

Česká zemědělská univerzita v Praze
Provozně ekonomická fakulta
Katedra informačního inženýrství



Diplomová práce

Multifunkční webový portál z oblasti kryptoměn

Jan Kalina

© 2023 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jan Kalina

Informatika

Název práce

Multifunkční webový portál z oblasti kryptoměn

Název anglicky

Multifunctional cryptocurrency portal

Cíle práce

Hlavním cílem práce je realizace návrhu webového portálu zaměřeného na kryptoměny.

Díličními cíli jsou:

- kryptokalkulačka – webová aplikace sloužící k přepočtu kryptoměn dle aktuálního i historického kurzu
- uživatelská sekce – umožňuje registrovanému uživateli vytváření přehledu dle vlastních preferencí
- informační sekce – slouží k přehlednému a ucelenému získání informací z oblasti kryptoměn, kladen důraz na bezpečnost

Závěrečným cílem práce je UX/UI návrh mobilní aplikace jakožto alternativy k webovému rozhraní.

Metodika

Teoretická část práce je založena na studiu odborné literatury a následování trendů ve vývoji webových stránek na internetu. Základem je literární rešerše zaměřená na tvorbu webových stránek.

Výstupem literární rešerše je soubor doporučení pro tvorbu webových stránek ve zvoleném prostředí. Což je dále ověřeno realizací funkčního webu. Na základě teoretické části a nezávislém hodnocení hotového řešení bude formulován závěr práce.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

Html a css/bootstrap, php/nette, ux/ui návrh, mysql, kryptoměny

Doporučené zdroje informací

BIERER, D. PHP 8 Programming Tips, Tricks and Best Practices. Birmingham, UK: Packt Publishing Ltd., 2021. ISBN 978-1-80107-187-1

CASTRO, E. – HYSLOP, B. HTML5 a CSS3. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8

JENSEN, S., J. The Missing Bootstrap 5 Guide. Birmingham, UK: Packt Publishing Ltd., 2021. ISBN 978-1-80107-643-2

NETTE.ORG. Dokumentace Nette. <https://doc.nette.org/>

VERONICA, A. The New 2022 UI/UX For Beginners And Experts. Amazon Digital Services LLC – Kdp, 2022. ISBN 979-84-900-2652-5

WELLING, L. – THOMSON, L. Mistrovství PHP a MySQL. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. David Buchtela, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 26. 3. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 28. 3. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 28. 03. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Multifunkční webový portál z oblasti kryptoměn" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2023

Poděkování

V této diplomové práci bych rád vyjádřil své upřímné poděkování Ing. Davidu Buchtelovi, Ph.D. za jeho vstřícný přístup, odborné rady a podporu při vypracování tohoto díla. Dále bych chtěl poděkovat svým kolegům a přátelům, zejména Bc. Dominiku Křížkovi a Ing. Martinu Stehlíkovi, za jejich cenné informace a nápady, které významně přispěly k úspěšné realizaci navrhovaného webu.

Multifunkční webový portál z oblasti kryptoměn

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací multifunkčního webového portálu zaměřeného na oblast kryptoměn prostřednictvím literární rešerše. Obsahuje také návrh grafického uživatelského rozhraní webu a jeho dynamických částí.

V teoretické části práce je prezentováno využití a potřeba webových technologií, včetně jejich historie a možného vývoje. Práce se dále zaměřuje na charakteristiku nástrojů a služeb, které se běžně využívají při vývoji webových stránek. Významná část práce je věnována oblasti kryptoměn a jejich roli v lidské společnosti.

V praktické části práce je popsán konkrétní návrh webového portálu, jehož hlavními součástmi jsou návrh uživatelského rozhraní, návrh aplikační části a postup možné realizace, který vychází z předpokladů uvedených v teoretické části. Tento obsah je doplněn vhodnými diagramy, obrázky, drátěnými modely, ukázkami vlastního návrhu a výsledky testování navrženého řešení.

Klíčová slova: Kryptoměny, html a css/bootstrap, php/nette, ux/ui návrh, mysql,

Multifunctional cryptocurrency portal

Abstract

This thesis contains a proposal for the implementation of a multifunctional web portal in the field of cryptocurrencies in the form of a literature search. It also includes a proposal for the graphical user interface of the website and its dynamic parts.

The theoretical part of the thesis describes the use and need of web technologies including their history and possible development. It also discusses the characteristics of the resources and services used in web development. A significant part of the thesis is devoted to the area of cryptocurrencies and their role in human society.

The practical part of the thesis describes the actual design of the web portal, whose main components include the design of the user interface, the design of the application part and the procedure of possible implementation, which is based on the assumptions described in the theoretical part. This content is complemented by appropriate diagrams, images, wireframes, examples of the actual design and the output of testing the actual design.

Keywords: Cryptocurrency, html a css/bootstrap, php/nette, design of the ux/ui, mysql

Obsah

1	Úvod	14
2	Cíl práce a metodika.....	15
2.1	Cíl práce	15
2.2	Metodika	15
3	Teoretická východiska.....	16
3.1	Internet a jeho vznik.....	16
3.2	World Wide Web	17
3.3	Přístup klient-server	18
3.3.1	Klient	19
3.3.1.1	Webové prohlížeče dnes	19
3.3.1.2	Mobilní zařízení	20
3.3.2	Server	21
3.3.3	Serverové prostředí pro web	22
3.3.4	Webový server	23
3.3.5	Hostování webových stránek	23
3.3.6	Hosting a freehosting	25
3.3.7	Protokoly.....	25
3.3.7.1	HTTP.....	27
3.3.7.2	FTP.....	28
3.4	Vývojové a uživatelská zkušenost	29
3.4.1	Vývojové fáze webu	29
3.4.2	UI/UX	30
3.4.3	Uživatelská zkušenost.....	31
3.4.4	Přístupnost	32
3.4.5	Responzivita	34
3.5	Nástroje a jazyky ke tvorbě webu	36
3.5.1	HTML	36
3.5.2	CSS	37
3.5.3	Frameworky	38
3.5.4	CSS frameworky.....	39
3.5.5	Bootstrap.....	40
3.5.6	Ukázky Bootstrapu	40
3.5.6.1	Navbar.....	40
3.5.6.2	Grid systém	41

3.5.6.3	Formuláře.....	42
3.5.7	Porovnání prostého CSS a Bootstrapu	43
3.5.7.1	Vytvoření tlačítka	43
3.5.7.2	Vytvoření sloupce.....	44
3.5.8	Javascript	45
3.5.9	jQuery	46
3.5.10	Node.js.....	47
3.5.11	Javascript frameworky	48
3.5.12	PHP.....	49
3.5.13	Nette	51
3.5.14	Porovnání prostého PHP a Nette	52
3.5.15	Další frameworky	53
3.5.16	Databáze	54
3.5.17	SQL	56
3.5.18	MySQL.....	56
3.5.19	SQL dotazy	57
3.5.20	SQL v PHP	58
3.5.21	SQL v Nette.....	61
3.5.22	Bezpečnost programování	64
3.5.23	API.....	66
3.5.24	Redakční systémy	67
3.5.25	Optimalizace pro vyhledávače	68
3.6	Kryptoměny	70
3.6.1	Principy fungování	71
3.6.2	Motivace k pořízení kryptoměn.....	72
3.6.3	Kritika.....	73
4	Vlastní práce	76
4.1	Frontend	79
4.1.1	Základní koncepce.....	79
4.1.1.1	Idea	79
4.1.1.2	Architektura	80
4.1.1.3	Informační architektura aplikace	80
4.1.2	Návrh uživatelského rozhraní.....	81
4.1.2.1	Desktop.....	81
4.1.2.2	Mobilní zařízení	85
4.1.3	Vývoj vlastního webu.....	87

4.1.3.1	Popis základní struktury.....	87
4.1.3.2	Popis vlastního HTML kódu.....	88
4.1.3.3	Ukázka vlastního návrhu.....	92
4.1.3.4	Popis vlastního Javascript kódu	97
4.1.4	Zpětná analýza	104
4.1.4.1	Použitelnost.....	104
4.1.4.2	Přístupnost.....	108
4.1.4.3	Rychlost	112
4.1.4.4	Validita kódu.....	113
4.2	Backend.....	115
4.2.1	Technologická architektura:	115
4.2.2	Architektura databáze	115
4.2.3	Popis používaných komponent	117
4.2.4	Implementace Nette	117
4.2.5	Připojení API Coingecko	118
4.2.6	Implementace skriptů.....	119
4.2.7	SEO strategie	126
4.3	Řízení projektu.....	128
4.3.1	Projektový plán.....	129
4.3.1.1	Důvody projektu	129
4.3.1.2	Cíle projektu.....	130
4.3.1.3	Předmětem projektu není	130
4.3.1.4	Očekávané přínosy projektu	130
4.3.1.5	Předpoklady realizace projektu.....	130
4.3.1.6	Průzkum	131
4.3.2	Bussines Model Canvas.....	131
4.3.3	Persony	134
4.3.3.1	Popis cílové skupiny	134
4.3.3.2	Ukázky person	136
4.3.4	UC scénáře.....	141
4.3.4.1	Aktéři	141
4.3.4.2	Případy užití	143
4.3.4.3	Scénáře.....	143
4.3.5	Harmonogram	146

4.3.6	Náklady projektu	150
4.3.7	Výběr webhostingu.....	154
4.3.8	Rizika.....	156
4.3.8.1	Identifikace rizik projektu.....	156
4.3.8.2	Návrh ošetření rizik projektu.....	157
5	Výsledky vlastní práce	158
6	Závěr	159
8	Seznam obrázků, tabulek, grafů a zkratek	164
8.1	Seznam obrázků.....	164
8.2	Seznam tabulek.....	165
8.3	Seznam grafů	165
8.4	Seznam použitých zkratek	165
9	Přílohy	166

1 Úvod

V současné době představují webové technologie, zejména webové stránky, klíčové prvky naší digitální společnosti. Tyto technologie umožňují vytváření dynamických a interaktivních webových stránek, což zlepšuje uživatelskou zkušenost a umožňuje větší interakci mezi uživatelem a webovou stránkou. Díky těmto technologiím se také zlepšila funkčnost a užitečnost webových stránek, což umožnilo vytváření nových aplikací a funkcí, jako jsou elektronické obchody, sociální sítě, online bankovníctví a mnoho dalších. Webové stránky se tak staly klíčovým nástrojem nejen pro firmy a organizace, ale i pro běžné jednotlivce. Potřeba webových technologií je v dnešní době nezbytná a stala se nedílnou součástí našich životů. Tento trend se zvýšil v posledních letech s rychlým rozvojem internetu a vývojem mobilních zařízení, což znamená, že vytváření, správa a rozvoj webových stránek je stále důležitější a potřebnější. Kromě webových stránek se práce zaměřuje i na kryptoměny, digitální aktiva fungující nezávisle na státních bankovních systémech, jejichž popularita a role v rámci naší společnosti neustále roste. Diplomová práce si klade za cíl přispět k rozvoji v této oblasti.

Hlavním cílem této práce je navrhnout postup realizace webového portálu, který nejen přispěje k osvětě v oblasti kryptoměn, ale také poskytne cenné informace, umožní uživatelům sledovat přehled svého portfolia a zajistí přehled o vývoji cen, včetně různých výpočtů či zobrazení v grafech. Využitá metodika v rámci této práce je založena na studiu odborné literatury a informací dostupných na renomovaných internetových stránkách. Nedílnou součástí práce je využití získaných poznatků při realizaci webu na základě vypracovaného návrhu.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem práce je realizace návrhu webového portálu zaměřeného na kryptoměny.

Dílními cíli jsou:

- Kryptokalkulačka – webová aplikace sloužící k přepočtu kryptoměn dle aktuálního i historického kurzu
- uživatelská sekce – umožňuje registrovanému uživateli vytváření přehledu dle vlastních preferencí
- informační sekce – slouží k přehlednému a ucelenému získání informací z oblasti kryptoměn, kladen důraz na bezpečnost

Závěrečným cílem práce je UX/UI návrh mobilní aplikace jakožto alternativy k webovému rozhraní.

2.2 Metodika

Teoretická část práce je založena na studiu odborné literatury a následování trendů ve vývoji webových stránek na internetu. Základem je literární rešerše zaměřená na tvorbu webových stránek.

Výstupem literární rešerše je soubor doporučení pro tvorbu webových stránek ve zvoleném prostředí. Což je dále ověřeno realizací funkčního webu. Na základě teoretické části a nezávislém hodnocení hotového řešení bude formulován závěr práce.

3 Teoretická východiska

3.1 Internet a jeho vznik

Internet představuje globální síť propojených počítačových sítí, která umožňuje sdílení informací a komunikaci mezi uživateli z celého světa. Počátky internetu sahají až do poloviny 20. století, kdy byly založeny první počítačové sítě.

V roce 1969 byl ve Spojených státech amerických založen ARPANET (Advanced Research Projects Agency Network) - první počítačová síť určená primárně pro vojenské a vědecké účely. Tato síť umožňovala sdílení informací mezi počítači umístěnými na různých univerzitách a výzkumných institucích.

Během 70. a 80. let se vyvíjely další počítačové sítě a protokoly, které daly za vznik současnému internetu, především pak TCP/IP (Transmission Control Protocol/Internet Protocol). V rámci této sítě bylo umožněno propojení různých počítačů a sítí po celém světě.

V roce 1989 navrhl britský vědec Tim Berners-Lee koncept World Wide Web (WWW) - hypertextový systém, který umožňoval propojování dokumentů na internetu prostřednictvím hypertextových odkazů. Tento systém se stal základem moderního internetu a umožnil vznik webových stránek.

V 90. letech se internet stával stále populárnějším a běžným nástrojem pro komunikaci, sdílení informací a nakupování. V této době vznikly také první vyhledávače, které umožňovaly uživatelům snadněji nalézt informace na internetu.

V průběhu následujících let se internet neustále rozšiřoval a stával se nedílnou součástí našeho každodenního života. V současnosti se stále více rozvíjejí různé technologie, jako jsou cloudová úložiště, umělá inteligence či internet věcí (IoT), které mění způsob, jakým využíváme internet [1].

3.2 World Wide Web

Úspěch WWW je založen na již zmíněném vynálezu hypertextu. První webová stránka byla vytvořena v roce 1991 a obsahovala informace o projektu World Wide Web. O rok později byl vytvořen první webový prohlížeč – WorldWideWeb, který umožňoval uživatelům procházet a zobrazovat webové stránky.

V roce 1993 byla vytvořena první verze webového prohlížeče Mosaic, který se rychle stal populárním a přispěl k rozšíření World Wide Webu mezi širokou veřejnost. V roce 1994 vznikla první webová kamera, která umožňovala živý přenos prostřednictvím internetu, a o rok později byla spuštěna první vyhledávací webová stránka – Yahoo!.

Mezi dalšími významnými událostmi v historii vývoje World Wide Webu lze zmínit například vznik JavaScriptu v roce 1995, který umožnil vývoj interaktivních webových stránek, nebo vznik webového prohlížeče Internet Explorer od společnosti Microsoft téhož roku, který se stal dominantním prohlížečem na trhu.

Stojí za zmínku také Netscape Navigator, jeden z prvních populárních webových prohlížečů, který byl vydán v roce 1994 společností Netscape Communications. Jako průkopník v oblasti webových prohlížečů hrál důležitou roli při popularizaci World Wide Webu. Netscape Navigator byl v té době konkurentem Internet Exploreru od Microsoftu.

Během let získával Netscape Navigator na popularitě a jeho verze 4.0 se stala v roce 1997 nejpoužívanější. Později se však začaly objevovat problémy s jeho vývojem a podporou, což vedlo k jeho postupnému úpadku. V roce 2002 byl Netscape Navigator nahrazen webovým prohlížečem Mozilla Firefox, který byl vyvinut na základě zdrojových kódů z Netscape Navigatoru.

Dnes již Netscape Navigator není v aktivním vývoji ani není používán. Nicméně jeho vliv na historii webových prohlížečů a rozvoj World Wide Webu je velký a nezpochybnitelný.

Od té doby neustále pokračoval rozvoj World Wide Webu, který se stal stále dostupnějším a snadno použitelným pro běžné uživatele. V roce 2004 byl uveden na trh webový prohlížeč Mozilla Firefox, který se stal významným konkurentem pro Internet Explorer, a v roce 2008 byl uveden mobilní prohlížeč Safari pro zařízení od společnosti Apple. V roce 2010 byl spuštěn webový prohlížeč Google Chrome, který rychle získal na popularitě a dnes patří mezi nejčastěji používané prohlížeče.

Vývoj World Wide Webu pokračoval a dnes je plně integrován do našich životů. Umožňuje nám komunikovat s lidmi z celého světa, získávat informace na nejrůznější témata, nakupovat a prodávat zboží, sledovat filmy, poslouchat hudbu a mnoho dalšího. WWW se stal základním stavebním kamenem moderní společnosti a výrazně ovlivnil způsob, jakým fungujeme jako celek [2].

3.3 Přístup klient-server

Obecně lze říci, že model klient-server představuje způsob komunikace mezi dvěma nebo více počítači v síti. V tomto modelu jeden počítač (server) poskytuje určité služby, zatímco ostatní počítače (klienti) tyto služby využívají.

V rámci modelu klient-server jsou role klienta a serveru striktně rozděleny. Server poskytuje služby, jako například sdílení dat, správu uživatelů, databáze a další. Na druhou stranu klienti tyto služby využívají. Například u webových stránek je klientem internetový prohlížeč, který zobrazuje obsah webových stránek, zatímco server poskytuje obsah těchto stránek.

V oblasti webových technologií se model klient-server často používá v souvislosti s protokolem HTTP (Hypertext Transfer Protocol), který umožňuje klientům (prohlížečům) komunikovat se servery a získávat informace. Klient (prohlížeč) pošle serveru (webovému serveru) požadavek (request) na získání určitého obsahu (např. webové stránky), načež server odpoví (response) odesláním požadovaného obsahu zpět klientovi.

Model klient-server se mimo jiné používá i v souvislosti s databázovými systémy, kde server (databázový server) poskytuje služby pro ukládání a zpracování dat, zatímco klienti (aplikace) tyto služby využívají pro ukládání a zpracování dat ve svých aplikacích.

Lze tedy konstatovat, že přístup klient-server je zásadním modelem komunikace v oblasti informačních technologií a je široce využíván v různých oblastech, jako jsou webové stránky, databázové systémy, cloudové služby a mnoho dalších [3] [4].

3.3.1 Klient

V roli klienta rozumíme především přístup k serverům prostřednictvím klientských aplikací, které odesílají požadavky směrem k serveru a následně obdrží odpověď ve formě dat, která jsou zpracována nebo určena ke zpracování v klientské aplikaci. V kontextu WWW tuto roli standardně zastávají webové prohlížeče.

3.3.1.1 Webové prohlížeče dnes

V současné době existuje mnoho webových prohlížečů, které se liší svou popularitou a funkcemi. Mezi nejznámější webové prohlížeče patří:

Google Chrome patří mezi nejpopulárnější webové prohlížeče na světě. Tento prohlížeč se vyznačuje rychlostí a stabilitou a poskytuje řadu užitečných funkcí. Mezi ně patří například synchronizace mezi různými zařízeními, vestavěný překladač či možnost použití rozšíření pro zlepšení uživatelského zážitku.

Mozilla Firefox představuje další oblíbený webový prohlížeč, který si získal pověst díky své rychlosti a bezpečnosti. Tento prohlížeč disponuje širokou škálou funkcí, mezi něž patří blokování reklam a sledování, vestavěný správce hesel či podpora pro rozšíření.

Microsoft Edge je webový prohlížeč vyvinutý společností Microsoft, který nahradil původní Internet Explorer. Edge je postaven na jádře Chromium, stejně jako Google Chrome, což mu poskytuje rychlost a stabilitu. Navíc nabízí funkce jako integraci s ekosystémem Microsoft, zabezpečení a možnost použití rozšíření.

Safari je webový prohlížeč vyvinutý společností Apple, který je určen primárně pro zařízení s operačním systémem macOS a iOS. Safari je zaměřen na rychlost, bezpečnost a soukromí a nabízí funkce, jako jsou inteligentní vyhledávání, blokování sledování a synchronizace mezi Apple zařízeními.

Opera je další webový prohlížeč, který je známý svou rychlostí a inovacemi. Nabízí funkce, jako jsou integrovaný blokátor reklam, bezplatná neomezená VPN služba a Turbo režim pro rychlejší načítání stránek na pomalejších připojeních.

Brave je relativně nový webový prohlížeč zaměřený na soukromí a bezpečnost uživatelů. Brave blokuje reklamy a sledování ve výchozím nastavení, což zajišťuje lepší soukromí. Navíc nabízí systém odměn pro uživatele a tvůrce obsahu založený na kryptoměně Basic Attention Token (BAT).

Tyto webové prohlížeče představují pouze některé z nejznámějších a nejpopulárnějších prohlížečů na trhu, ale existuje mnoho dalších, které se zaměřují na různé potřeby a preference uživatelů. Například Google Chrome se zaměřuje na rychlost a nabízí mnoho užitečných funkcí pro uživatele, jako je integrace s Google účtem. Mozilla Firefox se zaměřuje na bezpečnost a soukromí uživatele a nabízí mnoho užitečných funkcí pro blokování sledování a reklam. Microsoft Edge se zaměřuje na integraci s dalšími Microsoft produkty, jako je například operační systém Windows, Windows Office 365 a služby vyhledávače Bing. Apple Safari nabízí vynikající integraci s dalšími Apple produkty a podporu pro například Apple Pay či autentifikaci prostřednictvím Face ID případně Touch ID [5].

3.3.1.2 Mobilní zařízení

World Wide Web a mobilní zařízení se v průběhu času staly dvěma úzce propojenými odvětvími. Na mobilních platformách lze mezi nejpopulárnější webové prohlížeče zařadit následující:

- Google Chrome – Google Chrome je jedním z nejoblíbenějších prohlížečů na mobilních zařízeních s operačními systémy Android i iOS. Tento prohlížeč nabízí rychlé načítání stránek, intuitivní uživatelské rozhraní a synchronizaci s desktopovou verzí prohlížeče.
- Safari – Safari je standardní webový prohlížeč v operačním systému iOS od společnosti Apple. Tento prohlížeč nabízí rychlé vyhledávání a integraci s dalšími aplikacemi v systému.
- Firefox – Na mobilních zařízeních je k dispozici také verze tohoto oblíbeného open-source prohlížeče, Firefox. Nabízí řadu funkcí, jako je blokování reklam či ochrana soukromí.
- Opera – Opera Mobile přináší podobné funkce jako desktopová verze prohlížeče, včetně rychlého načítání stránek a blokování reklam.

Je však třeba poznamenat, že v závislosti na konkrétním regionu a používaném zařízení se mohou preference lišit [6].

3.3.2 Server

- Server je počítačový systém, který poskytuje určité služby pro klienty. Existuje několik různých typů serverů, které se mohou lišit v závislosti na použité technologii a architektuře. Mezi běžné formy serverů patří:
- Fyzický server je počítačový systém, který běží přímo na fyzickém hardwaru a poskytuje různé služby pro klienty.
- Virtualizovaný server je virtuální počítač, který běží na fyzickém hardwaru. Virtualizace umožňuje vytvoření několika virtuálních strojů na jednom fyzickém stroji a zajišťuje efektivní využití hardwarových zdrojů.

- Cloudový server je virtuální počítač, který běží na infrastruktuře cloudového poskytovatele. Cloudový server umožňuje uživatelům pronajmout si virtuální prostředí pro své webové stránky a další aplikace.

Každý typ serveru má své výhody a nevýhody, volba konkrétního typu závisí na potřebách a požadavcích správce a zejména pak na typu projektu, ke kterému má být určen.

3.3.3 Serverové prostředí pro web

Servery lze rozlišovat také podle jejich účelu, aniž bychom brali v úvahu, na jakém hardwaru či platformě jsou provozovány.

Webový server standardně poskytuje pouze obsah webových stránek v podobě HTML souborů, obrázků, CSS stylů a umožňuje běh aplikací na straně serveru. Obecně však bývá tento pojem užíván jako zastřešující pro celou řadu dalších serverů, jenž zaujímají svoji dílčí a zcela odlišnou roli.

Jedním z takových serverů je databázový server, který slouží k ukládání a správě dat, jako jsou uživatelská jména a hesla, informace o produktech nebo zákaznících. Databázové servery mohou být součástí webového serveru nebo být provozovány odděleně.

Dalším důležitým serverem pro webové stránky je aplikační server, například PHP. Tyto servery umožňují spouštění aplikací a skriptů na webovém serveru, což umožňuje větší flexibilitu a výkon pro webové stránky, ale zejména rozšiřuje jejich funkcionalitu. Aplikační servery mohou být součástí webového serveru nebo mohou být rovněž provozovány na jiném serveru.

Úložiště typu FTP (File Transfer Protocol) slouží k přenosu souborů mezi počítači a mohou být použity k uložení souborů, jako jsou obrázky, videa a další multimediální soubory, které jsou potřebné pro webové stránky.

Všechny tyto servery jsou důležité pro provoz plnohodnotných a moderních webových stránek. Je důležité zajistit, aby byly tyto servery správně nakonfigurovány, spravovány, aktualizovány a aby zajistily co nejlepší výkon a stabilitu pro webové stránky [7] [8].

3.3.4 Webový server

Jako konkrétní příklad webového serveru byl vybrán Apache HTTP Server (známý také jako Apache) jedná se o open source webový server, který poskytuje služby pro zpřístupňování webových stránek přes internet. Byl vyvinut v roce 1995 skupinou studentů na University of Illinois a od té doby se stal nejpoblárnějším webovým serverem na světě.

Apache je multiplatformní webový server, který běží na operačních systémech jako jsou Linux, Windows, Unix a další. Podporuje mnoho jazyků pro tvorbu dynamických webových stránek, jako jsou PHP, Python, Perl, Ruby a další.

Apache poskytuje řadu funkcí, jako jsou podpora pro SSL/TLS šifrování, správu virtuálních hostů, podporu pro autentizaci a autorizaci uživatelů a další. Díky těmto funkcím se Apache stal nejvíce používaným webovým serverem na světě.

Apache se používá v široké škále aplikací, jako jsou webové stránky, e-commerce aplikace, online zpravodajství, intranety a další. Díky jeho popularitě existuje mnoho podpůrných technologií a nástrojů, jako jsou moduly, které přidávají další funkce, a také různé nástroje pro monitorování a správu Apache serveru.

Apache je open source projekt, což znamená, že jeho zdrojový kód je dostupný a může být upraven a přizpůsoben potřebám konkrétního uživatele. To také znamená, že Apache je zdarma k použití a může být upravován bez nutnosti platit licenční poplatky [9].

3.3.5 Hostování webových stránek

Existuje několik způsobů, jak hostovat webové stránky, mezi ty nejčastější patří:

Hosting – Toto je nejčastější způsob hostování webových stránek. Zákazník si pronajme prostor na serveru poskytovatele webhostingu, který poskytuje webový server,

databázi a další služby potřebné pro provoz webových stránek. Tento způsob je vhodný pro malé a střední webové stránky, protože nevyžaduje velkou investici do infrastruktury a má nízké náklady na údržbu a provoz.

VPS (Virtual Private Server) – Tento způsob hostování poskytuje virtuální prostředí na sdíleném fyzickém serveru. Každý zákazník má vlastní prostor na disku a přístup k operačnímu systému, což umožňuje větší flexibilitu a možnost přizpůsobit si prostředí svým potřebám. VPS je vhodný pro středně velké webové stránky s většími nároky na výkon a specifické požadavky.

Dedikovaný server – Tento způsob hostování znamená pronájem celého fyzického serveru pro provoz webových stránek. Tento způsob je vhodný pro velké webové stránky s velkým provozem a vysokými nároky na výkon.

Cloudový hosting – Tento způsob hostování poskytuje virtuální prostředí v cloudu. Webové stránky jsou hostovány na virtuálních serverech, které jsou škálovatelné a mohou být snadno přizpůsobeny měnícím se potřebám. Cloudový hosting je vhodný pro velké a střední webové stránky s vysokou návštěvností a měnícími se nároky na výkon.

Domácí server – Tento způsob hostování znamená provozování webového serveru na vlastním počítači v domácím prostředí. Tento způsob je vhodný pro menší webové stránky nebo osobní projekty, ale vyžaduje vysokou úroveň technických znalostí a bezpečnostní opatření.

Každý z těchto způsobů má své výhody a nevýhody a volba závisí na konkrétních potřebách webových stránek a jejich provozovatele.

3.3.6 Hosting a freehosting

Hosting je služba, která umožňuje uživatelům publikovat své webové stránky na internetu. Existuje několik druhů hostingu, které se liší podle použité technologie, cenové politiky, úrovně poskytované podpory a dalších zásadních faktorů.

Sdílený hosting představuje nejrozšířenější typ hostingu, v němž si několik uživatelů sdílí jeden fyzický server. Tento typ hostingu bývá obvykle cenově dostupnější než VPS nebo dedikované servery, avšak je omezen v počtu souborů, velikosti úložiště, databázích, datové přenosové rychlosti a konfiguračních možnostech. Sdílený hosting je ideální pro menší a střední projekty, které nevyžadují velké množství datového prostoru nebo vysokou rychlost přenosu dat.

Freehosting poskytuje variantu hostingových služeb, které jsou obvykle zcela zdarma. Tyto služby stojí na stejném principu jako placené hostingy, ale jsou vhodné pro menší projekty a testování. Freehosting může mít omezené funkce a podporu, tudíž není vhodný pro větší projekty ani pro komerční využití. Standardní praxí je nucené vkládání reklam do webového obsahu, což finančně podporuje poskytování bezplatného hostování.

Při porovnání hostingových služeb je třeba zohlednit řadu faktorů, jako je cena, prostor na serveru, množství uvažovaných souborů, funkce, podpora a další. Zájemce by měl pečlivě zvážit své potřeby a vybrat hostingovou službu, která danému projektu nejlépe vyhovuje.

3.3.7 Protokoly

Následující část práce je věnována obecnému popisu protokolů, s nimiž je reálné setkat se během práce se serverem i v roli uživatele. Detailněji budou probrány protokoly TCP/IP, HTTP a FTP [10].

- TCP/IP – Rodina protokolů, které umožňují přenos dat v počítačových sítích.
- HTTP (Hypertext Transfer Protocol) – Protokol, který se používá pro přenos hypertextových dokumentů na webu.

- HTTPS (Hypertext Transfer Protocol Secure) – Bezpečná verze protokolu HTTP, která zajišťuje šifrovaný přenos dat mezi prohlížečem a webovým serverem.
- FTP (File Transfer Protocol) – Protokol, který se používá pro přenos souborů mezi počítači v síti.
- SSH (Secure Shell) – Zabezpečený protokol pro vzdálený přístup a ovládání příkazové řádky počítače.
- Telnet – Protokol pro vzdálený přístup a ovládání příkazové řádky počítače.
- SMTP (Simple Mail Transfer Protocol) – Protokol pro přenos elektronické pošty mezi mailovými servery.
- POP3 (Post Office Protocol version 3) – Protokol pro stažení elektronické pošty z mailového serveru na lokální počítač.
- IMAP (Internet Message Access Protocol) – Protokol pro správu elektronické pošty na mailovém serveru a umožňuje uživatelům přistupovat ke své poště z různých zařízení.
- DNS (Domain Name System) – Protokol pro překlad doménových jmen na IP adresy.
- DHCP (Dynamic Host Configuration Protocol) – Protokol pro automatické přidělování IP adres počítačům v síti.
- SNMP (Simple Network Management Protocol) – Protokol pro správu počítačových sítí a připojených zařízení.
- SIP (Session Initiation Protocol) – Protokol pro správu a řízení hlasového a videohovoru v IP sítích.
- FTPS (File Transfer Protocol Secure) – Zabezpečená verze protokolu FTP, která používá šifrování pro ochranu přenosu souborů mezi počítači.
- DNSSEC (Domain Name System Security Extensions) – Rozšíření protokolu DNS, které zajišťuje bezpečnost překladu doménových jmen na IP adresy.

3.3.7.1 HTTP

HTTP (Hypertext Transfer Protocol) je protokol, který se používá pro komunikaci mezi webovými servery a prohlížeči. Jeho hlavní funkcí je zajištění přenosu informací mezi klientem a serverem ve formě požadavků a odpovědí.

HTTP požadavek je zpráva, kterou klient (např. webový prohlížeč) odesílá serveru s cílem získat nějaký zdroj (např. webovou stránku). Tento požadavek se skládá z několika částí, jako jsou metoda (GET, POST, PUT, DELETE atd.), URI (Uniform Resource Identifier), HTTP verze, hlavičky požadavku a tělo požadavku (v případě metody POST).

HTTP odpověď je zpráva, kterou server odesílá klientovi jako odpověď na požadavek. Odpověď se skládá z několika částí, jako jsou HTTP verze, stavový kód, hlavičky odpovědi a tělo odpovědi.

Stavový kód je třímístný kód, který určuje stav odpovědi na HTTP požadavek. Tyto kódy jsou rozděleny do pěti tříd podle první číslice kódu:

- 1xx (Informational): Tato třída stavových kódů se používá pro informační zprávy, např. 100 (Continue), který znamená, že server pokračuje v přenosu dat.
- 2xx (Successful): Tato třída stavových kódů se používá pro úspěšné požadavky, např. 200 (OK), který znamená, že požadavek byl úspěšně zpracován.
- 3xx (Redirection): Tato třída stavových kódů se používá pro přesměrování, např. 301 (Moved Permanently), který znamená, že zdroj byl trvale přesunut na jinou adresu.
- 4xx (Client Error): Tato třída stavových kódů se používá pro chyby způsobené klientem, např. 404 (Not Found), který znamená, že požadovaný zdroj nebyl nalezen.

- 5xx (Server Error): Tato třída stavových kódů se používá pro chyby způsobené serverem, např. 500 (Internal Server Error), který znamená, že došlo k chybě na serveru.

Tyto HTTP stavy jsou důležité, protože umožňují rychlé zjištění stavu požadavku a umožňují rychlé opravy chyb, když se něco nepodaří [11].

3.3.7.2 FTP

FTP (File Transfer Protocol) je protokol pro přenos souborů mezi počítači připojenými k internetu. Tento protokol umožňuje uživatelům efektivně a bezpečně přenášet soubory mezi počítači pomocí internetového připojení a umožňuje jim pracovat se soubory na vzdáleném serveru.

FTP protokol se skládá ze dvou základních částí, a to řídicího kanálu a datového kanálu. Řídicí kanál slouží k ovládání spojení a výměně informací mezi klientem a serverem. Na druhé straně datový kanál se používá pro přenos souborů mezi klientem a serverem.

FTP definuje několik základních operací, které mohou být prováděny jak klientem, tak serverem. Tyto operace zahrnují:

- Přihlášení na server s uživatelským jménem a heslem
- Zobrazení seznamu souborů na serveru
- Přenos souborů mezi klientem a serverem
- Vytvoření nové složky na serveru
- Odstranění souborů nebo složek na serveru
- Přejmenování souborů nebo složek na serveru

Podobně jako u protokolu HTTP, existují i v protokolu FTP různé stavy, které signalizují různé situace, jež se mohou vyskytnout během přenosu dat. Některé z nejčastěji používaných FTP stavů jsou:

- 150 Opening data connection: Server oznámí klientovi, že je připraven na přenos dat.
- 226 Closing data connection: Server oznámí klientovi, že byl dokončen přenos dat a že spojení bylo uzavřeno.
- 331 User name okay, need password: Server požaduje od klienta zadání hesla pro přihlášení na server.
- 550 Requested action not taken: Server oznamuje klientovi, že požadovaná operace se nepodařila, například kvůli nedostatečným oprávněním.

Tyto stavy jsou důležité pro uživatele FTP, aby mohli pochopit, co se děje během přenosu dat a jaké jsou případné problémy [12].

3.4 Vývojové a uživatelská zkušenost

3.4.1 Vývojové fáze webu

Vývoj webu se skládá z několika fází, z nichž každá má své specifické cíle a úkoly [13].

- Analýza a plánování – v této fázi se shromažďují potřeby a požadavky zákazníka. Zároveň se stanoví cíle webu a formuluje se strategie vývoje. Do této fáze mohou patřit i první náčrtky nebo ideje ohledně designu webu, ale ještě nejsou v této fázi detailnější designové prvky použity. Jedním z nástrojů, který se může použít v této fázi, jsou i wireframy. Pomocí wireframů lze zobrazit hrubší strukturu webové stránky na základě cílů a potřeb uživatelů.
- Návrh – na základě výstupů z první fáze se vytváří důkladnější návrh vizuálního stylu webu a rozhraní. V této fázi se již věnuje pozornost detailům jako jsou typografie, barvy a typy grafiky. Wireframy mohou být opět použity jako nástroj k vizualizaci struktury a umístění jednotlivých prvků webové stránky.

- Vývoj – při této fázi se již vytváří samotný web včetně funkcí, logiky a databáze. Na základě hotového návrhu se tvoří konkrétní stránky a ovládací prvky, výstupem je prototyp určený k testování.
- Testování a opravy – po dokončení vývoje se provádí testování webu, aby se zkontrolovaly funkce a rozhraní. Jestliže jsou identifikovány problémy nebo chyby, je potřeba je opravit, než dojde k nasazení webu.
- Nasazení – pokud je web úspěšně otestován a schválen, je nasazen na internetový server. Tato fáze také může zahrnovat vytváření architektury webu, výběr hostingových řešení, optimalizace rychlosti webu a další.
- Údržba – v této poslední fázi se web průběžně spravuje a aktualizuje. Může jít o aktualizace designu nebo jiných funkčních prvků s cílem zajistit optimální provoz webu.

3.4.2 UI/UX

UI (User Interface) a UX (User Experience) jsou dva významné pojmy v oblasti designu webových stránek, mobilních aplikací a dalších digitálních projektů.

UI se zaměřuje na vývoj vizuálního designu, tedy jak se grafické prvky a informace na stránce organizují, aby byly pro uživatele co nejpřehlednější a atraktivní. UI designer se soustředí na práci s barvami, typografií, ikonami, widgety a dalšími grafickými prvky, které tvoří vzhled produktu.

UX naopak řeší celkovou uživatelskou zkušenost. UX designer pracuje na tom, aby uživatelé ocenili produkt a aby jim bylo plnění jejich potřeb co nejvíce usnadněno. UX designer se tedy soustředí na to, jak uživatelé používají produkt, zda snadno najdou informace, zda je produkt intuitivní a zda jsou s jeho používáním spokojeni.

UI a UX jsou dvě velmi důležité složky, které úzce spolupracují. Společnými silami vzniká funkční a esteticky příjemný digitální produkt, který zajišťuje co největší uživatelskou spokojenost [13].

3.4.3 Uživatelská zkušenost

Uživatelská zkušenost lze vyjádřit různými parametry, které určují úspěšnost webu. Mezi ně se řadí i to, zda mají uživatelé tendenci se na web vracet. Existuje celá řada způsobů, jak zlepšit uživatelskou zkušenost a získat tak uživatele, kteří se budou na web rádi vracet. Mezi ty hlavní patří [13]:

- Umožnit snadnou navigaci – Uživatelé by měli být schopni snadno nalézt to, co hledají. Proto je vhodné strukturu webu rozdělit do logických kategorií a zpřístupnit ji uživatelům pomocí jednoduché a intuitivní navigace.
- Zvýšit rychlost a výkon – Web s rychlým načítáním a vysokým výkonem přináší uživatelům pozitivní zážitek. Je důležité mít web s responzivním designem, který se přizpůsobí různým zařízením a rychle načte stránky.
- Vybudovat důvěru – Použít certifikáty zabezpečení, vybrat kvalitní hosting, zobrazit zpětnou vazbu od ostatních uživatelů, poskytnout kontaktní informace na podporu a zobrazit obchodní podmínky a další informace. Tímto vložíte do webu důvěru, což bude mít pozitivní vliv na zákaznickou důvěru.
- Sledovat uživatelské chování – Sledování uživatelského chování pomáhá pochopit, kde se uživatelé zastavují, co hledají a kde mohou být problémy. Tuto funkci lze například využít prostřednictvím Google Analytics.
- Vytvořit prvotřídní obsah – Zajistit, aby byl obsah vašeho webu zajímavý, přesný, relevantní a snadno čitelný. Pokud máte například blogovou sekci, vybírat témata, která budou čtenáře zajímat a obohacovat je novými informacemi.

- Zlepšit mobilní zážitek – Mobilní zařízení jsou stále více používána pro prohlížení webu. Proto je důležité, aby byl váš web responzivní a funkční pro mobilní zařízení.

3.4.4 Přístupnost

Přístupnost webu (v anglické terminologii „web accessibility“) zahrnuje navrhování webových stránek a aplikací tak, aby byly přístupné pro všechny uživatele včetně osob s různými typy postižení nebo omezením, jako jsou například zrakové, sluchové, motorické a kognitivní potíže. Tento koncept přístupnosti obnáší vytváření webových stránek a aplikací, které jsou snadno použitelné pro osoby se zdravotním postižením a zároveň dodržování mezinárodních standardů pro přístupnost webu.

Vývoj a návrh webových stránek musí být přizpůsoben tak, aby zohledňoval nároky na přístupnost již v průběhu celého procesu tvorby a provozu webové stránky. To zahrnuje návrh uživatelského rozhraní, výběr barev, editaci obsahu a testování uživatelského rozhraní. Důležité prvky přístupnosti webu zahrnují:

- Obsah webové stránky by měl být napsán jednoduchým a srozumitelným jazykem. Dále vyhnout se složitým slovům, frázím a větám a používat jasné a konkrétní formulace namísto záhadných nebo technických termínů.
- Použity by měly být fonty s dobrou čitelností a kontrastem. Například sans-serifové fonty jsou na webových stránkách obecně čitelnější než serifové fonty. Kontrast mezi pozadím a textem by měl být dostatečný, aby byl text snadno čitelný pro uživatele se zrakovými problémy.
- Uživatelům by mělo být umožněno měnit velikost textu pro zvýšení čitelnosti a srozumitelnosti. To lze docílit použitím relativních jednotek místo pevných hodnot.

- Vždy by měl být k dispozici popis a alternativní text pro všechny obrázky, aby mohly být snadno chápány uživateli se zrakovými problémy.
- Použití odpovídajících tagů HTML, aby bylo možné snadno určit popisky, titulky a nadpisy. To umožňuje uživatelům se zrakovými problémy prohlížet obsah stránky a obsahovat informace v logickém pořadí.
- Umožnit uživatelům procházet web použitím klávesových zkratk, například "Tab" pro přechod na další ovládací prvek a "Enter" pro jeho aktivaci.
- Testovat stránky na přístupnost pomocí nástrojů k testování přístupnosti, jako je například nástroj WebAIM Accessibility Checker. Tyto nástroje pomohou identifikovat problematické oblasti webové stránky a poskytnou určitá doporučení ke zlepšení.

Webová stránka s vysokou přístupností umožňuje všem uživatelům, bez ohledu na jejich individuální schopnosti či omezení, efektivní přístup k informacím a službám dostupným na internetu. Tímto způsobem se zajišťuje širší dostupnost a usnadňuje orientace na webových stránkách pro všechny uživatele.

Zde jsou některé ukázky HTML kódu pro použití popisků, titulků a dalších prvků, které zlepšují přístupnost:

Použití popisků u obrázku:

```

```

Použití titulků:

```
<h1>Titulek stránky</h1>
<h2>Podtitulek</h2>
```

Použití popisků k aktivním odkazům:

```
<a href="odkaz.html" title="Popisek odkazu">Text odkazu</a>
```

Atribut `aria-label` pro prvky, které nemají textový popisek:

```
<button type="submit" aria-label="Odeslat formulář">Odeslat</button>
```

Použití klávesových zkratk pro přístup k odkazu:

```
<a href="odkaz.html" accesskey="n">Název odkazu</a>
```

Atribut `tabindex` pro přizpůsobení pořadí klávesnicové navigace:

```
<input type="text" tabindex="2">  
<input type="submit" tabindex="3">  
<input type="reset" tabindex="4">
```

Tyto příklady demonstrují, jak lze prostřednictvím jednoduchých HTML prvků vytvořit více přístupné webové stránky pro uživatele s různými druhy zdravotních omezení. Během procesu tvorby webových stránek je důležité brát v úvahu tyto prvky, aby byl výsledný obsah vhodný pro všechny uživatele [14].

3.4.5 Responzivita

Responzivita je vlastnost webové stránky, která jim umožňuje přizpůsobit se různým velikostem obrazovek a zařízení, na kterých jsou zobrazeny. Cílem responzivního designu je zajistit optimální zobrazení a použitelnost stránky napříč širokou škálou zařízení, jako jsou stolní počítače, notebooky, tablety a chytré telefony.

Responzivní webové stránky automaticky upravují velikost textu, ovládacích prvků, obrázků a dalších komponent podle velikosti obrazovky uživatele, aby bylo zobrazení co nejjasnější a nejsnazší k použití. Je-li responzivita při vývoji webových stránek brána v úvahu, ovlivňuje samotný vývoj těmito způsoby:

Při navrhování responzivního webu je nezbytné zohlednit rozmanité velikosti obrazovek a zařízení. Designéři musí vytvářet flexibilní rozvržení, jež se přizpůsobí různým podmínkám. To zahrnuje vytvoření několika verzí designu pro rozličná zařízení a rozlišení.

Pro realizaci responzivního webu je nutné využít technologie, které umožňují flexibilní rozvržení a adaptivní prvky. Mezi běžně používané technologie patří HTML5, CSS3 a JavaScript. CSS media queries mají zvláštní význam, neboť umožňují vývojářům definovat různé styly pro odlišné velikosti obrazovek a zařízení.

Testování responzivního webu je klíčové pro zajištění jeho správné funkčnosti na různých zařízeních a obrazovkách. Vývojáři musí provádět průběžné testy na rozličných zařízeních, prohlížečích a rozlišeních, aby zajistili kvalitu a funkčnost stránky.

Optimalizace responzivního webu zahrnuje zajištění rychlého načítání a efektivního využití zdrojů. Je nutné provádět optimalizaci obrázků, minimalizovat velikost souborů CSS a JavaScript, a využívat techniky jako je lazy loading (načítání obsahu až při jeho potřebě).

Responzivní web vyžaduje pravidelnou údržbu a aktualizace, aby zůstal kompatibilní s novými zařízeními, prohlížeči a technologiemi. Vývojáři musí sledovat trendy a inovace v oblasti responzivního designu a průběžně aktualizovat a vylepšovat své webové stránky.

Responzivita má také významný dopad na SEO (optimalizace pro vyhledávače), jelikož vyhledávače upřednostňují stránky, které jsou responzivní. Z toho důvodu je velmi důležité, aby webová stránka byla optimalizována pro různá zařízení, jinak by mohla výrazně utrpět v oblasti SEO [15].

3.5 Nástroje a jazyky ke tvorbě webu

3.5.1 HTML

HTML (Hypertext Markup Language) je základní jazyk používaný pro tvorbu webových stránek. Od svého vzniku v roce 1991 prošel mnoha změnami a aktualizacemi, které přidávaly nové funkce a možnosti.

Přímým předchůdcem HTML byl standard SGML (Standard Generalized Markup Language), který byl vytvořen v roce 1986. SGML byl založen na myšlence oddělit strukturu dokumentu od jeho prezentace a popisovat strukturu pomocí značek, které definují jednotlivé části dokumentu.

HTML bylo poté vytvořeno Timem Bernersem-Leem jako zjednodušená a omezená verze SGML pro popisování struktury webových stránek. HTML 1.0 bylo poměrně jednoduché a obsahovalo pouze několik základních tagů, jako například `<head>`, `<title>` a `<body>`. Postupně se však HTML rozvíjelo a přibývaly mu nové funkce a značky.

HTML 2.0, první standardizovaná verze, byla uvedena v roce 1995 a zahrnovala nové prvky jako tabulky a formuláře. HTML 3.0, uvedený v roce 1996, přinesl další inovace, včetně rámců, kaskádových stylů (CSS) a skriptování v JavaScriptu.

HTML 4.0, která byla uvedena v roce 1997, byla první verzí plně podporovanou všemi hlavními prohlížeči. Představila nové prvky, jako jsou podpora vektorové grafiky, vylepšené styly a formuláře a nové prvky pro strukturování dokumentů.

HTML 5, uvedená v roce 2014, přinesla řadu nových funkcí, mezi nimiž patří značky pro přehrávání videa a audia, možnost offline ukládání webových stránek, pokročilé zpracování formulářů a další.

Kromě výše uvedených verzí byly v určitém období používány i verze HTML 2.1, 3.2 a XHTML. XHTML byl vytvořen jako nástupce HTML 4.0 a navržen jako kombinace

HTML a XML. Cílem bylo přinést více struktury do webových stránek a zlepšit kompatibilitu mezi různými prohlížeči.

V současnosti je HTML 5 stále nejnovější verzí a je široce používán pro tvorbu moderních webových stránek. Jeho přidané funkce, jako jsou geolokace, pokročilé zpracování formulářů a offline ukládání, poskytují webovým vývojářům více možností pro tvorbu interaktivních a uživatelsky přívětivých stránek.

Zajímavostí může být také fakt, že HTML byl původně vytvořen jako jednoduchý způsob sdílení dokumentů mezi vědci na CERNu, aniž by si kdokoli uvědomil obrovský dopad, který bude mít na celý svět a vývoj internetu [16] [17] [18].

3.5.2 CSS

CSS (Cascading Style Sheets) je technologie, která umožňuje oddělit vzhled webové stránky od jejího obsahu. Byla poprvé představena v roce 1996 a od té doby prošla mnoha změnami a vylepšeními. CSS bylo vytvořeno s cílem umožnit tvůrcům webových stránek lepší kontrolu nad prezentací obsahu a snadnější úpravu vzhledu celé webové stránky oproti složité úpravě jednotlivých tagů v rámci HTML.

První verze CSS byla velmi jednoduchá a obsahovala pouze základní vlastnosti jako barvu pozadí, barvu textu, velikost písma a ohraničení. Další verze postupně přidávaly nové vlastnosti, jako jsou stíny, přechody, animace, responzivní design a další.

CSS 2 bylo vydáno v roce 1998 a přineslo mnoho nových vlastností jako například podpora pro vrstvy, způsob řazení elementů nebo podpora pro tisk. CSS 2.1 bylo vydáno v roce 2004 a přineslo vylepšení a opravy chyb z předchozí verze.

V roce 2011 bylo vydáno CSS 3, které přineslo ještě více nových vlastností, jako jsou například nové selektory, podpora pro více sloupců, flexibilní design, transformace, přechody, animace a mnoho dalšího. CSS 3 se skládá z mnoha modulů, každý z nich obsahuje specifické vlastnosti. Tím se zajistí, že nové vlastnosti se mohou přidávat, aniž by byla narušena zpětná kompatibilita.

CSS je v současné době klíčovou technologií v oblasti tvorby webových stránek. Umožňuje tvůrcům oddělit prezentaci od obsahu, což zjednodušuje úpravy vizuálního vzhledu stránek a zlepšuje rychlost a efektivitu vývoje. Díky CSS mohou být webové stránky esteticky přitažlivější, přístupnější a uživatelsky přívětivější.

V posledních letech se jedním z významných trendů stalo využití CSS frameworků, které nabízejí hotová řešení pro běžné úkoly, jako je responzivní design, základní typografie a další aspekty. Tímto způsobem mohou tvůrci webových stránek pracovat rychleji a efektivněji.

V současnosti je CSS standardizováno a spravováno organizací World Wide Web Consortium (W3C).

Je také důležité zmínit, že v nedávné době se vývoj CSS posunul směrem k tzv. "CSS-in-JS", což je přístup, při kterém se styly definují přímo v kódu JavaScriptu. Toto řešení umožňuje větší modularitu a lepší správu kódu.

Závěrem lze konstatovat, že od svého vzniku prošlo CSS značným rozvojem a přineslo nové možnosti pro design a stylizaci webových stránek [20] [21].

3.5.3 Frameworky

Webový framework je softwarový nástroj určený pro vývoj webových aplikací a webových stránek. Skládá se z knihoven, funkcí, návodů a standardů, které usnadňují programování a vývoj webových aplikací. Umožňuje vývojářům využít hotových řešení a implementovat je do svých projektů.

Hlavním účelem frameworků je zjednodušit práci vývojářů a umožnit rychlejší a efektivnější vývoj webových aplikací. Frameworky nabízejí předem připravená řešení pro různé aspekty webového vývoje, jako je integrace s databázemi, autentizace a autorizace uživatelů, správa relací, cachování, routování a mnoho dalších funkcí. V případě CSS jsou to například třídy, které jednoduchým zavoláním aplikují celou řadu složitých stylů.

Použití webového frameworku přináší výhody, jako je zrychlení vývojového cyklu, zlepšení kvality a bezpečnosti kódu a snížení nákladů na vývoj a údržbu aplikací. Vývojáři mohou také využít rozsáhlých komunit spojených s jednotlivými frameworky, které poskytují podporu, dokumentaci a široký výběr rozšiřujících modulů a knihoven.

Existuje řada různých webových frameworků pro různé programovací jazyky, mezi něž patří například Ruby on Rails, Django pro Python, Laravel pro PHP nebo ASP.NET pro Microsoft .NET, následující část práce bude věnována frameworkům určeným pro CSS [22].

3.5.4 CSS frameworky

Existují také webové frameworky pro CSS, které usnadňují tvorbu a úpravu vzhledu webových stránek. Mezi nejpopulárnější patří:

- Bootstrap – nejpopulárnější framework pro tvorbu responzivního webdesignu, který kombinuje HTML, CSS a JavaScript.
- Foundation – další populární framework pro responzivní webdesign, který poskytuje širokou paletu připravených prvků a je využíván například společnostmi jako Zurb, Adobe nebo National Geographic.
- Bulma – moderní a flexibilní CSS framework s důrazem na jednoduchost a modularitu.
- Materialize – framework navržený podle Google Material Design specifikací, který poskytuje moderní vzhled a mnoho připravených prvků.
- Semantic UI – framework, který se zaměřuje na intuitivní a srozumitelný kód a snadnou úpravu vzhledu webových stránek.

Všechny tyto frameworky mají za úkol zjednodušit tvorbu webových stránek a usnadnit již popsanými způsoby činnost vývojářů při práci na frontendu [23].

3.5.5 Bootstrap

Bootstrap je opensource front-endový framework určený pro vývoj responzivních a mobilně přívětivých webových stránek. Byl vytvořen Markem Ottem a Jacobem Thorntonem ve společnosti Twitter a poprvé uveden na trh v roce 2011.

Framework Bootstrap nabízí sadu CSS stylů a JavaScriptových pluginů pro rychlé a jednoduché vytváření konzistentního vzhledu webových stránek. Cílem jeho vývojářů je dosáhnout konzistentního vzhledu a chování prvků na webových stránkách, čímž se zvyšuje jejich použitelnost a snižují náklady na vývoj. Bootstrap podporuje responzivní design a optimalizaci pro mobilní zařízení.

Bootstrap prošel několika aktualizacemi a v současné době je dostupný ve verzi 5. Tato verze přináší nové funkce a vylepšení, včetně nových komponent, aktualizovaných CSS vlastností a optimalizací pro rychlost.

Díky svým vlastnostem se Bootstrap stal populárním nástrojem pro vývoj webových stránek a aplikací. Mnoho známých webových stránek a aplikací, jako jsou Airbnb, Spotify nebo Microsoft, jej využívá. Bootstrap také disponuje rozsáhlou dokumentací a komunitou vývojářů, kteří sdílejí své zkušenosti a osvědčené postupy [24].

3.5.6 Ukázky Bootstrapu

Na ukázkou je zde uvedeno několik konkrétních příkladů využití Bootstrapu a jak může zefektivnit psaní CSS.

3.5.6.1 Navbar

Navbar, neboli navigační lišta, je jednou z nejčastěji používaných prvků na webových stránkách. Bootstrap umožňuje rychle a snadno vytvořit navbar, aniž byste museli psát velké množství CSS. Níže je ukázka kódu pro vytvoření navbaru s tlačítkem pro mobilní zobrazení:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
```

```

    <a class="navbar-brand" href="#">Logo</a>
    <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
    <li class="nav-item active">
    <a class="nav-link" href="#">Home <span class="sr-
only">(current)</span></a>
    </li>
    <li class="nav-item">
    <a class="nav-link" href="#">About</a>
    </li>
    <li class="nav-item">
    <a class="nav-link" href="#">Contact</a>
    </li>
    </ul>
</div>
</nav>

```

Výsledný navbar bude obsahovat logo a tlačítko pro mobilní zobrazení, které otevře menu s odkazy na stránky.

3.5.6.2 Grid systém

Bootstrap také nabízí jednoduchý způsob, jak vytvořit responzivní layout pomocí svého grid systému. Grid systém umožňuje rozdělit stránku na sloupce, které se automaticky přizpůsobují velikosti obrazovky.

Níže je ukázka kódu, který rozděluje stránku na 2 sloupce pro velké obrazovky, 2 sloupce pro střední obrazovky a 1 sloupec pro malé obrazovky:

```

<div class="container">
    <div class="row">
    <div class="col-lg-6 col-md-6 col-sm-12">
    <p>Sloupec 1</p>
    </div>
    <div class="col-lg-6 col-md-6 col-sm-12">
    <p>Sloupec 2</p>
    </div>
    </div>
</div>

```

3.5.6.3 Formuláře

Bootstrap usnadňuje tvorbu formulářů, které jsou často složité a časově náročné na vytvoření. Níže je ukázka kódu pro vytvoření jednoduchého formuláře s textovým polem a tlačítkem pro odeslání:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp" placeholder="Enter email">
    <small id="emailHelp" class="form-text text-muted">We'll never
share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control"
id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input"
id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me
out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Jak lze vidět, každý prvek formuláře je zabalen do div elementu s třídou `form-group`. Toto zjednodušení také umožňuje snadné přidání vlastních stylů, které by jinak byly velmi obtížné vytvořit.

Bootstrap také poskytuje mnoho užitečných prvků, jako například modální okna, rozbalovací menu, tooltipy a mnoho dalšího. Tyto prvky jsou opět velmi snadno použitelné a mohou výrazně zjednodušit vývoj webových stránek [25] [26].

3.5.7 Porovnání prostého CSS a Bootstrapu

Zde je porovnání několika ukázek kódu CSS a kódu Bootstrap, které ukazují, jak Bootstrap usnadňuje psaní CSS:

3.5.7.1 Vytvoření tlačítka

CSS kód:

```
button {
  display: inline-block;
  padding: 6px 12px;
  font-size: 14px;
  font-weight: 400;
  line-height: 1.42857143;
  text-align: center;
  white-space: nowrap;
  vertical-align: middle;
  -ms-touch-action: manipulation;
  touch-action: manipulation;
  cursor: pointer;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
  background-image: none;
  border: 1px solid transparent;
  border-radius: 4px;
  color: #fff;
  background-color: #337ab7;
  border-color: #2e6da4;
}
```

Bootstrap kód:

```
<button class="btn btn-primary">Primary Button</button>
```

Tento příklad jasně demonstruje, že Bootstrap kód je mnohem kratší a srozumitelnější. Využívá třídu `btn` a `btn-primary`, které definují vzhled tlačítka včetně barvy pozadí, ohraničení a dalších detailů.

3.5.7.2 Vytvoření sloupce

CSS kód:

```
.col-1 {  
  width: 8.33%;  
  float: left;  
}  
  
.col-2 {  
  width: 16.66%;  
  float: left;  
}  
  
.col-3 {  
  width: 25%;  
  float: left;  
}  
  
.col-4 {  
  width: 33.33%;  
  float: left;  
}  
  
.col-5 {  
  width: 41.66%;  
  float: left;  
}  
  
.col-6 {  
  width: 50%;  
  float: left;  
}  
  
.col-7 {  
  width: 58.33%;  
  float: left;  
}  
  
.col-8 {  
  width: 66.66%;  
  float: left;  
}  
  
.col-9 {  
  width: 75%;  
  float: left;  
}
```

```
.col-10 {
  width: 83.33%;
  float: left;
}

.col-11 {
  width: 91.66%;
  float: left;
}

.col-12 {
  width: 100%;
  float: left;
}
```

Bootstrap kód:

```
<div class="col-md-4">Obsah sloupce</div>
```

V Bootstrapu použijeme třídu `col-md-4`, což nám umožňuje jednoduše nastavit šířku sloupce bez nutnosti psát pro každou šířku samostatný CSS kód. Třída `col-md-4` nastaví šířku sloupce na 33,33%, což odpovídá `col-4` v předchozím příkladu.

V obou příkladech Bootstrap usnadňuje psaní CSS a umožňuje nám se rychleji zaměřit na tvorbu samotného obsahu.

3.5.8 Javascript

Javascript byl vyvinut v roce 1995 Brendanem Eichem ve společnosti Netscape Communications Corporation. Původně byl navržen jako jednoduchý jazyk pro webové stránky, který by umožnil reakci na uživatelské akce a interakce. V roce 1996 byl Javascript standardizován jako ECMAScript, a dnes patří mezi nejžádanější programovací jazyky pro vývoj webových aplikací.

Javascript se využívá pro vytváření interaktivních webových stránek, které reagují na uživatelské vstupy, jako jsou kliknutí, pohyby myši či klávesové zkratky. Dále umožňuje manipulovat s obsahem stránek, který je vytvořen pomocí HTML a CSS.

Kromě toho se Javascript uplatňuje při vývoji webových aplikací, jako jsou chatovací programy, hry a další aplikace, které fungují v prohlížeči. Javascript také nachází využití v backendových technologiích, kde se využívají nástroje jako Node.js [27].

Zde je ukázka kódu, který zobrazí vyskakovací okno se zprávou „Ahoj..“ po kliknutí na tlačítko:

```
<button onclick="myFunction()">Klikni mě</button>

<script>
function myFunction() {
    alert("Ahoj, tohle je ukázka Javascriptu!");
}
</script>
```

3.5.9 jQuery

jQuery je knihovna JavaScriptového kódu, která zjednodušuje práci s HTML dokumenty, manipulaci s DOM, ovládání událostí, zpracování formulářů, animací a načítání obsahu pomocí AJAXu. Byla vytvořena v roce 2006 Johnem Resigem a dodnes patří mezi nejpoblárnější JavaScriptové knihovny. Díky jQuery je možné snížit potřebu psát rozsáhlý kód v čistém JavaScriptu, což usnadňuje zejména práci vývojářům, ale také dále zlepšuje interakci uživatele s webovými stránkami a zlepšuje jejich výkon [28].

Zde je ukázka kódu s tlačítkem, které za pomoci jQuery po kliknutí změni barvu pozadí:

```
<!DOCTYPE html>
<html>
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></
script>
</head>
<body>

<button id="changeColor">Změnit barvu pozadí</button>

<script>
$(document).ready(function() {
```

```

        $("#changeColor").click(function(){
            $("body").css({"background-color": "yellow"});
        });
    });
</script>

</body>
</html>

```

Tento kód využívá jQuery k vyhledání tlačítka s id "changeColor" a připojení `onClick` funkce, která změní barvu pozadí na žlutou po kliknutí na tlačítko.

3.5.10 Node.js

Node.js je open-source, multiplatformní JavaScriptové runtime prostředí, které umožňuje vývojářům vytvářet serverové aplikace v JavaScriptu. Node.js byl poprvé představen v roce 2009 vývojářem Ryanem Dahlem, který reagoval na absenci řešení pro psaní serverového kódu v JavaScriptu.

Od svého vzniku se Node.js stal velmi populárním a široce používaným nástrojem pro webové vývojáře. Node.js je proslulý svou rychlostí a efektivitou, což z něj činí ideální volbu pro aplikace s vysokou zátěží, jako jsou například chatovací aplikace, realtime aplikace a streamovací platformy.

Některé z nejčastějších využití Node.js jsou:

- Vytváření backendových aplikací na straně serveru
- Vývoj API a mikroslužby
- Vytváření streamovaných aplikací a her
- Vytváření nástrojů pro automatizované testování a linting

Node.js také obsahuje mnoho užitečných knihoven, jako například Express pro vytváření webových aplikací a Socket.IO pro vytváření realtime aplikací [29].

Zde je ukázka kódu v Node.js:

```
// Připojte modul 'fs'
var fs = require('fs');

// Přečtete obsah souboru a vypište jej
fs.readFile('cesta/k/souboru.txt', function(err, data) {
  if (err) throw err;
  console.log(data);
});
```

Tento kód čte obsah souboru, jméno a cestu k souboru definujete změnou textu "cesta/k/souboru.txt". Jakmile se soubor načte, jeho obsah se vypíše na konzolu.

3.5.11 Javascript frameworky

Dále existuje mnoho populárních JavaScriptových frameworků, které se využívají při vývoji webových aplikací. Mezi nejpopulárnější frameworky patří:

- React – vyvinutý společností Facebook, používaný pro tvorbu interaktivních uživatelských rozhraní a jednostránkových aplikací (Single Page Applications, SPA).
- Angular – vyvinutý společností Google, používaný pro tvorbu SPA a pokročilých webových aplikací.
- Vue.js – oblíbený framework u webových vývojářů díky své jednoduchosti a svižnosti, používá se k tvorbě SPA a widgetů.
- Ember.js – pomáhá vývojářům upravit a spravovat složité webové aplikace, používá se k tvorbě SPA a webových aplikací jakéhokoli rozsahu.
- Backbone.js – poskytuje strukturu pro vývoj SPA, při jeho přímém využití se často využívají další knihovny.

Frameworky jsou oblíbené mezi webovými vývojáři pro svou schopnost zjednodušit vývoj webových aplikací a zlepšit výkon a udržitelnost kódu. Používají se k tvorbě jednostránkových aplikací (SPA), interaktivních uživatelských rozhraní a pokročilých

webových aplikací. Frameworky jsou velmi populární a používají se při vytváření webových stránek různých velikostí a složitosti [30].

Zde je jednoduchá ukázka React kódu:

```
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  function increment() {
    setCount(count + 1);
  }

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={increment}>Increment</button>
    </div>
  );
}

export default Counter;
```

Tento kód definuje komponentu `Counter`, která obsahuje stav proměnné `count`. Při kliknutí na tlačítko se tento stav inkrementuje o 1. Následně se zobrazí aktuální stav `count` v odstavci a tlačítko s popiskem "Increment".

3.5.12 PHP

PHP (Hypertext Preprocessor) je interpretovaný skriptovací jazyk, který se využívá pro vývoj webových aplikací. Tento jazyk byl založen Rasmusem Lerdorfem v roce 1994, přičemž první veřejná verze byla uvedena na trh v roce 1995.

V počáteční fázi byl PHP pouhá sbírka skriptů umožňujících zpracování webových formulářů, zobrazení obsahu z databází a vytváření jednoduchých dynamických webových stránek. V roce 1997 byla uvedena první verze PHP 3, jež zahrnovala řadu pokročilých funkcí, například podporu databází, regulární výrazy a mnohé další.

Verze PHP 4 byla vydána v roce 2000 a představovala plně objektově orientovaný jazyk s mnoha dalšími vylepšeními. Následně byla v roce 2004 uvedena verze PHP 5, která přidávala více objektových vlastností, jako jsou rozhraní a abstraktní třídy. PHP 5 zároveň přinesl řadu nových funkcí, mezi něž patřila podpora SimpleXML a nativní podpora pro práci s SQLite databázemi.

V roce 2014 byla uvedena verze PHP 5.6, jež nabídla další vylepšení a opravy chyb. Následně v roce 2015 byla vydána verze PHP 7, která představovala revoluční změnu díky výraznému zrychlení výkonu a snížení nároků na paměť. PHP 7 rovněž přinesl mnoho nových funkcí a vylepšení, mezi něž patřily deklarace typů, skalární typy a nové operátory.

V současné době je PHP verze 8.x.x velmi populárním jazykem pro vývoj webových aplikací různých typů včetně sociálních sítí, online obchodů a správy obsahu. Vývojáři ocení existence mnoha populárních frameworků pro PHP jako Laravel, Symfony, CakePHP a CodeIgniter, které usnadňují vývoj webových aplikací a zjednodušují práci s kódem.

V posledních letech se také stále více využívá PHP pro vývoj webových služeb (API) a aplikací v rámci mikroslužební architektury. Mezi největší výhody PHP patří snadná instalace a integrace s dalšími technologiemi, jako je například MySQL databáze nebo Apache web server. Tato vlastnost umožňuje rychlou instalaci a použití v různých prostředích. PHP také nabízí mnoho předdefinovaných funkcí, které usnadňují práci s daty, soubory nebo řetězci.

Nicméně, s narůstajícími nároky na bezpečnost a výkon se v posledních letech objevily i některé nevýhody PHP. Někteří vývojáři kritizují jeho slabé typování, což může vést k nechtěným chybám a problémům s výkonem. Také není ideální pro velké a složité aplikace, které mohou být náchylné k chybám a neefektivním řešením.

V každém případě, PHP zůstává důležitým a významným programovacím jazykem v oblasti webového vývoje. S pravidelnými aktualizacemi a zlepšeními se stále vyvíjí a přizpůsobuje novým trendům a potřebám [31] [32].

3.5.13 Nette

Nette je opensource framework určený pro vývoj webových aplikací v programovacím jazyce PHP. Jeho existenci lze datovat zhruba od roku 2005, kdy jej začal vyvíjet David Grudl. Od té doby se stal jedním z nejpoužívanějších PHP frameworků nejen v České republice a na Slovensku.

Jedním z hlavních cílů Nette je zefektivnit a zrychlit vývoj webových aplikací. Toho se snaží dosáhnout pomocí inovativní vývojové metodiky, jako je například oddělení aplikační logiky od prezentační vrstvy nebo umožnění jednoduchého testování. Pro tyto účely Nette využívá moderní technologie a nástroje, jako jsou Composer, Latte šablonovací systém nebo Tracy debugger.

Nette poskytuje vývojářům možnost vytvářet aplikace různých typů – od menších projektů až po velké komplexy systémů. Modulární architektura frameworku umožňuje jednoduché rozšiřování funkčnosti aplikace. Nette také nabízí integraci s různými databázovými systémy a podporu tvorby REST API.

Mezi výhody Nette pro vývojáře patří jednoduchost a přehlednost kódu, rychlý vývoj aplikací, snadné testování a ladění chyb díky využití Tracy debuggeru. Díky Latte šablonovacímu systému je možné snadno oddělit prezentační vrstvu od aplikační logiky a zlepšit tak přehlednost kódu. Nette také nabízí mnoho rozšíření a pluginů, které usnadňují vývoj a implementaci dalších funkcionalit.

Nette však pro nové vývojáře může být náročný na pochopení kvůli specifickým konvencím a řešení některých problémů. Taktéž může být narůstající složitost projektů náročná na hardware a může vyžadovat větší paměťovou a procesorovou kapacitu ve srovnání s optimalizovaným PHP.

Obecně lze ale konstatovat, že Nette je silným nástrojem pro vývoj webových aplikací, který poskytuje mnoho možností a umožňuje rychlý a efektivní vývoj. Vzhledem k jeho popularitě a rozšíření je také snadno dostupný a podporován komunitou vývojářů [33].

3.5.14 Porovnání prostého PHP a Nette

Jedním z typických příkladů, kde Nette framework umožňuje zkrácení délky kódu oproti čistému PHP, je práce s formuláři. Uvažujme následující příklad formuláře pro registraci uživatele:

Čistý PHP:

```
<?php
if($_SERVER['REQUEST_METHOD'] === 'POST'){
    //validace vstupů
    $name = isset($_POST['name']) ?
htmlspecialchars(trim($_POST['name'])) : null;
    $email = isset($_POST['email']) ?
htmlspecialchars(trim($_POST['email'])) : null;
    $password = isset($_POST['password']) ?
htmlspecialchars(trim($_POST['password'])) : null;

    //kontrola, zda byly vyplněny všechny údaje
    if($name === null || $email === null || $password === null){
        echo 'Musíte vyplnit všechna pole.';
        exit;
    }

    //hashování hesla
    $hashed_password = password_hash($password, PASSWORD_DEFAULT);

    //uložení uživatele do databáze
    //...

    echo 'Registrace proběhla úspěšně!';
}
?>

<form method="post">
    <label for="name">Jméno:</label>
    <input type="text" name="name" id="name" required><br>

    <label for="email">E-mail:</label>
    <input type="email" name="email" id="email" required><br>

    <label for="password">Heslo:</label>
    <input type="password" name="password" id="password"
required><br>

    <input type="submit" value="Registrovat">
```

```
</form>
```

Nette framework:

```
<?php
use Nette\Application\UI\Form;

$form = new Form;

$form->addText('name', 'Jméno:')
    ->setRequired('Musíte vyplnit jméno.');
```

```
$form->addEmail('email', 'E-mail:')
    ->setRequired('Musíte vyplnit e-mail.')
```

```
    ->addRule(Form::EMAIL, 'Zadejte prosím platný e-mail.');
```

```
$form->addPassword('password', 'Heslo:')
    ->setRequired('Musíte vyplnit heslo.');
```

```
$form->addSubmit('send', 'Registrovat');
```

```

$form->onSuccess[] = function($form, $values){
    $hashed_password = password_hash($values->password,
PASSWORD_DEFAULT);

    //uložení uživatele do databáze
    //...

    echo 'Registrace proběhla úspěšně!';
};

echo $form;
?>
```

V uvedené ukázce je využit Nette formulář, který značně usnadňuje práci s formuláři a zkracuje jejich zpracování. Na rozdíl od klasického zpracování formulářů není potřeba ručně zpracovávat vstupy ani je validovat. Nette formulář toto zpracování obstarává pomocí funkcí jako je `setRequired` a `addRule`. Dále je zpracování odeslání formuláře velmi snadné díky dostupnosti `onSuccess` události. Celkově tedy Nette framework výrazně zjednodušuje a usnadňuje práci s formuláři, což zvyšuje efektivitu vývoje aplikací [34].

3.5.15 Další frameworky

Kromě Nette existuje celá řada dalších PHP frameworků, které jsou často používány pro tvorbu moderních webových aplikací. Mezi nejpopulárnější z nich patří následující:

Laravel je moderní a elegantní framework, který přináší mnoho funkcí pro vývojáře, včetně ORM, routingu, autentizace, testování a mnoho dalšího. Laravel je velmi oblíbený díky své jednoduchosti a uživatelské přívětivosti.

Symfony je robustní a modulární framework, který poskytuje vývojářům mnoho nástrojů pro tvorbu velkých a komplexních webových aplikací. Mezi jeho nejvýznamnější funkce patří Dependency Injection, ORM, routing, validace a mnoho dalšího.

CodeIgniter je lehký a rychlý framework, který nabízí jednoduchý a snadno použitelný způsob pro tvorbu webových aplikací. CodeIgniter je vhodný zejména pro menší projekty a je oblíbený díky své rychlosti a jednoduchosti.

CakePHP je framework s důrazem na konvenci nad konfigurací, který umožňuje vývojářům rychle vytvářet webové aplikace s minimálním úsilím. Mezi jeho nejvýznamnější funkce patří ORM, routing, validace a mnoho dalšího.

Zend Framework je modulární framework, který nabízí vývojářům mnoho nástrojů pro tvorbu velkých a komplexních webových aplikací. Mezi jeho nejvýznamnější funkce patří Dependency Injection, ORM, routing, validace a mnoho dalšího.

Každý z těchto frameworků má své výhody a nevýhody a výběr závisí na konkrétních požadavcích projektu. Vývojáři obvykle volí framework podle jeho funkcionality, uživatelské přívětivosti, dostupnosti dokumentace a podpory komunity [35].

3.5.16 Databáze

Databáze představují klíčový prvek mnoha webových aplikací, neboť umožňují organizovat, ukládat a spravovat velké množství dat. Tyto databáze jsou nezbytné pro ukládání a zpracování dat na webových stránkách, včetně ukládání uživatelských účtů, produktů a dalších informací.

Historie databázových systémů sahá až do 60. let, kdy byla vytvořena první relační databáze. Postupem času se vyvinula celá řada různých typů databázových systémů, včetně

hierarchických, síťových, objektových, relačních a NoSQL databází. Každý z těchto typů databázových systémů má své výhody a nevýhody a každý se hodí pro různé účely.

Nejčastěji používaným typem databázových systémů na webu jsou relační databázové systémy. Tyto systémy ukládají data v řádcích a sloupcích v tabulkách, což umožňuje snadné vyhledávání a manipulaci s daty pomocí SQL (Structured Query Language). Relační databázové systémy jsou ideální pro ukládání strukturovaných dat, jako jsou například uživatelské účty, objednávky nebo produktové katalogy.

NoSQL databázové systémy, obvykle založené na objektových modelech, představují novější typy databázových systémů, které se v posledních letech staly populárními. Tyto systémy jsou navrženy tak, aby umožnily ukládání a manipulaci s nestrukturovanými daty, jako jsou například fotografie nebo video soubory. NoSQL databázové systémy jsou často používány v rámci aplikací pro zpracování velkého množství dat (Big Data), avšak mohou sloužit i jako alternativa klasických relačních databázových systémů.

Databázové systémy jsou klíčové pro webové aplikace, neboť umožňují snadnou a rychlou manipulaci s daty, včetně vyhledávání, aktualizace nebo mazání dat. Databázové systémy také umožňují ukládání dat na serveru, což znamená, že jsou data k dispozici všem uživatelům připojeným k webové aplikaci.

Webové aplikace mohou využívat různé typy databázových systémů, jako jsou relační databázové systémy (např. MySQL, PostgreSQL, Oracle, Microsoft SQL Server), dokumentové databázové systémy (např. MongoDB), grafové databázové systémy (např. Neo4j) a další. Každý typ databázového systému je vhodný pro různé účely a aplikace.

Vývoj databázových systémů stále pokračuje a objevují se nové technologie, jako například cloudové databázové systémy nebo databázové systémy s podporou strojového učení. Databázové systémy jsou tedy neodmyslitelnou součástí moderního světa IT a stále se rozvíjejí, aby dokázaly efektivně a spolehlivě ukládat a zpracovávat stále větší a rozmanitější množství dat [36].

3.5.17 SQL

SQL (Structured Query Language) představuje standardizovaný jazyk pro správu relačních databází. Byl vyvinut společností IBM v roce 1974 jako součást projektu System R a v roce 1986 byl standardizován organizací ANSI. Od té doby se stal jedním z nejpoužívanějších jazyků pro práci s databázemi.

SQL je deklarativní jazyk, což znamená, že programátor specifikuje pouze, jaká data chce získat nebo upravit, aniž by specifikoval, jak to má být provedeno. SQL využívá tabulky a sloupce k ukládání a organizování dat, což umožňuje snadný přístup k informacím a rychlé dotazování.

V průběhu let se SQL vyvíjel a rozšiřoval o nové funkce a vlastnosti. Byly přidány operátory, funkce a příkazy, jako jsou JOIN, GROUP BY, UNION, CREATE TABLE, ALTER TABLE a další. V současnosti existuje několik variant SQL, jako jsou MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database a mnoho dalších.

Uplatnění SQL se nachází celé řadě oblastí, jako jsou webové aplikace, e-commerce, bankovníctví, zdravotnictví, vládní organizace a mnoho dalších. V oblasti webových stránek je SQL využíván k ukládání informací o uživateli, produktech, objednávkách a dalších datech, které jsou nezbytné pro správné fungování webové aplikace. V e-commerce pak SQL slouží k ukládání informací o zákaznících, objednávkách, platbách a dalších datech týkajících se prodeje a samozřejmě mnoha dalších případech [36].

3.5.18 MySQL

MySQL je relační databázový systém, který byl poprvé vydán v roce 1995. Byl vytvořen švédskými vývojáři Michaelem Wideniusem a Davidem Axmarkem. V roce 1998 byl systém převzat firmou MySQL AB, která jej později prodala společnosti Sun Microsystems. V roce 2008 byla Sun Microsystems zakoupena společností Oracle Corporation, která ho nyní vlastní.

MySQL je opensource databázový systém, což znamená, že jeho zdrojový kód je volně dostupný a může být modifikován a použit podle potřeby. Systém je naprogramován v jazycích C a C++ a podporuje široké spektrum operačních systémů, včetně Windows, Linux a macOS.

MySQL se využívá v mnoha oblastech, jako jsou webové aplikace, e-commerce stránky, CRM systémy, online hry, aplikace pro Internet věcí (IoT) a mnoho dalších. Vzhledem k tomu, že se jedná o open-source systém, je velmi oblíbený mezi vývojáři a mnoho populárních opensource projektů ho využívá jako svůj primární databázový systém [37].

3.5.19 SQL dotazy

Pro představení dotazů v SQL je zde uvedeno několik jednodušších příkladů:

Vytvoření tabulky s názvem "zamestnanci" obsahující sloupce pro jméno, příjmení a číselný údaj o jejich mzdě:

```
CREATE TABLE zamestnanci (  
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  jmeno VARCHAR(30) NOT NULL,  
  prijmeni VARCHAR(30) NOT NULL,  
  mzda INT(10)  
);
```

Vložení nového zaměstnance do tabulky "zamestnanci":

```
INSERT INTO zamestnanci (jmeno, prijmeni, mzda) VALUES ('Jan',  
'Novák', 50000);
```

Získání všech zaměstnanců z tabulky "zamestnanci":

```
SELECT * FROM zamestnanci;
```

Získání všech zaměstnanců, kteří mají mzdu větší než 50000:

```
SELECT * FROM zamestnanci WHERE mzda > 50000;
```

Aktualizace mzdových údajů pro zaměstnance s ID 1:

```
UPDATE zamestnanci SET mzda = 60000 WHERE id = 1;
```

Odstranění zaměstnance s ID 2:

```
DELETE FROM zamestnanci WHERE id = 2;
```

Toto jsou pouze základní příklady dotazů jazyka SQL. Samozřejmě, existuje mnoho dalších pokročilých funkcí, které umožňují komplexní manipulaci s daty v databázích.

3.5.20 SQL v PHP

Následují jednoduché příklady pro představení SQL dotazů v jazyce PHP, využito bylo prostředí MySQL [38] [39]:

Připojení k databázi:

```
$servername = "localhost";
$username = "uzivatel";
$password = "heslo";
$dbname = "databaze";

// Vytvoření připojení
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Kontrola připojení
if (!$conn) {
    die("Připojení se nezdařilo: " . mysqli_connect_error());
}
echo "Připojení bylo úspěšné.";
```


Dotaz CREATE TABLE:

```
$sql = "CREATE TABLE zakaznici (  
    id INT PRIMARY KEY,  
    jmeno VARCHAR(50),  
    prijmeni VARCHAR(50),  
    email VARCHAR(100)  
)";  
  
if (mysqli_query($conn, $sql)) {  
    echo "Tabulka byla úspěšně vytvořena."  
} else {  
    echo "Chyba při vytváření tabulky: " . mysqli_error($conn);  
}
```

Dotaz INSERT INTO:

```
$sql = "INSERT INTO zakaznici (id, jmeno, prijmeni, email)  
VALUES (1, 'Jan', 'Novák', 'jan.novak@email.com)";  
  
if (mysqli_query($conn, $sql)) {  
    echo "Data byla úspěšně vložena."  
} else {  
    echo "Chyba při vkládání dat: " . mysqli_error($conn);  
}
```

Dotaz SELECT:

```
$sql = "SELECT jmeno, prijmeni FROM zakaznici";  
$result = mysqli_query($conn, $sql);  
  
if (mysqli_num_rows($result) > 0) {  
    // Výstup dat každého řádku  
    while($row = mysqli_fetch_assoc($result)) {  
        echo "Jméno: " . $row["jmeno"]. " - Příjmení: " .  
$row["prijmeni"]. "<br>";  
    }  
} else {  
    echo "Žádná data nebyla nalezena."  
}
```

Dotaz UPDATE:

```
$sql = "UPDATE zakaznici SET email='novy.email@email.com'
WHERE id=1";

if (mysqli_query($conn, $sql)) {
    echo "Data byla úspěšně aktualizována.";
} else {
    echo "Chyba při aktualizaci dat: " . mysqli_error($conn);
}
```

Dotaz DELETE:

```
$sql = "DELETE FROM zakaznici WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Data byla úspěšně smazána.";
} else {
    echo "Chyba při mazání dat: " . mysqli_error($conn);
}
```

3.5.21 SQL v Nette

Zde jsou uvedeny shodné příklady, nyní v jazyce PHP s použitím frameworku Nette [40]:

Připojení k databázi:

```
use Nette\Database\Connection;

$dsn = "mysql:host=localhost;dbname=databaze";
$username = "uzivatel";
$password = "heslo";

// Vytvoření připojení
$db = new Connection($dsn, $username, $password);

// Kontrola připojení
if ($db->getPdo()) {
    echo "Připojení bylo úspěšné.";
} else {
    die("Připojení se nezdařilo.");
}
```

Dotaz CREATE TABLE:

```
use Nette\Database\Table\Selection;

// Vytvoření tabulky
$db->query('CREATE TABLE zakaznici (
    id INT(11) PRIMARY KEY,
    jmeno VARCHAR(50),
    prijmeni VARCHAR(50),
    email VARCHAR(100)
)');

// Kontrola vytvoření tabulky
if ($db->table('zakaznici')->count('*') > 0) {
    echo "Tabulka byla úspěšně vytvořena.";
} else {
    die("Chyba při vytváření tabulky.");
}
```

Dotaz INSERT INTO:

```
use Nette\Database\Table\ActiveRow;

// Vložení dat
$db->table('zakaznici')->insert([
    'id' => 1,
    'jmeno' => 'Jan',
    'prijmeni' => 'Novák',
    'email' => 'jan.novak@email.com'
]);

// Kontrola vložení dat
if ($db->table('zakaznici')->get(1) instanceof ActiveRow) {
    echo "Data byla úspěšně vložena.";
} else {
    die("Chyba při vkládání dat.");
}
```

Dotaz SELECT:

```
use Nette\Database\Table\Selection;

// Výběr dat
$data = $db->table('zakaznici')->select('jmeno, prijmeni');

// Výstup dat každého řádku
foreach ($data as $row) {
    echo "Jméno: " . $row['jmeno'] . " - Příjmení: " .
    $row['prijmeni'] . "<br>";
}
```

Dotaz UPDATE:

```
use Nette\Database\Table\ActiveRow;

// Aktualizace dat
$row = $db->table('zakaznici')->get(1);
$row->update(['email' => 'novy.email@email.com']);

// Kontrola aktualizace dat
if ($db->table('zakaznici')->get(1)->email == 'novy.email@email.com')
{
    echo "Data byla úspěšně aktualizována.";
} else {
    die("Chyba při aktualizaci dat.");
}
```

Dotaz DELETE:

```
use Nette\Database\Table\ActiveRow;

// Smazání dat
$db->table('zakaznici')->where('id', 2)->delete();

// Kontrola smazání dat
if (!$db->table('zakaznici')->get(2) instanceof ActiveRow) {
    echo "Data byla úspěšně smazána.";
} else {
    die("Chyba při mazání dat.");
}
```

3.5.22 Bezpečnost programování

Vzhledem k faktu, že toto téma by vydalo na celou diplomovou práci, bude mu zde věnována pouze drobná část. Při vývoji softwaru je důležité dodržovat několik obecných pravidel, která zajišťují bezpečnost uživatelů a jejich dat. Následuje několik klíčových bodů, které je nutné mít na paměti:

- Validace vstupů: Všechny uživatelské vstupy je třeba ověřovat a očišťovat, aby bylo zabráněno vložení škodlivých dat do aplikace.
- Princip nejnižších oprávnění: Uživatelům, procesům a systémům by měla být udělena pouze nezbytná oprávnění pro provedení jejich úkolů.
- Důkladné testování: Provést důkladné testování kódu, zahrnující jednotkové, integrační a bezpečnostní testy, které odhalí a opraví chyby a zranitelnosti.
- Šifrování citlivých dat: Při přenosu či ukládání citlivých informací je nutné využívat silné šifrovací metody.
- Pravidelné aktualizace: Udržovat kód, knihovny a závislosti aktuální a aplikovat bezpečnostní záplaty.
- Zabezpečení zdrojového kódu: Zajistit, že zdrojový kód je bezpečně uložen a chráněn před neoprávněným přístupem.
- Principy bezpečného návrhu: Při návrhu aplikace dodržovat zásady bezpečného návrhu, jako je prevence zranitelností způsobených špatným návrhem.
- Důkladná dokumentace: Vytvářet podrobnou dokumentaci pro kód, která umožní ostatním vývojářům snadno pochopit a správně pokračovat v práci na projektu.

- Pravidelný monitoring a audit: Monitorovat a auditovat aplikaci, aby byl získán přehled o bezpečnostních událostech a umožněno včasné reagování na incidenty.
- Vzdělávání a školení: Neustále se vzdělávat a školit sebe i svůj tým v oblasti bezpečnosti, aby byli informováni o nejnovějších hrozbách a osvědčených postupech.

Dodržováním těchto obecných pravidel a postupů při programování lze snížit riziko zneužitelných chyb a zajistit bezpečnost výsledné aplikace jak pro uživatele, tak pro jejich data. V kontextu PHP, SQL a Javascriptu lze uvést následující příklady:

Prvním krokem k bezpečnému programování v PHP je validace vstupů. Vývojáři by měli využívat funkci, jako je `filter_var()` nebo regulární výrazy, aby zajistili správný typ dat. Při práci s databázemi je důležité používat předpřipravené dotazy (prepared statements) prostřednictvím PDO (PHP Data Objects) nebo MySQLi, což minimalizuje riziko SQL injection. Je rovněž nezbytné zamezit použití funkce `eval()` a zakázat zobrazování chyb na produkčním serveru..

JavaScript je univerzální skriptovací jazyk, který se používá především na straně klienta. Bezpečnost v Javascriptu zahrnuje použití `textContent` místo `innerHTML` při manipulaci s DOMem, což zabraňuje cross-site scripting (XSS) útokům. Vývojáři by měli také validovat a escapovat vstupy na straně klienta (i když validace na straně serveru je stále nezbytná) a používat Content Security Policy (CSP) pro omezení zdrojů obsahu.

SQL, jazyk pro správu databází, také vyžaduje zvláštní pozornost v oblasti bezpečnosti. Použitím předpřipravených dotazů (prepared statements) lze minimalizovat riziko SQL injection. Je také důležité omezit oprávnění uživatelů a aplikací, aby měly přístup pouze k potřebným datům a operacím. V neposlední řadě by vývojáři měli bezpečně ukládat hesla pomocí hašovacích algoritmů, jako je bcrypt, místo ukládání hesel jako čistý text.

Dodržováním osvědčených postupů a pravidel pro bezpečné programování v PHP, JavaScriptu a SQL lze snížit riziko zneužitelných chyb. Je důležité, aby vývojáři byli neustále informováni o nejnovějších bezpečnostních hrozbách a zranitelnostech, aby mohli své aplikace chránit proti stále se měnícím útokům. Spolehlivá a bezpečná webová aplikace je výsledkem nejen pečlivého návrhu a implementace, ale také průběžného sledování a udržování. Tímto způsobem lze dosáhnout optimální bezpečnosti a zabezpečení pro uživatele a jejich data [41].

3.5.23 API

API, neboli "Application Programming Interface" (aplikační programovací rozhraní), je sada pravidel, funkcí, protokolů a nástrojů, které umožňují různým softwarovým aplikacím komunikovat a vyměňovat si data. API slouží jako most mezi dvěma aplikacemi, který usnadňuje jejich spolupráci. API je potřebné pro:

- Integraci služeb: Programátoři mohou využívat API pro integraci různých služeb do svých aplikací. Například může být API použito pro získání přístupu k databázi, mapování nebo platebnímu systému.
- Škálování a modularitu: API umožňuje rozdělit aplikaci na menší, nezávislé části, které mohou být snadno škálovány a spravovány.
- Snadnější údržba a aktualizace: Díky API je snazší aktualizovat a udržovat software, protože změny v jedné části aplikace nemusí nutně ovlivňovat ostatní části.

Pokud je vyvíjena PHP aplikace, může být jako příklad použito API pro získání přístupu k externím službám, například získání počasí, převod měn nebo přístup k sociálním sítím [42].

3.5.24 Redakční systémy

Redakční systémy, známé také jako Content Management Systems (CMS), jsou softwarové aplikace či platformy, které umožňují spravovat, vytvářet a publikovat digitální obsah. Jsou navrženy tak, aby usnadnily práci s webovým obsahem pro redaktory a autory, a to i bez hlubokých znalostí programování.

Potřeba redakčních systémů vznikla s růstem internetu a potřebou snadného a efektivního spravování webových stránek. Tyto systémy umožňují uživatelům editovat obsah, řídit verze dokumentů a spravovat přístupová práva. Vývoj redakčních systémů začal v 90. letech a od té doby se neustále rozvíjí, aby vyhovoval moderním potřebám a technologiím.

Redakční systémy fungují na různých technologiích a platformách, ale většina z nich využívá kombinaci programovacích jazyků, databází a webových serverů. Některé z nejpoužívanějších technologií zahrnují PHP, JavaScript, HTML, CSS, MySQL a Apache. Mezi hlavní redakční systémy patří [43]:

- WordPress – nejpoblárnější redakční systém, který pohání více než 40 % všech webových stránek na světě. Je známý svou flexibilitou, širokou škálou šablon a pluginů.
- Joomla! – další populární redakční systém, který je známý svou robustností a bezpečností. Jeho výhodou je široká komunita a mnoho rozšíření.
- Drupal – tento redakční systém je oblíbený pro svou bezpečnost, škálovatelnost a výkonnost. Je ideální pro složitější projekty s vysokými nároky na správu obsahu.
- Wix – webová platforma, která umožňuje uživatelům vytvářet webové stránky pomocí přetahování prvků. Je zaměřen na snadné použití a rychlé zprovoznění webové stránky.

- Squarespace – tento redakční systém se zaměřuje na design a kvalitu šablon. Je vhodný pro jednoduché projekty a menší webové stránky.

3.5.25 Optimalizace pro vyhledávače

Optimalizace pro vyhledávače (SEO, z anglického Search Engine Optimization) je proces zlepšování viditelnosti webových stránek ve výsledcích vyhledávačů, jako je Google, Bing, popř. český Seznam. Cílem SEO je zvýšit návštěvnost a dosáhnout vyššího umístění ve výsledcích vyhledávání. Z pohledu vývojáře je dobré považovat SEO za důležitou součást tvorby webových stránek. Mezi hlavní body patří:

- Sémantické značky HTML: Používat správně značky, jako `<header>`, `<nav>`, `<main>`, `<article>`, `<section>` a `<footer>`.
- Meta tagy: Zajistit správně vyplněné meta tagy, jako `<title>` a `<meta name="description">`.
- Responzivní design: Vytvářet stránky, které se přizpůsobí různým velikostem obrazovek a zařízením.
- Rychlost načítání: Optimalizovat rychlost načítání stránek kompresí obrázků, minimalizací kódu a využitím cache.
- Čitelné URL adresy: Vytvářet strukturované a čitelné URL adresy s relevantními klíčovými slovy.
- Interní a externí odkazy: Udržovat interní odkazy strukturované a získávejte kvalitní externí odkazy.
- Kvalitní a aktuální obsah: Pravidelně aktualizovat a rozšiřujte obsah na vyvíjených stránkách.
- Klíčová slova: Provést výzkum klíčových slov a začlenit je do obsahu webu.
- Sitemapa: Vytvořit a pravidelně aktualizovat sitemapu pro lepší indexaci stránek vyhledávači.
- Strukturovaná data: Používat strukturovaná data (např. schema.org) pro poskytování dodatečných informací o obsahu.

- Lokální vyhledávání: Optimalizovat web pro lokální trh, včetně správného použití NAP informací a registrace v Google Místních službách.

Dodržováním těchto osvědčených postupů v oblasti SEO lze zajistit, že webové stránky budou lépe viditelné a atraktivní pro uživatele i vyhledávače. Je vhodné pravidelně sledovat trendy a novinky v oblasti SEO a odpovídajícím způsobem na ně reagovat [44].

3.6 Kryptoměny

Kryptoměny, někdy také nazývané digitální měny nebo virtuální měny, představují digitální aktiva sloužící jako prostředek směny. Základem jejich fungování je kryptografie, která zajišťuje bezpečnost transakcí, regulaci vytváření nových jednotek a ověření převodů aktiv mezi jednotlivými uživateli. Kryptoměny jsou založeny na decentralizovaných sítích postavených na technologii blockchain, což představuje veřejnou, neměnnou a distribuovanou účetní knihu evidující všechny transakce.

Bitcoin, založený v roce 2009 anonymním vývojářem či skupinou vývojářů pod pseudonymem Satoshi Nakamoto, je první a nejznámější kryptoměna. Od jeho vzniku bylo založeno tisíce dalších kryptoměn, často nazývaných altcoiny, mezi něž patří například Ethereum, Litecoin, Ripple, Cardano a mnoho dalších.

Kryptoměny umožňují rychlé a relativně levné transakce napříč hranicemi bez nutnosti účasti třetích stran, jako jsou banky nebo platební procesory. Tento přístup může přispět k větší finanční inkluzi pro jedince bez přístupu ke klasickým finančním systémům.

Nicméně, kryptoměny také čelí kritice a kontroverzím. Jejich cenová volatilita může způsobit značné ztráty pro investory, a jejich anonymita a decentralizace mohou usnadnit praní špinavých peněz, financování terorismu a daňové úniky.

Navzdory těmto výzvám se kryptoměny stále více uznávají jako platidlo a investiční nástroj. Vlády, finanční instituce a technologické společnosti zkoumají potenciál blockchainu a kryptoměn pro zlepšení finančních systémů, zabezpečení a další aplikace [45].

Mezi nejčastěji skloňované pojmy ve spojitosti s kryptoměnami jsou:

- Decentralizace: Snaha o odstranění centrální autority, jako jsou banky a vlády z finančního systému a umožnit peer-to-peer transakce.

- Finanční inkluze: Zjednodušení přístupu k finančním službám pro jedince bez bankovních účtů nebo s omezeným přístupem k tradičním finančním systémům.
- Ochrana soukromí: Poskytnutí vyšší úrovně anonymity a ochrany soukromí v porovnání s tradičními finančními transakcemi.
- Omezená inflace: Vytvoření deflační měny s pevným množstvím jednotek, které mohou být vytvořeny, čímž se snižuje riziko inflace.

3.6.1 Principy fungování

Vzhledem k popularitě Bitcoinu byl vybrán jako příklad pro obecný popis principů, na kterých funguje většina kryptoměn. Bitcoin je decentralizovaný blockchainový systém, který umožňuje převody mezi účty v podobě digitální měny – bitcoinu. Celá síť bitcoinu funguje na základě peer-to-peer principu, což znamená, že převody probíhají mezi uživateli bez nutnosti zprostředkovatele, jako jsou banky nebo směnárny. Tento přístup k transakcím zajišťuje anonymitu a snižuje náklady. Druhým dechem nutno dodat, že právě zmíněné směnárny, popř. burzy jsou pro běžné uživatele díky jejich přívětivějšímu rozhraní preferovaným způsobem, jak se svými prostředky manipulovat.

Provedené transakce se zaznamenávají do blockchainu – veřejně dostupné databáze, kde se každá nová transakce zaznamenává v rámci jednotlivých bloků v řetězci. Každá transakce musí být potvrzena, aby mohla být zapsána do blockchainu a stát se tak konečnou. K tomuto účelu slouží proces ověřování, nazývaný "mining", který je založen na konsensu typu "proof of work".

Proof of work spočívá v tom, že uživatelé, kteří se podílejí na těžbě, musí řešit složité matematické úlohy, za které obdrží odměnu v podobě získání měny a současně validují daný blok. Složitost matematické úlohy je dána množstvím zapojených těžařů, což eliminuje jejich nadbytečné množství. Navíc je zajištěna návaznost celého řetězce, protože každý blok obsahuje hash (jednoznačný identifikátor), který je odvozen z předchozího bloku, čímž je zajištěna nemožnost pozdější manipulace s bloky v blockchainu.

Množství bitcoinů, které mohou být vytvořeny, je omezené na 21 milionů. Další zajímavou zmínkou je půlení odměny za těžbu. Probíhá zhruba každé 4 roky a je jím zajištěna adekvátní výše odměny těžařům vzhledem k rostoucí ceně měny. Další technologie, jako například Lightning Network, umožňuje ještě rychlejší a levnější převody mezi účty a to díky operování v druhé vrstvě, tedy nad samotným blockchainem.

Veřejný a privátní klíč je další klíčový prvek bitcoinu. Každý uživatel bitcoinové peněženky má svůj privátní klíč, což je heslo nebo identifikační řetězec, kterým dokládá, že má disponibilní množství bitcoinů a je majitelem peněženky, kterých je možné mít i více. Tento klíč dále slouží k podepisování transakcí, aby se ověřila jejich platnost. Veřejný klíč slouží jako adresa peněženky, na kterou nebo ze které jsou prostředky převáděny.

Za zmínku stojí také další způsob ověření transakcí, využívaný například měnou Ethereum, Proof of stake. Jedná se o alternativním konsensus pro vytváření a ověřování transakcí v kryptoměně, který nahrazuje proof of work. Namísto těžařů, kteří vytěžují nové mince na základě vysokého výkonu svého hardwaru, jsou nové transakce ověřovány a bloky jsou generovány na základě stávajícího množství kryptoměny, které vlastní uživatelé. To znamená, že čím více kryptoměny má uživatel v držení, tím je pravděpodobnější, že bude zvolen pro ověřování bloků. Tento přístup k ověřování transakcí je považován za méně energeticky náročný a zajišťuje vyšší bezpečnost a stabilitu sítě. Na druhou stranu, může to také vést k vzniku oligarchie velkých vlastníků kryptoměn a tím ne zcela spravedlivě ovlivnit postavení všech zapojených validátorů [46].

3.6.2 Motivace k pořízení kryptoměn

Samotné důvody, proč lidé investují do kryptoměn mohou být různé a často závisí na individuálních cílech každého investora. Tyto důvody však lze obecně rozdělit na dvě hlavní kategorie.

Mnoho investorů vidí v kryptoměnách potenciál jako stabilní investiční příležitost. Tyto osoby nakupují kryptoměny s dlouhodobým záměrem držet je jako součást svého portfolia a věří, že se v budoucnu budou jejich hodnoty dále zvyšovat, což zajistí minimálně

udržení hodnoty peněz. Někteří investoři si kupují kryptoměny jako součást diverzifikace svého portfolia aktiv, aby snížili riziko nejistoty v jiných investicích.

Kryptoměny také mohou být považovány za alternativu k tradičním investicím, jako jsou například akcie nebo komodity. Tyto tradiční investice se vyznačují podobnými riziky jako kryptoměny, avšak jejich pořízení bývá podstatně složitější než u kryptoměn. Kromě toho tento proces bývá také nákladnější a z těchto důvodů hledají alternativu. Kryptoměny mohou být atraktivním řešením pro takového investora, protože se jedná o relativně novou a snadno dostupnou třídu aktiv.

Někteří lidé kupují a prodávají kryptoměny na základě krátkodobých výkyvů v cenách, čímž se snaží generovat zisk. Tento přístup k investování do kryptoměn se často nazývá trading. Trading se provádí například na burzách nebo mezi jednotlivými uživateli s kryptoměnami. Obchodování s kryptoměnami je hlavně pro ty, kteří se zajímají o aktivní obchodování, ale také pro ty, kteří věří v rychlé zisky. Časté jsou také případy, kdy začínající obchodníci bez předchozích zkušeností ke svému tréninku zvolí právě kryptoměny díky nízkým poplatkům, snadnému vstupu na trh a možnosti operovat i s relativně malými prostředky. Obchodníci s kryptoměnami sledují denně vývoj cen a snaží se odhadnout, kdy je nejlepší okamžik k nákupu nebo prodeji kryptoměn.

Investování do kryptoměn nabízí různé příležitosti pro různé typy investorů, ať už hledají stabilní dlouhodobé investice nebo rychlé zisky z krátkodobého tradingu. Kromě toho kryptoměny nabízejí investorům zážitek z investování do nové a neustále rostoucí tržní třídy. Je ovšem důležité stále mít na paměti, že stejně jako u tradičních aktiv jsou tyto investice rizikové, ne-li rizikovější a mohou přinést i nemalé ztráty [47].

3.6.3 Kritika

Kromě již uvedených a dobře míněných pohnutek ke vzniku a užívání kryptoměn existuje také celá řada negativních názorů. Tato část práce je věnována nejčastějším z nich:

- Energetická náročnost: Těžba kryptoměn, zejména Bitcoinu, vyžaduje značné množství energie. To má negativní dopad na životní prostředí a zvyšuje uhlíkovou stopu.
- Volatilita: Kryptoměny jsou extrémně nestabilní, což způsobuje prudké kolísání jejich hodnoty. Tato volatilita může být riskantní pro investory a narušovat ekonomickou stabilitu.
- Regulační otázky: Kryptoměny často operují mimo tradiční finanční systémy, což znesnadňuje jejich regulaci a kontrolu ze strany vlád a centrálních bank. To může vést k nelegálním aktivitám, jako je praní špinavých peněz, daňové úniky a financování terorismu.
- Nejasná užitečnost: Někteří kritici poukazují na to, že kryptoměny zatím nemají široké uplatnění v běžném životě jako platidlo. Jejich hlavním účelem zůstává spekulace a uchování hodnoty, což zpochybňuje jejich dlouhodobou udržitelnost a hodnotu.
- Kybernetická bezpečnost: Kryptoměny a související platformy mohou být náchylné k hackerským útokům a krádežím. Historie zaznamenala několik velkých případů, kdy byly zcizeny značné částky kryptoměn.
- Nerovnoměrné rozdělení bohatství: Někteří tvrdí, že kryptoměny zvýhodňují již bohaté jedince, protože ti mají větší přístup k technologiím a zdrojům potřebným pro těžbu a obchodování s kryptoměny. To může vést ke zvýšení ekonomických nerovností.
- Technická složitost: Pro mnoho lidí může být koncept kryptoměn a technologie blockchain obtížně srozumitelný. Tato technická složitost může způsobovat nedůvěru a omezený přístup k těmto technologiím.

- Centralizace: Ačkoli kryptoměny původně vznikly jako decentralizovaná alternativa k tradičním finančním systémům, některé kryptoměny a související služby jsou v současnosti provozovány centralizovaně. To znamená, že kontrola nad kryptoměnou a jejími transakcemi je soustředěna do rukou několika velkých společností nebo jedinců, což odporuje původnímu záměru decentralizace.
- Spekulace a bubliny: Kryptoměnový trh může být náchylný k spekulativním bublinám, které mohou vést k následným prudkým poklesům cen a ztrátám pro investory. Někteří kritici tvrdí, že kryptoměny představují spekulativní investice s omezenou skutečnou hodnotou.
- Odpovědnost vývojářů: V některých případech mohou být kryptoměny založeny na neúplných nebo chybných kódech, což může vést k problémům s bezpečností nebo užitečností. To zvyšuje otázky týkající se odpovědnosti vývojářů a zda jsou schopni zajistit dlouhodobou stabilitu a bezpečnost svých projektů.
- Neetické projekty a podvody: Kryptoměnový sektor je náchylný ke sporným praktikám a podvodům, jako jsou tzv. pump-and-dump schémata, v nichž dochází k manipulaci s cenou kryptoměny za účelem rychlého zisku. Rovněž existují projekty, které nabízejí nerealistické výnosy či se ukazují jako zcela podvodné.

Tyto skutečnosti přispívají k faktu, že kryptoměny čelí kritice ze strany veřejnosti, investorů, vlád a odborníků. Nicméně je důležité zdůraznit, že kryptoměnový průmysl se neustále vyvíjí a inovuje, a některé z těchto problémů mohou být postupem času řešeny. Budoucí regulace, technologický pokrok a celková osvěta o kryptoměnách by mohly přispět ke zmírnění některých z těchto obav a zlepšení jejich pověsti v očích kritiků. Je také důležité připomenout, že ačkoli některé z těchto důvodů jsou velmi vážné a pravdivé, jsou to právě lidé, kteří zneužívají kryptoměny k nelegálním činnostem, stejně jako je tomu ve vědě či v oblasti zcela jiných vynálezů [47].

4 Vlastní práce

Následující část diplomové je zaměřena na praktický návrh webových stránek, který je rozdělen do dvou základních oblastí podle druhu vývoje: frontend a backend. Dále následuje poslední část, zaměřená na web z pohledu řízení projektu.

Frontend, část webových stránek, s níž se primárně setkává uživatel, bude hlavním zaměřením první poloviny této sekce. Bude se podrobněji zabývat různými aspekty, jako jsou:

- Základní koncepce a myšlenka projektu, která definuje cíle a zaměření webových stránek.
- Informační struktura, která poskytuje ucelený přehled o organizaci obsahu a navigaci na stránkách.
- Návrh uživatelského rozhraní (UX) pomocí wireframů, které slouží jako vizuální náčrt struktury a rozložení prvků na stránce.
- Tvorba konkrétního grafického návrhu a jeho následná analýza z pohledu přístupnosti, rychlosti načítání a validitou HTML a CSS kódu.
- Použitelnost, která se soustředí na optimalizaci uživatelského zážitku a snadné ovládání stránek.

Druhá část, nazývaná backend, se zaměřuje na obecné rady a postupy pro tvorbu aplikační části webových stránek. Tato část bude zahrnovat:

- Architekturu a strukturu databáze, která zajišťuje efektivní ukládání a správu dat.

- Implementaci API pro načítání dat vztahujících se k vybraným měnám, což umožňuje automatické získávání aktuálních informací a jejich zobrazení na stránkách.
- Implementaci serverových skriptů a funkcí, které umožňují komunikaci mezi frontendem a databází.
- Bezpečnostní opatření a ochranu osobních údajů uživatelů, což je klíčový aspekt při vývoji webových aplikací.
- Optimalizaci výkonu a škálovatelnosti aplikace, aby byla schopna rychle reagovat na požadavky uživatelů a zvládat narůstající zátěž.
- SEO strategii, která zahrnuje optimalizaci webových stránek pro vyhledávače, s cílem zlepšit jejich pozici ve výsledcích vyhledávání a zvýšit návštěvnost.
- Možné návrhy na budoucí zlepšení, které se budou týkat jak technických aspektů, tak oblastí obsahu a uživatelského rozhraní.

Cílem této praktické části je demonstrovat, jak teoretické znalosti získané v předchozí části práce mohou být aplikovány na konkrétní projekt a poskytnout čtenáři ucelený přehled o procesu návrhu a vývoje webových stránek.

Třetí část vlastní práce bude zaměřena na řízení projektu z perspektivy výdělečného projektu. Tato část bude zahrnovat:

- Projektový plán, který popisuje hlavní cíle, úkoly, zdroje a časový rámec projektu.
- Opodstatnění projektu, které zdůvodňuje jeho realizaci z hlediska tržního potenciálu, konkurenceschopnosti a očekávaných přínosů.

- Očekávaný přínos, který se zaměřuje na odhadované výnosy a dopady projektu na trh, zákazníky a podnikání jako celek.
- Business Canvas Model, který představuje strukturovaný přehled klíčových aspektů podnikání, jako jsou zákazníci, hodnotová nabídka, kanály, vztahy se zákazníky, zdroje, činnosti, partnery, náklady a příjmy.
- Persony, které reprezentují typické uživatele webových stránek, jejich potřeby, chování a motivace.
- UC scénáře (use case scénáře), které popisují interakce mezi uživateli a systémem ve formě příběhů, jež demonstrují, jak webové stránky řeší různé potřeby a situace uživatelů.
- Harmonogram (Ganttův diagram), který znázorňuje časový rámeček projektu, jeho fáze a úkoly, a umožňuje sledovat jejich průběh.
- Náklady projektu neboli rozpočet v případě, že by byl zadán placenému vývojovému týmu, včetně odhadu nákladů na lidské zdroje, technologie, marketing a další související oblasti.
- Možná rizika neúspěchu a jejich ošetření, která identifikují potenciální problémy a navrhují opatření k jejich minimalizaci nebo řešení.
- Kritéria úspěchu projektu, která definují, jak bude úspěch projektu měřen a hodnocen, například pomocí ukazatelů návštěvnosti, konverzních sazeb, zákaznické spokojenosti nebo dosažení stanovených obchodních cílů.

- Plán komunikace a spolupráce s týmem, který popisuje, jak budou informace sdíleny a jak bude probíhat koordinace mezi členy týmu, včetně plánovaných schůzek, nástrojů pro správu projektů a komunikačních kanálů.
- Plán udržitelnosti a rozvoje projektu, který se zabývá strategiemi pro dlouhodobé udržení a zlepšení webových stránek, například pravidelnými aktualizacemi obsahu, technickým rozvojem nebo rozšířením nabídky služeb.
- Doporučení pro budoucí výzkum a rozvoj v oblasti projektu, která identifikují možné nové směry, nápady nebo příležitosti pro další práci na webových stránkách nebo v souvisejících oblastech.

Tato třetí část práce poskytne ucelený pohled na řízení a organizaci projektu z hlediska jeho výdělečného potenciálu a zajistí, že budou zohledněny nejen technické aspekty, ale také obchodní, strategické a uživatelské požadavky.

4.1 Frontend

4.1.1 Základní koncepce

Projektem je portál pro český trh, kde uživatel bez přihlášení může vidět základní informace ohledně vybrané kryptoměny, ceny vybraných měnových párů vč. Jejich ceny v minulosti a po založení účtu si uložit oblíbené měny včetně provedených transakcí.

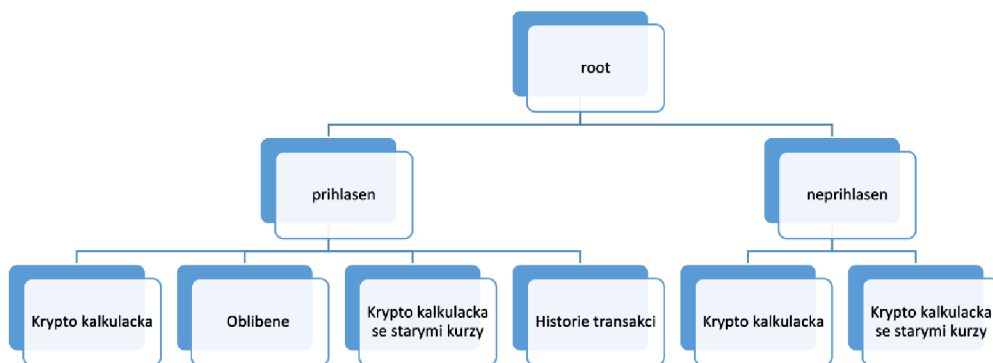
4.1.1.1 Idea

Cílem je vytvořit webovou aplikaci, která umožní uživatelům přepočít kryptoměn. Dále pak zobrazování historického vývoje kryptoměn, včetně zaniklých kryptoměn. Webová aplikace bude umožňovat přihlášení uživatele pro ukládání si oblíbených kryptoměn a dalších funkcionalit.

4.1.1.2 Architektura

- Nepřihlášený uživatel může přehledně vidět aktuální kurzy kryptoměn
 - uvidí historický vývoj kurzu jednotlivých kryptoměn
 - provádět přepočty v rámci těchto měn s aktuální cenou
 - provádět přepočty v rámci těchto měn s historickou cenou
- Přihlášený uživatel může dále
 - filtrovat seznam oblíbených kryptoměn
 - uvidí vlastní historii provedených transakcí
- Budoucí rozvoj
 - nepřihlášený uživatel uvidí aktuální kurzy i národních měn.
 - přihlášený uživatel si nastaví emailové notifikace dle vývoje ceny
 - diskusní fórum
 - knowledge base a rady zkušených investorů

4.1.1.3 Informační architektura aplikace.



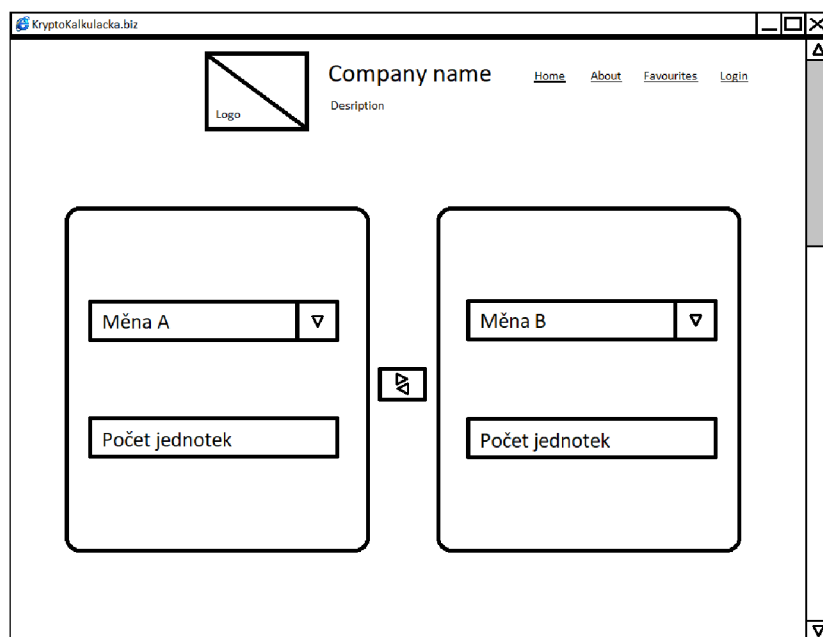
Obrázek 1: Informační architektura aplikace

4.1.2 Návrh uživatelského rozhraní

Návrhy drátěných modelů vychází z případů užití. V této části jsou uvedeny na ukázkou některé z nich. Při tvorbě návrhu uživatelského rozhraní byla snaha držet se čistého a střídmeho funkcionalistického designu.

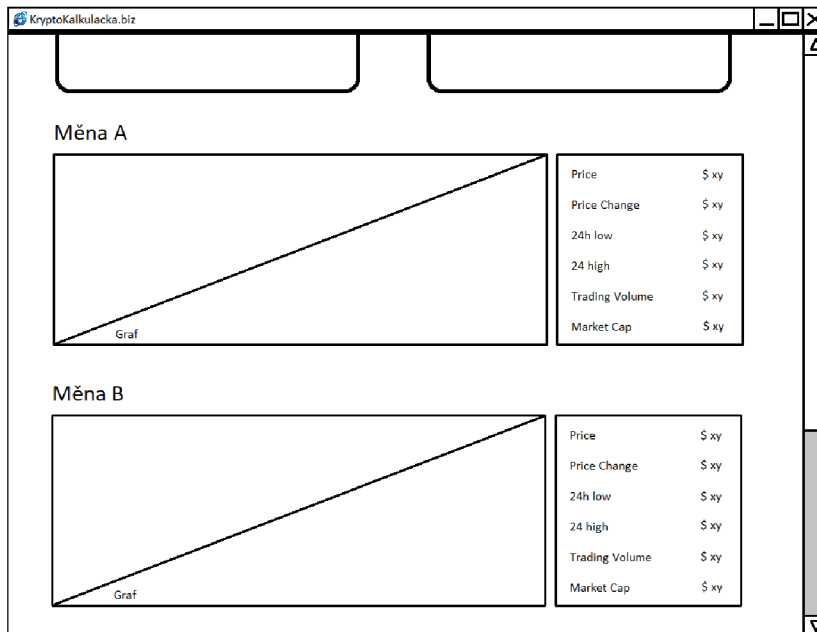
Následující model reprezentuje hlavní stránku a současně formulář pro převod měn. Ovládací prvky tohoto modelu slouží jako textové (vyhledávací) pole nebo rozbalovací seznam. Vidět je také menu, logo a nadpis webu.

4.1.2.1 Desktop



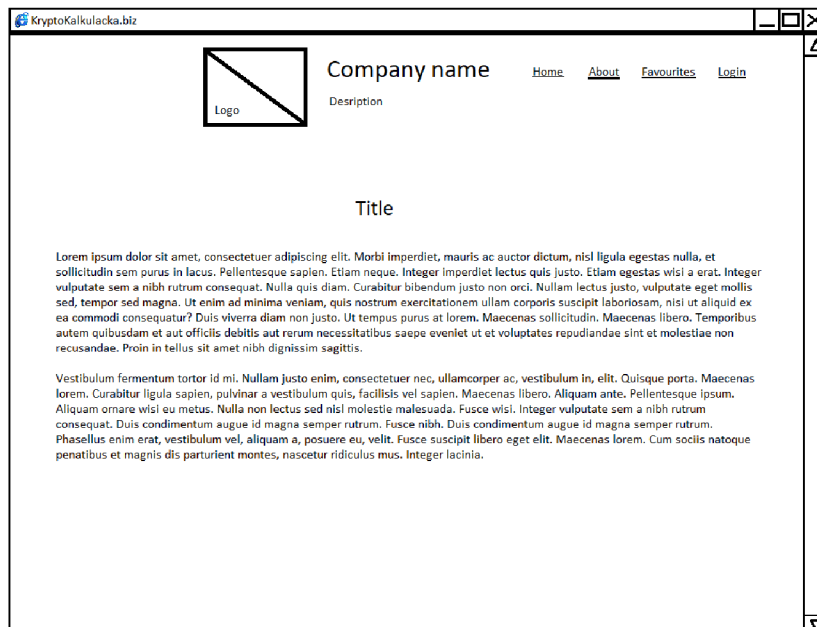
Obrázek 2: Návrh desktopového rozhraní

Následující model reprezentuje pokračování úvodní stránky, které nabízí podrobnější informace o vybraných měnách, napravo prostřednictvím údajů v tabulce. Nalevo pomocí grafu. Tyto prvky jsou pouze informativní a neumožňují další ovládání webu.



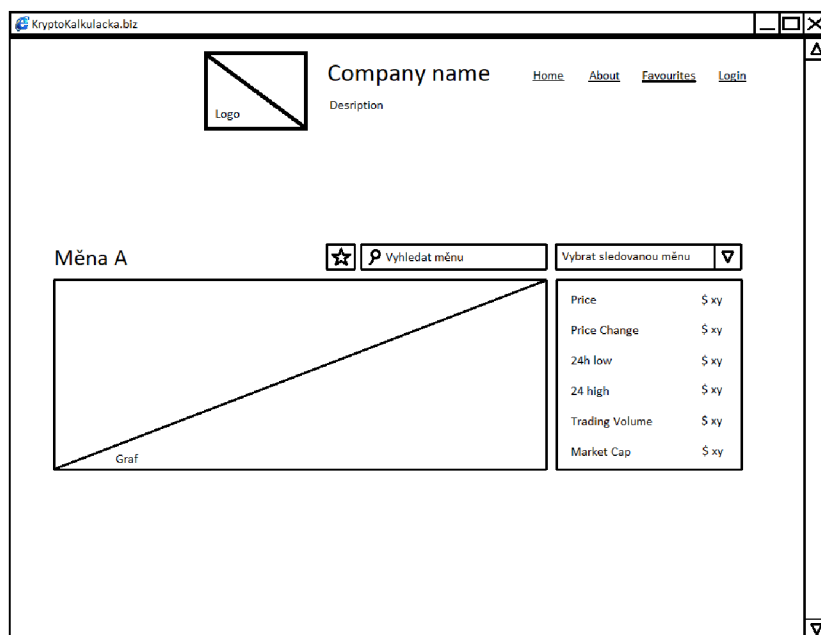
Obrázek 3: Pokračování úvodní stránky

Následující model je ukázkou textového obsahu stránky. Uplatnění by mohl nalézt při popisu kryptoměn či jiného sdělování informací.



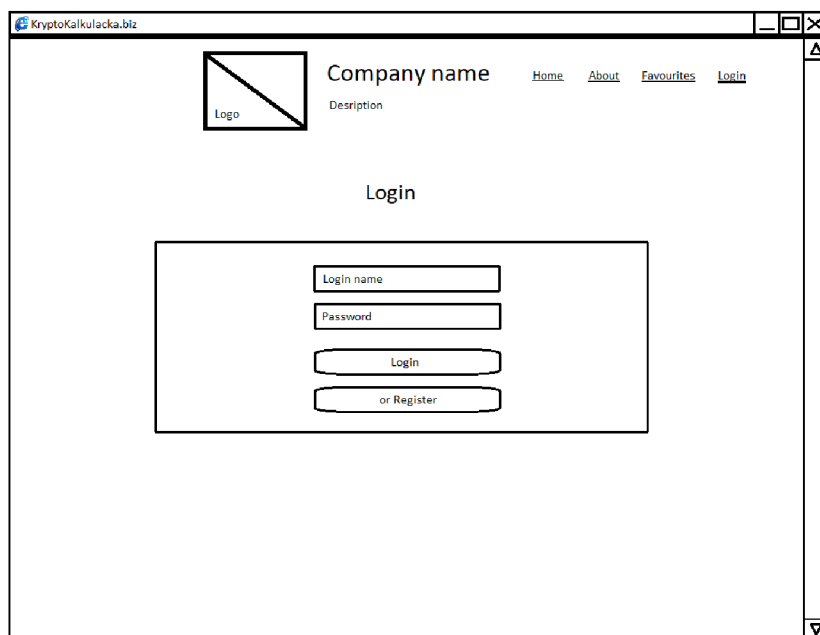
Obrázek 4: Návrh textového rozložení

Následující model ukazuje rozvržení uživatelského rozhraní v případě sledování pouze jedné měny v detailu. Zde se nachází textové pole pro vyhledávání a rozbalovací seznam pro výběr měny ze seznamu oblíbených.



Obrázek 5: Návrh uživatelského rozhraní

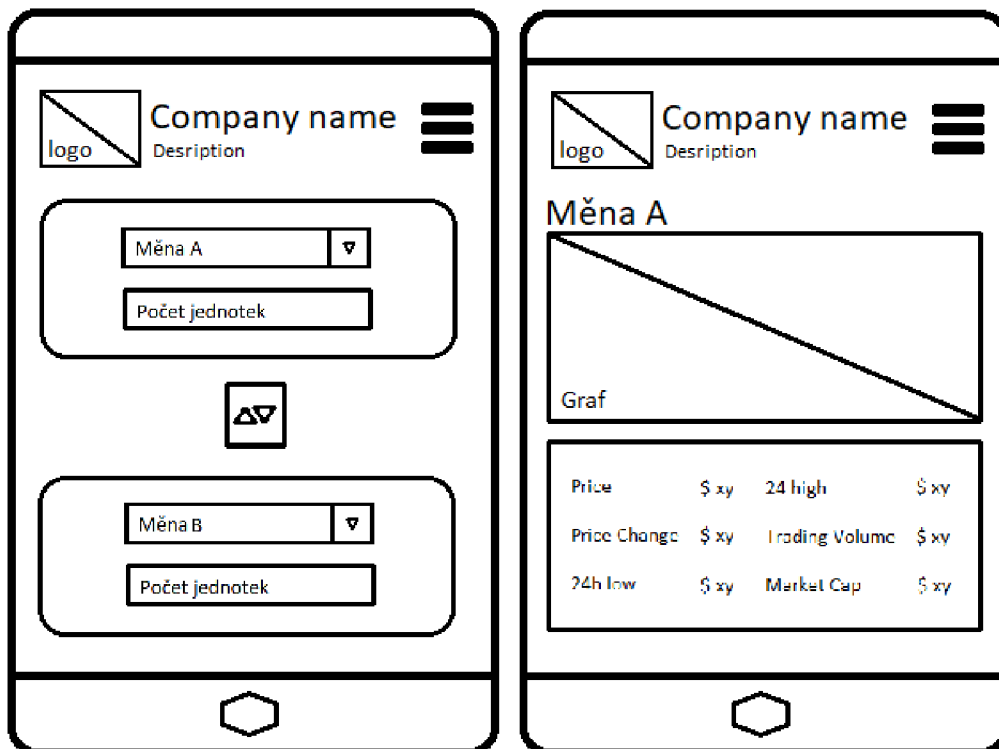
Poslední model reprezentuje přihlašovací obrazovku, nachází se zde formulář se vstupními poli pro zadání přihlašovacích údajů, potvrzovací tlačítko a tlačítko pro přesun na stránku s registrací.

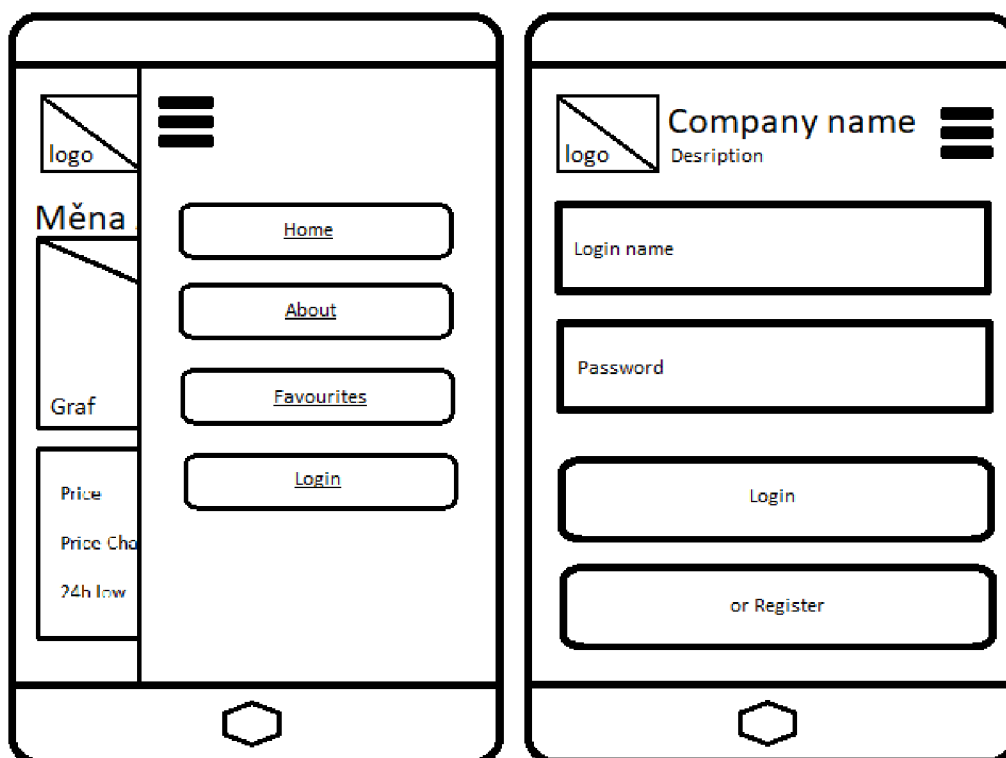


Obrázek 6: Návrh přihlašovací obrazovky

4.1.2.2 Mobilní zařízení

Následující modely zobrazují shodné části webu, avšak je zde kladen důraz na responzivitu. Popis funkcionality bude vynechán, jelikož se nijak zásadně neliší od verze pro Desktop. Za zmínku stojí pouze tzv. Hamburger menu, pod kterým se skrývá hlavní menu.





Obrázek 7: Soubor návrhů mobilního rozhraní

4.1.3 Vývoj vlastního webu

Tato část práce je zaměřena na vývoj frontendové části webu, což je klíčový prvek pro poskytování uživatelsky přívětivých a interaktivních webových zkušeností. Pro tento účel byla využita řada technologií a nástrojů, které jsou široce používány ve webovém vývoji. HTML a CSS tvoří základ pro strukturu a prezentaci webových stránek, dále Bootstrap, populární CSS framework, umožnil rychle a snadno navrhnout responzivní a atraktivní design.

Javascript, univerzální skriptovací jazyk pro webové aplikace, byl klíčový pro vytváření dynamických a interaktivních funkcí, které zajišťují lepší uživatelské zkušenosti. jQuery, knihovna založená na Javascriptu poté usnadnila práci s DOM (Document Object Model) a představuje dobrý základ pro budoucí práci s AJAXem k asynchronní komunikaci se serverem. Drátěné modely (wireframy) byly důležitou součástí procesu návrhu, které pomohly vizualizovat layout a strukturu webových stránek před samotným kódováním. Kromě praktických dovedností byly díky teoretické části práce získány důležité poznatky o zásadách uživatelského rozhraní (UI) a uživatelské zkušenosti (UX), které byly následně aplikovány při vývoji frontendové části.

4.1.3.1 Popis základní struktury

Pro položení základů layoutu webových stránek byl využit grid systém Bootstrapu. Grid systém je základní strukturální komponenta Bootstrapu, která umožňuje rychlé a jednoduché vytváření responzivních rozvržení. Bootstrap využívá flexibilní systém s dvanácti sloupci, který se snadno přizpůsobuje různým velikostem obrazovek a zařízení.

Tento systém umožňuje definovat rozvržení stránky pomocí řádků (rows) a sloupců (columns), které lze přizpůsobit potřebám konkrétního projektu. Sloupce lze dále rozdělit do menších částí a přizpůsobit jejich velikost a pozici dle potřeb. Díky tomu je možné vytvořit různorodé a responzivní layouty pro různá zařízení, což zajišťuje optimální zobrazení a použitelnost webových stránek napříč širokou škálou obrazovek a prohlížečů.

Použitím grid systému Bootstrapu je tedy možné vytvořit konzistentní a udržitelný základ webových stránek a usnadnit jejich budoucí údržbu a aktualizace.

4.1.3.2 Popis vlastního HTML kódu

Základní strukturu HTML stránky tvoří hlavička `<head>`, která definuje titulek webu, favico, znakovou sadu, načítá externí css styly, knihovny bootstrapu, jquery a vlastní javascript soubory. Dále stojí za zmínku meta tag `<meta name="viewport" content="width=device-width, initial-scale=1" />` který zajišťuje, že webová stránka bude responzivní a přizpůsobí se různým velikostem obrazovek a zařízení.

Zbylou část základní struktury tvoří tag `<body>`, ve kterém je obsažen veškerý kód stránky. Jeho struktura se dále dělí na `<header>` a tzv. container.

Obsahem již zmíněného headeru je úvodní část webu, která zahrnuje logo, nadpis webu a menu. Také se zde nachází kód modálního okna pro přihlášení, které je však defaultně skryto.

```
<nav class="navbar navbar-expand-lg navbar-dark">
  ...
</nav>
```

Tato část kódu vytváří navigační menu s odkazy na různé části webové stránky. Navigační menu se automaticky zmenší na hamburger menu na malých obrazovkách díky třídě "navbar-expand-lg", "navbar-dark" je další třída Bootstrapu, která probarví menu do odstínů tmavé šedi.

```
<div class="modal fade" id="loginModalForm" tabindex="-1" aria-
label="Přihlašovací okno" aria-hidden="true">
  ...
</div>
```

Tato část kódu vytváří přihlašovací formulář, který se zobrazí v modálním okně v momentě, kdy uživatel klikne na tlačítko "Přihlásit" a je opět součástí headeru.

Grid systém Bootstrapu je založen na flexboxu a zajišťuje flexibilní a responzivní layout pro stránky. Grid se skládá z následujících komponent:

- `container` nebo `container-fluid`: Třída obalovacího elementu, která omezuje šířku obsahu na různých velikostech obrazovek a zarovnává obsah na střed.
- `row`: Třída řádku, která zajišťuje správné zarovnání a rozložení sloupců. Řádky jsou umístěny v kontejneru.
- `col-*`: Třídy sloupců, které definují šířku sloupců pro různé velikosti obrazovek. Číslo za `col-` určuje, kolik sloupců ze 12 sloupců gridu bude daný prvek zaujímat.

Ve vlastním kódu lze použití popsat následovně:

Nejvyšší úroveň:

- `.container-fluid` pro horní panel a `.container` pro kalkulačku.

Druhá úroveň:

- `.row` pro zarovnání a rozložení sloupců uvnitř kontejnerů.

Třetí úroveň:

- Sloupce s třídami `col-*` pro různé komponenty, jako jsou logo, nadpis, navigace, měna A, měna B a obrácení hodnot mezi nimi.

Jako ukázka použití grid systému byl vybrán samotný obsah webu, tedy `container` v rámci tagu `body`.

```
<div class="container">
  <div class="row mt-5"></div>
  <div class="row align-items-center">
    <div class="col-md-5">
      ...
    </div>
    <div class="col-md-2">
```

```

        ...
    </div>
    <div class="col-md-5">
        ...
    </div>
</div>
<div class="row mb-5"></div>
</div>

```

V tomto příkladě je celková šířka mřížky rozdělena do 12 sloupců. Grid systém je použit pro rozdělení krypto kalkulačky na tři sloupce:

Třídy sloupců `col-*` zahrnují předpony, které specifikují chování sloupců na různých úrovních zlomových bodů (breakpoints):

- `col-xs-*` (není v Bootstrap 4) pro obrazovky menší než 576px.
- `col-sm-*` pro obrazovky 576px a větší.
- `col-md-*` pro obrazovky 768px a větší.
- `col-lg-*` pro obrazovky 992px a větší.
- `col-xl-*` pro obrazovky 1200px a větší.

Ve vlastní kódu je použita třída `col-md-5` pro sloupce měny A a měny B, což znamená, že na středních obrazovkách (od 768px) a větších budou tyto sloupce zabírat 5 ze 12 sloupců gridu. V případě malých obrazovek (do 767px) nebyla použita žádná konkrétní třída sloupce, takže sloupce budou zabírat 100% šířky kontejneru a řadit se pod sebe, čímž se přizpůsobí mobilním zařízením.

Sloupec pro obrácení hodnot mezi měnou A a B používá třídu `col-md-2`, která zajišťuje, že na středních obrazovkách (od 768px) a větších bude tento sloupec zabírat 2 ze 12 sloupců gridu. Stejně jako u sloupců měny A a měny B, pro malé obrazovky nebyla použita žádná konkrétní třída sloupce, tedy opět zabere celý řádek.

Měna A:

Kód pro měnu A se nachází v prvním divu s třídou `col-md-5`. Tento div obsahuje karty (cards) Bootstrapu, které zahrnují hlavičku karty a tělo karty.

Hlavička karty obsahuje formulářový prvek `input`, který slouží k výběru měny A. Je zde také `datalist` s ID `datalistMenaA`, který obsahuje seznam měn. Seznam měn je generován dynamicky pomocí JavaScriptové funkce `naplnDatalist()`.

Tělo karty obsahuje další formulářový prvek `input`, tentokrát pro zadání počtu jednotek měny A. Tento `input` je spojen s `` elementem, který zobrazuje název vybrané měny A.

Měna B:

Kód pro měnu B je velmi podobný kódu pro měnu A a nachází se v dalším divu s třídou `col-md-5`. Opět jsou zde karty s hlavičkou a tělem karty.

Hlavička karty obsahuje formulářový prvek `input` pro výběr měny B a `datalist` s ID `datalistMenaB`, který obsahuje seznam měn generovaný opět pomocí `naplnDatalist()`.

Tělo karty obsahuje formulářový prvek `input` pro zadání počtu jednotek měny B. Tento `input` je spojen s `` elementem, který zobrazuje název vybrané měny B.

Prvek uprostřed pro obrácení hodnot mezi měnou A a B:

Mezi divy pro měnu A a měnu B se nachází div s třídou `col-md-2`. V tomto divu je obrázek šipky, který slouží jako tlačítko pro otočení hodnot mezi měnou A a B. Při kliknutí na tento obrázek se zavolá JavaScriptová funkce `arrowBtn()`.

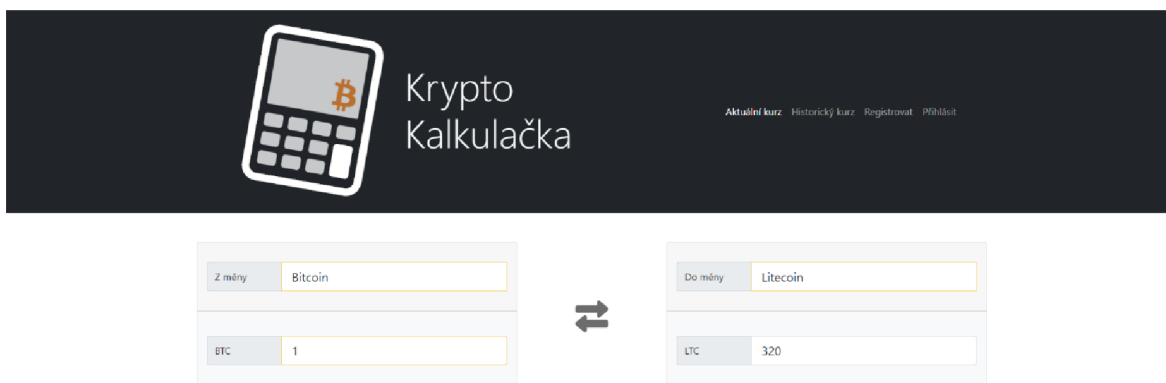
Funkce `arrowBtn()` zajišťuje prohození hodnot mezi měnou A a měnou B. Tato funkce by měla obsahovat logiku, která zajistí, že hodnoty zadané ve formulářových prvcích pro měny A a B se vymění a aktualizují na obrazovce.

Grid systém Bootstrapu usnadňuje responzivní design a zajišťuje, že aplikace pro převod měn bude vypadat a fungovat dobře na různých zařízeních a velikostech obrazovek.

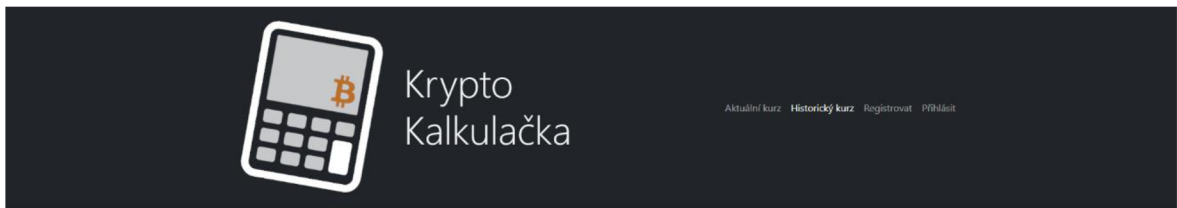
Součástí realizace prototypu webu byla stránka pro zobrazení historického kurzu, formulář registrace, informaci o uživateli a jeho oblíbených měnách. Vzhledem k možnostem rozsahu práce a aplikování do značné míry shodného kódu jim nebude věnován podrobný popis a budou následovat rovnou ukázky.

4.1.3.3 Ukázka vlastního návrhu

Desktop

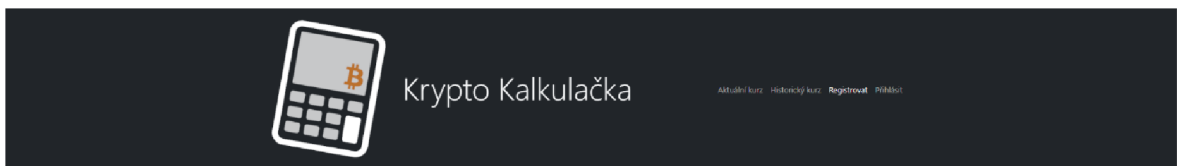


Obrázek 8: Realizace: Úvodní stránka



Historický kurz, přepočten na základě libovolně starých dat.

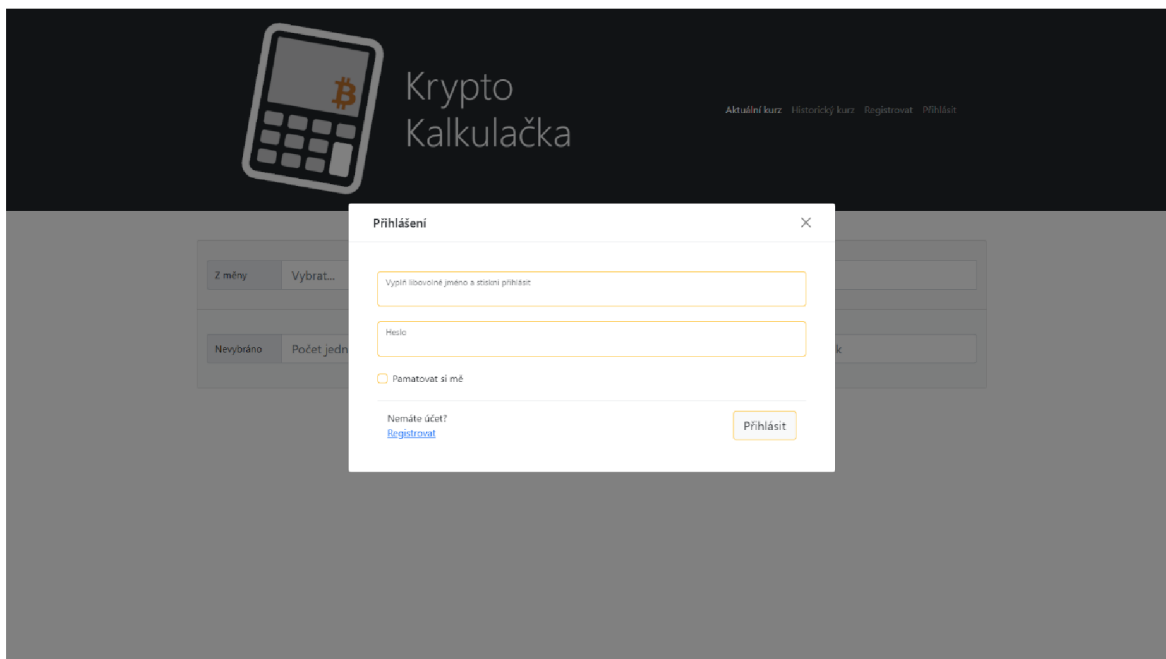
Obrázek 9: Realizace: Kalendář



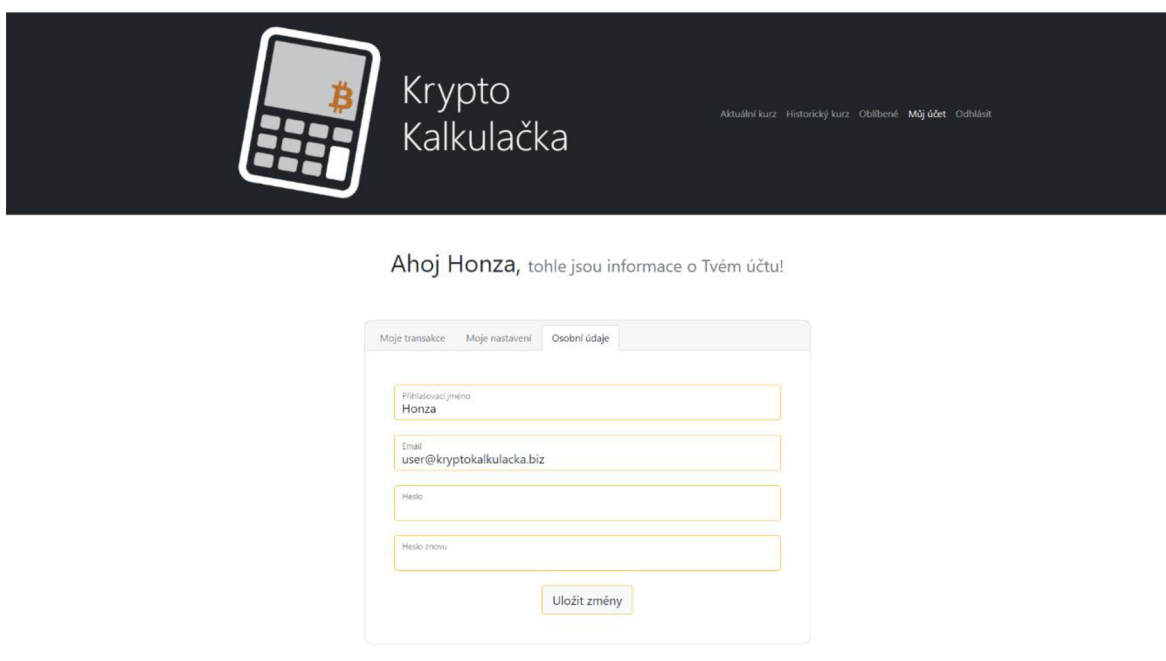
Registrace, získáte výhody přihlášeného uživatele.

Přihlásit se'."/>

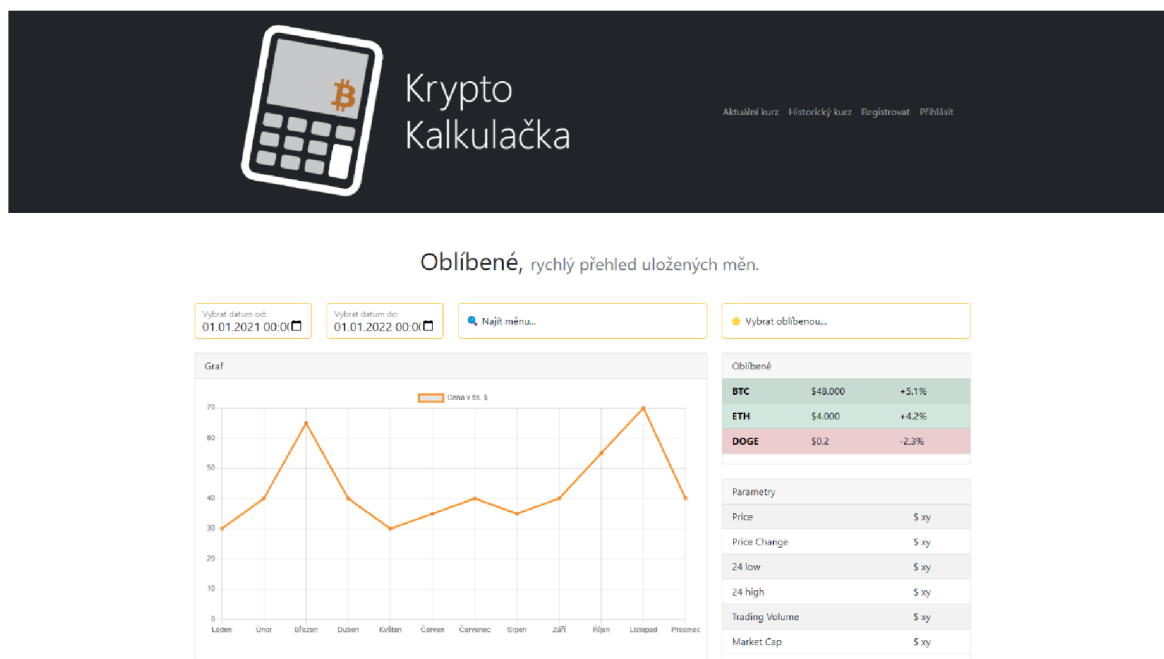
Obrázek 10: Realizace: Registrační formulář



Obrázek 11: Realizace: Okno přihlášení



Obrázek 12: Realizace: Uživatelský účet



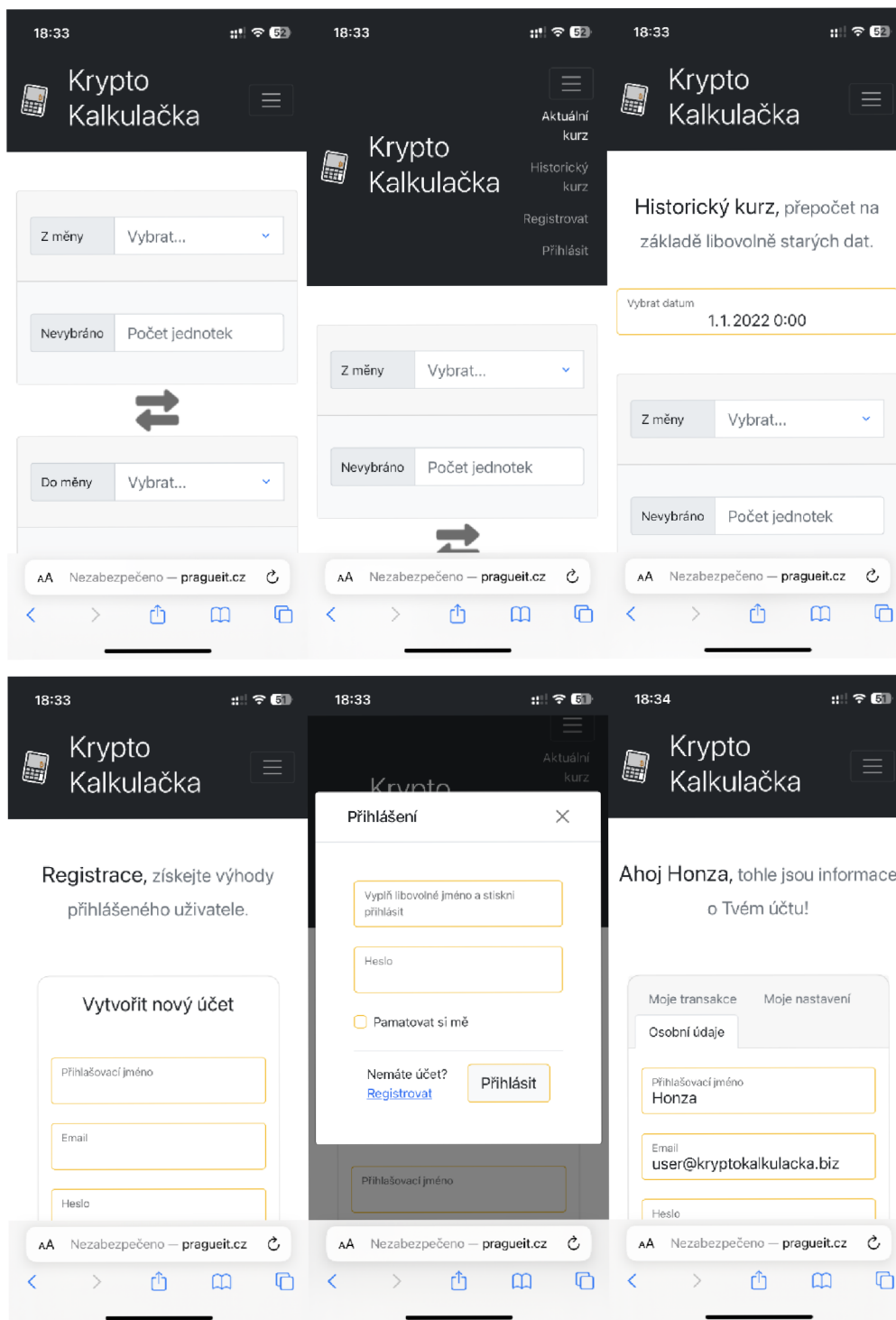
Obrázek 13: Realizace: Přehled oblíbených položek

V průběhu procesu vývoje webové stránky byly nejprve vytvořeny drátěné modely, které poskytovaly základní návrh struktury a uspořádání prvků na stránce. Tyto modely sloužily jako výchozí bod pro další práci na layoutu a designu webové stránky.

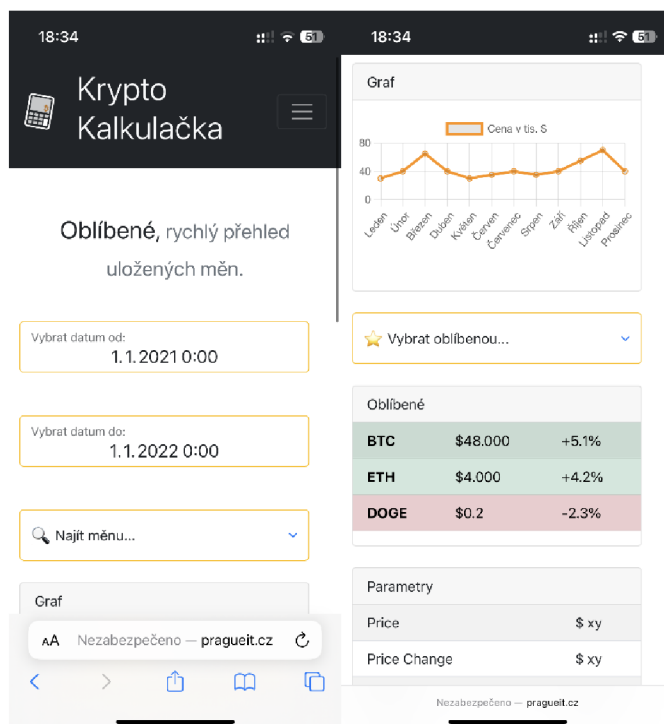
Přestože konečná podoba layoutu webové stránky v mnoha ohledech odpovídá původnímu drátěnému modelu, v průběhu vývoje došlo k několika odchylkám. Tyto odchylky byly výsledkem dalšího zamyšlení nad efektivitou, funkcionalitou návrhu a optimalizací designu, což vedlo k úpravám původního návrhu.

Vzhledem k těmto faktorům bylo nevyhnutelné, že konečný layout webové stránky se mírně liší od původního drátěného modelu. Nicméně, tyto odchylky byly výsledkem snahy o zlepšení uživatelského zážitku a zefektivnění designu webové stránky.

Mobilní zařízení



Obrázek 14: Realizace: Mobilní zobrazení



Obrázek 15: Realizace: Mobilní zobrazení – pokračování

4.1.3.4 Popis vlastního Javascript kódu

Javascript zde plní funkci přepočtu měn, kontrolu zadaných údajů a v kombinaci s jQuery také dynamickou manipulaci s DOM. Přepočet na straně klienta, tedy pomocí Javascriptu byl vybrán z toho důvodu, aby nedocházelo k nadbytečnému přetěžování serveru vysokou mírou požadavků.

Úvod této části bude věnován datalistu, který našeptává názvy kryptoměn, to slouží nejen jako přehled dostupných měn ke konverzi, ale také k urychlenému dokončení psaní.

Údaje o měnách jsou čerpány z pole `menyArr` a pomocí funkce `naplnDatalist` je generován seznam kryptoměn.

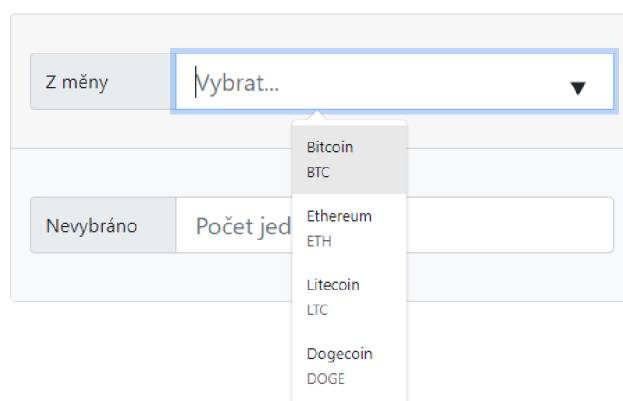
```
function naplnDatalist() {
    // Startujeme od klíče 1, dolar přeskakujeme
    let i = 1;
    // Cyklem vypíšeme vybrané hodnoty pole, stop hodnota je počet
    klíčů
```

```

while (i < Object.keys(menyArr).length) {
    document.write('<option value="' + menyArr[i]['celyNazev'] +
'" label="' + menyArr[i]['zkratka'] + '>');
    i += 1;
}
}

```

Během procházení pole `menyArr` je vytvářen řetězec HTML se značkami `<option>` pro každou kryptoměnu a to až do projití celého obsahu pole. Současně je tento řetězec vkládán do datalistů s ID "datalistMenaA" a "datalistMenaB" jako jednotlivé položky seznamu.



Obrázek 16: JavaScript výběr ze seznamu

Kontrola vstupních dat, tedy jak měny, tak počtu jednotek je prováděna současně během zadávání. K vybrání měny pro přepočítání dojde pouze v případě korektního zadání názvu, který se shoduje s měnami obsaženými v poli `menyArr` a propíše se přidáním její zkratky k počtu jednotek. Zadané údaje jsou kontrolovány během každého stisku klávesy či jiné změny, tím je zamezeno podsunutí nesprávných dat. Svou roli zde hraje také pole `prepocetArr`, které obsahuje údaje určené pouze ke konverzi získané pomocí formuláře.

```

// Po zvolení měny připiše zkratku měny před input "počet
jednotek" měny
$('#inputMenaA, #inputMenaB').change (function() {
    // Kontroluje, zda se obsah inputu stále shoduje se
selectováním názvem, aby s tam nedalo podsouvat kravinu
    if ($('#inputMenaA').val() === getVybranyNazev($('#inputMenaA'))) {
        // Při úspěšném průběhu kontroly zapíše zkratku měny před input
        $('#vybranaMena'+$('#inputMenaA').val()).text(getVybranaZkratka($('#inputMenaA').val()));
    }
});

```



```

        // Ještě nezapomeneme porovnat, jestli nejsou zvoleny
stejně měny, pokud ano, vynadáme
        if ($("#vybranaMenaA").text() !==
$("#vybranaMenaB").text()) {
            // Po úspěšné změně a zvolení nestejných měn
            // odbarvíme rámečky
            setWarning("#inputMenaA, #inputMenaB");
            // a přejdeme ke kontrole (pošle element id a jeho
hodnotu)
            kontrola("#vybranaMena"+getIdAB(this),
getVybranaZkratka(getId(this)));
        } else {
            // Obarvím rámeček na rudo
            setDanger("#inputMenaA, #inputMenaB");
        }
    } else { // defaultne nebo pri neshode
        $("#vybranaMena"+getIdAB(this)).text("Nevybráno");
        prepočetArr["mena"+getIdAB(this)]= "";
    }
});

```

Tato ukázka kódu slouží k výše zmíněné kontrole zadané měny, za zmínku stojí také funkce `setDanger` a `setWarning`.

```

function setDanger(id) {
    // Červená barva ohraničení
    $(id).removeClass("border-warning");
    $(id).addClass("border-danger");
}

function setWarning(id) {
    // Oranžová barva ohraničení
    $(id).removeClass("border-danger");
    $(id).addClass("border-warning");
}

```

Názvy jsou odvozené od tříd užívaných v Bootstrapu a nastavují oranžové ohraničení při správném zadání či výchozím stavu nebo červené v případě nekorektního zadání. Aby nedocházelo ke zbytečné redundanci kódu, identifikátor elementu je doplňován dynamicky. Identifikátor je získáván pomocí následující funkce

```

function getId(e) {
    return "#" + $(e).attr("id");
}

```

Máme-li korektně zvoleny obě měny, je možný přepočet, ten není třeba nijak potvrzovat, dochází k němu průběžně po každém jednotlivém zadání číslice a jeho výsledek je zobrazován v protějším poli. V úvahu se také bere úprava hodnoty v protějším poli s výsledkem a přepočet následně probíhá opačným směrem. Tomu ovšem předchází kontrola zadané číselné hodnoty inputu, následující kód slouží jako ukázka, jak toho bylo dosaženo. Jako u předchozích ukázek i zde je kód důsledně komentován pro lepší porozumění čtenáře.

```

// Hlídá zadávanou hodnotu v inputPocetA a inputPocetB
$('#inputPocetA, #inputPocetB').on("input keydown mousedown
mouseup select contextmenu drop", function() {
    // Odchytíme id políčka pro nějaký další využití
    var id = getId(this);
    // Preventivně přehodíme tečku za čárku
    $(this).val($(this).val().replace(".", ","));
    // Rozhodujeme, co budeme dělat na základě výsledku
testování vstupu
    if (testFloatNum(this.value)) { // Když true (tedy validní
vstup)
        this.oldValue = this.value; // Nastavíme odchycenou
hodnotu jako zadanou
        this.oldSelectionStart = this.selectionStart; // hraje
si s umístěním kurzoru
        this.oldSelectionEnd = this.selectionEnd;
        setWarning(id);
        // Zahodím okýnko
        popHide(id);
        // Test sice false, ale nechceme smazat již zadané znaky
    } else if (this.hasOwnProperty("oldValue")) {
        this.value = this.oldValue; // přepíšeme aktuální
hodnotu přechází (vrátíme zpět zadání)
        this.setSelectionRange(this.oldSelectionStart,
this.oldSelectionEnd); // kurzor
        setDanger(id);
        // Doplním třídu okýnka pro možné vyobrazení
        popShow(id);
        // Neorektní vstup (1. zadávaný znak)
    } else {
        this.value = "";
        setDanger(id);
        // Taky nadávací okýnko
        popShow(id);
    }
});

```

```

// Předpokládejme, že keydown osetřila radně vstup a můžeme udaje
// předat ke kontrole při keyup
$('#inputPocetA, #inputPocetB').on("keyup", function() {
    // Pro num není dobrá čárka
    var hodnota = $(this).val().replace(",", ".");
    // Jenom mrknem, jestli zbytečně nepošíláme ""
    if (hodnota !== "" && hodnota > 0) {
        // id, vlauce
        kontrola(getId(this), hodnota);
    } else {
        // Pokud je prázdná či nulová, odmazeme obsah protejsi
        strany
        $(getOpacnyPrvek(getId(this))).val("");
    }
});

// Vrací bool podle úspěšnosti testu, co dal přemysleme jinde
function testFloatNum(value) {
    return /^\d*[.,]?\d*$/.test(value);
}

```

Chybně zadaný znak je skriptem ihned odstraněn, není tedy možné pokračovat v nesmyslném zadání, zároveň však nedochází ke smazání celého obsahu inputu, tím nedochází ke zbytečnému trestání uživatele, který se pouze překlepl.

V uvedeném kódu je také možné pozorovat práci s tooltipy (popover), doplňkem Bootstrapu. Jejich účelem zde je informovat uživatele o zadání nesprávně hodnoty. V rámci html se můžeme setkat s touto podobou.

```

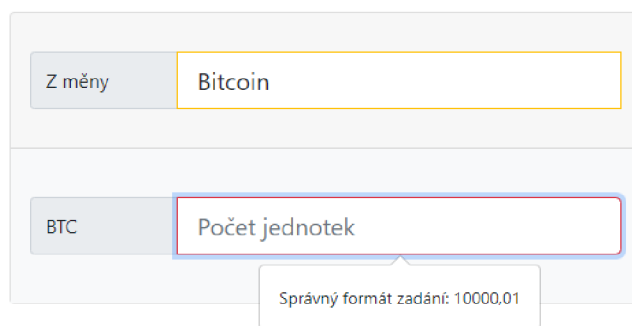
<input class="form-control form-control-lg"
    type="text"
    inputmode="decimal"
    pattern="\d*"
    id="inputPocetA"
    data-bs-content="Správný formát zadání: 10000,01"
    data-bs-placement="bottom"
    placeholder="Počet jednotek"
    aria-label="Počet jednotek měny A" />

```

Zobrazení či skrytí tooltipu je realizuje následující funkce, která je dle potřeby volána.

```
function popShow(id) {
    // Doplním třídu okýnka pro možné vyobrazení
    $(id).attr('data-bs-toggle', 'popover');
    // Zobrazím popover s chytrou hláškou
    $("[data-bs-toggle='popover']").popover('show');
}

function popHide(id) {
    // Odeberu popover
    $('.popover').remove();
    // Seberu třídu forcasse
    $(id).removeAttr('data-bs-toggle');
}
```



Obrázek 17: Javascript kontrola pole

Po kontrole vstupních polí je na řadě poslední kontrola hodnot v poli `prepocetArr` a následný přepočet, ten je realizován takto:

```
// Spustíme prepocet pouze v momente, kdy máme zadane měny a alespon jeden z počtů není undefined nebo prazdna
if ((typeof prepocetArr["menaA"] !== "undefined" && prepocetArr["menaA"] !== "") &&
    (typeof prepocetArr["menaB"] !== "undefined" && prepocetArr["menaB"] !== "") &&
    ((typeof prepocetArr["pocetA"] !== "undefined") || (typeof prepocetArr["pocetB"] !== "undefined"))) {
    // Nojo, jenže co když nějaký undefined je
    if (typeof prepocetArr["pocetA"] == "undefined") {
        // Pokud strana A nebyla definovana, nastala vyjimka, kdy otacime strany
        prepocet("B", "A");
    } else if (typeof prepocetArr["pocetB"] == "undefined") {
```

```

        // Pokud strana B nebyla definována, nastala vyjimka,
kdy otacime strany
        prepocet("A", "B");
    } else {
        /* V případě, byly již definovány obě strany voláme
prepocet dynamicky
        * pošle tedy poslední písmenka inputů původce (aktuálně
zadávaný) a cíl (protilehlý)
        * funguje i v případě změny měny
        */
        prepocet(getIdAB(element),
getIdAB(getOpacnyPrvek(element)));
    }
}

// Prepocet co se nelekne prohození stran ci vyberu jine meny
function prepocet(odkud, kam) {

$("#inputPocet"+kam).val((getCena(prepocetArr["mena"+odkud])*prepocetArr[
"pocet"+odkud])*(1/getCena(prepocetArr["mena"+kam])));
}

```

Jelikož je možné upravovat hodnoty u obou měn, je nezbytné vždy správně identifikovat, která je upravovaná a kam je třeba umístit výsledek. Identifikátor upravované lze zjistit jednoduše díky zachytávané události. Následně je určena protější strana pomocí jednoduché funkce.

```

function getOpacnyPrvek(id) {
    let opacnyPrvek;
    if (id.slice(-1) === "A") {
        opacnyPrvek = id.slice(0, -1)+"B";
    } else if (id.slice(-1) === "B") {
        opacnyPrvek = id.slice(0, -1)+"A";
    }
    return opacnyPrvek;
}

```

Funkce `arrowBtn` prohazuje hodnoty vstupních polí `"inputMenaA"` a `"inputMenaB"` a odpovídající textové elementy `"vybranaMenaA"` a `"vybranaMenaB"`. K tomu nejprve uloží hodnoty vstupních polí a textových elementů do dočasných proměnných. Poté nastaví hodnoty vstupních polí a textových elementů na hodnoty dočasných proměnných opačného pole (A na B, B na A).

4.1.4 Zpětná analýza

Cílem této části práce je podat přehled o jednotlivých testech, které byly provedeny, konkrétně se jedná o test použitelnosti, přístupnosti, rychlosti a validity kódu.

Test použitelnosti se zaměřil na hodnocení, jak snadno a intuitivně mohou uživatelé aplikace Krypto Kalkulačka ovládat. Byly identifikovány potenciální problémy v rozhraní a navrhnutá, případně realizována možná zlepšení.

Test přístupnosti se zaměřil na kontrolu, zda aplikace Krypto Kalkulačka splňuje standardy přístupnosti a je dostupná co nejširšímu spektru uživatelů, včetně těch s různými druhy zdravotních omezení. Byly prověřovány například kontrasty barev, velikost písma, navigace klávesnicí a alternativní texty pro obrázky.

Rychlostní test měl za úkol ověřit, jak rychle se aplikace Krypto Kalkulačka načítá a reaguje na uživatelské akce.

Validita kódu byla ověřena prostřednictvím testů, které kontrolovaly, zda zdrojový kód aplikace splňuje standardy a konvence pro kvalitní a udržitelný kód. Byly prověřovány syntaxe, sémantika, konzistence kódu a používání komentářů, což pomohlo identifikovat a opravit chyby či nedostatky v kódu.

4.1.4.1 Použitelnost

Pro projekt Krypto Kalkulačky byly hlavními parametry pro výběr vhodné testovací metody zvoleny časová a finanční nenáročnost. Z toho důvodu byla vybrána heuristická evaluace jako expertní testování bez uživatele a současně bude zmíněno pouze nutné minimum pro pochopení testovací metody a pro snadnější a korektní interpretace výsledků.

- Výběr částí klienta, které je důležité otestovat.
- Výběr použitých heuristik.
- Vytvoření seznamu typických úkolů, které uživatel řeší pomocí testované aplikace.

- Vykonání jednotlivých úkolů. U každého průchodu zaznamenat konflikt s heuristikami. Přiřazení významnosti dopadu zaznamenaných konfliktů z pohledu typického uživatele.
- Vytvoření seznamu nálezů konfliktů pro všechny úkoly, tak aby je bylo možné efektivně prezentovat tvůrcům aplikace.

Heuristickou evaluací je možné testovat libovolnou část aplikace, u které má smysl sledovat její použitelnost, test byl aplikován na hlavní stránku s převodem měn. Cílem níže provedených testů je primárně uživatelské rozhraní, jelikož celková funkcionality aplikace je zatím omezena, protože její klíčová část, backend, ještě není implementována ani emulována [51] [52].

Použité heuristiky

Heuristika č.	Popis
1	Viditelnost stavu, ve kterém se systém nachází.
2	Shoda v popisu situací a stavů mezi systémem a reálným světem.
3	Uživatelská volnost pohybu aplikací a možnost návratu (např. podpora undo a redo).
4	Konzistence v dodržování konvencí a standardů.
5	Prevence chyb.
6	Rozpoznatelnost nad zapamatovatelnost. Zbytečně nezatěžovat paměť uživatele.
7	Flexibilita a efektivita používání.
8	Estetičnost a minimalistický design.
9	Pomoc uživatelům rozpoznat, diagnostikovat a zotavit se z chyb.
10	Nápověda a dokumentace.

Tabulka 1: Seznam použitých heuristik

Seznam typických úkolů

Č. úkolu	Popis úkolu	Častost
1	Porovnání kryptoměn	Vysoká
2	Přihlášení uživatele	Střední

Tabulka 2: Seznam typických úkolů

Výsledky testů

Test proběhl dne 6.1.2022 mezi 1500 a 1700 CET. Výsledky obsahují pouze popis odhalených konfliktů s pravidly 1 až 10 podle tabulky 1 z předchozí části. Nálezy jsou pro přehlednost uvedeny na konci této kapitoly v jedné společné tabulce.

Porovnání kryptoměn

Porovnání probíhá vyplněním jednoduchého formuláře. Formulář je bez nadpisu a uživatel je tak nucen rozpoznat stav, kde se nalézá podle zvýrazněné položky navigačního menu, která má malý font textu, což je možný konflikt s pravidlem 1. Výběr měny umožňuje uživateli vkládat i nesmyslný text bez přímé indikace chyby. Ta je nepřímo naznačena v popisku pole „Počet jednotek“ hodnotou nevybráno. Problém s pravidly č.5 a č.9.

Přihlášení uživatele

Přihlášení lze provést z hlavního menu. Přihlašovací formulář umístěn na modálním okně. Vstupní pole pro uživatelské jméno neobsahuje informaci o znacích přípustných v uživatelském jménu nebo v heslu. Je možné vložit i neobvyklé znaky, jak je patrné z obrázku číslo 4. Pravděpodobně jde o kolizi s pravidly 5, 9 a 10.

Po odeslání formuláře je otevřena stránka s informacemi o uživatelském účtu. Informace jsou zobrazeny pomocí formuláře, který lze editovat a který umožňuje tyto změny ukládat. Vstupní pole neindikují chybný vstup. Kolize s pravidly 5 a 9. Rovněž v hlavním navigačním menu zůstává zvýrazněná jiná položka, očekávaná zvýrazněná položka by byla „Můj účet“. Možná kolize s pravidlem 1, jejíž dopad mírní dostatečně viditelný pozdrav uživateli s informací o obsahu stránky. Po stisku tlačítka „uložit změny“ chybí indikace, zda byla operace provedena a s jakým výsledkem, což vede ke kolizi s pravidlem 1.

Celkový přehled nálezů

Id nálezu	Č. úkolu	Popis	Umístění	Dopad
1	1	Stav systému rozpoznatelný jen z navigačního menu s položkami napsanými malým písmem.	class="nav-link"	Nízký
2	1	Možnost vložit chybný text bez upozornění na chybu.	id="inputMenaA"	Střední
3	2	Možnost vložit nepřípustné znaky bez upozornění na chybu.	id="formPrihlLogin"	Střední
4	2	Možnost vložit chybný text bez upozornění na chybu.	id="formMujUcetEmail"	Střední
5	2	Zvýrazněná špatná položka menu „Aktuální kurz“.	class="nav-link"	Nízká
6	2	Nezvýrazněná položka menu „Můj účet“.	class="nav-link"	Nízká

Tabulka 3: Přehled nálezů heuristické analýzy

Analýza výsledků

Při testování části aplikace, která je stále ve vývoji a jejíž funkcionality odpovídá spíše prototypu, než hotové aplikaci, je nutné tuto skutečnost zohlednit při analýze výsledků. Většina nálezů má úroveň dopadu střední a nízkou.

Z výsledků je patrné, že nejčastějšími porušovanými principy byly heuristiky 1,5,9 a 10. Tedy pravidla, která se zabývají zobrazováním stavu, ve kterém se aplikace při plnění konkrétního úkolu nalézá, prevencí chyb a jejich indikací uživateli či nabídnutím návodu pro řešení.

Doporučení vyplývající z analýzy výsledků testů tedy je klást důraz na výše zmíněné principy při další tvorbě aplikace, zejména prvků navigace a formulářů pro vkládání údajů.

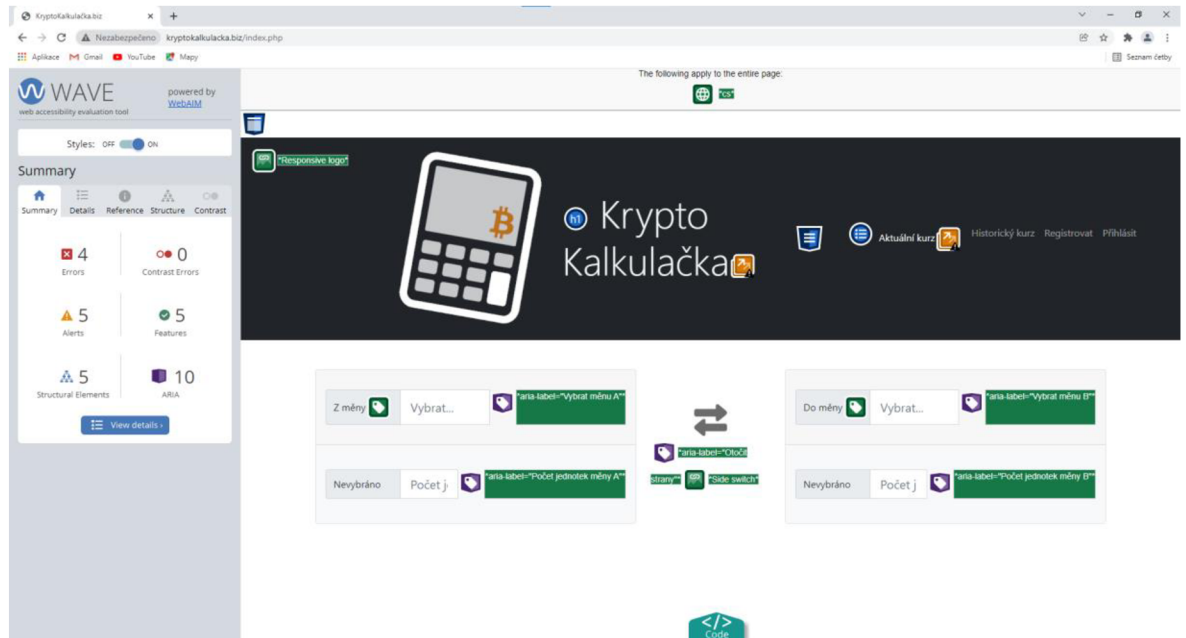
4.1.4.2 Přístupnost

Přístupnost je důležitou komponentou User Experience (UX). Aplikace by neměla klást překážky uživatelům při používání, naopak, měla by být vytvořena tak, aby umožnila přístup i uživatelům s různými hendikepy.

Vytváření aplikací s ohledem na přístupnost má široké spektrum pozitivních přínosů. Rozšíření množiny uživatelů může mít za následek více obchodních příležitostí, lepší viditelnost na internetu z hlediska SEO, dobré jméno a reklamu. Pro některé instituce je splnění podmínek přístupnosti nařízeno zákonem. V neposlední řadě jde i o respektování demografické reality, protože nezanedbatelné procento uživatelů internetu je hendikepovaných.

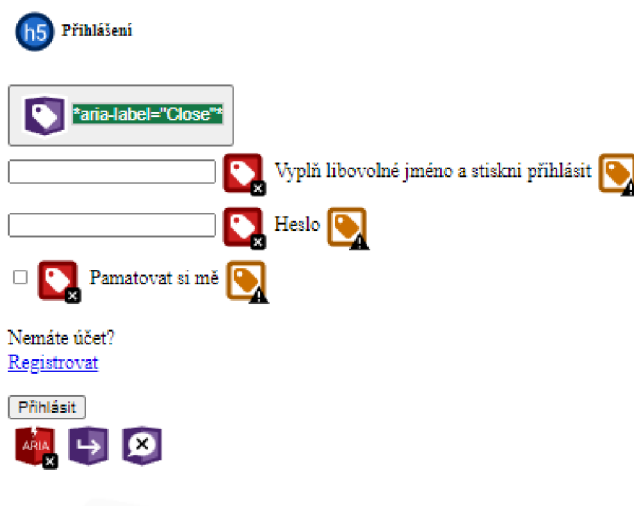
Existuje široká škála možností, jak testovat přístupnost. Pro testování relativně malé aplikace, jako je Krypto Kalkulačka, která je navíc stále ve stádiu vývoje klientské části, se jeví jako plně dostačující automatické testování pomocí k tomu vytvořeného nástroje a následné vytvoření reportu shrnujícího výsledky testu a návrhu opatření. Nástrojem pro testování byla zvolena aplikace WAVE Evaluation Tool. Test byl opět aplikován na hlavní stránce [53].

Výsledky testu



Obrázek 18: Výsledky testu WAVE s CSS

Nalezeny byly celkem 4 chyby a 5 výstrah. Všechny s nízkým dopadem, protože jsou viditelné pouze v případě vypnutých stylů (CSS).



Obrázek 19: Výsledky testu WAVE bez CSS

Nálezy pro úvodní stránku jsou shrnuty v následující tabulce.

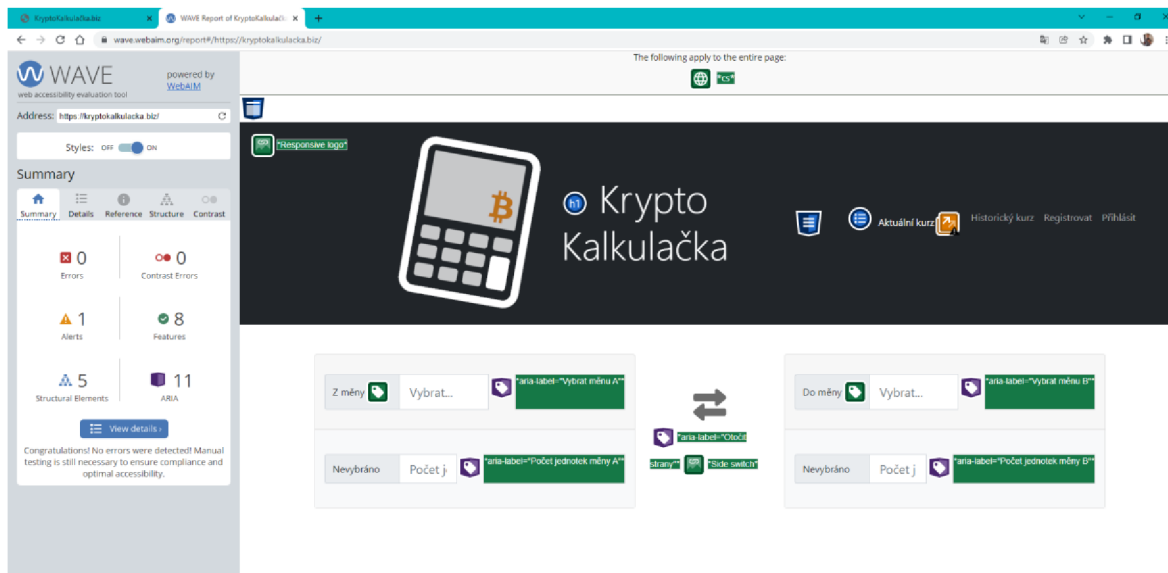
Id nálezu	Druh	Popis	Umístění	Dopad	Pozn.
1	Chyba	Chybějící popiska formuláře	id="formPrihlLogin"	Nízký	Bez CSS
2	Chyba	Chybějící popiska formuláře	id="formPrihlHeslo"	Nízký	Bez CSS
3	Chyba	Chybějící popiska formuláře	id="formPrihlPamatovat"	Nízký	Bez CSS
4	Chyba	Nefunkční odkaz ARIA	id="loginModalForm"	Nízký	Bez CSS
5	Výstraha	Popisek nesprávně asociován s prvkem formuláře	<label for="floatingInputValue"> Vyplň libovolné jméno a stiskni přihlásit </label>	Nízký	Bez CSS
6	Výstraha	Popisek nesprávně asociován s prvkem formuláře	<label for="floatingInputValue"> Heslo </label>	Nízký	Bez CSS
7	Výstraha	Popisek nesprávně asociován s prvkem formuláře	<label class="form-check-label" for="rememberMe"> Pamatovat si mě </label>	Nízký	Bez CSS
8	Výstraha	Nadbytečný odkaz	 Krypto Kalkulačka 	Nízká	

9	Výstraha	Nadbytečný odkaz	Aktuální kurz	Nízká	
---	----------	------------------	---	-------	--

Tabulka 4: Přehled nálezů WAVE analýzy

Testy přístupnosti odhalily 9 nálezů, z nichž většina má úroveň dopadu hodnocenu jako střední. Jde o chyby a výstrahy, které by mohly negativně ovlivnit zejména nástroje pro čtení obsahu obrazovky. Z hlediska standardů jde o úroveň priority A. V žádném případě tedy nejde o chyby zásadní a s přihlédnutím k faktu, že aplikace je stále ve stavu vývoje, kdy je k dispozici pouze část klienta, nelze o její konečné přístupnosti činit relevantní závěry.

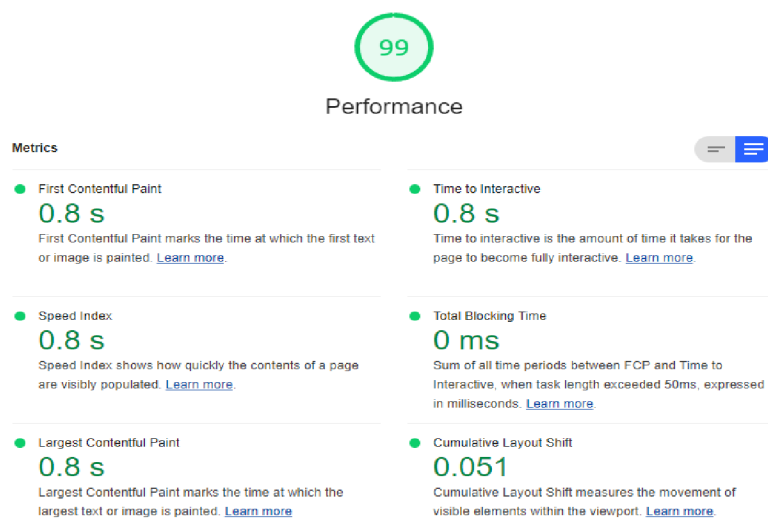
Oprava chyb



Obrázek 20: Výsledky testu WAVE po opravě chyb

4.1.4.3 Rychlost

Testování rychlosti aplikace je důležité, protože může nepřímo ovlivnit návštěvnost a tím pádem i její komerční úspěch. Pro získání alespoň orientační představy byl pro testování rychlosti použit nástroj Google Lighthouse [54].



Obrázek 21: Výsledky testu výkonnosti pomocí Google Lighthouse

Testování rychlosti aplikace v jejím současném stadiu vývoje má jen částečný význam. V současnosti se pracuje na implementaci klienta, který běží v internetovém prohlížeči. Serverová část ještě není implementována. Proto lze brát příznivé výsledky testu pouze jako orientační.

Z toho důvodu by bylo vhodnější provést test rychlosti aplikace, až bude implementována backendová část, nebo pokud bude vytvořen odpovídající mock backendu, který by ho co nejdříve emuloval.

4.1.4.4 Validita kódu

Tato část se zabývá validitou kódu aplikace Krypto Kalkulačka. Aplikace je implementována pomocí technologií HTML, CSS a jazyka Javascript. Validace kódu se bude primárně týkat HTML a CSS.

Z historického hlediska dokázaly a stále ještě dokážou webové prohlížeče interpretovat kód obsahující chyby. Z jejich pohledu jde o projev jisté robustnosti, který přispěl k růstu internetu. Proto by se mohlo zdát testování validity kódu téměř nadbytečné.

Problém je, že různé prohlížeče interpretují nevalidní kód různým způsobem. Proto je snaha o validitu způsobem, jak zajistit, aby byl kód v různých prohlížečích interpretován stejně. V neposlední řadě je validace také způsob, jak včas identifikovat chyby v kódu aplikace.

Validita kódu HTML je testována pomocí online nástroje Nu Html Checker [55].

Validita kódu CSS je testována pomocí online nástroje CSS Validation Service [56].

Nálezky HTML

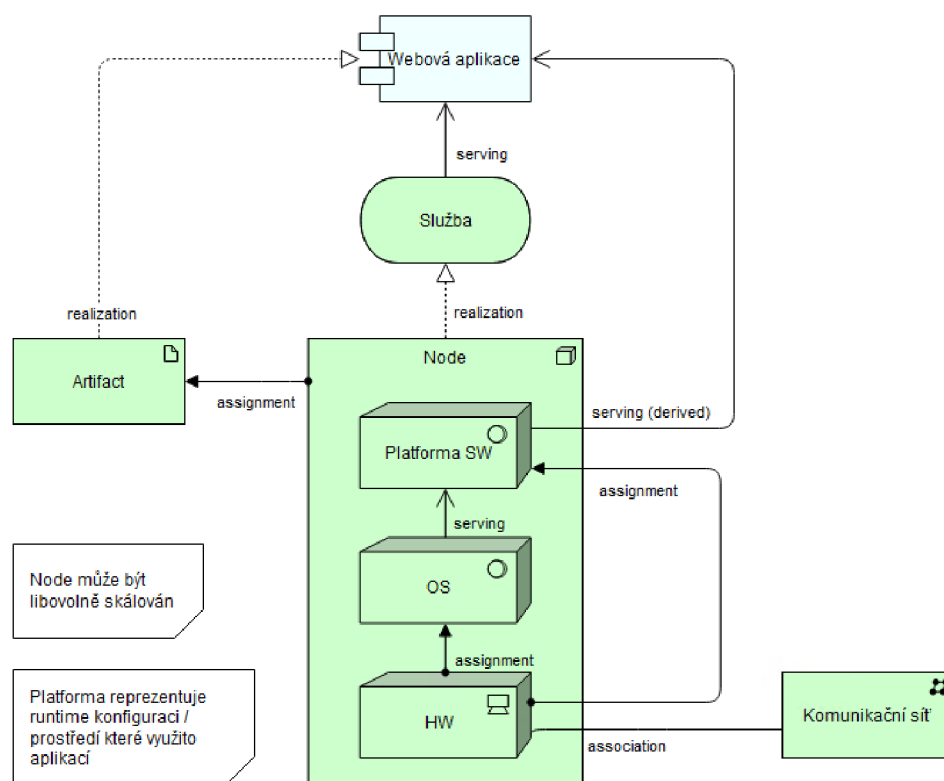
Č. nálezky	Popis	Umístění
1	Attribute min not allowed on element input at this point.	id="inputPocetA"
2	Attribute min not allowed on element input at this point.	id="inputPocetB"
3	The value of the for attribute of the label element must be the ID of a non-hidden form control.	<label for="floatingInputValue">Vyplň
4	The value of the for attribute of the label element must be the ID of a non-hidden form control.	<label for="floatingInputValue">Heslo
5	The value of the for attribute of the label element must be the ID of a non-hidden form control.	<label class="form-check-label" for="rememberMe">Pamato
6	The aria-labelledby attribute must point to an element in the same document.	id="loginModalForm"

Tabulka 5: Přehled nálezů HTML Validátoru

Nálezky CSS se nepodařilo nalézt, a to pravděpodobně díky nasazení Bootstrapu, jakožto odladěného nástroje.

4.2 Backend

4.2.1 Technologická architektura:



Obrázek 22: Technologická architektura

4.2.2 Architektura databáze

Tato část obsahuje návrh architektury a strukturu relační databáze. Předpokladem je, že bude použito SQL pro správu databáze. Zde je návrh tabulek a jejich atributů:

Tabulka users:

- user_id (integer, primární klíč, auto_increment)
- username (varchar, unikátní)
- email (varchar, unikátní)
- password_hash (varchar)
- registration_date (datetime)

Tabulka user_settings:

- settings_id (integer, primární klíč, auto_increment)
- user_id (integer, cizí klíč odkazující na users.user_id)
- preferred_currency (varchar)
- preferred_language (varchar)
- dark_mode (boolean)

Tabulka favorite_currencies:

- favorite_id (integer, primární klíč, auto_increment)
- user_id (integer, cizí klíč odkazující na users.user_id)
- currency_symbol (varchar)

Tabulka transactions:

- transaction_id (integer, primární klíč, auto_increment)
- user_id (integer, cizí klíč odkazující na users.user_id)
- from_currency (varchar)
- to_currency (varchar)
- conversion_rate (decimal)
- amount_converted (decimal)
- transaction_date (datetime)

Vztahy mezi tabulkami:

users má 1:1 vztah s user_settings přes user_id.

users má 1:M vztah s favorite_currencies přes user_id.

users má 1:M vztah s transactions přes user_id.

Tento návrh by měl pokrýt veškeré požadavky na databázi. Je třeba nezapomenout na správné indexování a ošetření bezpečnostních otázek, jako je například hashování hesel a ochrana proti SQL injection.

4.2.3 Popis používaných komponent

V Nette frameworku se pro zobrazení obsahu používá dvojice komponent, které se nazývají šablona a presenter. Každý presenter je třída, která obsahuje logiku pro obsluhu a zpracování požadavků od uživatele, zatímco šablona je šablonovací systém pro generování HTML obsahu na základě dat předaných z presenteru.

Šablona:

Šablona je soubor, který definuje strukturu a vzhled HTML výstupu, který se zobrazuje uživateli. Šablona může obsahovat proměnné, cykly, podmínky a další konstrukce, které umožňují dynamicky generovat obsah na základě dat předaných z presenteru. V Nette se pro šablony používá vlastní šablonovací systém Latte, který rozšiřuje syntaxi HTML o speciální značky a konstrukce pro práci s proměnnými a logikou.

Presenter:

Presenter je třída, která obsahuje logiku pro zpracování požadavků od uživatele a předávání dat do šablony. Presenter může obsahovat metody pro zpracování odeslaných formulářů, práci s databází a další logiku spojenou s aktuální stránkou nebo částí aplikace. Presenter také obsahuje metodu `render()`, která se volá při zpracování požadavku a je zodpovědná za předání dat do šablony. V Nette se presenter obvykle dědí od třídy `Nette\Application\UIPresenter`, která poskytuje základní funkcionalitu a strukturu pro práci s komponentami, formuláři a šablonami.

4.2.4 Implementace Nette

Pro implementaci statického layoutu do Nette frameworku se obvykle postupuje takto:

- Instalace Nette framework pomocí Composeru.
- Rozdělení na statický layout na základní části, jako hlavička, tělo a patička.
- Vytvořit základní šablonu `@layout.latte` a vložit do ní části statického layoutu s použitím bloků pro dynamický obsah.

- Vytvořit presentery a šablony pro jednotlivé části aplikace. Tyto šablony budou rozšířením základní šablony a obsahovat dynamický obsah pro presentery.

4.2.5 Připojení API Coingecko

Pro napojení API Coingecko a získání aktuálních kurzů měn lze následovat tyto kroky:

Navštívit dokumentaci Coingecko API na adrese:
<https://www.coingecko.com/api/documentation>

Pro získání aktuálních kurzů měn použít koncový bod `/simple/price`. Tento endpoint umožňuje získat cenu jedné nebo více kryptoměn v jedné nebo více fiat měnách.

Připravit si HTTP požadavek v Nette pomocí knihovny, jako je Guzzle nebo `Nette\Http\Client`. Například s Guzzle:

```
composer require guzzlehttp/guzzle
```

Vytvořit metodu pro získání aktuálních kurzů měn v presenteru nebo službě:

```
use GuzzleHttp\Client;

public function getCurrentPrices($cryptocurrencies, $fiatCurrencies)
{
    $client = new Client(['base_uri' =>
'https://api.coingecko.com/api/v3/']);

    $response = $client->get('simple/price', [
        'query' => [
            'ids' => implode(',', $cryptocurrencies),
            'vs_currencies' => implode(',', $fiatCurrencies),
        ]
    ]);

    $prices = json_decode($response->getBody(), true);
    return $prices;
}
```

Použit metodu pro získání aktuálních kurzů měn v presenteru nebo službě:

```
$cryptocurrencies = ['bitcoin', 'ethereum'];  
$fiatCurrencies = ['usd', 'eur'];  
$prices = $this->getCurrentPrices($cryptocurrencies,  
$fiatCurrencies);
```

4.2.6 Implementace skriptů

V této části práce je uvedeno několik příkladů implementace PHP kódu pro framework Nette, který pracuje s MySQL databází založenou na navržené architektuře. Následující kód je zjednodušený a pro kompletní aplikaci je nutné jej dále rozšířit a upravit.

Přihlášení uživatele:

V presenteru:

```
use Nette\Security\Passwords;  
  
class SignPresenter extends Nette\Application\UI\Presenter  
{  
    private $database;  
  
    public function __construct(Nette\Database\Context $database)  
    {  
        $this->database = $database;  
    }  
  
    protected function createComponentSignInForm():  
Nette\Application\UI\Form  
    {  
        $form = new Nette\Application\UI\Form;  
        $form->addText('username', 'Uživatelské jméno:')  
            ->setRequired('Zadejte prosím uživatelské jméno.');  
        $form->addPassword('password', 'Heslo:')  
            ->setRequired('Zadejte prosím heslo.');  
        $form->addSubmit('login', 'Přihlásit se');  
        $form->onSuccess[] = [$this, 'signInFormSucceeded'];  
        return $form;  
    }  
}
```

```

        public function signInFormSucceeded(Nette\Application\UI\Form
$form, \stdClass $values): void
    {
        try {
            $userRow = $this->database->table('users')-
>where('username', $values->username)->fetch();

            if (!$userRow) {
                throw new
Nette\Security\AuthenticationException('Uživatel nebyl nalezen.');
```

V šabloně:

```

{block content}
<h2>Přihlášení</h2>
{control signInForm}
{/block}
```

Registrace uživatele

V presenteru:

```

use Nette\Security\Passwords;

class RegisterPresenter extends Nette\Application\UI\Presenter
{
    private $database;

    public function __construct(Nette\Database\Context $database)
    {
```

```

        $this->database = $database;
    }

    protected function createComponentSignUpForm():
Nette\Application\UI\Form
    {
        $form = new Nette\Application\UI\Form;
        $form->addText('username', 'Uživatelské jméno:')
            ->setRequired('Zadejte prosím uživatelské jméno.');
```

\$form->addEmail('email', 'Email:')
 ->setRequired('Zadejte prosím email.');

\$form->addPassword('password', 'Heslo:')
 ->setRequired('Zadejte prosím heslo.');

\$form->addSubmit('register', 'Registrovat se');
 \$form->onSuccess[] = [\$this, 'signUpFormSucceeded'];
 return \$form;
 }

 public function signUpFormSucceeded(Nette\Application\UI\Form
\$form, \stdClass \$values): void
 {
 try {
 \$passwordHash = Passwords::hash(\$values->password);
 \$this->database->table('users')->insert([
 'username' => \$values->username,
 'email' => \$values->email,
 'password_hash' => \$passwordHash,
 'registration_date' => new Nette\Utils\DateTime(),
]);

 \$this->flashMessage('Registrace byla úspěšná. Nyní se
můžete přihlásit.');

\$this->redirect('Sign:in');

} catch (Nette\Database\UniqueConstraintViolationException
\$e) {
 \$form->addError('Uživatelské jméno nebo email již
existuje. Zvolte prosím jiné.');

}
 }
 }
}

V šabloně:

```

{block content}
<h2>Registrace</h2>

```

```
{control signUpForm}
{/block}
```

Vkládání oblíbené kryptoměny

V presenteru:

```
class AddFavoritePresenter extends Nette\Application\UI\Presenter
{
    private $database;

    public function __construct(Nette\Database\Context $database)
    {
        $this->database = $database;
    }

    protected function createComponentAddFavoriteForm():
    Nette\Application\UI\Form
    {
        $form = new Nette\Application\UI\Form;
        $form->addText('currency_symbol', 'Symbol kryptoměny:')
            ->setRequired('Zadejte prosím symbol kryptoměny.');
```

```
        $form->addSubmit('add', 'Přidat do oblíbených');
        $form->onSuccess[] = [$this, 'addFavoriteFormSucceeded'];
        return $form;
    }

    public function
addFavoriteFormSucceeded(Nette\Application\UI\Form $form, \stdClass
$values): void
    {
        $userId = $this->getUser()->getId();

        $this->database->table('favorite_currencies')->insert([
            'user_id' => $userId,
            'currency_symbol' => $values->currency_symbol,
        ]);

        $this->flashMessage('Kryptoměna byla úspěšně přidána do
oblíbených.');
```

```
        $this->redirect('Homepage:');
    }
}
```


V šabloně:

```
{block content}
<h2>Přidání oblíbené kryptoměny</h2>
{control addFavoriteForm}
{/block}
```

Výpis oblíbených kryptoměn

V presenteru:

```
class FavoriteCurrenciesPresenter extends
Nette\Application\UI\Presenter
{
    private $database;

    public function __construct(Nette\Database\Context $database)
    {
        $this->database = $database;
    }

    public function renderDefault(): void
    {
        $userId = $this->getUser()->getId();
        $this->template->favoriteCurrencies = $this->database-
>table('favorite_currencies')
        ->where('user_id', $userId);
    }
}
```

V šabloně:

```
{block content}
<h2>Oblíbené kryptoměny</h2>
{if $favoriteCurrencies}
    <table>
        <thead>
            <tr>
                <th>Symbol kryptoměny</th>
            </tr>
        </thead>
        <tbody>
            {foreach $favoriteCurrencies as $favoriteCurrency}
                <tr>
                    <td>{$favoriteCurrency->currency_symbol}</td>
                </tr>
            </foreach>
        </tbody>
    </table>
{/if}
```

```

                </tr>
            {/foreach}
        </tbody>
    </table>
{else}
    <p>Žádné oblíbené kryptoměny nebyly nalezeny.</p>
{/if}
{/block}

```

Vkládání provedené transakce

V presenteru:

```

class AddTransactionPresenter extends Nette\Application\UI\Presenter
{
    private $database;

    public function __construct(Nette\Database\Context $database)
    {
        $this->database = $database;
    }

    protected function createComponentAddTransactionForm():
Nette\Application\UI\Form
    {
        $form = new Nette\Application\UI\Form;
        $form->addText('currency_symbol', 'Symbol kryptoměny:')
            ->setRequired('Zadejte prosím symbol kryptoměny.');
```

```

        $form->addText('amount', 'Množství:')
            ->setRequired('Zadejte prosím množství.')
            ->addRule($form::FLOAT, 'Množství musí být číslo.');
```

```

        $form->addText('price', 'Cena:')
            ->setRequired('Zadejte prosím cenu.')
            ->addRule($form::FLOAT, 'Cena musí být číslo.');
```

```

        $form->addSelect('type', 'Typ transakce:', [
            'buy' => 'Nákup',
            'sell' => 'Prodej'
        ])->setRequired('Vyberte prosím typ transakce.');
```

```

        $form->addSubmit('add', 'Přidat transakci');
        $form->onSuccess[] = [$this, 'addTransactionFormSucceeded'];
        return $form;
    }

```

```

        public function
addTransactionFormSucceeded(Nette\Application\UI\Form $form, \stdClass
$values): void
    {
        $userId = $this->getUser()->getId();

        $this->database->table('transactions')->insert([
            'user_id' => $userId,
            'currency_symbol' => $values->currency_symbol,
            'amount' => $values->amount,
            'price' => $values->price,
            'type' => $values->type,
            'transaction_date' => new Nette\Utils\DateTime(),
        ]);

        $this->flashMessage('Transakce byla úspěšně přidána.');
```

```

        $this->redirect('Homepage:');
    }
}

```

V šabloně:

```

{block content}
<h2>Přidání provedené transakce</h2>
{control addTransactionForm}
{/block}

```

Výpis provedených transakcí

V presenteru:

```

class TransactionsPresenter extends Nette\Application\UI\Presenter
{
    private $database;

    public function __construct(Nette\Database\Context $database)
    {
        $this->database = $database;
    }

    public function renderDefault(): void
    {
        $userId = $this->getUser()->getId();
        $this->template->transactions = $this->database->
table('transactions')
->where('user_id', $userId);
    }
}

```

```
}  
}
```

V šabloně:

```
{block content}  
<h2>Provedené transakce</h2>  
{if $transactions}  
  <table>  
    <thead>  
      <tr>  
        <th>Symbol kryptoměny</th>  
        <th>Množství</th>  
        <th>Cena</th>  
        <th>Typ transakce</th>  
        <th>Datum transakce</th>  
      </tr>  
    </thead>  
    <tbody>  
      {foreach $transactions as $transaction}  
        <tr>  
          <td>{$transaction->currency_symbol}</td>  
          <td>{$transaction->amount}</td>  
          <td>{$transaction->price}</td>  
          <td>{if $transaction->type ==  
'buy'}Nákup{else}Prodej{/if}</td>  
          <td>{$transaction->transaction_date|date:'j.n.Y  
H:i:s'}</td>  
        </tr>  
      {/foreach}  
    </tbody>  
  </table>  
{else}  
  <p>Žádné provedené transakce nebyly nalezeny.</p>  
{/if}  
{/block}
```

4.2.7 SEO strategie

Pro projekt "Kryptokalkulačka" je důležité mít efektivní SEO strategii, aby se webová stránka objevila ve vyhledávačích, jako je Google, a přitahovala návštěvníky. Níže je uvedeno několik doporučení pro vhodnou SEO strategii:

- Klíčová slova a fráze: Výběr relevantních klíčových slov a frází, která jsou spojena s kryptoměnami, konverzí měn a souvisejícími tématy. Použití nástrojů, jako je Google Keyword Planner, pro nalezení klíčových slov s vysokým počtem vyhledávání a nízkou konkurencí.
- Optimalizace obsahu: Optimalizovat obsah na webové stránce pro vybraná klíčová slova a fráze. Zahrnout klíčová slova do nadpisů, podnadpisů, textu a meta tagů, jako jsou title tag a meta description. Ujistit se, že obsah je jedinečný, relevantní a užitečný pro uživatele.
- Responzivní design: Ujistit se, že web je responzivní a optimalizovaný pro různá zařízení a prohlížeče, aby poskytoval co nejlepší uživatelský zážitek. Google dává přednost webovým stránkám s responzivním designem, což může pozitivně ovlivnit SEO hodnocení.
- Rychlost načítání stránky: Optimalizovat rychlost načítání webové stránky, protože to je důležitý faktor pro SEO hodnocení. Zkomprimovat obrázky, minimalizovat počet externích skriptů a použít techniky cachování pro zvýšení rychlosti načítání stránky.
- Interní a externí odkazy: Vytvářet interní odkazy mezi stránkami na webu pro zlepšení navigace a externí odkazy na důvěryhodné zdroje, které poskytují užitečné informace. Tímto způsobem lze zlepšit autoritu webové stránky a zvýšit důvěru uživatelů i vyhledávačů.
- Sociální média a sdílení obsahu: Podporovat sdílení obsahu na sociálních médiích a získávat zpětné odkazy na webovou stránku. Vytvořit profily na populárních sociálních sítích, jako jsou Facebook, Twitter, LinkedIn a Instagram, a pravidelně zveřejňovat obsah, který se týká tohoto projektu. Sledovat aktivity na sociálních médiích, komunikovat s uživateli a přizpůsobit obsah jejich potřebám a zájmům.

- **Technické SEO:** Zajistit, zdali web splňuje technické požadavky vyhledávačů, jako je správné nastavení robots.txt, sitemap.xml a strukturovaná data (např. schema.org). Použít HTTPS pro zabezpečené připojení a ujistit se, že všechny URL adresy jsou dobře strukturované a jednoznačné.
- **Měření a sledování:** Sledovat a analyzovat výkon webu pomocí nástrojů jako Google Analytics, Google Search Console a dalších SEO nástrojů. Pravidelně zjišťovat, jak se web umísťuje ve výsledcích vyhledávání pro vybraná klíčová slova, a na základě těchto informací optimalizovat SEO strategii.
- **Lokalizace a jazyková podpora:** Pokud se projekt zaměřuje na uživatele z různých zemí a jazykových skupin, zvážit lokalizaci obsahu a překlady do různých jazyků. To může zlepšit viditelnost webu ve vyhledávačích pro uživatele z různých zemí a zvýšit důvěru uživatelů.
- **Kvalitní obsah:** Pravidelně publikovat články, návody, novinky a další obsah související s kryptoměny, který bude zajímavý a užitečný pro návštěvníky. Kvalitní obsah přitahuje více uživatelů, což vede ke zvýšení důvěryhodnosti a autority webu.

4.3 Řízení projektu

V rámci tvorby webových stránek je řízení projektu důležitým, avšak okrajovým prvkem této diplomové práce, kterému bude věnována pouze krátká pozornost. Práce se primárně zaměřuje na jiné aspekty tvorby webových stránek, nicméně je důležité zmínit základy řízení projektu pro úplnost a pochopení možné výdělečnosti.

Řízení projektu se týká koordinace různých fází webového projektu, jako je plánování, návrh, vývoj, testování a nasazení. Klíčovými složkami úspěšného řízení projektu jsou komunikace, organizace, časový management a řízení rizik.

V této stručné části práce bude představen základní přehled řízení projektu v kontextu tvorby webových stránek, aniž by se zabýval konkrétním projektem. Důraz bude kladen na možnosti výdělečnosti, které mohou být spojeny s efektivním řízením projektu.

Je důležité si uvědomit, že úspěšné řízení projektu může přispět k optimalizaci zdrojů, zvýšení efektivity a snížení nákladů, což může vést k lepšímu výsledku a potenciálnímu zisku. Zároveň může správné řízení projektu zvýšit spokojenost zákazníků a pověst firmy, což může mít pozitivní dopad na budoucí obchodní příležitosti.

4.3.1 Projektový plán

Název projektu	Kryptokalkulačka		
Identifikace projektu	KK	Vedoucí projektu	
Pracovník pověřený vypracováním		Termín zahájení stadia „Provedení projektu“	3. 1. 2022
Verze	1.0	Termín ukončení prací na projektu ve stadiu „Provedení projektu“	21. 2. 2022
Datum vypracování	7. 1. 2022	Termín ukončení stadia „Ukončení projektu“	21. 2. 2022

Tabulka 6: Projektový plán

4.3.1.1 Důvody projektu

Projekt vznikl na základě průzkumu trhu a často vyhledávaných služeb v oblasti kryptoměn. Vzhledem ke složitosti nákupu a prodeje kryptoměn a neexistence žádné žádného druhu této služby, který by tuto problematiku uživatelům dostatečně usnadnil.

4.3.1.2 Cíle projektu

ID	Cíl projektu	Metrika	Indikátor dosažení cíle (cílová hodnota metriky)
1	Webová kalkulačka kryptoměn	Funkční systém	Úspěšný beta test
2	Administrace obsahu	Redakční systém	Ověření funkčnosti obsluhou
3	Porovnávač kryptoburz	Počet burz	Tři největší
4	Internetová propagace aplikace	Hit per \$	Unique 1000 hits per month

Tabulka 7: Cíle projektu

4.3.1.3 Předmětem projektu není

Součástí projektu není převod národních měn např. usd, czk, eur a jiné... Dále pak není součástí projektu zprostředkování nákupů a prodeje kryptoměn, ale pouze poskytnutí k tomu vhodné burzy.

4.3.1.4 Očekávané přínosy projektu

Hlavními přínosy projektu jsou očekávány dodávky plně funkčních částí webové aplikace v rámci každé naplánované etapy projektu. Cílem je tak zajistit nepřetržitou dodávku funkčních částí webové aplikace zákazníkům. Projekt umožní uživatelům převádět a porovnávat jednotlivé kryptoměny, i s ohledem na historický vývoj jednotlivých kryptoměn, a to i včetně převodu v časovém období zpětně. V rámci předplatného bude umožněno uživatelům si často používané a převáděné měny uložit do oblíbených. V dnešním dynamickém trhu kryptoměn bude administrátorům umožněno libovolné přidávání nových, a i již zaniklých kryptoměn pro rozšíření dosavadního portfolia. Plná automatizace plnění obsahu webové aplikace při napojení na webové služby kryptoměnových burz.

4.3.1.5 Předpoklady realizace projektu

Pro zahájení realizace projektu musí být provedena analýza trhu a existence zadání definující rozsah, požadavky a očekávání projektu.

4.3.1.6 Průzkum

Podle průzkumu Eurobarometru z roku 2021 používalo v Evropské unii kryptoměny 30 % respondentů. Míra penetrace kryptoměn se však liší v různých zemích, s nejvyššími hodnotami v zemích jako Malta, Řecko nebo Rumunsko a nejnižšími v Německu, Dánsku nebo Belgii. Zdrojem informace o penetraci kryptoměn v EU je průzkum Eurobarometru z května 2021 [48].

Podle studie Cambridgeského centra pro alternativní finance z roku 2020 měla největší skupina uživatelů kryptoměn v Evropě příjem mezi €35,000 a €70,000 ročně, následovaná těmi s příjmem mezi €70,000 a €140,000 ročně. Podobná data o příjmových skupinách kryptoměnových uživatelů jsou k dispozici v průzkumech Cambridge Center for Alternative Finance [49].

Podle dat Eurostatu z roku 2021 byl průměrný hrubý roční příjem v Evropské unii 27 zemích 29 300 eur. S ohledem na tuto hodnotu lze uvedené příjmové skupiny pozicovat jako střední a vyšší střední třídu v rámci Evropské unie.

Podle dat z Českého statistického úřadu (ČSÚ) z roku 2020 mělo průměrné hrubé měsíční příjmy v ČR hodnotu 35 260 Kč. Předpokládáme-li, že rozdělení příjmů v ČR odpovídá Gaussovské křivce, horních 25 % příjmů by odpovídalo příjmu nad 49 230 Kč měsíčně, což by odpovídalo přibližně 2,4 milionům lidí v ČR. Zdrojem dat o průměrných a mediálních hrubých měsíčních příjmech podle vybraných skupin obyvatel v ČR za rok 2020 jsou údaje z Českého statistického úřadu, které jsou zveřejněny na webových stránkách ČSÚ [50].

4.3.2 Bussines Model Canvas

Business Model Canvas je vizuální nástroj pro plánování a analýzu podnikatelského modelu. Skládá se z devíti stavebních bloků, které reprezentují klíčové prvky podnikání:

- Zákaznické segmenty: Cílové skupiny, kterým firma poskytuje hodnotu.
- Hodnotová nabídka: Jedinečné výhody a hodnoty nabízené zákazníkům.

- Kanály: Způsoby doručení hodnoty zákazníkům.
- Vztahy se zákazníky: Typy interakcí mezi firmou a zákazníky.
- Příjmy: Zdroje příjmů generovaných firmou.
- Klíčové zdroje: Nezbytné zdroje pro fungování firmy.
- Klíčové činnosti: Hlavní činnosti prováděné firmou pro dosažení úspěchu.
- Klíčoví partneři: Důležití partneři a dodavatelé pro efektivní fungování firmy.
- Nákladová struktura: Hlavní náklady spojené s provozem podnikatelského modelu.

Business Model Canvas pomáhá rychle analyzovat silné a slabé stránky podnikání a identifikovat oblasti pro zlepšení.

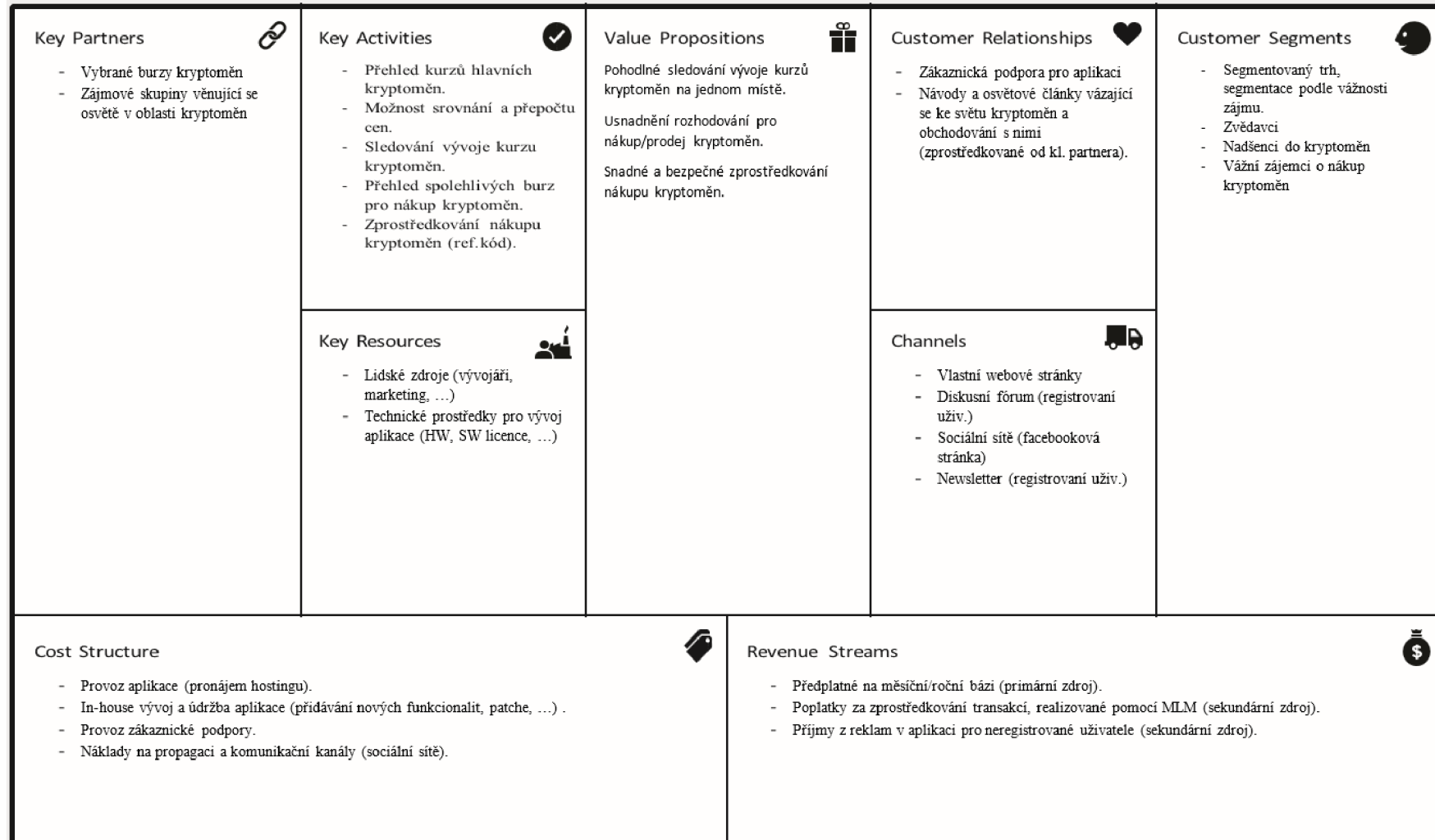
The Business Model Canvas

Designed for: Projekt Kryptokalkulátor

Designed by: Jan Kalina

Date: 10.1.2023

Version: 0.1



Obrřzek 23: The Business Model Canvas

4.3.3 Persony

Persony v oblasti vývoje webových stránek jsou fiktivní, ale detailní reprezentace skupin uživatelů, které mají společné potřeby, cíle nebo chování. Persony pomáhají návrhářům a vývojářům lépe pochopit potřeby a očekávání skutečných uživatelů, což umožňuje vytvářet uživatelsky přívětivější a efektivnější webové stránky.

Uplatnění person ve vývoji webových stránek zahrnuje:

- **Zaměření na skutečné potřeby uživatelů:** Persony umožňují týmu soustředit se na konkrétní potřeby a očekávání skupin uživatelů, což vede k lepšímu návrhu a funkcionalitě webových stránek.
- **Zlepšení komunikace v týmu:** Persony poskytují společný jazyk a referenční bod pro tým, což zlepšuje komunikaci a koordinaci během vývoje.
- **Rozhodování o prioritách:** Persony pomáhají určit, které funkce a návrhy jsou pro uživatele nejdůležitější, což usnadňuje rozhodování o prioritách.
- **Testování a validace:** Persony mohou být použity pro vytváření scénářů a testování uživatelského rozhraní, což pomáhá zlepšit kvalitu a uživatelskou zkušenost webových stránek.

4.3.3.1 Popis cílové skupiny

V posledních pěti letech zaznamenaly kryptoměny značný nárůst popularity. Přestože mnoho lidí dnes alespoň tuší, co je Bitcoin, kryptoměny stále nejsou považovány za mainstreamový fenomén. Pro velkou část populace zůstává tato oblast záhadná a s občasnými extrémními výkyvy kurzů některých kryptoměn i zdánlivě nebezpečná.

Projekt Kryptokalkulačka se zaměřuje na cílovou skupinu s vážnějším zájmem o kryptoměny. Členové této skupiny mají minimálně základní povědomí o kryptoměnách a hledají nástroj, který by usnadnil rozhodování o případném nákupu měn.

Cílovou skupinu lze dále rozdělit podle znalosti problematiky kryptoměn a ochoty investovat do krypta. Byly identifikovány tři částečně se překrývající segmenty: zvědavé lamy, kryptonerdi a investiční dobrodruzi.

Zvědavé lamy mají rády technologie a o kryptoměnách se zajímají hlavně kvůli hype okolo Bitcoinu. Není pro ně důležité znát podstatu kryptoměn do detailu, ale vnímají je jako způsob zhodnocení nebo zajištění části svého kapitálu. Jsou opatrnější a nechtějí investovat příliš mnoho, spíše se chtějí ujistit, že investování do krypta má budoucnost i pro menší investory.

Kryptonerdi jsou většinou mladí lidé, často ze sféry IT, studenti nebo čerstvě zaměstnaní absolventi. Rozumí kryptoměnám a považují je za pevnou součást budoucnosti. Zatím však nemají dostatek finančních prostředků pro vážné investování. Rádi sdílejí své znalosti na sociálních sítích nebo diskusních fórech a Kryptokalkulačku mohou vnímat jako nástroj pro simulaci investic a následnou prezentaci svých správných předpovědí.

Investiční dobrodruzi jsou majetnější jedinci, kteří hledají investice s potenciálem rychlého zhodnocení. Jsou přitahováni příběhy úspěchu jako u Bitcoinu nebo Ethereum, disponují finančními prostředky a nebojí se je investovat. Tato skupina představuje směsici zvědavých lam a zkušenějších nerdů, kteří si mohou dovolit investovat do krypta bez ohrožení své finanční stability. Zároveň se nejedná o těžké investory, kteří by si pro správu svého portfolia najali osobního makléře nebo investiční firmu.

V rámci diplomové práce je důležité zaměřit se na potřeby a očekávání těchto cílových skupin, aby bylo možné efektivně reagovat na jejich požadavky a přizpůsobit projekt Kryptokalkulačka tak, aby co nejlépe vyhovoval jejich potřebám.

4.3.3.2 Ukázky person

V této části jsou uvedeny osoby, které reprezentují jednotlivé zákaznické segmenty, popsané v předchozí části.

Persona 1 (zvědavé lamy)

Osobní údaje:

- Jméno: Jana Skoupá
- Věk: 35 let
- Pohlaví: žena
- Bydliště: Ústí nad Labem
- Rodinný stav: rozvedená, dcera Viktorie (5 let)
- Dosažené vzdělání: vysokoškolské bakalářské (VŠE v Praze)
- Pracovní pozice: ekonomka v malé soukromé společnosti
- Měsíční příjem: 35 500 Kč
- Volnočasové aktivity: péče o dceru, četba, sociální sítě, Netflix

Získávání a sdílení informací: Paní Jana informace získává hlavně z internetových magazínů a sociálních sítí. Tento mix doplňuje sledováním večerního zpravodajství ČT. Informace sdílí hlavně na Facebooku a diskusních fórech zpravodajských webů IDnes a Novinek.

Znalost a používání technologií: Paní Jana používá internet a sociální sítě na notebooku a mobilním telefonu.

Cíle, motivace: Paní Jana by ráda zhodnotila část svých peněz. Nové technologie ji přitahují a kryptoměny v jejích očích stále zahaluje opar magičnosti, prudký růst Bitcoinu v posledních letech ji ohromil.

Potřeby, očekávání, preference, faktory pro rozhodnutí: Paní Jana potřebuje aplikaci, která by ji bezpečně navigovala světem kryptoměn. Chtěla by mít možnost přepočítávat kurzy nebo sledovat jejich vývoj v čase a tak si zkusit nákupy nanečisto, bez obavy ze

ztráty vlastních peněz. To vše na základě důvěryhodných dat a bez agresivního marketingu a paywallů.

Bariéry a osobní averze: Paní Jana ví, že neexistuje nic jako oběd zdarma, a proto má odpor k agresivnímu marketingu jedinečných příležitostí a zaručeného pohádkového zisku. Na kryptoměnách jí nejvíce vadí volatilita jejich kurzů.

Persona 2 (kryptonerdi)

Osobní údaje:

- Jméno: Jáchym Kňourek
- Věk: 22 let
- Pohlaví: muž
- Bydliště: Kralupy nad Vltavou
- Rodinný stav: svobodný, bezdětný
- Dosažené vzdělání: vysokoškolské bakalářské (FIT ČVUT v Praze)
- Pracovní pozice: vývojář (částečný úvazek)
- Měsíční příjem: 59 500 Kč
- Volnočasové aktivity: četba fantasy a naučné literatury, larp, turistika a geocaching, aktivní přispívatel do diskusí o kryptu na sociálních sítích, filmy a seriály (Netflix, Hulu, HBO GO), počítačové hry (The Division, CoD: Warzone, WoW)

Získávání a sdílení informací: Pan Jáchym získává informace hlavně z internetu a zde je také i sdílí. Je členem různých skupin na sociálních sítích včetně těch věnovaných kryptoměnám (Facebook), má účet na Twitteru. Dále se účastní diskusí na webech věnovaných tématice kryptoměn, případně na diskusích na zpravodajských webech (IDnes).

Znalost a používání technologií: Pan Jáchym jako vývojář IT systémů používá technologie běžné pro výkon své profese. Má nadprůměrné znalosti v používání osobních počítačů, vývojářských nástrojů a programování. Rovněž rozumí principům, na kterých stojí kryptoměny (distribuovaný systém, blockchain, ...). Lze ho charakterizovat jako

pokročilého uživatele webu. Pro volný čas i práci používá sociální sítě a specializované weby. Pro práci a zábavu používá stolní PC, notebook, tablet a mobilní telefon.

Cíle, motivace: Pan Jáchym sleduje svět kryptoměn, rád o nich čte, a ještě raději o nich poučuje ostatní. Bere to jako měření sil, virtuální znalostní turnaj, na kterém buduje část svého sociálního statusu. Ve slabších chvílích se snaží dobře poradit ostatním. Jeho cílem je stát se zkušeným a uznávaným odborníkem na kryptoměny ve svém okruhu a rozšířit svoji znalostní základnu v této oblasti.

Potřeby, očekávání, preference, faktory pro rozhodnutí: Pan Jáchym potřebuje spolehlivý nástroj pro sledování kurzů a převodů vybraných kryptoměn. Chce mít možnost tyto informace sledovat i zpětně, včetně historických dat. Preferuje možnost provádět tyto akce na jednom místě. Jako student by ocenil, kdyby nástroj poskytoval alespoň část užitečných funkcí zdarma. Dále by rád měl možnost sledovat vývoj kryptoměnového trhu a diskutovat s ostatními uživateli, aby mohl rozvíjet své investiční teorie a zlepšovat své investiční dovednosti.

Bariéry a osobní averze: Při používání aplikací mu vadí částečně skryté paywally, nespolehlivá funkcionalita a data. Nerad používá aplikace, které plýtvají jeho časem, protože jsou z hlediska User eXperience špatně navrženy. Pan Jáchym očekává jasné a přehledné rozhraní, které usnadní jeho interakci s aplikací a ušetří jeho čas. Dále nesnáší, když je nucen se přihlašovat do aplikace přes sociální sítě nebo sdílet své osobní údaje, pokud to není nezbytně nutné.

Persona 3 (investiční dobrodruzi)

Osobní údaje:

- Jméno: Evžen Plachý
- Věk: 45 let
- Pohlaví: muž
- Bydliště: Praha 3
- Rodinný stav: ženatý, manželka Adéla (42), synové Petr (12) a Pavel (9)
- Dosažené vzdělání: vysokoškolské magisterské (ČZU v Praze)

- Pracovní pozice: ekonomický ředitel ve střední soukromé společnosti
- Měsíční příjem: 71 000 Kč
- Volnočasové aktivity: turistika, cyklistika, četba soudobé beletrie, sociální sítě, počítačové hry

Získávání a sdílení informací: Pan Evžen získává informace sledováním televize, sociálních sítí a z komunity svých známých a přátel, zejména v zaměstnání. Z televizi sleduje hlavně ČT24 a BBC, zejména pořady z oblasti businessu a technologií. Aktivně přispívá na Facebook, je členem skupin věnujících se technologiím, investování a kryptoměnám. Sleduje Twitter, i když na něj moc nepřispívá.

Znalost a používání technologií: Pan Evžen je z hlediska znalostí technologií průměrným počítačovým uživatelem. K práci i zábavě používá notebook a mobilní telefon. O technologiích týkajících se kryptoměn má povědomí, není však expertem.

Cíle, motivace: Pan Evžen má peníze a nebojí se je použít. Před svým okolím a zejména kamarády se rád prezentuje jako drobný investor, který rozumí a umí vydělat na nových technologiích. Kryptoměny pro to z jeho pohledu představují ideální nástroj pro investování. Hledá tedy aplikaci, která by mu poskytla spolehlivou podporu v tomto investičním rozhodování.

Potřeby, očekávání, preference, faktory pro rozhodnutí: Aplikace by měla umožnit převádět kryptoměny, sledovat jejich kurzy za časový úsek, pro snadnější rozlišení umožnit vytvořit sledování kurzu oblíbených kryptoměn s upozorněním na možné výkyvy v kurzu. Dále by pan Evžen uvítal, kdyby si mohl rovnou vybrat na tom samém místě konkrétní burzu, na které provede transakci, případně mít možnost sledovat kurz stejné kryptoměny na různých burzách a vybrat si tak tu výhodnější.

Bariéry a osobní averse: Pan Evžen ví, že investování je zatíženo již tak dostatečnou nejistotou, a proto nemá rád, když se jeho rozhodování má dít na základě nespolehlivých, nejednoznačných nebo zastaralých informací. Rovněž pevně věří, že čas jsou peníze, a proto ho jakékoliv nepotřebné zdržení při používání aplikací rozčiluje.

Persona 4 (investiční dobrodruzi)

Osobní údaje:

- Jméno: Igor Všehrab
- Věk: 52 let
- Pohlaví: muž
- Bydliště: Praha 1
- Rodinný stav: ženatý, bezdětný
- Dosažené vzdělání: středoškolské s maturitou
- Pracovní pozice: podnikatel v oblasti finančního poradenství a přepravy nenadměrných nákladů
- Měsíční příjem: neodvázili jsme se zeptat
- Volnočasové aktivity: střelba, turistika, sběratelství (drahé kovy, platné mince a bankovky s co nejvyšší nominální hodnotou).

Získávání a sdílení informací: Pan Igor je uživatelem Facebooku. Zprávy získává z této platformy. Dále sleduje televizní vysílání České televize a odebírá noviny MF Dnes.

Informace sdílí ústně v okruhu přátel a spolupracovníků.

Znalost a používání technologií: Pan Igor jde s dobou, internet využívá na stolním počítači, notebooku a telefonu. O technologiích spojených s kryptoměnami neví mnoho, tuší ale, že komplikují práci finančnímu úřadu.

Cíle, motivace a proces nákupu: Pan Igor je ze staré školy a ví, že každá párty jednou skončí. Podnikatelem v oblasti finančního poradenství a přepravy nenadměrných nákladů je od devadesátých let, takže také ví, že o tom, kdy párty končí, nerozhoduje tak úplně sám. Z toho důvodu chce být připraven, přeci jen mu z mládí v pionýru utkvělo v hlavě to podstatné. Kryptoměny se mu zdají jako skvělá příležitost, jak ochránit část svých peněz před slídily. Pokud by mu některý nástroj pomohl s tímto problémem, neváhal by projevit svou štedrost.

Potřeby, očekávání, preference, faktory pro rozhodnutí: Pan Igor si je vědom svého postavení ve své komunitě. Nikdy se nesnažil zapadnout, protože je to přesně opačný směr pohybu než ten, o který se snažil celý život, totiž cestu vzhůru. A chce, aby to tak zůstalo i v případě, že ho osud na nějaký čas posadí na střídačku. Proto potřebuje spolehlivý jednoduchý nástroj, který by mu pomohl zprostředkovat nákup kryptoměny na důvěryhodné burze. Ví, že takové nástroje již existují, ale žádný z nich není provozován místní společností. To zraňuje jeho vlastenecké cítění a také mu to neumožňuje zdůraznit naléhavost případné nespokojenosti se službami osobní návštěvou.

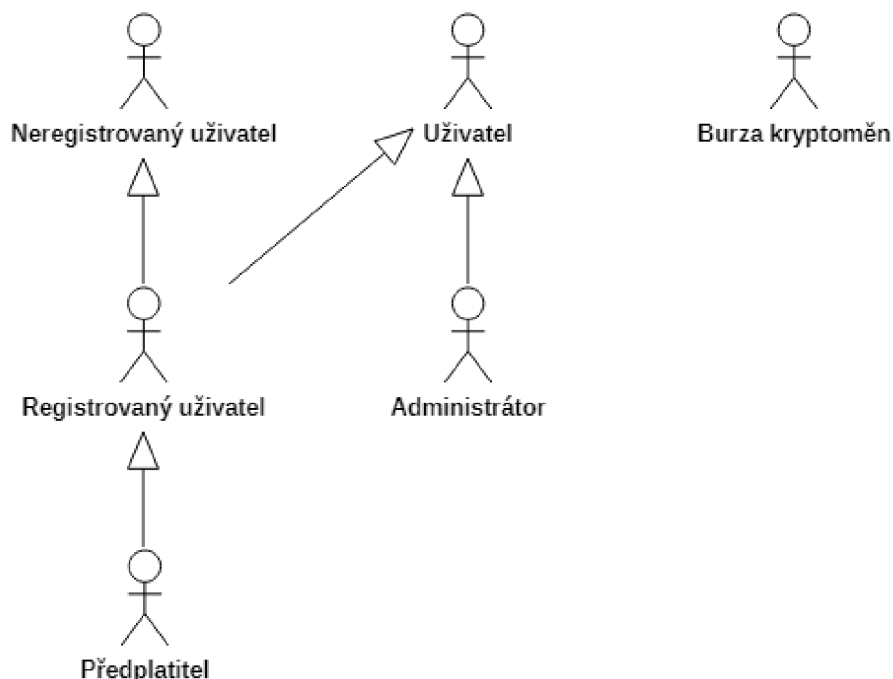
Bariéry a osobní averze: Pan Igor nemá rád vlezlost. Přímo nesnáší, pokud si někdo dovoluje uhadnout, na co právě myslí. Nemá rád průvodce a nerad čte v mapách. Ještě nepotkal člověka, ze kterého by správnou motivací nedostal potřebné informace. Co se týče aplikací, očekává jednoduchost, intuitivní ovládání a žádné zbytečné rušivé prvky. Pan Igor také upřednostňuje osobní zodpovědnost a očekává, že podpora bude reagovat rychle a efektivně na případné problémy nebo dotazy.

Tyto osoby by měly pokrýt širokou škálu potenciálních uživatelů webu. Při návrhu webu je na snaze brát v úvahu jejich potřeby, očekávání a zájmy, aby byl vytvořena uživatelsky přívětivá platforma, která je přizpůsobena různým úrovním zkušeností a znalostí v oblasti kryptoměn. Dobré je zaměřit se na snadnou orientaci, přístup k informacím a možnost komunikace s komunitou okolo webu případně s jeho podporou.

4.3.4 UC scénáře

4.3.4.1 Aktéři

Aktéry a jejich vzájemné vztahy znázorňuje diagram na Obrázek 24. Jejich podrobnější popis je v následujících odstavcích.



Obrázek 24: Diagram aktéři případů užití.

Neregistrovaný uživatel nemá v systému uloženy žádné informace. Systém používá pouze anonymně s omezenými možnostmi užití, zpravidla jen porovnání hodnoty dvou kryptoměn pomocí aktuálního kurzu.

Uživatel je abstraktním předkem, který sdružuje případy užití společné pro registrovaného uživatele a administrátora. Systém uchovává informace o jeho stavu přihlášení.

Administrátor je správcem systému a vykonává správu účtů ostatních uživatelů a seznamu nabízených burz kryptoměn.

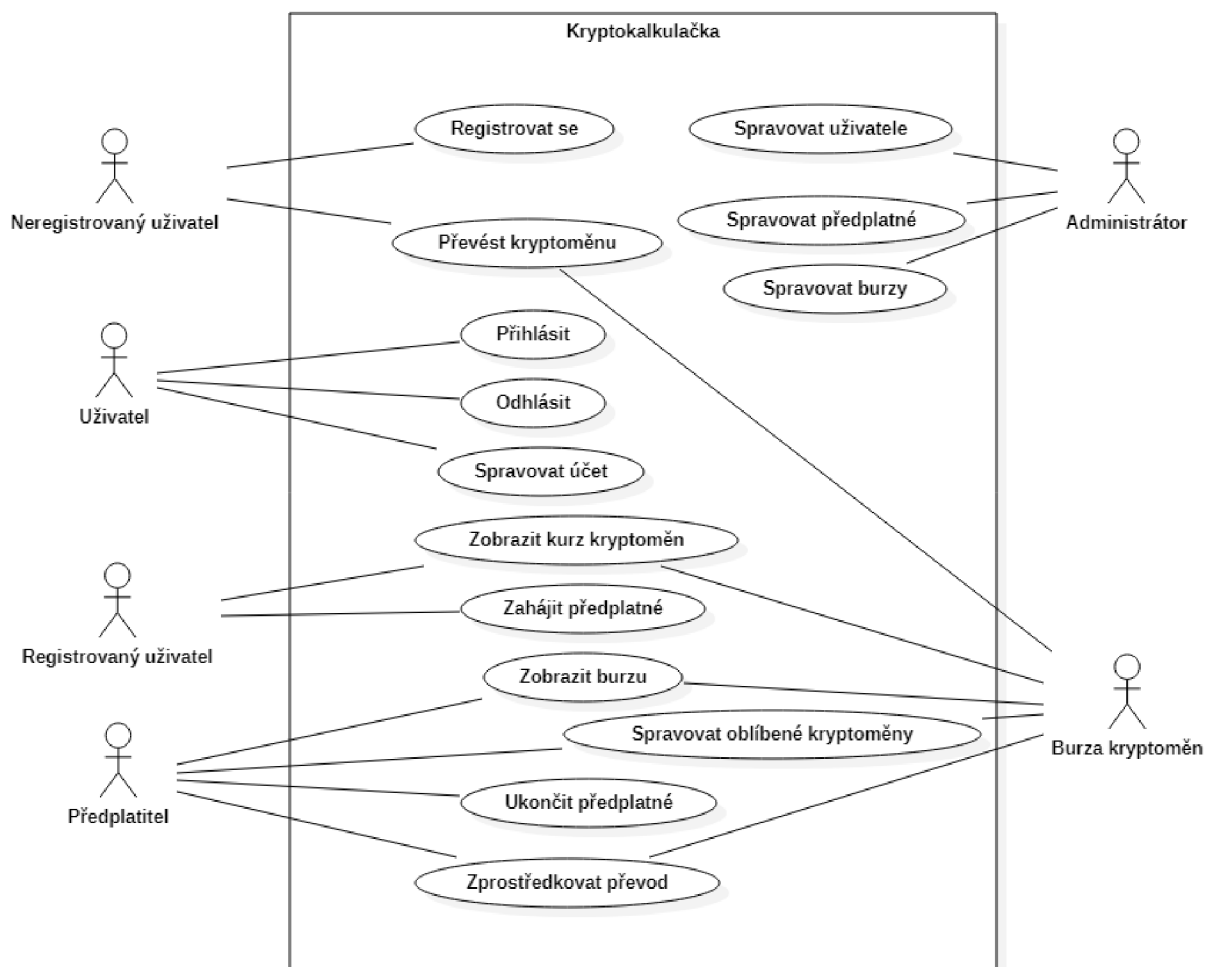
Registrovaný uživatel může v systému sledovat kurzy kryptoměn, včetně historických kurzů. Systém uchovává informace relevantní pro registraci a umožňuje některé změnit. Registrovaný uživatel se může stát předplatitelem.

Předplatitel využívá veškerou funkcionalitu systému spojenou s kryptoměnami, tedy převody, sledování kurzů a burz. Systém uchovává informace o jeho oblíbených měnách, převezech a naposledy prohlížených burzách.

Burza kryptoměn je sekundárním aktérem, se kterým systém komunikuje za účelem získání kurzů kryptoměn a dalších podstatných informací.

4.3.4.2 Případy užití

Případy užití jsou zachyceny na diagramu na Obrázek 25. Pro zachování přehlednosti byly v diagramu zakryty vztahy generalizace/specializace mezi aktéry. Tyto vazby je možné vidět v diagramu na předchozím Obrázek 24.



Obrázek 25: Diagram případů užití - kryptokalkulačka.

4.3.4.3 Scénáře

V této části jsou uvedeny dva scénáře, které se vážou k případům užití z předchozího diagramu na Obrázek 25. Hlavní a alternativní scénáře jsou uvedeny v samostatných tabulkách. Sloupec nazvaný alternativní scénář je přítomen pouze pokud v daném scénáři dochází k větvení na základě podmínky. V tom případě je v tomto sloupci uveden odkaz na alternativní scénář.

Scénář registrace nového uživatele

- Název UC: registrovat se
- Cíl: registrovat nového uživatele.
- Počáteční stav: neregistrovaný uživatel, nachází se na hlavní stránce aplikace.
- Konečný úspěšný stav: registrovaný uživatel.
- Konečný neúspěšný stav: nedokončená registrace.
- Aktéři: neregistrovaný uživatel.

Hlavní úspěšný scénář

Číslo kroku	Popis	Alternativní scénář
1.	Uživatel vybere volbu „registrovat“.	
2.	Systém zobrazí registrační formulář.	
3a.	Uživatel vyplní správné údaje do formuláře.	A1
4a.	Systém provede validaci údajů.	
5	Uživatel odešle vložené údaje.	
6.	Systém odešle email pro potvrzení registrace.	
7.	Systém zobrazí uživateli zprávu o proběhlé registraci a nutnosti jejího potvrzení.	
8a.	Uživatel potvrdí registraci odkazem z emailu z kroku 6.	A2
9.	Systém označí registrovaný účet za aktivní.	
10.	Systém odešle uživateli zprávu o úspěšném dokončení registrace.	

Tabulka 8: Registrace – hlavní scénář.

Alternativní scénář 1 - zadání chybných registračních údajů.

Číslo kroku	Popis
3b.	Uživatel vyplní některý údaj do formuláře chybně.
4b.	Systém provede validaci údajů.
5b.	Systém zvýrazní uživateli špatně zadané údaje ve formuláři a vyzve ho k jejich opravě.
6b.	Uživatel opraví všechny označené údaje.
7b.	Systém provede validaci údajů.
8b.	Uživatel pokračuje od bodu č.5 hlavního scénáře.

Tabulka 9: Registrace – alternativní scénář 1.

Alternativní scénář 2 – nepotvrzení registrace.

Číslo kroku	Popis
8c.	Uživatel nepotvrdí registraci odkazem z emailu z kroku č.6 hlavního scénáře.
9c.	Systém označí registrovaný účet za neaktivní.

10c.	Systém odešle před vypršením doby platnosti registrace poslední výzvu uživateli k jejímu dokončení.
11c.	Uživatel pokračuje krokem č.8a hlavního scénáře.

Tabulka 10: Registrace – alternativní scénář 2.

Scénář zobrazení kurzu kryptoměny

- Název UC: zobrazit kurz kryptoměn.
- Cíl: zobrazit uživateli kurz vybrané kryptoměny za vybraný časový úsek.
- Počáteční stav: přihlášený registrovaný uživatel, nachází se na hlavní stránce aplikace.
- Konečný úspěšný stav: zobrazený kurz vybrané kryptoměny za vybrané období v podobě grafu a doprovodných informací.
- Konečný neúspěšný stav: nezobrazený kurz, návrat na hlavní okno aplikace.
- Aktéři: registrovaný uživatel (primární), burza kryptoměn (sekundární).

Hlavní úspěšný scénář

Číslo kroku	Popis	Alternativní scénář
1.	Uživatel vybere volbu zobrazit kurz kryptoměny.	
2.	Systém zobrazí formulář pro výběr burzy, měny a sledovaného období.	
3a.	Uživatel vyplní správné údaje do formuláře.	A1
4a.	Systém provede validaci údajů.	
5	Uživatel odešle vložené údaje.	
6.	Systém najde uživatelem zvolenou burzu v seznamu burz.	
7.	Systém odešle požadavek na údaje o kurzu za určené období zvolené burze.	
8.	Burza odpoví na požadavek systému údaji o kurzu za období a tím zároveň potvrdí svou dostupnost.	A2
9.	Systém přijme údaje od burzy a transformuje je do podoby grafů a doprovodných informací, které zobrazí uživateli.	

Tabulka 11: Zobrazení kurzu kryptoměn – hlavní scénář.

Alternativní scénář 1 – zadání chybných dat

Číslo kroku	Popis
3b.	Uživatel vyplní některý údaj chybně.
4b.	Systém provede validaci údajů.
5b.	Systém zvýrazní chybně zadané údaje a vyzve uživatele k jejich opravě.
6b.	Uživatel pokračuje krokem č. 3a. hlavního scénáře.

Tabulka 12: Zobrazení kurzu kryptoměn – alternativní scénář 1.

Alternativní scénář 2 – nedostupná vybraná burza kryptoměn

Číslo kroku	Popis	Alternativní scénář
8c.	Burza neodpoví na požadavek v určeném časovém intervalu.	
9c.	Systém zobrazí uživateli informaci o nedostupnosti vybrané burzy.	
10c.	Systém vytvoří podseznam burz, které mají v nabídce kurz požadované kryptoměny.	A3
11c.	Systém zobrazí podseznam burz uživateli a vyzve ho k výběru náhradní burzy.	
12c.	Uživatel vybere burzu z podseznamu.	
13c.	Systém pokračuje krokem č. 6 z hlavního scénáře.	

Tabulka 13: Zobrazení kurzu kryptoměn – alternativní scénář 2.

Alternativní scénář 3 – žádná dostupná burza pro vybranou kryptoměnu

Číslo kroku	Popis
10d.	Systém nedokáže vytvořit podseznam dostupných burz pro zvolenou kryptoměnu.
11d.	Systém zobrazí uživateli informaci o nedostupnosti burz.
12d.	Systém nabídne uživateli návrat do hlavního okna aplikace.
13d.	Uživatel potvrdí návrat do hlavního okna systému.
14d.	Systém zobrazí hlavní okno aplikace.

Tabulka 14: Zobrazení kurzu kryptoměn – alternativní scénář 3.

4.3.5 Harmonogram

Ganttův diagram je vizuální nástroj používaný pro plánování a sledování průběhu projektů. Byl poprvé představen Henrym L. Ganttem na začátku minulého století a od té doby se stal oblíbenou metodou řízení projektů v mnoha oborech. Diagram zobrazuje jednotlivé úkoly projektu na časové ose, což umožňuje snadno zobrazit a porozumět plánovanému průběhu a aktuálnímu stavu projektu. V rámci projektu plní Ganttův diagram následující úlohy:

- **Plánování:** Ganttův diagram pomáhá týmu při plánování jednotlivých úkolů projektu, stanovení jejich trvání, závislostí a pořadí.
- **Koordinace:** Umožňuje lepší koordinaci mezi členy týmu a zajišťuje, že každý má jasný přehled o tom, co má dělat a kdy.

- Sledování pokroku: Ganttův diagram poskytuje vizuální přehled o aktuálním stavu projektu, což umožňuje týmu rychle identifikovat, zda jsou úkoly dokončeny včas nebo zda jsou zpožděny.
- Řízení zdrojů: Pomáhá při přidělování zdrojů, jako je čas, personál a finance, a při řízení jejich efektivního využití.
- Komunikace: Zjednodušuje komunikaci mezi členy týmu a s dalšími zainteresovanými stranami tím, že poskytuje jasný a srozumitelný přehled o plánu a průběhu projektu.

Ganttův diagram popisovaného projektu webových stránek zahrnuje následující fáze a klíčové činnosti:

- Zahájení
 - Definice cílů projektu
 - Identifikace stakeholderů
 - Finalizace cílů
 - Analýza a specifikace požadavků
 - Vytvoření plánu
 - Zahajovací schůzka (kickoff)
- Web design analýza
 - Výzkum klíčových slov a SEO
 - Vytvoření person a struktury webu
 - Navrhování wireframů
 - Design stránek
 - Práce na uživatelském zážitku (UX)
 - Vývoj prototypu
- Vlastní web

- Kontrola kvality (QA) prototypu
 - Oprava chyb
 - Kontrola kvality (QA) Build 1
 - Oprava chyb
 - Kontrola kvality (QA) finální sestavení
- Staging a akceptace
 - Přesun na hosting
 - Testování funkčnosti
 - Akceptace a předání pro provoz

Tento slovní popis Ganttova diagramu představuje postupný průběh projektu webových stránek, od zahájení až po akceptaci a předání hotového projektu. Každá fáze zahrnuje specifické úkoly a činnosti, které musí být dokončeny v rámci definovaného časového rámce, aby byl projekt úspěšně realizován. Ganttův diagram umožňuje sledovat pokrok jednotlivých úkolů a koordinovat práci týmu efektivně.

4.3.6 Náklady projektu

Tvorba rozpočtu projektu je zásadní součástí úspěšného řízení projektu, protože umožňuje efektivně alokovat a sledovat finanční zdroje potřebné pro dokončení projektu. Rozpočet pomáhá definovat očekávané náklady a příjmy spojené s projektem, což umožňuje projektovému týmu a stakeholderům zajistit dostatečné finanční prostředky a plánovat jejich využití.

Při tvorbě rozpočtu je důležité zohlednit všechny relevantní náklady a příjmy, jako jsou náklady na materiály, mzdy, provozní náklady, marketing a další související výdaje. Rozpočet by měl být pružný a realistický, aby dokázal reagovat na nečekané události a změny v průběhu projektu.

Správně sestavený a řízený rozpočet projektu pomáhá zajistit, že projekt bude dokončen včas a v rámci finančních možností, což minimalizuje rizika spojená s nedostatkem finančních prostředků a umožňuje lepší kontrolu nad vývojem projektu.

Personální zdroje

Požadované profese	Sazba za čld	Odhadovaná rámcová pracnost v čld	Rámcový náklad
Vedoucí projektu	4000	14	56000
Analytik	3200	7	22400
Programátor	2880	24	69120
Grafik	2240	4	8960
Tester	2400	10	24000
Systémový specialista	3200	4	12800
Dokumentarista	2000	5	10000
Celkem za požadované profese:		203280	

Predbežný rozpočet optimum

Company Data		Rate	
Pozadovaná návratnosť		10%	
Danová sadza		19%	

Prvotní investice do stránek	ROK	Predplatiteľa				
		1	2	3	4	5
Webhosting	10.000,00 Kč	1500	2700	4000	5500	7000
Marketing a reklama	80.000,00 Kč					
Vývoj a implementace	230.000,00 Kč					
Celkem investice	320.000,00 Kč					

Príjmy z provozu	ROK	Predplatiteľa				
		1	2	3	4	5
Predplatné		150.000,00 Kč	270.000,00 Kč	400.000,00 Kč	550.000,00 Kč	700.000,00 Kč
Provize z registrovaných klientů		30.000,00 Kč	100.000,00 Kč	200.000,00 Kč	250.000,00 Kč	280.000,00 Kč
Provize z reklam na portálu		25.000,00 Kč	25.000,00 Kč	25.000,00 Kč	25.000,00 Kč	25.000,00 Kč
Dodatečný příjem za poradenské služby		60.000,00 Kč	60.000,00 Kč	60.000,00 Kč	60.000,00 Kč	60.000,00 Kč
Publikace online		10.000,00 Kč	10.000,00 Kč	10.000,00 Kč	10.000,00 Kč	10.000,00 Kč
Total Benefits		275.000,00 Kč	465.000,00 Kč	695.000,00 Kč	895.000,00 Kč	1.075.000,00 Kč

Naklady	ROK	Predplatiteľa				
		1	2	3	4	5
Akce pro predplatitele, promoce		120.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč
Udržba portálu		80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč
Zákaznická podpora		150.000,00 Kč	150.000,00 Kč	150.000,00 Kč	150.000,00 Kč	150.000,00 Kč
Online reklama, vyhledavace, PPC		120.000,00 Kč	70.000,00 Kč	70.000,00 Kč	70.000,00 Kč	70.000,00 Kč
Pokuty za guerrilla marketing		20.000,00 Kč	50.000,00 Kč	0,00 Kč	0,00 Kč	0,00 Kč
Total Costs		490.000,00 Kč	430.000,00 Kč	380.000,00 Kč	380.000,00 Kč	380.000,00 Kč

Celkem	ROK	Predplatiteľa				
		1	2	3	4	5
cisty príjem		215.000,00 Kč	35.000,00 Kč	315.000,00 Kč	515.000,00 Kč	695.000,00 Kč
Dane hura na ne		40.850,00 Kč	6.650,00 Kč	59.850,00 Kč	97.850,00 Kč	132.050,00 Kč
Value after tax		174.150,00 Kč	28.350,00 Kč	255.150,00 Kč	417.150,00 Kč	562.950,00 Kč
Cash flow	320.000,00 Kč	174.150,00 Kč	28.350,00 Kč	255.150,00 Kč	417.150,00 Kč	562.950,00 Kč
Cumulative cash flow	320.000,00 Kč	494.150,00 Kč	465.800,00 Kč	210.650,00 Kč	206.500,00 Kč	769.450,00 Kč

Evaluation Metrics	
Net present value (NPV)	371.276,26 Kč
Internal rate of return (IRR)	28,16%
Návratnosť projektu v letech	3,50

Tabulka 15: Predbežný rozpočet – optimum

Predbežný rozpočet optimistický

Company Data		Rate	
Pozadovana navratnost		10%	
Danovazazba		19%	

Prvotni investice do stranek	ROK	Predplatitele				
		1	2	3	4	5
Webhosting	10.000,00 Kč	1800	3000	4200	6500	8000
Marketing a reklama	80.000,00 Kč					
Vyvoj a implentace	230.000,00 Kč					
Celkem investice	320.000,00 Kč					

Prijmy z provozu	ROK	Predplatitele				
		1	2	3	4	5
Predplatne		180.000,00 Kč	300.000,00 Kč	420.000,00 Kč	650.000,00 Kč	800.000,00 Kč
Provize z registrovaných klientů		30.000,00 Kč	100.000,00 Kč	200.000,00 Kč	250.000,00 Kč	280.000,00 Kč
Provize z reklam na portalu		25.000,00 Kč	25.000,00 Kč	25.000,00 Kč	25.000,00 Kč	25.000,00 Kč
Dodatečný příjem za poradenské služby		60.000,00 Kč	60.000,00 Kč	60.000,00 Kč	60.000,00 Kč	60.000,00 Kč
Publikace online		10.000,00 Kč	10.000,00 Kč	10.000,00 Kč	10.000,00 Kč	10.000,00 Kč
Total Benefits		305.000,00 Kč	495.000,00 Kč	715.000,00 Kč	995.000,00 Kč	1.175.000,00 Kč

Naklady	ROK	Predplatitele				
		1	2	3	4	5
Akce pro predplatitele, promoce		120.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč
Udržba portalu		80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč
Zakaznicka podpora		150.000,00 Kč	150.000,00 Kč	150.000,00 Kč	150.000,00 Kč	150.000,00 Kč
Online reklama, vyhledavace, PPC		120.000,00 Kč	70.000,00 Kč	70.000,00 Kč	70.000,00 Kč	70.000,00 Kč
Pokuty za guerrilla marketing		20.000,00 Kč	50.000,00 Kč	0,00 Kč	0,00 Kč	0,00 Kč
Total Costs		490.000,00 Kč	430.000,00 Kč	380.000,00 Kč	380.000,00 Kč	380.000,00 Kč

Celkem	ROK	Predplatitele				
		1	2	3	4	5
cisty prijem		185.000,00 Kč	65.000,00 Kč	335.000,00 Kč	615.000,00 Kč	795.000,00 Kč
Dane hura na ne		35.150,00 Kč	12.350,00 Kč	63.650,00 Kč	116.850,00 Kč	151.050,00 Kč
Value after tax		149.850,00 Kč	52.650,00 Kč	271.350,00 Kč	498.150,00 Kč	643.950,00 Kč
Cash flow	320.000,00 Kč	149.850,00 Kč	52.650,00 Kč	271.350,00 Kč	498.150,00 Kč	643.950,00 Kč
Cumulative cash flow	320.000,00 Kč	469.850,00 Kč	417.200,00 Kč	145.850,00 Kč	352.300,00 Kč	996.250,00 Kč

Evaluation Metrics	
Net present value (NPV)	531.239,83 Kč
Internal rate of return (IRR)	35,15%
Navratnost projektu v letech	3,29

Tabulka 16: Předběžný rozpočet – optimistický

Predbežny rozpočet pesimisticky

Company Data		Rate	
Pozadovana navratnost		10%	
Danova sazba		19%	

Prvotni investice do stranek		Predplatitele				
ROK		1	2	3	4	5
Webhosting	10.000,00 Kč	800	1500	2000	3500	6000
Marketing a reklama	80.000,00 Kč					
Vyvoj a implentace	230.000,00 Kč					
Celkem investice	320.000,00 Kč					

Prijmy z provozu		ROK				
		1	2	3	4	5
Predplatne		80.000,00 Kč	150.000,00 Kč	200.000,00 Kč	350.000,00 Kč	600.000,00 Kč
Provize z registrovaniych klientu		30.000,00 Kč	100.000,00 Kč	200.000,00 Kč	250.000,00 Kč	280.000,00 Kč
Provize z reklam na portalu		25.000,00 Kč	25.000,00 Kč	25.000,00 Kč	25.000,00 Kč	25.000,00 Kč
Dodatecny prijem za poradenske sluzby		60.000,00 Kč	60.000,00 Kč	60.000,00 Kč	60.000,00 Kč	60.000,00 Kč
Publikace online		10.000,00 Kč	10.000,00 Kč	10.000,00 Kč	10.000,00 Kč	10.000,00 Kč
Total Benefits		205.000,00 Kč	345.000,00 Kč	495.000,00 Kč	695.000,00 Kč	975.000,00 Kč

Naklady		ROK				
		1	2	3	4	5
Akce pro predplatitele, promoce		120.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč
Udrzba portalu		80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč	80.000,00 Kč
Zakaznicka podpora		150.000,00 Kč	150.000,00 Kč	150.000,00 Kč	150.000,00 Kč	150.000,00 Kč
Online reklama, vyhledavace, PPC		120.000,00 Kč	70.000,00 Kč	70.000,00 Kč	70.000,00 Kč	70.000,00 Kč
Pokuty za guerrilla marketing		20.000,00 Kč	50.000,00 Kč	0,00 Kč	0,00 Kč	0,00 Kč
Total Costs		490.000,00 Kč	430.000,00 Kč	380.000,00 Kč	380.000,00 Kč	380.000,00 Kč

Celkem		ROK				
		1	2	3	4	5
cisty prijem		285.000,00 Kč	85.000,00 Kč	115.000,00 Kč	315.000,00 Kč	595.000,00 Kč
Dane hura na ne		54.150,00 Kč	16.150,00 Kč	21.850,00 Kč	59.850,00 Kč	113.050,00 Kč
Value after tax		230.850,00 Kč	68.850,00 Kč	93.150,00 Kč	255.150,00 Kč	481.950,00 Kč
Cash flow		320.000,00 Kč	230.850,00 Kč	68.850,00 Kč	93.150,00 Kč	255.150,00 Kč
Cumulative cash flow		320.000,00 Kč	550.850,00 Kč	619.700,00 Kč	526.550,00 Kč	271.400,00 Kč

Evaluation Metrics	
Net present value (NPV)	43.255,57 Kč
Internal rate of return (IRR)	7,85%
Navratnost projektu v letech	4,56

Tabulka 17: Předběžný rozpočet – pesimistický

Předběžný rozpočet je nástroj, který projektovým manažerům a týmům pomáhá odhadnout náklady na projekt v jeho raných fázích. Tři typy předběžných rozpočtů – optimistický, pesimistický a optimum – představují různé scénáře odhadů nákladů, které pomáhají týmu připravit se na různé situace a nečekané události.

Optimistický rozpočet předpokládá, že vše půjde podle plánu a že nebudou žádné významné problémy ani zpoždění. Tento rozpočet zohledňuje nejnižší možné náklady a ideální časový rámec.

Pesimistický rozpočet naopak předpokládá, že projekt bude čelit významným překážkám, které způsobí zpoždění a zvýší náklady. Tento rozpočet je založen na nejvyšších možných nákladech a nejdelším časovém rámci.

Rozpočet optimum je kompromis mezi optimistickým a pesimistickým rozpočtem. Tento rozpočet zohledňuje pravděpodobné náklady a časový rámec, které zahrnují mírné

zpoždění a neočekávané události. Je to nejrealističtější odhad nákladů a časového rámce, který by měl být použit pro plánování a řízení projektu.

Tím, že projektový tým zohlední všechny tři typy rozpočtů, získává lepší představu o možných scénářích a rizicích, což mu umožní lépe plánovat a přizpůsobovat se nečekaným událostem v průběhu projektu.

4.3.7 Výběr webhostingu

Výběr hostingu byl uvážen na základě rozhodovací tabulky, která zahrnuje váhy pro každé kritérium. Cena má váhu 0,7 (70%), zatímco ostatní kritéria mají váhu 0,3 (30%).

Skóre pro každý typ hostingu je následující:

Kritérium	Váha	Full server hosting	Cloud hosting	Shared hosting	VPS hosting
Cena	0,7	2	3	5	4
Výkon	0,3	5	4	2	3
Flexibilita	0,3	4	5	1	3
Škálovatelnost	0,3	3	5	1	4
Bezpečnost	0,3	5	4	2	3
Správa serveru	0,3	2	3	5	4
Podpora	0,3	4	4	3	4
SLA / Uptime	0,3	4	5	2	3

Tabulka 18: Rozhodovací tabulka webhostingu

Aby byly zohledněny váhy, je třeba vypočítat vážené skóre pro každý typ hostingu:

- Full server hosting: $(2 \times 0,7) + (5 + 4 + 3 + 5 + 2 + 4 + 4) \times 0,3 = 9,1$
- Cloud hosting: $(3 \times 0,7) + (4 + 5 + 5 + 4 + 3 + 4 + 5) \times 0,3 = 11,6$
- Shared hosting: $(5 \times 0,7) + (2 + 1 + 1 + 2 + 5 + 3 + 2) \times 0,3 = 10,5$
- VPS hosting: $(4 \times 0,7) + (3 + 3 + 4 + 3 + 4 + 4 + 3) \times 0,3 = 10,2$

Typ hostingu	Vážené skóre
Full server hosting	9,1
Cloud hosting	11,6
Shared hosting	10,5
VPS hosting	10,2

Tabulka 19: Vážené skóre dle typu hostingu

Podle výsledků je nejlepší volbou cloud hosting. Tato metoda zohledňuje důraz na cenu (70%) a zároveň bere v úvahu ostatní důležitá kritéria (30%).

4.3.8 Rizika

4.3.8.1 Identifikace rizik projektu

ID	Příčina	Riziko	Popis dopadu rizika
1.	Dynamický vývoj trhu kryptoměn	Neaktuálnost obsahu kryptoměn	Rychle zanikající a vnikající nové kryptoměny
2.	Legislativní změny v oblasti kryptoměn a jejich správy	Zavádění nových opatření a podmínek v oblasti kryptoměn	Zákaz poskytování některých služeb v oblasti kryptoměn
3.	Příliš velké výkyvy a zneužívání kryptoměn	Ztráta důvěry uživatelů v kryptoměny	Nepotřebnost webové aplikace
4.	Příliš úzce definována cílová skupina	Nepokrytí nákladů projektu	Potřeba úpravy nebo zrušení webové aplikace
5.	Výpadek webové stránky	Nedostupnost služby	Uživatelé nebudou schopni zobrazit webovou stránku
6.	Krach dodavatele	Ztráta služby	Trvalá nedostupnost webové stránky
7.	Výpadek cloud	Nedostupnost služby	Uživatelé nebudou schopni zobrazit webovou stránku
8.	Nefunkční funkcionality webu	Nedostupnost funkcionalit	Uživatelé nebudou schopni zobrazit webovou stránku
9.	Nedostupnost databáze	Nefunkční předplatné pro uživatele	Nemožnost se přihlásit a používat předplacené funkcionality

Tabulka 20: Seznam rizik projektu

4.3.8.2 Návrh ošetření rizik projektu

ID	Riziko	Dopad rizika	P	R	Preventivní opatření – PO	Osoba zodpovědná za realizaci PO
1.	Neaktuálnost obsahu kryptoměn	Rychle zanikající a vnikající nové kryptoměny	8	6	Příprava api pro stahování aktuálního obsahu	Analytik
2.	Zavádění nových opatření a podmínek v oblasti kryptoměn	Zákaz poskytování některých služeb v oblasti kryptoměn	6	8	Nezávislost na obchodování nebo s tím nějak spojených služeb	Analytik
3.	Ztráta důvěry uživatelů v kryptoměny	Nepotřebnost webové aplikace	4	7	Rozšíření webové aplikace o další funkcionality	Vedoucí projektu
4.	Nepokrytí nákladů projektu	Potřeba úpravy nebo zrušení webové aplikace	6	8	Zavedení většího množství placených reklam	Vedoucí projektu
5.	Nedostupnost služby	Uživatelé nebudou schopni zobrazit webovou stránku	8	9	Zajištění redundanci řešení	Systémový specialista
6.	Ztráta služby	Trvalá nedostupnost webové stránky	2	10	Zajištění redundanci řešení	Systémový specialista
7.	Nedostupnost cloud	Uživatelé nebudou schopni zobrazit webovou stránku	3	9	Zajištění vyšší třídy poskytovaného cloud	Systémový specialista
8.	Nedostupnost funkcionalit	Uživatelé nebudou schopni zobrazit webovou stránku	8	9	Zajištění redundanci řešení	Systémový specialista
9.	Nefunkční předplatné pro uživatele	nemožnost se přihlásit a používat předplacené funkcionality	6	9	Zajištění redundanci řešení	Systémový specialista

Tabulka 21: Opatření rizik projektu

5 Výsledky vlastní práce

Výstupem diplomové práce je ucelený návrh webového portálu navržený na základě doporučení získaných z literární rešerše. Tato rešerše poskytla soubor doporučení pro tvorbu webových stránek ve zvoleném prostředí, což bylo následně aplikováno při vývoji funkčního webu.

Návrh webového portálu počítá s využitím moderních technologií, jako jsou HTML5, Bootstrap, Javascript, PHP a MySQL. Tento výběr technologií zajišťuje kompatibilitu, flexibilitu a udržitelnost řešení. Celý prototyp byl úspěšně otestován na hostingovém serveru, což potvrzuje jeho správnou funkčnost a stabilitu.

Na základě teoretické části a nezávislého hodnocení dokončeného řešení lze konstatovat, že diplomová práce splnila své cíle. Vytvořený webový portál s responzivním layoutem vhodným i pro mobilní zařízení představují hodnotný přínos pro oblast kryptoměn mající schopnost oslovit širokou skupinu uživatelů a stát se jedním z hlavních zdrojů informací a nástrojů v oblasti kryptoměn.

6 Závěr

V rámci diplomové práce byl úspěšně navržen a realizován webový portál zaměřený na kryptoměny, který splňuje všechny dílčí cíle. Kryptokalkulačka poskytuje uživatelům přehledný nástroj pro přepočítání kryptoměn dle aktuálního i historického kurzu. Uživatelská sekce umožňuje registrovaným uživatelům vytvářet přehledy dle vlastních preferencí, zatímco informační sekce poskytuje ucelený zdroj informací z oblasti kryptoměn s důrazem na bezpečnost. Jako další výstup práce byl navržen UX/UI návrh mobilní aplikace, který poskytuje uživatelům přístup k funkcím portálu prostřednictvím mobilního rozhraní.

Dodatečným benefitem práce je stanovení business modelu, který analyzuje tržní potenciál, konkurenci a možnosti monetizace projektu. Vlastní práce zahrnovala design, návrh a vývoj prototypu portálu, kde byla ověřena funkčnost celého řešení. Na zpracování celého projektu byly použity postupy a zkušenosti získané v průběhu studia. Byl sestaven předpokládaný rozpočet projektu včetně očekávané ceny vytvoření a provozu portálu.

Pokud vycházíme z dat jako Eurostat a ČSÚ, tak cílový segment potenciálních zákazníků dle příjmových skupin by byl okolo 30% z 2,4mil, což odpovídá 720000 lidí. Při 1 % penetraci možných uživatelů v ČR a výši měsíčního předplatného začínajících zhruba na 50 Kč, tento projekt vypadá jako výnosný s velmi dobrou návratností. Ke skutečné realizaci projektu by zcela jistě přispěly kroky jako oslovení potenciálních investorů a sestavení realizačního týmu.

V závěru lze konstatovat, že diplomová práce úspěšně splnila všechny stanovené cíle a přinesla hodnotný přínos pro oblast kryptoměn. Navržený webový portál má potenciál přilákat širokou škálu uživatelů a stát se jedním z hlavních zdrojů informací a nástrojů v oblasti kryptoměn. Navíc byl stanoven udržitelný business model, který umožňuje další rozvoj a růst projektu v dlouhodobém horizontu. Diplomová práce by mohla představovat významný přínos pro oblast kryptoměn a nabízí výnosnou příležitost pro budoucí investory a uživatele.

7 Seznam použitých zdrojů

- [1] LEINER Barry M., CERF Vinton G., CLARK David D., KAHN Robert E., KLEINROCK Leonard, LYNCH Daniel C., POSTEL Jon, ROBERTS Larry G., WOLF Stephen. *Brief History of the Internet*. Internet Society [online]. 1997. Dostupné z: https://www.internetsociety.org/wp-content/uploads/2017/09/ISOC-History-of-the-Internet_1997.pdf
- [2] *The Brief Timeline: The History of Web Browsers*. Vestra Inet [online]. Dostupné z: <https://vestrainet.com/the-brief-timeline-the-history-of-web-browsers.html>
- [3] *What is Client-Server Architecture?*. W3SCHOOL; World Wide Web Consortium [online]. Dostupné z: <https://www.w3schools.in/what-is-client-server-architecture>
- [4] *An overview of HTTP*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- [5] *Comparison of Web Browsers*. Eylenburg GitHub [online]. Dostupné z: https://eylenburg.github.io/browser_comparison.htm
- [6] *Mobile Browser Market Share Worldwide*. Global Stats [online]. Dostupné z: <https://gs.statcounter.com/browser-market-share/mobile/worldwide>
- [7] SUBEDI Hari. *Types Of Servers: Shared, Dedicated, Cloud, and Virtual*. Jones IT [online]. Dostupné z: <https://www.itjones.com/blogs/types-of-servers-shared-dedicated-cloud-and-virtual>
- [8] *15 Different Types of Servers in Computing*. Sunny Valley Networks [online]. Dostupné z: <https://www.sunnyvalley.io/docs/network-basics/types-of-servers>
- [9] *What is a Web server?*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
- [10] *Types of Network Protocols and their uses*. W3SCHOOL; World Wide Web Consortium [online]. Dostupné z: <https://www.w3schools.in/types-of-network-protocols-and-their-uses>
- [11] *HTTP Guide*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP>
- [12] POSTEL J., REYNOLDS J. RFC 959: *File Transfer Protocol (FTP)*. ISI, Network Working Group [online]. 1985. Dostupné z: <https://www.rfc-editor.org/rfc/rfc959>
- [13] KRUG, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd Edition)*. USA: New Riders, Peachpit Press, 2013. ISBN 978-0-32196-551-6
- [14] *WCAG 2 Overview*. W3C WAI (Web Access Initiative); World Wide Web Consortium [online]. Dostupné z: <https://www.w3.org/WAI/standards-guidelines/wcag/>
- [15] *Responsive Web Design Basics*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design
- [16] *HTML: HyperText Markup Language*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>

- [17] BERNERS-LEE T. *RFC 1866: Hypertext Markup Language - 2.0*. MIT/W3C, Network Working Group [online]. 1995. Dostupné z: <https://www.rfc-editor.org/rfc/rfc1866>
- [18] *HTML Living Standard*. WHATWG (Web Hypertext Application Technology Working Group) [online]. Dostupné z: <https://html.spec.whatwg.org/multipage/>
- [19] CASTRO, E. -- HYSLOP, B. *HTML5 a CSS3*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- [20] *CSS: Cascading Style Sheets*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [21] *The Evolution of CSS in 3 Decades*. BYBY DEV [online]. Dostupné z: <https://byby.dev/css-evolution>
- [22] AN E. *What's a Web Framework and Why Are They Useful?*. CareerFoundry [online]. Dostupné z: <https://careerfoundry.com/en/blog/web-development/responsiveness-with-a-front-end-framework/>
- [23] SUFYAN, bin Uzayr. *CSS Frameworks: The Ultimate Guide*. USA: CRC Press, 2023. ISBN 978-10-324-1322-8
- [24] *About Bootstrap*. MIT, Bootstrap team [online]. Dostupné z: <https://getbootstrap.com/docs/5.0/about/overview/>
- [25] JENSEN, S., J. *The Missing Bootstrap 5 Guide*. Birmingham, UK: Packt Publishing Ltd., 2021. ISBN 978-1-80107-643-2.
- [26] SUFYAN, bin Uzayr. *Bootstrap: The Ultimate Guide*. USA: CRC Press, 2023. ISBN 978-1-03231-361-0
- [27] KANTOR, I. *The Modern JavaScript Tutorial*. Techlead LLC [online]. Dostupné z: <https://javascript.info/>
- [28] *jQuery Learning Center*. jQuery Foundation; OpenJS Foundation [online]. Dostupné z: <https://learn.jquery.com/>
- [29] *Node.js*. Node.js; OpenJS Foundation [online]. Dostupné z: <https://nodejs.org/>
- [30] *Understanding client-side JavaScript frameworks*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks
- [31] *History of PHP*. The PHP Group [online]. Dostupné z: <https://www.php.net/manual/en/history.php.php>
- [32] BIERER, D. *PHP 8 Programming Tips, Tricks and Best Practices*. Birmingham, UK: Packt Publishing Ltd., 2021. ISBN 978-1-80107-187-1.
- [33] *Historie Nette*. NETTE.ORG [online]. Dostupné z: <https://nette.org/en/history>
- [34] *Dokumentace Nette*. NETTE.ORG [online]. Dostupné z: <https://doc.nette.org/>
- [35] *Which PHP Framework Should You Use in 2023?* The Dev Drawer, DEV Community [online]. Dostupné z: <https://dev.to/thedevdrawer/which-php-framework-should-you-use-in-2023-41m7>

- [36] MEIER, A., KAUFMANN, M. *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management*. Německo: Springer Fachmedien Wiesbaden, 2019. ISBN 978-3-65824-548-1
- [37] *MySQL Documentation*. Oracle Corporation [online]. Dostupné z: <https://dev.mysql.com/doc/>
- [38] WELLING, L. -- THOMSON, L. *Mistrovství PHP a MySQL*. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1
- [39] *PHP MySQL API Documentation*. The PHP Group [online]. Dostupné z: <https://www.php.net/manual/en/book.mysql.php>
- [40] *Nette Database Core*. NETTE.ORG [online]. Dostupné z: <https://doc.nette.org/cs/database/core>
- [41] *Web security*. MDN Web Docs, Mozilla Foundation [online]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/Security>
- [42] *What is an API?*. Red Hat, Inc. [online]. Dostupné z: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
- [43] *Introduction to Content Management System (CMS)*. W3SCHOOL, World Wide Web Consortium [online]. Dostupné z: <https://www.w3schools.in/wordpress/introduction-to-content-management-system-cms>
- [44] *Search Engine Optimization (SEO) Starter Guide*. Google Search Central Documentation, Google Developers, Google LLC [online]. Dostupné z: <https://developers.google.com/search/docs/fundamentals/seo-starter-guide>
- [45] JONES, E. *A Brief History of Cryptocurrency*. CryptoVantage Guides [online]. Dostupné z: <https://www.cryptovantage.com/guides/a-brief-history-of-cryptocurrency/>
- [46] NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Bitcoin Project [online]. Dostupné z: <https://bitcoin.org/bitcoin.pdf>
- [47] FRANKENFIELD, J.; MURRY, C.; KVILHAUG, S. *Cryptocurrency Explained With Pros and Cons for Investment*. Investopedia, IAC Inc [online]. Dostupné z: <https://www.investopedia.com/terms/c/cryptocurrency.asp>
- [48] *Survey: Attitudes Towards Crypto Assets and Technology*. Eurobarometer, European Commission, 2021 [online]. Dostupné z: https://ec.europa.eu/finance/docs/surveys/2021/eb_2021-attitudes-towards-crypto-assets-and-technology_en.pdf
- [49] BLANDIN, A.; DR. PIETERS, G.; WU, Y.; EISERMANN, T.; DEK, A.; TAYLOR, S.; NJOKI, D. *3rd Global Cryptoasset Benchmarking Study*. Cambridge Centre for Alternative Finance, University of Cambridge, 2020 [online]. Dostupné z: <https://www.jbs.cam.ac.uk/wp-content/uploads/2021/01/2021-ccaf-3rd-global-cryptoasset-benchmarking-study.pdf>
- [50] *Průměrná hrubá měsíční mzda a medián mezd*. Veřejná databáze, Český Statistický úřad [online]. Dostupné z: <https://vdb.czso.cz/vdbvo2/faces/index.jsf?page=vystup-objekt-parametry&pvo=MZD07&sp=A&pvokc=&katalog=30852&z=T>
- [51] *Heuristic Evaluation*. The Usability Body of Knowledge, User Experience Professionals' Association [online]. Dostupné z: <https://usabilitybok.org/heuristic-evaluation>

- [52] NIELSEN, J. *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group [online]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [53] *WAVE Web Accessibility Evaluation Tools*. WebAIM, Institute for Disability Research, Policy, and Practice [online]. Dostupné z: <https://wave.webaim.org/>
- [54] *Google Lighthouse Documentation*. Google Developers, Google LLC [online]. Dostupné z: <https://developer.chrome.com/docs/lighthouse/>
- [55] *Nu Html Checker*. World Wide Web Consortium [online]. Dostupné z: <https://validator.w3.org/nu/>
- [56] *CSS Validation Service*. World Wide Web Consortium [online]. Dostupné z: <https://jigsaw.w3.org/css-validator/>

8 Seznam obrázků, tabulek, grafů a zkratk

8.1 Seznam obrázků

Obrázek 1: Informační architektura aplikace	80
Obrázek 2: Návrh desktopového rozhraní	81
Obrázek 3: Pokračování úvodní stránky	82
Obrázek 4: Návrh textového rozložení	82
Obrázek 5: Návrh uživatelského rozhraní	83
Obrázek 6: Návrh přihlašovací obrazovky	84
Obrázek 7: Soubor návrhů mobilního rozhraní	86
Obrázek 8: Realizace: Úvodní stránka	92
Obrázek 9: Realizace: Kalendář	93
Obrázek 10: Realizace: Registrační formulář	93
Obrázek 11: Realizace: Okno přihlášení	94
Obrázek 12: Realizace: Uživatelský účet	94
Obrázek 13: Realizace: Přehled oblíbených položek	95
Obrázek 14: Realizace: Mobilní zobrazení	96
Obrázek 15: Realizace: Mobilní zobrazení – pokračování	97
Obrázek 16: JavaScript výběr ze seznamu	98
Obrázek 17: Javascript kontrola pole	102
Obrázek 18: Výsledky testu WAVE s CSS	109
Obrázek 19: Výsledky testu WAVE bez CSS	109
Obrázek 20: Výsledky testu WAVE po opravě chyb	112
Obrázek 21: Výsledky testu výkonnosti pomocí Google Lighthouse	112
Obrázek 22: Technologická architektura	115
Obrázek 23: The Business Model Canvas	133
Obrázek 24: Diagram aktéři případů užití.	142
Obrázek 25: Diagram případů užití - kryptokalkulačka.	143
Obrázek 26: Harmonogram projektu	149

8.2 Seznam tabulek

Tabulka 1: Seznam použitých heuristik	105
Tabulka 2: Seznam typických úkolů	105
Tabulka 3: Přehled nálezů heuristické analýzy	107
Tabulka 4: Přehled nálezů WAVE analýzy	111
Tabulka 5: Přehled nálezů HTML Validátoru	114
Tabulka 6: Projektový plán	129
Tabulka 7: Cíle projektu	130
Tabulka 8: Registrace – hlavní scénář	144
Tabulka 9: Registrace – alternativní scénář 1.	144
Tabulka 10: Registrace – alternativní scénář 2.	145
Tabulka 11: Zobrazení kurzu kryptoměn – hlavní scénář	145
Tabulka 12: Zobrazení kurzu kryptoměn – alternativní scénář 1	145
Tabulka 13: Zobrazení kurzu kryptoměn – alternativní scénář 2	146
Tabulka 14: Zobrazení kurzu kryptoměn – alternativní scénář 3	146
Tabulka 15: Předběžný rozpočet – optimum	151
Tabulka 16: Předběžný rozpočet – optimistický	152
Tabulka 17: Předběžný rozpočet – pesimistický	153
Tabulka 18: Rozhodovací tabulka webhostingu	154
Tabulka 19: Vážené skóre dle typu hostingu	155
Tabulka 20: Seznam rizik projektu	156
Tabulka 21: Opatření rizik projektu	157

8.3 Seznam grafů

Odkazovaný seznam grafů

8.4 Seznam použitých zkratk

Soupis a definování zkratk (vyskytuje-li se jich v textu velké množství)

9 Přílohy

Odkazovaný seznam příloh