

DELTA – Střední škola informatiky a ekonomie, Základní škola a Mateřská škola s.r.o.

Ke Kamenci 151, PARDUBICE

MATURITNÍ PROJEKT

DYNAMICKÝ WEB S HLAVOLAMY

Příjmení, jméno: Modráček Václav

Třída: 4.B

Studijní obor: Informační technologie 18-20-M/01

Školní rok: 2022/2023

Konzultant: Mgr. Jan Mottl

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Václav Modráček*
Pro školní rok: *2022/2023*
Třída: *4. B*
Obor: *Informační technologie 18-20-M/01*

Téma práce: *Dynamický web s hlavolamy*
Vedoucí práce: *Mgr. Jan Mottl*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem projektu bude vytvořit dynamický web s hlavolamy, databází a přihlašovacím systémem. Na webu se nejdříve uživatel zaregistruje a potom bude moci řešit různé hlavolamy a za každý splněný hlavolam dostane odznak. Zároveň se mu bude případně ukládat postup, který na jednotlivých hlavolamech dosáhl, aby se mohl později k dokončení hlavolamu vrátit. Dále bude přítomna správa uživatelských účtů v roli admina. Technologie bude v základu HTML, CSS, Javascript, PHP s tím, že se počítá s využitím frameworku, který bude teprve určen.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

Září – Rešerše možných řešení zpracování a podobných již fungujících webů

Říjen – Návrh a tvorba základního webu, bez dynamických prvků.

Listopad – Tvorba systému přihlašování, propojení s databází.

Prosinec – Vytváření samotných hlavolamů v prostředí webu.

Leden – Tvorba grafické reprezentace hlavolamů.

Únor – Systém získávání odznaků za vyřešené hlavolamy a ukládání postupu při řešení hlavolamů. Testování webu.

Březen – Tvorba dokumentace projektu.

Prohlášení

Prohlašuji, že jsem maturitní projekt vypracoval(a) samostatně, výhradně s použitím uvedené literatury.

V Pardubicích dne 31.3 2023

(vlastnoruční podpis)

Poděkování

Chtěl bych poděkovat Mgr. Janu Mottlovi za odborné vedení maturitního projektu a konzultování při vývoji aplikace a kontrolu dokumentace.

Anotace

Dokumentace mapuje tvorbu aplikace s hlavolamy ve webovém prostředí. Práce se zabývá technologiemi, které byly použité pro vytvoření této aplikace. Popisuje také, jak jednotlivé části aplikace fungují a hlouběji popisuje jednotlivé hlavolamy a jejich řešení v rámci kódu.

Klíčová slova

Hlavolamy, React, Firebase, Typescript, Logické hry

Annotation

The documentation covers the creation of an application with puzzles in web environment. The project deals with the technology used for creation of this application. It describes how different parts of the application works and in more depth describes how each puzzle works and its solution in code.

Key words

Puzzles, React, Firebase, Typescript, Logical game

Obsah

1. Úvod.....	7
1.1. Cílová skupina.....	7
2. Technologie.....	8
2.1. Node.js.....	8
2.2. Next.js.....	8
2.3. React.....	8
2.4. Firebase.....	9
2.4.1. Cloud Firestore.....	9
2.4.2. Firebase Authentication.....	9
2.5. Typescript.....	10
2.6. Chakra-UI.....	10
3. Architektura.....	11
3.1. Backend.....	11
3.2. Firestore databáze.....	11
3.3. Správa uživatelů.....	12
3.4. Dynamické prvky.....	13
4. Webová aplikace.....	14
4.1. Přihlášení a registrace.....	14
4.2. Domovská stránka.....	15
4.3. Profil.....	16
4.4. Achievements.....	16
5. Hlavalamy.....	17
5.1. Hlavalam lightsaber.....	17
5.2. Hlavalam vlk, koza a zelí.....	17
5.3. Pentomino hlavalam.....	18
5.4. Eternity hlavalam.....	19
6. Budoucnost webu.....	21
6.1. Co v aplikaci chybí?.....	21
6.2. Plán do budoucna.....	21
7. Závěr.....	22
8. Seznam použité literatury.....	23
9. Seznam obrázků.....	25

1. Úvod

Na internetu lze nalézt webové stránky, kde můžeme řešit různorodé hlavolamy. Většinou tyto stránky, ale obsahují příliš zjednodušené verze hlavolamů, které jsou nezábavné. Webové stránky se složitými hlavolamy také existují. Většina jich bývá zastaralých a málo graficky rozpracovaných. Toto jsou důvody, které mě vedli k vytvoření vlastní alternativy.

Cílem mého projektu bylo vytvořit dynamický web, který hlavolamy vizualizuje a bude uživatelsky přívětivý. V první řadě zde uživatel najde různé hlavolamy od těch nejjednodušších pro nenáročné hráče, až po složitější hlavolamy pro hráče, kteří mají rádi výzvu. Jako motivací k dokončení hlavolamu slouží systém odznaků, které hráč získává za každý dokončený hlavolam. To vede k větší motivaci vyřešit hlavolam a jít řešit další hlavolamy.

1.1. Cílová skupina

Cílovou skupinou tohoto webu jsou lidé od 15 let výše.

Jedním typem tak mohou být nováčci, kteří se o hlavolamy teprve začínají zajímat a hledají web, kde budou moci začít na jednoduchých hlavolamech a postupně přejít na ty složitější.

Druhý typ jsou hráči zkušení, co hledají složitější hlavolamy a nebojí se výzev.

2. Technologie

Tato část dokumentace se věnuje technologiím, které jsou v aplikaci použity. Webová aplikace je napsaná ve frameworku Next.js, který je postavený na prostředí Node.js. Backend řeší služba Firebase.

2.1. Node.js

Node.js je vývojové prostředí vytvořené pro jazyk javascript. Je postavený na javascript enginu V8, což je open-source engine vyvinutý společností google. Používá se pro psaní vysoce škálovatelných internetových aplikací. Node.js využívá asynchronní I/O a událostmi řízený model programování, což umožňuje zpracování velkého množství požadavků na serveru bez blokování procesu. Na Node.js běží celá webová aplikace.[1]

2.2. Next.js

Next.js je framework, který je postavený na vývojovém prostředí Node.js. Tento framework rozšiřuje známou Javascriptovou knihovnu React a proto nabízí mnoho užitečných funkcí a vlastností, jako je například serverové renderování, automatický optimalizace, routing a další. Next.js obsahuje také server side rendering. To znamená, že se HTML kód vygeneruje na straně serveru a poté ho odešle uživateli do prohlížeče.

Velkou výhodou Next.js je, že používá jen ten kód, který zrovna potřebuje. Framework rozpozná, který kód je potřeba na aktuální stránce a načte pouze ten. Proto je načítání webových stránek rychlejší.

Next.js také obsahuje Next link. Pomocí linku můžeme navigovat mezi adresáři na webu. Neboli můžeme díky němu přesměrovávat mezi stránkami. Tuto funkci využívám například v menu, kde pomocí linku přesměrovávám uživatele na jinou část webové aplikace. [2][3]

2.3. React

React je Javascriptová knihovna pro tvorbu uživatelského rozhraní, která umožňuje vývojářům snadno vytvářet interaktivní a dynamické webové aplikace. React je založen na konceptu komponent, které umožňují kód rozdělit a zpřehlednit. Tyto komponenty poté můžeme opětovně používat dle libosti. To vývojáři značně ulehčí práci, protože nemusí danou komponentu psát znovu.

React využívá virtuální DOM (Document Object Model), což je abstraktní reprezentace HTML stránky. Virtuální DOM zajišťuje rychlé a efektivní aktualizace stránky, protože React aktualizuje pouze ty části stránky, které se změnilly, a ne celou stránku. Například u hlavolamu Eternity, pokud v průběhu řešení hlavolamu kliknete na tlačítko restart, tak se v hracím poli aktualizují pouze ty dílky, které byly obsazeny. [4]

2.4. Firebase

Firebase je platforma poskytovaná společností Google pro vývoj mobilních a webových aplikací. Firebase nabízí širokou škálu služeb a nástrojů, které usnadňují vývojářům tvorbu aplikací a usnadňují jejich provozování. Je to serverless řešení backendu, tím pádem vývojář nemusí řešit ostatní věci spojené s backendem a usnadní si tak práci. Firebase obsahuje několik užitečných služeb jako jsou např.: Cloud Firestore, Firebase Authentication. Firebase používám jako databázi pro tuto webovou aplikaci. [5]

2.4.1. Cloud Firestore

Firestore je dokumentová NOSQL databáze, která je škálovatelná. Struktura databáze se skládá z kolekcí, ve kterých jsou dokumenty. Každá kolekce má svoje ID a může obsahovat mnoho dokumentů. Dokument má také svoje ID. Pomocí těchto ID můžeme buď do kolekcí přidávat nebo mazat dokumenty. Do každého dokumentu za pomoci jeho ID můžeme rovněž ukládat, mazat, nebo měnit data. Dokument obsahuje data ve formě JSON. Cloud Firestore rovněž nabízí ukládání dat v určitých datových typech jako např.: string, number, map atd.

Data z Firestore lze získat pomocí Firebase SDK, který se implementuje přímo do naší aplikace. To využíváme při načítání obsahu specifického pro uživatele. Například na profilové stránce uživatele, zde chceme uživateli předat informaci jaký je jeho aktuální email. Pošleme tedy dotaz na databázi, aby nám vrátila informace o právě přihlášeném uživateli. Zde se dozvíme, jaký email uživatel má a zobrazíme ho. [6]

2.4.2. Firebase Authentication

Firebase Authentication zprostředkovává registraci a přihlášení uživatelů. Tyto akce mohou probíhat přes poskytovatele, jako jsou Facebook, Google apod. Můžeme, ale také využít registraci pomocí emailu a hesla. Firebase Authentication také umožňuje ověření emailu, změnu hesla, změnu emailu. Firebase Authentication používáme například při přihlášení uživatele. Uživatel vyplní přihlašovací formulář, my pošleme na server, jestli

uživatel zadal správné a shodující údaje pro přihlášení a Firebase Authentication ověří, jestli uživatel zadal přihlašovací údaje správně.[7]

2.5. Typescript

Typescript je programovací jazyk, který je vyvinut firmou Microsoft. Jedná se o nadstavbu jazyka Javascript, to znamená, že přidává další funkcionality a možnosti k původnímu Javascriptu.

TypeScript přidává statický typový systém, který umožňuje vývojářům definovat typy pro proměnné, funkce a další prvky v kódu. Tímto způsobem TypeScript pomáhá zachytit chyby v kódu již během vývoje, což může ušetřit mnoho času a úsilí při testování a ladění aplikace.

Další výhodou TypeScriptu je, že podporuje moderní funkcionality jazyka JavaScript, jako jsou například třídy, moduly, arrow funkce a další. TypeScript také poskytuje možnosti pro rozhraní a dědičnost, což umožňuje vývojářům psát čistý a strukturovaný kód.

Typescript používám jako hlavní nástroj pro vytváření mé webové aplikace a její funkcionality. [8][9]

2.6. Chakra-UI

ChakraUI je open-source knihovna uživatelského rozhraní (UI) pro React, která umožňuje vývojářům rychle vytvářet moderní, responzivní a atraktivní uživatelská rozhraní pro webové aplikace. Je postavena na knihovně stylů CSS-in-JS, která umožňuje vývojářům psát styly pro komponenty v Reactu pomocí JavaScriptu.

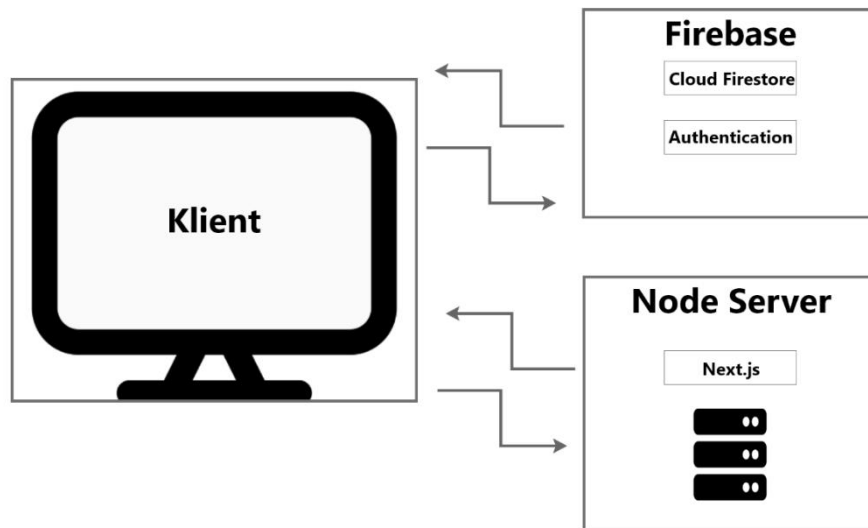
Chakra-UI nabízí mnoho předdefinovaných komponent, jako jsou tlačítka, formuláře, tabulky, modální okna, ikony a mnoho dalšího. Tyto komponenty jsou snadno přizpůsobitelné. Chakra-UI také poskytuje flexibilní systém mřížky, který umožňuje vývojářům rychle a snadno vytvářet responzivní uživatelská rozhraní.

Chakra-UI je navržena tak, aby byla jednoduchá na použití a přizpůsobení. Vývojáři mohou snadno přidat své vlastní styly a rozšířit funkcionality knihovny podle svých potřeb. Chakra-UI také podporuje tematizaci, což umožňuje vývojářům snadno měnit vzhled aplikace pomocí předem definovaných témat nebo vlastních vlastností.

Chakra-UI používám pro veškeré stylování webu. [10]

3. Architektura

Klient přímo komunikuje s databází Firebase, která slouží jako backend aplikace. Na Node serveru běží samotná Next.js aplikace, která se zobrazuje u klienta. Tak jak je vidět na ilustraci.



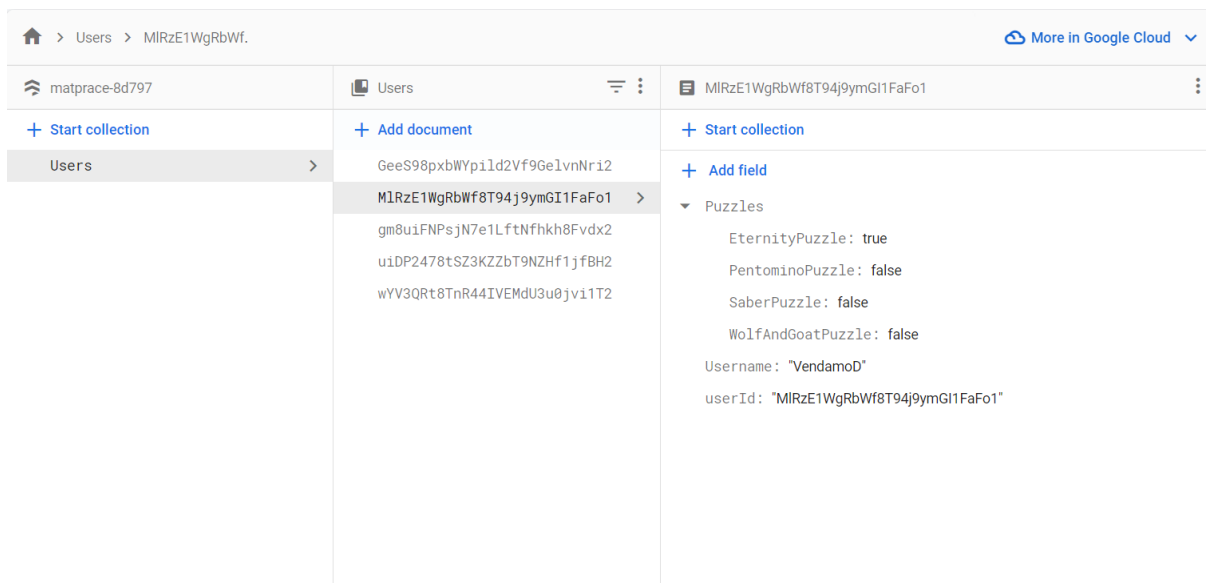
Ilustrace 1 – Architektura aplikace [zdroj: autor]

3.1. Backend

Roli backendu v této aplikaci plní Firebase. Ta je propojena s klientem pomocí webové SDK, kterou Firebase poskytuje. Komunikace tak probíhá přímo od klienta.

3.2. Firestore databáze

Databází pro tuto webovou aplikaci je Firestore. V ní jsou ukládány veškeré potřebné informace. Databáze se dělí na kolekce a v každé kolekci nalezneme dokumenty. V kolekci také můžeme mít uložený objekt, který obsahuje více dokumentů.

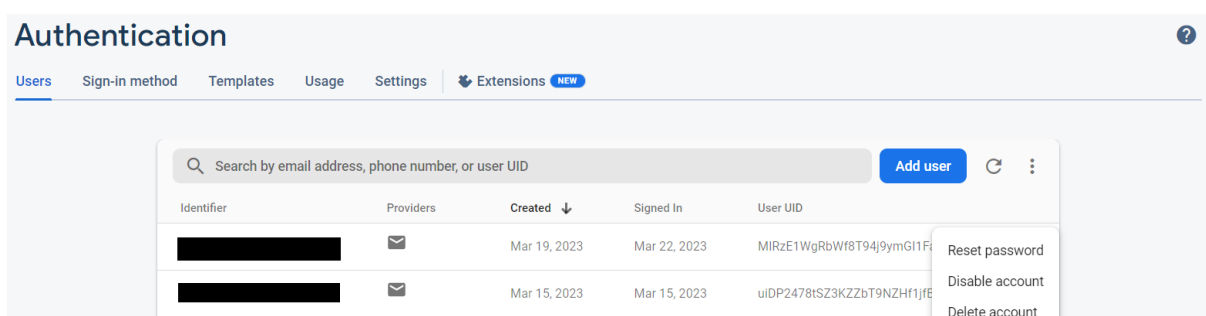


Ilustrace 2 – Firestore databáze [zdroj: autor]

Jak můžete vidět výše na ilustraci, tak ke každému uživateli ukládáme stav jeho hlavolamů, jestli je dokončil nebo ne. Dále ukládáme jeho userID a také jeho přezdívku.

3.3. Správa uživatelů

Správa uživatelů je řešena přes Firebase. Ta poskytuje službu Firebase Authentication, která za nás řeší přihlašování a registrování uživatelů. Firebase Authentication nabízí různé providery, takže si můžeme vybrat, jestli chceme registrovat a přihlašovat uživatele pomocí google účtu, emailu s heslem nebo třeba pomocí facebooku či microsoft účtu. V našem případě používáme pouze email a heslo. Po registraci se uživatel uloží do složky Users v kartě Authentication. Zde také nalezneme informace o uživateli a můžeme zde také spravovat uživatele, jako mu například smazat účet, změnit heslo nebo dočasně pozastavit účet.



Ilustrace 3 – Firebase Authentication [zdroj: autor]

Ve složce users, ale nemůžeme uživateli přidávat ani odebírat žádné další informace. Proto máme vytvořenou vlastní složku users ve Firestore databázi, jak je už popsané v sekci 3.2.

3.4. Dynamické prvky

Dynamické prvky v této webové aplikaci řešíme pomocí dotazů na databázi a následných odpovědí. Například pošleme dotaz, jaké hlavolamy má uživatel splněné a podle jejich hodnot true nebo false zobrazíme ty, které jsou mají hodnotu true. Tímhle způsobem postupujeme i na domovské stránce při zobrazování, jestli uživatel dokončil hlavolam nebo ne.

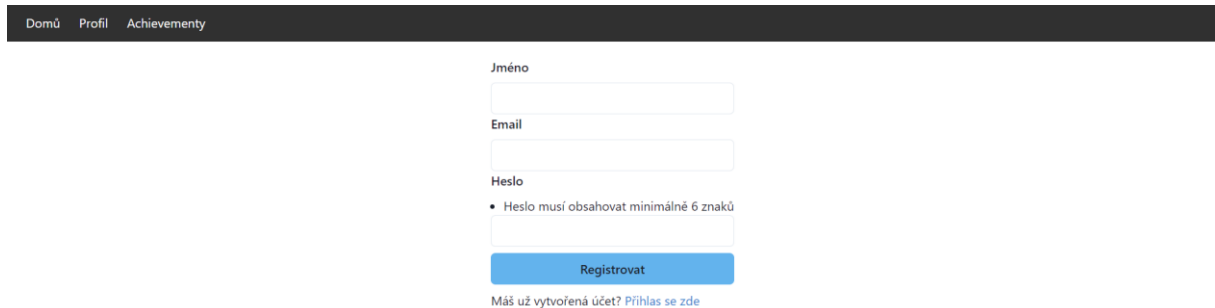
Další možnost řešení je používat takzvané dynamické URL To by znamenalo, že by se URL adresa webu měnila podle uživatele a každému uživateli by se zobrazoval jiný obsah na stránce.[13]

4. Webová aplikace

Tato část dokumentace popisuje, jak funguje celá webová aplikace. Sekce zahrnuje vše od registrace s přihlášením až po jednotlivé hlavolamy.

4.1. Přihlášení a registrace

Přihlášení a registraci řešíme pomocí Firebase authentication. Pro úspěšnou registraci je potřeba vyplnit validně všechny potřebné údaje, které můžete vidět na ilustraci níže.



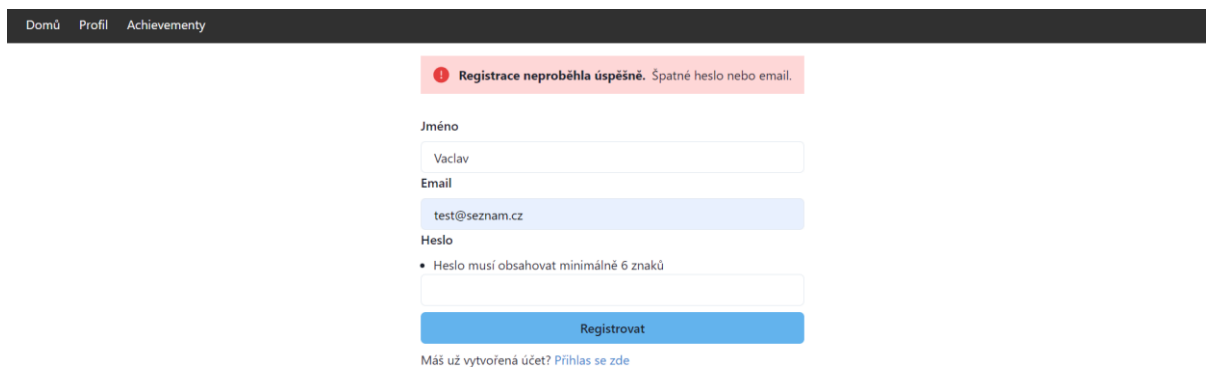
The image shows a registration form with a dark header containing the links 'Domů', 'Profil', and 'Achievements'. The form itself is centered and contains three input fields: 'Jméno' (Name), 'Email', and 'Heslo' (Password). Below the password field, there is a validation message: '• Heslo musí obsahovat minimálně 6 znaků'. At the bottom of the form is a blue button labeled 'Registrovat'. Below the button, there is a link: 'Máš už vytvořená účet? Přihlas se zde'.

Ilustrace 4 – registrační formulář [zdroj: autor]

Poté co uživatel vyplní registrační údaje a stiskne tlačítko registrovat se spustí funkce ve které voláme funkci register. Té předáme parametry email, password a username.

Funkci register máme vytvořenou v souboru userContext.tsx. V tomto souboru máme napsané všechny funkce pro komunikaci a práci s databází. Tato funkce používá už předdefinovanou funkci od Firebase authentication createUserWithEmailAndPassword a poté pomocí už naše funkce saveUser uloží uživatele do kolekce Users ve Firestore, jak už jsem zmínil v sekci 3.2.

Po úspěšné registraci se nám ukáže upozornění, že registrace proběhla úspěšně a překlikneme na přihlašovací stránku. Na přihlašovací stránce poté vyplníme náš email a heslo a přihlásíme se. Pokud se přihlášení nepovede, tak uživateli ukážeme upozornění, že se registrace nepovedla.



Ilustrace 5 – nepovedená registrace [zdroj: autor]

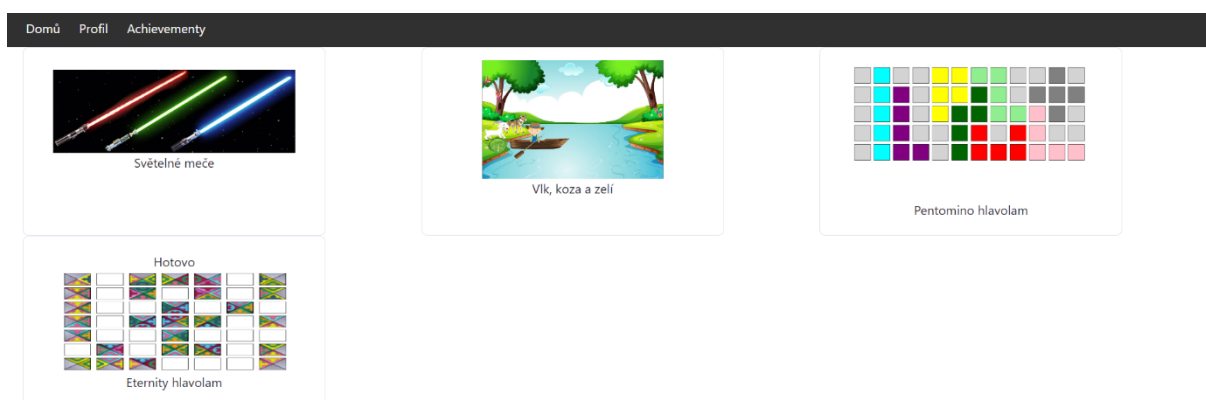
Po stisknutí tlačítka přihlásit se spustí funkce ve které zavoláme funkci login, které předáme parametry email a heslo.

Funkci login máme opět vytvořenou v useContextu odkud jsme si ji naimportovali. Funkci login předáme parametry email a heslo. Ve funkci používáme funkci signInWithEmailAndPassword of Firebase authentication a poté zavoláme další dvě námi vytvořené funkce pro získání dat o uživateli.

Alternativní řešení je nahradit Firebase třeba MongoDB databází. MongoDB také obsahuje vlastní autentifikační systém, který můžeme použít. [11][12]

4.2. Domovská stránka

Na domovské stránce nalezneme všechny hlavolamy, na které můžeme kliknout a přesunout se na stránku se samotným hlavolamem. V horní části najdeme taky navigační menu, kde nalezneme odkazy na domovskou stránku, profil a achievements. Pokud má hráč některý hlavolam už vyřešený, uvidí u hlavolamu hotovo.

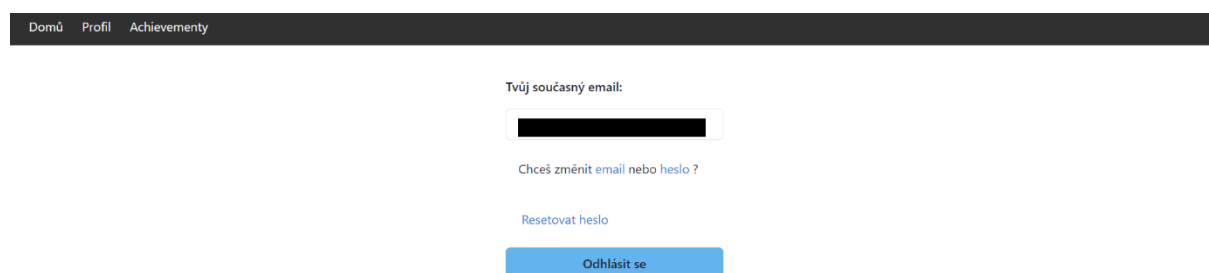


Ilustrace 6 – domovská stránka [zdroj: autor]

Při načtení stránky si pomocí useEffectu zavoláme funkci, pomocí které dostaneme data uživatele, podle kterých potom zobrazíme stav dokončení hlavolamu.

4.3. Profil

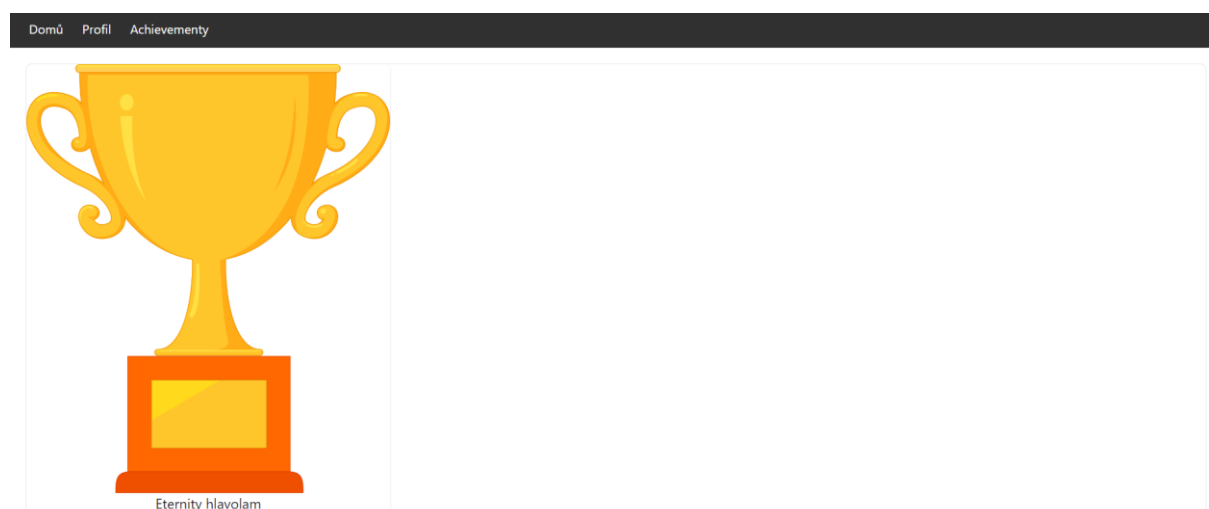
Na profilové stránce najde uživatel svůj současný email a také možnosti si změnit email anebo heslo a také si heslo resetovat za pomocí emailu. Kliknutím na jednotlivé odkazy se uživatel přesměruje na stránku buď se změnou emailu nebo hesla případně na stránku pro resetování hesla.



Ilustrace 7 – profil uživatele [zdroj: autor]

4.4. Achievements

Na stránce achievements najdeme všechny achievements, které jsme doposud získali. Uživatel samozřejmě vidí pouze ty achievements, které získal. U každého achievementu je jméno hlavolamu, ke kterému achievement patří. Pokud má splněné pouze 1 achievement, tak uvidí pouze 1.



Ilustrace 8 – achievements [zdroj: autor]

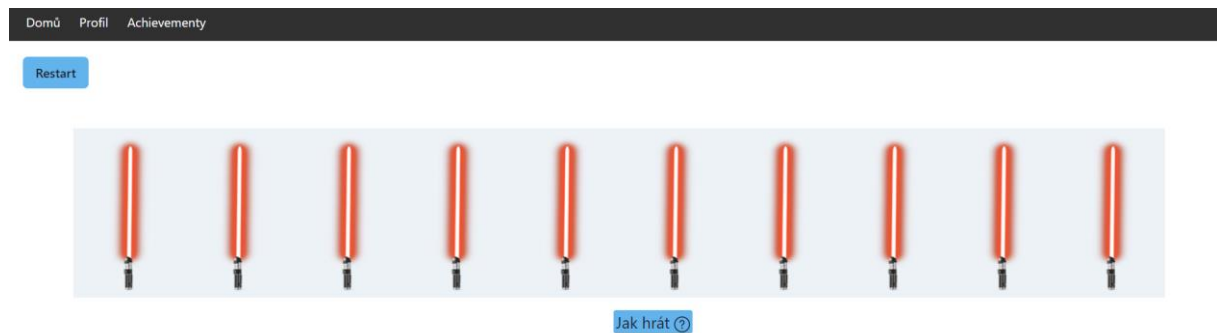
5. Hlavalamy

V této části dokumentace se věnuji popisu hlavalamů a tomu, jak fungují. Dále zde také popisují, jak jsem hlavalamy řešil v kódu.

5.1. Hlavalam lightsaber

Tento hlavalam obsahuje hrací pole 10 světelných mečů. Pro vyřešení hlavalamu musíme utvořit 5 párů mečů, které budou překřížené. Meč můžeme posouvat o 3 pole např. pokud chceme posunout meč na prvním poli, tak ho přesuneme doprava na čtvrté pole. Pokud přesouváme meč přes takovýto pár mečů, tak se pár počítá za dva meče.

V kódu řešíme hlavalam tak, že si vytvoříme pole obrázků a pole hodnot, které budou představovat hodnoty mečů. Při kliknutí na meč se zavolá funkce, která zkontroluje, jestli můžeme jít doleva nebo doprava. Pokud můžeme jít na obě strany, tak se uživateli zobrazí obě vyskakovací okna, kde bude mít na výběr, jestli chce táhnout mečem doprava nebo doleva. Pokud může hráč táhnout na jeden směr, tak zkontrolujeme pole hodnot a přesuneme se o 3 hodnoty do předu. Pokud uživatel nemůže zvoleným mečem nikam táhnout, tak se objeví hláška s upozorněním, že tímto mečem nemůže nikam táhnout. Pokud už hráč nemůže žádným mečem nikam táhnout, tak se opět zobrazí hláška s tím, že hráč nedokázal vyřešit hlavalam.



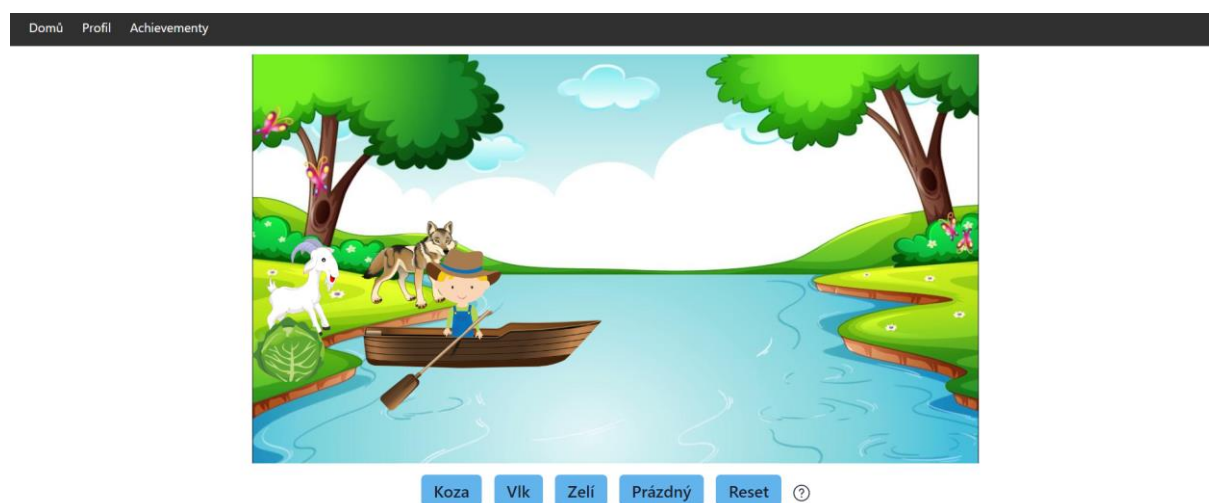
Ilustrace 9 – hlavalam se světelnými meči [zdroj: autor]

5.2. Hlavalam vlk, koza a zelí

V tomto hlavalamu řešíme problém farmáře, který potřebuje dostat na druhý břeh vlka, kozu a zelí, ale do loďky se mu vejde pouze jeden z nich. A navíc pokud nechá o samotě vlka s kozou, tak vlk kozu sežere anebo pokud zůstane koza sama se zelím, tak koza sežere zelí.

Hráč hlavolam hraje tak, že klikne na tlačítko toho, koho chce přesunout na druhou stranu. Může také přejet na druhou stranu i sám.

V kódu řešíme hlavolam, tak že si uchováváme u každé naší postavy pozici, na kterém břehu je. Poté jsme přidali tlačítka, na které když klikneme, tak se vybraná postava přesune na druhou stranu. Po každém přesunutí kontrolujeme, jestli na nějakém břehu nezůstala sama koza se zelím nebo vlk s kozou. A pokud jsme zjistili, že taková situace nastala, tak se hráči objeví vyskakovací okno s důvodem, že hlavolam nevyřešil a prohrál. [14]



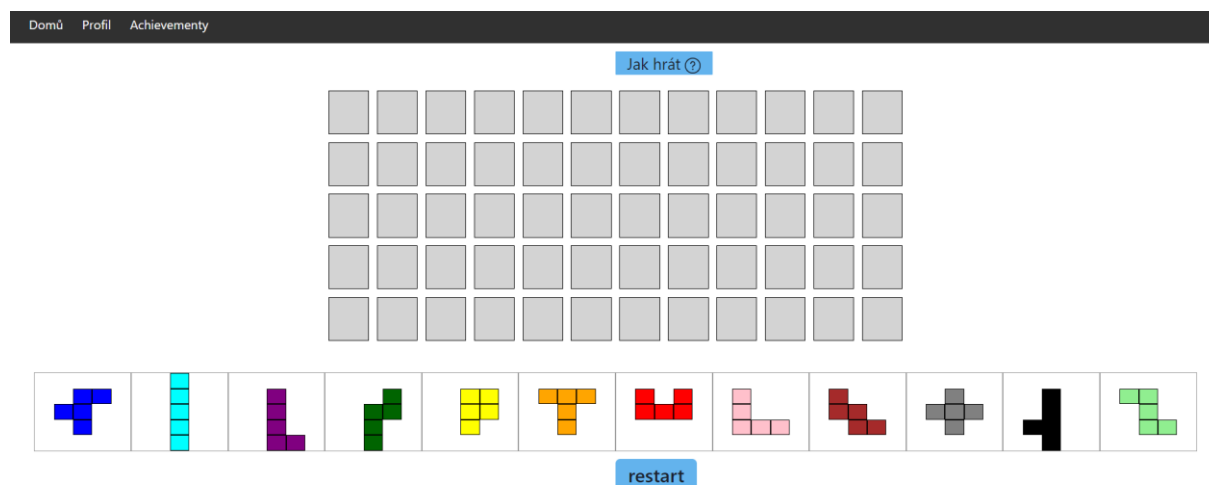
Ilustrace 10 – hlavolam vlk, koza a zelí [zdroj: autor]

5.3. Pentomino hlavolam

Pentomino hlavolam může mít různé hrací plochy. Pro náš případ jsem zvolil hrací plochu 12x5. Pro vyřešení hlavolamu musíme do této plochy složit 12 dílků každý složený z 5 kostiček, které jsou poskládané do různých tvarů. Hráč může dílky různě otáčet pro více možností řešení.

V kódu jsem si vytvořil dvou rozměrné pole pro dílky složených z hodnot 0 a 1 s tím, že 1 představuje kostičku ze které se dílek skládá. A také jsem si vytvořil pole plné 0, které představuje herní plochu. Když začneme přetahovat dílek, zavoláme funkci, která převezme, jakým dílkem hýbeme a na jakou souřadnici v poli hýbeme. Při přetahování dílku do hrací plochy, tak na event onDrop voláme funkci, která zkontroluje, jestli se dílek vejde do hranic pole anebo jestli se nepřekrývá s jiným dílkem. Pokud tyto podmínky splní, tak dílek položíme do pole hodnot hracího pole a vykreslíme položený dílek do hracího pole. Dále také máme připravené onKeyDown event pro každý dílek. Pokud nastane takový event spustí se funkce, ve které zkontrolujeme, jakou klávesu uživatel stiskl a pokud se jedná o klávesu R otočíme dílek o 90 stupňů po směru hodinových

ručiček a pokud se jedná o klávesu F, tak se dílek převrátí ze shora dolů. Hráč také může dílky z hrací plochy vyjmout. To řešíme tak, že při přetahování dílku zjišťujeme, kam dílek uživatel pokládá a pokud je například mimo hranice hrací plochy, tak dílek vrátíme zpět k ostatním dílkům odkud si jej hráč vybral. Po každém položení dílku zkontrolujeme stav hrací plochy, jestli není zaplněná dílkami v podobě hodnot v našem dvou rozměrném poli. Pokud zjistíme, že hráč úspěšně složil všechny dílky do hrací plochy, tak zobrazíme okno s gratulací a v databázi uživateli změním status u tohoto hlavolamu na true.



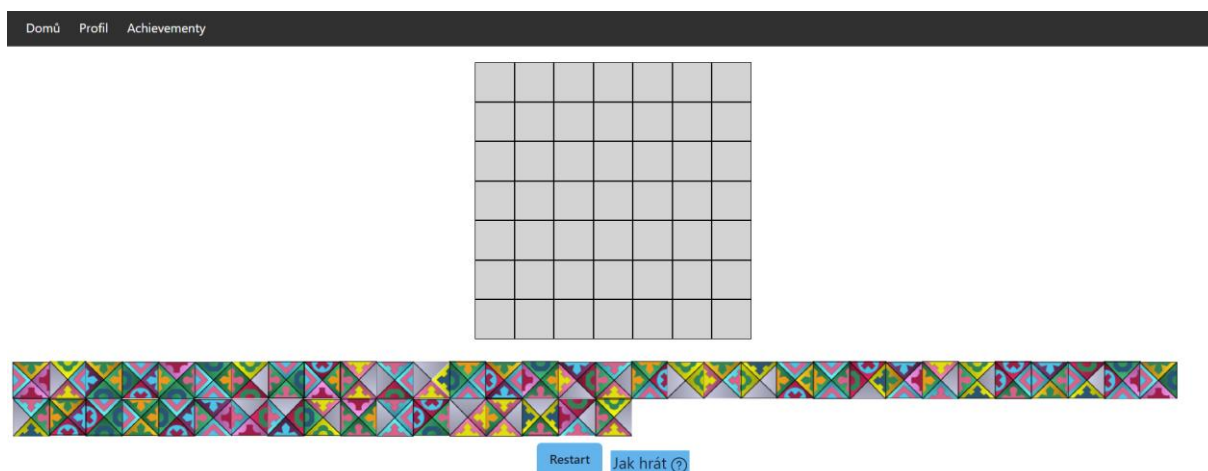
Ilustrace 11 – pentomino hlavolam [zdroj: autor]

5.4. Eternity hlavolam

Hlavolam Eternity obsahuje hrací plochu o velikosti 7x7. A do ní musí hráč složit celkově 49 dílků. Každý dílek má 4 strany a každá strana může být jiná, ale také může mít 2 shodné strany. Dílky musíme skládat tak, aby byly dílky shodnými stranami k sobě.

V kódu jsem si vytvořil dvou rozměrné pole představující hrací plochu a vyrenderoval ho pomocí funkce map. Při načtení stránky všechny dílky různě zamícháme mezi sebou, aby se neobjevovali na stejném místě a také je různě otáčíme, aby dílky nebyly hned od začátku otočeny správně. Hráč samozřejmě může dílky otáčet pomocí klávesy R. Proto máme připravený onKeyDown event, který nám zavolá funkci, ve které potom kontrolujeme, jestli uživatel stiskl klávesu R a pokud ano, tak otočíme dílek o 90 stupňů. Při přetahování dílku si uchováváme souřadnici, kam v hrací ploše se dílek pokládá a po položení v poli hodnot představující hrací plochu změním hodnotu na 1 představující obsazenou pozici. Při položení také kontrolujeme, jestli jsou v okolí nějaké dílky a pokud ano, tak porovnáváme, jestli se dílky k sobě pokládají shodnými stranami. Pokud ne, tak se dílek nepoloží. Po každém položení kontrolujeme, jestli není hlavolam vyřešený. To kontrolujeme, tak že projdeme pole hrací plochy, jestli jsou všechny pozice obsazeny a

pokud ano zobrazíme uživateli okno s gratulací a v databázi aktualizujeme uživateli stav hlavolamu na true.



Ilustrace 12 – eternity hlavolam [zdroj: autor]

6. Budoucnost webu

V této části popisuji, co jsem nestihl dokončit a co by bylo dobré přidat do mé aplikace.

6.1. Co v aplikaci chybí?

Aplikace v aktuální podobě není dokonalá to je určitě zřejmé. Grafické zpracování by určitě potřebovali výrazné změny. Já ale zrovna graficky nadaný člověk nejsem a podle toho to také vypadá. Uživatelé by si zasloužili přívětivější prostředí a nebylo by na škodu přidat např. profilové obrázky pro hráče a možnost jejich změny stejně tak možnost změnit si přezdívku a možnost si svůj profil smazat. Pro větší motivaci hráčů k vyřešení hlavolamu by také mohl sloužit žebříček nejlepších hráčů. Přidání dalších hlavolamů určitě také stojí za zmínku. V aplikaci se momentálně nacházejí 4 hlavolamy a do budoucna by to chtělo alespoň pár hlavolamů přidat. V projektu bohužel chybí pár věcí jako je ukládání postupu a správa uživatelů za pomoci administrativního prostředí. Ukládání postupu jsem u prvních tří hlavolamů neřešil, jelikož hlavolamy jsou velmi jednoduché a na jejich vyřešení není potřeba mnoho času. U posledního hlavolamu Eternity už tomu je ale jinak a mě bohužel nevystačil čas, abych stihl ukládání postupu zpracovat. Co se týče administrativního prostředí na to mi taktéž nezbyl čas, jakožto správce projektu mohu spravovat uživatele přímo v prostředí firebase, ale to není to uživatelské prostředí, které jsem měl vytvořit.

6.2. Plán do budoucna

Aplikaci bych chtěl pořádně testovat a podle feedbacku dále rozvíjet. Přidávat nové věci, které by podle testerů stálo za to přidat. Dále upravovat některé věci, které se testerům nelíbily. Chtěl bych se také domluvit s nějakým grafikem na přepracování vzhledu celkové aplikace, který by měl být přívětivější pro všechny uživatele.

7. Závěr

Cílem projektu bylo vytvořit webovou aplikaci, kde se uživatel přihlásí a poté bude moci řešit hlavolamy, spravovat svůj profil a bude se moci podívat na svoje získané achievementy. K tomu mělo být vytvořeno prostředí pro admina a případné ukládání postupu v hlavolamu.

Byl vytvořen přihlašovací systém s možnostmi změnit si heslo, email nebo si heslo resetovat. K tomuto řešení jsem použil Firebase a jejich rozšíření v podobě Firebase authentication a Firestore. Dále byli vytvořeny 4 hlavolamy, které jsem prvně nastudoval a poté jsem přišel s vlastním řešením, jak hlavolamy vizualizovat a hrát na webu. K hlavolamům byl také vytvořen systém achievementů, který motivuje hráče k vyřešení dalších hlavolamů. Bohužel nezbyl čas na to vytvořit administrativní prostředí pro admin uživatele a ukládání postupu u časově náročného hlavolamu.

8. Seznam použité literatury

1. What Is Node.js and Why You Should Use It. Kinsta® Cloud Hosting – All Your Projects in One Platform [online]. Copyright © 2023 Kinsta Inc. All rights reserved. [cit. 17.03.2023]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-node-js/#nodejs-architecture-and-how-it-works>
2. What is Next.js? - Enonic. Take control of your content: Composable content platform without limitations [online]. Copyright © Enonic AS. All Rights Reserved. [cit. 17.03.2023]. Dostupné z: <https://enonic.com/blog/what-is-next-js>
3. How Next.js Works | Learn Next.js. Next.js by Vercel – The React Framework [online]. Copyright © [cit. 17.03.2023]. Dostupné z: <https://nextjs.org/learn/foundations/how-nextjs-works>
4. React.js Introduction and Working – GeeksforGeeks. GeeksforGeeks | A computer science portal for geeks [online]. [cit. 17.03.2023]. Dostupné z: <https://www.geeksforgeeks.org/react-js-introduction-working/>
5. Firebase Documentation. Firebase [online]. [cit. 19.03.2023]. Dostupné z: <https://firebase.google.com/docs>
6. Firestore | Firebase. Firebase [online]. [cit. 17.03.2023]. Dostupné z: <https://firebase.google.com/docs/firestore>
7. Firebase Authentication. Firebase [online]. [cit. 17.03.2023]. Dostupné z: <https://firebase.google.com/docs/auth>
8. Lekce 1 - Úvod do TypeScriptu. itnetwork.cz - Učíme národ IT [online]. Copyright © 2023 itnetwork.cz. Veškerý obsah webu [cit. 17.03.2023]. Dostupné z: <https://www.itnetwork.cz/javascript/typescript/uvod-do-typescriptu>
9. What Is Typescript? | The New Stack. The New Stack | DevOps, Open Source, and Cloud Native News [online]. Copyright © The New Stack 2023 [cit. 17.03.2023]. Dostupné z: <https://thenewstack.io/what-is-typescript/>
10. Why You Should Start Using Chakra UI. [online]. [cit. 17.03.2023]. Dostupné z: <https://www.freecodecamp.org/news/why-should-you-start-using-chakraui/>
11. Introduction to MongoDB — MongoDB Manual. MongoDB: The Developer Data Platform | MongoDB [online]. Copyright © 2023 MongoDB, Inc. [cit. 23.03.2023]. Dostupné z: <https://www.mongodb.com/docs/manual/introduction/>
12. Authentication — MongoDB Manual. MongoDB: The Developer Data Platform | MongoDB [online]. Copyright © 2023 MongoDB, Inc. [cit. 23.03.2023]. Dostupné z: <https://www.mongodb.com/docs/manual/core/authentication/>
13. Dynamic URLs vs. static URLs | Google Search Central Blog | Google Developers. Google Developers [online]. [cit. 23.03.2023]. Dostupné z: <https://developers.google.com/search/blog/2008/09/dynamic-urls-vs-static-urls>
14. Michal Musílek. Forpsi.com [online]. [cit. 26.03.2023]. Dostupné z: <http://musilek.eu/michal/desetsirek.html?menu=hry&item=d&lang=cz>
15. Vlk, koza a zelí – Wikipedie. [online]. [cit. 26.03.2023]. Dostupné z: https://cs.wikipedia.org/wiki/Vlk,_koza_a_zel%C3%AD
16. Pentomino – Wikipedie. [online]. [cit. 27.03.2023]. Dostupné z: <https://cs.wikipedia.org/wiki/Pentomino>

17. eternity. MathPuzzle.com [online]. [cit. 27.03.2023]. Dostupné z:
<https://www.mathpuzzle.com/eternity.html>

9. Seznam obrázků

Ilustrace 1 – Architektura aplikace [zdroj: autor]	11
Ilustrace 2 – Firestore databáze [zdroj: autor]	12
Ilustrace 3 – Firebase Authentication [zdroj: autor]	12
Ilustrace 4 – registrační formulář [zdroj: autor]	14
Ilustrace 5 – nepovedená registrace [zdroj: autor]	15
Ilustrace 6 – domovská stránka [zdroj: autor]	15
Ilustrace 7 – profil uživatele [zdroj: autor]	16
Ilustrace 8 – achievements [zdroj: autor]	16
Ilustrace 9 – hlavolam se světelnými meči [zdroj: autor]	17
Ilustrace 10 – hlavolam vlk, koza a zelí [zdroj: autor]	18
Ilustrace 11 – pentomino hlavolam [zdroj: autor]	19
Ilustrace 12 – eternity hlavolam [zdroj: autor]	20