

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Bakalářská práce**

**Analýza programovacích jazyků pro tvorbu webu**

**Matěj Bílek**

**© 2023 ČZU v Praze**



# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Matěj Bílek

Informatika

Název práce

**Analýza programovacích jazyků pro tvorbu webu**

Název anglicky

**Analysis of programming languages for web development**

---

### Cíle práce

Cílem teoretické části bakalářské práce je komplexně popsat téma tvorby webových stránek, službu World Wide Web a klíčové prvky při tvorbě webových stránek: značkovací jazyk HTML a kaskádové styly. Dále práce analyzuje a porovnává jednotlivé programovací jazyky používané pro tvorbu webových stránek, jejich historii, výhody a nevýhody, funkčnost, ilustrovanou na praktických ukázkách kódu, a využitelnost daného jazyka v praxi.

Cílem praktické části práce je na základě syntézy poznatků získaných v teoretické části práce shrnout klady a zápory jednotlivých programovacích jazyků a na základě výsledků shrnutí vybrat jeden programovací jazyk a použít ho při tvorbě ukázkového webu, který bude demonstrovat jeho funkcionality a možnosti využití.

### Metodika

Metodika bakalářské práce je založena na shromáždění, studiu a analýze odborných informačních zdrojů. Na základě syntézy získaných teoretických poznatků budou shrnuty klady a zápory jednotlivých programovacích jazyků a vybrán jeden jazyk, který bude v praktické části práce použit na tvorbu ukázkového webu, demonstrujícího jeho funkce a možné využití v praxi. Ukázkový web a jeho podrobně popsany a okomentovaný kód bude výstupem této bakalářské práce.

**Doporučený rozsah práce**

30-60 stran

**Klíčová slova**

WWW, vývoj webu, programovací jazyky, front-end, back-end, HTML, CSS, PHP, Python, Javascript, Ruby, ASP.NET

---

**Doporučené zdroje informací**

KRUG, Steve. Web design – nenuťte uživatele přemýšlet!. 2., aktualiz. vyd. Brno: Computer Press, 2006. ISBN 8025112918.

LUTZ, Mark a David ASCHER. Naučte se Python: Pohotová příručka. Praha: Grada, 2003. ISBN 802470367x.

MUSCIANO, Chuck a Bill KENNEDY. HTML a XHTML: kompletní průvodce. Praha: Computer Press, 2000. ISBN 9788072264070.

PEHLIVANIAN, Ara a Don NGUYEN. JavaScript okamžitě. 2. vydání. Brno: Computer Press, 2021. ISBN 9788025150252.

POKORNÝ, Martin. PHP nejen pro začátečníky. Kralice na Hané: Computer Media, 2005. ISBN 8086686388.

---

**Předběžný termín obhajoby**

2022/23 LS – PEF

**Vedoucí práce**

Ing. Dana Vynikarová, Ph.D.

**Garantující pracoviště**

Katedra informačního inženýrství

---

Elektronicky schváleno dne 31. 10. 2022

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

Elektronicky schváleno dne 24. 11. 2022

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 25. 12. 2022

### **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci „Analýza programovacích jazyků pro tvorbu webu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15. 3. 2023

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D. za konzultaci a odborné vedení této bakalářské práce, cenné rady a trpělivost.

# Analýza programovacích jazyků pro tvorbu webu

## Abstrakt

Bakalářská práce se zabývá tvorbou dynamických webových stránek. Popisuje technologie zajišťující vytváření a fungování webů a představuje analýzu pěti skriptovacích jazyků, kandidátů pro použití v praktické části. Na základě shrnutí poznatků z teoretické části je pro praktickou ukázkou vybrán skriptovací jazyk PHP (*Hypertext Preprocessor*). Pro demonstraci funkcí jazyka je vytvořena jednoduchá webová aplikace s tematikou fitness. Vzhled aplikace udává jazyk HTML a responzivní CSS framework Bootstrap, funkcionalitu zajišťují vnořené PHP skripty.

V práci byla vytvořena webová aplikace *Fitter & Happier* nabízející uživatelům možnost se zaregistrovat, přihlašovat se zadanými údaji, zakládat osobní milníky a vést si statistiku o proběhlých cvičeních. Vzhledové i funkční prvky jsou podrobně vysvětleny v praktické části práce prostřednictvím okomentovaných výpisů zdrojového kódu. Na závěr jsou nabídnuta možná vylepšení aplikace, inspirace pro budoucí projekty.

**Klíčová slova:** webová aplikace, front-end, back-end, test programovacích jazyků, HTML, CSS, PHP, Python, JavaScript, Ruby, ASP.NET

# Analysis of programming languages for web development

## Abstract

Bachelor thesis is focused on the creation of dynamic web pages. It describes the technologies that ensure the creation and functioning of websites and presents an analysis of five scripting languages, candidates for use in the practical part. Based on the summary of findings from the theoretical part, the scripting language PHP (*Hypertext Preprocessor*) is selected for the practical demonstration. A simple web application with a fitness theme is created to demonstrate the language features. The appearance of the application is determined by HTML and responsive CSS framework Bootstrap, the functionality is provided by embedded PHP scripts.

In this thesis, a web application *Fitter & Happier* was created offering users the possibility to register, log in with the entered data, set personal milestones and keep statistics on the exercises performed. The design and functional elements are explained in detail in the practical part of the thesis through annotated source code listings. Finally, possible improvements to the application are offered, as well as inspiration for future projects.

**Keywords:** web application, front-end, back-end, programming language test, HTML, CSS, PHP, Python, JavaScript, Ruby, ASP.NET



# Obsah

<b>1 Úvod .....</b>	<b>11</b>
<b>2 Cíl práce a metodika .....</b>	<b>13</b>
2.1 Cíl práce.....	13
2.2 Metodika.....	13
<b>3 Teoretická východiska .....</b>	<b>14</b>
3.1 WWW.....	14
3.1.1 Klient.....	15
3.1.2 Server.....	16
3.1.3 URL.....	17
3.1.4 HTTP.....	17
3.2 HTML.....	18
3.3 CSS.....	20
3.4 Skriptovací programovací jazyky .....	21
3.4.1 Charakteristika .....	21
3.4.2 Kritéria výběru.....	22
3.4.3 ASP.NET (Core).....	23
3.4.4 JavaScript.....	25
3.4.5 PHP .....	27
3.4.6 Python.....	30
3.4.7 Ruby .....	33
3.5 Tvorba webových stránek.....	35
3.5.1 Zásady .....	35
3.5.2 Postup.....	36
3.5.3 Výběr jazyka pro praktickou ukázkou .....	37
3.5.4 Použitý software.....	39
<b>4 Vlastní práce .....</b>	<b>43</b>
4.1 Front-end .....	44
4.1.1 Návrh webu.....	44
4.1.2 Struktura webové aplikace.....	45
4.1.3 HTML hlavička.....	45
4.1.4 Záhlaví a navigační lišta .....	46
4.1.5 Footer .....	48
4.1.6 Domovská stránka.....	49
4.1.7 Stránka „O nás“.....	50
4.1.8 Registrační formulář.....	51
4.2 Back-end.....	52

4.2.1	Registrace .....	52
4.2.2	Přihlášení .....	55
4.2.3	Kontrola přístupu .....	56
4.2.4	Ukládání vstupů .....	56
4.2.5	Zobrazení vstupů .....	57
4.2.6	Smazání vstupů .....	58
4.3	Databáze .....	59
4.3.1	Tabulka Users .....	59
4.3.2	Tabulka Goals .....	59
4.3.3	Tabulka Exercises .....	60
4.4	Výsledná podoba aplikace .....	61
<b>5</b>	<b>Zhodnocení výsledků .....</b>	<b>62</b>
<b>6</b>	<b>Závěr .....</b>	<b>64</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>66</b>
<b>8</b>	<b>Seznam obrázků, tabulek a zkratk.....</b>	<b>70</b>
8.1	Seznam obrázků .....	70
8.2	Seznam výpisů zdrojového kódu .....	70
8.3	Seznam tabulek .....	70
8.4	Seznam použitých zkratk.....	71
<b>Přílohy</b>	<b>.....</b>	<b>72</b>

# 1 Úvod

Komunikace je základním stavebním kamenem lidské společnosti. Již první civilizace řešily potřebu funkčního předávání informací mezi členy pro zajištění chodu společenství. Ani v dnešní době se její význam nesnížil, ba právě naopak. Na schopnosti porozumět ostatním, umění předat své myšlenky a názory patřičným lidem a navázání a udržení mezilidských kontaktů závisí téměř veškerý úspěch či neúspěch jednotlivce ve společnosti.

S rozvojem informačních technologií a s příchodem internetu přichází i rozmach možností komunikace. Internet umožnil spojení s kýmkoliv na světě téměř v reálném čase, tedy za předpokladu, že příjemce disponoval připojením k internetové síti. V dnešní době nesou sdělovací funkci primárně webové stránky. Snaží se uživatele upoutat, předat mu svou zprávu a také reagovat na jeho vstupy a přizpůsobovat jim svou výslednou podobu. Roli prostředníka zajišťujícího hladký průběh komunikace mezi uživatelem a webovou stránkou sehrávají skriptovací programovací jazyky, které se starají o interaktivitu nabízeného obsahu.

Bakalářská práce se zabývá tématem tvorby dynamických webových stránek. Popisuje technologie zajišťující vytváření a fungování webů, dále představuje analýzu pěti skriptovacích jazyků, kandidátů pro použití v praktické části. Pro demonstraci vybrané technologie je vytvořena jednoduchá webová stránka tvořena na front-endu jazykem HTML doplněným o CSS a JavaScript framework Bootstrap, pro back-end je využito skriptování ve vybraném jazyce. Zde jsou demonstrovány možnosti jazyka, které nabízí při využití v praxi webového designu. Aplikace s názvem *Fitter & Happier* se věnuje odvětví zdraví a fitness, umožňuje uživatelům se na stránce registrovat, přihlašovat se zadanými údaji, vytvářet si vlastní cíle a zaznamenávat průběh své cvičební rutiny.

Důvodů pro volbu tématu této bakalářské práce je možné najít v současnosti mnoho. Webové stránky a jejich správa a tvorba nejsou již výsadou pouze hrstky nadšenců, ale prostoupily do téměř všech odvětví lidského působení. Jednotlivci, živnostníci, firmy, nadace a další zájmové skupiny, všichni využívají internet pro co nejlepší sebeprezentaci. Proto je znalost problematiky tvorby a údržby webů velmi ceněna a poptávka po ní se každým rokem zvyšuje. S vývojem technologií a příchodu nových softwarových řešení je možné libovolně vylepšovat a aktualizovat již existující projekty, nehrozí tedy zastarání a úpadek tohoto v současné době velice perspektivního odvětví.

Práce je rozdělena na teoretickou a praktickou část. Teoretická část obsahuje definice základních pojmů tématu vývoje webových aplikací – systému Word Wide Web a jeho jednotlivých dílčích částí, značkovacího jazyka HTML, stylovacího nástroje CSS, a také charakteristiku vybraných skriptovacích programovacích jazyků pro tvorbu webu. Ta zahrnuje seznámení s daným jazykem, jeho stručnou historii a kontext vývoje doplněný o ukázkou zdrojového kódu jazyka, a též klady a zápory jeho použití v praxi. V poslední řadě obsahuje teoretická část popis principů tvorby webových stránek, doporučení nejvhodnějšího jazyka pro praktickou ukázkou, a také výčet použitého softwaru pro vývoj výsledného webu.

V praktické části práce je popsán samotný vývoj ukázkové webové aplikace s fitness tematikou. Pro přehlednost je vývoj rozdělen do tří samostatných bloků. Front-end popisuje rozložení a vizuál aplikace, back-end nabízí náhled na praktické použití vybraného skriptovacího jazyka a kapitola databáze představuje řešení ukládání uživatelských dat do databázových tabulek.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Bakalářská práce se podrobně zabývá skriptovacími programovacími jazyky využívanými k tvorbě dynamických webových aplikací. Hlavním cílem bakalářské práce je demonstrovat vlastnosti a využití jednoho vybraného programovacího jazyka prostřednictvím ukázkové webové aplikace. Dílčími cíli je komplexně popsat téma tvorby webových stránek, analyzovat jednotlivé programovací jazyky používané pro tvorbu webů a vybrat vhodný programovací jazyk, který bude použit pro vývoj webové aplikace.

### **2.2 Metodika**

Metodika bakalářské práce je založena na shromáždění, studiu a analýze odborných informačních zdrojů. V literární rešerši jsou vytyčeny základní pojmy týkající se tvorby webů a webových aplikací: systém WWW a jeho dílčí složky, značkovací jazyk HTML a jazyk CSS. Dále se práce zabývá charakteristikou jednotlivých programovacích jazyků. V ní je zahrnuto představení jazyka a demonstrace jeho podoby a vlastností prostřednictvím ukázky zdrojového kódu. Pro zasazení do kontextu vývoje trendů v softwarové sféře jsou zmíněny indexy TIOBE a PYPL, na kterých je ilustrována popularita mezi veřejností. Charakteristika se také skládá z historie jazyka a výčtu pozitiv a negativ možného použití.

Na základě syntézy získaných teoretických poznatků jsou shrnuty klady a zápory jednotlivých programovacích jazyků a vybrán jeden jazyk, který je v analytické části práce použit na tvorbu ukázkového webu, demonstrujícího jeho funkce a možné využití v praxi. Ukázkový web a jeho podrobně popsany a okomentovaný kód bude výstupem této bakalářské práce.

### 3 Teoretická východiska

Teoretická část bakalářské práce popisuje základní pojmy tématu vývoje webových aplikací: jako první je představen nadřazený systém World Wide Web zajišťující zpřístupnění obsahu na internetu. Dále je kladen zvýšený důraz na technologie úpravy struktury a podoby webové stránky: značkovacího jazyka HTML a stylovacího nástroje CSS. Následně je představeno několik vybraných skriptovacích technologií na základě míry použití, oblíbenosti a relevance. Charakteristika zahrnuje seznámení s daným jazykem, stručný popis historie a kontextu vývoje doplněný o ukázkou zdrojového kódu, a též klady a zápory jeho použití pro tvorbu webových aplikací. V poslední kapitole je popsán princip tvorby webových stránek, na základě shromážděných poznatků doporučen nejvhodnější jazyk pro praktickou ukázkou, a také přiblížen použitý software pro vývoj výsledného webu.

#### 3.1 WWW

World Wide Web (WWW, W3, nebo pouze Web) je distribučním informačním systémem, vyvinutým za účelem vytváření, prohlížení a sdílení elektronických dokumentů (webových stránek) přes síť Internet. Umožňuje vzájemné hypertextové propojení, pomocí kterého lze odkazovat mezi libovolnými dokumenty uloženými kdekoliv na světě. Implementace systému vedla k jednodušší a produktivnější práci s internetem a započala tak jeho celosvětový rozmach. [1]

Za autora WWW je považován britský informatik Tim Berners-Lee, který v roce 1989 definoval první hypertextový systém při svém působení ve švýcarské CERN<sup>1</sup>. O rok později vytvořil první webový prohlížeč a nazval jej WordWideWeb. V roce 1994 založil W3 Konsorcium (W3C), které službu WWW spravuje a standardizuje dodnes. [2]

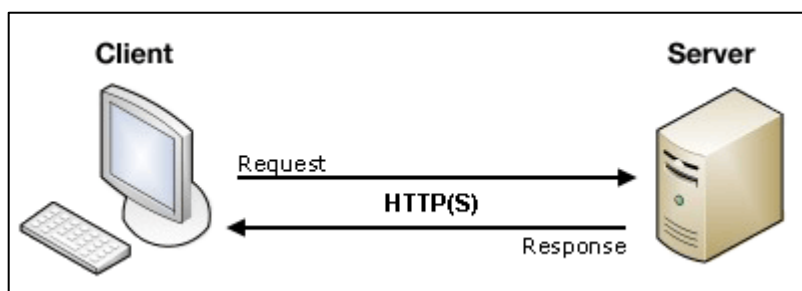
Podle Berners-Leeho byl Web vyvíjen proto, aby sloužil jako zdroj lidských vědomostí, který by umožňoval všem zúčastněným kdekoliv na světě sdílet své nápady a vynálezy v reálném čase. Idea Webu jako celosvětového projektu pro propojení dostupných relevantních informací se zrodila v CERN, kde byl otestován obdobně fungující personální hypertextový systém navržený tak, aby hlídal spojitosti mezi jednotlivými dokumenty uloženými na intranetu a podle relevance je mezi sebou propojoval odkazy.

---

<sup>1</sup> Conseil Européen pour la Recherche Nucléaire – Evropská organizace pro jaderný výzkum

Jednoduchá škálovatelnost celého projektu umožnila rychlé rozšíření za hranice organizace CERN po celé síti internetu. [3]

Fungování WWW je založeno na architektuře **klient – server** (Obrázek 1). Klientem je myšlen program (webový prohlížeč) běžící na lokálním počítači, který podle uživatelských vstupů zašle dotaz na příslušný server a zobrazí výslednou odpověď (většinou v podobě hypertextu nebo multimédia) na monitoru. Server je počítač připojený v síti, často fyzicky umístěný v jiné části světa než PC, jenž požadavek zaslal. Na něm jsou uložena data, která server po vyhodnocení dotazů a požadavků zašle příslušným klientům. Tento způsob fungování systému značně snižuje nároky na technické znalosti uživatele, neboť veškerou komunikaci se serverem zajišťuje webový klient a uživateli jsou předkládány pouze vygenerované výsledky v textové (či jiné) podobě. Před použitím WWW bylo nutné znát přístupové metody a příkazové jazyky pro jednotlivé typy serverů, což značně omezovalo dostupnost síťové komunikace pro běžného uživatele PC. [1]



Obrázek 1 – Princip architektury Klient – Server [4]

WWW se skládá z následujících složek:

- 1) univerzální adresní systém **URL** (Uniform Resource Locator)
- 2) internetový komunikační protokol **HTTP** (HyperText Transfer Protocol)
- 3) značkovací jazyk **HTML** (HyperText Markup Language) [2]

### 3.1.1 Klient

Webový klient je software nebo aplikace, která prostřednictvím internetu komunikuje s webovými servery. U webových dokumentů v jazyce HTML zajišťuje komunikaci program nazvaný webový prohlížeč (web browser). Ten zprostředkovává veškerou interakci mezi uživatelem a serverem a stará se o zasílání požadavků na server a zobrazení přijatých odpovědí uživateli. [1]

Ne vždy dokáže klient samostatně odpověď zpracovat a zobrazit, v takovém případě spustí pomocný software nebo zásuvný modul (plug-in) schopný daný formát zobrazit přímo (např. videosoubor). Dotazy i odpovědi vyměňované v komunikaci mezi klientem a serverem jsou formátovány podle pravidel definovaných standardem protokolu HTTP, kterému se práce věnuje v kapitole 3.1.4. [1]

Mezi technologie na straně klienta patří HTML, CSS<sup>2</sup> a JavaScript. Ty jsou interpretovány po stažení stránky ze serveru přímo webovým prohlížečem, ve kterém jsou uloženy. Značkovací jazyk HTML definuje, čím zobrazovaný text je (např. nadpis sekce), CSS upravují jeho atributy a výslednou podobu a JavaScript obstarává spouštění skriptů na straně klienta. [5]

### 3.1.2 Server

Pojem Webový server označuje buď počítač připojený v síti Internet sloužící jako zdroj dat, nebo software běžící na tomto počítači, který na základě dotazů poskytuje klientům webový obsah, většinou webovou stránku nebo soubor. Server využívá komunikační protokol HTTP pro přenos dat mezi klienty a serverem. Když klient pošle požadavek na server, například o zobrazení webové stránky nebo odeslání formuláře, server odpoví s požadovanými daty nebo chybovým hlášením. [6]

Odpovědi serveru se dle povahy klientských požadavků liší, proto je jejich součástí také příslušný stavový kód vyjadřující, s jakou úspěšností byl požadavek vyřízen a odpověď zpracována.

Odpovědi se dělí do následujících kategorií:

- **1xx** (Informační odpověď) – požadavek byl doručen, proces pokračuje: 100 Continue (Pokračování procesu), 101 Switching Protocols (Změna protokolu).
- **2xx** (Úspěch) – požadavek byl úspěšně přijat a zpracován: 200 OK (Úspěšný požadavek), 201 Created (Zdroj vytvořen), 202 Accepted (Požadavek přijat).

---

<sup>2</sup> Cascading Style Sheets – kaskádové styly

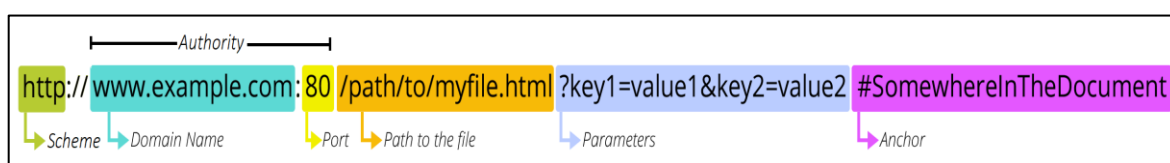


- **3xx** (Přesměrování) – je nutná další akce klienta pro splnění požadavku: 300 Multiple Choices (Více možností vyřízení), 301 Moved Permanently (Zdroj přesunut).
- **4xx** (Chyba na straně klienta) – chybně zapsaný nebo nespelnitelný požadavek: 400 Bad Request (Chybný požadavek), 403 Forbidden (Přístup odepřen), 404 Not Found (Zdroj nenalezen).
- **5xx** (Chyba na straně serveru) – validní požadavek nezpracovaný serverem: 500 Internal Server Error (Interní chyba serveru), 503 Service Unavailable (Služba nedostupná). [7]

### 3.1.3 URL

URL (Uniform Resource Locator) je řetězec znaků, který slouží jako specifikace cesty ke zdroji informací (dokumentu, souboru, webové stránce) uložených v síti Internet. Skládá se ze tří hlavních částí: protokolu, domény a přístupové cesty k souboru (Obrázek 2). [2]

Protokolová část specifikuje typ použitého protokolu pro zaslání požadavku (například HTTP, HTTPS<sup>3</sup>), doména (doménové jméno nebo IP adresa) slouží jako unikátní identifikátor webové stránky, ke které je přistupováno, a přístupová cesta upřesňuje informace, kde se hledaný soubor na serveru nachází. URL může obsahovat i další informace jako například port<sup>4</sup>, parametry (vstupní hodnoty pro skript běžící na stránce) nebo kotvu (přímý odkaz na určitou část webové stránky). [8]



Obrázek 2 – Příklad adresy URL [8]

### 3.1.4 HTTP

HTTP (HyperText Transfer Protocol) je komunikační protokol aplikační vrstvy používaný pro přenos dat v rámci Webu. Jedná se o základní stavební kámen WWW, který umožňuje prohlížečům a aplikacím zobrazovat webové stránky a komunikovat s webovými servery. Pro transport dat je využíván protokol TCP<sup>5</sup> na vyhrazeném portu 80. [8]

<sup>3</sup> HyperText Transfer Protocol Secure – zabezpečená verze protokolu HTTP [8]

<sup>4</sup> Zápis portu v URL je využit převážně na intranetech, pro protokol HTTP/HTTPS se standardně neuvádí [8].

<sup>5</sup> Transmission Control Protocol

HTTP pracuje na principu **dotaz – odpověď** (Obrázek 1). Klient pošle dotaz na server, například žádost o zobrazení webové stránky nebo odeslání formuláře, a server odpoví s požadovanými daty nebo chybovým hlášením. Metainformace o prohlížeči a serveru včetně dotazovací metody u požadavků a stavového kódu u odpovědí jsou obsaženy v HTTP hlavičce, tělo HTTP obsahuje data odesílaná klientem na server a vice versa, případně zůstává prázdné, pokud k výměně dat nedošlo. [1]

Jedná se o bezstavový protokol: neumožňuje uchovávat informace o komunikaci a každý dotaz je vyhodnocován samostatně bez jakékoliv souvislosti s ostatními. Proto je rozšířen o tzv. HTTP Cookies, malé dávkové soubory ukládané na straně klienta webovým serverem, které obsahují potřebné informace o uživateli. [10]

Mezi základní HTTP metody dotazování patří:

- **GET** – požadavek klienta na zobrazení dat (např. webových stránek) ze serveru
- **POST** – požadavek klienta na zaslání dat na server, která jím budou dále zpracována (např. vyplněný formulář)

Další HTTP metody: PUT, HEAD, DELETE, PATCH, OPTIONS, CONNECT

Veškerá probíhající komunikace přes protokol HTTP není šifrována, což nese mnohá bezpečnostní rizika, jako krádež citlivých údajů či odcizení účtu. Proto se většinou používá jeho zabezpečená verze: HTTPS. Zde jsou webové stránky opatřeny digitálními certifikáty ověřujícími jejich identitu a při komunikaci se serverem jsou mimo protokolu HTTP využívány i protokoly SSL<sup>6</sup> nebo TLS<sup>7</sup>, které se starají o šifrování přenášených dat. HTTPS využívá serverový port 443 TCP. [11]

## 3.2 HTML

HTML (HyperText Markup Language) je značkovací jazyk používaný pro tvorbu webových stránek. Definuje význam a strukturu webového obsahu. Slovo Hypertext v názvu jazyka označuje odkazy spojující webové stránky mezi sebou. Spolu s jazyky **CSS** a **JavaScript** patří HTML k technologiím stylizování webových stránek na straně klienta (front-end). [1], [5]

---

<sup>6</sup> Secure Sockets Layer

<sup>7</sup> Transport Layer Security

Jazyk HTML používá pro určení významu textu značky (elementy), které označují jednotlivé prvky na stránce, jako jsou nadpisy, odstavce, obrázky nebo odkazy. Tyto značky se zapisují přímo do textu webové stránky, prohlížeč provede jejich interpretaci a výsledná stránka je zobrazena uživateli. HTML dokumenty mají následující strukturu (Obrázek 3):

- Deklarace `<!DOCTYPE html>` – sděluje prohlížeči, že přistupuje k HTML5 dokumentu
- Kořenový element `<html>` – ohraničení celého dokumentu
- Hlavička `<head>` – obsahuje metadata dokumentu, například název, propojení s CSS nebo nastavení zobrazení pro odlišné velikosti displejů
- tělo `<body>` – sem spadá veškerý obsah dokumentu

Dále mohou obsahovat celou řadu určujících elementů, například `<title>`, `<article>` a další. [12]

```
<!DOCTYPE html>
<html>

  <head>
    <title> Title here </title>
  </head>

  <body>
    Web page content goes here.
  </body>

</html>
```

Obrázek 3 – Ukázka struktury HTML dokumentu [23]

Jednotlivé elementy se skládají z otevírací a zavírací značky. Například element `<p>` označuje odstavec textu a má otevírací značku `<p>` a uzavírací značku `</p>`. Mezi těmito značkami se nachází obsah, například text odstavce. Dále mohou elementy obsahovat atributy, které poskytují další informace o daném elementu. Například element `<img>` (image) má atribut `src` (source), který udává cestu k zobrazovanému obrázku. [12]

Počátky jazyka HTML se datují do roku 1990, kdy po spolupráci Tima Berners-Leeho a Roberta Cailliaua vznikl první prototyp jazyka HTML, vytvořený pro potřeby jednoduché tvorby webových stránek v nově založeném systému WWW. Do té doby používaný jazyk

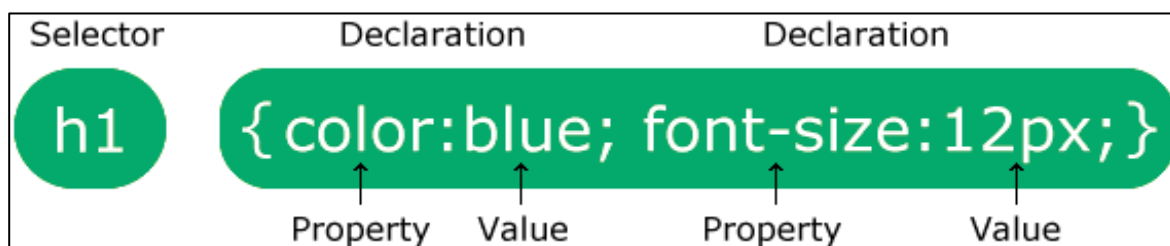
SGML<sup>8</sup> byl pro implementaci do systému příliš složitý, byl proto zjednodušen, což dalo vzniknout HTML. [1]

Současná verze k roku 2023, HTML 5, se již vzhledem k původní verzi velmi vyvinula a změnila. HTML 5 (vydán r. 2014) přinesl nové elementy pro lepší a přehlednější členění a atributy pro vylepšení funkčnosti webových stránek, například podporu pro multimédia v prohlížeči, lepší formulářové prvky nebo podporu offline aplikací. [12]

### 3.3 CSS

CSS (Cascading Style Sheets) je jazyk, který se používá pro stylování webových stránek. CSS umožňuje definovat vzhled prvků na stránce, jako jsou barva, velikost, font, pozice nebo animace. To umožňuje oddělit obsah webové stránky od jejího vzhledu a jednoduše měnit zobrazení celého webu na jednom místě, bez nutnosti zásahu do webové struktury. [13]

Definice kaskádových stylů se skládá z listu pravidel, z nichž každé obsahuje selektor a blok deklarácí. Selektor označuje HTML prvek na stránce, ke kterému se pravidlo vztahuje, například element `<p>`. Blok deklarácí obsahuje informace o stylu, tedy jaké vlastnosti stránky jsou upravovány a jaké hodnoty jsou jim přiřazeny (Obrázek 4). [13]



Obrázek 4 – Syntax jazyka CSS [24]

CSS nabízí tři základní způsoby, jakými lze stylování aplikovat prvkům na stránce. Lze je stylovat přímo uvnitř HTML elementu pomocí atributu `style`, v hlavičce dokumentu pomocí elementu `<style>`, případně vložením do externího souboru, který je následně propojen s HTML dokumentem pomocí elementu `<link>` (odkaz). [14]

Za vývojem kaskádových stylů stojí konsorcium W3C, které se stará o standardizaci nejen CSS, ale i jazyka HTML. Kaskádové styly byly navrženy v roce 1996, aby umožnily oddělení formátování od obsahu dokumentu. Původně bylo formátování dokumentů prováděno pouze pomocí HTML značek, což způsobovalo problémy s přenositelností

<sup>8</sup> Standard Generalized Markup Language – standardní jazyk pro popis značkovacích programů [1]

a údržbou kódu. CSS umožnily vytvořit jednotný styl pro celou webovou stránku nebo webovou aplikaci a výrazně snížily nutnost opakovaně kódovat totožné formátování na různých místech dokumentu. [1], [2]

CSS také umožnily vytvořit odlišné styly pro různé typy zařízení nebo podle velikosti okna prohlížeče, což umožnilo vytváření responzivních webových stránek, které se automaticky přizpůsobují velikosti obrazovky. Kromě toho CSS umožňují vytvářet částečně interaktivní prvky, jako jsou hover<sup>9</sup> efekty a další. Nic z uvedeného nebylo možné realizovat pouze v HTML, proto rozmach CSS vedl k strmému nárůstu kvality webů a uživatelského komfortu při prohlížení webových stránek. [13]

Při tvorbě webových stránek je možné využít širokou škálu CSS frameworků (rámců). Ty značně zjednodušují práci webovým developerům, protože nabízí přednastavené šablony, upravují rozložení stránky, poskytují sety komponent pro úpravu rozhraní a umožňují vytvářet přenositelné a responzivní webové stránky bez nutnosti začínat ve vývoji pokaždé „od nuly“. Nejznámějším a nejpoužívanějším CSS frameworkem je **Bootstrap**. [14]

## 3.4 Skriptovací programovací jazyky

### 3.4.1 Charakteristika

Skriptovací jazyky jsou typ jazyků programování, které se používají k automatizaci úloh a vykonávání specifických úkolů v rámci jiných aplikací nebo systémů. Na rozdíl od standardních jazyků jako C nebo Java, které se kompilují do binárních souborů, skriptovací jazyky se interpretují při běhu. To znamená, že kód je spuštěn v reálném čase a může být snadno změněn nebo opraven. [2]

Program zapsaný ve skriptovacím jazyce se označuje termínem skript. Princip fungování skriptovacích jazyků je založen na interpretaci skriptu napsaném v daném jazyce. Jedná se o příkaz vykonávající určitou strojovou instrukci, který se nahraje do paměti počítače a je interpretován po jednotlivých příkazech. Interpret se postará o přímé vykonání příkazu ve zdrojovém kódu, bez nutnosti převodu do strojového kódu procesoru, jako je tomu u kompilovaných jazyků. Tímto způsobem mohou skriptovací jazyky vykonávat různé úkoly a automatizovat práci s počítačem. [15][15]

---

<sup>9</sup> Efekt přechodu – změna vzhledu prvku po najetí myši, například jeho zvýraznění a přebarvení [13]

Mezi nejznámější skriptovací jazyky patří Python, JavaScript, Ruby a PHP. Tyto jazyky jsou nedílnou součástí procesu tvorby dynamických webových aplikací na straně serveru (v případě jazyka JavaScript i na straně klienta) a zajišťují interaktivní a přizpůsobenou komunikaci podle požadavků uživatele. [15]

Skriptovací jazyky ve valné většině obsahují knihovny a moduly, které poskytují předdefinované funkce a metody pro různé úkoly, jako je například práce s databázemi nebo sítě. Tyto knihovny a moduly umožňují vývojářům rychle a snadno implementovat složité funkce do svých skriptů, aniž by museli psát vlastní kód. [2]

### 3.4.2 Kritéria výběru

Skriptovací programovací jazyky zkoumané v teoretické části práce byly vybrány pomocí dvou kritérií, které zajišťují, že se jedná o relevantní a hojně využívané jazyky pro tvorbu webových rozhraní, a že jejich výběr reflektuje trendy na poli informačních technologií.

1. **W3Techs** průzkum použití technologií zkoumá procentuální zastoupení jednotlivých programovacích a skriptovacích jazyků na základě analýzy volně dostupných informací o webových stránkách. Podle publikované statistiky o využití programovacích jazyků pro server-side skriptování jsou nejvíce zastoupeny následující technologie: **PHP** (77,8 %), **ASP.NET** (7,3 %) a **Ruby** (5,2 %) [16]. Ve skriptování na straně klienta dominuje pouze jediný jazyk, a to **JavaScript** (98,2 %) [28], využívaný také na straně serveru, kde je zastoupen 2 % [16].
2. Index **TIOBE** a Index **PYPL** indikují popularitu jednotlivých programovacích jazyků na základě četnosti vyhledávání na internetu. TIOBE měří počet výsledků vyhledávání týkající se daného jazyka napříč nejpoužívanějšími internetovými vyhledávači. PYPL analyzuje, jak často jsou na Google search vyhledávány tutoriály na jednotlivé jazyky, slouží tedy jako ukazatel relevance. Tyto dva indexy se navzájem doplňují a dávají poměrně přesný obraz o popularitě programovacích jazyků. Oběma indexům vévodí na prvním místě jazyk **Python**, zastoupen z 28,2 % v PYPL a z 16,66 % v TIOBE. [17], [18]

Na základě výše zmíněných kritérií se práce dále věnuje pěti programovacím jazykům: PHP, ASP.NET, Ruby, JavaScript a Python.

### 3.4.3 ASP.NET (Core)

První představená technologie se od ostatních nápadně liší, nejedná se totiž o interpretovaný skriptovací jazyk ani jazyk programovací, nýbrž o multiplatformní aplikační framework zaměřený na vývoj dynamických webových aplikací. Pro jeho poměrně silné zastoupení v server-side technologiích (druhé místo s 7,3 % podle W3Techs) jej nebylo možné ve srovnání opomenout. [17]

ASP.NET (Active Server Pages Network Enabled Technologies) je open-source framework vyvíjený společností Microsoft pro potřebu tvorby webových aplikací pro systém Windows. Je založen na technologii CLR<sup>10</sup>, která umožňuje developerům psát ASP.NET kód v kterémkoliv z podporovaných programovacích jazyků, jako je například C# nebo Visual Basic.NET. [19]

ASP.NET nabízí širokou škálu funkcí pro vytváření webových aplikací, včetně podpory pro formuláře, validaci, autentizaci a autorizaci, správu stavu a mnoho dalšího. Framework také podporuje různé architektury, jako je MVC<sup>11</sup>, Web Forms a Web Pages, což umožňuje vývojářům zvolit nejvhodnější architekturu podle zaměření projektu. [19]

Původní verzi ASP.NET nahradil v roce 2016 její nástupce **ASP.NET Core** spojující v sobě funkcionality dvou přechozích aplikačních frameworků: ASP.NET Web API a .NET MVC. Core poprvé nabízí podporu nezávisle na operačním systémem, podporuje Windows, Linux i macOS. [20]

```
namespace HelloWorld
{
    class Hello {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World!");
        }
    }
}
```

Zdrojový kód 1 – Ukázka programu „Hello World“ v jazyce C#

---

<sup>10</sup> Common Language Runtime

<sup>11</sup> Model View Controller – rozdělení aplikace do tří nezávislých komponent: datového modelu, uživatelského rozhraní a řídicí logiky [19]

## Historie

ASP (Active Server Pages) byla první verze platformy pro tvorbu webových aplikací od společnosti Microsoft. Byla vydána v roce 1996 a umožňovala vytvářet dynamické webové stránky pomocí skriptovacích jazyků VBScript a JScript. [2]

ASP.NET vychází v roce 2002 jako nástupce technologie ASP a poskytoval vylepšenou podporu pro vývoj webových aplikací pro Microsoft Windows. Na rozdíl od svého předchůdce byl kód aplikace kompilovaný, což urychlilo běh a usnadnilo zachycování chyb při vývoji. Další výhodou poskytovala výše zmíněná technologie CLR, která značně rozšířila repertoár podporovaných programovacích jazyků. [19]

ASP.NET Core je nová generace platformy ASP.NET vydána v roce 2016. Jedná se o open-source a cross-platform framework, který poprvé nabízí využití i na operačních systémech mimo Windows. ASP.NET Core poskytuje vylepšenou flexibilitu a lepší optimalizaci aplikací pro internetové použití. Kompilace aplikace je prováděna za běhu, odpadá tedy nutnost rekompilace po každé dílčí změně, dále jsou zde spojeny funkce dříve separátních frameworků ASP.NET Web API a ASP.NET MVC. [20]

## Klady a zápory

**Klady** použití ASP.NET Core pro tvorbu webových aplikací:

- Cross-platform: ASP.NET Core je open-source framework, který lze použít na různých operačních systémech, jako jsou Windows, MacOS a Linux.
- Vysoký výkon: Framework poskytuje vysoký výkon a efektivní správu paměti, což umožňuje vytvořit robustní a rychlé webové aplikace.
- Bezpečnost: ASP.NET Core obsahuje built-in bezpečnostní funkce, jako je například ověřování přihlášení a autorizace.
- Integrace s Visual Studio: Core je integrován s populárním vývojovým prostředím Visual Studio od Microsoftu umožňujícím snadnější vývoj a debugging.
- Škálovatelnost: ASP.NET Core je navržen tak, aby bylo snadné rozšířit aplikaci na další servery, čehož je možné využít pro uspokojení potřeb rostoucího provozu.



### **Zápory:**

- Nižší kompatibilita: ASP.NET Core má menší kompatibilitu s knihovnamí a nástroji třetích stran než jiné webové frameworky.
- Náročnost konfigurace: Některé funkce ASP.NET Core vyžadují složitější konfiguraci, což může být pro začínající uživatele obtížné.
- Omezená přenositelnost: Projekty postavené na ASP nebo ASP.NET je velmi složité převést do ASP.NET Core. [20], [21]

**TIOBE** umístění: Dominantní jazyk rozhraní ASP.NET, C#, obsadil v Indexu TIOBE 5. místo se 4,9 % [17].

**PYPL** umístění: V Indexu PYPL se C# umístil na 4. místě se 7 % [18].

#### **3.4.4 JavaScript**

JavaScript je skriptovací programovací jazyk, jehož použití je klíčovým prvkem tvorby dynamických webových stránek. Zatímco HTML a CSS dávají stránce strukturu a styl, JavaScript zajišťuje tvorbu interaktivních prvků na webových stránkách, jako jsou tlačítka, formuláře a animace. [5]

Jedná se o interpretovaný nebo JIT<sup>12</sup> kompilovaný jazyk, který je primárně využíván pro výše zmíněné client-side skriptování, nicméně nabízí i další možnosti využití: tvorbu desktopových a mobilních aplikací skrze některý z dostupných aplikačních frameworků (Cordova, Electron, React Native), skriptování PDF dokumentů v rámci Adobe Acrobat, nebo server-side skriptování přes velice populární softwarový systém **Node.js**. [22]

Právě zmíněný Node.js slouží jako konkurence tradičně používaným skriptovacím jazykům pro tvorbu webových aplikací, jako je PHP nebo Ruby. Nespornou výhodou je použití JavaScript nejen pro client-side ale i pro server-side skriptování, což snižuje nároky na vývojáře a zjednodušuje proces vývoje webu. Node.js se od konkurence odlišuje architekturou podporující asynchronní I/O (Input/Output), která umožňuje běh více paralelních příkazů najednou. [22]

---

<sup>12</sup> Just In Time – kompilovaný za běhu programu [22]

```
<!DOCTYPE html>
<html>
  <body>
    <script>
      alert('Hello, world!');
    </script>
  </body>
</html>
```

Zdrojový kód 2 – Ukázka jazyka Javascript vnořeného do HTML

## Historie

JavaScript byl vyvinut roku 1995 Brendanem Eichem, za dobu jeho působení ve společnosti Netscape, jako univerzální skriptovací jazyk pro webové prohlížeče. Jeho původním účelem byla tvorba dynamických webových stránek, které do té doby byly pouze statické. Motivací pro vytvoření JavaScript byla nutnost odstranit tuto bariéru ve vývoji stránek v nově vydaném prohlížeči Netscape Navigator (1994). Jméno JavaScript bylo zvoleno jako marketingová strategie, díky velké popularitě jazyka Java v době vydání, jinak spolu oba jazyky nijak více nesouvisí. [2]

Roku 1997 vychází, z důvodu značné odlišnosti jednotlivých iterací jazyka JavaScript (například JScript od Microsoftu), jednotná standardizace jazyka nazvaná ECMAScript. Ta sjednotila zobrazení a chování skriptů v prohlížečích. V následujících letech se JavaScript rozšířil i do jiných odvětví, například serverových technologií (Node.js), aplikací pro mobilní telefony a desktopových aplikací. Standard ECMAScript 5 z roku 2009 v sobě obsahoval další podstatná vylepšení jazyka, představil například možnost JIT kompilace vedoucí k značnému zrychlení běhu skriptů. Jeho stabilně vysoká popularita a schopnost spolupracovat s technologiemi umožňujícími využití mimo webový prohlížeč, jako jsou například Node.js, React nebo Vue.js, učinila z JavaScript jeden z nejžádanějších jazyků v oboru vývoje software. [5]

## Klady a zápory

Mezi hlavní **klady** použití JavaScript patří:

- **Jednotný jazyk:** JavaScript je jazyk, který je používán jak na straně klienta, tak na straně serveru. To umožňuje vývojářům pracovat s jedním jazykem po celou dobu vývoje aplikace.
- **Dynamika:** Použití jazyka umožňuje vytvářet interaktivní a dynamické webové stránky, které reagují na akce uživatele.

- Rozšířenost: JavaScript patří k nejpobulárnějším jazykům a má rozsáhlou komunitu, která poskytuje podporu a pomoc při vývoji.
- Vstupní validace: JavaScript umožňuje validovat formuláře na straně klienta, což může snížit zatížení na serveru a rychleji zpracovávat uživatelské požadavky.

#### **Zápory:**

- Nesoulad verzí: Prohlížeče mohou obsahovat různé verze JavaScript, což může vést k problémům s kompatibilitou a potřebě přizpůsobení kódu podle prohlížeče.
- Bezpečnost: Použití JavaScript má některá bezpečnostní rizika, jako jsou cross-site scripting (XSS) a cross-site request forgery (CSRF).
- Výkon: JavaScript má omezený výkon v porovnání s jinými jazyky při zpracování velkého objemu dat nebo využití náročnějších algoritmů.
- Volná syntaxe: JavaScript má volnou syntaxi, což může vést k problémům se čtením a údržbou zdrojového kódu. [5], [22], [25], [28]

**TIOBE** umístění: TIOBE zařazuje jazyk na 7. místo s 2,9 % [17].

**PYPL** umístění: V Indexu PYPL se jazyk nachází na 3. místě s 9,6 % [18].

#### **3.4.5 PHP**

PHP (Hypertext PreProcessor) je open-source multiplatformní skriptovací programovací jazyk, který se nejčastěji používá pro tvorbu dynamických webových stránek a webových aplikací ve formátu HTML. Syntaxe jazyka vychází z několika dalších programovacích jazyků, primárně Perlu a C. Použití pro tvorbu webových skriptů dominuje, jazyk má ale k dispozici svou kompilovanou formu, jež nabízí možnost tvorby konzolových a desktopových aplikací. [2]

Skripty PHP jsou integrovány přímo v jazyce HTML, což značně zjednodušuje jejich zápis. Jedná se o interpretovaný jazyk, PHP kód je zpracován interpretem na straně serveru a výsledná stránka je zaslána a vygenerována v klientském rozhraní v podobě dokumentu HTML. [2]

PHP je nejpoužívanějším programovacím jazykem pro server-side skriptování, obsahuje jej přibližně 78 % všech známých webových serverů [16]. Míra popularity je dána primárně jednoduchostí použití, podporou mnoha rozšiřujících knihoven a provázaností s dalšími technologiemi, se kterými tvoří plně funkční open-source serverovou infrastrukturu. Příkladem použití může být systém LAMP, kde jednotlivá písmena zkratky vyjadřují využívanou technologii:

- **Linux** – použitý operační systém na webovém serveru
- **Apache** – populární webový HTTP server od stejnojmenné společnosti
- **MySQL** – relační databázový systém pro správu uložených dat v jazyce SQL<sup>13</sup>
- **PHP** – skriptovací jazyk tvořící výsledné dynamické webové stránky

Některé technologie v systému LAMP jsou zaměnitelné, je možné využít např. Python či Perl místo PHP, nebo MariaDB místo MySQL. [26]

```
<!DOCTYPE html>
<html>
  <body>
    <?php echo 'Hello, World!'; ?>
  </body>
</html>
```

Zdrojový kód 3 – Ukázka skriptu v jazyce PHP vnořeného do dokumentu HTML

## Historie

Počátky jazyka PHP se datují do roku 1995, kdy na jeho vývoji začal pracovat dánský programátor Rasmus Lerdorf. Ten jej zprvu zamýšlel pouze jako prostředek pro sledování návštěvnosti svého osobního webu a nazval jej Personal Home Page. Původní PHP byla pouze sada skriptů napsaných v jazyku Perl s omezenou použitelností. Proto ji Lerdorf spojil s dalším programem vlastní výroby, Form Interpreterem (FI). Výsledné spojení nazvané PHP/FI obsahovalo mimo jiné širokou implementaci pro jazyk C, umožňovalo komunikaci s databázemi, zpracování formulářů a možnost vkládat HTML kód, což umožnilo poprvé vytvářet jednoduché dynamické webové stránky. [2]

V roce 1997 se do vývoje PHP zapojili izraelští vývojáři Andi Gutmans a Zeev Suraski, kteří představili novou verzi jazyka s názvem PHP 3. Tato verze byla kompletně

---

<sup>13</sup> Structured Query Language – dotazovací jazyk pro práci s relačními databázemi [26]



- Omezená podpora pro větší projekty: PHP není vhodný pro větší projekty, které vyžadují silnou architekturu a plánování.
- Změna verzí: Pro přechod na vyšší verzi jazyka je často nutné celý webový projekt od základů přepracovat.
- Nekonzistentní syntaxe: Dosud neexistuje formální specifikace syntaxe jazyka a jeho chování (jedinou úplnou definicí je jeho interpret), což může vést k nepřehlednému a nefunkčnímu kódu. [2], [15], [16], [26], [27]

**TIOBE** umístění: Podle TIOBE je jazyk PHP na 10. místě s 1,6 % [17].

**PYPL** umístění: Index PYPL ukazuje odlišnou míru popularity: 6. místo s 5,3 % [18].

### 3.4.6 Python

Python je vysokoúrovňový programovací jazyk, který má velmi široké pole využití: používá se pro tvorbu webových aplikací, aplikací pro desktop, automatizaci procesů, analytiku dat a strojové učení. Python se vyznačuje snadnou syntaxí a čistým kódem, což z něj činí jazyk vhodný jak pro začínající programátory, tak pro tvorbu rozsáhlých projektů nadnárodních korporací. Jméno bylo odvozeno podle britské komediální skupiny Monty Python, již byl autor Guido van Rossum velkým fanouškem. [29]

Jedná se o interpretovaný a objektově orientovaný jazyk. Spojuje v sobě jednoduchost a přehlednost skriptovacích jazyků, zároveň obsahuje i pokročilé nástroje standardně dostupné pouze u kompilovaných jazyků (C, Java), jde tedy o jazyk „hybridní“. Kromě objektového paradigmatu podporuje i procedurální a funkcionální programování. Disponuje rozsáhlou knihovnou standardních modulů, která poskytuje širokou škálu funkcí pro práci s daty, sítí, I/O zařízeními a mnoha dalšími. Jde o open-source jazyk s velkou mírou přenositelnosti: funguje na mnoha Linux variantách, macOS i MS Windows. [30]

Python má širokou škálu praktického využití, což z něj činí jeden z nejoblíbenějších a nejpoužívanějších programovacích jazyků současnosti. Použit je mimo jiné v následujících odvětvích:

- Standardní knihovny Python podporují jednoduchou práci se systémem, a proto je ideálním jazykem pro psaní lehce přenositelných skriptů pro správu operačního systému.

- V jazyku Python lze snadno tvořit kvalitní a přenositelná GUI<sup>15</sup> aplikací, například pomocí knihovny TK přes rozhraní Tkinter.
- Pomocí dostupné knihovny NumPy (Numeric Python) lze použít Python jako pokročilý a dostupný nástroj pro numerické programování.
- Python obsahuje všechny funkce potřebné pro **serverové skriptování**, podporuje práci s CGI<sup>16</sup>, parsování<sup>17</sup> HTML a XML<sup>18</sup>, přenos souborů prostřednictvím FTP<sup>19</sup> a mnoho dalších nástrojů. [29], [30]

```
print('Hello, world!')
```

Zdrojový kód 4 – Ukázka jazyka Python – program „Hello World“

## Historie

Vývoj jazyka Python započal roku 1989 jako osobní projekt nizozemského programátora Guida van Rossuma, který se snažil vyvinout nástupce oblíbeného výukového programovacího jazyka ABC. V nově vytvořeném jazyku Python spojil jednoduchou syntaxi ABC doplněnou o nové funkce, například zpracování výjimek. Byl navržen jako snadno osvojitelný jazyk pro čitelné programování, který by zároveň nabízel možnosti využití srovnatelné s „klasickými“ programovacími jazyky jako je C nebo Java. Roku 1991 vyšla první oficiální verze, Python 0.9.0. [29]

V roce 2000 představila verze 2.0. mnoho důležitých novinek, jako je podpora UNICODE nebo implementace funkce garbage collector (automatického uvolňování nepoužívaných úseků paměti). Podpora a aktualizace řady 2.x skončily rokem 2020. [31]

Verze 3.0, uvolněná v roce 2008, představila novou syntaxi, která měla zamezit redundanci a repetitivnosti kódu. Šlo o poměrně drastickou úpravu struktury celého jazyka Python za účelem ošetření bezpečnostních problémů a vad designu předchozích verzí. I proto byly vyvíjeny verze 2.x a 3.x dlouho dobu paralelně, zatímco se pracovalo na převodu staršího softwaru na novou řadu jazyka. V roce 2023 je nejaktuálnější verze jazyka Python **3.11**. [31]

<sup>15</sup> Graphic User Interface – grafické uživatelské rozhraní

<sup>16</sup> Common Gateway Interface – protokol propojení webového serveru s externí aplikací [29]

<sup>17</sup> Syntaktická analýza obsahu [29]

<sup>18</sup> Extensible Markup Language

<sup>19</sup> File Transfer Protocol

## Klady a zápory

Mezi hlavní **klady** využití jazyka Python patří:

- **Použití:** Python má jednoduchou syntaxi a čistý kód, což umožňuje efektivnější vývoj odladěného softwaru.
- **Široké možnosti:** Python má širokou škálu knihoven a frameworků pro různé účely, jako je např. Django pro tvorbu webových aplikací, Flask pro jednoduché aplikace a Pyramid pro složitější projekty.
- **Dynamické typování:** Python podporuje dynamické typování, což znamená, že se typ proměnné může změnit při běhu programu. To dává vývojářům větší flexibilitu při tvorbě aplikací.
- **Komunita a dokumentace:** Python má silnou komunitu a širokou škálu dokumentace a nástrojů pro vývoj, což umožňuje rychlý a efektivní problem-solving a debugging.
- **Objektově orientovaný jazyk:** Python podporuje objektově orientované paradigma, což značně zvyšuje jeho účinnost při tvorbě velkých projektů modelovaných podle reálného světa.

## Zápory:

- **Bezpečnostní rizika:** Python obsahuje nedokumentované funkce a lokální proměnné, které mohou být zneužity pro spuštění příkazu v operačním systému.
- **Náchylnost k chybám:** Vlivem designových prvků jazyka jsou programy v Pythonu více náchylné na výskyt chyb. Tyto chyby je velmi často možné odhalit až za běhu programu, je proto nutné déle a rozsáhleji testovat a ladit programy před jejich uvedením do provozu.
- **Výkon:** Stejně jako další interpretované jazyky má Python spíše nižší výkon a rychlost běhu, proto je nutné aplikace před použitím pečlivě optimalizovat.  
[15], [29], [30], [31]

**TIOBE** umístění: V TIOBE indexu se Python umístil na 1. místě s 16,6 % [17].

**PYPL** umístění: Taktéž v PYPL je Python na 1. místě s 28,2 % [18].



### 3.4.7 Ruby

Pod názvem Ruby se skrývá interpretovaný skriptovací programovací jazyk, který je jako jediný z uvedených ryze objektově orientovaný, tedy všechny jeho součásti jsou objekty. Jazyk byl navržen s důrazem na produktivitu programování pomocí jednoduché a elegantní syntaxe. Ruby je znám pro svou velkou flexibilitu umožňující libovolně redefinovat a odstraňovat jeho dílčí části – podle tvůrce jazyk cílí na maximální možné přizpůsobení potřebám programátora. Svě jméno jazyk dostal podle drahokamu rubín, anglicky Ruby. [32]

Ruby v sobě spojuje prvky několika programovacích jazyků: Perl, Smalltalk a Python. Stejně jako Python a Perl je používán pro server-side skriptování, dosahuje však většího výkonu než Perl a na rozdíl od jazyku Python je ryze objektově orientovaný, stejně jako výše zmíněný Smalltalk. Také obsahuje mnoho užitečných funkcionalit, které mu pomáhají držet krok s konkurenčními jazyky, jde například o:

- Dynamické typování, které se stará o validování skriptů za běhu.
- Automatické čištění nepoužívané paměti pomocí funkce garbage collector
- „Chytání“ a ošetřování výjimek (Exception handling)
- Přetěžování operátorů (ad hoc polymorfismus)
- Podpora UNICODE a další [33]

Stejně jako Python je Ruby ideálním jazykem pro psaní server-side skriptů, je možné jej přímo vnořit do HTML pomocí ERB<sup>20</sup>, lze jej použít pro psaní CGI skriptů a může být propojen s databázovým systémem, například MySQL nebo DB2. [32]

Pro komfortnější tvorbu webových aplikací v Ruby je možné využít oblíbený framework **Ruby on Rails** založený na architektuře MVC. Ta rozděluje aplikaci do tří nezávislých komponent, které mohou být modifikovány nezávisle na sobě: datový model, uživatelské rozhraní a řídicí logiku. Filozofie frameworku je založena na principech CoC<sup>21</sup> a DRY<sup>22</sup>. CoC říká, že je nutné specifikovat pouze nekonvenční aspekty aplikace, o vše ostatní se stará přímo konvence jazyka, DRY cílí na eliminaci opakujícího se kódu. [34]

---

<sup>20</sup> Embedded Ruby – vnořené Ruby skripty [32]

<sup>21</sup> Convention over Configuration

<sup>22</sup> Don't Repeat Yourself

```
puts 'Hello, world!'
```

Zdrojový kód 5 – Ukázka jednoduché syntaxe jazyka Ruby – program „Hello World“

## Historie

Ruby byl vyvinut mezi lety 1993 a 1995 japonským programátorem Yukihirom „Matz“ Matsumotem, který jej navrhl kvůli nespokojenosti s ostatními dostupnými skriptovacími jazyky té doby. Jeho cílem bylo vytvořit ryze objektově orientovaný jazyk s nejen jednoduchou syntaxí ale i širokou škálou využití. [32]

První veřejná verze vyšla roku 1995 a rychle si získala značnou popularitu v Japonsku. Zaměření na user-friendly syntax, podpora dynamického typování a implementace funkce garbage collector jazyk rychle proslavila mezi japonskými vývojáři. Častá absence relevantních materiálů v angličtině zatím však bránila aktivnímu rozvoji komunity i mimo Japonsko, s postupem času však došlo k překladu všech klíčových dokumentů a návodů k použití jazyka. [33]

V roce 2003 vyšla první standardizovaná verze, Ruby 1.8, která umožnila snadnou přenositelnost a kompatibilitu kódu napříč platformami. Rozmach používání jazyka dále umocnilo vydání frameworku pro tvorbu webových aplikací, Ruby on Rails, o rok později. Ruby 3.0 z roku 2020 přidalo důležitou funkci JIT kompilace, což vedlo k značnému nárůstu výkonu jazyka. V roce 2023 je nejnovější verzí Ruby **3.2**. [33]

## Klady a zápory

**Klady** použití jazyka Ruby:

- Jednoduchost: Ruby je známý jednoduchou a elegantní syntaxí, což vede ke kratší době vývoje oproti konkurenčním jazykům.
- Ruby on Rails: Framework Ruby on Rails je velmi populární nástroj pro vytváření webových aplikací s minimalizovaným kódem.
- OOP: Ruby je plně objektově orientovaný jazyk, vše v Ruby je objektem.
- Podpora regulárních výrazů: Ruby je díky podpoře regulárních výrazů ideální skriptovacím jazykem pro práci s textovými soubory a jejich správou.
- Cross-platform: Kód Ruby je volně přenositelný mezi vícero platformami (Windows, Linux, Solaris apod.).

### **Zápory:**

- **Náročnost na paměť:** Ruby je jazyk více náročný na spotřebu paměti, což může vést k problémům s výkonem při spouštění velkých aplikací.
- **Nízká rychlost:** Ačkoliv má Ruby jeden z nejrychlejších časů potřebných pro vývoj, zaostává z hlediska rychlosti běhu aplikace i za ostatními skriptovacími jazyky, například Pythonem.
- **Chybějící dokumentace:** Vzhledem k původu jazyka není často existující dokumentace v anglickém nebo českém jazyce dostatečně obsáhla a kvalitní a některé méně využívané funkce nejsou zmapovány téměř vůbec. [32], [33], [34]

**TIOBE** umístění: Ruby se v TIOBE umístil na 16. místě s 0,8 % [17].

**PYPL** umístění: V PYPL najdeme Ruby na 15. místě s 1,1 % [18].

## **3.5 Tvorba webových stránek**

Před přistoupením k praktické ukázce webové tvorby jsou v následující kapitole představeny nutné předpoklady pro úspěšný vývoj webové aplikace. Jde o zásady a postupy, kterým obor web designu podléhá, dále pak výběr vhodného jazyka z výše zkoumaných technologií a popis použitého softwaru pro vývoj praktické ukázky.

### **3.5.1 Zásady**

Tvorba webových stránek se řídí mnoha pravidly a zásadami, které zajišťují kvalitu obsahu pro koncového uživatele. Mezi nejdůležitější zásady patří:

- **Použitelnost** – Použitelnost webové stránky se odvíjí podle schopnosti uživatelů se na webu orientovat, najít hledané informace, nebo podle celkového dojmu z používání stránky. Kvalitní web z hlediska použitelnosti by měl mít přehlednou úvodní stránku, která předá návštěvníku základní informace, dále by měla každá stránka webu obsahovat konzistentní umístění opakujících se prvků, například loga nebo navigačního menu. V poslední řadě by měly být dílčí stránky přehledně označeny, aby bylo jasné, kde v rámci aplikace se uživatel momentálně nachází. Všechny části aplikace by měly být také propojeny odkazy pro snadnou orientaci v rámci webu.

- **Přístupnost** – Přístupnost označuje přizpůsobení obsahu stránky uživatelům se zrakovým, sluchovým nebo pohybovým tělesným postižením. Přístupný web mohou lidé používat bez ohledu na svůj tělesný stav. Obsah musí být logicky strukturován a sekce opatřeny nadpisy, barevný kontrast písma musí být dostatečný a grafické a pohyblivé prvky na stránce by měly mít vyplněná klíčová slova či pomocná označení.
- **Rozložení** – Standardní rozložení úvodní stránky webu se skládá z pěti základních prvků: headeru obsahujícího většinou logo dané společnosti, navigačního menu s odkazy na další dílčí stránky, hero sekce neboli poutacího banneru, samotného obsahu stránky a footeru. Prvky by měly mít sladěný design komponent, jednotný font písma a stejné použité barevné kombinace.
- **Responzivita** – Responzivita webu zajišťuje korektní zobrazení jeho obsahu bez ohledu na velikost displeje zařízení, které ke stránce přistupuje. Potřeba responzivity vychází převážně z masového rozšíření chytrých mobilních telefonů a jejich využívání pro webové prohlížení. Responzivitu obsahu je možné dosáhnout používáním relativních jednotek (REM), které přizpůsobují svou velikost podle celkové velikosti displeje, a užitím „Media Queries“, což jsou bloky CSS vlastností aktivovaných podle velikosti zařízení. Také mnoho CSS frameworků (např. Bootstrap) je od základu responzivních a stará se o přizpůsobení obsahu automaticky. [13], [35]

### 3.5.2 Postup

Vytváření webového obsahu nezahrnuje pouze samotné programování v používaných jazycích, jde o komplexní vícefázový proces zahrnující ekonomické, grafické a programátorské segmenty, na jejichž konci by se měl nacházet funkční a marketingově úspěšný web, jehož struktura je uzpůsobena podle oboru zaměření. Mezi důležité fáze vývoje patří:

1. **Monitoring trhu** – Nejdříve je zjišťována motivace potenciálních zákazníků či uživatelů webu: jaké mají nároky, co očekávají, na co jsou zvyklí a podobně.
2. **Návrh designu webu** – Před zahájením samotného procesu tvorby obsahu webu je vytvořen návrh rozložení funkčních prvků na úvodní stránce, tzv. Wireframe. Ten slouží jako opora pro pozdější grafické návrhy a celkový

vzhled stránky. Wireframe existuje několik typů rozdělených podle míry věrnosti, kterou zobrazují požadované rozložení. Low-Fidelity Wireframy jsou pouze hrubé, většinou ručně kreslené skici přibližného rozmístění prvků na stránce. Mid-Fidelity přidávají přesnější reprezentaci kýženého designu, jednotlivé prvky už jsou seřazeny, viditelně odděleny a mohou být barevně rozlišeny pomocí odstínů šedé. High-Fidelity přidávají konkrétní grafické a textové prvky, která předchozí typy neobsahují. Jde o podrobný návrh téměř kopírující vzhled finálního produktu.

3. Tvorba obsahu webu – V tomto bodě probíhá samotná tvorba webového obsahu. Na základě průzkumu trhu a návrhů obsahu a grafiky je zahájen vývoj webové aplikace. Ten sestává z výsledného grafického zpracování (front-end) a technické realizace jednotlivých funkcí (back-end).
4. Testování a nasazení webu – Po vytvoření funkčního webu je nutné jej nejprve otestovat a následně nasadit. Dochází také k optimalizaci obsahu pro vyhledávače (SEO), aby se stránka uživatelů zobrazovala pro zadání klíčových slov například do Google vyhledávání. [13], [35]

Tato bakalářská práce se věnuje podrobně pouze třetímu bodu, a to samotné tvorbě webového obsahu pomocí front-end a back-end technologií.

### 3.5.3 Výběr jazyka pro praktickou ukázkou

Pro potřeby ukázky vývoje v praktické části práce je nutné z výše analyzovaných skriptovacích jazyků vybrat takový, který skupinu nejlépe reprezentuje a je nejvhodnější pro demonstraci jednoduchých softwarových funkcí v ukázkové webové aplikaci.

Vzhledem k stručné povaze ukázky a nutnosti vývoje aplikace „od nuly“ je vhodné vybrat takový jazyk, který neklade příliš vysoké nároky na programátora, poskytuje snadno osvojitelnou syntaxi, a umožňuje rychlý a komfortní vývojový proces. V tomto ohledu exceluje jazyk PHP nabízející „user friendly“ syntax vhodný pro začínajícího programátora. Integrace jazyka v přednastavených datových balíčcích (LAMP, XAMPP atd.) a možnost skripty přímo vkládat do HTML kódu dále usnadňují vývoj aplikací.

Nelze v tomto ohledu opomenout ani jazyky Python a Ruby, které jsou funkčně velmi podobné. Jejich syntax připomíná mluvený jazyk, což značně snižuje dobu nutnou pro

osvojení a zvyšuje přehlednost zdrojového kódu. Pouze nutnost ovládat základy objektivě orientovaného paradigmatu mírně navyšuje celkovou obtížnost.

Z hlediska nároků na programátora jsou méně vhodnými variantami jazyky JavaScript a C# (zastupující framework ASP.NET Core). V případě jazyka JavaScript je to z důvodu delší doby nutné pro pochopení jeho komplexní syntaxe vycházející z kompilovaných jazyků (C, Java), dále pak z důvodu jeho nepřilíšného zastoupení mezi serverovými technologiemi (jde primárně o front-end nástroj). C# je na rozdíl od ostatních kandidátů kompilovaný programovací jazyk, což se projevuje zvýšenou složitostí programů.

Pokud má vybraný jazyk reprezentovat současnou tvorbu dynamických webových stránek, mělo by se jednat o hojně využívaný a populární nástroj s aktivní komunitou. Zdaleka nejpopulárnějším se jeví jazyk Python, dominující žebříčkům popularity a počtů vyhledávání. To je ale dáno především jeho všestranným možnostem využití, v serverovém skriptování jde o méně často aplikovanou technologii (1,4 % podle W3Techs) [16].

Na rozdíl od Pythonu popularita PHP spíše stagnuje, jde ale o dominantní technologii pro tvorbu webových aplikací. C# nabízí rovnováhu mezi oběma faktory, zatímco všeobecně populární JavaScript je v odvětví spíše podreprezentován. Jako nejméně populární je označen jazyk Ruby, který ale používá nezanedbatelné procento webových stránek (5,2 %) [16].

Na základě poznatků shromážděných v literární rešerši je doporučeno použít pro tvorbu ukázkového webu skriptovací jazyk **PHP**. Jde o technologii se strmou křivkou učení, s jednoduchou syntaxí a hojně využívanou mezi současnými webovými stránkami a službami. Existence celosvětové a aktivní komunity svědčí o přetrvávajícím zájmu o jazyk a jeho relevanci pro odvětví tvorby webových aplikací.

### 3.5.4 Použitý software

Následující kapitola se zabývá přehledem softwaru, který v praktické části práce tvoří nezbytnou součást vývoje ukázkového webu s využitím vybraného skriptovacího programovacího jazyka PHP.

#### **Bootstrap**

Bootstrap je oblíbený CSS a JavaScript framework založený na flexboxu<sup>23</sup>. V práci je využíván z důvodu možnosti jednoduchého stylování webové stránky přímo v dokumentu HTML bez nutnosti připojovat vlastní CSS soubor, což značně snižuje náročnost zápisu. Použitím Bootstrap je zároveň v aplikaci zajištěna responzivita obsahu neboli jeho schopnost přizpůsobit své zobrazení podle velikosti displeje využívaného zařízení. Při tvorbě webové aplikace je využita aktuální verze frameworku: Bootstrap **5.3.2**. [14]

Práce s frameworkem začíná propojením se stylopisem Bootstrap v hlavičce HTML dokumentu pomocí odkazů na CDN<sup>24</sup> uvedených v ukázce zdrojového kódu č. 6. Stejným způsobem je propojena i JavaScript část frameworku, pomocí níž jsou v aplikaci tvořeny dynamické prvky stránky, například tlačítka, dropboxy, nebo vysunovací menu.

V ukázce kódu č. 6 je v části `<body>` ilustrováno jednoduché použití stylizace stránky pomocí frameworku. Výchozí rozdělení stránky v Bootstrap na 12 sloupců je ovládáno pomocí atributu `col-sm-6`. Ten říká, že pro displeje větší, než malé (576px) bude layout řádku tvořen dvěma sloupci zabírajícími šest z dvanácti dostupných míst pro sloupce, tedy polovinu stránky.

---

<sup>23</sup> CSS model rozložení webu pomocí responzivních pružných obsahových boxů [14]

<sup>24</sup> Content Delivery Network – veřejná síť zprostředkující obsah, v tomto případě umožňuje načítání CSS a JavaScript komponent ze svých serverů [36].

```

<!DOCTYPE html>
<html lang="cs">
<head>
  <title>Bootstrap ukázka</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!--Propojení HTML s CSS a JavaScript knihovnamí Bootstrap přes CDN -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.
js"></script>
</head>
<body>

<!--Vytvoření dvojice containeru obsahují řádek se dvěma sloupci s 3 REM
padding, okraji a změnou barvou pozadí pomocí syntaxe Bootstrap-->
<div class="container">
  <div class="row">
    <div class="col-sm-6"><div class="p-3 border bg light">Sloupec1</div></div>
    <div class="col-sm-6"><div class="p-3 border bg-light">Sloupec2</div></div>
  </div>
</div>

</body>
</html>

```

Zdrojový kód 6 – Ukázka propojení dokumentu HTML s CDN frameworku Bootstrap

## XAMPP

XAMPP je multiplatformní balíček otevřeného softwaru, který umožňuje jednoduše vytvořit lokální server pro vývoj a testování aplikací využívajících server-side skriptování v jazyce PHP (nebo Perl). Slouží jako bezplatná alternativa pořízení serveru pro nutnost vývoje aplikací v PHP a ve své instalaci zahrnuje mnoho přídavného softwaru pro usnadnění tvorby výsledného webu. Jeho použití ale není překážkou pro budoucí implementaci aplikace na internetu, naopak jsou jeho komponenty kompatibilní s většinou dostupných webových serverů, nasazení aplikace do praxe je tedy velmi jednoduché. XAMPP se skládá z:

- Lokálně hostovaného webového serveru Apache, který zajišťuje provedení PHP skriptů ukázkové aplikace a přenášení výsledků činnosti uživateli. Jde o multiplatformní server, aplikace může být tedy nasazena na libovolném operačním systému.
- Relační databáze MariaDB, nástupce MySQL, která se stará o uchovávání dat o uživateli aplikace a dalších dat nutných pro fungování skriptů. Databáze



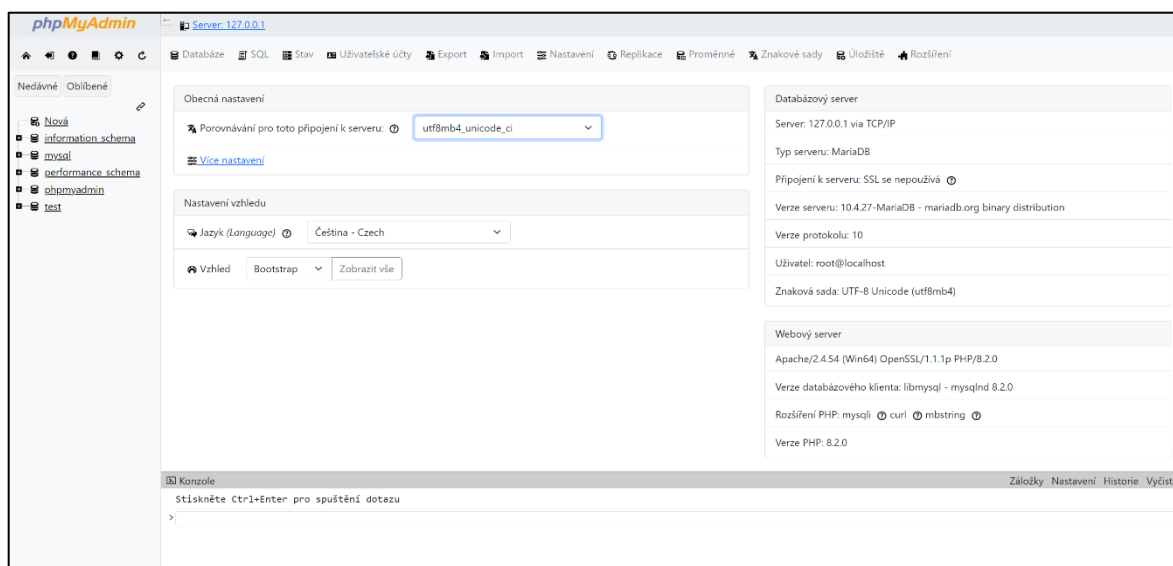
může být použita například pro uložení registračních údajů zadaných uživatelem do tabulky v databázi, kde k nim může být podle potřeby přistupováno.

- Skriptovacího jazyka PHP pohánějícího chod celé aplikace pomocí do HTML vnořených skriptů. PHP se stará o propojení stránky s databází, o ukládání vstupů od uživatele, a také zaslání finální podoby stránek do webového prohlížeče. [37]

XAMPP obsahuje i mnoho dalších softwarových doplňků, pro potřeby ukázky bude využit pouze nástroj phpMyAdmin, o kterém pojednává následující podkapitola. V práci je využívána aktuální verze XAMPP 8.2.0 (odvozeno od obsažené verze PHP 8.2.0).

## phpMyAdmin

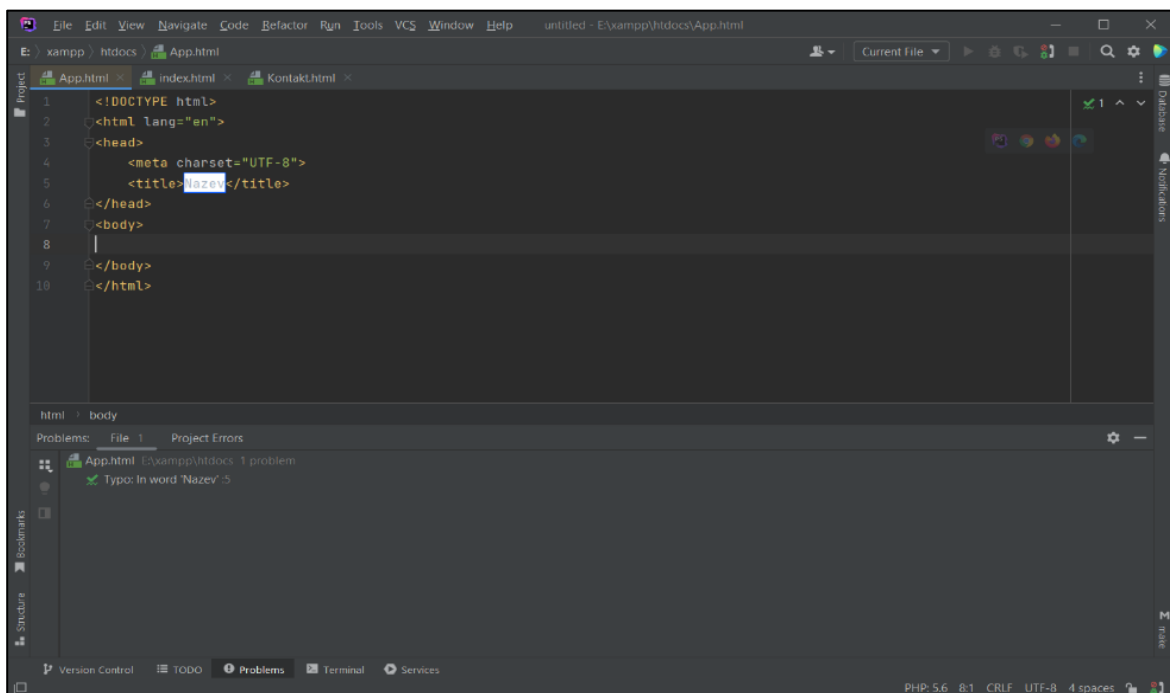
PhpMyAdmin je GUI nástroj napsaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MariaDB (Obrázek 5). Stará se o vytváření a rušení databází a tabulek, provádí SQL příkazy a spravuje primární a cizí klíče. Poskytuje mnohem pohodlnější a přehlednější přístup do databáze a k jednotlivým instancím. Aplikace jej využívá pro vytváření tabulek pro uchovávání informací o uživateli a jejich následné správě. [38]



Obrázek 5 – Ukázka UI aplikace phpMyAdmin [38]

## PHP Storm

PHP Storm je komerční vývojové prostředí pro jazyk PHP od společnosti JetBrains. Jde o placený software, který je ale poskytován studentům vysokých škol zdarma, čehož je pro potřeby práce využito. Tento editor byl vybrán z důvodu podpory nejen pro jazyk PHP, ale i další jazyky používané při webové tvorbě: HTML, CSS, JavaScript a SQL. Umožňuje psaní veškerého potřebného kódu pro aplikaci v rámci jednoho prostředí, což značně usnadňuje vývoj. Zvýraznění syntaxe a našeptávač kódu dále přispívají k přehlednosti a komfortu při programování. Design prostředí PHP Storm ilustruje Obrázek č. 6. [39]



Obrázek 6 – Ukázka UI textového editoru PHP Storm [39]

## 4 Vlastní práce

V praktické části této bakalářské práce je popsán samotný vývoj ukázkové aplikace, na které je demonstrováno praktické využití jazyka PHP a dalších technologií používaných při web designu: HTML, CSS, JavaScript a SQL. Praktická část je rozdělena na tři části:

- Front-end aplikace sestávající ze značkovacího jazyka **HTML**, který určuje rozložení stránky a dává jednotlivým prvkům význam a smysl. O stylování jednotlivých elementů stránky se starají kaskádové styly, v této ukázce reprezentované frameworkem **Bootstrap**, který zajišťuje responzivní a srozumitelný design aplikace. Pomocí něj je řešeno také ovládání interaktivních prvků v jazyce **JavaScript**, jde například o tlačítka, vysunovací menu, dropboxy a další pohyblivé části stránky.
- Back-end aplikace využívající skriptovací programovací jazyk **PHP**, který dále rozšiřuje původní HTML dokument o embedded (vnořené) skripty zajišťující výměnu informací s uživateli a zasílání dat databázovému systému.
- Databázový systém **MariaDB** využívající dotazovacího jazyka **SQL**, který slouží aplikaci na uchování dat o uživateli.

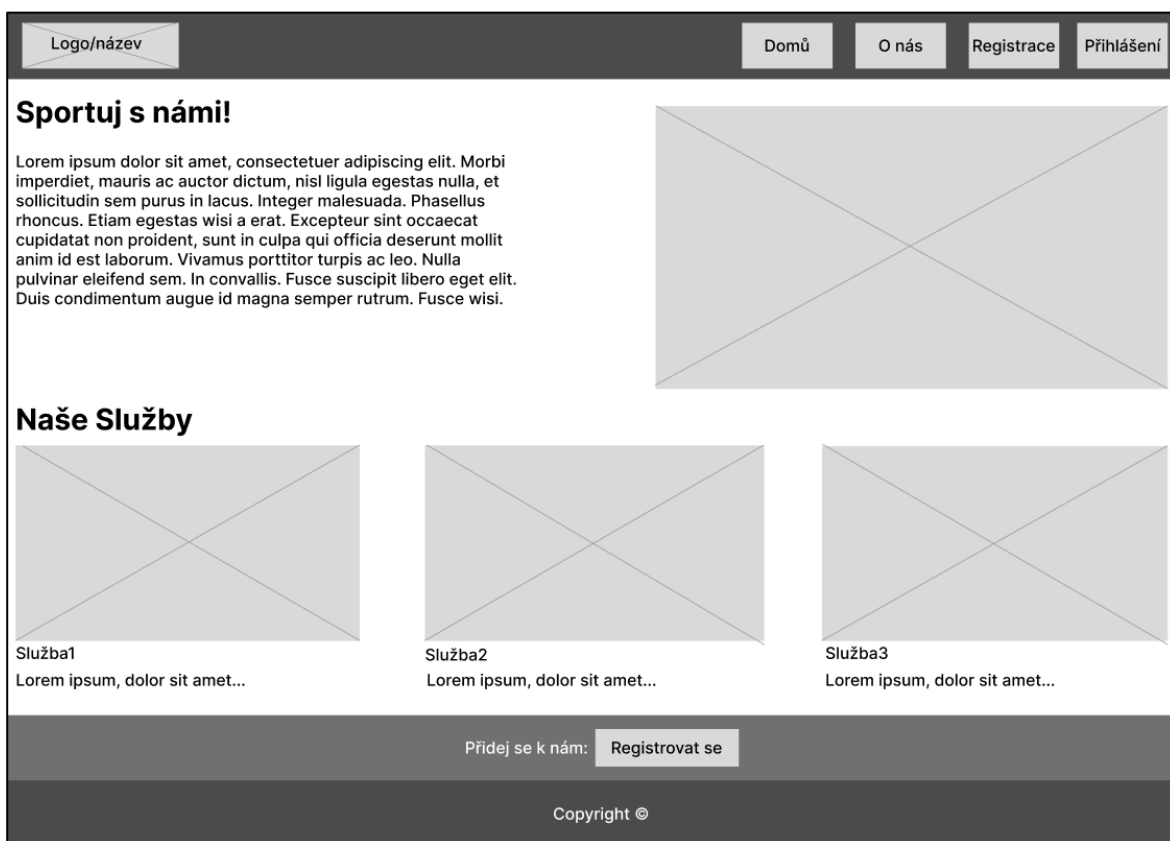
Ukázka webové aplikace s názvem *Fitter & Happier* je tvořena pro fiktivní stránku zabývající se odvětvím zdravého životního stylu a fitness. Svým uživatelům umožňuje se na stránce registrovat a přihlašovat se zvolenými údaji, vytvářet vlastní cíle a milníky pro cvičební účely a zaznamenávat průběh a intenzitu cvičení v čase. Tématika je zvolena cíleně pro snadné demonstrování funkcionalit jazyka PHP, jeho hojně využívaný způsob použití v praxi a provázanost s dalšími technologiemi pro webovou tvorbu. Aplikace a její používání může zároveň posloužit jako motivační nástroj pro fyzickou aktivitu a zdravější životní styl autora stránky i potenciálních budoucích uživatelů.

## 4.1 Front-end

V této části je představena front-end řešení webové aplikace, které spočívá v návrhu rozložení stránky (Wireframe), představením struktury aplikace a jednotlivých používaných prvků na stránce a rozboru zdrojového kódu v jazyce HTML.

### 4.1.1 Návrh webu

Před zahájením tvorby obsahu webové aplikace je nutné navrhnout rozložení grafických a textových prvků na stránce. Proto je vytvořen následující Medium-Fidelity Wireframe (Obrázek 7), který bude sloužit jako opora při vývoji front-end části aplikace. Uspořádání vychází ze standardního rozložení prvků používaného většinou moderních webových stránek. Domovská stránka se skládá z Headeru (záhlaví) obsahujícího navigační menu a logo firmy, Hero sekce (dále nazvané Představení) přibližující účel aplikace, sekce Služby tvořící vlastní obsah stránky a dále rozvíjející informace o projektu a Footer sekce (zápatí) ohraničující stránku zespodu. Sekce Představení a Služby budou pro ilustraci tématu stránky a nabízených funkcí doplněny o příslušné obrázky.



Obrázek 7 – Wireframe Domovské stránky ukázkového webu

#### 4.1.2 Struktura webové aplikace

Web se skládá z následujících dílčích webových stránek, které společně tvoří celou podobu aplikace *Fitter & Happier*:

- Domovská stránka – Úvodní stránka celé aplikace, slouží jako seznámení s projektem a souhrn základních informací o něm. Obsahuje dvě sekce: Představení a Služby.
- O nás – Tato stránka má za úkol uživatele informovat o kolektivu stojícím za webovou aplikací, jejich motivací za výběrem zaměření a výčet jejich schopností a vědomostí z oboru fitness. Kromě samotné sekce „O nás“ obsahuje také sekci Kontakty s potřebnými podklady pro komunikaci s autory aplikace.
- Registrace – Jedná se o jednoduchý formulář pro registraci nového uživatele pomocí přezdívky, e-mailu a hesla. Uživatel je obeznámen s kritérii výběru pro přezdívku a heslo
- Přihlášení – Formulář s obdobným designem jako registrace, zde se mohou zaregistrovaní uživatelé přihlásit do aplikace.
- Aplikace – Stránka, na kterou je uživatel po úspěšném přihlášení přesměrován, jedná se o samotné jádro aplikace obsahující služby inzerované na Domovské stránce.

#### 4.1.3 HTML hlavička

Před samotným rozbořením zdrojového kódu jednotlivých sekcí je nutné zmínit hlavičku dokumentu HTML neboli element `<head>` (Zdrojový kód 7). Ta v sobě nese základní informace o dokumentu (viz kapitola 3.2) a propojení s CDN frameworku Bootstrap (Zdrojový kód 6). Hlavička zůstává v rámci dílčích stránek stále stejná, pouze element `<title>` (název) se bude měnit podle tématu dané stránky. První použitý meta tag říká prohlížeči, jaké kódování pro text bude použito, zde jde o UTF-8. Tímto je zajištěna správná interpretace a zobrazení textu na stránce. Druhý meta tag upravuje šířku stránky podle šířky použitého zařízení (`width=device-width`) a určuje výchozí úroveň přiblížení stránky (`initial-scale=1.0`). Tyto vlastnosti jsou důležité pro správné zobrazení webové stránky na zařízeních s různými rozměry obrazovky.

```
<!-- Hlavička s meta tagy, názvem stránky a propojením s Bootstrap CDN -->
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fitter & Happier</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
```

Zdrojový kód 7 – Sdílená hlavička HTML dokumentů ve webové aplikaci

#### 4.1.4 Záhloví a navigační lišta

Hlavním prvkem každé stránky je záhlaví umístěné navrchu, které zároveň slouží i jako navigační lišta pro průchod jednotlivých částí webové aplikace. Záhlaví se skládá z názvu produktu se zvýrazněnou barvou pro snadnou čitelnost, dále pak z linků (prokliků) na ostatní stránky aplikace: Domovskou stránku, sekci „O nás“, Registraci a Přihlášení. Poslední dvě zmiňované jsou taktéž barevně odděleny od ostatních linků, aby je mohl uživatel na první pohled nalézt a použít.

V následujícím výčtu bude představeno řešení samotné navigační lišty (Zdrojový kód 8), obsažené elementy a jejich atributy:

- Element *nav* (navigační menu) – Tento element definuje navigační menu pro webovou stránku. Obsahuje Bootstrap třídy, které upravují jeho výsledný vzhled: *navbar-expand-lg* upřesňuje, od jaké velikosti displeje se má navbar rozložit (lg = 960px), *navbar-dark* a *bg-dark* mu přidávají tmavý vzhled.
- Element *div* (kontejner) – Následující element se používá jako obecný kontejner pro vkládání jiných elementů a pomáhá organizaci stránky. Obsahuje Bootstrap třídu *container* zajišťující responzivitu.
- Element *a* (odkaz) – Odkaz zde slouží zároveň jako název společnosti a proklik na Domovskou obrazovku. Třídy *navbar-brand* a *text-info* upravují vzhled a barvu použitého textu v odkazu (tyrkysově zvýrazněný text). Atribut *href* určuje přesměrování po kliknutí, zde je uveden odkaz na Index.html, což je domovská stránka aplikace.

- Element *button* (tlačítko) – Tento element definuje tlačítko, které lze na mobilních zařízeních použít k rozvinutí nebo sbalení navigačního menu. Třída *navbar-toggler* značí použitou stylizaci tlačítka, atribut *type* obsahuje informaci, že se jedná o tlačítko, a atributy *data-bs-toggle* a *data-bs-target* (JavaScript prvky Bootstrap) určují, že obsah bude ve výchozí poloze sbalený (*collapse*) a ovládán je cílový prvek s id *navbarNav*. Poslední trojice atributů (*aria-controls*, *aria-expanded* a *aria-label*) se stará o přístupnost pro zrakově postižené, varuje je před ve výchozí poloze skrytým menu a poskytuje popis využití tlačítka.
- Element *span* – Span vnořený do předcházejícího elementu určuje pomocí Bootstrap třídy *navbar-toggler-icon* design ikony použité pro dropdown navigační tlačítko.
- Element *ul* (seznam) – Jde o nečíslovaný seznam sestávající z jednotlivých položek navigačního menu. Celý seznam je uzavřen v divu (kontejneru), který pomocí tříd *collapse* a *navbar-collapse* definuje skrytí a organizaci obsahu navigačního seznamu. Samotný seznam má přidělený automatický levý okraj přes třídu *ms-auto*.
- Element *li* (položka seznamu) – Tento element definuje jednu položku seznamu. Obsahuje stylizační třídu *nav-item* a jednotlivé odkazy na dílčí stránky webu. Odkaz pro registraci a přihlášení má pomocí třídy *text-primary* změněnou barvu textu na modrou.

```

<!-- Navigační lišta -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container">
    <a class="navbar-brand text-info" href="Index.html">Fitter &
Happier</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Vysunout/Zasunout navigační menu">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item">
          <a class="nav-link" href="Index.html">Domů</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="Kontakt.html">O nás</a>
        </li>
        <li class="nav-item">
          <a class="nav-link text-primary"
href="Register.html">Registrace</a>
        </li>
        <li class="nav-item">
          <a class="nav-link text-primary"
href="Login.php">Přihlášení</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

```

Zdrojový kód 8 – Navigační lišta s responzivním dropdown menu

#### 4.1.5 Footer

Dalším opakujícím se prvkem většiny stránek (kromě forem pro registraci a přihlášení) je footer neboli zápatí (Zdrojový kód 9). Ten obsahuje proklik na registraci pro snadnější přístup i po sjetí po obsahu stránky dolů, dále je zde uveden copyright včetně názvu projektu a jméno tvůrce webové aplikace. Je složen z následujících prvků:

- Element *footer* – Tento element je určen pro definování sekce zápatí na webové stránce. Obsahuje třídy *text-center*, *text-white* a *bg-dark* upravující umístění textu, barvu textu a barvu pozadí. Footer se skládá z dvou vnořených elementů *div*, první obsahuje registrační tlačítko, druhý Copyright sekci.
- Sekce Registrační tlačítko – Jedná se o *div* s upraveným paddingem (vnitřními okraji) obsahující vnořený element *p* (paragraph) zobrazující text. Třídy *d-flex*, *justify-content-center* a *align-items-center* nastavují zarovnání a vycentrování obsahu paragrafu. Samotný paragraf se dále skládá z řádkového elementu *span*



a pomocí Bootstrap stylovaného tlačítka (*btn*, *btn-outline-light*, *btn-rounded*) s odkazem na registrační formu.

- Sekce Copyright – V tomto *divu* je vycentrován text a určeny vnitřní okraje (*text-center p-3*), dále je pomocí atributu *style* změněn barevný odstín (*background-color: rgba(0, 0, 0, 0.2)*) odlišující sekci od předcházející.

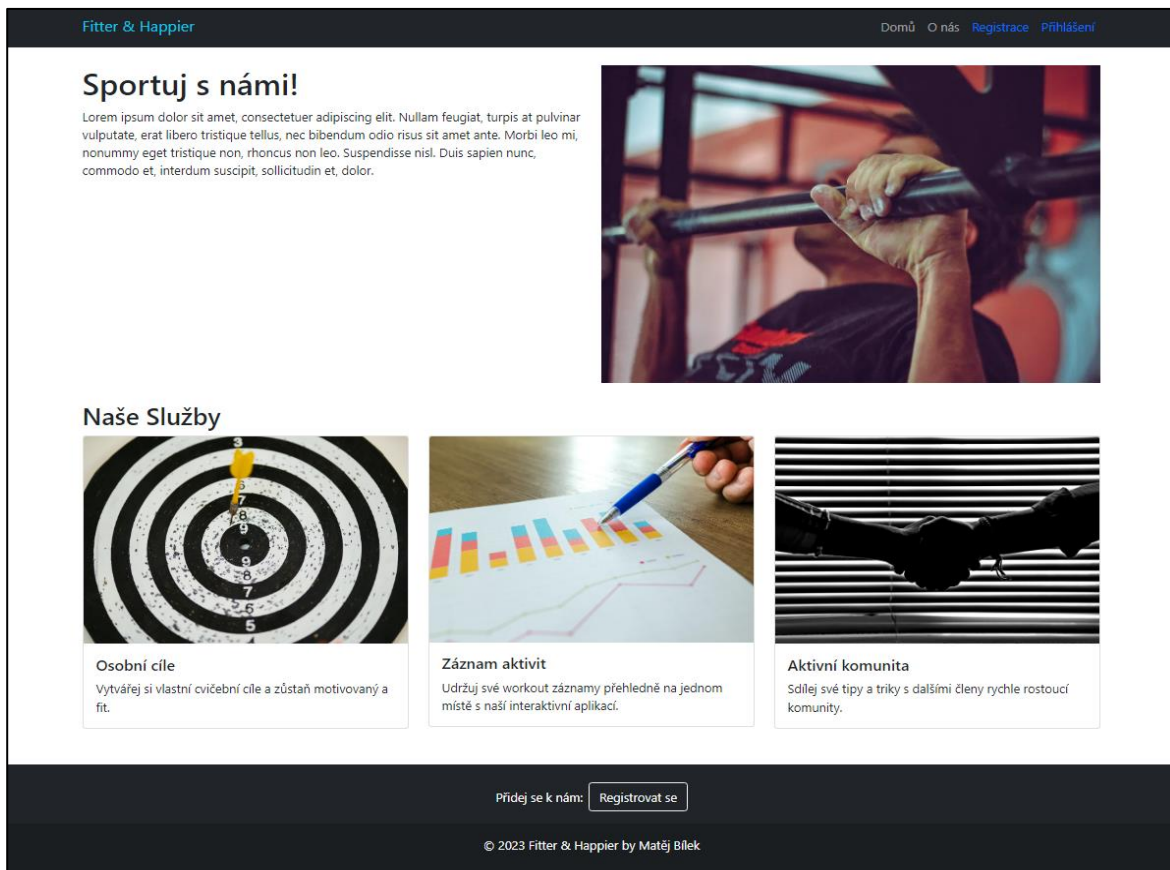
```
<!-- Footer -->
<footer class="text-center text-white bg-dark">
  <div class="container p-4 pb-0">
    <p class="d-flex justify-content-center align-items-center">
      <span class="me-2">Přidej se k nám:</span>
      <a href="Register.html" class="btn btn-outline-light
btn-rounded" role="button">Registrovat se</a>
    </p>
  </div>
  <!-- Copyright-->
  <div class="text-center p-3" style="background-color: rgba(0, 0, 0,
0.2);">
    <p class="text"> © 2023 Fitter & Happier by Matěj Bílek</p>
  </div>
</footer>
```

Zdrojový kód 9 – Footer se dvěma sekcemi

#### 4.1.6 Domovská stránka

V této subkapitole je představena domovská stránka aplikace (Obrázek 8), tedy první obsah, který návštěvník uvidí po zobrazení webu. Jsou zde proto obsaženy důležité informace o projektu prostřednictvím dvou sekcí (*section*):

- Představení – Sekce obsahující nadpis *h1*, který láká na přečtení úvodního textu o aplikaci a samotného paragrafu textu. Pro účely ukázky je vložen standardní pseudolatinský text „lorem ipsum“ sloužící jako výplň před osazením stránky finálním článkem. Vedle článku je zobrazen ilustrační obrázek s workout tématikou pro zdůraznění zaměření stránky. Veškeré použité obrázky na stránce pocházejí z webového archivu s volnou licencí: Pexels [40].
- Služby – Tato sekce se snaží zaujmout návštěvníka pomocí výčtu služeb nabízených aplikací. Využívá pro to Bootstrap stylovací kontejner *card*, který obsahuje náhledovou fotografii činnosti (element *img*), název služby (element *h2*) a popis služby (element *p*).



Obrázek 8 – Domovská stránka aplikace

#### 4.1.7 Stránka „O nás“


Tato stránka (Obrázek 9) slouží jako informační a kontaktní zdroj prostřednictvím dvou sekcí:

- O nás – Jde o představení kolektivu stojícím za aplikací, vizuál je řešen stejným způsobem jako sekce Představení, tedy jednoduchým kontejnerem s nadpisem, samotným textem (lorem ipsum) a obrázkem.
- Kontakty – V této sekci je s použitím Bootstrap vytvořen kontaktní formulář, který slouží pro zaslání zpětné vazby autorům aplikace. Dále jsou zde uvedeny důležité údaje pro komunikaci s autory: kontaktní adresa, telefonní číslo a emailová adresa.

Fitter & Happier Domů O nás Registrace Přihlášení

## O nás

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam feugiat, turpis at pulvinar vulputate, erat libero tristique tellus, nec bibendum odio risus sit amet ante. Morbi leo mi, nonummy eget tristique non, rhoncus non leo. Suspendisse nisl. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Nulla quis diam. Aenean placerat. Aliquam id dolor. Vivamus porttitor turpis ac leo. Maecenas aliquet accumsan leo. Quisque porta. Pellentesque ipsum. Duis risus. Aliquam erat volutpat. Suspendisse sagittis ultrices augue. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Etiam commodo dui eget wisi. Nullam justo enim, consectetur nec, ullamcorper ac, vestibulum in, elit. Etiam dui sem, fermentum vitae, sagittis id, malesuada in, quam. Aenean placerat.



## Kontaktuj nás

Jméno

Email

Zpráva

**Kontaktní adresa:**  
 ul. Fitness 123  
 101 00 Praha 10  
 Česká republika

**mobil:** +420 123 456 789  
**e-mail:** fitnessci@fitness.com

Přidej se k nám:

© 2023 Fitter & Happier by Matěj Bílek

Obrázek 9 – Stránka „O nás“

### 4.1.8 Registrační formulář

Stránka Registrace používá jednoduchý Bootstrap formulář (Obrázek 10) pro možnost registrování nového uživatele a jeho následné přihlášení do aplikace na obdobně stylované stránce „Přihlášení“. Formulář je spojen s externím PHP souborem *Register.php*, který data poskytnutá uživatelem zpracuje. Jde o jeden ze dvou možných způsobů, jak PHP skripty do stránky zakomponovat. Druhý způsob spočívá ve sloučení HTML zápisu a samotných skriptů do jednoho souboru s koncovkou *.php* zajišťujícího vizuál i funkcionality. Tento přístup bude představen v kapitole 4.3 na přihlašovací formuláři a obsahu dostupném uživateli po přihlášení do aplikace.

## Registrace

Přezdívka

Přezdívka musí být 3-20 znaků dlouhá a nesmí obsahovat mezery nebo speciální znaky.

E-mail

Heslo

Heslo musí být 8-20 znaků dlouhé a nesmí obsahovat mezery nebo speciální znaky.

**Registrovat se**

Už máš účet? [Přihlaš se zde](#)

Obrázek 10 – Registrační formulář

## 4.2 Back-end

V následující kapitole je představena nejdůležitější část tvorby dynamické webové stránky, samotné použití skriptovacího jazyka PHP pro zajištění funkcionalit vytyčených v úvodu a dále rozvedených v praktické části práce. Skripty zajišťují registraci a přihlašování uživatelů na stránce, ošetřují a validují jejich vstupy, ukládají a načítají informace do tabulek v databázi a umožňují přihlášeným uživatelům využívat funkcí aplikace: přidávání cvičebních cílů a záznam předchozích cvičení. Funkce byly vybrány tak, aby bylo na jednoduchých ukázkách ilustrováno praktické využití jazyka PHP.

### 4.2.1 Registrace

Princip registrace je první klíčovou funkcí, kterou v aplikaci zajišťují PHP skripty. Data vložená do registračního formuláře (kapitola 4.1.8) se odešlou HTTP metodou POST na server, kde jsou zadané vstupy zpracovány následujícím skriptem.

Jako první je načten konfigurační soubor obsahující citlivé údaje (jméno serveru, admin login, heslo a jméno databáze), který zároveň vytvoří nové spojení s danou databází pomocí metody *new mysqli()*. Vytvořené spojení je uloženo do proměnné *\$con*. Důvodem použití externího souboru je oddělení citlivých údajů k databázi od samotného zdrojového kódu, což snižuje riziko napadení. Pro ještě vyšší bezpečnost by se měl soubor *db\_config.php* nacházet mimo kořenový adresář serveru.

Skript (Zdrojový kód 10) se skládá z několika částí:

- Podmínky *if* kontrolující, zda byla data z formuláře odeslána metodou POST, pokud ano, skript se provede, pokud ne, je zobrazena chybová hláška.
- Uložení hodnot z registračního formuláře do proměnných a ošetření vstupů proti SQL injection použitím funkce *mysqli\_real\_escape\_string()*.
- Blok podmínek, které kontrolují splnění požadavků na vstupy uvedené v registračním formuláři pomocí regulárních výrazů a bloků *if* s použitím funkcí *preg\_match()* a *filter\_var()*. Po zadání neplatného vstupu je uživatel upozorněn na chybu a přesměrován zpět na stránku s registrací (funkcí *header*).
- Pokud validace vstupů proběhla úspěšně, je ověřeno, zda uživatel se zadaným e-mailem již není v aplikaci zaregistrován. Je proveden SQL dotaz na databázi, která vybere řádek, kde se uložený e-mail rovná zadanému. Pokud je počet nalezených řádků větší než nula (tedy e-mail již v databázi existuje), uživateli je zobrazena příslušná chybová hláška a je přesměrován zpět na stránku registrace.
- Pokud jsou vstupy validní a e-mail není používán, přezdívkou, e-mail a heslo jsou vloženy do databázové tabulky Users příslušným SQL dotazem. Heslo je před uložením zahashováno (funkcí *password\_hash*), aby při možném úniku dat nemohl být účet odcizen. Dále je úspěšně registrovaný uživatel přesměrován na přihlašovací formulář, kde se může se svým nově vytvořeným účtem přihlásit. V případě neúspěšné registrace je zobrazena chybová hláška s důvodem neúspěchu.
- Na konci skriptu je uzavřeno spojení s databází pomocí funkce *close()*, aby k ní dále nebylo možné přistupovat a vkládat hodnoty.

```

<?php

// Konfigurační soubor pro navázání spojení s databází
require_once('db_config.php');

// overeni, zda byl registracni formular odeslan
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // ziskani hodnot z registracniho formulare, osetreni proti SQL injection
    $name = mysqli_real_escape_string($conn, $_POST["name"]);
    $email = mysqli_real_escape_string($conn, $_POST["email"]);
    $password = mysqli_real_escape_string($conn, $_POST["password"]);
    // validace hodnot z registracniho formulare pres regularni vyrazy
    $name_pattern = "/^\S{3,20}$/"; // presne 3-16 neprazdnych znaku
    $password_pattern = "/^\S{8,20}$/"; // presne 8-20 neprazdnych znaku

    if (!preg_match($name_pattern, $name)) {
        header('Refresh: 3; URL=Register.html');
        die("Chybný formát přezdívky - Přezdívka musí být 3-20 znaků dlouhá a nesmí obsahovat mezery nebo speciální znaky."); }

    if (!preg_match($password_pattern, $password)) {
        header('Refresh: 3; URL=Register.html');
        die("Chybný formát hesla - Heslo musí být 8-20 znaků dlouhé a nesmí obsahovat mezery nebo speciální znaky."); }

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        header('Refresh: 3; URL=Register.html');
        die("Neplatná emailová adresa."); }

    // overeni, zda uzivatel jiz neni zaregistrovan
    $sql = "SELECT email FROM users WHERE email='$email'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        header('Refresh: 3; URL=Register.html');
        die("Tento e-mail už je používán."); }

    // zahashování hesla a vložení noveho uzivatele do databaze
    $password_hash = password_hash($password, PASSWORD_DEFAULT);
    $sql = "INSERT INTO users (name, email, password)
        VALUES ('$name', '$email', '$password_hash')";
    if ($conn->query($sql) === TRUE) {
        ob_clean();
        header('Refresh: 3; URL=Login.php');
        echo "Registrace proběhla úspěšně.";

    } else {
        echo "Registrace se nezdařila: " . $conn->error; }
} else {
    echo "Požadavek nebyl zpracován."; }

$conn->close();

?>

```

Zdrojový kód 10 – Kód zajišťující registrace do aplikace

## 4.2.2 Přihlášení

Po úspěšném provedení registrace může uživatel využít následující přihlašovací formulář na stránce *Login.php*, který v sobě spojuje HTML kód a PHP skripty v jednom dokumentu. Na rozdíl od Registrace jsou zde chybové hlášky zobrazovány přímo pod formulářem (pomocí proměnné *error* volané při chybném zadání jména či hesla).

Přihlašovací skript (Zdrojový kód 11) funguje na podobném principu, jako registrace. Je navázáno nové spojení s databází, zkontrolováno odeslání formuláře a ošetřeny vstupy. Následně se provede dotaz na databázi pro vyhledání uživatele s odpovídajícím emailem. Pokud je uživatel nalezen, porovná se nalezené hashované heslo s heslem zadaným v přihlašovacím formuláři pomocí funkce *password\_verify()*. Pokud jsou hesla shodná, uživatel je přihlášen do aplikace pomocí funkce *session\_start()* a jeho údaje jsou uloženy v proměnných *\$\_SESSION*. Nakonec je uživatel přesměrován na stránku *App.php* pomocí funkce *header()*.

```
<?php
// Připojení k databázi
require_once('db_config.php');

// Zpracování přihlašovacího formuláře
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// Ošetření vstupů proti SQL injection
$email = mysqli_real_escape_string($conn, $_POST["email"]);
$password = mysqli_real_escape_string($conn, $_POST["password"]);

// Hledání uživatele v databázi
$sql = "SELECT * FROM users WHERE email = '$email'";
$result = $conn->query($sql);

if ($result->num_rows == 1) {
// Uživatel nalezen, porovnání hesla
$row = $result->fetch_assoc();
if (password_verify($password, $row["password"])) {
// Heslo se shoduje, přihlášení úspěšné
session_start();
$_SESSION["email"] = $email;
$_SESSION["name"] = $row["name"];
header("Location: App.php");
exit();
}
} else {
// Uživatel nenalezen nebo zadáno chybné heslo, přihlášení se nezdařilo
$error = "Chybné jméno nebo heslo";
}
$conn->close();
?>
```

Zdrojový kód 11 – Princip přihlášení do aplikace

### 4.2.3 Kontrola přístupu

Stránka *App.php* tvoří jádro aplikace přístupné přihlášeným uživatelům. V hlavičce stránky je obsažen následující kód (Zdrojový kód 12), který vytváří session, kontroluje, zda je uživatel přihlášen (pokud není, je přesměrován zpět na Domovskou stránku) a stará se o odhlašování uživatelů a jejich následné přesměrování mimo aplikaci.

```
<?php
// start session
session_start();

// Kontrola, zda je uživatel přihlášen
if (!isset($_SESSION["email"])) {
    header("Location: Login.php");
    exit();
}

// Odhlášení uživatele
if (isset($_POST["logout"])) {
    // Odstranění dat o přihlášeném uživateli ze session
    session_unset();
    session_destroy();
    // Přesměrování na Domovskou stránku po odhlášení
    header("Location: Index.html");
    exit();
}
?>
```

Zdrojový kód 12 – Kontrola přístupu a odhlášení uživatele

### 4.2.4 Ukládání vstupů

Pro potřebu ukládání uživatelských vstupů, které budou následně zaneseny do tabulky cílů, je nutné nejdříve informace validovat a poté provést přiřazení cílů aktuálně přihlášenému uživateli. Toho je docíleno použitím následujícího skriptu (Zdrojový kód 13), který vloží validované vstupy do databázové tabulky *Goals* spolu s e-mailem uživatele, který akci vyvolal. Tím je zajištěno, že se uživatelům zobrazují pouze jejich vlastní cíle či cviky. Přidávání cviků je řešeno analogicky s příslušnými hodnotami a použitím tabulky *Exercises*.



```

<?php
// Připojení k databázi pomocí konfiguračního souboru
require_once('db_config.php');
// Získání dat z formuláře
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = mysqli_real_escape_string($conn,$_POST["goal"]);
    $date = mysqli_real_escape_string($conn,$_POST["due_date"]);
    $priority = mysqli_real_escape_string($conn,$_POST["priority"]);

    // Získání emailu přihlášeného uživatele ze session
    session_start();
    $email = $_SESSION["email"];

    // Příprava a provedení SQL dotazu pro vložení dat
    $sql = "INSERT INTO goals (goal, due_date, priority, email)
           VALUES ('$name', '$date', '$priority', '$email')";

    if ($conn->query($sql) === TRUE) {
        echo "Cíl byl úspěšně vytvořen.";
        header('Refresh: 2; URL=app.php');
    } else {
        echo "Nastala chyba při vytváření cíle: " . $conn->error;
    }
} else {
    echo "Požadavek nebyl zpracován."; }
// Uzavření spojení s databází
$conn->close();
?>

```

Zdrojový kód 13 – Funkce přidání nového cíle

#### 4.2.5 Zobrazení vstupů

Po uložení vstupů do databáze je nutné je zobrazit v příslušné tabulce na stránce aplikace (Zdrojový kód 14). SQL dotazem jsou vybrány řádky příslušející přihlášenému uživateli. Pokud SQL dotaz vrátí nenulový počet řádků, provede se cyklus *while*, který pro každý databázový řádek vytvoří řádek v HTML tabulce s informacemi o cílech uživatele. Po úspěšném vytvoření řádku je uživatel přesměrován zpět na stránku aplikace.

Pro každý řádek se zobrazí název cíle, datum splnění, priorita a tlačítko "Smazat", které umožňuje uživateli odstranit cíl. Pokud SQL dotaz nevrátí žádné řádky, zobrazí se placeholder v tabulce HTML s upozorněním, že dosud nebyly vytvořeny žádné cíle. Zobrazení cviků v tabulce funguje na stejném principu.

```

<?php
// Získání dat z databáze a vytvoření řádků v tabulce
require "db_config.php"; //konfigurační soubor pro vytvoření spojení s databází
$email = $_SESSION['email'];
$sql = "SELECT goal_id, goal, due_date, priority FROM goals WHERE email =
'$email'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<tr>
            <td>" . $row["goal"] . "</td>
            <td>" . $row["due_date"] . "</td>
            <td>" . $row["priority"] . "</td>
            <td>
                <form method='POST' action='delete_goal.php'>
                <input type='hidden' name='goal_id' value='" .
$row["goal_id"] . "'>
                <button type='submit' class='btn text-danger p-
0'>Smazat</button>
                </form>
            </td>
        </tr>";
    }
} else {
    //pokud jsou cíle prázdné, je použit placeholder
    echo "<tr><td colspan='4' class='text-center'>Žádné cíle nebyly
vytvoreny.</td></tr>"; }
$conn->close();
?>

```

Zdrojový kód 14 – Funkce zobrazení vytvořeného cíle v tabulce

#### 4.2.6 Smazání vstupů

Pokud si uživatel přeje cíl smazat, může tak učinit kliknutím na tlačítko „Smazat“ vedle příslušného řádku tabulky, které aktivuje následující skript *delete\_goal.php* (Zdrojový kód 15), který pomocí unikátního ID spáruje řádek s protějškem v databázi a provede SQL dotaz DELETE na jeho vymazání. Smazání cviků funguje na stejném principu.

```

<?php
//testování odeslání proměnné metodou POST
if (isset($_POST['goal_id'])) {
require_once "db_config.php";
}
$goal_id = $_POST['goal_id'];
// Smazání řádku z databáze
$sql = "DELETE FROM goals WHERE goal_id = $goal_id";
if ($conn->query($sql) === TRUE) {
    echo "Cíl byl úspěšně smazán.";
    header('Refresh: 2; URL=app.php');
} else {
    echo "Chyba při mazání cíle: " . $conn->error; }
$conn->close();
?>

```

Zdrojový kód 15 – Funkce smazání cíle

## 4.3 Databáze

Pro potřebu uchovávání dat o přihlašovacích údajích a činnostech uživatele byla pomocí rozhraní phpMyAdmin vytvořena databáze *FitterHappier* obsahující následující tři databázové tabulky, do kterých jsou potřebné vstupní údaje od uživatelů ukládány, a ke kterým může být podle potřeby přistupováno.

### 4.3.1 Tabulka Users

Tabulka *Users* (Tabulka 1) uchovává data o registrovaných uživatelských účtech. Jedinečným identifikátorem (primárním klíčem) účtu je e-mail, jelikož není možné vytvářet více účtů se stejnou e-mailovou adresou. Dále tabulka obsahuje přezdívku uživatele a zahashovanou podobu jeho hesla. Všechny tři hodnoty jsou ukládány v datovém typu VARCHAR, který se používá pro textové řetězce s variabilní délkou. Poslední ukládaná hodnota je pouze informační, jde o datum a čas vytvoření uživatelského účtu s datovým typem DATETIME.

Název	Datový typ	Popis
email	varchar	E-mail uživatele (primární klíč)
name	varchar	Přezdívka uživatele
password	varchar	Heslo uživatele v hashované podobě
created_at	datetime	Datum a čas registrace uživatele

Tabulka 1 – databázová tabulka Users

### 4.3.2 Tabulka Goals

Tabulka *Goals* (Tabulka 2) obsahuje cíle vytvořené uživateli. Každý cíl má svůj jedinečný identifikátor, což je automaticky inkrementované číslo datového typu INTEGER, které se při každém dalším vytvoření nového cíle zvýší o 1. Tabulka dále obsahuje název cíle, zamýšlené datum splnění (datový typ DATE), prioritu cíle a e-mail uživatele, který cíl vytvořil. E-mail zde hraje roli cizího klíče párujícího existující uživatele (z tabulky *Users*) s příslušnými řádky jimi vytvořených cílů.

Název	Datový typ	Popis
goal_id	integer	Identifikace cíle (primární klíč)
goal	varchar	Název cíle
due_date	date	Požadované datum splnění cíle
priority	varchar	Priorita cíle (vybraná z možností)
email	varchar	E-mail uživatele, který cíl vytvořil (cizí klíč)

Tabulka 2 – databázová tabulka Goals

#### 4.3.3 Tabulka Exercises

Tabulka *Exercises* (Tabulka 3) obsahuje záznamy o cvičení vytvořených uživateli. Analogicky s tabulkou *Goals* obsahuje jedinečný identifikátor cviku, dále pak jeho název, dobu trvání, datum uskutečnění cvičení a e-mail. Stejně jako pro předchozí tabulku je e-mail cizím klíčem odkazující na příslušného autora cvičení z tabulky *Users*.

Název	Datový typ	Popis
exercise_id	integer	Identifikace cvičení (primární klíč)
exercise_type	varchar	Název cvičení
exercise_duration	integer	Doba trvání cvičení
exercise_date	date	Datum uskutečnění cvičení
email	varchar	E-mail uživatele, který cvičení přidal (cizí klíč)

Tabulka 3 – databázová tabulka Exercises

## 4.4 Výsledná podoba aplikace

V této kapitole je pro úplnost ukázky zobrazena finální podoba jádra aplikace, která pomocí výše představených technologií komunikuje s uživateli a přizpůsobuje své zobrazení jimi zadaným vstupům a vykonaným akcím. Přihlášený uživatel je v záhlaví přivítán a jsou mu zpřístupněny sekce pro přidávání cílů (Obrázek 11) a cvičení (Obrázek 12).

**Zakládání osobních cílů**

Název cíle

Požadované datum splnění  
dd.mm.rrrr

Priorita cíle  
Vyberte prioritu

**Vytvořit cíl**

**Vaše cíle**

Název cíle	Splnit do	Priorita	Akce
testovací cíl	2023-03-15	Vysoká	Smazat

Obrázek 11 – Sekce Cíle + header aplikace

**Počítadlo cvičení**

Typ cvičení:

Délka cvičení (minuty):

Den vykonání cvičení  
dd.mm.rrrr

**Přidat cvičení**

**Přehled cvičení**

Typ cvičení	Délka cvičení (minuty)	Datum	Akce
testovací cvik	15	2023-02-27	Smazat

Obrázek 12 – Sekce Cvičení

## 5 Zhodnocení výsledků

Na základě syntézy shromážděných poznatků byl v závěru teoretické práce vybrán skriptovací programovací jazyk PHP, jehož funkce byly demonstrovány na praktické webové ukázce s tematikou fitness. Výsledek srovnání byl ovlivněn výběrem zdrojů použitých pro charakteristiku jednotlivých jazyků, dále také soudobou využívaností a uživatelskou oblíbeností zkoumané skupiny jazyků, která vyplývá z použitých statistických indexů TIOBE, PYPL a W3Techs.

Výsledná aplikace *Fitter & Happier* plní spíše úlohu jednoduchého internetového záznamníčku a proti komerční konkurenci by ve své současné podobě pravděpodobně neobstála. Svůj účel ilustrovat využití skriptovacích jazyků pro webovou tvorbu ale splňuje dostatečně a poskytuje vhled nejen do fungování PHP skriptování ale i webové tvorby jako celku. Před potenciální nasazením aplikace do provozu a uvedením do konkurenčního prostředí by bylo nutné přidat celou řadu funkcí, které jsou u podobných aplikací ve sféře fitness dnes již standardem:

- Editace uživatelských účtů – V podobě prezentované v práci aplikace nenabízí možnost editace uživatelských účtů, proto by bylo nutné před nasazením aplikace tuto funkci implementovat. Uživatelé by mohli měnit při registraci zvolené uživatelské údaje či přiřadit fotografii ke svému účtu.
- Seznam přátel a zprávy – V návaznosti na předchozí návrh by bylo možné využít editovatelné účty pro základ komunikace mezi návštěvníky aplikace. Uživatelé by si mohli vyměňovat postřehy a tipy mezi sebou prostřednictvím soukromých zpráv či přidávat své oblíbence do seznamu přátel.
- Mobilní aplikace – Přejít na mobilní formu aplikace by umožnil snadnější zadávání dat do aplikace z mobilního telefonu přímo po proběhnutém cvičení, bez nutnosti přistupovat do prohlížeče.
- Spojení se sociálními sítěmi – Možností sdílet své cíle a výkony na sociálních sítích (např. Twitter) by se aplikace dostala do povědomí více lidí, což by potenciálně navýšilo počet uživatelů.
- Integrace s chytrými hodinkami – Propojení se softwarem chytrých hodinek by snížilo nutnost ručně zadávat údaje o cvičení do aplikace. Zároveň by si

mohl uživatel udržovat statistiky kroků, aktivity a tepové frekvence na jednom místě spolu s dalšími podklady pro sledování své fyzické zdatnosti.

- Záznamy o dietách – K plnohodnotnému cvičení samozřejmě patří i vyvážený a hodnotný jídelníček, proto by vylepšená podoba aplikace měla tuto funkci pro své uživatele zpřístupnit.
- Výraznější grafické zpracování – Současná podoba aplikace má spíše úsporný a jednoduchý design bez výrazných prvků. Pro větší upoutání pozornosti potenciálního uživatele by design mohl být více barevný a dynamický (čtetnějším použitím elementu jazyka JavaScript), aby byla jasně zdůrazněna jeho tematika. Dále by bylo vhodné navrhnout a graficky zpracovat logo společnosti, které by tvořilo ústřední motiv v hlavičce stránky.

Pro vývoj rozšiřujících funkcí aplikace by bylo rozumné použít některý z dostupných PHP frameworků (např. Laravel), což by značně snížilo čas a práci nutnou pro vývoj a implementaci diskutovaných funkcí. Frameworky nabízí velké množství již hotových návrhů a není tedy nutné vždy začínat s vývojem „od nuly“.

V poslední řadě je nutné zmínit internetovou reklamu, která by se zobrazovala uživatelům stránky a při dostatečné návštěvnosti by poskytovala zdroj peněz pro autora a jeho týmu. V případě rozšíření mezi veřejnost by mohlo být registrovaným uživatelům nabídnuto placené členství, které by nabízelo sestavení cvičebních plánů, diet a jídelníčků najatými profesionály z oboru.

## 6 Závěr

Předmětem této bakalářské práce byla problematika tvorby dynamických webových stránek s využitím skriptovacích programovacích jazyků. Pro uvedení do tématu webového obsahu byl představen distribuční informační systém Word Wide Web zajišťující vytváření, prohlížení a sdílení webových stránek. Byl přiblížen princip fungování architektury *Klient – Server*, který v systému zprostředkovává interakci mezi uživatelem a webovou stránkou. Byly vytyčeny tři dílčí složky systému: adresní systém URL, transportní protokol HTTP a značkovací jazyk HTML. Právě HTML a návazné stylovací technologii CSS byla v práci věnována zvýšená pozornost z důvodu využití pro front-end část praktické ukázky.

Po vytyčení základních pojmů následovala samotná charakteristika pěti zkoumaných skriptovacích jazyků, které byly vybrány na základě dat ze statistických indexů W3Techs, TIOBE a PYPL. Tím bylo zajištěno, že jazyky dále analyzované v práci jsou hojně využívané, oblíbené a relevantní technologie tvorby dynamických webových stránek. Samotná charakteristika zahrnovala seznámení s daným jazykem, výčet vlastností, stručnou historii a popis vývoje funkcí, názornou ukázkou zdrojového kódu a výpis kladů a záporů využití dané technologie v praxi webového designu.

Na závěr teoretické části práce byly popsány postupy a zásady tvorby webových stránek. Na základě syntézy shromážděných poznatků byl doporučen nejvhodnější jazyk pro praktickou ukázkou, v této práci byl zvolen skriptovací jazyk PHP (*Hypertext Preprocessor*). Důvodem výběru byla jeho strmá křivka učení, jednoduchá syntaxe, aktivní komunita a hojné použití mezi současnými webovými stránkami a službami. Nakonec byl charakterizován potřebný software, který byl následně použit při vývoji výsledné dynamické webové aplikace.

Splnění dílčích cílů v teoretické části vedlo k úspěšnému uskutečnění ukázky funkcionalit jazyka PHP a dalších technologií pro tvorbu webu, realizované v praktické části práce. Ta se skládala ze tří částí: front-end, back-end a databáze. Front-end aplikace byl tvořen značkovacím jazykem HTML, který určil rozložení a význam jednotlivých prvků na stránce, a CSS/JavaScript frameworkem Bootstrap, jenž zajistil responzivní, interaktivní a přívětivý design aplikace. Před přistoupením k samotné tvorbě vizuálů byl vytvořen drátěný model (Wireframe) sloužící jako opora při vývoji stránky. Okomentovaný HTML kód byl pro názornost doplněn obrázkem výsledného designu aplikace.



V back-end části praktické ukázky byly původní HTML dokumenty rozšířeny o vnořené skripty jazyka PHP, které z původně statické stránky vytvořily výslednou dynamickou webovou aplikaci. Skripty zajistily výměnu informací mezi uživateli a serverem, registraci a přihlašování do aplikace, ovládání prvků uvnitř aplikace a komunikaci a přenos dat do databázového systému. Této problematice byla následně věnována samostatná kapitola obsahující řešení ukládání dat do příslušných databázových tabulek prostřednictvím rozhraní phpMyAdmin a dotazovacího jazyka SQL.

Výsledná webová aplikace *Fitter & Happier* nabídla uživatelům možnost se zaregistrovat, přihlašovat se zadanými údaji, zakládat osobní milníky a vést si statistiku o proběhlých cvičeních. Při její tvorbě byl kladen důraz na jednoduchost a přehlednost, což odpovídá výsledné podobě i repertoáru dostupných funkcí. Hlavní cíl práce, demonstrovat praktické využití jazyka PHP, ale aplikace splnila a nabídla několik podrobně vysvětlených možností využití technologie při tvorbě webových aplikací. Možná vylepšení a přiblížení komerční konkurenci byla následně představena v kapitole Zhodnocení výsledků, která může být použita jako inspirace pro budoucí projekty.

Tento projekt může sloužit jako základ pro tvorbu komplexnějších webových aplikací. Navrhované funkce mohou být libovolně přidávány a implementovány pro přiblížení aplikace komerčním variantám, které používají miliony lidí po celém světě. Nabyté vědomosti a zkušenosti při programování a vývoj mohou být užitečné v mnoha zaměstnáních v oboru IT: od UX designéra, přes front-end developera, po webového programátora. Práce jako celek může být použita jako učební informační zdroj, jelikož obsahuje ucelený náhled na problematiku serverového skriptování, případně jako opora začínajícím programátorům ve výběru skriptovací technologie.

Zdrojový kód webové aplikace je umístěn na příloženém CD. Pro zobrazení dílčích souborů je nutné mít vhodné vývojové prostředí. V práci je využíváno studentské verze editoru PHP Storm, nicméně je možné využít i bezplatné alternativy, například Atom nebo Visual Studio Code. Pro spuštění webové aplikace je nutné mít nainstalovaný softwarový balíček XAMPP, na jehož základu aplikace funguje. Přihlašovací údaje jsou obsaženy na příloženém CD.

## 7 Seznam použitých zdrojů

- [1] MUSCIANO, Ch., KENNEDY B. *HTML a XHTML: kompletní průvodce*. Praha: Computer Press, 2000. ISBN 9788072264070.
- [2] POKORNÝ, M. *PHP nejen pro začátečníky*. Kralice na Hané: Computer Media, 2005. ISBN 8086686388.
- [3] BERNERS-LEE, T. a kol. The World-Wide Web. *Communications of the ACM* [online]. 1994, vol 37 issue 8, s. 76-82 [cit. 2022-12-05]. ISSN 0001-0782.  
Dostupné z: <https://dl.acm.org/doi/10.1145/179606.179671>
- [4] UDN Web Docs. *Learn Web Development: Sending form data* [online obrázek]. 2020 [cit. 2022-12-06]. Dostupné z: [https://udn.realityripple.com/docs/Learn/Forms/Sending\\_and\\_retrieving\\_form\\_data](https://udn.realityripple.com/docs/Learn/Forms/Sending_and_retrieving_form_data)
- [5] PEHLIVANIAN, A., NGUYEN, D. *JavaScript okamžitě*. 2. vydání. Brno: Computer Press, 2021. ISBN 9788025150252.
- [6] AlegsOnline. *Klient-Server* [online]. 20. 6. 2021 [cit. 2022-12-08]. Dostupné z: <https://cs.alegsaonline.com/art/20972>
- [7] Internet Assigned Number Authority. *Hypertext Transfer Protocol (HTTP) Status Code Registry* [online]. 8. 6. 2021 [cit. 2022-12-08]. Dostupné z: <https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>
- [8] MDN Web Docs. *What is a URL?*. [online]. 23. 1. 2023 [cit. 2022-12-10].  
Dostupné z: [https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_URL](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL)
- [9] W3 Schools. *HTTP Request Methods* [online]. 2022 [cit. 2022-12-10]. Dostupné z: [https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp)
- [10] MDN Web Docs. *An overview of HTTP* [online]. 2022 [cit. 2022-02-02].  
Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- [11] GAUNT M., PALMER Ch. Web.Dev. *Enabling HTTPS on your servers* [online]. 27. 3. 2015 [cit. 2022-12-10]. Dostupné z: <https://web.dev/enable-https/>
- [12] W3 Docs. *HTML 5 Page Structure* [online]. [cit. 2022-12-12]. Dostupné z: <https://www.w3docs.com/snippets/html/html5-page-structure.html>

- [13] KRUG, S. *Web design – nenuťte uživatele přemýšlet. 2.*, aktualiz. vyd. Brno: Computer Press, 2006. ISBN 8025112918.
- [14] ALAWAR, M., ABU-NASER, S. *CSS-Tutor: An intelligent tutoring system for CSS and HTML*. [online]. 2017, vol. 2 issue 1, s. 94-99 [cit. 2023-12-12]. ISSN 2455-4197. Dostupné z: <http://dstore.alazhar.edu.ps/xmlui/handle/123456789/216>
- [15] FreeCodeCamp. *Interpreted vs Compiled Programming Languages: What's the Difference?* [online]. 1. 10. 2020 [cit. 2022-12-14]. Dostupné z: <https://www.freecodecamp.org/news/compiled-versus-interpreted-languages/>
- [16] W3 Techs. *Usage statistics of server-side programming languages for websites* [online]. 14. 12. 2022 [cit. 2022-12-14]. Dostupné z: [https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language)
- [17] TIOBE. *TIOBE Index for December 2022* [online]. 1.12.2022 [cit. 2022-12-14]. Dostupné z: <https://www.tiobe.com/tiobe-index/>
- [18] CARBONELLE, P. *PYPL. PYPL Popularity of Programming Language* [online]. 14. 12. 2022 [cit. 2022-12-14]. Dostupné z: <https://pypl.github.io/PYPL.html>
- [19] GALLOWAY, J. *Professional ASP.NET MVC 4*. Hoboken, N.J.: Wiley, 2012. ISBN 9781118348468.
- [20] LOCK, A. *ASP.NET Core in action*. Second edition. Shelter Island, NY: Manning, 2021. ISBN 9781617298301.
- [21] Redwerk. *ASP.NET Core Advantages and Disadvantages* [online]. 1. 2. 2021 [cit. 2022-12-16]. Dostupné z: <https://redwerk.com/blog/asp-net-core-pros-and-cons/>
- [22] SYED, B. *Beginning Node.js*. New York, NY: Apress, 2014. ISBN 9781484201886.
- [23] W3 Schools. *CSS Syntax* [online obrázek]. 2020 [cit. 2023-01-05]. Dostupné z: [https://www.w3schools.com/css/css\\_syntax.asp](https://www.w3schools.com/css/css_syntax.asp)
- [24] Web Design. *Basic structure of an HTML document* [online obrázek]. 2015 [cit. 2023-01-05]. Dostupné z: <https://stuyhsdesign.wordpress.com/basic-html/structure-html-document/>

- [25] PASSIER, H., STUURMAN, S., POOTJES H. Beautiful JavaScript. In: *Proceedings of the Computer Science Education Research Conference* [online]. New York, NY, USA: ACM, 2014, 2014-11-05, s. 65-76 [cit. 2023-01-05]. ISBN 9781450333474. Dostupné z: <https://dl.acm.org/doi/10.1145/2691352.2691358>
- [26] WikiSofia. *PHP* [online]. 2020 [cit. 2023-01-07]. Dostupné z: <https://wikisofia.cz/wiki/PHP>
- [27] W3 Techs. *Usage statistics of content management systems* [online]. 7.1. 2023 [cit. 2023-01-07]. Dostupné z: [https://w3techs.com/technologies/overview/content\\_management](https://w3techs.com/technologies/overview/content_management)
- [28] W3 Techs. *Usage statistics of client-side programming languages for websites* [online]. 7. 1. 2023 [cit. 2023-01-07]. Dostupné z: [https://w3techs.com/technologies/overview/client\\_side\\_language](https://w3techs.com/technologies/overview/client_side_language)
- [29] LUTZ, M. *Programming Python*. 2 vydání. Sebastopol, CA: O'Reilly, 2001. ISBN 9780596000851.
- [30] LUTZ, M., ASCHER, D. *Naučte se Python: Pohotová příručka*. Praha: Grada. 2003. ISBN 802470367x.
- [31] Python Institute. *Python® – the language of today and tomorrow* [online]. [cit. 2023-01-10]. Dostupné z: <https://pythoninstitute.org/about-python>
- [32] Ruby Community. *About Ruby* [online]. 2018 [cit. 2023-01-16]. Dostupné z: <https://www.ruby-lang.org/en/about/>
- [33] FLANAGAN, D., MATSUMOTO, Y. *The Ruby Programming Language: Everything You Need to Know*. Sebastopol, California: O'Reilly, 2008. ISBN 9780596516178.
- [34] Code Academy. *What is Ruby on Rails?* [online]. 2021 [cit. 2023-01-16]. Dostupné z: <https://www.codecademy.com/resources/blog/what-is-ruby-on-rails/>
- [35] KOŘDOUSKOVÁ, B. Rascasone. *Web Design pro začátečníky: Tipy, pravidla a trendy 2022*. [online]. 8. 2. 2022 [cit. 2023-02-02]. Dostupné z: <https://www.rascasone.com/cs/blog/webdesign-tipy-pravidla-trendy>
- [36] JAHODA, B. Je Čas. *K čemu slouží CDN?* [online]. 8. 2. 2016 [cit. 2023-02-02]. Dostupné z: <https://jecas.cz/cdn>

[37] Apache Friends. *About XAMPP* [online]. 2022 [cit. 2023-02-05]. Dostupné z:  
<https://www.apachefriends.org/about.html>

[38] PhpMyAdmin. *About phpMyAdmin* [online]. 2022 [cit. 2023-02-06]. Dostupné z:  
<https://www.phpmyadmin.net/>

[39] JetBrains. *PHP Storm: The Lightning-Smart PHP IDE* [online]. 2023 [cit. 2023-02-07]. Dostupné z: <https://www.jetbrains.com/phpstorm/>

[40] Pexels. *License* [online]. 2023 [cit. 2023-02-23]. Dostupné z:  
<https://www.pexels.com/license/>

## 8 Seznam obrázků, tabulek a zkratk

### 8.1 Seznam obrázků

Obrázek 1 – Princip architektury Klient – Server [4] .....	15
Obrázek 2 – Příklad adresy URL [8] .....	17
Obrázek 3 – Ukázka struktury HTML dokumentu [23].....	19
Obrázek 4 – Syntax jazyka CSS [24].....	20
Obrázek 5 – Ukázka UI aplikace phpMyAdmin [38] .....	41
Obrázek 6 – Ukázka UI textového editoru PHP Storm [39] .....	42
Obrázek 7 – Wireframe Domovské stránky ukázkového webu .....	44
Obrázek 8 – Domovská stránka aplikace .....	50
Obrázek 9 – Stránka „O nás“ .....	51
Obrázek 10 – Registrační formulář .....	52
Obrázek 11 – Sekce Cíle + header aplikace .....	61
Obrázek 12 – Sekce Cvičení.....	61

### 8.2 Seznam výpisů zdrojového kódu

Zdrojový kód 1 – Ukázka programu „Hello World“ v jazyce C# .....	23
Zdrojový kód 2 – Ukázka jazyka Javascript vnořeného do HTML.....	26
Zdrojový kód 3 – Ukázka skriptu v jazyce PHP vnořeného do dokumentu HTML .....	28
Zdrojový kód 4 – Ukázka jazyka Python – program „Hello World“ .....	31
Zdrojový kód 5 – Ukázka jednoduché syntaxe jazyka Ruby – program „Hello World“ .....	34
Zdrojový kód 6 – Ukázka propojení dokumentu HTML s CDN frameworku Bootstrap....	40
Zdrojový kód 7 – Sdílená hlavička HTML dokumentů ve webové aplikaci .....	46
Zdrojový kód 8 – Navigační lišta s responzivním dropdown menu .....	48
Zdrojový kód 9 – Footer se dvěma sekcemi.....	49
Zdrojový kód 10 – Kód zajišťující registrace do aplikace .....	54
Zdrojový kód 11 – Princip přihlášení do aplikace .....	55
Zdrojový kód 12 – Kontrola přístupu a odhlášení uživatele .....	56
Zdrojový kód 13 – Funkce přidání nového cíle.....	57
Zdrojový kód 14 – Funkce zobrazení vytvořeného cíle v tabulce .....	58
Zdrojový kód 15 – Funkce smazání cíle .....	58

#### Zdroj výpisů: Vlastní tvorba

### 8.3 Seznam tabulek

Tabulka 1 – databázová tabulka Users .....	59
Tabulka 2 – databázová tabulka Goals.....	60
Tabulka 3 – databázová tabulka Exercises.....	60

## 8.4 Seznam použitých zkratk

Zkratka	Význam zkratky
WWW	World Wide Web
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
HTTP	HyperText Transfer Protocol
URL	Uniform Resource Locator
PHP	Hypertext PreProcessor (dříve Personal Home Page)
ASP.NET	Active Server Pages Network Enabled Technologies
SQL	Structured Query Language
GUI	Graphic User Interface
CDN	Content Delivery Network

## **Přílohy**

Příloha 1 – CD se zdrojovým kódem ukázkové aplikace