



Gymnázium, Praha 6, Arabská 14  
předmět Programování, vyučující Mgr. Jan Lána



# Smart website builder

Ročníkový projekt

Rád bych poděkoval Mgr. Jan Lána za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování ročníkové práce. Mé poděkování patří též Ing. Václav Strnad za spolupráci při kontrole ročníkové práce.

# Prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne ..... David Pavlův .....

## Anotace

V této dokumentaci ročníkové práci najdete postupy a řešení, která doprovázela vývoj webové aplikace a rozšíření do prohlížeče určené pro vývoj a usnadnění vytváření webových stránek. Aplikace obsahuje dva editory – vizuální a programovací. Vizuální editor využívá šablony a jednoduché editování. Textový je určený pro programátory. Uživatel může mezi editory přepínat a využívat ten adekvátnější.

Součástí ročníkového projektu je rozšíření pro prohlížeč Firefox. Pomocí něj můžete kopírovat zdrojové kódy z různých stránek a vložením použít u vlastních stránek.

## Annotation

In this year's documentation, you will find procedures and solutions that accompanied the development of the web application and the browser extension designed to develop and facilitate the creation of websites. The application contains two editors - visual and programming. The visual editor uses templates and easy editing. Programming is intended for programmers. The user can switch between editors and use the more adequate one.

Part of the year's project is an extension for the Firefox browser. You can use it to copy source code from different pages and paste it into your own pages.

## Zadání

Webová aplikace na tvorbu jednoduchých stránek. Uživatel bude schopen si z šablon (bloků) postavit stránku a pak jen upravit obsah (podobně jako u Wix.com). Aplikace mu následně vygeneruje potřebný HTML a CSS.

Součástí projektu bude rozšíření pro Firefox, který si uživatel nainstaluje do prohlížeče. Po jeho aktivování si může načíst libovolnou stránku a pomocí rozšíření vybrat přímo blok. Po kliknutí se mu zkopíruje HTML a CSS bloku.

Stěžejní částí práce bude naprogramovat nějaký editor pro tvorbu stránek. Editor bude zrychlovat vývoj stránek, avšak uživatel stejně bude muset umět programovat, aby docílil cíleného výsledku.

Technologie webové aplikace: Javascript, ReactJS, NodeJs, NextJs

Technologie webového rozšíření: Javascript, Firefox

# Obsah

<b>1. Úvod</b>	<b>1</b>
1.1. <i>Cíl</i>	1
<b>2. Technologie</b>	<b>1</b>
2.1. <i>Webová aplikace</i>	1
2.2. <i>Rozšíření na Firefox</i>	2
<b>3. Webová aplikace</b>	<b>2</b>
3.1. <i>Editování v oknech</i>	2
3.1.1. Vizuální editor	2
3.1.2. Textový editor	3
3.2. <i>Sekce</i>	3
3.2.1. Databáze šablon	3
3.2.2. Manipulace sekcí	3
3.3. <i>UI a UX design</i>	4
3.3.1. Postup	4
<b>4. Rozšíření na Firefox</b>	<b>5</b>
4.1. <i>Účel nástroje</i>	5
4.2. <i>Vývoj</i>	5
4.3. <i>Ovládání</i>	6
<b>5. Nasazení</b>	<b>6</b>
5.1. <i>Hosting webové aplikace na Vercel</i>	7
5.2. <i>Rozšíření na Firefox</i>	7
<b>6. Závěr</b>	<b>7</b>
6.1. <i>Možné vylepší</i>	7
6.2. <i>Autorovo zhodnocení</i>	7

# 1. Úvod

V dnešní době je několik možností, jak vytvořit webovou stránku (například blog, informační či komerční stránky apod.). Uživatelé, kteří neumí programovat, mohou sáhnout po nástrojích typu wix.com nebo wordpress.com (patří mezi nejpopulárnější). A vývojáři si mohou vytvořit webovou stránku sami.

Výhody a nevýhody obou přístupů jsou rozličné, avšak zde budou zmíněny ty klíčové.

Tak například použití výše zmíněných nástrojů je časově velice efektivní. Uživatel si může rychle naklikat sekce a změnit obsah dle vlastního uvážení. To vše, aniž by musel ovládat jakýkoliv programovací jazyk. Navíc dostupné šablony jsou designované a naprogramované profesionály, takže jsou responzivní<sup>1</sup> a vzhledově moderní. Tímto se zároveň dostáváme k nevýhodě. Jelikož je mnoho klientů a počet šablon daný, tak se začnou opakovat. Je běžné, že vícero klientů mají totožné šablony a jsou tím pádem neoriginální. To je z hlediska brandingů vlastní značky a společnosti negativní. Internet je šablonami zahlcený.

Naopak druhý přístup, tedy vlastní vývoj, je časově náročnější. I když s rostoucí zkušeností programátora platí pro časovou náročnost nepřímá úměra, tak je stále faktem, že je vlastní vývoj pomalejší. Zároveň je kritické, aby vývojář/i byl schopný jak v kódování, tak i v designování. Nekvalitní weby se ve vysoké konkurenci spíše neprosadí. Výsledkem jsou ale velice vyladěné a optimalizované weby. Jsou sice dražší, ale jen na úkor originality a správné funkčnosti, to se vyplácí.

Tady se dostáváme k hlavní myšlence tohoto projektu. Vytvořit aplikaci, která pomůže přiblížit oba přístupy vytvoření webu.

## 1.1. Cíl

Cílem bylo vytvořit webovou aplikaci, která bude kombinovat oba přístupy. Bude disponovat výhodami z obou přístupů. Toho bude docíleno použitím dvou editorů. Vizuálního (používáním šablon) a textového (programováním) editoru. Uživatel může mezi nimi kdykoliv přepínat a používat ten, který se více hodí. Zároveň, bude vše zachováno jednoduché a přehledné.

Součástí ročníkového projektu je rozšíření pro prohlížeč Firefox. Pomocí něj můžete kopírovat zdrojové kódy bloků z různých stránek a tím je jednoduchým vložením použít u vlastních stránek.

# 2. Technologie

V této podkapitole jsou popsány použité technologie a programovací jazyky.

## 2.1. Webová aplikace

Webová aplikace je vyvinuta v Javascriptu za použití frameworku NextJs. NextJs umožňuje vykreslování na straně serveru a generování statických webů pro webové aplikace založené na ReactJS. (1)

---

<sup>1</sup> Jedná se o způsob stylování HTML dokumentu, které zaručí, že zobrazení stránky bude optimalizováno pro všechny druhy nejrůznějších zařízení (mobily, notebooky, tablety atd.). (11)

Dále se používá HTML<sup>2</sup> a CSS<sup>3</sup> jak pro vývoj editoru, tak v dostupných šablonách.

## 2.2. Rozšíření na Firefox

Rozšíření je napsané také v Javascriptu v souladu s dokumentací pro vývoj rozšíření na Firefox, kterou naleznete na oficiálních stránkách Firefox<sup>4</sup>.

Dále se zde používá JSON<sup>5</sup> hlavně pro definování práv, popisu, načítání skriptů a verzování rozšíření. (2)

## 3. Webová aplikace

V této kapitole je popsán vývoj stěžejních částí projektu. Zabývá se hlavními částmi aplikace. Jednotlivé podkapitoly pak obsahují vysvětlení a řešení.

### 3.1. Editování v oknech

Oba editory, vizuální a textový, fungují nezávisle. Uživatel mezi nimi může přepínat pomocí horní záložky hned nad editorem (vždy je otevřen jeden z nich). Právě přepínání je okamžik, kdy se z jednoho editoru převádí změněný dokument do druhého. Čili uživatel může upravovat dokument, jak chce, ale ve chvíli, kdy se rozhodne přepnout editor, tak se přesunou změny do nově zvoleného editoru. Předchozím přístupem byla snaha vytvořit jednotný úložný prostor pro oba editory zároveň. V tom případě by se musela hlídat každá změna v dokumentu a ihned ukládat, ale to se ukázalo jako optimalizačně nešetrné, protože uživatel může psát například delší část kódu/textu (v závislosti na aktuálním editoru) a každý úder do klávesy by postupně vedl k velkému množství požadavků. Tento problém mnoha požadavků se by se dal vyřešit metodou debounce<sup>6</sup> (dokument se uloží až po posledním úderu – uživatel dopíše), ale zde přetrvává odezva, takže ukládání jen při přepínání editorů se ukázalo jako lepší řešení jak optimalizačně, tak i v jednoduchosti.

Oba editory jsou stejně velké a jsou na stejném místě, takže jeden překrývá druhý. Pomocí jednoduchého skriptu bylo docíleno přepínání editorů, tak že po kliknutí na danou záložku se nastaví korespondujícímu editoru vyšší z-index, takže se dostane nahoru a překryje druhý. Tento přechod je doprovázen animací.

#### 3.1.1. Vizuální editor

Vizuální editor zobrazuje vizuální stav stránky. Jinými slovy v něm uživatel vidí aktuální stav jeho práce na webové stránce. Toho je docíleno vytvořením Iframe elementu a následnou úpravou jeho obsahu.

---

<sup>2</sup> Hypertext Markup Language (zkratka HTML) je jazyk používaný pro tvorbu webových stránek. (12)

<sup>3</sup> Cascading Style Sheets (zkratka CSS) je jazyk pro popis způsobu zobrazení elementů na stránkách HTML.

<sup>4</sup> Oficiální stránky: [https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Your\\_first\\_WebExtension](https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Your_first_WebExtension)

<sup>5</sup> JavaScript Object Notation (zkratka JSON) je způsob zápisu dat nezávislý na počítačové platformě. (13)

<sup>6</sup> Odposlouchávání změn s určitým časovým intervalem a provedení dané operace po uplynutí intervalu. (14)

Navíc může uživatel do prostoru klikat a upravovat jeho obsah pouhým psaním i vkládáním zkopírovaných obrázků. To je umožněno pomocí nastavení atributu body elementu v dokumentu. Konkrétně atributu `contenteditable` na `true`. To dává pohodlnou a rychlou možnost k úpravám. Důležité je tento atribut odstranit na konci, aby jej neobsahovala výsledná stránka. (3)

Manipulace s dokumentem je po získání reference na element velice přímočará. Obsah můžeme zadávat i získávat v `iframe` elementu. Reference je JSON objekt, který obsahuje `contentDocument` objekt. V něm je uložena celá struktura dokumentu, která je potřeba pro ukládání i nastavování obsahu. (4)

### 3.1.2. Textový editor

Textový editor zobrazuje zdrojový kód dokumentu. V podstatě jde o klasický input pro editování textu (kódu). Uživatel v něm může programovat a není ničím limitován.

Jelikož je v klasickém inputu těžké programovat z důvodu nepřehlednosti (žádné formátování, bez zvýraznění klíčových slov apod.), bylo potřeba implementovat příjemnější editor. Vhodným řešením bylo použití editoru s formátováním – Ace<sup>7</sup>. Ace je populární editor pro Javascript (ale také pro HTML a CSS). Editor byl navíc stylován, tak aby splňoval designérské požadavky a vyhovoval celému vzhledu webové aplikace.

## 3.2. Sekce

Jednotlivé sekce, které uživatel může vkládat pomocí kliknutím na znak `+`, se nachází v levé části webové aplikace. Tato část je příjemně stylovaná a je doprovázena animacemi. Implementovaný je seznam šablon, který vychází z levé strany. V akordeonu si může zvolit nejprve kategorii, poté konkrétní sekci. Po kliknutí se ihned zobrazí ve vizuálním editoru a může pokračovat v editování.

### 3.2.1. Databáze šablon

Velký kredit patří týmu `mdbootstrap.com`, poskytují velký výběr šablon, založených na `bootstrap.com`, a to dokonce i pro komerční použití úplně zdarma. Šablony jsou responzivní a příjemně stylované. K správnému fungování bylo potřeba nainportovat některé skripty a styly. V projektu byl použit CDN<sup>8</sup> v souladu s dokumentací `mdbootstrap.com`. (5)

Dostatečný počet šablon byl uložen do složky `templates` v kořenovém adresáři projektu. V této složce jsou jednotlivé moduly reprezentující kategorii šablon. Moduly exportují tyto šablony k použití rovnou i s odkazem náhledového obrázku, který je použit pro reprezentaci dané šablony. Uživatel to má tímto vybírání vhodné šablony mnohem snazší.

Pokud uživatel použije některou sekci z šablon, není vůbec omezen jejím základním zdrojovým kódem. Může použít vizuální nebo textový editor a dopravit si šablonu k tíženému výsledku.

### 3.2.2. Manipulace sekcí

Uživatel se sekcemi může manipulovat několika způsoby. Po zvolení nějaké sekce s ní uživatel může pohybovat nahoru/dolu ve vazbě s dalšími sekcemi. Dále může být sekce úplně odstraněna nebo přidána další.

---

<sup>7</sup> Oficiální webová stránka: <https://ace.c9.io>

<sup>8</sup> Content Delivery Network (zkratka CDN) je síť vzájemně propojených počítačů skrze internet, která umožňuje dostupnost obsahu nebo dat (obvykle velký multimediální obsah) uživatelům. (15)



Pokud se sekci uživatel pohybuje nahoru/dolu, její aktuální pozice se zobrazuje v levé části aplikace. V ní má uživatel, kromě struktury, také nástroje k manipulaci – jsou reprezentovány ikonami. Každá sekce má identifikační číslo, aby nedošlo k různým chybám. Při pohybu sekcemi bylo potřeba ošetřit krajní body, kdy je již sekce úplně nahoře či dole. Uživatel dále může přidávat nové sekce přímo do zvolených pozic (i mezi jiné sekce).

Přidávání, mazání i pohyb se sekcí má okamžitou účinnost. Změna se projeví automaticky a je k dispozici ihned ke kontrole v editoru.

### 3.3. UI a UX design

V této podkapitole jsou popsány řešení a rozvržení aplikace pro pohodlnou práci s editorem společně s profesionálním a moderním designem. Design by měl být přehledný a ovladatelnost programu jednoduchá.

#### 3.3.1. Postup

Nejdříve bylo potřeba zvolit paletu barev. Důležité je aby 2 základní barvy byly kontrastní, takže jejich použití v blízkosti vede k dobře čitelnosti jak textu, tak grafických objektů. Na obrázku 1. můžete vidět paletu těchto 2 barev. Zbytek použitých barev se odvíjí od jejich odstínu. Pro výběr barev byl použit nástroj pro generování ověřených barev Colors<sup>9</sup>. Jedná se o populární databázi palet a barev.

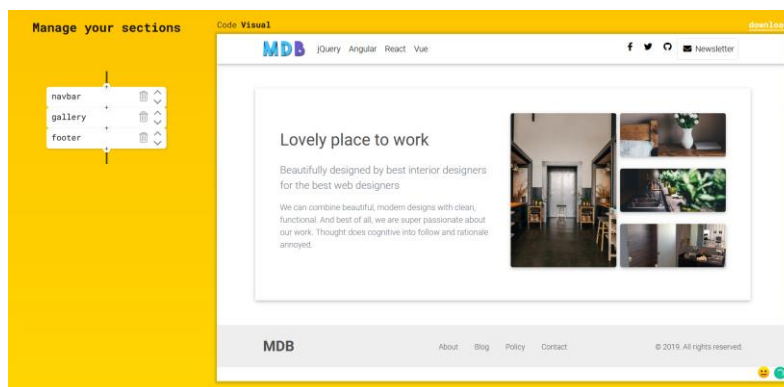


Obrázek 1- paleta kontrastních barev

Pro vytvoření návrhu je dobré použít nějaký grafický editor. Pro tento projekt byla použita Figma. Figma<sup>10</sup> je editor vektorové grafiky a prototypový nástroj. Na obrázku 2 je vidět výsledek. V levé části je prostor jen pro editování sekcí a jejich manipulování. V pravé části můžete vidět nahoře nejprve hlavičku obsahující záložky 2 editorů (Code a Visual) a tlačítko pro stažení webové stránky (download) a poté velký prostor pro okno editoru samotného (v případě výstřižku: vizuální editor)

<sup>9</sup> Oficiální webová stránka: <https://colors.co>

<sup>10</sup> Oficiální webová stránka: <https://www.figma.com>



Obrázek 2- výstřih aplikace

Ke stylování a animování aplikace byl použit Sass<sup>11</sup>. Co se týče kódování je totožný s CSS, ale má výhodu skládání podobných stylů a kratší selektory, takže vývoj v něm je rychlejší, a hlavně méně namáhavý.

## 4. Rozšíření na Firefox

### 4.1. Účel nástroje

Rozšíření je nástroj, jehož účelem je urychlit vývoj webových aplikací. Po jeho instalaci do prohlížeče na Firefox může uživatel ze stránek, které má zrovna načtené, pouhým kliknutím na element stránky zkopírovat jeho zdrojový kód – HTML a CSS. Poté může tento blok vložením použít do svého projektu.

Toto rozšíření je zcela nezávislé na vytvořeném editoru, a může být tedy používáno pro jakékoliv účely.

### 4.2. Vývoj

Vývoj aplikace byl směřován na Firefox. Bylo tedy nutné postupovat správně dle dokumentace rozšíření pro Firefox. Prvním krokem bylo vytvoření souboru manifest, ve kterém se rozšíření definuje. V něm bylo potřeba, krom názvu a verze, vytvořit odkazy na skripty, reference na logo (vytvořeno pomocí dříve zmíněného nástroje Figma), požadavky a práva pro spuštění. (2)

Po aktivování a spuštění rozšíření celá plocha stránky ztmavne. To je indikace proto, aby uživatel začal vybírat element. Když uživatel začne myší projíždět začnou se elementy ohraničovat, takže uživatel přesně ví, jaká část stránky se mu zkopíruje. Zde bylo potřeba vytvořit funkci pro selektování a vytváření ohraničení v reálném čase. Ve webových stránkách jsou elementy vnořené do dalších atd., takže pokud uživatel míří na nějaký element, tak ve skutečnosti míří také na všechny jeho rodiče. Bylo potřeba vybrat ten nejnižší v hierarchii HTML struktury.

Během vybírání se vytváří ohraničení pro daný element. Pokud se přejede myší na různý další element tak se ohraničení smaže a vytvoří u nového elementu. Z toho důvodu bylo nutné porovnávat elementy (zda se jedná o ten samý či o jiný po projetí myší). HTML elementy jsou objekty a nepatří mezi primitivní datový typ. To znamená, že pokud porovnáваме objekty porovnáваме jejich referenci v paměti, nikoliv jejich hodnoty jako u primitivních datových typech.

<sup>11</sup> Sass je preprocesorový skriptovací jazyk, který je interpretován do kaskádových stylů. (16)

Takže pro posouzení, zda se jedná o jiný element posloužila jeho identifikační vlastnost, kterou byla jeho přesná pozice v dokumentu.

Po kliknutí na element se vybírání ukončí a přejde se k dalšímu kroku. Tím je získání potřebných zdrojových kódů a následné zkopírování.

Vnitřní HTML struktura elementu je uložena v jeho referenci (atribut innerHTML) (6). CSS vlastnosti, se získají výčtem a filtrováním stylůpisů dokumentu. V tomto kroku může nastat CORS<sup>12</sup> výjimka. A to právě tehdy pokud je stylůpis načítán externě z jiné domény. (7)

Po získání zdrojových kódů se automaticky zkopírují. To se pomocí skriptu dělá vytvořením divu vložením textu do něj, poté se div vloží do dokumentu. Pak se daný text označí, a až poté je možné jej zkopírovat. Nakonec se div z dokumentu odstraní. To se udělá zcela automaticky a velice rychle.

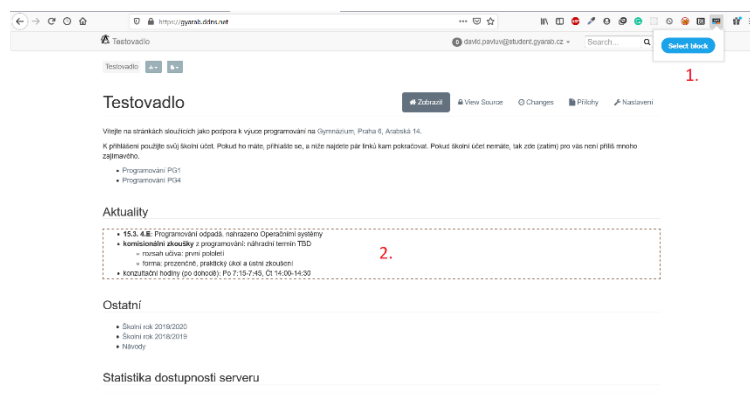
Během jakéhokoliv kroku může potenciálně dojít k výjimce, takže to, co není ošetřeno je stejně zachyceno a po ukončení funkce je uživateli vypsaná zpráva o úspěšnosti všech kroků.

### 4.3. Ovládání

Po instalaci rozšíření se ikonka nástroje uloží do pravé části v panelu prohlížeče. Na obrázku 3 lze vidět postup při kopírování bloku.

Krok 1 – nejdříve rozklikneme rozšíření a stiskneme modré tlačítko. Po kliknutí se ihned spustí režim vybírání bloků. Zapnutý režim se také projeví ztmavením stránky.

Krok 2. – poté stačí kliknout na jakýkoliv blok stránky. Pouhým přejížděním se jednotlivé elementy stránky ohraničují rámečkem. Po kliknutí se zobrazí na stránce textová zpráva o úspěšnosti zkopírování.



Obrázek 3- ovládání rozšíření

Krok 3. – po zkopírování můžeme vložit zdrojový kód elementu do aplikace. V aplikaci si přepneme režim na „Code“ a vložíme na vhodné místo v kódu. Zkopírovaný kód můžeme klidně použít i mimo aplikaci.

## 5. Nasazení

Webová aplikace i rozšíření pro Firefox bylo potřeba nasadit na produkci, kde můžou být pro vzdělávací i komerční účely používány zcela zdarma.

<sup>12</sup> Cross-origin resource sharing (zkratka CORS) je mechanismus umožňující sdílení zdrojů. (17)

## 5.1. Hosting webové aplikace na Vercel

Pro hosting<sup>13</sup> webové aplikace byla zvolena cloudová platforma Vercel. Po založení účtu jej stačí propojit s GitHub účtem a vybrat konkrétní repozitář k nasazení. Jakoukoliv změnu na master větvi si Vercel hlídá a nasazuje změny automatizovaně. Výhodou (srovnáváme balíčky, které jsou zdarma) oproti například Heroku je neuspávání serveru po 30 minutové neaktivitě. Pozdější načítání aplikace je ihned k dispozici a nemusí se znova načítat jako u Heroku. (8) (9)

Webová aplikace je úspěšně nasazena na adrese: <https://smartwebsitebuilder.vercel.app>

## 5.2. Rozšíření na Firefox

Nahrání rozšíření bylo provedeno na stránkách Firefox pro rozšíření. Je zdarma a kdykoliv k dispozici ke stažení přímo do prohlížeče Firefox. Předem bylo potřeba popsat funkcionalitu a kategorii rozšíření. Dále byly poslány zdrojové kódy rozšíření na kontrolu a po schválení bylo rozšíření udělena licence. (10)

Rozšíření na Firefox je k dispozici na adrese: <https://addons.mozilla.org/cs/firefox/addon/block-copier>

## 6. Závěr

Tato kapitola pojednává o možném vylepšení ročníkového projektu. V závěru je autorovo vlastní zhodnocení ročníkového projektu a jeho splnění.

### 6.1. Možné vylepší

Momentálně je v aplikaci možnost stažení vytvořené webové stránky, takže uživatel může dále nahrát webovou stránku na hostingovou službu dle svého vlastního výběru. V budoucnosti, při navýšení hostingového výkonu, by bylo možné implementovat možnost automatického nasazení vytvořeného webu a vygenerování domény.

### 6.2. Autorovo zhodnocení

Aplikace byla úspěšně naprogramovaná a nasazena. Stejně jako rozšíření. Obě části splňují své účely dle zadání.

Samotné programování a plánování funkcí, designování mi přineslo mnoho zkušeností a dobrých znalostí do budoucna. Bylo přínosné dotáhnout projekt do konce i přes nečekané potíže. Jsou části, které bych nyní udělal trochu jinak, ale celkově jsem velice spokojený.

---

<sup>13</sup> Hosting je pronájem prostoru pro webové stránky na cizím serveru. (18)

## Seznam obrázků

Obrázek 1- paleta kontrastních barev.....	4
Obrázek 2- výstřížek aplikace.....	5
Obrázek 3- ovládání rozšíření.....	6

## Bibliografie

1. Next.js - Wikipedia. [online]. Dostupné z: <https://en.wikipedia.org/wiki/Next.js>. [Online]
2. Your first extension - Mozilla | MDN. [online]. Copyright © 2005 [cit. 09.01.2021]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Your\\_first\\_WebExtension](https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Your_first_WebExtension). [Online]
3. HTML iframe tag. W3Schools Online Web Tutorials [online]. Dostupné z: [https://www.w3schools.com/tags/tag\\_iframe.ASP](https://www.w3schools.com/tags/tag_iframe.ASP). [Online]
4. HTML DOM IFrame contentDocument Property. W3Schools Online Web Tutorials [online]. Dostupné z: [https://www.w3schools.com/jsref/prop\\_frame\\_contentdocument.asp](https://www.w3schools.com/jsref/prop_frame_contentdocument.asp). [Online]
5. Material Design for Bootstrap 5 & 4 - Material Design for Bootstrap. Material Design for Bootstrap 5 & 4 - Material Design for Bootstrap [online]. Copyright © 2021 [cit. 10.01.2021]. Dostupné z: <https://mdbootstrap.com/>. [Online]
6. JavaScript and HTML DOM Reference. W3Schools Online Web Tutorials [online]. Dostupné z: <https://www.w3schools.com/jsref/>. [Online]
7. jquery - Accessing cross-domain style sheet with .cssRules - Stack Overflow. Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/questions/3211536/accessing-cross-domain-style-sheet-with-cssrules>. [Online]
8. Introduction to Vercel - Vercel Documentation. Develop. Preview. Ship. For the best frontend teams – Vercel [online]. Copyright © [cit. 12.01.2021]. Dostupné z: <https://vercel.com/docs>. [Online]
9. Documentation | Heroku Dev Center. Heroku Dev Center [online]. Copyright © [cit. 12.01.2021]. Dostupné z: <https://devcenter.heroku.com/categories/reference>. [Online]
10. Submitting an add-on | Firefox Extension Workshop. Firefox Extension Workshop. [online]. Copyright ©1998 [cit. 12.01.2021]. Dostupné z: <https://extensionworkshop.com/documentation/publish/submitting-an-add-on/>. [Online]
11. Responzivní web design – Wikipedie. [online]. Dostupné z: [https://cs.wikipedia.org/wiki/Responzivn%C3%AD\\_web\\_design](https://cs.wikipedia.org/wiki/Responzivn%C3%AD_web_design). [Online]
12. Hypertext Markup Language – Wikipedie. [online]. Dostupné z: [https://cs.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](https://cs.wikipedia.org/wiki/Hypertext_Markup_Language). [Online]
13. JavaScript Object Notation – Wikipedie. [online]. Dostupné z: [https://cs.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://cs.wikipedia.org/wiki/JavaScript_Object_Notation). [Online]

14. JavaScript Debounce Function. David Walsh Blog - JavaScript Consultant [online]. Copyright © David Walsh 2007 [cit. 10.01.2021]. Dostupné z: <https://davidwalsh.name/javascript-debounce-function>. [Online]
15. Content delivery network – Wikipedie. [online]. Dostupné z: [https://cs.wikipedia.org/wiki/Content\\_delivery\\_network](https://cs.wikipedia.org/wiki/Content_delivery_network). [Online]
16. Sass (stylesheet language) - Wikipedia. [online]. Dostupné z: [https://en.wikipedia.org/wiki/Sass\\_\(stylesheet\\_language\)](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)). [Online]
17. CORS – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/CORS>. [Online]
18. Webhosting – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/Webhosting>. [Online]