

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

Školní rok 2020/2021

4.B

DELTA

MATURITNÍ PROJEKT

MMTrading

Příjmení, jméno: Votava Vladimír

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

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Vladimír Votava*

Školní rok: *2020/2021*

Třída: *4.B*

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

Téma práce: *Webový prostředník v obchodu s virtuálními předměty*

Vedoucí práce: *Bc. Vlad'ka Janů*

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

Cílem projektu je vytvoření webové aplikace, která by umožňovala vyměnit virtuální předměty z online her na platformě Steam za reálné peníze bez potřeby prostředníka. Uživatel si bude moct vybrat předměty, které by chtěl prodat a určí si cenu. Server poté vygeneruje link a převezme si předměty od uživatele, po platbě od druhého uživatele dostane druhý uživatel předměty a první uživatel dostane peníze popř. virtuální měnu na stránce. Aplikace by měla sloužit k automatizaci těchto obchodů, které by se normálně museli dělat přes třetí osobu, kde často vznikají podvrhy a krádeže těchto předmětů, které mohou být i velmi drahé.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- návrh backendu a frontendu aplikace (září)
- hotový backend a první prototyp frontend části (říjen - prosinec)
- ladění a funkční aplikace (leden - březen)
- dokumentace (leden - březen)

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze zdroje a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne

Poděkování

Děkuji paní bc. Vladěce Janů za vedení projektu a její odborný pohled z hlediska vývoje.

Anotace

Cílem projektu je vytvořit aplikaci, která bezpečně a pohodlně zprostředkuje výměnu virtuálních předmětů z platformy Steam za reálné peníze.

Klíčová slova

Webová Aplikace; Steam; Javascript; NextJs; Express; React; GraphQL

Annotation

The main goal of this project is to create a web application that would securely and comfortably handle trade with Steam items for real money.

Keywords

Web Application; Steam; Javascript; NextJs; Express; React; GraphQL

OBSAH

1	ÚVOD	7
2	STEAM	8
2.1	SKINY (PŘEDMĚTY)	8
2.2	VÝMĚNA SKINŮ	8
2.3	TRADE LINK	9
2.4	TRADE BAN	10
3	PROBLEMATIKA OBCHODU SE STEAM SKINY	10
3.1	PŘÍMÝ OBCHOD MEZI UŽIVATELI	10
3.2	OBCHOD PŘES TŘETÍ STRANU	11
3.3	ŘEŠENÍ	11
3.4	PROJEKT Z LEGISLATIVNÍ HLEDISKA	12
4	FUNKCE SYSTÉMU	13
4.1	WEBOVÉ ROZHRANÍ	13
4.1.1	ÚVODNÍ STRÁNKA	14
4.1.2	DASHBOARD	14
4.1.3	OFFERS	15
4.1.4	SETTINGS	16
4.1.5	ROZHRANÍ NABÍDKY	17
4.2	PROCES OBCHODU	18
4.2.1	INICIÁLNÍ VYTVOŘENÍ NABÍDKY	18
4.2.2	NABÍDKA BYLA ZRUŠENA	18
4.2.3	BOT ČEKÁ NA TRADE BAN	18
4.2.4	BOT JE PŘIPRAVEN	18
4.2.5	MAJITEL RUŠÍ NABÍDKU	19
4.2.6	KUPUJÍCÍ ZAPLATÍ	19
4.2.7	KUPUJÍCÍ SI VYBÍRÁ PŘEDMĚTY	19
5	SYSTÉM	20
5.1	FRONT-END	20
5.2	BACK-END	21

5.3	ZABEZPEČENÍ	21
5.4	GRAPHQL	22
5.5	DATABÁZE	23
5.5.1	UŽIVATEL	23
5.5.2	OFFER.....	24
5.6	STEAM BOT.....	24
6	TECHNOLOGIE.....	25
6.1	FRONT-END	25
6.1.1	REACT	26
6.1.2	STYLED COMPONENTS.....	26
6.1.3	APOLLO CLIENT	27
6.2	BACK-END.....	27
6.2.1	EXPRESS	27
6.2.2	PASSPORT	27
6.2.3	EXPRESS GRAPHQL	28
6.2.4	MONGODB	29
6.2.5	STEAM API.....	29
6.3	GIT.....	30
7	ZÁVĚR	30
8	ZDROJE.....	31
9	SEZNAM OBRÁZKŮ	32

1 ÚVOD

Obchod s virtuálními předměty se dnes stává realitou, např. bitcoin a jiné virtuální měny také mají svoji hodnotu a dá se s nimi obchodovat jelikož je jich omezené množství. Toto stejné platí u vzácných virtuálních předmětů do různých her. Platforma Steam to však vzala na jinou úroveň a v rámci jejich platformy dovolila jak vývojářům přidávat tyto předměty do svých her tak uživatelům tyto předměty mezi sebou vyměňovat přes Steam trade nebo prodávat na komunitním Steam marketu.

Problém však nastává když si chce uživatel vyměnit tyto předměty za reálné peníze. Potom nastává riziko že bude okraden, jelikož tyto obchody si musí uživatelé zprostředkovávat samy a Steam na to nemá žádnou službu.

Cílem mého projektu je vytvořit uživatelsky jednoduchou a přívětivou webovou aplikaci, která by umožnila bezpečnou výměnu těchto vzácných předmětů za reálné peníze s použitím moderní technologie jako je NextJs a React. Projekt zatím funguje pouze pro hru Dota 2, ale v budoucnu je plánovaná implementace i jiných známých titulů.

2 STEAM

Steam je nejpopulárnější a největší herní platforma na světě vlastněná firmou Valve, která funguje už od roku 2003. Začala jako jednoduchá knihovna her od společnosti Valve, ale postupně se přidávali i jiné tituly a postupně se tak propracoval do verze, kterou známe dnes. Kromě kupování her je zde i plno jiných a zajímavých služeb, které Steam nabízí. Například to jsou uživatelské profily, chatování, workshop který umožňuje komunitě přidávat vlastní obsah do her, ale pro tento projekt jsou nejdůležitější Skiny (virtuální Steam předměty).



Obrázek 1 - Logo Steam

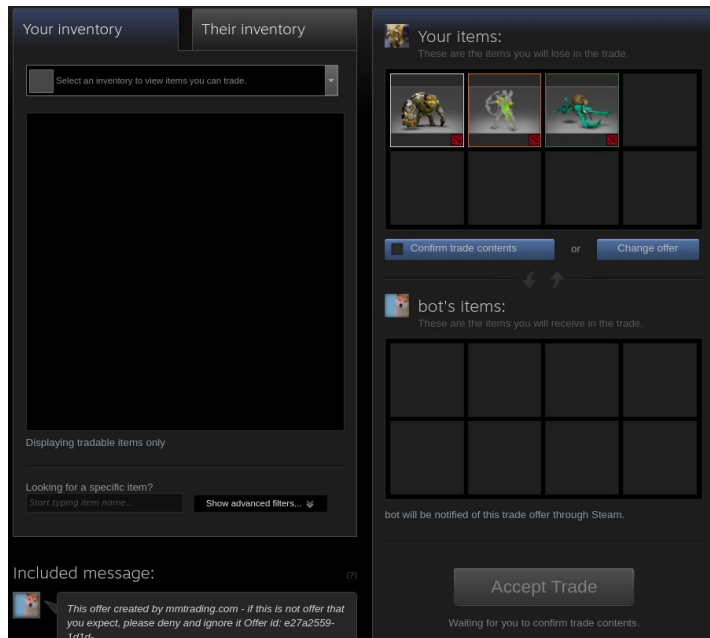
2.1 SKINY (PŘEDMĚTY)

Skiny jsou virtuální předměty, které uživatel může vlastnit a většinou upravují vizuálně např. zbraně nebo jiné předměty ve hře. Skiny byli původně přidány jen pro tituly her od Valve, ale Steam poté umožnil vývojářům přidávat vlastní skiny do jejich her. Skin lze získat vícero způsoby, např. za nějaký milník ve hře (dokončení tutoriálů apod.), ale populárnější jsou dneska tzv. bedny. Hráč si může za nějaký poplatek otevřít bednu z které dostane skin s určitou vzácností. Samozřejmě skiny, na které je menší šance jsou dražší, jelikož je jich méně, ale cenu také může ovlivnit vzhled anebo nějaké speciální efekty předmětu.

2.2 VÝMĚNA SKINŮ

Skiny lze vyměňovat mezi hráči dvěma způsoby. První z nich je služba Steam trade, které umožní uživatelům platformu udělat “výměnný obchod”, např. uživatel-2 má skin co chce uživatel-1, uživatel-1 pošle uživateli-2 Steam trade se skiny co nabízí za skiny které chce, uživatel-2 má poté možnost nabídku přijmout, odmítnout nebo učinit protinabídku. Celý projekt je řízený právě pomocí těchto Steam trade, které zpracovává projektový Steam bot.

Další možnost, jak obchodovat se skiny je Steam market, což je další oficiální služba od Steamu. Jednoduše můžete svůj skin nabídnout za nějakou cenu, nebo můžete nějaký skin koupit. Výhoda oproti Steam tradu je že nemusíte hledat nikoho s konkrétním skinem, prostě si ho můžete koupit. Celý Steam market funguje jako každý jiný online trh, cena je řízené stylem nabídka-poptávka.



Obrázek 2 - Steam Trade

2.3 TRADE LINK

Abyste mohli posílat obchodní nabídky, tak buď musíte být přátelé v rámci platformy Steam anebo potřebujete tkz. trade link. Je to url odkaz, v kterém jsou dva parametry. Jeden z nich je id uživatele a druhý je unikátní token. Uživatel má taky možnost si token zrušit a nechat si vygenerovat nový. Toto je cesta Steamu aby předešla spamu uživatele.

<https://steamcommunity.com/tradeoffer/new/?partner=104887507&token=rYP8tn3i>

TRADE URL STEAM ID TOKEN

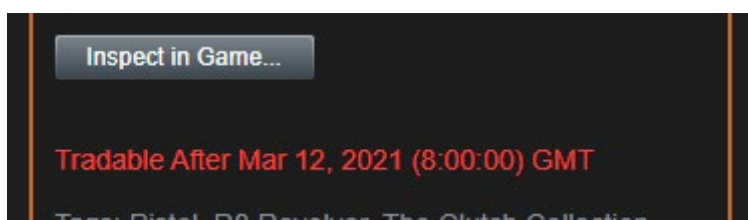
Obrázek 3 - Rozbor Steam Trade Link

Trade link je v tomto projektu důležitý, jelikož by jinak Steam bot nemohl posílat uživatelům automaticky generované obchodní nabídky.

2.4 TRADE BAN

Kvůli rychlému přeprořádání a tím že lidé obchodovali s předměty jako např. s burzou za účelem velkého zisku, což nebyl záměr Steam marketu, tak se Valve rozhodlo přidávat procentuální poplatky za použití Steam marketu a při Steam tradu je na skin udělen trade ban. Pokud jste s někým vyměnili nějaký skin nebo jste ho koupili na Steam marketu, tak na něj bude přidán 7denní trade ban. S tímto předmět se pak 7 dní nemůže obchodovat, tudíž se před další manipulací musí 7 dní počkat.

Tato část je v rámci projektu taky velmi důležitá, jelikož na předměty z hry Dota 2 na kterou je tento projekt zaměřen, byla v průběhu vývoje zavedená stejná pravidla (což při začátku vývoje nebylo) tak musel být projekt poupraven, aby mohl dále fungovat.



Obrázek 4 - Snímek - Trade Ban

3 PROBLEMATIKA OBCHODU SE STEAM SKINY

Steam sice umožňuje získávání, obchod a výměnu předmětů, které mají reálnou hodnotu. Problém je že tyto peníze máte jen v rámci služby Steam, takže si za to můžete koupit novou hru na platformě Steam, ale peníze si už nemůžete vybrat za reálnou měnu. Tudíž pokud si přidám nějaký menší obnost na svojí Steam peněženku, koupím si bednu, ve které otevřu a dostanu z něj skin za 100€ tak i kdybych tento předmět prodal na Steam marketu tak tyto peníze mám jen v rámci služby Steam.

3.1 PŘÍMÝ OBCHOD MEZI UŽIVATELI

Uživatelé co chtěli dostat peníze ze Steamu do reálného světa začali obchodovat tak že se domluvili dvě strany, jedna strana poslala peníze z bankovního účtu prodávajícímu a druhá strana předala skiny přes službu Steam trade. Skin byl většinou prodáván lehce pod cenou, aby se to kupující straně vyplatilo a prodávající za to mohl dostat reálné peníze.

Problém této metody je však to, že často docházelo ke okradení. Jedna strana nedodržela část svojí dohody, takže například nezaplatila za skin, nebo skin vůbec nevydala kupujícímu. Okradení lidé se bohužel nemají jak bránit, jelikož Steam tento druh obchodů oficiálně nepodporuje takže tento styl se nevyužívá.

3.2 OBCHOD PŘES TŘETÍ STRANU

Populární a dodnes používaný styl obchodu se skiny za reálnou měnu je tkz. obchod přes middleman. Stejně jako v předešlém případě jste se domluvili s druhou stranou na výměně a ceně obchodu, ale do toho se ještě zakomponuje tkz. middleman, což je v podstatě člověk, kterého si přes reference najdete na nějakém fóru a ten nejdřív převezme peníze a skiny od obou stran a poté je vydá.

Tímto se zamezuje možnému okradení nebo nedodržení dohody. Problém této metody je že i přesto můžete být okradeni právě middlemanem, ale to se stává málo, protože middleman musíte mít dobrou reputaci a při okrádání zákazníků by si vás už nikdo nenajal.

Další problém je že místo domlouváním pouze s kupujícím musíte ještě komunikovat s middlemanem a dohodnout se na poplatku a ceně za jeho služby a u lepších middlemanů, kteří mají dobrou reputaci a u kterých vše probíhá rychle jsou bohužel větší poplatky. Někdy to jsou až desítky procent z obchodu.

3.3 ŘEŠENÍ

Řešení s kterým jsem přišel já je v podstatě virtuální a automatizovaný middleman. K celé službě MMTrading je udělané jednoduché webové rozhraní kde si uživatel může vše nastavit. Do služby MMTrading se přihlásíte pomocí služby Steam a vytvoříte si nabídku, který pak pošlete kupujícímu. Vše je zautomatizované, takže prodávající nemusí nic domlouvat s další osobou a celá služba je bez poplatků (v rámci business modelu by mohl být implementován velmi malý procentuální poplatek).

3.4 PROJEKT Z LEGISLATIVNÍ HLEDISKA

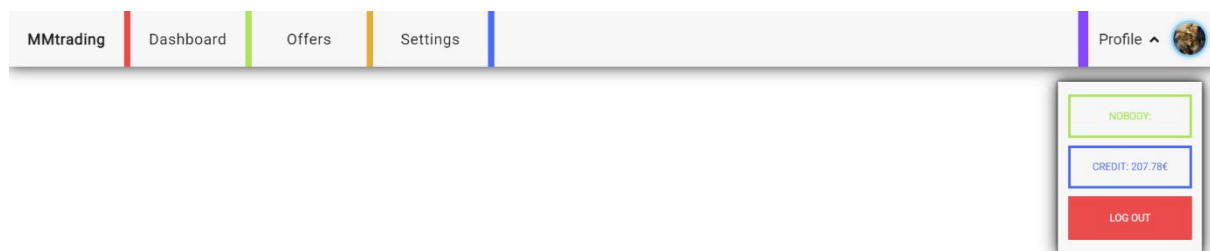
Jelikož projekt zahrnuje výměnu předmětů s reálnou hodnotou a v budoucnu je plánovaná integrace platební brány tak je potřeba brát v potaz i legální stránku tohoto projektu. Steam ve svých podmínkách tyto metody obchodu nezakazuje, ale zároveň nejsou podporovány. Což znamená že je můžete provozovat, ale Steam si za ně nebere žádnou odpovědnost. Steam má také pár podmínek pro služby, které manipulují se Steam předměty (např. neobcházení trade banu). Pokud tyto podmínky porušíte tak vám Steam v lepším případě zabanuje API klíče a Steam boty, v horším na vás mohou podat i žalobu za porušení podmínek.

Ohledně legislativní stránky v České Republice je to vyřešené tak, že používáním služby uživatel konkludentně souhlasí s podmínkami (které jsou dostupné k nahlédnutí na úvodní stránce) s tím že se ze mě stává pouze zprostředkovatel obchodu mezi prodávajícím a kupujícím. Kdy prodávající mi poskytne předmět smlouvy (v tomto případě skiny které si vybral) do úschovy a po obdržení příslušné finanční částky od kupujícího. Kupujícímu předmět vydám a prodávajícímu vyplatím celou zaplacenou částku. Tuto část pak obecně upravuje Občanský zákoník - část čtvrtá "Relativní majetková práva".

4 FUNKCE SYSTÉMU

4.1 WEBOVÉ ROZHRAŇÍ

Celé webové rozhraní je složeno ze čtyř stránek. Všechny jsou uživateli přístupné i když není přihlášený, ale funkčnost stránek se odemkne až po přihlášení, která probíhá přes službu Steam.



Obrázek 5 - Snímek rozhraní - Menu

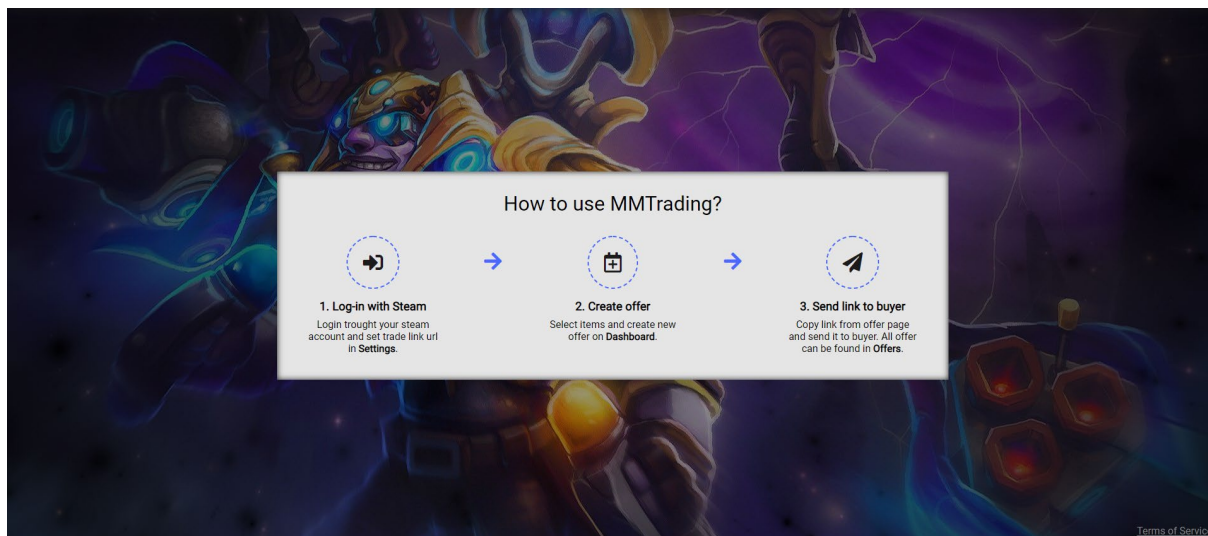
V celém webovém rozhraní se dá pohybovat pomocí navigačního menu. Jsou zde odkazy na všechny stránky aplikace a v pravé části navigačního menu se také nachází tlačítko na přihlášení přes Steam a pokud už je uživatel přihlášen, tak je zde jednoduché dropdown menu. Uživatel tam může najít za jaký účet je přihlášen, svůj kredit a tlačítko na odhlášení.

V celém webovém rozhraní jsou také zavedené toast notifications, což jsou jednoduché notifikace, které např. uživatele upozorní že není přihlášený, nebo mu oznámí že nabídka, kterou chce vytvořit byla úspěšně vytvořena.

4.1.1 ÚVODNÍ STRÁNKA

Úvodní stránka slouží k prezentaci MMTrading službě a zároveň je tam menší návod pro uživatele, jak tato služba funguje. Stránka má naimplementovanou menší funkcionalitu, které spočívá v tom, že při každém načtení se náhodně mění před definované schéma a podle toho se také upraví pozadí.

Na stránce lze také nalézt odkaz na podmínky použití.

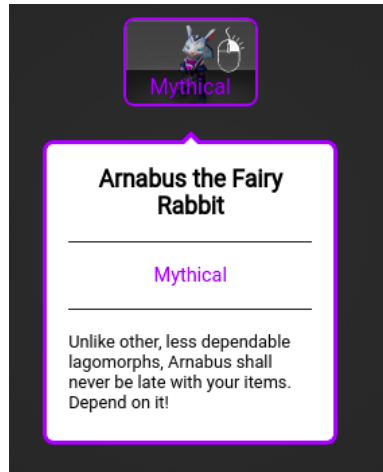


Obrázek 6 - Snímek rozhraní - Úvodní stránka

4.1.2 DASHBOARD

Na stránce Dashboard si uživatel může vytvořit novou nabídku. Pro použití zde musí být uživatel přihlášen. Je zde vytvořeno jednoduché rozhraní kde uživatel vidí svoje skiny co vlastní. Kolonka kde uživatel vidí, které skiny si vybral a pod kolonkou s vybranými skiny je ukázka, jak by objednávka vypadala. Tam si uživatel nastaví cenu nabídky.

Skiny které chce uživatel vybrat do nabídky jednoduše levým kliknutím na skin přesune do předmětů které chce koupit a naopak z kolonky předmětů, které si vybral je může levým kliknutím odebrat. U jednotlivých skinů jde také pravým tlačítkem zobrazit detailní informace, jako je například název, popisek anebo jestli jsou na skinu nějaké speciální efekty.



Obrázek 7 - Snímek rozhraní - Skin a popisek

V kolonce s předměty, které vlastní uživatel je při určitém počtu aktivováno stránkování, které sníží náročnost vykreslování na prohlížeč. Na stránce jsou ještě tři pomocné tlačítka, které umožní uživateli znova načíst skin, přesunout všechny předměty z inventáře do nabídky a nebo odstranit všechny předměty z nabídky.



Obrázek 8 - Snímek rozhraní - Dashboard

4.1.3 OFFERS

Offers je stránka kde uživatel může najít list všech nabídek, okno s nabídkami je při větším počtu nabídek rozděleno na stránky aby se zmírnila náročnost vykreslování. V horní části stránky také můžeme najít přepínání mezi vytvořenými nabídky a koupenými nabídky, také je zde možnost znovu načíst všechny nabídky a aktualizovat jejich stav.

4.1.4 SETTINGS

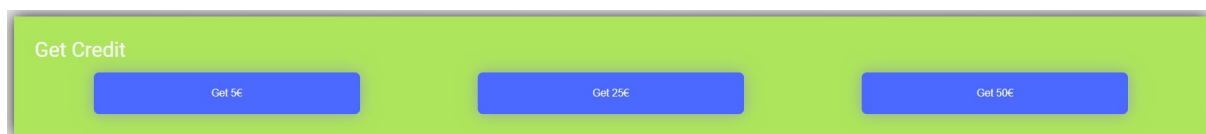
Na stránce settings je nastavení trade url. Pokud jí už má uživatel nastavenou k profilu tak se mu zde načte, ale pokud jí ještě nemá nastavenou nebo se mu trade url změnila tak jí stačí aktualizovat, při problému se přes toast notifikace uživateli zobrazí zpráva co je špatně. (např. že tento token není validní, nebo nepatří přihlášenému uživateli)



Obrázek 9 - Snímek rozhraní - Steam Trad Link Settigns

V okně s nastavením trade url také najdeme tlačítko, které otevře nové okno přímo se Steam stránkou, kde náš trade link najdeme.

Na stránce settings také najdeme okno pro získání kreditů, jelikož zatím aplikace není napojená na platební bránu tak lze získat kredity i bez zaplacení. V budoucnu je integrace platební brány v plánu.

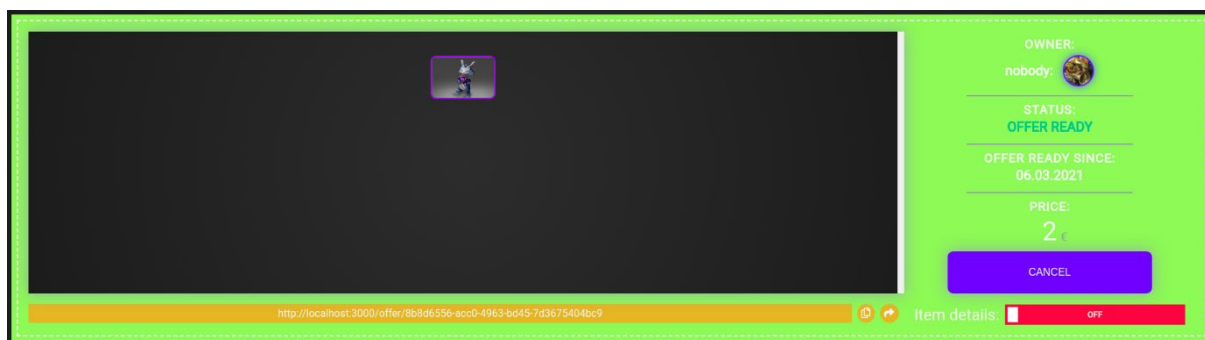


Obrázek 10 - Snímek rozhraní - Kredity

4.1.5 ROZHRANÍ NABÍDKY

Nabídku můžeme najít na stránce Offers, nebo na konkrétním odkazu kde se ukáže pouze jedna nabídka podle ID. Na komponentně se nachází:

- Okno se skinu - co jsou přidáné k nabídce, pokud je nabídka ve stádiu kdy MMTrading Steam bot nemá skinu u sebe (např. je nabídka zrušená nebo je nabídka už dokončená a kupující si skinu už vybral) tak se přes okno s předměty místo předmětů zobrazí hláška podle stavu. (např. “Nabídka již byla dokončena”)
- Link s odkazem - na nabídku u kterého je možnost link zkopírovat přímo do clipboardu nebo se nechat přesměrovat přímo na stránku s konkrétní nabídkou
- Detaily nabídky - jsou zde vypsané informace jako majitel nabídky, stav nabídky, cena a časová informace podle stavu nabídky (např. za jak dlouho vyprší trade ban, nebo kdy byla nabídka dokončena)
- Tlačítko akce - je přístupná pouze při určitém stavu nabídky a podle toho který uživatel je právě přihlášen, defaultně je tlačítko akce disable. (aktivní je např. pokud kupující zaplatí, v tu chvíli jde zde možnost akce “Vybrat předměty”)
- Tlačítko na zapnutí detailů u skinů (ty jsou defaultně vypnuté z náročnosti vykreslování na prohlížeč)



Obrázek 11 - Snímek rozhraní - Nabídka

4.2 PROCES OBCHODU

V této části dokumentace se pokusím vysvětlit, jak funguje proces nabídky. Od jejího vytvoření až po dokončení a jaké role v tom hrají jednotlivé stránky. Počítám s tím, že si už prodávající nastavil platnou trade url na stránce Settings.

4.2.1 INICIÁLNÍ VYTVOŘENÍ NABÍDKY

Prodávající vytvoří nabídku na stránce Dashboard a poté mu Steam bot zašle Steam trade kde žádá o skiny, které si uživatel vybral. Na stránce s nabídkou zatím nejsou vidět vybrané itemy a u času je napsaný za jak dlouho iniciální Steam trade vydrží.

4.2.2 NABÍDKA BYLA ZRUŠENA

Pokud prodávající odmítne Steam trade při iniciálním vytvoření objednávky nebo uběhne časový limit tak se nabídka označí jako zrušená. U kolonky času bude napsaný datum kdy se objednávka zrušila.

4.2.3 BOT ČEKÁ NA TRADE BAN

Po úspěšném převzetí skinů od prodávajícího je na skiny od Steamu udělen 7denní trade ban. Při tomto stavu nabídky se musí čekat 7 dní. Systém každých několik minut kontroluje všechny nabídky, jestli už u nich nevypršel trade ban, poté označí nabídku jako aktivní. V kolonce času je uvedeno za jak dlouho vyprší trade ban.

4.2.4 BOT JE PŘIPRAVEN

Po uplynutí trade banu se nabídka stává aktivní. Je možné s ní dále manipulovat a v kolonce času je uvedeno od kdy s ní jde manipulovat dál. Pokud jste majitel nabídky tak je aktivní action tlačítko, které umožní nabídku zrušit. Pokud jste jiný uživatel tak můžete nabídku koupit.

4.2.5 MAJITEL RUŠÍ NABÍDKU

Pokud jako majitel nabídky zrušíte aktivní nabídku tak vám bot zašle Steam trade s předměty, které byli v nabídce. Zrušení nabídky dokončíte tak že předměty přijmete od Steam bota. Pokud je aktivní Steam trade, kde si majitel ruší nabídku a vybírá si předměty tak nabídka nejde koupit jiným uživatelem.

4.2.6 KUPUJÍCÍ ZAPLATÍ

Pokud si uživatel koupil některou z nabídek, tak se nabídka stane zaplacená a z tohoto uživatele se stává kupující. Kupující tuto nabídku od teď může najít i přes stránku Offers v sekci zakoupených nabídek. Při tomto procesu se převedou kredity od kupujícího k majiteli nabídky. V tuto chvíli jsou předměty pořád u Steam bota, ale majitel ani nikdo jiný už nabídku nemůže koupit ani zrušit.

4.2.7 KUPUJÍCÍ SI VYBÍRÁ PŘEDMĚTY

Po úspěšném zakoupení nabídky je z pozice kupujícího aktivní action tlačítko s možností vybrání předmětů, pokud si kupující zažádá tak mu Steam bot zašle Steam trade s předměty. Po převzetí předmětů od Steam bota se nabídka stává dokončenou a v kolonce času bude datum kdy se tak stalo.

5 SYSTÉM

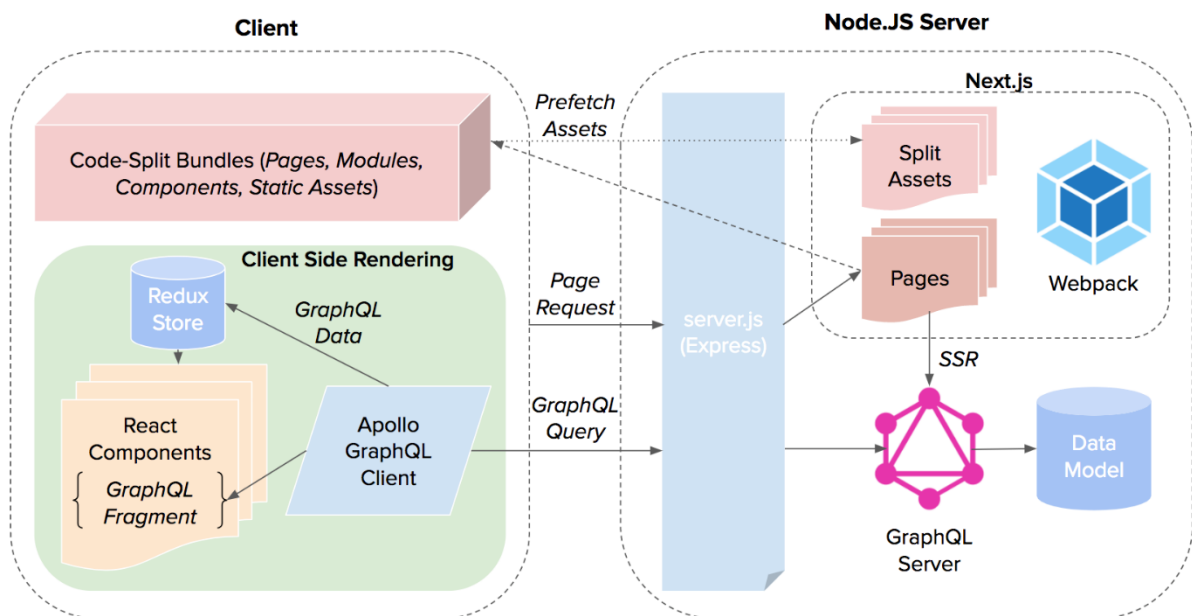
5.1 FRONT-END

Front-end se stará o prezentaci dat z back-endu (např. výpis nabídek vlastněné uživatelem) a také aby uživatel mohl dělat různé akce (např. vytvoření objednávky, nastavení trade url).

Celý front-end je v technologii react a je psaný v typescriptu (typescript je typový jazyk který se pak kompiluje na javascript). Pro front-end bylo vytvořeno přes 20 komponent a 5 stránek a k tomu je zde několik helpUtils, soubory s redundantními typy.

Vizuální aspekt projektu není nijak zajímavý a vytvořil jsem ho podle sebe tak aby to splnilo aspoň funkční účel. Stylizace stránky není hlavní pointa projektu, takže není dělaná podle nějakého css frameworku jako je bootstrap nebo materialize.

Jelikož projekt začal, když jsem ještě jako vývojář neměl žádné zkušenosti s vývojem front-endu a neznal jsem pojmy jako mobile-first tak stránky responzivní nejsou. Pokud bych na projektu pokračoval dál tak by to určitě zahrnovalo předělání designu podle css frameworku, zakomponování responzivity a využití moderní css knihovny jako je tailwind.



Obrázek 12 - Diagram architektury projektu

5.2 BACK-END

Back-end propojuje všechny části aplikace dohromady. K funkčnosti back-endu jsou taky potřeba speciální klíče a přihlašovací údaje, které jsou uschované v souboru s klíči. (nejsou součástí git repozitáře, ale místo toho je tam uvedený soubor se vzory klíčů)

Na back-endu je vytvořen cron job (aktivita která se v určitém cyklu opakuje). Ten kontroluje jestli nabídka, které čeká na vypršení trade banu už může být označena jako aktivní. Každých 10min (čas se dá změnit v konfiguračním souboru) se provede kontrola všech nabídek a u kterých již vypršel trade ban tak nastaví jako aktivní.

5.3 ZABEZPEČENÍ

K zabezpečení projekt využívá přihlášení přes Steam. Pokud se uživatel úspěšně přihlásí tak Steam poskytne back-endu informace o uživateli jako id, jméno nebo avatar. To se pak uloží do databáze MMTrading a při opětovném přihlášení se tyto data při změně (např. jména) zaktualizují.

Aby se uživatel při každém načtení stránky nemusel znova přihlašovat tak dostane od back-endu seassion-id, kterým se pak při každém requestu ověřuje. Všechny seassion-id jsou uloženy v databázi. Pro použití graphql api uživatel musí mít platné seassion-id.

5.4 GRAPHQL

Pro komunikaci s front-endem je na back-endu vytvořen graphql server. Všechny query které graphql api poskytuje naleznete na sub stránce “/api”. Tam si v jednoduchém menu můžete api vyzkoušet a zároveň zde naleznete popisky jednotlivých query. (Pro většinu query se však musíte nejdřív přihlásit)

Jednotlivé query se mohou zdát být jednoduché, ale jelikož je aplikace napojená na Steam tak se musí kontrolovat mnoho, např:

- zda je uživatel přihlášen
- pokud uživatel doopravdy vlastní předměty
- jestli token který uživatel poskytl je validní
- zda může provést akci na konkrétní nabídce
- pokud má uživatel nastavenou správnou trade url

Některá z těchto validačních pravidel se mohou zdát zbytečné, jelikož by se k nim uživatel nikdy neměl dostat, ale základní pravidlo při vývoji webových aplikací je nikdy nebrat data od uživatele (front-endu) jako důvěryhodná, jelikož mohou být zmanipulované.

Všechny query u sebe mají error parametr a pokud nastane chyba tak se v něm pošle kódové číslo, pod kterým pak front-end vypíše chybovou hlášku co se stalo přes notifikace.

5.5 DATABÁZE

Pro ukládání dat používám noSql dokumentovou databázi MongoDB. Ukládají se data o uživateli a o nabídkách. Také se využívá k ukládání session-id's, které si spravuje knihovna express-session.

Manipulace s databází se nedělá napřímo, ale vše je zabalené do asynchronních funkcí, které jsou uloženy v jednom souboru a distribuované po celém projektu. Je to z důvodů přehlednosti, snížení redundantnosti a zlepšení kontroly toku dat

V databázi se používá několik unikátních identifikátorů. Pro uživatele se používá SteamID, které nám poskytne Steam při přihlášení uživatele. Pro nabídku v projektu používá UUID verze 4, kterou nabídka dostane při vytvoření. Pro skiny a Steam trade nám id získává Steam bot a v průběhu nabídky mění a aktualizuje.

Pro čas v nabídce používám standard ISO-8601, který se pak ukládá jako řetězec do databáze.

5.5.1 UŽIVATEL

Uživatele ukládáme do databáze poté co se úspěšně přihlásí přes službu Steam. Ta nám poskytne základní informace o uživateli - id, nickname a avatara. K tomu mu stránka přidá ještě kredity (ze začátku defaultně nastaví 0) a trade url (defaultně prázdný řetězec). Při opětovném přihlášení se zkontroluje jestli nějaký údaj (např. Steam nickname) změnilo a pokud ano tak se aktualizuje i v databázi.

Users	
id	steamUserID
name	string
avatar	stringUrl
credit	number
tradeUrl	stringUrl

Obrázek 13 - User Model

5.5.2 OFFER

Nabídka se prvotně uloží do databáze v první iniciální fázi po vytvoření nabídky uživatelem v dashboardu. Poté se nabídce postupně mění stav, čas a při převzetí předmětů od prodávajícího se musí přepsat id skinů, jelikož při každém Steam tradu se mění.

Offer	
id	uuid
trade_id	steamTradeID
owner_id	steamUserID
buyer_id	steamUserID
items	[steamItemID]
price	number
date	ISODate
status	number

Obrázek 14 - Offer Model

5.6 STEAM BOT

Steam bot je klíčový aspekt v celém projektu, je to v podstatě uživatelský účet na Steamu, který je ovládán přes Steam api (od Steamu je vygenerovaný api klíč který nám umožňuje využívat Steam api). V celém projektu zajišťuje několik klíčových funkcí, proto je potřeba aby byl stále aktivní.

Při načítání stránky dashboard si Steam bot podle id uživatele najde, které předměty vlastní a poté je předá graphql, také zajišťuje chybové hlášky např. pokud má uživatel soukromý profil a z toho důvodu Steam bot nemůže načíst jeho předměty.

Při nastavování nové trade url si Steam bot ověří validnost tokenu u Steamu a také ověřuje jestli daný token opravdu patří uživateli.

Vytváří jednotlivé Steam trade pokud si o ně uživatel zažádá (např. při vytvoření nabídky nebo při výběru zakoupených předmětů). Steam bot si při přebírání skinů u iniciálního vytvoření nové nabídky musí zkontrolovat nové id skinů, jelikož při každém Steam tradu se u skinů změní jejich id. Bot poté musí tyto ID přepsat v i databázi.

Bot má také event listener na Steam trady, které vytvořil. Pokud se změní některá nabídka, kterou odeslal tak si nejdřív zjistí v systému, o kterou nabídku se jedná a podle toho vyhodnotí co s ní dál udělá (např. pokud si kupující vybere předměty které si koupil tak se tato nabídka označí za dokončenou)

6 TECHNOLOGIE

Celý systém je zabalený v NextJs frameworku, který funguje na technologii NodeJs, což je Back-end technologie. NodeJs se programuje v Javascriptu a funguje na V8 enginu od Googlu, který je velmi populární. Celý NodeJs je ještě obalené v c++.

NextJs umožňuje server side rendering aplikace, takže se sníží doba vykreslování a může se tak i zlepšit SEO stránky. NextJs také umožňuje např. custom server nebo routing (přepínání mezi stránky bez načítání nové url).

Celá aplikace je hostovaná na službě Heroku, která umožňuje hosting NodeJs aplikací. Díky jednoduchému propojení se službou Git je implementace a budoucí vývoj a nasazení do produkce velmi jednoduchý.

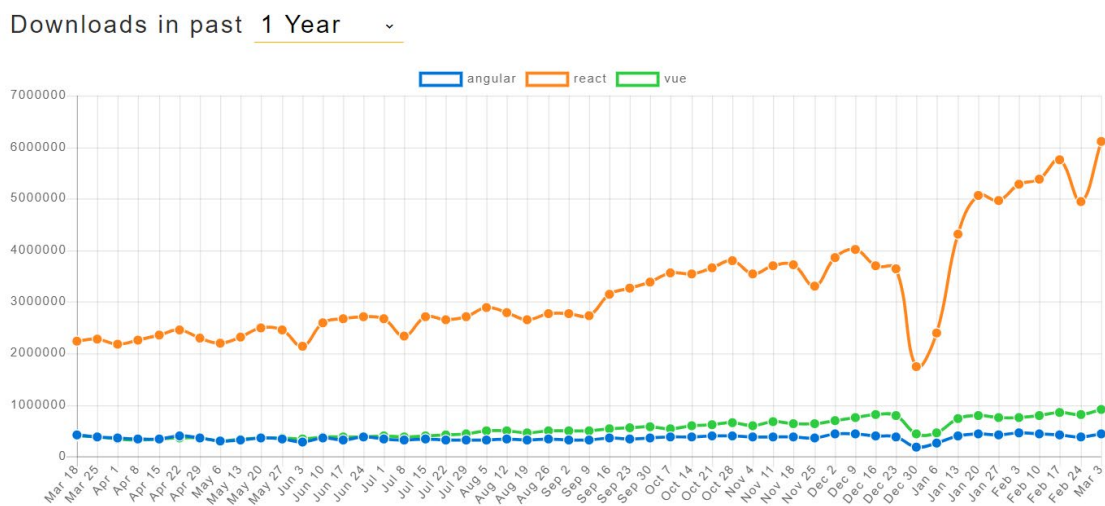
6.1 FRONT-END

Kromě hlavních knihoven jsou v NextJs zakomponované i menší knihovny pro front-end jako je:

- Babel - vezme všechny javascriptové soubory, které jsou psané v moderním javascriptu a převede je do starší verze javascriptu, tak aby se zajistila kompatibilita se staršími prohlížeči
- Webpack - vezme všechny soubory jako .js, .css, .html a převede je tak aby vznikl jen jeden soubor od každého a ten se pak poslal uživateli. Díky tomu se stránka bude načítat rychleji a pro uživatele a potenciálního útočníka se kód bude hůře číst.
- Hot Reload - technologie která se využívá při developementu a umožňuje vývojáři vidět změnu na stránce hned bez reloadu stránky

6.1.1 REACT

React je nejpopulárnější Front-endový framework vyvíjen společností Facebook. Funguje na bázi komponent a moderně se programuje v funkcionálním Javascriptu s využitím hooků. React je beginners friendly, ale zároveň v něm může fungovat i velký komerční projekt. Díky parametrizovatelnosti a znovuvyužití komponent je vývoj velmi rychlý a pohodlný.



Obrázek 15 - Graf porovnání Reactu

6.1.2 STYLED COMPONENTS

Ke stylizaci stránek jsem si vybral Styled Components, jelikož dobře funguje s Reactem.

Funguje na principu komponent, takže hotovou komponentu nebo její část je obalena v Styled komponentně, která má předem nastavené styly.

Jelikož znovupoužitelnost Styled Components jsem nevyžíval, takže jsou všechny styly přímo u jednotlivých komponent. Při větším projektu s jednotným designem je určitě lepší si Styled Componenty psát do jiného souborů a distribuovat po všech komponentách aby byl styl jednotný a snížila se redundandnost.

6.1.3 APOLLO CLIENT

Pro komunikaci Front-endu a Back-endu jsem si vybral Apollo client. Ten se stará o request i response GraphQL requestu na server. Apollo má připravené React hooky, takže je jednoduchý na implementaci do Reactu a podporuje Typescript.



Obrázek 16 - Logo Apollo

6.2 BACK-END

Back-end je vytvořen jako custom express server v NextJs. Při vývoji aplikace jsem chtěl několikrát rozdělit Back-end a NextJs Front-end, ale nakonec jsem se rozhodl z důvodů udržení integrity projektu, jednodušší hosting a jednoduchosti vývoje (jelikož front-end i Back-end fungují na NodeJs) že projekt nechám jako jeden celek.

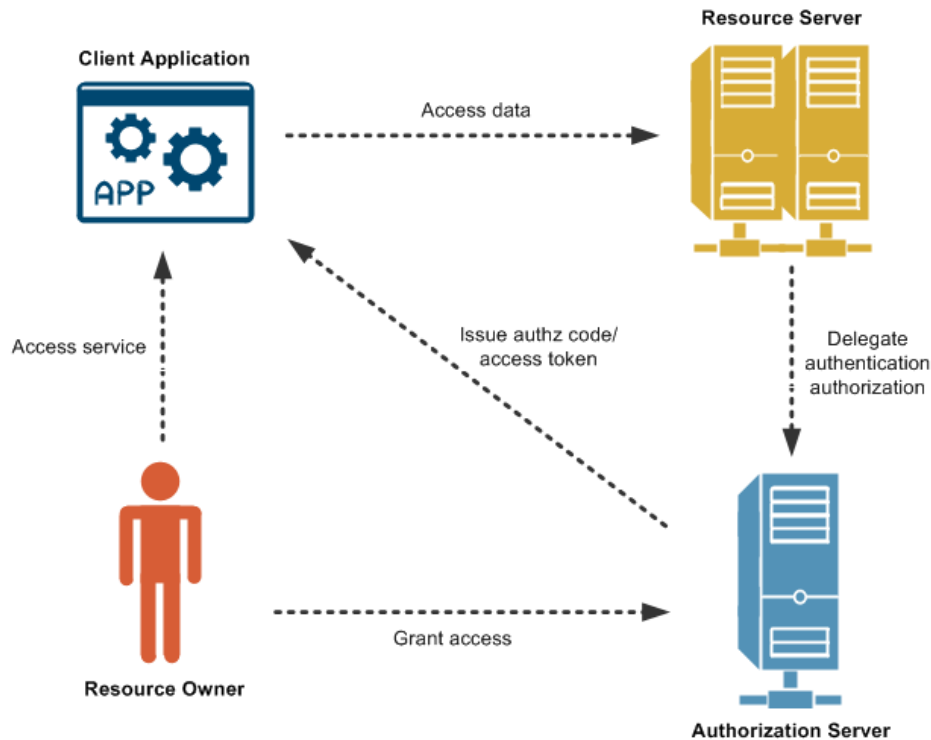
6.2.1 EXPRESS

Express je nejpopulárnější NodeJs web framework, jelikož je minimalistický a jednoduchý na použití. Další důvod proč používám je velká podpora jiných balíčků jako je např. passport nebo express-graphql. NextJs má oficiální podporu pro express custom server. Jednoduše řečeno, předtím než si request převezme NextJs tak se custom server podívá jestli to náhodou není request na předem definované url např. na GraphQL.

6.2.2 PASSPORT

Passport knihovna zajišťuje ověřování uživatele a využívá k tomu technologie OAuth2. Jedna možnost u PassportJs je využít lokální (vlastní) ověření, nebo využít jednu Passport strategy, které knihovna poskytuje. Pro tento projekt se využívá Steam strategy. Passport se tak postará o tokeny a vše ostatní a nám poskytne jen data od uživatele.

Při přihlášení přesměruje uživatele na stránku Steam, kde se uživatel přihlásí a při úspěšném přihlášení Passport zavolá metodu, která vyhledá uživatele v naší databázi podle ID, které dostane od Steamu. Poté tento uživatel dostane session-id díky kterému se uživatel ověří při dalším requestu bez přihlášení.



Obrázek 17 - Diagram OAuth2.0

6.2.3 EXPRESS GRAPHQL

Express-graphql je knihovna která umožní vytvořit GraphQL server na Back-endu. Díky typům a jednoduchému definování objektů je zde vytvořeno několik kombinovaných query.

Každá query má definované jméno, typ, prvky a resolver. Pokud se zavolá query tak se spustí resolver, který např. zavolá z databáze všechny nabídky vlastněné uživatelem a poté data pošle.

6.2.4 MONGODB

Používám dokumentovou noSql databázi MongoDB, je to velmi populární databáze právě v kombinaci s NodeJs.

Celá MongoDB je hostovaná na službě Mongo Atlas, což je oficiálně podporovaná služba od MongoDB. Tam si můžete vybrat od jaké cloudové (AWS, Azure, Google cloud) a kde chcete databázi hostovat. Databázi jde přes jednoduché webové rozhraní spravovat a na propojení s aplikací vám stránka vygeneruje přihlašovací URI.



Obrázek 18 - Logo MongoDB

6.2.5 STEAM API

Na komunikaci se Steamem projekt využívá npm balíček Steam-user. Díky tomuto balíčku se přes jednoduchou api můžeme přihlásit na účet Steam bota, vytvořit Steam trade, nebo si zjistit inventář uživatele.

Jelikož k funkčnosti Steam-user knihovny je potřeba mít účet na Steamu tak v rámci projektu bylo vytvořeno několik účtů, ale aktivně se používá jen jeden. Kromě přihlašovacího jména a hesla musí být v souboru s klíči ještě `shared_secret` a `identity_secret`, tyto údaje slouží např. automatickému dvoufázovému ověření nebo k mobilnímu potvrzení Steam trade.

6.3 GIT

Všechny zdrojové kódy byli při vývoji ukládané na Github. Při stažení projektu je potřeba nainstalovat všechny potřebné knihovny přes příkaz “npm i” a vytvořit soubor klíčů “keys.json” podle vzorového “exampleKeys.json”

Odkaz na git repozitář: <https://github.com/prosteNoBody/mmtrading>



Obrázek 19 - Logo GIT

7 ZÁVĚR

Po práci na tomto projektu můžu s klidem říct, že jsem úspěšně vytvořil platformu, kterou dosud nikdo jiný nevytvořil a může pomoci mnoha lidem. Aplikace umí zařídit bezpečný obchod mezi prodávajícím a kupujícím. Během práce na tomto projektu jsem se naučil plno věcí ze světa vývoje webových aplikací, seznámil se s různými technologiemi a zkusil si vývoj aplikace od začátku až do konce.

Do budoucna bych chtěl určitě přestylovat stránky, zajistit responzivitu stránek a zkusit nasadit i platební bránu a tím zakomponovat i možnost vybrat a přidat si kredity za reálné peníze.

8 ZDROJE

- [1] MEGA KNIHY, „HTML and CSS (anglicky),“ [Online]. Available: <https://www.megaknihy.cz/programovani/20001671-html-and-css.html>. [Přístup získán 15 března 2020].
- [2] GitConnected, „Deploy your Next.js app to Heroku in 5 minutes,“ [Online]. Available: <https://levelup.gitconnected.com/deploy-your-next-js-app-to-heroku-in-5-minutes-255e829a9966>. [Přístup získán 15 března 2020].
- [3] CodingTheSmartWay, „Creating A GraphQL Server With Node.js And Express,“ [Online]. Available: <https://medium.com/codingthesmartway-com-blog/creating-a-graphql-server-with-node-js-and-express-f6dddc5320e1>. [Přístup získán 15 března 2020].
- [4] Vercel, „NextJs Documentation,“ [Online]. Available: <https://nextjs.org/docs>. [Přístup získán 15 března 2020].
- [5] Heroku, „Heroku platform,“ [Online]. Available: <https://www.heroku.com/>. [Přístup získán 15 března 2020].
- [6] mongoDB, „MongoDB Atlas,“ [Online]. Available: <https://www.mongodb.com/cloud/atlas>. [Přístup získán 15 března 2020].
- [7] GitHub, „Andrew's Guide to Steam Bots,“ [Online]. Available: <https://github.com/andrewda/node-steam-guide>. [Přístup získán 15 března 2020].
- [8] Apollo, „Apollo Documentation,“ [Online]. Available: <https://www.apollographql.com/docs/>. [Přístup získán 15 března 2020].
- [9] Steam, „Trading and Market Restrictions,“ [Online]. Available: https://support.steampowered.com/kb_article.php?ref=1047-EDFM-2932. [Přístup získán 15 března 2020].

9 SEZNAM OBRÁZKŮ

Obrázek 1 - Logo Steam	8
Obrázek 2 - Steam Trade	9
Obrázek 3 - Rozbor Steam Trade Link.....	9
Obrázek 4 - Snímek - Ttrade Ban.....	10
Obrázek 5 - Snímek rozhraní - Menu	13
Obrázek 6 - Snímek rozhraní - Úvodní stránka	14
Obrázek 7 - Snímek rozhraní - Skin a popisek	15
Obrázek 8 - Snímek rozhraní - Dashboard	15
Obrázek 9 - Snímek rozhraní - Steam Trad Link Settigns.....	16
Obrázek 10 - Snímek rozhraní - Kredity	16
Obrázek 11 - Snímek rozhraní - Nabídka.....	17
Obrázek 12 - Diagram architektury projektu	20
Obrázek 13 - User Model.....	23
Obrázek 14 - Offer Model	24
Obrázek 15 - Graf porovnání Reactu.....	26
Obrázek 16 - Logo Apollo	27
Obrázek 17 - Diagram OAuth2.0	28
Obrázek 18 - Logo MongoDB	29
Obrázek 19 - Logo GIT	30