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

Ke Kamenci 151, PARDUBICE

MATURITNÍ PROJEKT

Webová aplikace pro nabízení a vyhledávání brigád

Příjmení, jméno: Bukai Sebastian

Třída: 4.B

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

Školní rok: 2023/2024

Konzultant: *Bc. Vlaďka Janů*

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Sebastian Bukai*

Školní rok: *2023/2024*

Třída: 4.A

Obor: *Informační technologie 18-20-M/01*

Téma práce: *Aplikace pro vyhledávání a nabízení brigád*

Vedoucí práce: *Bc. Vlaďka Janů*

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

Projekt si klade za cíl vytvořit efektivní platformu, která umožní uživatelům snadně a rychle vytvořit brigády nebo najít pomocníky pro splnění různorodých úkolů. S pomocí intuitivního a přehledného rozhraní nabízí uživatelům možnost prezentovat své dovednosti, zkušenosti a dostupnost.

Základní technologie:

- React
- NextJS
- Typescript
- Firebase
- MUI

Stručný časový harmonogram

- Studování technologií, návrh aplikace a začátek s vytvářením samotné aplikace. (září)
- Implementace Firebase Databáze, práce ve frameworku NextJs. Dále vytvoření funkčnosti vytváření a vyhledávaní inzerátů. Práce na registraci a přihlašování a bezpečnosti. (říjen-listopad)
- Práce na vzhledu aplikace a testování doposud implementovaných funkcí. (prosinec)
- Práce na dokumentaci a dodělávání

Prohlášení

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

V Pardubice dne 7.2.2024 ……

Poděkování

Děkuji především Bc. Vlaďce Janů za vedení a konzultace mého projektu a dokumentace.
Dále bych rád poděkoval svým spolužákům za pomoc při testovaní a konzultacích projektu.

Anotace

Aplikace "Hodinový syn" propojuje lidi hledající brigády s těmi, kteří jsou ochotni pomoci. Klíčové funkce zahrnují vytváření, zobrazování a filtrování inzerátu. Dále hodnocení uživatelů mezi sebou a komunikační systém mezi uživateli.

Klíčová slova

Webová aplikace, Next.js, React, Firebase, MUI, brigády

Annotation

The "Hourly Helper" application connects people looking for part-time jobs with those willing to help. Key features include creating, displaying, and filtering job listings. Additionally, it includes a user rating system and a communication platform for users.

Key words

Web application, Next.js, React, Firebase, MUI, part time jobs.

Obsah

1. Úvod 1
1.1. Jak jsem se dostal k nápadu 1
2. Technologie 1
2.1. Next.js 1
2.2. React 2
2.3. Typescript 2
2.4. Firebase 3
2.5. NextAuth 3
2.6. MUI 4
3. Pojmy 4
4. Architektura 5
4.1. Databáze 6
5. Funkce aplikace 7
5.1. Přihlašování 8
5.2. Vytváření inzerátů 8
5.3. Zobrazování inzerátů 9
5.4. Filtrování Inzerátů 9
5.5. Profil 9
6. Frontend/ Grafický návrh 11
7. Uživatelská příručka 11
8. Co mi chybí 15
9. Závěr 16
10. Zdroje 17
11. Seznam obrázků 17

1. Úvod

Projekt Hodinový Syn představuje webovou aplikaci zaměřenou na usnadnění vyhledávání a nabízení příležitostí pro brigády. Cílem je poskytnout mladým lidem možnost najít si vhodnou příležitost ke krátkodobé práci, aniž by byli omezeni velkými korporacemi, jako je například McDonald's.

Aplikace umožňuje uživatelům vytvořit si vlastní inzerát, kde specifikují požadavky na odměnu a své dovednosti. To jim poskytuje možnost prezentovat své schopnosti a stanovit si odměnu podle vlastních preferencí. Druhá skupina uživatelů, kteří potřebují pomoc nebo pracovníky na krátkodobé úkoly, mohou následně reagovat na tyto inzeráty a navázat kontakt s vhodnými brigádníky. Aplikace je intuitivní a přehledná.

1.1. Jak jsem se dostal k nápadu

Nápad mě napadl prostřednictvím mého rodného místa, konkrétně vesnice, kde jsem vyrůstal. Každá vesnice obvykle disponuje malým nebo větším náměstím, na kterém se nachází nástěnky s různými pozvánkami na události jako koncerty nebo plesy. Jednoho dne jsem narazil na inzerát od 15letého chlapce, který nabízel své služby za 100 korun za hodinu, s nabídkou péče o zahradu. To mě přivedlo k myšlence přenést podobný koncept do digitálního prostoru.

2. Technologie

V následující části textu se věnuji technologiím, které jsem v aplikaci využil. Webová aplikace je napsána ve frameworku Next.js, jenž je postaven na javascriptové knihovně React. Jako Databázový systém slouží Firebase. A na stylování aplikace MUI.

2.1. Next.js

Next.js je framework postavený na vývojovém prostředí Node.js, rozšiřující známou JavaScriptovou knihovnu React. Tato kombinace vytváří optimální prostředí pro vývoj webových aplikací. Jednou z klíčových funkcí Next.js je podpora server side rendering (SSR). Tato technika generuje HTML kód na straně serveru, který je následně odeslán uživateli do prohlížeče. Až poté jsou aktivovány javascriptové prvky, což zaručuje rychlé načítání obsahu stránky.

Výhody:

- **Server Side Rendering (SSR):** Next.js nabízí vestavěnou podporu pro SSR, což umožňuje generovat HTML obsah na straně serveru a poskytovat uživatelům rychlejší a více SEO příznivé načítání stránek.
- **Hot Module Replacement (HMR)**: Next.js podporuje HMR, což umožňuje rychlé a snadné zobrazení změn v kódu v prohlížeči bez nutnosti ručního obnovení stránky.

2.2. React

React je JavaScriptová knihovna, která se specializuje na tvorbu uživatelského rozhraní. Její klíčovou předností je schopnost efektivně pracovat s dynamickými daty. Základním stavebním prvkem React aplikací jsou komponenty, což jsou znovupoužitelné bloky kódu, které umožňují strukturovat aplikaci a zlepšují přehlednost kódu.

Je důležité poznamenat, že React často není používán samostatně, ale je kombinován s dalšími nástroji a frameworky. V uvedeném případě je zmíněna integrace s frameworkem Next.js

Výhody:

- **Vysoká efektivita:** React používá virtuální DOM (Document Object Model), což umožňuje efektivní manipulaci s UI a minimalizuje potřebu přímých manipulací s DOM, což vede k lepší výkonnosti aplikací.
- **Rozsáhlá komunita a ekosystém:** React má velkou a aktivní komunitu vývojářů, kteří neustále přispívají novými knihovnami, nástroji a návody, což usnadňuje vývoj a řešení problémů.
- **Jednoduchá integrace:** React lze snadno integrovat s dalšími nástroji a frameworky, což umožňuje vytvoření robustních a plně funkčních aplikací.

2.3. Typescript

TypeScript představuje open-source nadstavbu pro JavaScript, která přináší statické typování a další pokročilé vlastnosti pro vývojáře. S využitím transpileru je možné převést TypeScript kód na ekvivalentní JavaScript, což umožňuje kombinovat moderní konstrukce jazyka s širší kompatibilitou. Zavedení datových typů, jako jsou boolean, number, string a any, přináší do vývoje jistotu a umožňuje odhalování chyb již v raných fázích vývoje. Tím se zvyšuje spolehlivost kódu, zjednodušuje proces debugování a usnadňuje vytváření komplexních aplikací.

Výhody:

- **Statické typování:** TypeScript přináší statické typování do JavaScriptu, což umožňuje definovat typy proměnných, parametrů funkcí a návratových hodnot funkcí. Tímto způsobem pomáhá odhalovat chyby v kódu již v době kompilace, což vede ke zvýšení spolehlivosti a kvality kódu.
- **IntelliSense a doplňování kódu**: Díky statickému typování TypeScript poskytuje lepší podporu pro IntelliSense a automatické doplňování kódu v IDE (Integrovaném vývojovém prostředí), což zvyšuje produktivitu vývojářů.

2.4. Firebase

Firebase je platforma poskytující širokou škálu nástrojů a služeb pro vývoj mobilních a webových aplikací. Nabízí vývojářům cloudové služby, které zahrnují autentizaci, databázi, hosting, notifikace a mnoho dalšího. S Firebase lze rychle a snadno integrovat backendovou funkcionalitu do projektů, což usnadňuje vývoj a umožňuje koncentrovat se na samotnou logiku aplikace.

Jedním z klíčových prvků Firebase je Firestore, která je flexibilní, škálovatelná a plně spravovaná databáze dokumentů. Firestore umožňuje ukládat a synchronizovat data mezi různými klienty v reálném čase, stejně jako Realtime Database, avšak nabízí pokročilejší možnosti dotazování a strukturování dat.

Výhody:

- **Dokumentace a podpora**: Firebase má rozsáhlou dokumentaci a aktivní komunitu uživatelů, což usnadňuje učení se a řešení problémů během vývoje aplikací.
- **Škálovatelnost**: Firestore je plně spravovaná databáze, která automaticky škáluje podle potřeby, což zajišťuje spolehlivost a výkonnost aplikací i při náhlých nárůstech uživatelské základny.

2.5. NextAuth

NextAuth je knihovna pro autentizaci a autorizaci ve webových aplikacích postavených na frameworku Next.js. Jedním z hlavních benefitů NextAuth je jeho jednoduché a flexibilní použití.

NextAuth také podporuje použití CSRF (Cross-Site Request Forgery) tokenů k ochraně před útoky, které by mohly vést k neautorizovaným žádostem vůči vaší aplikaci. Tím,

že automaticky generuje a ověřuje CSRF tokeny pro každou autentizační žádost, NextAuth přispívá k zvýšení bezpečnosti aplikace.

Výhody:

- **Flexibilita**: NextAuth poskytuje širokou škálu možností a konfiguračních parametrů, které umožňují přizpůsobit autentizaci a autorizaci potřebám konkrétní aplikace.
- **Jednoduchá integrace**: NextAuth je navržen tak, aby byl snadno integrovatelný do aplikací postavených na frameworku Next.js, což zjednodušuje proces implementace autentizace a autorizace.
- **Dobrá dokumentace**: NextAuth disponuje kvalitní dokumentací a komunitní podporou.

2.6. MUI

Material-UI (MUI) je jednou z nejpopulárnějších knihoven UI pro React, která poskytuje sadu komponent navržených v souladu s Material Design koncepty od společnosti Google.

- **Jednoduchost použití a flexibilita:** Material-UI poskytuje širokou škálu předdefinovaných komponent, které lze snadno integrovat do projektu. Zároveň umožňuje jednoduché přizpůsobení stylu a chování komponent podle potřeb projektu.

- **Responsivní design:** Material-UI umožňuje snadno vytvářet responsivní webové aplikace, které se automaticky přizpůsobují různým velikostem obrazovky a zařízením. To je klíčové pro moderní webové stránky a aplikace, které chtějí poskytnout uživatelům plynulý zážitek bez ohledu na používané zařízení.

3. Pojmy

3.1. Frontend

Pojem frontend vznikl v oblasti programování webových aplikací a označuje část webové stránky, kterou běžní návštěvníci vidí.

3.2. Backend

Backend představuje opak frontendu a slouží k administraci webové aplikace. Zde se řídí a upravuje obsah, který poté frontend zobrazuje uživatelům. Tato část webové aplikace není běžně viditelná a slouží k zajištění funkcionality a manipulaci s daty, které se následně zobrazují na stránce.

3.3. API

API (Application Programming Interface) je rozhraní umožňující komunikaci mezi frontendem a backendem a přenášení dat mezi nimi.

4. Architektura

Tento projekt je postaven na JavaScriptovém frameworku React, který je používán pro vývoj frontendu. Kód je napsán v TypeScriptu, což je staticky typovaný jazyk založený na JavaScriptu, který přidává statické typy a další pokročilé funkce pro vývoj velkých aplikací.

Frontend aplikace je postaven na Reactu a používá Material-UI pro komponenty uživatelského rozhraní.

Backendová logika je implementována v serverless funkcích pomocí Next.js API routes. Tyto funkce jsou nasazeny na cloudovou platformu a jsou volány z frontendové části aplikace pomocí HTTP požadavků. Backend se stará o autentizaci uživatelů, správu inzerátů a manipulaci s daty v databázi. Autentizace uživatelů je zajištěna pomocí NextAuth, což je knihovna poskytující jednoduché a flexibilní řešení pro autentizaci s podporou různých poskytovatelů identity.

Jako databázové řešení je v projektu použita Firestore, která je NoSQL databází poskytovanou Firebase. Firestore je dokumentově orientovaná databáze, což znamená, že data jsou organizována do kolekcí a dokumentů. Každý dokument obsahuje sadu klíč-hodnota, kde hodnota může být jakýkoli datový typ, včetně dalších dokumentů nebo kolekcí.

Routing je řešen pomocí next/router, což je součástí frameworku Next.js. Next.js je framework pro React, který poskytuje funkce pro server-side rendering a statické ítigenerování stránek, což zlepšuje výkon a SEO.

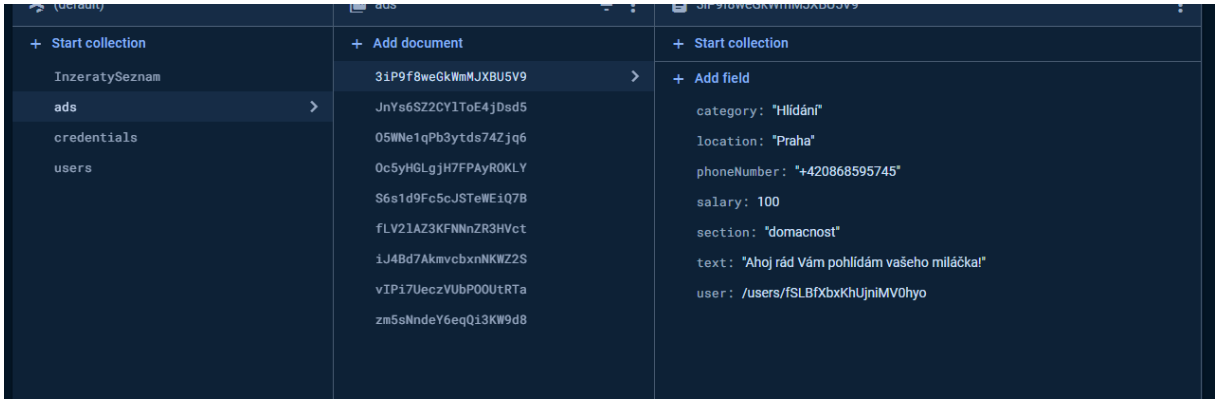
4.1. Databáze

Kolekce "ads": Obsahuje inzeráty. Každý inzerát je reprezentován dokumentem, který obsahuje různé informace o inzerátu, jako je kategorie, umístění, telefonní číslo, mzda, text inzerátu a uživatel, který inzerát vytvořil. Jednotlivé inzeráty jsou odděleny a uloženy jako dokumenty v rámci kolekce "ads".

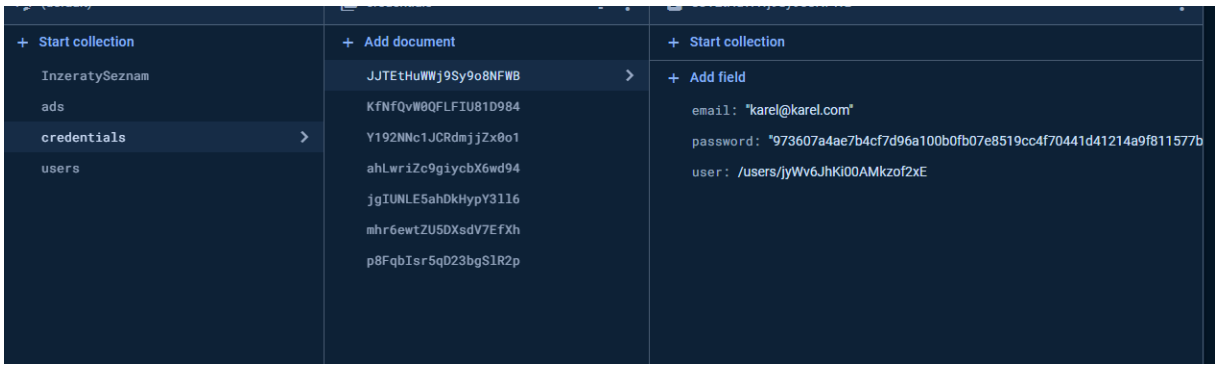
Kolekce "credentials": Obsahuje údaje pro přihlášení. Každý uživatel má v této kolekci uloženy údaje pro přihlášení, jako je e-mailová adresa a heslo. Každý uživatel je reprezentován dokumentem, který obsahuje e-mailovou adresu a hashované heslo (pro zajištění bezpečnosti hesel).

Kolekce "users": Obsahuje informace o uživatelích. Každý uživatel má v této kolekci svůj vlastní dokument, který obsahuje různé informace, jako je jméno, e-mailová adresa atd.

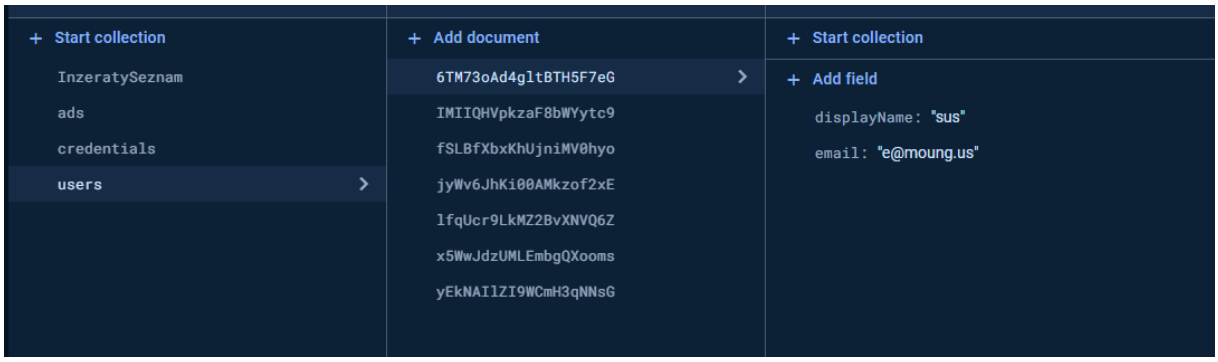
Reference s ID jsou implementovány prostřednictvím textových polí v dokumentech, která obsahují ID cílových dokumentů. Tyto odkazy umožňují vytváření vazeb mezi různými částmi databáze. Například v dokumentu inzerátu může pole obsahovat odkaz na uživatele, který inzerát vytvořil.



Obrázek 1 ukázka databáze



Obrázek 2ukázka databáze



Obrázek 3 ukázka databáze

5. Funkce aplikace

V této části se budu podrobně zabývat všemi funkcemi, které jsem doposud do aplikace implementoval.

Přihlašování:

- Implementace autentifikace pomocí Next Auth
- Zavedení procesu registrace a přihlášení uživatelů

Vytváření inzerátů

- Vytvoření formuláře umožňujícího uživatelům zadávat informace o inzerátu jako rubrika, cena, lokalita apod.
- Propojení inzerátů s konkrétními uživateli pomocí identifikátorů.

Zobrazování inzerátů

- Zobrazení inzerátu, vytvořeného konkrétním uživatelem.

Filtrování inzerátů

- Dynamická aktualizace seznamu inzerátů na základě zvolených filtrů uživatele.
- Použití různých prvků pro výběr kategorií, lokací a cenového rozpětí.

Profil

- Implementace profilu uživatele s možností zobrazení a aktualizace informací.

Moje inzeráty

- Zobrazení inzerátů, které vytvořil přihlášený uživatel.
- Ověření přihlášení uživatele a případná přesměrování na stránku přihlášení.

Detail inzerátu

- Zobrazení detailů konkrétního inzerátu, včetně informací o tvůrci.
- Využití ID inzerátu pro načtení potřebných dat z databáze.

5.1. Přihlašování

Původně jsem implementoval autentifikaci pomocí Firebase Auth, ale nakonec jsem se rozhodl přejít na Next Auth. Hlavním důvodem bylo, že Firebase Auth nabízí pouze základní funkcionality pro správu uživatelských účtů. Next Auth mi poskytuje vyšší flexibilitu a kontrolu, což je více vyhovující p ro můj projekt.

Jak jsem již zmínil projekt implementuje autentizaci pomocí Next Auth a poskytuje stránky pro registraci a přihlášení uživatelů. Při registraci uživatel zadá své jméno, email a heslo, které jsou odeslány na server. Server ověřuje, zda uživatel s daným emailem již neexistuje. Pokud neexistuje, vytvoří nový uživatelský záznam v databázi a uloží hashované heslo. Poté odpoví klientovi zprávou "ok". Při přihlášení uživatel zadá svůj email a heslo. Tyto údaje jsou rovněž odeslány na server. Server ověří, zda jsou poskytnuté údaje platné. Pokud ano, uživatel je přihlášen a přesměrován na další stránku. Pokud údaje nejsou platné, server odpoví s chybovou zprávou.

Pro ochranu formulářů na stránkách registrace a přihlášení jsou využity CRSF tokeny. Při každém načtení těchto stránek je generován nový CRSF token a předám klientovi. Klient poté musí tento token předat zpět na server při odesílání formulářů. Server pak ověřuje platnost tohoto tokenu při zpracování požadavku. Pokud token neodpovídá očekávané hodnotě, server odmítne požadavek jako potenciálně nebezpečný.

5.2. Vytváření inzerátů

Vytváření inzerátu pomocí formuláře je klíčovou funkcí této aplikace. Formulář umožňuje uživatelům zadávat informace o inzerátu, jako je rubrika, kategorie, cena, lokalita, telefonní číslo a samotný text inzerátu.

- **Formulář pro vytvoření inzerátu:** Na stránce je zobrazen formulář, který obsahuje vstupní pole pro každý požadovaný údaj. Uživatelé mohou vybírat z různých možností pomocí rozbalovacích nabídek nebo zadávat hodnoty do textových polí.
- **Zpracování formuláře:** Po vyplnění formuláře může uživatel odeslat data stisknutím tlačítka "Vytvořit". Po stisknutí tlačítka se spouští funkce addNewAd, která zpracovává odeslaná data. Nejprve ověřuje, zda uživatel vyplnil všechna povinná pole formuláře. Pokud ano, data jsou připravena k odeslání na server.

- **Odeslání dat na server:** Data vyplněná uživatelem jsou odeslána na server pomocí metody POST. Tyto data obsahují informace o inzerátu, včetně vybrané rubriky, kategorie, ceny, lokality, telefonního čísla a textu inzerátu. Důležitou součástí odeslaných dat je také identifikátor uživatele, který inzerát vytváří. Tímto způsobem jsou inzeráty propojeny s konkrétními uživateli.

5.3. Zobrazování inzerátů

První významnou částí je funkce getServerSideProps, která je zodpovědná za načítání inzerátů ze serverového prostředí. Tato funkce je asynchronní a používá metodu getAllAds, aby získala všechny inzeráty. Poté pomocí Promise.all načte informace o uživatelích pro každý inzerát a vrátí je jako vlastnost props komponenty.

Druhou částí je samotná komponenta Ads, která využívá React stavový hook useState pro uchování stavu seznamu inzerátů a aktuálního filtru. Funkce handleFilter slouží k filtraci inzerátů na základě nastavených kritérií a aktualizuje stav seznamu inzerátů a filtru.

5.4. Filtrování Inzerátů

Filtrování inzerátů v tomto kódu je založeno na dynamické aktualizaci seznamu inzerátů v závislosti na vybraných filtracích uživatele. Když uživatel provádí změny ve filtru (například výběr kategorie, lokace nebo cenového rozpětí), komponenta AdsFilter zachycuje tyto změny a aktualizuje seznam inzerátů zobrazených na stránce Ads.

- Komponenta AdsFilter obsahuje různé prvky pro výběr kategorií, lokací a cenového rozpětí, jako jsou dropdown menu a textová pole. Tyto prvky slouží k nastavení filtru pro inzeráty.
- Každý prvek má přiřazenou událost (např. onChange), která reaguje na změny hodnoty vybrané uživatelem. Když uživatel provede změnu ve filtru, daná událost je spuštěna a aktualizuje se stav filtru.
- Jakmile je změněný filtr uložen v místním stavu (state) komponenty Ads, je zavolána funkce handleFilter, která aktualizuje seznam inzerátů zobrazených na stránce. Tato funkce porovnává každý inzerát s aktuálním stavem filtru a vybírá pouze ty, které splňují zadaná kritéria.

5.5. Profil

Na začátku jsou importovány potřebné moduly a komponenty, včetně několika vlastních. Poté je definována funkce getServerSideProps, která se volá na serverové straně pro získání informací o uživateli.

V samotné komponentě Inzeraty jsou využívány React hooky jako useState pro uchování stavu (např. otevřeného/zavřeného modálního okna) a useEffect pro práci s efekty, jako je získání dat po načtení komponenty.

Funkce handleAddIconClick a handleModalClose reagují na události kliknutí na ikonu plus a zavření modálního okna. Funkce handleSaveText slouží k uložení textu z modálního okna do stavu komponenty.

5.6. Moje Inzeráty

Tento kód reprezentuje stránku pro zobrazení inzerátů, které vytvořil přihlášený uživatel. Stránka nejprve ověřuje, zda je uživatel přihlášen, a pokud ne, přesměruje ho na stránku přihlášení. Poté načte všechny inzeráty, které uživatel vytvořil, pomocí funkce getAllAdsOfUser. Tyto inzeráty jsou poté zobrazeny na stránce.

- **Ověření přihlášení uživatele:** Na začátku kódu se provádí ověření přihlášení uživatele pomocí funkce getServerSession. Pokud uživatel není přihlášen, je přesměrován na stránku přihlášení.
- Načtení inzerátů uživatele: Poté, co je uživatel úspěšně ověřen, se načtou všechny inzeráty, které vytvořil. Tyto inzeráty jsou následně zobrazeny na stránce v komponentě AdsDetails, která zobrazuje detaily inzerátu, včetně možnosti smazání inzerátu.

5.7. Detail Inzerátu

První komponenta DetailAd přijímá inzerát jako vstupní props a renderuje jednotlivé detaily inzerátu, jako je jméno uživatele, cena, kategorie, lokace a telefonní číslo. Každý detail je zobrazen jako samostatný box s ikonou a odpovídajícím textem. Tato komponenta je navržena tak, aby byla responzivní a vhodná pro zobrazení na různých zařízeních.

Využití ID inzerátu v tomto kódu umožňuje aplikaci načíst potřebná data z databáze. Když uživatel navštíví stránku s detaily inzerátu, ID inzerátu z URL adresy je použito k získání informací o inzerátu a jeho tvůrci z databáze.

6. Frontend/ Grafický návrh

Dalším důležitým aspektem projektu byl kromě funkcí aplikace také její grafický návrh. Můj cíl byl zajistit, aby aplikace byla co nejpřehlednější a esteticky přitažlivá. Zaměřil jsem se na to, aby uživatelé ihned po přihlášení do aplikace věděli, co a jak funguje.

6.1. CTA (Call to action)

Nejdříve jsem se zaměřil na udržení pozornosti uživatelů na stránce. K tomu jsem využil známou marketingovou strategii CTA, která má za úkol přimět uživatele k provedení určité akce. V případě tohoto projektu se jedná o tlačítka, která vyzývají uživatele k vytváření/hledání inzerátu, stejně jako o úderné nadpisy, jež mají za úkol vtáhnout uživatele do aplikace.

6.2. Obsah

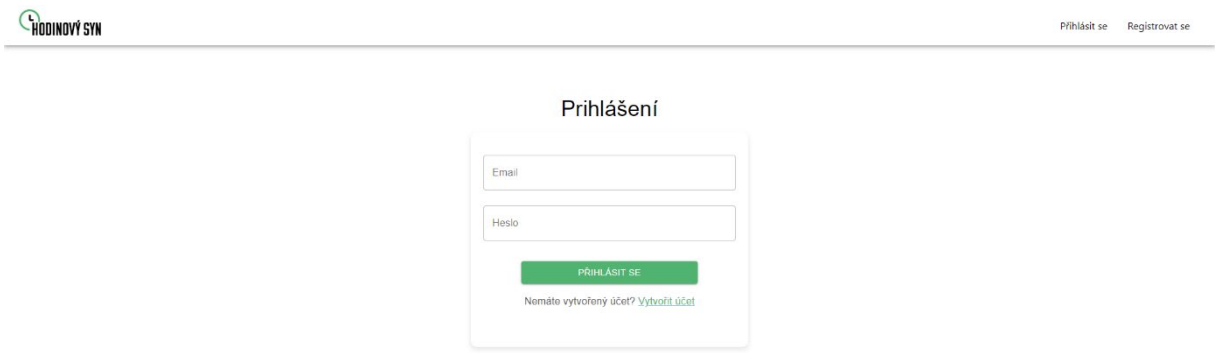
Dále jsem se věnoval samotnému obsahu na stránce, který je sestaven tak, aby byl pro uživatele co možná nejjednodušší na pochopení a zároveň příjemný. K tomu přispívá čistý design stránky s různými zelenými prvky. Také obrázky byly pečlivě vybrány tak, aby si je uživatel jednoduše zapamatoval a zároveň se mu líbily. Důležitou roli hraje i samotný text, včetně nadpisů, textů ve formuláři atd.

7. Uživatelská příručka

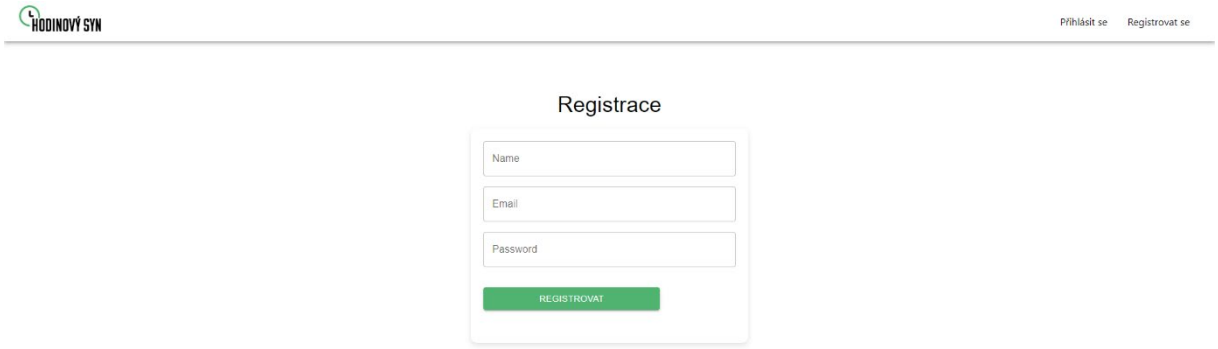
V této kapitole bych se rád v rychlosti pověnoval vysvětlení a provedení aplikací pro uživatele.

7.1. Registrace/ Přihlášení

Poté, co se uživatel dostane na webovou stránku, je nejprve nutné buď se zaregistrovat, nebo přihlásit. Bez účtu není možné v aplikaci provádět většinu funkcí. Při registraci stačí vyplnit pouze email, uživatelské jméno a heslo. Při přihlášení je potřeba zadat pouze email a heslo.



Obrázek 4 přihlašování v aplikaci



Obrázek 5 registrace v aplikaci

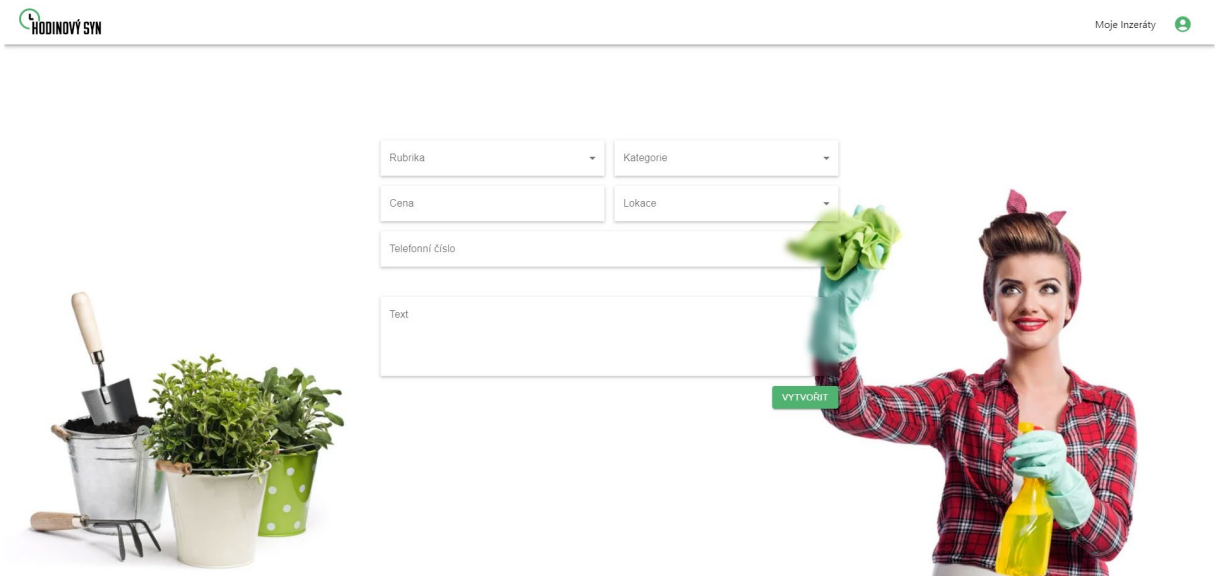
7.2. Vytváření inzerátů/Vyhledávání inzerátů

Potom, co se uživatel do aplikace přihlásí se může rozhodnout, zda chce vytvořit, či najít ideální inzerát.



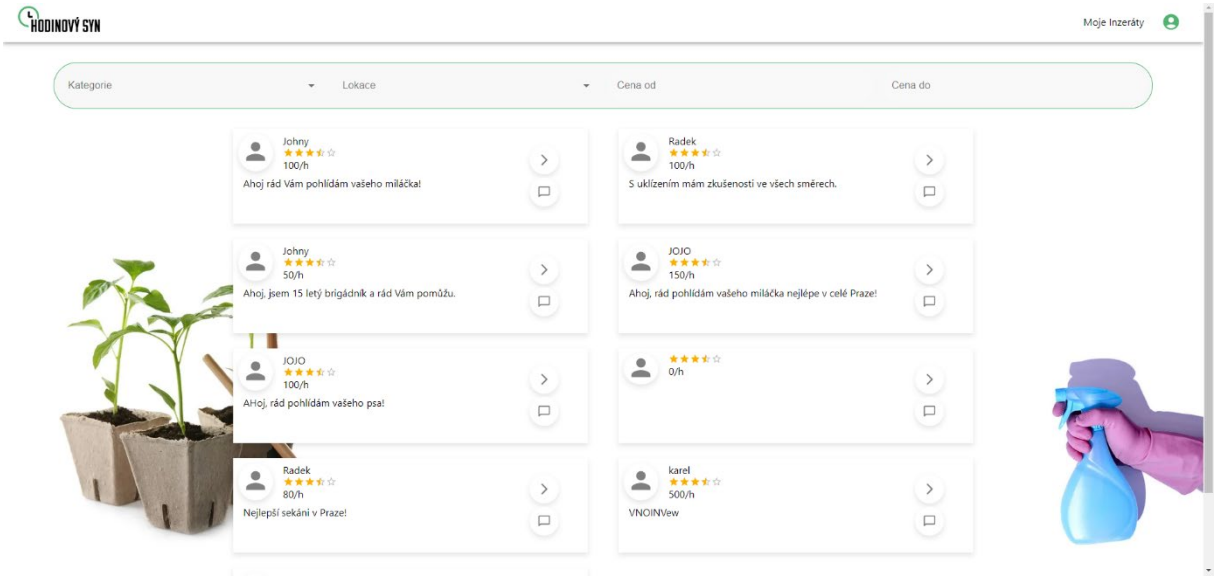
Obrázek 6 hlavní stránka v aplikaci

Pokud se rozhodneme si vytvořit svůj inzerát a klikneme na příslušné tlačítko, aplikace nás přesměruje na formulář pro vytváření inzerátu, kde si nejprve vybereme rubriku, následně kategorii, cenu, lokaci, telefonní číslo a libovolný text, potom stačí pouze kliknout na tlačítko vytvořit.



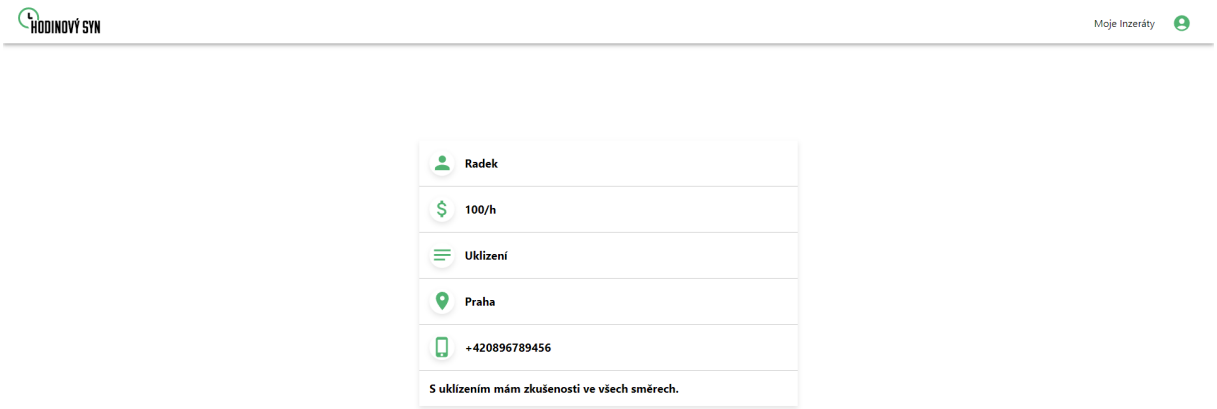
Obrázek 7 vytváření inzerátu v aplikaci

V případě, že by si rád uživatel nějakého brigádníka našel, stačí kliknout na tlačítko, najdi brigádníka a aplikace ho přesměruje na stránku, kde uvidí všechny zatím vytvořené inzeráty uživatelů. Nad inzeráty si lze všimnout filtru, který pomáhá si najít ideálního brigádníka.



Obrázek 8 stránka inzerátů v aplikaci

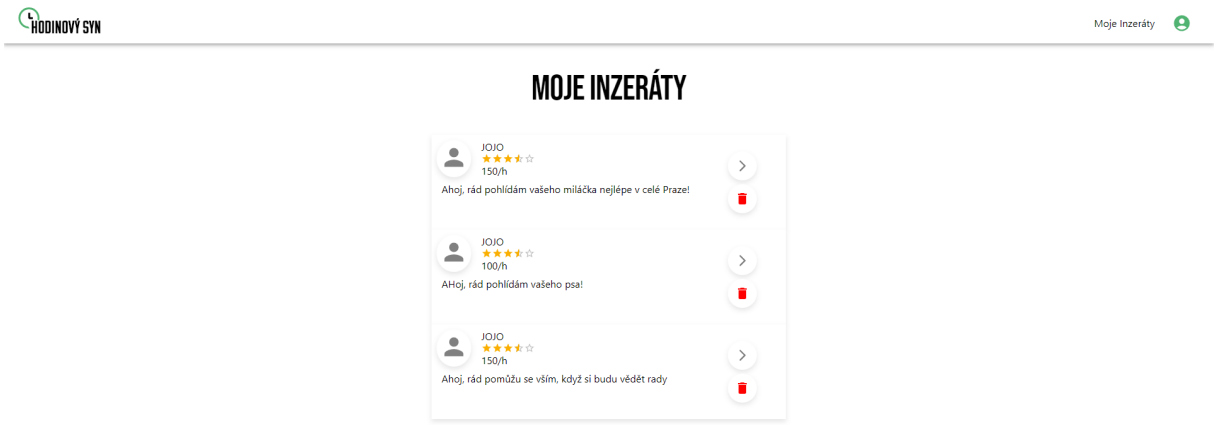
Pokud se uživateli podaří tedy najít svého favorita, může kliknout na ikonu šipky, která ho přesměruje na detail inzerátů, kde je možné vidět např. lokaci, kategorii a telefonní číslo.



Obrázek 9 detail inzerátů v aplikaci

7.3. Moje inzeráty

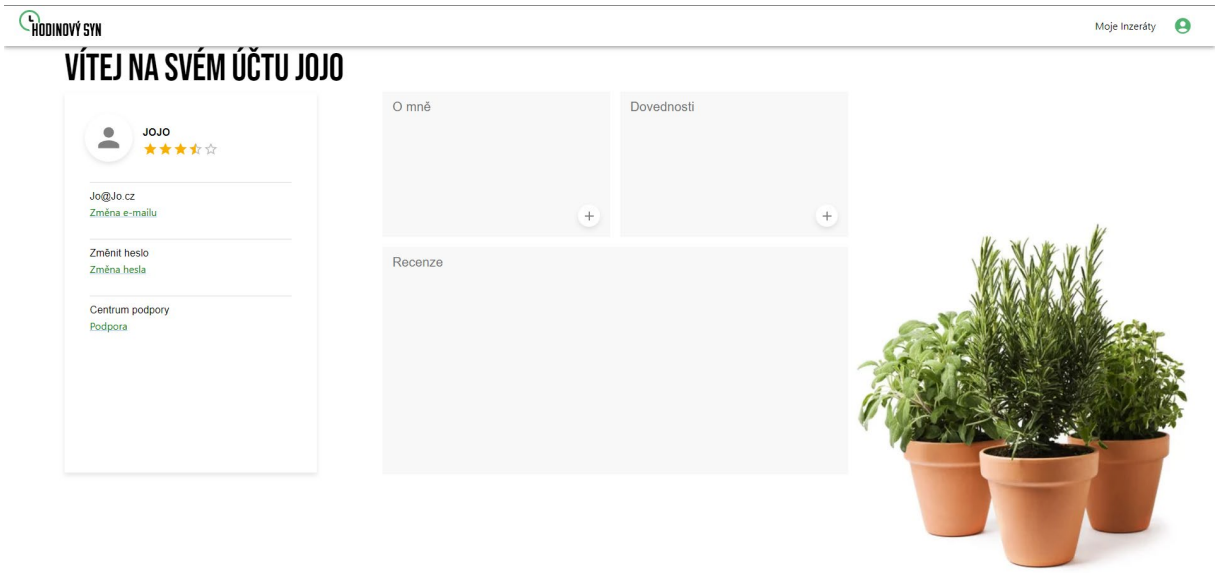
Dále si uživatel může prohlédnout své inzeráty, které zatím vytvořil.



Obrázek 10 moje inzeráty v aplikaci

7.4. Profil

Uživatel si také může prohlédnout svůj uživatelský profil.



Obrázek 11 profil v aplikaci

8. Co mi chybí

Většina funkcí je již implementována, ale stále zbývá práce na recenzích uživatelů a chatovém systému. Recenze plánuji dodělat, převážně kvůli bezpečnosti aplikace, a lépe ohodnocení uživatelé si budou moci zároveň například stanovit vyšší cenu. Chatový systém bude sloužit k usnadnění komunikace mezi uživateli. I když je již možné nalézt

telefonní číslo uživatele v detailu inzerátu, chatový systém bude další důležitou součástí komunikace mezi uživateli.

9. Závěr

Ačkoli většina klíčových funkcí aplikace Hodinový Syn je již implementována, stále zbývá několik aspektů, na kterých je třeba pracovat, abych zajistil, že aplikace bude plně funkční a efektivní. Dokončení a vypuštění aplikace plánuji ještě v tomto roce.

10. Zdroje

CTA (Call to Action) – co to je a jak ji napsat. *MarketingPPC* [online]. [cit. 2024-03-19]. Dostupné z: https://www.marketingppc.cz/ppc/co-je-to-cta/

What Is Node.js and Why You Should Use It. *Kinsta* [online]. listopad 2023 [cit. 2024-03-19]. Dostupné z: https://kinsta.com/knowledgebase/what-is-node-js/#nodejs-architecture-and-how-it-works

(M.7) Zvýraznění textu. *IPK* [online]. září 2020 [cit. 2024-03-19]. Dostupné z: https://prirucky.ipk.nkp.cz/pristupnost/manual_zvyrazneni_textu

Lekce 1 - Úvod do TypeScriptu. *Itnetwork.cz* [online]. [cit. 2024-03-19]. Dostupné z: https://www.itnetwork.cz/javascript/typescript/uvod-do-typescriptu

Lekce 3 - Serverové renderování s Next.js. *Itnetwork.cz* [online]. [cit. 2024-03-19]. Dostupné z: https://www.itnetwork.cz/javascript/react/pokrocily/serverove-renderovani-s-nextjs

Cloud Firestore. *Google* [online]. březen 2024 [cit. 2024-03-19]. Dostupné z: https://firebase.google.com/docs/firestore

Lekce 1 - Úvod do React. *Itnetwork.cz* [online]. [cit. 2024-03-24]. Dostupné z: https://www.itnetwork.cz/javascript/react/zaklady/uvod-do-react

11. Seznam obrázků

Obrázek 1 ukázka databáze 6
Obrázek 2ukázka databáze 6
Obrázek 3 ukázka databáze 7
Obrázek 4 přihlašování v aplikaci 12
Obrázek 5 registrace v aplikaci 12
Obrázek 6 hlavní stránka v aplikaci 13
Obrázek 7 vytváření inzerátu v aplikaci 13
Obrázek 8 stránka inzerátů v aplikaci 14
Obrázek 9 detail inzerátů v aplikaci 14
Obrázek 10 moje inzeráty v aplikaci 15
Obrázek 11 profil v aplikaci 15