



Středoškolská technika 2023

Setkání a prezentace prací středoškolských studentů na ČVUT

Využití HTML5, CSS3 a Bootstrapu na webu

Matěj Rešl

Vyšší odborná škola, Střední škola, Centrum odborné přípravy
Budějovická 421, Sezimovo Ústí

Anotace

Závěrečná práce se zabývá využitím HTML5, CSS3, Javascriptu, frameworku Bootstrap a základy SEO optimalizace webových stránek. Na konkrétních příkladech webových stránek demonstruje kdy, kde a jak používat sémantické značky HTML5, aplikace frameworku Bootstrap do HTML5, použití Javascriptu, ladění webových stránek a jejich umístění na webový server.

Klíčová slova: HTML5, CSS3, Bootstrap, Javascript, SEM, SEO optimalizace, PPC, PPV, CPC, Adobe XD, webdesign

Abstract

The final thesis deals with the use of HTML5, CSS3, Javascript, Bootstrap framework and the basics of SEO optimization of websites. Using specific website examples, it demonstrates when, where and how to use HTML5 semantic tags, application of the Bootstrap framework to HTML5, use of Javascript, debugging web pages and their placement on a web server.

Keywords: HTML5, CSS3, Bootstrap, Javascript, SEM, SEO optimization, PPC, PPV, CPC, Adobe XD, webdesign

Seznam zkratek

- HTML: HyperText Markup Language
- CSS: Cascaded Style Sheets
- SEM: Search Engine Marketing
- SEO: Search Engine Optimization
- PPC: Pay Per Click
- PPV: Pay Per View
- CPC: Cost Per Click
- DOM: Document Object Model
- AJAX: Asynchronous Javascript and XML
- W3C: World Wide Web Consortium

1	Úvod.....	6
1.1	HTML5.....	6
1.2	CSS3	7
1.3	Javascript	7
1.4	Bootstrap	7
1.5	SEM – Search Engine Marketing.....	8
1.5.1	SEO optimalizace	8
1.5.2	Platba za proklik	9
1.5.3	Platba za zobrazení	9
1.5.4	Platba za klik.....	9
1.5.5	Přednostní katalogové zápisy.....	9
1.6	Responzivní webdesign	9
1.6.1	Media queries.....	10
1.6.2	Flexibilní obrázky	10
1.6.3	Flexibilní struktura.....	10
2	Příprava grafického layoutu webu	11
2.1	Prototypovací nástroje pro tvorbu webové grafiky	11
2.2	Práce s Adobe XD.....	11
2.2.1	Karta Design	13
2.2.2	Karta Prototype	13
3	Kódování webu	15
3.1	Tvorba menu.....	16
3.2	Tvorba hlavičky webu	17
3.3	Textové pasáže webu	20
3.4	Fotogalerie na webu	23
3.4.1	Lightboxy.....	25
3.4.2	Medailonky	26
3.5	Formuláře na webu.....	26
3.6	Mapa na webu	30
3.6.1	Statická mapa.....	31
3.6.2	Interaktivní mapa	31
3.7	Patička webu.....	33
3.7.1	Superpatička.....	36
3.8	Převod od responzivní podoby	36
4	Ladění webu.....	38
4.1.	Průzkumník	39

4.2.	Konzole prohlížeče.....	39
4.3.	Responzivní design	40
5	Závěr	42
6	Zdroje.....	43
	Bibliografie.....	43
	Webové stránky.....	43
7	Seznam obrázků	44
8	Seznam příloh	44
	Nástroje použité pro zpracování	45

1 Úvod

Toto téma jsem si zvolil, protože se velice zajímám o tvorbu webových stránek od grafického návrhu až po naprogramování, a také z důvodu, že mnoho vývojářů webových stránek nevyužívá možnosti HTML5. Mnoho lidí si totiž neuvědomuje, že dobře vytvořený web vydělává firmám peníze. Kvalitní optimalizace pro vyhledávače zajistí zobrazení na vyšších příčkách. Přestože je to dnes už skoro povinnost, stále některé webové stránky nejsou vůbec responzivní, nevyužívají SEO optimalizaci, nejsou ověřené certifikační autoritou, nemají HTTPS zabezpečení. Kód bývá velmi nepřehledný a pro ostatní vývojáře špatně čitelný.

Tato práce má za cíl poukázat na hlavní problémy, kterých se začátečníci ve tvorbě webových stránek nejvíce dopouštějí, a je určena nejen profesionálním vývojářům webových stránek, ale i běžnému uživateli, který by rád pronikl do tvorby webových stránek v HTML5, CSS3, Bootstrapu, Javascriptu.

1.1 HTML5

V předchozích verzích značkovacího jazyka HTML se používaly tagy, kterým ale webový vyhledávač nedával žádnou váhu. Značkovací jazyk HTML5 přinesl oproti předchozím verzím několik změn, které usnadňují vyhledávači vyhledávat stránky.

Mezi tyto změny patří především responzivita, díky které se webová stránka přizpůsobí tomu zařízení, které je využíváno k prohlížení webové stránky. Dříve se dělaly webové stránky tak, že se musely tvořit pro každé zařízení zvlášť, což bylo časově náročné a velmi neefektivní z důvodu, že ne všechny prohlížeče (např. Internet Explorer) dokázaly zobrazovat weby podle nejnovějších standardů. HTML5 samozřejmě přináší mnoho funkcí a nových tagů tzv. **sémantické tagy**. Tyto tagy je potřeba používat s rozmyslem, protože ne každý web je využije. Ale co určitě použijí všichni vývojáři webových stránek, jsou tagy pro definování layoutu webu:

Název tagu	Vlastnost tagu
nav	navigace (menu)
header	hlavička webu
main	hlavní obsah webu
section	sekce webu např. galerie
article	článek, textové pasáže
figure	obrázky na webu

footer	patička webu
--------	--------------

Další užitečná a velmi žádoucí věc, kterou HTML5 přinesl, je zákaz použití HTML stylů. HTML styly byly nahrazeny CSS3 stylováním.

1.2 CSS3

Druhým jazykem, kterým se budu v této práci zabývat, je CSS3, který spravuje (stejně jako HTML5) **W3C** konsorcium. Tento jazyk přinesl oproti předchozím verzím nové možnosti stylování a odstranil některé neduhy, např. obrázky se zaoblenými rohy, které se dříve tvořily v programech na úpravu obrázků. Dnes lze toto tvořit pomocí atributu *border-radius*. Dále byla přidána funkce *calc()*, kdy místo konkrétní hodnoty pouze vložíme matematickou operaci.

CSS3 obsahuje mnoho dalších funkcí, kterými se podrobně nebudu zabývat, protože se využívají na specifické účely a nepoužívají se tak často.

1.3 Javascript

Javascript je dynamický skriptovací jazyk, který pracuje primárně v klientské části webu, ačkoliv existují frameworky pro serverovou část webu. Je velmi jednoduchý a zároveň velmi žádaný. Pro práci s ním slouží velmi mnoho frameworků, které usnadňují práci. V tomto jazyce není potřeba deklarovat datový typ (stejně jako u PHP), a proto se mu říká dynamicky typovaný jazyk. Mezi nepoužívanější JS frameworky patří:

- jQuery
- DOM
- Node.js
- AJAX
- Angular
- React

My budeme používat frameworky jQuery a DOM.

Framework jQuery se používá primárně na animace a interaktivní prvky, např. načítací kolečka, pokročile „*paralax scrolling*“ efekty, a jiné.

DOM je zkratka pro **Document Object Model** a slouží k propojení HTML a JS. Díky tomu lze měnit strukturu HTML pomocí Javascriptu.

1.4 Bootstrap

Bootstrap je framework, který nám pomáhá s responzivitou webu a eliminuje ruční tvorbu responzivního zobrazení.

Jedná se o dvanáctisloupcový systém, kde **12 sloupců** představuje rozvržení pro desktop, **8 sloupců** představuje rozvržení pro tablety a **4 sloupce** představují rozvržení pro mobilní telefony. Tento systém je velmi užitečný, když tvoříme web, který se bude přizpůsobovat tomu zařízení, na kterém si ho zobrazíme. Bootstrap byl navržen primárně pro tzv. **mobile-first**, což je metoda tvorby webu, kdy začínáme prvně mobilním zařízením a až poté desktopem. Tuto metodu lze používat z toho důvodu, že 90 % uživatelů si prohlíží web na telefonu, jelikož chytré telefony mají možnost internetového připojení.

Bootstrap je samozřejmě open-source framework a lze si ho přizpůsobit podle svých potřeb pomocí jazyka HTML, CSS, Javascriptu. Bootstrap se vkládá do HTML pomocí `<link rel="stylesheet" type="text/css" href="absolutni_cesta_k_bootstrap_souboru">` a poté se volá pomocí tříd jako na obrázku 6, kde `<header class="col-12"></header>` znamená, že tento element bude mít šířku o dvanácti sloupcích (col-12 znamená column 12).

1.5 SEM – Search Engine Marketing

SEM patří mezi formu internetového marketingu. Využívá potenciálu webových vyhledávačů. Pomocí SEM můžeme třídit a hledat webové stránky. Do SEMu patří SEO optimalizace a PPC reklamy.

1.5.1 SEO optimalizace

SEO optimalizace či optimalizace pro vyhledávače je technika SEMu. SEO optimalizace nám pomocí klíčových slov, angl. **Keywords**, zajistí, že se web bude umisťovat na vyšší příčky. Není to ale jen o klíčivých slovech. Důležitá je optimalizace HTML dokumentu, tj. využití sémantických tagů. Dříve se používal tag `<meta name="keywords" content="klicova slova">`, do kterého se klíčová slova zapisovala. Google Chrome tento tag považuje za nedůležitý, ale neznamená to, že se nesmí používat. Klíčová slova by měla být volena tak, aby nejvíce definovala webovou stránku, tj. na základě jakých slov by uživatel web hledal. Tato oblast vyžaduje odborníka z oblasti online marketingu. Klíčová slova se nejčastěji vyskytují v textových částech webové stránky. Před aplikací klíčových slov na webu by měl SEO specialista vytvořit seznam relevantních témat a podle nich klíčová slova vymýšlet. Náš web se zabývá ekologií, takže seznam témat by se měl zabývat otázkami ekologie, důležitostí ekologie, výhodami ekologie, aplikací ekologie na životní prostředí atp. Klíčová slova na našem webu tedy mohou být například tato:

- Ekologie
- Životní prostředí
- Příroda

1.5.2 Platba za proklik

Jedná se o typ reklamy, které se také říká PPC, což je zkratka **Pay Per Click**. Za PPC reklamu platí majitel v případě, že na ni někdo klikne. Tyto reklamy jsou nejčastěji ve formě bannerů a odkazují se na e-shopy. Pokud tedy na reklamu kdokoliv klikne, platí majitel příslušného e-shopu.

1.5.3 Platba za zobrazení

Dále existuje PPV (Pay Per View). Zde platíme za počet zobrazení. Cena této reklamy se udává v tisících. V této reklamě používáme zkratky:

- CPM – Cost Per Mile
- CPT – Cost per Thousand

Obojí znamená cenu za tisíc zobrazení. Cena CPM/CPT se většinou udává takto: Pokud je banner dobře cílen, cena se pohybuje přibližně mezi 20 až 600 Kč. Pokud ale je banner umístěn špatně, cena se pohybuje okolo 1 až 10 Kč.

1.5.4 Platba za klik

Potom existuje CPC nebo také Cost Per Click. Za tento typ reklamy se platí v případě, že uživatel sleduje video a reklamu nepřeskočí. V současné době na youtube vyskakují nepřeskočitelné reklamy, takže na tom vlastně nikdo neprodělá. Dříve stačilo shlédnout 5 vteřin a reklama se dala přeskočit. Dnes obecně platí, že čím více reklam uživatel vidí, tím více plátce reklamy vydělá.

1.5.5 Přednostní katalogové zápisy

Toto je jedna z forem internetové reklamy, díky které se náš web dostává na vyšší příčky, protože je snáze dohledatelný. Zápisy v katalogích jsou placené a neplacené. V České republice se používají neplacené zápisy. Typický příklad jsou **firmy.cz**. Placený typ zápisů se jako reklama nedá nijak přeskočit a nepomůže ani organické vyhledávání. Je dobré velmi pečlivě zvážit, kdy, kde a jaký typ zápisu zvolíme. Katalogové zápisy najdeme například zde:

1. zboží.cz
2. hyperinzerce.cz
3. podnikatel.cz

1.6 Responzivní webdesign

Responzivní webdesign je jeden ze základních kamenů při tvorbě webových stránek, protože uživatelé k prohlížení webových stránek velmi často používají chytré telefony.

S termínem „*responzivní webdesign*“ přišel americký vývojář **Ethan Marcotte** v jeho stejnojmenném článku na blogu „*A LIST APART*“.

Responzivní webdesign je způsob stylování HTML5 pomocí CSS3, který zaručí, že se web zobrazí perfektně na jakémkoliv zařízení. Responzivní webdesign využívá tři základní prvky pro lepší zobrazení:

- Media queries
- Flexibilní obrázky
- Flexibilní strukturu

1.6.1 Media queries

Media queries je podmínka v CSS3, která definuje, při kterém rozlišení se změní obsah webu. Pokud si tedy zobrazíme webovou stránku na telefonu, bude mít podmínka šířku rozlišení mobilního telefonu a přizpůsobí obsah webu.

1.6.2 Flexibilní obrázky

Touto technikou docílíme, aby se naše obrázky přizpůsobovaly každému zařízení totožně s flexibilní strukturou. Hodnota maximální šířky by se měla uvádět vždy v % a výška by měla být „*auto*“.

1.6.3 Flexibilní struktura

Technika flexibilní struktury je závislá na procentuálních šířkách elementů. Všechn obsah našeho webu by tedy měl mít výšku a šířku v %. Na výpočet šířky se používá následující vzorec:

$$\text{procentuální šířka} = (\text{požadovaná šířka (v px)} / \text{předchozí oddíl (v px)}) * 100$$

Do vzorce si dosazujeme hodnoty podle potřeby. Pokud máme oddíl s šířkou 800px a chceme další oddíl v šířkou 600px, bude dosazení vypadat následovně:

$$75 = (600 / 800) * 100$$

Procentuální šířka se nesmí zaokrouhlit. Hodnoty musí být vždy co nejpřesnější.

2 Příprava grafického layoutu webu

Na začátek je potřeba říci, že programování a kódování není jedno a to samé, i když si to většina lidí myslí. Programování je zápis algoritmu pomocí programovacího jazyka a kódování je definování jakéhokoliv obsahu pomocí tagů.

2.1 Prototypovací nástroje pro tvorbu webové grafiky

Webová stránka je většinou kódována z předem připraveného layoutu, který nám dodá webdesignér. Webdesignér je osoba, která tvoří grafický layout webu v prototypovacím programu, např. Adobe XD, Figma, InVision, Sketch, popř. Adobe Photoshop.

Software	Adobe XD	Figma	InVision	Sketch	Photoshop
Operační systém	Windows	Windows	Windows	Windows	Windows
	Mac OS X	Mac OS X	Mac OS X	Mac OS X	Mac OS X
	Android	Android	-	-	-
	iOS	iOS	-	-	-
	-	Linux	-	-	-

Výše v tabulce vidíme, že nejvyšší pokrytí operačních systémů má software Figma. My si budeme příklady ukazovat v programu Adobe XD.

2.2 Práce s Adobe XD

Adobe XD je prototypovací nástroj od firmy Adobe. V plánu „*Started plan*“ je zdarma s omezením jednoho sdílení s ostatními. Program jinak stojí v samostatné edici 12,09 € měsíčně, což je přibližně 300 Kč. Nám pro demonstraci postačí plán zdarma.

Uživatelské rozhraní se skládá ze tří karet:

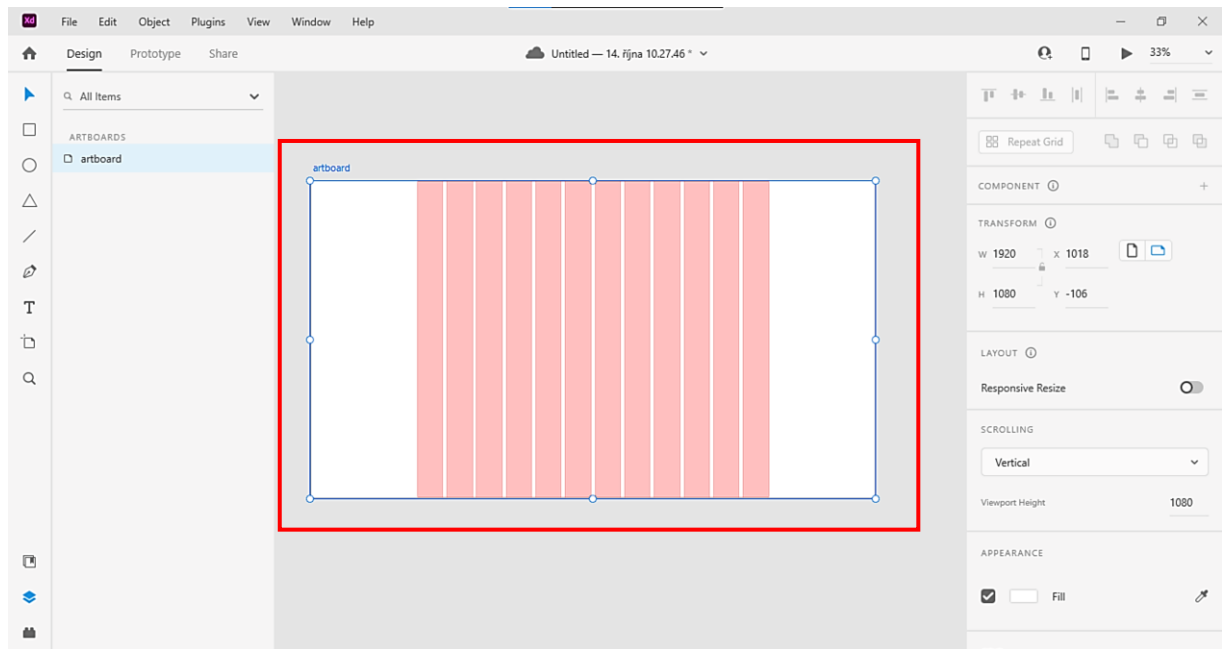
- **Design:** pro grafický návrh.
- **Prototype:** pro přidání interakcí, animací, přechodů.
- **Sharing:** pro sdílení s ostatními.

Nás budou primárně zajímat první dvě karty, kde si ukážeme tvorbu bootstrap gridu.

Adobe XD pracuje s tzv. artboardy, což jsou plátna, na která umísťujeme naše grafické elementy. Těmto objektům můžeme přiřadit tyto vlastnosti:

- Název artboardu
- Šířka × výška
- Responzivní šířka
- Skrolování

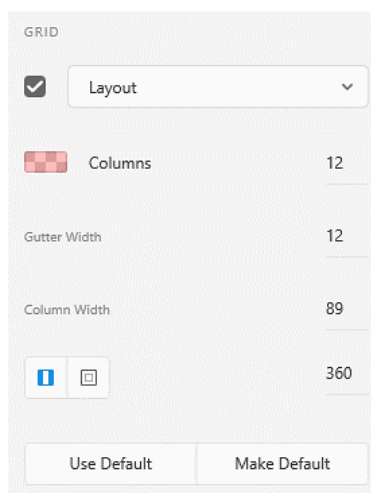
- Barva pozadí
- Grid



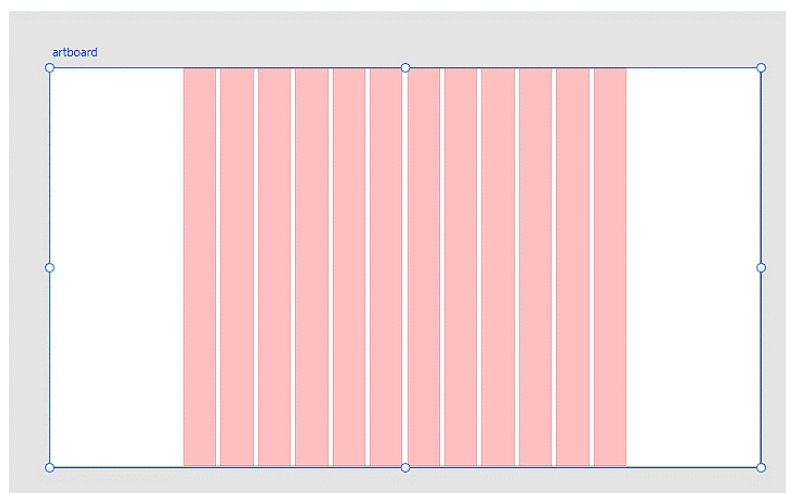
Obr. 1: Artboard v Adobe XD (bílá plocha v červeném obdelníku)

V artboardu můžeme měnit velikost, barvu, zaškrtnout responzivní zobrazení, ale nejdříve je potřeba mu přidat bootstrap grid. Ten přidáme pomocí zaškrťovacího políčka dole v pravé části „Grid“. Layout ve výběrovém poli necháme a zadáme hodnoty, které jsou na *obrázku 2*.

Na *obrázku 3* vidíme, jak bude výsledný grid vypadat. Délka všech 12 sloupců i s mezerami mezi sloupci bude cca 1200px. Do této šířky se nám správně zobrazí jakýkoliv obsah webu.



Obr. 2: Hodnoty zadané do políček

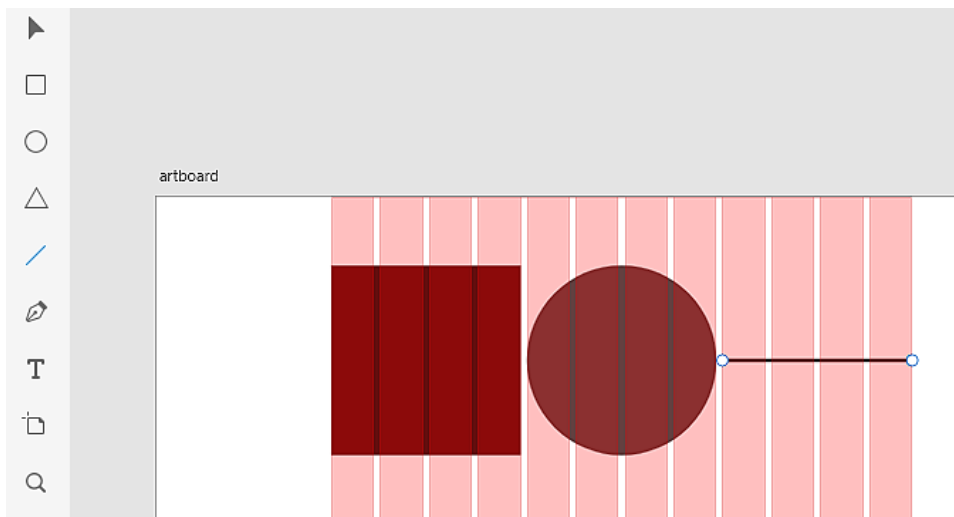


Obr. 3: Výsledný bootstrap grid, po zadání hodnot

2.2.1 Karta Design

Program samozřejmě podporuje přidání geometrických tvarů nebo bezierové křivky (v Adobe programech se bezierova křivka nazývá nástroj Pero). Další nástroje slouží k přidání textu, přidání artboardu a přibližování.

Na Artboard můžeme přidávat také obrázky, a to buď jako samostatné objekty, nebo lze obrázky vkládat do tvarů. V nejnovější verzi Adobe XD lze přidávat video a animace.



Obr. 4: Nástroje programu Adobe XD

2.2.2 Karta Prototype

Prototypování v Adobe XD slouží k demonstraci toho, jak bude web fungovat a jestli bude mít animace, přechody apod.

V programu lze vytvořit téměř cokoliv, ale záleží na tom, jak s programem umíme.

Program nabízí tzv. trigger, které mezi artboardy vykonávají akci, viz seznam níže

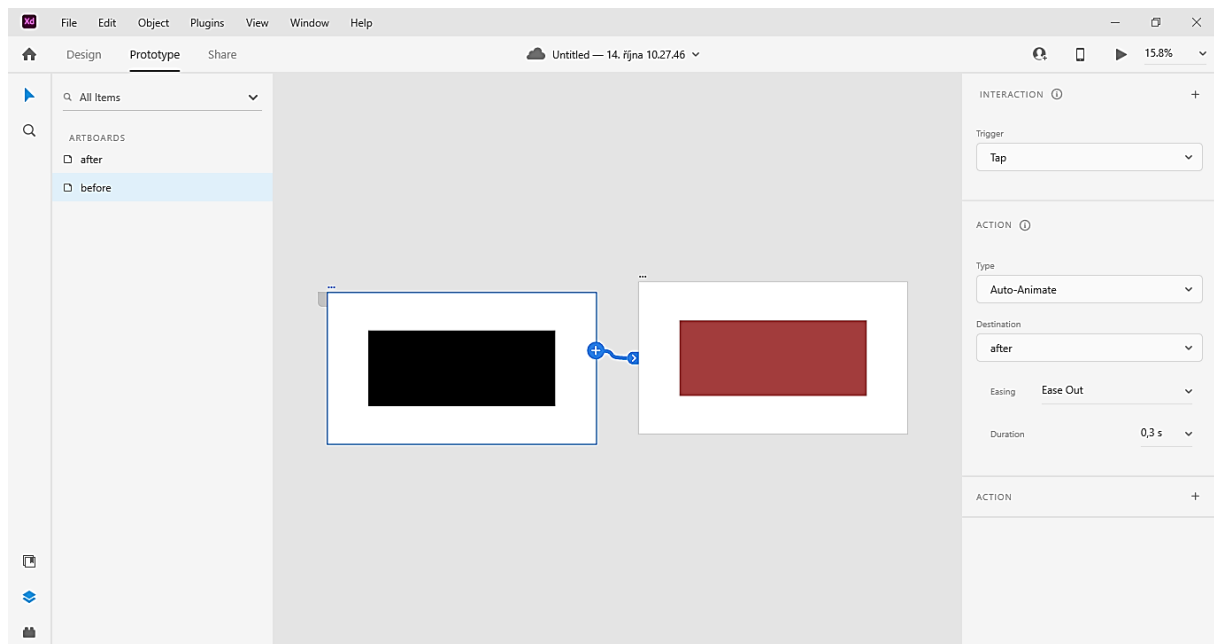
- Hypertextové odkazy
- Přechody
- Automatické animace
- Skrolování
- Překrytí
- Přehrávání řeči
- Přehrávání zvuku

„Trigger“ jsou interakce mezi uživatelem a webovou stránkou v Adobe XD a patří mezi ně:

- Kliknutí
- Stisknutí klávesy
- Časovač

- Táhnutí (pouze pro dotykové obrazovky)

Na obrázku níže je karta Prototype, kde se po kliknutí změní obdélník z černé barvy na vínovou.

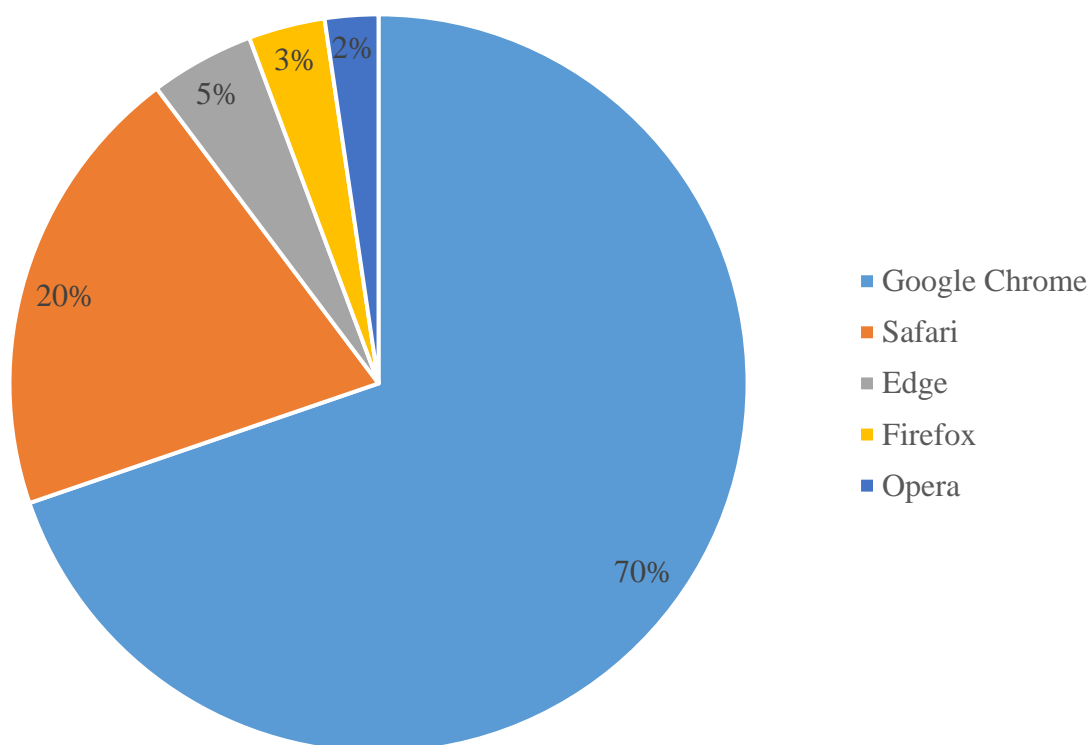


Obr. 5: Karta Prototype

3 Kódování webu

Pro kódování webu slouží výše zmíněný jazyk HTML5, protože obsahuje tagy, které nám umožňují optimalizovat strukturu webu pro vyhledávače. Při samotné tvorbě webu je důležité mít prohlížeč, který obsahuje *nástroje pro vývojáře* a server. Jsou dvě možnosti, jak si vytvořit webový server. Buď využít náš počítač jako server s tím, že bude přístupný pouze nám, nebo druhá varianta, a to použít doménu zdarma např. od Endory, což je poskytovatel webhostingu. Při ladění webu se využívají v každém prohlížeči tzv nástroje pro webové vývojáře. Ty obsahují velmi užitečné funkce jako např. Responzivní design, který nám simuluje různé šířky zařízení, kdy díky tomu můžeme vidět vzhled webu na tabletu, telefonu, a dalších zařízeních. Tomu se budeme věnovat později v další části. Co se týče samotného prohlížeče, je to volba čistě na programátorovi V grafu na další stránce uvádím nejoblíbenější prohlížeče mezi běžnými uživateli za rok 2022.

Oblíbenost vyhledávačů



Z grafu je patrné, že nejvíce oblíbený je Google Chrome. To je z toho důvodu, že běží téměř na všech telefonech se systémem Android a na počítačích.

3.1 Tvorba menu

Struktura webu se od verze HTML5 definuje pomocí výše zmíněných sémantických tagů, kdy každý představuje nějakou část webu.

Webový vyhledávač takto chápe, co jednotlivé části představují na webu, a posouvá ho na vyšší příčku oproti ostatním webům, které tyto tagy nepoužívají. Pro ukázkou zde předvedu, jak může vypadat definování menu v HTML5:

```
<nav class="topnav" id="navigation">
  <a class="nav-link" href="#"></a>
  <a class="nav-link" href="#Intro">ÚVOD</a>
  <a class="nav-link" href="#DetailInfo">DETAILNÍ INFO</a>
  <a class="nav-link" href="#gallery">GALÉRIE</a>
  <a class="nav-link" href="#contact">KONTAKT</a>
  <a href="javascript:void(0);" class="icon" onclick="MenuZapVyp()">
    <i class="fa fa-bars"></i>
  </a>
</nav>
```

Výše máme nadefinováno menu v HTML5, a to tak, že jsme si vytvořili třídu *topnav* a identifikátor *navigaton*. Jednotlivé položky v menu jsou tvořeny pomocí odkazů ``, který nás po kliknutí odroluje na vybranou část webu. Tato část webu musí být tvořena tak, že se k příslušné pasáži webu přiřadí identifikátor a ten se poté napíše do atributu `href="#"` za hashtag. Další věc, kterou je potřeba vytvořit, je „*Media queries*“ podmínka, kde nastavíme maximální šířku např. 600px, což je běžné zobrazení mobilních telefonů.

```
@media screen and (max-width: 600px) {
  .topnav a:not(:first-child) {
    display: none;
  }
  .topnav a.icon {
    float: left;
    display: block;
  }
}
```

Do „*media queries*“ podmínky se napíše vnořený selektor třídy „*topnav*“. Následující část slouží k výběru odkazů, které jsou „*potomci prvního <a> tagu*“. Ve zkratce to znamená, že první tag nebude stylován, ale ostatní ano.

V tuto chvíli bude menu funkční na desktopu, ale na telefonu funkční nebude. Abychom měli menu plně responzivní, je potřeba vytvořit nějaký skript, který se spustí po načtení webu. Níže si uvedeme skript, který používá náš web.

```
function MenuZapVyp() {
    var menu = document.getElementById("navigation");
    if (menu.className === "topnav") {
        menu.className += "responsive";
        console.log("MenuRozkliknuto");
    } else {
        menu.className = "topnav";
        console.log("MenuNerozkliknuto");
    }
}
```

„**Function**“ je ekvivalent ručně vytvořených metod z jiných programovacích jazyků (např. C#, Java) a volá se v případě, že se vykonává nějaká událost *angl. event*. Každá událost má mít název takový, aby bylo poznat, k čemu slouží. Každý programovací jazyk pracuje s proměnnými *angl. variables*. V našem skriptu máme proměnnou „*menu*“ svázanou pomocí metody „*Document.getElementById()*“ HTML tagem, který má přiřazený identifikátor „*navigation*“. Tento řádek dá vyhledávači najevo, že se ve skriptu proměnná odkazuje na HTML tag. Na dalších řádcích kódu máme podmínku se striktním operátorem, který je zapisován třemi rovnítky. Striktní rovnost slouží k přetypování hodnot v proměnných. V našem případě, pokud podmínka platí, se místo klasického menu zobrazí ikona tří čárek, které se říká **Hamburger menu**. Pokud ale podmínka neplatí, zobrazí se klasické řádkové menu.

Na obrázcích 6 a 7 níže vidíme porovnání menu, pokud si web zobrazíme na desktopu a telefonu.



Obr. 6: Menu na desktopu

Obr. 7: Menu na telefonu

3.2 Tvorba hlavičky webu

Hlavička webu je sekce, kde by měly být úplně nezákladnější informace, které jasně říkají, k čemu web slouží. Hlavička může obsahovat fotografie, video, různé animace a krátké texty.

Na webu se používají tři typy hlavičky:

- Fullframe,
- Fullframe-fix,

- Bannerová hlavička.

My na webu použijeme „Bannerovou hlavičku,“ což je typ, který má na šířku 1920px a na výšku 600px. Protože chceme, aby náš web byl responzivní, nepoužíváme fixní hodnoty pro šířku, ale musíme nastavit procentuální hodnoty, v našem případě 100 %, což je ekvivalent poměru 1:1. Obsahem hlavičky může být nějaký multimediální obsah, což je v našem případě logotyp

a fotografie. Když začneme na webu skrolovat dolů, fotografie „pluje“ s námi. Tomuto efektu se říká „*Parralax scrolling*“. Jedná se o takovou základnější interakci webu s uživatelem. Níže uvádím HTML kód pro hlavičku webu.

```
<header class="col-12">
  <picture>
    <source srcset="header/logotype.svg" type="image/svg" />
    
  </picture>
</header>
```

Hlavička webu se deklaruje tagem `<header>`, zakončuje se stejně. Naše hlavička má atribut „*col-12*“, jak už jsem avizoval v kapitole Bootstrap, kde se zmiňuji o 12 sloupcovém systému. Třída „*col-12*“ je 12 sloupec s šířkou 100 %, což znamená, že hlavička bude 1:1. Obsahem hlavičky je logotyp, který byl vytvořen ve vektorovém programu **Affinity Designer**. Logotyp je ve dvou grafických formátech:

- SVG: pro nové prohlížeče
- PNG: pro starší prohlížeče

Logotyp má přiřazen třídu „*img-fluid img-thumbnail*“. Ta v Bootstrapu zajistí, že se logotyp bude zmenšovat na základě šířky zařízení. Z hlediska SEO optimalizace je důležité přidat do `` vlastnost „*alt=""*“. Tato vlastnost přidá alternativní popis, pokud se Obr. nezobrazí. Obrázky by měly mít cca 20 kB, protože čím menší obrázky, tím rychleji se web načte. Optimálně by se web měl načítat do 2 vteřin. Pokud se načítá déle, 99 % uživatelů místo čekání bude hledat jiný web.

CSS styly hlavičky vypadají následovně:

```
header {
  width: 100 %;
  height: 696px;
  background-image: url("header/hlavicka.png");
  background-size: cover;
```

```

background-repeat: no-repeat;
background-attachment: fixed;
text-align: center;
vertical-align: middle;
padding-top: 5rem;
}

```

Použijeme selektor header, kterým vytvoříme vazbu na tag `<header>`. Tím docílíme, že se nám bude stylovat pouze tento element `<header>`. Hlavička má v pozadí fotografii, kterou nahrajeme přes CSS atribut „`background-image: url(„cesta_k_obrazku“);`“. Atributy „`background-size: cover;`“ a „`background-repeat: no-repeat;`“ zajistíme, že se Obr. přizpůsobí šířce hlavičky a nebude se nikde opakovat. Vlastnost „`background-attachment: fixed;`“ způsobí, že s námi Obr. bude „putovat“ jak bylo výše zmíněno. Následující atributy nejsou tolik důležité. Pouze říkají, aby se texty a jiné elementy zobrazovaly na střed stránky. Atribut „`padding-top: 5rem;`“ nám vytvoří vnitřní šířku v horní části hlavičky. Proto je logotyp posunutý ve středu obrázku.



Obr. 8: Hlavička webu na desktopu



Obr. 9: Hlavička webu na telefonu

Jakmile začneme procházet web, vyskočí na nás oranžová šipka a putuje s námi. Tomuto prvku se říká **backlink**. Když na tuto šipku klikneme, vrátí nás zpět na začátek webu. Je to velmi užitečná věc, protože uživatel webu nemusí skrolovat myší zpátky na začátek. HTML kód vypadá následovně:

```

<button onclick="scrollUp()" id="backlink"></button>

```

Protože se jedná o tlačítko, je potřeba, aby mělo vlastnost „`onclick=""`“. Obsah onclick bude „`scrollUp()`“. Další atribut je identifikátor „`backlink`“. Uvnitř máme Obr. šipky. CSS kod backlinku vypadá následovně:

```
#backlink {
  display: none;
  position: fixed;
  bottom: 20px;
  z-index: 99;
  border: none;
  outline: none;
  background-color: rgba(255, 255, 255, 0);
  width: 159px;
}
```

V CSS vybereme selektorem identifikátor „**#backlink**“. Použijeme atribut „**display: none;**“ a to z toho důvodu, že backlink se zobrazí až poté, co začneme skrolovat. Pozice šipky je „**fixed**“. Tím docílíme, aby s námi šipka putovala při prohlížení webu. Javascriptový kód vypadá takto:

```
var backbutton = document.getElementById("backlink");
window.onscroll = function () { TopScroll() };

function TopScroll() {
  if (document.body.scrollTop > 20 || document.documentElement.scrollTop >
  20) {
    backbutton.style.display = "block";
    console.log("ŠipkaZobrazena");
  }
  else {
    backbutton.style.display = "none";
    console.log("ŠipkaNezobrazena");
  }
}
function scrollUp() {
  document.body.scrollTop = 0;
  document.documentElement.scrollTop = 0;
  console.log("SkrolujuNahoru")
}
```

Máme vytvořenou proměnnou s názvem „backbutton“. Ta je propojena s identifikátorem „**#backlink**“. Poté zvolíme metodu window.onscroll a přiřadíme jí funkci „**TopScroll()**“. Následně vytvoříme funkci, která nám zajistí, že když uživatel skroluje webem dolů a překročí 20px, ukáže se backlink. Funkce „ScrollUp()“ způsobí, že po kliknutí na backlink se vrátíme na začátek webu.

3.3 Textové pasáže webu

Textové pasáže jsou oblast webu, kde se nejčastěji vyskytují klíčová slova. Nejčastěji je připravuje **copywriter**. Texty je důležité mít např. ve Wordu předtím, než se vloží do webové

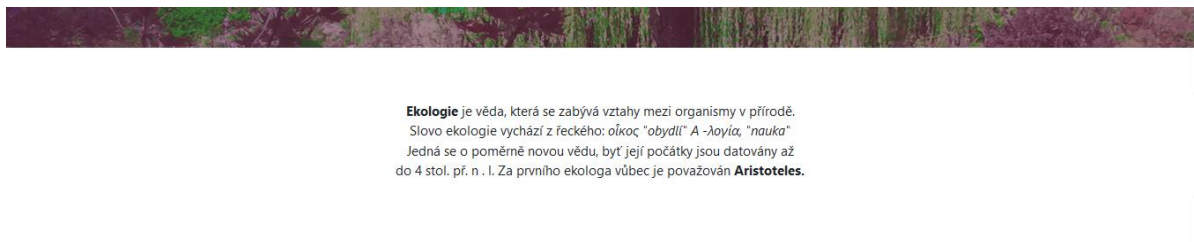
grafiky v Adobe XD. Z programu Adobe XD se potom vykopírují do HTML. Obecně platí, že na webu by mělo být textu co nejméně. Sazba textu do webu se dělí na tři typy:

1. jednosloupcový text
2. půlsloupec ½
3. 1/3 sloupce

Jednosloupcový text je vždy úvodní povídání o webu a je přes celých 12 sloupců. Text je zhruba o čtyřech řádcích a v kódu vypadá následovně:

```
<main class="col-12" id="Intro">
  <p><strong>Ekologie</strong> je věda, která se zabývá vztahy mezi organismy v
  přírodě.<br> Slovo ekologie vychází z řeckého: <i>oἶκος "obydlí" A-λογία,
  "nauka"</i><br> Jedná se o poměrně novou vědu, byť její počátky jsou datovány až
  <br> do 4 stol. př. n . l. Za prvního ekologa vůbec je považován
  <strong>Aristoteles.</strong>
  </p>
</main>
```

Tag **<main>** označuje hlavní část webu. V ní se nejčastěji nachází nejdůležitější části webu. Např. u e-shopu se může jednat o tlačítka, která nás hned nasměrují na produkty atp. Do hlavní části vložíme náš text. Textový odstavec se vždy deklaruje pomocí tagu **<p>**. Poté do něj vložíme náš text, který se zobrazí. Následně použijeme tag ****, který má úplně stejnou funkci jako tag ****, s tím rozdílem, že dává jednotlivým slovům větší váhu a vyhledávač takové slovo označuje za důležité. Text poté ještě potřebujeme na třech řádcích zalomit pomocí tagu **
**. Tento tag nám vytvoří výsledek na následujícím obrázku 10:



Obr. 10: Příklad jednosloupcového textu

Další velmi používaný případ textových odstavců jsou půlsloupcové texty. Ty se používají nejčastěji v novinových článcích, kde jsou dlouhé texty. Tyto texty jsou vždy doplněny nějakou grafikou, jako jsou např. různé geometrické tvary, ikony, diagonály apod. My pro ukázkou použijeme dva textové sloupce, které obsahují tagem **** zvýrazněná klíčová slova. Níže uvádím kód v HTML.

```
<section class="col-12" id="DetailInfo">
  <article id="odstavec1">
    <p class="col-6" id="prvniText">
```

```
Mezi témata, jimiž se ekologie zabývá, patří rozmanitost, distribuce, hmota
<strong>(biomasy)</strong> a počet (populace) jednotlivých organismů, jakož i
spolupráce a konkurence mezi organismy, a to jak uvnitř, tak mezi ekosystémy.
<strong>Ekosystémy</strong> se skládají z dynamicky navzájem propojených částí
včetně organismů, komunit, jež vytváří z neživých složek jejich prostředí.
<strong>Ekosystémové procesy</strong> jako primární produkce, pedogeneze, koloběh
živin, a další různé činnosti směřující k vytváření niky regulují toky energie a
hmoty skrze prostředí. Tyto procesy jsou udržovány organismy se specifickými
životními vlastnostmi, pro označení rozmanitosti organismů se používá termín
<strong>biodiverzita (biologická rozmanitost).</strong> Biodiverzita poukazuje na
rozdílnost druhů, genů a ekosystémů a podporuje některé ekosystémové služby.
</p>
</article>
```

Pasáž s texty je vnořena do tagu `<section>` s atributem `id="DetailInfo"`. Každý odstavec musí být součástí tagu `<article>`, aby byla dodržena optimalizace struktury webu. Konkrétní text je součástí tagu `<p>`. Níže máme CSS kód.

```
#DetailInfo {
    width: 100%;
    height: auto;
    padding-top: 10rem;
    text-align: left;
    padding-bottom: 10rem;
}
#prvniText {
    margin-right: auto;
    margin-left: auto;
    display: block;
    text-align: justify;
}
#odstavec1 {
    width: 50%;
    height: auto;
    float: left;
}
```

Identifikátor `#DetailInfo` má nastavenou šířku 100 %, a `#prvniText` s `#odstavec1` mají nastavenou šířku na 50 %, protože druhý odstavec má také šířku 50 %. Výsledek na webu vypadá jako na obrázku 11:

Mezi témata, jimiž se ekologie zabývá, patří rozmanitost, distribuce, hmota (**biomasy**) a počet (populace) jednotlivých organismů, jakož i spolupráce a konkurence mezi organismy, a to jak uvnitř, tak mezi ekosystémy. **Ekosystémy** se skládají z dynamicky navzájem propojených částí včetně organismů, komunit, jež vytváří z neživých složek jejich prostředí. **Ekosystémové procesy** jako primární produkce, pedogeneze, koloběh živin, a další různé činnosti směřující k vytváření niky regulují toky energie a hmoty skrze prostředí. Tyto procesy jsou udržovány organismy se specifickými životními vlastnostmi, pro označení rozmanitosti organismů se používá termín **biodiverzita (biologická rozmanitost)**. Biodiverzita poukazuje na rozdílnost druhů, genů a ekosystémů a podporuje některé ekosystémové služby.

Existuje mnoho praktických aplikací ekologického myšlení v ochraně přírody, řízení přírodních zdrojů (např. **agroekologie, zemědělství, lesnictví, agrolesnictví, rybolov**), urbanismu (**ekologie města**), komunitním zdraví, ekonomii, základní a aplikované vědě, stejně jako v lidských sociálních interakcích (ekologie člověka). V původním a správném významu je tedy ekologie věda, která se zabývá vztahem organismů a jejich prostředí a vztahem organismů navzájem. Jako první tak nazval a definoval tento vědní obor **Ernst Haeckel** v roce **1866**. Dále se ekologie užívá chybně v širokém smyslu jako ochrana životního prostředí nebo dokonce místo přírodní prostředí (např. **ekologicky šetrný výrobek** znamená výrobek šetrný k životnímu prostředí).

Obr. 11: Půlsloupcové odstavce na webu

3.4 Fotogalerie na webu

Galerie na webu patří mezi stěžejní části, protože pomocí ní můžeme návštěvníkovi blíže představit naše služby. Galerie se obecně dělí na dva typy:

1. interní
2. externí

Interní galerie je vždy součástí webu a obsahuje minimálně 8 fotografií. Fotografie mohou mít různé interakce, např. změna fotografie ze stupně šedi na barevnou. Galerie může být ale také série snímků, kde se jednotlivé fotky střídají mezi sebou. Tyto galerie se nazývají „*slideshow*“, „*image carousel*“. Obrázky mohou být podtrženy různou grafikou nebo animacemi. Externí galerie se používá v případě, že fotky umístíme na sociální síť jako Facebook, Instagram, Rajce.net. Většinou funguje tak, že na web vloží `<iframe>`, který zobrazí třeba dvě fotky. Pokud chce návštěvník vidět více, musí na fotky kliknout, a to nás přesměruje na konkrétní sociální síť.

My jsme použili na webu interní galerii s výše zmíněnou slideshow. Obrázky se po 5ti vteřinových intervalech střídají. Toto se vždy tvoří pomocí HTML, CSS a Javascriptu. My používáme W3C.css framework, který je zdarma poskytován W3C konsorciem. Jedná se o takový menší Bootstrap, přičemž tento framework má menší datový objem, ale není tak komplexní. HTML kód galerie vypadá následovně:

```
<section class="col-12" id="gallery">
  <picture class="img-responsive">
    <div class="col-8" id="photos">
      <div class="w3-content w3-section">
        
          <!--dalsi_fotky-->
        </div>
      </div>
    </div>
  </div>
```

```
</picture>
</section>
```

Galerie je tvořena tagem `<section>` a má dva atributy, a to třídu „*col-12*“ a identifikátor „*gallery*“.

Uvnitř tagu máme vnořený tag `<picture>`, který je určen pro více fotografií. Tento tag má dvě třídy „*img-responsive*“. Tím zajistíme vazbu na bootstrap styl, který zajistí responzivní šířku webu. Pro zarovnání fotografií na šířku máme vytvořen blokový tag `<div>` s identifikátorem „*photos*“. Jednotlivé fotky přidáváme pro pomocí tagu ``. Ten poté musí mít atributy. My, pokud chceme nahrát Obr. do HTML, potřebujeme atribut `src="cesta_k_obrazku"`. Další atribut je třída „*mySlides w3-animate-left*“. Ta je součástí w3c.css frameworku a zajistí nám přechody mezi fotkami s animacemi. Poslední atribut, který potřebujeme, je identifikátor „*foto*“. CSS kód uvádím níže:

```
#gallery {
  width: 100%;
  height: auto;
  background: #064600;
  padding-top: 5rem;
  padding-bottom: 5rem;
}
#photos {
  text-align: center;
  display: block;
  margin-left: auto;
  margin-right: auto;
}
#foto {
  width: 100%;
  height: auto;
  margin-bottom: 6px;
  border-radius: 12px;
}
```

V CSS je potřeba pomocí identifikátoru *#gallery* dát naší galerii šířku 100 % a výšku bude mít auto, čímž se bude přizpůsobovat fotografiím. Další styl slouží k vystředění fotografií, To se tvoří pomocí vlastností *display: block*, *margin-left*, a *margin-right*, kterým dáme hodnotu **auto**, případně můžeme zarovnání na střed docílit vlastností *text-align: center*. Poslední styl nám zajistí lepší responzivní fotografie, vnější šířku 6px a zakulacení rohů, které je dostupné od CSS3. Samotné přepínání je řešeno Javascriptem pomocí tohoto kódu:

```
var myIndex = 0;
slideshow();

function slideshow() {
  var i;
```



```

var slides = document.getElementsByClassName("mySlides");

for (i = 0; i < slides.length; i++)
{
    slides[i].style.display = "none";
}

myIndex++;

if (myIndex > slides.length)
{
    myIndex = 1
}

slides[myIndex - 1].style.display = "block";
setTimeout(slideshow, 3000);
console.log("DalsiFotka")
}

```

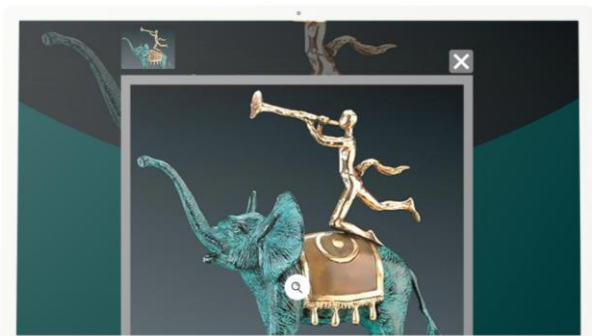
Slideshow funguje tak, že se zavolá funkce a poté cyklus for projde všechny obrázky v galerii, přičemž zobrazí vždy ten, na kterém se nachází index i. Poté se nastaví funkce setTimeout() s argumenty slideshow a hodnotou 3000 milisekund.

3.4.1 Lightboxy

Lightbox je webový prvek, který slouží k zobrazení větší fotografie, než je na webu. Mezi nejtypičtější variantu patří překrytí webu obrázkem s tmavším průhledným pozadím po kliknutí. Lightbox nemusí mít jenom fotografie, ale také video, popř. nějaké animace a 3D modely. Lightboxy můžou mít různé podoby:

- posouvání
- přechody
- rotace

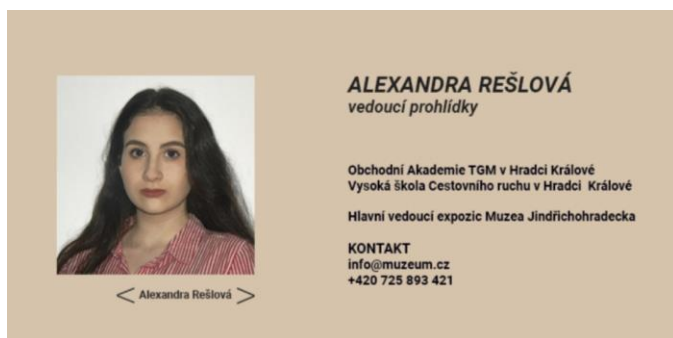
Společná vlastnost těchto prvků je vypínací ikonka, ikony šipek pro další obrázky. Na našem webu se lightboxy nezobrazují z toho důvodu, že naše galerie má velké obrázky. Lightbox musí být plně responzivní. Na obrázku 12 máme jeden s příkladů lightboxu:



Obr. 12: Příklad lightboxu

3.4.2 Medailonky

Pomocí medailonků můžeme představit návštěvníkům webu pracovníky např. nějaké společnosti. Typicky se jedná například o pedagogický sbor s kontakty na pracovníky a jejich funkci ve společnosti. Medailonky jsou většinou fotografie v různých tvarech (čtverce, kruhy, a jiné). Pod medailonky by měl být kontakt na pracovníka ve formátu emailové adresy a telefonního čísla. Emailová adresa by měla být bez klasického zavináče kvůli internetovým botům, kteří skenují weby. Zavináč by měl být nahrazen buď obrázkem, nebo v těchto formátech: **(at)**, **zavinac** apod. Níže máme příklad medailonku. V tomto případě si může návštěvník stránky pomocí šipek přepínat mezi pracovníky, viz Obr. 13:



Obr. 13: Příklad medailonku

3.5 Formuláře na webu

Formuláře patří mezi stěžejní části webu. Díky formuláři s námi můžou komunikovat naši potenciální zákazníci. Formulář se vždy skládá s textových polí tlačítka, zaškrťovacího políčka a CAPTCHA ověření. Formuláře se dělí na několik typů:

Každý formulář se skládá ze dvou částí:

1. Vzhled: vizuál formuláře

2. Logika: funkčnost formuláře

Další trochu speciálnější varianta je formulář pro odběr newsletterů, který je vždy v patičce. Ten se skládá s textového pole pro zadání emailu a tlačítka pro odebírání novinek.

My používáme kontaktní formulář a zabýváme se pouze jeho vzhledem. Logiku nebudeme tvořit, protože je potřeba pomocí **backendu** nastavit poštovní server, šifrování přenesených dat, a mnoho dalšího. To je vždy záležitost **backend vývojáře**. Náš formulář se skládá s textových polí pro zadání:

- jména,
- příjmení,
- e-mailu,
- telefonního čísla,
- zprávy, popř. dotazu.

Dále je potřeba, aby uživatel zaškrtl klasické „*Souhlasím se zpracováním osobních údajů*“. Poslední prvek, který potřebujeme, je tlačítko pro odeslání dat. Níže uvádím HTML kód.

```
<form method="POST" class="col-12" id="contact">
  <div class="inputFields">
    <h1 id="nadpis">KONTAKTNÍ FORMULÁŘ</h1>
    <input type="text" id="name" value="*Jméno" name="jmeno" class="col-5"><br>
    <input type="text" id="surname" value="*Příjmení" name="prijmeni" class="col-5"><br>
    <input type="email" id="email" value="*Email" name="email" class="col-5"><br>
    <input type="tel" id="phone" value="*Telefon" name="mobil" class="col-5"><br>
    <input type="text" id="message" value="*Zpráva" name="zprava" class="col-5"><br>
    <input type="checkbox" name="souhlas" id="souhlas">
    <label for="souhlas">Souhlasím se zpracováním osobních údajů</label><br>
    <input type="submit" id="odeslat" class="col-5">
  </div>
<!--KOD_POKRACUJE-->
```

Formulář se deklaruje pomocí tagu `<form>` a má tři atributy. První atribut je metoda „POST“.

Tato metoda slouží k hashování údajů při odeslání formuláře. Tento atribut se týká jazyka PHP, kterým se nezabýváme, protože se jedná o backendovou část. Druhým atributem je třída odkazující se na bootstrap a třetí atribut je identifikátor pro stylování formuláře.

Druhý tag `<div>` slouží k pozicování části formuláře. Další prvek je nadpis první úrovně, který ve formuláři může a nemusí být. Poté už následují formulářové prvky. V HTML5 se používá pro deklaraci prvků tag `<input type="" id="" value="" name="" class="">`. Prvky mají samozřejmě také atributy. Vlastností `type` bude `text`. Tím webovému vyhledávači řekneme, že

máme ve formuláři textové pole. Druhá vlastnost je *identifikátor* pro stylování prvku. Třetí vlastnost je *value*, do které patří text, který se bude v poli defaultně zobrazovat. Čtvrtá vlastnost je *name*. Ta slouží k propojení HTML a PHP, kdy můžeme z textového pole odesílat data na server. Poslední vlastností je odkaz na bootstrap. Další textová pole se dělají totožně s tím rozdílem, že se jim změní atributy.

Formulář musí obsahovat zaškrťovací políčko pro zpracování osobních údajů. Ten se tvoří pomocí `<input type="checkbox" name="souhlas" id="souhlas">`. Tím vytvoříme samotné políčko bez doprovodného textu. Text se tvoří pomocí `<label for="souhlas">Souhlasím se zpracováním osobních údajů</label>`. Formulář je zakončen tlačítkem. Tlačítko se tvoří dvěma způsoby. První možností je tag `<button>Odeslat</button>`, což je starší varianta. Druhou možností je `<input type="submit" id="odeslat" class="col-5">`. Náš web používá druhou možnost, protože je prohlížečem více upřednostňována před starší variantou `<button>`.

Níže máme tabulku s příklady vlastností pro `<input type="vlastnost">`

<code><input type="*"></code>	Použití vlastnosti
Text*	vytvoří textové pole
Submit*	vytvoří tlačítko
Email*	vytvoří textové pole pro email
Tel*	vytvoří textové pole pro tel. číslo
Number*	Vytvoří pole pro zadání čísel pomocí šipek

Formulář je potřeba nastylovat, aby nevypadal jako z devadesátých let. Níže máme CSS kód, který si rozeberme:

```
#contact {
  width: 100%;
  height: auto;
  padding-top: 5rem;
  padding-bottom: 5rem;
}
input {
  height: 56px;
  border-radius: 30px;
  border-color: #064600;
  border-style: solid;
  border-width: 3px;
  margin-bottom: 12px;
  background: white;
}
```

```

#message {
  height: 146px;
  border-radius: 30px;
  border-color: #064600;
  border-style: solid;
  border-width: 3px;
  margin-bottom: 12px;
  padding-bottom: 80px;
  word-wrap: break-word;
  background: white;
}
#message:hover {
  border-color: orangered;
  border-style: solid;
  border-width: 3px;
  border-radius: 30px;
  color: orangered;
}
input:hover {
  border-color: orangered;
  border-style: solid;
  border-width: 3px;
  border-radius: 30px;
  color: orangered;
}
.inputFields {
  width: 40%;
  height: auto;
  text-align: right;
  float: left;
}
#nadpis {
  font-size: 1.45vw;
  color: #653057;
  margin-bottom: 1.2rem;
}

```

První věc, kterou musíme nastylovat, je oblast formuláře pomocí selektoru **#contact**. Tomu dáme šířku 100 % a výšku auto současně s vnitřní šířkou v dolní a horní části. Ty budou mít 5rem. Dále je potřeba dát vzhled jednotlivým prvkům ve formuláři. My je stylujeme globálně pomocí selektoru **input**. Tím zajistíme stylování všech formulářových prvků v HTML s názvem „input“. Od roku 2022 je ve webdesignu velký trend mít na webu prvky se zaoblenými rohy. Toto se řeší pomocí vše zmíněné vlastnosti „**border-radius**“ *hodnota v jednotkách*;. Jednotky jsou buďto v pixelech nebo procentech. Jednotlivé prvky by měly mít ohraničení. Na webu máme šířku ohraničení 3px. Toto se tvoří pomocí „**border-width: 3px;**“. Ohraničení prvků chceme mít plnobarevné. To vytvoříme pomocí vlastnosti „**border-style: solid;**“. Poslední důležitou věc, kterou potřebujeme, je barva ohraničení, na kterou máme vlastnost „**border-**

color: barva;“. Barvy můžeme zadat jako název nebo v RGB, popř. hexadecimálně. Naše barva bude v hexadecimální hodnotě tmavě zelená. Druhá věc, kterou potřebujeme vytvořit se selektorem input, je událost **„hover“**. Toto po najetí myši způsobí nějakou akci, např. obarvení prvku. Pro využití události se používá selektor, v tomto případě **„input:hover { css atribut }“**. CSS kód uvnitř selektoru bude úplně stejný jako předchozí input s tím rozdílem, že vlastnosti **„border-color: barva;“** a **„color: barva;“** bude nastaven na **„orangered“**. Toto způsobí, že po najetí se obarví ohraničení a text na oranžovočervenou barvu. Další věc, kterou potřebujeme upravit, je textové pole s obsahem zprávy. Na toto máme identifikátor **„#message“**. Obsah **„#message“** bude totožný se selektorem input s tím rozdílem, že výška pole bude mít 146 px. **„#message:hover“** bude také stejný jako **„input:hover“**. Poslední prvek, který musí mít styl, je nadpis **„KONTAKTNÍ FORMULÁŘ“**. Nadpis vybereme selektorem **„#nadpis“**. Velikost fontu dáme 1vw, což je přibližně 1 % viewport width. Barva fontu bude v hexadecimální hodnotě vínová. Vnější dolní šířka nadpisu bude 1.2rem, což představuje jednotku, která není ovlivněna ostatními elementy. Výsledný formulář vypadá jako na obrázku 14 a na obrázku 15:

KONTAKTNÍ FORMULÁŘ

*Jméno

*Příjmení

*Email

*Telefon

*Zpráva

Souhlasím se zpracováním osobních údajů

Odeslat dotaz

Obr. 14: Formulářové prvky v defaultním stavu

KONTAKTNÍ FORMULÁŘ

*Jméno

*Příjmení

*Email

*Telefon

*Zpráva

Souhlasím se zpracováním osobních údajů

Odeslat dotaz

Obr. 15: Formulářové prvky po najetí myši

3.6 Mapa na webu

Mapy na webových stránkách jsou případ, kdy se používá tzv. iframe neboli interaktivní rámce. Jsou součástí kontaktního formuláře nad patičkou webu. Mapy mohou mít různé formy, např. mít tvar kruhu, čtverce atp. Mapy máme dva typy:

1. statické,

2. interaktivní.

3.6.1 Statická mapa

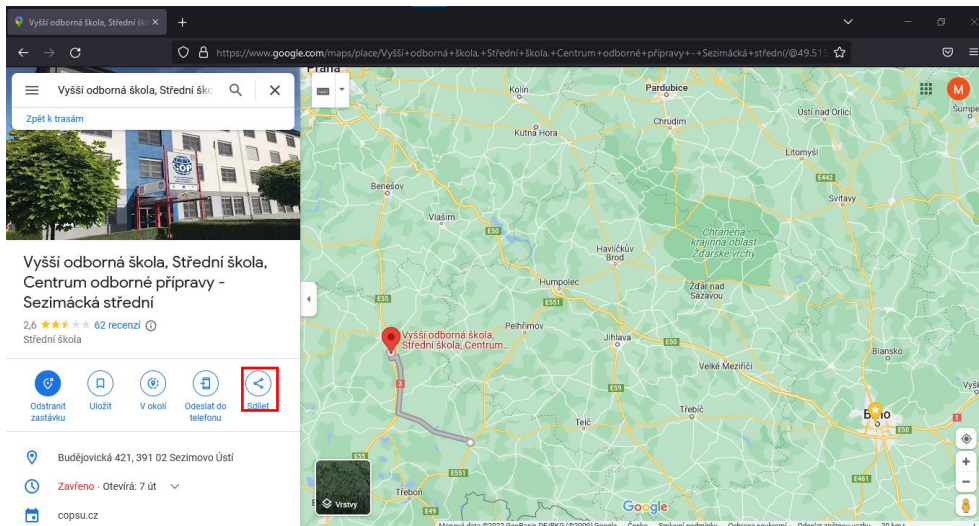
Jsou to klasické obrázky vytvořené jako výstřižek obrazovky a v grafickém editoru přidané do libovolného tvaru jako maska. Nejčastější formáty obrázků map jsou jpg, png a svg (pokud je mapa ručně nakreslená ve vektorovém editoru). Statické mapy se rychle načítají a mají podporu v každém prohlížeči. Mezi jejich nevýhody patří malé rozlišení a není možno si do nich umístit trasové body (musí se připravit v grafickém programu).

3.6.2 Interaktivní mapa

Tyto mapy poskytuje Google/Seznam. Jsou upřednostňované webovými vyhledávací, protože hledané místo se nám integruje do přednostních katalogových zázpisů, když na této mapě umístíte trasovací bod. Interaktivní mapy mají velké možnosti přizpůsobení:

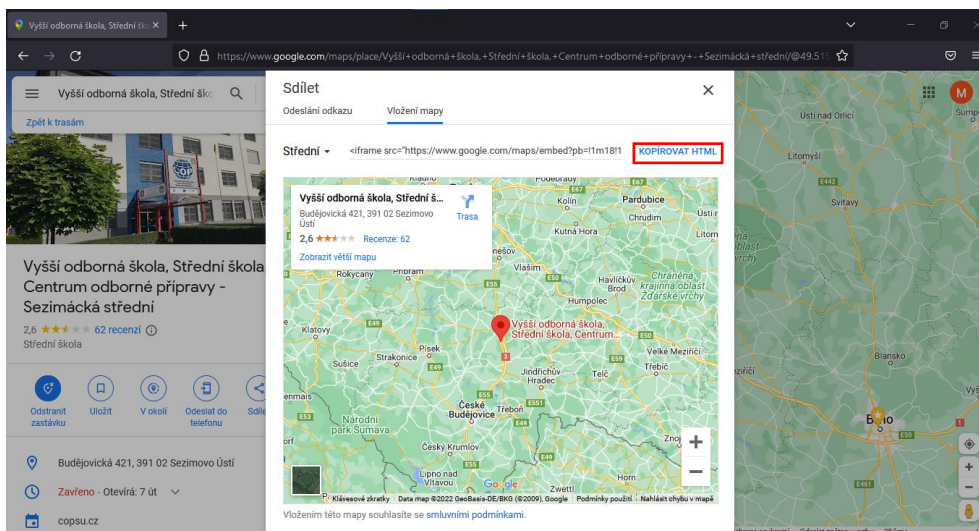
- Street view: procházení pomocí šipek prostředím.
- Přidání lokací.
- Filtrování map (letecké, satelitní, terénní).

Interaktivní mapy se vkládají jako masky to různých geometrických obrazců pomocí skriptů. Nevýhodou interaktivních map je to, že prodlužují načítání webové stránky. Mapa je vnořena do blokového tagu `<div class="mapFrame"></div>`. Tím vytvoříme samostatný oddíl. Do tohoto oddílu potom vložíme mapu. Ta se vkládá pomocí tagu `<iframe>`. Na Google mapách si najdeme náš cíl a klikneme levým tlačítkem na trasovací bod. V tu chvíli se nám zobrazí postranní panel s informacemi o hledaném místě. Mnoho vývojářů netuší, že lze mapy takto jednoduše přidávat do HTML. Některé mapy mívají také pomocí javascriptu vytvořen zámeček. Je to z důvodu, že uživatelé při prohlížení webu najedou nechtěně kurzorem myši na mapu. V tu chvíli se jim posouvá mapa. Google mapy mají toto ošetřené tak, že se mapa aktivuje klávesou CTRL a kolečkem myši. Na další stránce máme návod, jak přidat Google mapy do HTML.



Obr. 16: Detaily o hledaném místě

Poté klikneme na ikonku „Sdílet“ a vyskočí na nás okno jako níže na obrázku. V okně vybereme panel „Vložení mapy“ a klikneme na odkaz „Kopírovat HTML“.



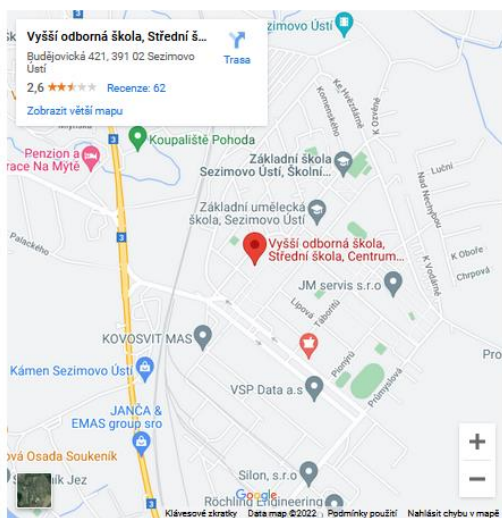
Obr. 17: Okno s tagem <iframe> obsahujícím naši mapu

Na další stránce máme HTML kód s mapou:

```
<div class="mapFrame">
  <iframe class="col-7" id="mapa"
  src=https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d10390.422203248354!2d14.690629235319493!3d49.37853620000001!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x470cbd1795d92eff%3A0x44181db621f7619!2zVnnFocWhw60gb2Rib3Juw6EgxaFrb2xhLCBtdMWZZWRuw60gxaFrb2xhLCBdZW50cnVtIG9kYm9ybsOpIHFDFmcOtcHJhdkgLSBTZXppbcOhY2vDoSBzdMWZZWRuw60!5e0!3m2!1scs!2scz!4v1656950451174!5m2!1scs!2scz allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>
</div>
```

Tag „<iframe>“ má od Google definovaných několik atributů. První je odkaz na bootstrap a druhý je odkaz do našeho externího CSS souboru. Třetí atribut „src“ za nás automaticky

vyplní Google mapy. Čtvrtý atribut „*allowfullscreen*“ má dvě hodnoty: „*true*“ nebo „*false*“. V podstatě se jedná o zobrazení na celou obrazovku. Naše mapa zobrazení na celou obrazovku nemá. Pokud totiž nevybereme ani jednu hodnotu, bude mapa bez „*fullscreenu*“. Předposlední atribut je „*loading*“ a týká se načítání obsahu v iframe. Máme zde pouze dvě volby. První volba se nazývá „*eager*“. Tato volba načte obsah iframe bez ohledu na to, jestli je mimo viditelný výřez. Druhá volba „*lazy*“ vystředí obsah iframe tak, aby se trasovaný bod zobrazil vertikálně a horizontálně uprostřed. Poslední možností je „*referrerpolicy*“. Tento atribut odesílá data na stránku, kterou iframe zobrazuje. Volba, kterou jsme použili v naší mapě, slouží k tomu, aby se neodesílala data zpátky bez zabezpečení HTTPS/TLS.



Obr. 18: Vzhled mapy webu

3.7 Patička webu

Patička webu patří mezi poslední pasáž webové stránky. Většinou obsahuje odkazy na různé sociální sítě, např. pomocí ikon, formulář pro odběr newsletteru, vratnou šipku (angl. backlink). Pokud se jedná o e-shop nějakého existujícího obchodu, může patička obsahovat tabulku s otevírací dobou. Patička našeho webu se skládá ze dvou odkazů, formuláře pro odběr newsletteru a ikon. HTML kód patičky se skládá z následujících tagů:

```
<footer class="col-12" id="paticka">
  <div class="col-1" id="odkazy">
    <ul style="color: white;">
      <strong><a href="https://cs.wikipedia.org/wiki/Ekologie" target="_blank">
        <li>Wikipedie</li>
      </a></strong><br>
      <strong><a href="https://www.thispersondoesnotexist.com/" target="_blank">
        <li>Avatary</li>
      </a></strong><br>
    </ul>
```

```

</div>
<form class="col-7" id="novinky" action="/SubscribeNews.php">
  <input type="email" id="Odber" value="*Email" class="col-4">
  <input type="submit" id="odebirat" class="col-2" value="Odebírat">
</form>
<figure class="col-2" id="icons">
  <a href=""></a>
  <a href=""></a>
  <a href=""></a>
</figure>
</footer>

```

Patička se deklaruje tagem `<footer>` s dvěma atributy. Uvnitř patičky jsou vnořené tři oddíly. První oddíly jsou odkazy v seznamu na webové stránky. Seznam je buď uspořádaný `` nebo neuspořádaný `` (označení piktogramy). Jednotlivé položky seznamu se deklarují tagem ``. Druhý oddíl je formulář s akcí, která se tvoří pomocí programovacího jazyka PHP. Jak bylo výše zmíněno, tuto část tvoří Backend vývojář. Poslední oddíl jsou tři ikony. Oddíl s obrázky se deklaruje tagem `<figure>`. Ikony jsou vnořeny do odkazů, protože se budou odkazovat na sociální sítě. Styly patičky vypadají následovně:

```

footer {
  width: 100%;
  height: auto;
  background-color: #064600;
  padding-top: 5rem;
  padding-bottom: 5rem;
}
ul,
li {
  list-style-type: none;
  text-align: center;
}
#odkazy {
  width: 20%;
  height: auto;
  display: block;
  float: left;
  font-size: 14pt;
  text-decoration: none;
  margin-right: 101px;
}
#Odber {
  border-top-left-radius: 30px;
  border-top-right-radius: 0px;
  border-bottom-right-radius: 0px;
  border-bottom-left-radius: 30px;
}

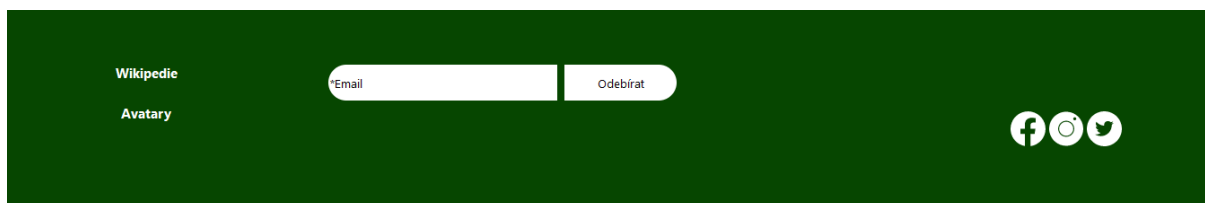
```

```

#odebirat {
    border-top-left-radius: 0px;
    border-top-right-radius: 30px;
    border-bottom-right-radius: 30px;
    border-bottom-left-radius: 0px;
    text-align: center;
}
#novinky {
    float: left;
    display: block;
}
#icons {
    display: block;
    float: right;
}

```

Pomocí CSS selektoru „*ul, li*“ vybereme globálně všechny seznamy a jako CSS vlastnost jim přiřadíme vlastnost „*list-item-style: none;*“. Tím docílíme, aby se naše odkazy nezobrazovaly s puntíky. Dále vybereme v CSS identifikátor „*#odkazy*“. Šířku oddílu s odkazy dáme 20 % a výšku dáme „*auto*“. Vlastnost „*float*“ slouží k tomu, aby byl oddíl obtékán ostatními oddíly. Vlastnost „*text-decoration: none*“ nám zajistí, že nebudou odkazy podtrženy. Následující oddíl s identifikátorem „*#novinky*“, textovým polem a tlačítkem má zaoblené rohy pomocí vlastností „*border-top-left: 30px;*“ a „*border-bottom-left: 30px*“. Zbylé rohy budou normálně ostré. U tlačítka uděláme totožnou věc s tím rozdílem, že místo „*left*“ dáme „*right*“. Oddíl „*#novinky*“ bude obtékán totožně jako předchozí oddíl. Poslední oddíl s identifikátorem „*icons*“ bude stejný jako předchozí oddíl, ale bude mít „*float: right;*“. Naše patička vypadá následovně:



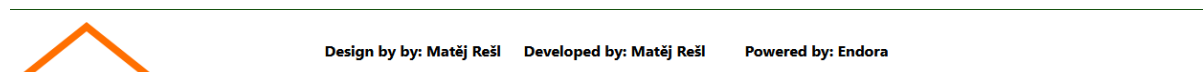
Obr. 19: Patička webu

3.7.1 Superpatička

Superpatička obvykle slouží jako tiráž webu. Většinou jsou zde informace o webdesignérovi, programátorovi a poskytovateli webhostingu. Informace mohou být ve formě odkazů nebo pouze klasických popisků. Programátor a webdesignér by měli mít před jménem znak copyrightu. Poslední věc, kterou může superpatička obsahovat, je mapa webu. Mapa webu je hierarchická struktura, kde každá část webu je dělena na větve. Tuto vlastnost lze vygenerovat do xml souboru pomocí webové stránky *xml-sitemaps.com*. Generování mapy pro náš web nepotřebujeme. Níže si projdeme HTML kód superpatičky:

```
<section class="col-12" id="superfooter">
  <label class="col-2"><strong>Design by by: Matěj Rešl</strong></label>
  <label class="col-2"><strong>Developed by: Matěj Rešl</strong></label>
  <label class="col-2"><strong>Powered by: <a href="https://www.endora.cz"
    style="color: black;">Endora</a></strong></label>
</section>
```

Super patička bude sekce s identifikátorem „*superfooter*“. Jednotlivé popisky budou deklarovány tagem `<label>`. Obsah „labelů“ bude v tagu ``. Poslední popisek bude mít slovo „Endora“ jako odkaz na poskytovatele webhostingu. Superpatička je tímto hotová, viz Obr. 20:



Obr. 20: Vzhled superpatičky

3.8 Převod od responzivní podoby

Tímto je desktop verze hotová a následuje převod do responzivní podoby. To se tvoří pomocí CSS3 vlastnosti „*@media queries*“. K media queries se potom přiřadí buďto „*max-width: rozlišení v px*“ nebo „*min-width: rozlišení v px*“. Uvnitř vlastnosti je potom vnořen ten selektor, který chceme přizpůsobit pro jiná rozlišení. Pro tablety se používá většinou rozlišení 768px a pro telefon rozlišení 414px. Počet #media queries vlastností je neomezen, protože většina vnořených selektorů bude mít stejně šířku 100 %. Níže máme příklad použití media queries:

```
@media (max-width: 800px)
{
  .inputFields {
    width: 100%;
    height: auto;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
    display: block;
  }
  #nadpis {
    font-size: 24pt;
    color: #653057;
  }

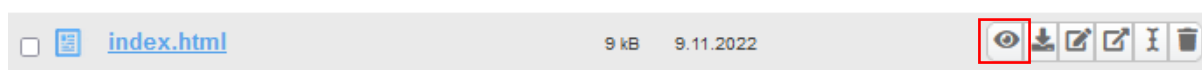
  .mapFrame {
    width: 100%;
    height: auto;
    text-align: center;
    padding-top: 5rem;
  }
}
```

4 Ladění webu

Ladění webu patří mezi velmi důležité procesy tvorby webových stránek. Díky ladění webu můžeme vychytat různé chyby, díky kterým náš web nefunguje tak, jak by měl. V zásadě jsou dva způsoby, jak odladit webovou stránku.

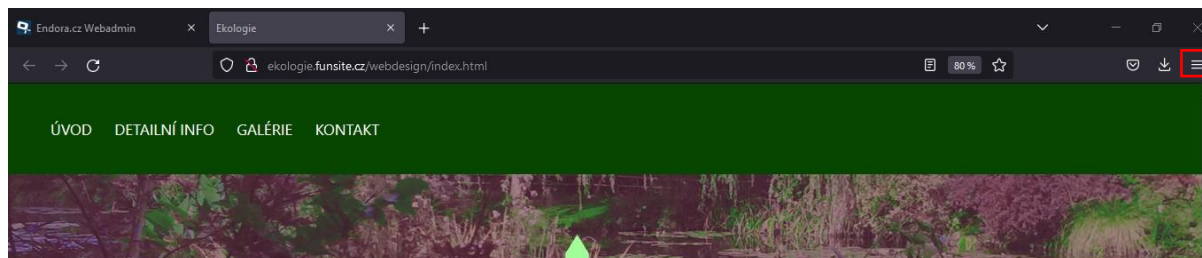
1. Lokálně na počítači
2. Pomocí FTP klienta se připojit na webový server např. Endora.

My využijeme druhý případ. Operační systém Windows 10 umožňuje v průzkumníkovi souborů připojit se na webový server pomocí protokolu FTP, ale je to velmi nepřehledné. Je potřeba se přihlásit do webového rozhraní, kam nahrajeme náš web. Po přihlášení klikneme na položku „Správce souborů“ a v seznamu nic nemáme. Klikneme na tlačítko nahrát soubory a vybereme náš zip archív. Musíme ale zaškrtnout políčko „Rozbalit nahrané archívy“. Archív se po chvílce nahraje a hned nato uvidíme složku se soubory našeho webu. Přejdeme do složky s naším webem a klikneme na tlačítko, viz Obr. č. 21:



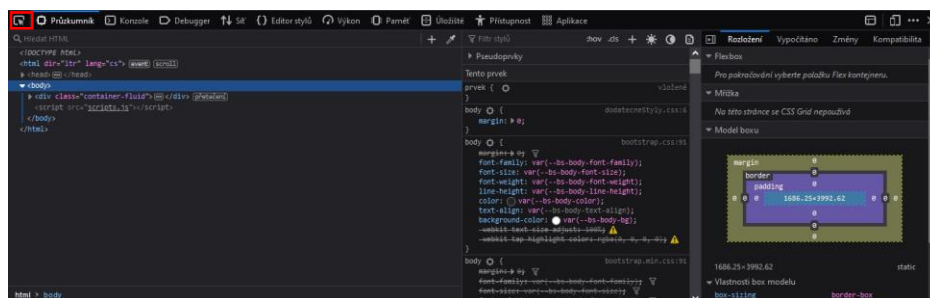
Obr. 21: Kliknutím na ikonku oka (v červeném obdélníku) zobrazíme náš web v novém panelu

Poté se náš web zobrazí v novém panelu. V tuto chvíli klikneme na hamburger menu našeho webu v pravém horním rohu našeho prohlížeče, viz Obr. č. 22:



Obr. 22: Hamburger menu našeho prohlížeče (v červeném obdélníku)

Po kliknutí na hamburger menu vybereme položku „Další nástroje“. V tu chvíli se objeví položky, kde vybereme „Nástroje pro webové vývojáře“. Poté se nám v dolní části zobrazí panel, viz Obr. č 23:

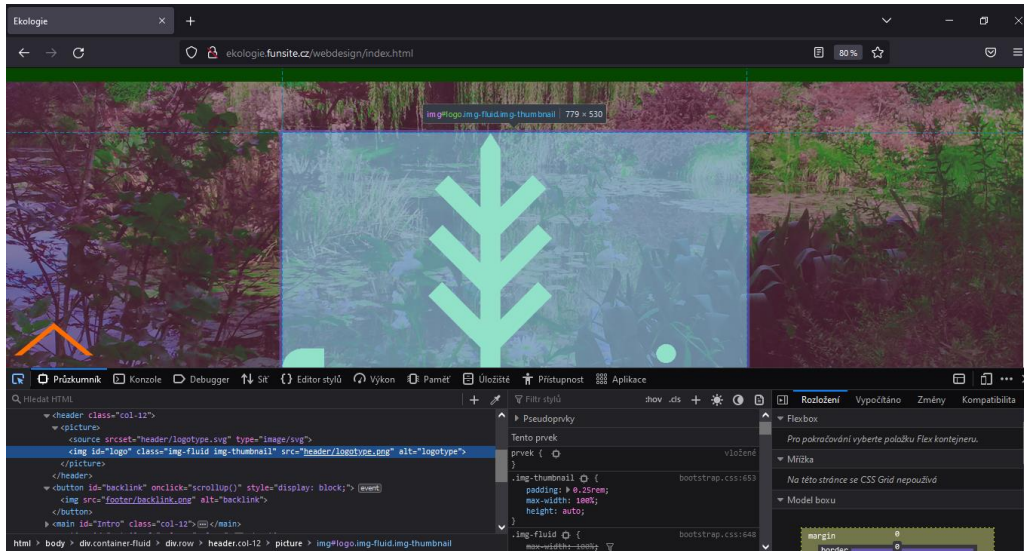


Obr. 23: Nástroje pro webové vývojáře

Máme zde 10 panelů, ale ne všechny se vždy využívají.

4.1. Průzkumník

Panel průzkumník slouží k prohlížení našeho HTML kódu. Pokud klikneme vlevo nahoře v našem panelu na ikonku označenou červeným čtvercem a najedeme na nějaký prvek na našem webu, průzkumník najde v HTML kód tohoto prvku.

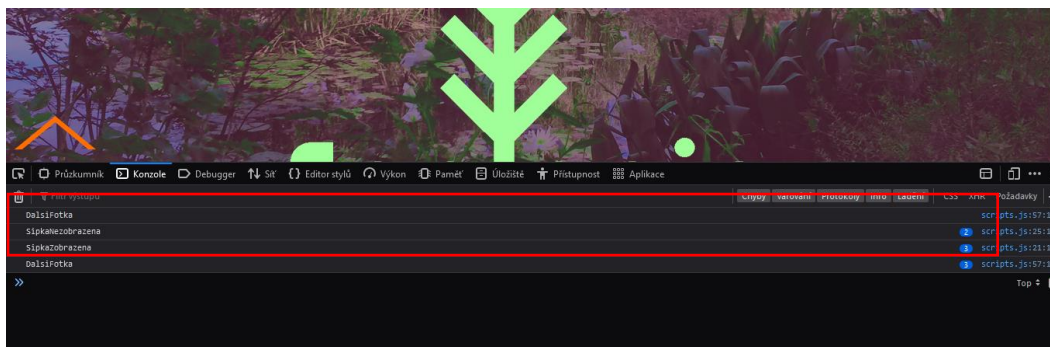


Obr. 24: Výběr prvku na stránce

Vybraný prvek se podbarví vždy modrou barvou. Toto je důležité, pokud chceme zjistit něco o prvcích na stránce.

4.2. Konzole prohlížeče

Konzole nám vždy vypisuje chyby na našem webu. Používá se ale také v Javascriptu. Příkaz `console.log("vypis");` slouží k vypsání textu v konzoli. Na obrázku 25 máme příklad výstupu v konzoli. V našem Javascriptovém kódu si necháváme vypsát „*události*“, které se stanou, když se skript vykoná.

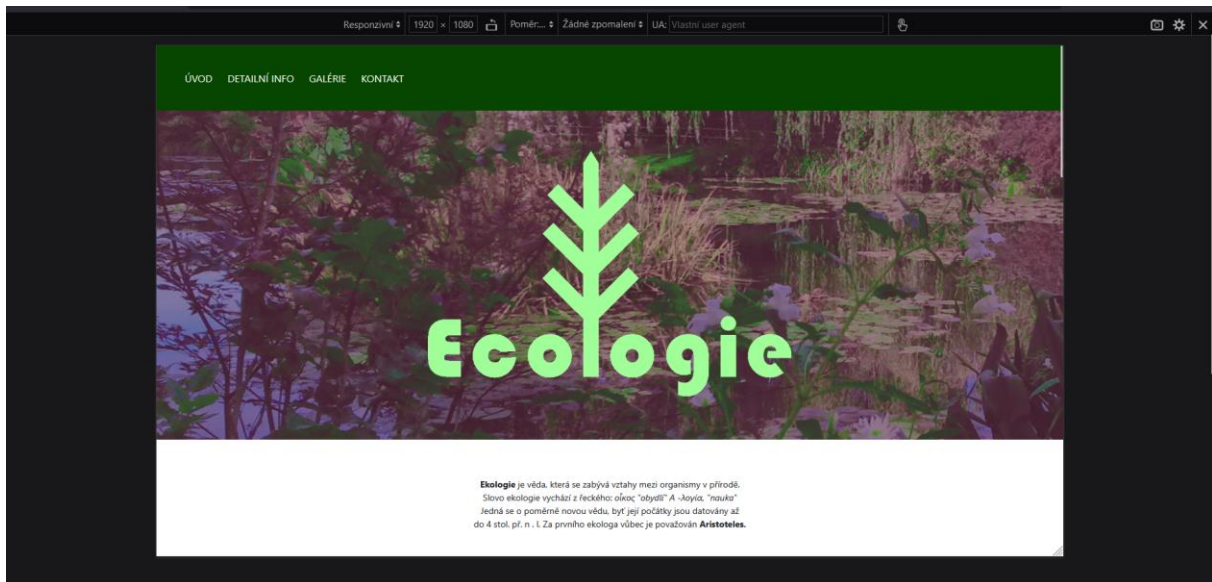


Obr. 25: Výpis javascriptu v konzoli

V našem případě se nám vypisuje „*SipkaZobrazena*“ a „*SipkaNezobrazena*“. První případ se vypíše vždy, když začnu skrolovat. Druhý případ se vypíše v případě, když se vrátím na začátek webové stránky. Dále se nám v konzoli vypisuje „*DalsiFotka*“. Ta se vypíše při změně obrázku v galerii.

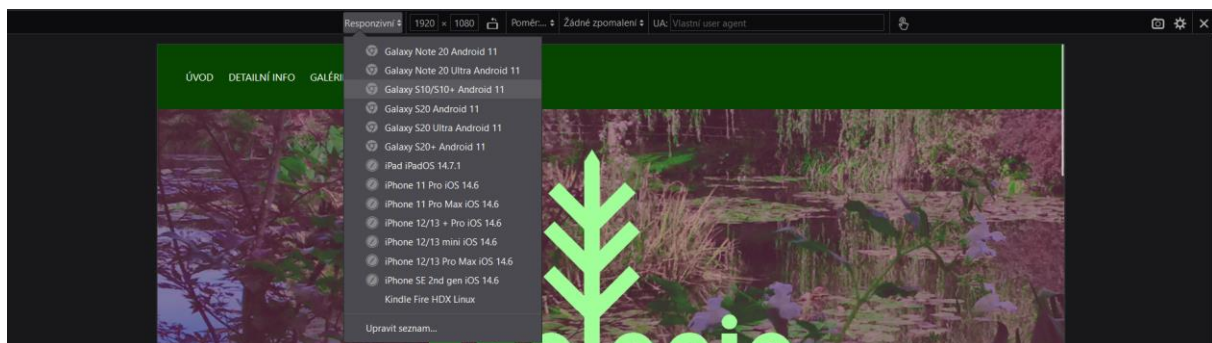
4.3. Responzivní design

Nástroje pro vývojáře umožňují jednu velmi užitečnou věc, a to emulátor responzivního rozvržení. Dostaneme se do něj tak, že klikneme na „*Nástroje pro vývojáře*“ a poté klikneme na položku „*Responzivní design*“. V tu chvíli se nám v našem vyhledávači zobrazí rozhraní, viz Obr. 26.



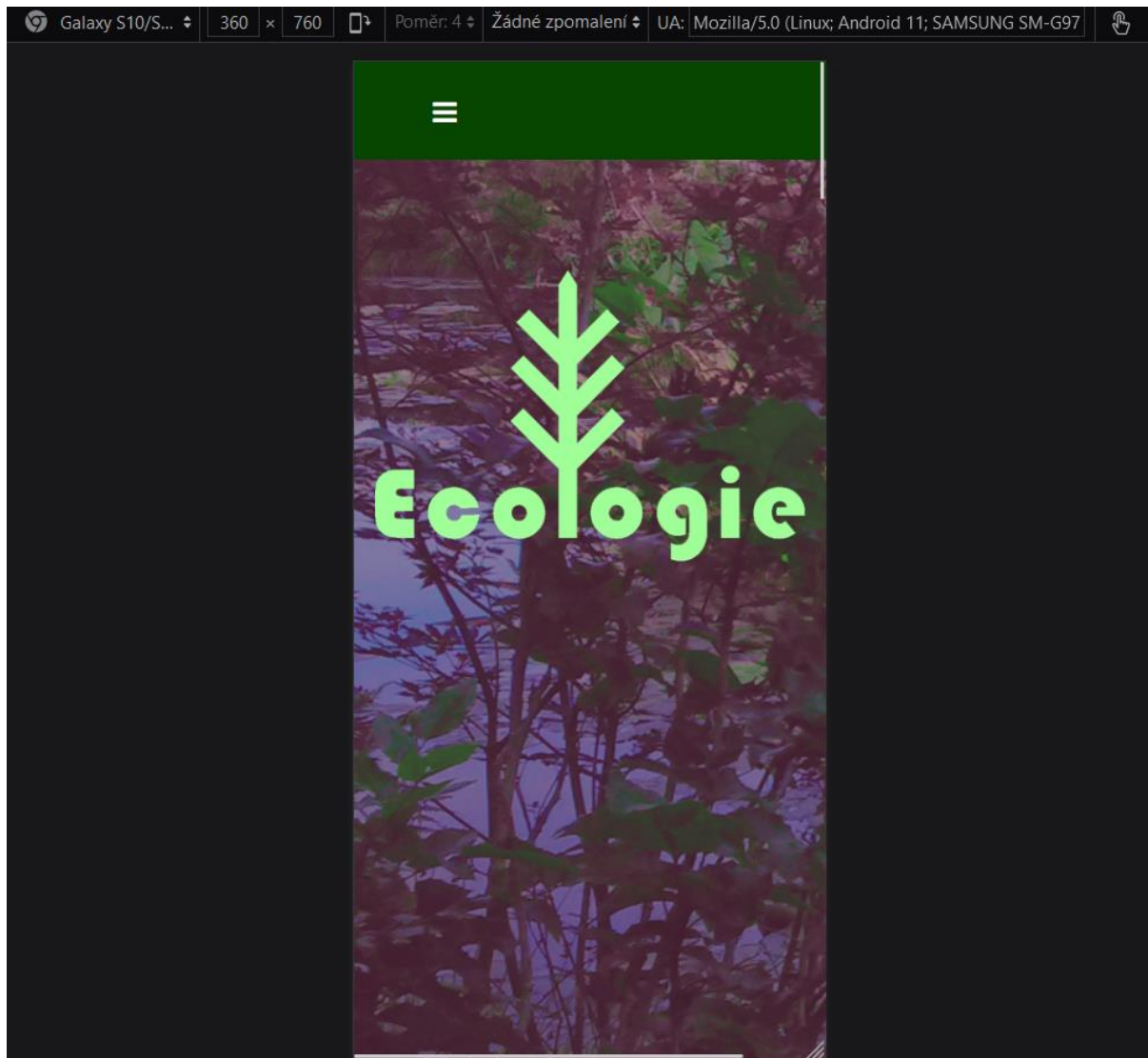
Obr. 26: Rozhraní "Responzivní design"

Rozhraní responzivní design má několik vlastností, které si můžeme přizpůsobit podle našich potřeb. V horní části, kde máme výběrové políčko „responzivní“, si můžeme zvolit jednotlivá rozlišení, na kterých chceme web testovat. Stačí na něj kliknout a můžeme si vybrat zařízení, které chceme simulovat, viz Obr. 27.



Obr. 27: Výběr zařízení

Po kliknutí na zařízení se nám zobrazí emulované zařízení, viz obr. 28.



Obr. 28: Simulace chytrého telefonu

Pokud vybereme zařízení, u kterého předpokládáme, že je dotykové, klikneme na ikonku „*ručičky*“ a můžeme posouvat web, jako by byl na fyzickém dotykovém zařízení. Samozřejmě toto je pouze emulace, takže ne vždy je to přesné jako na fyzickém zařízení. V textových políčkách máme hodnoty rozlišení „*šířka*“ x „*výška*“. Ty můžeme měnit na libovolné hodnoty. Kolonka „*Žádné zpomalení*“ nám emuluje komunikaci mezi klientem a serverem pomocí technologií, např. Wi-Fi, 4G, DSL apod. Toto se hodí v případě, že chceme zjistit funkčnost webu v závislosti na tom, pomocí jaké technologie budou uživatelé web prohlížet.

5 Závěr

Cílem maturitní práce bylo poukázat na použití HTML5, CSS3 a Bootstrap frameworku při tvorbě webových stránek. Záměrem autora bylo poukázat na výhody použití sémantiky HTML5 a důležitost responzivity webových stránek. V kapitole 2 se práce zabývá tím, jak v prototypovacím nástroji Adobe XD vytvořit bootstrap grid. Práce se nezabývá porovnáním starších verzí jazyka s nejnovější verzí, jelikož starší verze není doporučeno používat. Kapitola 3 pojednává o kódování webové stránky. Čtenář se zde dozví, jak vytvořit např. responzivní menu, propojení Bootstrapu s HTML, aplikování sémantických tagů v HTML5. Kapitola 3 obsahuje samozřejmě ukázky kódu, které jsou na webu.

V praktické části byla vytvořena webová stránka, kde HTML5, CSS3 a Javascriptové kódy slouží k demonstraci jednotlivých částí webu. Na těchto konkrétních příkladech autor ukazuje, jak používat jednotlivé tagy. Webová stránka je přístupná pod touto webovou adresou:

[Odkaz na webovou stránku](#). Webová stránka je plně responzivní, aby si ji mohl kdokoliv zobrazit na libovolném zařízení. Výsledkem maturitní práce je příručka nejen pro webové designéry a programátory, ale také pro začátečníky, kteří pronikli do základu tvorby webu a chtěli by si rozšířit znalosti v oblasti tvorby webových stránek.

6 Zdroje

Bibliografie

Baše, O. (2012). *jQuery pro neprogramátory: průvodce využitím knihovny jQuery UI*. (1. vyd). Brno: Computer Press.

Castro, E., & Hyslop, B. (2022). *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. (2. vydání). Brno: Computer Press.

Croft, J., Lloyd, R., & Rubin, D. (2007). *Mistrovství v CSS: pokročilé techniky pro webové designéry a vývojáře*. (Vyd. 1). Brno: Computer Press.

Odell, D. (2010). *JavaScript: průvodce programováním ajaxových aplikací*. (Vyd. 1). Brno: Computer Press.

Pehlivanian, A., & Nguyen, D. (2021). *JavaScript okamžitě*. (2. vydání). Brno: Computer Press.

Webové stránky

MARCOTTE, Ethan. Responsive Web Design [online]. May 25, 2010. Obsah článku [cit. 23. 10. 2022]. Dostupné z: <https://alistapart.com/article/responsive-web-design/>

W3Schools Online Web Tutorials. *W3Schools Online Web Tutorials* [online]. Dostupné z: <https://www.w3schools.com>. Veškerý obsah webu [cit. 23. 10. 2022].

Responzivní webdesign. *itnetwork.cz - Učíme národ IT* [online]. Copyright © 2022 itnetwork.cz. Veškerý obsah webu [cit. 23. 10. 2022]. Dostupné z: <https://www.itnetwork.cz/html-css/responzivni-webdesign>

7 Seznam obrázků

Obr. 1: Artboard v Adobe XD (bílá plocha v červeném obdelníku).....	12
Obr. 2: Hodnoty zadané do políček.....	12
Obr. 3: Výsledný bootstrap grid, po zadání hodnot	12
Obr. 4: Nástroje programu Adobe XD	13
Obr. 5: Karta Prototype	14
Obr. 6: Menu na desktopu	17
Obr. 7: menu na telefonu.....	17
Obr. 8: hlavička webu na desktopu	19
Obr. 9: Hlavička webu na telefonu.....	19
Obr. 10: Příklad jednosloupcového textu	21
Obr. 11: Půlsloupcové odstavce na webu.....	23
Obr. 12: Příklad lightboxu.....	26
Obr. 13: Příklad medailonku	26
Obr. 14: Formulářové prvky v defaultním stavu.....	30
Obr. 15: Formulářové prvky po najetí myši	30
Obr. 16: Detaily o hledaném místě.....	32
Obr. 17: Okno s tagem <iframe> obsahujícím naši mapu.....	32
Obr. 18: Vzhled mapy webu	33
Obr. 19: Patička webu	36
Obr. 20: Vzhled superpatičky.....	36
Obr. 21: Kliknutím na ikonku oka (v červeném obdelníku) zobrazíme náš web v novém panelu	38
Obr. 22: Hamburger menu našeho prohlížeče (v červeném obdelníku).....	38
Obr. 23: Nástroje pro webové vývojáře	38
Obr. 24: Výběr prvku na stránce	39
Obr. 25: Výpis javascriptu v konzoli	39
Obr. 26: Rozhraní "Responzivní design"	40
Obr. 27: Výběr zařízení.....	40
Obr. 28: Simulace chytrého telefonu.....	41

8 Seznam příloh

Zdrojový kód webové stránky

Nástroje použité pro zpracování

- Adobe XD: Návrh grafiky webové stránky.
- Affinity Designer: Návrh logotypu pro web.
- Visual Studio Code: Programování webu v HTML5, CSS3, JS, bootstrap
- MS Word: Tvorba teoretické části maturitní práce.
- MS Excel: Tvorba tabulek a grafů.

Software z výše uvedeného seznamu je řádně zakoupený na oficiálních stránkách výrobce, Microsoft Store, nebo je jeho vlastníkem toho času Vyšší odborná škola, Střední škola, Centrum odborné přípravy, Sezimovo Ústí, Budějovická 421, kde autor toho času studoval.