

STREDNÁ PRIEMYSELNÁ ŠKOLA INFORMAČNÝCH TECHNOLOGIÍ

Nábřežná 1325, 024 01 Kysucké Nové Mesto

ONLINE HERNÝ OBCHOD

Vlastný projekt na praktickú časť odbornej zložky maturitnej skúšky

Školský rok 2022/2023

Vedúci projektu: Mgr. Zuzana Polláková

Autor projektu: Miloš Bardáč IV.CI

STREDNÁ PRIEMYSELNÁ ŠKOLA INFORMAČNÝCH TECHNOLOGIÍ

Nábrežná 1325, 024 01 Kysucké Nové Mesto

ONLINE HERNÝ OBCHOD

Vlastný projekt na praktickú časť odbornej zložky maturitnej skúšky

Školský rok 2022/2023

Študijný odbor: 2561 M informačné a sieťové technológie

Vedúci projektu: Mgr. Zuzana Polláková

Autor projektu: Miloš Bardáč IV.CI

Čestné prehlásenie

Čestne prehlasujem, že som predkladanú prácu vypracoval samostatne, za použitia uvedenej literatúry. Taktiež vyhlasujem, že danú prácu som neprihlásil a neprezentoval v žiadnej inej súťaži vyhlásenej – MŠVVaŠ SR.

V Kysuckom Novom Meste, dňa

.....

podpis

ABSTRAKT

Táto práca sa zaoberá tvorbou webovej aplikácie využitím frameworkov Laravel, Vue.JS, Bulma a Tailwind CSS. Herný obchod umožňuje používateľom komunikovať s priateľmi pomocou textových správ, nakupovať a recenzovať rôzne hry v digitálnej podobe. Aplikácia je predovšetkým určená pre milovníkov počítačových hier a pre ľudí, ktorí radi trávia čas komunikáciou pomocou textových správ z pohodlia domova len za použitia webového prehliadača.

Kľúčové slová: webová aplikácia, php, javascript, framework, laravel, vue.js, tailwind

Rozsah: 20 s. vrátane príloh, z toho 17 s. textovej časti

ABSTRACT

This project deals with the development of a web application using Laravel, Vue.JS, Bulma and Tailwind CSS frameworks. The game store allows users to communicate with friends using text messages, purchase and review various games in digital form. The application is primarily intended for computer game lovers and for people who like to spend time communicating using text messages from the comfort of their home using only a web browser.

Keywords: web application, php, javascript, framework, laravel, vue.js, tailwind

Size: 20 p. including appendix, 17 p. of main part

OBSAH

0	Úvod.....	8
1	Problematika a prehľad literatúry.....	9
1.1	HTML.....	9
1.2	CSS.....	9
1.3	PHP.....	9
1.4	Laravel.....	10
1.5	Vue.js.....	11
1.6	Bulma.....	11
1.7	Tailwind CSS.....	12
1.8	Stripe.....	12
1.9	MySQL.....	12
1.10	Pusher.....	12
1.11	Figma.....	13
1.12	Visual Studio Code.....	13
2	Ciele práce.....	14
3	Materiál a metodika.....	15
3.1	ER diagram.....	15
3.2	Inštalácia a prvotné nastavenie technológií.....	16
3.3	Migrácie a seeding.....	18
3.4	Modely, šablóny, cesty a riadiace jednotky.....	18
3.5	Verejný, súkromný chat a upozornenia.....	19
3.6	Platobná brána Stripe.....	19
3.7	Viacjazykové rozhranie.....	20
3.8	Dizajn.....	21
3.9	Pridanie aplikácie na webový server.....	21
4	Výsledky práce a diskusia.....	22
5	Závery práce.....	23
6	Zhrnutie.....	24
	Zoznam použitej literatúry.....	25

ZOZNAM TABULIEK, GRAFOV A ILUSTRÁCIÍ

ZOZNAM TABULIEK

Tab. 1 Porovnanie PHP frameworkov v závislosti od počtu požiadaviek za sekundu.....	10
---	----

ZOZNAM ILUSTRÁCIÍ

Obr. 1 Využitie PHP oproti ostatným technológiám.....	10
Obr. 2 Tlačidlá z frameworku Bulma.....	11
Obr. 3 Snímka grafického editora Figma	13
Obr. 4 Snímka editora kódu Visual Studio Code	13
Obr. 5 ER diagram vytvorený pomocou webu drawSQL	16
Obr. 6 Snímka pôvodného súboru .env.....	16
Obr. 7 Koreňový adresár Laravelu.....	17
Obr. 8 Import frameworkov Tailwind CSS a Bulma.....	18
Obr. 9 Vzhľad platobnej brány Stripe	20
Obr. 10 Porovnanie rozdielných jazykov pre hlavnú šablónu.....	20
Obr. 11 Snímka tvorby dizajnu pomocou aplikácie Figma	21

0 ÚVOD

V dnešnej dobe môžeme sledovať rapidný nárast popularity v hernom priemysle. Rozhodli sme sa preto navrhnuť a vytvoriť webovú aplikáciu na tému herný obchod. Práca sa zaoberá návrhom a tvorbou webovej aplikácie za pomoci frameworkov Laravel, Vue.js, Bulma a Tailwind CSS – programovacie jazyky Javascript a PHP. Cieľom tohoto projektu je vytvoriť aplikáciu, pomocou ktorej bude môcť používateľ vyhľadávať, nakupovať a recenzovať hry v digitálnej podobe. Užívateľ bude schopný taktiež uzatvárať priateľstvá s ostatnými participantmi a následne s nimi komunikovať pomocou textových správ v reálnom čase.

Táto dokumentácia je dedikovaná do niekoľkých kapitol, v ktorých sa popisujú teoretické východiská, ciele, samotný návrh a tvorba, výsledky, záver a zhrnutie práce. Prvou kapitolou sú problematika a prehľad literatúry, táto kapitola za použitia citácií informuje o poznatkoch, ktoré boli v danej oblasti už publikované. Za ňou nasleduje ďalšia kapitola, ktorá obsahuje primárne a sekundárne ciele práce.

Nasleduje tretia kapitola, v ktorej sa nachádza samotný opis postupu tvorby aplikácie a naplnenie cieľov práce. Nasledujúcou kapitolou sú výsledky práce a diskusia, táto kapitola zahŕňa naše zistenia a postrehy ku skúmanej problematike. Piatou kapitolou je zhodnotenie dosiahnutých výsledkov a splnenie vytýčených cieľov. V nasledujúcej kapitole sa nachádza popis cieľu práce, súhrn najdôležitejších zistení a výsledkov práce. Na konci sa nachádza zoznam literatúry, ktorú sme pri tvorbe dokumentácie použili.

1 PROBLEMATIKA A PREHĽAD LITERATÚRY

Úlohou tejto práce je navrhnutie a vytvorenie webovej aplikácie na vyhľadávanie, nakupovanie, hodnotenie hier, komunikáciu pomocou verejného alebo súkromného chatu za použitia backend technológie PHP. Pre rýchlejšiu a kvalitnejšiu prácu sme použili PHP framework Laravel a pre frontend javascript framework Vue.js a CSS frameworky Bulma a Tailwind CSS. Pre prácu s databázou sme sa rozhodli pre technológiu MySQL z dôvodu spoľahlivosti a skvelej kompatibility s Laravelom. Ako spôsob platby sme vybrali platobnú bránu využitím technológie Stripe. Pre správnu funkčnosť textových správ a oznámení sme implementovali technológiu Pusher, ktorá v kombinácii s frameworkami Laravel a Vue.js umožňuje zobrazovať dáta v reálnom čase. Nasledujúca časť obsahuje podrobný opis použitých technológií a aplikácií.

1.1 HTML

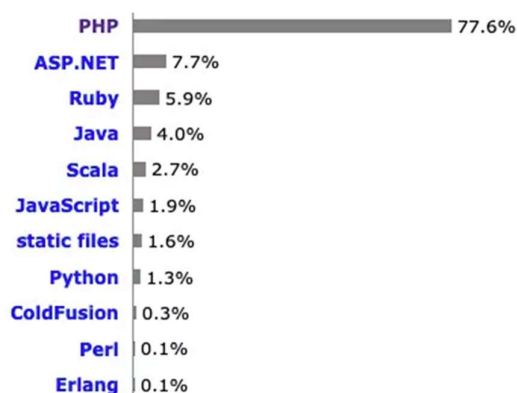
HTML (Hyper Text Markup Language) je hypertextový značkovací jazyk, ktorý sa používa na vytváranie obsahu webovej stránky. Jazyk HTML sa skladá z elementov, ktoré možno použiť na vyjadrenie toho, ako sa má obsah prvku prezentovať používateľovi. Každý prvok má úvodnú značku, obsah, ktorý sa má prezentovať, a zvyčajne aj uzatváraciu značku. Prvky zvyčajne obsahujú atribúty ako napríklad id, class... [15]

1.2 CSS

Pomocou CSS (Cascading Style Sheets) sa štýluje HTML kód, bez ktorého by stránky vyzerali fádne a nezaujímavo. Používa sa najmä na rozmiestnenie jednotlivých prvkov na stránke, štýluje farebnosť textu a pozadia a vyberajú sa fonty pre texty. CSS má jednoduchú konštrukciu a používa určité anglické slová pre určenie vlastností štýlovaného elementu. [3]

1.3 PHP

PHP (Hypertext Preprocessor) patrí medzi najrozšírenejšie, voľne dostupné skriptovacie jazyky pre tvorbu webu. Keďže sa jedná o veľmi populárny jazyk, tak na internete môžeme nájsť veľké množstvo rád a návodov. Taktiež sa nemusíme báť podpory, pretože je podporovaný takmer každým webhostingom. Zároveň táto technológia umožňuje relatívne ľahké programovanie aplikácií na strane serveru. [11]



Obr. 1 Využitie PHP oproti ostatným technológiám [4]

1.4 Laravel

Framework (Rámec)

Framework je rozsiahla sada vývojových nástrojov, ktoré pomáhajú a definujú, ako má programátor vyvíjať aplikáciu. Pomocou rámca programátor používa, upravuje a rozširuje určité hotové softvérové komponenty, čo vedie k efektívnejšej a jednoduchšie vykonanej práci. [7]

Laravel je kompletný framework pre programovací jazyk PHP, ktorý výrazne zjednodušuje tvorbu webových aplikácií. Ide o rýchly, kvalitne objektovo navrhnutý a zároveň vo svete najrozšírenejší PHP framework. Vďaka svojej popularite existuje aj veľa návodov a obsiahla dokumentácia na zvládnutie tohto programovacieho jazyka. Sú na ňom postavené väčšie i menšie webové aplikácie, ako napríklad: Twitch, BBC, Pfizer, 9GAG a mnoho ďalších.

Tab. 1 Porovnanie PHP frameworkov v závislosti od počtu požiadaviek za sekundu [14]

Framework / verzia PHP	PHP 7.3	PHP 8.0
Laravel	574.67	623.78
Wordpress	189.14	233.40
Symfony	529.06	563.90

Kontroléry (riadenie)

Kontrolér je časť, ktorej používateľ odovzdáva parametre a ona mu vráti dáta. Kontrolér typicky parametre odovzdá modelom, od ktorých získa dáta. Tieto dáta odovzdá pohľadom (views), ktoré dáta začlení do nejakého HTML kódu. Tento HTML kód pošle kontrolér užívateľovi do prehliadača.

Modely (logika)

Modely obsahujú logiku aplikácie, ako napríklad prácu s databázou alebo rôzne výpočty. Každá dátová entita napr. užívateľ, hra či správa, by mala mať svoj model.

Pohľady (výstup)

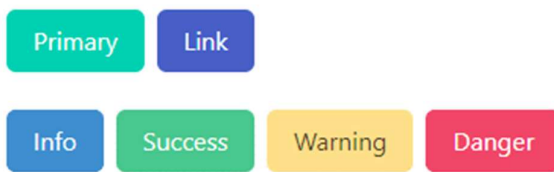
Obsahujú Blade šablóny (s HTML kódom). Blade je šablónovací systém, ktorý na rozdiel od ostatných systémov nezakazuje používanie PHP kódu. Blade používa špeciálna syntax, ktorá je potom prevedená na normálny PHP kód, a zároveň tento vygenerovaný pohľad je následne uložený do medzipamäte pre budúce použitie kvôli rýchlejšiemu načítaniu. [8]

1.5 Vue.js

Vue.js patrí medzi najpopulárnejšie javascript frameworky pre tvorbu jednostránkových a reaktívnych stránok. Veľká výhoda Vue spočíva v tom, že je oproti konkurencii minimalistický, rýchly a pomerne jednoduchý na naučenie. Na internete je možné nájsť množstvo kvalitne spracovanej dokumentácie a návodov pre prácu s Vue. [12]

1.6 Bulma

Bulma je bezplatný framework s voľne dostupným zdrojovým kódom, ktorý poskytuje hotové komponenty frontendov, ktoré môžete jednoducho kombinovať a vytvárať tak responzívne webové rozhrania. Nie je preto potrebná ani znalosť kaskádových štýlov. [5]



Obr. 2 Tlačidlá z frameworku Bulma

1.7 Tailwind CSS

Framework Tailwind umožňuje písať CSS vo forme skratiek priamo do HTML kódu. Zatiaľ čo iný populárny framework Bulma poskytuje základné nastavenia typografie a predpripravené komponenty, Tailwind má iný prístup a ponúka len všeobecné užitočné triedy na kompletné vlastné štylizovanie. Hlavná výhoda spočíva v urýchlení práce vývojára. [13]

1.8 Stripe

Stripe je externé platobné riešenie, ktoré umožňuje internetovým obchodom prijímať platby platobnou kartou. Po vyplnení potrebných údajov je možné cez Stripe prijímať platby online. Služba Stripe je využívaná ako v Česku, Slovensku, tak i na celom svete. Samotné nastavenie účtu Stripe je dostupné iba v anglickom jazyku, je však pomerne veľmi jednoduché a intuitívne. Zákazníci vlastný Stripe účet nepotrebujú, Stripe je platobnou bránou, stačí teda, pokiaľ majú k dispozícii kreditnú alebo debetnú kartu Visa, Mastercard či Maestro. [1]

1.9 MySQL

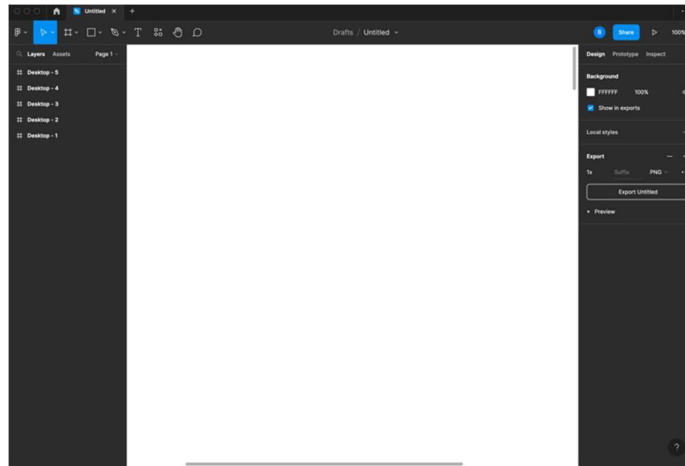
MySQL je voľne dostupný systém pre správu relačnej databázy (RDBMS), ktorý pomocou príkazov umožňuje ukladanie a správu dát v databáze. Správa dát je možná aj za pomoci niektorých skriptovacích jazykov, predovšetkým pomocou jazyka PHP. MySQL je veľmi široko používaný pri tvorbe webových aplikácií vďaka svojej výkonnosti a stabilite. [16]

1.10 Pusher

Pusher je hostovaná služba API, ktorá umožňuje bezproblémové pridávanie údajov a funkcií v reálnom čase do webových alebo mobilných aplikácií. Funguje ako komunikačná vrstva v reálnom čase medzi klientom a serverom. V porovnaní s podobnými službami, kde musíte prvotné spustenie spravovať sami, Pusher vyžaduje menej úkonov na spustenie. V skratke táto technológia umožňuje rýchlo a spoľahlivo zobrazovať dáta ako napríklad správy a upozornenia v reálnom čase. [10]

1.11 Figma

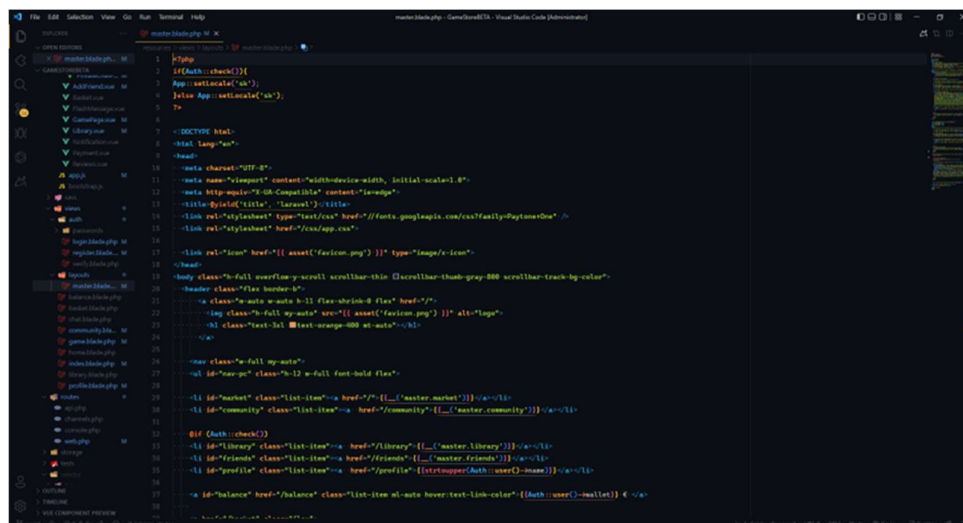
Figma je grafický editor určený najmä pre webdesign, ktorý umožňuje tvorbu interaktívnych prototypov webových stránok, prvkov rozhrania (tlačidlá, ikony, okna) alebo vektorových ilustrácií. Táto služba je dostupná vo forme počítačovej alebo webovej aplikácie. [2]



Obr. 3 Snímka grafického editora Figma

1.12 Visual Studio Code

Visual Studio Code je bezplatný, jednoduchý a výkonný editor kódu. Má zabudovanú podporu pre jazyky JavaScript, TypeScript a Node.js, ale vďaka jeho rozšíreniam si môžeme pridať podporu na drvivú väčšinu moderných technológií vrátane PHP a Vue.js. Obsahuje taktiež vstavaný terminál, ktorý umožňuje efektívnejšie vkladanie príkazov. [9]



Obr. 4 Snímka editora kódu Visual Studio Code

2 CIELE PRÁCE

Hlavné ciele:

- Navrhnutie a vytvorenie webovej aplikácie pre nákup počítačových hier
- Publikovanie aplikácie na vlastnú webovú doménu
- Implementácia platobnej brány

Vedľajšie ciele:

- Vytvorenie viacjazyčného rozhrania
- Realizácia textovej komunikácie v reálnom čase

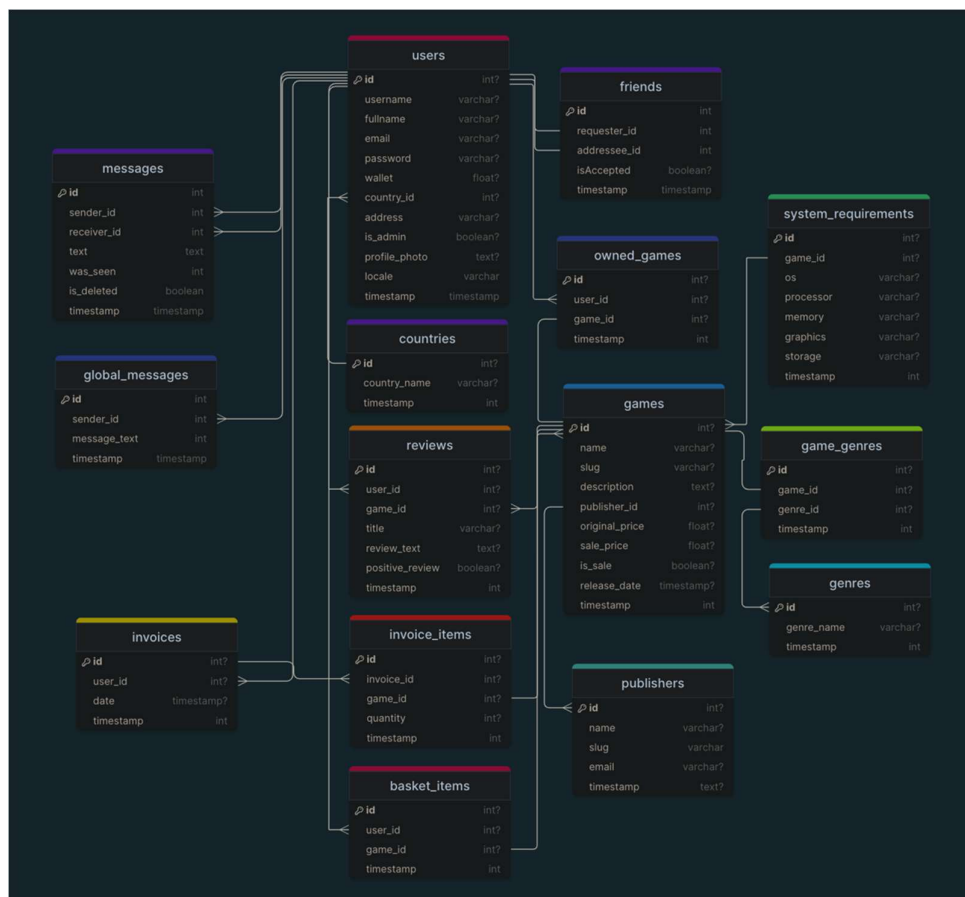
3 MATERIÁL A METODIKA

Táto kapitola obsahuje opis postupu pri práci, ktorý je potrebný k dosiahnutiu zadaných cieľov. Tento postup zahŕňa návrh a vyhotovenie databázy, inštaláciu a prvotné nastavenie potrebných technológií, vytvorenie databázy pomocou migrácií, prvotné naplnenie pomocou funkcie seeder, vytvorenie modelov, ktoré nám umožňujú nastaviť vzťahy medzi údajmi v databáze a tým uľahčiť prácu s dátami. Ďalej tento postup obsahuje nastavenie ciest a prepojenie ich s jednotlivými podstránkami. Pre každú z týchto podstránok je potrebné vytvorenie Blade šablóny, do ktorej je písaný samotný obsah danej stránky. Dôležitou súčasťou projektu sú taktiež controllers, ktoré majú za úlohu spracovať dáta a následne vrátiť výsledok. Pomocou nich je možné na stránku odosielať požadované dáta. Použitím controllers spolu s Vue.js možno pridať webu funkčnosť a reaktivitu. Nasledovne je potrebné upraviť vzhľad a vykonať posledné úpravy webovej aplikácie. Posledným krokom je pridanie aplikácie na server a priradenie vlastnej webovej domény.

V nasledujúcej časti sa nachádza podrobný opis postupu práce k tvorbe webovej aplikácie a splneniu stanovených cieľov.

3.1 ER diagram

Keďže základom našej stránky sú používatelia a hry, našim prvým cieľom bolo zostrojiť návrh databázy alebo taktiež ER (entity-relationship) diagram, ktorý zjednoduší nasledovnú prácu s databázou. Na tvorbu modelu databázy sme použili bezplatnú webovú aplikáciu drawSQL, z dôvodu jednoduchého ovládania a pekného vzhľadu. Aj keď bezplatná verzia tejto služby poskytuje vytvorenie iba maximálne pätnástich tabuliek, toto obmedzenie pre nás nespôsobovalo žiadne obmedzenie, keďže naša databáza spĺňa daný limit. Následne sme začali s návrhom samotného systému. Tento systém obsahuje tabuľky pre užívateľov, hry, vydavateľov hier, súkromné a verejné správy, vlastnené hry, recenzie, zoznam objednávok, zoznam štátov, priateľov, herné žánre či položky v nákupnom košíku. Každé z týchto tabuliek sme priradili stĺpce a ich vlastnosti. Následne sme pomocou nich upravili vzťahy medzi samotnými tabuľkami.



Obr. 5 ER diagram vytvorený pomocou webu drawSQL

3.2 Inštalácia a prvočné nastavenie technológií

V prvom rade je potrebné nainštalovať a nastaviť voľne dostupnú aplikáciu XAMPP, ktorá obsahuje webový server Apache. Ten nám umožňuje pohodlne pracovať s lokálnou databázou MySQL. Následne sme vykonali inštaláciu bezplatného editora kódu Visual Studio Code. Keďže náš inštalačný program prevedie používateľa krok po kroku inštaláciou, tak obe tieto inštalácie sú veľmi jednoduché. Ďalším krokom bola registrácia do služieb Stripe, Pusher a následné vytvorenie aplikácií na oficiálnych webových stránkach služieb. Na inštaláciu samotných technológií sme použili vstavaný terminál v programe Visual Studio Code. Potrebná bola inštalácia najnovšej verzie PHP, táto inštalácia sa skladá z rozbalenia súboru na zvolené miesto, a nastavenia požadovaných systémových premenných php. Potom sme vykonali inštaláciu Laravel a inicializáciu projektu zadáním príkazov do vstavaného príkazového riadka.

```
$ composer global require "laravel/installer"
```

```
$ composer create-project laravel/laravel online-herny-obchod
```


Po vytvorení Laravel projektu sme prešli na nastavenie potrebných premenných, ktoré sa nachádzajú v projektovom súbore s názvom `.env`. V tomto súbore sa nachádzajú vopred vyplnené premenné, pre zmenu týchto premenných stačí len prepísať pôvodnú hodnotu.

```
GameStoreBETA > .env
1  APP_NAME=Laravel
2  APP_ENV=local
3  APP_KEY=
4  APP_DEBUG=true
5  APP_URL=http://localhost
6
7  LOG_CHANNEL=stack
8  LOG_DEPRECATIONS_CHANNEL=null
9  LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
```

Obr. 6 Snímka pôvodného súboru `.env`

```
# The app Directory
# The bootstrap Directory
# The config Directory
# The database Directory
# The lang Directory
# The public Directory
# The resources Directory
# The routes Directory
# The storage Directory
# The tests Directory
# The vendor Directory
```

Obr. 7 Koreňový adresár Laravelu

Do riadkov s databázou sme zadali údaje z lokálneho webového servera, ktorá beží v aplikácii XAMPP. Ako ovládač vysielania sme nastavili pusher a následne sme nastavili ostatné premenné služieb Pusher a Stripe, tieto údaje sme skopírovali z aplikácií, ktoré sme vytvorili na oficiálnych stránkach týchto služieb. Pre inštaláciu ostávajúcich technológií je potrebné nainštalovať knižnicu súborov npm.

```
$ npm install && npm run dev
```

Po úspešnej inštalácii sme prešli na inštaláciu knižnice, ktorá obsahuje rámec Vue.js a autentifikáciu používateľa.

```
$ php artisan ui vue --auth
```

Ďalej prišla na rad inštalácia rámcov Tailwind CSS a Bulma, zadaním príkazov.

```
$ npm install -D tailwindcss postcss autoprefixer
```

```
$ npx tailwindcss init
```

```
$ npm install bulma
```

Následne sme v novovytvorenom súbore s názvom `tailwind.config.js`, pridali cesty k súborom v ktorých chceme pracovať s frameworkom. Potom sme do CSS súboru s názvom `app.css` pridali kód, ktorý určí v ktorom súbore má rámec pracovať. Zároveň sme pridali riadok kódu, ktorý ma za úlohu importovať framework Bulma.

```
@tailwind base;
@tailwind components;
@tailwind utilities;
@import 'bulma/css/bulma.min.css';
```

Obr. 8 Import frameworkow Tailwind CSS a Bulma

Posledným krokom inicializácie je spustenie procesu `watch`, ktorý bude neustále kontrolovať zmeny v projekte. Tento proces slúži na správny chod Tailwind CSS.

```
$ npm run watch
```

3.3 Migrácie a seeding

Ďalšou časťou bolo vytvorenie migrácií, ktoré nám zabezpečili bezproblémové vytvorenie databázy. Súbor s migráciami sa nachádza v adresári `database`. Pre každú tabuľku v databáze sme vytvorili migráciu pomocou príkazu a následne sme migráciu naplnili za pomoci ER diagramu.

```
$ php artisan make:migration "názov tabuľky"
```

Následne sme vytvorili seeder pre užívateľov, ktorý naplnil databázu náhodnými údajmi. Toto nám ušetrilo značné množstvo času pri tvorbe aplikácie, pretože nebolo nutné ručné vytváranie riadkov. Pre nemenné tabuľky sme ručne pridali požadované dáta. A následne sme príkazom spustili migrácie a seeding.

```
$ php artisan migrate:refresh --seed
```

3.4 Modely, šablóny, cesty a riadiace jednotky

Po vytvorení a čiastočnom naplnení databázy sme vytvorili modely, pomocou ktorých sme upravili vzťahy medzi tabuľkami a tým si umožnili jednoduchšiu prácu s dátami. Podadresár s modelmi sa nachádza v hlavnom adresári `app`. Ďalej sme vytvorili hlavnú šablónu, ktorú bude obsahovať každá podstránka. Táto šablóna obsahuje napríklad navigáciu, nastavenie ikony pre stránku, tlačidlo pre verejný chat... Potom sme pre každú

podstránku vytvorili podšablónu, ktorá sa vložila do hlavnej šablóny. Následne sme sa pustili do nastavenia ciest a riadiacich jednotiek. Pre každú z podstránok sme vytvorili cestu, pomocou ktorej sa užívateľ vie dostať na dané miesto. Vytvorili sme hlavný controller pre stránky, ktorý sa stará o to aká šablóna sa použije a aké dáta danej šablóny odošle. Po dokončení hlavnej riadiacej jednotky, sme vytvorili controller pre platby, viacjazyčné rozhranie a taktiež riadiacu jednotku pre každý z modelov. Cestu sme priradili aj riadiacim jednotkám, ktoré to pre budúcu prácu vyžadujú.

3.5 Verejný, súkromný chat a upozornenia

Na tvorbu textovej komunikácie v reálnom čase sme sa rozhodli použiť službu Pusher. Výhodou tejto technológie je pomerne jednoduchá konfigurácia. Pre prácu s API sme potrebovali vysielat' dáta pomocou funkcie Laravelu. Ako prvé sme si vytvorili súkromný kanál pre súkromnú konverzáciu a verejný pre verejnú komunikáciu. Adresár s kanálmi sa nachádza v rovnakom adresári ako cesty. Následne sme vytvorili eventy, ktoré majú za úlohu prijať dáta, dáta spracovať, výsledok nasledovne vložiť do databázy a zároveň pomocou vysielania odoslať výsledok do API služby Pusher. Potom sme si v šablóne pre konverzáciu vytvorili Vue.js komponent, ktorý ma za úlohu neustále kontrolovať či sa v službe objavili nové dáta a v takom prípade zobrazí správu na stránke. Podobný princíp ako sme použili pri správach použijeme pri upozorneniach. Pri vykonaní požadovanej akcie, odošleme dáta a pomocou hlavného javascript súboru dáta zobrazíme na stránke. V tejto časti sme sa stretli s veľkým počtom rôznych problémov, v podobe nesprávnej funkčnosti zo strany služby Pusher.

3.6 Platobná brána Stripe

Ako spôsob platby sme si vybrali platobnú bránu od spoločnosti Stripe. Táto služba poskytuje rýchlu, pomerne jednoduchú a zároveň bezpečnú možnosť úhrady rôznymi spôsobmi. Pomocou Stripe javascript knižnice sme si vytvorili tlačidlo, ktoré vytvorí unikátne číslo platby a spolu s čiastkou, ktorú chceme vložiť na konto nás bezpečne presmeruje k samotnej platbe. Tu si zvolíme typ úhrady a zadáme požadované údaje. Tieto údaje sa v riadiacej jednotke spracujú a v prípade, že sú správne prebehne platba a na účet sa pripočíta zvolená čiastka. Následne sme vytvorili logiku nakupovania hier. Pomocou tlačidla sa pridá hra do nákupného košíka. Po kliknutí na tlačidlo uhradiť sa vykoná funkcia, ktorá skontroluje či máme na konte dostatok financií. V prípade, že všetko prebehne bez problémov pridajú sa nám zvolené hry do knižnice s hrami.

← TEST MODE

Order - Blast
€10.00

Contact information

Email

Payment method

Card IDEAL Bancontact ...

Card information

1234 1234 1234 1234 VISA ...

MM / YY CVC

Name on card

Country or region

Slovakia

Securely save my information for 1-click checkout
Pay faster on this site and everywhere Link is accepted.

0912 123 456 Optional

link · More info

Powered by stripe Terms Privacy

Pay

Obr. 9 Vzhľad platobnej brány Stripe

3.7 Viacjazykové rozhranie

Jeden zo stanovených cieľov bol urobiť viacjazyčnú aplikáciu. Na tento účel sme použili funkciu, ktorú ponúka sám Laravel. Predvolene obsahuje adresár lang, do ktorého sme vytvorili podadresáre s požadovanými jazykmi. Pre každý jazyk sme vytvorili súbor s obsahom fráz v danom jazyku, ktoré šablóna obsahuje. Toto sme vykonali pre každú šablónu. Následne sme vytvorili funkciu, ktorá ako vstupné parametre dostane jazyk a šablónu. Výsledkom tejto funkcie je zoznam fráz v danom jazyku, ktoré daná šablóna obsahuje. Túto funkciu sme implementovali pre každú podstránku.

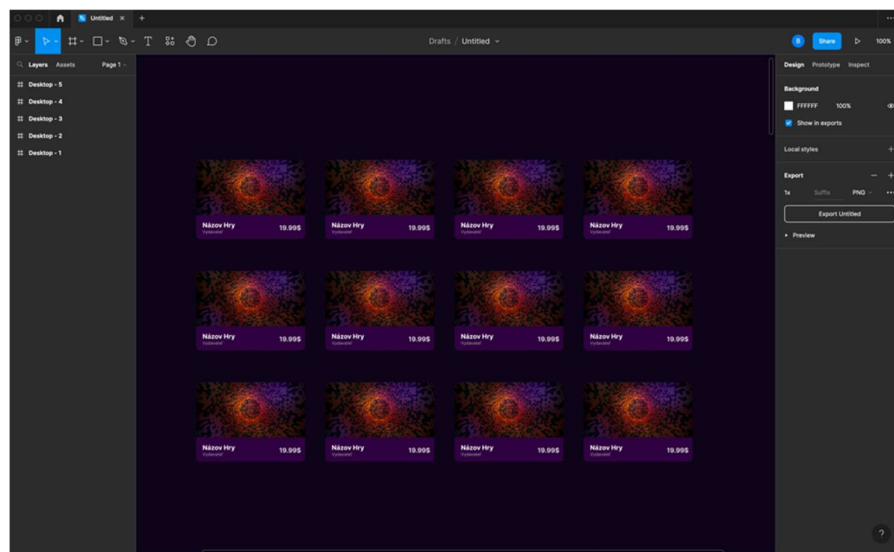
```
<?php
return [
    'market' => 'MARKET',
    'community' => 'COMMUNITY',
    'library' => 'LIBRARY',
    'friends' => 'FRIENDS',
    'logout' => 'LOGOUT',
    'globalchat' => 'GLOBAL CHAT',
    'login' => 'LOGIN',
    'register' => 'REGISTER',
]
?>
```

```
<?php
return [
    'market' => 'OBCHOD',
    'community' => 'KOMUNITA',
    'library' => 'KNIŽNICA',
    'friends' => 'PRIATELIA',
    'logout' => 'ODHLÁSIŤ SA',
    'globalchat' => 'VEREJNÝ CHAT',
    'login' => 'PRIHLÁSIŤ SA',
    'register' => 'VYTVORIŤ ÚČET',
]
?>
```

Obr. 10 Porovnanie rozdielných jazykov pre hlavnú šablónu

3.8 Dizajn

Potom čo sme všetky potrebné dáta dostali na stránku, prešli sme na návrh a samotnú tvorbu dizajnu. Pre návrh dizajnu sme použili webovú aplikáciu Figma. Po pomerne dlhom skúšaní sme vyhotovili návrh pre všetky podstránky. Ako hlavnú farbu stránky sme zvolili tmavo fialovú. Následne sme prešli na úpravu vzhľadu webovej aplikácie pomocou kaskádových štýlov. Primárne sme z dôvodu ľahšieho a rýchlejšieho zápisu používali Tailwind. Veľmi zriedkavo sme použili aj framework Bulma.



Obr. 11 Snímka tvorby dizajnu pomocou aplikácie Figma

3.9 Pridanie aplikácie na webový server

Keďže sme mali prístup k VPS Linuxovému serveru od spoločnosti Oracle, tak sme sa rozhodli práve pre túto možnosť. Funkčnosť aplikácie sme docielili takmer rovnako ako na lokálnom webovom serveri. Keďže server bol pripravený na beh webovej aplikácie, tak mal všetky potrebné technológie už predinštalované a pripravené na správny chod. Stačilo teda len pomocou vzdialeného prístupu nahráť náš Laravel projekt. Po úspešnom nahraní projektu sme potrebovali spustiť migrácie spoločne so seederom. Vďaka tomu sme mohli vytvoriť tabuľky v databáze a naplniť ich základnými dátami. Tento krok bol kľúčový pre správny beh našej aplikácie, pretože zabezpečil, že naše aplikácie môžu vytvárať, čítať a aktualizovať údaje v databáze. Posledným krokom v tomto procese bolo jednoduché reštartovanie webového servera. Po úspešnom reštartovaní servera sme mohli pristúpiť k našej aplikácii pomocou prehliadača a overiť si jej funkčnosť na vzdialenom serveri.

4 VÝSLEDKY PRÁCE A DISKUSIA

Výsledkom našej práce je plne funkčná webová aplikácia s korektne pracujúcou platobnou bránou, možnosťou nákupu, recenzentstiev rôznych herných titulov, možnosťou vytvárať priateľstva a následne komunikovať prostredníctvom chatu v reálnom čase. Aplikácia taktiež obsahuje bezpečnú a funkčnú autentifikáciu, nákupný košík, knižnicu vlastnených hier a upozornenia v reálnom čase.

Popri práci na webovej aplikácii sme boli konfrontovaní s množstvom banálnych, či signifikantných problémov. Medzi pomerne veľkú komplikáciu s ktorou sme sa stretli, bola nie tak dobrá znalosť v oblasti tvorby dizajnu. Celý proces tvorby nám zabral mnoho desiatok hodín, pričom najviac času sme venovali návrhu a tvorbe databázy. Počas tvorby aplikácie nám bola do vysokej miery nápomocná znalosť programovacích jazykov, nadobudnutá pri tvorbe projektov v minulosti.

Pri porovnávaní našej práce s inými simultánnymi webovými aplikáciami sme zaregistrovali, že v diferencii s konkurenciou sme dosiahli porovnateľnú úroveň kvality a funkčnosti. Naša práca ponúka užívateľom mnoho užitočných funkcií a vlastností, ktoré sú pre širokú škálu jedincov príjemné a jednoducho použiteľné.

5 ZÁVERY PRÁCE

Cieľom našej práce bolo navrhnuť a zostrojiť plne funkčnú webovú aplikáciu za pomoci frameworkov Laravel a Vue.js. Aplikácia mala obsahovať bezpečnú autentifikáciu, možnosť nákupu hier, možnosť vytvárať priateľstva, súkromnú a verejnú komunikáciu pomocou chatu a platobnú bránu.

Pri práci na webovej aplikácii sme postupovali v krokoch, ktoré sme si vopred vytýčili. V procese tvorby nám skomplikovali realizáciu mnohé problémy, ktoré sme postupne vyriešili. Pri rekapitulácii našej práce sme sa zhodli, že najobťažnejšia vec, s ktorou sme sa stretli bola tvorba prvkov, ktoré vyžadujú zmeny v reálnom čase. Aj keď v procese tvorby nastalo mnoho problémov tak, sa nám všetky hlavné aj vedľajšie stanovené ciele podarilo splniť. Môžeme tvrdiť že finálny výstup práce sa zhoduje s našimi počiatočnými predstavami.

6 ZHRNUTIE

Výsledkom našej práce je plne funkčná webová aplikácia, ktorá umožňuje používateľom nakupovať a hodnotiť herné tituly, vytvárať si priateľstvá a komunikovať v reálnom čase prostredníctvom chatu. Aplikácia zahŕňa aj bezpečnú autentifikáciu, nákupný košík, knižnicu vlastnených hier a upozornenia v reálnom čase. Počas vývoja aplikácie sme sa stretli s rôznymi problémami, od banálnych až po závažné. Napriek tomu, že táto práca predstavovala pre nás novú skúsenosť, využili sme naše znalosti v oblasti programovania a dokázali sme ju úspešne zvládnuť. Práca na tomto projekte nám poskytla množstvo zaujímavých skúseností, ktoré bezpochyby využijeme v budúcnosti.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] Ako nastaviť platby kartou cez platobnú bránu Stripe [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <https://www.webnode.info/sk_SK/e-shop-2-0/ako-nastavit-platby-kartou-cez-platobnu-branu-stripe>
- [2] Co to je Figma? [online]. 2022, [cit. 2023-02-07]. Dostupné na internete: <<https://www.drawplanet.cz/co-to-je-figma/>>
- [3] CSS [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://www.fine.sk/pojmy-a-skratky/css/>>
- [4] Dillon [online]. 2022, [cit. 2023-02-07]. Dostupné na internete: <<https://iknowthatnow.com/2022/04/11/2022-php-statistics/>>
- [5] Dokumentácia Bulma [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://bulma.io/documentation/>>
- [6] Dokumentácia Laravel [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://laravel.com/docs/>>
- [7] Framework [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://www.damidev.com/slovník/framework>>
- [8] Jan Lupčík [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://www.itnetwork.sk/php/laravel/uvod-do-laravel-frameworku-pre-php>>
- [9] Martin Heller [online]. 2022, [cit. 2023-02-07]. Dostupné na internete: <<https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>>
- [10] Rachit Solanki [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://www.velotio.com/engineering-blog/scalable-real-time-communication-with-pusher>>
- [11] Skriptovací/programovací jazyk PHP [online]. 2023, [cit. 2023-02-07]. Dostupné na internete: <<https://napoveda.slovaknet.sk/inpage/php/>>

[12] Šafo [online]. 2019, [cit. 2023-02-07]. Dostupné na internete:

<<https://blog.bart.sk/uvod-vue-js/>>

[13] Tailwind CSS [online]. 2021, [cit. 2023-02-07]. Dostupné na internete:

<<https://jecas.cz/tailwind-css>>

[14] Tomzur [online]. 2021, [cit. 2023-02-07]. Dostupné na internete:

<<https://dev.to/tomzur/php-benchmarks-2021-for-20-different-php-platforms-on-seven-different-php-versions-5-6-7-0-7-1-7-2-7-3-7-4-8-0-3nbp>>

[15] Značkový jazyk (HTML) [online]. 2023, [cit. 2023-02-07]. Dostupné na internete:

<<https://web.vavyskov.cz/znackovaci-jazyk.html>>

[16] Základné informácie o databázach [online]. 2023, [cit. 2023-02-07]. Dostupné na

internet: <<https://napoveda.slovaknet.sk/inpage/co-je-to-mysql/>>

PRÍLOHA A

USB disk príloha

Priložený USB disk obsahuje:

- Dokumentáciu v elektronickej podobe
- Fotografiu ER diagramu
- Laravel projekt
- Fotografie a video záznam finálneho stavu webovej aplikácie
- Powerpoint prezentáciu