

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



## **Bakalářská práce**

**Tvorba webové stránky stavební firmy pomocí  
frameworku Bootstrap**

**Martin Dub**

© 2022 ČZU v Praze



# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Martin Dub

Informatika

Název práce

**Tvorba webové stránky stavební firmy pomocí frameworku Bootstrap**

Název anglicky

**Creation of a Construction Company's Website Using the Bootstrap Framework**

---

### Cíle práce

V teoretické části budou popsány základní stavební kameny při tvorbě webu, tedy značkovací jazyk HTML5, kaskádový styl CSS3, framework JavaScript společně s programovacími jazyky PHP a metodami responzivního webu. Dále budou popsány nedostatky stávající stránky společně s porovnáním staré a nové stránky z hlediska kritérií vyhledávání a SEO.

Cílem praktické části bude vytvoření webové stránky stavební firmy pomocí frameworku Bootstrap.

### Metodika

V práci budou použity metody a standardy softwarového inženýrství. Řešení stránky bude responzivní díky Bootstrap frameworku. Dočasný hosting bude zajišťovat stránka GitHub.

Na finální stránce bude implementovaný kontaktní formulář přes programovací jazyk PHP a informace se budou předávat na e-mailovou adresu společnosti.

## Doporučený rozsah práce

30-60 stran

## Klíčová slova

webová grafika, HTML, CSS, JavaScript, PHP, GitHub, JIRFA s.r.o.

---

## Doporučené zdroje informací

HOGAN, Brian P. HTML5 a CSS3: Výukový kurz webového vývojáře. Brno: Computer Press, 2011. ISBN 9788025135761.

MATSINOPOULOS, Panos. Practical Bootstrap: Learn to Develop Responsively with One of the Most Popular CSS Frameworks. New York: Apress, 2020. ISBN 9781484260708.

TSITOARA, Mariot. Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. New York: Apress, 2019. ISBN 9781484253120.

---

## Předběžný termín obhajoby

2021/22 LS – PEF

## Vedoucí práce

Ing. Dana Vyníkarová, Ph.D.

## Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 1. 11. 2021

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 23. 11. 2021

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 09. 02. 2022

## **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "Tvorba webové stránky stavební firmy pomocí frameworku Bootstrap" 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 datum odevzdání

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Daně Vynikarové. Ph.D. za odborné vedení a profesionální pomoc při psaní této práce, za konzultace i cenné rady. Dále bych chtěl poděkovat firmě JIRFA s. r. o. za dodané materiály a účelné připomínky k webu.

# Tvorba webové stránky stavební firmy pomocí frameworku Bootstrap

## Abstrakt

Bakalářská práce zkoumá problematiku tvorby webové stránky a je určena pro vývojáře, kteří se chtějí naučit vyvíjet vlastní stránky pomocí jazyků HTML, CSS, JavaScript a PHP. Cílem práce bylo zjistit, jakým způsobem lze webovou stránku zmodernizovat a upravit na moderní, přístupnou stránku. Teoretická část obsahuje shrnutí technologií tvorby stránky, vhodné příklady a vlastnosti stránky. Praktická část obsahuje analýzu dvou populárních frameworků Bootstrap a Tailwind CSS. Oba byly v rámci práce srovnány a na základě analýzy byl vybrán finální framework. V této části je také popsána každá sekce webu, a to způsob tvorby, využití frameworku a finální snímek obrazovky sekce. Výstupem práce je responzivní webová stránka, která má zakomponované všechny požadavky zadavatelů.

**Klíčová slova:** webová stránka, stavební firma, JIRFA s. r. o., HTML, CSS, JavaScript, PHP, Bootstrap, GitHub

# **Creation of a Construction Company's Website Using the Bootstrap Framework**

## **Abstract**

The bachelor thesis explores the issue of website development and is intended for all developers who want to learn how to develop their own websites using HTML, CSS, JavaScript, and PHP. The thesis aimed to find out how a website could be modernized and adapted into a modern, accessible site. The theoretical part includes a summary of page creation technologies, suitable examples, and page features. The practical part contains an analysis of two very useful frameworks Bootstrap and Tailwind CSS, both were compared within the thesis, and based on the analysis the final framework was chosen. This section also describes each section of the site, namely how it was created, the use of the framework, and the final screenshot of the section. The output of the thesis is a responsive website that includes all the requirements of the clients.

**Keywords:** Web page, Construction Company, JIRFA s. r. o., HTML, CSS, JavaScript, PHP, Bootstrap, GitHub



# Obsah

<b>1 Úvod.....</b>	<b>- 1 -</b>
<b>2 Cíl práce a metodika .....</b>	<b>- 2 -</b>
2.1 Cíl práce .....	- 2 -
2.2 Metodika .....	- 2 -
<b>3 Teoretická východiska .....</b>	<b>- 3 -</b>
3.1 Zásady moderního webu .....	- 3 -
3.1.1 Responzivita.....	- 3 -
3.1.2 Mobile-first .....	- 3 -
3.1.3 SEO .....	- 5 -
3.1.4 Přístupnost .....	- 6 -
3.2 Existující nástroje tvorby webu.....	- 7 -
3.2.1 WordPress .....	- 7 -
3.2.2 Drupal .....	- 7 -
3.2.3 Ostatní CMS systémy .....	- 7 -
3.3 Tvorba vlastní stránky .....	- 8 -
3.3.1 HTML .....	- 8 -
3.3.1.1 Psaní HTML kódu .....	- 8 -
3.3.1.2 Šablony HTML.....	- 8 -
3.3.1.3 HTML značky .....	- 9 -
3.3.1.4 Nové strukturální značky a atributy HTML5 .....	- 12 -
3.3.2 CSS .....	- 13 -
3.3.2.1 Propojení HTML a CSS .....	- 14 -
3.3.2.2 CSS metodiky .....	- 14 -
3.3.3 JavaScript.....	- 17 -
3.3.3.1 Vložení Javascript kódu .....	- 18 -
3.3.4 Frameworky .....	- 20 -
3.3.4.1 Twitter Bootstrap.....	- 20 -
3.3.4.2 Tailwind CSS.....	- 22 -
3.3.5 PHP .....	- 23 -
3.4 Úložiště a hosting .....	- 24 -
3.4.1 GitHub .....	- 24 -
3.4.2 Verzovací nástroj Git .....	- 24 -
3.4.3 GitHub Pages .....	- 26 -
3.4.3.1 Jekyll.....	- 27 -

3.4.3.2	Hugo .....	- 27 -
3.4.3.3	Porovnání Jekyll a Hugo .....	- 27 -
<b>4</b>	<b>Vlastní práce .....</b>	<b>- 29 -</b>
4.1	Požadavky zadavatele .....	- 29 -
4.1.1	Funkční požadavky .....	- 29 -
4.1.2	Nefunkční požadavky .....	- 29 -
4.2	Analýza řešení .....	- 30 -
4.2.1	Závěr .....	- 30 -
4.3	Recenze původního webu .....	- 31 -
4.3.1	Nedostatky starého webu .....	- 31 -
4.4	Tvorba nové stránky .....	- 31 -
4.4.1	Návrh struktury .....	- 31 -
4.4.2	Metodika .....	- 33 -
4.5	Popis tvorby vytyčených cílů .....	- 34 -
4.5.1	Reference .....	- 34 -
4.5.2	Kontaktní formulář .....	- 36 -
4.5.3	Certifikáty .....	- 37 -
4.5.4	Ostatní sekce .....	- 38 -
4.6	Výsledky měření .....	- 40 -
<b>5</b>	<b>Závěr.....</b>	<b>- 41 -</b>
<b>6</b>	<b>Seznam použitých zdrojů .....</b>	<b>- 42 -</b>
<b>7</b>	<b>Přílohy .....</b>	<b>- 49 -</b>

## Seznam obrázků

Obrázek 1	- HTML značka meta a atributy pro stoprocentní šířku. ....	- 3 -
Obrázek 2	- Ukázka CSS kódů (nejen) s přístupem mobile-first .....	- 4 -
Obrázek 3	- Snímek obrazovky Lighthouse reportu webu Provozně ekonomické fakulty ČZU pořízený autorem práce.....	- 5 -
Obrázek 4	- HTML šablona editoru WebStorm .....	- 8 -
Obrázek 5	- Obsah šablony HTML5 Boilerplate.....	- 9 -
Obrázek 6	- Konvence pojmenování jednotlivých částí HTML značky .....	- 10 -
Obrázek 7	- Nadpis s upravenou vlastností velikosti písma. ....	- 10 -
Obrázek 8	- Jednotlivé druhy textové značky a. ....	- 10 -
Obrázek 9	- Ukázka rozdílu uzavírání HTML značek.....	- 11 -
Obrázek 10	- Tři příklady používání atributů .....	- 11 -
Obrázek 11	- Rozložení blogu podle nových sémantických značek.....	- 13 -
Obrázek 12	- Ukázka propojení CSS souboru s HTML dokumentem .....	- 14 -
Obrázek 13	- Metodika BEM – blok .....	- 14 -
Obrázek 14	- Metodika BEM – elemen .....	- 15 -
Obrázek 15	- Metodika BEM – modifikátor.....	- 15 -

Obrázek 16 - Metodika OOCSS .....	- 16 -
Obrázek 17 - Jak různé metody vložení souboru ovlivňují načasování stažení a provedení JavaScriptového kódu .....	- 18 -
Obrázek 18 - Vložení JavaScript kódu do HTML souboru .....	- 19 -
Obrázek 19 - Diagram vztahu framework - kód - knihovna.....	- 20 -
Obrázek 20 - Mřížkový systém frameworku Bootstrap.....	- 20 -
Obrázek 21 - Rozložení objektů podle tříd justify-content-start, end, center, between a around.....	- 22 -
Obrázek 22 - Fragment kódu s frameworkem Tailwind CSS.....	- 22 -
Obrázek 23 - Model využití PHP při vyplnění dotazníku .....	- 23 -
Obrázek 24 - Funkcionalita verzovacího systému Git.....	- 25 -
Obrázek 25 - Demonstrace podmíněného příkazu pomocí generátoru Jekyll.....	- 28 -
Obrázek 26 - Demonstrace podmíněného příkazu pomocí generátoru Hugo.....	- 28 -
Obrázek 27 - Analýza původního webu nástrojem Lighthouse.....	- 31 -
Obrázek 28 - Snímek drátěného modelu.....	- 32 -
Obrázek 29 - Snímek drátěného modelu.....	- 33 -
Obrázek 30 - Nastavení tříd pro upravené zobrazení reference.....	- 35 -
Obrázek 31 - Kompletní kód zobrazení jedné reference. ....	- 35 -
Obrázek 32 - Zobrazení reference ve výchozím stavu. ....	- 35 -
Obrázek 33 - Zobrazení reference při přejetí myši. ....	- 35 -
Obrázek 34 - Náhled kontaktního formuláře v desktop prostředí. ....	- 37 -
Obrázek 35 - Kód pro zobrazení certifikátu společnosti. ....	- 38 -
Obrázek 36 - Náhled sekce certifikátů.....	- 38 -
Obrázek 37 - Zobrazení navigace přes navbar-toggler.....	- 38 -
Obrázek 38 - Zobrazení navigace v desktop prostředí.....	- 38 -
Obrázek 39 - Hlavička stránky. ....	- 39 -
Obrázek 40 - Rozcestník s kategoriemi referencí.....	- 39 -
Obrázek 41 - Sekce partnerů a dodavatelů. ....	- 39 -
Obrázek 42 - Sekce partnerů a dodavatelů po najetí myši na prvek.....	- 40 -
Obrázek 43 - Patička stránky.....	- 40 -
Obrázek 44 - Analýza nového webu nástrojem Lighthouse .....	- 40 -

## Seznam tabulek

Tabulka 1 - Body zlomu frameworku Bootstrap .....	- 21 -
Tabulka 2 - Základní příkazy verzovacího nástroje Git .....	- 26 -

## Seznam použitých zkratek

BEM	Block Element Modifier
CDN	Content Delivery Network
CGI	Computer-Generated Image
CMS	Content Management System
CSS	Cascade Style Sheet
DOM	Document Object Model
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP(S)	Hypertext Transfer Protocol (Secured)
IDE	Integrated Development Environment
IoC	Inversion of Control
JS	JavaScript
kB	Kilobyte
MB	Megabyte
MIT	Massachusetts Institute of Technology
MS	Microsoft
MVČR	Ministerstvo vnitra České republiky
NPM	Node Package Manager
OOCSS	Object-Oriented Cascade Style Sheet
PHP	Hypertext Preprocessor
PWA	Progressive Web Application
SEO	Search engine optimization
SMACSS	Scalable and Modular Architecture Cascade Style Sheet
SRI	Subresource Integrity
UI	User Interface
URL	Uniform Resource Locator
USA	United States of America
UTF-8	Universal Transformation Format-8
UX	User Experience
W3C	World Wide Web Consortium

WAI

Web Accessibility Initiative

WAI-ARIA

Web Accessibility Initiative-Accessible

Rich Internet Applications

WYSIWIG

What You See Is What You Get

# 1 Úvod

V dnešní době je internet neodmyslitelnou součástí každodenního života nejednoho občana. Dnes má každá firma, každý podnik i každá společnost své vlastní webové stránky. Dostupnost dat nebyla nikdy rychlejší, otevřenější ani svobodnější. V případě hledání informací jen málokdo použije jinou věc než mobilní telefon, tablet či počítač. Na internetu je nepřehledné množství dat a informací, a proto jsou webové stránky našimi pomocníky v bádání.

Společnost JIRFA s.r.o. vlastní webové stránky, jež nejsou optimalizované, responzivní ani aktuální. S majiteli této společnosti byla dohodnuta spolupráce na modernizaci webových stránek v rámci této bakalářské práce.

Práce je rozdělena na teoretickou a praktickou část. V první kapitole jsou popsány technologie a metody tvorby moderního webu, dále zásady moderního webu a verzovací systém GitHub užitečný nejen pro tuto práci, ale i jako nástroj pro spolupráci při ostatních projektech. V teoretické části lze nalézt fragmenty kódu pro ukázky popisovaných problémů nebo metodik zápisu. V rámci práce byla vypracována analýza na existující frameworky pro tvorbu webu, aby byl zajištěn cíl práce.

Praktická část se věnuje popisu vytyčených cílů a jejich dosažení, včetně metodiky tvorby každé části webové stránky. Na finálním řešení této práce není možné demonstrovat funkčnost odesílání kontaktních formulářů, protože ze strany poskytovatele hostingu GitHub Pages nejsou povoleny dynamické jazyky, ačkoliv kód bude na stránce přítomný a funkční.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

V teoretické části budou popsány základní stavební kameny při tvorbě webu, tedy značkovací jazyk HTML5, kaskádový styl CSS3 a skriptovací jazyk JavaScript. Do finální verze stránky bude implementován framework Bootstrap a programovací jazyk PHP pro kontaktní formulář. Nový responzivní web bude porovnán se starým webem společně s nedostatky starého webu a inovacemi webu nového. Výsledný projekt bude také obsahovat metody moderního vyhledávání „SEO“, jež v překladu znamená „optimalizace pro vyhledávače“.

### **2.2 Metodika**

V práci budou použity metody a standardy softwarového inženýrství. Řešení stránky bude responzivní díky frameworku Bootstrap a postavené metodou mobile-first. Dočasný hosting bude zajišťovat stránka GitHub společně se službou GitHub Pages. Nová stránka také bude splňovat podmínky přístupnosti.

## 3 Teoretická východiska

### 3.1 Zásady moderního webu

Každý nově vznikající web by měl mít moderní vlastnosti a funkcionality. Tím je myšlena primárně responzivita a celý web je postavený metodou mobile-first. To znamená, že výchozí zobrazení je optimalizováno pro mobilní telefony. Moderní web musí mít dobré SEO skóre, což je metoda vyhodnocení, jak dostat stránku na nejvyšší pozice ve vyhledávání.

#### 3.1.1 Responzivita

Úplným základním kamenem každé webové stránky by měla být responzivita. To znamená, že se zobrazení stránky přizpůsobí šířce okna, který stránku zobrazuje uživatel. Díky responzivitě není potřeba mít oddělené stránky pro jednotlivá zařízení, web se zaobalí automaticky podle uživatelského rozhraní. Responzivita se nastavuje pomocí HTML značky `<meta>`. Například

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

*Obrázek 1 - HTML značka meta a atributy pro stoprocentní šířku (W3schools.com, 2022h).*

vygeneruje stránku se sta procenty šířky (W3Schools.com 2022h).

Pojem responzivita vznikl v roce 2010 Ethanem Marcottem, jenž sjednotil tehdy existující tři metodiky (rozložení mřížky, flexibilní obrázky a media-queries) do jednoho principu zvaného responzivní webdesign. Ačkoliv jsou ve výchozím nastavení texty na webové stránce responzivní (díky blokovým elementům, viz kapitola 3.3.1.4), je lepší vždy zavést již zmíněný `meta` tag na Obrázek 1 pro přívětivější zobrazení. Při vkládání obrázků na stránku je dobré přidat do CSS souboru `img {max-width: 100%}`, aby se obrázky, které jsou širší než šířka zařízení, přizpůsobily (Frain, 2020).

#### 3.1.2 Mobile-first

Novým trendem ve vytváření webových stránek je princip takzvaný mobile-first. Tento krok zapřičinili sami uživatelé, kteří se ve většině (55,04%) připojují na internet skrz mobilní telefon (Johnson, 2021).

V červenci 2010 byl přístup na internet z mobilních zařízení necelá tři procenta, o rok déle přesáhl podíl sedm procent. V polovině roku 2014 se zvedl poměr používání telefonů



na 29,48% a v září 2019 přesáhly přístupy padesát procent, konkrétně 51,11% (Frain, 2020). V době psaní této práce se tato hodnota pohybuje na úrovni 55% s odchylkou maximálně dvou procent za uplynulý rok (Statcounter.com, 2022). Z celkového počtu 4,66 miliardy všech aktivních uživatelů se na internet v lednu 2021 takto připojilo 4,32 miliard uživatelů (Johnson, 2021).

Tuto novou metodu psaní webových stránek začali implementovat i vývojáři, kteří nyní kódují již zmíněným stylem *mobile-first*. To znamená, že v CSS souboru se vytvoří tzv. *media-query*, což jsou podmínky, které umožňují aktivaci či deaktivaci stylů a funkcí podle aktuální velikosti obrazovky a vývojář začne kódovat stránku od nejmenších typů zobrazení (Bittner, 2021).

```
/* Výchozí styl */
#header-image {
    background-repeat: no-repeat;
    background-image: url('obrazek.gif');
}

/* Uživateli se zobrazí větší obrázek, pokud na stránku přistupuje ze
širšího zařízení */
@media screen and (min-width: 1200px) {
    #header-image {
        background-image: url("obrazek-velky.gif");
    }
}

/* Mimo zobrazení se dá nastavit např. schování obrázku při tisku */
@media print {
    #header-image {
        display: none;
    }
}
```

Obrázek 2 - Ukázka CSS kódů (nejen) s přístupem *mobile-first* (Felsen, 2017).

Kompletní logika se dá i invertovat přidáním slova *not* na začátek atributu *media*, celé nastavené zobrazení se tedy bude aplikovat na všechna rozlišení kromě daného rozlišení v atributu *media* (Frain, 2020).

Media-queries se v praxi využívají především ve spojitosti se šířkou. V určitých případech lze namísto použití šířky zkombinovat rozlišení společně s výškou nebo dynamické nastavení přes poměr stran (*aspect-ratio*). Další segment, který lze nastavit pomocí *media-query*, je nastavení barvy. Hodnota atributu *color* značí počet bitů na komponentu, *color-index* počet vstupů do barevné tabulky (tabulka udává způsob změny z jedné barvy na druhou barvu) a *monochrome* počet bitů na pixel v paměťové

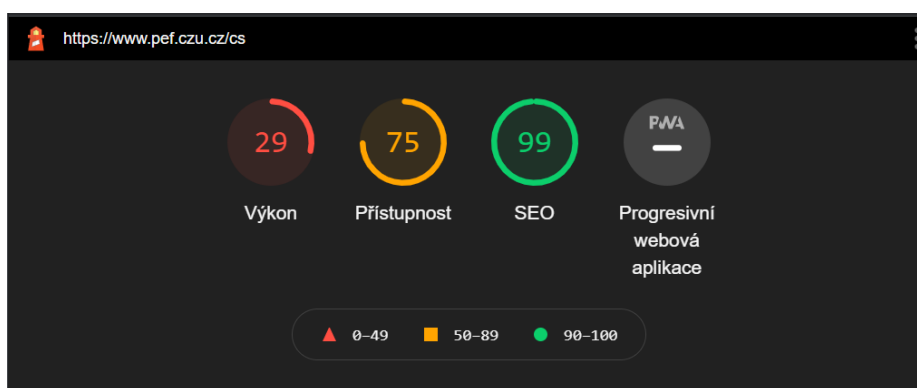
sběrnici (anglicky frame buffer). Poslední dva příklady (color-index a monochrome) musí mít nezáporné celočíselné hodnoty (Frain, 2020).

### 3.1.3 SEO

Aby existovala soutěž o přední příčky vyhledávání, vznikla, přibližně ve stejné době jako první webové stránky vůbec, i optimalizace vyhledávání neboli SEO (Search engine optimization). Tento nástroj měl být řešením na problematiku rychle rostoucího počtu vznikajících webových stránek a jejich třídění při vyhledávání podle relevance. Ze začátku se řešilo pouze SEO. Správci stránek tedy přidávali velmi dlouhé texty klíčových slov, jež byly například zabarveny do pozadí, aby je zákazník neviděl, ale SEO naopak ano a potažmo stránku umisťovalo výše ve výsledcích vyhledávání (A Brief History of SEO, 2017).

Později se příležitosti chopila americká firma Google, jež přetvořila SEO na algoritmus, který odměňoval obsah dle kvality a relevance a následně danou webovou stránku spojil s uživatelem, který ji chtěl najít (A Brief History of SEO, 2017).

Dnes lze jednoduše zjistit nejen SEO výsledky jakékoliv stránky, je k tomu potřeba internetový prohlížeč Google Chrome. Po stisku klávesy F12 se objeví panel pro vývojáře, a na záložce *Lighthouse* se objeví modré logo Google Lighthouse. Tento nástroj analyzuje otevřenou stránku z hlediska výkonu, přístupnosti, postupů při tvorbě webu, SEO a PWA (progresivní webová aplikace). Po dokončení analýzy se pro každou kategorii zobrazí výsledky do kruhu, viz Obrázek 3 včetně nápovědy, jak danou kategorii vylepšit, aby měla vyšší skóre, a tedy i lepší umístění ve vyhledávání (Tools for Web Developers, 2022).



Obrázek 3 - Snímek obrazovky Lighthouse reportu webu Provozně ekonomické fakulty ČZU pořízený autorem práce (Tools for Web Developers, 2022).

Tento report byl vygenerován autorem práce 22.2.2022 pomocí programu Lighthouse.

### 3.1.4 Přístupnost

Účelem přístupnosti je odstranění překážek a zajištění dostupnosti informací pro širokou škálu uživatelů. Web byl založen v roce 1989, nicméně iniciativa přístupnosti stránek vznikla až o devět let později konsorciem W3C a dostala pojmenování WAI (Web Accessibility Initiative) (Dowden, Dowden, 2019).

Dle webu <http://pristupnost.cz> se mezi základní pravidla přístupnosti řadí dostupnost a čitelnost obsahu webových stránek, srozumitelné a přehledné informace, jasné a pochopitelné ovládání webu, technicky způsobilý kód či prohlášení o přístupnosti. Tyto metodické pokyny vydává MVČR v podobě vyhlášek (Pristupnost.cz, 2022)

V dnešní době existují čtečky obrazovky, což je software na překlad grafického kontextu na stránce do mluvené podoby. Nevýhodou čteček je jejich zaostávání za současnými trendy, nedokáží si poradit se živými oblastmi, například hlasování na stránce (Hogan, 2011).

Přístupnost dynamických aplikací zajišťuje metodika WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications). Některé prvky této specifikace převzalo HTML5 (Hogan, 2011). Hodí se hlavně pro práci s JavaScript moduly (Hogan, 2011; Dowden, Dowden, 2019). ARIA mimo jiné využívá tzv. role, které označují „body zájmu“ na stránce, jako je banner, navigace nebo vyhledávací oblast, takže je čtečka obrazovky může rychleji rozeznat. Implementace ARIA specifikace lze provést přidáním atributu `role` do HTML značky a nastavit ji hodnotu. V roce 2011 byly validní hodnoty `application`, `banner`, `complementary`, `contentinfo`, `main`, `navigation` a `search` (Hogan, 2011).

O devět let později byl seznam aktualizován na validní hodnoty `alert`, `alertdialog`, `application`, `contentinfo`, `dialog`, `document`, `log`, `main`, `marquee`, `presentation`, `search` a `status`. Všechny ostatní hodnoty přeloží jako výchozí, a tedy nenastavené. Díky těmto rolím lze usnadnit přístupnost čtečkám společně s používáním správných značek. HTML tag `header` je z hlediska přístupnosti vhodnější než `<div class="header">`, případně tag `<button>` oproti ostatním substitucím tlačítka (Frain, 2020).

## **3.2 Existující nástroje tvorby webu**

V dnešní době existují dva způsoby, jak psát webovou stránku: metodou tzv. *WYSIWYG* (What You See Is What You Get) nebo psaním HTML značek v editoru (Janovský, 2021a). Editorem je myšlen software pro vývojové prostředí, jenž usnadňuje práci vývojářům z hlediska psaní kódu, protože každé IDE již obsahuje základní konfiguraci pro psaní různých kódů programovacími jazyky. Výhodou je i mimo jiné kontrola psaného kódu a hledání případných chyb a snaha o jejich nápravu (RedHat.com, 2019).

*WYSIWIG* editory jsou určeny pro začátečníky v oblasti tvorby stránek, nevyžadují totiž žádný ručně psaný kód. Veškeré prvky jsou tzv. klikací, uživatel si vybere aplikaci a přizpůsobí si ji dle svých požadavků. Mezi nejznámější online editory patří WordPress, Drupal a jiné popsané v následujících podkapitolách (Janovský, 2022).

### **3.2.1 WordPress**

WordPress je redakční systém s otevřeným zdrojovým kódem založený na PHP, který lze použít pro vytváření webových stránek. V dnešní době funguje přes systém WordPress více než 42 % všech existujících webových stránek (W3techs.com, 2022). Je to zapříčiněno především jeho snazším ovládním a oblíbeností mezi uživateli. Velkou předností je bezplatný přístup, bezpečnost, responzivita nebo správa SEO a medií. V rámci systému WordPress existují také placené prémiové šablony nebo pluginy (Wordpress.org, 2022).

### **3.2.2 Drupal**

Dalším populárním redakčním systémem je Drupal, který je sice méně rozšířený než WordPress, ale je využíván i vládními institucemi, např. v USA, Austrálii, Velké Británii nebo Německu. Zaměřuje se primárně na bezpečnost, UX (User Experience, uživatelská přívětivost) a multijazyčnost. V tuzemsku tento CMS systém využívá webový portál iPrima, Česká televize nebo Český rozhlas (Drupal.cz, 2021).

### **3.2.3 Ostatní CMS systémy**

Za zmínku jistě stojí i další CMS jako je například Joomla!, který je zdarma a stejně jako WordPress založený na PHP (Hypertext Preprocessor). Uživatelská oblíbenost zde

trochu klesá a používá se spíše na pokročilejší, obsahově propracovanější stránky. Dalším placeným systémem je ExpressionEngine, který je stejně jako všechny předchozí CMS založený na PHP. Pro triviální webové stránky je tento systém až příliš složitý, a proto se využívá hlavně v obsahově rozsáhlejších projektech (Interval.cz, 2011).

### 3.3 Tvorba vlastní stránky

#### 3.3.1 HTML

Pro vytvoření jednoduché webové stránky je potřeba znát značkovací jazyk HTML (Janovský, 2021b). Kompletní dokumentace (nejen) HTML je popsána včetně příkladů na stránkách organizace W3C [w3schools.com](http://w3schools.com) (World Wide Web Consortium), která udává standardy tohoto jazyka (Hogan, 2011). V českém jazyce jsou HTML a CSS popsány na webu [jakpsatweb.cz](http://jakpsatweb.cz) (Janovský, 2022).

##### 3.3.1.1 Psaní HTML kódu

Každý HTML dokument musí obsahovat deklaraci typu dokumentu, pro HTML je tato deklarace `<!DOCTYPE html>`, následovaný značkami `<html></html>`, které označují začátek a konec HTML kódu. Viditelná oblast pro uživatele je napsaná vně značek `<body></body>` (W3schools.com, 2022c). Jednoduchá šablona obsažená v IDE WebStorm od společnosti JetBrains následně vypadá takto (Jetbrains.com, 2021):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>

</body>
</html>
```

Obrázek 4 - HTML šablona editoru WebStorm (Jetbrains.com, 2021).

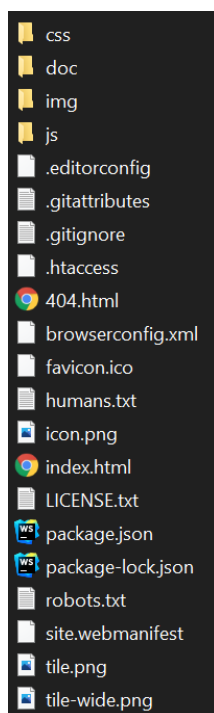
Jednotlivé šablony se od sebe odlišují, jejich základ je nicméně stejný – nastavit vývojové prostředí tak, aby bylo připravené pro vývojáře na psaní kódu.

##### 3.3.1.2 Šablony HTML

Na internetu existuje nepřeberné množství šablon, které nabízejí možnost začít pracovat na stránce téměř ihned po stažení. Nejuniverzálnější šablonou je

HTML5 Boilerplate, jež obsahuje všechny potřebné prvky i doplňky pro začátečníky. Kompletní dokumentace je dostupná online v GitHub repozitáři na stránce [html5boilerplate.com](https://html5boilerplate.com) (Meloni, 2014).

Na Obrázku 1 níže lze najít šablonové soubory jako *.htaccess*, který umožňuje autorovi upravit nastavení serveru bez požádání správce (Github.com, 2020). Dále dvojice souborů *humans.txt* a *robots.txt*, kdy první obsahuje informace o vývojářích (pokud informace někdo vyplnil) a druhý zakazuje přístup robotům na různé stránky. Každý robot by si měl před vstupem na stránku přečíst soubor *robots.txt*, jestli náhodou nemá někde zakázaný vstup (Janovský, 2021c). Boilerplate šablona také obsahuje balíček *package.json* pro případ, že by vývojář chtěl projekt běžet společně s NPM, základní soubory JS a CSS a spoustu dalšího (Github.com, 2020).



Obrázek 5 - Obsah šablony HTML5 Boilerplate (HTML5 Boilerplate, 2022).

### 3.3.1.3 HTML značky

Značky pomáhají určit a stylizovat jednotlivé bloky textu. Podle W3schools.com existuje k dnešnímu datu 131 HTML značek, potažmo elementů, nicméně 119 z nich je validní z hlediska nového formátu HTML5 (W3schools.com, 2022d). V následujících

odstavcích budou popsány nejdůležitější a nejznámější značky jazyka HTML, včetně atributů a jejich vlastností.

```
<značka atribut="vlastnost">Zobrazený text</značka>
```

Obrázek 6 – Konvence pojmenování jednotlivých částí HTML značky (W3Schools.com, 2022b).

Nadpisy se píšou do značek <hX>. Písmeno X značí číslo, které udává důležitost nadpisu. Nadpis v tagu <h1> by měl obsahovat název stránky a také umístěn nejvýše v kódu. Další nadpisy následují sekvenčně (značka <h3> nesmí být nad značkou <h2> atp.) Další nastavení nadpisů, například velikost písma, by měla odpovídat i prioritě nadpisu. Každý nadpis má předem nastavenou velikost písma, která se dá změnit atributem `style` a CSS vlastností `font-size`. Obrázek 7 níže zobrazuje upravený nadpis h1 s velikostí písma 42 pixelů (W3schools.com, 2022c):

```
<h1 style="font-size: 42px">Ahoj světe!</h1>
```

Obrázek 7 - Nadpis s upravenou vlastností velikosti písma (W3Schools.com, 2022c).

Značky pro textová pole se rozlišují podle typu textu. Značka <a> definuje hypertextový odkaz a cílové umístění odkazu se zapisuje do atributu `href`. Odkaz nemusí být pouze URL adresa, přesměrování bude fungovat i s odkazem na obrázek v adresáři nebo na sekci stránky. V posledním případě lze odkazovat pouze na sekci se zavedeným atributem `id`, viz Obrázek 8. Odkazy se projevují třemi způsoby: nenavštívený link se na stránkách zobrazuje modrou barvou a text odkazu je podtrhnutý, navštívený fialovou barvou a text podtrhnutý, aktivní link červenou barvou a text taktéž podtrhnutý (W3schools.com, 2022c).

```
<a href="https://google.com">Google</a> <!-- Odkaz s URL adresou -->  
<a href="obrazek.gif">Obrázek</a> <!-- Odkaz na obrázek -->  
<a href="#kontakty">Kontakty</a> <!-- Odkaz na sekci Kontakty -->
```

Obrázek 8 - Jednotlivé druhy textové značky a (W3Schools, 2022c).

Značka <p> značí odstavec a text vždy začíná na novém řádku. Některé prohlížeče ještě automaticky přidávají mezery před a za značku. Zároveň upravuje zarovnání podle šířky obrazovky a odebírá zbytečné mezery obsažené v textu. Pokud ale uživatel potřebuje zachovat mezery či jiné úpravy textu, které by byly smazány, lze použít značku <pre> (předformátovaný text) a po spuštění kódu se zobrazí fontem Courier (W3schools.com, 2022f).

Zkrášení stylu stránky lze docílit vložení obrázků pomocí tagu `<img>`. Tato značka je narozdíl od výše zmíněných self-closing (samouzavíratelná), tedy nepotřebuje druhý tag, který by značil konec, viz Obrázek 9 (W3Schools.com, 2022c).

```

```

```
<a>Vzorový text na ukázkou self-closing značek</a>
```

Obrázek 9 - Ukázka rozdílu uzavírání HTML značek (W3Schools.com, 2022c).

Atributy každé značky umožňují dále upravovat její nastavení. V předchozích příkladech byly uvedeny některé z nich. Na Obrázek 10 je demonstrován atribut `style`, který se v dnešní době téměř nevyužívá. Jeho funkci totiž dokáže plně zastat připojený CSS soubor a díky němu lze měnit vzhled HTML stránky, aniž by se do ní muselo zasahovat, jak je zobrazeno na Obrázek 10 (Blender, 2013).

```
<!-- Značka se špatnou validitou -->  
<h2 style="font-size: 36px">Méně podstatný nadpis</h2>
```

```
<!-- Využití propojení HTML a CSS -->  
<!-- model.html -->  
<h2>Méně podstatný nadpis</h2>
```

```
/* model.css */  
h2 {  
    font-size: 36px;  
}
```

```
<!-- Validní zápis značky <h2> -->  
<h2 class="heading2">Méně podstatný nadpis</h2>  
  
/* Název třídy musí být definován i v CSS souboru */  
.heading2 {  
    font-size: 36px;  
}
```

Obrázek 10 - Tři příklady používání atributů (Blender, 2013).

Dalšími neopomenutelnými atributy jsou:

- `href` a `src` popsané v kapitole 3.3.1.3,
- `alt`, který je z hlediska přístupnosti povinný a poskytuje alternativní text obrázku,
- `lang` ve značce `<html>`, jenž specifikuje jazyk webové stránky,
- `title` přidávající extra informace o elementu.

(W3schools.com, 2022b)



### 3.3.1.4 Nové strukturální značky a atributy HTML5

Před vznikem HTML5 se veškerý obsah dával do blokových elementů `<div>`. Blokový element znamená, že celý obsah funguje alespoň na jednom celém řádku a text uvnitř se, v případě potřeby, zalamuje, oproti řádkovým elementům (například značka `a`), které takovou funkci postrádají. „S příchodem HTML5 se zavedly nové sémantické značky popisující obsah, jež označují, viz seznam níže:

`<header>`

*Definuje oblast záhlaví nebo její části,*

`<footer>`

*Definuje oblast zápatí nebo její části,*

`<nav>`

*Definuje oblast s navigací stránky nebo její části,*

`<section>`

*Definuje logickou oblast stránky nebo její části,*

`<article>`

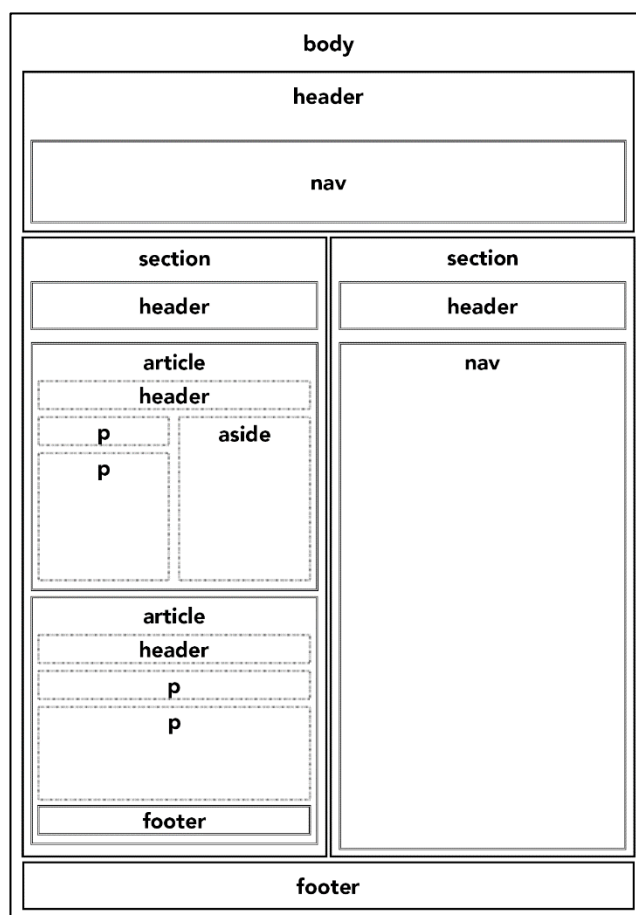
*Definuje článek nebo kompletní část obsahu,*

`<aside>`

*Definuje sekundární nebo příbuzný obsah“ (Hogan, 2011).*

Tyto značky jsou kompatibilní v prohlížečích od těchto verzí: Chrome verze 26, Firefox verze 21, Opera verze 15, MS Edge verze 12, Internet Explorer bez podpory dle [caniuse.com](http://caniuse.com) (Hogan, 2011).

Podle těchto nových značek lze mentální model webové stránky rozvrhnout přibližně takto:



Obrázek 11 - Rozložení blogu podle nových sémantických značek (Hogan, 2011).

### 3.3.2 CSS

Jak již bylo v předchozí kapitole naznačeno, je nedílnou součástí každé webové stránky i kaskádový styl CSS. Tento sešit udává, jak se mají HTML prvky zobrazovat. V původní vizi konsorcia W3C nebyl tento typ dokumentu zahrnut, počítalo se pouze s HTML značkami, které měly vystačit na poptávku vývojářů a zákazníků.

Ve verzi HTML 3.2 přibyly nové značky, například `<font>`, který zkomplikoval práci vývojářům a vytváření rozsáhlejších stránek přinášelo problémy. Právě kvůli této novince zavedlo konsorcium W3C nový typ souboru propojený s HTML dokumentem (W3schools.com, 2022a).

### 3.3.2.1 Propojení HTML a CSS

Pro vzájemné propojení HTML a CSS souborů je potřeba oba dokumenty propojit značkou `<link>` uvedenou v hlavičce HTML dokumentu (Janovský, 2020a):

```
<head>  
  <link rel="stylesheet" type="text/css" href="model.css">  
</head>
```

Obrázek 12 - Ukázka propojení CSS souboru s HTML dokumentem (Janovský, 2020a).

### 3.3.2.2 CSS metodiky

CSS kód je velmi jednoduchý na porozumění a psaní, ale při větším množství kódu není obtížné se v něm ztratit. Postupně vznikaly jednotlivé metodiky, které si kladly za cíl standardizovat postupy při vytváření a následné správě CSS souboru (Michálek, 2020a). V následujících podkapitolách jsou představeny nejpoužívanější metodiky BEM, OOCSS a SMACSS.

#### 3.3.2.2.1 BEM

BEM je zkratka pro Block Element Modifier, tedy prvky, kteří se navzájem upravují. Idejí této metodiky je rozdělit celé UI (User Interface, uživatelské rozhraní) do nezávislých bloků. Tím pádem se zjednoduší a zrychlí vývoj rozhraní a umožní opětovné použití kódu bez kopírování.

Blok je základní stavební kámen celé metodiky, je nezávislý a znovupoužitelný. Bloky se mohou navzájem a bez omezení vnořovat. V HTML se blok píše do atributu `class`, přičemž název by měl být výstižný pro danou sekci a vysvětlovat, co to je, nikoliv jak to vypadá (BEM Methodology, 2021):

```
<!-- Příklad správného použití bloku pro chybovou hlášku -->  
<div class="error"></div>
```

```
<!-- Příklad špatného použití bloku pro chybovou hlášku -->  
<div class="red-text"></div>
```

```
/* model.css */  
.error {  
  color: red;  
}  
.red-text {  
  color: red;  
}
```

Obrázek 13 - Metodika BEM – blok (BEM Methodology 2021).

Element je část bloku, která má být jiná než zbytek bloku. Konvence pojmenování je *block-name\_\_element-name*, kdy název bloku a název elementu jsou od sebe odděleny dvojitým podtržítkem *\_\_*. Stejně jako blok se může i element neomezeně vnořovat, na druhou stranu je vždy součástí bloku, tedy konvence *block\_\_element\_\_element* je špatně. Každý blok může mít nejvýše jeden element a zároveň každý element musí být součástí bloku (BEM Methodology, 2021).

```
<!--  
Správný zápis, struktura se drží konvence zápisu `block-name__element-name`  
-->  
<form class="search-form">  
  <div class="search-form_content">  
    <input class="search-form_input">  
  
    <button class="search-form_button">ZDE</button>  
  </div>  
</form>
```

Obrázek 14 - Metodika BEM – element (BEM Methodology 2021).

Poslední částí této metodiky je modifikátor, který definuje vzhled, stav nebo chování bloku nebo elementu. Stejně jako element nemůže být modifikátor použit samostatně, měl by pouze měnit vlastnosti již existujícího bloku nebo elementu. Existují dva typy modifikátorů, datový typ *boolean* nebo hodnotu *key-value*. Zápis modifikátoru v Boolean podobě je *block-name\_modifier-name* a *block-name\_\_element-name\_modifier-name* a záleží na přítomnosti elementu. Modifikátor může mít i svou hodnotu, v tom případě by zápis (bez elementu) vypadal takto: *block-name\_modifier-name\_modifier-value* (BEM Methodology, 2021).

```
<!-- Blok `search-form` má Boolean modifikátor `focused` -->  
<form class="search-form search-form_focused">  
  <input class="search-form_input">  
</form>  
  
<!-- Blok `search-form` obsahuje modifikátor `theme` s nastavenou hodnotou  
`islands` -->  
<form class="search-form search-form_theme_islands">  
  <input class="search-form_input">  
  
<!-- Element `button` obsahuje modifikátor `size` s hodnotou `m` -->  
<button class="search-form_button search-form_button_size_m">ZDE</button>  
</form>
```

Obrázek 15 - Metodika BEM – modifikátor (BEM Methodology 2021).

### 3.3.2.2.2 OOCSS

Následující metodika OOCSS (Object-Oriented CSS, objektově orientované CSS) byla představená z kraje roku 2009. Podobně jako BEM se i OOCSS se snaží dodržet princip *structure-skin*. Slovem *structure* je myšleno pozicování jednotlivých prvků na stránku: logo společnosti nahoru doleva, napravo vyhledávací box a pod oba prvky umístit navigaci se stoprocentní šířkou. *Skin* nastavuje téma stránky, například font písma, barevné rozhraní nebo pozadí stránky (Canziba, 2018) Třídy uvedené na Obrázek 16 jsou flexibilní, tudíž se dají použít na náš daný element i kdekoliv jinde (Sikora, 2016):

```
<!-- Flexibilní názvy tříd umožňují jejich použití i mimo div značku -->
<div class="product">
  <a class="product-name">Jméno produktu</a>
  <p class="product-desc">Popis produktu</p>
</div>

/* CSS soubor by pak vypadal následovně */
.product {}
.product-name {}
.product-desc {}
```

Obrázek 16 - Metodika OOCSS (Sikora, 2016).

### 3.3.2.2.3 SMACSS

V roce 2011, o 2 roky později než OOCSS, vyšla nová architektura pro psaní CSS SMACSS (Scalable and Modular Architecture CSS, škálovatelná a modulární architektura CSS). Primární přístup má stejný jako OOCSS, nicméně celý kód se dělí do pěti kategorií:

- Základní pravidla
- Pravidla rozložení
- Moduly
- Pravidla stavu
- Pravidla motivu

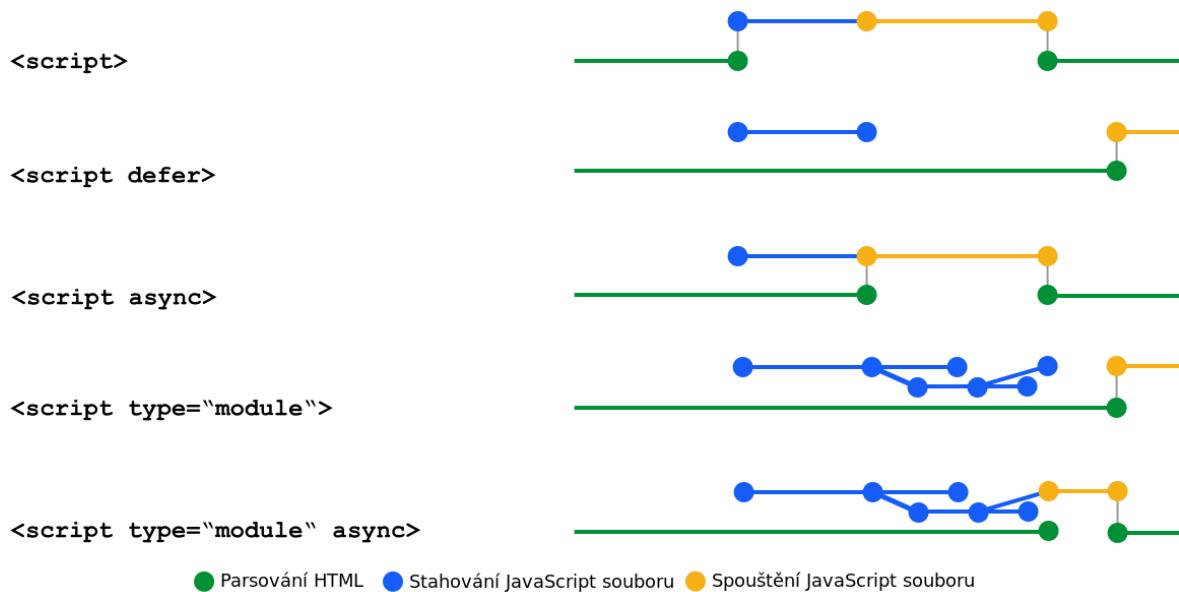
Společně s těmito pravidly vznikla i konvence pojmenování, které se vývojáři drží při vytváření CSS kódu. Pokud se stylizace týkala například zobrazení, bylo dobré to zaznamenat do názvu třídy pomocí prefixu *l-* nebo *layout-*, u jednotlivých komponent pak lze použít třídy *is-hidden*, *is-active* nebo *is-clicked*. Tato metodika zajišťuje čistý a přehledný kód s minimem nevýhod (Canziba, 2018). Každá z pěti kategorií má svůj CSS soubor, kam se píše kód. Pokud se tato metodika SMACSS spojí se Sass (rozšíření pro CSS), který má schopnost slučovat vícero souborů a importovat je jako jeden velký soubor, tzv. manifest (Watts, 2016).

### 3.3.3 JavaScript

Designovou stránku webu a užitečné funkce poskytuje skriptovací jazyk JavaScript. Od svého počátku v polovině 90. let minulého století, kdy sloužil jako rozšíření vydáváno společností Netscape, k jazyku Visual Basic společnosti Microsoft. V průběhu času začal nahrazovat zásuvný modul Flash. Zdrojový kód JavaScriptu běží ve webovém prohlížeči, ale existuje i v prostředí příkazového řádku.

Dalším způsobem, jak psát JavaScript, je přímo do HTML dokumentu. To lze vyřešit vložením elementu `<script></script>`, který se vkládá na konec stránky, aby se stihl načíst model DOM a prvotní obsah stránky. Pokud je potřeba vložit element do záhlaví HTML dokumentu (do elementu `<head>`), dá se načítání nastavit až po sestavení objektového modelu dokumentu skrz funkci `window.onload()` (Pehlivanian, Nguyen, 2014).

Může také nastat situace, kdy je potřeba umístit na stránku velký JavaScript soubor. V tomto případě nastává komplikace s umístěním takového souboru, protože umístěním značky `<script>` do hlavičky se zpomalí načítání stránky, takže zákazník uvidí jen bílou stránku bez jakéhokoliv obsahu, dokud se celý skript nenačte. Naopak přidání značky na konec stránky zajistí vykreslení obsahu zákazníkovi, nicméně stránka bude nefunkční z hlediska interakce ze strany klienta a prázdnými boxy, které by měly již mít nějaký obsah. Řešením tohoto problému může být třídění skriptů, tím pádem se méně důležité skripty zpozdí a načtou se až po načítání kompletní stránky s prioritními skripty (Burnham, 2017). K tomuto řešení dopomáhají HTML atributy `async`, `defer` a `type="module"`. Rozdíl mezi nimi je ve způsobu načítání JavaScript kódu. Na Obrázek 17 - Jak různé metody vložení souboru ovlivňují načasování stažení a provedení JavaScriptového kódu (Michálek, 2020b). jsou zobrazeny časové osy parsování (syntaktická analýza) HTML, stahování a spouštění JavaScript kódu při návštěvě uživatele.



Obrázek 17 - Jak různé metody vložení souboru ovlivňují načasování stažení a provedení JavaScriptového kódu (Michálek, 2020b).

Vložení skriptu přes jednoduchý tag `script` zastaví parsování, načte se skript a následně se obnoví parsování HTML. S atributem `defer` se odloží spouštění, pouze se skript stáhne současně s HTML kódem. Atribut `async` se oproti `defer` liší – spouští se totiž asynchronně. Ačkoliv se skript stahuje současně s procesem parsování, plné načtení skriptu parsování zastaví, vyhodnotí skript a až poté pokračuje v analýze HTML. Značka `script` s atributem `type="module"` má jako jediné garantováno pořadí provedení. Skript uvedený v této značce se stahuje, stejně jako `async`, souběžně s parsováním. Výhodou je spuštění skriptu až po dokončení analýzy HTML, kdy nedochází k přerušení (Michálek, 2020b).

### 3.3.3.1 Vložení Javascript kódu

Jak již bylo v přechozí kapitole popsáno, JavaScript kód se může zapisovat několika způsoby. Pokud je skript krátký, lze jej psát mezi `script` tag přímo do HTML souboru. Je však potřeba jej správně umístit na konec kódu. Dále je možné jej spojit značkou `link` v hlavičce stránky nebo jen přidat odkaz na kód v internetu (Pehlivanian, Nguyen, 2014). Na Obrázek 18 jsou demonstrovány tři příklady vložení JavaScript kódu do HTML souboru.

```

<!-- Externě napojený JS soubor přítomný v adresáři -->
<link rel="script" href="model.js">

<!-- Jednoduchý JavaScript kód zapsaný mezi značkami script -->
<button onclick="myFunction()">Try it</button>
<script>
    function myFunction() {
        alert("Hello world!");
    }
</script>

<!-- Kód je umístěn na webu, ale je možné jej převzít z CDN-->
<script
    src="https://code.jquery.com/jquery-3.6.0.js"
    integrity="sha256-H+K7U5CnX11h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
    crossorigin="anonymous">
</script>

```

Obrázek 18 - Vložení JavaScript kódu do HTML souboru (Pehlivanian, Nguyen, 2014).

V posledním případě na Obrázek 18 výše je kód převzat ze skupiny serverů CDN (Content Delivery Network). Tyto servery uchovávají duplicitní kopie dat, aby mohly rychle poskytovat data možným koncovým uživatelům. CDN se značně využívá pro distribuci CSS a JS souborů nebo knihoven, jako je například Bootstrap, jQuery atd. Poskytování statických knihoven přes CDN snižuje zátěž požadavků na servery spravující webové stránky. Zároveň díky bohatému rozmístění po celém světě je možné, že CDN servery jsou k uživatelům geograficky blíže než samotné servery poskytovatele, a tím pádem se snižuje latence (zeměpisná šířka je přímo úměrná k výši latence) (MDN Web Docs, 2021).

Atribut `integrity` obsahuje název šifrovacího algoritmu (v tomto případě sha-256) následovanou samotnou šifrou. Tato šifra, anglicky hash, umožňuje ověření zachování integrity a případně upozornit na kompromitaci souboru buď při převzetí z CDN nebo v průběhu přenosu. Každý soubor z CDN má svůj jedinečný hash, který se dá zjistit na stránce [srihash.org](https://srihash.org) nebo zadáním následujícího příkazu do příkazové řádky: `openssl dgst -sha384 -binary FILE.js | openssl base64 -A`. Tento příkaz lze využít i pro zabezpečení lokálních souborů (Padolsey, 2020).

Poslední atribut `crossorigin` musí být přítomen kvůli kontrole integrity. Pokud by tento atribut chyběl, prohlížeč bude ignorovat atribut `integrity` a přenos přijde o veškeré zabezpečení (SRI Hash Generator, 2022). Celý tento proces se nazývá Subresource Integrity, zkráceně SRI (Padolsey, 2020).



### 3.3.4 Frameworky

Software framework je struktura, jenž obsahuje již vytvořené funkce. Framework je svým způsobem i aplikace, jelikož zahrnuje téměř vše potřebné k vytvoření uživatelské aplikace.

Rozdíl mezi frameworkem a knihovnou je v takzvaném IoC (Inversion of Control). Pokud voláme metodu z knihovny, máme celý proces pod kontrolou. U frameworku je ovládaní invertované, tedy framework volá vývojářův kód, viz Obrázek 19 (Geeksforgeeks.com, 2020).

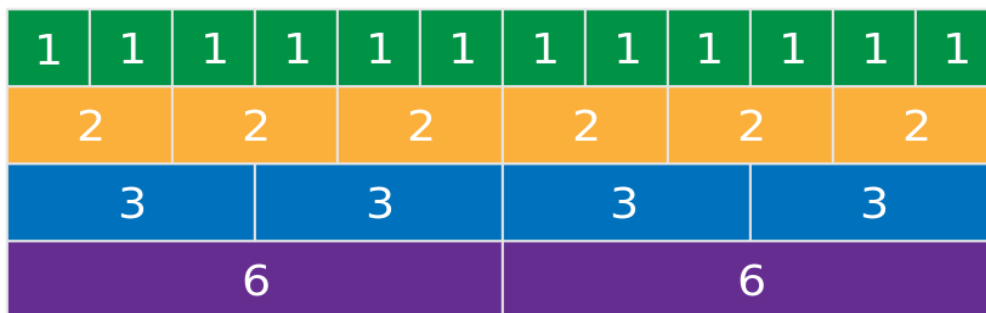


Obrázek 19 - Diagram vztahu framework - kód - knihovna (Geeksforgeeks, 2020).

#### 3.3.4.1 Twitter Bootstrap

V JavaScriptu a CSS je napsán framework Bootstrap vytvořený společností Twitter a vydávaný licencí MIT (Massachusetts Institute of Technology). Kód frameworku lze stáhnout přes webové stránky [getbootstrap.com](https://getbootstrap.com) nebo pomocí CDN (Matsinopoulos, 2020).

Celý framework využívá dvanáctisloupcový mřížkový systém, který využívá současně řádky i sloupce a je responzivní. Zjednodušeně řečeno, princip spočívá v definici třídy `container`, kde se bude vyskytovat 2D mřížka. Do každého kontejneru se poté přidá třída `row` ve značce `<div>` a do ní se přidá stejný tag, akorát s definicí počtu sloupců. Obrázek 20 demonstruje příklad 2D mřížky tohoto frameworku (Matsinopoulos, 2020):



Obrázek 20 - Mřížkový systém frameworku Bootstrap (Matsinopoulos, 2020).

Framework Bootstrap umožňuje používat různé třídy pro různé šířky zařízení. Je definováno pět klíčových bodů zlomu pro pět různých typů zařízení, viz Tabulka 1. Reference k tomuto rozdělení se udává přímo v názvu třídy, například třída `col-lg-12` odkazuje na zařízení se šířkou zařízení větší nebo rovno 992 pixelům a bude mít jeden sloupec přes celou šířku třídy *container* (pokud je obsažena).

Tabulka 1 - Body zlomu frameworku Bootstrap (Matsinopoulos, 2020).

Název	Symbol	Šířka displeje zařízení
Extra malé	xs	< 576 px
Malé	sm	≥ 576 px
Střední	md	≥ 768 px
Velké	lg	≥ 992 px
Extra velké	xl	≥ 1200 px

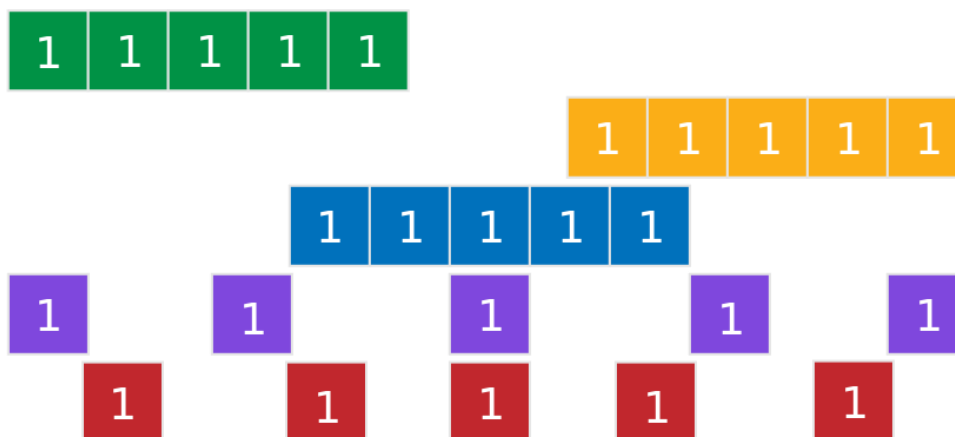
Třídy napojené na již dříve definované body zlomu pak vypadají takto:

- `col-1` až `col-12` pro extra malá zařízení,
- `col-sm-1` až `col-sm-12` pro malá zařízení,
- `col-md-1` až `col-md-12` pro střední zařízení,
- `col-lg-1` až `col-lg-12` pro for velká zařízení,
- `col-xl-1` až `col-xl-12` pro for extra velká zařízení.

Stěžejní je rozdíl mezi jednotlivými kategoriemi. Pro příklad, `col-sm-2` a `col-md-2` zabírají dva sloupce z dvanácti. Obsah ve třídě pro malá zařízení se zobrazí do dvou sloupců pro všechna zařízení širší než 576 pixelů, přičemž obsah ve druhé třídě se zobrazí do dvou sloupců pouze na zařízeních širších než 768 pixelů. V opačném případě se dva sloupce sloučí do jednoho a obsah se vykreslí nad sebe (Matsinopoulos, 2020).

Mimo systém zarovnání existují vně frameworku i jiné techniky, například zarovnávání položek na střed pomocí třídy `align-items-center`, lze však zarovnat i na vršek (`align-items-start`) a spodek (`align-items-end`) sloupce. Existuje i možnost zarovnání i jediného sloupce na různé strany. K tomu slouží třída `justify-content-*`, kde \* představuje množinu slov `start` (výchozí), `end`, `center`, `between` a `around`. Jakým stylem třída zarovnává položky je graficky znázorněno na Obrázek 21 -

Rozložení objektů podle tříd `justify-content-start`, `end`, `center`, `between` a `around` (Matsinopoulos, 2020). (Matsinopoulos, 2020).



Obrázek 21 - Rozložení objektů podle tříd `justify-content-start`, `end`, `center`, `between` a `around` (Matsinopoulos, 2020).

### 3.3.4.2 Tailwind CSS

Novějším trendem ve vývoji webů je framework Tailwind CSS, který je napsán, jak je vidno v názvu, v CSS a přináší s sebou nový způsob tvorby webu. Cílením na metodiku *utility-first* (volně přeloženo jako užitečnost na prvním místě) se snaží framework o přehledný komplexní kód postavený z omezené sady tříd. Například CSS třída `m-4` nastavuje vlastnost `margin` na hodnotu `1rem` (Rappin, 2021).

Instalace se na rozdíl od frameworku Bootstrap provádí příkazem `npm install -D tailwindcss`. Celkově je konfigurace frameworku složitější než v případě frameworku Bootstrap. Na druhou stranu má Tailwind CSS intuitivní konvenci pojmenování, je tedy pravděpodobnější, že vývojář se rychleji zorientuje v kódu s tímto frameworkem (Get Started with Tailwind CSS, 2022).

```
<!-- Fragment kódu s frameworkem Tailwind CSS -->
<div class="flex justify-center">
  <div class="mx-4 order-last">
    <%= image_pack_tag("media/images/music.svg", size: "100x100") %>
  </div>
  <div class="mx-4 self-center">
    <h1>Hello World</h1>
    <h2>How you doin'?</h2>
    <button>Learn More</button>
  </div>
</div>
```

Obrázek 22 - Fragment kódu s frameworkem Tailwind CSS (Rappin, 2021).

### 3.3.5 PHP

PHP je skriptovací programovací jazyk na straně serveru. Slouží pro programování dynamických stránek a webových aplikací ve formátech HTML, XHTML nebo WML. Mezi nejznámější stránky, které používají WordPress, se řadí i mj. Facebook, Wikipedie a z tuzemska např. Slevomat nebo Mall (Blažková, 2021).

Největší předností PHP je jeho cena, kompletní licence a využívání tohoto jazyka je zcela zdarma. Lze jej také používat na jakémkoliv operačním systému a při přepínání mezi nimi nedochází ke ztrátě funkčnosti PHP programu. S PHP se nejčastěji využívá webový server Apache (a dále také nginx, IIS (Microsoft Internet Information Server) nebo jakýkoliv jiný s podporou standardu CGI) spolu s několika typy databází, například MySQL, MongoDB, PostgreSQL, Oracle atp. Přes 200 milionů webových stránek běží s pomocí PHP, například sociální síť Facebook, encyklopedie Wikipedia nebo messenger Slack. Fragment kódu níže demonstruje triviální PHP kód, který se spustí po vyplnění textového pole a zmáčknutí tlačítka odeslání (Sklar, 2016):

```
<!-- model.html -->
<form method="POST" action="model.php">
  Your Name: <input type="text" name="user" />
  <br/>
  <button type="submit">Say Hello</button>
</form>

// model.php
<?php
print "Hello, ";
// Print what was submitted in the form parameter called 'user'
print $_POST['user'];
print "!";
?>
```

Obrázek 23 - Model využití PHP při vyplnění dotazníku (Sklar, 2016).

Začátek PHP kódu se značí <?php a ukončuje se značením ?>. Ve starších verzích PHP je možné použít i zkrácenou verzi značení <? a ?>, nicméně tento postup není doporučený a první uvedenou verzi značení podporuje každý moderní PHP engine. PHP engine vše okolo ignoruje a interpretuje pouze obsah vně těchto závorek. Do PHP kódu lze psát i samotný HTML kód díky syntaxi *here document*, místo dvou soborů stačí tedy jen jeden .php soubor. Toto funguje i *vice versa*, v tom případě je možné mít i PHP kód uvnitř HTML značky (Sklar, 2016).

Důležitým aspektem jazyka PHP je jeho case-insensitive vlastnost (malá a velká písmena jsou zaměnitelná), například obě funkce `PRINT number_format(98)`

a `print NUMBER_format(98)` vrátí stejný výsledek. Velkou část jazyka PHP tvoří proměnné. Pojmenování proměnných má svá pravidla, mezi povolené znaky patří malé a velké znaky latinky, čísla (nesmí na číslo začínat) a každý další znak přeložitelný stylem kódování *UTF-8*. Dle těchto pravidel lze pojmenovat proměnnou \$😊, na druhou stranu nelze pojmenovat proměnnou například \$bang! (Sklar, 2016).

## 3.4 Úložiště a hosting

### 3.4.1 GitHub

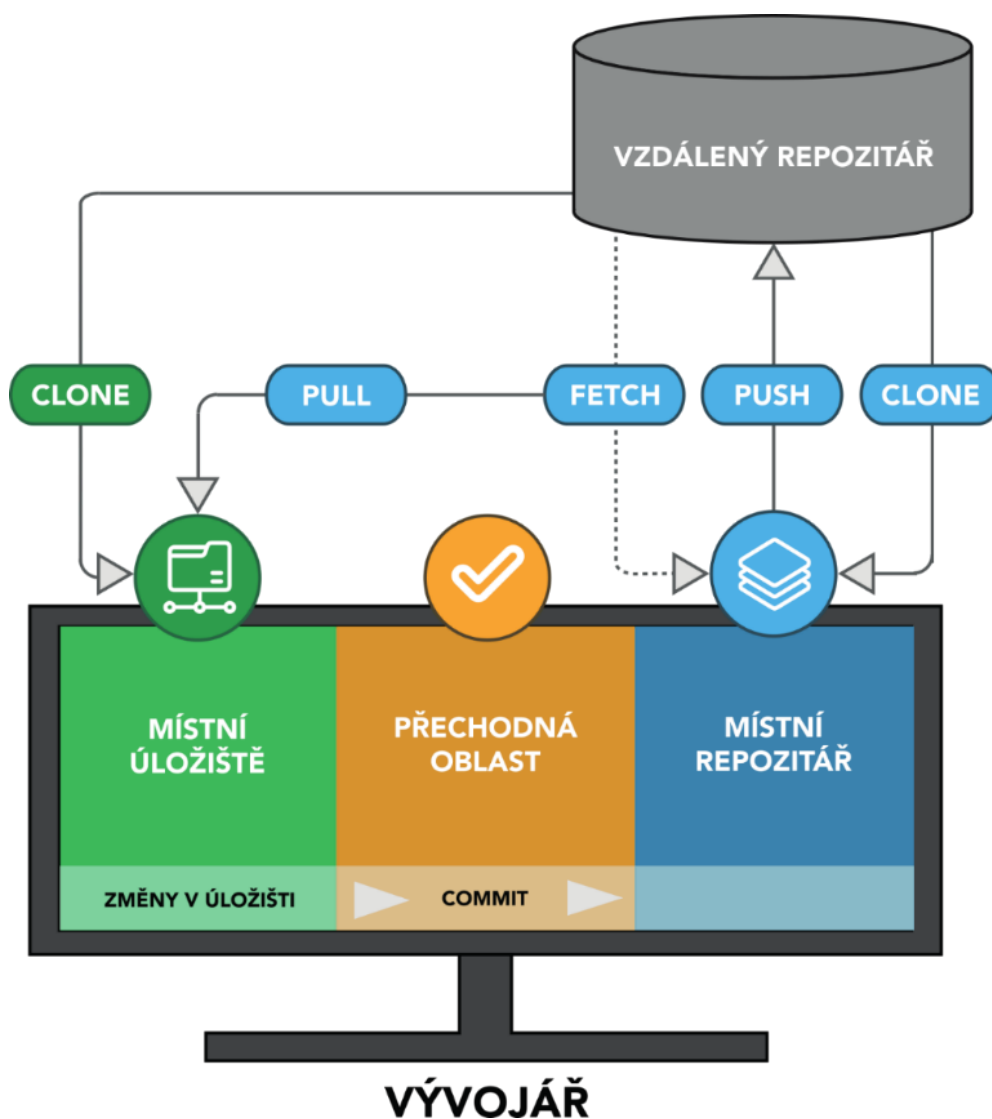
GitHub je webová stránka, která hostí repozitáře více než 73 milionu vývojářů. Zároveň je poskytovatelem internetového hostingu, viz 3.4.3. Vývojářům nabízí distribuované verzování skrz nástroj Git (Github.com, 2022). Pro založení nového repozitáře je potřeba být zaregistrovaný na stránce GitHub.com. Schéma URL adresy každého založeného repozitáře vypadá takto: <https://github.com/<přezdívk>/<repozitář>>. Po založení nového repozitáře se zobrazí Git příkazy, které navedou uživatele na nahrání prvního commit záznamu a jeho následné nahrání na daný repozitář. Každý commit obsahuje mimo unikátního hash řetězce i informace o autorovi, datum sestavení záznamu a komentář. Veškeré potřebné příkazy je možné najít v Tabulka 2 a detailnější postup v kapitole Verzovací nástroj Git (Pipinellis, 2018). Mimo založení vlastního repozitáře je možné i spolupracovat na cizím repozitáři přes tzv. pull request. Tyto žádosti umožňují ukázat ostatním vývojářům změny, které byly nahrané do repozitáře. Jakmile se tato žádost objeví, mohou všichni komentovat chystané změny, včetně všech commit příkazů, které přibyly dodatečně. Jakmile se pull request sloučí s větví repozitáře, tato žádost zanikne a stane se součástí projektu (Github.com, 2022). Jak stáhnout Git, včetně instalace a základních příkazů je popsáno v další podkapitole 3.4.2

### 3.4.2 Verzovací nástroj Git

Software je možné stáhnout na webové stránce <https://git-scm.com/downloads>, kde je nutné vybrat distribuci. V balíčku je obsažen Git GUI a Git Bash, což jsou aplikace, přes které se zadávají příkazy. Oba zmíněné prostředky lze spustit v kořenové složce projektu, aby příkazy fungovaly rekurzivně na celý projekt (Loeliger, 2012).

Dále je potřeba porozumět terminologii Git. Příkaz `git commit` provede tzv. zápis, kdy Git uloží změny v souborech nacházejících se v oblasti změn připravených k nahrání

a uloží jej trvale do adresáře Git. S commitem souvisí příkaz `push`, který vezme všechny commity a nahraje je na server GitHub a následně se zobrazí i na stránce repozitáře projektu. V případě potřeby lze stáhnout i kompletní kopii repozitáře, včetně historie commit záznamů, přes příkaz `git clone <URL>`. Všechny příkazy se zadávají prostřednictvím e-mailové adresy spojené s GitHub účtem, proto je potřeba nastavit globální jméno a e-mail příkazem `git config` (Loeliger, 2012). Na obrázku níže jsou vyobrazené základní příkazy pro pohyb adresáře mezi lokálním a vzdáleným repozitářem.



Obrázek 24 - Funkcionalita verzovacího systému Git (Prajapati, 2019).

V následující tabulce je popsáno devět příkazů, které stačí k základnímu využívání verzovacího nástroje Git (Tagliaferri, 2021):

Tabulka 2 - Základní příkazy verzovacího nástroje Git (Tagliaferri, 2021).

<b>Příkaz</b>	<b>Popis</b>
<code>git init</code>	Vytvoří prázdný Git repozitář v podobě skryté <i>.git</i> složky
<code>git add -A</code>	Přidá (rekurzivně) veškerý obsah složky do repozitáře
<code>git commit -m "text"</code>	Vytvoří commit se všemi soubory, které byly přidány. Argument <i>-m</i> značí interní zprávu commitu, která slouží k popisu commitu (co se změnilo oproti minulé verzi).
<code>git remote add origin URL</code>	Propojí aktuální repozitář v počítači s repozitářem na GitHubu
<code>git remote -v</code>	Zobrazí URL adresy vzdáleného repozitáře
<code>git push -f origin master</code>	Nahraje všechny commity na GitHub do zvoleného repozitáře a použije poslední commit jako aktuální verzi
<code>git pull origin master</code>	Aktualizuje repozitář v počítači
<code>git clone URL</code>	Naklonuje jakýkoliv veřejný repozitář do místa v počítači, odkud je příkaz spouštěn
<code>git revert commitHash</code>	Zruší nahraný commit a použije předešlý commit jako aktuální (nejnovější) verzi

### 3.4.3 GitHub Pages

GitHub nabízí i online hosting repozitáře obsahující HTML, CSS a JS soubory zvaný GitHub Pages. Ten vždy vezme aktuální commit a do pár minut aktualizuje i uživatelsky přístupnou verzi, i když je repozitář skrytý (Github.com, 2022).

Výhodou je přítomnost HTTPS protokolu, který je na rozdíl od svého předchůdce HTTP šifrovaný (písmeno S ve zkratce značí anglické slovo secured – zabezpečený). Dalším

plus je i nastavení vlastní domény. Obecnou adresou GitHub Pages je <https://<přezdívk>.github.io> nebo <https://<organizace>.github.io> (Github.com, 2022).

Co se týče nevýhod služby Pages, tak největší z nich je absence verzování. Vždy je dostupná jen aktuální verze, která se při aktualizaci repozitáře přepíše. Staré verze se dají zobrazit jen stáhnutím určitého commitu jako .zip soubor nebo příkazem `git checkout <CommitHash>` (v tomto případě je potřeba první stáhnout celý repozitář se všemi commit záznamy přes příkaz `git clone`) (Longair, 2011).

Další nevýhodou hostingu je absence jazyků na straně serveru, tedy PHP, Ruby nebo Python (Github.com, 2022). Kvůli tomuto problému je hosting GitHub Pages pouze dočasným řešením a bude sloužit zejména pro optimalizaci SEO.

#### 3.4.3.1 Jekyll

Pokud je vývojář zároveň i administrátorem repozitáře, může mimo založení GitHub Pages vytvořit i stránku přes generátor stránek Jekyll. K tomu je potřeba nainstalovat ještě jazyk Ruby a prostředí (pro Ruby) Bundler. Tím se stává používání tohoto typu webu komplikovanější, a proto se v současnosti (alespoň na méně náročné weby) nepoužívá (Github.com).

#### 3.4.3.2 Hugo

Hugo je generátor statického webu, který vyšel v roce 2013. V porovnání s Jekyll generátorem je Hugo v rámci velikosti podstatně menší, na druhou stranu si zakládá na rychlosti a flexibilitě. Právě díky rychlosti se Hugo drží na prvních příčkách používání. Je napsán v jazyce Go, který vyvinul americký korporát Google. Má otevřený kód, každý tedy může přispět na jeho vylepšení (CloudCannon 2021).

#### 3.4.3.3 Porovnání Jekyll a Hugo

Pro demonstraci obou generátorů je na obrázcích X a Y zobrazen podmíněný příkaz (anglicky if statement):



```
{% if page.is_post and page.hascaniuse %}

    {% komentář %}
    Provede příkaz(y), pokud je podmínka pravdivá
    {% konec komentáře %}

{% endif %}
```

*Obrázek 25 - Demonstrace podmíněného příkazu pomocí generátoru Jekyll (CloudCannon, 2021).*

```
{{ if (and (eq .Type "blog") (isset .Params "hascaniuse")) }}
```

*<--! Provede příkaz(y), pokud je podmínka pravdivá -->*

```
{{ end }}
```

*Obrázek 26 - Demonstrace podmíněného příkazu pomocí generátoru Hugo (CloudCannon, 2021).*

Ačkoliv se na první pohled zdá, že příkaz v Jekyll (Obrázek 26) je napsaný čitelněji, statistika (Hugo Usage Statistics, 2020 & Jekyll Usage Statistics, 2020) tvrdí opak: Hugo vede v počtu webových stránek, v počtu hvězd na GitHubu i ve sledujících na platformě Twitter (CloudCannon, 2021).

## 4 Vlastní práce

### 4.1 Požadavky zadavatele

Hlavním požadavkem na základě hovoru s majiteli společnosti je mít nejpřehlednější reference společnosti. Firma touto cestou může reprezentovat své minulé projekty a potenciální klient má možnost si projekty zobrazit. Dalším důležitým aspektem je kontaktní formulář, který funguje na bázi odesílání e-mailových zpráv s pomocí jazyka PHP. Poslední důležitá sekce je zobrazení certifikátů pro demonstraci kvalifikace společnosti a jejích zaměstnanců.

#### 4.1.1 Funkční požadavky

Pro splnění cílů práce byly definovány tyto funkční požadavky:

- Přesměrování uživatele z navigace do konkrétní sekce
- Zobrazení miniatur certifikátů
- Filtrování referencí
- Miniatury certifikátů a referencí budou zvětšitelné
- Interaktivní mapa
- Kontaktní údaje budou umožňovat *call-to-action*
- Možnost kontaktovat stavební firmu
- Upozornění uživatele na nevyplněné pole kontaktního formuláře
- Informování o úspěšnosti odeslané poptávky
- Možnost vrácení uživatele zpět na začátek webu

#### 4.1.2 Nefunkční požadavky

Nová webová stránka bude splňovat tyto následující nefunkční požadavky:

- Responzivita webu
- Rychlé načítání webu
- Optimalizace SEO
- Uživatelská přívětivost

## 4.2 Analýza řešení

Na základě analýzy byl vybrán vhodný framework pro tuto práci. Těchto frameworků existuje velké množství a každý se liší v několika faktorech. Pro framework byly vybrány tato hodnotící kritéria: velikost balíčku, čistý kód, přehlednost dokumentace, popularita a četnost používání. Následující odstavce srovnávají dva frameworky – Bootstrap a Tailwind CSS, oba byly stručně popsány v kapitolách 3.3.4.1 a 3.3.4.2.

Oba frameworky mají relativně dobře zpracovanou dokumentaci, u obou lze vyhledávat klíčová slova a většinou najde správný výsledek. Obě dokumentace mají také příklady využití, Tailwind CSS pak demonstruje i kód u každé třídy. Jak Bootstrap, tak i Tailwind CSS dokáží udržovat při svém používání čistý kód. Ve zbytku kritérií se oba frameworky méně či více rozcházejí.

Bootstrap má za sebou bohatou historii. Vznikl již v roce 2011 na mezinárodní soutěži, jež hostovala společnost Twitter. Tento framework je open-source a je navržen pro vytváření lepších a rychlejších webových stránek jednoduchými postupy. Obsahuje HTML, CSS i JavaScript. Bootstrap lze využít s jakýmkoliv server-side technologiemi (Java, PHP atp.). Tento framework také obsahuje šablony pro spoustu webových komponent: tlačítka, typografie, formuláře, ikony, navigace a jiné. Celý framework je objektově orientovaný. Velikost celého balíčku je necelých 60 kB (kilobytů) a při minifikaci (přepisování kódu a mazání nepotřebných prvků za účelem zmenšení velikosti souborů) a kompresi se celý balíček zmenší až na 18 kB. Bootstrap je celkově populárnější, jelikož se používá již přes 10 let.

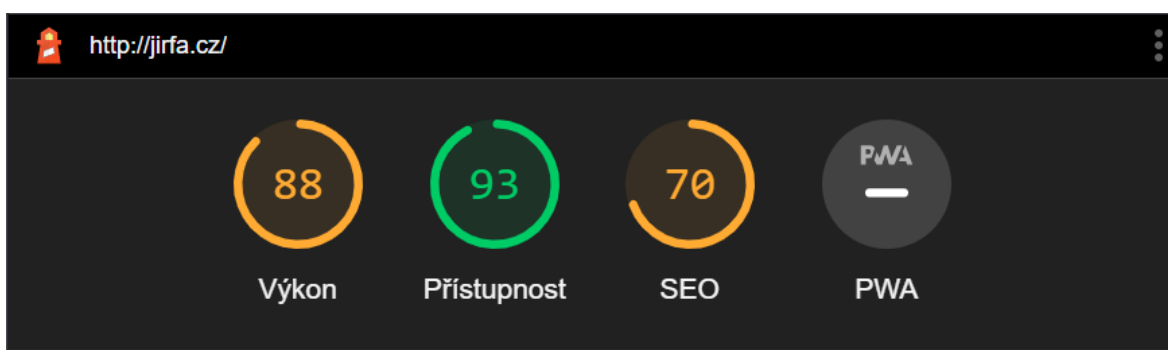
Tailwind CSS patří mezi novější frameworky, a proto jeho popularita teprve poroste (první verze vznikla 1. listopadu 2017). Nabízí širokou škálu přizpůsobitelnosti a flexibility. Na stejné bázi jako je Tailwind CSS vzniklo již několik frameworků, tento je ze všech velikostně nejmenší a jeden z nejpoblárnějších. V rámci velikosti balíčku velmi zaostává oproti předchozímu frameworku, výchozí velikost se totiž blíží k 0,5 MB a na síti 3G by stahování trvalo tři vteřiny. Po kompresi je velikost stále velká, a to 135 kB.

### 4.2.1 Závěr

Oba frameworky, Bootstrap i Tailwind CSS, mají své výhody i nevýhody. Vzhledem ke zkušenostem autora práce, velikosti balíčku a popularitě byl vybrán framework Bootstrap jako vhodnější varianta pro tvorbu nové webové stránky.

### 4.3 Recenze původního webu

V rámci analýzy cílů práce byla provedena recenze starého webu. Aktuální verze webové stránky není ani responzivní, ani postavená přístupem mobile-first. Web byl zveřejněn v roce 2013, kdy byl přístup na internet přes telefony velmi výjimečný. Nástroj Lighthouse zobrazuje u kritéria *Výkon* hodnotu 88 ze sta. Hůře dopadlo kritérium SEO (70) i optimalizace PWA (1/7), viz Obrázek 27 - Analýza původního webu nástrojem Lighthouse (Tools for Web Developers, 2022).



Obrázek 27 - Analýza původního webu nástrojem Lighthouse (Tools for Web Developers, 2022).

#### 4.3.1 Nedostatky starého webu

Hlavním nedostatkem webové stránky je absence využívání CDN serverů. Tím je zapříčiněná nižší rychlost načítání a potažmo i SEO výsledky. Kvůli špatným výsledkům pak může být zredukovaný dosah na potenciální klienty firmy.

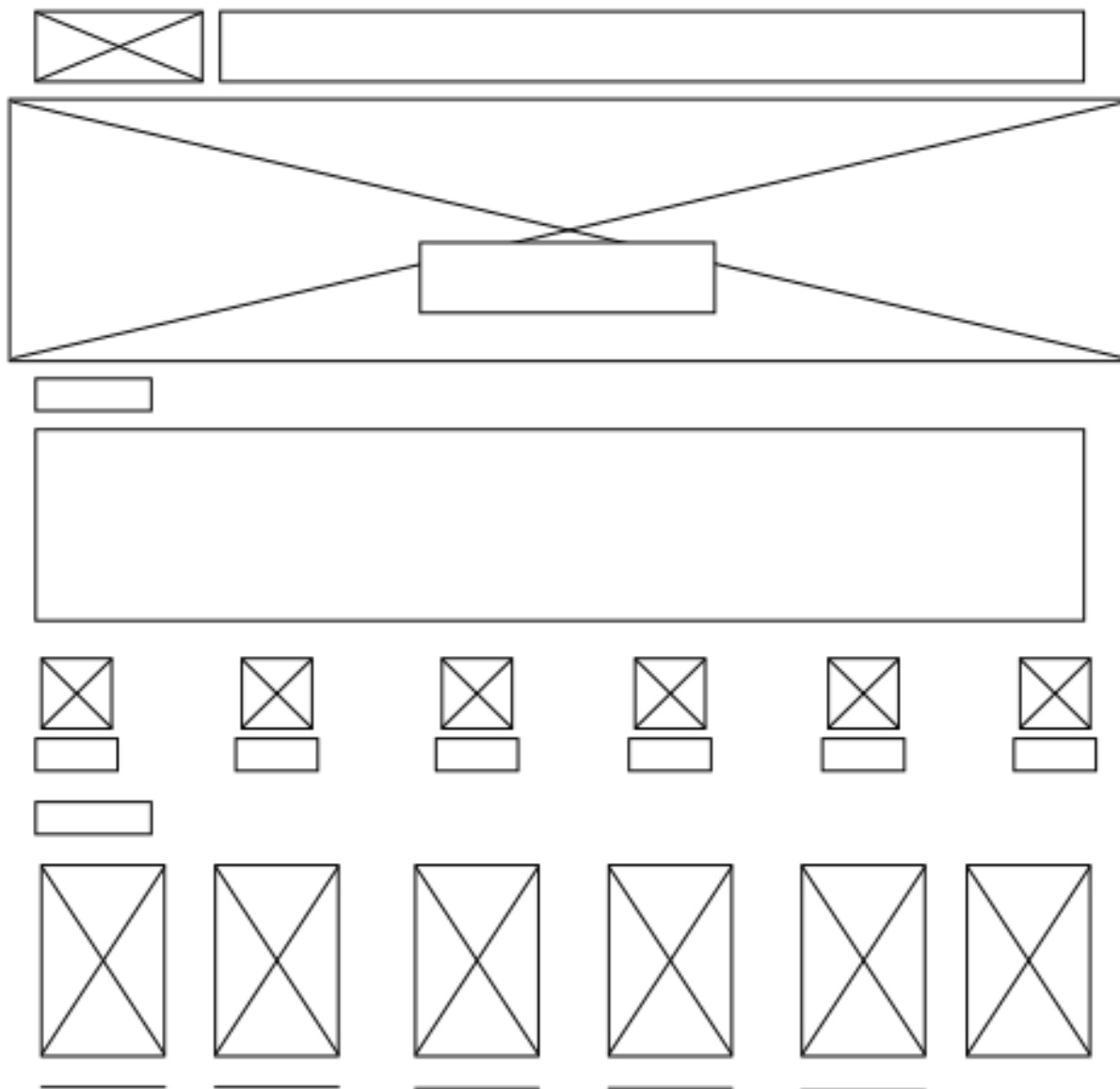
Stránka také postrádá šifrovanou komunikaci mezi klientem a serverem. Například prohlížeč Google Chrome upozorňuje při vstupu na stránku s prefixem http://, protože se jedná o bezpečnostní riziko. Existují případy, kdy prohlížeč může zcela zablokovat přístup na danou stránku. Nezabezpečená stránka také snižuje SEO skóre.

### 4.4 Tvorba nové stránky

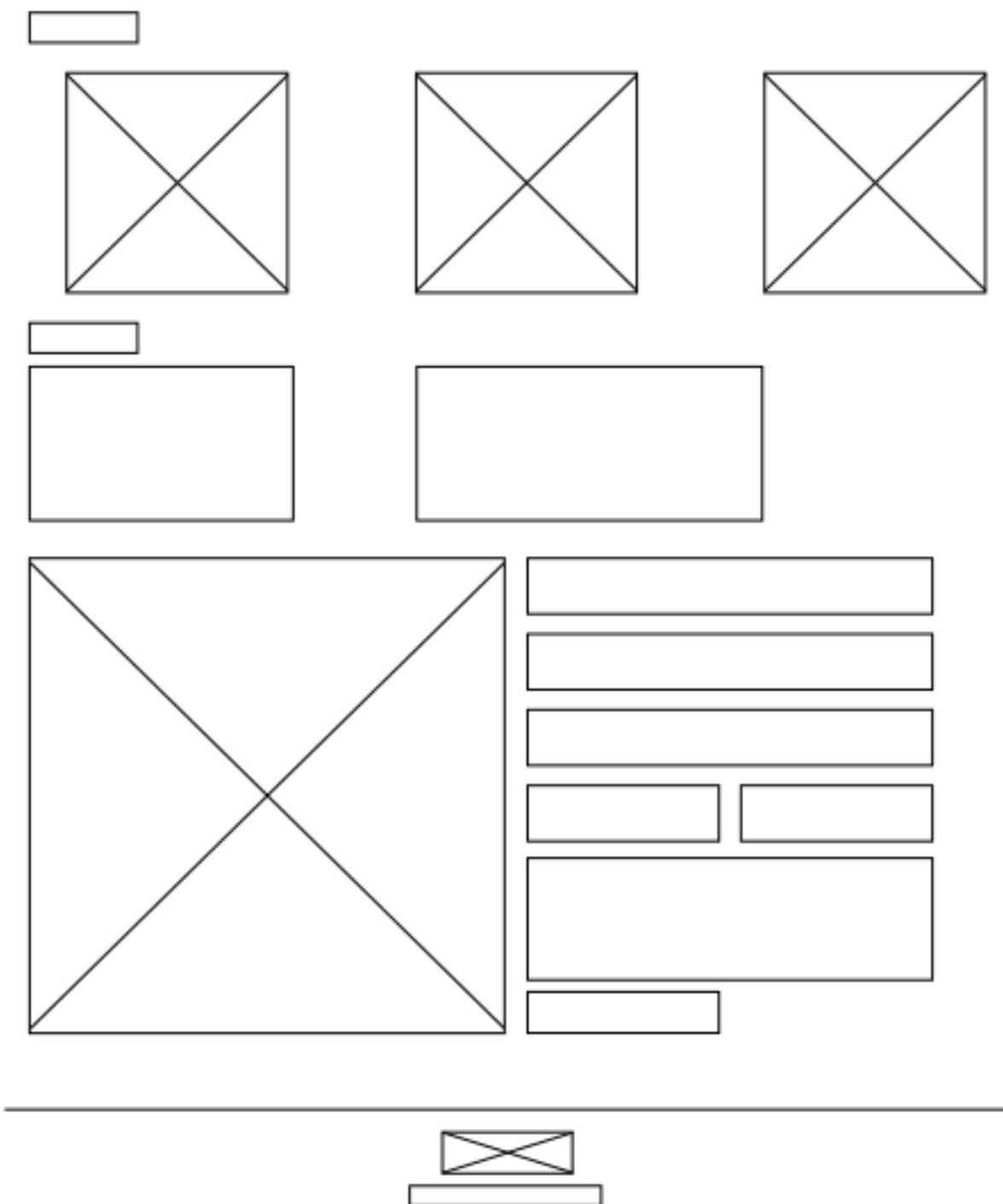
#### 4.4.1 Návrh struktury

Pro znázornění konceptu zadavatelům byl vytvořen drátěný model (wireframe) pro vyřešení nesrovnalostí s tvorbou nové stránky. První model představuje náhled prvních čtyř sekcí: Navigace, hlavička stránky, úvod s kategoriemi referencí a miniatury

certifikátů. Druhý model zobrazuje zbylé sekce stránky: položky referencí, údaje o společnosti, kontaktní formulář, mapa sídla a patička webu.



Obrázek 28 - Snímek drátěného modelu.



Obrázek 29 - Snímek drátěného modelu.

#### 4.4.2 Metodika

Metodika tvorby aktualizované webové stránky je nastavena formou popisu jednotlivých sekcí, včetně ukázky kódu a vysvětlení vlastního řešení. U popisu bude přidán odkaz na *commit*, který čtenáře odkáže na aktuálně popisovanou verzi kódu. V případě

nejasností lze repozitář stáhnout a aktuální commit dohledat přes příkazovou řádku a příkazem `git checkout <commit>`.

## 4.5 Popis tvorby vytyčených cílů

V této kapitole jsou popsány tři stěžejní cíle, které byly zadané od majitelů společnosti. Vedlejší cíle jsou poté popsány hromadně v kapitole 4.5.4. Webová stránka je řazena do třech komplementárních barev – bílé (v pozadí), červené (primární barva) a modré (sekundární barva). Textový obsah je doplněn o grafické ikony z webové stránky Font Awesome, které se stahují externě z GitHub repozitáře. Hosting poskytuje taktéž GitHub se službou Pages, nicméně finální poskytovatel hostingu zůstane stejný, jelikož služba GitHub Pages nepovoluje dynamický obsah a tím pádem je omezena funkčnost kontaktního formuláře.

### 4.5.1 Reference

Část stránky s referencemi byla vytyčena jako primární cíl této práce. Reference na staré webové stránce byly rozděleny pouze do základních kategorií a veškeré související obrázky byly smíchané dohromady. Tento způsob byl chaotický a v nové verzi jsou reference rozříděny dvakrát: podle kategorie, po vzoru starého webu, a potom podle projektu. Tento filtr je přehlednější a návštěvník webu si může prohlédnout určitou referenci klikem na náhledový obrázek. Při přejetí myši se obrázek překryje červenou vrstvou informující o typu a místě projektu. Na mobilních zařízeních je tento popis skryt (na mobilních zařízeních nelze „najat myši“ na obrázek). Tento problém byl vyřešen přidáním titulku reference do knihovny Lightbox2, jež zobrazuje vyšší rozlišení obrázků na obou verzích webu. Na novém webu byly ponechány stejné kategorie jako v předešlé verzi a ke každé kategorii je přiřazena minimálně jedna reference. Sekce využívá mřížkový systém a je rozdělena na jeden sloupec pro telefony, dva pro tablety a tři sloupce pro zařízení se šířkou větší než 992 pixelů. Toto zobrazení je nastavené třídami viz Obrázek 30.

Na Obrázek 32 a Obrázek 33 je zobrazena reference ve výchozím stavu a ve stavu aktivované pseudo-třídy `:hover`, tedy pokud návštěvník přejede přes referenci myší. Mimo nastavení zobrazení je u každé reference nastavený filtr přes třídy `filterClass` a danou kategorii. K těmto třídám náleží i JavaScript kód, který filtruje reference.

```
<div class="my-5 col-12 col-md-6 col-lg-4 filterClass pergoly">  
  <!-- Obsah reference -->  
</div>
```

Obrázek 30 - Nastavení tříd pro upravené zobrazení reference.

Kompletní kód pro zobrazení jedné reference je zobrazen níže. Každá reference obsahuje náhled ve značce `img`, detaily reference v seznamu `ul` a proklik na galerii přes odkazovou značku `a`. Každá reference, miniatura i související obrázky, je propojena atributem `data-lightbox`. Zobrazovaný název reference zajišťuje atribut `data-title`, který se hodí při mobilním zobrazení. Pro zlepšení výkonu stránky je miniatura optimalizována na nižší rozlišení, aby bylo načítání stránky rychlejší.

```
<div class="my-5 col-12 col-md-6 col-lg-4 filterClass pergoly">
  <div class="reference-overlay">
    <a href="assets/images/reference/pergoly/07.jpeg"
      data-lightbox="reference-pergoly-3"
      data-title="Domov důchodců Praha 4"
      title="Domov důchodců Praha 4">
      <div class="reference-desc hide w-100 h-100">
        <ul class="lst-none p-0 m-auto">
          <li class="fw-bold reference-desc-title">
            Domov důchodců<br>Praha 4</li>
          <li class="mt-5"><i class="fas fa-search-plus fs-2"></i></li>
          <li class="reference-desc-menu">Zobrazit</li>
        </ul>
      </div>
      
    </a>
    <a href="assets/images/reference/pergoly/08.jpeg"
      data-lightbox="reference-pergoly-3"
      data-title="Domov důchodců Praha 4" title="Domov důchodců Praha 4"></a>
    <a href="assets/images/reference/pergoly/09.jpeg"
      data-lightbox="reference-pergoly-3"
      data-title="Domov důchodců Praha 4" title="Domov důchodců Praha 4"></a>
    <a href="assets/images/reference/pergoly/10.jpeg"
      data-lightbox="reference-pergoly-3"
      data-title="Domov důchodců Praha 4" title="Domov důchodců Praha 4"></a>
  </div>
</div>
```

Obrázek 31 - Kompletní kód zobrazení jedné reference.



Obrázek 32 - Zobrazení reference ve výchozím stavu.



Obrázek 33 - Zobrazení reference při přejetí myší.



Jelikož je reference zobrazována značkou `<img>`, bylo potřeba změnit kurzor myši na `pointer`, aby bylo zřejmé, že je obrázek interaktivní. Po kliknutí na miniaturu se zobrazí optimalizovaná knihovna Lightbox2 s galerií fotografií reference.

Do referencí je taktéž implementována funkce vymazání aplikovaného filtru, která po stisknutí uživatelem vrátí filtrování do výchozí hodnoty (zobrazení všech položek). Reference se také filtrují výběrem typu reference z navigačního pole. Pokud se uživatel nachází mimo pole referencí, atribut `href` ve značce `a` (`anchor`) návštěvníka přesune na oblast s referencemi.

#### 4.5.2 Kontaktní formulář

Kontaktní formulář slouží k propojení mezi jednateli společnosti a potenciálními klienty. V této sekci se nachází interaktivní mapa se sídlem společnosti a kontaktní formulář pro vyplnění celého jména, e-mailové adresy pro možnost odepsání, telefonního čísla pro osobnější komunikaci, nepovinných prvků města, poštovního směrovacího čísla a textu zprávy. Každé textové pole má i svůj `placeholder`. Jedná se o nápovědu, jak by měl text v daném poli přibližně vypadat. Zároveň textové pole pro telefonní číslo a PSČ upravené, aby lidé na telefonu měli k přístupu pouze numerickou klávesnici. Komunikace je řešena programovacím jazykem PHP společně s jazykem JavaScript a přenos obsahu formuláře je nastaven na bázi SMTP serveru. Data z každého textového pole se uloží do proměnných v PHP souboru `action.php`. Ve formuláři je v JavaScriptu nastavený `eventListener` na tlačítko *Odeslat e-mail*, který slouží k tomu, aby byl proveden určitý skript, poté co uživatel odešle formulář. Tento skript je asynchronní, což znamená, že dojde k odeslání požadavku na PHP server, který provede odeslání e-mailu s parametry uživatele. Některé prvky formuláře jsou povinné, tudíž jejich uložení do proměnných je ošetřeno podmínkou `IF` a operandem `AND`. Do dalších proměnných se ukládá i tělo e-mailové zprávy, jenž se bude odesílat v případě správného vyplnění datových polí uživatelem. Celý proces je vykonáván v blocích `try{} a catch{}` , aby případné chyby byly zachyceny a vypsalý se do vývojářské konzole. Úspěšné i neúspěšné odeslání e-mailu zobrazuje uživateli hlášku informující o stavu zprávy.

Pokud uživatel přistupuje na stránku skrz mobilní telefon, tak se uživateli skryje miniatura mapy, nicméně mapu lze zobrazit i přes substituční tlačítko *Mapa* umístěné pod adresou společnosti. V případě potřeby je pro klienty zobrazeno Identifikační číslo osoby.

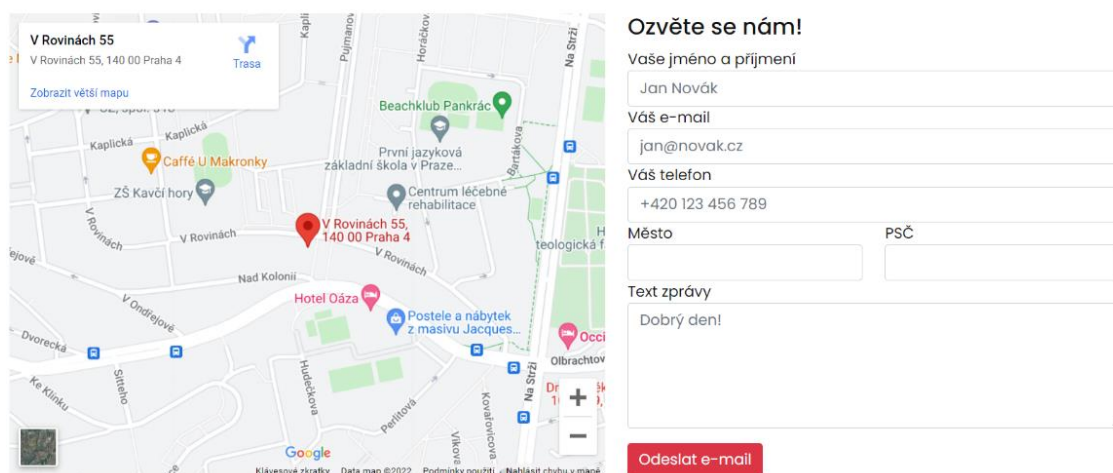
Jednatelé firmy jsou vypsáni v pravé části obrazovky a jejich telefonní čísla jsou interaktivní – dá se kliknutím na telefonní číslo rovnou zavolat.

## Kontakty

**JIRFA, s.r.o.**  
V Rovinách 55  
140 00  
Praha 4  
IČO: 27143287  
[jirfa.cz](http://jirfa.cz)

### Jednatelé:

Jiří Šteker  
[+420 606 094 834](tel:+420606094834)  
František Doubek  
[+420 603 269 054](tel:+420603269054)



**Ozvěte se nám!**  
Vaše jméno a příjmení  
  
Váš e-mail  
  
Váš telefon  
  
Město  PSČ   
Text zprávy

Obrázek 34 - Náhled kontaktního formuláře v desktop prostředí.

Obě sekce s adresou i kontaktem na jednatele společnosti využívají třídu `col-6`, tedy na všech zařízeních zabírají 50% šířky zařízení. V případě nejmenších zařízení je ikona sluchátka telefonu odstraněna, aby se telefonní číslo nezalamovalo na více řádků. Stejně rozložení platí i pro mapu a kontaktní formulář, akorát pro menší zobrazení se mapa skryje a kontaktní formulář využívá 100% šířky vykreslení.

### 4.5.3 Certifikáty

Sekce certifikátů informuje o kvalifikaci společnosti. Funguje velmi podobně jako sekce referencí. Pro testování funkčnosti byl vytvořen v programu Adobe Illustrator tzv. *mockup*, jenž nahradil v této fázi skutečné certifikáty. Každý certifikát lze rozkliknout a prohlédnout si jej. Náhled certifikátu je optimalizovaný pro rychlé načítání stránky. Tyto miniatury jsou tedy rozmazané, aby měly co nejmenší velikost. Na zobrazení certifikátů je uplatněn mřížkový systém frameworku Bootstrap, viz Obrázek 35 - Kód pro zobrazení certifikátu společnosti. Výchozí nastavení jsou dva certifikáty na jeden řádek, pouze u malých zařízení jsou zobrazeny tři certifikáty na řádek a na velkých zařízeních je na řádek

zobrazeno certifikátů šest. Další Obrázek 36 - Náhled sekce certifikátů. je převzat z finální verze webové stránky společnosti.

```
<div class="col-6 col-md-4 col-lg-2 mb-2">
  <a href="assets/images/certificates/bramac1.jpg" data-
lightbox="certifikat" data-title="Certifikát">
    
  <hr class="certificate-hr">
</a>
</div>
```

Obrázek 35 - Kód pro zobrazení certifikátu společnosti.

## Certifikáty



Obrázek 36 - Náhled sekce certifikátů.

### 4.5.4 Ostatní sekce

Sekce navigace využívá šablonu Bootstrap frameworku. Do navigace je implementována i rozevírací nabídka na jednotlivé sekce referencí. V případě užších



Obrázek 37 - Zobrazení navigace přes navbar-toggler.

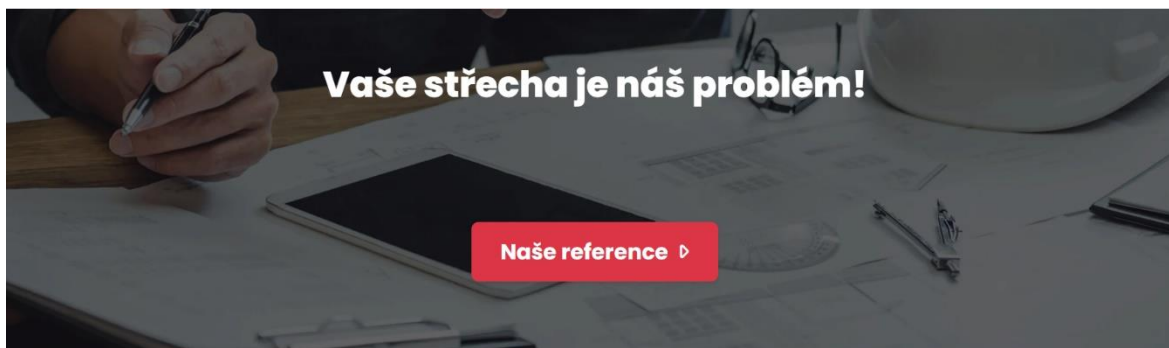


Obrázek 38 - Zobrazení navigace v desktop prostředí.

zařízení je navigace skryta za přepínač nastavený třídou navbar-toggler. Tuto funkci využívá mnoho webů, aby zachovaly integritu webu. V navigaci je na levé straně zobrazeno logo firmy a kliknutím na něj se uživatel dostane na vršek stránky. K tomu samému přesunu slouží i tlačítko v pravém dolním rohu stránky, které se zobrazí, pokud se návštěvník stránky nachází pod horním okrajem stránky.

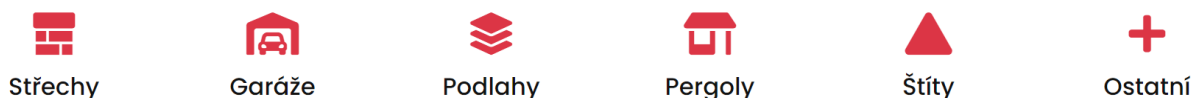
Sekce hlavičky stránky obsahuje upravený obrázek, motto firmy a tlačítko odkazující na reference společnosti. Obrázek na pozadí je vložen přes CSS vlastnost background-image. Použité pozadí muselo být kvůli snížení kontrastu ztmaveno, aby

bylo možné použití bílé barvy písma. Tímto způsobem lépe vyniknou oba prvky (motto i tlačítko). V tlačítku referencí se nachází ikona Font Awesome, která se mění při najetí myši na tlačítko ze šipky doprava na šipku dolů.



Obrázek 39 - Hlavička stránky.

Následující sekce obsahuje informace o společnosti. Texty jsou vloženy do značky `p` a důležité informace jsou vymezeny do značky `strong`, který daný text přemění na tučný. Tuto vlastnost umí i značka `b`, nicméně syntakticky správně je první možnost. Značka `strong` je vhodná pro obsah, který je důležitý a značka `b` pouze přitahuje pozornost čtenáře. Alternativně lze použít CSS vlastnost `font-weight: bold`. Pod informacemi o firmě lze najít ikonky Font Awesome s kategoriemi referencí.



Obrázek 40 - Rozcestník s kategoriemi referencí.

Před sekci s kontaktním formulářem je vystavena sekce s partnery a dodavateli. Všechna loga firem jsou zabarvená do stupních šedi, aby se udržela jednotná kompozita. Při najetí na logo myši se zobrazí originální vzhled loga společnosti.

#### Partneři a dodavatelé



Obrázek 41 - Sekce partnerů a dodavatelů.

## Partneři a dodavatelé



Obrázek 42 - Sekce partnerů a dodavatelů po najetí myši na prvek.

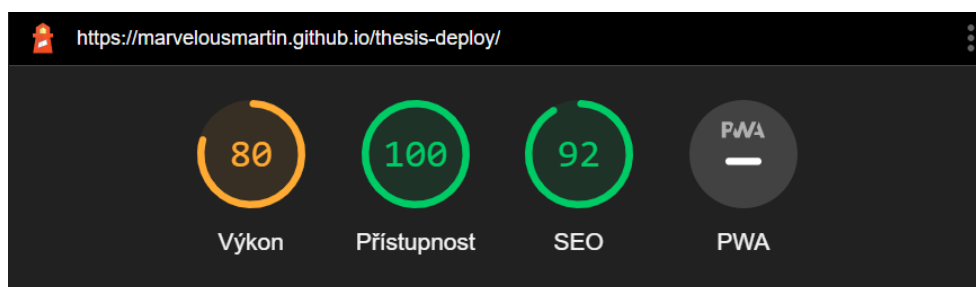
Poslední nepopsaná sekce je patička stránky. Do patičky se umísťuje logo společnosti a copyright spolu s rokem vytvoření webu. Rok vytvoření webu je vždy aktualizován pomocí JavaScript kódu `document.write(\/\d{4}\/.exec(Date())[0])`.



Obrázek 43 - Patička stránky.

## 4.6 Výsledky měření

Po vyhotovení aktualizované verze webu byla stránka změřena programem Lighthouse, stejným způsobem jako původní web. Výkon webu nijak neovlivňuje pozici ve vyhledávání a hodnota SEO je snížena knihovnou Lightbox2, jenž obsahuje komponenty, které nejsou sémanticky správně. Jelikož je tato knihovna stahována z CDN, nelze tuto chybu nijak opravit.



Obrázek 44 - Analýza nového webu nástrojem Lighthouse (Tools for Web Developers, 2022).

## 5 Závěr

Předmětem této práce byla modernizace webových stránek stavební společnosti JIRFA s. r. o. Ke splnění cíle bylo zapotřebí použít framework Bootstrap, který byl v práci vysvětlen a názorně demonstrován. Spolu s tímto frameworkem byly vysvětleny i principy tvorby webu moderního webu, kam patří například responzivita, přístupnost či dobré skóre SEO. Teoretická část také obsahovala popis existujících nástrojů, které lze využít na tvorbu webu bez znalosti programování. Vytvořená stránka obsahuje HTML, CSS, JavaScript a PHP. Všechny tyto jazyky byly popsány včetně vhodných příkladů použití. V první sekci práce je vysvětlen také verzovací nástroj Git a způsob jeho použití – základní příkazy, nahrávání práce do online úložiště GitHub, správa demo stránky přes službu GitHub Pages a celkový princip fungování tohoto nástroje. Pro zajímavost jsou v teoretické části uvedeny nástroje Hugo a Jekyll, což jsou generátory statických stránek.

Praktická část pojednávala o tvorbě vlastní práce, jaké byly vytyčeny cíle, požadavky od zadavatelů a podmínky, jež je třeba splnit – certifikáty, kontaktní formulář a reference. Obě verze, stará i nová, byly v rámci této bakalářské práce porovnány nástrojem od společnosti Google Lighthouse, aby se zjistila výkonnost, diagnostika a přístupnost webu. Podle Obrázek 27 - Analýza původního webu nástrojem Lighthouse (Tools for Web Developers, 2022). a Obrázek 44 je vidět, že cíle práce byly splněny.

## 6 Seznam použitých zdrojů

A Brief History of SEO, 2017. *Blue Frog DM* [online]. 5.9. [cit. 30. 1. 2022].  
Dostupné z: <https://www.bluefrogdm.com/blog/a-brief-history-of-seo>.

About GitHub Pages, 2022. *Github.com* [online]. [cit. 31. 1. 2022]. Dostupné z:  
<https://docs.github.com/en/pages/getting-started-with-github-pages/about-github-pages>.

*BEM Methodology*, 2021. Dostupné z: <https://en.bem.info/methodology/quick-start>.

BITTNER, Jan, 2021. *Responzivní webdesign pomocí Media Queries v CSS 3* [online].  
10.4. [cit. 22.1.2022]. Dostupné z: <https://www.itnetwork.cz/html-css/responzivni-webdesign/mobile-first-responzivni-design>.

Blender, 2013. Why is using the style-attribute in html bad. In: *stackoverflow.com*  
[online]. 2.3. [cit. 19.1.2022]. Dostupné z:  
<https://stackoverflow.com/a/15170320/11426324>

BURNHAM, Trevor, 2017. *Async JavaScript*. Raleigh (North Carolina, USA): The Pragmatic Programmers. ISBN 9781937785277.

CANZIBA, Elvis, 2018. *Hands-On UX Design for Developers*. Birmingham: Packt Publishing, 350 s. ISBN 9781788626699.

CloudCannon, 2021. Jekyll vs. Hugo – choosing the right tool for the job. In:  
*cloudcannon.com*. [online]. [cit. 12.1.2022]. Dostupné z:  
<https://cloudcannon.com/blog/jekyll-vs-hugo-choosing-the-right-tool-for-the-job>

DOWDEN, Martine, DOWDEN, Michael, 2019. *Approachable Accessibility: Planning for Success*. New York: Apress. ISBN 9781484248812.

Drupal.cz, 2021. O Drupalu. In: *drupal.cz* [online]. [cit. 22.11.2021]. Dostupné z:  
<https://www.drupal.cz/o-drupalu>

FELSEN, Lili Ouaknin, 2017. What are CSS Media Queries?.  
In: *medium.com* [online]. 23.10. [cit. 22.1.2022]. Dostupné z:  
<https://medium.com/@LiliOuakninFelsen/what-are-css-media-queries-4bee18de7650>

FRAIN, Ben, 2020. *Responsive Web Design with HTML5 and CSS*. 3. vyd.  
Birmingham: Packt Publishing. ISBN 9781839211560.

Framework vs. Library, 2020. In: *Geeksforgeeks.com* [online]. [cit. 30. 1. 2022].  
Dostupné z: <https://www.geeksforgeeks.org/software-framework-vs-library>

Get Started with Tailwind CSS, 2022. *Tailwindcss.com* [online]. 4.2. [cit. 9. 2. 2022].  
Dostupné z: <https://tailwindcss.com/docs/installation>

Github.com, 2020. HTML5 Boilerplate | Usage. In: *github.com* [online].  
[cit. 29.11.2021]. Dostupné z: <https://github.com/h5bp/html5-boilerplate/blob/v8.0.0/dist/doc/usage.md>

Github.com, 2021. Creating a GitHub Pages site with Jekyll. In: *github.com* [online].  
28.11. [cit. 29.11.2021]. Dostupné z: <https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll/creating-a-github-pages-site-with-jekyll>

GitHub.com: About pull requests, 2022. GitHub.com [online]. [cit. 31. 1. 2022].  
Dostupné z: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>

Github.com: Where the world builds software, 2022. *GitHub* [online].  
[cit. 31. 1. 2022]. Dostupné z: <https://github.com>.

HAACK, Phil a GUTHALS, Sarah, 2019. *GitHub For Dummies*. New York: John Wiley. ISBN 9781119572671.

HOGAN, Brian P., 2011. *Výukový kurz webového vývojáře*. Brno: Computer Press. ISBN 9788025135761.



- HTML5 Boilerplate* [online], 2022. [cit. 26. 1. 2022].  
Dostupné z: <https://html5boilerplate.com>
- Hugo Usage Statistics, 2020. In: *builtwith.com* [online]. 9.11. [cit. 26. 1. 2022].  
Dostupné z: <https://trends.builtwith.com/cms/Hugo>
- Humanstxt.org, 2021. O čem je humans.txt. In: *humantxt.org* [online]. 18.8. [cit. 29.11.2021]. Dostupné z: <https://humanstxt.org/CZ>
- Interval.cz, 2011. 10 nejlepších redakčních systémů (CMS). In: *interval.cz* [online]. [cit. 29.11.2021]. Dostupné z: <https://www.interval.cz/clanky/10-nejlepsich-redakcnich-systemu-cms>
- BLAŽKOVÁ, Tereza, 2021. Má smysl začínat s PHP? In: *itnetwork.cz* [online]. 24.7. [cit. 29.11.2021]. Dostupné z: <https://www.itnetwork.cz/blog/ma-smysl-zacinat-s-php>
- JANOVSKÝ, Dušan, 2020a. CSS prakticky. Jak psát web [online]. ©1999 – 2021 [cit. 22. 1. 2022]. Dostupné z: <https://www.jakpsatweb.cz/css/css-prakticky.html>
- JANOVSKÝ, Dušan, 2020b. Programové vybavení. In: *Jakpsatweb.cz* [online]. [cit. 16.1.2022]. Dostupné z: <http://polopate.jakpsatweb.cz/index.php?page=programy>
- JANOVSKÝ, Dušan, 2021a. HTML Editory. *Jakpsatweb.cz* [online]. 5.7. [cit. 6. 2. 2022]. Dostupné z: <https://www.jakpsatweb.cz/editory.html>
- JANOVSKÝ, Dušan, 2021b. Jak si vytvořit stránku. Jak psát web [online]. ©1999 - 2022 [cit. 22. 1. 2022]. Dostupné z: <https://www.jakpsatweb.cz/barvy-zapis.html>
- JANOVSKÝ, Dušan, 2021c. Řízení přístupu vyhledávačům. *Jakpsatweb.cz* [online]. 12.5. [cit. 6. 2. 2022]. Dostupné z: <https://www.jakpsatweb.cz/robots-txt.html>
- JANOVSKÝ, Dušan, 2022. *Jak psát web* [online]. ©1999 - 2022 [cit. 26. 1. 2022]. Dostupné z <https://www.jakpsatweb.cz>

Jekyll Usage Statistics, 2022. In: *builtwith.com* [online]. 8.1. [cit. 26. 1. 2022].  
Dostupné z: <https://trends.builtwith.com/cms/Jekyll>

Jetbrains.com: HTML, 2021. *Jetbrains.com* [online]. 16.11. [cit. 31. 1. 2022].  
Dostupné z: <https://www.jetbrains.com/help/webstorm/editing-html-files.html>

LOELIGER, Jon, 2012. *Version Control with Git*. 3. vyd. Sebastopol: O'Reilly Media, Inc. ISBN 9781449316389.

LONGAIR, Mark 2011. How to get certain commit from GitHub project.  
In: *stackoverflow.com*. [online]. 20.10. [cit. 29.11.2021]. Dostupné z:  
<https://stackoverflow.com/a/7832839/11426324>

MATSINOPOULOS, Panos, 2020. *Practical Bootstrap: Learn to Develop Responsively with One of the Most Popular CSS Frameworks*. New York: Apress. ISBN 9781484260708.

MDN Web Docs, 2021. *Developer.mozilla.org* [online]. 8.10. [cit. 7. 2. 2022].  
Dostupné z: <https://developer.mozilla.org/en-US/docs/Glossary/CDN>

MELONI, Julie C., 2014. *HTML, CSS and JavaScript All in One*. 2. vyd. Carmel (Indiana, USA): Sams, 704 s. ISBN 9780672337147.

MICHÁLEK, Martin, 2020a. Metodiky pro organizaci CSS kódu: Pište styly bez boolehavů. In: *vzhurudolu.cz* [online]. 21.7. [cit. 19.1.2022] Dostupné z:  
<https://www.vzhurudolu.cz/prirucka/css-metodiky>

MICHÁLEK, Martin, 2020b. Vkládání JavaScriptu jako async, defer a type="module" versus rychlost webu. In: *vzhurudolu.cz* [online]. 6.4. [cit. 6. 2. 2022]. Dostupné z:  
<https://www.vzhurudolu.cz/prirucka/js-async-defer-module>

PADOLSEY, James, 2020. *Clean Code in JavaScript*. Birmingham: Packt Publishing. ISBN 9781789957648.

PEHLIVANIAN, Ara a Don NGUYEN, 2014. *JavaScript okamžitě [Jump Start JavaScript]*. Brno: Computer Press, 160 s. ISBN 9788025141632.

PIPINELLIS, Achilleas, 2018. *GitHub Essentials: Unleash the power of collaborative development workflows using GitHub*. 2. vyd. Birmingham: Packt Publishing. ISBN 9781789138337.

PRAJAPATI, Ameet 2019. What is git commit, push, pull, log, aliases, fetch, config & clone. In: *medium.com* [online]. 20. 9. [cit. 14.1.2022]. Dostupné z: <https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c>

Pristupnost.cz, 2022. Česká pravidla přístupnosti. In: *pristupnost.cz* [online]. 22.1. [cit. 12. 03. 2022]. Dostupné z: <http://www.pristupnost.cz/jak-tvorit-pristupny-web/pravidla-pristupnosti/ceska-pravidla-pristupnosti>

RAPPIN, Noel, 2021. *Modern CSS with Tailwind*. Raleigh (North Carolina, USA): The Pragmatic Programmers. ISBN 9781680508185.

RedHat.com, 2019. What is an IDE?. In: *redhat.com* [online]. 8.1. [cit. 16.1.2022]. Dostupné z: <https://www.redhat.com/en/topics/middleware/what-is-ide>

SIKORA, Piotr, 2016. *Professional CSS3*. Birmingham: Packt Publishing, 362 s. ISBN 9781785880940.

SKLAR, David, 2016. *Learning PHP*. Sebastopol: O'Reilly Media. ISBN 9781491933572.

SRI Hash Generator, 2022. *Srihash.org* [online]. 26.1. [cit. 7. 2. 2022]. Dostupné z: <https://www.srihash.org>

StatCounter: Desktop vs Mobile vs Tablet Market Share Worldwide, 2022. *StatCounter.com* [online]. 14.1. [cit. 1. 2. 2022]. Dostupné z: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>

TAGLIAFERRI, Lisa, 2021. How To Use Git: A Reference Guide. In: *Digitalocean.com* [online]. 2.9. [cit. 26. 1. 2022]. Dostupné z: <https://www.digitalocean.com/community/cheatsheets/how-to-use-git-a-reference-guide>

W3schools.com, 2022a. CSS Introduction. In: *w3schools.com* [online]. [cit. 22.1.2022] Dostupné z: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

W3schools.com, 2022b. HTML Attributes. In: *w3schools.com* [online]. 13.1. [cit. 16.1.2022] Dostupné z: [https://www.w3schools.com/html/html\\_attributes.asp](https://www.w3schools.com/html/html_attributes.asp)

W3schools.com, 2022c. HTML Basic Examples. In: *w3schools.com* [online]. 14.1. [cit. 16.1.2022]. Dostupné z: [https://www.w3schools.com/html/html\\_basic.asp](https://www.w3schools.com/html/html_basic.asp)

W3schools.com, 2022d. HTML Element Reference. In: *w3schools.com* [online]. [cit. 22.1.2022]. Dostupné z: <https://www.w3schools.com/TAGS/default.asp>

W3schools.com, 2022e. HTML Links. In: *w3schools.com* [online]. 14.1. [cit. 16.1.2022] Dostupné z: [https://www.w3schools.com/html/html\\_links.asp](https://www.w3schools.com/html/html_links.asp)

W3schools.com, 2022f. HTML Paragraphs. In: *w3schools.com* [online]. 13.1. [cit. 16.1.2022] Dostupné z: [https://www.w3schools.com/html/html\\_paragraphs.asp](https://www.w3schools.com/html/html_paragraphs.asp)

W3schools.com, 2022g. JavaScript Tutorial. In: *w3schools.com* [online]. [cit. 31.1.2022] Dostupné z: <https://www.w3schools.com/js/default.asp>

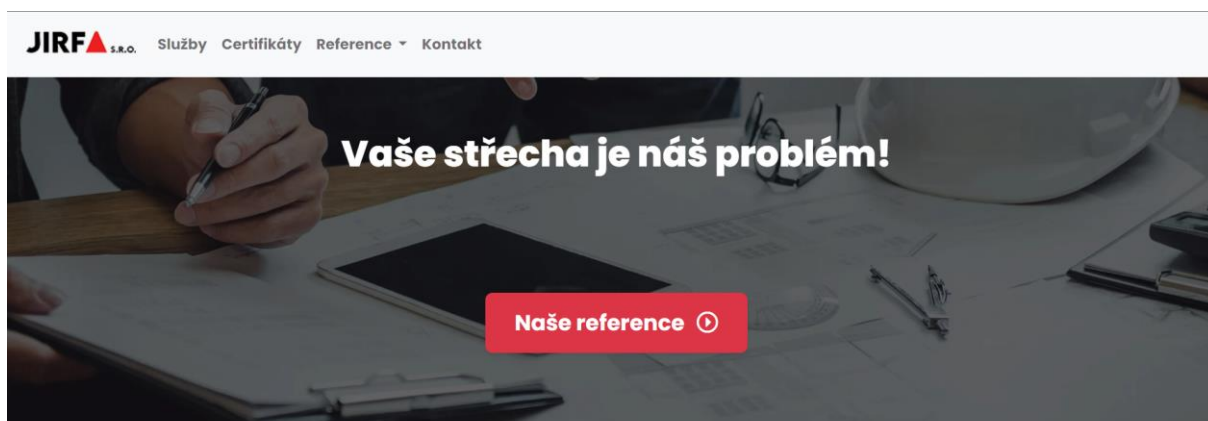
W3Schools.com, 2022h. Responsive Web Design – The Viewport. In: *W3schools.com* [online]. [cit. 15. 2. 2022]. Dostupné z: [https://www.w3schools.com/css/css\\_rwd\\_viewport.asp](https://www.w3schools.com/css/css_rwd_viewport.asp)

W3techs.com, 2021. Usage statistics and market share of WordPress. In: *w3techs.com* [online]. [cit. 14.11.2021]. Dostupné z: <https://w3techs.com/technologies/details/cm-wordpress>

WATTS, Luke, 2016. *Mastering Sass*. Birmingham: Packt Publishing, 318 s. ISBN 9781785883361.

## 7 Přílohy

Příloha 1 - Snímek obrazovky první části nového webu. ....	49
Příloha 2 - Snímek obrazovky druhé části nového webu. ....	50
Příloha 3 - Snímek obrazovky třetí části nového webu. ....	50
Příloha 4 - CD se zdrojovým kódem webové stránky	



### Čím se zabýváme

**Stavební firma JIRFA, s.r.o.** se specializuje na dodávky a montáže všech typů střech a krytin, pergol, garážových stání a dřevostaveb, včetně celkových oprav střešních pláštů, pokrývačských prací a dále na veškeré sádrokartonářské práce. Také se zaměřujeme na opravu památkově chráněných objektů pod dozorem památkového úřadu.

Na všechny tyto práce máme proškolené, odborně způsobilé zaměstnance, kteří pravidelně zvyšují svoji kvalifikaci.

Firma JIRFA, s.r.o. má sídlo v Praze 4, ale pobočku také v Načeradci (okr. Benešov). V naší načeradské pobočce máme plně vybavenou truhlářskou dílnu (truhlářské a tesářské stroje a nářadí na výrobu hoblovaných pergol a vazeb). Součástí areálu jsou také sklady o rozloze 3000m<sup>2</sup>, kde máme uskladněno a připraveno ihned k prodeji řezivo, střešní latě, prkna, fošny nebo hranoly.

Stavební firma JIRFA, s.r.o. je držitelem certifikátů: **BRAMAC**, **VELUX**, **ISOVER**, protipožární systémy **RIGIPS**, hydropojistné fólie a další. JIRFA, s.r.o. má uzavřenou smlouvu na pojištění odpovědnosti za způsobenou škodu třetím osobám při výkonu své činnosti až do výše 10 mil.Kč. Všichni naši zaměstnanci jsou řádně proškoleni v oblasti bezpečnosti práce a práce ve výškách.

V případě zájmu o naše služby Vám **zdarma** vypracujeme cenovou nabídku.



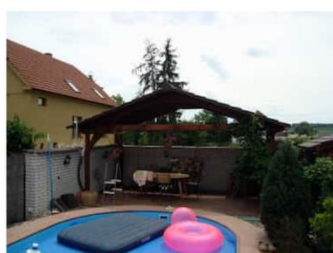
*Příloha 1 - Snímek obrazovky první části nového webu.*

## Certifikáty



## Reference

Vše



Příloha 2 - Snímek obrazovky druhé části nového webu.

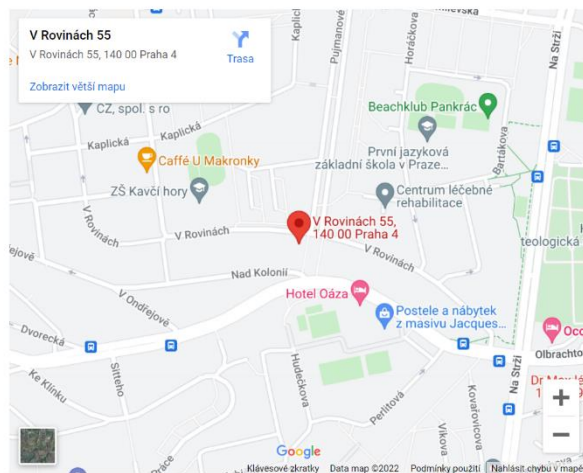
## Partneři a dodavatelé



## Kontakty

**JIRFA, s.r.o.**  
V Rovínách 55  
140 00  
Praha 4  
IČO: 27143287  
[jirfa.cz](http://jirfa.cz)

**Jednatelé:**  
Jiří Šteker  
[+420 606 094 834](tel:+420606094834)  
František Doubek  
[+420 603 269 054](tel:+420603269054)



## Ozvěte se nám!

Vaše jméno a příjmení

Váš e-mail

Váš telefon

Město  PSC

Text zprávy

**JIRFA** s.r.o.  
© 2022 Martin Dub, JIRFA s.r.o.

*Příloha 3 - Snímek obrazovky třetí části nového webu.*