

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Generování zdrojového kódu front–end aplikace z wireframe modelu
Lukáš Bajer

Bakalářská práce
2023

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Lukáš Bajer**
Osobní číslo: **I19063**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Téma práce: **Generování zdrojového kódu front-end aplikace z wireframe modelu.**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

V rámci bakalářské práce bude provedena rešerše aktuálních nástrojů pro designování front-endu aplikací a jejich porovnání. Student se dále zaměří na nástroj Figma, předvede samotnou aplikaci a plugíny do aplikace tvořené uživateli, prozkoumá plugíny zaměřené na generování React kódu z wireframe modelu a nakonec vytvoří jednoduchý plugin pro odzkoušení možnosti generování React kódu z komponent knihovny Material UI (MUI).

Rozsah pracovní zprávy: **min. 30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Designing and Prototyping Interfaces with Figma. 1. London: Packt Publishing, 2022. ISBN 180056418X.

Advanced Web Development with React: SSR and PWA with Next.js using React with advanced concepts. 1. London: BPB PUBN, 2020. ISBN 9389423597.

ŽÁRA, Ondřej. JavaScript: programátorské techniky a webové technologie. Brno: Computer Press, 2015. ISBN 978-80-251-4573-9.

Vedoucí bakalářské práce: **Ing. Jan Panuš, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **16. prosince 2022**
Termín odevzdání bakalářské práce: **12. května 2023**

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

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

V Pardubicích dne 28. února 2023

Prohlašuji:

Tuto práci 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., autorský zákon, 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. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 11. 5. 2023

Lukáš Bajer

PODĚKOVÁNÍ

Tímto bych rád poděkoval svému vedoucímu Ing. Janu Panušovi, Ph.D. za vedení mé práce. Dále bych chtěl poděkovat svým kolegům v práci, za jejich cenné rady a podporu při řešení této práce. Velké poděkování patří také mé rodině, která mě v mém studiu podporovala a stála po mém boku.

ANOTACE

Bakalářská práce se zaměřuje na vytvoření pluginu pro aplikaci Figma, který umožňuje generovat kód v jazyce JavaScript z návrhů, které využívají komponenty knihovny MUI, které jsou stylizované společností Generali Česká pojišťovna. Teoretická část nejprve představí pojmy UI a UX design a problematiku návrhu grafického modelu. Dále bude představeno a porovnáno několik existujících nástrojů pro tvorbu grafických návrhů. Zbytek teoretické části se bude věnovat aplikaci Figma, která bude podrobně představena a rozebere problematiku pluginů. V praktické části budou nejprve představeny technologie použité při vývoji pluginu a poté bude popsán postup návrhu, vývoje a testování pluginu.

KLÍČOVÁ SLOVA

UX design, UI design, Figma, Figma plugin, Sketch, Adobe XD, InVision, Generátor React kódu.

TITLE

Generating the source code of the front–end application from the wireframe model

ANNOTATION

The bachelor's thesis focuses on the creation of a plugin for the Figma application, which allows generating JavaScript code from designs that use components of the MUI library, which are stylized by the company Generali Česká pojišťovna. The theoretical part will first introduce the concepts of UI and UX design and the issue of graphic model design. Next, several existing tools for creating graphic designs will be presented and compared. The rest of the theoretical part will be devoted to the Figma application, which will be presented in detail and the issue of plugins will be discussed. The section will first introduce the technologies used in the development of the plugin, and then describe the process of design, development and practical testing of the plugin.

KEYWORDS

UX design, UI design, Figma, Figma plugin, Sketch, Adobe XD, InVision, Generator of React code.

OBSAH

Seznam obrázků	8
Seznam zkratk	9
Úvod	10
1 Problematika návrhu grafické designu.....	11
1.1 Definice pojmu UI a UX design	11
1.2 Role UI designu v procesu vývoje aplikace	11
1.3 Příklad UI/UX designu na webové stránce	12
2 Rešerše nástrojů pro design front-endu aplikací.....	13
2.1 Přehled nástrojů	13
2.2 Shrnutí a porovnání nástrojů.....	14
3 Figma.....	15
3.1 Úvod do Figma	15
3.2 Klíčové vlastnosti a funkce	15
3.3 Výhody	17
3.4 Nevýhody	18
3.5 Pluginy	18
3.5.1 Pluginy zaměřené na generování React kódu	19
3.6 Figma pro vývojáře	22
3.7 Figma pro UI/UX designéry	22
4 Použité technologie	23
4.1 Programovací jazyky.....	23
4.2 Značkovací a stylové jazyky.....	24
4.3 Frameworky a knihovny.....	25
4.4 Vývojová prostředí.....	26
4.5 Programy	28
5 Vývoj pluginu.....	29
5.1 Stanovení cílů	29
5.2 Tvorba a nastavení projektu	29
5.3 Návrh pluginu	32
5.4 Vývoj pluginu	35
6 Testování pluginu	38
Závěr.....	39
7 Použitá literatura.....	40

SEZNAM OBRÁZKŮ

Obrázek 1: Rozhraní pluginu Anima. Zdroj vlastní.....	20
Obrázek 2: Rozhraní pluginu DhiWise. Zdroj vlastní.....	20
Obrázek 3: Rozhraní pluginu TeleportHQ. Zdroj vlastní.....	21
Obrázek 4: Generování šablony pluginu. Zdroj vlastní.....	30
Obrázek 5: Souborová struktura projektu. Zdroj vlastní.	31
Obrázek 6: Návrh UI + možnosti výstupu pluginu. Zdroj vlastní.	33
Obrázek 7: Diagram návrhu pluginu. Zdroj vlastní.	35

SEZNAM ZKRATEK

PDF	Portable Document Format
API	Application Programming Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
UI	User Interface
UX	User Experience
JSX	JavaScript XML
URL	Uniform Resource Locator
OS	Operating System
IDE	Integrated Development Environment
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation

ÚVOD

V dnešní době, kdy se vše odehrává na internetu, funguje každý malý i velký podnik nějakým způsobem na internetu. Může to být prostřednictvím sociálních sítí, webových stránek nebo webových aplikací. Vzhledem k tomu, že se vše stále více vyhledává na internetu, je pro podniky velmi důležité mít co nejlepší webové stránky, aby oslovily co nejvíce potenciálních zákazníků. Návrh, implementace a nasazení webové stránky nejsou lehkým úkolem, a tak tento proces trvá dlouho a může být velmi drahý. Vzhledem k obrovskému množství nových stránek, které se každý rok na internetu objevují, se hledají způsoby, jak tento proces urychlit a zlevnit.

Proces návrhu webové stránky, zejména rozsáhlých webových aplikací, je zdoluhavým a iterativním procesem, který velmi ulehčuje aplikace pro design front–endů. Další možností, jak urychlit a zlevnit vývoj webové stránky, je přímo ze těchto návrhů strojově generovat zdrojový kód webové stránky, což zautomatizuje velkou část implementace. To umožňuje například aplikace Figma pomocí uživatelsky tvořených pluginů, které fungují přímo v aplikaci.

Hlavním cílem této práce je vytvořit plugin pro aplikaci Figma, který umožní strojové generování zdrojového kódu v jazyce JavaScript z návrhu vytvořeného v aplikaci Figma. Tento plugin je určený pro návrhy vytvořené s využitím knihovny MUI a jeho úkolem je z velké části automatizovat samotnou implementaci návrhu.

Teoretická část práce se zaměřuje na představení problematiky návrhu grafického designu. Nejprve jsou vysvětleny základní pojmy a poté je popsán proces návrhu. Další část se věnuje rešerši nástrojů pro design front–endů aplikací, kde jsou popsány populární nástroje a porovnány mezi sebou. Následně je v práci představen a detailně prozkoumán nástroj Figma, včetně jeho klíčových funkcí. Současně jsou představena existující řešení pluginů, které umožňují generování kódu z návrhu. Poslední částí je představení technologií použitých při vývoji tohoto konkrétního pluginu.

Praktická část nejprve popisuje postup tvorby projektu pro tento plugin a představuje nastavení, které je potřeba udělat. Následně popisuje vytvořenou souborovou strukturu. Další část je věnována návrhu uživatelského rozhraní a výkonné části pluginu. Poslední část práce se zaměřuje na popis implementace jednotlivých důležitých modulů pluginu a tomu, jak plugin funguje.

1 PROBLEMATIKA NÁVRHU GRAFICKÉ DESIGNU

1.1 Definice pojmu UI a UX design

UI/UX design je termín, který popisuje dva odlišné aspekty návrhu digitálního produktu. UI je zkratka pro výraz user interface neboli uživatelské rozhraní. Tento pojem se týká celkového vzhledu produktu, jako jsou vizuální prvky v aplikacích, programech, na webových stránkách nebo v jiných digitálních produktech. Jedná se například o tlačítka, ikony, ilustrace a další grafické prvky, volbu textových fontů a barev. Zkratka UX znamená user experience neboli uživatelský zážitek. Zaměřuje se na to, jak uživatel interaguje s produktem, jaký pocit to v něm vyvolává, jaký má k produktu vztah a jak si užívá jeho používání. Konkrétně se jedná například o rozložení grafických komponentů, přehlednost a snadnou navigaci. To je důležité, protože čím lepší zážitek má uživatel s produktem, tím spíše ho bude používat do budoucna. Ačkoli se na první pohled tyto dva pojmy mohou zdát velmi podobné, každý z nich zodpovídá za jiné funkcionality produktu. Nicméně v konečném důsledku je kvalita výsledného produktu závislá na kvalitním zpracování jak UI, tak UX designu.

1.2 Role UI designu v procesu vývoje aplikace

Definování kontextu a uživatelů

Prvním krokem v jakémkoli procesu UI designu je porozumění myšlenky produktu a jeho uživatelům. Vývojový tým musí zjistit, pro jaké uživatele je produkt vyvíjen, jaké jsou jejich potřeby a jaký je nejlepší způsob, jak jim vyhovět. S těmito informacemi je možné začít vytvářet prototyp produktu a wireframe modely, ze kterých nakonec vznikne konečný produkt. Tyto počáteční informace jsou velmi důležité pro vystižení potřeb uživatelů a poskytnutí kvalitního uživatelského prožitku v konečném produktu.

Návrh vizuálního rozhraní

V této fázi vývoje prototypu začíná tvorba rozhraní, které je klíčovým krokem, protože se od něj bude odvíjet celý budoucí vývoj. Nejprve je nutné stanovit cíle rozhraní, jako například to, co uživatelé budou moci dělat a jak budou s rozhraním interagovat. Na základě těchto cílů lze začít vytvářet návrh. (1), (2)

Návrh navigačních a systémových rozhraní

Tento krok se zaměřuje na samotný návrh uživatelského rozhraní, který typicky řeší otázky týkající se vzhledu, navigace a systémového rozhraní, stejně jako vliv systému na uživatele. V případě navigace je cílem vytvořit návrh, který poskytne uživatelům snadné a jednoduché navigování v systému. Systémový design se soustředí na to, aby celý systém působil jednotně a soudržně. Nakonec probíhá testování vytvořeného návrhu na uživatelích a následné vyhodnocení získaných dat, aby se zjistilo, zda navržené rozhraní splňuje uživatelské potřeby.

Návrh prototypu výrobku

Tento proces se zaměřuje na tvorbu prototypů a wireframe modelů s vysokou přesností, které již dobře odráží výsledný produkt. Detailnost jednotlivých modelů a prototypů se odvíjí od toho, jak daleko je produkt ve vývoji. Tyto modely již obsahují barvy, obrázky, ikony a další prvky. Hlavním cílem tvorby těchto modelů je zjištění proveditelnosti návrhu a získání časné zpětné vazby od uživatelů.

Testování použitelnosti prototypu

Pevně spojeno s návrhem prototypu produktu je jeho testování, které je nezbytné pro odhalení případných problémů před uvedením produktu na trh. Mezi běžné metody testování patří například žádost na uživatele o splnění zadaného úkolu při používání prototypu návrhu. Tento postup umožňuje odhalit nejasné a matoucí prvky. Důležitým faktorem je také uživatelský prožitek, aby uživatel nebyl frustrován nebo znuděn některými částmi produktu. Díky provedení těchto kroků lze zajistit uživatelsky přívětivý a úspěšný produkt.

1.3 Příklad UI/UX designu na webové stránce

Představme si například textovou aplikaci, která bude mít vysokou vizuální úroveň, ale odeslání zprávy bude trvat nepřiměřeně dlouho. V takovém případě bude konverzace jedním dlouhým čekáním na odeslání a uživatelský zážitek bude velmi špatný. Uživatelé pak budou raději používat jinou aplikaci. Na druhou stranu může být aplikace velmi dobře optimalizovaná a vyladěná, ale vizuálně nebude přívětivá, text bude nepřehledný, barvy nebudou ladit a celkový vizuální zážitek bude špatný. Tento příklad ukazuje, že pro tvorbu úspěšného produktu je nutné kvalitně navrhnout jak UI, tak UX. (3)

2 REŠERŠE NÁSTROJŮ PRO DESIGN FRONT-ENDU APLIKACÍ

UI/UX design je prvním krokem vývoje jakéhokoli digitálního produktu, ať už se jedná o webové stránky, mobilní nebo desktopové aplikace. Kvalitní design poskytuje uživatelům pozitivní zkušenost s produktem a zvyšuje tak jejich spokojenost, což může vést k další interakci a větší loajalitě k produktu. V dnešní době existuje mnoho takových nástrojů, od jednoduchých řešení až po pokročilé a profesionální aplikace. Mezi nejpopulárnější aplikace patří Figma, Sketch, Adobe XD a InVision, které jsou v této práci více rozebrány. (4) Nicméně existuje mnoho dalších jako například Marvel, UXPin, Framer nebo jiné. Každý z těchto nástrojů má své výhody a nevýhody, které je třeba zohlednit před tvorbou nového projektu, aby odpovídal jeho požadavkům.

2.1 Přehled nástrojů

Nástroj Sketch je aplikace zaměřená na design a vektorovou grafiku, která je populární mezi designéry pro tvorbu návrhů, ikon a dalších webových elementů. Aplikace disponuje rozsáhlou kolekcí editačních nástrojů, možností prototypování, podporuje sdílené knihovny, styly, komponenty a znovupoužitelné prvky. Navíc do velké míry umožňuje přizpůsobovat fonty. Sketch je možné používat jak offline, tak online. Při práci online lze pracovat společně s ostatními designéry v reálném čase na stejném projektu. Aplikace ukládá soubory návrhů do vlastního formátu, který je podporován i jinými nástroji jako Adobe Photoshop a Illustrator, lze zvolit i jiné formáty, nebo návrh exportovat do různých obrazových formátů. Samotná aplikace Sketch je dostupná pouze pro operační systém MacOS, a pro ostatní platformy je k dispozici webová aplikace, která umožňuje sdílení, komentování a předávání podkladů mezi týmy. Aplikace Sketch existuje od roku 2010 a byla vytvořena nezávislou společností. Je k dispozici s třicetidenní zkušební lhůtou, po jejímž vypršení je nutné mít předplatné za každého editora. Pro studenty a organizace existují zvýhodněné plány. Designéři mohou do projektů pozvat neomezené množství lidí, kteří mohou prohlížet, komentovat a stahovat součásti návrhů zdarma. (5) (6)

Aplikace Adobe XD je navržena s důrazem na výkon a umožňuje vytváření věrných návrhů, prototypů a wireframe modelů pro navrhovanou aplikaci. Je snadno použitelná a založená na vektorovém prostředí. Poskytuje nástroje pro spolupráci a vytváření návrhů, a pomáhá vývojářům a designérům skrze celý proces návrhu až po předání návrhů vývojářům.

Aplikace disponuje funkcí kolaborace v reálném čase a usnadňuje práci díky časově úsporným funkcím a pracovním postupům. Podporuje znovupoužitelné komponenty, sady odsazení, funkci opakovatelné mřížky, 3D transformace, sdílené knihovny, responzivní návrhy a pluginy. Sdílení a předávání souborů funguje skrze sdílení odkazů a aplikace také umožňuje komunikaci přímo v pracovním prostoru pomocí komentářů, špendlíků a zmínek. Adobe XD je k dispozici pro operační systémy Windows a macOS a je kompatibilní mezi těmito platformami. Aplikace však neobsahuje webovou aplikaci, která by umožňovala její použití bez nutnosti stahování a instalace. Je součástí Adobe Creative Cloud a vyžaduje placení měsíčních poplatků. (7)

Aplikace InVision je cloudové řešení pro tvorbu, testování a experimentování s designem. Designéři a vývojáři ji využívají pro tvorbu designů, wireframů a prototypů navrhovaných aplikací. (8) InVision disponuje funkcemi prezentování prototypů v reálném čase i na mobilních zařízeních, sdílení prototypů a podporu desktopových i mobilních gest v prototypu. Dalšími funkcemi jsou plánování v reálném čase skrze zabudovaný plánovač, komentování návrhů v reálném čase skrze webovou aplikaci a obecně spolupráce designerů na jednom projektu. InVision nabízí široké možnosti integrace s ostatními aplikacemi. (9) Aplikace je zdarma pro jednotlivce. Pro větší skupiny je k dispozici měsíční předplatné a pro společnosti a organizace nabízí speciální plány. (10)

2.2 Shrnutí a porovnání nástrojů

Každá aplikace pro tvorbu UI/UX designu je jiná a každá má své plusy a mínusy. Aplikace Sketch je známá svou širokou škálou nástrojů, podporou návrhových systémů, sdílených knihoven a komponentů. Umožňuje integraci s ostatními nástroji pro design a umožňuje kolaboraci v reálném čase. Aplikace je však dostupná pouze pro operační systém macOS, neobsahuje pluginy je placená a nedisponuje webovým rozhraním. Adobe XD také nabízí široké spektrum funkcí a nástrojů, k tomu navíc podporuje pluginy, sdílené knihovny, znovupoužitelné komponenty a umožňuje kolaboraci v reálném čase. Její nevýhody jsou integrace pouze s Adobe produkty, absence webového rozhraní a fakt že je placená. InVision je oproti tomu nabízí zdarma verzi, dobrou integraci s ostatními produkty a disponuje webovou aplikací. Na druhou stranu obsahuje méně nástrojů, jelikož se zaměřuje spíše na prototypování a kolaboraci. Aplikace může být také náročnější na naučení, a pro větší týmy je placená. Shrnutím má každá aplikace své výhody, nevýhody a specifické případy použití. Pro některé projekty může být dobrá ta, pro jiné ona což záleží na požadavcích konkrétních návrhů a projektů.

3 FIGMA

Tato kapitola je zaměřena výhradně na grafický nástroj Figma, který představí. Dále se bude věnovat funkcím, které tato aplikace nabízí, pluginům, které rozšiřují její funkcionality. Současně také představí pluginy, které řeší stejný problém jako tato práce. Následně zhodnotí výhody a nevýhody této aplikace, a nakonec popíše pohled na ni z pohledu designéra a vývojáře.

3.1 Úvod do Figmy

Figma je cloudový kolaborační design nástroj pro úpravu grafiky, návrh uživatelského rozhraní a UX. Umožňuje provádět jakoukoli grafickou práci od grafických návrhů, tvorby wireframů webových stránek, navrhování mobilních a desktopových aplikací, prototypování návrhů až po tvorbu příspěvků na sociální sítě. Od ostatních alternativ se odlišuje tím, že se jedná o webovou aplikaci, která funguje přímo v prohlížeči počítače, nebo to že její základní verze je zcela zdarma. Významnou částí Figmy je také možnost spolupracovat na projektech v reálném čase ve více lidech. Aplikace poskytuje velké množství funkcí, díky kterým umožňuje tvořit komplexní grafické návrhy, funkce, které nejsou v základní aplikaci aplikace doplňují pluginy, kterých existují tisíce jak přímo od Figmy, tak od rozsáhlé komunity uživatelů. (11), (12), (13)

3.2 Klíčové vlastnosti a funkce

Dostupnost a kolaborace

Figma je cloudová webová aplikace, která funguje samostatně v prohlížeči. Tento způsob umožňuje aplikaci Figma fungovat s minimálními nároky na hardware počítače. Pokud má uživatel internetové připojení a internetový prohlížeč, aplikace Figma funguje bez potíží. Jelikož se jedná o cloudovou aplikaci, projekty jsou ukládány a stahovány z internetu a existuje vždy pouze jedna verze. To přináší výhodu v tom, že odpadá starost s různými verzemi a vývojáři mají jistotu, že pracují s nejnovější verzí. Projekty lze stáhnout nebo otevřít předem a poté na nich lze pracovat bez internetového připojení. Nicméně, pokud uživatel pracuje bez připojení k internetu, projekt nebude uložen v cloudu. Figma umožňuje spolupráci v reálném čase až s 200 uživateli najednou na jednom projektu. (13) Do jednoho projektu může být pozváno nanejvýše 500 uživatelů. (14)

Flexibilita

Z pohledu kompletního vývoje a testování nového návrhu aplikace je Figma velmi flexibilní nástroj. V počátečních fázích vývoje umožňuje designerům spolupracovat přímo v aplikaci a přicházet s novými nápady díky funkci živého audia a chatu, které jsou integrovány přímo do pracovního prostředí. tyto funkce umožňují kreativní a interaktivní vytváření prvních návrhů. Aplikace Figma dále umožňuje iterativní práci na návrhu, což znamená, že designéři se mohou průběžně vracet k návrhu a postupně jej vylepšovat. To umožňuje vytváření pokročilejších wireframů až do stádia hotového a komplexního návrhu. Celý tento proces probíhá v aplikaci Figma, což zajišťuje efektivitu a plynulost vývojového cyklu. (15)

Prototypování a testování

Aplikace disponuje funkcí zvanou prototypování, která umožňuje vytvářet interaktivní aplikace ze statických návrhů bez nutnosti jakéhokoli programování. Zprovoznění této funkce je velmi snadné, stačí vybrat záložku prototypování, zvolit prvek, ke kterému chceme přidat interakce, animace nebo přechody, a kliknout na ikonu plus v sekci interakce v záložce prototypování. Otevře se tabulka, která umožňuje definovat jednotlivé interakce na prvky. Nejprve je potřeba zvolit typ interakce, jako například kliknutí, tažení nebo podržení myši nad prvkem, a následně definovat, co se má po této interakci stát, například navigovat na jinou obrazovku, otevřít modální okno nebo zobrazit hlášku. Nakonec vybereme, ke kterému prvku chceme tuto interakci napojit po jejím spuštění. Tímto způsobem lze tvořit komplexní prototypy, které odrážejí designérovu vizi. Prototypy jsou velmi užitečné například při prezentování návrhu nebo při testování v rané fázi vývoje. Hotový prototyp lze spustit kliknutím na ikonu šipky v pravém horním rohu aplikace v panelu nástrojů. Nyní se prototyp chová jako samotná aplikace s definovanými interakcemi. Vytvořený prototyp je možné sdílet například s testovacími uživateli pomocí modrého tlačítka "Sdílet prototyp" v pravém horním rohu aplikace v panelu nástrojů. Tím se vytvoří hypertextový odkaz, který lze nasdílet testovacím uživatelům, kteří mohou prototyp libovolně testovat a interagovat s ním. (16), (15)

Návrhové systémy

Návrhový systém je kolekce znovupoužitelných komponent vycházejících z jasných pravidel a může být použit k tvorbě neomezeného množství aplikací. Součástí designového systému mohou být mimo komponent a stylů také samotný kód nebo nejlepší praktiky pro jeho používání. (17)

Návrhový systém je tedy takový soubor návrhů jednotlivých komponent, které, ať jsou použity kdekoli, vždy vycházejí z tohoto návrhu a s jeho změnou se mění i všechny jeho instance. Aplikace Figma umožňuje tyto systémy vytvářet a pohodlně je exportovat do souboru nebo publikovat jako knihovny. Tyto knihovny pak poskytují všechny předpřipravené komponenty ihned k použití a zároveň se synchronizují s původním návrhem, takže vždy existuje pouze jedna verze. (15)

Sdílení

Předávání návrhů a wireframů mezi designéry a vývojáři se výrazně zjednodušuje díky aplikaci Figma. Tento nástroj umožňuje práci až pro 200 uživatelů najednou v cloudu, což znamená, že není třeba fyzicky předávat projekty. Dokonce i v bezplatné verzi Figmy mohou vývojáři prohlížet návrhy a získat užitečné informace pro vývoj. Protože existuje pouze jedna verze v cloudu, do které se mohou připojit všichni pozvaní, není potřeba předávat návrhy jiným způsobem. (15) Sdílení projektů a prototypů aplikace se provádí pomocí tlačítka pro sdílení, které se obvykle nachází v pravém horním rohu. Po kliknutí na něj se otevře modální okno, které umožňuje nastavit přístupová práva a poté se vygeneruje odkaz, který pozvaným lidem poskytne přístup k projektu s určenými oprávněními pro přístup a editaci. (18)

Mobilní aplikace

Mobilní aplikace Figma umožňuje prohlížení a přístup k souborům, testování prototypů a zrcadlení obrazovek z počítače na mobilní zařízení. Díky těmto funkcím má široké využití, například pro prezentaci prototypů přímo na mobilních zařízeních, prohlížení návrhů kdekoliv na světě s internetovým připojením nebo testování návrhů na mobilních zařízeních s různými rozměry. (19) Mobilní aplikace je zdarma, což umožňuje každému zúčastněnému zobrazit si prototyp na vlastním mobilním zařízení a otestovat návrh aplikace po svém. (13) Výhody mobilní aplikace Figma zahrnují ulehčení práce designérům při testování návrhu na různých zařízeních a celkové zjednodušení a zpříjemnění procesu designu pro všechny zúčastněné.

3.3 Výhody

Figma má mnoho výhod oproti konkurenčním aplikacím. Jednou z největších výhod je možnost práce v reálném čase na jednom projektu s více lidmi, což šetří mnoho hodin práce. Navíc aplikace umožňuje provádět mnoho různých úkolů v jedné aplikaci, jako například navrhování, prototypování a předávání podkladů. Figma je první aplikací, která disponuje všemi těmito vlastnostmi. Dříve se na to muselo používat několik různých aplikací. Figma funguje v cloudu, což umožňuje přístup k souborům odkudkoli na světě.

Nicméně, k používání webové aplikace je nutné mít připojení k internetu. Další výhodou je základní neplacený plán, který umožňuje využívat Figma pro malé projekty zcela zdarma. Prohlížení kteréhokoli projektu a získávání užitečných informací pro vývoj je s tímto plánem také možné, díky čemuž si vývojáři nemusí platit licence. Funkce, které Figma standardně nepodporuje, lze doplnit pluginy, které jsou vytvářeny uživateli. Existuje tisíce pluginů, které výrazně rozšiřují funkčnost a možnosti práce s aplikací. Figma je k dispozici jako desktopová aplikace pro macOS a Windows a jako webová aplikace, která lze spustit v prohlížeči. Díky neplacené verzi je začít s používáním Figma velmi snadné a může ji používat kdokoli. (20), (21)

3.4 Nevýhody

Webová aplikace Figma je cloudová aplikace a jednou z nevýhod tohoto typu aplikací je závislost na internetovém připojení. Pokud uživatel není online, aplikace nefunguje. Figma však nabízí také desktopovou aplikaci, která funguje offline, ale ke svému běhu potřebuje počítač s vyššími výpočetními schopnostmi. Samotná aplikace nabízí menší počet editačních nástrojů v porovnání s konkurenčními aplikacemi, což může být pro někoho nevýhodou, pokud je zvyklý na širší spektrum nástrojů. Další nevýhodou aplikace je problém s integrací a spoluprací s jinými aplikacemi pro tvorbu grafiky a grafických návrhů. (20)

3.5 Pluginy

Pluginy jsou programy nebo aplikace vytvořené komunitou, které rozšiřují funkcionalitu aplikace Figma. Tyto pluginy umožňují uživatelům provádět jednu nebo více uživatelsky definovaných akcí přímo v souborech Figma. V důsledku toho je možné přizpůsobit prostředí aplikace, nebo zjednodušit a zefektivnit rutinní pracovní postupy. Stejně jako aplikace Figma, jsou pluginy založeny na webových technologiích. To znamená, že fungují nezávisle na operačních systémech a lze je spouštět prostřednictvím webového prohlížeče na libovolných operačních systémech jako například Windows, MacOS, Linux, ChromeOS a případně i jiných. Pro vývoj pluginů je třeba mít znalosti webových technologií, jako je JavaScript, HTML a CSS, které jsou pro tvorbu pluginů využívány. Komunikace mezi pluginy a aplikací Figma je zajištěna pomocí Plugin API, které umožňuje využívání i externích webových rozhraní API. Pluginy mohou být vytvořeny pro zpracování jedné operace, v takovém případě se plugin spustí, provede operaci a následně se ukončí. Druhý typ pluginu disponuje uživatelským rozhraním, které se zobrazí po spuštění pluginu. Toto rozhraní je definováno pomocí HTML struktury a umožňuje uživateli ovládat chování pluginu.

Plugin API poskytuje vývojářům funkce čtení a zápisu, což umožňuje prohlížení, vytváření a úpravu obsahu návrhu v aplikaci. (22)

Co pluginy neumožňují

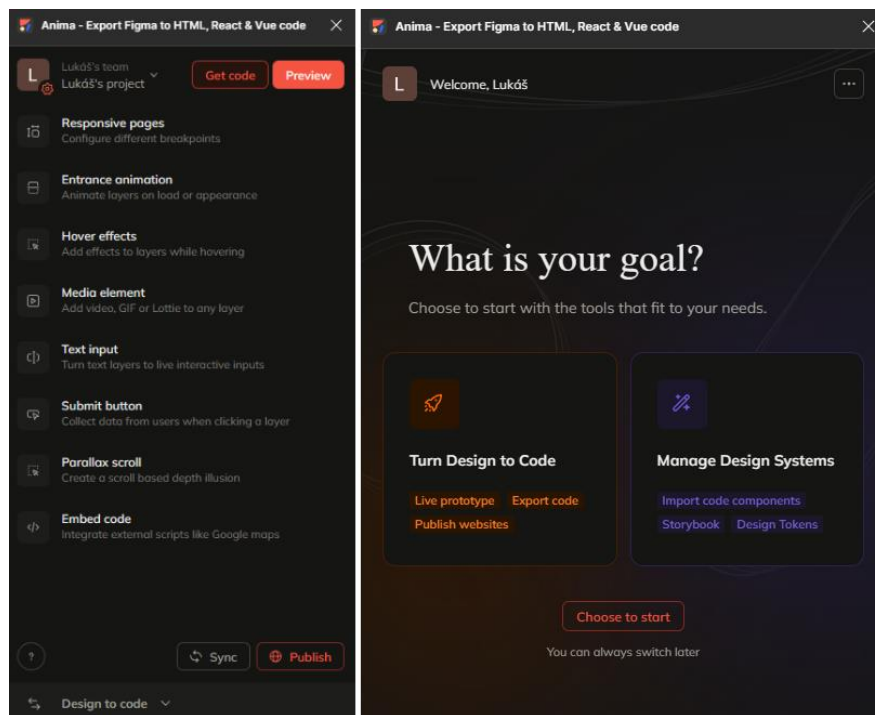
Většina funkcí v plugin API je přístupná skrze globální objekt jménem Figma. Nicméně tento objekt umožňuje přístup pouze k prvkům, které se nacházejí na plátně aplikace, nikoli mimo něj. Proto pomocí pluginů není možné přebírat styly ani komponenty z týmových nebo organizačních knihoven, přístup mají pouze k lokálním instancím. Dále pluginy neumožňují přístup k externím nebo webovým písmům, která jsou přístupná pomocí adresy URL. Přístup mají pouze k lokálním písmům nebo písmům sdíleným v rámci organizace. Plugin nemá přístup k dalším metadatům souboru, jako je umístění souboru, tým, oprávnění, uživatelé nebo jakékoli komentáře, které soubor obsahuje. Dále není možné vytvářet pluginy, které běží na pozadí. A není možné spouštět více pluginů nebo akcí najednou. (22)

3.5.1 Pluginy zaměřené na generování React kódu

Když je návrh v aplikaci Figma hotový, prototyp vytvořen a vše pečlivě otestováno, nastává čas implementace návrhu. Samotný návrh aplikace dává vývojáři dobrý základ pro zahájení implementace funkční webové aplikace. Tento úkol obvykle řeší vývojáři a není to snadná práce. Vývojáři často hlouběji nerozumí navržené aplikaci nebo musí vytvářet všechny komponenty od nuly, což může být časově náročné. V těchto případech je užitečné mít připravený návrhový systém, který umožňuje vývojářům i designérům pracovat na produktu jako na sestavovací sadě s předem připravenými komponentami. Pokud takový systém není k dispozici, musí vývojář ručně převádět návrh do kódu, což může být složité a časově náročné. Jelikož aplikace Figma umožňuje tvořit pluginy naskýtá se možnost vytvoření generátoru, který vezme návrh a vygeneruje z něj validní React kód. Tento kód nemusí být vždy dokonalý, nicméně umožňuje vývojáři ušetřit čas. vývojář poté už jen rozdělí vygenerovaný kód na jednotlivé komponenty, vytvoří strukturu souborů a případně upraví některé nesrovnalosti s původním návrhem. Existuje mnoho pluginů, které tento úkol zjednodušují, například tato práce je zaměřena na tvorbu jednoho takového pluginu. Tato kapitola představí některá z již existujících řešení. (23)

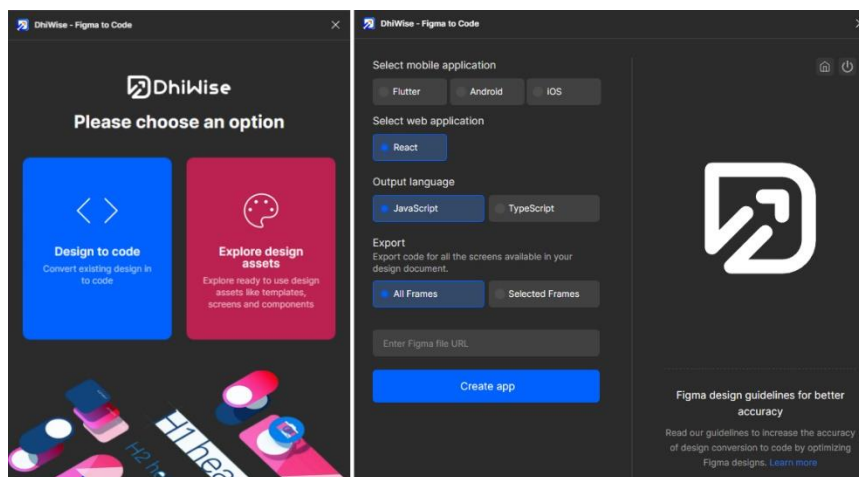
Prvním pluginem je Anima, která umožňuje vytváření interaktivních a responzivních prototypů vysoké kvality na základě návrhů v aplikacích Figma, Sketch a Adobe XD. Navržené prototypy lze exportovat do několika různých jazyků, což usnadňuje implementaci webových stránek a aplikací.

Základní verze umožňuje vytvořit jeden projekt s až třemi obrazovkami a je zdarma. Pro větší projekty je nutné platit měsíční nebo roční předplatné. Vygenerovaný kód lze zobrazit ve webové aplikaci pluginu. (24), (23) Ukázka grafického uživatelského rozhraní pluginu Anima je zobrazeno na obrázku číslo 1.



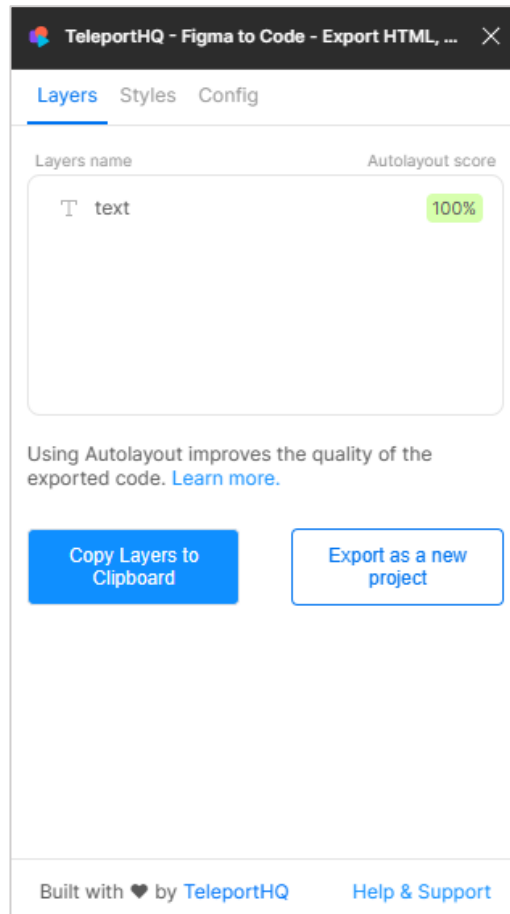
Obrázek 1: Rozhraní pluginu Anima. Zdroj vlastní.

Další možností je plugin DhiWise. Jedná se o plugin, který umožňuje generovat kód pro uživatelské rozhraní aplikace přímo z návrhu v aplikaci Figma. Plugin podporuje jazyky Flutter, React, Kotlin a Swift. Po registraci je plugin k dispozici zdarma a vygenerovaný kód lze zobrazit ve webové aplikaci pluginu. (25) Ukázka grafického uživatelského rozhraní pluginu DhiWise je zobrazeno na obrázku číslo 2.



Obrázek 2: Rozhraní pluginu DhiWise. Zdroj vlastní.

Jedním z možných pluginů je také TeleportHQ, který je zaměřený na převod návrhu z aplikace Figma na kód některých z podporovaných jazyků. Umožňuje exportovat návrhy do jazyků HTML/CSS, React, Next, Vue, Nuxt a Angular. Vygenerovaný kód je možné exportovat jako projekt nebo zkopírovat do schránky. (26) Ukázka grafického uživatelského rozhraní pluginu TeleportHQ je zobrazeno na obrázku číslo 3.



Obrázek 3: Rozhraní pluginu TeleportHQ. Zdroj vlastní.

Shrnutí

Existuje mnoho pluginů, které umožňují generovat kód z návrhu v aplikaci Figma. Zde jsou uvedeny pouze tři příklady z mnoha dostupných pluginů. Cílem všech těchto pluginů je vytvářet kvalitní kód z návrhu v aplikaci Figma pro požadovaný jazyk. Různé pluginy podporují různé jazyky a některé z nich podporují i více jazyků. Některé pluginy jsou zdarma, zatímco jiné vyžadují předplatné. Je důležité si uvědomit, že kvalita vygenerovaného kódu závisí na kvalitě návrhu. Proto je důležité dodržovat obecné praktiky při tvorbě návrhu a také specifické požadavky jednotlivých pluginů.

3.6 Figma pro vývojáře

Aplikace Figma je v základní verzi zdarma a tato verze umožňuje mimo jiné i prohlížení návrhů, což vývojáři s výhodou využívají. Prohlížením návrhu mohou vývojáři získat mnoho užitečných informací o návrhu které potřebují pro vývoj, a to bez nutnosti konzultace s designéry. Jedná se o informace, jako je odsazení prvků, barvy, fonty, efekty a další vizuální údaje. Všechny tyto údaje jsou přehledně zobrazené pro každou komponentu přímo v aplikaci. Aplikace však umožňuje získat pouze vizuální údaje, nedokáže exportovat žádnou logiku ani JavaScriptový kód. (27) Vývojáři jsou ti, kteří programují tyto komponenty a vytvářejí tak knihovnu komponentů, použitelnou napříč projekty. Komponenty vždy vycházejí z návrhů v aplikaci Figma a jsou měněny na základě změn v návrhu, aby vždy reflektovaly návrh. Aby byly všechny barvy, fonty, efekty a další vlastnosti napříč projekty jednotné, je vhodné zavést návrhový systém. Na vývoji návrhového systému by se měli podílet jak designéři, kteří tvoří vzhled a interakce, tak i vývojáři, kteří designéry usměrňují v tom, co je možné a co není. Z vytvořeného návrhového systému mají užitek obě strany, které nové aplikace skládají z předpřipravených komponent, do nichž vývojáři již jen přidají potřebnou funkcionalitu pomocí JavaScriptu v kódu aplikace. (28)

3.7 Figma pro UI/UX designéry

Figma disponuje menší paletou nástrojů, se kterými může designer pracovat než některé konkurenční produkty. Nicméně, disponuje přesně těmi nástroji, které umožňují vývojáři vytvořit jakýkoli návrh díky tomu, že všechny nástroje jsou do velké míry přizpůsobitelné. Absence nadbytečných funkcí umožňuje designerům vytvářet jednodušší a přehlednější návrhy a soustředit se na důležité aspekty, jako jsou jednotné fonty nebo správná volba barevných kombinací. Vlastnost, kterou designéři velmi ocení, je možnost kolaborace v reálném čase na stejném projektu. To jim umožňuje společně vytvářet jeden návrh, ať už jsou kdekoli na světě. Přímo v aplikaci Figma mohou vkládat komentáře přímo do pracovního prostoru, používat funkce chatování nebo audio hovoru. Vývoj návrhu je tak jednodušší, zábavnější a rychlejší. Designéři také s výhodou využívají návrhových systémů, které jim usnadňují práci tím, že využívají již definovanou paletu barev, volbu fontů nebo již dříve navržené celé komponenty, ze kterých tvoří nové návrhy aplikací. (28) Figma je vynikajícím nástrojem pro předávání projektů z jednoho člověka na druhého. Jelikož se jedná o cloudovou aplikaci, existuje vždy pouze jedna verze, ke které může být přístup udělen komukoli. Tak odpadá jakýkoli export návrhu a předávání fyzicky. Návrh je vždy dostupný online. (29)

4 POUŽITÉ TECHNOLOGIE

Tato kapitola je zaměřená na představení technologií, které byly využity při návrhu, implementaci a testování pluginu. U každého typu technologie je nejprve uvedeno, jakým způsobem byly technologie ve vývoji pluginu použity, poté jsou jednotlivě popsány.

4.1 Programovací jazyky

Před jakýmkoli vývojem nové aplikace je důležité zvolit, v jakém programovacím jazyce bude implementována. V případě vývoje pluginu pro aplikaci Figma však není tolik možností, jelikož pluginy do Figmy jsou webové aplikace, které komunikují s Figmou pomocí plugin API. Aby tato komunikace správně fungovala, musí být výsledný kód pluginu napsán v programovacím jazyce JavaScript. K vývoji pluginu tak může být použit libovolný jazyk, který lze následně převést do JavaScriptu. V případě tohoto pluginu byl zvolen jazyk TypeScript, který je doporučený pro vývoj pluginu do Figmy, jelikož nabízí vylepšené funkcionality editoru a disponuje našeptáváním funkcí z plugin API přímo ve vývojovém prostředí. Kód napsaný v jazyce TypeScript je následně pomocí technologie Webpack převeden na JavaScript.

JavaScript

JavaScript je objektově orientovaný skriptovací jazyk, který se používá hlavně pro tvorbu webových stránek. Byl vytvořen s cílem dodat statickým webovým stránkám dynamické chování a umožnit jim spolupracovat se servery. Dynamické stránky mohou poskytnout uživatelům zajímavější uživatelské rozhraní a umožnit tvorbu komplexních interaktivních webových aplikací. JavaScript se obvykle používá pro tvorbu klientské části aplikací, ale s využitím technologií, jako je například Node.js, může být použit i na serverové straně. Jméno JavaScript může být matoucí, protože se zdá, že je nějakým způsobem spojené s jazykem Java. Ve skutečnosti s ním však JavaScript sdílí pouze podobnou syntaxi a jinak vychází ze specifikace jazyka ECMAScript. Mezi výhody jazyka patří jeho rychlost a schopnost dynamického načítání stránek, což umožňuje změnu obsahu na obrazovce bez nutnosti nového načtení stránky. Další výhodou je jeho spolupráce s HTML a CSS, což umožňuje tvorbu robustních a interaktivních webových aplikací. (30) Existuje mnoho různých frameworků pro JavaScript, které usnadňují tvorbu webových aplikací. Mezi nevýhody jazyka patří riziko zneužití kódu, který se nachází volně v prohlížeči, možná nekompatibilita mezi různými internetovými prohlížeči a nemožnost práce s daty v jiných formátech, než jsou textové soubory webového serveru cookies. (31), (32)

TypeScript

TypeScript je silně typovaný, objektově orientovaný programovací jazyk, který je založen na jazyku JavaScript a je kompilován pomocí TypeScript kompilátoru. Tento kompilátor převádí TypeScript kód na ekvivalentní kód v JavaScriptu, což znamená, že každý platný kód napsaný v JavaScriptu je také platný v TypeScriptu. TypeScript poskytuje nadstavbu pro JavaScript a přidává do něj vylepšené nástroje, jako jsou například nová klíčová slova nebo typové anotace. TypeScript kompilátor také poskytuje přidanou vrstvu do vývojových prostředí, která zahrnuje funkce jako doplňování řádků, formátování kódu a barevné odlišení. Jeden z největších přínosů TypeScriptu je silné typování a kontrola kódu v čase kompilace. To umožňuje detekovat chyby v kódu dříve než při spuštění aplikace, což zjednodušuje proces vývoje a snižuje riziko chyb. TypeScript je zejména užitečný pro vývoj serverových aplikací, kde bez řádné typové kontroly a kontroly chyb při kompilaci nastávají komplikace. Díky TypeScriptu může být JavaScript vhodné řešení i pro serverovou stranu. Celkově lze tedy říci, že TypeScript je mocný nástroj pro vývoj webových a serverových aplikací v jazyce JavaScript, který umožňuje větší jistotu a efektivitu vývoje a snižuje riziko chyb. (33)

4.2 Značkovací a stylové jazyky

Pluginy do aplikace Figma mohou disponovat grafickým uživatelským rozhraním. Struktura tohoto rozhraní je implementována pomocí značkovacího jazyka HTML. Vzhled grafického rozhraní je implementován pomocí souborů kaskádových stylů CSS, které jsou psány nebo importovány do dokumentu HTML. Při vývoji pluginu byl jazyk HTML využit pro tvorbu základní kostry uživatelského rozhraní pluginu a CSS pro vizuální úpravy této kostry.

HTML

HTML je zkratka pro HyperText Markup Language, což je hypertextový značkovací jazyk, určený k tvorbě jednoduchých internetových stránek. Hypertext může být slovo, fráze nebo blok textu, který lze propojit s jiným dokumentem nebo textem. (34) language neboli značkovací jazyk, je standardní systém kódování textu, který se skládá z řady symbolů vložených do textového dokumentu pro řízení jeho struktury a formátování. (35) HTML není považován za programovací jazyk, protože nedokáže vytvářet dynamické funkce. Umožňuje vytváření struktur, oddílů, odstavců a odkazů použitím elementů, tagů a atributů. HTML je často využíván společně se soubory kaskádových stylů, které umožňují grafické úpravy internetových stránek a programovacím jazykem JavaScript, který umožňuje tvorbu dynamických funkcí.

Mezi hlavní využití tohoto značkovacího jazyka patří kromě vývoje internetových stránek také navigace internetem, jelikož HTML je silně používán pro vkládání hypertextových odkazů, nebo jako webová dokumentace, jelikož pomocí konkrétních HTML tagů dokáže organizovat a formátovat dokumenty podobně jako aplikace Microsoft Word. HTML disponuje čistým a konzistentním značením a jeho osvojení v základní formě je jednoduché, je tedy příjemný pro začátečníky. Jedná se o nejrozšířenější jazyk pro vývoj internetových stránek, existuje tedy velké množství materiálů a velká komunita, díky čemuž je snadné s ním pracovat a řešit problémy. Jedná se o open–source a zdarma dostupnou technologii, která funguje ve všech internetových prohlížečích. Je flexibilní a snadno funguje s jazyky a technologiemi komunikujícími se servery, jako je například PHP nebo Node.js. Mezi nevýhody HTML patří to, že dokáže tvořit pouze statické stránky a pro tvorbu dynamických stránek je potřeba použít nějaký serverový jazyk. Nelze znovu používat HTML dokumenty, pro každou stránku musí být vytvořen nový soubor. Může se také stát, že některé prohlížeče implementují nové HTML prvky pomalu, a tak se stránky, které využívají nejnovější HTML tagy nebudou zobrazovat správně. (36)

CSS

Zkratka CSS pochází z anglického názvu Cascading Style Sheets, což v češtině znamená kaskádové styly. (37) CSS je jazyk, který definuje vizuální vzhled prvků HTML na webové stránce. Soubory kaskádových stylů umožňují kontrolovat rozložení prvků, barvy, fonty, animace, přechody a další aspekty vizuálního charakteru. Tyto soubory mohou být zapsány přímo do souboru, do externího souboru nebo k jednotlivým HTML elementům. Kaskádové styly se řídí jasnými pravidly a syntaxí, což umožňuje vývojářům tvořit konzistentní a responzivní webové stránky pro různá zařízení a prohlížeče. (38) CSS se v pluginu využívají pro vizuální stránku grafického uživatelského rozhraní, ale také v samotném generátoru. Generátor vytváří některé styly komponentů, které jsou tvořeny na základě atributů v návrhu v aplikaci Figma.

4.3 Frameworky a knihovny

Framework a knihovna jsou dva různé způsoby, jak lze poskytovat znovupoužitelný kód vývojářům. Zatímco knihovna je kolekce funkcí nebo tříd, které mohou být importovány do projektu v libovolnou chvíli vývoje a použity v různých na sobě nezávislých projektech, framework je soubor pravidel, který určuje celkovou strukturu a způsob, jakým se kód projektu píše již od počátku vývoje. (39)

MUI

MUI je zkratka pro Material UI, což je open–source knihovna pro React, která implementuje Material Design od společnosti Google. Knihovna poskytuje širokou kolekci React komponent, které jsou připraveny k použití ihned po přidání do projektu. MUI se využívá při tvorbě uživatelského rozhraní grafických mobilních nebo webových aplikací. Použití této knihovny přináší mnoho výhod, jako je například rychlost vývoje aplikace. Díky předem vytvořeným a dostupným komponentům je návrh grafického uživatelského rozhraní velmi rychlý. Navíc jsou všechny komponenty pečlivě implementovány podle Material Designu a splňují nejvyšší standardy kvality a funkčnosti. Jednotlivé komponenty disponují širokou možností přizpůsobitelnosti, čehož lze dosáhnout pomocí atributů konkrétních komponent nebo pomocí stylování komponent pomocí JSX. Materiál UI také disponuje velmi dobře zpracovanou dokumentací, což zjednodušuje dohledání informací a způsob použití konkrétních komponent. V neposlední řadě je velkou výhodou také velká komunita používající tuto knihovnu, díky které je k dispozici mnoho materiálů a návodů. (40), (41)

React

React.js je open–source framework a front–end knihovna programovacího jazyka JavaScript, vyvinutá společností Facebook. Používá se k rychlému a efektivnímu vývoji webových aplikací s uživatelským rozhraním, přičemž jeho použití výrazně snižuje množství kódu potřebného k dosažení stejného výsledku v čistém JavaScriptu. Klíčovou vlastností knihovny je zpracování vrstvy zobrazení aplikace podle vzoru MVC, což umožňuje nejlepší a nejrychlejší vykreslování. Vývojáře povzbuzuje k tvorbě znovupoužitelných komponent místo jednoho rozsáhlého uživatelského rozhraní. Tyto komponenty se pak stávají stavebními kameny uživatelského rozhraní a poskytují také možnost vylepšeného vykreslování tím, že React vykresluje pouze komponenty, které to vyžadují. Knihovna React byla představena společností Facebook v roce 2013 z důvodu vlastní potřeby bohatšího, dynamičtějšího a citlivějšího uživatelského rozhraní, které by bylo rychlé a spolehlivé. (42)

4.4 Vývojová prostředí

Vývojové prostředí je program, se kterým vývojář pracuje a vytváří v něm nové aplikace. Jeho hlavním účelem je usnadnit a zpříjemnit práci vývojářům. Pro vývoj pluginu je doporučeným vývojovým prostředím Visual Studio Code, tento editor zdrojového kódu byl použit pro implementaci pluginu.

Pro testování bylo využito integrované vývojové prostředí WebStorm, jelikož je v něm implementovaný projekt, který má vytvořené téma pro knihovnu MUI. Vygenerovaný kód byl nakopírován do tetovací aplikace, byl spuštěn server node.js a aplikace byla zobrazena na webové stránce. Zobrazená aplikace byla testována pozorováním a pomocí různých interakcí

Visual Studio Code

Visual Studio Code je editor zdrojového kódu, který má mnoho výhod a je oblíbeným nástrojem mezi vývojáři. K dispozici je pro různé operační systémy, jako jsou Windows, macOS, Linux a Raspberry Pi OS nebo jako webová aplikace. Jeho hlavní předností je rychlost, a to hlavně při spuštění kde vývojáři nemusí na nic čekat. Editor také podporuje mnoho jazyků a běhových prostředí, včetně JavaScriptu, TypeScriptu, Node.js, C#, C++, Go, Java a Pythonu. Má také bohatý ekosystém rozšíření, která umožňují přidávat další funkce a podporu pro další jazyky. Mezi užitečné funkce editoru patří zvýrazňování syntaxe, porovnávání zdrojového kódu, automatické formátování kódu, intuitivní klávesové zkratky a snadné přizpůsobení celé aplikace. Díky robustním nástrojům pro ladění a krokování kódu mohou vývojáři snadno odhalit a odstranit chyby v kódu. Editor také nabízí nativní podporu pro Git, což umožňuje vývojářům pracovat se správou zdrojového kódu přímo v editoru, včetně zobrazení rozdílů a čekajících změn. Celkově je Visual Studio Code výkonným a efektivním nástrojem pro vývojáře, který nabízí mnoho užitečných funkcí a podporu pro mnoho jazyků a běhových prostředí. Díky své rychlosti a rozsáhlé knihovně rozšíření je ideální pro vývojáře všech úrovní. (43), (44), (45), (46)

WebStorm

WebStorm je integrované vývojové prostředí, zkráceně IDE, určené pro programování v jazyce JavaScript a souvisejících technologiích, jako jsou React, TypeScript, Node.js, HTML a další. Jeho cílem je usnadnit a zpříjemnit práci vývojářům prostřednictvím automatizace rutinních úkonů a pomoci při provádění komplexních úkolů. Klíčovými výhodami prostředí jsou nativní podpora programovacího jazyka JavaScript a s ním souvisejících technologií. Díky inteligentnímu editoru nabízí automatické dokončování kódu, detekci chyb, snadnou navigaci napříč kódovou základnou a bezpečný refaktoring kódu. Obsahuje také řadu integrovaných vývojářských nástrojů pro ladění a testování aplikací. Podporuje Git, díky čemuž umožňuje práci se správou verzí přímo v programu. Kromě toho obsahuje nástroje pro sestavení, terminál a HTTP klienta, a umožňuje přizpůsobení pracovního prostředí pomocí motivů a pluginů. (47), (48)

4.5 Programy

Desktopová aplikace Figma

Desktopová aplikace Figma je k dispozici pro operační systémy Windows i macOS a nabízí stejné funkce jako webová verze. Některé funkce jsou však dostupné pouze pro předplatitele. Díky desktopové aplikaci je možné pracovat ve Figmě i bez připojení k internetu, ale změny se nebudou ukládat do cloudu, dokud se neobnoví připojení k internetu. Nevýhodou desktopové aplikace je to, že vyžaduje výkonnější počítač. Pro vývoj pluginů je nutné používat desktopovou aplikaci, protože umožňuje vývoj a testování pluginů ve stejnou chvíli. Právě z tohoto důvodu byla desktopová aplikace Figma využita při implementaci a testování pluginu.

(49)

5 VÝVOJ PLUGINU

5.1 Stanovení cílů

Jelikož vývoj pluginu, který dokáže převést návrh z aplikace Figma na kód v jazyce JavaScript s využitím knihovny React.js a frameworku MUI, představuje rozsáhlý úkol, je nutné stanovit funkční požadavky, které bude plugin schopný na konci svého vývoje v rámci této práce schopný provádět. Práce na pluginu bude pokračovat i do budoucna a budou stanoveny nové cíle. Plugin bude také potřeba neustále ladit, ze začátku z důvodu oprav případných chyb, které se neobjevili během testování, a poté aby nadále odrážel potřeby designerů a adaptoval se novým postupům, které vývojáři používají při návrhu aplikací.

Cíle:

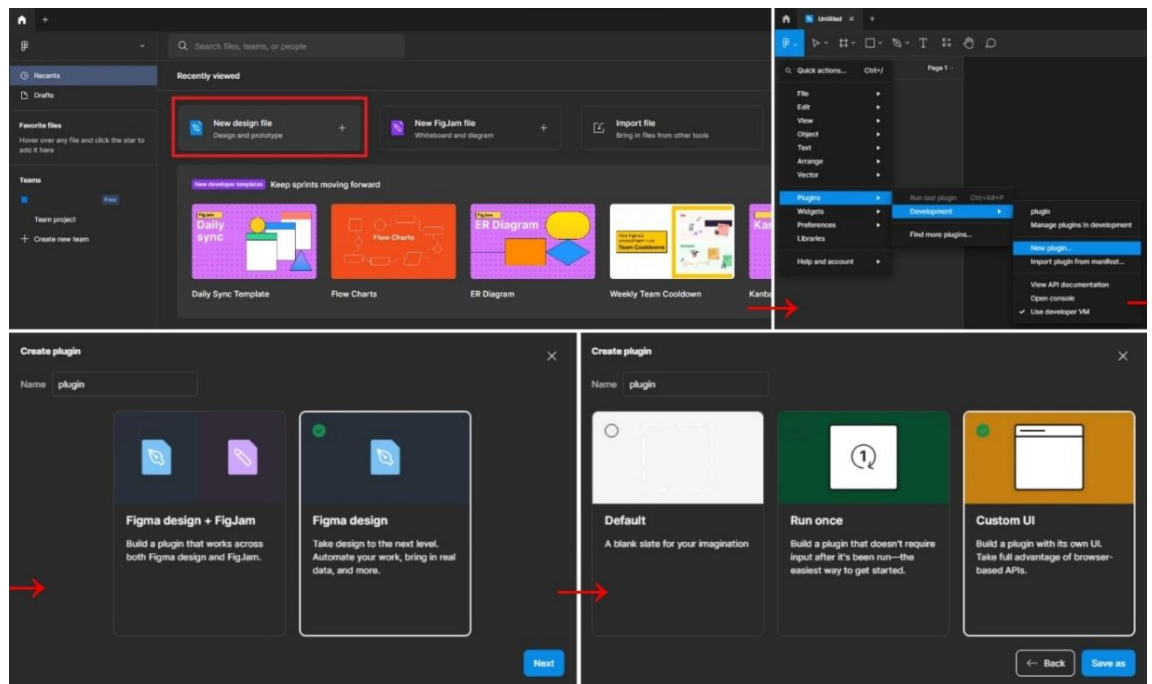
- Plugin bude podporovat tyto MUI komponenty: Alert, Avatar, Badge, Button, Card, CardContent, CardHeader, CheckBox, Chip, Dialog, DialogActions, DialogContent, DialogTitle, Divider, Icon, IconButton, Link, Radio, Select, TextField, ToggleButton, ToggleButtonGroup, Typography.
- Plugin bude generovat strukturu pouze MUI komponent s jejich atributy a importy.
- Plugin bude generovat kompletní strukturu sestavenou z MUI komponenty, odrážející původní návrh.

5.2 Tvorba a nastavení projektu

Generování šablony pluginu

Prvním krokem při vývoji pluginu je jeho vytvoření. Postup tvorby tohoto pluginu je znázorněn na obrázku číslo 4. Jednou z možností je vytvořit všechny soubory ručně, to ale není nutné, protože desktopová aplikace implementuje rozhraní, které provede uživatele tvorbou nového pluginu a podle voleb vygeneruje základní šablonu pluginu. Nejprve je třeba stáhnout desktopovou aplikaci z oficiálních stránek Figma, nainstalovat ji a přihlásit se. Dalším krokem je vytvoření nového návrhového souboru pomocí tlačítka "New design file", které se nachází napravo od záložky "Favorite files" v levém menu aplikace. Poté je třeba kliknout na ikonu loga Figma v levém horním rohu aplikace, což otevře menu. V tomto menu je potřeba zvolit cestu Plugins/Development/New plugin. Tím se otevře modální okno, kde je třeba vyplnit jméno pluginu a vybrat jeden ze dvou typů pluginu: Figma design + FigJam nebo pouze Figma design. Pro tento plugin je vybrán typ Figma design. Po kliknutí na tlačítko "Next" se zobrazí další modální okno s možnostmi šablon pluginu.

Pro tento plugin je zvolena možnost Custom UI z nabízených možností Default, Run once a Custom UI. Kliknutím na tlačítko "Uložit" se otevře průzkumník souborů, kde je třeba zvolit umístění pro uložení šablony, a tím je projekt vytvořen. (50)



Obrázek 4: Generování šablony pluginu. Zdroj vlastní.

Konfigurace technologie Webpack

Projekt používá Webpack, což široce rozšířená javascriptová knihovna, která umožňuje balíčkování souborů a spouštění úloh. (51) Webpack je v projektu využíván hlavně z důvodu dělení projektu na menší logické celky, protože samotný plugin do aplikace Figma se implementuje celý do jednoho JavaScriptového souboru. Použitím Webpacku můžeme rozdělit kód do jednotlivých souborů, a ty rozdělit do adresářů a vnést do projektu větší strukturovanost a přehlednost. Webpack také může zlepšit rychlost načítání projektů díky tomu, že vygenerovaný výsledný kód je v jednom souboru.

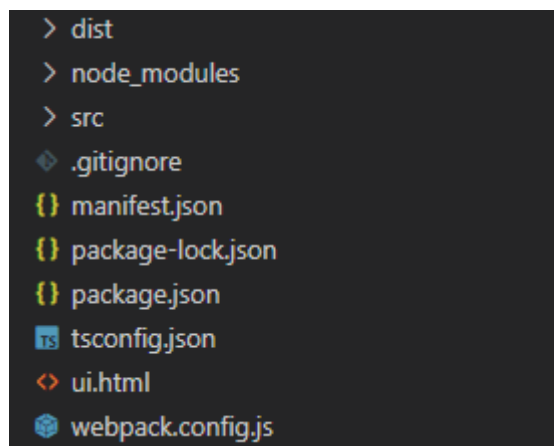
Instalace a konfigurace

Složku projektu otevřeme v editoru, v našem případě Visual Studio Code, a v terminálu spustíme příkaz "npm install webpack webpack-cli --save-dev", který do projektu uloží Webpack a nainstaluje další závislosti. Pokud byl příkaz úspěšně proveden, přidá se Webpack do sekce "devDependencies" v souboru package.json. Poté je třeba nakonfigurovat Webpack tak, aby využíval TypeScript, což je jazyk, který používáme při vývoji pluginu.

Toho dosáhneme nainstalováním TypeScript kompilátoru a zavaděče příkazem "npm install --save-dev typescript ts-loader", což přidá novou závislost do sekce "dependencies" v souboru package.json. Následně v projektu vytvoříme novou složku pojmenovanou "dist" a přesuneme do ní soubor code.js. Aby plugin správně fungoval, musíme nyní v souboru manifest.json změnit umístění souboru code.js tak, že za "main" napíšeme správnou cestu k souboru. Poté aktualizujeme skripty v souboru "package.json", aby používaly nainstalovaný Webpack. Skript "build" změníme na "webpack". Dále nakonfigurujeme soubor "tsconfig.json", kde nastavíme možnosti kompilátoru tak, aby zahrnoval soubory ze složky "src" a pro výstup kompilátoru používal vytvořenou složku "dist". Jako poslední krok musíme vytvořit soubor webpack.config.js a vyplnit ho podle vzoru, který je dostupný na oficiální stránce Figma Developers. (52) Tento soubor je v projektu potřeba jelikož plugin je vyvíjen v desktopové aplikaci Figma, která funguje jinak než prohlížeč. Po nakonfigurování Webpacku můžeme pomocí skriptu "build" plugin zkompileovat. Výsledek bude vygenerován do složky "dist". Pro vývoj používáme skript s názvem "build", který sleduje změny v kódu a v případě změny provede automatickou kompilaci. (52)

Popis struktury projektu

Obrázek číslo 5 ukazuje souborovou strukturu vygenerovaného projektu. Jednotlivé soubory jsou popsány v této kapitole.



Obrázek 5: Souborová struktura projektu. Zdroj vlastní.

Složka "dist" obsahuje vygenerované soubory balíčkovacím modulem webpack. V rámci pluginu se zde nachází pouze soubor code.js, který obsahuje vygenerovaný výkonný kód pluginu, který je převeden do jazyka JavaScript aby fungoval v aplikaci Figma.

Složka "node_modules" obsahuje všechny závislosti nebo moduly, které jsou potřebné pro běh aplikace.

Tyto moduly jsou do projektu nainstalovány pomocí nástroje správy balíčků npm (Node Package Manager) a jsou umístěny v této složce, aby byly snadno dostupné pro aplikaci. Každý modul v této složce je umístěn v samostatné podsložce, která má název odpovídající jménu balíčku.

Složka "src" obsahuje veškeré složky a soubory s výkonným kódem pluginu, zároveň se zde nachází soubor "code.ts", který je výchozím souborem pro kompilátor TypeScriptu. Do tohoto souboru se také importují všechny ostatní soubory projektu, aby byli zahrnuti do kompilace.

Soubor "manifest.json" je povinný soubor každého pluginu pro aplikaci Figma. Tento soubor obsahuje metadata o pluginu, jako jsou název, popis, ikona, verze, která se používá pro distribuci a správu pluginu v aplikaci Figma. Kromě toho také definuje, jakým způsobem má být plugin zaveden do aplikace a jaké jsou jeho funkce. Popisuje seznam příkazů, které plugin poskytuje uživateli a definuje jakým způsobem s nimi může být interagováno. V manifestu je také specifikováno, jaké oprávnění musí mít uživatel, aby mohl plugin používat a jakým způsobem se má plugin propojit s externími službami API.

Soubor "package.json" je soubor využívaný v Node.js projektech a slouží k definici různých informací o projektu jako je název, verze, seznam závislostí a další metadata. V tomto souboru jsou uvedeny všechny závislosti projektu, které je zde také možné spravovat. Závislosti je následně potřeba pomocí příkazu "npm install" nainstalovat, čímž je zajištěno že projekt bude mít k dispozici veškeré potřebné knihovny a balíčky pro svůj běh. Soubor také specifikuje skripty, které umožňují provádět různé úkoly jako například spuštění testů, sestavení kódu a jiné. (53)

Soubor "tsconfig.json" a "webpack.config.json" jsou konfigurační soubory pro kompilátor TypeScriptu. Soubor ui.html, je soubor značkovacího jazyka HTML, který definuje strukturu a vzhled pluginu.

5.3 Návrh pluginu

Uživatelské rozhraní

Návrh uživatelského rozhraní pluginu je velmi jednoduchý, neboť plugin nepotřebuje složité ovládací prvky pro provádění jednosměrné akce generování React kódu. Požadované vlastnosti pluginu jsou generování kódu, který obsahuje pouze komponenty MUI ve stejné hierarchii jako v návrhu ve Figmě, generování kódu, který bude odrážet celkový vzhled a rozložení prvků podle navrženého designu ve Figmě.

A generování importů potřebných pro vygenerovaný kód z návrhu v aplikaci Figma. Uživatelské rozhraní pluginu obsahuje ovládací prvky, které umožňují splnění těchto požadovaných vlastností.

Pro tyto účely jsou k dispozici tlačítka "MUI only", které po kliknutí generuje pouze MUI komponenty, "All design", které generuje celkový design a všechny komponenty, a "Imports", které generuje importy pro zvolený podstrom.

Kromě těchto základních funkcí pluginu jsou k dispozici dvě možnosti, které usnadňují práci s vygenerovaným kódem. První možnost je zabalení vygenerovaného kódu do JavaScriptové funkce s názvem "app". Druhá možnost vloží nad vygenerovaný kód importy. Tyto dvě možnosti lze kombinovat a vytvořit tak funkční React komponentu s vygenerovaným kódem, kterou lze nahrát přímo do souboru projektu. Po spuštění projektu se zobrazí vygenerovaný návrh odpovídající návrhu ve Figmě.

Vzhled uživatelského rozhraní pluginu je definován pomocí kaskádových stylů, které určují pozicování a barvy prvků. Plugin nese název "Figma to React" a je použit jako nadpis vložený do struktury souboru. Pod nadpisem se nacházejí již zmíněná tlačítka a dvě zaškrťovací pole pro volbu možností generátoru. Tyto prvky jsou stylovány v červené barvě s bílým nebo červeným textem a ohraničením, které se mění po najetí myši. Finální návrh a jednotlivé výstupy tohoto pluginu je možné vidět na obrázku číslo 6.



Obrázek 6: Návrh UI + možnosti výstupu pluginu. Zdroj vlastní.

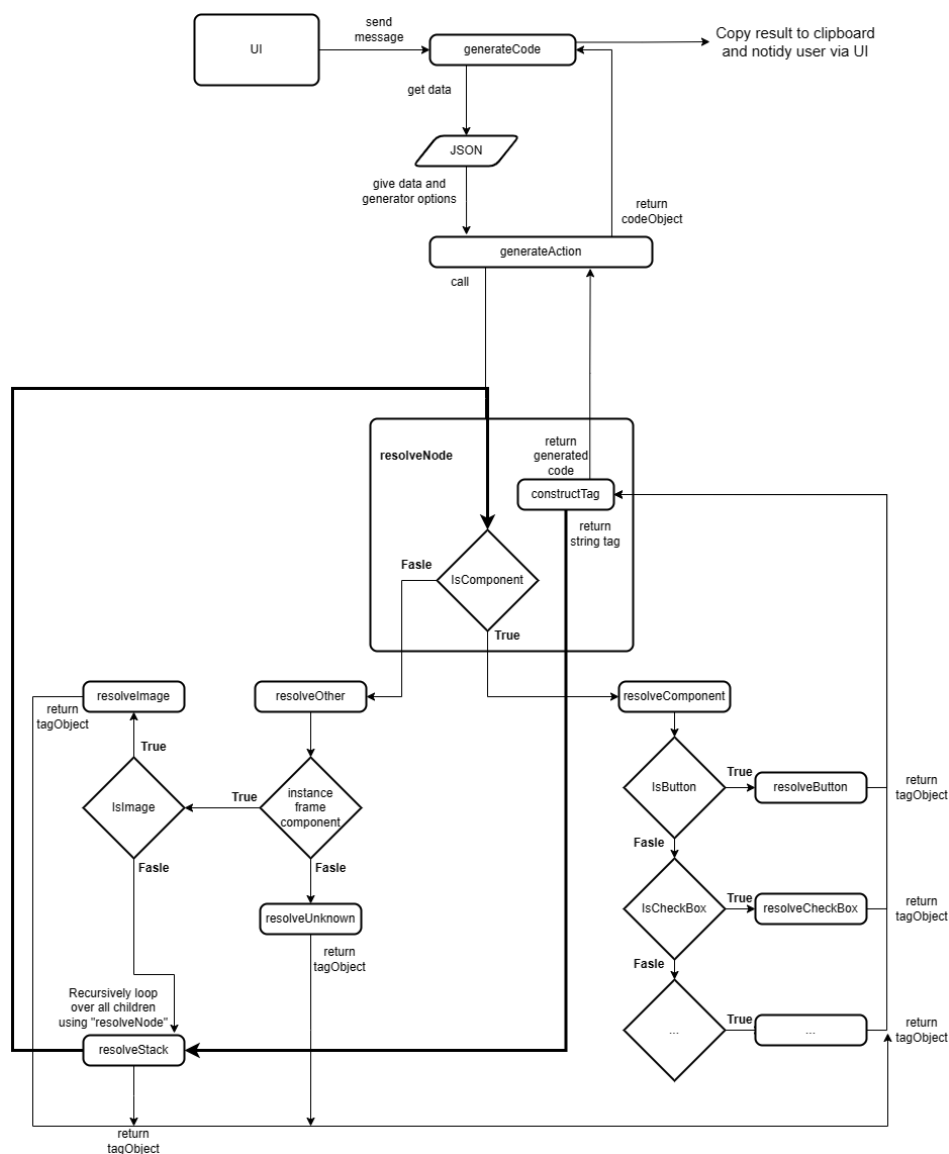
Výkonná část

Plugin je ovládán pomocí grafického uživatelského rozhraní, které definuje tlačítka a přepínače. Na základě interakce s těmito ovládacími prvky jsou generovány zprávy, které jsou posílány do výkonné části pluginu do funkce "generateCode", tak jak je znázorněno na obrázku číslo 7. Tato funkce přečte zvolený podstrom objektů v aplikaci Figma, který je reprezentován ve formátu JSON a obsahuje veškeré informace o jednotlivých objektech a jejich struktuře v návrhu.

Funkce "generateAction" přijímá tento JSON strom jako argument a spouští generátor. Generátor projde každý objekt v JSON souboru a rozhodne, zda je to MUI komponenta nebo není. Pokud se jedná o MUI komponentu, generátor spustí funkci "resolveComponent", která individuálně řeší každou komponentu pomocí její vlastní obslužné funkce. Z těchto funkcí se vrací objekt, který slouží jako předpis pro tvorbu tagu. Pokud se nejedná o MUI komponentu, zavolá se funkce "resolveOther", která řeší veškeré ostatní případy. Většina těchto funkcí také vrací předpisový objekt. Pokud se jedná o komponentu Stack, tak potomci této komponenty jsou postupně čtení a každý z nich je posílán přímo do funkce "resolveNode". Tímto způsobem se zajistí, že dojde ke zpracování všech potomků všech objektů.

Po projití všech potomků se z funkce "resolveOther" nebo "resolveComponent" vrací předpisový objekt, který se vloží jako atribut do funkce "constructTag". Tato funkce přečte tento předpis a vytvoří z něj textovou reprezentaci tagu. Pokud už jsou všechny objekty v JSON souboru projité tak se do funkce "generateAction" vrátí vygenerovaný kód. Pokud však stále dochází k procházení potomků Stack komponenty, textový předpis se vrací zpět do funkce "resolveStack", kde je vložen nakonec atributu předpisového objektu.

Po dokončení cyklu se vygenerovaný kód zabalí do objektu a pošle se zpět do funkce "generateCode". Tato funkce upozorní uživatele, že generování je dokončeno a výsledek se zkopíruje do schránky.



Obrázek 7: Diagram návrhu pluginu. Zdroj vlastní.

5.4 Vývoj pluginu

Uživatelské rozhraní

Uživatelská část pluginu je implementována do souboru, který má výchozí pojmenování `ui.html`. Toto pojmenování nemusí být pravidlem a lze jej uživatelsky změnit v konfiguračním souboru `manifest.json`. Jak již přípona souboru napovídá, jedná se o dokument typu HTML, takže vývoj probíhá stejným způsobem jako vývoj internetové stránky. Základní struktura pluginu, jako jsou tlačítka, nadpisy nebo texty, je definována pomocí prvků HTML. Vzhled a interaktivita dokumentu je doplněna pomocí kaskádových stylů, které mění vzhled určeným prvkům HTML struktury. Nedílnou částí front-endové části je také JavaScriptová část, která se musí nacházet přímo v dokumentu mezi tagy `<script>`. Za pomoci JavaScriptu interaguje front-endová část s výkonnou částí pluginu pomocí Plugin API.

Samotná komunikace probíhá pomocí jednoduchých zpráv a event listenerů. Například při kliknutí na tlačítko se nastaví event listener a pokud je zaregistrován, vytvoří se zpráva, která je pomocí funkce `parent.postMessage` odeslána do výkonné části kódu, kde je zpráva zpracována. Zprávy jsou většinou objekty, které obsahují atributy s identifikátorem a případně data, která plugin získává od uživatele z uživatelského rozhraní pluginu.

Uživatelská část obsahuje tři tlačítka a dva checkboxy, jak je uvedeno v návrhu. Po každém kliknutí na tlačítko se vytvoří zpráva obsahující požadovanou akci a zvolené předvolby. Tato zpráva je poslána do výkonné části, kde je zpracována. Z výkonné části se zpět do uživatelského rozhraní již nic nevrací, uživatel je notifikován o dokončení výkonnou částí pomocí notifikace přímo v programu Figma pomocí Figma API. Uživatelské rozhraní umožňuje generovat kód opakovaně, dokud není plugin ukončen.

Výkonná část

Zdrojové soubory projektu se nacházejí v adresáři "src", který je rozdělen do několika podsložek, které popisují, co zdrojové soubory v nich dělají. Hlavním souborem je "code.ts", který komunikuje s UI a po uživatelské interakci spustí funkci "generateCode". Tato funkce pomocí Figma API získá zvolený podstrom ve formě JSON souboru. Generátor se spustí pomocí funkce "generate", která se nachází v zdrojovém souboru "generateAction". Tento soubor obsahuje globální proměnné pro importy a funkce, které jsou dostupné v celém projektu. V souboru "generateAction" se spouští funkce "resolveNode", která prochází JSON strukturu návrhového souboru a řeší každý objekt. Pro každý objekt se rozhodne, zda je to komponenta nebo ne, a podle toho se zavolá příslušná obslužná funkce, která vrátí předpisový objekt. Tento objekt se používá k vytvoření textové reprezentace tagu pomocí funkce "constructTag", která vrátí výslednou reprezentaci tagu zpět do souboru "generateAction" nebo do souboru "resolveStack".

Pokud se jedná o komponentu, spustí se funkce "resolveComponent", která nejprve získá instanční atributy komponenty, které jsou nastaveny v návrhu a dostupné v JSON souboru, a poté pomocí switch bloku rozhodne, o jakou komponentu se jedná podle jména a zavolá příslušnou obslužnou funkci pro danou komponentu. Obslužná funkce komponenty zpracovává její název, atributy, strukturu a obsah komponenty a vše zabalí do objektu, který vrátí. Taktéž se přidá import dané komponenty do globálního objektu v souboru "generateAction". Vytvořený objekt slouží jako předpis pro budoucí tvorbu tagu.

Pokud se obslužná funkce nenajde, vrátí se objekt, který obsahuje pouze informaci o tom, že daná komponenta není implementována.

Pokud se nejedná o komponentu, je vyvolána funkce „resolveOther“, která řeší všechny ostatní případy, které mohou nastat, zpracovávají se základní typy objektů ve Figmě, ilustrace, obrázky a komponenty Stack a Box, které jsou výjimečně řešeny zde. Pokud se jedná o objekty typu Instance, Frame nebo Component, zjišťuje se, zda se jedná o ikonu případně ilustraci nebo Stack. V prvním případě se pomocí obslužné funkce zjišťuje, o jakou ilustraci nebo ikonu se jedná, a vytváří se předpisový objekt obsahující tyto informace a přidává se import dané komponenty. Pokud není nalezena shoda, a žádná ikona nebo ilustrace nebyla nalezena, vrátí se předpisový objekt pro komponentu Box, která obsahuje atributy tak, aby vytvořila obdélník o rozměrech odpovídajících rozloze ilustrace nebo ikony. Tento obdélník zabírá místo na výsledné stránce a signalizuje, že zde je potřeba doplnit ilustraci nebo ikonu.

V případě Stacku se pomocí cyklu procházejí všechny jeho potomci a posílají se do funkce „resolveNode“. Každý z potomků je vyřešen a pomocí funkce „constructTag“ se vytvoří nový tag, který se následně vrací zpět do funkce „resolveStack“. Tato funkce tvoří nový předpisový objekt a postupně přidává vytvořené tagy do jeho atributu „content“. Po projití všech potomků je tento objekt vrácen zpět do funkce „createTag“, která vytvoří celý podstrom. Pokud ve funkci "resolveOther" není nalezena žádná shoda, vrátí se předpisový objekt s informací, že nebylo možné rozpoznat daný JSON objekt, který je pak vykreslen na stránce.

Všechny funkce vždy vrací předpisový objekt, který je poté předán funkci "createTag". Ta přečte atributy předpisového objektu, jako jsou název, atributy a obsah, a vytvoří z nich textovou reprezentaci tagu. Tato textová reprezentace je buď vrácena zpět do funkce "resolveStack" nebo "generateAction", kde je uložena do proměnné. Funkce "generateAction" vrací objekt, který obsahuje vygenerovaný kód. Na základě možností, které lze zvolit v UI, je kód obohacen o importy nebo zabalen do funkce. Tento objekt je poté přijat v souboru "code.ts", kde je uživatel informován o úspěšné generaci kódu a kód je zkopírován do schránky.

6 TESTOVÁNÍ PLUGINU

Na základě zvolených předvoleb v grafickém uživatelském rozhraní a volby typu generátoru proběhne vygenerování kódu. Tento kód je vypsán do konzole a zároveň automaticky zkopírován do schránky. Vygenerovaný kód je poté nutné nakopírovat do již existujícího React projektu, který lze vytvořit například pomocí příkazu “`npx create-react-app <název>`“ v příkazové řádce. Ve vygenerovaném projektu pomocí desktopové aplikace Figma lze nahradit celý kód rodičovské komponenty “`App.js`“ vygenerovaným kódem nebo vytvořit novou komponentu.

Při testování tohoto pluginu bylo využito vývojové prostředí WebStorm a zkušební projekt využívající vytvořené téma, které definuje výchozí stylování, barvy, fonty a další vlastnosti MUI knihovny. Jednotlivé části pluginu, jako jsou konkrétní komponenty a jejich nastavení, jako velikost, barva a jiné individuální vlastnosti, byly testovány pomocí zmíněného postupu. Po každé změně byl kód znovu vygenerován a nakopírován do spuštěné React aplikace. Po znovunačtení stránky v prohlížeči byly pozorovány změny, které se s danou změnou projevíly.

Po otestování jednotlivých komponent byly testovány i větší wireframy, které se skládaly z více prvků. Důležitou součástí bylo pozicování prvků, aby se vše zobrazovalo na správném místě podle návrhu. Po úspěšném vygenerování a zobrazení wireframu se začal ladit zdrojový kód pluginu. V kódu se vyskytovaly redundantní funkce, překlepy ve stylování, atributech nebo špatná struktura kódu. Generátor pak musel být upraven, aby generoval správný výstup, a ten byl znovu testován pomocí zmíněného postupu.

Tímto způsobem byl plugin laděn až do stavu použitelného výstupu, který vývojářům usnadní vývoj. Plugin na konci této práce generuje solidní a použitelný základ pro budoucí vývoj aplikací s navrženým front-endem. Nicméně v budoucnosti se mohou objevit, některé neobjevené chyby, nebo dojde ke změnám přímo v knihovně MUI nebo React a generátor bude třeba upravit znovu testovat což je s neustálým vývojem technologií nevyhnutelné.

ZÁVĚR

Cílem této práce bylo vytvořit plugin pro aplikaci Figma, který umožní generovat zdrojový kód v programovacím jazyce JavaScript z návrhů využívajících knihovnu MUI, které jsou vytvořeny v aplikaci Figma. Plugin měl splňovat tři klíčové vlastnosti. Podporovat definované MUI komponenty, generovat kód pouze s MUI komponenty a generovat kód odpovídající návrhu v aplikaci Figma. Tyto vlastnosti byly implementovány pomocí ovládacích prvků grafického uživatelského rozhraní a funkcí pluginu. Plugin tedy umožňuje uživatelům generovat oba typy kódu a podporuje všechny definované MUI komponenty, což znamená, že splňuje všechny cíle, které byly stanoveny.

Dále plugin umožňuje uživatelům zabalit kód do JavaScriptové funkce nebo vygenerovat importy pro jednotlivé komponenty. V budoucnu by bylo možné tuto práci rozšířit přidáním nových podporovaných MUI komponent, implementací vylepšených funkcionalit nebo optimalizací kódu. Práce na pluginu bude pokračovat, aby nadále odrážela rozhraní MUI a pracovní postupy uživatelů. Plugin může být využíván designéry nebo vývojáři, kteří používají aplikaci Figma pro tvorbu návrhů svých aplikací. Tento plugin jim umožňuje výrazně zkrátit čas potřebný k implementaci návrhu.

7 POUŽITÁ LITERATURA

- [1] The Designest. The Designest. *Main Steps for UI Design Process*. [Online] The Designest, 24. 10. 2022. [Citace: 27. 4. 2023.] Dostupné z: <https://thedesigntest.net/ui-design-process-steps/>.
- [2] UX Design Institute. What does a UI designer do? *UX Design Institute*. [Online] UX Design Institute, 29. 3. 2022. [Citace: 2023. 27. 4.] Dostupné z: <https://www.uxdesigninstitute.com/blog/what-does-a-ui-designer-do/>.
- [3] Coursera. UI vs. UX Design: What's the Difference? *Coursera*. [Online] Coursera, 15. 03. 2023. [Citace: 27. 4. 2023.] Dostupné z: <https://www.coursera.org/articles/ui-vs-ux-design>.
- [4] Cahil, Carl a May, Tom. The best UI design tools in 2023. *Creative Bloq*. [Online] Creative Bloq, 20. 3. 2023. [Citace: 27. 4. 2023.] Dostupné z: <https://www.creativebloq.com/how-to/20-best-ui-design-tools>.
- [5] Ahmad, Karim. What Is Sketch and What Can You Do With It? [Online] makeusof, 27. 2. 2022. [Citace: 27. 4. 2023.] Dostupné z: <https://www.makeuseof.com/what-is-sketch/>.
- [6] Sketch. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.sketch.com/>.
- [7] Rae, Mat. What is Adobe XD and what is it used for? *Adobe*. [Online] Adobe, 26. 10. 2020. [Citace: 27. 4. 2023.] Dostupné z: <https://www.adobe.com/cz/products/xd/learn/get-started/what-is-adobe-xd-used-for.html>.
- [8] Appsrhino. What is InVisionapp? Everything you need to know. *Appsrhino*. [Online] 29. 3. 2023. [Citace: 27. 04 2023.] Dostupné z: <https://www.appsrhino.com/blogs/what-is-invisionapp-everything-you-need-to-know>.
- [9] Wells, Jeremy. 7 Reasons Why I Use InVision for Rapid Prototyping. *Medium*. [Online] 14. 2. 2014. [Citace: 27. 4 2023.] Dostupné z: <https://medium.com/@mrjeremywells/7-reasons-why-i-use-invision-for-rapid-prototyping-ed1c33d5b86>.
- [10] Invision. Half the price of Miro and Mural. *Invision*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.invisionapp.com/freehand/pricing>.
- [11] themejunkie. What is Figma? (And How to Use Figma for Beginners). *themejunkie*. [Online] 26. 4. 2023. [Citace: 27. 4. 2023.] Dostupné z: <https://www.themejunkie.com/what-is-figma/>.
- [12] Cousins, Carrie. What Is Figma? a 101 Intro. *Design shack*. [Online] 20. 11. 2019. [Citace: 27. 4. 2023.] Dostupné z: <https://designshack.net/articles/software/what-is-figma-intro/>.
- [13] Jůn, Šimon. Proč je Figma dar z nebes? *Simonjun*. [Online] 8. 5. 2022. [Citace: 27. 4. 2023.] Dostupné z: <https://www.simonjun.cz/blog/proc-je-figma-dar-z-nebes#toc-co-je-to-figma-2>.

- [14] Figma. Active file limits. *Figma*. [Online] [Citace: 27. 04 2023.] Dostupné z: <https://help.figma.com/hc/en-us/articles/1500006775761-Active-file-limits>.
- [15] Capacci, Manuele. Mini Tutorial – Leveraging Figma’s Features for the Entire Design Process. *Toptal*. [Online] 7. 1. 2020. [Citace: 27. 4. 2023.] Dostupné z: <https://www.toptal.com/designers/figma/figma-features>.
- [16] Figma. Prototype while you design, and vice versa. *Figma*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.figma.com/prototyping/>.
- [17] Uxpin. What is a Single Source of Truth, and Why Do You Need it? *Uxpin*. [Online] 27. 5. 2021. [Citace: 27. 4. 2023.] Dostupné z: [What is a Single Source of Truth, and Why Do You Need it? \(uxpin.com\)](https://uxpin.com)
- [18] Figma. Share files and prototypes. *Figma*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://help.figma.com/hc/en-us/articles/360040531773-Share-files-and-prototypes>.
- [19] Figma. Guide to the Figma mobile app. *Figma*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://help.figma.com/hc/en-us/articles/1500007537281-Guide-to-the-Figma-mobile-app>.
- [20] Bayo, Ferdian. Pros & Cons of Using Figma (Updated). *Medium*. [Online] 16. 7. 2019. [Citace: 27. 4. 2023.] Dostupné z: <https://medium.com/gizalab/pros-cons-of-using-figma-317115f762be>.
- [21] Vo, Aine. Pros & Cons Of Figma. *Uimega*. [Online] 19. 8. 2022. [Citace: 27. 4. 2023.] Dostupné z: <https://blog.uimega.com/pros-cons-of-figma>.
- [22] Figma. Introduction. *Figma*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.figma.com/plugin-docs/>.
- [23] Singh, Ajay. 5 Awesome Plugins That Can Convert Figma To ReactJs Code. *Medium*. [Online] 23. 3. 2022. [Citace: 27. 4. 2023.] Dostupné z: <https://medium.com/@ajay.singh07/5-awesome-plugins-that-can-convert-figma-to-reactjs-code-9614c7d7cc8d>.
- [24] Anima. Anima + Figma. *Anima*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.animaapp.com/figma>.
- [25] DhiWise Pvt Ltd. DhiWise - Figma to Code. *Figma*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.figma.com/community/plugin/1037309320238203168/DhiWise---Figma-to-Code>.
- [26] TeleportHQ. TeleportHQ - Figma to Code - Export HTML, CSS, React & Vue. *Figma*. [Online][Citace: 27. 4. 2023.] Dostupné z: <https://www.figma.com/community/plugin/992726161890204477/TeleportHQ---Figma-to-Code---Export-HTML%2C-CSS%2C-React-%26-Vue>.
- [27] Santangelo, Mirko. An overview of Figma for developers. *Figma*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.figma.com/best-practices/tips-on-developer-handoff/an-overview-of-figma-for-developers/>.

- [28] Jůn, Šimon. Figma očima grafika a vývojáře. *Simonjun*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.simonjun.cz/blog/figma-ocima-grafika-a-vyvojare>.
- [29] Jůn Šimon. Loom: Nástroj, který zlepšil předávání podkladů o 100 %. *Simonjun*. [Online] [Citace: 27. 4. 2023.] Dostupné z: <https://www.simonjun.cz/blog/loom-nastroj-pro-predavani-podkladu>.
- [30] Blažková, Tereza. Proč je JavaScript nejžádanější technologií 2021? *Itnetwork*. [Online] 17. 8. 2021. [Citace: 27. 4. 2023.] Dostupné z: <https://www.itnetwork.cz/blog/proc-je-javascript-nejzadanejsi-technologie-2021>.
- [31] Cihak, Lukas. Co jsou cookies a proč vás s nimi každý web otravuje? *CDR*. [Online] 3. 3. 2020. [Citace: 27. 4. 2023.] Dostupné z: <https://cdr.cz/clanek/co-jsou-cookies-proc-vas-s-nimi-kazdy-web-otravuje>.
- [32] Koďousková, Barbora. JAVASCRIPT PRO ZAČÁTEČNÍKY: CO TO JE A JAK FUNGUJE. *Rascasone*. [Online] 28. 1. 2022. [Citace: 27. 4. 2023.] Dostupné z: <https://www.rascasone.com/cs/blog/co-je-javascript-pro-zacatecniky>.
- [33] Shubel, Meredith. What Is TypeScript? *The new stack*. [Online] 26. 7 2022. [Citace: 27. 4. 2023.] Dostupné z: https://www.google.com/search?q=m%C4%9Bs%C3%ADce+anglicky&sxsrf=APwXEdebY3uZ5HHrreJKWVi4KQuma-uYBA:1682601268032&source=Inms&tbm=isch&sa=X&ved=2ahUKEwieiYSXksr-AhWMgP0HHdYkAkWQ_AUoAXoECAEQAw&biw=1863&bih=929&dpr=1#imgrc=tLM-nExsjdNKkM.
- [34] Rouse, Margaret. Hypertext. *Techopedia*. [Online] 29. 10. 2012. [Citace: 27. 4. 2023.] Dostupné z: <https://www.techopedia.com/definition/5177/hypertext>.
- [35] The editors od Encyclopedia Britannica. markup language. *Britannica*. [Online] 12. 10. 2022. [Citace: 29. 4. 2023.] Dostupné z: <https://www.britannica.com/technology/markup-language/additional-info#history>.
- [36] Astari, S. What Is HTML? Hypertext Markup Language Basics Explained. *Hostinger*. [Online] 10. 4. 2023. [Citace: 29. 4. 2023.] Dostupné z: <https://www.hostinger.com/tutorials/what-is-html>.
- [37] Vorbová, Renata. Co je nutné vědět o CSS – 1. díl. *Shoptet*. [Online] 23. 9. 2009. [Citace: 29. 4. 2023.] Dostupné z: https://blog.shoptet.cz/co-je-nutne-vedet-o-css-1-dil/?gclid=CjwKCAjw3POhBhBQEiwAqTCuBqQ6onLzwG_1H1j8e6yRu9KQe-OawwdXFkvpvXZjKykE9FZagE9bwBoCyKAQAvD_BwE.
- [38] Mozilla. What is CSS? *Mdn web docs*. [Online] 24. 2. 2023. [Citace: 29. 4. 2023.] https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS.
- [39] Wozniewicz, Brandon. The Difference Between a Framework and a Library. *freeCodeCamp*. [Online] 1. 2. 2019. [Citace: 29. 4. 2023.] Dostupné z: <https://www.freecodecamp.org/news/the-difference-between-a-framework-and-a-library-bd133054023f/>.

- [40] Mui. Material UI - Overview. *Mui*. [Online] [Citace: 29. 4. 2023.] Dostupné z: <https://mui.com/material-ui/getting-started/overview/>.
- [41] Sirotko, Alesia. What is Material UI? *Flatlogic*. [Online] 15. 3. 2022. [Citace: 29. 4. 2023.] Dostupné z: <https://flatlogic.com/blog/what-is-material-ui/>.
- [42] Herbert, David. What is React.js? (Uses, Examples, & More). *Hubspot*. [Online] 27. 6. 2022. [Citace: 29. 4. 2023.] Dostupné z: <https://blog.hubspot.com/website/react-js>.
- [43] Heller, Martin. What is Visual Studio Code? Microsoft's extensible code editor. *InfoWorld*. [Online] 8. 7. 2022. [Citace: 29. 4. 2023.] Dostupné z: <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>.
- [44] Visual Studio Code. Getting Started. *Visual Studio Code*. [Online] 11. 3. 2021. [Citace: 29. 4. 2023.] Dostupné z: <https://code.visualstudio.com/docs>.
- [45] Visual Studio Code. Setting up Visual Studio Code. *Visual Studio Code*. [Online] 11. 3. 2021. [Citace: 29. 4. 2023.] Dostupné z: <https://code.visualstudio.com/docs/setup/setup-overview>.
- [46] Visual Studio Code. Why did we build Visual Studio Code? *Visual Studio Code*. [Online] 11. 3. 2021. [Citace: 29. 4. 2023.] Dostupné z: <https://code.visualstudio.com/docs/editor/whyvsc>.
- [47] WebStorm. Meet WebStorm. *WebStorm*. [Online] 27. 2. 2023. [Citace: 29. 4. 2023.] Dostupné z: <https://www.jetbrains.com/help/webstorm/meet-webstorm.html#first-steps>.
- [48] WebStorm. *WebStorm*. [Online] [Citace: 29. 4. 2023.] Dostupné z: <https://lp.jetbrains.com/webstorm-ide/>.
- [49] Angelica, Laura. Detailed Steps to Download Figma for Windows. *Wondershare*. [Online] 29. 3. 2023. [Citace: 29. 4. 2023.] Dostupné z: <https://mockitt.wondershare.com/figma/figma-windows.html>.
- [50] Figma. Plugin Quickstart Guide. *Figma Developers*. [Online] [Citace: 29. 4. 2023.] Dostupné z: <https://www.figma.com/plugin-docs/plugin-quickstart-guide/>.
- [51] Michálek, Martin. Webpack: Úplné základy a tutoriál k tomu. *Vzůru dolů*. [Online] 9. 10. 2019. [Citace: 29. 4. 2023.] Dostupné z: <https://www.vzhurudolu.cz/prirucka/webpack>.
- [52] Figma. Figma Developers. *Libraries and bundling*. [Online] [Citace: 29. 4. 2023.] Dostupné z: <https://www.figma.com/plugin-docs/libraries-and-bundling/>.
- [53] Npmjs. *npm Docs*. [Online] [Citace: 29. 4. 2023.] Dostupné z: <https://docs.npmjs.com/cli/v9/configuring-npm/package-json#description>.