



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**PLATFORMA PRO PODPORU INTERAKTIVITY VE
VÝUCE**

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DOMINIK HOFMAN

VEDOUcí PRÁCE

SUPERVISOR

Ing. VOJTĚCH MRÁZEK, Ph.D.

BRNO 2024

Zadání bakalářské práce



156469

Ústav: Ústav počítačových systémů (UPSY)
Student: **Hofman Dominik**
Program: Informační technologie
Název: **Platforma pro podporu interaktivity ve výuce**
Kategorie: Web
Akademický rok: 2023/24

Zadání:

1. Seznamte se s nástroji určenými pro podporu interaktivity mezi posluchači a prezentujícími s cílem zvýšit míru zapojení posluchačů a udržet pozornost.
2. Navrhněte systém pro podporu výuky IT kurzů, kde jedním ze stěžejních prvků je úsek programového kódu. Při návrhu dbejte na ergonomii z hlediska uživatele.
3. Navržený systém implementujte ve formě aplikace určené pro mobilní zařízení. Zvolte vhodnou platformu (např. PWA) tak, aby použití aplikace nepředstavovalo významnou vstupní bariéru.
4. Ověřte funkčnost implementace a vyhodnoťte její parametry z hlediska funkčních i nefunkčních požadavků. Při vyhodnocení využijte dotazníkové šetření.
5. Diskutujte dosažené parametry a možnosti dalšího rozšíření.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodu 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Mrázek Vojtěch, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 2.5.2024

Abstrakt

Tato bakalářská práce se zaměřuje na vývoj systému pro podporu interaktivity mezi posluchači a prezentujícími, klade si za cíl zvýšit míru zapojení posluchačů a udržet tak jejich pozornost. Hlavním cílem práce je navrhnout a implementovat systém pro podporu výuky IT orientovaných kurzů, přičemž klíčovým prvkem bude možnost vkládání a editace programového kódu. Platforma je realizována formou progresivní webové aplikace.

Aplikace dokáže ověřit geolokaci uživatele, tedy zda je posluchač přítomen na fakultě informatiky VUT. Prezentující vytvoří kvíz s otázkami a správnými odpověďmi - tím je vygenerován přístupový kód ke kvízu. Posluchači se pomocí tohoto kódu připojí ke kvízu a čekají na jeho zahájení. Poté se uživatelům vždy již postupně zobrazují otázky s jednotlivými odpověďmi, ze kterých posluchač vybírá ty správné. Přednášející si po skončení kvízu dokáže zobrazit jeho detailní statistiky.

Práce dále zahrnuje ověření funkčnosti implementace a hodnocení jejich parametrů z hlediska funkčních i nefunkčních požadavků. Pro vyhodnocení výsledků je využito dotazníkového šetření, které poskytuje informace o přívětivosti uživatelského prostředí a spokojeností s aplikací.

Abstract

This bachelor's thesis focuses on the development of a system to support interactivity between listeners and presenters, aiming to increase the level of listener engagement and thus maintain their attention. The main goal of the work is to design and implement a system to support the teaching of IT-oriented courses, the key element of which will be the possibility of inserting and editing program code. The platform is implemented in the form of a progressive web application.

The application can verify the geolocation of the user, i.e. whether the listener is present at the faculty of information technology VUT. The presenter creates a quiz with questions and correct answers - this generates an access code for the quiz. Listeners use this code to join the quiz and wait for it to start. Afterwards, users are always presented with questions with individual answers, from which the listener selects the correct ones. After finishing the quiz, the presenter can view its detailed statistics.

The work also includes verification of the functionality of the implementation and evaluation of their parameters from the point of view of functional and non-functional requirements. A questionnaire survey is used to evaluate the results, which provides information on the friendliness of the user environment and satisfaction with the application.

Klíčová slova

progresivní webová aplikace, interaktivita, javascript, react, firebase, QR kód, GPS, PWA, vite, webpack

Keywords

progressive web application, interactivity, javascript, react, firebase, QR code, GPS, PWA, vite, webpack

Citace

HOFMAN, Dominik. *Platforma pro podporu interaktivity ve výuce*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vojtěch Mrázek, Ph.D.

Platforma pro podporu interaktivity ve výuce

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vojtěcha Mrázka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Dominik Hofman
6. května 2024

Poděkování

Rád bych poděkoval panu doc. Ing. Zdeňkovi Vašíčkovi, Ph.D. za jeho ochotu a cenné rady během mé bakalářské práce. Dále bych chtěl poděkovat panu Ing. Vojtěchu Mrázkovi, Ph.D. za vedení mé bakalářské práce, přínosné podněty pro její zlepšení a také za možnost otestovat aplikaci přímo na jedné z jeho přednášek.

Obsah

1	Úvod	4
2	Progresivní webové aplikace	5
2.1	Role a význam knihovny React pro uživatelské rozhraní	6
2.2	Sestavování webových aplikací	7
2.2.1	Webpack	7
2.2.2	Vite	8
2.3	Ekosystém Firebase	9
2.3.1	Realtime databáze	9
2.3.2	Autentizace	9
2.3.3	Databáze Firestore	9
2.3.4	Firebase úložiště	10
2.3.5	Cloudové funkce	10
2.3.6	Nasazení aplikace do produkčního prostředí	11
2.3.7	Cloud Messaging	11
2.4	Využití GPS v progresivních webových aplikacích	12
2.5	QR kódy	12
3	Průzkum existujících řešení	14
3.1	Mentimeter	14
3.2	Kahoot	15
3.3	Slido	16
4	Návrh a architektura systému	18
4.1	Požadavky, předpoklady a omezení	19
4.2	Bezpečnost systému	22
4.3	Uživatelské rozhraní	23
4.4	Programový kód	25
5	Implementace	27
5.1	Klientská část	27
5.1.1	Registrace a přihlášení uživatele	27
5.1.2	Připojení do kvízu	28
5.1.3	Nástěnka	31
5.1.4	Navigace aplikace	32
5.1.5	Tvorba kvízu	33
5.1.6	Historie odehraných kvízů	34
5.1.7	Detail kvízu	35

5.1.8	Průběh hraní	36
5.2	Serverová část	39
5.2.1	Cloudové funkce	39
5.2.2	Produkční prostředí	40
6	Zhodnocení systému a jeho další vývoj	41
6.1	Ověření funkčnosti implementace	41
6.2	Budoucí vývoj a rozšíření aplikace	43
7	Závěr	44
	Literatura	45
A	Obsah přiloženého paměťového média	47

Seznam obrázků

2.1	Základní princip fungování progresivní webové aplikace. Převzato z [14] . . .	5
2.2	Porovnání časů sestavení pro Webpack a Vite. Převzato z [9]	8
2.3	Diagram demonstrující princip fungování Firebase Cloud Messaging	11
2.4	Ukázka QR kódu, odkazujícího na web FIT VUT	13
3.1	Uživatelské rozhraní tvorby prezentace v aplikaci Mentimeter	15
3.2	Uživatelské rozhraní kvízu v aplikaci Kahoot	16
3.3	Uživatelské rozhraní kvízu v aplikaci Slido	17
4.1	Diagram případu užití z pohledu uživatelů navrhovaného systému	18
4.2	Prvotní návrh stránky pro přidávání kvízů	23
4.3	Prvotní návrh stránky vytvořených kvízů	24
4.4	Prvotní návrh stránky detailu kvízu	24
4.5	Prvotní návrh stránky pro připojení se ke kvízu	25
4.6	Ukázka markdown editoru pro psaní programového kódu	26
5.1	Přihlašování v aplikaci	28
5.2	Registrace v aplikaci	28
5.3	Připojení do kvízu pomocí číselného kódu	29
5.4	Připojení do kvízu pomocí QR kódu	29
5.5	Připojení do kvízu - přihlášený uživatel	30
5.6	Alternativní formulář pro připojení při využití QR kódu	31
5.7	Nástěnka s vytvořenými kvízy	32
5.8	Navigace aplikace	32
5.9	Tvorba kvízu	33
5.10	Historie odehraných kvízů	34
5.11	Stránka podrobností kvízu	35
5.12	Čekací místnost	36
5.13	Informace pro připojení ke kvízu	37
5.14	Hra kvízu - pohled přednášejícího	38
5.15	Správná odpověď studenta	38
5.16	Nesprávná odpověď studenta	38
5.17	Žebříček uživatelů s nejvíce získanými body	39

Kapitola 1

Úvod

Udržet pozornost studentů během výuky představuje pro vyučujícího často náročný úkol. Tato výzva vyžaduje nejen pedagogické dovednosti na velmi dobré úrovni, ale také schopnost udržet zájem studentů po celou dobu výuky. S narůstajícím množstvím distrakcí a vnějších vlivů ve vzdělávacím prostředí se stává stále důležitějším nalézt efektivní metody učení, které budou nejen informativní, ale také podněcující. Zapojení posluchačů do výuky se ukázalo jako jedna z těchto účinných metod, poskytující prostor pro aktivní interakci a rozvoj kognitivních funkcí. To vše přispívá ke zlepšenému učebnímu prostředí, kde studenti nejsou pouze pasivními příjemci informací, ale aktivně se účastní výuky.

Cílem této bakalářské práce je vyvinout webovou aplikaci, která umožní přednášejícím vytvářet atraktivní kvízy pro posluchače. Tato aplikace posluchačům nabídne možnost aktivně se zapojit do výuky tím, že se připojí k vytvořeným kvízům a vybírají správné odpovědi na otázky prezentované během výuky.

Kapitola 2 této práce se zaměřuje na poskytnutí podrobných informací o technologiích použitých při vývoji Progresivní Webové Aplikace (PWA). Bude probíráno, jaké nástroje a postupy jsou nezbytné pro úspěšnou implementaci PWA, včetně aspektů spojených s offline funkcionalitou, manifesty a service workery.

Následně, v kapitole 3, bude proveden průzkum existujících řešení v oblasti interaktivních vzdělávacích platforem. Tato část zkoumá dostupné technologie a přístupy, které již existují na trhu, a analyzuje jejich výhody a nevýhody. Cílem je získat ucelený přehled o stávajících řešeních, který může sloužit jako podklad pro návrh a implementaci nové aplikace.

Kapitola 4 se následně věnuje definici požadavků a omezení navrhovaného systému. Zabývá se bezpečnostními aspekty a návrhem uživatelského rozhraní, což jsou klíčové prvky pro úspěšný vývoj interaktivní vzdělávací platformy. Tato část je klíčová pro následný návrh a implementaci systému, který bude odpovídat potřebám uživatelů a zároveň zajistí bezpečný a uživatelsky příjemný zážitek.

Následující kapitola této práce se zaměřuje na samotnou implementaci navrženého systému. Bude detailně popsáno, jakým způsobem byly využity technologie a postupy z kapitoly 2 při vytváření webové aplikace. Tato část obsahuje informace o konkrétních krocích a rozhodnutích při implementaci, včetně práce s backendem a frontendem, vytváření interaktivních prvků a integrování klíčových funkcionalit systému.

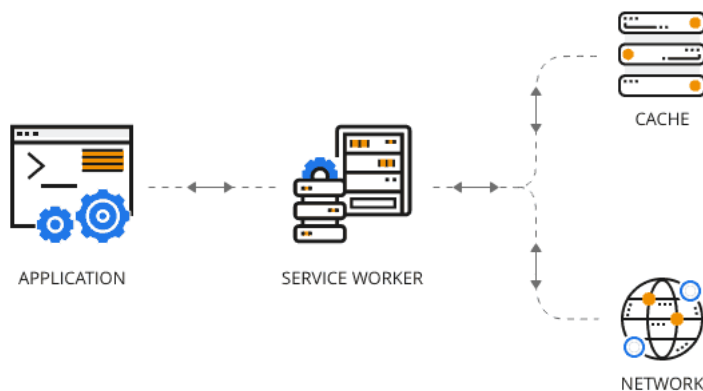
Kapitola 6 se zabývá dotazníkovému šetření a vyhodnocení funkčních i nefunkčních požadavků, dále je zde rozebrán také vývoj aplikace do budoucna.

Poslední kapitola 7 bude sloužit k celkovému zhodnocení dosažených výsledků, dosažených cílů a přínosu navrženého systému.

Kapitola 2

Progresivní webové aplikace

Progresivní webové aplikace představují moderní přístup k vývoji webových stránek a aplikací, který klade důraz na výkonnost a uživatelskou zkušenost. Tyto inovativní aplikace přinášejí několik klíčových výhod. Jednou z hlavních vlastností progresivních webových aplikací je schopnost pracovat offline, což uživatelům umožňuje využívat aplikaci i bez internetového připojení. Využívají konceptu tzv. service workers, což jsou skripty pracující v pozadí a umožňující aplikaci provádět úkoly, jako je synchronizace dat či ukládání dat do vyrovnávací paměti prohlížeče, i když není aktuálně otevřena v prohlížeči [8]. Tato funkce přispívá k lepší uživatelské zkušenosti tím, že umožňuje plynulý přechod mezi online a offline režimem. Aplikace jsou navrženy tak, aby byly responzivní na různá zařízení a obrazovky, nabízejí konzistentní uživatelský zážitek, ať už je používáno jakékoliv zařízení. Rychlá odezva aplikací je dosažena právě díky optimalizovanému kódu a schopnosti ukládání obsahu do vyrovnávací paměti.



Obrázek 2.1: Základní princip fungování progresivní webové aplikace. Převzato z [14]

Další významnou vlastností je možnost automatických aktualizací, což zajišťuje, že uživatelé mají vždy přístup k nejnovějším funkcím aplikace. V oblasti bezpečnosti využívají progresivní webové aplikace standardy, jako je HTTPS [12], které chrání citlivá data uživatelů před neoprávněným přístupem.

Integrované notifikace přispívají k lepší angažovanosti uživatelů i v případě, kdy nejsou aktivně připojeni k aplikaci. Progresivní webové aplikace jsou také optimalizovány pro mo-

bilní zařízení, využívají GPS (viz sekce 2.4), fotoaparát a další dostupné funkce. Díky své škálovatelnosti mohou progresivní webové aplikace efektivně reagovat na rostoucí množství uživatelů a dat. Distribuce těchto aplikací je jednoduchá, a nemusí probíhat přes oficiální obchody s aplikacemi, což zjednodušuje celý proces. Progresivní webové aplikace podporují různé prohlížeče, což zajišťuje jejich širokou dostupnost. Jejich interaktivní uživatelské rozhraní přispívá ke zlepšení celkové uživatelské zkušenosti.

Díky schopnosti automatických aktualizací mají vývojáři možnost rychle reagovat na zjištěné chyby a přidávat nové funkce bez nutnosti čekání na schválení od obchodů s aplikacemi. Vývoj progresivních webových aplikací zahrnuje také optimalizaci načítání obsahu. Využívání technik jako lazy loading, která umožňuje načítání obrázků a jiných prvků pouze tehdy, když jsou potřeba, přispívá celkově k efektivnějšímu načítání stránek a šetří šířku pásma.

Progresivní webové aplikace mají modulární strukturu, což znamená, že lze snadno přidávat a odebírat funkce podle potřeby. Tyto aplikace umožňují personalizaci obsahu a nastavení podle individuálních preferencí uživatelů. V neposlední řadě vývoj těchto aplikací přináší inovace v oblasti webových technologií, posouvá hranice možností pro interakci a zvyšuje přístupnost na webu.

2.1 Role a význam knihovny React pro uživatelské rozhraní

React.js je populární knihovna jazyka JavaScript pro tvorbu uživatelských rozhraní. Jedním z klíčových principů Reactu je používání komponent, což jsou samostatné a znovupoužitelné části kódu, které mohou existovat izolovaně a starají se o svůj vlastní stav a vykreslování na obrazovku.

Tento koncept komponent umožňuje snadno strukturovat kód, vytvářet hierarchie komponent, které lze propojovat, a dosahovat tak lepší modularizace a opakovatelnosti kódu. Každá komponenta v Reactu může mít svůj vlastní stav, který ovlivňuje její chování a vzhled. Změny stavu v komponentě vedou k automatickému překreslení, což zajišťuje aktualizaci uživatelského rozhraní podle aktuálních dat a stavu aplikace.

React umožňuje psát komponenty buď ve funkčním nebo třídním stylu, přičemž funkční komponenty získávají na popularitě díky tzv. hooks [4], jako jsou `useState`¹ a `useEffect`², které umožňují práci s vnitřním stavem a efekty i ve funkčním kontextu. Hierarchie komponent umožňuje propojování a předávání dat mezi komponentami pomocí tzv. `props`, což jsou parametry předávané z rodičovských komponent do potomků.

V Reactu hraje významnou roli koncept jednosměrného toku dat (one-way data binding), kdy informace proudí pouze z rodičovských komponent k potomkům. Tento přístup zvyšuje přehlednost datového toku a usnadňuje ladění kódu.

Každá komponenta v Reactu prochází životním cyklem, který zahrnuje fáze jako inicializace, aktualizace a odstranění. Díky těmto životním cyklům lze vykonávat operace v průběhu různých etap života komponenty, což umožňuje optimalizovat chování a výkon celé aplikace. Kromě toho React podporuje podmíněné vykreslování elementů³, což umožňuje dynamické zobrazování částí uživatelského rozhraní na základě určitých podmínek.

Příklad jednoduché funkční komponenty:

¹Umožňuje komponentám udržovat a aktualizovat svůj vlastní stav

²Umožňuje vykonávat akce v reakci na změny stavu nebo fáze životního cyklu komponenty

³Základní stavební bloky uživatelského rozhraní

```

function Pocitadlo(props) {
  const [count, setCount] = useState(props.defaultCount);
  useEffect(() => {
    console.log(count)
  }, [])
  return (
    <div>
      <p>Aktuální stav count: {count}</p>
      <button onClick = {() => setCount(count + 1)}>Zvýšit count</button>
    </div>
  );
}
export default Pocitadlo;

```

React používá virtuální DOM (Document Object Model⁴), což je abstrakce nad reálným DOM, a tím minimalizuje manipulaci s reálným DOM pro efektivnější aktualizaci obsahu stránky. To zvyšuje výkon aplikace a umožňuje plynulé interakce.

React je ekosystémem doplněn o knihovny a nástroje, jako například Redux pro správu stavu aplikace, React Router pro navigaci v aplikaci nebo Jest pro testování. Tyto nástroje společně s Reactem poskytují komplexní řešení pro vývoj robustních a testovatelných webových aplikací.

V Reactu lze vytvářet jak jednostránkové aplikace (Single Page Applications), tak i vícestránkové aplikace, díky schopnosti vkládat komponenty do různých částí aplikace. Jeho popularita spočívá v jeho jednoduchosti, efektivitě a schopnosti být integrován s různými nástroji a knihovnami, což umožňuje tvorbu moderních a výkonných webových aplikací.

2.2 Sestavování webových aplikací

Sestavování webových aplikací představuje klíčový proces vývoje, který se zabývá konfigurací a optimalizací zdrojových kódů tak, aby bylo možné efektivně a spolehlivě vytvářet a nasazovat webové aplikace. Tato fáze vývoje je klíčová pro dosažení optimálního výkonu, správy závislostí, minimalizace souborů a zajištění kompatibility v různých prohlížečích.

2.2.1 Webpack

Webpack⁵, jeden z nejpoužívanějších nástrojů pro sestavování, slouží k balení a optimalizaci zdrojových kódů aplikací. Tento modulární nástroj umožňuje spojovat a transformovat různé typy souborů, což zjednodušuje správu závislostí a zvyšuje výkonnost aplikace.

Webpack dokáže efektivně zpracovávat moduly, což usnadňuje rozdělení kódu do menších částí a tím zlepšuje udržitelnost a znovupoužitelnost. Navíc, díky zásuvným modulům a loaderům⁶, je možné integrovat různé technologie a optimalizace, jako například kompresi, lazy loading⁷, nebo automatickou minifikaci [1] kódu, což je činnost zahrnující odstraňování nadbytečných prvků a úpravu kódu s cílem zredukovat velikost souboru.

V průběhu sestavování jsou také provedeny další kroky, jako například správa prostředí, tvorba záloh a konfigurace serveru pro produkční nasazení. To vše má za cíl vytvořit efek-

⁴Stromová reprezentace struktury webových dokumentů

⁵<https://webpack.js.org/>

⁶Modul, umožňující transformaci souborů do podoby, kterou lze použít v rámci webové aplikace

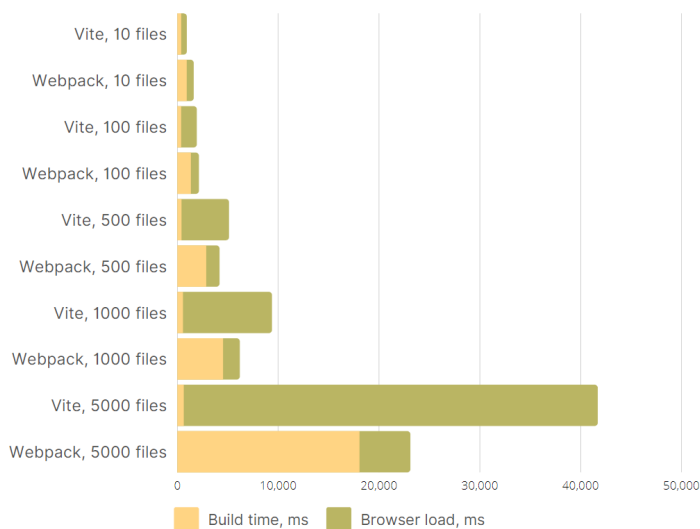
⁷Technika odložení načítání částí kódu

tivní workflow⁸ pro vývojáře a zajistit, že webové aplikace dosahují optimálního výkonu a stability ve finálním produkčním prostředí.

2.2.2 Vite

Vite⁹ je moderní nástroj pro sestavování webových aplikací, který vychází z konceptů, které představil Webpack, a snaží se posunout vývojářský proces na další úroveň. Vite a Webpack jsou oba významné nástroje pro sestavování webových aplikací, ale Vite se často vnímá jako modernější alternativa, zejména pro vývoj jednostránkových aplikací. Jedním z hlavních důvodů, proč někteří vývojáři preferují Vite, je jeho mimořádná rychlost v procesu sestavování.

Vite používá nový koncept, zvaný esbuild, pro sestavování, což umožňuje rychlý vývojový cyklus a snížení doby načítání aplikace. To je zásadní zejména při vývoji, kdy je časté provádět opakované změny v kódu. Webpack na druhou stranu může mít delší dobu sestavování, což může být vnímáno jako nevýhoda v prostředích, kde je kladen důraz na okamžitou zpětnou vazbu a efektivní vývoj.



Obrázek 2.2: Porovnání časů sestavení pro Webpack a Vite. Převzato z [9]

Z grafu výše lze vyčíst, že Vite vždy jednoznačně vítězí nad Webpackem, co se týče času sestavení aplikace. Nicméně je ale patrné, že projekt, který obsahuje 500 a více souborů pro Vite vykazuje delší načítání samotné webové stránky. U webpacku se zdá, že přírůstek času načítání samotné stránky se zvyšuje lineárně s rostoucím počtem souborů, kdežto u Vite se tento čas zvětšuje s rostoucím počtem souborů rapidně.

Celkově lze konstatovat, že jak Webpack, tak Vite, představují velmi silné nástroje pro sestavování webových aplikací, a jejich výběr závisí na konkrétních potřebách a preferencích vývojářského týmu.

⁸Posloupnost kroků, které mají za cíl dosažení konkrétního výsledku

⁹<https://vitejs.dev/>

2.3 Ekosystém Firebase

Firestore¹⁰ je komplexní platforma, nabízející široké spektrum nástrojů pro vývoj moderních webových a mobilních aplikací. Jeho hlavním cílem je usnadnit vývojářům práci tím, že poskytuje soubor plně integrovaných služeb, které pokrývají mnoho klíčových aspektů vývoje.

2.3.1 Realtime databáze

Služba Realtime Database, která představuje nerelační databázi v reálném čase, která umožňuje efektivní synchronizaci dat mezi různými klienty. Data jsou ukládána ve formě JSON objektů, a díky konceptu posluchačů (listenerů) mohou klienti reagovat na okamžité změny v databázi. Tato technologie je vhodná pro aplikace, které vyžadují rychlou aktualizaci dat v reálném čase, například aplikace pro zasílání zpráv nebo kolaborativní nástroje.

Realtime Database podporuje transakce [2], což zajišťuje konzistentní stav dat při současných aktualizacích. Kromě toho umožňuje strukturování dat pomocí klíčů a hodnot, což usnadňuje organizační aspekty vývoje aplikací. Své využití nachází také v offline režimu, kdy klienti mohou pracovat s daty i bez připojení k internetu.

Vývojáři mohou využívat možnosti, jako jsou indexy, filtrování a řazení dat, aby optimalizovali výkon aplikace. Tato flexibilita a schopnost reagovat na změny v reálném čase dělá Firebase Realtime Database atraktivní volbou pro dynamické a interaktivní webové a mobilní aplikace.

2.3.2 Autentizace

Firestore Authentication poskytuje komplexní a bezpečné možnosti správy uživatelských identit v aplikacích. Uživatelé mohou být snadno registrováni pomocí e-mailové adresy a hesla nebo přihlášení prostřednictvím sociálních médií. Mimo jiné je podporována také anonymní autentizace, která umožňuje uživatelům interagovat s aplikací bez nutnosti tradičního přihlašování pomocí e-mailu a hesla. Tato funkce umožňuje přidělení dočasného anonymního účtu, což je užitečné například pro ukládání dočasných dat nebo poskytování omezených funkcí bez potřeby odhalení skutečné identity uživatele. Dvoufázové ověření uživatelů je dalším nástrojem, který posiluje bezpečnost uživatelských účtů. Uživatelé mohou aktivovat tuto funkci, což znamená, že kromě běžného přihlašování potřebují pro ověření poskytnout další údaje, například kód zprávy z mobilní aplikace apod.

Díky nástrojům pro správu uživatelů mohou vývojáři získat informace o uživatelských profilech, sledovat aktivity a zabezpečit přístup k určitým částem aplikace. Tato funkce je klíčová pro zajištění bezpečnosti uživatelských účtů.

2.3.3 Databáze Firestore

Firestore databáze je moderní nerelační databáze od Firebase, která poskytuje škálovatelné a flexibilní řešení pro ukládání a synchronizaci dat. Na rozdíl od Realtime Database (viz sekce 2.3.1), Firestore využívá datový model založený na kolekcích a dokumentech. Tato struktura umožňuje organizovat a strukturovat data mnohem efektivněji. Kolekce slouží k seskupování dokumentů, přičemž každý dokument obsahuje data ve formátu klíč-hodnota.

¹⁰<https://firebase.google.com/>

Firestore umožňuje dotazování nad daty a filtraci výsledků, což usnadňuje získávání specifických informací. Využívá také možnosti offline režimu a automatické synchronizace dat, což zajišťuje konzistentní uživatelskou zkušenost i při omezeném nebo žádném připojení k internetu.

Integrace Firestore s ostatními službami Firebase umožňuje vytvářet komplexní aplikace, které efektivně využívají data pro personalizaci obsahu, analýzy a sledování uživatelské aktivity.

2.3.4 Firebase úložiště

Firestore je cloudová služba od společnosti Google, která umožňuje bezpečné a efektivní ukládání, načítání a sdílení souborů v cloudu. Služba je navržena tak, aby byla snadno integrovatelná s dalšími službami Firebase, což umožňuje vývojářům jednoduše spravovat multimediální obsah v jejich aplikacích.

Ukládání a Načítání Souborů Firestore poskytuje úložiště pro různé typy souborů, jako jsou obrázky, videa, audia nebo dokumenty. Soubory jsou organizovány do tzv. „buckets“ (kbelíků), což jsou základní úložné jednotky. Tato struktura usnadňuje organizaci a řízení přístupu k souborům.

Bezpečnost a Omezení Přístupu Firestore umožňuje nastavit pravidla zabezpečení pro kontrolu přístupu k souborům. Tato pravidla lze konfigurovat na základě uživatelských identit nebo jiných parametrů, což zajišťuje, že soubory jsou chráněny a přístup k nim je spravován podle potřeb aplikace.

2.3.5 Cloudové funkce

Služba Cloud Functions umožňuje vývojářům psát a nasazovat části kódu, běžící na serveru bez nutnosti spravovat vlastní serverovou infrastrukturu. Tyto funkce mohou být spouštěny v reakci na různé události, jako jsou změny v databázi, přihlášení uživatele nebo HTTP¹¹ požadavky.

Cloud Functions poskytují možnost vytvářet vlastní aplikační logiku na straně serveru a reagovat na specifické události v aplikaci. Vývojáři mohou využívat různé programovací jazyky, ať už JavaScript, Python nebo Go, a mít tak flexibilitu ve výběru nástrojů pro konkrétní úlohy. Další výhodou je možnost reagovat na různé události v aplikaci, bez ohledu na to, zda jde o změny v databázi, přihlášení uživatele nebo HTTP požadavky, vývojáři mohou definovat, kdy a jak mají být jejich funkce spouštěny. Tímto způsobem mohou vývojáři efektivně reagovat na specifické potřeby aplikace a optimalizovat její výkon.

Cloudové funkce lze využít například pro:

- Zaslání uvítacího e-mailu po úspěšné registraci uživatele
- Zaslání e-mailového potvrzení při přihlášení k odběru novinek aplikace
- Odeslání SMS s potvrzovacím kódem při vytváření účtu uživatele
- Automatické zpracování profilového obrázku při nahrání nového souboru

Při správném využití mohou Cloud Functions zvýšit modularitu aplikací tím, že umožní vyčlenit specifickou funkcionalitu do samostatných částí kódu. Tímto způsobem se minimalizuje komplexnost aplikace a je umožněno jednoduché rozšiřování aplikace a její údržba.

¹¹Protokol určený pro přenos hypertextových dokumentů po internetu

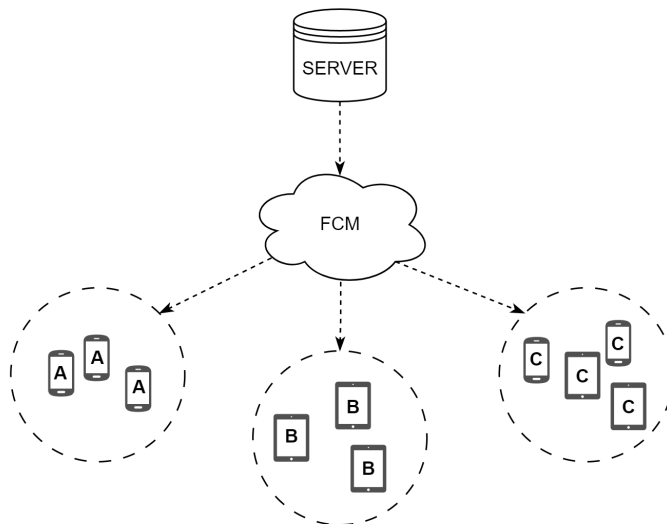
V neposlední řadě Cloud Functions přináší škálovatelnost, což znamená, že aplikace mohou bezpečně a účinně zvládat nápor uživatelských požadavků. To je klíčový faktor pro poskytování stabilních a rychlých služeb uživatelům bez nutnosti velkých investic do výpočetní infrastruktury.

2.3.6 Nasazení aplikace do produkčního prostředí

Pro účely nasazení aplikace do produkčního prostředí poskytuje Firebase službu Firebase Hosting. Jednou z hlavních výhod této služby je její snadná konfigurace a nasazení. S použitím příkazů v příkazovém řádku lze jednoduše nahrát obsah webové stránky na cloudové servery Firebase. Při nasazování aplikace lze využít také možnosti tzv. „deployment hooks“, což jsou skripty, které se spustí při každém úspěšném nasazení. To umožňuje automatické provádění dalších akcí, jako je spuštění testů nebo aktualizace dalších služeb. Hosting automaticky spravuje SSL [11] šifrování pro zabezpečené připojení, což přispívá k bezpečnosti přenášovaných dat. Firebase Hosting zajišťuje rychlé doručení obsahu webové stránky po celém světě díky distribuci souborů na Content Delivery Network (CDN) serverech¹². Tato distribuce na geograficky vzdálené servery minimalizuje latenci a umožňuje uživatelům rychlý přístup ke stránkám bez ohledu na jejich polohu.

2.3.7 Cloud Messaging

Firebase Cloud Messaging (FCM) přináší vývojářům možnost odesílat zprávy na jakékoliv zařízení, čímž podporuje různé formy komunikace s koncovými uživateli. Tato služba není omezena pouze na notifikace¹³, ale umožňuje i přenos datových zpráv, což je užitečné pro aplikace, které potřebují předávat komplexnější informace. FCM podporuje cílení zpráv na skupiny zařízení dle různých kritérií, tedy například podle demografických údajů, podle daného tématu či pouze na jedno konkrétní koncové zařízení.



Obrázek 2.3: Diagram demonstrující princip fungování Firebase Cloud Messaging

¹²Distribuované servery umístěné po celém světě, sloužící k efektivnímu doručování obsahu

¹³Zprávy, upozorňující na události nebo aktivity v aplikaci, i když aplikace běží na pozadí nebo je zařízení v režimu spánku

S FCM mohou vývojáři odesílat zprávy prostřednictvím konzoly Firebase nebo pomocí API na straně serveru. Tato flexibilita umožňuje cílení specifických skupin uživatelů a optimalizaci komunikace. Díky možnosti cílení specifické skupiny zařízení lze pro každou takovou to skupinu či samotné zařízení poskytovat personalizovanou zprávu s unikátním obsahem.

2.4 Využití GPS v progresivních webových aplikacích

Globální polohový systém (GPS) představuje satelitní systém určování polohy, který poskytuje přesné geografické informace o daném místě na zeměkouli. Tato technologie umožňuje mobilním či jiným zařízením získat aktuální polohu a využívá se v různých kontextech, včetně implementace v progresivních webových aplikacích (PWA).

V PWA znamená integrace GPS novou úroveň možností pro získávání geografických dat. Díky schopnosti webových aplikací přistupovat k hardwarovým funkcím zařízení, jako je GPS, mohou poskytovat lokalizovaný obsah, navigaci nebo sledování polohy uživatele. To je zvláště užitečné v oblasti geograficky zaměřených služeb, jakými jsou například mapové aplikace, navigační systémy nebo služby na sledování aktivity. Implementace GPS v PWA přináší možnost získávat a využívat geografická data bez nutnosti interakce s uživatelem. Aplikace mohou automaticky získávat aktuální polohu zařízení a využívat tuto informaci pro různé účely, jako je lokalizovaný obsah nebo sledování pohybu. V rámci bezpečnosti je však důležité pečlivě přistupovat k ochraně uživatelského soukromí. Správné a odpovědné zacházení s geografickými údaji je nezbytné pro udržení důvěry uživatelů.

Výhodou GPS v PWA je také offline funkcionality. Aplikace mohou ukládat určité geografické informace a umožňovat uživatelům využívat navigaci nebo další lokalizované funkce i bez aktivního internetového připojení. To může být zvláště užitečné v oblastech s nekonzistentním internetovým pokrytím.

2.5 QR kódy

QR kódy představují formu vizuálních kódů, které obsahují zakódované informace a umožňují rychlý přenos dat. Princip fungování spočívá ve skenování kódu fotoaparátem mobilního či jiného zařízení, což uživatelům umožňuje okamžitý přístup k obsahu bez potřeby manuálního zadávání URL nebo identifikátoru. Hlavním rozdílem oproti klasickému čárovému kódu je to, že QR kódy obsahují druhý rozměr, což umožňuje uchovávat několikanásobně větší množství dat [16].

V kontextu progresivních webových aplikací (PWA) se QR kódy stávají klíčovým prvkem pro zlepšení uživatelské interakce. Využívají se k usnadnění přístupu uživatelů k různým částem aplikace, informacím nebo funkcím, eliminujíc potřebu složitého zadávání identifikátorů. Mohou být implementovány při navigaci ke specifickým úkolům, zobrazování obsahu a dalším interaktivním prvkům, což výrazně zvyšuje pohodlí a rychlost uživatelské interakce.

S postupem technologie se QR kódy stávají více interaktivními, umožňující zakódování komplexnějších dat a interaktivních prvků, jako jsou tlačítka, hlasové příkazy nebo dynamické odkazy. To vytváří prostředí s větší možností personalizace a bohatšími uživatelskými zážitky.

Výhoda QR kódů spočívá v jejich schopnosti fungovat offline. Data zakódovaná v kódu lze uchovat přímo v zařízení uživatele, což umožňuje přístup k nim i v situacích, kdy není dostupné aktivní internetové připojení.



Obrázek 2.4: Ukázka QR kódu, odkazujícího na web FIT VUT

V offline režimu mohou být využity k přístupu k předem staženým informacím, což zvyšuje jejich výhody v oblasti dostupnosti. Tato schopnost umožňuje uživatelům stále využívat klíčové funkce aplikace, i když není k dispozici aktivní internetové připojení.

Kapitola 3

Průzkum existujících řešení

V dnešní době se vyskytuje široká škála systémů, které se věnují interaktivní prezentaci ve vzdělávacím prostředí mezi učiteli a studenty. Avšak žádný z těchto systémů nenabízí přímou podporu integrace programového kódu mezi prezentujícím a účastníky, což je pro tuto bakalářskou práci klíčovým prvkem.

Tato kapitola se zabývá stručnou analýzou již existujících podobných systémů v oblasti interaktivní prezentace v reálném čase. Cílem této analýzy je poskytnout ucelený přehled o stávajících technologických řešeních, která se věnují podobným výzvám a úkolům, jaké představuje tato bakalářská práce. V průběhu této analýzy budou rozebírány klíčové aspekty existujících systémů, jako jsou jejich funkcionalita, uživatelské rozhraní, podpora pro interaktivní prvky a další relevantní parametry.

3.1 Mentimeter

Mentimeter¹ je interaktivní platforma, která umožňuje prezentujícím vytvářet a řídit prezentace s vysokým stupněm interaktivity. Tato služba umožňuje snadné vytváření různých typů otázek, včetně možností s více volbami, hodnocení a otázek s otevřenou odpovědí.

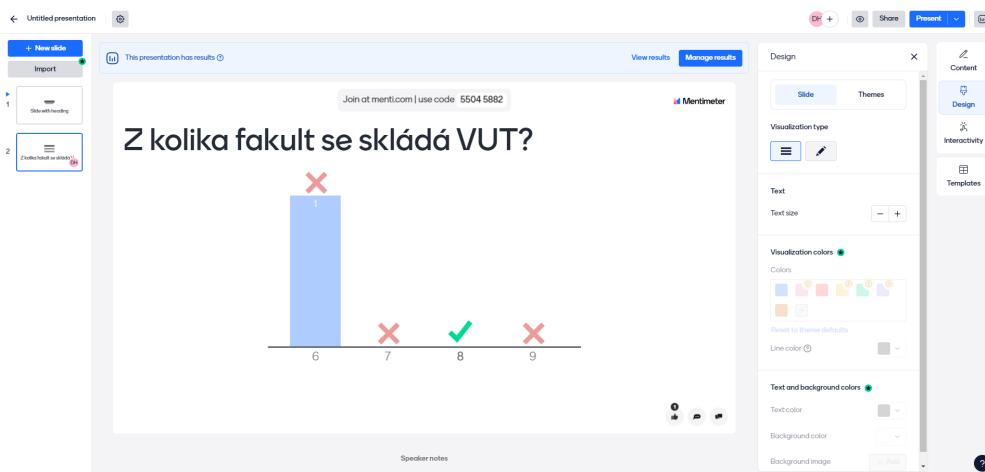
Uživatelé mohou sledovat odpovědi v reálném čase, což dovoluje uživatelům okamžitě vidět výsledky hlasování, nicméně často to bohužel také vede k ovlivnění výběru odpovědi uživatele.

Platforma umožňuje používání aplikace zdarma, nicméně jsou zde zavedena jistá omezení. Mezi nejvíce zásadní omezení tohoto předplatného patří:

- Počet hlasujících uživatelů za měsíc nesmí překročit 50
- Počet snímků prezentace je limitován na 34
- Nemožnost exportu dat do tabulky
- Nemožnost změny atributů pozadí snímku či elementů na snímku (barva pozadí, textu apod.)
- Nelze položit více než jednu otázku na snímku
- Nelze nastavit, kdo se může připojit k prezentaci

¹<https://www.mentimeter.com/>

Při vytváření samotných prezentací či kvízů má prezentující na výběr z několika typů snímků, což mírně kompenzuje nemožnost tyto snímky detailnějším způsobem upravovat. Lze také při tvorbě prezentace zobrazit náhled, jak prezentaci uvidí samotní diváci, nicméně tento náhled poté při samotném prezentování již není k dispozici.



Obrázek 3.1: Uživatelské rozhraní tvorby prezentace v aplikaci Mentimeter

Aplikace neumožňuje automatické přepínání mezi snímky pomocí časovače, tedy prezentující je mezi nimi nucen přepínat ručně. Mentimeter dovoluje tvorbu prezentací, kvízů či dotazníků a poskytuje mnoho šablon, se kterými prezentující mohou začít tvořit. Aplikace umožňuje připojení do konkrétní prezentace pomocí osmimístného číselného kódu nebo také pomocí QR kódu. Aplikace po skončení prezentace umožňuje zobrazit její základní vyhodnocení a statistiky. Celkově lze říct, že Mentimeter představuje silné řešení pro ty, kteří hledají efektivní prostředek k zapojení svého publika a zvýšení interaktivity svých prezentací.

3.2 Kahoot

Kahoot² je interaktivní aplikace, která se stala oblíbeným nástrojem pro vytváření a správu kvízů, testů a her s vysokým stupněm zapojení publika. Jedná se o uživatelsky přívětivou službu, která umožňuje učitelům, trenérům či jiným prezentujícím snadno vytvářet a sdílet vzdělávací hry.

Tvorba kvízů v Kahoot je snadná a intuitivní. Uživatelé mohou vytvářet otázky s více volbami, maximálně však šesti, položky s otevřenou odpovědí a další. Platforma umožňuje přidávat obrázky a videa, čímž se zvyšuje vizuální atraktivita kvízů.

Aplikaci je taktéž možné používat zdarma, avšak mezi hlavní omezení tohoto předplatného patří:

- Maximální počet uživatelů ve kvízu je omezen na 10
- Obrázky nelze označit jakožto odpovědi
- Nelze používat jezdec (slider) pro výběr odpovědi

²<https://kahoot.com/>

- Pouze jeden typ rozložení objektů na snímku

Herní prvky, jako jsou body a žebříčky, přispívají k vytváření soutěživého prostředí, které zvyšuje motivaci účastníků. Kahoot rovněž poskytuje možnost okamžité zpětné vazby po každé otázce, což umožňuje uživatelům rychle porozumět správným odpovědím a udržovat aktivitu publika. Po každém kvízu si prezentující může zobrazit jeho detailní výsledky a statistiky. Lze také zobrazit statistiky pro jednotlivé uživatele, kteří na kvíz odpovídali. Kahoot neumožňuje uživatelům z publika pokládat otázky na prezentujícího, nicméně lze po skončení kvízu odeslat jednoduchou zpětnou vazbu.



Obrázek 3.2: Uživatelské rozhraní kvízu v aplikaci Kahoot

Během samotné prezentace otázek se zobrazuje jednoduché rozhraní, které zahrnuje otázku a možnosti odpovědí. Barvy jsou často velmi výrazné. Po každé otázce jsou výsledky okamžitě zobrazeny ve formě grafu. To přináší do hry dojem soutěže a interaktivity.

Platforma Kahoot je přístupná prostřednictvím webového prohlížeče i mobilních zařízení, což umožňuje připojení z téměř kteréhokoli zařízení. Mimo jiné také používá Vite (viz sekce 2.2.2), což umožňuje sestavovat a spouštět aplikaci několikanásobně rychleji než při využití klasického Webpacku (viz sekce 2.2.1)[5].

3.3 Slido

Slido³ je významná platforma, která se specializuje na interaktivní způsoby zapojení publika během různých událostí, konferencí a prezentací. Tato aplikace nabízí široké spektrum nástrojů, které zvyšují zapojení publika a umožňují mu tak sdílet své názory a otázky v reálném čase.

Jedním z klíčových prvků Slido je možnost vytvářet otázky a ankety, které mohou účastníci okamžitě zodpovídat. Tato interaktivní forma komunikace umožňuje prezentátorům okamžitě reagovat na potřeby publika, ať už se jedná o zjišťování názorů, sběr otázek nebo společnou tvorbu obsahu.

Samotná tvorba prezentace je intuitivní a jednoduchá, po zadání jména prezentace se zobrazí výběr různých typů anket a odpovědí. Uživatelské rozhraní je jednoduché a nepůsobí

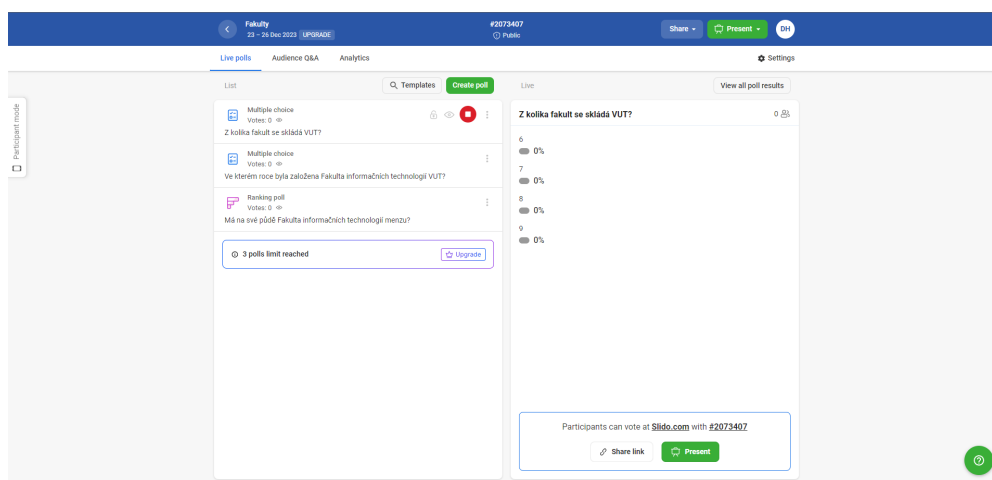
³<https://www.slido.com/>

rušivě, jednotlivé bloky tohoto rozhraní jsou strategicky umístěny tak, aby je uživatel velmi rychle našel.

Platformu lze, stejně jako předchozí dvě, používat zcela bezplatně s jistými omezeními, mezi které patří například:

- Maximální počet účastníků, připojených na dané události je limitován na 100
- Počet vytvořených otázek prezentujícím může být nejvíce 3

Slido podporuje i mód, kde účastníci mohou klást otázky přímo prezentujícím a ti pak mohou buď odpovídat v průběhu prezentace nebo si otázky ponechat pro speciální část. Slido dále nabízí funkcionalitu, která umožňuje moderovat otázky a odpovědi, což pomáhá udržovat vysokou úroveň obsahu a eliminuje riziko nevhodných či opakujících se dotazů.



Obrázek 3.3: Uživatelské rozhraní kvízu v aplikaci Slido

Díky analytickým nástrojům Slido mohou organizátoři událostí sledovat úroveň účasti, zhodnocovat oblíbené otázky či témata a získávat cenné zpětné vazby pro budoucí akce. V oblasti bezpečnosti Slido pečlivě chrání osobní údaje účastníků a zajišťuje, že veškerá komunikace, včetně přenosu hesel a dalších citlivých informací, je důsledně šifrovaná a chráněná.

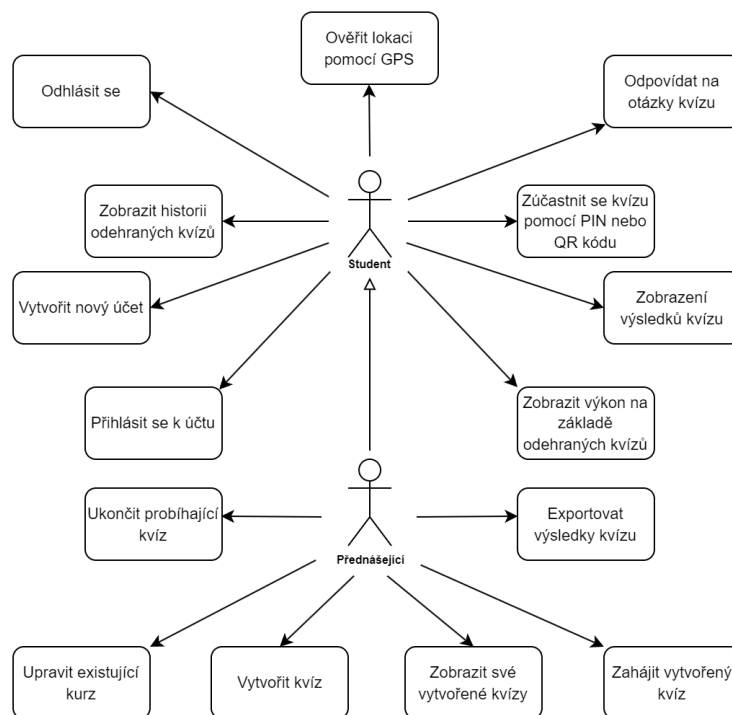
Slido není omezeno pouze na konkrétní typy událostí – využívá se na konferencích, školeních, týmových setkáních i virtuálních akcích. Aplikace je dostupná online a je kompatibilní s různými platformami, což umožňuje širokou dostupnost a flexibilitu při jejím používání.

Kapitola 4

Návrh a architektura systému

Tato kapitola se zabývá podrobným popisem návrhu a architektury systému pro platformu podporující interaktivitu ve výuce, která je realizována jakožto progresivní webová aplikace (viz sekce 2).

Pro plné pochopení funkčnosti systému je klíčové získat přehled o akcích, které budou uživatelé schopni v systému provádět. Jedním z efektivních nástrojů pro tuto definici je vytvoření diagramu případu užití. Tento diagram vizualizuje různé scénáře a interakce mezi uživateli a systémem, identifikuje klíčové funkce a pomáhá navrhnout uživatelské rozhraní tak, aby co nejlépe odpovídalo potřebám uživatelů. Vytvoření takového diagramu umožňuje lepší pochopení očekávaného chování systému a přispívá k jeho efektivnímu designu a implementaci.



Obrázek 4.1: Diagram případu užití z pohledu uživatelů navrhovaného systému

I když je v diagramu uživatel oddělen na přednášejícího a studenta, aplikace systém rolí neřeší. Jedná se pouze o rozdělení kvůli přehlednosti diagramu. Pro výběr vhodných technologií, následnou implementaci a nadimenzování systému za účelem spolehlivosti je nutné získat informace o následujících základních bodech.

- Jaký typ zařízení bude využíván pro přístup k aplikaci?
- Je nutné, aby systém reagoval s minimální odezvou nebo lze připustit malá zpoždění?
- Jakým způsobem lze zajistit bezpečnost aplikace?
- Kolik místa bude průměrně zabírat jeden kvíz v databázi?

4.1 Požadavky, předpoklady a omezení

Aplikace je primárně cílena do výuky a lze předpokládat, že většina posluchačů bude používat mobilní zařízení, tablet nebo notebook. Z tohoto důvodu bude aplikace přizpůsobena a optimalizována právě pro tato zmíněná zařízení.

Existuje mnoho služeb, poskytující databázová řešení, nicméně rozhodl jsem se využít ekosystém Firebase (viz sekce 2.3), který poskytuje sadu služeb a nástrojů pro správu serverové části aplikace. Jedná se o intuitivní, moderní a spolehlivý ekosystém, který lze využívat do jisté míry také zdarma.

Aby bylo možné vypočítat celkové potřebné místo v databázi pro parametry popsané výše, je nutné vědět, jakým způsobem Firebase ukládá jednotlivé datové typy a dokumenty do své databáze¹.

Uvažujme následující:

Mějme jeden kvíz, ke kterému je připojeno 100 aktivních uživatelů. Kvíz má pět otázek (každá má v průměru 45 znaků) a každá otázka má na výběr ze čtyř možných odpovědí (každá odpověď má průměrně 20 znaků) a pouze jedna odpověď je správná. Odpovědi jsou u každé otázky vždy v kombinaci čtyři textové bloky. Kolik místa bude v databázi zabírat jeden takový kvíz?

Tento problém je neprve nutno pro přehlednost rozdělit do čtyř samostatných výpočtů. Nejdříve je nutno vypočítat potřebné místo pro kolekci s uživateli, dále je potřeba vypočítat místo pro kolekci otázek, odpovědí a jako poslední je třeba vypočítat potřebné místo atributů daného kvízu. Poté již zbývá sečíst vše dohromady. Pro celkovou velikost daného dokumentu si samotná databáze ještě přičítá dalších 32 B a pro dokumenty 16 B. Ve výpočtech se bude vyskytovat proměnná `PATH`, která značí velikost cesty k danému dokumentu.

¹<https://firebase.google.com/docs/firestore/storage-size>

Atribut	Velikost v bajtech	Popis
id	24	Jednoznačné označení kvízu
vytvoreno	18	Určuje datum vytvoření kvízu
vytvoril	30	Určuje tvůrce kvízu
pin	11	PIN kód pro připojení uživatelů do kvízu
live	6	Značí, zda je kvíz momentálně spuštěn
pocet_odpovedi	23	Celkový počet zaznamenaných odpovědí v průběhu kvízu
pocet_otazek	21	Celkový počet otázek kvízu
pocet_uzivatelu	24	Celkový počet uživatelů kvízu
vyherce	29	Výherce kvízu
otazka	53	Text otázky
odpoved	29	Odpověď pro danou otázku
spravny	9	Slouží k označení správné odpovědi
pocet_zvoleni	22	Celkový počet uživatelů, kteří odpověď označili jako správnou
overen	8	Slouží k identifikaci, zda účastník kvízu ověřil svou lokaci
body	13	Celkový počet bodů, které účastník získal během kvízu
aktualni_otazka	24	Udržuje informace, na kterém snímku se kvíz nachází
poradi	15	Informuje, kolikátá v pořadí je daná otázka / odpověď kvízu
nazev	27	Název kvízu
typ	7	Slouží k rozlišení odpovědi: 't'- text, 'i'- image (obrázky), 'c'- code (programový kód), 'cq' - code question(programový kód součástí otázky)

Tabulka 4.1: Uchovávané atributy kvízu v databázi

1) Výpočet velikosti kolekce uživatelů:

$$\text{uzivatele} = (\text{PATH} + \text{body} + \text{overen} + 16 + 32) * 100$$

Po dosazení čísel získáme následující rovnici.

$$\begin{aligned}\text{uzivatele} &= (50 + 13 + 8 + 16 + 32) * 100 \\ \text{uzivatele} &= \underline{\underline{11\ 900\ B}}\end{aligned}$$

2) Výpočet velikosti kolekce odpovědí:

$$\begin{aligned}\text{odpovedi} &= (\text{PATH} + \text{odpoved} + \text{pocet_zvoleni} + \text{poradi} \\ &\quad + \text{spravny} + \text{typ} + 16 + 32) * 20\end{aligned}$$

Opět dosadíme čísla a získáme následující rovnici.

$$\begin{aligned}\text{odpovedi} &= (82 + 29 + 22 + 15 + 9 + 7 + 16 + 32) * 20 \\ \text{odpovedi} &= \underline{\underline{4\ 240\ B}}\end{aligned}$$

3) Výpočet velikosti kolekce otázek:

$$\begin{aligned}\text{otazky} &= (\text{PATH} + \text{otazka} + \text{poradi} + \text{vyherce} \\ &\quad + 16 + 32) * 20\end{aligned}$$

Pokud znovu dosadíme čísla, získáme následující rovnici.

$$\begin{aligned}\text{otazky} &= (54 + 53 + 15 + 29 + 16 + 32) * 5 \\ \text{otazky} &= \underline{\underline{995\ B}}\end{aligned}$$

4) Výpočet velikosti dokumentu kvízu:

$$\begin{aligned}\text{kviz} &= (\text{PATH} + \text{aktualni_otazka} + \text{id} + \text{live} + \text{nazev} + \text{pin} \\ &\quad + \text{pocet_odpovedi} + \text{pocet_otazek} + \text{pocet_uzivatelu} \\ &\quad + \text{vyherce} + \text{vytvoreno} + \text{vytvoril} + 16 + 32)\end{aligned}$$

Pokud naposledy dosadíme čísla, získáme následující rovnici.

$$\begin{aligned}\text{kviz} &= (28 + 24 + 24 + 6 + 27 + 11 + 23 + 21 + 24 + 29 + 18 + 30 + 16 + 32) \\ \text{kviz} &= \underline{\underline{313\ B}}\end{aligned}$$

Pokud nyní sečteme veškeré dílčí výpočty, získáme tak představu o tom, kolik místa bude pro daný kvíz potřeba vyhradit v databázi. Tedy rovnice bude vypadat následovně:

$$\begin{aligned}\text{celkem} &= \text{kviz} + \text{uzivatele} + \text{otazky} + \text{odpovedi} \\ \text{celkem} &= 313 + 11\,900 + 995 + 4\,240 \\ \text{celkem} &= 17\,448 \text{ B} \approx \underline{17.04 \text{ KiB}}\end{aligned}$$

Databáze Firestore je v rámci svého bezplatného plánu limitována velikostí dat, které zde mohou být uloženy. Tento limit je 1 GiB (1 048 576 KiB) a nelze ho překročit bez příplatku. Pokud bychom uvažovali, že takových kvízů bude provozováno maximálně deset denně, tedy bude ukládáno $10 * 17.04 \text{ KiB} = 170.4 \text{ KiB}$ za den, lze říct, že takovýto systém bude schopen provozu po dobu danou vztahem $1\text{GiB}/170.4\text{KiB} \approx 6\,153$ dní ≈ 17 let. Jedná se o nejhorší možný případ, jelikož celková velikost je počítána se sto aktivními uživateli, kteří zabírají nejvíce místa v databázi pro daný kvíz. Dále se jedná o případ, kdy by kvízy nebyly vymazávány. Lze předpokládat, že tato velikost bude mnohem menší a tedy maximální doba provozu systému se exponenciálně zvýší.

4.2 Bezpečnost systému

Bezpečnostní opatření jsou klíčovým prvkem v návrhu a implementaci systému, zejména při využití Firebase Authentication (viz sekce 2.3.2), Cloud Functions (viz sekce 2.3.5) a SSL šifrování. Následuje podrobný popis bezpečnostních opatření a jejich implementace v rámci aplikace.

Firebase Authentication bude hrát centrální roli v zajištění bezpečného přístupu uživatelů k aplikaci. Hesla uživatelů budou zabezpečena pomocí pokročilého hašovacího algoritmu s názvem Scrypt. Scrypt je navržen tak, aby poskytoval vysokou úroveň odolnosti vůči útokům typu brute force a dalším metodám prolomení hesel.

Tímto způsobem budou citlivá hesla uživatelů zakódována do nečitelné podoby, což výrazně snižuje riziko odhalení citlivých informací v případě, že by došlo k neoprávněnému přístupu k úložišti dat. V kombinaci s dalšími bezpečnostními opatřeními poskytuje scrypt efektivní ochranu proti různým formám kybernetických hrozeb, a to i v případě, že by došlo k úniku uživatelských údajů.

Využití Cloud Functions představuje klíčový prvek v architektuře aplikace, kde slouží k efektivní správě serverové logiky a zabezpečení operací prováděných v aplikaci. Distribuce serverové logiky na serverovou infrastrukturu umožňuje centralizovanou a bezpečnou správu operací, což snižuje riziko neoprávněného přístupu a zvyšuje úroveň bezpečnosti aplikace. Cloud Functions zajišťují, že klíčové operace a výpočetně náročné úlohy jsou prováděny na serverové straně, což snižuje zátěž na zařízení uživatele a zlepšuje celkový výkon aplikace. To je zásadní zejména v kontextu bezpečnosti, protože minimalizuje možnost zneužití klientského prostředí. Tímto způsobem Cloud Functions přispívají k bezpečnému a efektivnímu provozu aplikace, což je klíčový aspekt při návrhu a implementaci robustního a spolehlivého systému. Vstupní data budou pečlivě ověřována a filtrována tak, aby se zabránilo SQL nebo XSS [6] injekcím. Cloud Functions budou používat parametrizované dotazy k minimalizaci rizika bezpečnostních incidentů. Při manipulaci s citlivými daty, jako jsou osobní informace uživatelů, budou Cloud Functions využívat šifrování, což zajišťuje, že data zůstanou

bezpečná i v případě, že dojde k neautorizovanému přístupu. Přístup k jednotlivým Cloud Functions bude spravován pomocí Firebase Authentication, což zaručí, že pouze oprávnění uživatelé budou moci volat a provádět tyto funkce. Jelikož i Cloud Functions mají svá omezení, která zahrnují maximální počet volání funkce při čtení nebo zápisu, je nutno je volat pouze pro klíčové účely, tedy v určitých případech bude nutno počítat s případným malým zpožděním.

Pro zabezpečení komunikace mezi klienty a serverem bude využito SSL šifrování, které je poskytováno službou Firebase Hosting. Tato služba přináší automatickou konfiguraci SSL šifrování, což eliminuje potřebu ruční konfigurace a zjednodušuje proces zajištění bezpečné komunikace. Certifikáty použité pro SSL šifrování jsou pravidelně aktualizovány a pečlivě monitorovány. Tím bude zajištěno, že komunikace probíhá s důvěryhodnými a aktuálními certifikáty, což minimalizuje riziko útoku typu „man-in-the-middle“². Důraz na pravidelnou aktualizaci certifikátů je klíčový pro udržení vysoké úrovně bezpečnosti a ochrany před potenciálními bezpečnostními hrozbami.

Celkově tedy využití SSL šifrování poskytovaného Firebase Hostingem představuje efektivní a spolehlivý způsob, jak zabezpečit veškerou komunikaci v rámci aplikace a chránit citlivá data před neoprávněným přístupem a manipulací.

4.3 Uživatelské rozhraní

Uživatelské rozhraní představuje klíčový prvek aplikace, který ovlivňuje celkový dojem uživatelů a jejich schopnost pracovat se samotným systémem [7]. V této sekci si popíšeme jeho návrh a strukturu nejdůležitějších částí.

Panel tvorby kvízu

Tento panel byl navržen tak, aby měl uživatel vše potřebné po ruce, tedy prvky, které lze do kvízu přidávat se nachází na vrchním místě obrazovky, nalezneme zde možnost přidávání bloku kódu, textu či obrázků.



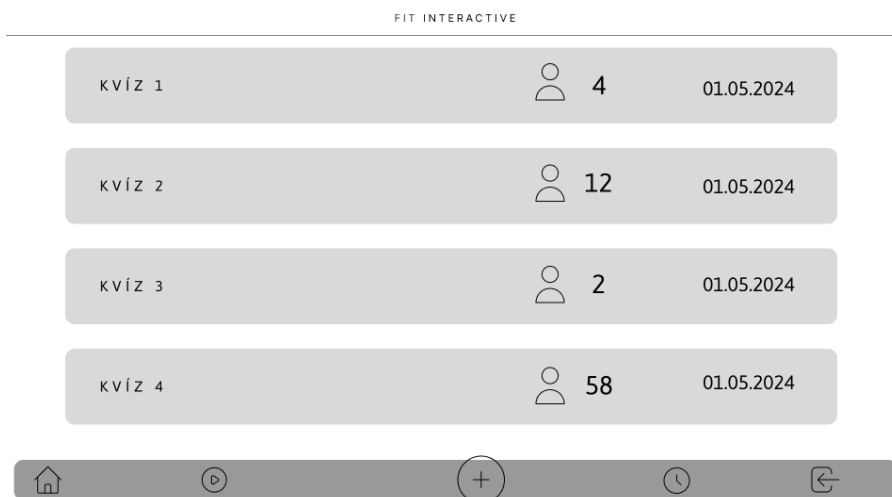
Obrázek 4.2: Prvotní návrh stránky pro přidávání kvízů

²https://cs.wikipedia.org/wiki/Man_in_the_middle

Přehled vytvořených kvízů

Tato stránka byla navržena s cílem poskytnout uživatelům přehled o jejich vytvořených kvízech. Umožňuje uživatelům procházet a editovat vytvořené kvízy, tedy samozřejmě poskytuje detaily o každém jednotlivém kvízu po kliknutí na něj.

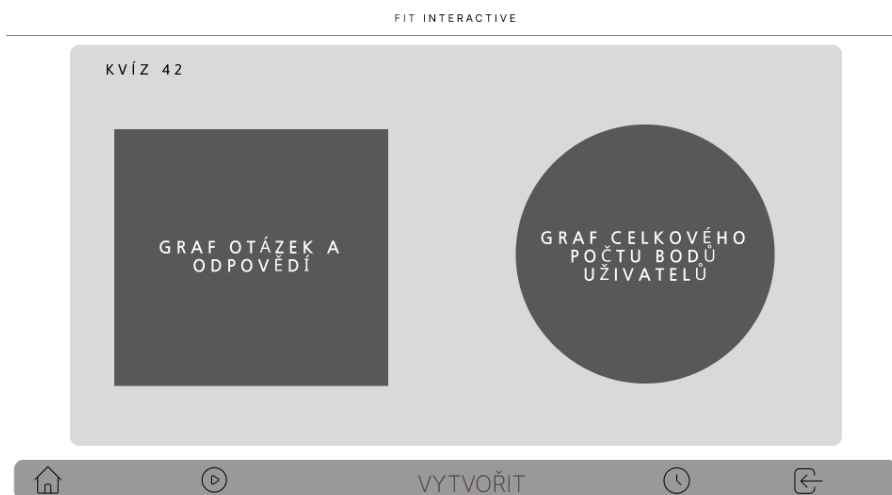
Na stránce budou zobrazeny informace o dříve vytvořených kvízech, včetně jejich názvu či počtu zúčastněných uživatelů a data vytvoření. Uživatel má možnost procházet jednotlivé kvízy a získávat podrobnosti, jako jsou otázky, na které uživatelé odpověděli, a jejich výsledky.



Obrázek 4.3: Prvotní návrh stránky vytvořených kvízů

Detaily odehraného kvízu

Po rozkliknutí kvízu vidíme detaily, které obsahují kompletní informace o daném kvízu. Zahrnují seznam otázek, na které uživatelé odpovídali, spolu s odpověďmi, které poskytli. Tímto způsobem přednášející získává přehled o silných a slabých stránkách studentů.



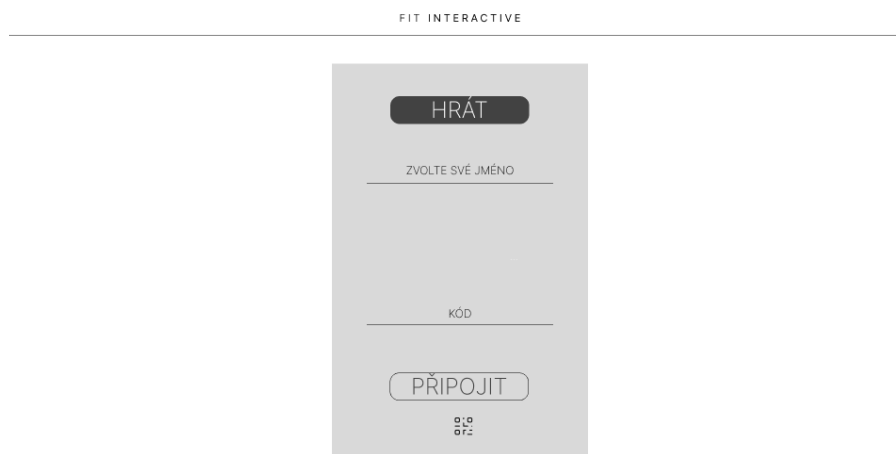
Obrázek 4.4: Prvotní návrh stránky detailu kvízu

Detaily kvízu také budou obsahovat celkové skóre, které uživatelé dosáhli. Tato data slouží k posouzení výkonnosti a úspěšnosti uživatelů v daném kvízu. Kromě toho mohou uživatelé porovnávat své výsledky mezi různými kvízy a sledovat pokrok v průběhu času.

Připojení se ke kvízu

Stránka pro připojení se do kvízu je klíčovým prvkem aplikace, který umožňuje uživatelům aktivně se zapojit do výuky. Na této stránce mohou uživatelé zvolit možnost připojení pomocí PIN kódu nebo snadno naskenovat QR kód, což zjednodušuje a urychluje celý proces. Aplikace automaticky rozpozná kód a provede uživatele k připojení do odpovídajícího kvízu.

Pro připojení pomocí PIN kódu uživatel jednoduše zadá přidělený šestimístný unikátní kód do příslušného pole a stiskne tlačítko pro připojení. Tato metoda je rychlá a efektivní, což je zejména užitečné v prostředích, kde není možné nebo vhodné skenovat QR kódy.



Obrázek 4.5: Prvotní návrh stránky pro připojení se ke kvízu

Aplikace bude také poskytovat uživatelům možnost efektivně spravovat své účty a udržovat je aktuální. To zahrnuje jednoduchý proces změny uživatelských údajů jako jsou jméno či heslo. Na stránce správy účtu bude uživatel mít možnost aktualizovat své osobní informace prostřednictvím jednotlivých polí, která umožní editaci konkrétních údajů. Tímto způsobem bude proces změny údajů intuitivní a přístupný pro všechny uživatele.

4.4 Programový kód

Vkládání programového kódu je klíčovým prvkem návrhu, a právě k tomu bude využit existující Markdown editor pro React. Tento editor představuje uživatelsky příjemné rozhraní pro zadávání a formátování programových částí během tvorby kvízů a prezentací. Editor umožňuje uživatelům snadno vkládat a upravovat kód v textové podobě a rozpoznává syntaxi různých programovacích jazyků. Díky této funkci bude psaní kódu pro uživatele efektivní a pohodlné. V editoru lze psát kód a zároveň lze vidět jeho finální vizuální stránku.

Součástí editoru jsou také nástroje pro zvýrazňování syntaxe, což přispívá k lepší vizualizaci kódu. Uživatelé tak mohou s jistotou vkládat a prezentovat části kódu s optimálním

The image shows a dark-themed code editor with two panels. The left panel contains Python code with syntax highlighting: a docstring 'python', a function definition 'def calculate(num):' with 'return 2 * num' inside, and a call 'print(calculate(2))'. The right panel shows the same code rendered in a lighter font, demonstrating how the code is displayed in a markdown environment. The code is:

```
python
def calculate(num):
    return 2 * num

print(calculate(2))
```

Obrázek 4.6: Ukázka markdown editoru pro psaní programového kódu

formátováním a barevným zvýrazněním klíčových prvků. Využití existujícího Markdown editoru pro React usnadňuje integraci této funkcionality do celkové struktury projektu.

Zahrnutí takového editoru do systému poskytuje široké možnosti pro vytváření interaktivních a obsahově bohatých výukových materiálů. Uživatelé mohou snadno vytvářet kvízy a prezentace s přesným formátováním kódu a jeho názornou vizualizací.

Díky možnosti vkládání programového kódu budou prezentace a kvízy atraktivnější a lépe odpovídající potřebám IT orientovaných kurzů. Učitelé a lektori tak mohou efektivněji prezentovat a zdůraznit klíčové aspekty výuky prostřednictvím konkrétních kódových příkladů.

Nabízelo se využívat markdownu i pro vkládání obrázků, nicméně od tohoto jsem upustil, jelikož spousta uživatelů s markdownem nemusí umět pracovat a tedy pro ně bude pohodlnější svůj obrázek pouze přetáhnout a nahrát jej do aplikace.

Kapitola 5

Implementace

Tato kapitola se podrobně zaměřuje na samotnou tvorbu webové aplikace, nastavením databáze a implementací interaktivních prvků, podporujících aktivní zapojení studentů během výuky. Byla zvolena architektura klient-server. Tato architektura umožňuje centralizované zpracování dat a logiky aplikace na serverové straně, což zajišťuje konzistenci a bezpečnost dat, zatímco uživatelé mohou interagovat s aplikací pomocí svých prohlížečů, aniž by museli provádět složité operace, či operace nesoucí bezpečnostní riziko, na svých zařízeních. Pro tyto účely bylo využito cloudových funkcí 2.3.5, které budou detailně popsány níže, a poskytnutého API [3] ekosystému Firebase (viz sekce 2.3).

Aplikace je vytvořena s ohledem na responzivní design, což znamená, že svůj vzhled dokáže přizpůsobit zejména velikosti zařízení uživatele.

5.1 Klientská část

Pro tvorbu klientské části aplikace byla zvolena knihovna React (viz sekce 2.1). Jednotlivé části aplikace byly následně vhodně rozděleny do samostatných komponent, což umožňuje lepší organizaci kódu a snadnější údržbu. Každé komponentě je přidělena specifická role uživatelského rozhraní, což umožňuje lépe řídit různé aspekty aplikace, jako je přihlašování, zobrazování dat nebo interakce s uživatelem. Použití Reactu a komponentové architektury je zdrojem celkové efektivity a flexibility klientské části aplikace. Pro vzhled jednotlivých prvků bylo použito kaskádových stylů [15]. Aplikace je schopna fungovat v offline režimu, pokud ji má uživatel nainstalovanou, uživatel je na toto upozorněn.

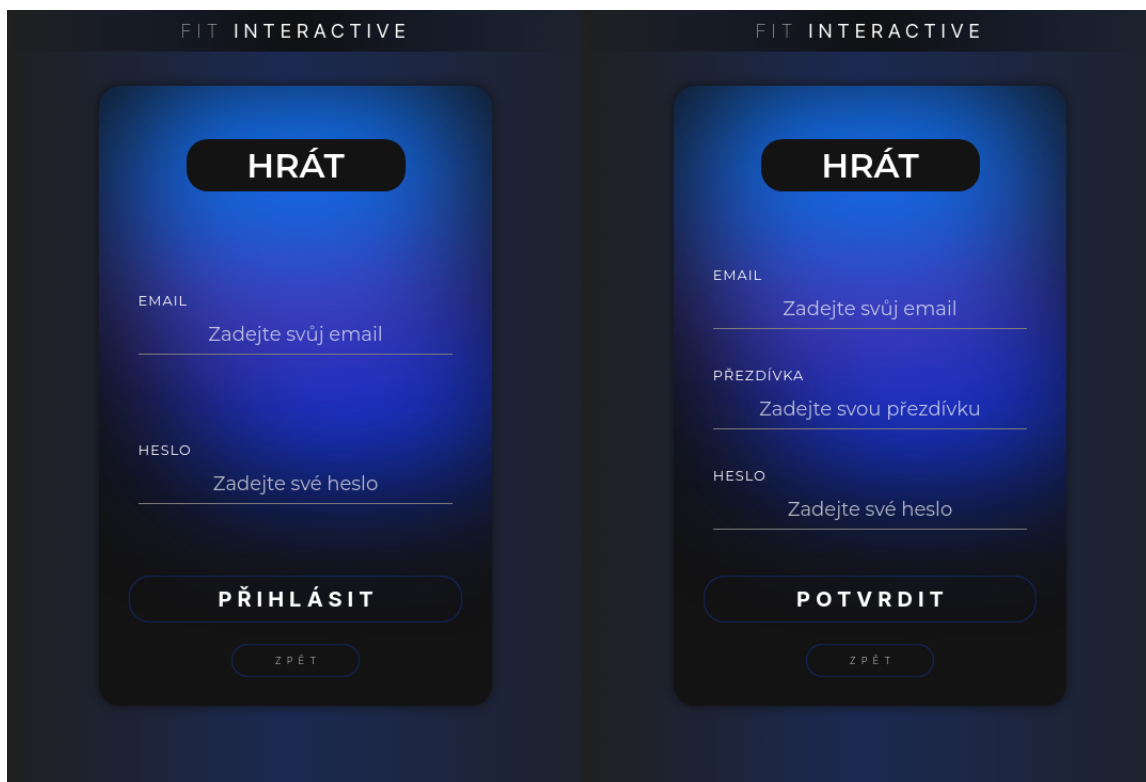
5.1.1 Registrace a přihlášení uživatele

Komponentou „Registrace a přihlašování uživatele“ je zajištěna autentizace uživatelů a umožněn přístup k funkcím spojeným s účastí na kvízech a jejich správou. Uživatelům je umožněno zobrazovat detaily kvízů, vytvářet je, a také sledovat svůj výkon, na základě již odehraných kvízů.

V rámci této komponenty je obsažen formulář pro přihlášení, který uživatelům umožňuje zadat své přihlašovací údaje, jako je jejich e-mail a heslo. Dále je poskytnuta možnost registrace nových uživatelů, kteří ještě nemají účet v systému, a to prostřednictvím formuláře pro vytvoření nového účtu.

Pro autentizaci uživatelů je využívána služba Firebase Authentication (viz sekce 2.3.2), která v tomto případě poskytuje robustní a bezpečnou autentizaci pomocí e-mailových adres a hesel. Tímto způsobem je zajištěna bezpečnost a spolehlivost autentizačního procesu.

Po úspěšném přihlášení nebo registraci je uživateli poskytnut přístup k dalším funkcím aplikace, jako je prohlížení dostupných kvízů, účast na kvízech a sledování výsledků.



Obrázek 5.1: Přihlašování v aplikaci

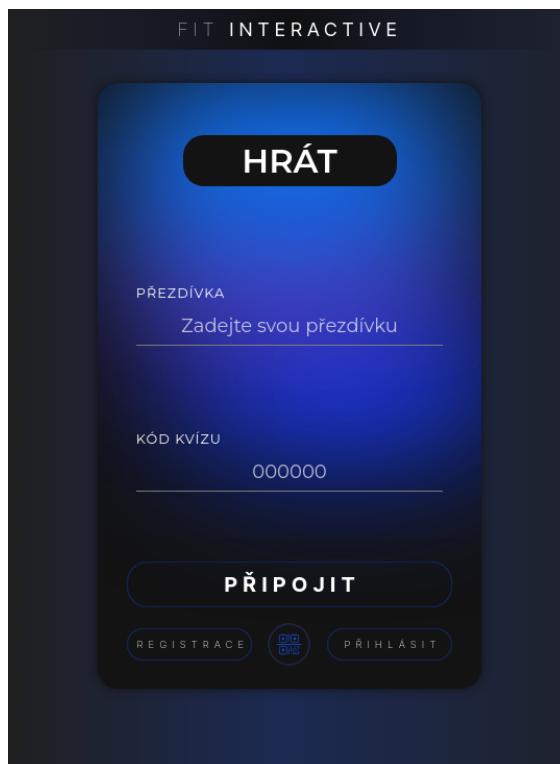
Obrázek 5.2: Registrace v aplikaci

5.1.2 Připojení do kvízu

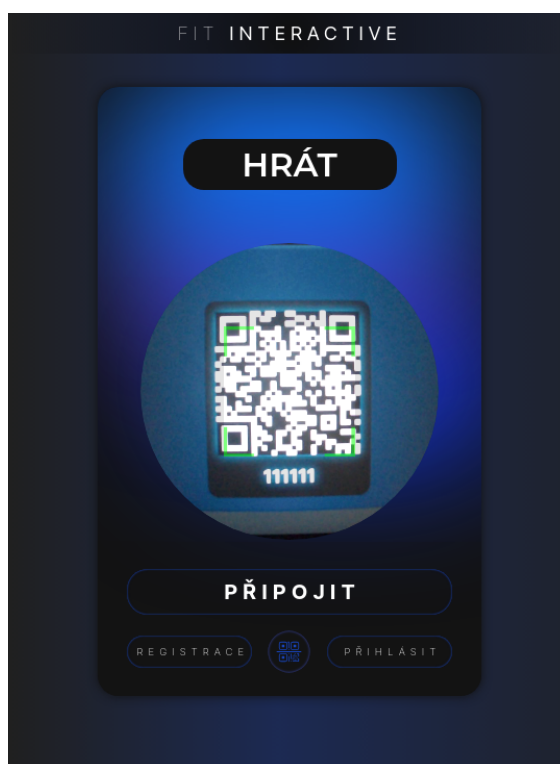
Bez této klíčové části aplikace by uživatelé měli problém s úspěšným připojením a aktivním zapojením se do průběhu jakéhokoli kvízu. Proto byl implementován intuitivní a snadno použitelný mechanismus, který umožňuje uživatelům rychle a bezproblémově se přihlásit k účasti na dostupných kvízech. Pro připojení do kvízu byla znovupoužita komponenta pro přihlašování a registraci uživatele. Tato komponenta poskytuje uživatelům jednoduchý a intuitivní způsob, jak se přihlásit k účasti na dostupných kvízech.

Pokud však uživatel nebude chtít ručně zadávat celý číselný kód kvízu, má také možnost použít funkci skenování QR kódu. Tento proces je rychlý a pohodlný. Stačí uživateli pouze namířit fotoaparát svého zařízení na vygenerovaný QR kód, který poskytne tvůrce kvízu. Po naskenování QR kódu je uživatel automaticky přesměrován přímo do daného kvízu, což zjednodušuje a urychluje připojení do kvízu.

Tímto způsobem je uživatelům poskytnuta flexibilita a volba, jakým způsobem se chtějí připojit k účasti na kvízu, ať už pomocí ručního zadání kódu nebo pomocí skenování QR kódu.



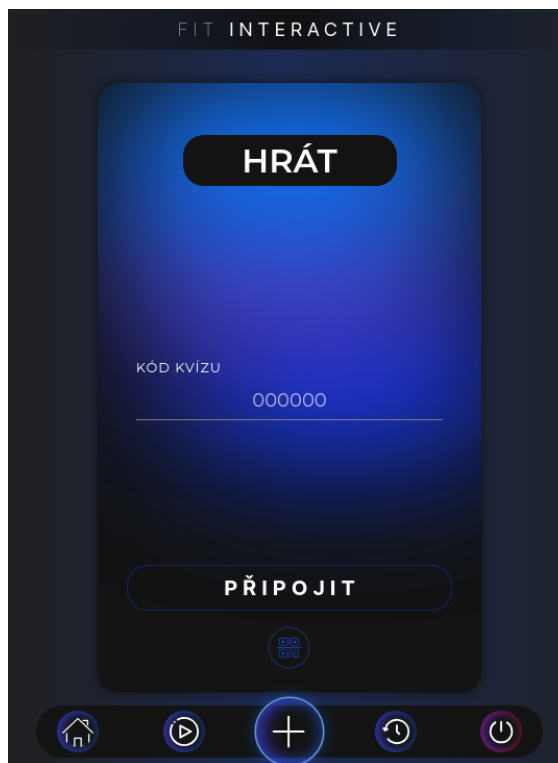
Obrázek 5.3: Připojení do kvízu pomocí číselného kódu



Obrázek 5.4: Připojení do kvízu pomocí QR kódu

V použité komponentě skeneru¹ QR kódů je zabudována indikace správnosti přečtení QR kódu. Jedná se o čtyři pravoúhlé rohy, svítící červeně v případě, že je QR kód nesprávně přečten. Pokud se QR kód podaří dekódovat, rohy se rozsvítí zeleně.

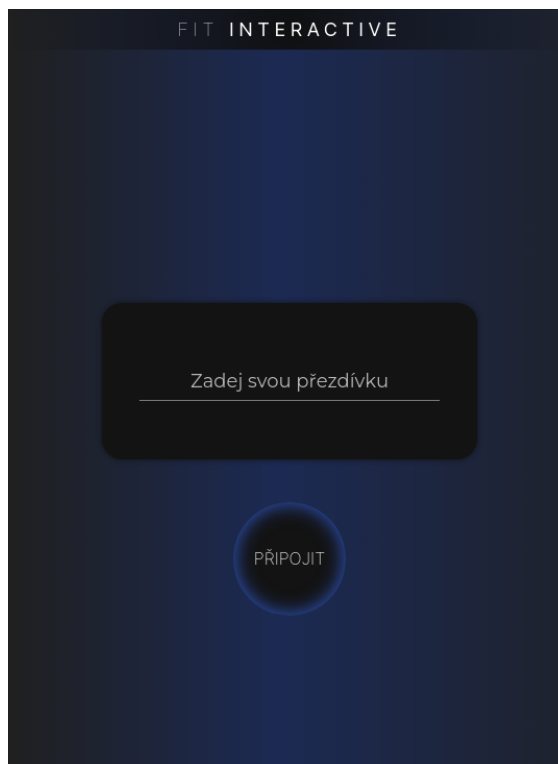
Pokud je uživatel přihlášený, je samozřejmostí, že se mu zobrazí pouze formulář, který již neobsahuje tlačítko pro registraci či přihlášení. Je zobrazeno pouze pole pro vložení číselného kódu kvízu s případnou možností načíst QR kód.



Obrázek 5.5: Připojení do kvízu - přihlášený uživatel

V případě, že se uživatel potřebuje pouze rychlým způsobem dostat do kvízu (použil QR kód, vygenerovaný vlastníkem kvízu), není nutno mu zobrazovat celý formulář s nutností zadání veškerých povinných údajů a tedy byla vytvořena ještě jedna verze pro připojení uživatele do kvízu. Tato verze přihlašovacího formuláře zahrnuje pouze potřebu zadat uživatelskou přezdívku a neobsahuje žádné jiná rozptylující prvky. Po kliku na tlačítko PŘIPOJIT je uživatel přemístěn do „čekací místnosti“ kvízu do doby, než bude zahájen.

¹<https://github.com/yudielcurbelo/react-qr-scanner>



Obrázek 5.6: Alternativní formulář pro připojení při využití QR kódu

5.1.3 Nástěnka

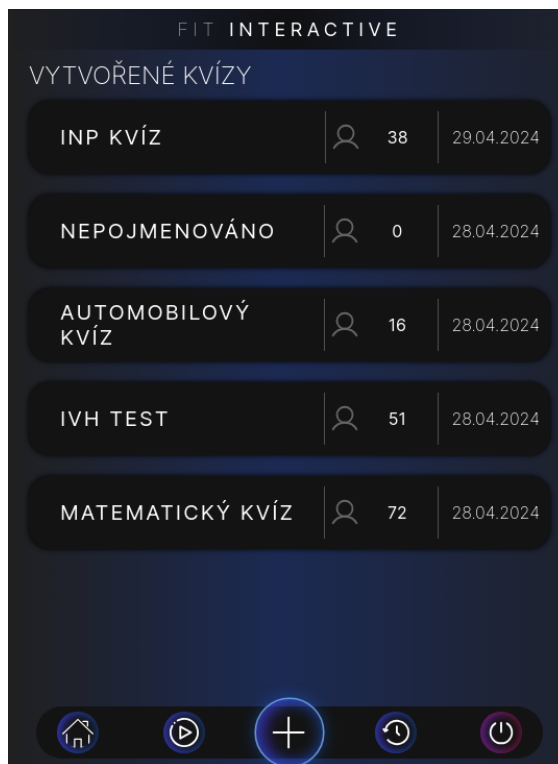
Nástěnka, také známá jako Dashboard, představuje důležitou část aplikace, která má za úkol poskytnout uživateli přehled o všech jeho vytvořených kvízech. Tento prostor slouží jako centrální místo, kde uživatel může snadno a rychle nalézt informace o svých kvízech a spravovat je podle svých potřeb.

Na nástěnce jsou zobrazovány klíčové informace o každém kvízu, jako je název, datum vytvoření a počet zúčastněných uživatelů. Uživatelé tak mají možnost rychle identifikovat jednotlivé kvízy a získat přehled o jejich stavu a obsahu.

Samotné položky seznamu (kvízy), tak obsahují dostatek informací, aby si o nich tvůrce kvízu dokázal vytvořit představu, ale pouze tolik informací, aby Dashboard pro uživatele již nebyl nepřehledný nebo matoucí.

Nástěnka také poskytuje uživatelům možnost různých akcí spojených s každým kvízem, jako je jejich úprava, spuštění, zobrazení detailů nebo dokonce jejich smazání. Tímto způsobem má uživatel plnou kontrolu nad svými kvízy přímo z nástěnky, což přispívá k pohodlné a efektivní správě jejich aktivit.

Vytvořené kvízy se uživateli zobrazí v seznamu pod sebou sestupně podle data vytvoření kvízu nikoli ve formě dlaždic vedle sebe z důvodu, že by se mohlo jednat o matoucí řazení pro uživatele, kteří nemají s podobným rozložením zkušenosti.



Obrázek 5.7: Nástěnka s vytvořenými kvízy

5.1.4 Navigace aplikace

Navigace v aplikaci je považována za velmi důležitou komponentu, bez které by se aplikace neobešla. Obsahuje celkem pět prvků v podobě ikonek, z nichž každá má svou specifickou funkci. Prvním prvkem je ikonka domečku, která uživatele přeměrovává na hlavní obrazovku, nazývanou také nástěnka (viz sekce 5.1.3). Dalším prvkem je ikonka spuštění, která slouží k připojení uživatele ke kvízu. Pokud uživatel chce vytvořit nový kvíz, může využít ikonku třetí ve tvaru matematické operace plus. Pro zobrazení historie odehraných kvízů je využívána ikonka čtvrtá, která znázorňuje čas se zpětnou šipkou. Nakonec je zde ikonka, která umožňuje uživatelům odhlášení z aplikace. Jako jediná z ikonek je odlišena jinou barvou. Jedná se o barvu vínovou, ostatní ikonky mají barvu modrou.



Obrázek 5.8: Navigace aplikace

Tento jednoduchý a intuitivní systém navigace usnadňuje pohyb uživatelů po aplikaci a přispívá k lepšímu uživatelskému zážitku. Díky němu jsou uživatelé schopni snadno a rychle se dostat ke všem důležitým funkcím aplikace a využívat je podle svých potřeb.

5.1.5 Tvorba kvízu

Tvorba kvízu je pro přednášející jedním z nejvíce kritických a důležitých míst aplikace. Bylo tedy nutno dbát na pohodlí a přehlednost při přidávání jednotlivých interaktivních bloků či snímků při vytváření kvízů.



Obrázek 5.9: Tvorba kvízu

Rozhraní obsahuje elementy, které uživatelé mohou poznat z jiných programů, ať už se jedná o programy, zaměřené na úpravu dokumentů, vytváření prezentací nebo jiné programy. Nebylo tedy vhodné vytvářet pro ikonky úlohy jiné než ty, které uživatelé dobře znají a jsou na ně zvyklí.

Nahoře uživatelského rozhraní se nachází miniatury snímků, které uživatel v daném kvízu již vytvořil. Díky tomu uživatel může provádět velmi rychlé úpravy požadovaných snímků a jejich interaktivních bloků.

Pod miniaturami snímků se nachází sekce interaktivních bloků, které uživatelům umožňují přidávat obsah do svých kvízů jednoduše a efektivně. Uživatel si může vybrat ze tří základních bloků: textové pole, obrázek a kus kódu. Textové pole umožňuje vložit textový obsah, obrázek umožňuje přidání grafických prvků a ikonka pro kód integruje markdown editor². Tento editor umožňuje formátování textu, přidávání obrázků z externích zdrojů, zvýrazňování syntaxe kódu či vykreslování diagramů. Tyto bloky poskytují uživatelům flexibilitu a možnost snadno vytvářet interaktivní a zajímavé vzdělávací materiály přímo v rámci aplikace.

V rámci uživatelské přívětivosti byly na pravou stranu přidány tlačítka pro přiblížení / oddálení aktuálního snímku.

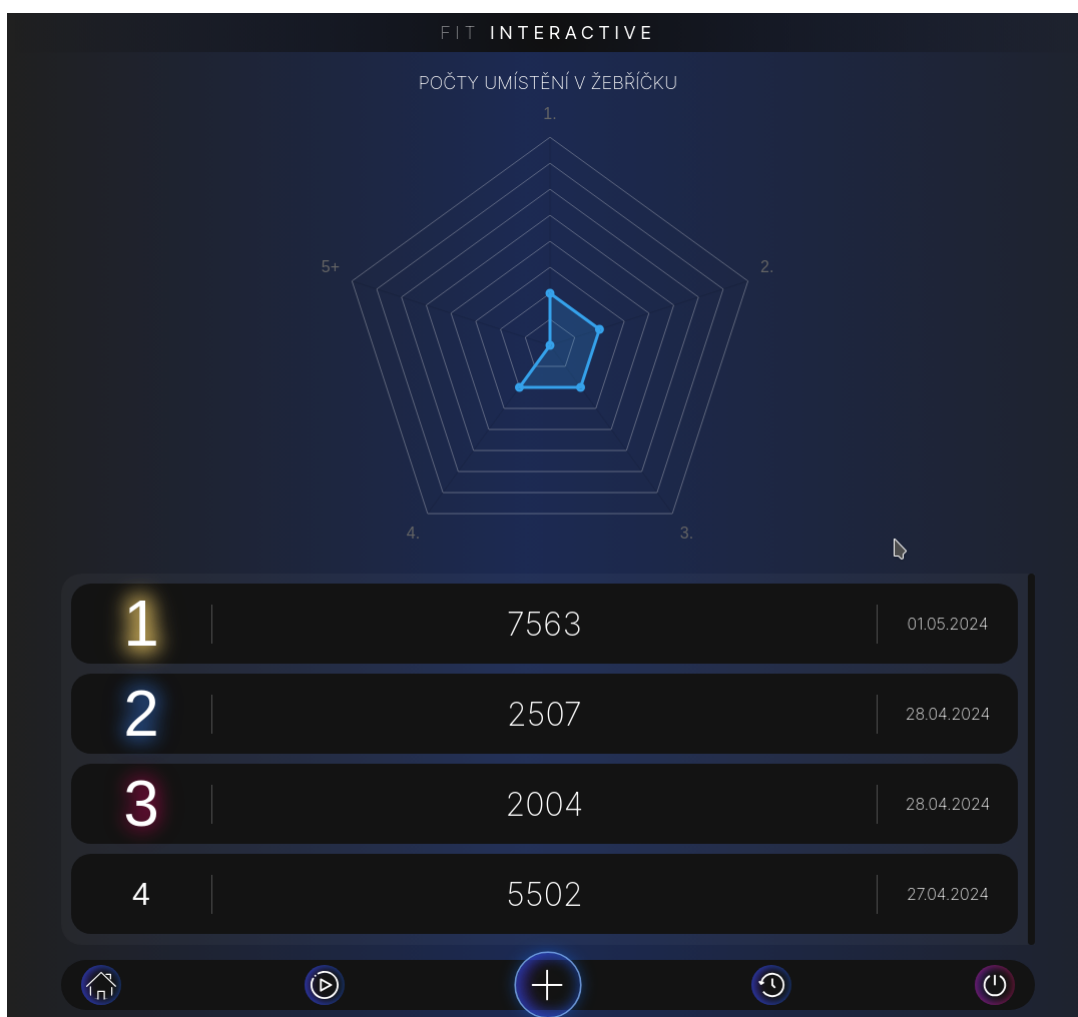
²<https://uiwjs.github.io/react-md-editor/>

Pod sekcí s interaktivními bloky se nachází samotný snímek, na jehož vrcholu uživatel zadá svou otázku. Jednotlivé bloky se skládají pod ni. Po kliku na jakýkoliv blok uživatel může nastavit, zda se jedná o součást otázky a nikoliv odpověď, či je odpověď správná apod.

Po kliku na tlačítko VYTVOŘIT je zaslán asynchronní požadavek do Firestore databáze a pokud nenastala nějaká chyba, na kterou je uživatel upozorněn, je kvíz vytvořen a uživatel přeměrován zpět na nástěnku s ostatními vytvořenými kvízy.

5.1.6 Historie odehraných kvízů

V pořadí čtvrtá ikonka navigace uživatele přeměruje na stránku, kde jsou v seznamu zobrazeny odehrané kvízy. U každého kvízu lze zjistit kolikrát uživatel skončil, kolik získal bodů v průběhu kvízu a jako poslední, kdy byl kvíz odehrán. Pokud uživatel obsadil jednu z prvních třech pozic, je pozice obarvená a velikostně se odlišuje například od pozice čtyři a výše.



Obrázek 5.10: Historie odehraných kvízů

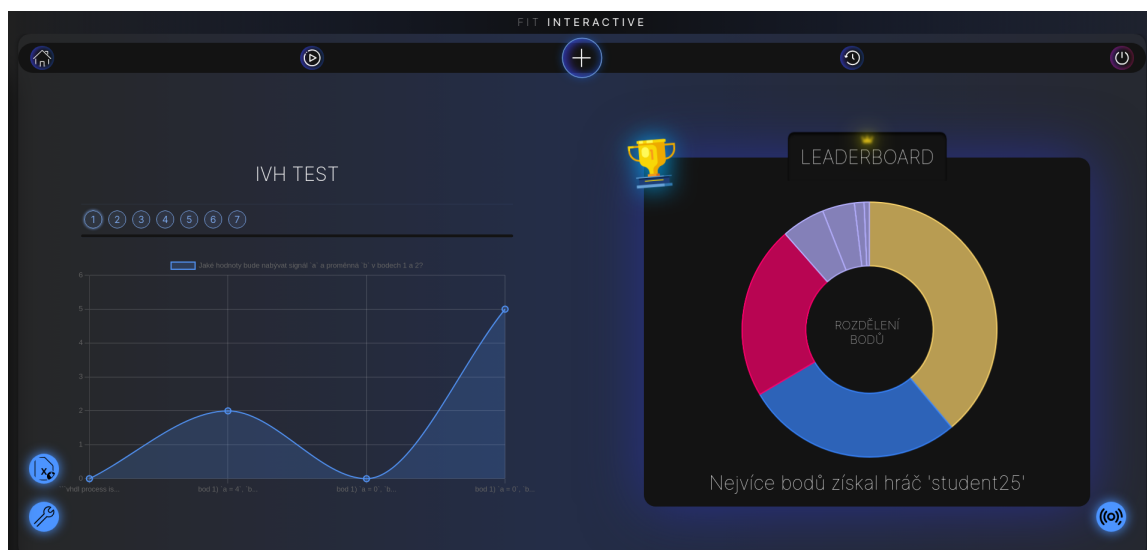
Nad seznamem kvízů se nachází radarový graf, který zobrazuje, kolikrát se uživatel umístil na jednotlivých pozicích v rámci historie odehraných kvízů. Tento graf poskytuje rychlý přehled o uživatelových umístěních a výkonech. Je jednoduchý a snadno čitelný.

Graf je součástí uživatelsky přívětivého rozhraní, které usnadňuje navigaci v aplikaci. Poskytuje uživatelům intuitivní způsob sledování svého pokroku v čase. Umožňuje uživatelům rychle identifikovat vzory a trendy ve svých výsledcích. Je to užitečný nástroj pro analýzu a zlepšování výkonu.

Pro implementaci všech grafů napříč aplikací byla využita knihovna Chart.js³. Tato knihovna poskytuje širokou škálu možností pro vytváření různých typů grafů, včetně radarových grafů⁴, které byly použity pro zobrazení historie odehraných kvízů. Pro každý graf byla definována data a nastavení, která odpovídala požadavkům uživatelského rozhraní. Chart.js umožnilo jednoduše integrovat do této aplikace grafy a poskytla jistou flexibilitu při jejich úpravách a nastaveních podle mých potřeb. Díky této knihovně bylo možno rychle a efektivně implementovat grafické prvky, které zlepšily uživatelský zážitek a usnadnily interpretaci dat uživatelům. Použití chart.js bylo klíčové pro úspěšnou realizaci grafických prvků v aplikaci a přispělo k jejímu profesionálnímu vzhledu a funkčnosti.

5.1.7 Detail kvízu

Na stránce detailu kvízu je k dispozici přednášejícímu přehled o průběhu a výsledcích kvízu, což je zásadní pro posouzení porozumění studentů a úspěšností výukového procesu. Jednou z klíčových funkcí je graf typu line⁵, umístěný na levé straně stránky, který zobrazuje, jaké odpovědi byly studenty vybrány nejčastěji. Tento graf poskytuje přednášejícímu možnost analyzovat, které možnosti byly preferovány, jak se vyvíjela interakce s otázkami a zda existují nějaké vzory nebo tendence ve výběru odpovědí.



Obrázek 5.11: Stránka podrobností kvízu

Na pravé straně stránky je umístěn graf typu doughnut⁶, který zobrazuje jednotlivé účastníky kvízu a počet bodů, které každý získal. Tento graf poskytuje přednášejícímu snadnou vizualizaci výsledků a umožňuje mu rychle porovnat úspěšnost jednotlivých studentů. Identifikace studentů s výraznými úspěchy nebo potížemi může pomoci přizpůsobit

³<https://www.chartjs.org/docs/latest/>

⁴<https://www.chartjs.org/docs/latest/samples/other-charts/radar.html>

⁵<https://www.chartjs.org/docs/latest/charts/line.html>

⁶<https://www.chartjs.org/docs/latest/charts/doughnut.html>

výuku a poskytnout individuální podporu tam, kde je to nezbytné. Celkově tyto grafy vytvářejí důležitý nástroj pro zhodnocení výkonu studentů a zlepšení výukového procesu. Pod grafy je umístěna tabulka účastníků, kde lze vyčíst, zda byl účastník ověřen pomocí GPS či nikoliv a kolik získal bodů.

Na tuto stránku byla přidána taktéž tři respektive čtyři funkční tlačítka. Podle jejich významu byla rozložena na levou a pravou stranu uživatelského rozhraní. Na levé straně se nachází skupina tlačítek, která mají co dočinění s manipulací dat kvízu. Tedy nalezneme zde tlačítko, vypadající jako klíč, pro možnost úprav kvízu a pod ním dále tlačítko pro export dat do souboru formátu csv [13], díky kterému lze snadno ukládat tabulková data.

Na pravé straně se poté nachází pouze vždy jedno funkční tlačítko. Ze začátku je tlačítko zbarveno modrou barvou a slouží pro nastavení kvízu do tzv. veřejného módu, který zajistí, že je kvíz nakopírován do Realtime Database. Po kliku na toto tlačítko je jeho barva změněna na červenou a symbol je změněn na čtverec, který znázorňuje, že po kliku na něj bude kvíz ukončen. Pod tímto čtvercem je taktéž zobrazen číselný kód kvízu, na jehož kliknutí je tvůrce kvízu rovnou přemístěn do čekací místnosti a nemusí nikam kód ručně zadávat.

5.1.8 Průběh hraní

V momentě, kdy tvůrce kvízu nastaví kvíz jako veřejný. Jsou potřebná data pro hraní kvízu z databáze Firestore nakopírována do Realtime Database (viz sekce 2.3.1). Odezva z této databáze je mnohem rychlejší a pro uživatele se solidním internetovým připojením téměř instantní. Tento proces zajišťuje, že všichni účastníci kvízu mají aktuální a synchronizované informace během hraní. Díky Realtime Database mohou uživatelé vidět okamžité změny v otázkách, odpovědích a výsledcích kvízu, což zlepšuje celkovou uživatelskou zkušenost a umožňuje plynulé hraní.



Obrázek 5.12: Čekací místnost

Jakmile se uživatel připojí pomocí číselného nebo QR kódu do kvízu, je před jeho zahájením umístěn společně s dalšími účastníky do čekací místnosti, tzv. lobby. V lobby uživatelé čekají, až tvůrce kvízu samotný kvíz spustí. Všichni uživatelé vidí, kolik hráčů se momentálně v lobby nachází. Tvůrce kvízu má poté zobrazeny navíc dvě tlačítka. Jedno pro zahájení kvízu a druhé, které zobrazí číselný a QR kód pro připojení. Bylo nutno nastavit bílé pozadí za QR kódem, v momentě, kdy bylo nastaveno tmavé pozadí, QR kód šel špatně skenovat z úhlu, což by na přednáškách nebylo ideální.



Obrázek 5.13: Informace pro připojení ke kvízu

Po zahájení kvízu je proces již přímočarý. Uživatelé si přečtou otázku, která je vždy nahoře aplikace a z daných interaktivních bloků vybírají správnou odpověď. Mohou si také po stisku tlačítka ověřit svou lokaci, pokud je o to přednášející požádá. Vlastník kvízu toho vidí o něco více. K dispozici má tlačítka pro manuální vyhodnocování a přepínání otázek nebo tlačítko, které po nastavení časového intervalu automaticky vyhodnotí otázku. Toto tlačítko se nachází v pravém horním rohu. Přepínání na další otázku je vždy pouze manuální, jelikož přednášející danou odpověď může chtít vysvětlit a nemá smysl zde nastavovat nějaký časový limit.

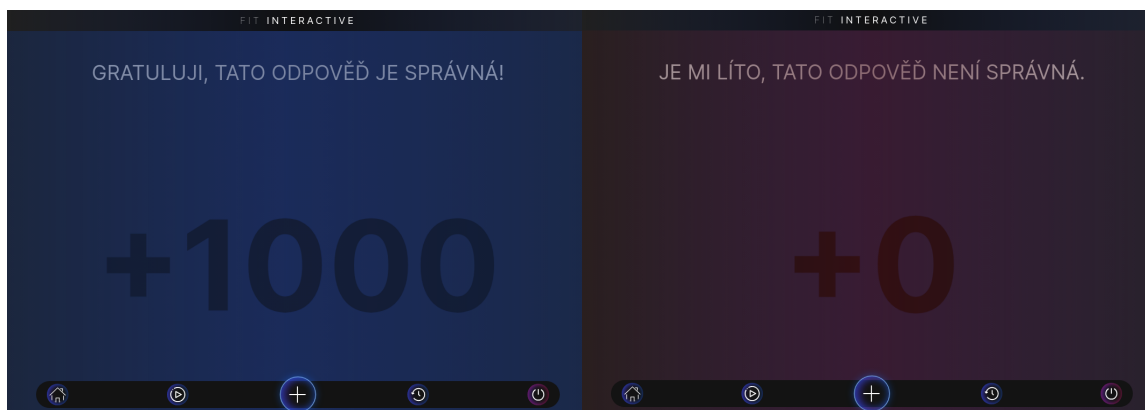
Na základě zpětné vazby při testování aplikace byl implementován indikátor počtu odeslaných odpovědí uživateli pro přednášejícího. Toto číslo přednášející vidí v levém horním rohu. Má tak přehled o tom, zda již může vyhodnotit otázku nebo by měl dát účastníkům více času.

Po vyhodnocení otázky se přednášejícímu správná odpověď označí zeleně a účastníkům tak dále může vysvětlovat, z jakého důvodu je odpověď správná a proč jsou ostatní odpovědi nesprávné.



Obrázek 5.14: Hra kvízu - pohled přednášejícího

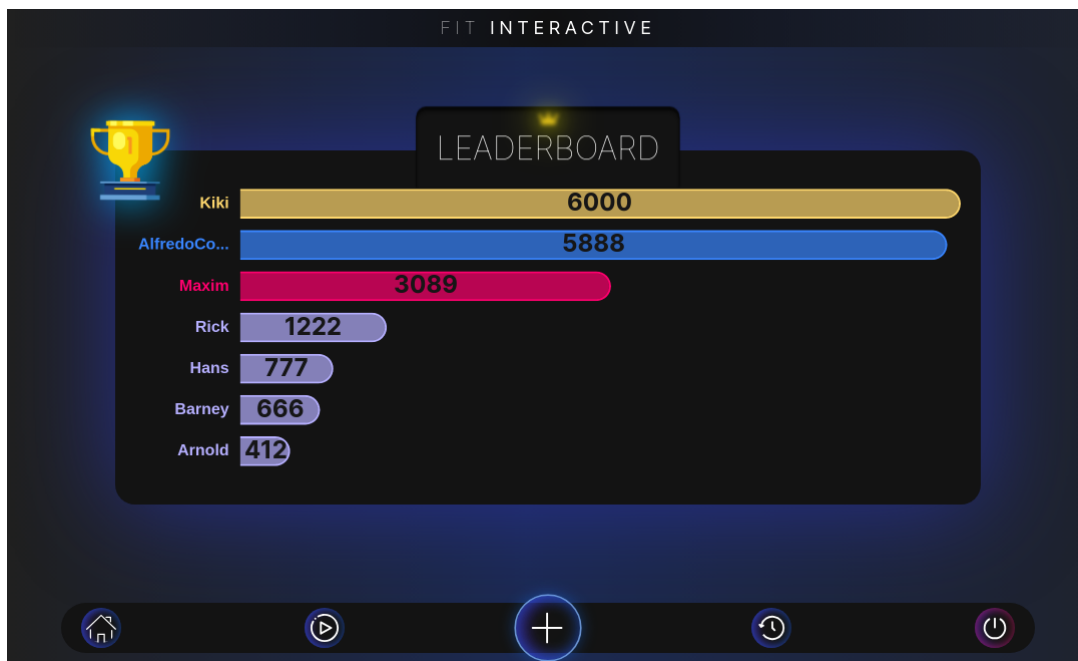
Ostatní uživatelé však vidí něco naprosto odlišného. Jelikož je potřeba, aby bylo hraní kvízu pro uživatele atraktivní, rozhodl jsem se implementovat přidělování bodů na základě času odeslání odpovědi a správnosti odpovědi (viz sekce 5.2.1). Uživatelé tak vidí, zda měli odpověď správně či nikoli a počet získaných bodů je jim zobrazen hned, jakmile přednášející vyhodnotí otázku. Tímto je zajištěna do jisté míry okamžitá zpětná vazba.



Obrázek 5.15: Správná odpověď studenta Obrázek 5.16: Nesprávná odpověď studenta

Po poslední otázce má možnost vyučující ukončit kvíz a zobrazit tak účastníkům kvízu jejich výsledky. Bodové hodnocení prvních sedmi uživatelů, kteří získali nejvíce bodů jsou vykresleny do grafu typu bar⁷. První tři obsazená místa jsou barevně odlišena od ostatních.

⁷<https://www.chartjs.org/docs/latest/charts/bar.html>



Obrázek 5.17: Žebříček uživatelů s nejvíce získanými body

Pokud tedy přednášející například před začátkem kvízu prohlásil, že uživatelům, kteří obsadí první tři místa, přidá jeden bonusový bod, může takto velmi rychle zjistit, komu bod dát.

5.2 Serverová část

Serverová část aplikace byla vytvořena s využitím ekosystému Firebase (viz sekce 2.3). Nejprve budou popsány cloudové funkce a jejich využití v aplikaci, poté bude probráno, na kterém místě bude aplikace k dispozici.

5.2.1 Cloudové funkce

Cloudové funkce bylo potřeba implementovat z důvodu větší bezpečnosti či automatizace databáze tak, aby se operace nemusely provádět přímo na zařízení uživatele. Níže je výčet implementovaných funkcí, včetně jejich rolí v aplikaci.

quizzCreated – Tato funkce je spuštěna, pokud je zaznamenána změna v databázi Firestore, konkrétně v části `/quizzes/quizzDocId`. Stará se o inicializaci nově vytvořeného kvízu. Pokud kvíz nemá nastavený název, automaticky je přiřazen název „Nepojmenováno“. Tímto způsobem je zajištěno, že každý nový kvíz má přidělené základní atributy a je připraven k použití.

quizzStarted – Při nastavení veřejného módu kvízu se tato funkce stará o inicializaci všech atributů a hodnot potřebných pro správný průběh kvízu. To zahrnuje generování číselného kódu pro připojení do kvízu, seřazení otázek a odpovědí podle jejich atributu `poradi`. Tyto data jsou tedy získávána z Firestore, zpracována a poté umístěna do Realtime Database.

quizzEnded – Po skončení kvízu tato funkce provádí několik důležitých úkolů. Nejprve přenáší data zpět z Realtime Database do Firestore, aby byla zachována konzistence a integrita dat. Vypočítá pozici každého uživatele v rámci kvízu na základě jeho výsledků. Poté přiřazuje uživatelům kvíz do seznamu odehraných kvízů společně s jejich body, pozicí a datem odehrání kvízu.

slideSwitched – Jak již název napovídá, funkce je spuštěna v momentě, kdy se změní snímek, tedy při přechodu na další otázku kvízu. Funkce nastaví pro novou otázku časový údaj **startedTimestamp**. Tato informace je klíčová pro výpočet získaných bodů pro daného uživatele.

evaluateQuestion – Funkce je spouštěna pokaždé, když při hraní kvízu proběhne vyhodnocování dané otázky přednášejícím (kliknutím na tlačítko „Vyhodnotit“ nebo automaticky). Funkce nejdříve aktualizuje atributy otázky, jako je **endedTimestamp**, který označuje čas ukončení otázky. Poté zkontroluje, kdy uživatel odeslal odpověď vzhledem k zahájení a ukončení otázky, s ohledem na časovou toleranci. Vzorec pro výpočet bodů uživatele je: **maxPoints** – (**userAnswer/endedTimestamp**) * **maxPoints**. Časová tolerance **userAnswer/endedTimestamp**, činí maximálně 10 % uběhlého času. Za tuto dobu tedy uživatel nedostane bodovou srážku.

Maximální počet bodů, které uživatel může za danou otázku získat je 1 000 a minimální 0. Záleží, zda a kdy odpověděl správně na otázku či odpověděl nesprávně a tedy celkově získal 0 bodů.

5.2.2 Produkční prostředí

Rozhodnutí o výběru produkčního prostředí pro hostování aplikace bylo důkladně promyšlené a zvážené. Přestože existuje mnoho alternativních platforem, jako jsou Vercel⁸, Google Cloud⁹ nebo Heroku¹⁰, rozhodl jsem se zůstat v rámci ekosystému Firebase.

Toto rozhodnutí bylo ovlivněno několika faktory. Zaprvé, integrace s ostatními službami a nástroji ve Firebase, které jsou již součástí vývojového procesu, představovala klíčovou výhodu. Do budoucna lze tedy aplikaci nadále vyvíjet jednoduše v rámci ekosystému, případně sledovat její vývoj z pohledu jedné platformy. Dále Firebase Hosting (viz sekce 2.3.6) nabízí snadné a rychlé nasazení webové aplikace v cloudu s minimálními nároky na konfiguraci a správu infrastruktury. Tímto způsobem bylo možné minimalizovat čas vynaložený na nastavení a údržbu.

Dalším důležitým faktorem byla spolehlivost a škálovatelnost Firebase Hosting, která poskytuje vysokou dostupnost a stabilitu pro provozování aplikace v produkčním prostředí. Tato platforma je navržena tak, aby zvládala zvyšující se zátěž a umožňovala rychlou reakci na různé požadavky uživatelů.

Celkově tedy výběr Firebase Hostingu jako produkčního prostředí pro hostování aplikace přináší řadu výhod a synergií s existujícími nástroji a službami ve Firebase ekosystému, což usnadňuje a zefektivňuje vývoj a provoz aplikace.

Webovou aplikaci této bakalářské práce lze nalézt na: <https://fit-interactive.web.app/>.

⁸<https://vercel.com/>

⁹<https://cloud.google.com/>

¹⁰<https://www.heroku.com/>

Kapitola 6

Zhodnocení systému a jeho další vývoj

Tato kapitola se zabývá vyhodnocením funkčnosti implementace z hlediska funkčních i nefunkčních požadavků. Bylo provedeno dotazníkové šetření na jedné z přednášek předmětu s názvem „Seminář VHDL“. Tento předmět je vyučován na Fakultě informačních technologií Vysokého učení technického v Brně.

6.1 Ověření funkčnosti implementace

Aplikace byla otestována na přednášce, uvedené výše. Ve kvízu se nacházely programové a textové otázky z předmětu „Seminář VHDL“ a studenti si tak aplikace mohli vyzkoušet, jak s aplikací pracovat. Vybírali vždy jednu správnou odpověď. Po ukončení kvízu byli studenti přesměrováni na dotazník, jenž se týkal samotné aplikace. Celkem dotazník vyplnilo jedenáct studentů či studentek a bylo položeno celkem sedm otázek.

Jaký je Váš názor na vizuální stránku uživatelského rozhraní aplikace?

Otázka si kladla za cíl zjistit, jaký je subjektivní dojem respondentů ohledně vizuální podoby uživatelského rozhraní aplikace.

Jde o to, jak jsou uživatelé spokojeni s designem, rozložením prvků na obrazovce, použitými barvami, typografií a dalšími estetickými aspekty. Tento názor může poskytnout užitečné informace o tom, jak aplikace působí na uživatele a zda je vizuálně přitažlivá a snadno použitelná.

Uživatelé z **90.9 %** zvolili, že aplikace vypadá skvěle nebo vypadá dobře, ale něco jim neseďí. Lze tedy usoudit, že po estetické stránce má aplikace u uživatelů úspěch.

Je pro Vás aplikace srozumitelná / přehledná?

Podle **90.9 %** respondentů je aplikace srozumitelná a přehledná, což naznačuje vysokou úroveň uživatelského komfortu a intuitivního designu. Tato pozitivní odezva signalizuje, že uživatelé mají snadný přístup k funkcím a informacím a zároveň si cení přehlednosti a strukturovanosti uživatelského rozhraní.

Použili jste k připojení do kvízu QR kód?

72.7 % dotazovaných studentů pro připojení do kvízu využilo QR kód, zbytek studentů využil číselný kód. Z tohoto je patrné, že si uživatelé vyberou možnost připojení, která je jim pohodlnější nebo v momentální situaci vhodnější, což bylo také jedním respondentem zmíněno i ve zpětné vazbě.

Líbila se Vám funkce bodování po každé otázce?

Reakce respondentů na funkci bodování po každé otázce byla velmi pozitivní. **100 %** účastníků vyjádřilo, že se jim tato funkce zalíbila. Tato odezva naznačuje, že možnost získávat body za každou správnou odpověď byla vnímána jako motivační prvek, který podněcuje k aktivnímu zapojení do kvízování a sledování výsledků. Bodování po každé otázce tak pravděpodobně přispívá k zábavnějšímu a interaktivnějšímu průběhu kvízů, což má pozitivní vliv na celkový uživatelský zážitek a zapojení do vzdělávacího procesu.

Myslíte si, že je vyhodnocování pořadí uživatelů podle získaných bodů vhodným interaktivním prvkem mezi studenty?

81.8 % respondentů souhlasilo s tím, že vyhodnocování pořadí uživatelů na základě získaných bodů je vhodným interaktivním prvkem mezi studenty. Tato odpověď naznačuje, že studenti hodnotí tuto funkci jako motivující a stimulující pro jejich aktivní účast v kvízování. Vyhodnocování pořadí jim umožňuje měřit svůj pokrok a porovnávat své výsledky s ostatními, což může podnítit zdravou soutěživost a snahu dosahovat lepších výsledků. Tento interaktivní prvek pravděpodobně přispívá k vytváření angažovaného a interaktivního prostředí ve vzdělávacím procesu.

Uvítali byste použití aplikace i na ostatních přednáškách?

Z počátku jsem se domníval, že toto téma bude mezi studenty velmi kontroverzní. Přeci jen se jedná o jistou formu testování, což studenti mnohdy nemají v oblibě. Nicméně data mluví jasně a zřetelně. Z dotazníku vyplývá, že **90.9 %** studentů by uvítalo použití aplikace i na ostatních přednáškách. Lze předpokládat, že použití aplikace je pro studenty přínosné a že vnímají její využití jako podporu interaktivity a zapojení ve výuce i v jiných předmětech. Tato pozitivní odezva naznačuje, že aplikace může být užitečným nástrojem pro podporu aktivního učení a interakce ve vzdělávacím prostředí.

Jak byste ohodnotili aplikaci? (1 - Výborná, 5 - Velmi špatná)

Celkem **72.8 %** respondentů hodnotí aplikaci známkou 1 nebo 2, což představuje **72.8 %** pozitivní ohlasů. Tento výsledek naznačuje, že většina uživatelů aplikaci hodnotí jako kvalitní a pozitivně vnímají její funkcionalitu a uživatelský zážitek.

6.2 Budoucí vývoj a rozšíření aplikace

V budoucím vývoji aplikace je potřeba zaměřit se na rozšíření funkcionalit, aby aplikace lépe vyhovovala potřebám uživatelů. Jedním z hlavních cílů je implementace dalších interaktivních prvků do procesu vytváření kvízů, aby bylo možné vytvářet ještě zajímavější a poutavější vzdělávací materiály. Tedy například výběr více správných odpovědí či slider pro výběr odpovědi.

Dalším rozšířením aplikace by mohlo být sdílení vytvořených kvízů mezi uživateli, což umožní širší využití aplikace nejen pro individuální vzdělávání, ale i pro spolupráci mezi uživateli a skupinami.

V rámci budoucího vývoje bude také klíčové zdokonalení algoritmů pro vyhodnocování odpovědí a bodování, aby bylo dosaženo ještě přesnějšího a spravedlivějšího hodnocení výkonu uživatelů v průběhu kvízování.

V současné době je možné vkládat do kvízu pouze obrázky z externích zdrojů, protože možnosti uložení souborů jsou v rámci Firebase Storage omezené. Vzhledem k bezplatnému používání Firebase byla tato možnost implementována jako primární způsob vkládání obrázků. Tato volba umožňuje udržet aplikaci pro uživatele funkční a dostupnou, aniž by byly překročeny limity bezplatného tarifu Firebase. V budoucnu však lze zvážit možnosti rozšíření a optimalizace uložení souborů v aplikaci, aby bylo možné vkládat i obrázky přímo ze zařízení uživatele.

Dále je v budoucím rozšíření aplikace zahrnout mechanismus, který po určitém čase bude z Realtime Database mazat již odehrané kvízy, aby nedošlo k přeplnění databáze. V aktuálním stavu lze pozorovat výsledky již odehraného kvízu kdykoliv. V současném stavu, kdy aplikace funguje na bázi bezplatného tarifu Firebase, toto není ideální skrze limity cloudových funkcí.

Dalším velmi důležitým rozšířením by mohlo být přihlašování pomocí VUT loginu pomocí SAML [10]. Studenti by tak nemuseli své účty vytvářet, ale pouze by je propojili s již existujícím účtem. Toto v momentální situaci na straně serveru opět není možné bez využití zpoplatněného tarifu Firebase.

Kapitola 7

Závěr

Hlavním cílem této bakalářské práce bylo vyvinout systém, který by umožňoval snadnou tvorbu, správu a sledování kvízů, a to s důrazem na uživatelskou přívětivost a flexibilitu. Bylo nutno nalézt efektivní a uživatelsky přívětivý způsob, jak spravovat a hrát kvízy.

Byly prostudovány již existující podobné platformy a analyzovány jejich přednosti či nevýhody.

Během návrhu a implementace tohoto systému byly zohledněny potřeby uživatelů, přičemž byl kladen důraz na jednoduchost použití. Systém byl vyvinut formou progresivní webové aplikace, což usnadňuje přístup z mnoha zařízení a nepředstavuje to pro uživatele zásadní bariéru.

Systém byl navržen s ohledem na možnost dalšího rozšíření a vylepšení v budoucnosti. Jednou z klíčových vlastností je modularita, která umožňuje snadné přidávání nových funkcí a integraci s existujícími systémy. To zajišťuje, že systém může být přizpůsoben potřebám uživatelů v průběhu času a může se vyvíjet společně s rostoucími požadavky a technologickými možnostmi. Výsledný systém poskytuje uživatelům efektivní nástroj pro správu kvízů, který je snadno použitelný.

Dále bylo provedeno dotazníkové šetření, které odhalilo pozitivní stránky systému, ale také jeho nedostatky, které mohou být námětem na další rozšíření aplikace.

Na aplikaci bych rád dále pokračoval, chtěl bych vytvořit i samotnou nativní mobilní aplikaci s podporou více funkcí.

Literatura

- [1] *Minifikace* [online]. [cit. 2023-23-12]. Dostupné z: <https://tech-lib.eu/definition/minification.html>.
- [2] BERNSTEIN, P. A. a NEWCOMER, E. *Principles of transaction processing*. Morgan Kaufmann, 2009. ISBN 1558606238.
- [3] BIEHL, M. *API Architecture: The Big Picture for Building APIs (API University Series)*. API-University Press, 2015. ISBN 150867664X.
- [4] BUGL, D. *Learn React Hooks: Build and refactor modern React.js applications using Hooks*. Packt Publishing Ltd, 2019. ISBN 1838641440.
- [5] GOLUBEVA, D. Webpack to Vite. [online]. Květen 2022, [cit. 2023-18-12]. Dostupné z: <https://kahoot.com/tech-blog/webpack-to-vite/>.
- [6] GUPTA, S. a GUPTA, B. B. Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*. Springer. 2017, sv. 8, s. 512–530.
- [7] HELAN, P. *Vývoj uživatelského rozhraní webové aplikace pro správu projektů*. Zlín, 2010. Diplomová práce. Univerzita Tomáše Bati ve Zlíně.
- [8] HUME, D. *Progressive web apps*. Simon and Schuster, 2017. ISBN 9781617294587.
- [9] KALMANOVYCH, A. Is Vite Really Faster Than Webpack? [online]. Listopad 2022, [cit. 2023-12-12]. Dostupné z: <https://betterprogramming.pub/is-vite-really-faster-than-webpack-b414f6cc751c>.
- [10] NIE, F., XU, F. a QI, R. SAML-based single sign-on for legacy system. In: IEEE. *2012 IEEE International Conference on Automation and Logistics*. 2012, s. 470–473.
- [11] OPPLIGER, R. *SSL and TLS: Theory and Practice*. 3. vyd. Artech House, 2023. 23-82 s. ISBN 1685690157.
- [12] RESCORLA, E. a SCHIFFMAN, A. *The secure hypertext transfer protocol* [Internet Requests for Comments]. RFC 2660. Srpen 1999. Dostupné z: <https://www.rfc-editor.org/rfc/rfc2660>.
- [13] SHAFRANOVICH, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files* [Internet Requests for Comments]. RFC 4180. říjen 2005. Dostupné z: <https://www.rfc-editor.org/rfc/rfc4180>.

- [14] SINHA, D. Progressive Web Apps (PWAs): The future of mobile web apps. [online].
Únor 2020, [cit. 2023-07-12]. Dostupné z:
<https://www.techaheadcorp.com/blog/progressive-web-apps/>.
- [15] WILSON, D., HASSAN, S.-U., ALJOHANI, N. R., VISVIZI, A. a NAWAZ, R.
Demonstrating and negotiating the adoption of web design technologies: Cascading
Style Sheets and the CSS Zen Garden. *Internet Histories*. Taylor & Francis. 2023,
sv. 7, č. 1, s. 27–46.
- [16] WINTER, M. *Scan me-everybody's guide to the magical world of qr codes*. Westsong
Publishing, 2011. ISBN 9780965900034.

Příloha A

Obsah přiloženého paměťového média

Příložené paměťové médium obsahuje veškeré zdrojové soubory aplikace. Tyto soubory lze nalézt v adresáři `aplikace`. V adresáři `doc` se nachází veškerá dokumentace ohledně instalace aplikace, dále se zde nachází uživatelská příručka a jako poslední samotná bakalářská práce v psané podobě. V podadresáři `zdrojove_soubory` se nachází zdrojové `LATEX` soubory pro bakalářskou práci. Soubor `README.md` obsahuje popis adresářové struktury přiloženého média.

```
DVD-R
├── doc/
│   ├── BP.pdf
│   ├── prirucka.pdf
│   ├── instalace.pdf
│   └── zdrojove_soubory/
├── aplikace/
└── README.md
```