



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

**Tvorba interaktivní webové aplikace s využitím
frameworku Qwik se zaměřením výuky
anglického jazyka na 2. stupni základní školy**

**Development of a interactive web application
using framework Qwik for teaching English
language at the lower secondary school level**

Bakalářská práce

Vypracoval: Jakub Moravec

Vedoucí práce: PaedDr. Petr Pexa, Ph.D.

České Budějovice 2025

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Jakub MORAVEC
Osobní číslo: P22409
Studijní program: B0114A300110 Oborové studium se zaměřením na vzdělávání na 2. stupni základní školy
Specializace: Anglický jazyk se zaměřením na vzdělávání na 2. stupni ZŠ
Téma práce: Informační technologie se zaměřením na vzdělávání na 2. stupni ZŠ
Zadávající katedra: Tvorba interaktivní webové aplikace s využitím frameworku Qwik se zaměřením výuky anglického jazyka na 2. stupni základní školy.
Katedra informatiky

Zásady pro vypracování

Cílem této bakalářské práce je navrhnout, sestavit a otestovat aplikaci vytvořenou pomocí frameworku Qwik se zaměřením na výuku anglického jazyka na 2. stupni základní školy. Práce bude rozdělena na teoretickou a praktickou část, přičemž teoretická část bude zaměřena na porovnávání frameworku Qwik s existujícími frameworky v oblasti efektivity, jednoduchosti. Praktická část bude zaměřena na tvorbu samotné aplikace a následné testování na žácích 2. stupně základní školy. Aplikace by měla poskytnout učitelům i žákům možnost efektivnější a interaktivnější výuky a rodičům přehled o tom, jaké má žák znalosti. Aplikace bude sloužit jako nástroj k procvičování látky, zadávání úkolů, známkování a případným testům.

Rozsah pracovní zprávy: 40
Rozsah grafických prací: CD ROM
Forma zpracování bakalářské práce: tištěná

Seznam doporučené literatury:

- Qwik. (n.d.). Qwik documentation. [online]. Dostupné z: <https://qwik.dev/docs/>
- GRAVES, Kathleen. DESIGNING LANGUAGE COURSES. National Geographic learning. ISBN 9780838479094.
- Stack overflow. The largest, most trusted online community for developers to learn, share their programming knowledge, and build their careers. [online]. (2008) [cit. 30. 3. 2023] Dostupné z: <https://stackoverflow.com/>
- Watson, J. (2023). Mastering Qwik(framework): The Essential Crash Course for Beginners to Qwik. Independently published. (ISBN 979-8391368540)
- BuilderIO. (n.d.). qwik: Instant-loading web apps, without effort. GitHub. [online]. Získáno z <https://github.com/BuilderIO/qwik>

Vedoucí bakalářské práce: PaedDr. Petr Pexa, Ph.D.
Katedra informatiky

Datum zadání bakalářské práce: 2. dubna 2024
Termín odevzdání bakalářské práce: 30. dubna 2025



doc. RNDr. Helena Koldová, Ph.D.
děkanka



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 2. dubna 2024

Prohlášení

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích dne 17. dubna 2025

Jakub Moravec

Podpis:

Abstrakt / Anotace

Cílem této bakalářské práce je navrhnout, sestavit a otestovat aplikaci vytvořenou pomocí frameworku Qwik se zaměřením na výuku anglického jazyka na 2. stupni základní školy. Práce bude rozdělena na teoretickou a praktickou část, přičemž teoretická část bude zaměřená na porovnávání frameworku Qwik s existujícími frameworky v oblasti efektivity a jednoduchosti. Praktická část bude zaměřená na tvorbu samotné aplikace a následné testování na žácích 2. stupně základní školy. Aplikace by měla poskytnout učitelům i žákům možnost efektivnější a interaktivnější výuky a rodičům přehled o tom, jaké má žák znalosti. Aplikace bude sloužit jako nástroj k procvičování látky, zadávání úkolů, známkování a případným testům.

Klíčová slova

Qwik, framework, porovnání frameworků, webová aplikace, 2. stupeň ZŠ, anglický jazyk

Abstract/Annotation

The goal of this bachelor's thesis is to design, construct, and test an application created using the Qwik framework, focusing on English language instruction for lower secondary school students. The thesis will be divided into theoretical and practical parts. The theoretical part will compare the Qwik framework with existing frameworks in terms of efficiency and simplicity. The practical part will focus on creating the application itself and subsequently testing it on lower secondary school students. The application should provide teachers and students with a more effective and interactive learning experience and give parents an overview of the student's knowledge. The application will serve as a tool for practicing material, assigning tasks, grading, and conducting potential tests.

Keywords

Qwik, framework, framework comparison, web application, lower secondary school, English language

Poděkování

Především bych chtěl poděkovat svému vedoucímu práce, panu PaedDr. Petru Pexovi, Ph.D., za jeho cenné rady, odborné vedení a trpělivost, které mi pomohly tuto práci úspěšně dokončit. Dále bych chtěl poděkovat rodině, přítelkyni a přátelům za jejich podporu během celého studia.

Obsah

1	Úvod	10
1.1	Východiska práce	10
1.2	Cíle práce	11
1.3	Metody práce	11
2	Teoretická část	12
2.1	Současný stav interaktivních vzdělávacích aplikací	12
2.1.1	Populární vzdělávací aplikace a jejich funkce	12
2.1.2	Trendy ve vývoji vzdělávacích aplikací	13
2.1.3	Výzvy při implementaci technologií ve vzdělávání	14
2.2	Framework Qwik a jeho charakteristiky	16
2.2.1	Základní charakteristiky frameworku Qwik	16
2.2.2	Porovnání Qwiku s jinými frameworky	16
2.2.3	Výhody a nevýhody frameworku Qwik	18
3	Praktická část	22
3.1	Motivace k vytvoření aplikace Sea Quest	22
3.2	Identifikace problémů ve výuce angličtiny	22
3.3	Přínos aplikace Sea Quest	22
3.4	Představení aplikace Sea Quest	24
3.4.1	Responzivita aplikace	24
3.4.2	Přihlášení a registrace	24
3.4.3	Systém tříd	30
3.4.4	Procvičování	40
3.4.5	Profil	53
3.4.6	Systém úrovní	55
3.4.7	Žebříček	60
3.4.8	Ocenění	63
3.5	Testování aplikace	67

4 Závěr	69
4.1 Naplnění cílů a přínos pro výuku	69
4.2 Technologie a inovativnost řešení	69
4.3 Shrnutí testování	70
4.4 Limit současné verze a potenciál dalšího rozvoje	71
4.5 Porovnání frameworků	72
4.6 Závěrečné zhodnocení	73
Seznam přečtené literatury a zdrojů	74
Seznam obrázků	79
Seznam zdrojových kódů	80
A Příloha	81

1 Úvod

V současné době se stále více využívají webové aplikace a služby. Mnoho firem a odvětví už tuto technologii nějakým způsobem využívá, ale ve školách ještě nedokáží využít její plný potenciál. Ve školství se ještě nepoužívá aplikace, která by měla leaderboard (žebříček nejlepších), odemykání nových úrovní, trofejí nebo skinů (jiný vzhled avatara, postav, prostředí). Pokud aplikace bude obsahovat herní prvky, které jsou atraktivní pro děti, tak budou mít motivaci stále pokračovat v učení, právě k tomu použiji framework Qwik, který dodá aplikaci moderní vzhled a interaktivní prvky.

1.1 Východiska práce

Ve školství se stále více objevuje potřeba inovativních a interaktivních přístupů k výuce, které by efektivně zapojily žáky a zlepšily jejich motivaci. Tradiční výukové metody se často nedají srovnávat s moderními technologiemi. Kdyby žák stál před otázkou: „Učit se z učebnice nebo učit se na počítači?“ Tak 90 % žáků si vybere učit se na počítači, protože je jim bližší. Příkladem toho je nedostatek gamifikovaných nástrojů v rámci školních aplikací, které by mohly přimět žáky k vyššímu zapojení a motivovat žáky prostřednictvím herních prvků. Kdyby žáci soutěžili o to, kdo má nejvyšší skóre, kdo udělá největší rekord nebo kdo má nejlepší skin, tak je to bude motivovat, aby pracovali jak ve škole, tak doma. Zavedení těchto nových technologií a přístupu ve školství představuje příležitost nejen pro zlepšení výuky, ale také pro lepší přizpůsobení vzdělávacích metod současným technologickým trendům a potřebám mladých lidí.

1.2 Cíle práce

Hlavním cílem mé bakalářské práce je navržení a sestrojení interaktivní webové aplikace pro výuku anglického jazyka na 2. stupni základních škol. Aplikace bude obsahovat prvky gamifikace, s cílem zvýšit motivaci žáků a jejich zapojení do výuky. Bude dostupná pro učitele i žáky. Po určité době používání uživatelé poskytnou zpětnou vazbu, aby se zhodnotila intuitivnost a celkový dojem. Dále bude provedeno srovnání frameworku Qwik s ostatními podobnými technologiemi od obtížnosti učení syntaxe a přehlednosti až po výkon.

1.3 Metody práce

Bakalářská práce začne definicí klíčových termínů a technologií, které budou použity při vývoji aplikace pro výuku angličtiny. V teoretické části práce bude framework Qwik porovnán s ostatními frameworky např. React, Vue.js, Next.js aj. Celý proces vývoje začne až v praktické části, od počátečního návrhu uživatelského prostředí, přes vytváření obsahu, až po propojení backendu pro správu uživatelských dat. Po dokončení aplikace a jejím následném otestování budou uživatelé, tj. učitelé a žáci vyzváni k poskytnutí zpětné vazby ohledně intuitivnosti, přehlednosti apod.

2 Teoretická část

V současné době se interaktivní vzdělávací technologie stávají čím dál větší částí moderního vzdělávání. Webové aplikace a služby nabízejí nové možnosti pro výuku, které mohou výrazně zvýšit zapojení a motivaci žáků k učení.

2.1 Současný stav interaktivních vzdělávacích aplikací

Interaktivní aplikace se v posledních letech staly důležitým nástrojem pro pedagogy, kteří hledají způsoby, jak oživit výuku a více zapojit žáky do hodiny. Tyto aplikace nabízí různorodé funkce a interaktivní prvky, které zvyšují zájem žáků o probíranou látku.

2.1.1 Populární vzdělávací aplikace a jejich funkce

V současné době existuje mnoho aplikací, které nabízejí různé přístupy k interaktivnímu vzdělávání. Jednou z nejoblíbenějších je Kahoot!, který umožňuje vytvářet různé kvízy a soutěže, do nichž se žáci přihlašují pomocí svých mobilních zařízení. Aplikace poskytuje okamžitou zpětnou vazbu a automaticky vyhodnocuje vítěze, což přináší do hodin prvky soutěživosti a zábavy. Tato forma gamifikace motivuje žáky k aktivnímu zapojení do výuky a může zlepšit jejich vztah k probíranému učivu [1] [2].

Například Wordwall je online platforma, která nabízí širokou škálu různých her jako jsou kvízy, křížovky a úkoly na spojování pojmů. Významnou výhodou této aplikace je české rozhraní a dostupnost velkého množství obsahu vytvořeného českými pedagogy. Podobně jako Quizlet představuje efektivní nástroj pro trénování slovní zásoby při výuce cizích jazyků. Quizlet pracuje s konceptem flashcards, které mají vždy dvě strany – dva jazyky a nabízí funkce jako rychlý překlad, ukázky výslovnosti a možnost importu vlastních, již hotových slovíček [2].

Další aplikací je například Nearpod, která obohacuje vzdělávací proces tím, že umožňuje učitelům prezentovat multimediální lekce, ke kterým mají žáci při-

stup prostřednictvím svých zařízení. Aplikace například dovoluje učiteli vytvořit kvíz, hlasování, kolaborativní nástěnku (Collaborate Board) nebo dovoluje žákům kreslit (Draw it). Tato aplikace tedy podporuje vytváření prezentací s interaktivními prvky, díky kterým učitel ihned vidí zpětnou vazbu od žáků [1] [3].

2.1.2 Trendy ve vývoji vzdělávacích aplikací

1. Gamifikace: Gamifikační přístup se čím dál více ukazuje jako efektivní nástroj pro zvýšení motivace žáků k procvičování dané látky. Příkladem úspěšné implementace gamifikace je Duolingo, které využívá systém úrovní a denních výzev k udržení motivace uživatelů při učení různých cizích jazyků. Tato platforma je postavena na principu postupného odemykání nových lekcí na základě získaných bodů, přičemž každý den je uživatel motivován k udržení tzv. „streaku“ - nepřetržitá řada dnů, během kterých se uživatel aktivně učil [4] [5]. Gamifikace využívá prvky jako jsou body, odznaky, žebříčky, výzvy a příběhy k tomu, aby proměnila jinak nudné učební aktivity na zajímavé a zábavné procvičování [5].
2. Personalizace: Současným trendem je také personalizace výukového obsahu podle individuálních potřeb a schopností žáků. Některé moderní vzdělávací aplikace využívají algoritmy, analýzu dat a umělou inteligenci k vytvoření individualizovaných učebních plánů, které se přizpůsobují tempu a stylu učení každého žáka. Tento přístup pomáhá žákům postupovat vlastním tempem a zaměřit se na oblasti, které pro ně představují největší výzvu. Aplikace jako Khan Academy nabízejí adaptivní učení, kde systém automaticky upravuje obtížnost úkolů a lekce podle výkonnosti žáka za pomoci umělé inteligence [6] [7].
3. Využití umělé inteligence: Umělá inteligence (AI) se stále více prosazuje ve vzdělávacích technologiích. AI může analyzovat výsledky žáků

a identifikovat mezery v jejich znalostech, což umožňuje poskytovat cílené výukové materiály pro překonání těchto nedostatků. Automatizace administrativních úkolů prostřednictvím AI také může uvolnit čas pedagogům, což jim umožní více se zaměřit na pedagogické interakce a rozvoj kritických kompetencí žáků [6] [8].

4. M-learning: V dnešní době už většina žáků má mobil nebo tablet a díky tomu se m-learning (forma online učení za využití mobilních technologií) stává stále významnějším trendem. Mobilní aplikace umožňují žákům přístup k výukovým materiálům kdykoli a kdekoli, což podporuje koncept celoživotního vzdělávání. Tyto aplikace často využívají mikroučení - rozdělení obsahu na malé, snadno zpracovatelné části, které lze studovat v krátkých časových intervalech, například během cesty do školy. Výzkumy ukazují, že žáci, kteří používají mobilní vzdělávací aplikace, vykazují lepší výsledky v oblasti slovní zásoby a plynulosti projevu [9] [10].

2.1.3 Výzvy při implementaci technologií ve vzdělávání

1. Integrace technologií: Efektivní integrace technologií do vzdělávacího procesu vyžaduje systematický přístup a překonání řady překážek. Nestačí pouze vybavit školy novými zařízeními, protože je na to v tuto chvíli rozpočet, ale je také třeba zajistit, aby tyto technologie byly smysluplně začleněny do výukových metod a kurikula. Tzv. „problém dvojí inovace“ znamená, že učitelé musí nejprve zvládnout samotnou technologii, a teprve poté mohou efektivně integrovat tuto technologii do výuky, což představuje významnou časovou investici [11]. Pro úspěšnou integraci je nezbytné, aby školy měly jasnou vizi a strategii, jak budou technologie používat k podpoře vzdělávacích cílů [12].
2. Digitální gramotnost učitelů: Jednou z hlavních výzev při zavádění nových technologií ve vzdělávání je úroveň digitální gramotnosti pedagogů.

Mnozí učitelé se cítí nejistí při používání nových technologií a postrádají technické a pedagogické dovednosti potřebné k jejich efektivnímu začlenění do své výuky. Programy celoživotního vzdělávání jsou nezbytné pro vybavení učitelů potřebnými dovednostmi [8] [11]. Tyto programy by měly řešit nejen technické používání nástrojů, ale také pedagogické strategie pro integraci těchto technologií do kurikula, podporu aktivního učení a zapojení žáků [13].

3. Bezpečnost a ochrana osobních údajů: S rostoucím využíváním digitálních platforem ve vzdělávání se stává kritickou otázkou ochrana osobních údajů žáků a učitelů. Incidentsky narušení bezpečnosti dat a kybernetické útoky mohou znehodnotit důvěryhodnost vzdělávacích technologií a vystavit citlivé informace riziku [13] [14]. Školy musí implementovat odpovídající bezpečnostní opatření, jako jsou k12 webové filtry, které mohou pomoci omezit přístup k rozptylujícímu obsahu a umožnit pedagogům udržet soustředěné učební prostředí [13].
4. Měření efektivity: Hodnocení skutečného dopadu vzdělávacích technologií na výsledky žáků představuje další výzvu. Je těžké úplně izolovat vliv technologií od ostatních faktorů, které ovlivňují učení. Efektivní monitorovací a evaluační systémy jsou nezbytné pro posouzení přínosu technologií a identifikaci oblastí pro zlepšení [15]. Sběr a analýza dat o používání technologií a výsledcích žáků mohou poskytnout cenné informace pro optimalizaci [12].
5. Přístupnost: Zajištění rovnocenného přístupu k technologiím pro všechny žáky je pro školy zásadní výzvou. Tzv. „digitální propast“ - nerovnost v přístupu k technologiím a internetu - může prohloubit sociální nerovnosti, zejména v kontextech socioekonomické zranitelnosti [8] [13]. Tato nerovnost vytváří překážky pro dokončení úkolů, provádění výzkumu a přístup ke vzdělávacím materiálům online. V důsledku toho se žáci s omezeným přístupem nemusí plně zapojit do digitálních vzdělávacích

zkušeností, což ovlivňuje jejich akademický výkon [13]. Řešení této výzvy vyžaduje investice do technologické infrastruktury a programů digitálního začleňování žáků ze všech socioekonomických vrstev [14].

2.2 Framework Qwik a jeho charakteristiky

2.2.1 Základní charakteristiky frameworku Qwik

Qwik je moderní JavaScriptový framework, který se zaměřuje na poskytování okamžitého načítání webových aplikací bez ohledu na jejich velikost nebo složitost. Byl vyvinut týmem kolem Builder.io, v němž působí i autor původního Angularu, Miško Hevery [16] [17]. Qwik dosahuje vysokého výkonu tím, že minimalizuje množství JavaScriptu, které je potřeba stáhnout při prvním načtení stránky [19].

Klíčovou inovací frameworku Qwik je koncept „resumability“ (obnovitelnosti), který eliminuje potřebu hydratace. Hydratace je proces, při kterém webová aplikace po načtení na klientovi musí zpracovat velké množství JavaScriptu, aby se stala interaktivní. Qwik tento krok přeskakuje, což umožňuje aplikacím být okamžitě interaktivní s minimálním množstvím počátečního JavaScriptu - přibližně 1 kB oproti desítkám až stovkám kB u jiných frameworků. To vede k výrazně plynulejšímu pocitu z aplikace, zejména na mobilních zařízeních nebo při pomalejším internetovém připojení [19].

Qwik je navržen jako „server-agnostický“, což znamená, že může být nasazen na jakémkoliv serveru nebo statickém hostingu. Je optimalizován pro okamžité obnovení stavu aplikace přímo v prohlížeči uživatele a využívá tzv. „progressive loading“ (lazy-loading) - načítá pouze ten JavaScript, který je aktuálně potřeba [19].

2.2.2 Porovnání Qwiku s jinými frameworky

Výkon a rychlost načítání

Při porovnání s ostatními populárními frameworky jako jsou React, Vue.js

nebo Next.js, Qwik vyniká v několika klíčových aspektech. Podle dostupných metrik má Qwik nejmenší počáteční velikost balíčku - pouhý 1 kB oproti 42 kB u Reactu, 33 kB u Vue.js a 75 kB u Next.js. Také v měřeních času do interaktivity (Time To Interactive - TTI) dosahuje Qwik nejlepších výsledků - je přibližně 4,2krát rychlejší než React, 3,5krát rychlejší než Vue.js a 2,8krát rychlejší než Next.js. Také v porovnání s Next.js má Qwik o 0,5 s rychlejší index rychlosti (Speed Index) [20]. V benchmarku Lighthouse dosahuje Qwik skóre 68, což je shodné s Next.js, ale vyšší než React (67) and Vue.js (66). Při zátěžových testech si Qwik udržuje stabilní výkon, zatímco skóre Angularu a Reactu klesá až na 26 [21].

Qwik vyniká v rychlosti načítání díky své architektuře „resumable execution“ a lazy-loadingu.

```
1 import { component$, useClientEffect$ } from
2     '@builder.io/qwik';
3
4     export const LazyLoadedExercise = component$(()
5         => {
6             useClientEffect$(() => {
7                 // Kod se provede pouze na klientovi
8             });
9     return <div>Cviceni se nacte, az kdyz bude potreba</
10    div >;
11 });
```

Příklad 1: Příklad Qwik komponentu s lazy-loadingem

Na druhou stranu, tradiční frameworky jako React nebo Angular často vyžadují načtení většího množství Javascriptu.

SSR a hydratace

Co se týče schopností server-side renderingu (SSR), Qwik patří mezi nejpokročilejší frameworky. V porovnání s Reactem, Vue.js a Next.js nabízí Qwik efektivnější přístup díky eliminaci hydratace, což umožňuje aplikacím dosahovat okamžité interaktivity s minimálním množstvím JavaScriptu. Právě toto zlepšuje výkon zejména na pomalejším internetu nebo zařízeních [22].

Vývojářská náročnost

V oblasti uživatelské zkušenosti vývojářů (developer experience) dosahuje Qwik hodnocení 3,5 z 5, což je stejné jako React (také 3,5), ale například Vue.js dosahuje hodnocení 4,5 z 5 [23]. Qwik poskytuje jednodušší a intuitivnější způsob práce se server-side a client-side kódem bez nutnosti explicitního rozdělování komponentů [22].

Komunita

V současné chvíli je největší nevýhoda Qwiku menší komunita ve srovnání s již existujícími frameworky. Zatímco například React dosahuje v této oblasti hodnocení 5 z 5, Qwik dosahuje pouze 3,5 z 5. Tato situace se však může změnit s rostoucí popularitou frameworku [23].

2.2.3 Výhody a nevýhody frameworku Qwik

Výhody:

Qwik přináší několik inovativních funkcí:

- **Resumabilita je základ:** Qwik serializuje stav aplikace na server, což umožňuje okamžité pokračování na klientovi bez nutnosti inicializovat celý framework.

1. **Embedování stavu do HTML:** Qwik při SSR generuje HTML s vloženým JSON objektem (`<script type="qwik/json">`), který

obsahuje:

- Stav aplikace (aktuální data formulářů).
- Kontext frameworku (informace o komponentách, jejich vztazích a event handlerech)
- QRL (Qwik Resource Locators) - cesty k funkcím (např. `./components/button#handleClick`).

2. **Komponentní metadata:** Každý DOM element dostává atributy jako `q:host` (identifikace komponentu) a `q:obj` (ID serializovaného objektu v JSON) [19].

- **Klientovská deserializace**

1. **Lazy rekonstrukce stavu:** Qwikloader (1kB runtime) parsuje JSON a připraví proxy objekty, které načítají skutečná data až při interakci (tzv. lazy execution).

2. **Dynamický import kódu:** Při kliknutí na tlačítko s atributem `on:click=".button#clickHandler"`:

(a) Qwikloader detekuje QRL v atributu

(b) Načte pouze soubor `button.js` (pomocí `import(.button.js)`).

(c) Spustí funkci `clickHandler` z načteného modulu [19].

- **QRL (Qwik Resource Locator**

1. **Struktura:** `cesta_k_souboru#název_funkce?parametry`.

2. **Příklad:**

```
1 <!-- Serverem vygenerovane HTML -->
2 <button on:click = "./src/components/Form#submit?
  data=123" > Odeslat </button>
```

Příklad 2: Příklad QRL

- `.src/components/Form` - cesta k modulu s funkcí
- `submit` - název handleru
- `data=123` - serializované parametry [19]

- **Výkonové optimalizace Incrementální streamování:** Qwik serializuje komponenty po částech (např. nejprve nadpis, poté obsah), což umožňuje rychlejší TTI (Time To Interactive) [19]

Deduplikace kódu: Pokud více komponentů používá stejnou funkci (např. `formatDate`), Qwik ji načte pouze jednou [19].

Nevýhody:

- **Menší komunita a omezená podpora**

Velikost komunity: Qwik má stále relativně malou uživatelskou základnu ve srovnání s již existujícími frameworky, jako jsou React nebo Vue.js. To znamená méně dostupných zdrojů, tutoriálů a diskusních fór, což může zpomalit učení a řešení problémů [24] [25].

Dostupnost knihoven třetích stran: Kvůli omezené popularitě je počet knihoven optimalizovaných pro Qwik výrazně nižší než u ostatních frameworků. Vývojáři musí dost často vytvářet vlastní řešení nebo dokonce opravovat existující knihovny, což zvyšuje časovou náročnost [24].

- **Náročnost na naučení**

Inovativní přístup: Qwik vyžaduje od vývojářů pochopení unikátních konceptů, jako například *resumability*. Tyto funkce nejsou běžné u ostatních frameworků, což může ztížit adaptaci pro vývojáře, kteří jsou zvyklí na tradiční modely [25].

Přechod na DOM-centric přístup: Vývojáři musí změnit svůj způsob práce, konkrétně musí změnit klasickou orientaci na paměť (heap-centric) na orientaci na DOM (DOM-centric), což vyžaduje přeučení některých základních principů [26].

- **Omezení serializace**

Komplexní datové struktury: Qwik má problémy se serializací složitých datových struktur, jako jsou objekty s metodami (např. `Date`). Tyto struktury musí být přepsány do jednodušších formátů, což může vést k chybám nebo nečekanému chování aplikace [25] [26]. Vývojáři kvůli tomu musí používat různé workarouny pro serializaci, což zvyšuje složitost kódu a může i zpomalit vývoj [26].

- **Problémy s kompatibilitou**

Integrace s již existujícími systémy: Kvůli odlišné architektuře může být složitější integrovat Qwik do již existujících projektů založených na jiných frameworkcích. To platí zejména pro systémy využívající tradiční hydrataci nebo synchronní rendering [19] [25].

3 Praktická část

3.1 Motivace k vytvoření aplikace Sea Quest

Výuka cizích jazyků se v současné době neustále vyvíjí a moderní technologie poskytují stále více inovativních způsobů, jak motivovat žáky a zlepšit výuku. Motivací pro vytvoření aplikace Sea Quest byla snaha o vytvoření interaktivní aplikace s prvky gamifikace pro lepší zapojení žáků a zároveň snaha o poskytnutí efektivního nástroje pro učitele a rodiče.

3.2 Identifikace problémů ve výuce angličtiny

Tradiční metody výuky angličtiny se často potýkají s nízkou motivací žáků a nedostatečným individuálním přístupem. Mezi hlavní problémy patří:

- **Pasivní učení** - žáci si často neosvojují jazyk aktivně a chybí jim možnost okamžité zpětné vazby.
- **Nedostatek personalizace** - učebnice a standardizované metody ne vždy odpovídají individuálním potřebám žáků.
- **Nízká motivace** - monotónní čtení textů, vypracovávání cvičení z učebnic snižuje zájem žáků o studium.

Aplikace Sea Quest byla navržena tak, aby tyto problémy řešila a nabídla moderní přístup k výuce angličtiny prostřednictvím interaktivních cvičení a herních mechanismů.

3.3 Přínos aplikace Sea Quest

Sea Quest kombinuje několik klíčových prvků pro zlepšení jazykové vzdělání:

- **Gamifikace** - využití herních prvků, jako jsou zkušenosti (XP), úrovně a odměny, žebříčky, podporuje motivaci žáků k pravidelnému studiu.

- **Interaktivní výuka** - žáci si mohou procvičovat slovní zásobu a grammatiku prostřednictvím kvízů, flashcards a skládání vět.
- **Správa tříd** - učitelé mohou efektivně organizovat výuku, přidělovat úkoly a zadávat testy.
- **Zapojení rodičů** - rodiče mají přístup k přehledu pokroku svého dítěte a mohou sledovat jeho výsledky a zda plní školní povinnosti.

Díky těmto vlastnostem poskytuje Sea Quest efektivní a zábavný způsob, jak se učit angličtinu, a zároveň usnadňuje práci učitelům.

3.4 Představení aplikace Sea Quest

3.4.1 Responzivita aplikace

Aplikace je navržena tak, aby byla responzivní, což znamená, že se přizpůsobuje různým typům zařízení, ať už jde o telefony, tablety nebo notebooky. Díky tomu má uživatel komfortní rozložení obsahu bez ohledu na velikost obrazovky.

Adaptivní layout

Na desktopových zařízeních je menu (sidebar) v levé části obrazovky a hlavní obsah aplikace se objevuje uprostřed. Na druhou stranu, u mobilních zařízení se toto menu přemístí dolů na spodní stranu obrazovky, kde je snadno dosažitelné pomocí prstů bez jakýchkoliv složitějších pohybů. Hlavní obsah je poté také zobrazován uprostřed obrazovky. Tento přístup zajišťuje, aby uživatel mohl aplikaci otevřít kdekoliv a pohodlně ji využívat.

3.4.2 Přihlášení a registrace

Přihlášení do aplikace

Přihlášení do aplikace je prvním krokem přístupu do aplikace, který umožňuje přístup k dalším funkcím. Aplikace využívá Firebase autentizaci, která zajišťuje bezpečné ověřování uživatelů prostřednictvím emailu a hesla.

Přihlašovací formulář

Přihlašovací formulář je první stránka, kterou uživatel uvidí po spuštění aplikace (pokud již není přihlášený). Obsahuje dvě vstupní pole: jedno pro email a druhé pro heslo. Dále jsou zde tlačítka pro potvrzení přihlášení, pro změnu okna na přihlášení rodiče a také dva odkazy: jeden pro registraci a druhý pro zapomenuté heslo.

The image shows a login form titled "Přihlášení" (Login) on a dark background. It features two input fields: "Email" and "Heslo" (Password). Below the fields are two buttons: a large blue button labeled "PŘIHLÁSIT SE" (Login) and a smaller purple button labeled "PŘIHLÁSIT SE JAKO RODIČ" (Login as Parent). At the bottom, there are two links: "Ještě nemáte vytvořený účet? [Registrovat se](#)" (Don't you have an account yet? [Register](#)) and "Zapomněli jste heslo? [Obnovit heslo](#)" (Forgot your password? [Reset password](#)).


Obrázek 1: Přihlašovací formulář

```
1     if (state.isRegistering) {
2         const userCredential = await
3             createUserWithEmailAndPassword(auth, state.
4                 email, state.password);
5         const user = userCredential.user;
6
7         await setDoc(doc(db, "users", user.uid), {
8             userId: user.uid,
9             name: state.name,
10            role: state.role,
11            parentCode: await generateParentCode(),
12            avatar: "/avatars/default.png",
13            createdAt: new Date().toISOString(),
14            experience: 0,
15            level: 1,
16            perfectScores: 0,
17            quizCompleted: 0,
18            achievements: [],
19            enrolledClassrooms: [],
20            homework_completed: 0
21        });
22     } else {
23         const userCredential = await
24             signInWithEmailAndPassword(auth, state.
25                 email, state.password);
26         const user = userCredential.user;
27         console.log("User logged in:", user.uid);
```

Příklad 3: Kód pro přihlašování a registraci

Přihlašovací formulář pro rodiče

Přihlášení pro rodiče funguje na principu přihlašování kódem, kdy se při vytvoření žákovského účtu vygeneruje šestimístný kód pro rodiče, který umožní nahlédnout do statistik žáka.



Přihlášení

Rodičovský kód

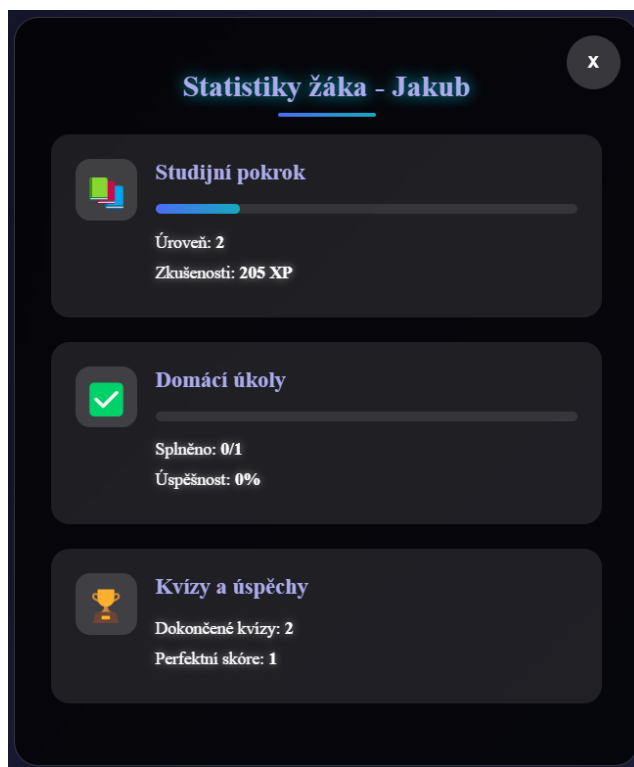
PŘIHLÁSIT SE

PŘIHLÁSIT SE JAKO STUDENT

Ještě nemáte vytvořený účet? [Registrovat se](#)

Obrázek 2: Přihlašovací formulář pro rodiče

Rodič může zkontrolovat, zda je žák aktivní a využívá aplikaci, jestli plní domácí úkoly nebo jak se mu daří v procvičování.



Obrázek 3: Statistiky žáka po přihlášení rodiče

```
1 if (state.isParentLogin) {  
2   const childQuery = query(collection(db, "users"),  
3     where("parentCode", "=", state.parentCode));  
4   const childSnap = await getDocs(childQuery);  
5   // [...]  
6 }
```

Příklad 4: Kód pro rodičovské přihlášení

```
1 <div class="stats-card">
2   <div class="stats-icon">IKONA_CHECK_MARK</div>
3   <div class="stats-content">
4     <h4>Domaci ukoly</h4>
5     <div class="progress-bar">
6       <div class="progress-fill" style={{ width:
7         `${state.childStats.totalHomeworks ?
8         (state.childStats.completedHomeworks / state.
9           childStats.totalHomeworks)
10          * 100 : 0}%` }}></div>
11     </div>
12     <p>Splneno: <span class="highlight">{state.
13       childStats.completedHomeworks}
14       /{state.childStats.totalHomeworks || 0}</span></p>
15     <p>Uspesnost: <span class="highlight">{state.
16       childStats.totalHomeworks ?
17       Math.round((state.childStats.completedHomeworks /
18         state.childStats.totalHomeworks) * 100) : 0}%</span>
19       </p>
20   </div>
21 </div>
```

Příklad 5: Statistika žáka z pohledu rodiče

Přihlášení pro učitele

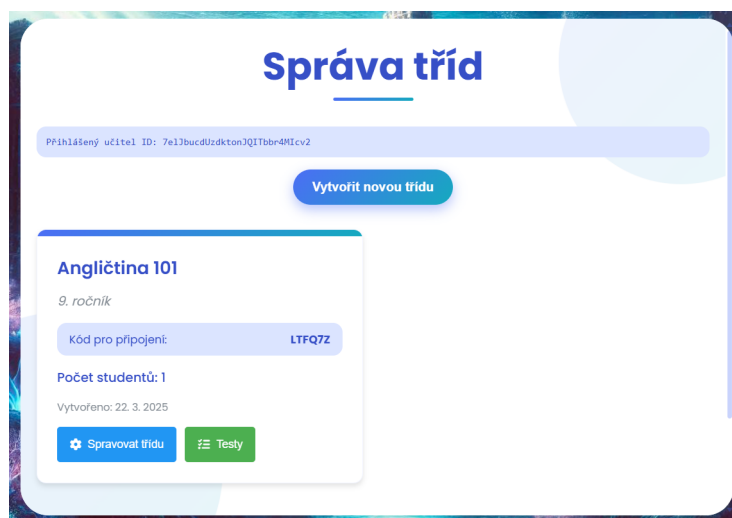
Učitelé se přihlašují stejným způsobem jako žáci, a to přes přihlašovací formulář (viz. obr.1) za pomoci emailu a hesla. Také si zatím ale nemohou vytvořit účet prozatím je aplikace navržena tak, aby se mohli registrovat pouze žáci, protože aplikace nebyla testována v takové míře, aby bylo nutné vytvářet separátní kód pro učitele.

3.4.3 Systém tříd

Aplikace Sea Quest implementuje systém správy tříd, který slouží jako základní organizační prvek pro učitele. Byla navržena tak, aby měl učitel přehled o žácích, jejich výsledcích a organizaci hodin.

Hlavní funkce systému tříd

- **Správa tříd:** Funkce, která umožňuje učiteli vytvářet a mít přehled o svých třídách. U každé třídy lze vidět základní informace: název třídy, ročník, počet studentů a hlavně kód pro připojení do třídy. Také jsou tam dvě tlačítka, která umožňují přejít na dashboard třídy a nebo přímo na testy.



Obrázek 4: Správa tříd

Formulář pro vytvoření třídy obsahuje tři povinná pole: název třídy, ročník a jednoduchý popis třídy.

Obrázek 5: Formulář pro vytvoření třídy

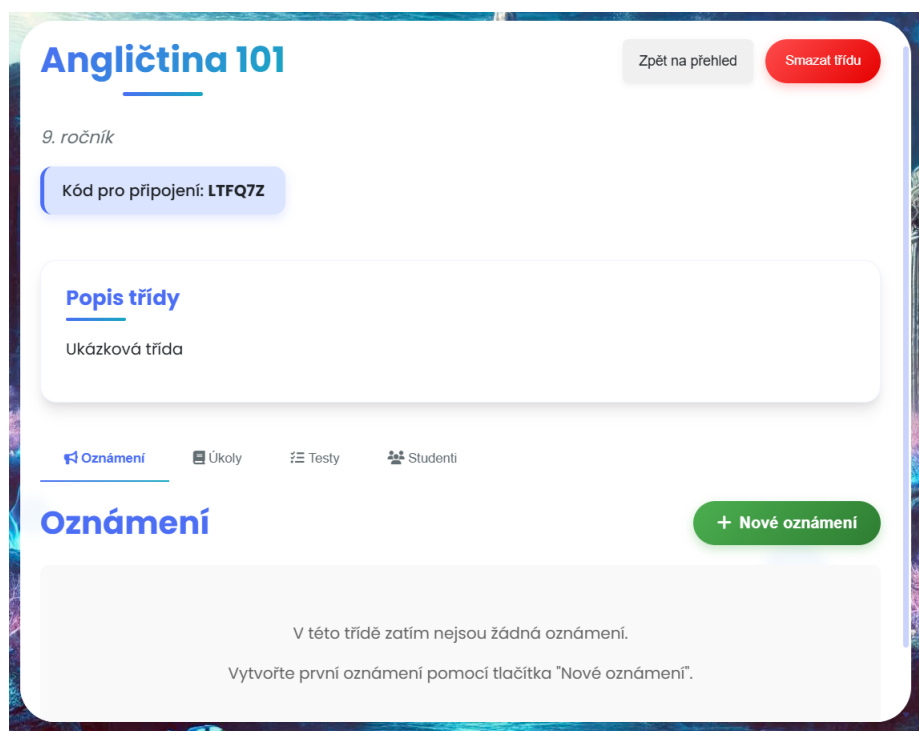
```
1 const handleJoinClassroom = $(async () => {
2   if (!joinCode.value.trim()) {
3     joinErrorMessage.value = "Zadejte kod tridy";
4     return;
5   } try {
6     isJoining.value = true;
7     joinErrorMessage.value = '';
8     if (!user.value?.uid) {
9       joinErrorMessage.value = "Nejste prihlaseni";
10      return;}
11    const classroom = await joinClassroom(joinCode.
12      value.trim(), user.value.uid);
13    if (!classrooms.value.find(c => c.id === classroom
14      .id)) {
15      classrooms.value = [...classrooms.value,
16        makeSerializable(classroom)];
17    }
18    isJoinModalOpen.value = false;
19    joinCode.value = '';
20    const updatedProfile = await getUserProfile();
21    user.value = makeSerializable(updatedProfile);
22  } catch (error: any) {
23    console.error("Error joining classroom:", error);
24    joinErrorMessage.value = error.message || "
    Nepodarilo se pripojit ke tride. Zkontrolujte
    kod a zkuste to znovu.";
```

Příklad 6: Kód pro připojení do třídy

- **Dashboard třídy:** Přehled třídy umožňuje učitelům i studentům přístup k důležitým informacím, jako jsou oznámení, úkoly, testy a také kód pro připojení do třídy.

1. Záhloví:

- Obsahuje název třídy (Angličtina 101), ročník (9. ročník) a unikátní šestimístný kód pro připojení do třídy (LTFQ7Z).
- K dispozici jsou také dvě tlačítka:
 - * *Zpět na přehled:* Vrábí uživatele na přehled tříd.
 - * *Smazat třídu:* Funkce, která je dostupná pouze pro učitele, umožňuje odstranit třídu.



Obrázek 6: Dashboard třídy

- **Popis třídy:** Zobrazuje stručný popis třídy (*Ukázková třída*), pro lepší orientaci.
- **Záložky:** Jsou dostupné čtyři záložky:

- *Oznámení*: Důležitá oznámení učitele.
- *Úkoly*: Učitel zde může zadávat a hodnotit úkoly. Žáci mají přehled o jednotlivých úkolech.
- *Testy*: Prostor pro testování žáků.
- *Studenti*: Seznam studentů, kteří jsou ve třídě.

Oznámení

Záložka *Oznámení* slouží ke sdělení důležitých informací mezi učitelem a studenty. Na obrázku č.7 je zobrazen příklad oznámení z pohledu učitele. Pohled studenta je totožný jako pohled učitele s výjimkou ovládacích tlačítek, která jsou unikátní jen pro učitele.

Hlavní prvky:

1. Nové oznámení:

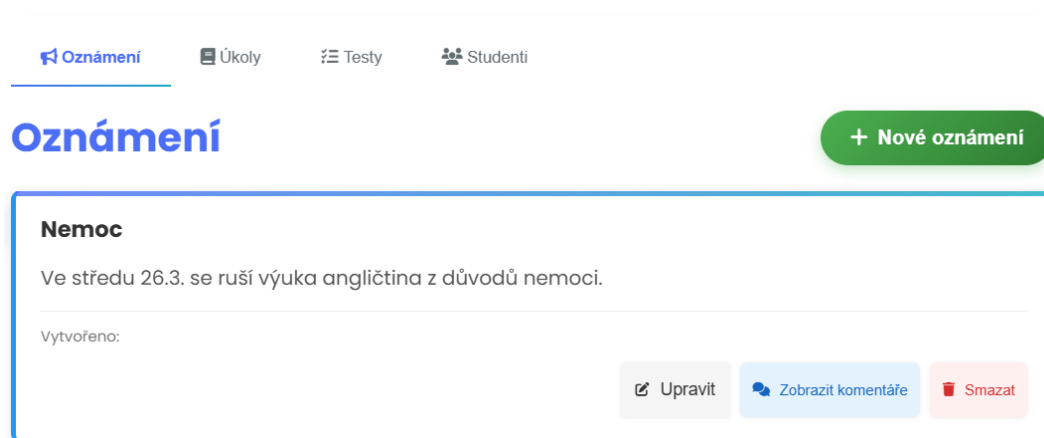
- Učitel má možnost přidat nové oznámení pomocí tlačítka *+ Nové oznámení*. Pro lepší přehlednost je zvýrazněno zelenou barvou.

2. Seznam oznámení:

- Oznámení jsou zobrazována jednotlivě a obsahují:
 - **Nadpis:** Stručný název oznámení (např. *Nemoc*)
 - **Obsah:** Text sdělení (např. *Ve středu 26.3.se ruší výuka angličtiny z důvodu nemoci*)
 - **Datum vytvoření:** Informace o datu zveřejnění oznámení.

3. Ovládací tlačítka:

- Každé oznámení obsahuje tři tlačítka:
 - *Upravit:* Umožňuje dodatečně upravit oznámení.
 - *Smazat:* Slouží k odstranění oznámení.
 - *Zobrazit komentáře:* Zobrazuje komentáře, které mají žáci možnost napsat pod jakékoliv sdělení.



Obrázek 7: Příklad oznámení

Úkoly

Záložka *Úkoly* slouží k zadávání, správě a sledování domácích úkolů. Na obrázku č.8. Stejně jako u *Oznámení* je pohled studenta stejný, s výjimkou ovládacích tlačítek.

Hlavní prvky:

1. Nový úkol:

- Učitel má možnost přidat nový úkol pomocí tlačítka *+ Nový úkol*. Pro lepší přehlednost je zvýrazněno zelenou barvou.

2. Seznam Úkolů:

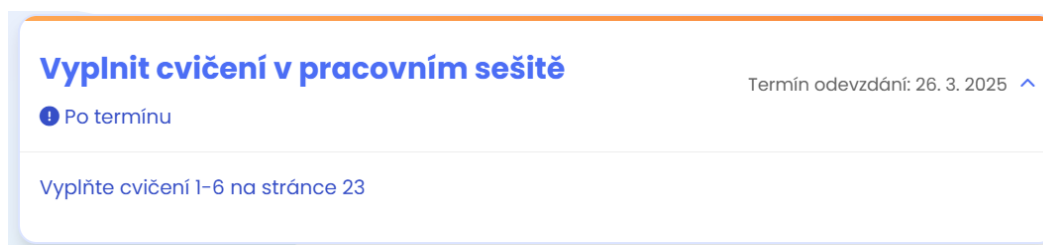
- Úkoly jsou zobrazovány jednotlivě a obsahují:
 - **Nadpis:** Stručný název úkolu (např. *Vyplnit cvičení v pracovním sešitě*)
 - **Obsah:** Popis úkolu (např. *Vyplňte cvičení 1-6 na stránce 23*)
 - **Termín odevzdání:** Informace o datu odevzdání.
 - **Odevzdáno:** Položka, kterou vidí pouze učitel, pro lepší organizaci.

3. Ovládací tlačítka:

- Každý úkol obsahuje tři tlačítka:
 - *Upravit*: Umožňuje dodatečně upravit úkol.
 - *Smazat*: Slouží k odstranění úkolu.
 - *Zobrazit stav*: Zobrazí seznam studentů a dovolí učiteli zapsat splnění úkolů.



Obrázek 8: Příklad úkolu z pohledu učitele



Obrázek 9: Příklad úkolu z pohledu studenta

Úkoly

V záložce *Testy* se dají vytvořit a vypracovat. Na obrázku č.10

Hlavní prvky:

1. Nový úkol:

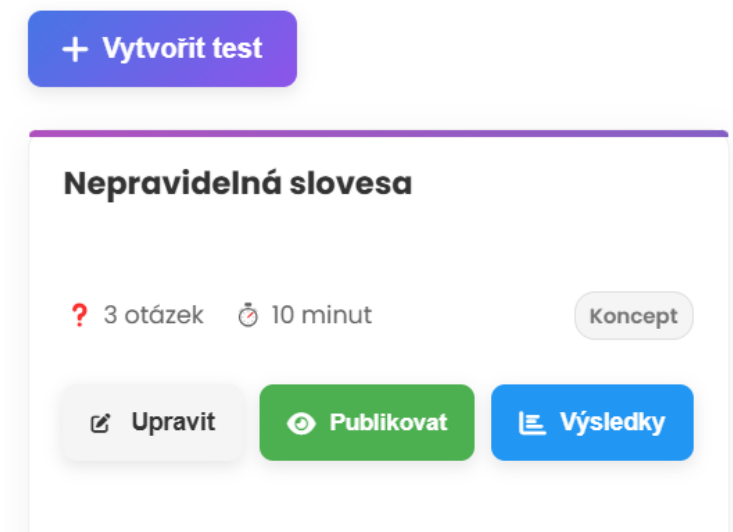
- Učitel má možnost přidat nové testy pomocí tlačítka + *Vytvořit test*. Pro lepší přehlednost je zvýrazněno fialovým gradientem.

2. Seznam testů:

- Testy jsou zobrazovány jednotlivě a obsahují:
 - **Nadpis:** Stručný název testu (např. *Nepřavidelná slovesa*)
 - **Základní informace:** Kolik má test otázek, jaký má časový limit.
 - **Stav:** Pokud test ještě není dostupný pro studenty, tak je označen jako koncept. Pokud je již dostupný pro studenty, tak je označen jako publikovaný.

3. Ovládací tlačítka:

- Každý test obsahuje tři tlačítka:
 - *Upravit:* Umožňuje dodatečně upravit test.
 - *Publikovat:* Po vytvoření je test skrytý pro žáky. Učitel ho musí ručně publikovat, aby ho otevřel žákům, kteří ho poté mohou vypracovat.
 - *Výsledky:* Zobrazí seznam studentů s přehledem o tom, jaké známky mají.



Obrázek 10: Příklad testu

Na obrázku č. 11 je příklad průběhu testu, jak ho vidí žák.

Hlavní části:

- **Otázky:** Jasná a stručná otázka (*Jaký je druhý tvar slovesa go?*).
- **Odpovědi:** Aplikace prozatím umožňuje vytvářet pouze testy typu abcd.



Obrázek 11: Průběh testu

3.4.4 Procvičování

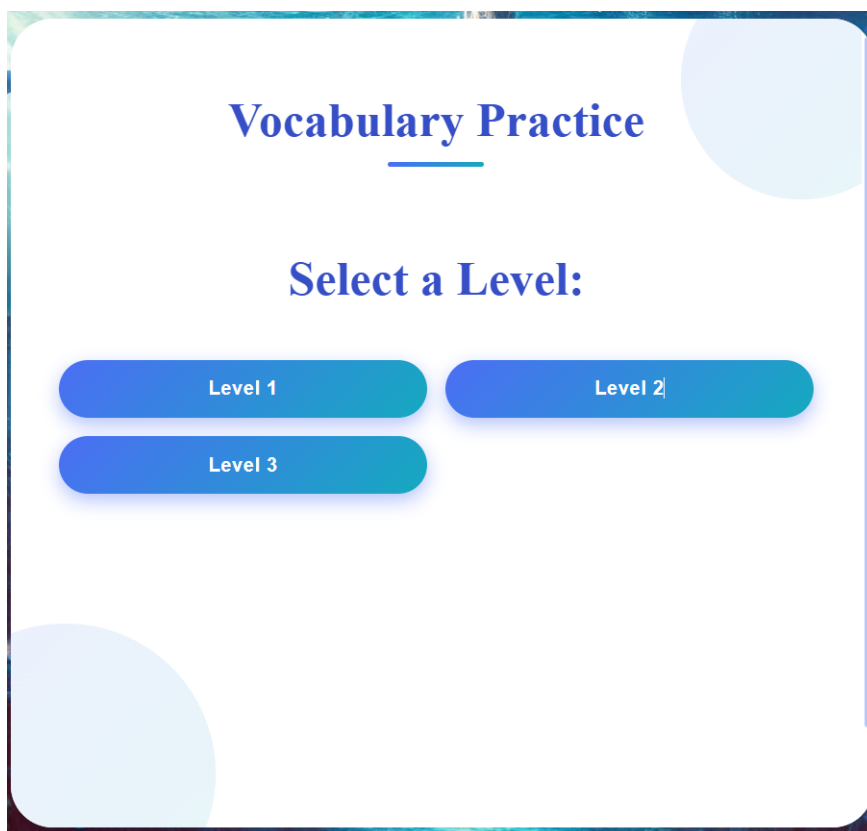
Sekce *Cvičení* nabízí volbu mezi třemi různými typy cvičení na procvičování svých dovedností. Na obrázku č. 12 jsou zobrazeny možnosti: *Vocabulary*, *Letter hunt* a *Sentences*.



Obrázek 12: Výběr cvičení

Vocabulary:

Sekce *Vocabulary* slouží k procvičování stávající a nové slovní zásoby. Je možné si vybrat z několika úrovní rozdělených podle náročnosti. Momentálně jsou tam 3 základní úrovně.

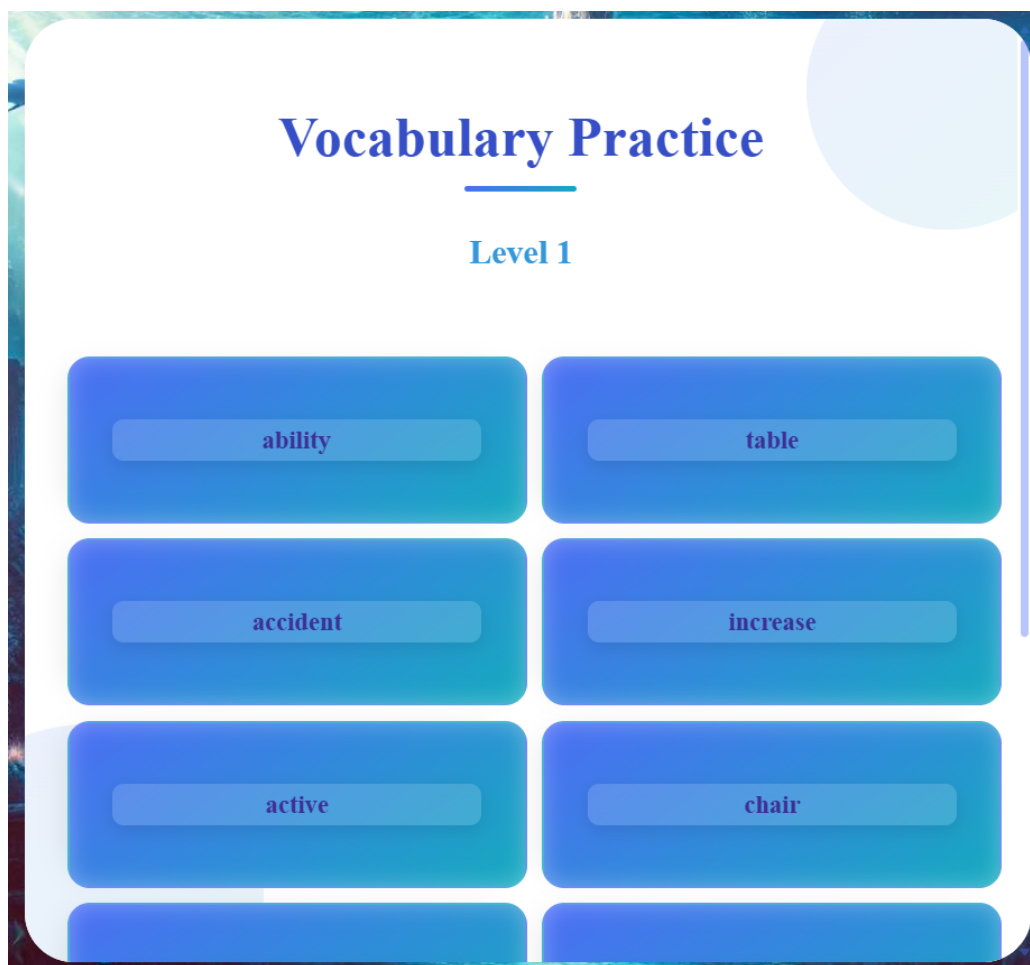


Obrázek 13: Výběr úrovně

Po výběru úrovně se uživateli zobrazí seznam kartiček se slovy, která se procvičují v dané úrovni. Každá kartička má dvě strany:

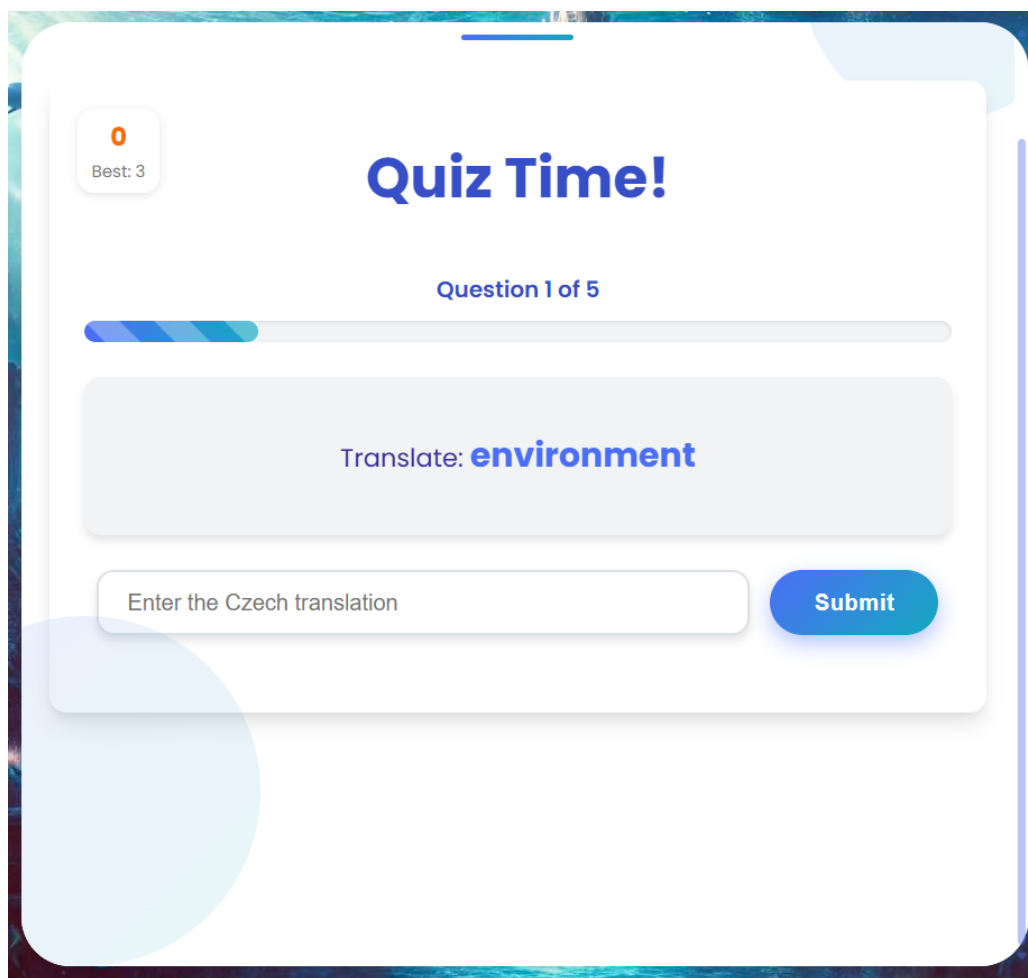
- **Modrá strana:** Přední strana s anglickým názvem
- **Oranžová strana:** Zadní strana s českým názvem

Poté co si uživatel bude jistý svou znalostí procvičovaných slov, může přejít na test pomocí tlačítka *Start Quiz*.



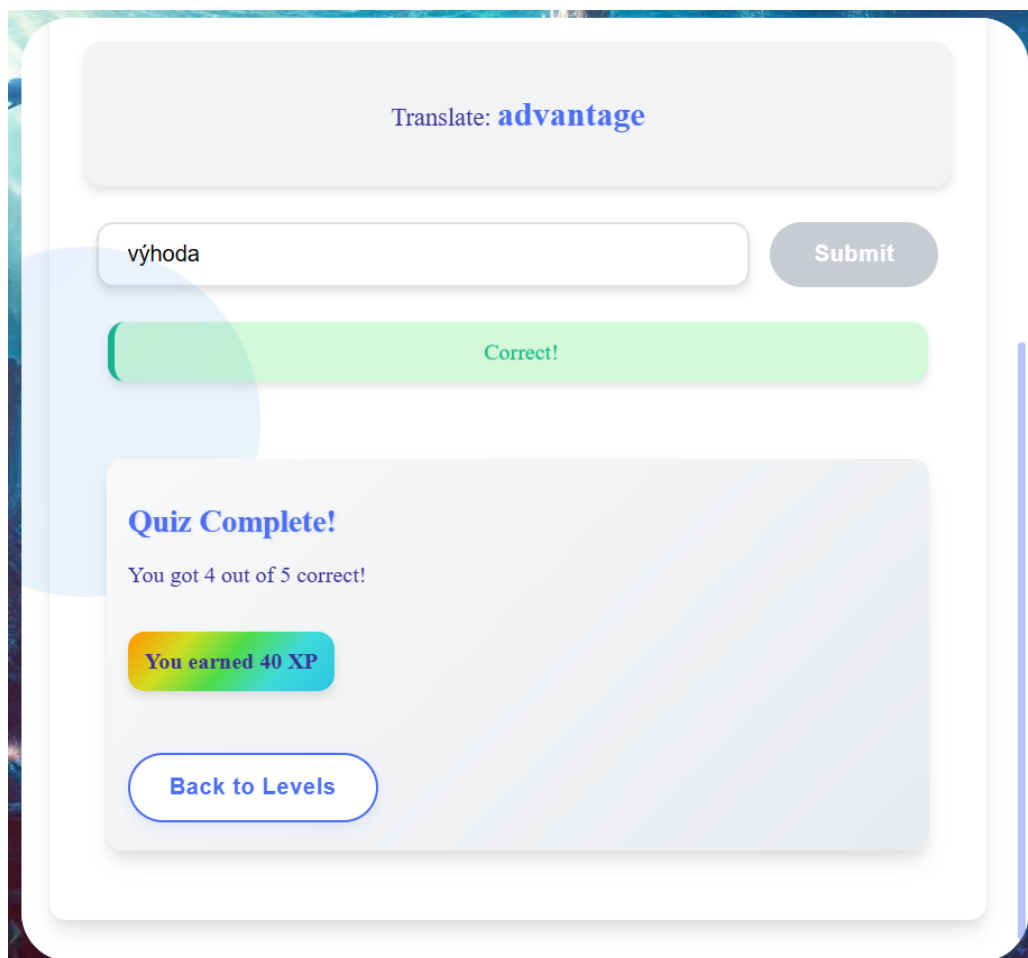
Obrázek 14: Flashcards

V samotném kvízu má uživatel za úkol přeložit 5 náhodných anglických slovíček, která se objevila v první části aktivity, jak lze vidět na obrázku č.14. Není stanoven žádný časový limit, takže uživatel může odejít z aktivity kdykoliv, ale pokud ji nedokončí, nedostane žádné odměny za splnění.



Obrázek 15: Aktivita Vocabulary

Po dokončení kvízu se uživateli zobrazí výsledky společně se zkušenostmi (*XP*) a tlačítko *Back to Levels* pro výběr úrovně.



Obrázek 16: Výsledky kvízu

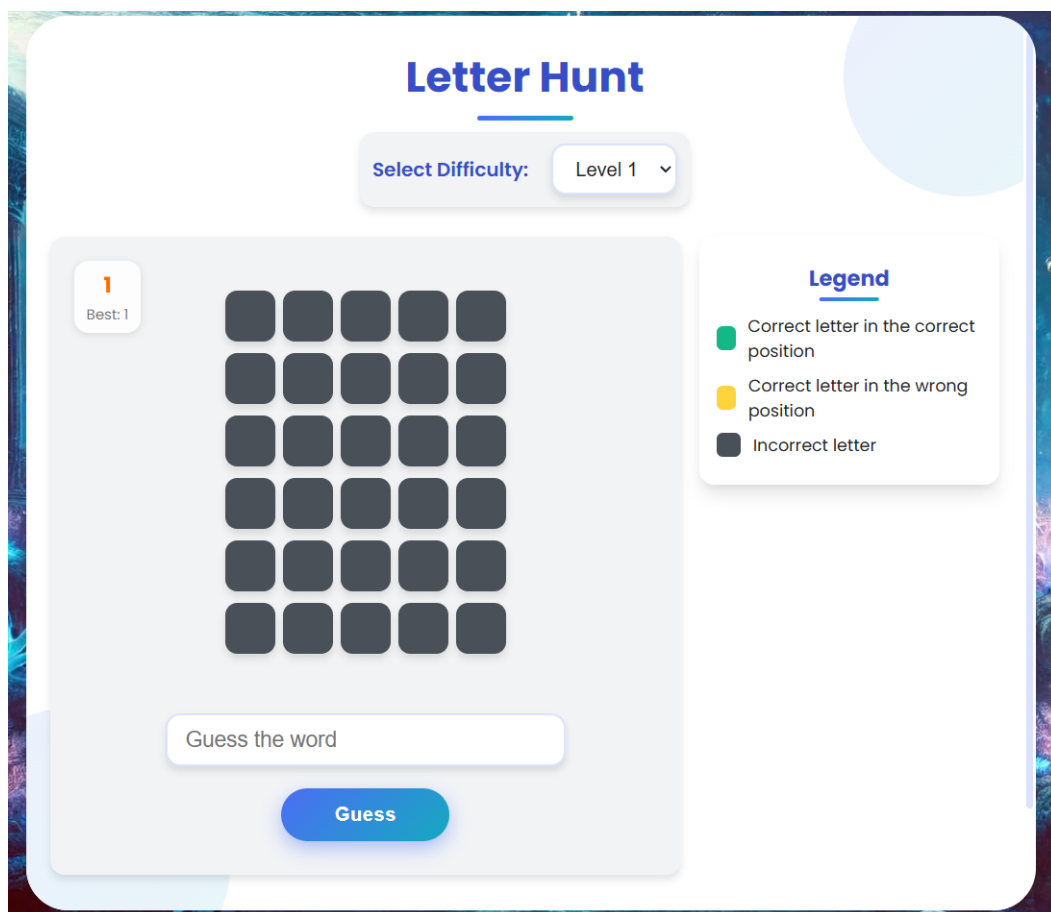
```
1 export default component$(() => {
2   const location = useLocation();
3   const levelId = location.params.id;
4
5   const vocabularyResource = useResource$<VocabularyItem
6     []>(async () => {
7     if (levelId) {
8       const levelDocRef = doc(db, 'sections/vocabulary/
9         levels/${levelId}');
10      const levelSnapshot = await getDoc(levelDocRef);
11
12      if (levelSnapshot.exists()) {
13        return levelSnapshot.data().vocabularyItems ||
14          [];
15      } else {
16        console.error('No such level!');
17        return [];
18      }
19    }
20    return [];
21  });
```

Příklad 7: Kód pro získání dat levelu z Firestore

Letter hunt

Aktivita *Letter hunt* spočívá v hádání pětímístných slov. Po zadání slova se zkontroluje, zda vůbec existuje. Pokud ano, tak se vepíše do prvního řádku a každé písmeno se vybarví jednou ze tří barev:

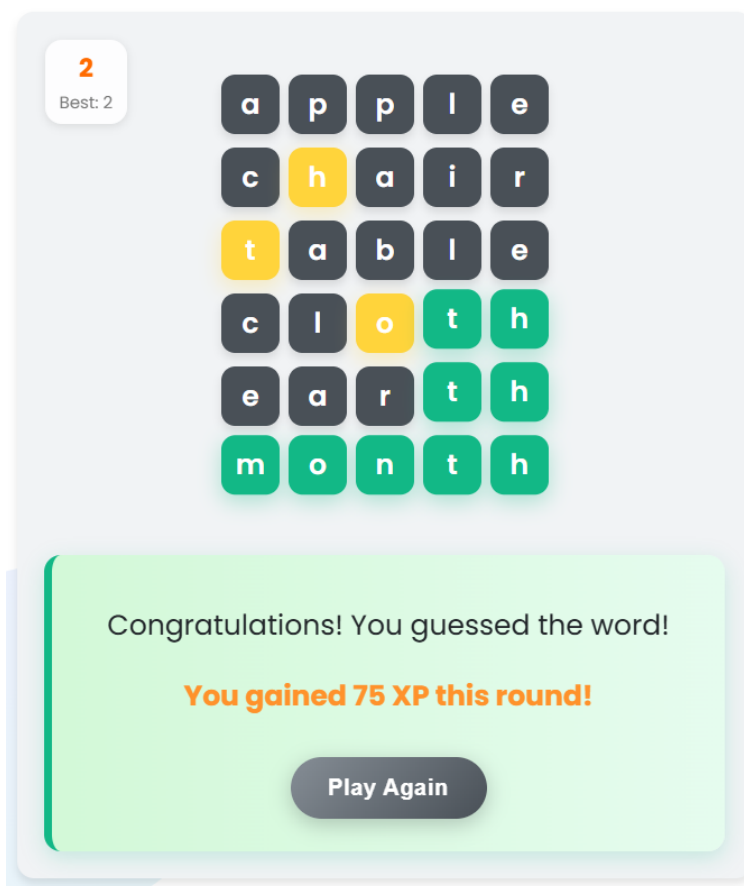
- **Černá/Žádná** - Písmeno není ve slově.
- **Žlutá** - Písmeno ve slově je, ale je na jiném místě.
- **Zelená** - Písmeno ve slově je a je také na správném místě.



Obrázek 17: Aktivita Letter hunt

Na obrázku č. 18 lze vidět výsledek aktivity. Uživatel dostane nejmenší možný počet zkušeností, protože bylo slovo uhodnuté až na poslední pokus

(75 XP).



Obrázek 18: Výsledek aktivity Letter hunt

```
1 const analyzeGuess = $(async (guess: string, word:  
  string) => {  
2   const result = new Array(word.length).fill('wrong');  
3   guess.split('').forEach((letter, index) => {  
4     if (letter === word[index]) {  
5       result[index] = 'correct';  
6     } else if (word.includes(letter)) {  
7       result[index] = 'wrong-place';  
8     }  
9   });  
10  return result;  
11 });
```

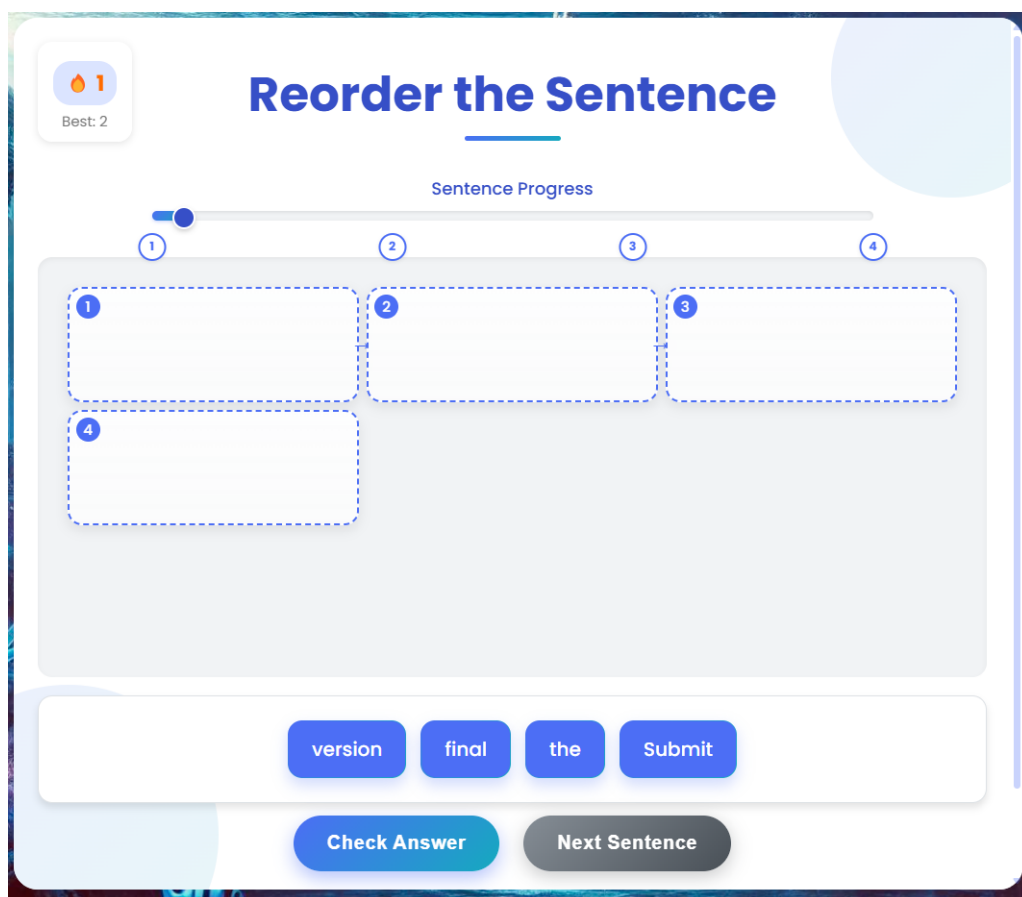
Příklad 8: Kontrola hádaného slova

```
1 const checkIfValidWord = $(async (word: string) => {
2     if (!word) return false;
3
4     if (word.length < 2 || word.length > 15) {
5         return false;
6     }
7
8     const url = 'https://api.dictionaryapi.dev/api/v2/
9         entries/en/${word}';
10    try {
11        const response = await axios.get(url);
12        return response.data.length > 0;
13    } catch (error) {
14        return false;
15    }
16 });
```

Příklad 9: Kontrola zda slovo opravdu existuje

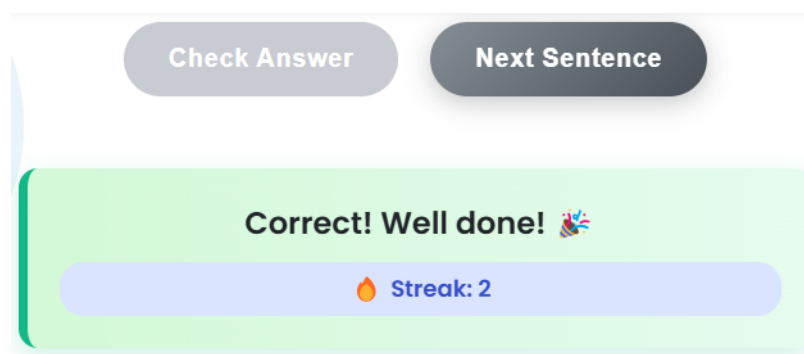
Skladba vět

Poslední aktivita *Sentences* je unikátní v tom, že se nejedná o samostatná slova, ale o celé věty. Uživatel má za úkol přeskládat větu do správného pořadí. Věta je vždy rozložená na slova a ta jsou náhodně přeházená. Uživatel se je následně snaží poskládat do boxů tak, aby věta byla gramaticky správně.



Obrázek 19: Aktivita Sentences

Po dokončení každé věty uživatel zjistí, zda je správně nebo špatně. Pokud je správně, tak získá zkušenosti (*XP*).



Obrázek 20: Výsledek aktivita Sentences

```
1 const setupNewSentence = $(async () => {  
2   const newSentence = await getRandomSentence();  
3   state.sentence = newSentence;  
4   state.correctOrder = state.sentence.split(' ');  
5   state.shuffledWords = await shuffleArray([...state.  
   correctOrder]);  
6   state.userOrder = Array(state.correctOrder.length).  
   fill('');  
7   state.gameOver = false;  
8   state.correct = false;  
9   selectedWordIndex.value = null;  
10 });
```

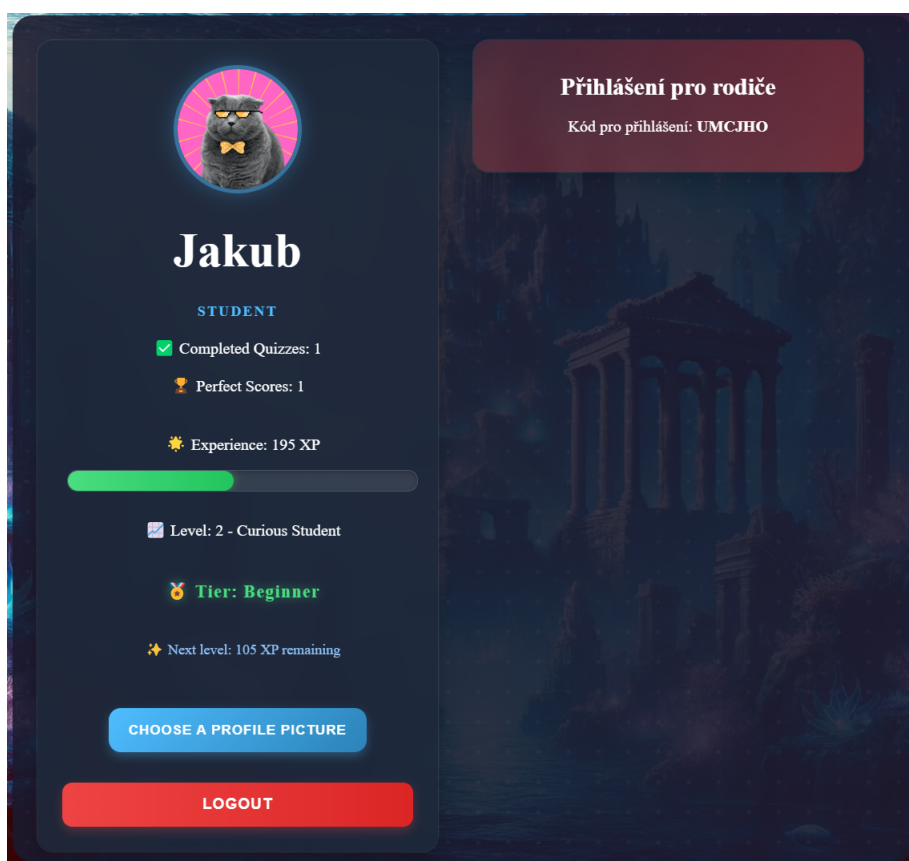
Příklad 10: Kód pro rozložení věty

```
1 const handleDragStart = $((event: DragEvent, index:
   number) => {
2   if (event.dataTransfer) {
3     event.dataTransfer.setData('text/plain', state.
       shuffledWords[index]);
4     event.dataTransfer.effectAllowed = 'move';
5   }
6   state.draggedIndex = index;
7   state.dragging = true;});
8
9 const handleDrop = $((event: DragEvent, dropIndex:
   number) => {
10  event.preventDefault();
11  if (state.draggedIndex !== -1) {
12    const draggedWord = state.shuffledWords[state.
       draggedIndex];
13    const targetWord = state.userOrder[dropIndex];
14
15    state.userOrder[dropIndex] = draggedWord;
16    state.shuffledWords[state.draggedIndex] =
       targetWord;
17
18    state.draggedIndex = -1;
19    state.dragging = false;
20    state.overIndex = -1;
21  }));
```

Příklad 11: Ukázka části kódu pro přetahování slov

3.4.5 Profil

Profil je jedním z hlavních gamifikačních prvků aplikace. Souží k motivaci žáků a sledování jejich pokroku. Uživatelé zde vidí počet splněných kvízů a kvízů splněných bez chyby (*Perfect Scores*), svou úroveň a další.

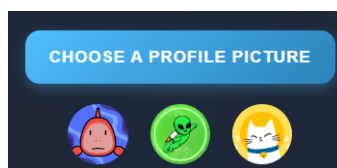


Obrázek 21: Ukázka profilu

```
1 const handleSignOut = $(async () => {  
2   try {  
3     await signOutUser();  
4     user.value = null;  
5     window.location.href = "/login";  
6   } catch (error) {  
7     console.error("Sign-out failed:", error);  
8   }  
9 });
```

Příklad 12: Kód pro odhlašování

Uživatelé si také mohou změnit profilový obrázek. Na výběr jsou čtyři druhy profilových obrázků. Na obrázku č. 22 jsou ukázány pouze tři, protože jeden obrázek je vždy aktivní.



Obrázek 22: Výběr profilového obrázku

```
1 const toggleAvatarSelection = $(() => {  
2   if (user.value) {  
3     availableAvatars.value = availableAvatars.value.  
4       filter(  
5         (avatar) => avatar !== user.value?.avatar  
6       );  
7     showAvatars.value = !showAvatars.value;  
8   });
```

Příklad 13: Kód pro výběr avataru

3.4.6 Systém úrovní

Úrovně (levely) jsou mechanikou používanou ve hrách, která motivuje uživatele k postupnému zlepšování a dosažení vyšších cílů.

Úrovně

- **Definice:** Úrovně reprezentují pokrok uživatele nebo postavy. Uživatel zpravidla začíná na nejnižší úrovni (např. Level 1) a časem postupuje výše sbíráním zkušenostních bodů (XP).
- **Funkcionalita:** Každá úroveň vyžaduje vyšší množství XP než předchozí, což zajišťuje, že s rostoucí úrovní roste i obtížnost.
- **Odměny:** Po dosažení nové úrovně uživatel získá nový název úrovně (např. Level 2 se nazývá *Curious Student*) a pokud splnil požadavky, tak i odznak.

Tiers (hodnost)

- **Rozdělení:** Úrovně jsou rozděleny do různých tierů, každý Tier má jiný název a barvu pro lepší přehlednost:

- *Level 1-5*: Beginner
- *Level 6-10*: Novice
- *Level 11-15*: Intermediate
- *Level 16-20*: Advanced
- *Level 21-25*: Expert
- *Level 26-30*: Master
- *Level 31+*: Grandmaster

```
1 export async function addExperience(userId: string,
  amount: number) {
2   try {
3     const userRef = doc(db, "users", userId);
4     const userSnap = await getDoc(userRef);
5
6     if (!userSnap.exists()) return false;
7
8     const userData = userSnap.data();
9     const newExperience = (userData.experience || 0) +
      amount;
10    let newLevel = userData.level || 1;
11    for (let i = newLevel; i < levels.length; i++) {
12      if (newExperience >= levels[i].requiredExperience)
13        {
14          newLevel = levels[i].id;
15        } else {
16          break;
17        }
18    }
```

Příklad 14: Úryvek kódu pro přidávání zkušeností

```
1 export const levels = [  
2     { id: 1, name: "Novice Learner",  
3       requiredExperience: 0, tier: "Beginner" },  
4     { id: 6, name: "Emerging Speaker",  
5       requiredExperience: 1000, tier: "Novice" },  
6     { id: 11, name: "Skilled Communicator",  
7       requiredExperience: 4000, tier: "Intermediate  
8     " },  
9     { id: 16, name: "Linguistic Explorer",  
10      requiredExperience: 11000, tier: "Advanced"  
11      },  
12     { id: 21, name: "Language Virtuoso",  
13      requiredExperience: 26000, tier: "Expert" },  
14     { id: 26, name: "Language Master",  
15      requiredExperience: 54000, tier: "Master" },  
16 ];
```

Příklad 15: Funkce pro získání zbývajících zkušeností do další úrovně

Zkušenosti (Experience points - XP)

- **Co jsou XP:** Zkušenosti měří pokrok uživatele. Lze je získat za splnění kvízů ve *Cvičení* nebo za splnění odznaků.
- **Jak je získat:** Každé cvičení a odznak má určený počet zkušeností, jakým odměňuje uživatele za splnění.

```
1     const baseXp = 200;
2     const unsuccessfulAttempts = 6 - state.
      attempts - 1;
3     const attemptPenalty = unsuccessfulAttempts *
      25;
4     const difficultyMultiplier = {
5       'A1': 1,
6       'A2': 1.5,
7       'B1': 2,
8     }[state.difficulty] || 1;
9
10    const finalXp = Math.max(50, Math.round((
      baseXp - attemptPenalty) *
      difficultyMultiplier));
11    await awardXp(finalXp);
```

Příklad 16: Příklad kalkulace zkušeností z aktivity *Letter hunt*

V kódu č. 16 lze vidět, jak jsou zkušenosti řešené v aktivitě *Letter hunt*. Vždy je základem 200 XP a za každý nevydařený pokus se odečítá 25 XP. To se ve výsledku násobí indikátorem obtížnosti.

3.4.7 Žebříček

Je dalším z hlavních gamifikačních prvků aplikace. Podporuje soutěživost a motivuje uživatele, aby byli nejlepší.

Popis žebříčku

1. Zobrazované údaje

- **Jméno:** Zobrazuje se jméno uživatele společně s profilovým obrázkem.
- **Level a tier:** Zobrazuje se také úroveň a tier uživatele a následně také počet XP. Podle XP se určuje pozice v žebříčku.

2. Rozdělení žebříčku: Žebříček se může přepínat mezi dvěma režimy:

- **All Time:** Zobrazuje se žebříček uživatelů za celou dobu.
- **Monthly:** Zobrazuje se žebříček uživatelů jen na aktuální měsíc.
- **Top n:** Uživatelé mohou filtrovat zobrazení na prvních 10, 20 nebo 50 pozic.

Přínosy žebříčku

1. Podpora soutěživosti:

- Uživatelé se snaží vylepšit své pořadí tím, že získávají XP body, což vede k tomu, aby pokračovali stále dál.
- Žáci mohou soutěžit mezi sebou nebo mezi ostatními uživateli v tabulce.

2. Jasná zpětná vazba:

- Uživatelé vidí svůj pokrok s ostatními, což je může motivovat k dalšímu zlepšování. Jsou to reálné výsledky ostatních, takže nejsou nastaveny nereálné cíle.

3. Jednoduchost:

- Design žebříčku je snadný a přehledný se všemi důležitými informacemi - každý uživatel okamžitě vidí svou pozici a soupeře.



RANK	USER	LEVEL	TIER	XP
1	 Anton	5	Beginner	2300 XP
2	 Zuzana	5	Beginner	2290 XP
3	 daniel	5	Beginner	2060 XP
4	 Ananas	5	Beginner	1310 XP
5	 Artem	4	Beginner	980 XP
6	 marek	4	Beginner	800 XP

Obrázek 23: Ukázka žebříčku s reálnými daty z testování

```
1 const leaderboardData = useResource$(async ({ track })
  => {
2   track(timeFilter);
3   track(limitCount);
4
5   try {
6     const usersRef = collection(db, "users");
7     let q;
8     const currentDate = new Date();
9     const monthStart = getStartOfMonth(currentDate);
10
11    if (timeFilter.value === 'monthly') {
12      q = query(
13        usersRef,
14        where('monthlyXP.${monthStart}', '>', 0),
15        orderBy('monthlyXP.${monthStart}', "desc"),
16        limit(limitCount.value)
17      );
18    } else {
19      q = query(
20        usersRef,
21        orderBy("experience", "desc"),
22        limit(limitCount.value)
23      );
24    }
25  }
26 });
```

Příklad 17: Získávání dat z databáze

3.4.8 Ocenění

Struktura achievementů

1. Hlavní statistiky:

- **Earned achievements:** Počet získaných achievementů.
- **Completion percentage:** Procento dokončených achievementů, které ukazuje celkový postup uživatele.

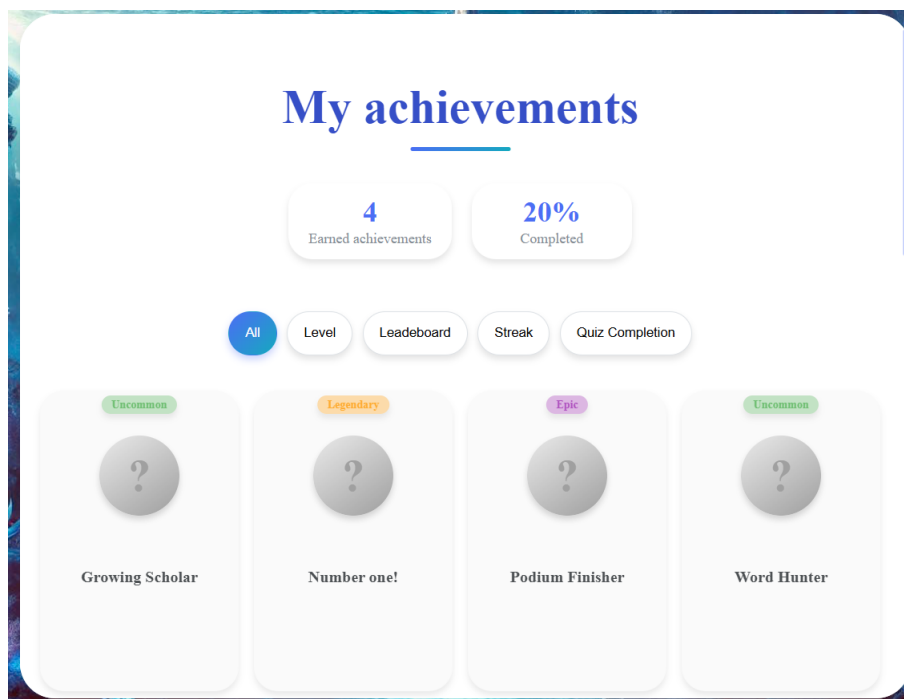
2. Kategorie achievementů: Uživatelé mohou filtrovat achievementy podle různých kategorií:

- *All:* Zobrazení všech achievementů.
- *Level:* Achievementy, které jsou spojené s úrovněmi (např. „Dosáhní úrovně 5“).
- *Leaderboard:* Achievementy spojené s žebříčkem (např. dosažení prvního místa v žebříčku).
- *Streak:* Tyto achievementy jsou udělovány za dosažení určitého počtu *Streaku* (např. Streak 20).
- *Quiz Completion:* Achievementy, které se získají za plnění cvičení. (např. „Splň 20 kvízů“).

3. Rarita:

- *Common:* Nejsnazší achievementy na získání (např. za první přihlášení, nebo splnění prvního kvízu). Jsou označovány šedou barvou a jsou ohodnoceny méně než 100 XP.
- *Uncommon:* Stále lehké achievementy, ale už těžší na získání než *Common*. Jsou označovány zelenou barvou a jsou ohodnoceny 100-200 XP.
- *Rare:* Středně těžké achievementy, nejsou ani snadné, ani těžké. Jsou označovány modrou barvou a jsou hodnoceny 200-300 XP.

- *Epic*: Složitější achievements, které vyžadují více práce a času. Jsou označované fialovou barvou a jsou ohodnoceny 300-500 XP.
- *Legendary*: Nejvzácnější a nejobtížnější achievements, které jsou prestižní (např. být první v tabulce), jsou označované zlatou barvou a jsou ohodnoceny více než 500 XP.



Obrázek 24: Ukázka achievementů

```
1 const achievementsData = achievementsSnapshot.docs.map(  
  doc => {  
2   const achievementData = doc.data();  
3   const processedData = processObject(achievementData);  
4  
5   const achievementId = processedData.uniqueId ||  
     processedData.id || doc.id;  
6   const unlocked = userAchievementIds.includes(  
     achievementId);  
7  
8   return {  
9     id: doc.id,  
10    uniqueId: achievementId,  
11    ...processedData,  
12    title: processedData.name,  
13    category: processedData.type || 'general',  
14    description: processedData.description || '',  
15    rarity: getRarityFromReward(processedData.reward),  
16    icon: getIconForType(processedData.type),  
17    unlocked: unlocked,  
18    isSecret: processedData.isSecret || false,  
19    secretDescription: processedData.secretDescription  
    || 'Tajny odznak ceká na odkryti!',  
20    requirement: processedData.description || 'Spln  
    pozadovany cil '  
21  };  
22 });
```

Příklad 18: Implementace odznaků

```
1 function getRarityFromReward(reward: number): string {
2     if (!reward) return 'common';
3     if (reward >= 500) return 'legendary';
4     if (reward >= 300) return 'epic';
5     if (reward >= 200) return 'rare';
6     if (reward >= 100) return 'uncommon';
7     return 'common';
8 }
9
10 function getIconForType(type: string): string {
11     switch(type) {
12         case 'level': return 'IKONA_LEVEL';
13         case 'streak': return 'IKONA_STREAK';
14         case 'practice': return 'IKONA_PRACTICE';
15         case 'vocabulary': return 'IKONA_VOCABULARY';
16         case 'perfect': return 'IKONA_PERFECT';
17         case 'social': return 'IKONA_SOCIAL';
18         case 'quiz': return 'IKONA_QUIZ';
19         case 'exercises': return 'IKONA_EXERCISES';
20         case 'secret': return 'IKONA_SECRET';
21         default: return 'IKONA_DEFAULT';
22     }
23 }
```

Příklad 19: Zkušenosti na základě rarity a typ odznaku

3.5 Testování aplikace

Testování aplikace proběhlo na druhém stupni základní školy v Jirkově, konkrétně na Městském gymnáziu a Základní škole Jirkov u paní učitelky Mgr. Barbory Šantorové. Cílem bylo ověřit funkčnost aplikace, její intuitivnost a celkový dojem za pomoci zpětné vazby ze strany žáků a paní učitelky.

Průběh testování Testování probíhalo v šesté a sedmé třídě, a to po dobu dvou hodin. Shodli jsme se s paní učitelkou, že na otestování aplikace bude tato doba dostačující. Žáci dostali adresu, kterou zadali do vyhledávače na svých zařízeních. Prvním krokem byla registrace, což zahrnovalo zadání e-mailové adresy. Tento krok byl jediným problémem, který nastal, protože minimálně polovina žáků nevěděla, jak vypadá správný tvar emailové adresy a jen dva z nich uměli napsat zavináč. Po krátké instruktáži, jak správně vypadá e-mailová adresa a jak napsat zavináč, se většina dětí přihlásila. Těm, kterým to nešlo jsme pomohli s paní učitelkou, aby mohli přejít k aplikaci.

Jakmile se žáci dostali do aplikace, už neměli žádné instrukce, jen ať objevují aplikaci. Jednoznačně nejoblíbenější aktivita byla *Letter hunt*, která je zaujala svým vzhledem a funkcemi. Nejvíce se totiž podobá hře. Během testování dostala také paní učitelka svůj učitelský přístup, kdy testovala funkce tříd a všechny aktivity jako žáci. Na konci hodiny byli všichni dotázáni na poskytnutí zpětné vazby.

Zpětná vazba od žáků Reakce od žáků byly pouze pozitivní a ukázaly, že aplikace dokáže zaujmout i motivovat:

- „Woow, to vypadá jako podvodní svět!“
- „Super stránka.“
- „Vypadá to hezky.“
- „Žebříček super!“
- „Dá se tam soutěžit!“

- „Zábavný.“
- „Nebyl jsem první, ale jinak super.“
- „Nic špatného nevím.“

Zpětná vazba od paní učitelky

Paní učitelka měla z aplikace velmi pozitivní dojmy. Ocenila zejména gamifikační prvky, přehlednost a jednoduchost aplikace. Při dotázání, zda by aplikace měla reálné využití ve výuce odpověděla, že určitě ano. Ona sama by jí prý určitě používala a ještě dodala, pokud v ní budu pokračovat, měl bych ji určitě zpoplatnit.

4 Závěr

Na základě testování aplikace na druhém stupni základní školy Městské gymnázium a Základní škola Jirkov lze konstatovat, že aplikace splnila cíle stanovené na začátku práce. Hlavním cílem bylo vytvořit a otestovat interaktivní webovou aplikaci pro výuku anglického jazyka, která by obsahovala gamifikační prvky, zvyšovala motivaci u žáků a podporovala jejich zapojení do výuky. Testování ukázalo, že tyto cíle byly úspěšně naplněny.

4.1 Naplnění cílů a přínos pro výuku

Aplikace Sea Quest byla navržena s jasným záměrem - vytvořit interaktivní nástroj, který by efektivně řešil klíčové problémy výuky, zejména nízkou motivaci žáků a nedostatečnou interaktivitu tradičních výukových metod. Testování jasně prokázalo, že tento cíl byl úspěšně naplněn. Žáci se během testovacích hodin aktivně zapojovali, což potvrzuje atraktivitu aplikace pro žáky.

Významným přínosem aplikace je její schopnost propojit normálně nudné vzdělávání s herními prvky, které jsou žákům blízké. Implementování žebříčku, úrovní, achievementů a vizuálně přívětivou aplikací ve stylu „podmořského světa“ výrazně přispívá k vytvoření poutavého vzdělávacího prostředí. Toto prostředí zvyšuje nejen zájem žáků o studium anglického jazyka, ale také prodlužuje dobu, kterou u učení žáci stráví.

4.2 Technologie a inovativnost řešení

Volba frameworku Qwik se pro vývoj aplikace ukázala jako velice vhodná vzhledem k požadavkům na rychlost a responsivitu. Během testování v reálném školním prostředí vynikaly zejména schopnost rychlého načítání a optimalizace výkonu i na zařízeních s pomalejším připojením (školní Wi-Fi nebyla velice kvalitní).

Z technologického hlediska tudíž vynikaly zejména tyto aspekty frameworku

Qwik:

- Implementace „resumability“, který umožňuje okamžitou interaktivitu bez nutnosti hydratace.
- Efektivní správa stavu aplikace prostřednictvím serializace na server.
- Využití progresivního načítání (lazy-loading) pro optimalizaci spotřeby dat.

Během testování žáci neměli problém s rychlostí reakcí aplikace (nezaznamenali žádné dlouhé načítání), což přispívalo k plynulosti procvičování.

4.3 Shrnutí testování

Testování na Městském gymnáziu a Základní škole Jirkov přineslo cenné poznatky o reálném fungování aplikace v cílovém prostředí. Během testování, se objevilo několik částí, kde by se aplikace dala vylepšit:

1. **Preference aktivit:** Žáci jednoznačně preferovali aktivitu *Letter hunt*, která nejvíce připomíná herní prostředí. Toto zjištění potvrzuje, že implementace herních mechanik do výuky zvyšuje míru zapojení žáků. Na druhou stranu, nejméně používaná aktivita byla *Vocabulary*, zejména část s Flashcards. I přes to, že si žáci nejdříve měli projít Flashcards a poté pokračovat na kvíz, tak většina žáků pokračovala ihned na kvíz a nevěnovala se přehledu slov.
2. **Sociální aspekty učení:** Komentáře žáků typu „Dá se tam soutěžit“ nebo „Nebyl jsem první, ale jinak super“ poukazují na důležitost sociální aspektu učení. Žebříčky a obecně možnost porovnávat výsledky se spolužáky vytvářejí pozitivní soutěživé prostředí, které žáky dokáže dále motivovat. Soutěživost by ale neměla být ve smyslu „Kdo získá více bodů z testu“, ale obecná „Kdo za daný měsíc bude v TOP 10“.

3. **Intuitivnost rozhraní:** Po překonání počátečních problémů s registrací (neznalost formátu e-mailové adresy a neznalost zkratky pro zavináč) se žáci bez dalších instrukcí a problémů pustili do objevování funkcí aplikace, což svědčí o tom, že aplikace je uživatelsky přívětivá.
4. **Responzivita aplikace:** Aplikace Sea QUest je navržena tak, aby byla plně responzivní a přizpůsobila se jakýmkoliv typům zařízení, včetně počítačů, tabletů a mobilních telefonů. Díky této vlastnosti je zajištěna pohodlnost uživatelům bez ohledu na velikost obrazovky nebo zařízení.
5. **Pozitivní přijetí učitelem:** Zpětná vazba od paní učitelky potvrdila potenciál aplikace pro reálné využití ve výuce, přičemž byla oceněna zejména přehlednost, jednoduchost a gamifikační prvky. Její komentář o možném zpoplatnění aplikace také naznačuje, že aplikace má potenciální komerční hodnotu.

4.4 Limit současné verze a potenciál dalšího rozvoje

Ačkoliv má aplikace úspěch u žáků, tak má současná verze aplikace Sea Quest určité limity, které by mohly být zlepšené v budoucích verzích:

1. **Rozšíření typů cvičení:** Aktuální stav aplikace nabízí tři typy aktivit (*Vocabulary*, *Letter hunt*, *Sentences*). Pro komplexnější rozvoj jazykových cvičení by bylo vhodné implementovat další typy cvičení zaměřené na poslech, mluvení a různé gramatické struktury.
2. **Adaptivní cvičení:** Implementace algoritmů nebo umělé inteligence pro adaptivní učení, které by automaticky přizpůsobovaly obtížnost cvičení podle výkonů žáka, by mohla dále zvýšit efektivitu a pedagogickou hodnotu aplikace.
3. **Rozšířené možnosti pro učitele:** Vylepšení analytických nástrojů, které by učitelům umožnili komplexněji organizovat výuku, detailněji sle-

dovat pokrok žáků a identifikovat problematické oblasti, by mohlo zvýšit efektivitu výuky a celkově pedagogickou hodnotu aplikace.

4.5 Porovnání frameworků

Frameworky Qwik, React, Vue.js a Next.js představují různé přístupy k vývoji webových aplikací. Každý má různé vlastnosti, výkon a vhodnosti pro různé typy projektů.

1. Výkon a rychlost načítání

- **Qwik:** Vyniká díky svému konceptu „resumability“, který eliminuje potřebu hydratace. To umožňuje okamžitou interaktivitu s minimálním množstvím JavaScriptu (pouze 1kB při prvním načtení). Díky progresivnímu načítání (lazy-loading) dosahuje Qwik nejlepšího času do interaktivity (TTI) ve srovnání s ostatními frameworky.
- **React:** Nabízí solidní výkon, ale jeho hydratace vyžaduje větší počáteční objem JavaScriptu (42 kB), což může zpomalit interaktivitu na pomalejších zařízeních.
- **Vue.js:** Podobně jako React používá hydrataci, ale díky menší velikosti počátečního stahování (33 kB) je o něco rychlejší při prvním načtení.
- **Next.js:** Přináší výhody server-side renderingu (SSR) a progresivní hydratace, avšak při úplném obnovení stránky na serveru může být pomalejší než Qwik.

2. Složitost učení:

- **Qwik:** Zatím malá komunita a omezený počet knihoven třetích stran mohou zpomalit vývoj komplexnějších aplikací. Vývojáři často musí vymýšlet vlastní řešení.

- **React:** Díky rozsáhlé komunitě a bohaté dokumentaci je nejoblíbenější mezi vývojáři. Poskytuje flexibilitu, ale může být složitější na naučení.
- **Vue.js:** Snadno se učí díky intuitivnímu syntaxu a kvalitní dokumentaci.
- **Next.js:** Nabízí robustní nástroje pro vývojáře, ale vyžaduje rozhodování mezi serverovými a klientskými komponentami.

Finální volba: Framework Qwik se ukázal jako ideální volba pro vzdělávací aplikace Sea Quest díky:

1. **Okamžité interaktivitě** při procvičování slovíček.
2. **Efektivní správě paměti:** Průměrná spotřeba RAM o 40 % nižší než u Reactu.
3. **Kompatibilitě se školními prostředky** - bezproblémový provoz na starších školních zařízeních.

4.6 Závěrečné zhodnocení

Aplikace Sea Quest představuje vhodný nástroj, který má potenciál pro začlenění do výuky. Kombinuje pedagogické principy s moderními technologickými možnostmi a gamifikačními prvky, čímž vytváří efektivní vzdělávací nástroj pro žáky.

Úspěšné testování v reálném školním prostředí potvrdilo nejen funkčnost a atraktivitu aplikace, ale také potenciál reálného využití ve výuce. Framework Qwik se v tomto kontextu ukázal jako vhodná technologická volba, která podporuje rychlost a responzivitu.

V širším kontextu tato bakalářská práce ukazuje, jak mohou moderní webové technologie a gamifikační přístupy proměnit tradiční vzdělávací metody a přispět k vytváření poutavějšího a efektivnější vzdělávacího prostředí pro žáky.

Seznam přečtené literatury a zdrojů

- [1] MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY ČR. *Na dálku: Metodická podpora pro učitele* [online]. Bez data [cit. 2025-03-26]. Dostupné z: <https://nadalku.msmt.cz/cs/rozvoj-pedagogu/nastroje-pro-vzdelavani>
- [2] HUDECOVÁ, Darina. 5 aplikací pro zpestření výuky [online]. VMS, 2022 [cit. 2025-03-26]. Dostupné z: <https://www.vms.cz/blog/5-aplikaci-pro-zpestreni-vyuky>
- [3] HOLEC, Jakub. *Zkušenosti s využitím Nearpodu v online výuce* [online]. Národní pedagogický institut České republiky, 2023 [cit. 2025-04-07]. Dostupné z: <https://clanky.rvp.cz/clanek/c/Z/22920/zkusenosti-s-vyuzitim-nearpodu-v-online-vyuce.html>
- [4] UNIVERSITY OF WATERLOO. *Gamification and Game-Based Learning* [online]. 2025 [cit. 2025-03-28]. Dostupné z: <https://uwaterloo.ca/centre-for-teaching-excellence/catalogs/tip-sheets/gamification-and-game-based-learning>
- [5] ANONYMOUS. *Gamification in Education* [online]. 2025 [cit. 2025-03-28]. Dostupné z: <https://geniusee.com/single-blog/gamification-in-education>
- [6] STAND TOGETHER. *How AI and personalized learning will redesign education and reengage students* [online]. Bez data [cit. 2025-04-07]. Dostupné z: <https://standtogether.org/stories/education/how-ai-personalized-learning-will-redesign-education-reengage>
- [7] BULGER, Monica. *Personalized Learning: The Conversations We're Not Having* [online]. Data & Society Research Institute, 2016 [cit. 2025-04-07]. Dostupné z: https://www.datasociety.net/pubs/ecl/PersonalizedLearning_primer_2016.pdf

-
- [8] COSTA, Maraisa de Cássia da; OLINDA, Ana Lucia Sgrancio; SANTOS, Alex Pedrucci dos. *Digital technologies in education: Challenges and opportunities for teaching and learning* [online]. Proceedings of the 7th International Conference on Education and E-Learning (ICEL), 2023 [cit. 2025-03-27]. Dostupné z: <https://sevenpublicacoes.com.br/anais7/article/download/4823/9027/19970>
- [9] MEGUMETHOD. *M-learning: Efektivní a rychlá forma vzdělávání, nejen v době pandemie* [online]. Bez data [cit. 2025-04-07]. Dostupné z: <https://www.megumethod.com/blog/m-learning>
- [10] SPEECHIFY. *Mikroučení: Revoluce ve vzdělávání v malých krocích* [online]. 2023 [cit. 2025-04-07]. Dostupné z: <https://speechify.com/cs/blog/microlearning-revolutionizing-learning-in-bite-sized-steps/>
- [11] JOHNSON, Amy M.; JACOVINA, Matthew E.; RUSSELL, Devin G.; SOTO, Christian M. *Challenges and solutions when using technologies in the classroom* [online]. Arizona State University, 2016 [cit. 2025-03-28]. Dostupné z: <https://files.eric.ed.gov/fulltext/ED577147.pdf>
- [12] ANONYMOUS. *Digital Technology* [online]. 2025 [cit. 2025-03-28]. Dostupné z: <https://www.leadertech.ch/resources/digital-technology>
- [13] ANONYMOUS. *The Challenges Associated With Technology in Schools - Smart Strategies for Successful Living* [online]. 2025 [cit. 2025-03-28]. Dostupné z: <https://www.agegracefullyamerica.com/the-challenges-associated-with-technology-in-schools/>
- [14] ANONYMOUS. *How can we solve education's technology challenges?* [online]. 2025 [cit. 2025-03-28]. Dostupné z: <https://edsembli.com/how-can-we-solve-educations-technology-challenges/>

- [15] ANONYMOUS. *Barriers of ICT in Education* [online]. 2025 [cit. 2025-03-28]. Dostupné z: <https://www.hmhco.com/blog/barriers-of-ict-in-education>
- [16] ANONYMOUS. *Qwik: Revoluční framework pro moderní web* [online]. 2025 [cit. 2025-03-27]. Dostupné z: <https://mydreams.cz/cz/wiki/4901-qwik-revolucni-framework-pro-moderni-web.html>
- [17] GLASER, Petr. *Podcast o javascriptovém frameworku Qwik* [online]. 2025 [cit. 2025-03-27]. Dostupné z: <https://www.vzhurudolu.cz/podcast/229-podcast-qwik>
- [18] ANONYMOUS. *Qwik Documentation* [online]. 2025 [cit. 2025-03-27]. Dostupné z: <https://qwik.dev/docs/>
- [19] QWIK. *Qwik framework* [online]. Bez data [cit. 2025-03-26]. Dostupné z: <https://qwik.dev>
- [20] LIOU, Daw-Chih. Qwik vs Next.js: Speed Comparison. *HackerNoon* [online]. 2023-01-28 [cit. 2025-03-29]. Dostupné z: <https://hackernoon.com/qwik-vs-nextjs-speed-comparison>
- [21] SMILE. What Frontend Framework is Fastest? *SMILE Europe* [online]. 2023 [cit. 2025-03-30]. Dostupné z: <https://smile.eu/en/publications-and-events/what-frontend-framework-fastest>
- [22] OUTSHIFT. *Qwik vs. Next.js: Which framework is right for your next web project?* [online]. Bez data [cit. 2025-03-29]. Dostupné z: <https://outshift.cisco.com/blog/qwik-vs-nextjs>
- [23] ROADMAP.SH. *Top 7 Frontend Frameworks to Use in 2025: Pro Advice* [online]. Bez data [cit. 2025-04-11]. Dostupné z: <https://roadmap.sh/frontend/frameworks>

-
- [24] HECKS, Eleanor. Is Qwik the Future of Scalable Web Development? *Frontend Dogma* [online]. 2024 [cit. 2025-03-30]. Dostupné z: <https://frontenddogma.com/posts/2024/is-qwik-the-future-of-scalable-web-development/>
- [25] LOGROCKET. Qwik adoption guide: Overview, examples, and alternatives. *LogRocket Blog* [online]. Bez data [cit. 2025-03-29]. Dostupné z: <https://blog.logrocket.com/qwik-adoption-guide/>
- [26] DAVID, Ekanem. Build a Quiz Application with Qwik Framework. *Bejamas Hub* [online]. 2024-11-18 [cit. 2025-03-30]. Dostupné z: <https://bejamas.com/hub/tutorials/build-a-quiz-application-with-qwik-framework>
- [27] PRAŽSKÁ ŠKOLA. *Top 10 bezplatných vzdělávacích aplikací a webů pro efektivní učení* [online]. Bez data [cit. 2025-03-27]. Dostupné z: <https://www.prahaskolska.eu/top-10-bezplatnych-vzdelavacich-aplikaci-a-webu-pro-efektivni-uceni/>
- [28] ANONYMOUS. *Qwik vs Svelte: A Technical Comparison for Developers* [online]. 2025 [cit. 2025-03-27]. Dostupné z: <https://caisy.io/blog/qwik-vs-svelte>
- [29] ANONYMOUS. *Qwik UI* [online]. 2025 [cit. 2025-03-27]. Dostupné z: <https://qwikui.com>
- [30] EUROPEAN COMMISSION. *Gamification in Education* [online]. European Union, 2023 [cit. 2025-03-28]. Dostupné z: https://ec.europa.eu/programmes/erasmus-plus/project-result-content/63bd8c6a-86c0-4a75-9196-7ac990770101/Gamification_book%20Gamest.pdf

- [31] KABISA. *A Qwik Look Into Next Gen* [online]. 2025 [cit. 2025-03-29]. Dostupné z: <https://archive.kabisa.nl/tech/a-qwik-look-into-next-gen/>
- [32] UNESCO. *Global Education Monitoring Report - Technology in Education* [online]. 2023 [cit. 2025-03-29]. Dostupné z: <https://gem-report-2023.unesco.org/technology-in-education/>
- [33] LINGIO. *Gamification in Education* [online]. 2023 [cit. 2025-03-29]. Dostupné z: <https://www.lingio.com/blog/gamification-in-education>
- [34] STACKADEMIC. Why Qwik Has the Potential to Be Next React in Web Development. *Medium* [online]. Bez data [cit. 2025-03-30]. Dostupné z: <https://blog.stackademic.com/why-qwik-has-the-potential-to-be-next-react-in-web-development-7eb71a949209>
- [35] THIS IS LEARNING. Qwik: The post-modern framework. *dev.to* [online]. Bez data [cit. 2025-03-30]. Dostupné z: <https://dev.to/this-is-learning/qwik-the-post-modern-framework-3c50>
- [36] INFOQ. Qwik JavaScript Framework Aims to Deliver Instant-On Performance. *infoq.com* [online]. 2021 [cit. 2025-03-30]. Dostupné z: <https://www.infoq.com/news/2021/09/qwik-javascript-framework/>
- [37] GAMBLE, Jonathan. Qwik Framework: King. *Telerik Blogs* [online]. Bez data [cit. 2025-03-30]. Dostupné z: <https://www.telerik.com/blogs/qwik-framework-king>

Seznam obrázků

1	Přihlašovací formulář	25
2	Přihlašovací formulář pro rodiče	27
3	Statistiky žáka po přihlášení rodiče	28
4	Správa tříd	31
5	Formulář pro vytvoření třídy	31
6	Dashboard třídy	33
7	Příklad oznámení	36
8	Příklad úkolu z pohledu učitele	37
9	Příklad úkolu z pohledu studenta	37
10	Příklad testu	39
11	Průběh testu	39
12	Výběr cvičení	40
13	Výběr úrovně	41
14	Flashcards	42
15	Aktivita Vocabulary	43
16	Výsledky kvízu	44
17	Aktivita Letter hunt	46
18	Výsledek aktivity Letter hunt	47
19	Aktivita Sentences	50
20	Výsledek aktivita Sentences	51
21	Ukázka profilu	53
22	Výběr profilového obrázku	54
23	Ukázka žebříčku s reálnými daty z testování	61
24	Ukázka achievementů	64

Seznam zdrojových kódů

1	Příklad Qwik komponentu s lazy-loadingem	17
2	Příklad QRL	19
3	Kód pro přihlašování a registraci	26
4	Kód pro rodičovské přihlášení	28
5	Statistiky žáka z pohledu rodiče	29
6	Kód pro připojení do třídy	32
7	Kód pro získání dat levelu z Firestore	45
8	Kontrola hádaného slova	48
9	Kontrola zda slovo opravdu existuje	49
10	Kód pro rozložení věty	51
11	Ukázka části kódu pro přetahování slov	52
12	Kód pro odhlašování	54
13	Kód pro výběr avataru	55
14	Úryvek kódu pro přidávání zkušeností	57
15	Funkce pro získání zbývajících zkušeností do další úrovně	58
16	Příklad kalkulace zkušeností z aktivity <i>Letter hunt</i>	59
17	Získávání dat z databáze	62
18	Implementace odznaků	65
19	Zkušenosti na základě rarity a typ odznaku	66

A Příloha

 GitHub Repository se zdrojovým kódem