

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2025

Filip Hnát

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Generátor barevných schémat webových aplikací
Bakalářská práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Filip Hnát**
Osobní číslo: **I22097**
Studijní program: **B0688A140009 Informační technologie**
Téma práce: **Generátor barevných schémat webových aplikací**
Zadávací katedra: **Katedra informačních technologií**

Zásady pro vypracování

Cílem této bakalářské práce je navrhnout a implementovat generátor barevných schémat, která budou sloužit pro tvorbu grafického systému webových aplikací. Práce se zaměří na vytvoření nástroje, který umožní uživatelům snadno a efektivně generovat harmonická barevná schémata, která budou esteticky příjemná a funkční.

V teoretické části bakalářské práce budou představeny základy teorie barev a principy tvorby barevných schémat. Dále bude provedena rešerše existujících nástrojů a metod pro generování barevných schémat, které jsou dostupné na trhu.

V aplikační části práce bude proveden návrh a následná implementace systému pro generování barevných schémat s ohledem na uživatelskou přívětivost a flexibilitu. K vytvořenému systému budou zpracovány metodické pokyny pro uživatele systému.

Rozsah pracovní zprávy: **30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

DANNHOFEROVÁ, Jana. Velká kniha barev: kompletní průvodce pro grafiky, fotografy a designéry. Brno: Computer Press, 2012. ISBN 978-80-251-3785-7.
FLANAGAN, David. JavaScript: the definitive guide. Seventh edition. Sebastopol: O'Reilly, 2020. ISBN 1491952024.
ŽÁRA, Ondřej. JavaScript: programátorské techniky a webové technologie. Brno: Computer Press, 2015. ISBN 9788025145739.
BEASLEY, Michael. Practical web analytics for user experience: how analytics can help you understand your users. Amsterdam: Morgan Kaufmann, an imprint of Elsevier, 2013.
ANISOVÁ, Hana a MÜLLER, Miroslav. UML srozumitelně. 2. aktualiz. vyd. Brno: Computer Press, 2007. ISBN 8025110834.
SOMMERVILLE, Ian. Softwarové inženýrství. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.
DUCKETT, Jon; STONE, Emme a ULLMAN, Chris. PHP & MySQL: server-side web development. Hoboken, New Jersey: Wiley, [2022]. ISBN 1119149223.

Vedoucí bakalářské práce: **Ing. Lukáš Čegan, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. prosince 2024**
Termín odevzdání bakalářské práce: **16. května 2025**

prof. Ing. Petr Doležel, Ph.D. v.r.
děkan

L.S.

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2025

Prohlašuji:

Práci s názvem Generátor barevných schémat webových aplikací jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 12. 5. 2025

Filip Hnát v.r.

PODĚKOVÁNÍ

Rád bych zde poděkoval svému vedoucímu Ing. Lukáši Čeganovi, Ph.D. za odborné vedení, trpělivost a věnovaný čas při tvorbě této práce.

ANOTACE

Tato bakalářská práce se zabývá návrhem a implementací generátoru barevných schémat pro vývoj webových aplikací. V rámci práce byla vytvořena webová aplikace využívající technologie HTML, CSS a JavaScript. Aplikace slouží k jednoduché a efektivní tvorbě harmonických a lehce použitelných barevných schémat.

KLÍČOVÁ SLOVA

barvy, CSS, návrhové tokeny, soulad barev, webová aplikace, webová přístupnost

TITLE

Color schemes generator for web applications

ANNOTATION

This bachelor's thesis focuses on the design and implementation of a color scheme generator for web application development. As part of the thesis, a web application was created using HTML, CSS, and JavaScript technologies. The application is intended for the simple and efficient creation of harmonious and easily usable color schemes.

KEYWORDS

colors, CSS, design tokens, color harmony, web application, web accessibility

OBSAH

SEZNAM ILUSTRACÍ A TABULEK.....	9
SEZNAM ZKRATEK A ZNAČEK	10
ÚVOD.....	11
1 BARVA A JEJÍ ÚČEL VE WEBOVÝCH APLIKACÍCH.....	12
1.1 Barvy.....	12
1.2 Vliv barev na člověka	13
1.3 Barevné modely	14
1.4 Soulad barev	16
1.5 Návrhové tokeny.....	18
1.6 Barvy v CSS.....	20
1.7 Práce s barvou s ohledem na přístupnost webu	22
1.8 Přehled aplikací pro práci s barevnými schématy	22
2 NÁVRH WEBOVÉ APLIKACE	24
2.1 Účel a požadavky na aplikaci	24
2.2 Analýza požadavků na aplikaci	31
2.3 Architektura	31
2.4 Návrh uživatelského rozhraní aplikace	33
3 IMPLEMENTACE WEBOVÉ APLIKACE	37
3.1 Návrhové tokeny.....	37
3.2 Generátor barevných schémat.....	39
3.2.1 Script.js	39
3.3 PWA.....	47
3.3.1 Service-worker.js	47
3.3.2 Manifest.json.....	48
4 UŽIVATELSKÁ PŘÍRUČKA	49
ZÁVĚR	53
POUŽITÁ LITERATURA	54
SEZNAM OBRÁZKŮ.....	56

SEZNAM PŘÍLOH.....	57
--------------------	----

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1: Barevné spektrum.....	12
Obrázek 2: Model HSL	15
Obrázek 3: Modely RGB a CMYK	16
Obrázek 4: Soulad barev	17
Obrázek 5: Soulad barev	17
Obrázek 6: Struktura návrhového tokenu od Adobe	18
Obrázek 7: Návrhový token v Material Design 3	19
Obrázek 8: Design tokens v Carbon Design System	20
Obrázek 9: Případy užití	25
Obrázek 10: Struktura aplikace.....	32
Obrázek 11: Rozhraní aplikace.....	33
Obrázek 12: Generátor palety	34
Obrázek 13: Generátor barevných schémat	35
Obrázek 14: Tabulka kontrastů.....	36
Obrázek 15: Aplikace	49
Obrázek 16: Generátor palet	49
Obrázek 17: Generátor barevných schémat	50
Obrázek 18: Špatné schéma	51
Obrázek 19: Špatné kontrasty	51
Obrázek 20: Správné schéma.....	52
Obrázek 21: Správné kontrasty.....	52

SEZNAM ZKRATEK A ZNAČEK

nm – nanometr

CSS - Cascading Style Sheets

W3C - World Wide Web Consortium

WCAG - Web Content Accessibility Guidelines

ADA - The Americans with Disabilities Act

PDF - Portable Document Format

SASS - Syntactically Awesome Style Sheets

XML - Extensible Markup Language

PWA - Progressive Web App

JSON - JavaScript Object Notation

SPA - Single Page Application

HTML - Hypertext Markup Language

UI – User interface (uživatelské rozhraní)

ÚVOD

V dnešní době je stále větším trendem digitalizace, tvořit vlastní weby a aplikace. Důležitou součástí všech těchto aplikací je design. Design, který upoutá, zpříjemní zážitek a vyvolá důvěru. Jednou ze stěžejních částí designu je volba barev. Jedná se o velmi těžkou volbu, se kterou nám pomáhají grafici. Z tohoto důvodu se bude tato práce zabývat fungováním barev na obrazovkách a jak pomocí aplikace usnadnit výběr barev, a tím také zjednodušit práci vývojářům a grafikům.

V první kapitole je nastíněno, jak barvy fungují pro účel webových aplikací, jak lze s barvami pracovat na počítačích a jak s nimi pracují různé aplikace. Jsou zde popsány návrhové tokeny, které usnadňují spolupráci mezi grafikem a vývojářem, standardy pro práci s barvami s ohledem na přístupnost a aplikacemi, které již můžeme najít na trhu řešící stejnou nebo podobnou problematiku.

V druhé kapitole je vytvořen návrh aplikace, který by danou problematiku řešil. Jsou popsány požadavky na aplikaci, případy užití aplikace, analýza daných požadavků, architektura aplikace a popsání grafický návrh.

Ve třetí kapitole jsou popsány důležité části implementace navržené aplikace pro generování barevných schémat.

Čtvrtou kapitolou je uživatelská příručka k vytvořené aplikaci. Je zde popsáno, jak s vytvořenou aplikací správně pracovat a co jednotlivé části aplikace dělají.

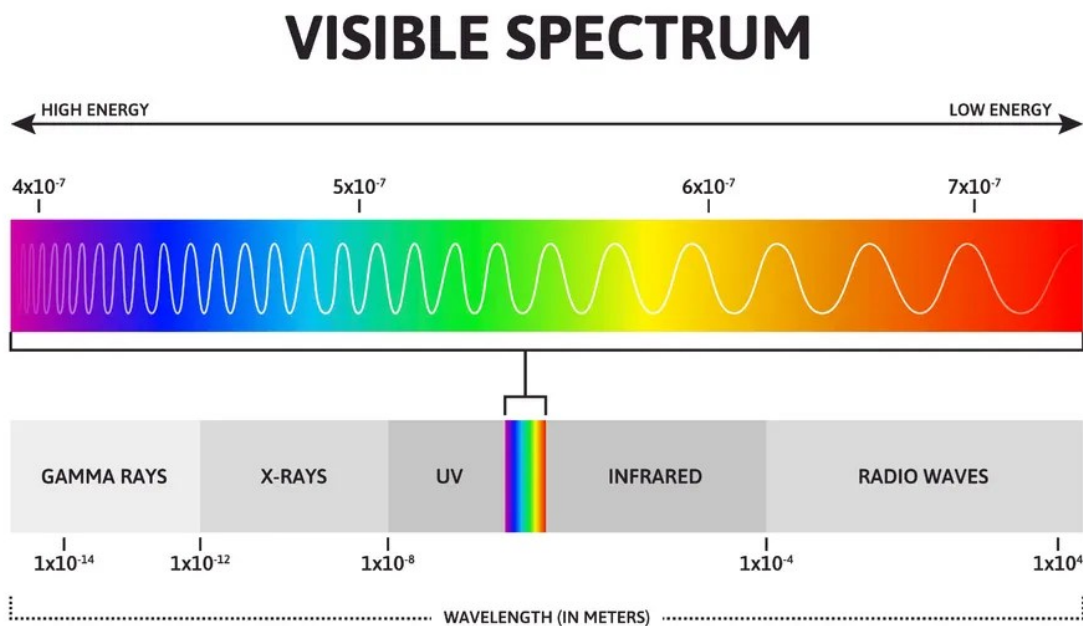
Cílem této práce je vytvořit nástroj, který usnadní práci s barvami v rámci vývoje webových aplikací. Dalším cílem je zlepšit technologické propojení mezi grafikem a vývojářem.

1 BARVA A JEJÍ ÚČEL VE WEBOVÝCH APLIKACÍCH

V této části se na barvy a barevná schémata podíváme z pohledu fyziky, designéra a počítače.

1.1 Barvy

Co je to vlastně barva? Pod slovem barva, si jistě každý dokáže představit něco jiného. Barvy se učíme rozlišovat už od narození. Barvu, v našem případě, musíme chápat jako světlo, což je elektromagnetické záření, různých vlnových délek. Světlo je elektromagnetické záření, které tvoří spektrum. Proto abychom mohli řádně vnímat barvy, je nutností mít dostatek světla. Člověk zrakem vnímá až elektromagnetické záření v rozmezí vlnových délek 700 nm až 400 nm. Uvádí se, že 80 až 90 % informací získává člověk ze svého okolí právě pomocí očí, které jsou následně zpracovávány v mozku.[1]



Obrázek 1: Barevné spektrum

„Vnímání barev ovlivňuje celá řada faktorů. To je důvod, proč různé vědní disciplíny nahlíží na tento proces zcela odlišným způsobem. Důsledkem je mnohdy nejednotná terminologie, ale také obtížné vymezení samotného pojmu barva, který nelze definovat bez bližší charakteristiky.“
[2, s. 13]

Člověk barvy vnímá a samozřejmě barvy na něj také působí. Existuje fyzikální, fyziologické, psychologické a vizuální působení barev.

„Fyzikální působení barev – spočívá ve viditelné části elektromagnetického spektra, které zahrnuje různé druhy světelných paprsků, ale také například v různé míře odrazivosti ploch (například světlé plochy světlo více odrážejí a tmavé plochy jej více pohlcují).

Fyziologické působení barev – je založeno na účincích světla na lidský organismus, zejména na lidský zrak a mozek, ale také na další anatomické funkce (například kontrasty barev mohou vyvolat následné (negativní) paobrazy, barvy rovněž ovlivňují náš vegetativní systém).

Psychologické působení barev – vychází z účinků barev na naši psychiku, zahrnuje také různé asociace, symboliku, syntézu či stimulace, které v nás barvy vyvolávají (na příklad teplé barvy jsou více aktivní a působí vzrušivě, kdežto studené barvy nás většinou uklidňují).

Vizuální působení barev – je dáno tím, jak se barvy projevují v ploše či prostoru (například teplé barvy mají tendenci v obraze vystupovat do popředí, chladné barvy naopak ustupovat do pozadí).“ [2, s. 14]

1.2 Vliv barev na člověka

Psychologie barev se zaměřuje na účinek barev na člověka. Různé barvy si můžeme spojovat např. i s různými zkušenostmi. Jejich vliv může být velmi subjektivní, přesto existují určité předpoklady, jak na nás dané barvy působí. [2, s. 42]

Vlivem barev na člověka se zabývaly již starověké civilizace, starověký Egypt, Čína, Řecko a Indie. Všechny zmíněné kultury se snažily pomocí barev léčit. [3] Průkopníkem v této oblasti se stal Niels Finsen (1860 – 1904). Ten pomocí světla a barev léčil kožní nemoci, regulovat krevní tlak, a dokonce ulevoval od bolestí. Experimentoval také, jaký vliv má barva místnosti na člověka. Dalším průkopníkem je Edwin B. Babbitt (1828 -1905), který své poznatky sepsal do knihy Podstata světla a barev. Babbitt používal například fialovou barvu pro uklidnění nebo žlutou jako nervový stimulant.[4]

Modrá barva je velmi oblíbená, to lze vidět například také u korporací jako je Facebook nebo LinkedIn. Modrou barvu si můžeme spojovat loajalitou, stabilitou, harmonií a klidem. Můžeme si ji spojovat s mořem či oblohou. Někdy si ji však spojujeme s negativní náladou.

Zelená je barva přírody. Může nám evokovat pocit svěžesti, čerstvosti, mládí a zdraví, proto tuto barvu můžeme vidět u výrobků, které se snaží být ekologické nebo přírodní.

Žlutá barva je velmi výrazná. V přírodě může značit nebezpečí, např. u vos. Nejspíše právě z tohoto důvodu používáme tuto barvu u varovných symbolů. Žlutou barvou můžeme symbolizovat teplo, štěstí a optimismus. Poslední dobou se s touto barvou setkáváme stále více. Je populární např. u „smajlíků“.

Oranžová barva v nás probouzí pocit čerstvosti, tepla a energie. Někdy se může využít jako alternativa červené, pokud by byla příliš výrazná.

Červená barva zobrazuje nějaký extrém. Často ji potkáme u různých varování nebo zákazech. Zároveň ji asociujeme s touhou, láskou, agresí nebo zlostí. Červená barva je také velmi výstižná, toho například využila značka Coca-Cola.

Na růžovou barvu velmi často nahlížíme jako na dívčí barvu. Ne vždy tomu tak musí být, jedná se o barvu, která může působit hravě a vesele.

Fialová barva je velmi nezvyklá a v přírodě vzácná. Vzbuzuje luxus, vznešenost, moudrost, magii a nadpřirozeno. Tato barva se stále používá velmi zřídka.

Černá barva je velmi oblíbená, jelikož se s ní dobře pracuje a lze jí jednoduše kombinovat s dalšími barvami. Černá barva vyjadřuje eleganci, sílu a tajemno. Často si ji ale můžeme spojovat se záparem, depresí nebo smutkem.

Bílá barva se často používá jako neutrální barva, bílý list papíru nebo bílé pozadí webu. Většinou symbolizuje čistotu, nevinnost a dobro.

Šedá barva je také neutrální barva, kterou lze použít na místech, kde nechceme narušit barevný design. Jedná se o barvu, která zobrazuje rovnováhu a pokrok. Musíme si dát ale pozor, protože může působit nudně, mdlé a negativně.

Hnědá barva působí přírodně, útulně, hřejivě. Nejedná se sice o příliš oblíbenou barvu, ale může tvořit čistý, přírodní, nadčasový vzhled. [5]

1.3 Barevné modely

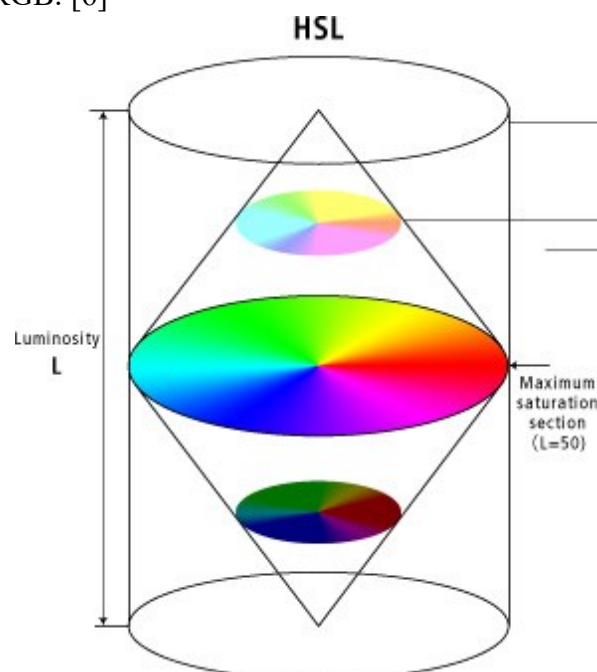
Barevný model je systém, pomocí kterého můžeme definovat a popsat barvy. Barevné modely mohou být popsány různými způsoby.

RGB

RGB model využívá aditivní míchání barev. Má tři základní barvy: R – červená (red), G – zelená (green), B – modrá (blue). Tyto barvy dosahují hodnot 0 až 255 a určují sytost barvy. Jelikož se jedná o aditivní systém, tak nízké hodnoty produkují tmavé barvy a vysoké hodnoty světlé barvy. Jedná se o velmi rozšířený model v digitálním světě. Tento model může být někdy omezený a nemusí umět zobrazit všechny barvy. Je omezen na 16,7 miliónů barev. [6]

HSL

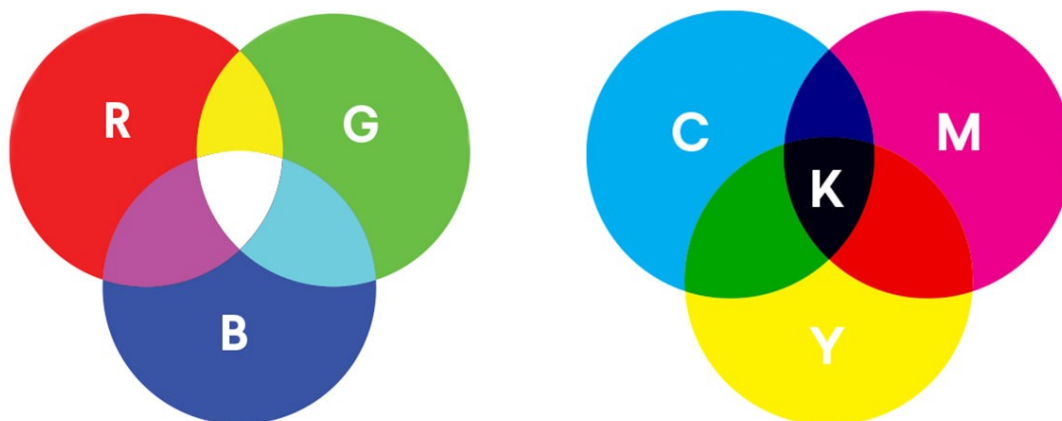
HSL je definován odstínem (hue), sytostí (saturation) a světlostí (Lightness). Matematicky se jedná o stejný model jako RGB. [6]



Obrázek 2: Model HSL

CMYK

CMYK je subtraktivní model, který využívá 4 barvy: azurová (cyan), purpurová (magenta), žlutá (yellow), černá (black). Teoreticky lze využít pouze 3 barvy, můžeme totiž vynechat černou. Bohužel barviva, která používáme nejsou dokonalé a s velkou pravděpodobností bychom při smíchání všech tří barev získaly odstín hnědé. Ideálně by všechna smíchaná barviva absorbovala celé barevné spektrum a získaly bychom černou. [6]



Obrázek 3: Modely RGB a CMYK

Model RGB využíváme v případech, kde získáváme barvy přidáváním světla, např. monitory nebo projektory. CMYK pro změnu používáme na fyzických materiálech. HSL je zase pro člověka intuitivní a dobře se s ním pracuje.

1.4 Soulad barev

Soulad barev je rozdělení barevného spektra tak, aby daná věc, například web, působil příjemně a harmonicky. Přestože barvy mohou být velice subjektivní a na každého člověka můžou působit jinak, tak máme několik principů. Principy můžeme rozdělit do skupin pro toho, kolik barev používáme. Dyády mají 2 barvy, triády 3, tetrády 4 a hexády 6. Barevné spektrum lze zakreslit různě, ale nejpoužívanější způsob je pomocí barevného kruhu, případně barevného trojúhelníku.

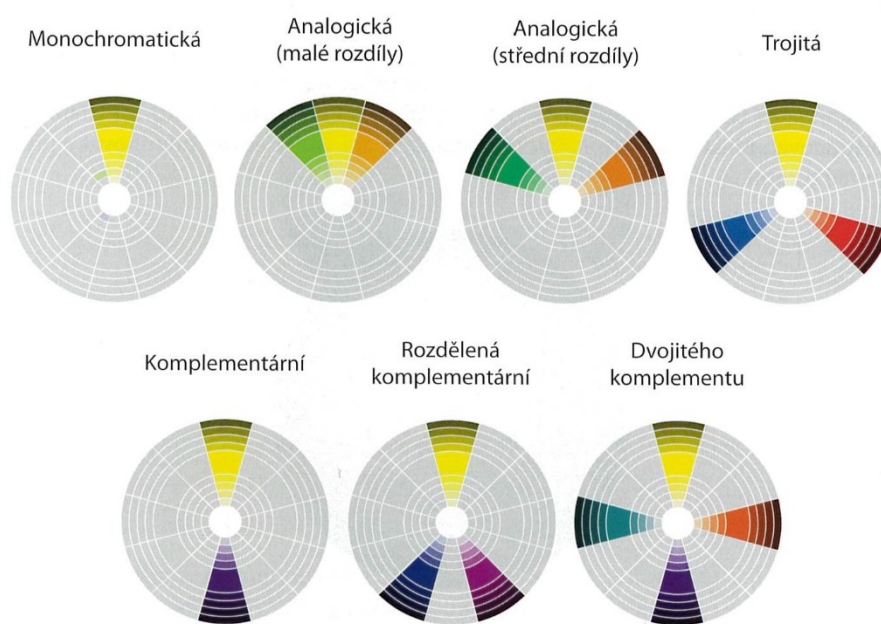
Dyády tvoří dvojici barev, které můžeme rozdělit na statické a dynamické. Ve statické dyádě má 2 podobné barvy, například červená a oranžová. V dynamické dyádě máme komplementární dvojici, tedy 2 barvy na opačné straně barevného kruhu, například červená a zelená.

Triády se skládají ze třech barev. Trojice mohou být buď vyvážené nebo akcentované. Vyvážené trojice barev v barevném kruhu vytváří rovnostranný trojúhelník, zatímco akcentovaná triáda obsahuje barvu, která se zásadně odlišuje a tvoří rovnoramenný trojúhelník.

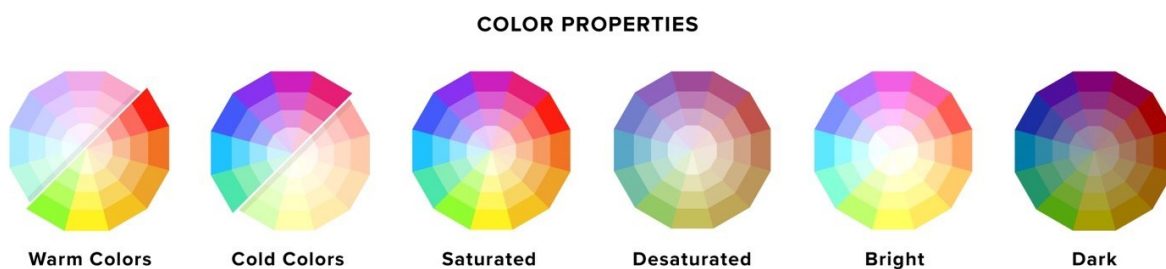
Tetrády tvoří na barevném kruhu čtverec nebo obdélník. Lze také říci, že se jedná o dvojici dynamických dyád, jelikož vytvoří právě čtverec nebo obdélník.

Posledním principem jsou hexády. Ty tvoří šestiúhelník. Hexády se používají velice zřídka a je třeba je používat s obezřetností.

Za pomoci těchto principů můžeme vytvořit barevné sestavy. Může se například jednat o následující rozdělení: Monochromatická, analogická, trojitá, komplementární, rozdělená komplementární, dvojitého komplementu. [1, s. 163-168]



Obrázek 4: Soulad barev



Obrázek 5: Soulad barev

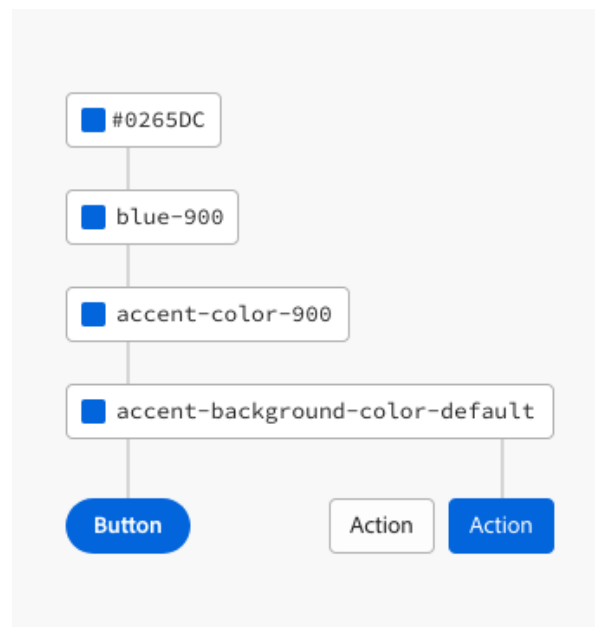
Různé aplikace na trhu mohou nabízet trochu jiné sestavy s jinými názvy, ale v základu jsou založené na stejných principech. Adobe Color využívá tyto principy: analogické, jednobarevné, trojúhelník, doplňkové, rozdělené, rozdělené doplňkové, čtverec, skládané a stíny. Paletton

využívá: monochromatický, sousedící, triáda a tetráda. Hexáda se v aplikacích většinou nevyskytuje. [7]

1.5 Návrhové tokeny

Pro vývoj nejen webových aplikací je důležité, aby grafik nebo designér byl schopen efektivně předat svůj návrh vývojáři. Zatímco grafik by měl vytvořit sadu barev, které byly využity v jeho návrhu, tak vývojář by měl získat takovou sadu barev, kterou lze jednoduše použít. Pokud by takový vývojář dostal seznam hexadecimálních kódů barev, které byly v návrhu použity, tak webovou aplikaci sice vytvoří, ale nejedná se efektivní způsob, pokud by každému prvku přímo určil, jakou má mít barvu. Lepší způsob je využít takzvané návrhové tokeny – design tokens. Jedná se o definice, které obsahují informace, v našem případě například CSS styly. Na obrázku 6. vidíme, jak mohou design tokens vypadat v podání od Adobe. Potřebujeme například tlačítko, které bude ve výchozí barvě, kterou používáme pro tlačítka. Místo toho abychom použili specifickou barvu, použijeme token, který používáme pro tato tlačítka. Vývojář se tedy nemusí zajímat o barevné schéma webu, protože to již vypracoval grafik. Vývojář tak může vytvořit web za pomoci pouze abstraktních prvků. [8]

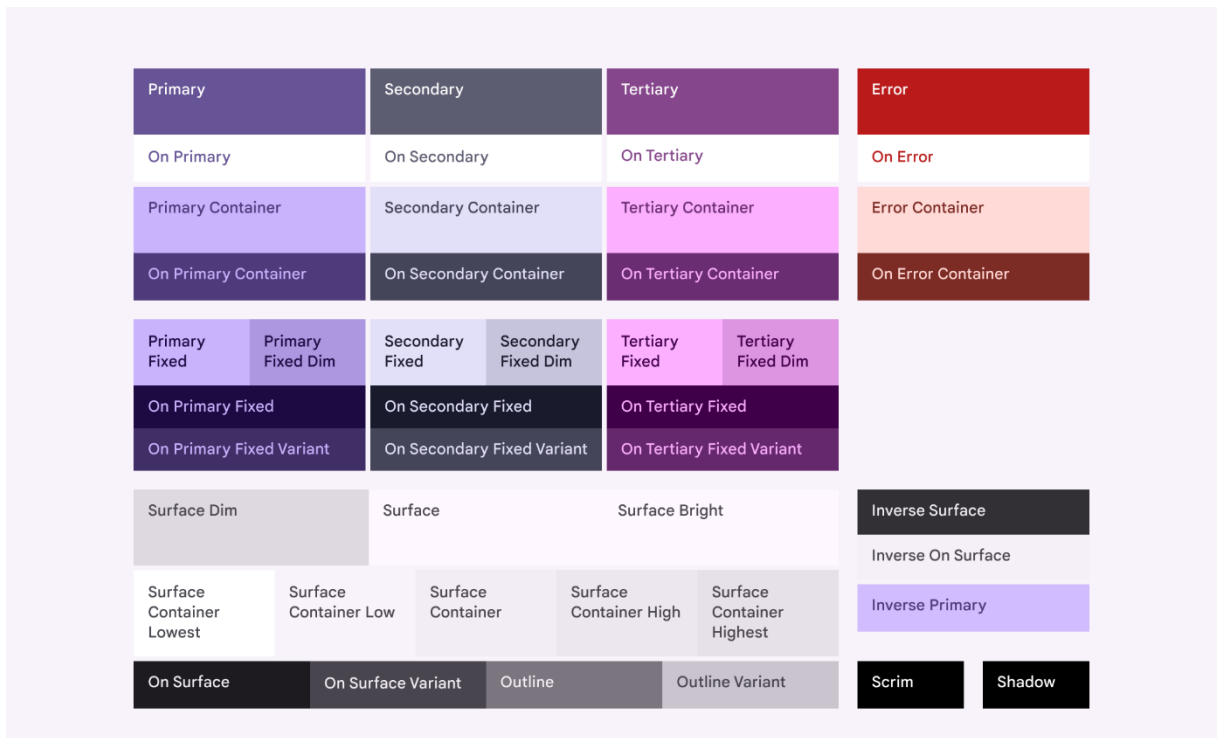
Existuje několik návodů, jak také design tokens používat. V této práci se podíváme na standardy od společností Google, IBM a Atlassian.



Obrázek 6: Struktura návrhového tokenu od Adobe

Google

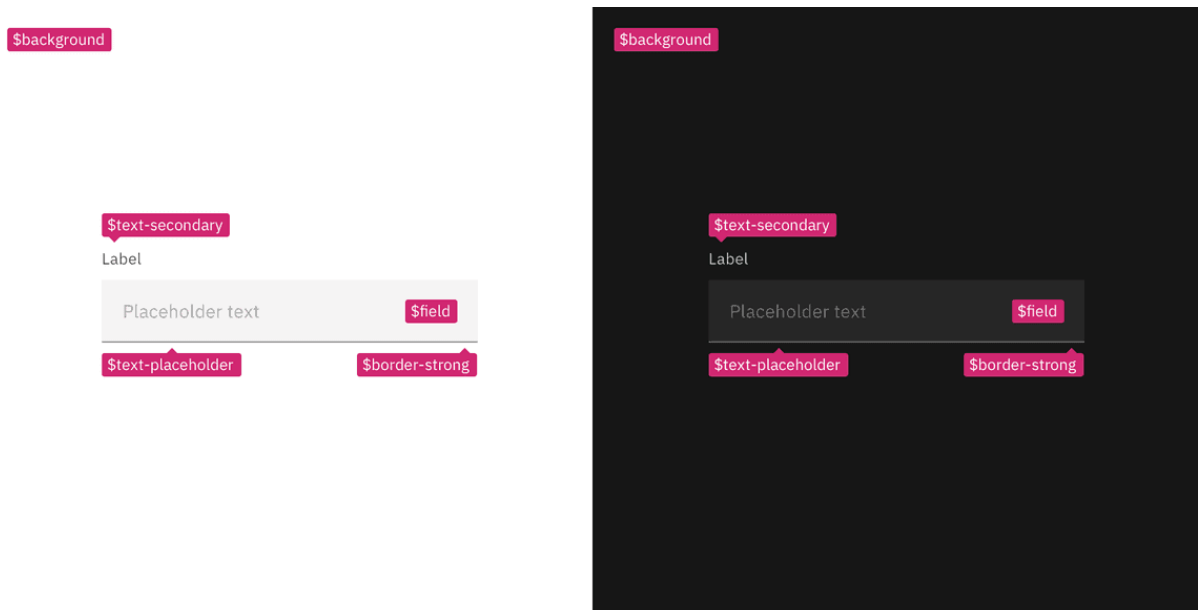
Design tokens od Google máme v rámci Material Design 3. Pro barvy jsou zde implementovány design tokens jako „Color roles“. Token Surface se používá pro pozadí velké oblasti. Primary, Secondary a Tertiary jsou hlavní barvy, které bude web využívat. Container je pro prvky jako tlačítka. „On“ je prefix pro prvky, které se nacházejí na rodičovském prvku, který využívá token bez prefixu „On“. [9]



Obrázek 7: Návrhový token v Material Design 3

IBM

IBM využívá svůj systém Carbon Design System. Ten rozděluje tokeny na 2 skupiny: Core tokens a Component tokens. Core tokens jsou globální tokeny, které jsou využívány napříč webem. Takový token může vypadat například takto: „\$text-primary“ nebo „\$link-visited“. Component token je pro využití na specifickém prvku, který se odlišuje. Například „\$notification-background-error“ nebo „\$tag-color-blue“. [10]



Obrázek 8: Design tokens v Carbon Design System

Atlassian

Atlassian Design System využívá tokeny, které jsou rozděleny na 3 části: základ (foundation), vlastnost (property) a modifikátor (modifier). Token se potom skládá následujícím způsobem: „základ.vlastnost.modifikátor“. Výsledně token může vypadat takto: „color.icon.information“. [11]

1.6 Barvy v CSS

Barevné moduly CSS jsou součástí specifikace CSS, které vydává W3C (World Wide Web Consortium). Barevné moduly přidávají možnosti, jak definovat barvy pomocí kaskádových stylů a jak jsou interpretovány.

Byly vydány Barevný modul CSS 3, Barevný modul CSS 4 a Barevný modul CSS 5 a nyní se pracuje na Barevném modulu CSS 6 (Barevný modul CSS 1 a Barevný modul CSS 2 neexistují, protože definice byly tehdy zaváděny přímo do specifikací CSS 1 a CSS 2). [12] [13] [14] [15]

Barevný modul CSS 3

Úroveň 3 přidává vlastnosti `color` a `opacity`. `Color` vyjadřuje barvu textu. Přijímá RGB hodnoty a hodnoty, které lze převést do RGB nebo RGBA. `Opacity` je vlastnost s hodnotou 0 až 1, která určuje průhlednost daného objektu.

Přidává také jednotku `<color>`, která může mít jak číselné vyjádření, tak slovní. Můžeme mít žlutou barvu, kterou můžeme definovat různými způsoby, např.: slovo „yellow“, hexadecimální zápis RGB „#FFFF00“, RGB „255,255,0“, HSL „60°, 100%, 50%“ a další.

Zápisy v CSS mohou vypadat následovně:

```
color: yellow
```

```
color: #ffff00
```

```
color: rgb(255,255,0)
```

```
color: hsl(60, 100%, 50%)
```

Všechny zápisy vyjadřují stejnou barvu. [12]

Barevný modul CSS 4

Úroveň 4 rozšiřuje práci s barvami. Přidává nové barevné prostory: HSL, HSLA, HWB, LAB, LCH, Oklab a Oklch. Přidává také předdefinované barevné prostory Display-P3 a Rec.2020. [13]

Barevný modul CSS 5

Tato úroveň přináší nové funkce `color-mix()`, `light-dark()` a `contrast-color()`. `Color-mix` umožňuje míchaní dvou nebo více barev v daném barevném prostoru. `Light-dark` poskytuje dynamický výběr barev na základě světlého nebo tmavého režimu. `Contrast-color` vybírá barvu

textu s dostatečným kontrastem vůči pozadí. Přibyla podpora pro vlastní barevné prostory a CMYK. [14]

Barevný modul CSS 6

V současné době vzniká šestá úroveň. V tuto chvíli je navrhována například funkce `contrast-color()`, která dokáže vybrat barvu ze seznamu splňující minimální kontrastní poměr. Další funkcí nejspíš bude např. `color-layers()` umožňující vracet barvu na základě barevných vrstev. [15]

1.7 Práce s barvou s ohledem na přístupnost webu

Krásný design není vše. Weby by také měly splňovat tzv. přístupnost. Jedná se o to, že by měly být dostupné všem, včetně lidí se zdravotním postižením. Celkově zahrnuje všechna postižení: sluchová, kognitivní, neurologická, fyzická, řečová a zraková. Samozřejmě z takového designu profitují i lidé bez postižení.

S přístupností se setkáme hlavně u státních a veřejných webů, například školy, nemocnice, úřady. Tyto organizace musejí splňovat různé standardy, například standard WCAG (Web Content Accessibility Guidelines) od W3C (World Wide Web Consortium) nebo standardy od ADA (The Americans with Disabilities Act). [16]

V případě barev je důležité, aby byly barvy jednotlivých prvků dostatečně kontrastní, například text a pozadí za textem. Kontrast zapisujeme v poměru, s tím že největší hodnota je 21:1 a nejmenší 1:1. WCAG 2.1 určuje 3 úrovně přístupnosti: A, AA a AAA. Úroveň A vyžaduje, aby barva Úroveň AA vyžaduje kontrastní poměr 4,5:1 pro normální text a 3:1 pro velký nebo tučný text. Úroveň AAA vyžaduje kontrastní poměr 7:1 pro normální text nebo 4,5:1 pro velký nebo tučný text. [17]

1.8 Přehled aplikací pro práci s barevnými schémata

Na trhu se nenachází moc aplikací, které by naplňovaly daná očekávání. Většinou se jedná o předvolená schémata barev nebo pouhé skládání barev. Byly vybrány 4 webové aplikace, které zdály býti jako nejužitečnější.

My Color Space (<https://mycolor.space/>)

Tato aplikace sice vezme od uživatele pouze jednu barvu, ale vygeneruje velké množství palet, které můžeme využít pro svůj web. Bohužel se zde nenachází žádné další možnosti přizpůsobení a všechny barvy si uživatel musí kopírovat po jednom.

Canva (<https://www.canva.com/colors/color-wheel/>)

Canva má nástroj Color wheel, který nám umožní vybrat barvy pomocí souladu barev. Barvy si uživatel může zkopírovat v hexadecimálním formátu nebo exportovat do PDF, což není vůbec užitečné.

Coolors (<https://coolors.co>)

Coolors má funkcí už značně více. Můžeme si zde vytvořit vlastní paletu o libovolném počtu barev, exportovat do mnoha různých formátů včetně CSS nebo zkontrolovat, jak vypadají barvy pro různou barvoslepost. Bohužel určité nástroje jsou až v placené verzi, jedná se například o generování barev pomocí souladu barev nebo zjištění kontrastu mezi barvami.

Paletton (<https://paletton.com>)

Paletton se zdá z vybraných nástrojů jako nejlepší. Umožňuje vybírat barvy pomocí souladu barev, má náhledy, jak barvy vypadají na demonstrační stránce, a to včetně z pohledu barvoslepeho. Obsahuje paletu, kterou lze ze všech zmíněných aplikací nejvíce přizpůsobit. Barvy lze exportovat do mnoha různých formátů včetně CSS, SASS nebo XML. Nevýhodou této aplikace je však její nepřívětivý design a některé nástroje ne vždy fungují, např. export. Od pohledu je zřejmé, že se jedná o zastaralou aplikaci.

Na základě posouzení výše uvedených aplikací lze konstatovat, že na trhu zatím není aplikace, která by byla uživatelsky přívětivá a obsahovala dostatečné množství nástrojů. My Color Space pouze generuje barvy na základě jedné barvy na vstupu. Canva zase využívá pouze soulad barev. Coolors a Paletton už nabízejí širší škálu nástrojů, bohužel v případě Coolors je hodně nástrojů placených a Paletton je již zastaralý, není responzivní, a ne každá funkce vždy funguje.

2 NÁVRH WEBOVÉ APLIKACE

2.1 Účel a požadavky na aplikaci

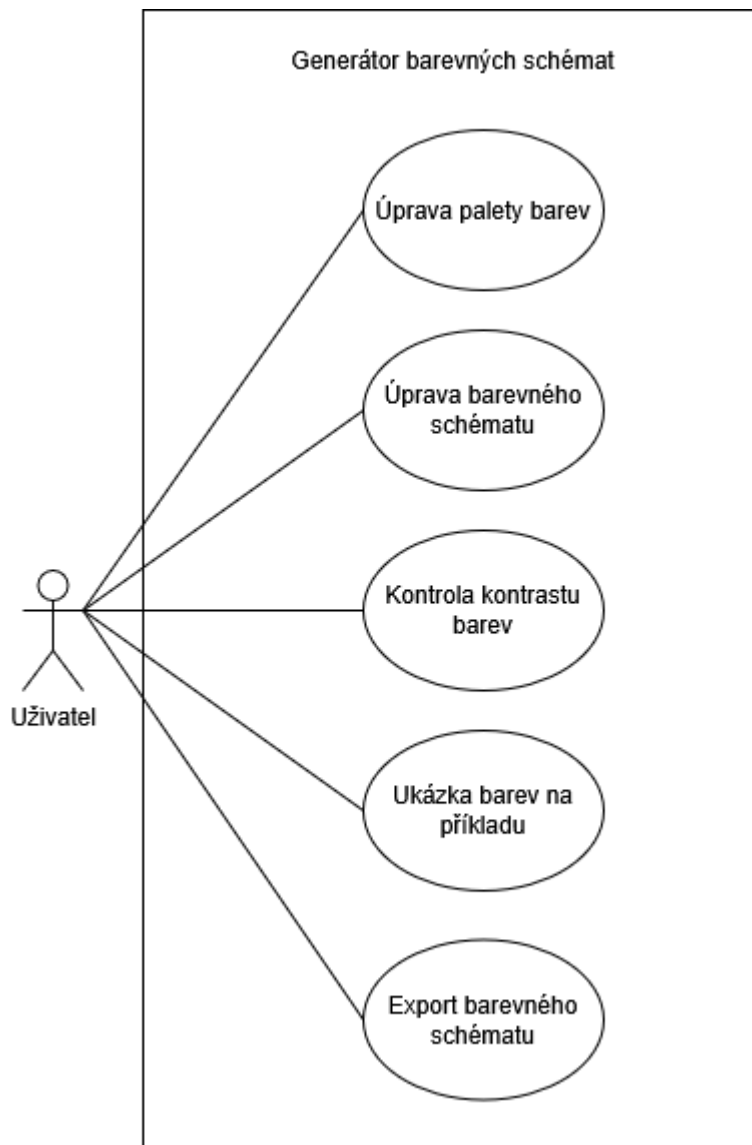
Aplikace by měla být navržena tak, aby byla dostupná co nejširšímu publiku grafiků a frontendových vývojářů. Klíčovým požadavkem je její intuitivnost a jednoduchost. Důležitá je také maximální dostupnost napříč různými zařízeními, což znamená, že musí být responzivní a přizpůsobit se jak stolním počítačům, tak tabletům. Zároveň by měla fungovat i v offline režimu, a proto využijeme technologii Progressive Web App (PWA), která umožní cachování potřebných souborů a pokračování v práci i bez internetového připojení.

Hlavní funkcionalitou aplikace je práce s barvami, přičemž uživatelům musí nabídnout dostatek nástrojů pro jejich správu a analýzu. Mezi klíčové funkce patří tvorba a generování barevných palet, které lze definovat na základě různých algoritmů, jako jsou analogické, komplementární či triadické kombinace. Součástí aplikace by měl být také nástroj pro porovnání kontrastu mezi dvěma barvami dle standardů WCAG, který pomůže uživatelům zajistit dostatečnou čitelnost textu v různých barevných kombinacích. Dále je vhodné umožnit vizualizaci vybraných barev v konkrétních designových situacích, například na tlačítkách, pozadích nebo textových blocích, aby si uživatel mohl představit jejich reálné použití.

Protože cílovou skupinou jsou také frontendoví vývojáři, měla by aplikace podporovat snadnou integraci do vývojářského workflow. To znamená, že vytvořená barevná schémata musí být rychle a jednoduše exportovatelná do formátu, který vývojáři běžně používají, tedy CSS.

Cílem aplikace je efektivní nástroj pro práci s barvami, který zjednoduší proces tvorby vizuálně konzistentních návrhů a usnadní jejich implementaci do webových projektů.

Případy užití



Obrázek 9:Případy užití

Případ užití 1: Úprava palety barev

Cíl: Generování palety barev

Aktér: Uživatel

Popis: Tento use case umožňuje uživateli vytvořit nebo upravit paletu barev prostřednictvím interaktivního nástroje. Uživatel si může vybrat základní barvu různými způsoby a definovat soulad barev pro generování doplňkových odstínů.

Základní scénář:

1 Uživatel vybere základní barvu

- Uživatel může zvolit základní barvu jedním z následujících způsobů:

1.1 Výběr barvy pomocí barevného kruhu

- Uživatel na barevném kruhu klikne na požadovanou barvu
- Na barevném kruhu se daná základní barva označí indikátor černé kružnice

1.2 Zadání hexadecimálního kódu

- Uživatel zadá do pole barvu pomocí hexadecimálního kódu
- Nástroj rozpozná zadání validní hodnoty (např. „#0FE9F9“) a aktualizuje výběr

2 Uživatel upraví jas základní barvy

- Uživatel posuvníkem upravuje jas základní barvy

3. Uživatel vybere typ harmonie barev

- Uživatel otevře nabídku s možnými typy souladu barev a jeden zvolí (např. komplementární)
- Po výběru jsou doplňkové barvy zobrazené pomoví bílých kružnic

4. Uživatel nastaví úhel souladu barev

- Uživatel může zvolit úhel souladu barev jedním z následujících způsobů:

4.1 Zadání úhlu souladu barev pomocí posuvníku

- Uživatel posuvníkem upravuje úhel souladu barev (např, 30°)

4.2 Zadání úhlu souladu barev posunutím na barevném kruhu

- Uživatel táhnutím bílého indikátoru na barevném kruhu upravuje úhel souladu barev

5 Systém zobrazí výslednou paletu barev

- Na základě všech nastavení se generuje paleta barev, kterou lze používat dále v aplikaci. Každá vygenerovaná barva je znázorněna jak vizuálně, tak zapsána pomocí hexadecimálního kódu.

Alternativní scénáře:

2A Uživatel nezmění jas barvy

- Uživatel se rozhodne nezměnit jas a ponechá výchozí jas barvy.

5A Uživatel nezmění úhel souladu

- Uživatel ponechá výchozí úhel souladu přednastavený typem souladu barev

Případ užití 2: Úprava barevného schématu

Cíl: Generování a úprava barevného schématu

Aktér: Uživatel

Popis: Tento use case umožňuje uživateli vytvořit nebo upravit barevné schéma. Uživatel si může zvolit základní barvy, pomocí kterých vygeneruje kompletní schéma a dále upravit jednotlivé návrhové tokeny pomocí různých metod.

Základní scénář:

1 Uživatel volí základní barvy

- Uživatel zvolí primární a sekundární barvu, které budou sloužit pro generování kompletního barevného schématu složeného z návrhových tokenů.
- Barvy lze zvolit následujícími způsoby:
 - Color picker – barva je zvolena pomocí komponenty color picker, jehož implementace je dána webovým prohlížečem
 - Hexadecimální kód – kód barvy je zadán do pole (např. „#0FE9F9“)
 - Táhnutí – barva je přetáhuta z palety barev

2 Uživatel generuje barevné schéma

- Na základě zvolené primární a sekundární barvy se vygenerují návrhové tokeny barevného schématu

3 Uživatel upravuje návrhové tokeny

- Po vygenerování barevného schématu může uživatel jednotlivé návrhové tokeny upravovat.

3.1 Změna barvy pomocí color pickeru

- Barva návrhového tokenu je zvolena pomocí komponenty color picker, jehož implementace je dána webovým prohlížečem

3.2 Změna barvy pomocí hexadecimálního kódu

- Kód barvy je zadán do pole (např. „#0FE9F9“)

3.3 Úprava jasu, odstínu a sytosti tlačítkem

- Uživatel může změnit vlastnosti barvy pomocí předpřipravených tlačítek:
 - Tlačítka pro zvýšení a snížení jasu
 - Tlačítka pro posun odstínu v jednom nebo druhém směru
 - Tlačítka pro zvýšení a snížení sytosti

3.4 Změna barvy pomocí přetáhnutím jiné barvy

- Barvu lze změnit přetáhnutím barvy z palety nebo jiného návrhového tokenu

3.5 Reset barvy

- Barva tokenu je vrácena do původního stavu

4 Dokončení barevného schématu

- Návrhové tokeny jsou automaticky uloženy. Pokud chce uživatel pokračovat v úpravách, může se vrátit k jakémukoliv kroku.

Alternativní scénáře:

3A Uživatel neprovede žádné úpravy návrhových tokenů

- Pokud uživatel neprovede žádnou změnu, ponechají původní hodnoty barevného schématu.

Případ užití 3: Kontrola kontrastu barev

Cíl: Kontrola, zda kontrasty barev splňují standard WCAG pro přístupnost

Aktér: Uživatel

Popis: Tento use case umožňuje uživateli zkontrolovat, zda jednotlivé barevné kombinace v rámci barevného systému (barva textu a barva pozadí) splňují pravidla pro kontrastní poměr dle standardu přístupnosti WCAG. Uživatel má přehlednou tabulku použitých návrhových tokenů a informace o tom, zda jednotlivé barvy splňují požadavky.

Základní scénář:

1 Uživatel otevře tabulku pro kontrolu kontrastů

- Uživatel klikne na tlačítko, které otevře okno s tabulkou kontroly kontrastů.

2 Uživatel analyzuje výsledky

- Uživatel může jednoduše analyzovat, které barevné kombinace splňují přístupnost a které ne.

3 Uživatel zavře okno

- Uživatel po analýze zavře okno a může pokračovat jiným use case.

Případ užití 4: Demonstrace na ukázkových aplikacích

Cíl: Zobrazení ukázek s použitým barevným schématem

Aktér: Uživatel

Popis: Tento use case umožňuje uživateli zobrazit, jak se aktuálně nastavené barvy (návrhové tokeny) projevují na konkrétních příkladech. Uživatel se může přepínat mezi různými ukázkami a sledovat aplikaci barev v různých scénářích.

Základní scénář:

1 Uživatel otevře okno s ukázkami

- Uživatel klikne na tlačítko, které otevře okno s ukázkami

2 Uživatel si prohlíží aktuální ukázkou

- Uživatel sleduje, jak jednotlivé prvky vypadají s použitým schématem.

3 Uživatel se přepíná mezi různými příklady

- Pomocí tlačítek se může uživatel přepnout na jinou ukázkou. Stejné návrhové tokeny jsou aplikovány v různých způsoby.

4 Uživatel analyzuje barevné schéma

- Uživatel se může na základě ukázek rozhodnout, zda mu barevné schéma vyhovuje či nikoliv a je třeba udělat nějaké úpravy.

5 Uživatel zavře okno

- Uživatel po analýze zavře okno a může pokračovat jiným use case.

Případ užití 5: Export barevného schématu

Cíl: Export barevného schématu na zařízení uživatele

Aktér: Uživatel

Popis: Tento use case umožňuje uživateli exportovat aktuálně vytvořené nebo upravené barevné schéma ve formě design tokenů do různých formátů. Cílem je umožnit snadné využití schématu v dalších nástrojích nebo přímo v kódu.

Základní scénář:

1 Uživatel dokončí úpravy barevného schématu

- Uživatel má hotové a upravené barevné schéma, případně právě dokončil jeho tvorbu nebo úpravu.

2 Uživatel si zvolí formát exportu

- Uživatel má na výběr z několika tlačítek, které mu umožní export v různých formátech.
- Uživatel má na výběr z následujících možností:
 - Export do schránky – uživateli se do schránky zkopíruje CSS kód se schématem
 - Export do CSS souboru – uživateli se stáhne CSS soubor se schématem

- Export do JSON souboru – uživatel se stáhne JSON soubor se schématem

2.2 Analýza požadavků na aplikaci

Funkční požadavky

Aplikace si bude pamatovat nastavení z poslední relace.

Výsledky se zobrazují v reálném čase bez odesílání formulářů.

Obsahuje paletu s výběrem barev pomocí barevného kruhu a souladu barev.

Export barevného schématu.

Kontrola kontrastu barev a zda splňují přístupnost podle standardu WCAG.

Demonstrace barev na ukázkových aplikacích.

Nefunkční požadavky

Aplikace bude responzivní.

Bude se o jednostránkovou webovou aplikaci.

Aplikace bude splňovat Lighthouse skóre alespoň 85.

Aplikace bude splňovat standard přístupnosti WCAG AA.

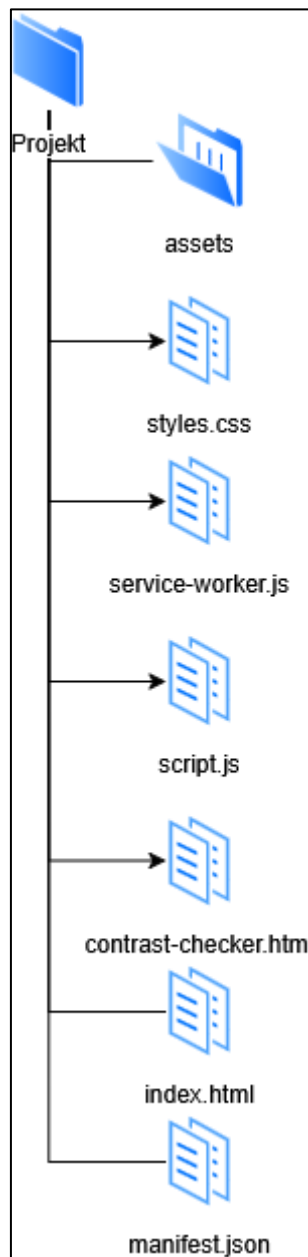
Aplikace umí pracovat i v režimu offline.

Aplikace je instalovatelná na koncovém zařízení uživatele.

2.3 Architektura

Bude se jednat o jednostránkovou aplikaci (Single Page Application - SPA), která bude mít veškerý obsah html v souboru index.html + contrast-checker.html s propočty kontrastů. Stylistické css nebude pomocí žádného frameworku, ale bude se jednat o zcela vlastní implementaci již od základu. Javascript bude rozdělen do 2 hlavních modulů: script.js (hlavní

logika celé aplikace) a service-worker.js (cachování pro PWA). Aby byla implementace PWA kompletní, tak k service workeru přidáme ještě manifest.json s konfigurací.



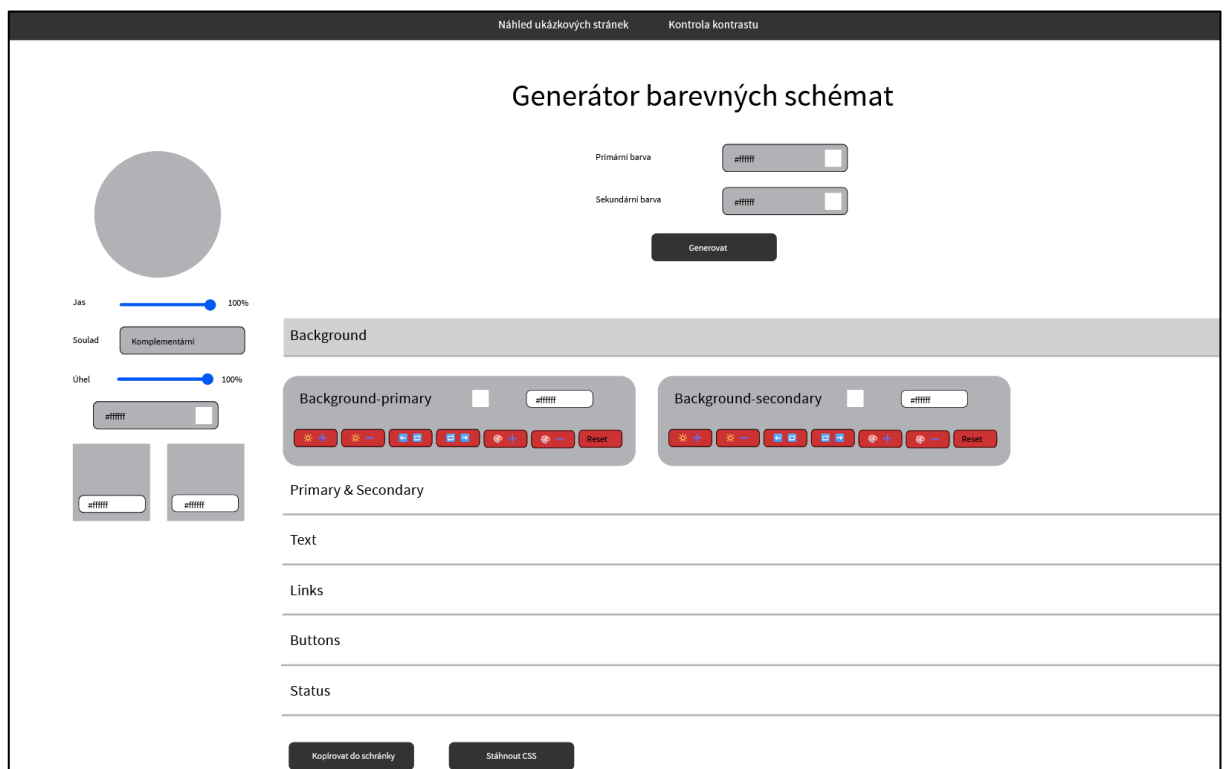
Obrázek 10: Struktura aplikace

Aplikace využívá Local Storage k ukládání zvolené primární a sekundární barvy a barvy jednotlivých tokenů. Tímto způsobem zůstávají data dostupná i po zavření nebo obnovení stránky. Aplikace je implementována jako Progressive Web App (PWA), což znamená, že podporuje offline režim a rychlejší načítání díky cachování prostřednictvím Service Workeru. Statické soubory aplikace (HTML, CSS, JavaScript) jsou uloženy v cache, což umožňuje jejich

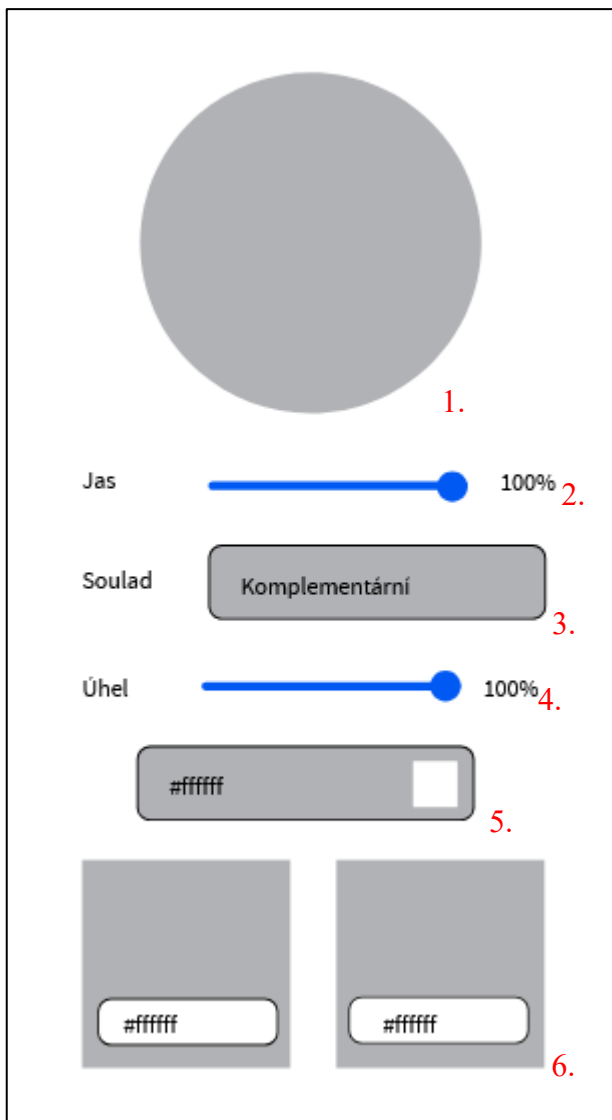
okamžité načtení i při pomalejším připojení nebo bez internetového připojení. Pokud se v aplikaci objeví nová verze, Service Worker zajistí aktualizaci souborů při dalším načtení stránky.

2.4 Návrh uživatelského rozhraní aplikace

Aplikace je rozdělena na 2 hlavní sekce: Paleta a Generátor barevných schémat. Na horním okraji se nachází menu. Rozhraní je navrženo s ohledem na to, že uživatel bude aplikaci používat na počítači případně tabletu, kde bude vyvíjet webové aplikace. Barevné schéma aplikace je z velké části udělané v barevných odstínech šedi, aby byla co nejméně ovlivněna tvorba barevných schémat.

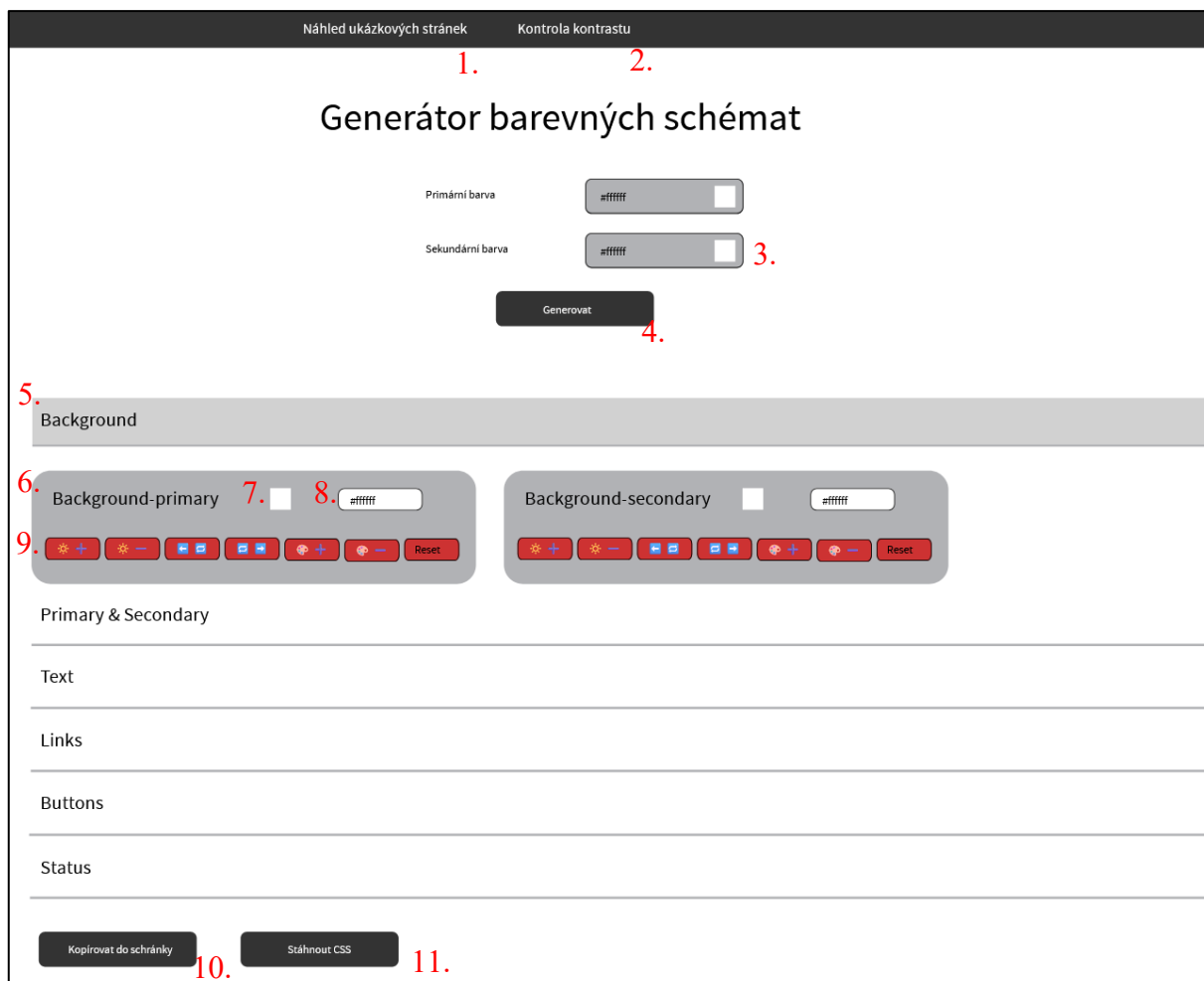


Obrázek 11: Rozhraní aplikace



Obrázek 12: Generátor palety

1. Barevný kruh, na kterém uživatel kliknutím nebo táhnutím navolí odstín a sytost barvy. Společně s prvkem 2. tvoří HSV barevný model.
2. Posuvník pro jas vybrané barvy. Společně s prvkem 1. tvoří HSV barevný model.
3. Výběr druhu souladu barev, pomocí kterého se vygenerují barvy v prvku 6.
4. Posuvník pro výběr úhlu souladu barev pro vygenerování barev prvku 6.
5. Textové pole s hexadecimálním kódem barvy vybrané pomocí prvku 1. a 2. Alternativně lze zadat hexadecimální kód barvy a aktualizuje prvek 1. a 2. podle zadané barvy. Vedle hexadecimálního kódu je daná barva také zobrazena.
6. Prvek, který znázorňuje vygenerované barvy podle předvoleb prvků 1. až 5. V dolní části je zobrazen hexadecimální kód dané barvy. Počet prvků 6. se mění v závislosti na souladu barev v prvku 3.



Obrázek 13: Generátor barevných schémat

1. Tlačítko, které otevře okno s ukázkovými stránkami.
2. Tlačítko, které otevře okno s tabulkou, zda jsou splněny kontrasty pro přístupnost podle standardu WCAG.
3. Primární a sekundární barva pro generaci barevného schématu. Barvu lze vybrat kliknutím na čtverec, kde se zobrazuje aktuální barva nebo vepsáním hexadecimálního kódu barvy.
4. Tlačítko, které pomocí vybrané primární a sekundární barvy vygeneruje barvy barevného schématu.
5. Sekce barevného schématu. Kliknutím lze návrhové tokeny dané sekce zobrazit nebo zase schovat.
6. Jednotlivý návrhový token. Barva pozadí je stejná jako barva tokenu.
7. Tlačítko, kde lze vybrat barvu tokenu.
8. Textové pole s hexadecimálním kódem dané barvy. Barvu tokenu lze změnit vepsáním hexadecimálního kódu.

9. Doplnková tlačítka pro dodatečnou změnu barvy tokenu. Tlačítka zleva doprava mají následující funkcionalitu: zvýšení jasů o 10 %, snížení jasů o 10 %, změna odstínu o 10 %, změna odstínu o 10 % opačným směrem, zvýšení sytosti o 10 %, snížení sytosti 10 %, resetování barvy tokenu na barvu, která je uložena v local storage.
10. Tlačítko, které vygeneruje a zkopíruje do schránky CSS kód z navoleného barevného schématu.
11. Tlačítko, které vygeneruje a stáhne CSS soubor se styly navoleného barevného schématu.

Tabulka kontrastů (WCAG)

Token textu	Token Pozadí	Ukázkový text	Kontrastní poměr	Výsledek
Text-primary	Background-primary	Ukázkový text	21.00:1	✅ AA
Text-secondary	Background-secondary	Ukázkový text	17.57:1	✅ AA
Text-on-primary	Primary	Ukázkový text	5.68:1	✅ AA

Obrázek 14: Tabulka kontrastů

Okno pro tabulku kontrastů je složeno z názvu tokenu textu, tokenu pozadí, ukázkového textu v barvách daných tokenů, kontrastního poměru barev daného tokenu a výsledku, zda splňují standard WCAG AA nebo pouze pro velké texty anebo vůbec nesplňují.

3 IMPLEMENTACE WEBOVÉ APLIKACE

3.1 Návrhové tokeny

Bylo rozhodnuto o vytvoření pěti hlavních skupin design tokenů, a to: základ, text, odkazy, tlačítka a status.

Základ obsahuje 2 základní barvy a 2 barvy pozadí. Byly zvoleny právě 2 barvy (primární a sekundární), protože předpokládáme, že další barvu použijeme až u specifických případů, kde si uživatel daný prvek doplní.

U textu byly zvoleny 3 hlavní druhy tokenů. Primární a sekundární text (Text-primary a Text-secondary), které budou na jednom z barev pozadí (Background-primary nebo Background-secondary). Ve speciálních případech můžeme mít text na základních barvách (Primary a Secondary), zde použijeme Text-on-primary nebo Text-on-secondary. Jelikož se jedná o speciální prvky, tak jsou připraveny také Text-on-primary-hover a Text-on-secondary-hover budou-li zapotřebí. Posledním tokenem v textové skupině je Text-disabled pro označení neplatných prvků.

Skupina odkazů je velmi jednoduchá. Máme primární a sekundární odkaz (Link-primary a Link-secondary), hover efekt pro tyto odkazy (Link-primary-hover a Link-secondary-hover) a barvu pro již navštívené odkazy (Link-visited).

Předposlední skupinou jsou tlačítka. Každé tlačítko má barvu pozadí (Button-primary a Button-secondary) a barvu textu (Button-primary-text a Button-secondary-text). To stejné musíme mít také pro hover efekt (Button-primary-hover, Button-secondary-hover, Button-primary-text-hover a Button-secondary-text-hover). Zároveň je třeba také odlišit neaktivní tlačítka (Button-disabled a Button-disabled-text).

Poslední skupinou je status. Ten je pro prvky, které označují různé zprávy nebo oznámení pro uživatele. Byly vytvořeny 4 druhy různých zpráv. Pozitivní status označující určitý úspěch, negativní status označující právě opak, status pro upozornění a v poslední řadě neutrální status pro zbytek případů.

Základ

- Background-primary
- Background-secondary
- Primary
- Secondary

Text

- Text-primary
- Text-secondary
- Text-on-primary
- Text-on-secondary
- Text-on-primary-hover
- Text-on-secondary-hover
- Text-disabled

Odkazy

- Link-primary
- Link-primary-hover
- Link-secondary
- Link-secondary-hover
- Link-visited

Tlačítka

- Button-primary
- Button-secondary
- Button-primary-hover
- Button-secondary-hover
- Button-primary-text
- Button-secondary-text
- Button-primary-text-hover
- Button-secondary-text-hover
- Button-disabled
- Button-disabled-text

Status

- Status-positive
- Status-positive-text
- Status-positive-hover
- Status-negative
- Status- negative-text
- Status- negative-hover
- Status-caution
- Status- caution-text
- Status- caution-hover

- Status-neutral
- Status-neutral-text
- Status-neutral-hover

3.2 Generátor barevných schémat

Hlavními soubory aplikace jsou index.html, styles.css a script.js. Přestože index.html i styles.css jsou nedílnou součástí script.js je alfou a omegou celé aplikace.

3.2.1 Script.js

Soubor script.js obsahuje funkce pro fungování hlavní části aplikace, a to generování palety, generování barevného schématu, import, export a další.

Funkce „updatePalette“ umožňuje pomocí zvolené základní barvy a zvoleného souladu barev vygenerovat paletu, kterou lze využít pro generování nebo úpravu barevného schématu. Každá barva palety je zobrazena jak vizuálně, tak pomocí hexadecimálního kódu. Barvu palety lze do barevného schématu aplikovat také přetáhnutím pomocí funkce „Drag and Drop“.

```
// Vykreslení palety
function updatePalette(baseColor) {
  var harmony = document.getElementById("harmony").value;
  var colors = [];
  if (harmony === "complementary") {
    colors = getComplementaryColors(baseColor);
  } else if (harmony === "analogous") {
    colors = getAnalogousColors(baseColor);
  } else if (harmony === "triadic") {
    colors = getTriadicColors(baseColor);
  } else if (harmony === "tetradic") {
    colors = getTetradicColors(baseColor);
  } else {
    colors = [baseColor];
  }
  var paletteContainer = document.getElementById("palette");
  paletteContainer.innerHTML = "";
  colors.forEach(function (color, index) {
    var swatch = document.createElement("div");
    swatch.className = "color-swatch";
    swatch.style.backgroundColor = color;

    var colorCode = document.createElement("div");
    colorCode.className = "color-code";
    colorCode.textContent = color;
    swatch.appendChild(colorCode);

    var indicator = createHarmonyIndicator(index);
    swatch.appendChild(indicator);

    enablePaletteDragAndDrop(swatch)
  });
}
```

```

        paletteContainer.appendChild(swatch);
    });
}

```

Funkce „drawColorWheel“ vykresluje pomocí komponenty Canvas barevný kruh pro výběr základní barvy, která se používá pro generování barev palety. Kruh je vykreslen pomocí 360 klínů otočených vždy o 1°, čímž následně vytvoří daný kruh s celým spektrem barev. Základní zvolená barva je na barevném kruhu znázorněna pomocí indikátoru černé kružnice. Další barvy, které získáme pomocí souladu barev jsou znázorněny indikátory bílé kružnice.

```

// Vykreslení barevného kruhu
function drawColorWheel() {
    var canvas = document.getElementById("colorWheel");
    var ctx = canvas.getContext("2d");
    var width = canvas.width;
    var height = canvas.height;
    var cx = width / 2;
    var cy = height / 2;
    var radius = Math.min(cx, cy);
    ctx.clearRect(0, 0, width, height);
    ctx.lineJoin = "round";
    ctx.lineCap = "round";
    // Kruh pomocí 360 klínů, kde každý klín má startAngle = (angle - 1) a
    endAngle = (angle + 1)
    for (var angle = 0; angle < 360; angle++) {
        var startAngle = (angle - 1) * Math.PI / 180;
        var endAngle = (angle + 1) * Math.PI / 180;
        ctx.beginPath();
        ctx.moveTo(cx, cy);
        ctx.arc(cx, cy, radius, startAngle, endAngle, false);
        ctx.closePath();
        var grad = ctx.createRadialGradient(cx, cy, 0, cx, cy, radius);
        grad.addColorStop(0, "#FFFFFF");
        grad.addColorStop(1, hsvToHex(angle, 1, 1));
        ctx.fillStyle = grad;
        ctx.fill();
    }
    // Základní černý indikátor
    var rad = selection.hue * Math.PI / 180;
    var indX = cx + selection.saturation * radius * Math.cos(rad);
    var indY = cy + selection.saturation * radius * Math.sin(rad);
    ctx.beginPath();
    ctx.arc(indX, indY, 8, 0, 2 * Math.PI, false);
    ctx.lineWidth = 3;
    ctx.strokeStyle = "#000000";
    ctx.stroke();

    // Další bílé indikátory
    var harmony = document.getElementById("harmony").value;
    var harmonyAngle =
parseFloat(document.getElementById("harmonyAngle").value);
    var additionalIndicators = [];
    if (harmony === "complementary") {

```

```

        additionalIndicators.push({hue: (selection.hue + harmonyAngle) %
360, saturation: selection.saturation});
    } else if (harmony === "analogous") {
        additionalIndicators.push({hue: (selection.hue - harmonyAngle +
360) % 360, saturation: selection.saturation});
        additionalIndicators.push({hue: (selection.hue + harmonyAngle) %
360, saturation: selection.saturation});
    } else if (harmony === "triadic") {
        additionalIndicators.push({hue: (selection.hue + harmonyAngle) %
360, saturation: selection.saturation});
        additionalIndicators.push({hue: (selection.hue - harmonyAngle +
360) % 360, saturation: selection.saturation});
    } else if (harmony === "tetradic") {
        additionalIndicators.push({hue: (selection.hue + harmonyAngle) %
360, saturation: selection.saturation});
        additionalIndicators.push({hue: (selection.hue + 180) % 360,
saturation: selection.saturation});
        additionalIndicators.push({hue: (selection.hue + 180 +
harmonyAngle) % 360, saturation: selection.saturation});
    }
    additionalIndicators.forEach(function (ind) {
        var radInd = ind.hue * Math.PI / 180;
        var indX = cx + ind.saturation * radius * Math.cos(radInd);
        var indY = cy + ind.saturation * radius * Math.sin(radInd);
        ctx.beginPath();
        ctx.arc(indX, indY, 8, 0, 2 * Math.PI, false);
        ctx.lineWidth = 3;
        ctx.strokeStyle = "#FFFFFF";
        ctx.stroke();
    });
}

```

Funkce na komplementární, analogické, triadické a tetradické soulady barev, které vytvoří ze základní barvy další barvy na základě daného souladu barev a úhlu.

```

// === Vypočty souladů barev ===
function getComplementaryColors(baseHex) {
    var rgb = hexToRgb(baseHex);
    var hsl = rgbToHsl(rgb.r, rgb.g, rgb.b);
    var angle = parseFloat(document.getElementById("harmonyAngle").value);
    var compHue = (hsl.h + angle) % 360;
    var compHex = hslToHex(compHue, hsl.s, hsl.l);
    return [baseHex, compHex];
}

function getAnalogousColors(baseHex) {
    var rgb = hexToRgb(baseHex);
    var hsl = rgbToHsl(rgb.r, rgb.g, rgb.b);
    var angle = parseFloat(document.getElementById("harmonyAngle").value);
    var leftHue = (hsl.h - angle + 360) % 360;
    var rightHue = (hsl.h + angle) % 360;
    return [hslToHex(leftHue, hsl.s, hsl.l), baseHex, hslToHex(rightHue,
hsl.s, hsl.l)];
}

function getTriadicColors(baseHex) {
    var rgb = hexToRgb(baseHex);
    var hsl = rgbToHsl(rgb.r, rgb.g, rgb.b);

```

```

    var angle = parseFloat(document.getElementById("harmonyAngle").value);
    var firstHue = (hsl.h + angle) % 360;
    var secondHue = (hsl.h - angle + 360) % 360;
    return [baseHex, hslToHex(firstHue, hsl.s, hsl.l), hslToHex(secondHue,
hsl.s, hsl.l)];
}

function getTetradicColors(baseHex) {
    var rgb = hexToRgb(baseHex);
    var hsl = rgbToHsl(rgb.r, rgb.g, rgb.b);
    var angle = parseFloat(document.getElementById("harmonyAngle").value);

    var hue1 = (hsl.h + angle) % 360;
    var hue2 = (hsl.h + 180) % 360;
    var hue3 = (hue2 + angle) % 360;

    return [
        baseHex,
        hslToHex(hue1, hsl.s, hsl.l),
        hslToHex(hue2, hsl.s, hsl.l),
        hslToHex(hue3, hsl.s, hsl.l)
    ];
}

```

Funkce zajišťující táhnutí indikátoru na barevném kruhu – rozlišuje se, zda měníme úhel souladu nebo základní barvu. Následně se přepočítá zvolená barva, aby se mohly aktualizovat barvy vygenerované paletou.

```

// Táhnutí na barevném kruhu
(function () {
    var canvas = document.getElementById("colorWheel");
    var width = canvas.width;
    var height = canvas.height;
    var cx = width / 2;
    var cy = height / 2;
    var radius = Math.min(cx, cy);

    var dragging = false;
    var dragMode = null; // "base" nebo "angle"

    canvas.addEventListener("mousedown", function (event) {
        var rect = canvas.getBoundingClientRect();
        var x = event.clientX - rect.left;
        var y = event.clientY - rect.top;
        var dx = x - cx;
        var dy = y - cy;
        var distance = Math.sqrt(dx * dx + dy * dy);
        if (distance > radius) return;

        // Pozice dodatečných indikátorů pro všechny soulady
        var harmony = document.getElementById("harmony").value;
        var currentHarmonyAngle =
parseFloat(document.getElementById("harmonyAngle").value);
        var additionalIndicators = [];
        if (harmony === "complementary") {
            additionalIndicators.push({

```

```

        hue: (selection.hue + currentHarmonyAngle) % 360,
        saturation: selection.saturation
    });
} else if (harmony === "analogous" || harmony === "triadic") {
    additionalIndicators.push({
        hue: (selection.hue - currentHarmonyAngle + 360) % 360,
        saturation: selection.saturation
    });
    additionalIndicators.push({
        hue: (selection.hue + currentHarmonyAngle) % 360,
        saturation: selection.saturation
    });
} else if (harmony === "tetradic") {
    additionalIndicators.push({
        hue: (selection.hue + currentHarmonyAngle) % 360,
        saturation: selection.saturation
    });
    additionalIndicators.push({hue: (selection.hue + 180) % 360,
saturation: selection.saturation});
    additionalIndicators.push({
        hue: (selection.hue + 180 + currentHarmonyAngle) % 360,
        saturation: selection.saturation
    });
}
// Zjistíme, zda klik proběhl v blízkosti některého z dodatečných
indikátorů
var threshold = 10; // prahová vzdálenost v pixelech
var nearAdditional = false;
additionalIndicators.forEach(function (ind) {
    var radInd = ind.hue * Math.PI / 180;
    var indX = cx + ind.saturation * radius * Math.cos(radInd);
    var indY = cy + ind.saturation * radius * Math.sin(radInd);
    var dInd = Math.sqrt((x - indX) * (x - indX) + (y - indY) * (y
- indY));
    if (dInd < threshold) {
        nearAdditional = true;
    }
});

if (nearAdditional) {
    dragMode = "angle";
} else {
    dragMode = "base";
}
dragging = true;

if (dragMode === "base") {
    var angleDeg = Math.atan2(dy, dx) * 180 / Math.PI;
    if (angleDeg < 0) angleDeg += 360;
    selection.hue = angleDeg;
    selection.saturation = distance / radius;
    updateSelectedColor();
} else if (dragMode === "angle") {
    var mouseAngle = Math.atan2(dy, dx) * 180 / Math.PI;
    if (mouseAngle < 0) mouseAngle += 360;
    var newAngle = mouseAngle - selection.hue;
    if (newAngle < 0) newAngle += 360;
    if (newAngle > 180) newAngle = 360 - newAngle;
    document.getElementById("harmonyAngle").value = newAngle;
    document.getElementById("harmonyAngleVal").textContent =

```

```

newAngle.toFixed(0) + "°";
        updateSelectedColor();
    }
});

canvas.addEventListener("mousemove", function (event) {
    if (!dragging) return;
    var rect = canvas.getBoundingClientRect();
    var x = event.clientX - rect.left;
    var y = event.clientY - rect.top;
    var dx = x - cx;
    var dy = y - cy;
    var distance = Math.sqrt(dx * dx + dy * dy);
    if (distance > radius) distance = radius;
    if (dragMode === "base") {
        var angleDeg = Math.atan2(dy, dx) * 180 / Math.PI;
        if (angleDeg < 0) angleDeg += 360;
        selection.hue = angleDeg;
        selection.saturation = distance / radius;
        updateSelectedColor();
    } else if (dragMode === "angle") {
        var mouseAngle = Math.atan2(dy, dx) * 180 / Math.PI;
        if (mouseAngle < 0) mouseAngle += 360;
        var newAngle = mouseAngle - selection.hue;
        if (newAngle < 0) newAngle += 360;
        if (newAngle > 180) newAngle = 360 - newAngle;
        document.getElementById("harmonyAngle").value = newAngle;
        document.getElementById("harmonyAngleVal").textContent =
newAngle.toFixed(0) + "°";
        updateSelectedColor();
    }
});

canvas.addEventListener("mouseup", function () {
    dragging = false;
    dragMode = null;
});
canvas.addEventListener("mouseleave", function () {
    dragging = false;
    dragMode = null;
});
})();

```

Funkce, která vytvoří token v barevném schématu – přiřadí se název tokenu a barva, aktivuje se funkcionalita „drag and drop“ a vytvoří všechny potřebné prvky k elementu včetně vytvoření tlačítek na úpravu barvy tokenu.

```

// Vytvoření token elementu
async function createTokenElement(parent, key, value) {
    const div = document.createElement("div");
    div.className = "token";
    div.dataset.key = key;
    div.dataset.defaultColor = value;

    enableDragAndDrop(div); // Aktivace drag & drop

```

```

const main = document.createElement("div");
main.className = "token-main";

const label = document.createElement("label");
label.textContent = key;

const colorInput = document.createElement("input");
colorInput.type = "color";
colorInput.value = value;
colorInput.oninput = () => updateTokenColor(div, colorInput.value);

const textInput = document.createElement("input");
textInput.type = "text";
textInput.maxLength = 7;
textInput.value = value.toUpperCase();
textInput.oninput = () => {
  if (/^[0-9A-F]{6}$/i.test(textInput.value)) {
    colorInput.value = textInput.value.toUpperCase();
    updateTokenColor(div, textInput.value.toUpperCase());
  }
};

const buttonsContainer = document.createElement("div");
buttonsContainer.className = "token-buttons";

const buttons = await Promise.all([
  createAdjustButtonWithIcon("assets/sun.svg", "brightness-plus", ()
=> adjustTokenColor(div, adjustBrightness, 10)),
  createAdjustButtonWithIcon("assets/sun2.svg", "brightness-minus",
() => adjustTokenColor(div, adjustBrightness, -10)),
  createAdjustButtonWithIcon("assets/hue.svg", "hue-left", () =>
adjustTokenColor(div, adjustHue, -10)),
  createAdjustButtonWithIcon("assets/hue2.svg", "hue-right", () =>
adjustTokenColor(div, adjustHue, 10)),
  createAdjustButtonWithIcon("assets/saturation.svg", "saturation-
plus", () => adjustTokenColor(div, adjustSaturation, 10)),
  createAdjustButtonWithIcon("assets/saturation2.svg", "saturation-
minus", () => adjustTokenColor(div, adjustSaturation, -10)),
  createAdjustButtonWithIcon("assets/reset.svg", "reset", () =>
resetSingleToken(div))
]);

buttons.forEach(btn => buttonsContainer.appendChild(btn));

main.appendChild(label);
main.appendChild(colorInput);
main.appendChild(textInput);
div.appendChild(main);
div.appendChild(buttonsContainer);

parent.appendChild(div);

applyTokenStyle(div, value);
}

```

Funkce na uložení a načtení z localStorage – uložení tokenů pracuje tak, že se vezmou tokeny, jejich název a barva, které se následně uloží v JSON formátu do localStorage. Načtení funguje

tak, že se prvně načte primární a sekundární barva pro generování schémat a poté se načtou tokeny do jednotlivých sekcí a následně se vykreslí. Jestliže nejsou uloženy žádné tokeny, vygenerují se.

```
function saveTokensToLocalStorage() {
  const tokens =
Array.from(document.querySelectorAll(".token")).reduce((acc, token) => {
  const key = token.dataset.key;
  const color = token.querySelector('input[type="color"]').value;
  acc[key] = color;
  return acc;
}, {});
  localStorage.setItem("designTokens", JSON.stringify(tokens));
}

// Načtení z Local Storage
async function loadFromLocalStorage() {
  const savedTokens = JSON.parse(localStorage.getItem("designTokens"));
  const savedColor1 = localStorage.getItem("color1");
  const savedColor2 = localStorage.getItem("color2");

  if (savedColor1) {
    document.getElementById("color1").value = savedColor1;
    document.getElementById("color1-hex").value =
savedColor1.toUpperCase();
  }

  if (savedColor2) {
    document.getElementById("color2").value = savedColor2;
    document.getElementById("color2-hex").value =
savedColor2.toUpperCase();
  }

  const tokensContainer = document.getElementById("tokens");
  tokensContainer.innerHTML = ""; // Vyčištění kontejneru

  if (savedTokens) {
    // Skupiny tokenů
    const tokenGroups = {
      "Background": {},
      "Primary & Secondary": {},
      "Text": {},
      "Links": {},
      "Buttons": {},
      "Status": {}
    };

    // Rozdělení tokenů do skupin
    for (const [key, value] of Object.entries(savedTokens)) {
      if (key.includes("Background")) {
        tokenGroups["Background"][key] = value;
      } else if (key.includes("Primary") ||
key.includes("Secondary")) {
        tokenGroups["Primary & Secondary"][key] = value;
      } else if (key.includes("Text")) {
        tokenGroups["Text"][key] = value;
      } else if (key.includes("Link")) {
        tokenGroups["Links"][key] = value;
      }
    }
  }
}
```

```

    } else if (key.includes("Button")) {
      tokenGroups["Buttons"][key] = value;
    } else if (key.includes("Status")) {
      tokenGroups["Status"][key] = value;
    }
  }

  // Vytvoření collapsibles
  for (const [group, tokens] of Object.entries(tokenGroups)) {
    if (Object.keys(tokens).length > 0) {
      await createCollapsibleGroup(group, tokens);
    }
  }
} else {
  await generateTokens(); // Pokud nejsou uložené tokeny,
  vygenerujeme nové
}
}

```

3.3 PWA

Abychom z aplikace vytvořili PWA, potřebujeme service worker, který zajišťuje komunikaci s webem a cachování. Dále potřebujeme manifest obsahující metadata pro správné fungování PWA.

3.3.1 Service-worker.js

Service-worker.js je hlavní soubor pro fungování aplikace jako PWA. Zajišťuje správnou instalaci service workeru, kdy se soubory načtené z internetu uloží do cache. Při aktivaci service workeru smaže všechny staré cache. Když se stránka pokusí načíst zdroje, tak se prvně prohledá cache, a až poté se načítá z webu.

```

// Instalace Service Workeru
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => {
        return Promise.all(
          urlsToCache.map(url =>
            fetch(url)
              .then(response => {
                if (!response.ok) {
                  throw new Error(`Failed to fetch
${url}: ${response.statusText}`);
                }
                return cache.put(url, response);
              })
          )
        )
      })
      .catch(error => console.error('Caching error:',
error))
  );
});

```

```

});

// Aktivace a čištění staré cache
self.addEventListener('activate', event => {
  event.waitUntil(
    caches.keys().then(cacheNames =>
      Promise.all(
        cacheNames.filter(name => name !== CACHE_NAME)
          .map(name => caches.delete(name))
      )
    )
  );
});

// Zachycení požadavků
self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request)
      .then(response => response || fetch(event.request))
  );
});

```

3.3.2 Manifest.json

Manifest.json slouží ke správné konfiguraci PWA. Zvolíme název a zkrácený název aplikace pomocí „name“ a „short_name“. V „start_url“ definujeme základní stránku, která se nám zobrazí při spuštění aplikace. Nastavení „display“ na „standalone“ nám zajistí, aby se spuštěná aplikace co nejvíce přibližovala nativní aplikaci bez UI prohlížeče. Každá aplikace by měla mít také ikonu pro její rychlé vyhledání. K tomu slouží dvě ikony, které jsou rozdílné v jejich rozlišení, první má 192x192 px a druhá má 512x512 px.

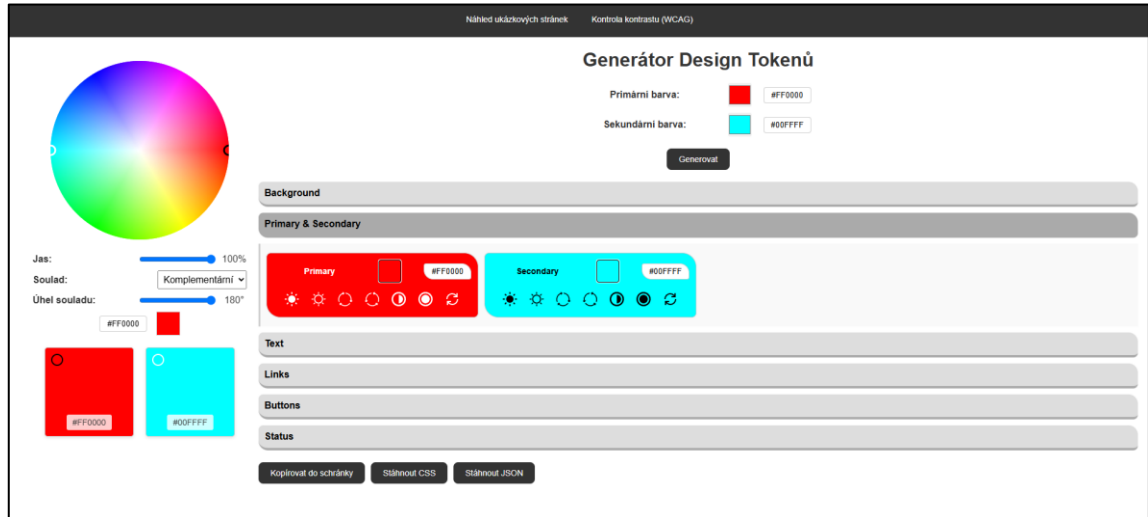
```

{
  "name": "Generátor barevných schémat",
  "short_name": "GBS",
  "start_url": "/index.html",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#000000",
  "icons": [
    {
      "src": "icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}

```

4 UŽIVATELSKÁ PŘÍRUČKA

Uživatelská příručka vás provede, jak pracovat s aplikací pomocí psaného návodu doplněného o obrázky aplikace.

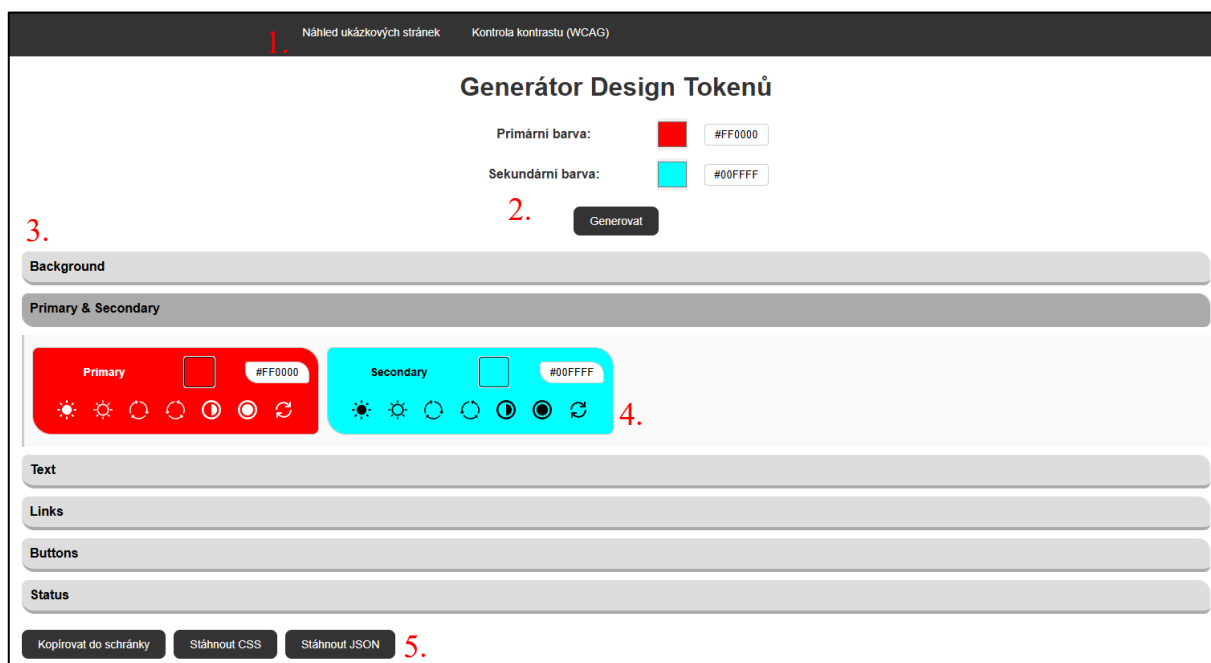


Obrázek 15: Aplikace



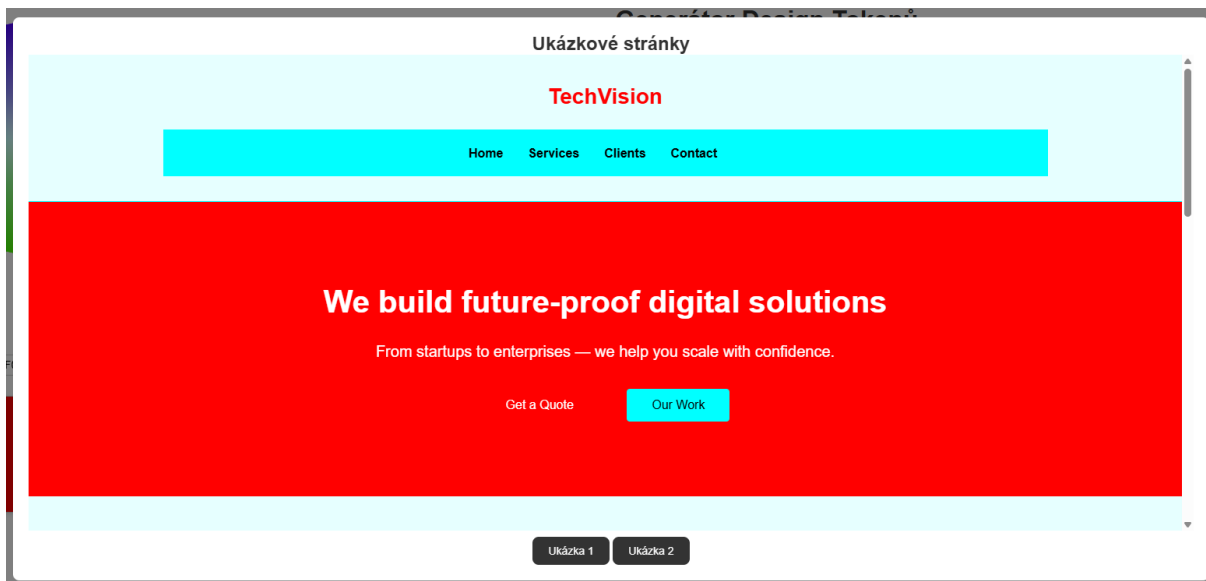
Obrázek 16: Generátor palet

1. Barevný kruh, na kterém lze kliknutím nebo táhnutím zvolit barvu nebo změnit úhel souladu barev
2. Indikátor doplňkových barev
3. Indikátor zvolené barvy
4. Nastavení generování palety obsahující jas, soulad barev a úhel souladu barev
5. Textové pole s kódem zvolené barvy a vizuální prezentace zvolé barvy
6. Paleta vygenerovaných barev s kódy barev a vizuální prezentací



Obrázek 17: Generátor barevných schémat

1. Menu s odkazy na okna s ukázkovými stránkami a kontrolou kontrastu
2. Nastavení primární a sekundární barvy pro generování tokenů
3. Sekce barevného schématu
4. Token barevného schématu s názvem tokenu, color pickerem, kódem barvy a tlačítky pro rychlou úpravu barvy: zvýšení jasu, snížení jasu, změna odstínu, změna odstínu v opačném směru, zvýšení sytosti, snížení sytosti a reset
5. Tlačítka pro export: zkopírování CSS do schránky, stáhnutí CSS souboru, stáhnutí JSON souboru

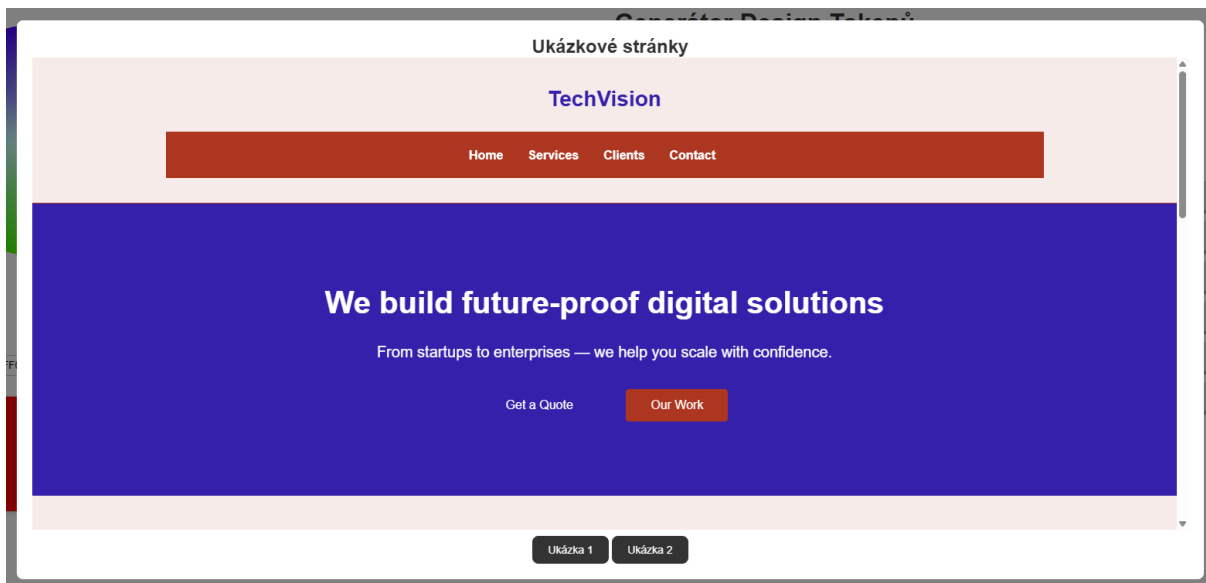


Obrázek 18: Špatné schéma

Token textu	Token pozadí	Ukázkový text	Kontrastní poměr	Výsledek
Text-primary	Background-primary	Ukázkový text	21.00:1	✅ AA
Text-secondary	Background-secondary	Ukázkový text	17.30:1	✅ AA
Text-on-primary	Primary	Ukázkový text	4.00:1	⚠️ AA velký text
Text-on-secondary	Secondary	Ukázkový text	16.75:1	✅ AA
Link-primary	Background-primary	Ukázkový text	4.00:1	⚠️ AA velký text
Link-secondary	Background-secondary	Ukázkový text	1.20:1	❌ WCAG nespĺňuje
Button-primary-text	Button-primary	Ukázkový text	4.00:1	⚠️ AA velký text
Button-secondary-text	Button-secondary	Ukázkový text	16.75:1	✅ AA
Button-disabled-text	Button-disabled	Ukázkový text	5.35:1	✅ AA
Status-positive-text	Status-positive	Ukázkový text	7.24:1	✅ AA

Obrázek 19: Špatné kontrasty

Při generování a tvorbě barevného schématu potřebujeme zajistit dobrou čitelnost. Na obrázcích 18. a 19. můžeme vidět, že vytvořené schéma nespĺňuje přístupnost. Také můžeme vidět, že zvolené barvy nemusí být vždy designově přívětivé, to si musí uživatel pohlídat už sám s pomocí nástroje generující palety. Na obrázcích 20. a 21. je vyobrazeno správné použití aplikace, kontrasty barev splňují standard WCAG a zároveň byly zvoleny barvy, které k sobě ladí.



Obrázek 20: Správné schéma

Kontrola kontrastu (WCAG)

Token textu	Token pozadí	Ukázkový text	Kontrastní poměr	Výsledek
Text-primary	Background-primary	Ukázkový text	21.00:1	✓ AA
Text-secondary	Background-secondary	Ukázkový text	17.21:1	✓ AA
Text-on-primary	Primary	Ukázkový text	10.75:1	✓ AA
Text-on-secondary	Secondary	Ukázkový text	6.32:1	✓ AA
Link-primary	Background-primary	Ukázkový text	10.75:1	✓ AA
Link-secondary	Background-secondary	Ukázkový text	5.42:1	✓ AA
Button-primary-text	Button-primary	Ukázkový text	10.75:1	✓ AA
Button-secondary-text	Button-secondary	Ukázkový text	6.32:1	✓ AA
Button-disabled-text	Button-disabled	Ukázkový text	5.35:1	✓ AA
Status-positive-text	Status-positive	Ukázkový text	7.24:1	✓ AA

Obrázek 21: Správné kontrasty

ZÁVĚR

Zadáním práce bylo navrhnout a implementovat generátor barevných schémat, který uživatelům umožní snadně generovat barevná schémata. V teoretické části bylo popsáno, jak fungují barvy a jak se s barvami pracuje v počítačovém světě. Dále byly popsány návrhové tokeny od různých společností, specifikace barev CSS pro práci s barvami a přístupnost na webu. Byly popsány a porovnány různé aplikace pro generování barev a barevných schémat na trhu, kde bylo zjištěno, že se jedná o odvětví, které ještě není příliš probádané a již existující nástroje mají ještě velký prostor pro zlepšení.

V aplikační části byl vytvořen podrobný návrh aplikace, který by mohl mít šanci ujmout se na trhu. Jsou popsány požadavky, které by mohl mít uživatel takové aplikace. Dále byly popsány případové studie o používání požadované aplikace. Tyto požadavky byly analyzovány a zpracovány do funkčních a nefunkčních požadavků aplikace. Následně byla vytvořena architektura potřebná pro takovou aplikaci, se záměrem udělat ji co nejvíce jednoduchou a univerzální tak, aby mohla být v budoucnu použita s různými technologiemi a frameworky. Na základě všech požadavků a poznatků byl vytvořen grafický návrh celé aplikace s popisem všech komponent.

Z připraveného návrhu byla vytvořena webová aplikace splňující dané zadání. Tato aplikace pomáhá s výběrem barev pomocí jejich souladu a následně generuje barevná schémata, která jsou estetická a připravená pro zakomponování vývojáři.

Pro uživatele této aplikace byla také vytvořena příručka, která uživatele provede aplikací a vysvětlí postupy, jak vytvořit ideální barevné schéma. K příručce jsou dodány obrázky pro rychlou orientaci v návodu.

Aplikace může být jednoduše doplněna o další funkcionality dle potřeby nebo zakomponována do většího systému. Velkým přínosem by mohla být větší standardizace návrhových tokenů nebo doplnění funkcionality vytvářet vlastní tokeny v schématu.

POUŽITÁ LITERATURA

1. JANČOVIČ, Adam. *Vnímání barev*. Diplomová práce. Brno: Masarykova univerzita, 2005.
2. DANNHOFEROVÁ, Jana. *Velká kniha barev: kompletní průvodce pro grafiky, fotografy a designéry*. Brno: Computer Press, 2012. ISBN 978-80-251-3785-7.
3. RÖHRICHOVÁ, Gabriela. *Terapeutické působení barev a světla má vliv na naše zdraví i psychiku, věděli to už pradávne kultury*. Online. ŽENA-IN. 2024. Dostupné z: <https://zena-in.cz/clanek/terapeuticke-pusobeni-barev-a-svetla-ma-vliv-na-nase-zdravi-i-psychiku-vedeli-to-uz-pradavne-kultury>. [cit. 2025-03-05].
4. *Colorterapie – historie*. Online. Via Human. Dostupné z: https://viahuman.cz/leceni-svetlem/?srsltid=AfmBOoqrTKTqIQXsia7neRe_rz-UYZXggw00xaWbC7kMbT0ct7WmH8JC. [cit. 2025-03-05].
5. MEIR, Dana. *Color psychology: how colors impact human behavior and emotion*. Online. In: Wix. 2020. Dostupné z: <https://www.wix.com/blog/color-psychology>. [cit. 2025-01-02].
6. PANTONE. *What are the different color models?* Online. Pantone. 2007. Dostupné z: <https://www.pantone.com/articles/color-fundamentals/color-models-explained>. [cit. 2024-12-04].
7. ADOBE. *Adobe Color*. Online. Dostupné z: <https://color.adobe.com/cs/>. [cit. 2024-12-05].
8. *Design tokens*. Online. In: ADOBE. Spectrum. Dostupné z: <https://spectrum.adobe.com/page/design-tokens/>. [cit. 2024-12-04].
9. *Color roles*. Online. In: GOOGLE. Material Design. Dostupné z: <https://m3.material.io/styles/color/roles>. [cit. 2024-12-07].
10. *Color*. Online. In: IBM. Carbon Design System. Dostupné z: <https://carbondesignsystem.com/elements/color/overview/>. [cit. 2024-12-07].
11. *Design tokens*. Online. In: ATLISSIAN. Atlasian Design System. Dostupné z: <https://atlassian.design/tokens/design-tokens>. [cit. 2025-01-08].
12. W3C. *CSS Color Module Level 3*. Online. W3C. 2022. Dostupné z: <https://www.w3.org/TR/css-color-3/>. [cit. 2024-12-27].
13. W3C. *CSS Color Module Level 4*. Online. W3C. 2024. Dostupné z: <https://www.w3.org/TR/css-color-4/>. [cit. 2024-12-27].

14. W3C. *CSS Color Module Level 5*. Online. W3C. 2024. Dostupné z: <https://www.w3.org/TR/css-color-5/>. [cit. 2025-01-08].
15. W3C. *CSS Color Module Level 6*. Online. W3C. CSS Working Group Editor Drafts. 2024. Dostupné z: <https://drafts.csswg.org/css-color-6/>. [cit. 2025-01-09].
16. W3C. *Accessibility Principles*. Online. W3C. 2024. Dostupné z: <https://www.w3.org/WAI/fundamentals/accessibility-principles/>. [cit. 2024-12-13].
17. RANK, Sojin. *Accessible Colors: A Complete Guide for Web Design*. Online. In: AudioEye. 2024. Dostupné z: <https://www.audioeye.com/post/accessible-colors/>. [cit. 2024-12-13].

SEZNAM OBRÁZKŮ

1. *A diagram showing the visible color spectrum.* Online. In: Live Science. 2022. Dostupné z: <https://www.livescience.com/50678-visible-light.html>. [cit. 2025-03-12].
2. *Understating the "HSL" color expression system.* Online. In: CANON. Canon. Dostupné z: <https://global.canon/en/imaging/picturestyle/editor/matters05.html>. [cit. 2024-12-04].
3. *RGB vs CMYK: What's the difference?* Online. In: VistaPrint. 2024. Dostupné z: <https://www.vistaprint.com/hub/correct-file-formats-rgb-and-cmyk>. [cit. 2025-01-09].
4. DANNHOFFEROVÁ, Jana. *Velká kniha barev: kompletní průvodce pro grafiky, fotografie a designéry.* Brno: Computer Press, 2012. ISBN 978-80-251-3785-7.
5. *Color Theory in Design: How to Choose the Right Palette for Your Brand.* Online. In: Franco. Dostupné z: <https://franco.com/blog/creative/color-theory-in-design-how-to-choose-the-right-palette-for-your-brand/>. [cit. 2025-01-09].
6. *Design tokens.* Online. In: ADOBE. Spectrum. Dostupné z: <https://spectrum.adobe.com/page/design-tokens/>. [cit. 2024-12-04].
7. *Color roles.* Online. In: GOOGLE. Material Design. Dostupné z: <https://m3.material.io/styles/color/roles>. [cit. 2024-12-07].
8. *Color.* Online. In: IBM. Carbon Design System. Dostupné z: <https://carbondesignsystem.com/elements/color/overview/>. [cit. 2024-12-07].
9. Případy užití
10. Struktura aplikace
11. Návrh aplikace
12. Návrh generátoru palety
13. Návrh generátoru barevných schémat
14. Návrh tabulky kontrastů
15. Aplikace
16. Generátor palet
17. Generátor barevných schémat
18. Špatné schéma
19. Špatné kontrasty
20. Správné schéma
21. Správné kontrasty

SEZNAM PŘÍLOH

Příloha A: Zdrojový kód aplikace

PŘÍLOHA A: Zdrojový kód aplikace

Zdrojový kód je uložen v archivu s názvem:

HnatF_GeneratorBarevnych_LC_aplikace_2025.zip