



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

**Responzivní webová rozhraní s využitím CSS
frameworku Bulma**

**Responsive web interface using the Bulma CSS
framework**

Bakalářská práce

Vypracoval: Jakub Burda

Vedoucí práce: PaedDr. Petr Pexa, Ph.D

České Budějovice 20. dubna 2023

Zadání bakalářské práce

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta
Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jakub BURDA**
Osobní číslo: **P20486**
Studijní program: **B7507 Specializace v pedagogice**
Studijní obor: **Informační technologie a e-learning**
Téma práce: **Responzivní webová rozhraní s využitím CSS frameworku Bulma.**
Zadávající katedra: **Katedra informatiky**

Zásady pro vypracování

Cílem bakalářské práce je představení možnosti využití CSS frameworku Bulma, který má usnadnit a urychlit kódování webových stránek. Bulma je open source CSS framework založený na CSS Flexible Box Layoutu (tzv. flexboxu), jehož použitím docílíme automatického uspořádání responzivních prvků v závislosti na velikosti obrazovky, webová stránka je tedy plně responzivní. Framework Bulma je knihovna CSS tříd, kterou je možné použít buď jako předkompilovaný .css soubor nebo jako .sass soubor, aby bylo možné framework přizpůsobit potřebám webdesignéra. V případě použití .sass souboru je pak Bulma plně modulárním frameworkem.

V teoretické části bakalářské práce budou vymezeny základní pojmy, úzce související s frameworkem Bulma (např. flexbox, CSS flexbox), uvedeny jeho vlastnosti a popsána instalace potřebného softwaru pro práci s technologií Bulma. Dále pak budou představeny další typy CSS frameworků jako Bootstrap, Foundation a Semantic k porovnání s frameworkem Bulma a následnému zpracování výhod a nevýhod samotného frameworku Bulma. V praktické části bakalářské práce bude zhotovena responzivní webová stránka pomocí frameworku Bulma k představení možností, které tato technologie přináší a zároveň také k demonstraci praktických výhod a nevýhod frameworku Bulma.

Rozsah pracovní zprávy: **40**
Rozsah grafických prací: **interaktivní modely**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

1. Bulma.io: Bulma: the modern CSS framework that just works. [online]. [cit. 2022-03-30]. Dostupné z: <https://bulma.io/>
2. Bulma.io: Documentation: Everything you need to create a website with Bulma [online]. [cit. 2022-03-30]. Dostupné z: <https://bulma.io/documentation/>
3. Mobiosolutions: What is Bulma ? Advantages & disadvantages of using Bulma [online]. [cit. 2022-03-30]. Dostupné z: <https://mobiosolutions.com/what-is-bulma-advantages-disadvantages-of-using-bulma/>
4. Devmountain: Bulma CSS: What It Is and Why It's a Framework That Developers Love [online]. [cit. 2022-03-30]. Dostupné z: <https://blog.devmountain.com/why-bulma-css-could-be-your-new-favorite-framework/>
5. HubSpot: The Bulma CSS Framework: What It Is and How To Get Started [online]. [cit. 2022-03-30]. Dostupné z: <https://blog.hubspot.com/website/bulma-css>
6. Tutorialspoint: Bulma [online]. [cit. 2022-03-30]. Dostupné z: <https://www.tutorialspoint.com/bulma/>

Vedoucí bakalářské práce: PaedDr. Petr Pexa, Ph.D.
Katedra informatiky

Datum zadání bakalářské práce: 4. dubna 2022
Termín odevzdání bakalářské práce: 30. dubna 2023

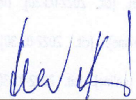
ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(Faint, mirrored text from the reverse side of the page, including university name and department information.)

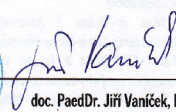
Zásady pro vypracování

(Faint, mirrored text from the reverse side of the page, detailing the rules for completing the thesis.)

(Faint, mirrored text from the reverse side of the page, likely containing contact information or administrative notes.)



doc. RNDr. Helena Koldová, Ph.D.
děkanka



doc. PaedDr. Jiří Vaniček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 4. dubna 2022

Prohlášení

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval(a) pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích dne 20. dubna 2023.

Jakub Burda

Anotace

Cílem bakalářské práce je představení možností využití CSS frameworku Bulma, který má usnadnit a urychlit kódování webových stránek nebo webových aplikací. Bulma je open source CSS framework založený na CSS Flexible Box Layoutu (tzv. flexboxu) a CSS Grid Layoutu (tzv. gridu), jehož použitím docílíme automatického uspořádání responzivních prvků v závislosti na velikosti obrazovky, webová stránka se stává plně responzivní. Framework Bulma je knihovna CSS tříd, kterou je možné použít buď jako předkompilovaný .css soubor nebo jako .sass soubor, aby bylo možné framework přizpůsobit potřebám webdesignéra. V případě použití .sass souboru je pak Bulma plně modulárním frameworkem.

V teoretické části bakalářské práce budou vymezeny základní pojmy úzce související s frameworkem Bulma (např. flexbox), uvedeny jeho vlastnosti, představeny technologie související s frameworkem a popsána instalace potřebného softwaru pro práci s technologií Bulma. Dále pak budou představeny další typy CSS frameworků jako Bootstrap, Foundation a Pure k porovnání s frameworkem Bulma a následnému zpracování výhod a nevýhod samotného frameworku Bulma.

V praktické části bakalářské práce bude zhotovena responzivní webová stránka pomocí frameworku Bulma k představení možností, které tato technologie přináší a zároveň také k demonstraci praktických výhod a nevýhod frameworku Bulma.

Klíčová slova

HTML, CSS, Javascript, framework, Bulma, front-end, vývojové prostředí

Abstract

The point of bachelor thesis is presentation of utilisation options CSS Bulma framework, which is meant to make coding website or web applications easier and faster. Bulma is open source CSS framework, which is based on CSS Flexible Box Layout (a.k.a. flexbox) and CSS Grid Layout (a.k.a. grid), which we use to achieve an automatic arrangement of responsive elements depending on the screen size, website is therefore fully responsive. Framework Bulma is library of CSS classes, which can be used like precompiled .css file or .sass file so that the framework can be adapted to the needs of a web designer. In case of use .sass file is Bulma fully modular framework.

In the theoretical part of bachelor thesis will be defined basic concepts, which are closely related with framework Bulma (e.g. flexbox), listed his properties, introduced technologies related to the framework and description of the installation of the necessary software for work with Bulma technology. Then will be introduced other types of CSS framework like Bootstrap, Foundation and Pure to compare with Framework Bulma and subsequent processing advantages and disadvantages of the Bulma framework.

In the practical part of the bachelor thesis will be made a responsive website with help of framework Bulma to introduce possibility that this technology brings and at the same time also to demonstration of practical advantages and disadvantages Bulma framework.

Keywords

HTML, CSS, Javascript, framework, Bulma, front-end, Integrated Development Environment

Poděkování

Chtěl bych poděkovat panu doktorovi Petrovi Pexovi za důsledné vedení mé bakalářské práce a jeho trpělivost.

Obsah

1	Úvod	11
1.1	Cíle práce	11
1.2	Metody práce	11
1.3	Východiska práce	12
2	Technologie užívané při vývoji	13
2.1	HTML	13
2.1.1	Historie HTML	13
2.1.2	Co nám HTML umožňuje	14
2.1.3	HTML5	15
2.2	CSS	16
2.2.1	Historie CSS	16
2.2.2	Co nám CSS umožňuje	16
2.2.3	CSS3	16
2.3	JavaScript	18
2.3.1	Historie JavaScriptu	18
2.3.2	Co nám JavaScript umožňuje	18
2.3.3	Jak JavaScript vypadá	19
3	Bulma	20
3.1	Framework	20
3.2	Co nám Bulma umožňuje	21
3.3	Jak Bulma vypadá	22
3.4	Výhody Bulmy	23
3.4.1	Výhody Bulmy v bodech	25
4	Vývojové prostředí	26
4.1	Visual Studio Code	26

5	Visual Studio Code	28
5.1	Instalace Visual Studio Codu	28
5.2	Použití Visual Studio Codu	28
6	Vývoj webové stránky	30
6.1	Adresáře webové stránky	30
6.2	Soubory webové stránky	31
6.3	Instalace frameworku Bulma	31
6.4	Použití frameworku Bulma	32
6.5	Vývoj webové stránky pomocí Bulmy	33
6.5.1	Začátek HTML dokumentu	33
6.5.2	Navigační menu	34
6.5.3	Hlavička	40
6.5.4	Sekce	43
6.5.5	Patička	51
7	Vývoj komponent k porovnání	57
7.1	Vývoj komponenty pomocí Bootstrapu	57
7.1.1	Vývoj patičky pomocí Bootstrapu	58
7.2	Vývoj komponenty pomocí Foundationu	62
7.2.1	Vývoj sekce pomocí Foundationu	63
7.3	Vývoj komponenty pomocí Pure	66
7.3.1	Vývoj sekce pomocí Pure	66
8	Porovnání frameworků	73
8.1	Porovnání Bulmy s Bootstrapem	73
8.1.1	Podobnosti Bulmy a Bootstrapu	73
8.1.2	Rozdíly Bulmy a Bootstrapu	74
8.2	Porovnání Bulmy s Foundationem	75
8.2.1	Podobnosti Bulmy a Foundation	75
8.2.2	Rozdíly Bulmy a Foundation	75

8.3	Porovnání Bulmy s Pure.css	76
8.3.1	Podobnosti Bulmy a Pure.css	77
8.3.2	Rozdíly Bulmy a Pure.css	77
9	Závěr	79
	Seznam použité literatury a zdrojů	81
A	Příloha	86
B	Příloha	87

1 Úvod

Bakalářská práce se zabývá open source frameworkem Bulma pro tvorbu responzivních front-end systémů. Framework Bulma využívá nové vlastnosti CSS3 technologie, a to vlastnosti flexibilního prvku tzv. flexboxu a gridu pro usnadnění a zároveň urychlení tvorby a vývoje plně responzivních webových stránek či webových aplikací. Vývojáři Bulmy se snaží zejména cílit na jednoduchost a intuitivnost frameworku, oproti jiným dostupným CSS frameworkům.

1.1 Cíle práce

Cílem bakalářské práce je teoreticky popsat funkci, tvorbu a možnosti využití CSS frameworku Bulma pro vývoj responzivní webové stránky. Následně vytvořit responzivní webovou stránku a fyzicky demonstrovat možné výhody a nevýhody této front-end frameworkové technologie na reálném produktu. Zjistit, do jaké míry je tato technologie na bázi tzv. flexboxu a gridu použitelná v reálném vývoji responzivních webů a shrnout zjištěné výhody a nevýhody frameworku. Po dosažených znalostí při tvorbě responzivního webu porovnat možné zjištěné výhody a nevýhody frameworku z praktické části s jinými dostupnými CSS frameworky.

1.2 Metody práce

V úvodní části bakalářské práce vymezím základní pojmy spojené s front-end technologiemi, potažmo webovými systémy. Následně v teoretické části se budu zabývat stručným popisem základních front-end technologií, nutných k prouzkoumání samotného tématu bakalářské práce, a to responzivního CSS frameworku Bulma, které mimo jiné budou využity k tvorbě praktické části. Na základě postupu vývoje praktické části budu demonstrovat výhody a nevýhody frameworku, které porovnam s dalšími konkurenčními frameworky. Ze získaných znalostí souhrně zhodnotit míru intuitivnosti a jednoduchosti re-

sponzivního frameworku Bulma.

1.3 Východiska práce

Vývoj profesionálních responzivních webových stránek nebo webových aplikací na širokou škálu chytrých mobilních zařízení a počítačů bez aplikace frameworků je zdlouhavý proces práce, a proto jsou vyvíjeny nejrůznější frameworky na podporu tvorby webových stránek a webových aplikací, mezi které patří i CSS responzivní framework Bulma. Bulma reaguje na tuto širokou škálu zařízení tzv. flexibilním rozvržením elementů na stránce, případně mřížkovým rozvržením elementů a tím by měla usnadnit vývoj, oproti jiným frameworkům, které tuto vlastnost CSS3 doposud nepodporují. Bulma se mimo jiné snaží i důrazně cílit na jednoduchost a intuitivnost frameworku při samotném použití.

2 Technologie užité při vývoji

Responzivní frameworky jsou stále důležitější součástí dnešního vývoje webových stránek a aplikací. Tyto frameworky nám umožňují vytvářet webové stránky či aplikace pomocí přesné struktury a metodiky pro řešení problému v dané oblasti, zjednodušovat práci koderů a programátorů, umožnit jim soustředit se na implementaci konkrétních funkcionalit, opakovaně používat kód a především zvýšení efektivního řešení problémů, který se projevuje zkrácením času potřebného k vývoji webové stránky nebo aplikace.

Responzivní open source framework Bulma je jedním z frameworků, který umožňuje za pomoci CSS a SCSS knihoven výrazně zkrátit design, vývoj webové stránky nebo aplikace. Bulma je tedy jednoduchým nástrojem pro začátečníky, ale i pro pokročilé k vývoji vizuálně moderních a plně responzivních webových stránek nebo aplikací. Framework se stává velmi populárním nástrojem u vývojářů díky své jednoduchosti, intuitivnosti, flexibilitě a kompatibilitě s dalšími využívanými frameworky, jako je např. Vue.js, React, Angular.

Nicméně k využití této frameworkové technologie potřebujeme i řadu dalších znalostí, zejména znalostí front-end technologií, jako je například HTML, CSS a JavaScriptový jazyk.

2.1 HTML

Hypertext Markup Language (zkratka HTML) je v informatice název značkovacího jazyka používaného pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. HTML je hlavním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.[7]

2.1.1 Historie HTML

V roce 1990 vynalezl Berners-Lee jazyk HTML pro vytváření elektronických stránek, které se zobrazují na World Wide Webu, přičemž každá stránka ob-

sahuje řadu odkazů na jiné stránky, tzv. hypertextové odkazy. Jazyk HTML byl navržen speciálně pro uchovávání obsahu webové stránky, potřeboval ještě něco, co by uchovávalo styl a design jako samostatnou záležitost, něco jako tabulku stylů (kaskádové styly), což je soubor pravidel stylů, která prohlížeči říkají, jak mají být různé styly aplikovány na značky HTML pro prezentaci dokumentu.[13] Více o kaskádových stylech v podkapitole CSS.

2.1.2 Co nám HTML umožňuje

Co nám vlastně HTML umožňuje? HTML nám umožňuje strukturovat obsah webových stránek pomocí značek neboli tagů a elementů, které popisují jednotlivé části stránky pro interpretaci webovému prohlížeči, který zobrazuje uživateli obsah jako stránku. HTML tvoří základní prvek pro tvorbu a zobrazení obsahu uživateli v kombinaci s CSS jazykem, který HTML obsah stylizuje a JavaScriptem, který tvoří logickou strukturu a dynamiku webové stránky nebo aplikace. Tyto technologie tvoří základ pro moderní webdesign. HTML používá několik základních sad značek, které se používají pro zobrazení obsahu. Tyto značky lze dále kombinovat (vnořovat) a upravovat pomocí dalších atributů jako např. atribut třídy (class). Některé z nejčastěji používaných značek v HTML jsou:

- `<html>` Označuje začátek a konec celého dokumentu HTML.
- `<head>` Obsahuje metadata dokumentu jako je název stránky, odkazy na CSS a JavaScript soubory a další informace.
- `<body>` Obsahuje veškerý viditelný obsah stránky jako je text, obrázky a odkazy.
- `<h1>` Označuje značku interpretující nadpis hlavní úrovně.
- `<p>` Označuje odstavec textu.
- `<div>` Slouží k oddělení a seskupení obsahu v rámci stránky.

- `` Vloží obrázek na stránku.
- `` a `` Používají se pro vytvoření seznamu s odrážkami nebo číslováním.

2.1.3 HTML5

Finální specifikace byla vydána 28. října 2014. Proti předchozí verzi HTML4 z roku 1997 přináší podstatné změny, přičemž mezi nejdůležitější patří přímá podpora přehrávání multimédií v prohlížeči a podpora pro aplikace, které fungují i bez připojení k Internetu. HTML5 obsahuje podrobné modely zpracování pro podporu interoperabilnějších implementací; rozšiřuje, zlepšuje a racionalizuje značkování dostupné pro dokumenty a zavádí značkovací a aplikační programovací rozhraní (API) pro složité webové aplikace.[8]

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,
7     initial-scale=1.0">
8   <title>Example</title>
9 </head>
10 <body>
11   <h1>Header Tag Example</h1>
12 </body>
13 </html>
```

Příklad 1: Ukázka HTML5 hlavičky se základními meta daty

2.2 CSS

Kaskádové styly (CSS) jsou jazykem stylů, který se používá k popisu prezentace dokumentu napsaného v jazyce HTML nebo XML (včetně dialektů XML, jako jsou SVG, MathML nebo XHTML). CSS popisuje, jak mají být prvky zobrazeny na obrazovce, na papíře, v řeči nebo na jiných médiích. Jazyk CSS patří mezi základní jazyky otevřeného webu a je standardizován ve všech webových prohlížečích podle specifikací W3C. Dříve probíhal vývoj různých částí specifikace CSS synchronně, což umožňovalo verzování nejnovějších doporučení. Možná jste již slyšeli o CSS1, CSS2.1 nebo dokonce CSS3. Spíše je nyní vše CSS bez čísla verze.[9]

2.2.1 Historie CSS

CSS (Cascading Style Sheets) poprvé navrhl Håkon Wium Lie 10. října 1994 při práci s Timem Bernersem-Lee v CERNu a zbytek je historie. CSS nebyl v té době jediným vyvíjeným stylovacím jazykem, ale právě prvek kaskádování a rozvíjení posloupnosti jej odlišoval od ostatních.[12]

2.2.2 Co nám CSS umožňuje

Co nám vlastně CSS umožňuje? CSS nám umožňuje základní stylizaci webových stránek a aplikací. Stylizace značek neboli tagů se provádí za pomoci tříd (class) nebo jedinečného identifikátoru (id). CSS přesně definuje, jak by měly být prvky na stránce zobrazeny, včetně pozice, barvy, velikosti, velikosti fontů a dalších vlastností. Nutností ke stylizování HTML tagů a elementů je potřeba CSS stylovací soubor tzv. nalinkovat do hlavičky souboru HTML pomocí linkovacího stylesheet příkazu.

2.2.3 CSS3

CSS3 je nejnovější verzí jazyka CSS, který přináší mnoho nových funkcí a vlastností, oproti předchozím verzím CSS. Tyto nové vlastnosti jsou pro vývojáře

webových stránek nebo aplikací velmi užitečné. Součástí nového CSS3 je i k dispozici uspořádání prvků pomocí nové funkce flexibilního boxu tzv. flexboxu. Do verze CSS3 bylo možné řešit uspořádání prvků pouze jinými způsoby, např. float a pozice. Tyto způsoby nejsou tak flexibilní a snadno ovladatelné jako je flexbox CSS3. Důvod zaměření se na tuto konkrétní vlastnost CSS3 je, že právě responzivní framework Bulma staví přesně na této nové funkci. Některé z dalších nejvýznamnějších nových funkcí CSS3 jsou:

- Media queries - umožňující vytvářet responzivní design, přizpůsobující se různým zařízením a velikostem obrazovek
- Grid layout - umožňuje vytvářet komplexní mřížkové uspořádání prvků na stránce
- Nové selektory - umožňují selektovat prvky na základě různých vlastností
- Animace a přechody - umožňují animovat prvky na stránce, což umožňuje vytvářet interaktivní a poutavé webové stránky
- Podpora pro stínování - (box-shadow, text-shadow)
- Pokročilé vlastnosti - pro typografii (např. @font-face)
- Flexbox - flexibilní layout pro snadné uspořádání prvků v rámci stránky

```
1 /* Basic settings for resetting browser window properties */
2 * {
3     margin: 0;
4     padding: 0;
5     box-sizing: border-box;
6 }
7 /* Change the background colour and change the text colour
   for the tag <body> */
8 body {
9     background-color: #212121;
```

```
10 color: #ffffff;
11 }
```

Příklad 2: Ukázka designování značek neboli tagů pomocí CSS stylů

2.3 JavaScript

JavaScript je skriptovací nebo programovací jazyk, který umožňuje implementovat na webové stránky složité funkce. Pokaždé, když webová stránka dělá něco víc, než že se jen na ní díváte, např. zobrazuje statické informace jako včasné aktualizace obsahu, interaktivní mapy, animovanou 2D/3D grafiku, rolovací video jukeboxy atd. Můžete se vsadit, že je v ní pravděpodobně použit JavaScript. Je to třetí vrstva z koláče standardních webových technologií (HTML a CSS).[10] Podstatnou informaci tvoří, že JavaScript je implementován do webového prohlížeče a spouští se přímo ve webovém prohlížeči. To znamená, že není nutné stahovat nebo instalovat další potřebný software. Kromě toho existuje mnoho dalších knihoven a frameworků, které usnadňují vývoj webových aplikací v JavaScriptu, jako např. Vue.js, React nebo Angular.

2.3.1 Historie JavaScriptu

JavaScript vynalezl Brendan Eich v roce 1995. Byl vyvinut pro Netscape 2 a v roce 1997 se stal standardem ECMA-262. Poté, co společnost Netscape předala JavaScript organizaci ECMA, pokračovala nadace Mozilla ve vývoji JavaScriptu pro prohlížeč Firefox. Poslední verze Mozilly byla 1.8.5. (Totožná s ES5). Internet Explorer (IE4) byl prvním prohlížečem, který podporoval ECMA-262 Edition 1 (ES1). Internet Explorer (IE4) was the first browser to support ECMA-262 Edition 1 (ES1).[11]

2.3.2 Co nám JavaScript umožňuje

Co nám JavaScript vlastně tedy umožňuje? JavaScript nám umožňuje vytvářet moderní, plně interaktivní a dynamické webové stránky nebo webové aplikace.

Původní zaměření programovacího jazyka bylo na front-end neboli klientskou část webového vývoje. To znamená využití JS k interakci s uživatelem, manipulaci s webovým obsahem, validaci formulářů, k animacím a ke spoustě dalším účelům na klientské straně. S příchodem Node.js prostředí je možný běh JavaScriptových aplikací i na straně serveru tzv. serverových aplikací.

2.3.3 Jak JavaScript vypadá

Funkce přijímající číslo v parametru a vyhodnocující sudost nebo lichost čísla.

```
1 /**
2  * Javascript function is an even number
3  */
4 function isEven(number) {
5     if(number % 2 == 0) {
6         console.log("The ", number, " is even.");
7     } else {
8         console.log("The ", number, " is odd.");
9     }
10 }
11
12 /**
13  * Calling javascript function is an even number
14  */
15 isEven(2);
```

Příklad 3: Ukázka funkce vyhodnocení sudosti čísla pomocí JavaScriptu

3 Bulma

Bulma je framework vytvořený a udržovaný společností Jeremy Thomas, tvůrcem dalších projektů jako například Bloomer, Vuepery a další. Bulma vlastní mnoho užitečných prvků pro efektivní tvorbu responzivních a zejména moderních webů nebo moderních webových aplikací.

Jedním z hlavních prvků je, že samotná Bulma je postavena na nové funkci tzv. flexboxu a gridu s příchodem nové CSS3 verze. Flexbox a grid je zásadní změnou v možnosti flexibilního uspořádání prvků na webové stránce nebo ve webové aplikaci. Do příchodu CSS3 bylo nutné řešit uspořádání pouze pomocí pozice nebo floatu, což neumožňuje takové flexibilní zobrazení prvků.

Bulma také obsahuje spousty dalších funkcí, jako je například rozdělení na komponenty (navbar, menu, cards, panels etc.), rozdělení layoutace (container, section, footer etc.), mřížkové rozdělení na sloupce a řádky (columns, column, nesting etc.), tlačítka, formuláře, různé velikosti titulků a nadpisů, základní moderní paletu barev a spousty dalšího. Všechny tyto prvky jsou navrženy tak, aby byly responzivní. To znamená, že se prvky přizpůsobují různým velikostem obrazovek pro správné zobrazení obsahu.

Kromě všech těchto funkcí nám framework nabízí i možnost přizpůsobení, ať už barevného spektra, typografických pravidel či dalších jiných vlastností. Pro vývoj webových aplikací nám Bulma umožňuje instalovat Sass knihovnu kompatibilní s front-endovými technologiemi, jako je například Vue.js, Angular nebo React skrze službu npm.

3.1 Framework

Framework (česky aplikační rámec nebo vývojová platforma) je softwarová struktura neboli soubor abstraktních konceptů a nástrojů, které slouží k usnadnění vývoje webových stránek, aplikací a jiného dalšího softwaru, pro který je framework navržen.

Framework tvoří základní kostru nebo základní konstrukci na podporu

komplexního programového řešení. Základní konstrukci, na kterou lze dále navazovat další tvorbu už konkrétního programového řešení. Framework poskytuje programátorům předem připravené nástroje, komponenty a funkce, které mohou použít pro rychlejší, bezpečnější a efektivnější vývoj softwaru.

Kromě toho framework zajišťuje i standardní způsob práce s kódem, a tím docílí konzistentnosti a snadné čitelnosti kódu pro další vývojáře, potažmo programátory. Framework je často tvořen sadami nástrojů, knihoven, modulů a standardů, které se spolu vzájemně doplňují a spolupracují. Tím umožňují vývojářům vytvářet složitější webové stránky, webové aplikace, aplikace, ale i jiný další software v krátkém časovém horizontu bez nutnosti vyvíjet veškeré funkcionality od začátku.

Framework nám vlatně jen umožňuje snažší a rychlejší manipulaci s daty, zpracování vstupů a výstupů, případně práci s databázemi atd. Při použití frameworku lze snížit náklady a urychlit vývoj daného softwaru, protože umožníme programátorům zaměřit se na řešení specifických problémů a implementaci pouze nových funkcí na místo implementace základní struktury a základních funkcí softwaru.

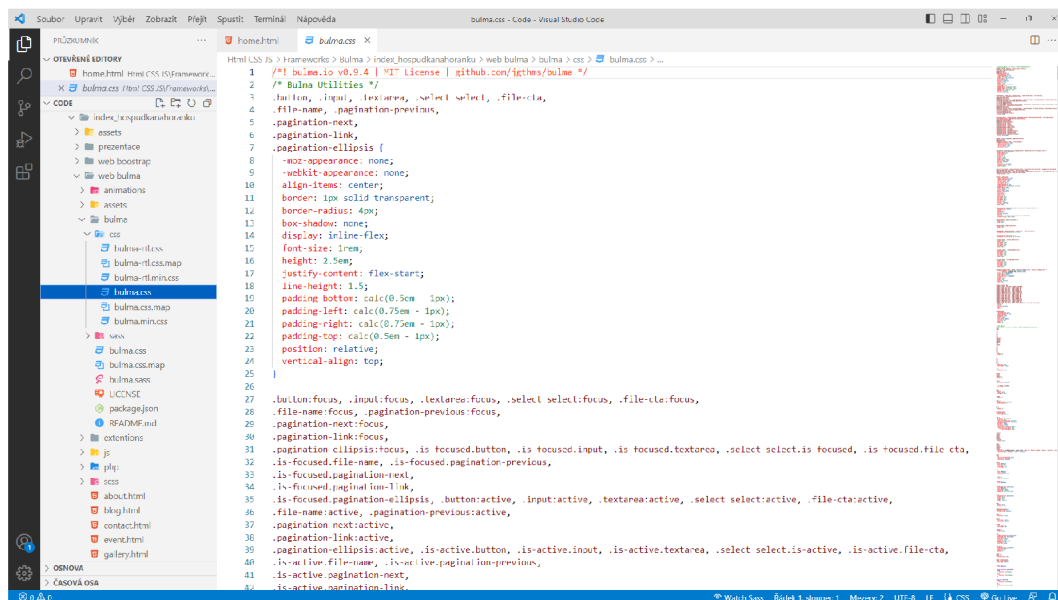
Další výhodou použití frameworku je, že framework je často vyvíjený jako open source software. To znamená, že framework je dostupný pro kohokoliv a má velkou komunitu vývojářů a uživatelů.

3.2 Co nám Bulma umožňuje

Co nám Bulma umožňuje? Poskytuje nám možnost rychle a správně tvořit moderní designové webové stránky či aplikace i s možností tzv. mobile first. Jasnou strukturu a metodiku řešení obecných problémů, jako je například responzivita na různých zobrazovacích zařízeních nebo správná layotace webu. Umožňuje nám opakované použití funkčního kódu, potažmo funkcionalit z knihoven a především zkrácení času potřebného k vývoji webové stránky nebo aplikace, a tím zvýšit naši efektivitu práce.

3.3 Jak Bulma vypadá

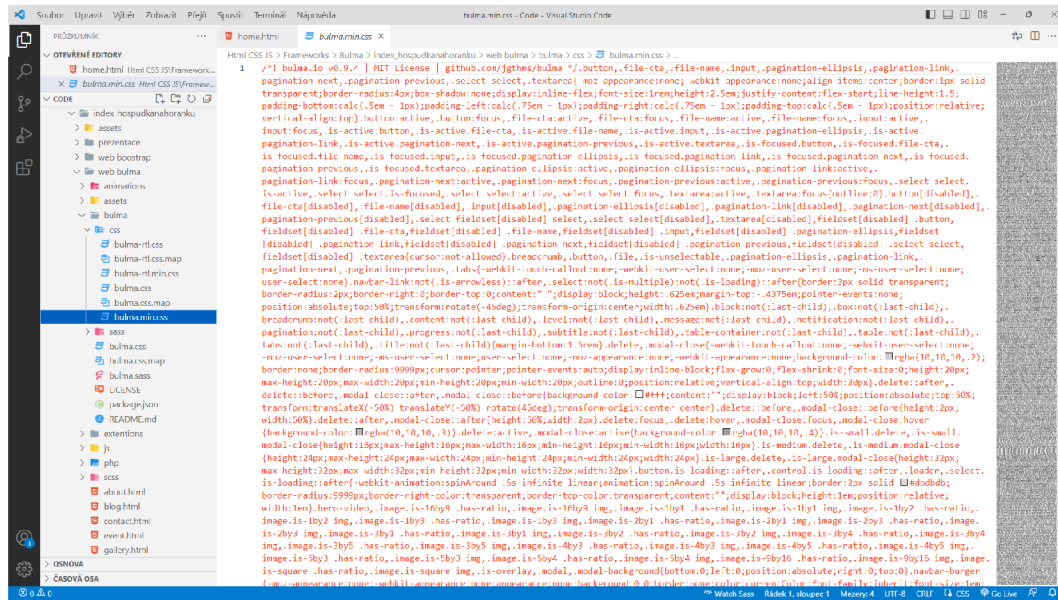
CSS framework Bulma je distribuovaný ve dvou základních verzích, podobně jako u jiných CSS frameworků. Verze jsou bulma.min.css a bulma.css. Soubor bulma.css je plnohodnotná knihovna CSS stylů s veškerým kódem a jasně čitelným formátováním pro vývojáře. Obě verze Bulmy lze používat standardním způsobem pro linkování nebo importování lokálních souborů, případně způsobem pomocí linkování nebo importování online souborů. Soubor bulma.css je především určen k přizpůsobení frameworku vlastním konkrétním úpravám.



Obrázek 1: Framework Bulma v souboru bulma.css jako plná verze

Soubor bulma.min.css je verze frameworku odstraněná o nepotřebný kód ze zdrojových souborů CSS s cílem zmenšit velikost souboru, aniž by se učinil zásah do již navrženého samotného fungování frameworku nebo zásahu fungování frameworku v prohlížeči. Samotný soubor bulma.min.css po odebrání nepotřebného kódu se pak stává těžko čitelným pro vývojáře pro případné budoucí úpravy samotného frameworku. To znamená, že verze bulma.min.css není vhodná pro případné upravování konkrétních knihoven CSS frameworku.

Úpravy vyvíjené stránky nebo aplikace lze však jednoduše provádět standardním způsobem přidání alternativní třídy s vynucením přepsání stávajícího stylu knihovny.



Obrázek 2: Framework Bulma v souboru bulma.min.css jako minimální verze

3.4 Výhody Bulmy

Mezi hlavní výhody frameworku Bulma patří jeho jednoduchost a intuitivnost. To lze snadno vidět z následujícího obrázku, kde obrázek tvoří jednoduchou soustavu HTML divů, které jsou jednotlivě nastýlované třídami Bulmy, aby nám tvořily jednoduché a flexibilní sloupce.

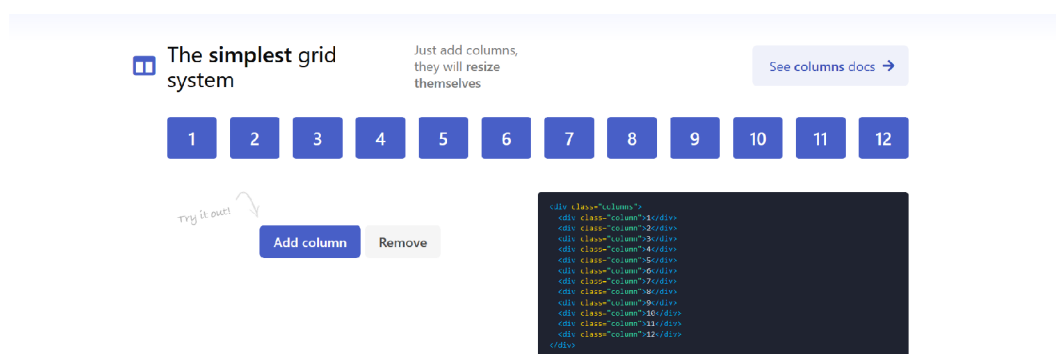
```

1 <div class="columns">
2   <div class="column">1</div>
3   <div class="column">2</div>
4   <div class="column">3</div>
5   <div class="column">4</div>
6 </div>

```

Příklad 4: Ukázka HTML divů s atributem třídy Bulmy

Jednotlivé sloupce nesou anglický název `column` a sada sloupců neboli řádek je nazýván `columns`. Tento jednoduchý a dobře čitelný kus kódu nám umožňuje vytvořit až 12 flexibilních sloupců v jedné řadě. Každý sloupec má potom stejnou délku vypočítanou v závislosti na šířce obrazovky. Už jen z uvedeného kódu lze říci, že Bulma je velmi snadná k použití jak pro začínající, tak pokročilé vývojáře.



Obrázek 3: Autor: bulma.io, Nejjednodušší systém mřížky

Zdroj: <https://bulma.io>

Bulma nese spousty dalších výhod, jako je například flexibilita, modulárnost, responzivita a dostupnost. Programátoři mají možnost snadno přizpůsobit design svým potřebám pomocí vlastních stylů a tříd nebo případně pomocí úpravy už předdefinovaných tříd v souboru frameworku.

Responzivní design: Bulma umožňuje snadné vytváření responzivních webů, které se přizpůsobí různým zařízením a velikostem obrazovky. Flexibilita: Bulma je velmi flexibilní a umožňuje programátorům snadno přizpůsobit vzhled svého webu pomocí vlastních stylů a tříd. Modulárnost: Bulma je modulární a umožňuje programátorům snadno používat pouze ty části frameworku, které potřebují. Volná distribuce a komunita: Bulma je open source framework, což znamená, že je k dispozici zdarma a má velkou komunitu uživatelů, kteří mohou pomoci s problémy a vylepšením. Lze tedy říci, že Bulma je velmi užitečný framework pro vývojáře, kteří potřebují jednoduchý a flexibilní CSS framework

pro tvorbu moderních a responzivních webových stránek.

3.4.1 Výhody Bulmy v bodech

- Jednoduchost
- Intuitivnost
- Flexibilita
- Modulárnost
- Responzivita
- Dostupnost

4 Vývojové prostředí

Vývojové prostředí (IDE - Integrated Development Environment) je softwarová aplikace usnadňující vývoj konkrétní webové stránky, aplikace a jiného dalšího softwaru. Nástroj vývojářů, potažmo programátorů pro psaní, editování, testování a lazení kódu vyvíjeného softwaru. Vývojová prostředí se využívají v mnoha oblastech, jako jsou webové aplikace, desktopové aplikace, mobilní aplikace, herní vývoj a ve spoustě dalších oblastech.

Součástí vývojového prostředí je sada funkcionalit, jako je například textový editor pro vizualizaci, psaní a úpravu již zmíněného samotného programového kódu, kompilátor pro přeložení kódu do strojového jazyka pro procesor nebo debugger pro snažší odhalování chyb v navrhnutém kódu. IDE je nástrojem, který nám usnadňuje a urychluje proces vývoje softwaru pro různé programovací jazyky na různých operačních systémech.

Příkladem vývojového prostředí je od společnosti Microsoft Visual Studio nebo Visual Studio Code pro Windows, Linux či MacOS. Dále PyCharm, PhpStorm, RubyMine od společnosti JetBrains také pro Windows, Linux i MacOS.

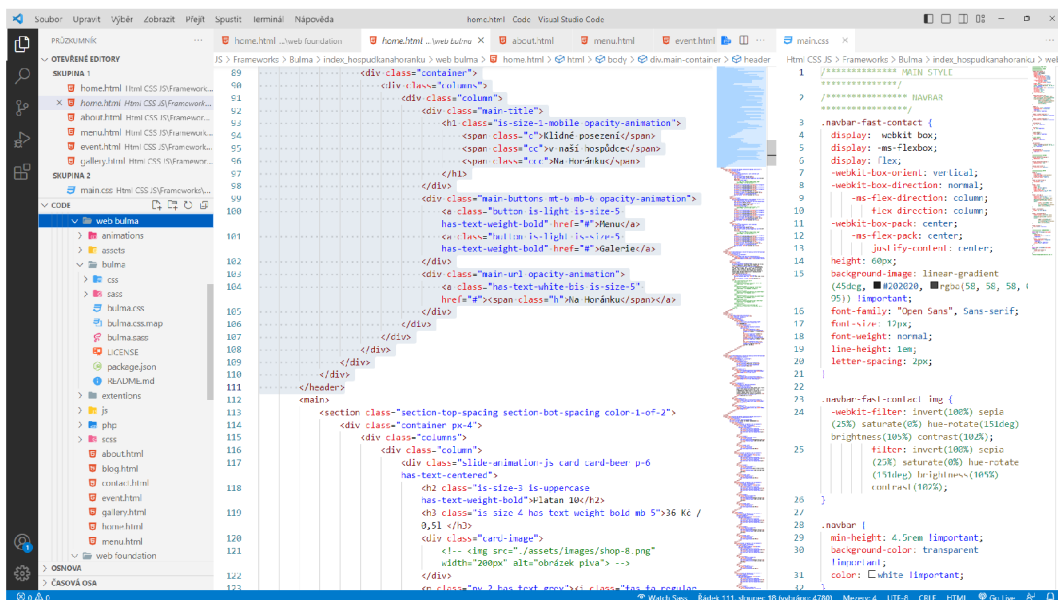
4.1 Visual Studio Code

Visual Studio Code je bezplatný, lehký, ale výkonný editor zdrojového kódu, který běží na počítači i na webu a je k dispozici pro operační systémy Windows, macOS, Linux a Raspberry Pi. Má vestavěnou podporu pro JavaScript, TypeScript a Node.js a bohatý ekosystém rozšíření pro další programovací jazyky (například C++, C#, Java, Python, PHP a Go), runtimey (například .NET a Unity), prostředí (například Docker a Kubernetes) a cloudy (například Amazon Web Services, Microsoft Azure a Google Cloud Platform).[14]

Kromě toho, že Visual Studio Code je lehké, anglicky označováno „lightweight“ a rychle začíná, má doplňování kódu IntelliSense pro proměnné, metody a importované moduly, grafické ladění, linting, editaci pomocí více kurzorů, ná-

povědy parametrů a další výkonné editační funkce, elegantní navigaci v kódu, refaktoring a integrovanou správu zdrojového kódu včetně podpory Git. Mnohé z toho bylo převzato z technologie Visual Studio.[15]

Vlastní Visual Studio Code je vytvořeno pomocí shellu Electron, Node.js, TypeScriptu a protokolu Language Server a je aktualizováno každý měsíc. Mnoho rozšíření je aktualizováno tak často, jak je potřeba. Bohatost podpory jednotlivých programovacích jazyků a jejich rozšíření je různá, od jednoduchého zvýrazňování syntaxe a porovnávání závorek až po ladění a refaktoring. Pokud nemáte k dispozici žádný jazykový server, můžete si základní podporu svého oblíbeného jazyka přidat prostřednictvím TextMate colorizérů. [16]



Obrázek 4: Vývojové prostředí Visual Studio Code

Praktická část

V praktické části bakalářské práce jsem vytvořil responzivní webovou stránku právě za použití zmíněného responzivního frameworku Bulma k zjištění reálných výhod a nevýhod frameworku při vývoji. Dále jsem vytvořil do jisté míry shodné jednotlivé komponenty, jak jen mi to samotný framework umožňoval pomocí dalších alternativních frameworků, jako je například Bootstrap, Foundation a Pure k porovnání frameworků mezi sebou.

5 Visual Studio Code

K vlastnímu vývoji webové stránky jsem použil vývojové prostředí Visual Studio Code (VSC) od společnosti Microsoft. Visual Studio Code je tvořeno jako open source projekt, to znamená, že je zdarma dostupný pro kohokoliv. Lze ho tedy bezplatně používat pro své konkrétní vývojové účely, a proto se stal vhodným nástrojem pro tvorbu mé webové stránky pomocí responzivního frameworku Bulma.

5.1 Instalace Visual Studio Codu

Jako uživatele operačního systému Windows od společnosti Microsoft jsem VSC nainstaloval standardním způsobem, a to stažením si VSC instalátoru z originální webové stránky poskytovatele produktu. Dále jsem provedl jednoduchou instalaci webového prostředí v operačním systému Windows skrze vestavěnou jednoduchou instalační nabídku.

5.2 Použití Visual Studio Codu

Pro použití VSC na vývoj webových stránek není potřeba instalovat dalšího softwaru nebo další alternativní doplňky pro podporu základních webových jazyků. VSC poskytuje základní podporu pro kódování HTML, CSS jazyka

a programování JavaScriptového jazyka ihned po instalaci. Kromě toho obsahuje také další alternativní funkce pro usnadnění vývoje, jako je například zvýraznění syntaxe, inteligentní dokončení, přizpůsobitelné formátování a obsahuje také podporu pro tzv. Emmeta, což nám umožňuje psaní kódu ve zkratkách. To znamená, že vývojové prostředí jsem nemusel nijak upravovat nebo doplňovat o další software.

Nicméně jsem využil dalších alternativních doplňků pro efektivní vývoj. Mezi ně patří například tzv. live server, který nám umožňuje živě promítat změny v nastaveném časovém zpoždění.

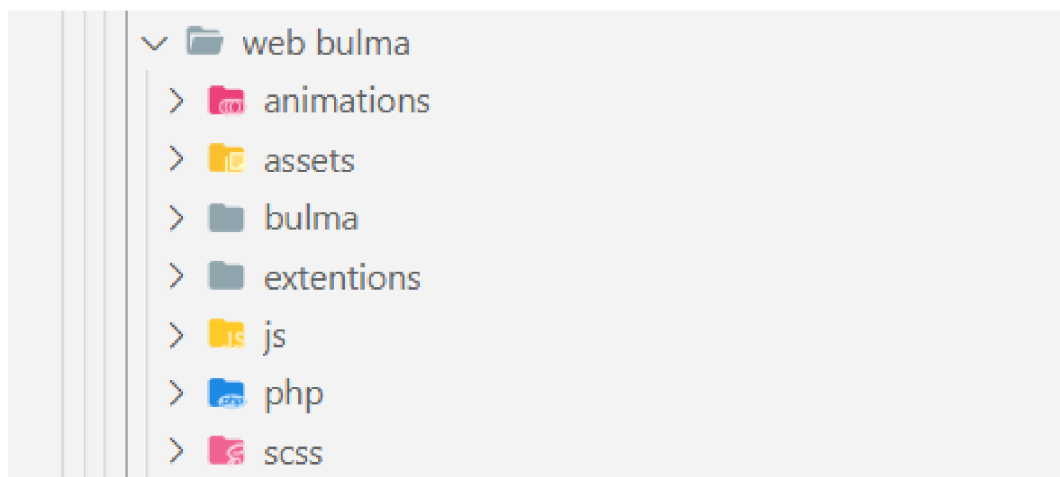
6 Vývoj webové stránky

Po instalaci a spuštění vývojového prostředí jsem začal se samotným vývojem webové stránky. Vytvořil jsem kořenový adresář pro ukládání veškerých souborů souvisejících s projektem a nazval jsem ho vlastním názvem reprezentujícím můj projekt.

6.1 Adresáře webové stránky

Dále jsem v kořenovém adresáři vytvořil další adresáře pro strukturované třídění informací, potažmo souborů. Tyto adresáře jsem nazval standardními názvy pro jednoznačnou rozeznatelnost obsahu v adresářích.

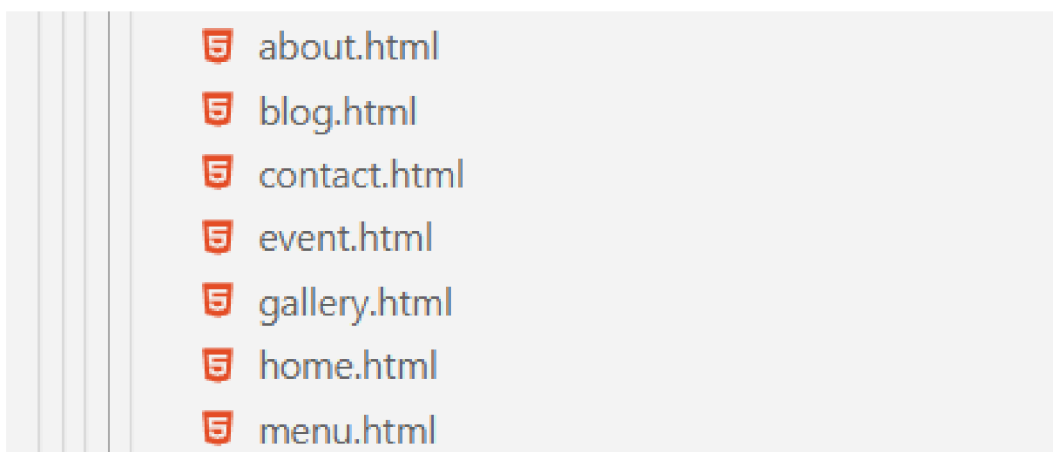
Adresáře s názvy „animations“ (animace), „assets“ (rekvizity), kde jsou uloženy další potřebné soubory pro tvorbu obsahu na stránce, jako jsou například obrázky, ikony, logo a podobně. Dále „extentions“ (rozšíření) pro případné doplňky nebo pluginy k vývoji, „js“ pro programový kód v jazyce JavaScript, „php“ pro programový kód v jazyce php, „scss“ pro generování CSS kódu k úpravám designu samotného frameworku Bulmy ke konkrétnímu vzhledu.



Obrázek 5: Ukázka kořenového adresáře webové stránky

6.2 Soubory webové stránky

V poslední řadě jsem vytvořil v kořenovém adresáři samotné webové soubory s názvy „home.html“ jako domovskou stránku, „menu.html“ jako stránku se základní nabídkou, „about.html“ jako stránku o nás, „blog.html“ jako stránku s blogem, „event.html“ jako stránku s událostmi, „gallery.html“ jako stránku prezentující obrázky a v poslední řadě „contact.html“ jako kontaktní stránku.



Obrázek 6: Ukázka HTML webových souborů

6.3 Instalace frameworku Bulma

Pro webový vývoj lze Bulmu instalovat čtyřmi základními způsoby. Jedním ze způsobů je pomocí HTML tagu tzv. nalinkovat onlinový zdroj neboli server z globálního internetu nebo z CDN (Content Delivery Network) serveru v blízkosti vašeho umístění. Tyto servery poskytují stažení binárního souboru samotného frameworku, stávají se tzv. online úložištěm.

Dále je možné onlinový zdroj frameworku importovat pomocí tzv. import příkazu přímo do CSS stylů vašeho projektu. Pro tyto dvě formy musí být vždy aktivní onlinový zdroj frameworku. V případě, že onlinový zdroj z jakéhokoliv důvodu spadne nebo není dostupný, je web zobrazován jen v základním stylu HTML kódu. Z tohoto důvodu je také možné nalinkovat nebo naimportovat

framework lokálně.

Bulmu jsem si stáhnul z dokumentace webové stránky vývojáře frameworku do svého kořenového adresáře a pomocí HTML tagu viz. příklad č. 5 na řádku č. 4 jsem ho lokálně nalinkoval v hlavičce HTML souboru. Na této webové stránce jsem k použití frameworku nepotřeboval žádnou další instalaci. Tento postup jsem zvolil i pro své další HTML soubory, potažmo stránky.

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <link rel="stylesheet" href="./bulma/bulma.css">
5   </head>
6 </html>

```

Příklad 5: Ukázka nalinkování Bulmy do projektu

6.4 Použití frameworku Bulma

CSS framework Bulma se používá standardním způsobem pro CSS frameworky. Vývojář využívá dokumentace frameworku k použití HTML kódu a předdefinovaných CSS tříd. HTML kód implementuje do svého webu, případně navrhne svůj vlastní HTML kód a ten doplňuje o zmíněné CSS třídy frameworku.

```

1 <div class="column">
2   <div class="card card-beer has-text-centered p-6">
3     <h2 class="is-size-3 is-uppercase has-text-weight-
4     bold">Platan 11</h2>
5     <h3 class="is-size-4 has-text-weight-bold mb-5">36 K
6     č / 0,51 </h3>
7     <p class="py-2 has-text-grey">Vestibulum</p>
8     <p class="py-2 has-text-grey">Vestibulum</p>
9     <p class="py-2 has-text-grey">Vestibulum</p>
10    <p class="py-2 mb-5 has-text-grey">Vestibulum</p>

```

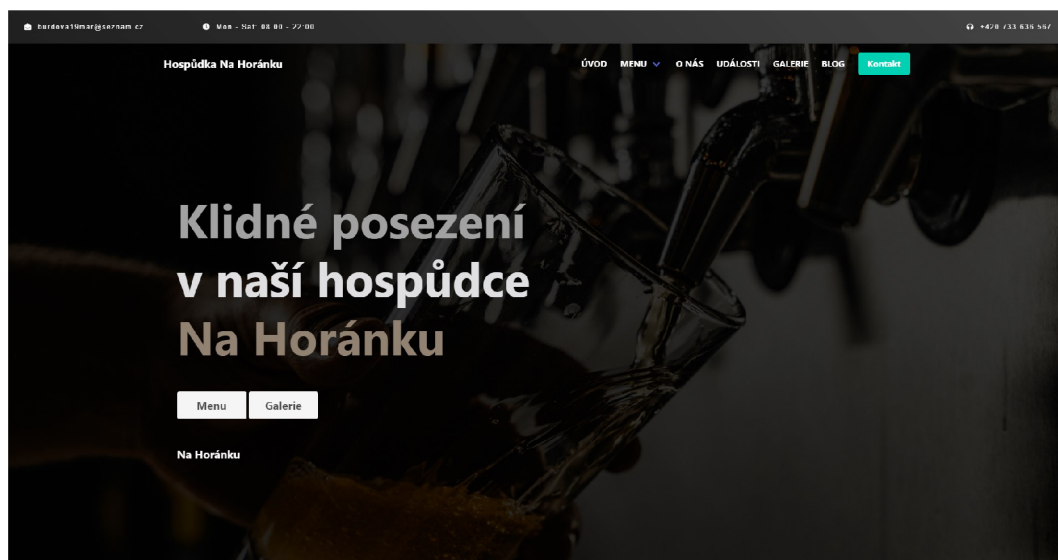


```
9     <a class="button py-2 has-background-dark has-text-  
10     white" href="#">Dozvědět se více</a>  
11 </div>
```

Příklad 6: Ukázka implementace CSS Bulmy do HTML souborů

6.5 Vývoj webové stránky pomocí Bulmy

Vývoj webu jsem prováděl standardním způsobem od tvoření navigace, hlavičky a dále jednotlivých sekcí webu až po samotný konec stránky, a to patičku.



Obrázek 7: Ukázka úvodní strany webové stránky vytvořené Bulmou

6.5.1 Začátek HTML dokumentu

HTML hlavička slouží k poskytnutí základních informací o webové stránce nebo webové aplikaci pro webové prohlížeče a jiné další programy, které zpracovávají obsah webových stránek. Obsahuje tzv. meta informace. Mezi základní meta informace patří například název stránky, popis obsahu stránky, klíčová

slova, autor a další. Tyto informace jsou důležité pro vyhledávače a další jiné nástroje.

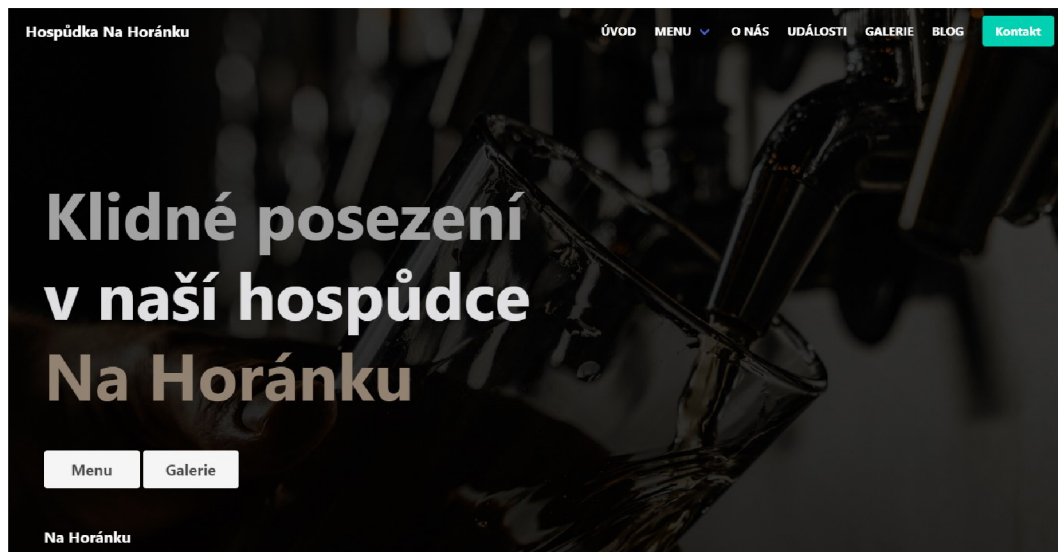
HTML hlavička je důležitou součástí každé webové stránky, protože pomáhá vyhledávačům a dalším nástrojům porozumět obsahu stránky a zlepšit její viditelnost a přístupnost.

```
1 <html lang="en">
2 <head>
3   <meta charset="UTF-8">
4   <meta http-equiv="X-UA-Compatible" content="IE=edge">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta name="author" content="Jakub Burda">
7   <meta name="description" content="Tradiční česká hospůdka s rodinnou atmosférou v okolí Lhenic s venkovním posezením.">
8   <meta name="keywords" content="Hospůdka Na Horánku, hospoda, hospůdka, Horánek, tradiční hospody, turistika na Horánku, pivní menu,
9     nejlepší hospody v ČR">
10  <link rel="stylesheet" href="/scss/main.css">
11  <link rel="stylesheet" href="/scss/home.css">
12  <link rel="stylesheet" href="/animations/animations.css">
13  <link rel="stylesheet" href="/bulma/bulma.css">
14  <title>Hospůdka na Horánku</title>
15  <link rel="icon" type="image/x-icon" href="/assets/favicon/icons8-pub-flaticons-lineal-color-32.png">
16  <script defer src="/js/animations.js"></script>
</head>
```

Obrázek 8: Ukázka nalinkování Bulmy na řádku č. 12

6.5.2 Navigační menu

Navigační menu jsem zhotovil pomocí dokumentace Bulmy. Využil jsem dostupný HTML kód pro vytvoření navigačního menu s nadpisem, s odkazy na další webové stránky webu a jedním tlačítkem jako odkaz na kontakt.



Obrázek 9: Ukázka designu navigačního menu Bulmy

Celé navigační menu poskytuje Bulma v dokumentaci. Kromě HTML kódu pro menu Bulma zde poskytuje i integraci JavaScriptového kódu tzv. hamburgerového menu pro mobilní telefony. Předdefinovaný HTML jsem implementoval do své webové stránky a přizpůsobil jsem ho vlastním potřebám vzhledu.

```

1 <nav class="navbar" role="navigation" aria-label="main
  navigation">
2 <div class="navbar-brand">
3   <a class="navbar-item" href="./index.html"><p>Hospůdka
  Na Horánku</p></a>
4   <a role="button" class="navbar-burger" aria-label="menu"
  aria-expanded="false" data-target="navbarBasicExample">
5     <span aria-hidden="true"></span>
6     <span aria-hidden="true"></span>
7     <span aria-hidden="true"></span>
8   </a>
9 </div>
10 <div id="navbarBasicExample" class="navbar-menu">

```

```

11 <div class="navbar-end">
12   <a href="/home.html">úvod</a>
13   <div class="navbar-item has-dropdown is-hoverable">
14     <a href="/menu.html">menu</a>
15     <div class="navbar-dropdown">
16       <a href="/menu.html#alcohol">alkoholické ná
17 poje</a>
18       <a href="/menu.html#non-alcohol">
19 nealkoholické nápoje</a>
20       <a href="/menu.html#food">občerstvení</a>
21     </div>
22   </div>
23   <a href="/about.html">o nás</a>
24   <a href="/event.html">události</a>
25   <a href="/gallery.html">galerie</a>
26   <a href="/blog.html">blog</a>
27 </div>
28 <div class="navbar-start">
29   <div class="navbar-item">
30     <a href="/contact.html">
31       <strong>Kontakt</strong>
32     </a>
33   </div>
34 </div>
</nav>

```

Příklad 7: HTML implementace navigace pomocí Bulmy

Integraci JavaScriptového kódu lze provést dvěma způsoby, a to vnořením přímo JavaScriptu do HTML souboru pomocí HTML tagu `<script>` nebo pomocí nalinkování JavaScriptového souboru pomocí tagu `<script>` s atributem „src“ a v něm odkaz na JavaScriptový soubor.

Pro svůj projekt jsem zvolil variantu samostatného JavaScriptového souboru, aby byl projekt lépe strukturovaný, případně pro snazší budoucí implementaci dalšího JavaScriptového kódu.

```
1 document.addEventListener('DOMContentLoaded', () => {
2
3   // Get all "navbar-burger" elements
4   const $navbarBurgers = Array.prototype.slice.call(document
5     .querySelectorAll('.navbar-burger'), 0);
6
7   // Add a click event on each of them
8   $navbarBurgers.forEach( el => {
9
10    // Get the target from the "data-target" attribute
11    const target = el.dataset.target;
12    const $target = document.getElementById(target);
13
14    // Toggle the "is-active" class on both the "navbar-
15    burger" and the "navbar-menu"
16    el.classList.toggle('is-active');
17    $target.classList.toggle('is-active');
18  });
19 });
20 });
```

Příklad 8: Úprava menu pomocí JavaScriptové integrace Bulmy

Celou navigaci jsem poté vnořil do tagu `<header>`, protože jsem se rozhodl, že navigace bude součástí hlavičky. Nicméně dalším částem hlavičky se budu věnovat v další podkapitole. Do hlavičky jsem vnořil tag `<div>` s atributem třídy. Tuto třídu jsem nazval `header-overlay`. Tato třída je má vlastní a je

použita jako tzv. overlay (pokrytí) přes celou hlavičku. Tento div je jen můj vlastní designový prvek, a proto ho nebudu dále rozebírat. Celou navigaci jsem poté vnořil ještě do posledního divu se třídou `container` pro zarovnání prvků zhruba na dvě třetiny stránky. Tento kontejner budu používat ještě velmi často v dalších podkapitolách, protože je to jedna ze základních tříd pro zarovnávání prvků.

```
1 <header>
2   <div class="header-overlay"></div>
3   <div class="container">
4     <!-- Navigation -->
5   </div>
6 </header>
```

Příklad 9: Vnoření navigace Bulmy do tagu `<header>`

V závěru jsem tuto navigaci nadesignoval pomocí tříd Bulmy pro úpravu textu. Na příkladu vidíme třídu `is-size-6`, která zvětšuje velikost písma, dále `has-text-white`, která obarvuje text na barvu bílou, dále `is-uppercase`, která nám z malých písmen tvoří písmena velká a třídu `has-text-weight-bold`, která nám písmo ztuční.

```
1 <div id="navbarBasicExample" class="navbar-menu">
2 <div class="navbar-end">
3   <a class="navbar-item is-size-6 is-uppercase has-text-
4     white has-text-weight-bold" href="/home.html">úvod</a>
5   <div class="navbar-item has-dropdown is-hoverable">
6     <a class="navbar-link is-size-6 is-uppercase has-
7       text-weight-bold has-text-white" href="/menu.html">menu<
8     /a>
9     <div class="navbar-dropdown">
10      <a class="navbar-item has-text-white" href="/
11        menu.html#alcohol">alkoholické nápoje</a>
12      <a class="navbar-item has-text-white" href="/
```

```
menu.html#non-alcohol has-text-white">nealkoholické ná  
poje</a>  
9         <a class="navbar-item has-text-white" href="./  
menu.html#food">občerstvení</a>  
10        </div>  
11    </div>  
12    <a class="navbar-item is-size-6 is-uppercase has-text-  
white has-text-weight-bold" href="./about.html">o nás</a>  
13    <a class="navbar-item is-size-6 is-uppercase has-text-  
white has-text-weight-bold" href="./event.html">události<  
/a>  
14    <a class="navbar-item is-size-6 is-uppercase has-text-  
white has-text-weight-bold" href="./gallery.html">galerie  
</a>  
15    <a class="navbar-item is-size-6 is-uppercase has-text-  
white has-text-weight-bold" href="./blog.html">blog</a>  
16 </div>  
17 <div class="navbar-start">  
18     <div class="navbar-item">  
19         <div class="buttons">  
20             <a href="./contact.html" class="button is-  
primary">  
21                 <strong>Kontakt</strong>  
22             </a>  
23         </div>  
24     </div>  
25 </div>  
26 </div>
```

Příklad 10: Designování prvků navigace hlavičky pomocí Bulmy

6.5.3 Hlavička

Pro hlavičku jsem chtěl navrhnout něco jednoduššího a zároveň zajímavého. Design navigace jsem se rozhodl spojit s designem hlavičky. Celou navigaci jsem proto vložil do tagu `<header>` a nastavil jsem navigaci transparentní pozadí.



Obrázek 10: Ukázka designu hlavičky pomocí Bulmy

```
1 <header>
2   <div class="header-overlay"></div>
3   <div class="container">
4     <!-- Navigation -->
5   </div>
6   <div class="container">
7     <!-- Second part of header -->
8   </div>
9 </header>
```

Příklad 11: Vnoření druhé části hlavičky do tagu `<header>`

Druhou část hlavičky jsem také umístil do kontejneru pomocí divu a třídy `container` pro stejné zarovnání obsahu. Do tohoto kontejneru jsem začal vno-

řovat obsah pro svoji hlavičku. Tím byl tag `<h1>` hlavního nadpisu, dále `div`, do kterého jsem vnořil tlačítka jako odkaz na galerii a menu a posledním vnořením do kontejneru byl opět `div` s ještě jedním tagem odkazu.

```

1 <div class="container">
2   <h1 class="is-size-1-mobile">
3     <span>Klidné posezení</span>
4     <span>v naší hospůdce</span>
5     <span>Na Horánku</span>
6   </h1>
7   <div class="main-buttons mt-6 mb-6">
8     <a href="#">Menu</a>
9     <a href="#">Galerie</a>
10  </div>
11  <div class="main-url">
12    <a href="#"><span class="h">Na Horánku</span></a>
13  </div>
14 </div>

```

Příklad 12: Vnoření HTML obsahu druhé části hlavičky

Tyto `div`y a tagy jsem nadesignoval zejména pomocí Bulmy. Bulma mi však úplně neumožňovala design hlavního nadpisu a samotné celé hlavičky, který byl v mé představě. Hlavní nadpis a hlavičku jsem proto musel nadesignovat pomocí vlastního CSS kódu.

```

1 <div class="container">
2   <h1 class="is-size-1-mobile">
3     <span class="has-text-grey-light">Klidné posezení</span>
4     <span class="has-text-light">v naší hospůdce</span>
5     <span class="has-text-primary">Na Horánku</span>
6   </h1>
7   <div class="mt-6 mb-6">

```

```

8     <a href="#">Menu</a>
9     <a href="#">Galerie</a>
10  </div>
11  <div>
12     <a href="#"><span class="has-text-grey-light">Na Hor
ánku</span></a>
13  </div>
14 </div>

```

Příklad 13: Design druhé části hlavičky pomocí Bulmy

```

1 header {
2     height: 95vh;
3     background-image: url( '../assets/carousels/beer-carousel
.jpg ' );
4     background-position: top;
5     background-repeat: no-repeat;
6     background-size: cover;
7 }
8 h1 {
9     font-weight: bold;
10    font-size: 85px;
11    line-height: 105px;
12 }
13 h1 span {
14    display: block;
15 }

```

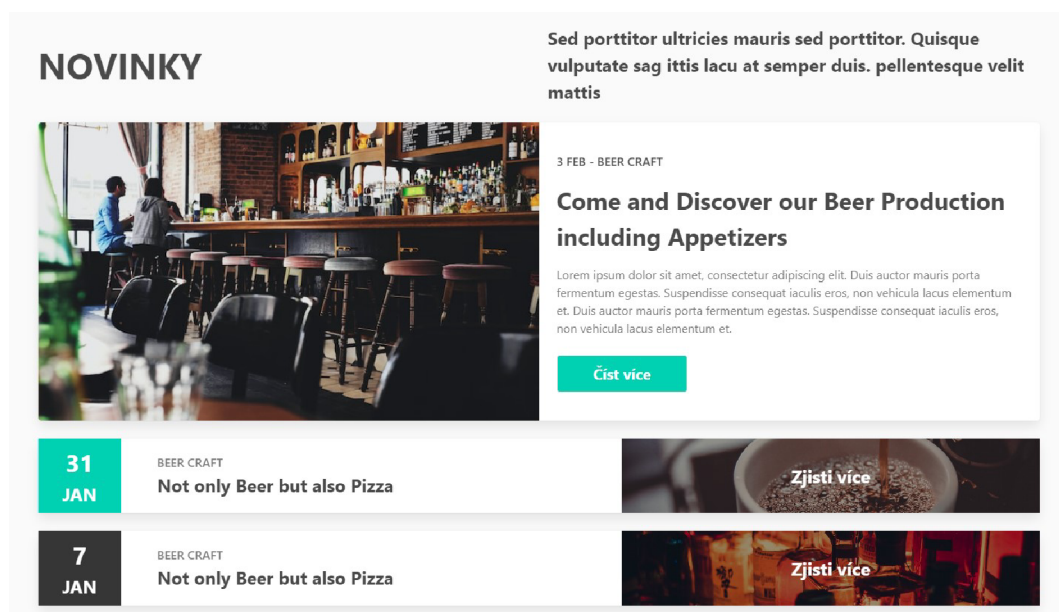
Příklad 14: Úprava hlavičky pomocí vlastního CSS kódu

V poslední řadě jsem tagu hlavního nadpisu přidal třídu `is-size-1-mobile`, abych zajistil responzivního chování na mobilních zařízeních.

6.5.4 Sekce

Design sekcí jsem nijak moc dopředu nepromýšlel a spíše jsem využil své fantazie a zkušeností z předešlých webových vývojů. Rozhodl jsem se pro každou jednotlivou stránku ze své navigace vytvořit několik základních sekcí pro prezentaci webového obsahu. Z těchto sekcí budu rozebírat pouze jednu konkrétní sekci, kterou jsem určil jako přínosnou pro bakalářskou práci, jelikož popis kompletního vývoje webu by byl velmi rozsáhlý a nepodstatný.

Sekce, kterou jsem vybral ze své praktické části pro reprezentaci vývoje pomocí Bulmy, byla sekce novinky. Mojí představou bylo vytvořit sekci o více řádcích a sloupcích se základními prvky, jako je nadpis sekce, základní text, tlačítko a základní informace k samotným novinkám.



Obrázek 11: Sekce webové stránky novinky vytvořena pomocí Bulmy

Jako první jsem si vytvořil hlavní obal celé sekce pomocí HTML tagu `<section>`. Do této sekce jsem vnořil `div` s atributem třídy. Tuto třídu jsem nazval `news-section` pro jednoznačné rozlišení sekce a pro případné další používání této části sekce. Dále jsem vnořil další `div` s třídou `container` pro

využití dvou třetin stránky jako u všech ostatních sekcí. Poté jsem si sekci rozdělil na mřížku 4x2. Mřížku čtyř řádků a dvou sloupců jsem rozdělil stejně pomocí divů s třídami `columns` a vnořenými divy s třídami `column`.

```
1 <body>
2   <section>
3     <div class="news-section">
4       <div class="container">
5         <div class="columns">
6           <div class="column">...</div>
7           <div class="column">...</div>
8         </div>
9         <div class="columns">
10          <div class="column">...</div>
11          <div class="column">...</div>
12        </div>
13        <div class="columns">
14          <div class="column">...</div>
15          <div class="column">...</div>
16        </div>
17        <div class="columns">
18          <div class="column">...</div>
19          <div class="column">...</div>
20        </div>
21      </div>
22    </div>
23  </section>
24 </body>
```

Příklad 15: Rozdělení sekce novinky pomocí Bulmy

Dále jsem začal vyplňovat jednotlivé řádky a sloupce webovým obsahem. V prvním řádku jsem se zaměřil na úvodní nadpis sekce pomocí HTML tagu `h2` nadpisu druhé úrovně a základním textem pro popis sekce pomocí HTML tagu `p`.

pro paragraf. Tyto tagy jsem poté nadesignoval pomocí dalších typografických tříd Bulmy, a to `is-size-1` pro větší velikost písma, dále třídou `is-uppercase` pro změnu z malých písmen na písmena velká, dále `has-text-weight-bold` pro tučnější písmo v sekci a `is-size-4` pro změnu velikosti písma paragrafu. V příkladu vidíme i třídu s názvem `is-vcentered`, která mi zajistila vertikální vycentrování obsahu sloupců.

```

1 <div class="columns is-vcentered">
2   <div class="column">
3     <h2 class="is-size-1 is-uppercase has-text-weight-
4 bold">News</h2>
5   </div>
6   <div class="column">
7     <p class="is-size-4 has-text-weight-bold">Sed
8 porttitor ultricies mauris sed porttitor. Quisque
vulputate sagittis lacu at semper dui. Pellentesque
velit mattis</p>
9   </div>
10 </div>

```

Příklad 16: První řádek sekce novinky pomocí Bulmy

Následně jsem začal vyplňovat obsahem druhý řádek rozdělené mřížky sekce. V prvním sloupci jsem umístil HTML tag pro obrázek. V druhém sloupci jsem poté umístil nadpis druhé úrovně, dále nadpis třetí úrovně, paragraf a tag pro odkaz jako budoucí tlačítko. Tyto prvky jsem nadesignoval pomocí tříd `has-text-weight-semibold` pro mírně tučné písmo, dále `has-text-grey` pro změnu barvy textu na šedivou a `has-text-weight-bold` pro tučné písmo. Nejvíce změn jsem provedl na samotném tagu pro odkaz, aby tag dostal podobu tlačítka. Zde jsem využil třídy `button` jako základní třídu Bulmy pro tlačítko, dále `is-size-5` určující velikost tlačítka, dále `has-background-primary` pro nastavení barevného pozadí tlačítka a `has-text-white` pro změnu barvy textu v tlačítku. V poslední řadě jsem prvky od sebe mírně začal odsazovat pomocí

třídy `mt-4`, která přiřadí tagu tzv. `margin-top` (odsazení shora) o velikosti 1 rem.

```

1 <div class="columns">
2   <div class="column">
3     
5   </div>
6   <div class="column">
7     <h2 class="has-text-weight-semibold mt-4 ">3 FEB —
8     BEER CRAFT</h2>
9     <h3 class="is-size-3 has-text-weight-bold mt-4 ">
10    Come and Discover our Beer Production including
11    Appetizers</h3>
12    <p class="has-text-grey mt-4">Lorem ipsum dolor sit
13    amet, consectetur adipiscing elit. Duis auctor mauris
14    porta fermentum egestas. Suspendisse consequat iaculis
15    eros, non vehicula lacus elementum et. Duis auctor mauris
16    porta fermentum egestas. Suspendisse consequat iaculis
17    eros, non vehicula lacus elementum et.</p>
18    <a class="button is-size-5 mt-5 has-background-
19    primary has-text-white has-text-weight-bold px-6" href="#"
20    ">Read more</a>
21  </div>
22 </div>

```

Příklad 17: Druhý řádek sekce novinky pomocí Bulmy

Poté jsem druhý řádek sekce chtěl trochu zvýraznit, a tak jsem využil třídy `card`, která z celého řádku vytvoří kartu s pozadím a mírným stínem. Tuto třídu jsem navázal na už použitý `div` se třídou `columns`. Karta byla s nadefinovaným vnitřním rámcem (`padding`). Tento `padding` jsem nechtěl zachovat, a proto jsem v dokumentaci vyhledal další třídu, a to třídu `is-gap-less`, která

mi poskytla možnost se zbavit paddingu.

```

1 <div class="columns card is-gapless news-content">
2   <div class="column">
3     <!-- Column content -->
4   </div>
5   <div class="column">
6     <!-- Column content -->
7   </div>
8 </div>

```

Příklad 18: Úpravy druhého řádku sekce novinky pomocí Bulmy

Závěrem jsem pro tento řádek pevně upravil šířku celého sloupce pomocí vlastní CSS třídy. Tuto třídu jsem nazval `news-content` a přiřadil jí k divu s třídou `columns`, abych šířku aplikoval na celý řádek. V dalším příkladu vidíme CSS kód pro úpravu šířky sloupce.

```

1 /****** SECTION NEWS *****/
2 .news-section {
3   height: 400px;
4 }

```

Příklad 19: CSS Úprava druhého řádku sekce novinky

V poslední části této sekce jsem designoval poslední dva řádky z původních čtyřech řádků. Z těchto dvou řádků budu popisovat pouze vývoj jednoho z nich, jelikož kód pro tyto dva řádky je téměř totožný.

V první řadě jsem se rozhodl rozdělit třetí i čtvrtý řádek na tři další sloupce z původní mřížky 4x2. První sloupec pro datum konání novinky, druhý sloupec pro popis novinky a třetí sloupec pro ukázkou a tlačítko jako přesměrování na novinku.

```

1 <div class="columns">
2   <div class="column"> ... </div>
3   <div class="column"> ... </div>

```

```

4 </div>
5 <div class="columns">
6   <div class="column"> ...</div>
7   <div class="column"> ...</div>
8 </div>
9 <div class="columns">
10  <!-- Changes -->
11  <div class="column"> ...</div>
12  <div class="column"> ...</div>
13  <div class="column"> ...</div>
14 </div>
15 <div class="columns">
16  <!-- Changes -->
17  <div class="column"> ...</div>
18  <div class="column"> ...</div>
19  <div class="column"> ...</div>
20 </div>

```

Příklad 20: Úprava rozdělení posledních řádků sekce novinky pomocí Bulmy

U posledních dvou řádků jsem využil stejných metod. Do jednotlivých sloupců jsem začal vnořovat HTML tagy pro obsah. Tyto tagy jsem nadesignoval pomocí typografických tříd Bulmy. Kromě těchto tříd jsem zde využil další už použité třídy, jako například třídu `card`, `is-gapless`, `is-vcentered`, které byly už zmíněny v úvodní části této sekce.

Mezi novými typografickými třídami pro tagy v této konkrétní sekci se objevila třída `has-text-centered` pro horizontální vycentrování textu. Dále jsem zde použil i další nové netyografické třídy, a to `p-2`, která mi vytvořila vnitřní rám (`padding`) o velikosti 0.5 rem, dále třídu `p1-6`, která mi vytvořila taktéž `padding` o velikosti 3 rem, ale pouze z levé strany jako mírné odsazení pro popis novinky.

Design sloupců jsem měl skoro hotový, nicméně třída `column` zajišťuje rov-

noměrnou šířku sloupců. Tato technika díky různé délce obsahu ve sloupcích se mi zde zrovna nehodila, a proto jsem použil další třídy pro změnu délky jednotlivého sloupce. V příkladu můžeme vidět třídu `is-1-desktop`, `is-6-desktop`.

Abychom pochopili, jak tyto třídy fungují, musíme vědět, že každý kontejner (`container`) sloupců je rozdělený na 12 sloupečků. Třída `is-6-desktop` vlastně znamená, že konkrétní sloupec s touto třídou bude zabírat polovinu těchto kontejnerových sloupečků, tedy 50 % kontejneru.

```
1 <div class="columns is-gapless card is-vcentered">
2   <div class="column is-1-desktop has-background-primary">
3     <div class="news-date has-text-centered p-2">
4       <h2 class="is-size-3 is-uppercase has-text-white
5         has-text-weight-bold">31</h2>
6       <h3 class="is-size-4 is-uppercase has-text-white
7         has-text-weight-bold">jan</h3>
8     </div>
9   </div>
10  <div class="column is-6-desktop">
11    <div class="new-describe pl-6">
12      <h2 class="has-text-grey is-uppercase has-text-
13        weight-semibold">beer craft</h2>
14      <h3 class="is-size-4 has-text-weight-bold">Not
15        only Beer but also Pizza</h3>
16    </div>
17  </div>
18  <div class="column">
19    <div class="news-image-overlay"></div>
20    <a class="is-size-4 has-text-weight-bold has-text-
21      white" href="#">
22      <div class="news-image">Read more</div>
23    </a>
24  </div>
```

```
20 </div>
```

Příklad 21: Design třetího a čtvrtého řádku sekce novinky pomocí Bulmy

Do posledního sloupce jsem vnořil další div bez jakéhokoliv obsahu. Tomuto divu jsem dal třídu `news-image-overlay`. Tato třída byla vytvořena mnou jako lehký overlay přes pozadí posledního sloupce. Tuto funkci jsem si musel napsat sám, jelikož Bulma takový overlay neobsahuje.

```
1 <div class="column">
2   <div class="news-image-overlay"></div>
3   <a class="is-size-4 has-text-weight-bold has-text-white"
4     href="#">
5     <div class="news-image">Read more</div>
6   </a>
7 </div>
```

Příklad 22: Vlastní HTML kód pro tzv. overlay posledních řádků sekce novinky

```
1 .news-image-overlay {
2   position: absolute;
3   width: 100%;
4   height: 100%;
5   background-color: #2d2d2d;
6   opacity: 0.5;
7   mix-blend-mode: multiply;
8 }
```

Příklad 23: Vlastní CSS kód pro tzv. overlay posledních řádků sekce novinky

Závěrem jsem vnořil obsah posledního sloupce do jednoho divu s vlastní třídou `news-more` a vlastním identifikátorem `news-image-1`. Třídu `news-more` jsem využil pro roztážení třetího sloupce na šířku celého řádku, abych mohl pomocí již zmíněného identifikátoru `news-image-1` vložit do pozadí obrázek. Tento design jsem musel provést svým kódem.

```

1 <div class="column">
2   <div id="news-image-1" class="news-more">
3     <div class="news-image-overlay"></div>
4     <a class="is-size-4 has-text-weight-bold has-text-
white" href="#">
5       <div class="news-image">Read more</div>
6     </a>
7   </div>
8 </div>

```

Příklad 24: Úprava sekce novinky třídou news-image třetího a čtvrtého řádku

```

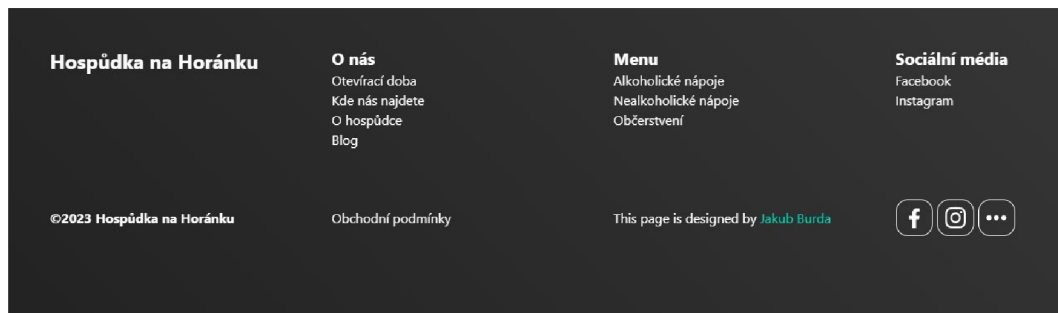
1 .news-section #news-image-1 {
2   background-image: url("../assets/images/caffe.jpg");
3 }
4 .news-section #news-image-2 {
5   background-image: url("../assets/images/alcohol.jpg");
6 }
7 .news-section .news-more {
8   position: relative;
9   background-position: center;
10  background-repeat: no-repeat;
11  background-size: cover;
12 }

```

Příklad 25: CSS přidání obrázku do třetího a čtvrtého řádku sekce novinky

6.5.5 Patička

Design patičky jsem si předem promyslel a navrhnul dle svých potřeb pro snažší budoucí vývoj této komponenty pomocí Bulmy. Rozhodl jsem se vytvořit patičku o matici 2x4. S úvodním logem, nadpisy a dalším webovým obsahem pro patičky.



Obrázek 12: Ukázka designu patičky pomocí Bulmy

Pro vytvoření patičky webu jsem použil HTML tag `<footer>`, který mi zaobaluje celý obsah patičky. Dále jsem potřeboval obsah patičky zarovnat na dvě třetiny stránky a tento obsah vycentrovat jako i u jiných dalších sekcí. I zde jsem si vytvořil div vnořený v patičce a k zarovnání prvků jsem tomuto divu přidal třídu s názvem `container`. Poté jsem rozdělil kontejner na další jednotlivé sloupce. K tomu jsem použil vnořené třídy `columns` a `column`. Pro vytvoření mřížky 1x4 jsem tedy využil jeden div se třídou `columns` a vnořil 4 další divy se třídou `column`.

```

1 <body>
2   <footer>
3     <div class="container">
4       <div class="columns">
5         <div class="column">...</div>
6         <div class="column">...</div>
7         <div class="column">...</div>
8         <div class="column">...</div>
9       </div>
10    </div>
11  </footer>
12 </body>

```

Příklad 26: Rozdělení patičky pomocí Bulmy na mřížku 1x4

Dále jsem začal patičku stránky doplňovat o obsah. Jednotlivé sloupce jsem začal plnit dalšími konkrétními HTML tagy pro zveřejnění obsahu, který měla patička prezentovat. To znamená, že divy s třídami `column` jsem začal vyplňovat konkrétním obsahem. V příkladu vidíme HTML tag pro paragraf a vnořený další tag pro odkaz.

```
1 <div class="column">
2   <p>O nás</p>
3   <p><a href="#">Otevírací doba</a></p>
4   <p><a href="#">Kde nás najdete</a></p>
5   <p><a href="#">O hospůdce</a></p>
6   <p><a href="#">Blog</a></p>
7 </div>
8 <div class="column">
9   <p>Menu</p>
10  <p><a href="#">Nápojový lístek</a></p>
11  <p><a href="#">Občerstvení</a></p>
12 </div>
13 <div class="column">
14  <p>Sociální média</p>
15  <p><a href="#">Facebook</a></p>
16  <p><a href="#">Instagram</a></p>
17 </div>
18 <div class="column">
19  <p>Sociální média</p>
20  <p><a href="#">Facebook</a></p>
21  <p><a href="#">Instagram</a></p>
22 </div>
```

Příklad 27: Vyplňování obsahu sloupcové mřížky Bulmy

V dalším kroku jsem začal HTML tagy vnořené do sloupců designovat ke své představě. K tomu jsem využil další CSS třídy poskytované Bulmou. V pří-

kladu vidíme třídu `is-size-5` určující velikost prvního paragrafu, dále vidíme například třídu `has-text-weight-bold`, která paragraf ztuční nebo třídu `has-text-white-bis`, která změní barvu odkazu na bílou. Kromě toho HTML tag pro odkaz obsahuje i samotný atribut odkazu, který na webové stránce odkazuje na alternativní stránky webu. Tento atribut není v příkladu vyplněn pro jednoduchost příkladu.

```

1 <p class="has-text-weight-bold is-size-5">O nás</p>
2 <p><a class="has-text-white" href="#">Otevírací doba</a></p>
3 <p><a class="has-text-white" href="#">Kde nás najdete</a></p>
  >
4 <p><a class="has-text-white" href="#">O hospůdce</a></p>
5 <p><a class="has-text-white" href="#">Blog</a></p>

```

Příklad 28: Design patičky webové stránky pomocí Bulmy

Poté jsem vytvořil druhou část patičky pro tzv. copyright, obchodní podmínky, tvůrce a ikony sociálních sítí. Pro vytvoření těchto sloupců v jednom řádku jsem využil znova rozdělení v podobě už několikrát zmíněného divu s atributem třídy a názvem třídy `columns`. Do divu `columns` jsem vnořil další 4 divy s názvem třídy taktéž `columns` pro vytvoření stejné mřížky 1x4 jako u první části patičky. Abych zarovnal tyto sloupce pod předchozí sloupce, umístil jsem celý div `columns` s vnořenými divy pod div `columns` v první části patičky. Tím jsem vytvořil vlastně mřížku 2x4.

```

1 <body>
2   <footer>
3     <div class="container">
4       <div class="columns">
5         <div class="column">...</div>
6         <div class="column">...</div>
7         <div class="column">...</div>
8         <div class="column">...</div>
9       </div>

```

```
10     <div class="columns">
11         <div class="column">...</div>
12         <div class="column">...</div>
13         <div class="column">...</div>
14         <div class="column">...</div>
15     </div>
16 </div>
17 </footer>
18 </body>
```

Příklad 29: Rozdělení patičky pomocí Bulmy na mřížku 2x4

V posledním sloupci druhé části jsem umístil tři ikony sociálních sítí. Tyto ikony jsem chtěl seřadit vedle sebe. K tomu jsem použil znova vnoření dalšího řádkové divu s dalšími divy pro sloupec. Ve vývoji se tento jev nazývá tzv. nesting.

```
1 <div class="container">
2     <div class="columns">
3         <div class="column">...</div>
4         <div class="column">...</div>
5         <div class="column">...</div>
6         <div class="column">...</div>
7     </div>
8     <div class="columns">
9         <div class="column">...</div>
10        <div class="column">...</div>
11        <div class="column">...</div>
12        <div class="column">
13            <div class="columns">
14                <div class="column">...</div>
15                <div class="column">...</div>
16                <div class="column">...</div>
17            </div>
```

```

18         </div>
19     </div>
20 </div>

```

Příklad 30: Rozdělení posledního sloupce na další mřížku pomocí Bulmy

Dále jsem už pokračoval stejným způsobem, a to vnořením dalších HTML tagů pro obsah do sloupců druhé části patičky. Tyto tagy jsem poté nadesignoval stejně jako u předchozích částí patičky pomocí CSS tříd Bulmy. Na závěr bylo potřeba ještě zkontrolovat samotnou responzivitu designu patičky. Patičku jsem zkontroloval vývojářským nástrojem v prohlížeči pro kontrolu responzivity na jiných zařízeních. Patička byla skoro celá responzivní už jen díky systému `columns` a `column`. Jedinou úpravu, kterou jsem musel doladit, byla na druhé části patička v posledním sloupci, kde jsem zarovnával další tři sloupce vedle sebe. Díky třem ikonám s konkrétním designem docházelo k překrývání prvků přes sebe. To jsem vyřešil dalšími alternativními třídami `divů` pro úpravu responzivity, a to třídou `is-2-desktop` a `is-3-desktop`.

```

1 <div class="columns">
2     <div class="column"> ...</div>
3     <div class="column"> ...</div>
4     <div class="column"> ...</div>
5     <div class="column"> ...</div>
6 </div>
7 <div class="columns">
8     <div class="column"> ...</div>
9     <div class="column"> ...</div>
10    <div class="column"> ...</div>
11    <div class="column">
12        <div class="columns">
13            <div class="column is-2-desktop is-3-tablet"> ...
14        </div>
15        <div class="column is-2-desktop is-3-tablet"> ...

```



```
15     </div>
16         <div class="column is-2-desktop is-3-tablet">...
17     </div>
18 </div>
```

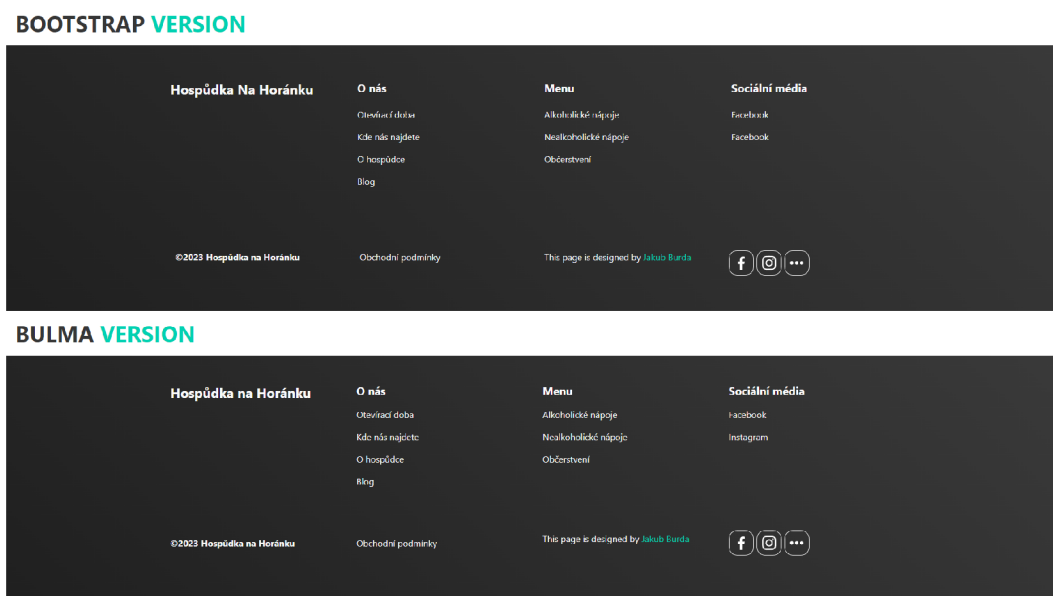
Příklad 31: Lazení responzivity Bulmy druhé části patičky

7 Vývoj komponent k porovnání

Pro porovnání jednotlivých frameworků jsem zvolil cestu vytvoření vždy dvou stejných komponent, vždy dvojici frameworků na jedné webové stránce. Vývoj celého webu jiným frameworkem by byl náročný a příliš zdlouhavý proces pro základní porovnání těchto frameworků.

7.1 Vývoj komponenty pomocí Bootstrapu

K porovnání Bulmy a Bootstrapu jsem zvolil vytvoření komponenty tzv. patičky. Patičku jsem vytvořil jak pomocí Bulmy, tak pomocí Bootstrapu, abych zjistil rozdíly mezi těmito konkrétními frameworky, a případně také jejich výhody a nevýhody.



Obrázek 13: Ukázka vytvořené komponenty Bulmou a Bootstrapem

7.1.1 Vývoj patičky pomocí Bootstrapu

Jelikož frameworky fungují na stejné bázi, tak se vývoj pomocí Bootstrapu nijak zvlášť neliší od vývoje pomocí frameworku Bulma. Zásadním klíčem pro můj vývoj byla Bootstrapová dokumentace, díky které jsem mohl prvky rozmisťovat dle své představy.

V první řadě jsem si vytvořil jednu hlavní sekci s tagem `<footer>`, která mi zaobaluje celý obsah patičky. Dále jsem potřeboval obsah patičky umístit zhruba na dvě třetiny stránky a tento obsah vycentrovat. Vytvořil jsem si div vnořený do tagu `<footer>` a tomu divu jsem dal Bootstrapovou třídu `container`. Dále jsem rozdělil celý vnitřní kontejner na 4 další sloupce. K tomu nám Bootstrap umožňuje vytvořit tzv. `row` neboli řádek a do něj vnořit další divy s názvy `col`, který nám reprezentují jednotlivé sloupce. Tím jsem získal tzv. matici neboli mřížku 1x4.

```
1 <body>
2   <footer>
```

```

3     <div class="container">
4         <div class="row">
5             <div class="col">...</div>
6             <div class="col">...</div>
7             <div class="col">...</div>
8             <div class="col">...</div>
9         </div>
10    </div>
11 </footer>
12 </body>

```

Příklad 32: Rozdělení patičky na mřížku 1x4 pomocí Bootstrapu

Dále jsem začal vnořovat další HTML tagy jako obsah jednotlivých sloupců. Příkladem je nadpis sloupce nebo také url odkazy. Tyto tagy jsem vnořoval do jednotlivých divů s názvem col, abych zachoval svou původně vytvořenou mřížku 1x4.

```

1 <div class="col">
2     <h6>Hospůdka na Horánku</h6>
3 </div>
4 <div class="col">
5     <h6>O nás</h6>
6     <p><a href="#">Otevírací doba</a></p>
7     <p><a href="#">Kde nás najdete</a></p>
8     <p><a href="#">O hospůdce</a></p>
9     <p><a href="#">Blog</a></p>
10 </div>
11 <div class="col">
12     <h6>Menu</h6>
13     <p><a href="#">Alkoholické nápoje</a></p>
14     <p><a href="#">Nealkoholické nápoje</a></p>
15     <p><a href="#">Občerstvení</a></p>
16 </div>

```

```

17 <div class="col">
18   <h6>Sociální média</h6>
19   <p><a href="#">Facebook</a></p>
20   <p><a href="#">Facebook</a></p>
21 </div>

```

Příklad 33: Vložení obsahu do jednotlivých sloupců Bootstrapové komponenty

Poté jsem jednotlivé prvky patičky designoval pomocí Bootstrapových tříd. Tyto třídy jsem přiřazoval jednotlivým tagům pomocí tzv. class atributu stejně tak jako u samotných divů, abych docílil své představy designu. Tyto třídy jsou součástí dokumentace. V příkladu vidíme typickou třídu `font-weight-bold`, která nám zajišťuje tučnost HTML nadpisu úrovně šest. Dále vidíme třídu `text-light`, která nám definuje barvu samotného textu nebo třídu `h5`, která nám určuje velikost nadpisu.

```

1 <h6 class="font-weight-bold mb-4 text-light h5">O nás</h6>
2 <p><a href="#" class="text-reset text-light">Otevírací doba
   </a></p>
3 <p><a href="#" class="text-reset text-light">Kde nás
   najdete</a></p>
4 <p><a href="#" class="text-reset text-light">O hospůdce</a>
   </p>
5 <p><a href="#" class="text-reset text-light">Blog</a></p>

```

Příklad 34: Designování obsahu jednoho sloupce patičky pomocí Bootstrapu

V poslední řadě jsem patičku přizpůsobil různým velikostem zařízení pomocí dalších tříd Bootstrapu. V podstatě jsem pomocí Bootstrapu nadefinoval, jak se budou sloupce zalamovat na různých velikostech zobrazovací plochy. V příkladu vidíme třídu `col-md-3`, která určuje, že tento konkrétní sloupec s touto třídou bude mít na zařízeních v rozlišení tzv. medium velikost 3 sloupečků. K tomu abych věděl, jak tento princip funguje, je nutné znát, že každý Bootstrapový kontejner je rozdělen na 12 sloupečků. Třídou `col-md-3` vlastně

říkáme, že tento sloupec bude zabírat jednu čtvrtinu kontejneru. Počet pixelů pro rozlišení medium určuje samotný Bootstrap.

```
1 <div class="col-md-3 col-lg-4 col-xl-3 mb-4">
2   <!-- Column content -->
3 </div>
4 <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-3">
5   <!-- Column content -->
6 </div>
7 <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-3">
8   <!-- Column content -->
9 </div>
10 <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-3">
11   <!-- Column content -->
12 </div>
```

Příklad 35: Přizpůsobení designu na více zařízeních pomocí Bootstrapu

Na designu si můžeme všimnout ještě druhé části patičky, a to úzkého pruhu na konci patičky, který obsahuje tzv. copyright. Tento copyright nebudu dále popisovat, jelikož jeho tvorba probíhala přesně jako u první části patičky, jen s vnořením jiného obsahu.

```

<!-- BOOTSTRAP VERSION -->
<h2 class="m-3 h1">Kontingenciální Bootstrap </h2>
<div class="text-light text-underline-underline">
  <div class="container text-white text-muted p-3">
    <div class="row mt-2">
      <div class="col-md-3 col-lg-4 col-xl-3 mb-4">
        <h3 class="font-weight-bold mb-4 h4">Hospůdka
        na Horáňku/h6>
      </div>
      <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-3">
        <h3 class="font-weight-bold mb-4 text-light h5">Staruc/h6>
        <span href="#" class="text-light">Otevírací doba/a></span>
        <span href="#" class="text-light">Kde nás najdete/a></span>
        <span href="#" class="text-light">Hodnocení/a></span>
        <span href="#" class="text-light">Blog/a></span>
      </div>
      <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-3">
        <h3 class="font-weight-bold mb-4 text-light h5">Staruc/h6>
        <span href="#" class="text-light">Alkoholické nápoje/a></span>
        <span href="#" class="text-light">Nealkoholické nápoje/a></span>
        <span href="#" class="text-light">Občerstvení/a></span>
      </div>
      <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-3">
        <h3 class="font-weight-bold mb-4 text-light h5">Sociální média/h6>
        <span href="#" class="text-light">Facebook/a></span>
        <span href="#" class="text-light">Instagram/a></span>
      </div>
    </div>
  </div>
</div>
<div class="container p-4 text-nd-start mx-auto mt-5">
  <div class="row mt-2">
    <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-4">
      <a class="text-light font-weight-bold" href="https://ndbootstrap.com">82022 Hospůdka na Horáňku/a>
    </div>
    <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-4">
      <a class="text-light" href="https://github.com">Obchodní podmínky/a>
    </div>
    <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mb-4">
      <p class="text-light">This page is designed by <span class="text-primary">Jakub Burdák/a>
    </div>
  </div>
</div>

```

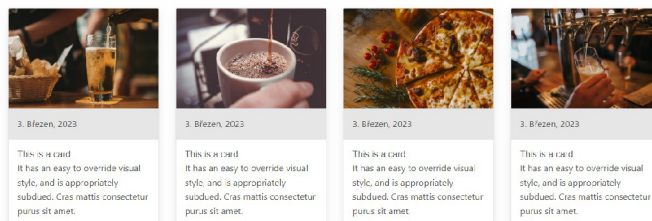
Obrázek 14: Ukázka kódu vytvořené komponenty pomocí Bootstrapu

Závěrem se mi podařilo velice snadno vytvořit komponentu pomocí Bootstrapu dle svého designu bez použití vlastního CSS jako v mé hlavní části bakalářské práce. Tyto dvě stejné komponenty mi posloužily k získání zkušeností s dalším CSS frameworkem a k následnému porovnání těchto dvou frameworků v kapitole porovnávání.

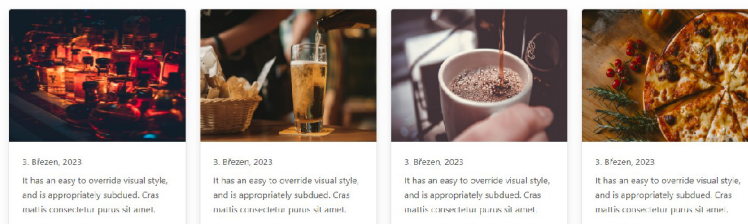
7.2 Vývoj komponenty pomocí Foundationu

K porovnání Bulmy a Foundationu jsem zvolil vytvoření komponenty sekce blogu. Tyto dvě komponenty jsem vytvořil jak pomocí Bulmy, tak pomocí Foundationu, abych zjistil rozdíly mezi těmito konkrétními frameworky a případně také jejich výhody a nevýhody.

FOUNDATION VERSION



BULMA VERSION



Obrázek 15: Ukázka vytvořené komponenty Bulmou a Foundationem

7.2.1 Vývoj sekce pomocí Foundationu

Vývoj za pomocí frameworku Foundation byl příjemnou změnou. Framework jako robustní a všestranný nástroj. Framework má širokou škálu možností využití už předdefinovaných pravidel, komponent či modulů. S frameworkem se mi pracovalo velmi dobře a rychle, možná i díky velmi rozsáhlé a dobré dokumentaci.

V první řadě jsem si vytvořil hlavní sekci pomocí HTML tagu `section`. Do této sekce jsem vnořil vlastní div se třídou `blog-section` reprezentující obsah této sekce. Dále jsem začal vnořovat už Foundation kód pro vytvoření požadované mřížky. Tímto kódem byl hlavní kontejner pomocí tagu `div` se třídou `grid-container`. Tímto jsem si implementoval rozložení mého obsahu do zhruba dvou třetin stránky. V další řadě jsem dle dokumentace Foundationu vnořil div se třídami `grid-x` a `grid-margin-x`, které nastavují základní parametry sloupců jako například odsazení zleva a zprava. Do toho divu jsem začal vkládat samotné sloupce pro obsah v podobě dalšího divu se třídou `cell`.

```
1 <section>
```

```

2 <div class="blog-section">
3   <div class="grid-container">
4     <div class="grid-x grid-margin-x">
5       <div class="cell">...</div>
6       <div class="cell">...</div>
7       <div class="cell">...</div>
8       <div class="cell">...</div>
9     </div>
10  </div>
11 </div>
12 </section>

```

Příklad 36: Rozdělení sekce na mřížku 1x4 pomocí Foundationu

V dalším kroku jsem začal do jednotlivých sloupců vnořovat další kód Foundationu pro vytvoření karet blogu už s připravenými tagy pro obsah těchto sloupců. Tento kód je poskytován také v dokumentaci.

```

1 <div class="cell">
2   <div class="radius bordered shadow card">
3     
4     <div class="card-divider">3. Brezen , 2023</div>
5     <div class="card-section">
6       <h4>This is a card.</h4>
7       <p>It has an easy to override visual style , and
is appropriately subdued. Cras mattis consectetur purus
sit amet.</p>
8     </div>
9   </div>
10 </div>
11 <div class="cell">
12   <!-- Same content -->
13 </div>
14 <div class="cell">

```



```

15 <!-- Same content -->
16 </div>
17 <div class="cell">
18 <!-- Same content -->
19 </div>

```

Příklad 37: Vnoření dalšího kódu do sloupců pomocí Foundationu

Sekci blog jsem měl díky Foundationu téměř hotovou, díky síle tohoto frameworku byl vývoj opravdu rychlý a jednoduchý.

Závěrem bylo potřeba doladit responzivitu, tedy zalomování sloupců na jednotlivých zařízeních. K responzivnímu chování nám Foundation poskytuje další třídy na sloupcové divy. Třídy s názvy, `small`, `medium` a `large`, doplněné o počet sloupců, které má jeden sloupec z celkového kontejneru zabírat.

I tento framework využívá rozdělení kontejneru na 12 sloupečků, obdobně jako například Bootstrap, případně Bulma. Počtem sloupečků s názvem zařízení definujeme šířku celkového sloupce.

```

1 <div class="cell medium-3 large-3">
2 <!-- Same content -->
3 </div>
4 <div class="cell medium-3 large-3">
5 <!-- Same content -->
6 </div>
7 <div class="cell medium-3 large-3">
8 <!-- Same content -->
9 </div>
10 <div class="cell medium-3 large-3">
11 <!-- Same content -->
12 </div>

```

Příklad 38: Responzivita sloupců pomocí Foundationu

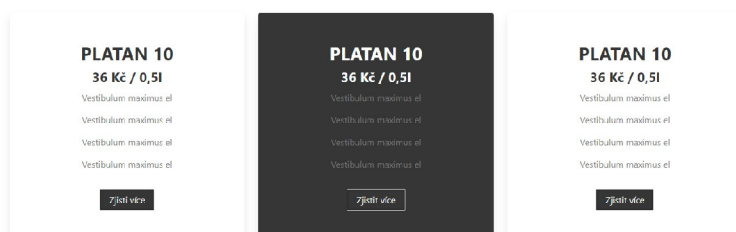
Blogová sekce pomocí Foundationu byla velmi rychle a kvalitně hotová. Nebylo nutné psát žádný další CSS kód pro dodesignování této sekce. Tato kompo-

nenta mi posloužila k získání zkušenosti a k budoucímu možnému porovnání s frameworkem Bulmou v další části bakalářské práce.

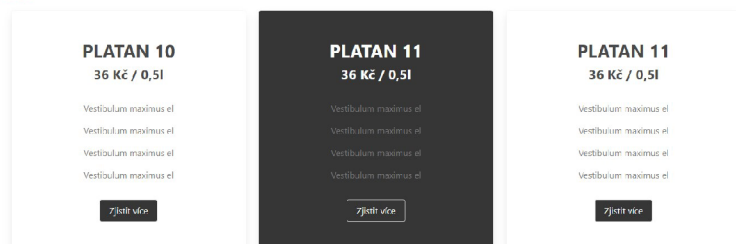
7.3 Vývoj komponenty pomocí Pure

K porovnání Bulmy a Pure jsem zvolil vytvoření jedné komponenty, a to konkrétní sekce z první části praktické části. Tyto dvě komponenty jsem vytvořil jak pomocí Bulmy, tak pomocí Pure, abych zjistil rozdíly mezi těmito konkrétními frameworky a případně také jejich výhody a nevýhody.

PURE VERSION



BULMA VERSION



Obrázek 16: Ukázka vytvořené komponenty Bulmou a Purem

Tvorba komponenty pomocí Pure byla o dost zdlohavější oproti vývoji s Bulmou. Pure jako velmi malý framework mi neumožňoval jednoduše použít mnoho věcí a musel jsem si tyto věci dopsat sám. Nicméně porovnání se budu blíže věnovat v kapitole porovnání frameworků.

7.3.1 Vývoj sekce pomocí Pure

Na začátek jsem vytvořil sekci pomocí HTML tagu `<section>`. Do této sekce jsem dále vnořil div se třídou `beers-section` pro jednoznačné označení sekce

a pro případné další budoucí používání obsahu této sekce. Dále jsem začal do tohoto divu vnořovat Pure kód pro vytvoření mřížky, a to div se třídou `pure-g` a do tohoto divu jsem vnořil ještě další tři divy s názvem třídy `pure-u-1-3`. Třída `pure-g` funguje obdobně jako u Bulmy třída `columns`, to znamená, že vytváříme kontejner nebo také řádek pro obsah. Divy se třídami `pure-u-1-3` jsou pak jednotlivé sloupce se šířkou 33 %.

```
1 <section>
2   <div class="beers-section">
3     <div class="pure-g">
4       <div class="pure-u-1-3">...</div>
5       <div class="pure-u-1-3">...</div>
6       <div class="pure-u-1-3">...</div>
7     </div>
8   </div>
9 </section>
```

Příklad 39: Rozdělení sekce na mřížku 1x3 pomocí Pure

Dále jsem jednotlivé sloupce začal vyplňovat dalšími HTML tagy pro obsah už s konkrétním obsahem.

```
1 <div class="pure-u-1-3">
2   <h2>Platan 10</h2>
3   <h3>36 Kč / 0,5l</h3>
4   <p>Vestibulum maximus el</p>
5   <p>Vestibulum maximus el</p>
6   <p> Vestibulum maximus el</p>
7   <p>Vestibulum maximus el</p>
8   <a href="#">Zjistí více</a>
9 </div>
10 <div class="pure-u-1-3">
11   <!-- Same content -->
12 </div>
```

```

13 <div class="pure-u-1-3">
14     <!-- Same content -->
15 </div>
16 <div class="pure-u-1-3">
17     <!-- Same content -->
18 </div>

```

Příklad 40: Vyplňování Pure sekce HTML obsahem

V dalším kroku jsem HTML tagy uvnitř sloupců začal designovat. Zde se objevil asi největší nedostatek frameworku Pure. Framework neposkytuje skoro žádné předdefinované třídy pro design HTML tagů. Většinu tříd jsem si zde musel navrhnout sám. Tyto třídy jsem pro jednoznačné rozlišení pojmenovával počátečním slovem „custom“. Na příkladu vidíme třídy s názvy `custom-h2` pro nastavení velikosti nadpisu úrovně dvě, dále `custom-p` pro nastavení rámece (paddingu) paragrafu, `custom-bt-bg` pro nastavení spodního rámece (paddingu) tlačítka atd.

```

1 <h2 class="custom-h2">Platan 10</h2>
2 <h3 class="custom-h3">36 Kč / 0,5l </h3>
3 <p class="custom-p">Vestibulum maximus el</p>
4 <p class="custom-p">Vestibulum maximus el</p>
5 <p class="custom-p">Vestibulum maximus el</p>
6 <p class="custom-p custom-mb">Vestibulum maximus el</p>
7 <a class="pure-button pure-button-active custom-bt-bg" href=
  "#">Zjistí více</a>

```

Příklad 41: Designování Pure sekce vlastními CSS třídami

Na následném příkladu vidíme CSS soubor s CSS příkazy designujícími jednotlivé tagy pro vytvoření mého konkrétního designu.

```

1 .custom-h2 {
2     font-weight: bold;
3     text-transform: uppercase;

```

```
4     font-size: 2rem;
5     color: #363636 !important;
6 }
7 .custom-h3 {
8     font-weight: bold;
9     font-size: 1.5rem;
10    color: #363636 !important;
11 }
12 .custom-h2a {
13     font-weight: bold;
14     text-transform: uppercase;
15     font-size: 2rem;
16     color: white !important;
17 }
18 .custom-h3a {
19     font-weight: bold;
20     font-size: 1.5rem;
21     color: white !important;
22 }
23 .custom-bt-bg {
24     background-color: #363636 !important;
25     color: white !important;
26     border: 1px solid #dbdbdb !important;
27 }
28 .custom-p {
29     padding-top: 0.5rem !important;
30     padding-bottom: 0.5rem !important;
31 }
32 .custom-mb {
33     margin-bottom: 1.5rem !important;
34 }
```

```

35 .custom-p-col {
36     padding: 0.75rem;
37 }

```

Příklad 42: Pure sekce doplněná o vlastní CSS kód

Nicméně bylo potřeba doladit dalších pár věcí, které mi framework sám o sobě nezajistil.

V první řadě jsem potřeboval zarovnat sekci na dvě třetiny stránky a doprostřed. U Bulmy mi tuto funkci zajišťoval div se třídou `container`. U Pure jediný kontejner, který můžeme použít je div se třídou `pure-g`, nicméně tento kontejner nemá žádná nastavená pravidla pro využití jen určité části stránky. Tuto funkci jsem si dopsal pomocí HTML třídy `custom-container`, kterou jsem navázal na div se třídou `pure-g`. Tento div jsem pak doplnil ještě o třídu `custom-center`, která mi centrovala obsah toho divu.

```

1 <section>
2     <div class="beers-section">
3         <div class="pure-g custom-container custom-center">
4             <div class="pure-u-1-3">...</div>
5             <div class="pure-u-1-3">...</div>
6             <div class="pure-u-1-3">...</div>
7         </div>
8     </div>
9 </section>

```

Příklad 43: Úprava Pure kontejneru doplněním vlastních CSS tříd

Tyto třídy jsem nadesignoval podle svých potřeb pomocí CSS příkazů. Bylo nutné nadefinovat v CSS i základní media queries pro nastavení šířky samotného kontejneru.

```

1 .custom-center {
2     text-align: center;
3 }

```

```
4 .custom-container {
5     margin-right: auto;
6     margin-left: auto;
7 }
8 @media (min-width: 1024px) {
9     .custom-container {
10         width: 960px;
11     }
12 }
13 @media (min-width: 1200px) {
14     .custom-container {
15         width: 1152px;
16     }
17 }
18 @media screen and (min-width: 1408px) {
19     .custom-container {
20         width: 1344px;
21     }
22 }
```

Příklad 44: Ukázka Pure kontejneru doplněného o vlastní CSS

V závěrečném kroku jsem ještě zaobalil obsah sloupců do jednoho divu se třídou `custom-card`, abych docílil designu karet. Tento design jsem si také musel pomocí vlastního CSS dokódovat, protože Pure neobsahuje ani žádné předdefinované komponenty jako typické karty apod. Poté už jsem jen dořešil responzivní zalamování, protože původní nastavení sloupců se stávalo na malých zařízeních nečitelný. To jsem udělal pomocí tříd `pure-u-1`, `pure-u-md-1-3` a `pure-u-lg-1-3`, které drží tři sloupce vedle sebe do velmi malého rozlišení obrazovky a poté sloupce zalomí pod sebe.

```
1 <section>
2     <div class="beers-section">
```

```

3     <div class="pure-g custom-container custom-center">
4         <div class="pure-u-1 pure-u-md-1-3 pure-u-lg-1-3
">
5             <div class="custom-card">
6                 <!-- Column content -->
7             </div>
8         </div>
9         <div class="pure-u-1 pure-u-md-1-3 pure-u-lg-1-3
">
10            <div class="custom-card">
11                <!-- Column content -->
12            </div>
13        </div>
14        <div class="pure-u-1 pure-u-md-1-3 pure-u-lg-1-3
">
15            <div class="custom-card">
16                <!-- Column content -->
17            </div>
18        </div>
19    </div>
20 </div>
21 </section>

```

Příklad 45: Přidání divu karty a řešení responzivity Pure sekce

```

1 .custom-card {
2     background-color: #fff;
3     border-radius: 0.25rem;
4     box-shadow: 0 0.5em 1em -0.125em rgba(10,10,10,.1), 0 0
0 1px rgba(10,10,10,.02);
5     color: #7a7a7a!important;
6     padding: 3rem;

```


7 }

Příklad 46: Úprava Pure sekce doplněním o vlastní CSS designu karty

Závěrem se mi podařilo vytvořit komponentu pomocí frameworku Pure dle svého návrhu, nicméně úsilí, které bylo potřeba vyvinout pro vytvoření této komponenty bylo o dost větší než například frameworkem Bulmou. Bylo nutné použít spousty vlastního CSS k dosažení mého požadavku. Tyto zkušenosti následně promítnu do porovnání těchto dvou frameworků v kapitole porovnávání frameworků.

8 Porovnání frameworků

Dnešní CSS frameworky využívají většinou nejmodernější vlastnosti přidané do CSS3 verze. Těmito vlastnostmi jsou především flexbox a grid. Při výběru konkrétního frameworku pro vývoj bychom se měli zaměřit především na sílu jejich základních vlastností jako například responzivita, flexibilita, dokumentace, komunita či výkon. Tyto faktory by měly být brány v úvahu při výběru CSS frameworku. Nicméně výběr závisí také na specifických potřebách a osobních preferencích vývojáře.

8.1 Porovnání Bulmy s Bootstrapem

Bulma a Bootstrap frameworky dnes již postavené na stejných vlastnostech CSS3 technologie, i když dříve tomu tak nebylo. Základem frameworku je flexbox a grid systém. Proto mají frameworky řadu podobností, navzdory tomu však mají také stále své specifické vlastnosti či použití.

8.1.1 Podobnosti Bulmy a Bootstrapu

Hlavní podobnost frameworků je, že oba jsou určeny pro vytváření responzivních webových stránek, případně webových aplikací pomocí předdefinovaných CSS tříd. Další podobnosti lze najít v dokumentaci v podobě různých modulů,

komponentů a mnoho dalších užitečných prvků, jako jsou tlačítka, tabulky, formuláře atd. Tyto prvky jsou zarovnávány stejně pomocí flexboxu a gridu, jak už bylo řečeno. Nicméně návrh použití a syntaxe frameworků se samozřejmě liší.

8.1.2 Rozdíly Bulmy a Bootstrapu

Výhodou a hlavním rozdílem frameworku Bulma je jeho modulárnost designu, který nám poskytuje větší volnost při tvorbě rozvržení prvků a samotných stylů. Bootstrapové styly, moduly a komponenty jsou více předdefinované, to nám však umožňuje rychlejší vývoj produktu oproti Bulmě. Dalším rozdílem Bulmy je její tzv. odlehčení. Bulma obsahuje méně kódu. Lehčí verze frameworku nám tím zajišťuje rychlejší načítání webu, případně webové aplikace. Oproti tomu síla Bootstrapu je v počtu možné integrace jednotlivých stylů, modulů či komponent, kterých Bootstrap má opravdu mnoho. Nelze opomenout ani rozdíly mezi samotným návrhem funkčnosti a syntaxí frameworků. Bulma se prezentuje jako intuitivní a jednoduchý framework pro používání. Tyto vlastnosti jsou spíše individuální a na posouzení konkrétního vývojáře. Design samotné Bulmy je modernější, nicméně Bulma není tolik rozšířená mezi vývojáři. Bootstrap má dlouhou historii na poli CSS frameworků, a proto je Bootstrap současně používanějším nástrojem pro vývoj. Tím sebou Bootstrap nese i dobrou dokumentaci a širší komunitu uživatelů. Jako poslední rozdíl bych rád uvedl i možnost integrace JavaScriptových komponent předdefinovaných samotným Bootstrapem, jako jsou modální okna a karusely. Bulma nám sice umožňuje použití vlastního JavaScriptu a přizpůsobení, ale už tyto komponenty sama o sobě prozatím neobsahuje.

Oba tyto frameworky jsou velmi užitečné pro vytváření moderních a responzivních webových stránek. Bulma je vhodnější pro projekty, které vyžadují vyšší úroveň flexibility a modularity. Bootstrap je vhodnější pro projekty, které vyžadují rychlý a efektivní vývoj s minimálním množstvím vlastního kódu a pro vývojáře, kteří preferují konzistentní vzhled.

8.2 Porovnání Bulmy s Foundationem

Bulma a Foundation je na tom podobně jako Bulma a Bootstrap. Foundation dnes již využívá také nejnovějších vlastností CSS3 technologie pro responzivní uspořádání prvků na stránce v rámci kontejneru. Základem Foundationu je flexibilní box a grid stejně jako u Bulmy nebo Bootstrapu. I zde jsme schopni díky tomu pozorovat řadu podobností. Nicméně Foundation se stále od Bulmy liší řadou věcí a svou použitelností.

8.2.1 Podobnosti Bulmy a Foundation

I tyto dva frameworky umožňují vývojáři vytvářet rychle responzivní webový design pomocí předdefinovaných CSS tříd, což nám zvyšuje efektivitu webového, případně aplikačního vývoje. Framework Foundation se toho snaží docílit stejně jako Bulma s použitím nové flexibilní a mřížkové vlastnosti CSS3 technologie. Ani v dokumentaci nenalezneme příliš rozdílů. Struktura dokumentace je podobná jako u dalších CSS frameworků. Nicméně Foundation má své specifické použití a zásadní rozdíly, oproti Bulmě, kterým se budu věnovat v další podkapitole porovnání frameworků.

8.2.2 Rozdíly Bulmy a Foundation

Mezi Bulmou a Bootstrapem byl jeden z hlavních rozdílů modulárnost. Bulma nám oproti Bootstrapu poskytuje větší možnost přizpůsobení našim konkrétním potřebám místo využití předdefinovaných komponent nebo modulů. Hlavním rozdílem mezi Bulmou a Foundationem je, že zde je to s modulárností přesně naopak. Foundation je mnohem robustnější a více všestranný nástroj, zatímco Bulma je více minimalistická a spíše moderní. Robustnost a všestrannost Foundationu ovlivňuje i jeho další faktory, jako je například rozsáhlost. Foundation je mnohem rozsáhlejší, a tím nám umožňuje přizpůsobit každý prvek webové stránky nebo webové aplikace. Bulma nám poskytuje spíše rychlou a snadnou flexibilní úpravu těchto prvků. Foundation nám díky své rozsáhlosti

nabízí i více možností pro přizpůsobení responzivního chování na různých zařízeních. Rozsáhlost Foundationu ovlivňuje i samotný výkon frameworku. Bulma má vynikající výkon, což nám umožňuje velmi rychlé načítání webových stránek, oproti tomu Foundation nabízí dobrý výkon, ale v některých případech může být pomalejší než Bulma. Bulma i Foundation poskytují velmi dobrou dokumentaci kódu a dostatek příkladů. Nicméně Foundation má podrobnější popis funkčnosti a kromě toho i rozsáhlejší komunitu vývojářů, kteří přispívají k vylepšení frameworků a je více rozšířený, nejspíše i díky své delší působnosti na poli CSS frameworků. Nelze opomenout ani rozdíly mezi samotnou syntaxí frameworků. Posledním rozdílem bych rád uvedl, že Foundation stejně jako Bootstrap umožňuje integraci JavaScriptových komponent předdefinovaných samotným Foundationem. Typicky karusel, slidery apod. Bulma prozatím integraci těchto JavaScriptově předdefinovaných komponent neumožňuje.

Oba frameworky mají své výhody i nevýhody a výběr závisí na specifických potřebách projektu a osobních preferencích vývojáře. Nicméně, Bulma je vhodnější pro webové projekty s minimalistickým a moderním designem. Foundation je vhodnější pro projekty, které vyžadují robustní a všestranný design.

8.3 Porovnání Bulmy s Pure.css

Pure funguje stejně jako předešlé frameworky na základě předdefinovaných CSS tříd, které vývojář využívá, případně upravuje pro své konkrétní potřeby vývoje. Stejně jako předešlé frameworky i Pure je open source projektem, a proto ho mohou vývojáři volně používat ve svých projektech. I přesto, že má knihovna Pure řadu dalších podobných vlastností s knihovnou Bulmy, existují mezi nimi i rozdíly v použití a stylu, takže je i zde důležité zvážit, který framework bude nejvhodnější pro konkrétní projekt.

8.3.1 Podobnosti Bulmy a Pure.css

I tento poslední framework z už představených frameworků přišel také s příchodem CSS3 verze na nové vlastnosti této technologie, a to flexibilní box a grid systém. Jak Bulma, tak i Pure nám také umožňují urychlit vývoj responzivních webových stránek nebo webových aplikací na zařízeních s různým rozlišením, díky předdefinovaným CSS třídám, jak už bylo zmíněno. Dokumentace je strukturovaná obdobným způsobem pro CSS dokumentace jako u jiných dalších CSS frameworků. Posledním bodem, který bych rád uvedl u těchto dvou frameworků je, že Pure se zaměřuje na čisté CSS. To znamená, že u frameworku nenalezneme žádnou možnou integraci JavaScriptových komponentů v podobě karuselu, případně nějakých sliderů apod. Prozatím je tomu stejně i u Bulmy.

8.3.2 Rozdíly Bulmy a Pure.css

Hlavním rozdílem mezi Bulmou a Pure je jejich rozsáhlost. U Bulmy a Foundation jsem avizoval na velkou rozsáhlost Foundationu oproti minimalistické Bulmě. Pure je oproti Bulmě ještě více minimalistický. Zaměřuje se také na jednoduchost, čistotu a samotnou rychlost. Bulma nám nabízí větší výběr komponent a stylů, což může být pro některé projekty vhodnější. Faktor velikosti komponentů a stylů samozřejmě ovlivňují i velikost samotných frameworků. Pure má velmi malou velikost souboru oproti Bulmě, a tím je i jeho samotný výkon vyšší. V Bulmě nalezneme také řadu variant pro různé komponenty a prvky, jako jsou například navigace, karty, tlačítka, formuláře atd. Pure nám nabízí jen základní sadu komponent a prvků. Z pohledu flexibility jsou oba dva frameworky silnými nástroji pro usnadnění responzivního designu. Nicméně Bulma nám nabízí více možností úprav a rozložení samotných prvků nebo komponent na stránce. Nelze opomenout ani rozdíly mezi syntaxí frameworků. Jedním z dalších zásadních rozdílů je také závislost a návrh frameworků. Knihovna Pure je čistým CSS souborem. Není tedy potřeba pro úpravu samotného fra-

meworku nějaké další znalosti. V případě Bulmy lze upravovat CSS soubor, nicméně samotná Bulma je navržena v tzv. Sassu, což by se dalo nazvat rozšířením CSS. Sass je předzpracovávající CSS, což znamená, že je to nástroj, který umožňuje psát CSS s použitím rozšířené syntaxe a následně tento kód přeložit na běžné CSS. Dokumentace Bulmy je díky možnostem a její rozsáhlosti ještě rozsáhlejší než samotná dokumentace Pure. Pure má poměrně jednoduchou dokumentaci. Bulma má i širší komunitu uživatelů než Pure.

Oba tyto frameworky mají své výhody a nevýhody a i zde výběr závisí na specifických potřebách a preferencích vývojáře nebo samotného projektu. Pure.css je vhodnější pro menší projekty nebo projekty, které nevyžadují mnoho různorodých komponent. Pro větší a flexibilnější projekt bude vhodnějším řešením framework Bulma.

9 Závěr

Hlavním cílem této bakalářské práce bylo seznámit uživatele s poměrně novým CSS frameworkem na poli CSS3 frameworků jako je například Bootstrap, Foundation apod. Zjistit a popsat základní funkci a tvorbu webové stránky pomocí CSS responzivního frameworku Bulma. Dále zjistit do jaké míry je tento framework použitelný při reálném vývoji responzivních webových stránek a zjistit, případně fyzicky demonstrovat reálné výhody a nevýhody frameworku. Následně framework Bulma porovnat s dalšími alternativními frameworky a zjistit jejich rozdíly, případné výhody a nevýhody.

Responzivní design webové stránky pomocí CSS3 flexibilního boxu a gridu, na kterém staví samotná Bulma od svého začátku, jsou dnes velmi silnými vlastnostmi. Důkazem toho je, že i drtivá většina CSS frameworků je dnes již také postavená na těchto vlastnostech. Flexibilní box a grid nám umožňují velmi dobře a efektivně pracovat a zarovnávat prvky na webové stránce nebo webové aplikaci.

Samotný vývoj frameworkem byl velmi rychlý a efektivní. Bulma splňovala přesně to, co jsem od ní jako od CSS frameworku očekával. Kód je velmi srozumitelný a dobře strukturovaný. To znamená, že i má vlastní modulace kódu byla velmi snadná a hlavně rychlá. Nemusel jsem nijak dlouze procházet nebo zorientovávat se v kódu. Pomocí Bulmy jsem tedy docílil poměrně snadno a rychle responzivní a zároveň moderní webové stránky. Nicméně ani Bulma není nedotknutelná a setkal jsem se zde s určitými komplikacemi.

V další části jsem vytvářel komponenty z webové stránky dalšími alternativními frameworky. Konkrétně jsem vytvořil patičku pomocí frameworku Bootstrap, jednu sekci pomocí frameworku Foundation a jednu sekci pomocí frameworku Pure, k porovnání jednotlivých frameworků s Bulmou.

Typickým příkladem, se kterým jsem se setkával, je nižší nebo vyšší počet možností použitých předdefinovaných komponent, modulů nebo i implementací samotných JavaScriptových komponent, oproti samotné Bulmě. Některé

frameworky mi usnadňovaly práci více, některé zase méně. Konkrétnější porovnání frameworků a výhod, případně nevýhod rozebírám v kapitole porovnání frameworků.

Nicméně vývoj pomocí Bulmy byl opravdu jednoduchý a intuitivní. S Bulmou jsem během vývoje neměl žádný konkrétní problém, a to i díky skvělé dokumentaci. Bulma mi zajistila při tvorbě praktické části přesně to, co bych od takového frameworku očekával, a to dostatek názorných HTML kódů a předdefinovaných CSS tříd pro vytvoření moderního a poměrně modulárního webu v poměrně krátkém čase. Ano, jsou i silnější frameworky na poli CSS frameworků, ale Bulma se dle mého stala jednoduchým nástrojem pro rychlé vytvoření funkčního a moderního webu.

Seznam použité literatury a zdrojů

- [1] Bulma.io: Bulma: the modern CSS framework that just works. *[online]*. [cit. 2022-03-30]. Dostupné z: <https://bulma.io/>
- [2] Bulma.io: Documentation: Everything you need to create a website with Bulma. *[online]*. [cit. 2022-03-30]. Dostupné z: <https://bulma.io/documentation/>
- [3] Mobiosolutions: What is Bulma ? Advantages & disadvantages of using Bulma *[online]*. [cit. 2022-03-30]. Dostupné z: <https://mobiosolutions.com/what-is-bulma-advantages-disadvantages-of-using-bulma/>
- [4] Devmountain: Bulma CSS: What It Is and Why It's a Framework That Developers Love. *[online]*. [cit. 2022-03-30]. Dostupné z: <https://blog.devmountain.com/why-bulma-css-could-be-your-new-favorite-framework/>
- [5] HubSpot: The Bulma CSS Framework: What It Is and How To Get Started. *[online]*. [cit. 2022-03-30] Dostupné z: <https://blog.hubspot.com/website/bulma-css>
- [6] Tutorialspoint: Bulma *[online]*. [cit. 2022-03-30]. Dostupné z: <https://www.tutorialspoint.com/bulma/>
- [7] HTML5 specification finalized, squabbling over specs continues *[online]*. [cit. 2023-03-20]. Dostupné z: <https://arstechnica.com/information-technology/2014/10/html5-specification-finalized-squabbling-over-who-writes-the-specs-continues/>
- [8] HTML5 *[online]*. [cit. 2023-03-20]. Dostupné z: https://cs.wikipedia.org/wiki/HTML5#cite_note-finalars-1/
- [9] CSS: Cascading Style Sheets *[online]*. [cit. 2023-03-20]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>

- [10] What is JavaScript? *[online]*. [cit. 2023-03-20]. Dostupné z:
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [11] What is JavaScript? *[online]*. [cit. 2023-03-20]. Dostupné z:
https://www.w3schools.com/js/js_history.asp
- [12] A star was born *[online]*. [cit. 2023-03-20]. Dostupné z:
<https://byby.dev/css-evolution>
- [13] A star was born *[online]*. [cit. 2023-03-20]. Dostupné z:
<https://byby.dev/css-evolution>
- [14] What is Visual Studio Code? Microsoft's extensible code editor *[online]*.
[cit. 2023-03-26]. Dostupné z:
<https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>
- [15] What is Visual Studio Code? Microsoft's extensible code editor *[online]*.
[cit. 2023-03-26]. Dostupné z:
<https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>
- [16] What is Visual Studio Code? Microsoft's extensible code editor *[online]*.
[cit. 2023-03-26]. Dostupné z:
<https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>

Seznam obrázků

1	Framework Bulma v souboru bulma.css jako plná verze	22
2	Framework Bulma v souboru bulma.min.css jako minimální verze	23
3	Autor: bulma.io, Nejjednodušší systém mřížky	24
4	Vývojové prostředí Visual Studio Code	27
5	Ukázka kořenového adresáře webové stránky	30
6	Ukázka HTML webových souborů	31
7	Ukázka úvodní strany webové stránky vytvořené Bulmou	33
8	Ukázka nalinkování Bulmy na řádku č. 12	34
9	Ukázka designu navigačního menu Bulmy	35
10	Ukázka designu hlavičky pomocí Bulmy	40
11	Sekce webové stránky novinky vytvořena pomocí Bulmy	43
12	Ukázka designu patičky pomocí Bulmy	52
13	Ukázka vytvořené komponenty Bulmou a Bootstrapem	58
14	Ukázka kódu vytvořené komponenty pomocí Bootstrapu	62
15	Ukázka vytvořené komponenty Bulmou a Foundationem	63
16	Ukázka vytvořené komponenty Bulmou a Purem	66

Seznam příkladů

1	Ukázka HTML5 hlavičky se základními meta daty	15
2	Ukázka designování značek neboli tagů pomocí CSS stylů	17
3	Ukázka funkce vyhodnocení sudosti čísla pomocí JavaScriptu	19
4	Ukázka HTML divů s atributem třídy Bulmy	23
5	Ukázka nalinkování Bulmy do projektu	32
6	Ukázka implementace CSS Bulmy do HTML souborů	32
7	HTML implementace navigace pomocí Bulmy	35
8	Úprava menu pomocí JavaScriptové integrace Bulmy	37
9	Vnoření navigace Bulmy do tagu <header>	38
10	Designování prvků navigace hlavičky pomocí Bulmy	38
11	Vnoření druhé části hlavičky do tagu <header>	40
12	Vnoření HTML obsahu druhé části hlavičky	41
13	Design druhé části hlavičky pomocí Bulmy	41
14	Úprava hlavičky pomocí vlastního CSS kódu	42
15	Rozdělení sekce novinky pomocí Bulmy	44
16	První řádek sekce novinky pomocí Bulmy	45
17	Druhý řádek sekce novinky pomocí Bulmy	46
18	Úpravy druhého řádku sekce novinky pomocí Bulmy	47
19	CSS Úprava druhého řádku sekce novinky	47
20	Úprava rozdělení posledních řádků sekce novinky pomocí Bulmy	47
21	Design třetího a čtvrtého řádku sekce novinky pomocí Bulmy	49
22	Vlastní HTML kód pro tzv. overlay posledních řádků sekce novinky	50
23	Vlastní CSS kód pro tzv. overlay posledních řádků sekce novinky	50
24	Úprava sekce novinky třídou news-image třetího a čtvrtého řádku	50
25	CSS přidání obrázku do třetího a čtvrtého řádku sekce novinky	51
26	Rozdělení patičky pomocí Bulmy na mřížku 1x4	52
27	Vyplňování obsahu sloupcové mřížky Bulmy	53

28	Design patičky webové stránky pomocí Bulmy	54
29	Rozdělení patičky pomocí Bulmy na mřížku 2x4	54
30	Rozdělení posledního sloupce na další mřížku pomocí Bulmy . .	55
31	Lazení responzivity Bulmy druhé části patičky	56
32	Rozdělení patičky na mřížku 1x4 pomocí Bootstrapu	58
33	Vložení obsahu do jednotlivých sloupců Bootstrapové komponenty	59
34	Designování obsahu jednoho sloupce patičky pomocí Bootstrapu	60
35	Přizpůsobení designu na více zařízeních pomocí Bootstrapu . . .	61
36	Rozdělení sekce na mřížku 1x4 pomocí Foundationu	63
37	Vnoření dalšího kódu do sloupců pomocí Foundationu	64
38	Responzivita sloupců pomocí Foundationu	65
39	Rozdělení sekce na mřížku 1x3 pomocí Pure	67
40	Vyplňování Pure sekce HTML obsahem	67
41	Designování Pure sekce vlastními CSS třídami	68
42	Pure sekce doplněná o vlastní CSS kód	68
43	Úprava Pure kontejneru doplněním vlastních CSS tříd	70
44	Ukázka Pure kontejneru doplněného o vlastní CSS	70
45	Přidání divu karty a řešení responzivity Pure sekce	71
46	Úprava Pure sekce doplněním o vlastní CSS designu karty . . .	72

A Příloha

Webová stránka praktické části: <https://hospudkanahoranku.cz>

B Příloha

Na přiloženém CD/DVD se nachází plné znění mé bakalářské práce, zdrojové kódy praktické části a odkaz na webovou stránku.