

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Porovnání CSS frameworků
pro tvorbu responzivních webových stránek
Bakalářská práce

Autor: Mikuláš Slavík
Studijní obor: Aplikovaná informatika (ai3-p)

Vedoucí práce: Mgr. Hana Rohrová

Hradec Králové

duben 2023

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 24.4.2023

Mikuláš Slavík

Poděkování:

Děkuji vedoucí bakalářské práce Mgr. Haně Rohrové za metodické vedení práce a trpělivé konzultace. Dále bych rád poděkoval svým rodičům za výraznou podporu studia vysoké školy a pýchu, mé přítelkyni Evě za trpělivost a dodávanou odvahu a lásku, švagru Ing. Jakubu Hořickému a strýci prof. Václavu Slavíkovi za inspirativně nastavené cíle nejen v oblasti vzdělávání.

Anotace

Tato práce seznamuje čtenáře s oblastí vývoje responzivních webových stránek a představuje dostupné nástroje pro usnadnění vývoje. Práce je zaměřena na CSS frameworky, jejich historii, použití, primární uživatele a porovnání těch nejznámějších. Na základě průzkumu byly zvoleny dva nejpopulárnější frameworky, Bootstrap a TailwindCSS, které jsou detailně porovnávány. Nahlíženo je na teoretické oblasti, jimiž jsou licencování, instalace, nabízené funkce, individualizace a dokumentace. Dále jsou v rámci práce vypracovány dvě webové stránky na obou porovnávaných frameworkcích, na kterých jsou zkoumány praktické aspekty, jako způsoby vývoje, jeho obtížnost a vliv na výsledný produkt.

Annotation

Title: Comparison of CSS frameworks for responsive web design

This thesis introduces the field of responsive web design and presents tools available to facilitate development. The thesis focuses on CSS frameworks, their history, usage, primary users, and a comparison of the most well-known ones. Based on a survey, the two most popular frameworks, Bootstrap and TailwindCSS, are selected and compared in detail. The theoretical aspects such as licensing, installation, offered features, individualization and documentation are reviewed. Further, the same websites are developed on both the compared frameworks and practical aspects such as development methods, its difficulty and its impacts on the final product are examined.

Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	Obecná teorie oblasti CSS frameworků	3
3.1	Vymezení oblasti a pojmů	3
3.1.1	Responzivní webové stránky a požadavky na ně.....	3
3.1.2	CSS framework	3
3.2	Historie a další vývoj frameworků	4
3.3	Primární důvody využití frameworků	5
3.4	Uživatelé a způsoby použití frameworků	6
3.5	Výhody a nevýhody frameworků	7
3.6	Obvyklé obsahové složení frameworků	9
4	Nejpopulárnější frameworky.....	11
4.1	Bootstrap	11
4.2	TailwindCSS.....	12
4.3	Materialize CSS.....	13
4.4	Foundation.....	13
5	Porovnání vybraných frameworků	14
5.1	Licencování	14
5.2	Možnosti instalace	14
5.2.1	CDN – Content Delivery Network.....	14
5.2.2	Lokální umístění souborů	15
5.2.3	NPM – Node Package Manager	15
5.2.4	PurgeCSS v nástroji Tailwind CLI.....	16
5.3	Teoretické porovnání	16

5.3.1	Vybrané způsoby instalace	16
5.3.2	Individualizace frameworku	17
5.4	Praktické porovnání.....	18
5.4.1	Navbar.....	19
5.4.2	Container	20
5.4.3	Flexbox layout.....	21
5.4.4	CSS Grid	23
5.4.5	Table	25
5.4.6	Accordion.....	26
5.5	Optimalizace.....	27
5.6	Dokumentace	29
5.7	Vliv použití na výsledný produkt	30
5.7.1	Velikost balíčku	30
5.7.2	Nástroj Google Lighthouse	32
6	Shrnutí výsledků.....	34
7	Závěry a doporučení	35
8	Seznam použité literatury.....	36
9	Přílohy	37

Seznam obrázků

Obrázek 1 - Grafy dat z dotazníků studie StateOfCSS.com.....	11
Obrázek 2 - Graf spokojenosti s aktuálním stavem CSS frameworků	11
Obrázek 3 – Navbar	19
Obrázek 4 – Porovnání rozdílu mezi container a fluid-container.....	21
Obrázek 5 – Flexbox layout.....	21
Obrázek 6 – CSS Grid	23
Obrázek 7 – Table	25
Obrázek 8 – Accordion.....	26
Obrázek 9 – Dokumentace frameworku Bootstrap.....	29
Obrázek 10 – Dokumentace frameworku TailwindCSS.....	30
Obrázek 11 – Velikost výsledných balíčků frameworků ve Windows 11	31
Obrázek 12 – Velikosti souborů frameworku Bootstrap.....	31
Obrázek 13 – Velikost souboru frameworku TailwindCSS.....	31
Obrázek 14 – Výsledek nástroje Google Lighthouse frameworku Bootstrap.....	32
Obrázek 15 – Faktory hodnocení Google Lighthouse frameworku Bootstrap	33
Obrázek 16 – Výsledek měření Google Lighthouse frameworku TailwindCSS	33

1 Úvod

Vývoj webových stránek a aplikací se za dobu existence internetu dočkal série výrazných obměn. Při vývoji jsou používány stejné, či obdobné technologie, které zde byly již na počátku vývoje webových stránek. Tyto technologie jsou ale postupem času rozšiřovány o nové funkce, případně jsou upravovány tak, aby vždy vyhovovali aktuálním trendům a standardům. Vytváření webových stránek je nyní díky různým postupům a nástrojům výrazně jednodušší a efektivnější. Nástroje také rozšiřují dostupnost vývoje i pro zájemce s menšími, nebo dokonce minimálními zkušenostmi v této oblasti. Webové stránky lze vyvíjet několika způsoby a postupy.

Původním způsobem je vývoj tzv. na zelené louce. V tomto způsobu je vývojář nucen si v projektu tvořit vše zcela sám. V dnešní době, kdy je požadováno mít vše co možná nejkvalitnější v co nejkratším čase, se jedná o nejméně efektivní a nejvíce náročný způsob vývoje.

Dalším způsobem je zakoupení, případně stažení, již hotové webové šablony, kterou uživatel pouze naplní vlastním obsahem, případně ji upraví dle své vlastní potřeby. Vzhledem k tomu, že tento způsob vyžaduje minimální znalosti, je velmi oblíbený.

Jedním z posledních způsobů je využití některého z dostupných online nástrojů, které fungují na principu interaktivní webové šablony. Uživatel jednoduše v počítačové nebo webové aplikaci, za použití systému drag-and-drop, sestavuje stránky z předpřipravených komponent, které naplní vlastním obsahem. Pro obsluhu takového nástroje je požadovaná znalost vývoje nulová.

Nejméně omezujícím, ale velmi používaným a efektivním způsobem vývoje je využití dostupných frameworků, které nabízejí velké množství předpřipravených komponent a funkcí, které může vývojář použít a upravit si je podle požadavků a případného grafického návrhu webových stránek.

2 Cíl práce

Cílem práce je seznámení s oblastí vývoje responzivních webových stránek a představení dostupných trendy nástrojů pro jeho usnadnění. Práce se zaměřuje na CSS frameworky, objasňuje jejich účel a využití. V rámci seznámení jsou představeny nejpoužívanější frameworky, jejich historie, aktuální vývoj, výhody a nevýhody, doplňující služby, použití a primární uživatelé.

Na základě dotazníkového průzkumu budou vybrány a detailně porovnány dva nejpoblárnější frameworky, pomocí kterých budou vyvinuty dvě jednostránkové webové stránky, které poslouží k porovnání praktických aspektů, zahrnující rozdíl ve vývoji, a jak jednotlivé frameworky uspokojují klíčové požadavky pro vývoj, jako je flexibilita a použitelnost. Dále budou porovnávány rozdíly v licencování, instalaci, dokumentaci, doplňujících funkcích a vlivu na výsledný produkt.

3 Obecná teorie oblasti CSS frameworků

Pojem CSS (Cascading Style Sheets) framework se řadí do oblasti front-end vývoje webových stránek. Touto oblastí se rozumí přední část webových stránek, která se zabývá především základním rozložením webu, jeho grafickou podobou, responzivitou, případně dalšími funkcemi, které nejsou závislé na komunikaci klienta se serverem, nebo klienta s klientem.

3.1 Vymezení oblasti a pojmů

Tato kapitola vymezuje klíčové oblasti a pojmy práce, které jsou nezbytné pro pochopení náplně a dané problematiky.

3.1.1 Responzivní webové stránky a požadavky na ně

Téma responzivních webových stránek se stává stále důležitějším, protože v dnešní době je používání různých zařízení a velikostí obrazovek k procházení internetu stále běžnější. Tuto skutečnost je nutné zohlednit při tvorbě webových stránek, aby byly použitelné a funkční na jakémkoliv zařízení, ať už se jedná o stolní počítač, tablet nebo chytrý telefon. Responzivní webové stránky musí být schopné automaticky přizpůsobit své rozložení, prostředí a obsah tak, aby byly čitelné a použitelné na jakémkoliv zařízení.

K dosažení této funkčnosti je klíčové využít flexibilního designu, který se automaticky přizpůsobí různým velikostem obrazovky. Tyto flexibilní weby také musí být schopny zobrazit různé typy médií, jako jsou obrázky, ale také by měly umožňovat snadnou navigaci. Tyto požadavky na responzivní webové stránky vyžadují využití specializovaných nástrojů a technik, jako jsou CSS frameworky.

3.1.2 CSS framework

CSS framework je knihovna obsahující předpřipravený soubor stylů, skriptů a komponent, které jsou připravené pro okamžité použití při vývoji webových stránek a webových aplikací jakékoliv komplexnosti anebo využití.

Webové stránky jsou tvořeny pomocí kódovacího jazyka HTML – HyperText Markup Language, ve kterém je tvořena základní struktura webových stránek a rozložení

prvků včetně obsahu. Tato základní kostra může být následně vzhledově vylepšována a zdokonalována stylovacím jazykem CSS – Cascading Style Sheets. Pomocí stylovacího jazyku je také možné kompletně přestavět pozice prvků, jejich velikost i obsah. Základní rozložení prvků vytvořené v HTML tak není zásadní a vzhledem k maximální responzivitě ani konečné.

Pro ještě svobodnější tvorbu webových stránek je možné při vývoji použít JS – JavaScript, který výrazně rozšiřuje možnosti stránek o dynamické funkce, které nedokáží zajistit jazyky HTML a CSS.

3.2 Historie a další vývoj frameworků

První CSS frameworky se začaly objevovat okolo roku 2005, přičemž jejich primárním cílem bylo urychlit a usnadnit proces vývoje webových stránek. Mezi první průkopníky ve vývoji webových stránek patří například Blueprint a YUI Grids, které nabízely základní nástroje pro stylování, a hlavně pro tvorbu základního rozložení a pozicování komponent ve smysluplném, snadno udržitelném prostředí, tzv. LAYOUT, pomocí funkce GRID system. Z počátku nebylo toto prostředí dokonalé a mělo samozřejmě i své nedostatky, ale i přesto to byl, na tu dobu, průlomový nástroj. Nyní je GRID základní funkcí všech dostupných frameworků a je na tolik vylepšený a spolehlivý, že je velmi snadné vytvořit responzivní webové stránky pro zařízení všech různých rozměrů a typů bez ohledu na poměr stran nebo velikost obrazovky.

Mezi další historické frameworky je možné zařadit také 960 Grid a YAML framework. [1] Zmíněné frameworky jsou i v roce 2023 stále dostupné a je možné je při vývoji i nadále používat. Jejich použití již však není pochopitelně doporučeno, protože příchod nových, mnohem komplexnějších frameworků v následných letech, vývoj těchto původních velmi ovlivnil. V některých případech byl zpomalen, případně úplně zastaven. Jen malé pozastavení způsobilo rychlé zastarání, jelikož nejen že vývoj jazyků HTML a CSS jde stále kupředu, ale velmi rychle se mění také vývojářské trendy.

Frameworky jsou stejně jako technologie HTML a CSS i nadále vyvíjeny a rozšiřovány o další funkce. V průběhu let se tento vývoj soustředil především na následující oblasti.

- **Responzivní design** – frameworky nabízejí komponenty, které se automaticky přizpůsobují velikosti obrazovky, což vývojářům umožňuje snadno vytvářet webové stránky s dobrou uživatelskou přívětivostí.
- **Kompatibilita** – frameworky se snaží zajistit kompatibilitu s co možná největším počtem prohlížečů, což vývojářům umožňuje věnovat méně času testování a optimalizaci pro různé prohlížeče.
- **Funkce** – frameworky poskytují širokou škálu funkcí, jako jsou například navigační lišty, formuláře, tlačítka, modální okna a mnoho dalšího, což vývojářům umožňuje využití hotové komponenty, aniž by je museli sami vyvíjet.
- **Design** – vybrané frameworky nabízejí také širokou škálu designových možností, od jednoduchých a minimalistických po sofistikované a komplexní, což vývojářům umožňuje vybrat si styl, který nejlépe vyhovuje jejich potřebám a požadavkům.

3.3 Primární důvody využití frameworků

Obecně CSS frameworky výrazně usnadňují práci webovým vývojářům tím, že nemusí vytvářet každý projekt zcela od nuly, ale mají dostupné nástroje, které umožňují v relativně krátkém čase vytvořit, minimálně, základní uživatelské rozhraní, které až na výjimky vyhovuje aktuálním standardům, kterých je doporučeno se při vývoji držet. Tyto nástroje tak neusnadňují pouze práci vývojářů, ale jsou také uživatelsky přívětivé pro návštěvníky, protože díky nim jsou na stránkách udržována obdobná rozložení prvků nebo obdobné funkce.

Frameworky zajišťují také základní vzhled, který většinou není příliš přívětivý a moderní, ale lze jej použít například pro nějaké interní systémy, nebo pro tvorbu kostry webových stránek – tzv. WIREFRAME. – pro takové použití je poskytován základní vzhled zcela dostačující. Tento základní vzhled je následně předěláván a vylepšován pomocí externích CSS stylů, které mohou být tvořeny na základě předpřipravených grafických návrhů, případně dle potřeb v průběhu vývoje.

Jedním z dalších důvodů vzniku CSS frameworků byla touha po ucelení vývoje webových stránek, o které se sice vývojáři HTML snažili, ale často se jednalo pouze o nepsaná pravidla, jak psát web. Ucelení vývoje bylo provedeno převážně v oblasti rozložení základních prvků, například pozice základní navigace (navigation bar) nebo patičky (footer) a dalších prvků, které mohou být na konkrétní stránce podstatné. Nejedná se pouze o prvky, ale také o přístup k tvorbě stránek, který naznačuje, že by měly být stránky tvořeny tak, aby se snadno přizpůsobovaly uživatelskému prostředí, tzv. responzivita. Responzivita byla historicky zajišťována pomocí CSS funkce flex, ale díky dalšímu vývoji začíná být nahrazována CSS funkcí grid. Do příchodu oficiální funkce grid bylo rozložení prvků zajišťováno právě funkcemi frameworku.

3.4 Uživatelé a způsoby použití frameworků

Používání CSS frameworků není nijak zásadně omezováno nebo ovlivňováno. Díky softwarové licenci MIT (podle Massachusettského technologického institutu, jenž tuto licenci vytvořil), pod kterou je většina takových frameworků publikována a distribuována, je možné s nimi nakládat dle uvážení vývojáře bez výrazného omezení. Jedinou podmínkou, kterou zmíněná licence nařizuje, je korektní uvedení původního autora a upozornění ohledně licencování. Tato uvedení jsou ve většině případů již automaticky vložena na začátku každého souboru ve frameworku, tedy jediná povinnost pro korektní licencování je do tohoto textu nijak nezasahovat, ani ho výrazně neupravovat, případně nemazat. Když je tato podmínka dodržena, vývojářům je umožněno s frameworkem jakkoliv nakládat. Vývojář tak může framework používat, upravovat, případně projekt stavěný na tomto frameworku distribuovat, a to za komerčním i nekomerčním účelem. [2]

Díky této licenci jsou frameworky používány jak na amatérských projektech, tak i na profesionálních stránkách velkých korporací. Pro příklad je možné zmínit, že jeden z nejpopulárnějších frameworků Bootstrap je využíván desktop webovou aplikací WhatsApp, kterou vlastní technologický konglomerát Meta Platforms, Inc. (dříve Facebook, Inc.), nebo webová aplikace Trello, která je jednou z nejznámějších aplikací pro správu projektů. [3]

Na nejpopulárnějších frameworkcích, které jsou podrobně rozebrány níže, jsou také vytvářeny webové šablony, které umožňují ještě snazší vytváření webových stránek. Obvykle stačí tyto šablony lehce upravit, naplnit obsahem a webové stránky jsou hotové a funkční. Tyto šablony je možné získat i používat zdarma, případně za zcela zanedbatelný poplatek od různých freelancerů nebo společností, které se vývoji webových šablon věnují. Šablony jsou prodávány na online tržištích, které jsou k tomuto prodeji určené. Mezi taková tržiště se řadí například stránka Colorlib nebo ThemeForest. Na těchto tržištích často dochází k tomu, že jsou nabízeny velmi podobné šablony, které mají sice různé barevné palety, zaoblení prvků a další detaily, ale i přesto si lze povšimnout znatelné podobnosti. To je způsobeno tím, že jsou tyto šablony tvořeny jedním subjektem, který vytvoří základní wireframe, na který následně nasazuje několik různých designů. Na internetu jsou již vyšší jednotky tisíc dostupných šablon pro všechny typy webových stránek, od administrátorských prostředí, přes webové prezentace firem, komplexní osobní portfolia, až po jednoduché stránky s galerií, např. pro svatební prezentaci. [4]

Lidé, kteří se vývoji webových stránek věnují pouze okrajově, nebo vůbec, mohou být díky šablonám, schopni vyprodukovat velmi dobře vypadající a fungující stránky, přičemž jim k tomu stačí minimální znalost CSS. Stačí se pouze držet přiložené dokumentace vybraného frameworku nebo šablony a vývojář nemusí mít ani velmi širokou znalost HTML. Velké korporace se k použití frameworku mohou rozhodnout v případě, že potřebují spolehlivě připravené, zpracované a responzivní prostředí, anebo v případě, kdy chtějí maximálně ušetřit i přesto, že potřebují co možná nejkvalitnější webové stránky. Aktuálně se ale jedná spíše o přežití, protože díky nově vyvíjeným funkcím, které jsou do CSS neustále implementovány, ztrácí použití frameworku pouze z důvodu gridu a responzivity smysl. Stále se ale jedná o rychlejší, jednodušší a spolehlivější řešení.

3.5 Výhody a nevýhody frameworků

Použití CSS frameworku prozatím stále převažuje výhodami, ale v dnešní době, kdy se velmi lpí na optimalizaci pro webové vyhledávače, uživatele datových připojení

i monitorů různých rozměrů, začíná být velikost souborů webových stránek a jejich zdrojů velmi sledovaným prvkem a je důležité na to při vývoji brát ohled.

Základní výhodou je výrazné zjednodušení práce při vývoji, a to nejen pro kodéra, ale může být také pro UX (User Experience – uživatelská přívětivost) a UI (User Interface – uživatelské rozhraní) designera. UX je ve frameworku již převážně připraveno díky předpřipraveným komponentám. Mezi tyto UX komponenty patří například formuláře, tlačítka, navigační lišta, různé karty a další vymoženosti, které jako celek tvoří důležitou část webových stránek. Webový vývojář zase může stránky skládat z již hotových prvků, které v případě potřeby graficky upravuje podle návrhů UI designera externími CSS styly, které přepisují nebo rozšiřují základní design, který je základní součástí frameworku. [5] Zmiňované zjednodušení a urychlení práce je ovšem velmi individuální věc. Může se pohybovat v rámci uspoření několika minut, ale také několika hodin, což se odvíjí hlavně od znalostí a zkušeností vývojáře s webovým vývojem a také použitým frameworkem. Tvorba webových stránek na míru a bez frameworku je každopádně časově náročná i pro zkušeného webového vývojáře a vyžaduje středně pokročilé znalosti HTML, CSS a případně JS.

Nevýhodou použití jakéhokoliv frameworku je navýšení počtu souborů, jejich velikostí, a především velikosti celého balíčku finálního produktu. Toto navýšení je u některých frameworků méně znatelné, ale u některých velmi výrazné, až kritické. Navýšení je způsobováno „hloupým“ fungováním množství frameworků, jenž jsou sice připraveny na všechna možná použití, obsahují mnohé funkce a skripty, ale taková připravenost si vyžaduje velké množství kódu, který balíček obsahuje. Tento přebytečný kód se označuje jako „CSS bloat“, tedy že jsou soubory zbytečně nafukovány. To poté způsobuje, že jsou jeho soubory velké, a to i v produkční verzi. Produkční verze je verze webové stránky, která je již nasazená a je používána uživatelem. Takové navýšení způsobuje nejen snížení skóre u vyhledávače Google, dle kterého vyhledávač řadí výsledky vyhledávání a web tak může být zcela potlačen, ale také zvyšuje zátěž pro server, a především pro klienta, kvůli skriptům, které fungují na pozadí i v případě, že nejsou využívány. Vyhledávač Google určuje relevantnost výsledků a webů pomocí různých algoritmů. Jeden z nich kontroluje

kvalitu obsahu a použitelnost webu, přičemž tento faktor je výrazně ovlivněn již při vývoji. [6]

K této nevýhodě je v rámci práce provedeno vlastní šetření, jehož cílem je zjistit, jak použití různých frameworků ovlivňuje velikost celého balíčku, ale také rychlost webových stránek a skóre, které Google stránkám udělí.

3.6 Obvyklé obsahové složení frameworků

Jelikož je každý CSS framework vyvíjen jiným subjektem, mají různé vlastnosti, funkce a doplňkové služby. I přesto je jejich základní složení obvykle shodné a udržuje se v mezích pomyslného standardu, který zajišťuje podobnost základní struktury. Tato struktura je následně každým subjektem doplňována o různé další služby, funkce a prvky. Díky ucelené základní struktuře není složité mezi těmito frameworky přecházet, aniž by se musel vývojář sžívat se zásadními změnami. Tato základní struktura se skládá z několika již zmíněných kategorií.

- **Reaktivita a responzivita** – kategorie obsahující především pomocné prvky, které výrazně usnadňují vývoj webových stránek z hlediska responzivity a přizpůsobení různým velikostem obrazovek.
- **Rozložení stránek** – kategorie poskytující třídy pro vytváření různých typů rozložení, včetně tříd pro sloupce, řádky, kontejnery apod.
- **Stylizace text a písma** – kategorie obsahující stylizaci textu a písma, včetně velikosti, zarovnání, označení, barvy apod., vhodná pro změnu vzhledu základních prvků, například nadpisů, paragrafu, číslovaných a nečíslovaných seznamů.
- **Formuláře** – kategorie obsahující zpracované formulářové komponenty, mezi které se řadí například textová pole, checkboxy, selektory, ale také různá tlačítka.

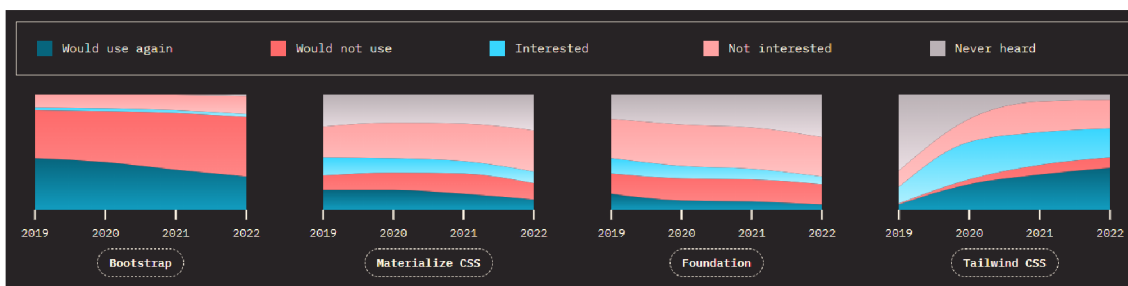
Frameworky jsou také často rozšiřovány o další prvky, které ke své správné funkčnosti potřebují využití příložených JavaScript souborů. Tyto prvky jsou následně využívány hlavně uživateli webových stránek, ale frameworky implementují také technologie, které jsou pouze pro vývojáře. Mezi tyto technologie se řadí například SASS – Syntactically Awesome StyleSheet. Tato technologie je

pomocné rozšíření pro CSS, konkrétně se jedná o pre-processor CSS. Tento pre-processor mění a zásadně usnadňuje způsob zapisování CSS a umožňuje tak používat funkce, které v obyčejném CSS originálně neexistují. Mezi tyto funkce se řadí například možnost využívání proměnných, vnořených pravidel, dědičnosti a dalších vychytávek, které mohou vývojáři ušetřit ještě více času a práce. Vývojář tak může využít tyto funkce a při každém uložení souboru je tento pre-processor transpiluje (zpracuje) a vytvoří z nich běžný CSS soubor s jeho běžnými funkcemi. Výsledný CSS soubor tak může být výrazně delší než původní SASS, což je ale v pořádku a je tak možné vidět, kolik práce bylo vývojáři ušetřeno. [7]

4 Nejpopulárnější frameworky

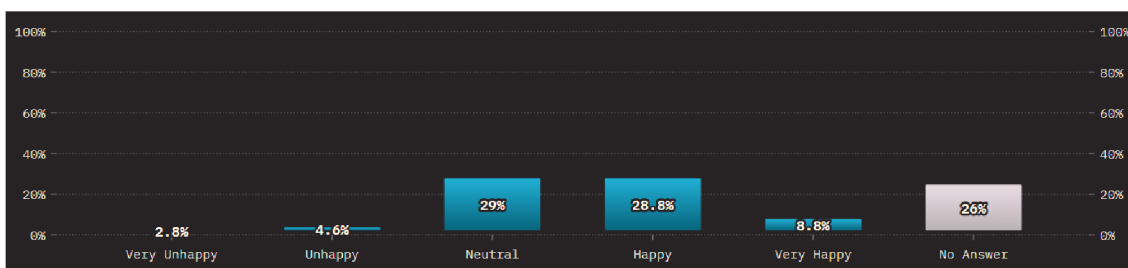
Od roku 2019 je kolektivem Devographics prováděn každoroční průzkum pomocí veřejné ankety, jehož cílem je sledovat popularitu CSS – jeho funkcí, jak se nadále rozvíjí, jak s ním jsou vývojáři spokojeni a mnoho dalších faktorů. Součástí ankety jsou také otázky na CSS frameworky. Dotázaní respondenti reagují na otázky, zda a případně jaký framework používají, zda jsou s ním spokojeni a zda by vyzkoušeli nějaký jiný framework, případně který. Vzhledem k tomu, že se dotazníku účastní široká veřejnost, jedná se o dostatečně velký vzorek. [8]

Jak lze vidět na přiloženém obrázku níže, dle studie se mezi ty nejpopulárnější frameworky řadí Bootstrap, Materialize CSS, Foundation a TailwindCSS.



Obrázek 1 - Grafy dat z dotazníků studie StateOfCSS.com

Zdroj: stateofcss.com [8]



Obrázek 2 - Graf spokojenosti s aktuálním stavem CSS frameworků

Zdroj: stateofcss.com [8]

4.1 Bootstrap

Framework Bootstrap byl oficiálně publikován v druhé polovině roku 2011. Za jeho vývojem stáli Jacob Thornton a Mark Otto, dva členové vývojářského týmu ve společnosti Twitter. Vývojářský tým měl původně za cíl vytvořit open-source framework obsahující předpřipravené komponenty, který měl fungovat jako nástroj

zaručující vývoj konzistentních webových stránek napříč celým internetem. Dle aktuálního výzkumu stránky W3Techs.com se tato vize zjevně vydařila, protože nyní 21.3 % všech stránek na světě framework Bootstrap využívá, [9] včetně velkých technologických jmen jako WhatsApp, Envato, Trello nebo FontAwesome. [3]

Jak uvádí studie, popularita tomuto frameworku za poslední roky výrazně klesá, a to pravděpodobně na úkor úspěchu jednoho z nejnovějších gigantů, frameworku TailwindCSS. Bootstrap se dlouhá letá držel na prvním místě žebříčku, ale výsledky dotazníku roku 2022 prokázali, že byl v popularitě jeho výrazným konkurentem překonán. [8]

4.2 TailwindCSS

Framework TailwindCSS, který je jedním z nejmladších frameworků, je aktuálně také nejpopulárnějším. Tento framework vytvořil Adam Wathan v roce 2017 společně s malým týmem několika vývojářů. Velkému úspěchu se tomuto frameworku dostalo na začátku roku 2020, kdy TailwindCSS v popularitě doslova vybuchl. Tento úspěch umožnil dosavadním vývojářům rozšířit své řady, a tak vznikla společnost Tailwind Labs, která stojí ze jeho dalším vývojem. Základním konceptem tohoto frameworku bylo vytvořit snadno použitelný a efektivní framework pro vývojáře, který usnadní psaní CSS kódu a urychlí proces vytváření webových stránek. Autor Wathan vytvořil TailwindCSS s cílem poskytnout vývojářům jednoduché a opakovatelné způsoby, jak vytvářet různé prvky na webu. Namísto psaní vlastního CSS kódu pro každý element umožňuje framework aplikovat styly pomocí definovaných tříd. Celkově se původní koncept opírá o myšlenku, že pokud jsou k dispozici standardní třídy pro designové prvky, může to vývojářům ušetřit spoustu času a usnadnit vývoj webových stránek. [10]

Od počátku svého vývoje se TailwindCSS vyvinul také do mnoha dalších projektů, jako jsou rozšíření pro vývojová prostředí (např. Visual Studio Code a Atom), pluginy pro React a Vue, knihovny komponent pro React a další.

Dle studie se jedná o nejrychleji rostoucí framework v popularitě, tedy minimálně v posledních letech. Na počátku roku 2019 bylo minimální procento vývojářů, kteří by TailwindCSS používalo, nýbrž už v tu dobu velké množství z nich o použití

uvažovalo. Tato čísla se rok od roku zvyšovala, až nakonec došlo k překonání nejpobulárnějšího frameworku na trhu – Bootstrap. Nyní existuje pouze minimální procento vývojářů, kteří by tento framework neznali, protože je nyní hojně využíváný a velké procento vývojářů je s ním spokojeno. [8]

4.3 Materialize CSS

Framework Materialize CSS vychází z konceptu Material Design navrženého společností Google a dle slov jeho vývojářů kombinuje klasické principy úspěšného designu společně s ověřenými inovacemi a technologiemi. V grafech studie si lze všimnout, že si tento framework udržoval do roku 2020 svoji stálou klientelu, tedy minimálně v číslech. Od toho roku mu ale zájemci klesají. [11]

4.4 Foundation

Framework Foundation je, ze čtyř nejpobulárnějších, právě čtvrtý, tedy poslední, nejméně používaný a nejméně oblíbený mezi vývojáři. Foundation framework byl publikován v roce 2011 a měl za úkol usnadnit vývojářům tvorbu responzivních webových stránek. Díky studii lze vidět postupný a pomalý pád v počtu zájemců, kteří by měli zájem tento framework využívat. [12]

5 Porovnání vybraných frameworků

Pro porovnání vybraných frameworků jsou v rámci práce vyvinuty dvě webové stránky dle stejného grafického návrhu. Při vývoji je zkoumáno, jak jednotlivé frameworky uspokojují klíčové požadavky jako jsou flexibilita, přizpůsobení různým velikostem obrazovky a podpora různých typů medií. Porovnání také zahrnuje praktické aspekty, jako je instalace, rychlost vývoje, snadnost použití, dokumentace, doplňující funkce a vliv na výsledný produkt.

K porovnání jsou vybrány dva aktuálně nepoužívanější frameworky, konkrétně Bootstrap (verze 5.3) a TailwindCSS (verze v3.3.1). Webové stránky jsou vytvořeny na základě grafického návrhu p. Tkáče, jehož návrh byl pro tuto práci vhodný a poskytnutý zdarma. Na frameworku Bootstrap jsou využity nabízené komponenty, které jsou u frameworku TailwindCSS vytvořeny systémem utility-first a jsou porovnávány rozdíly.

5.1 Licencování

Frameworky Bootstrap [5] i TailwindCSS [13] jsou distribuovány pod licencí MIT, což je tzv. open source licence, která vývojářům umožňuje volné užití a další distribuci kódu, ať pro komerční tak i nekomerční účely. Licence MIT je vytvořena Massachusettským technologickým institutem a je velmi krátká a jednoduchá. [2]

5.2 Možnosti instalace

Zvolené frameworky mají několik obdobných možností instalace, nebo způsobů, jak je možné je v projektu použít. Mezi ty prakticky shodné se řadí následující možnosti.

5.2.1 CDN – Content Delivery Network

Vložení absolutních odkazů na CDN server, ze kterého si webové stránky převezmou dané CSS a JS soubory. Toto připojení se provádí přidáním následujících řádků kódu do hlavičky HTML souboru:

```
<link rel="stylesheet" href="absolutniOdkazNaCssSoubor"/>  
<script src="absolutniOdkazNaJavaScriptSoubor "></script>
```

Výhoda použití CDN spočívá převážně v tom, že se soubor načítá z jiného serveru, než kde jsou hostované webové stránky, což může vést k lepšímu výkonu a větší stabilitě webového serveru. Server, který je zvolený jako CDN, má své soubory uložené v mezipaměti a je tak optimalizovaný pro rychlé poskytování souborů. Nespornou výhodou je také to, že je soubory možné snadno udržovat aktuální, jelikož jejich originální zdroj se nachází pouze na jednom místě na internetu.

5.2.2 Lokální umístění souborů

Stažení potřebných souborů přímo z webových stránek vybraného frameworku, vložení do souborů projektu a připojení do HTML relativním odkazem. Tyto odkazy jsou používány právě v případě, kdy je odkazováno na soubory, které jsou uloženy lokálně. Soubory lze do HTML souboru připojit pomocí následujících řádků kódu:

```
<link rel="stylesheet" href="relativniOdkazNaCssSoubor"/>  
<script src="relativniOdkazNaJavaScriptSoubor "></script>
```

Tento způsob je oproti CDN stabilnější, jelikož je minimalizováno riziko přerušení přenosu z jiného serveru, než na kterém jsou umístěné webové stránky. Vývojář má také větší kontrolu nad těmito soubory a může je dle potřeby upravovat, případně s nimi pracovat offline, bez přístupu k internetu.

5.2.3 NPM – Node Package Manager

NPM je nástrojem pro správu balíčků a knihoven. Přes tento nástroj je možné potřebné soubory stáhnout přímo do projektu a s nimi poté pracovat obdobně jako v předchozím kroku. Pro stažení je nutné použít konzoli v použitém vývojovém prostředí, do které je nutné vložit následující příkaz:

```
npm install nazevBalicku
```

5.2.4 PurgeCSS v nástroji Tailwind CLI

Framework TailwindCSS, oproti konkurenčnímu frameworku Bootstrap, nabízí vývojářům jednu velmi praktickou funkci, která umožňuje minimalizaci velikosti výsledného CSS souboru odstraněním nevyužitého CSS kódu ze všech dostupných tříd. Tato funkce je více popsána níže, nýbrž pro její funkčnost je při prvotní instalaci nutné nastavit nástroj PurgeCSS jako součást PostCSS pomocí nástroje Tailwind CLI (Command-Line Interface).

5.3 Teoretické porovnání

Teoretické porovnání zmiňuje přípravu prostředí, ve kterém je následně prováděno praktické porovnání – seznamuje s vybranými způsoby instalace a následujícím přizpůsobením vybraných frameworků.

5.3.1 Vybrané způsoby instalace

Pro vývoj webových stránek pomocí frameworku **Bootstrap** je zvolena možnost instalace pomocí lokálního umístění souborů, které jsou staženy přímo z webových stránek frameworku. Konkrétně se jedná o soubory *bootstrap.min.css* a *bootstrap.min.js*, což jsou minifikované soubory obsahující veškeré styly a skripty.

Pro vývoj webových stránek pomocí frameworku **TailwindCSS** je zvolena možnost instalace pomocí nástroje Tailwind CLI. Aby bylo možné použít tento postup instalace, je nutné mít v počítači nainstalovaný Node.js. Poté stačí ve vývojovém prostředí zadat do příkazové řádky následující příkazy, které stáhnou všechny potřebné soubory.

```
npm install -D tailwindcss
```

```
npx tailwindcss init
```

Poslední příkaz, který je vhodné do příkazové řádky zadat, umožní správnou funkci nástroje PurgeCSS v daném vývojovém prostředí.

```
npx tailwindcss -i ./src/input.css -o ./dist/output.css --watch
```

5.3.2 Individualizace frameworku

Protože **Bootstrap** obsahuje již předpřipravené komponenty, jeho individualizace je oproti konkurenčnímu frameworku mírně omezená. I přesto je ale možné pomocí něho vytvořit webové stránky přesně dle potřeby. Individualizace může probíhat dvěma způsoby. Jedním z nich je ruční přepisování dodaného CSS souboru. Tím druhým je vytvoření externího CSS souboru, který je nutné do HTML vložit až za ten dodaný, čímž je možné přepisovat jeho hodnoty. Do tohoto souboru je také vhodné vkládat vlastní kód pro další stylizace, případně vytváření dalších komponent. V tomto souboru je možné pro individualizaci používat proměnné, které vývojář nadefinuje na začátku a poté má možnost je používat v jakékoliv části souboru.

```
:root {
  --primary-color: #9DC08B; /* DEKLARACE PROMĚNNÉ */
}
section {
  color: var(--primary-color); /* POUŽITÍ PROMĚNNÉ */
}
```

TailwindCSS lze individualizovat pomocí konfiguračního souboru *tailwind.config*, který je automaticky vygenerován při instalaci a jeho obsah vypadá následovně:

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [],
  theme: {
    extend: {},
  }
}
```

Content specifikuje všechny soubory, které obsahují HTML nebo JS, ve kterých je vyžadováno použití tříd frameworku. Tyto soubory jsou následně analyzovány, a použité třídy jsou zahrnuty do generovaného CSS souboru.

```
content: ["/src/*.html,js"],
```

Theme vývojáři umožňuje vlastní nastavení hodnot pro barvy včetně různých odstínů, breakpoint hodnoty, velikosti písma nebo např. různá odsazení.

```
theme: {
  extend: {
    colors: {
      primary: '#9DC08B',
      secondary: '#40513B',
      white: {
        DEFAULT: 'rgb(255, 255, 255)',
        100: 'rgba(255, 255, 255, 0.1)',
        200: 'rgba(255, 255, 255, 0.2)',
        300: 'rgba(255, 255, 255, 0.3)',
        400: 'rgba(255, 255, 255, 0.4)',
        500: 'rgba(255, 255, 255, 0.5)',
        600: 'rgba(255, 255, 255, 0.6)',
        700: 'rgba(255, 255, 255, 0.7)',
        800: 'rgba(255, 255, 255, 0.8)',
        900: 'rgba(255, 255, 255, 0.9)'
      }
    },
    fontFamily: {
      'manrope': ['Manrope', 'sans-serif'],
      'saira': ['Saira', 'sans-serif']
    }
  }
}
```

Tyto hodnoty lze poté snadno aplikovat v kódu. Například pro obarvení textu bude automaticky vygenerována třída *text-white-200*, která textu nastaví bílou barvu s viditelností, neboli alfa kanálem, 0.2. Pro použití vlastního fontu je vygenerována třída *text-manrope*, která vybranému textu nastaví zvolený font.

5.4 Praktické porovnání

Praktické porovnání testuje vhodnost vybraných frameworků takovým způsobem, že je použito několik komponent, které jsou na moderních webových stránkách běžně využívány. V případě frameworku Bootstrap je využíváno co nejvíce předpřipravených komponent, které jsou doplněny o vlastní stylování, aby vyhovovali grafickému návrhu. TailwindCSS je testován tak, jak je navržen, tedy

použitím pouze atomických tříd utility-first způsobem s minimálním použitím vlastního stylování externím souborem.

Části kódu, které se opakují jsou pro přehlednost vynechány. Webové stránky jsou umístěné na webovém serveru a lze na ně přistoupit pomocí adresy <https://uhk.neoloop.cz/slavik>, kde se nachází rozcestník, ze kterého je možné zobrazit webové stránky postavené na obou frameworkcích.

5.4.1 Navbar

Navbar neboli navigace je nejdůležitějším prvkem webových stránek, kterou návštěvník využívá pro jejich procházení.



Obrázek 3 – Navbar

Zdroj: Vlastní zpracování

Bootstrap nabízí konkrétní komponentu, kterou je možné snadno modifikovat, např. lze před položky přidat logo, nebo je na pravou stranu navigace možné přidat tlačítko, případně jakýkoliv jiný prvek. Navigace je při použití tohoto frameworku díky JS knihovně velmi dobře responzivní.

```
<nav class="navbar navbar-expand-lg fixed-top">
  <div class="container">
    <button class="navbar-toggler" type="button">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarContent">
      <ul class="navbar-nav me-auto">
        <li class="nav-item"><a class="nav-link"
href="#home">Domů</a></li>
        <li class="nav-item"><a class="nav-link"
href="#about">O nás</a></li>
      </ul>
      <ul class="navbar-nav">
        <li class="nav-item nav-button"><a class="nav-link"
href="#contact">Kontaktujte nás</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Kód u frameworku **TailwindCSS** je kvůli utility-first způsobu vývoje výrazně delší. Výsledný produkt je však vzhledově prakticky totožný. Responzivita je v tomto případě ovšem nulová a pro její funkčnost je nutné vytvořit vlastní skript, který by tuto funkci zajišťoval.

```
<nav class="bg-white border-b-3 border-b-primary sticky top-0 z-50">
  <div class="max-w-screen-xl flex flex-wrap items-center justify-
between mx-auto py-3">
    <div class="hidden w-full md:block md:w-auto" id="navbar-
default">
      <ul class="font-medium flex flex-col p-4 md:p-0 mt-4
border border-gray-100 md:flex-row md:space-x-8 md:mt-0 md:border-0
md:bg-white">
        <li>
          <a href="#home" class="block py-1 pl-0 pr-6 text-
black-600 hover:text-black-800 transition">Domů</a>
        </li>
        <li>
          <a href="#about" class="block py-1 pl-0 pr-6 text-
black-600 hover:text-black-800 transition">O nás</a>
        </li>
      </ul>
    </div>
    <div class="hidden w-full md:block md:w-auto">
      <ul class="font-medium flex flex-col">
        <li>
          <a href="#contact" class="block py-3 px-8 text-
black-600 border border-black-400 rounded-2xl hover:text-black-800
hover:bg-black-100 transition">Kontaktujte nás</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

5.4.2 Container

Container je důležitý prvek, který ohraničuje část, nebo celou webovou stránku, a určuje šířku obsahu tak, aby byl vždy zobrazen konzistentně na různých zařízeních.

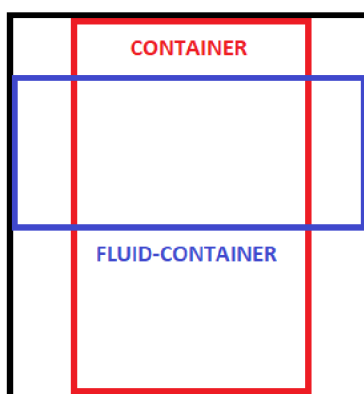
V **Bootstrapu** existují dva předdefinované typy:

- Container – je fixní šířky a vždy uprostřed webové stránky,
- Fluid-Container – není fixní šířky a přizpůsobuje se velikosti okna prohlížeče.

```
<div class="container">
  Obsah sekce nebo celé webové stránky
</div>
```

TailwindCSS tento prvek předdefinovaný nemá, a proto je více modulární. Pro vytvoření takového prvku se obdobně použije třída *container*, kterou je nutné doplnit třídou *mx-auto*, která zajistí to, že container bude vždy uprostřed. Pro nastavení fixní šířky je ještě nutné přidat třídu *max-w-screen-xl*, která nastaví, že container bude široký maximálně tak, jak definují písmena *xl*.

```
<div class="container mx-auto max-w-screen-xl">  
  Obsah sekce nebo celé webové stránky  
</div>
```



Obrázek 4 – Porovnání rozdílu mezi container a fluid-container

Zdroj: Vlastní zpracování

5.4.3 Flexbox layout

Sekce – Proč si vybrat právě nás? - je zaměřená na ověření funkce, která zajišťuje rozdělení tří položek vedle sebe, tzv. Flexbox layout. Tato funkce usnadňuje manipulace s umístěním prvků na stránce, což zajišťuje správnou responzivitu.

Proč si vybrat právě nás?

Naše jedinečné znalosti o lesích a práci s dřevem zajišťuje, že naše služby dosahují vysokých kvalit. Navíc jsme také specialisté v poradenství v oblasti životního prostředí a lesního hospodářství, což nám umožňuje pomoci zákazníkům s jakýmkoli otázkami a potížemi. Pokud hledáte spolehlivou firmu s vynikajícími znalostmi a zkušenostmi, není lepší volbou než naše firma.

<p>ČAS</p> <p>Příjezd zaměstnanců a techniky na místo již do druhého dne od rezervace termínu.</p>	<p>VYBAVENÍ</p> <p>Nadstandardní přístrojové i softwarové vybavení určené speciálně pro naše potřeby.</p>	<p>VSTŘÍCNOST</p> <p>Individuální řešení situací vycházející z 23leté praxe našich zaměstnanců.</p>
---	--	--

Obrázek 5 – Flexbox layout

Zdroj: Vlastní zpracování

Bootstrap využívá základní funkci *row*, ve které jsou položky rozděleny za pomoci sloupcového systému, konkrétně *col-4*. V kódu níže jsou mezi slovem *col* a číslicí vložena také dvě písmena, která signalizují, pro které breakpoint hodnoty má rozdělení prvků platit a poté se změnit.

```
<div class="row gx-5">
  <div class="col-lg-4">
    <div class="item">
      <div class="text">
        <h3>Čas</h3>
        <p>Příjezd zaměstnanců...</p>
      </div>
    </div>
  </div>
  <div class="col-lg-4">
    <div class="item">
      <div class="text">
        <h3>Vybavení</h3>
        <p>Nadstandardní přístrojové...</p>
      </div>
    </div>
  </div>
</div>
```

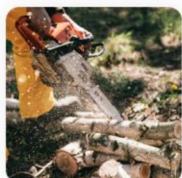
V případě **TailwindCSS** je použita funkce *flex*, která v podstatě odpovídá stejné funkci, která je použita u konkurenčního frameworku. Konkrétní rozdělení položek už ale není zajištěno sloupcovým systémem, nýbrž vlastností *flex-grow*, která zajišťuje, že prvek s třídou *flex-1* zabírá maximální dostupný prostor v rodičovském prvku. Tento konkrétní prvek bude tedy roztažen na celou dostupnou šířku. V případě, kdy je v rodičovském prvku více potomků, které mají tuto třídu, tak si dostupný prostor rozdělí spravedlivě.

```
<div class="flex gap-x-16">
  <div class="after-border relative flex-1 rounded-2xl px-6 py-8 bg-gradient-to-r from-primary to-secondary">
    <h3 class="uppercase font-bold text-lg mb-1 text-white font-saira">Čas</h3>
    <p class="text-white">Příjezd zaměstnanců...</p>
  </div>
  <div class="after-border relative flex-1 rounded-2xl px-6 py-8 bg-gradient-to-r from-primary to-secondary">
    <h3 class="uppercase font-bold text-lg mb-1 text-white font-saira">Vybavení</h3>
    <p class="text-white">Nadstandardní přístrojové...</p>
  </div>
</div>
```

5.4.4 CSS Grid

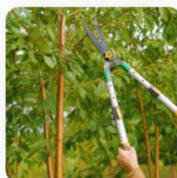
Sekce – Naše služby – je zaměřená na ověření funkce grid, která umožňuje vytvářet responzivní rozdělení položek do více řádků a sloupců, což umožňuje vytvářet komplexnější rozdělení.

Naše služby



LESNÍ HOSPODÁŘSTVÍ A ÚDRŽBA

Lesy jsou důležitou součástí našeho ekosystému a správná péče o ně je klíčová pro udržení biologické rozmanitosti a ochranu přírody. Naše služba má za cíl zachovat zdravý a silnou stavbu lesního ekosystému.



OŘEZ OVOCNÝCH STROMŮ

Pro krásné a plodné ovocné stromy je nejlepší pravidelný ořez. Ořez ovocných stromů pomáhá omezit počet malých, nevýhodných větví, a tím zvýšit počet a množství plodů. Náš zkušený tým nabízí profesionální ořez ovocných stromů, který je zaměřen na zvýšení kvality Vaší úrody.



VÝROBA SUBSTRÁTU

Výroba dřevěného substrátu je proces, při kterém se zpracovává dřevní materiál na malé kousky, které se používají jako součást směsi substrátu pro rostliny. Dřevěný substrát je ideální pro růst mnoha rostlin včetně okrasných rostlin, květin, zeleniny a ovocných stromů.



VÝKUP DŘEVA

Naše společnost se specializuje na výkup dřeva z celého území. Nabízíme rychlý a spolehlivý výkup dřeva od malých přírodních vlastníků i velkých lesních provozů. Výkup dřeva není jen pro nás způsob, jak získat kvalitní materiál pro naše služby, ale i pro životní prostředí a lesní hospodářství.

Obrázek 6 – CSS Grid

Zdroj: Vlastní zpracování

Bootstrap tuto funkci zatím vůbec neposkytuje a řeší to vlastním způsobem, tedy opět použitím obyčejného Flexbox layoutu. V tomto případě jsou prvky opět zabaleny do rodičovského prvku *row*, a každý prvek potomek má nastavený *col-6*, který mu nastavuje šířku na polovinu rodičovského prvku. Ve chvíli, kdy součet čísel, které jsou nastaveny v *col-x* přesáhne 12, což je maximální počet sloupců v sloupcovém systému frameworku Bootstrap, první takový prvek je přemístěn na další řádek.

```

<div class="row">
  <div class="col-xl-6">
    <div class="item">
      <div class="image d-none d-lg-block mb-5"></div>
      <div class="text">
        <h3>Lesní hospodářství a údržba</h3>
        <p>Lesy jsou důležitou součástí...</p>
      </div>
    </div>
  </div>
  <div class="col-xl-6">
    <div class="item">
      <div class="image d-none d-lg-block mb-5"></div>
      <div class="text">
        <h3>Ořez ovocných stromů</h3>
        <p>Pro krásné a plodné ovocné...</p>
      </div>
    </div>
  </div>
</div>

```

TailwindCSS díky své filozofii poskytuje všechny funkce, které ovládá samotné CSS, tudíž v něm lze snadno využít vlastnosti funkce GRID. V tomto konkrétním případě má rodičovský prvek třídu *grid* společně s dalšími třídami, které nastavují počet řádků a sloupců, tedy *grid-columns-2* a *grid-rows-2*. Každý přímý potomek v takovém rodičovském elementu se poté drží tohoto nastavení a opět přes funkci *flex-1* je nastaven stejnými pravidly jako v předchozím případě.

```

<div class="grid grid-cols-2 grid-rows-2 gap-5">
  <div class="flex-1">
    <div class="flex gap-x-5">
      <div class="w-4/12 hidden lg:block mb-5 w-52 h-52 rounded-2xl bg-center bg-no-repeat bg-cover"></div>
      <div class="w-8/12">
        <h3 class="uppercase font-bold text-lg mb-1 font-saira">Lesní hospodářství a údržba</h3>
        <p>Lesy jsou důležitou...</p>
      </div>
    </div>
  </div>
  <div class="flex-1">
    <div class="flex gap-x-5">
      <div class="w-4/12 hidden lg:block mb-5 w-52 h-52 bg-red-400 rounded-2xl bg-center bg-no-repeat bg-cover"></div>
      <div class="w-8/12">
        <h3 class="uppercase font-bold text-lg mb-1 font-saira">Ořez ovocných stromů</h3>
        <p>Pro krásné a plodné ovocné...</p>
      </div>
    </div>
  </div>
</div>

```

5.4.5 Table

Secke – Ceník – je obyčejná HTML tabulka, která reprezentuje především to, jak může být vlastní stylování obyčejného prvku ve frameworku TailwindCSS komplikované a zdlouhavé.

Ceník

Lesní hospodářství a údržba	
Jarní a podzimní ošetření	750 Kč <small>/ hektar</small>
Pravidelná lesní péče	2500 Kč <small>/ hektar</small>
Kácení malých stromů (do 5 metrů)	1500 Kč <small>/ strom</small>
Kácení středně vysokých stromů (do 10 metrů)	4000 Kč <small>/ strom</small>
Kácení vysokých stromů (nad 10 metrů)	od 8000 Kč <small>/ strom</small>

Obrázek 7 – Table

Zdroj: Vlastní zpracování

V případě frameworku **Bootstrap** kód neobsahuje téměř žádné třídy dodané externím souborem, ale vše je nastýlováno přes HTML tagy.

```
<table>
  <tbody>
    <tr>
      <td colspan="2" class="title">Lesní hospodářství a údržba</td>
    </tr>
    <tr>
      <td>Jarní a podzimní ošetření</td>
      <td>750 Kč<small>/ hektar</small></td>
    </tr>
    <tr>
      <td>Pravidelná lesní péče</td>
      <td>2500 Kč<small>/ hektar</small></td>
    </tr>
  </tbody>
</table>
```

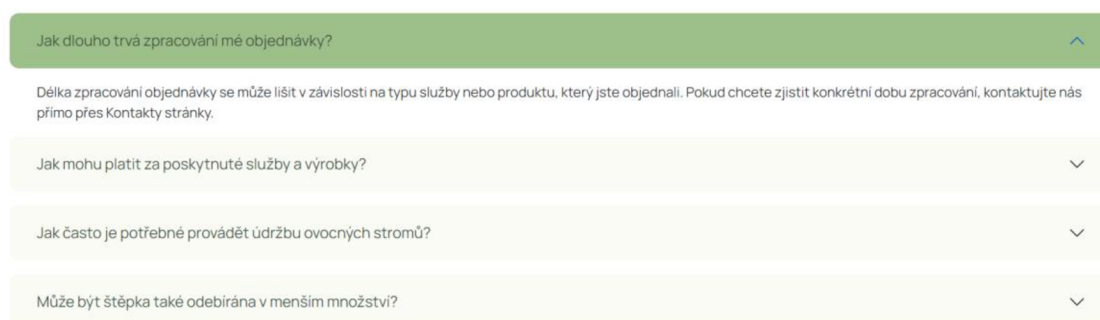
Naproti tomu **TailwindCSS** vyžaduje použití velkého množství tříd, aby bylo možné tabulku nastylovat takovýmto způsobem.

```
<table class="table-auto w-full">
  <tbody>
    <tr class="border-b border-b-black-200 leading-10">
      <td colspan="2" class="py-3 text-lg leading-10 font-
bold">Lesní hospodářství a údržba</td>
    </tr>
    <tr class="border-b border-b-black-200 leading-10">
      <td>Jarní a podzimní ošetření</td>
      <td class="flex flex-col justify-center text-end font-bold py-
2 leading-10 text-lg">750 Kč<small class="-mt-6 font-normal">/
hektar</small></td>
    </tr>
    <tr class="border-b border-b-black-200 leading-10">
      <td>Pravidelná lesní péče</td>
      <td class="flex flex-col justify-center text-end font-bold py-
2 leading-10 text-lg">2500 Kč<small class="-mt-6 font-normal">/
hektar</small></td>
    </tr>
  </tbody>
</table>
```

5.4.6 Accordion

Sekce – Často kladené dotazy (FAQ) – zobrazuje méně používanou komponentu, kterou Bootstrap nabízí. Jde o typ uživatelského rozhraní, které umožňuje uživatelům zobrazovat a skrývat obsah pomocí rozbalovací funkce.

Často kladené dotazy (FAQ)



Obrázek 8 – Accordion

Zdroj: Vlastní zpracování

Bootstrap, jak již bylo zmíněno, tuto komponentu obsahuje, a to i zcela funkční pomocí JS knihovny. Běžně se tato komponenta skládá z titulku, který v kódu figuruje jako tlačítko, a obsahu, který je dynamicky zobrazován nebo skrýván.

```

<div class="accordion" id="accordionList">
  <div class="accordion-item">
    <div class="accordion-header" id="accordionHeading1">
      <button class="accordion-button collapsed" type="button"
data-bs-toggle="collapse" data-bs-target="#accordionItem1" aria-
expanded="true" aria-controls="accordionItem1">
        Titulek prvku č.1
      </button>
    </div>
    <div id="accordionItem1" class="accordion-collapse collapse"
aria-labelledby="accordionHeading1" data-bs-parent="#accordionList">
      <div class="accordion-body">
        Textový obsah prvku č.1
      </div>
    </div>
  </div>
</div>

```

Ve frameworku **TailwindCSS** je kvůli absenci JS nemožné tuto komponentu vytvořit pouze za použití dodaných funkcí a vývojář je tedy nucen skript vytvořit sám. Základní struktura v HTML vypadá následovně.

```

<div id="accordionList">
  <div class="text-lg">
    <div id="accordionHeading1">
      <button class="w-full text-left rounded-lg bg-gray-100 py-
5 px-8" type="button">
        Titulek prvku č.1
      </button>
    </div>
    <div id="accordionItem1" class="text-left p-4">
      Textový obsah prvku č.1
    </div>
  </div>
</div>

```

5.5 Optimalizace

Výsledný produkt je po dokončení vývoje vhodné optimalizovat tak, aby potřebné zdroje pro jejich funkčnost byly co nejmenší.

Optimalizace frameworku **Bootstrap** je možná pomocí technologie SASS, která je zmíněna výše. Technologie SASS v použití s tímto frameworkem umožňuje vytvoření CSS souboru, pomocí kterého jsou do produktu importovány pouze ty části frameworku, které jsou opravdu použity, čímž je zajištěna minimalizace finálního stylového souboru. Tato implementace je nutná provést manuálně vloženými řádky o struktuře *@import nazev_funkce*. [5]

```
@import "functions";  
@import "variables";  
@import "mixins";  
@import "utilities";
```

Vytvořený stylový soubor je v posledním kroku optimalizace také možné minifikovat pomocí externího nástroje, které jsou dostupné i online. Minifikace je proces, který snižuje velikost souboru tak, že odstraňuje veškeré nepotřebné komentáře, znaky a mezery, kromě těch, které jsou součástí vlastností. Tento proces neovlivňuje funkčnost, ale snižuje velikost souboru. Výsledný soubor tak není strukturovaný, ale všechny kód je vložen pouze na jeden řádek souboru, čímž se stává pro vývojáře nečitelným a je tedy doporučeno tento proces aplikovat až jako poslední krok.

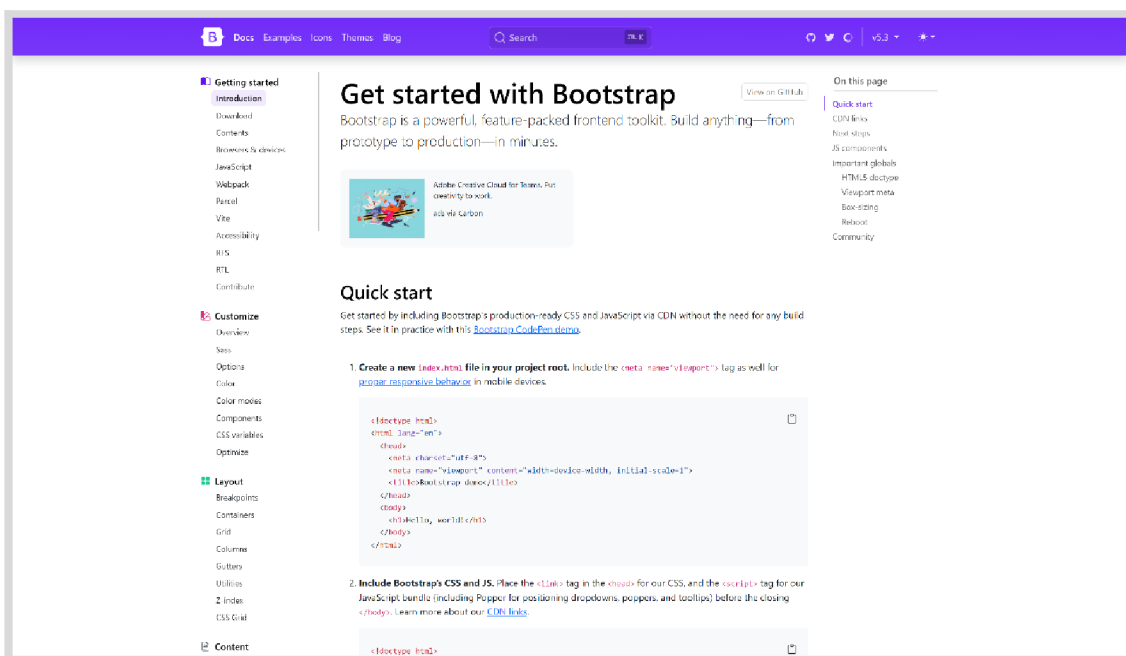
TailwindCSS lpí na optimalizaci podstatně více. V případě použití nástroje Tailwind CLI v průběhu vývoje, je možné využít nástroj PurgeCSS, který na pozadí vývojového prostředí sleduje, které konkrétní třídy vývojář používá a pouze ty následně ukládá do výsledného CSS souboru. Tím je zajištěno, že tento soubor neobsahuje žádné nadbytečné třídy a opravdu pouze ty potřebné a použité. Tento většinou velmi malý soubor je ještě možné vložením následujícího příkazu do příkazové řádky také minifikovat. [14]

```
npx tailwindcss -o build.css --minify
```

5.6 Dokumentace

Obecně lze říci, že oba vybrané frameworky mají velmi odlišné způsoby poskytování dokumentace. Bootstrap se zaměřuje více na přesné a detailní informace ke každé nabízené funkci, zatímco TailwindCSS se soustředí spíše na koncept a poskytuje větší volnost při vývoji.

Bootstrap poskytuje velmi podrobnou dokumentaci a mnoho ukázkových kódů, které je možné způsobem „copy and paste“ kdykoliv použít. Dokumentace vysvětluje každou nabízenou komponentu, funkce frameworku a poskytuje spoustu tipů a různých triků pro vývojáře. Tento framework také výrazně nahlíží na vývoj s ohledem na mobilní zařízení a responzivní design.



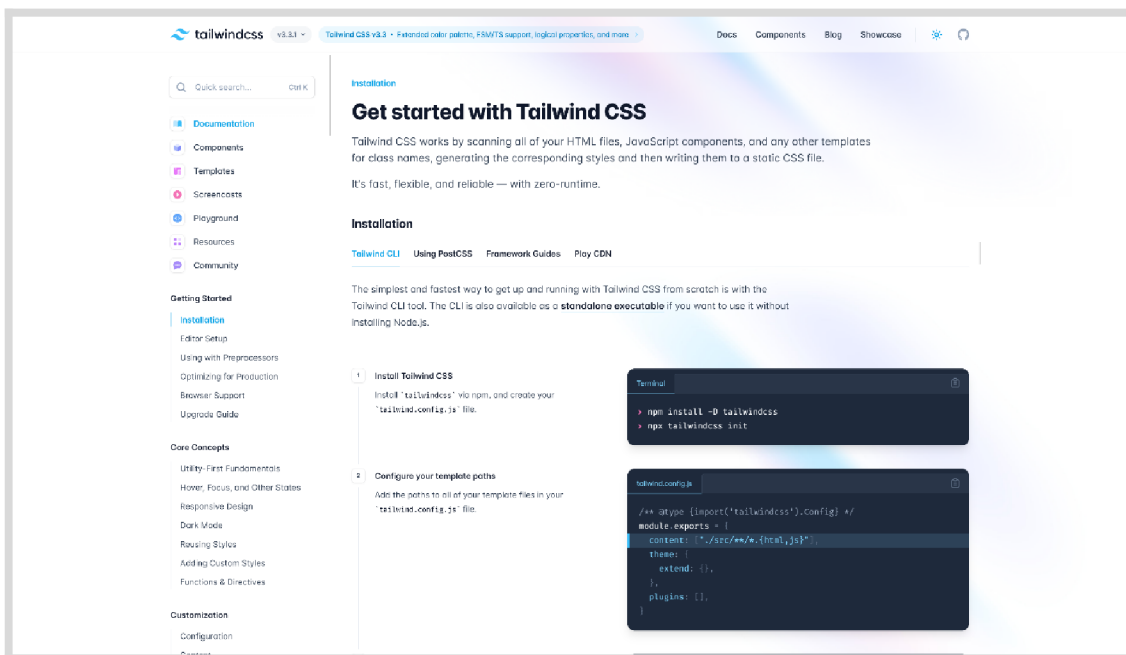
Obrázek 9 – Dokumentace frameworku Bootstrap

Zdroj: getbootstrap.com [5]

Vzhledem k tomu, že je Bootstrap oproti své konkurenci podstatně starší, jsou pro něj dostupnější šablony a různé komunitní návody, které tvoří přímo jeho uživatelé.

TailwindCSS poskytuje dokumentaci, která se více zaměřuje na filozofii a koncepty frameworku a méně na přímé použití komponent. Dokumentace se tak může zdát poměrně složitá a vývojáři s malými zkušenostmi s ní mohou mít problémy. Je to také způsobeno tím, že se framework zaměřuje na poskytování maximální flexibility

při vývoji, což je pro profesionální použití vhodné, ale pro začátečníky komplikované.



Obrázek 10 – Dokumentace frameworku TailwindCSS

Zdroj: tailwindcss.com [14]

V porovnání s konkurencí je dokumentace frameworku TailwindCSS vzhledově přívětivější, přehlednější a také svižnější.

5.7 Vliv použití na výsledný produkt

Vliv je sledován dvěma pohledy: velikost výsledného balíčku a výkon neboli rychlost načítání webových stránek v internetovém prohlížeči Google Chrome.

5.7.1 Velikost balíčku

Velikost adresáře, ve kterém se nachází veškeré soubory webových stránek je sledován správcem souborů Explorer v operačním systému Windows 11.

Vzhledem k velikostem testovacích stránek se může zdát vliv na velikost balíčku nevýrazný. Oba frameworky se zdají být obdobně velké, konkrétně se jedná přibližně o 3,5 MB. Konkrétní čísla je možné vidět na přiloženém obrázku níže.

Bootstrap		TailwindCSS	
Velikost	3,67 MB (3 854 973 bajtů)	Velikost	3,41 MB (3 586 043 bajtů)
Velikost na disku:	3,71 MB (3 895 296 bajtů)	Velikost na disku:	3,45 MB (3 620 864 bajtů)



Obrázek 11 – Velikost výsledných balíčků frameworků ve Windows 11

Zdroj: explorer.exe, operační systém Windows 11

Při zaměření na konkrétní soubory je již rozdíl více patrný, především u CSS souborů, které jsou u frameworku Bootstrap oproti konkurenci výrazně větší.

U frameworku **Bootstrap** obsahuje adresář se styly dva CSS soubory:

- bootstrap.min.css – veškeré styly, které Bootstrap poskytuje,
- style.css – přidané styly pro individualizaci.

 bootstrap.min	Soubor CSS	216 kB
 style	Soubor CSS	9 kB

Obrázek 12 – Velikosti souborů frameworku Bootstrap

Zdroj: explorer.exe, operační systém Windows 11

V celkovém součtu tak tento adresář zabírá 225 kB místa v paměti.

Adresář se styly u frameworku **TailwindCSS** obsahuje pouze jeden jediný soubor, který obsahuje jak jeho třídy, tak i styly navíc, kterými je framework individualizován.

 style	Soubor CSS	19 kB
---	------------	-------

Obrázek 13 – Velikost souboru frameworku TailwindCSS

Zdroj: explorer.exe, operační systém Windows 11

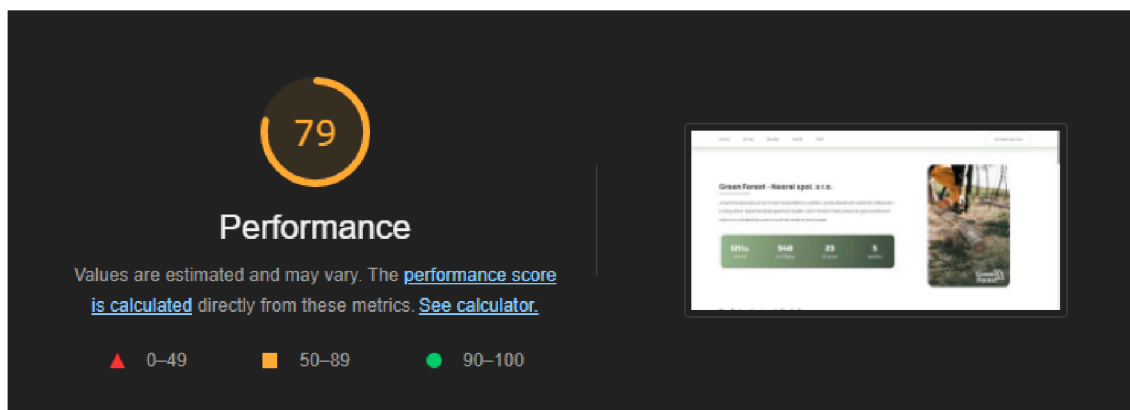
Tento soubor zabírá pouhých 19 kB paměti. Tento výrazný rozdíl je způsobený výše zmíněným nástrojem PurgeCSS, který do výsledného CSS souboru vkládá pouze ty třídy, které jsou v projektu opravdu použity, čímž výrazně snižuje jeho velikost.

5.7.2 Nástroj Google Lighthouse

Google Lighthouse je open-source nástroj, který je vhodný pro diagnostiku, analýzu výkonu, přístupnosti a SEO optimalizace. Je určený především pro vývojáře webových stránek a měří, jak rychle se stránka načítá, jak dobře je použitelná a přístupná i pro lidi se zdravotním znevýhodněním a také jak dobře se přizpůsobuje různým zařízením. [15]

V rámci měření je spuštěna pouze diagnostika výkonu, která dokáže odhalit i to, které konkrétní soubory ovlivňují rychlost načítání. Nástroj je spuštěn na jednom zařízení a ihned po sobě, aby byly zachovány obdobné podmínky a nedošlo k ovlivnění výsledků. Testování trvá přibližně deset vteřin.

Frameworku **Bootstrap** udělil tento nástroj hodnocení 79 bodů ze 100. Toto skóre bylo uděleno na základě několika faktorů, ale tím stěžejním je velikost zdrojů, konkrétně CSS a JS souborů frameworku.



Obrázek 14 – Výsledek nástroje Google Lighthouse frameworku Bootstrap

Zdroj: Google Lighthouse [15]

▲ Eliminate render-blocking resources 0.96s ^

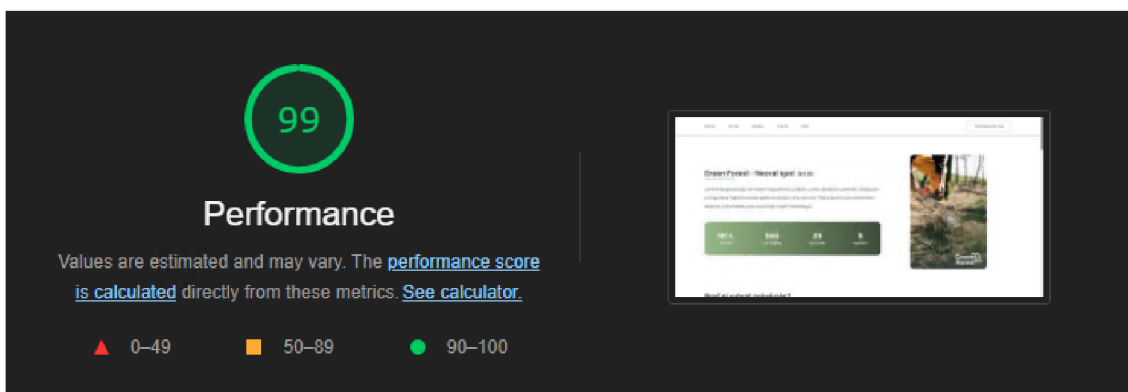
Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn how to eliminate render-blocking resources.](#) (FCP) (LCP)

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.neoloop.cz)	215.9 KiB	200 ms
...js/bootstrap.min.js (cdn.neoloop.cz)	59.5 KiB	80 ms

Obrázek 15 – Faktory hodnocení Google Lighthouse frameworku Bootstrap

Zdroj: Google Lighthouse [15]

Naproti tomu **TailwindCSS** obdržel výrazně lepší hodnocení, 99 bodů ze 100. Nástroj zdrojové soubory ani nezmiňuje, jelikož jsou dle něho v pořádku a výkon webu neovlivňují, minimálně ne nějak zásadně.



Obrázek 16 – Výsledek měření Google Lighthouse frameworku TailwindCSS

Zdroj: Google Lighthouse [15]

6 Shrnutí výsledků

Na základě veřejného dotazníkového šetření byly vybrány dva nejpopulárnější CSS frameworky, Bootstrap a TailwindCSS, které byly detailně porovnávány. V tomto porovnání byly sledovány klíčové aspekty, které jsou důležité pro tvorbu responzivních webových stránek. Ačkoliv jsou oba porovnávané frameworky nástroje určené pro vývoj webových stránek, jejich přístup k vývoji jako takovému se výrazně liší.

Bootstrap je framework s předdefinovanými třídami pro komponenty, což urychluje tvorbu webových stránek. Pro oblast responzivity jsou v tomto frameworku předdefinovány také breakpoint hodnoty. I přesto, že je Bootstrap určitým způsobem všeobecně flexibilnější, je komplikovanější upravit jeho základní styly. Mezi předem navržené komponenty se řadí navigace, carousel, formuláře, modal, ale i méně používané komponenty, jako např. accordion, breadcrumb a další, které jsou realizovány pomocí CSS a dodaných JS funkcí.

Framework TailwindCSS je utility-first framework, což znamená, že vývojářům umožňuje pracovat pomocí atomických tříd a je vhodnější pro tvorbu vlastního designu. Atomické třídy jsou takové třídy, které jsou velmi specifické a slouží pouze k jednomu účelu. Framework tak obsahuje třídu pro každou jednu CSS funkci a poskytuje rychlý způsob úpravy vzhledu webových stránek bez omezování vývojáře předem navrženými komponenty a styly. Pro komplikovanější, nebo dynamičtější funkce je ale vývojář nucen použít externí JavaScript knihovny, nebo si vytvořit skripty vlastní. V oblasti responzivity framework nabízí předem definované breakpoint hodnoty, které se od konkurenčního Bootstrapu liší, ale vývojáři je umožněno si je snadno upravit podle svého uvážení.

TailwindCSS poskytuje velmi přehlednou, svižnou a srozumitelnou dokumentaci, ve které je snadné cokoliv vyhledat, včetně vysvětlení, které je doplněno o praktickou, případně obrazovou ukázkou.

Konkurenční Bootstrap dokumentace není tak svižná, ani příliš graficky přívětivá, ale obsahuje velké množství použitelného kódu s ukázkami jeho využití, případně dalšími způsoby.

7 Závěry a doporučení

Cílem této práce bylo zjistit, které CSS frameworky jsou nejpoblárnější, a ty velmi detailně porovnat tak, aby bylo pro čtenáře možné, se na základě obsahu této práce rozhodnout, kterou cestou se vydat, nebo který framework použít pro konkrétní projekt. V této práci byly zkoumány teoretické i praktické rozdíly mezi dvěma nejpoblárnějšími frameworky, Bootstrap a TailwindCSS.

Praktickým výzkumem bylo zjištěno, že framework Bootstrap, který byl několik posledních let tím nejpoblárnějším a je stále velmi oblíbeným, je vhodný především pro začínající, nebo středně zkušené webové vývojáře, protože funguje na principu skládání již předpřipravených komponent a funkcí, ke kterým je poskytnuta velmi obsáhlá dokumentace. Z tohoto důvodu je oproti konkurenčnímu frameworku TailwindCSS mnohem hůře optimalizovaný a není tak vhodný pro velké webové projekty, u kterých je vyžadována maximální úspora prostředků. Vývojáři, kteří nemají ještě tolik zkušeností s HTML a CSS mohou pomocí použití tohoto frameworku zkušenosti nabývat pomocí odvozování funkcionalit z komponent.

TailwindCSS v posledních letech překonal svého konkurenta a aktuálně je největší špičkou v tomto oboru. I přes stručnou a přehlednou dokumentaci však může být, z důvodu utility-first způsobu vývoje a ne předpřipravených funkcí, pro začínající vývojáře komplikované s ním pracovat, a proto tento framework doporučuji středně pokročilým až velmi zkušeným vývojářům. TailwindCSS neposkytuje žádné JS skripty, takže pro dynamické funkce je nutné využít vlastního kódu. Díky optimalizačním nástrojům, se kterými dokáže framework spolupracovat, je kód výsledného produktu velmi úsporný a neovlivňuje tak výkon webových stránek ani v případě, kdy postupným vývojem nabývají na velikosti. Tento faktor má také pozitivní dopad na hodnocení internetových vyhledávačů a požitky pro uživatele s omezenými prostředky. Proto je TailwindCSS vhodný pro profesionální použití.

8 Seznam použité literatury

- [1] WANYOIKE, Michael. History of front-end frameworks [online]. LogRocket, Říjen 2018, [cit. 2023-04-16]. Dostupné z: <https://blog.logrocket.com/history-of-frontend-frameworks/>
- [2] The MIT License [online]. Open Source Initiative, Únor 2023, [cit. 2023-04-16]. Dostupné z: <https://opensource.org/license/mit/>
- [3] Websites using Bootstrap [online]. Wappalyzer, Červenec 2020, [cit. 2023-04-16]. Dostupné z: <https://www.wappalyzer.com/technologies/ui-frameworks/bootstrap/>
- [4] HTML Website Templates [online]. ThemeForest, [cit. 2023-04-16]. Dostupné z: <https://themeforest.net/category/site-templates>
- [5] Bootstrap · The most popular HTML, CSS, and JS library in the world [online]. Bootstrap, [cit. 2023-04-16]. Dostupné z: <https://getbootstrap.com/>
- [6] Vyhledávání Google [online]. Google, [cit. 2023-04-16]. Dostupné z: <https://www.google.com/search/howsearchworks/>
- [7] Sass Basics [online]. Sass, [cit. 2023-04-16]. Dostupné z: <https://sass-lang.com/guide>
- [8] The State of CSS 2022 [online]. The State of CSS 2022, [cit. 2023-04-16]. Dostupné z: <https://2022.stateofcss.com/>
- [9] Usage statistics and market share of Bootstrap for Websites [online]. W3Techs, [cit. 2023-04-16]. Dostupné z: <https://w3techs.com/technologies/details/js-bootstrap>
- [10] GONSCHOREK, Anne. The Story of Tailwind CSS feat. Adam Wathan [online]. The OfferZen Community Blog, [cit. 2023-04-16]. Dostupné z: <https://www.offerzen.com/blog/adam-wathan-story-tailwind-css>
- [11] Documentation – Materialize [online]. Materialize, [cit. 2023-04-16]. Dostupné z: <https://materializecss.com/>
- [12] The most advanced responsive front-end framework in the world. | Foundation [online]. Foundation, [cit. 2023-04-16]. Dostupné z: <https://get.foundation/>
- [13] tailwindcss/LICENSE [online]. GitHub, [cit. 2023-04-16]. Dostupné z: <https://github.com/tailwindlabs/tailwindcss/blob/master/LICENSE>
- [14] Rapidly build modern websites without ever leaving your HTML [online]. Tailwind CSS, [cit. 2023-04-16]. Dostupné z: <https://tailwindcss.com/>
- [15] Lighthouse overview [online]. Chrome Developers, [cit. 2023-04-16]. Dostupné z: <https://developer.chrome.com/docs/lighthouse/overview/>

9 Přílohy

1. Souhlas p. Tkáče s použitím jeho grafického návrhu webových stránek
2. Snímek celých webových stránek vytvořených pro praktické porovnání

Já, níže podepsaný Dávid Tkáč, tímto potvrzuji, že souhlasím s využitím mého grafického návrhu webových stránek v bakalářské práci *Porovnání CSS frameworků pro tvorbu responzivních webových stránek* zpracovávané Mikulášem Slavíkem. Jsem si plně vědom toho, že je můj návrh použit jako inspirace pro vývoj fiktivních webových stránek související s výzkumem rozdílu mezi využitými frameworky a vlivu jejich použití na výsledný produkt.

Tímto prohlašuji, že grafický návrh je pouze mé autorské dílo a že jsem seznámen s tím, jak bude v uvedené bakalářské práci použito a zpracováno.

Rovněž potvrzuji, že jsem si vědom toho, že tento souhlas není možné vzít zpět.

Dne 18. 04. 2023

Místo podpisu: Košice, Slovensko



Dávid Tkáč

Zadání bakalářské práce

Autor:	Mikuláš Slavík
Studium:	I2000411
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	Porovnání CSS frameworků pro tvorbu responzivních webových stránek
Název bakalářské práce AJ:	Comparison of CSS frameworks for responsive web design

Cíl, metody, literatura, předpoklady:

Cíl práce:

Cílem práce je seznámení s oblastí CSS frameworků, objasnění jejich účelu a využití. V rámci seznámení budou představeny dva nejpoužívanější frameworky, jejich historie, aktuální vývoj, výhody a nevýhody a doplňující služby. V rámci práce budou vypracované dvě webové stránky na porovnávání frameworků, které byly využity pro porovnání vývoje, flexibility, doplňujících funkcí, dokumentace a vlivu na výsledný produkt.

Osnova:

1. Teorie oblastí CSS frameworků
 1. Vysvětlení pojmů
 2. Historie frameworků
 3. Použití frameworků
 4. Uživatelé frameworků
 5. Výhody a nevýhody frameworků
 6. Obvyklé složení frameworku
2. Nejpoužívanější frameworky
3. Porovnání vybraných frameworků
 1. Instalace
 2. Teoretické porovnání
 3. Praktické porovnání
 4. Dokumentace
 5. Vliv na výsledný produkt

1. WANYOIKE, Michael. History of front-end frameworks [online]. LogRocket, Říjen 2018. Dostupné z: <https://blog.logrocket.com/history-of-frontend-frameworks/>
2. Bootstrap · The most popular HTML, CSS, and JS library in the world [online]. Bootstrap. Dostupné z: <https://getbootstrap.com/>
3. Rapidly build modern websites without ever leaving your HTML [online]. Tailwind CSS. Dostupné z: <https://tailwindcss.com/>
4. The State of CSS 2022 [online]. The State of CSS 2022. Dostupné z: <https://2022.stateofcss.com/>

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: Mgr. Hana Rohrová

Datum zadání závěrečné práce: 15.10.2021