Szakképesítés megnevezése: **Webfejlesztés**

A képzés azonosítója: **06135001**

VIZSGAREMEK

Artist page – Művészbemutató oldal

Halász Gábor Obertik Orsolya

témavezető 1/15WL

Budapest, 2022

*Nyilatkozat*

*Alulírott Obertik Orsolya kijelentem, hogy ez a záródolgozat saját tudásom, önálló munkám terméke.*

*Dátum:2022. április. 30*

*hallgató aláírása*

[*1.* Bevezeté*s* 4](#_Toc102415744)

[*1.1* *A program rövid áttekintése* 4](#_Toc102415745)

[*1.2* *A program fejlesztését megelőző kitűzött célok* 4](#_Toc102415746)

[*2.* *Témaválasztás indoklása* 5](#_Toc102415747)

[3. Fejlesztői dokumentáció 5](#_Toc102415748)

[*3.1* *Alkalmazott fejlesztői eszközök* 5](#_Toc102415749)

[*3.2* *Adatmodell leírása* 6](#_Toc102415750)

[3.2.1 Adatbázis-kapcsolati ábra 6](#_Toc102415751)

[3.2.2 Adattáblák 6](#_Toc102415752)

[3.3 Részletes feladatspecifikáció, algoritmusok 7](#_Toc102415753)

[3.3.1 Backend megvalósítása Laravel keretrendszer használatával 7](#_Toc102415754)

[3.3.2 API működése 7](#_Toc102415755)

[3.3.3 API végpontok 7](#_Toc102415756)

[3.3.4 Controllerek leírása 9](#_Toc102415757)

[3.3.5 Szerveroldali renderelés 9](#_Toc102415758)

[3.4 A program telepítésének és konfigurálásának leírása 10](#_Toc102415759)

[3.4.1 Rendszerkövetelmények 10](#_Toc102415760)

[3.4.2 Adatbázis létrehozása 10](#_Toc102415761)

[3.4.3 Alkalmazás telepítése 10](#_Toc102415762)

[3.4.4 A Backend beállítása és elindítása 11](#_Toc102415763)

[*3.5 Frontend megvalósítása React keretrendszer használatával* 11](#_Toc102415764)

[*3.5.1*  *A React alkalmazás felépítése és az adatkérések megvalósítása* 11](#_Toc102415765)

[*3.5.2*  *bejelentkezés nélkül használható komponensek* 12](#_Toc102415766)

[*3.5.3*  *Bejelentkezéshez kötött komponensek* 12](#_Toc102415767)

[*3.6 Tesztelési dokumentáció* 13](#_Toc102415768)

[*3.7 Ismert hibák* 13](#_Toc102415769)

[*3.8 Továbbfejlesztési lehetőségek* 13](#_Toc102415770)

[*4.* *Felhasználói dokumentáció* 14](#_Toc102415771)

[*4.1 A program általános specifikációja és használatának részletes leírása* 14](#_Toc102415772)

[5. Összegzés 18](#_Toc102415773)

[Forrásjegyzék, irodalomjegyzék 18](#_Toc102415774)

[Ábrajegyzék 19](#_Toc102415775)

# Bevezetés

## A program rövid áttekintése

Ezen képzés végével egy vizsgaremeket/ záródolgozatot kellett elkészítenünk. Én ehhez egy hozzám közel álló témát szerettem volna választani, ami egy Művész bemutató oldal egy regisztrációs felülettel kiegészítve plusz rendelés leadásos opcióval.

Képi Ábrázolás szakon végeztem az ELTE Savaria Egyetemi Központban 2017-ben. Jelenleg is megrendelés alapon készítek különböző grafikai anyagokat, emellett saját magam szórakoztatására, fejlődésére készítek további munkákat, amit jelenleg is meg lehet találni online különböző platformokon. Művészi tevékenységeimhez más nevet használok, így olykor a sajátommal elég nehezen találnak meg, de ettől függetlenül vannak platformok, ahol a rendes nevemet is megtalálják. Ezzel a Vizsgaremekkel szerettem volna egy művész bemutató oldalt készíteni a munkásságaimmal. Mivel nagyon sok különböző platformon is fent vagyok, mégis egy saját oldal, amin összefoglalóan együtt van minden munkám, elérhetőségem és egy helyen lehetne kezelni a megrendeléseket. A sok különböző platform miatt néha nehézkessé válik követni ki merre adott le különböző megrendeléseket, erre szerettem volna egy megoldást találni, hogy ne kelljen a megrendelőnek a platformok közül válogatni, és a művésznek is könnyebb legyen tájékozódni.

## A program fejlesztését megelőző kitűzött célok

Már az elejétől kezdve a célom egy művész bemutató oldal volt. Nagy hangsúlyt szerettem volna fektetni a munkák bemutatására, mint plusz funkciók ellátására. Egy kép leírásával megadhatnánk mit tartalmaz, de sokkal inkább az összhatás és az, hogy mi van a képen kell, hogy magáért beszéljen, ha már magyarázni kell akkor valamit nem adtunk jól át. Mivel egy megrendeléshez különböző szabályok kötődnek és nem feltétlen lehet megosztani azt, ezért a nagy része a képeknek, amik kint szerepelnek, azok fanartok, vagyis már létező karakterről készített vizuális megvalósítás a saját stílusomban. A megrendelések leadására mindenképpen szerettem volna egy felületet, hiszen általában erre nem kerül sor egy ilyen oldal készítésénél, de én úgy gondoltam, hogy ez nagyon fontos lehet mert így nem kell a különböző oldalak között navigálni, hogy éppen hol merre kell egy megrendelést leadni (sok művész ahány platform annyira válogatja meg, hogy éppen melyiken szereti fogadni a megrendeléseket).

Az alkalmazás egyik fontos koncepciója az volt, hogy felhasználó barát legyen, azért könnyen kezelhetőre és lényegre törőre szerettem volna készíteni. Nagy hangsúlyt fektetve, hogy lehessen látni a képeket egymás mellett, illetve a megrendelés pontosan értelmezhető legyen. Természetesen a megrendelést követően a befejezett munkáról emailben kap tájékoztatást a megrendelő, hiszen ott a képeket nagy formátumban továbbítani lehet.

# *Témaválasztás indoklása*

Egy művész nagyon sok féle platformon megosztja a munkáit, hogy minél többen ismerhessék meg, de kevés rendelkezik saját oldallal. Ezeket az oldalakat ugyan arra a célra használják, megosztják a munkáikat, viszont itt egy olyan plusz lehetőséget szerettem volna megadni, hogy a megrendeléseket egy külön oldalon le lehessen adni, könnyen értelmezhető legyen a működése és ne kelljen ezer meg egy kéretlen üzenetet elfogadni, ami azzal kezdődik, hogy milyen szép a munkád én is szeretnék egy ilyet, hanem lényegre törő legyen, itt a felület, töltsd ki.

Angol nyelven írtam a leírásokat főleg azért, mert a célközösség az egész világ és nem csak egy ország, és mivel az angol az egyik legelterjedtebb nyelv minden platform esetén, így mindenki el tudja olvasni.

Legkönnyebben ennek a témának a feldolgozásához a saját munkáimat véltem érdemesnek felhasználni. Ebben a témában ezért érzem kényelmesnek magam, hiszen személyes tapasztalatokra támaszkodva tudom bemutatni azt amire a jelenlegi - nevezzük kortárs - művészeknek, szüksége lehet.

# Fejlesztői dokumentáció

## Alkalmazott fejlesztői eszközök

A szerver oldali kód elkészítéséhez használtam:

* Visual Studio Code 1.66.2

Frontendhez használt keretrendszer és programozási nyelvek:

* React ^17.0.2
* Sass ^1.49.9
* Bootstrap ^5.1.3

Backendhez használt keretrendszer és programozási nyelvek:

* Laravel/framework ^9.2
* php ^8.0.2
* MySQL (phpMyAdmin 5.1.1, MariaDB 10.4.22)

Használt böngésző:

* Opera GX 85.0.4341.68

Dokumentáció elkészítéséhez:

* Microsoft Word 365

A képi anyagok elkészítéséhez használt programok

* CLIP STUDIO PAINT PRO 1.11.14 202203231709
* KRITA 4.4.1

## Adatmodell leírása

### Adatbázis-kapcsolati ábra

1. ábra adatbázis-kapcsolati ábra

### Adattáblák

* **users:** regisztrált felhasználók
* **commissions:** regisztrált felhasználók által leadott megrendelések, user\_id tartalmazza az éppen bejelentkezett users tábla id-t
* **artworks:** a képek adatait tartalmazza
* **password\_resets:** jelszó visszaállítás tábla
* **personal\_acces\_tokens:** tokeneket tartalmazó tábla
* **failed\_jobs:** Hibásan lefutott service tábla
* **migrations:** migrációs tábla

## 3.3 Részletes feladatspecifikáció, algoritmusok

### 3.3.1 Backend megvalósítása Laravel keretrendszer használatával

Az oldalam backend részéhez a Laravel keretrendszert választottam. A Laravel egy PHP nyelven írt, nyílt forráskódú keretrendszer. Az MVC elvet követi, miszerint a kód egyes részeit a modul alapfunkciója alapján választjuk szét. Adatbázis kezeléshez a Lavarel Eloquent-t használ, amely az alap SQL adatbázis műveleteket tudja kiváltani az ORM modul segítségével (mint például a SELECT, INSERT, UPDATE, DELETE).

A bejelentkezéshez, regisztrációhoz (Authentikációkhoz) token alapú, szerver oldali jogosultságkezelést használtam, ehhez a Laravel, Sanctum modulját alkalmaztam.

### 3.3.2 API működése

API kérések alkalmazásával oldottam meg a frontend és a backend közti kommunikációt, mely http protokollon keresztül történik meg.

Az API minden kérés küldése független egymástól és nem tárol állapotot.

A kérések eredményeire JSON formátumot ad vissza

### 3.3.3 API végpontok

Az API a HTTP protokoll miatt valamilyen szinten védett végpontokat használ, vannak melyek az CSRF modellt alkalmazza, viszont az én vizsgaremekemben ezt a modult(CSRF) „ki kellett venni”, mert olyan hibába ütközött, amit nem tudtam megoldani.

Az „API/AuthController” -en belül megtalálhatóak.

Az alábbi API végpontokat használtam

#### *3.3.3.1 nem bejelentkezéshez kötött API-k*

|  |  |
| --- | --- |
| **signin/login (felhasználó bejelentkezése)** | |
| Endpoint | api/api.js login |
| Method | POST |
| Function | Bejelentkezés a „Request Body” -ban megadottak alapján |
| Authentication | nem igényel |
| Request Body | email: szöveg  password: szöveg |
| Sample Response | {      "success": **true**,      "data": {          "token": "11|LArExxwEV2HWZhkKyQ67LkXmm29aNB3jrbvznocS",          "name": "Obertik Orsolya",          "userType": 0      },      "message": "User signed in"  } |
| Status code | 200 OK |

|  |  |
| --- | --- |
| **signup/register (felhasználó reigsztrációja)** | |
| Endpoint | api/api.js signup |
| Method | POST |
| Function | Regisztráció a „Request Body” -ban megadottak alapján |
| Authentication | nem igényel |
| Request Body | name: szöveg  email: szöveg  password: szöveg |
| Sample Response | {      "success": **true**,      "data": {          "token": "10|XU6YoadLKrSF6dQmtsYzDNFndxNQSZxURYnCjDAM",          "name": "Obertik Orsolya",          "userType": 0      },      "message": "User created successfully."  } |
| Status code | 200 OK |

#### *3.3.3.2 bejelentkezéshez kötött API-k*

**index:** Ellenőrzi, hogy a felhasználó típusa (userType), a nullával rendelkezik e, minden regisztrált felhasználó a nulla-t kapja meg alapértéknek, az egyes értékkel rendelkezőek az adminok. Ha nullával rendelkezik, akkor kiírja, hogy az adott megrendelő adott e le már korábban megrendelést, és ha igen akkor azt azokat felsorolva kiírja az adott user\_id alapján, amit automatikusan elment a tábla amikor a rendelés leadás megtörténik.

**store (rendelés felvétel/tárolás):** A commissions táblában felveszi az éppen bejelentkezett felhasználó által megadott adatokat, és hozzáfűzi a megrendelő user\_id -ját amit az user táblában található id-ből nyer ki, megrendelés státuszának megadja a 0- értéket amit korábban a RULES, vagyis szabályzatokban kifejtetve leírtam „under checking” státusszal megfelelő. Ezt a státuszkódot látja a felhasználó is a felsorolásnál, ezzel tudja ellenőrizni milyen stádiumban tart a megrendelése.

**handleLogout (felhasználó kijelentkeztetése):** eldobja a local storage-ben található, az épen bejelentkezett felhasználó tokenjét.

### 3.3.4 Controllerek leírása

A Controllerekben zajlanak le az adatbázis műveletek. Az alábbiakat tartalmazza:

**ArtworkController:** Használja az artworks modelt.

**AuthController:** Bejelentkezés és a regisztrációs műveleteket tartalmazza.

**BaseController:** a visszatérési értékek JSON formátumra való átalakítása, illetve a sikeres API kérések, és a hibaüzenetek kezelését végzi.

**CommissionController:** Megrendelések felvétele a commissions táblába és lekérdezi a megadott felhasználó userType alapján.

### 3.3.5 Szerveroldali renderelés

A leadott rendelések statisztikai adatait jeleníti meg.

## 3.4 A program telepítésének és konfigurálásának leírása

### 3.4.1 Rendszerkövetelmények

Az alkalmazáshoz használt technológiát már korábban említettem, Laravel 9 és React 17-t használtam. Ezekhez a következő rendszerkövetelményeket várja.

Szerveroldali megfeleléshez a minimum követelmény:

* MySQL 10.4
* PHP 8
* 350MB tárhely
* Microsoft Server 2016, Ubuntu 20.10, Linux 10

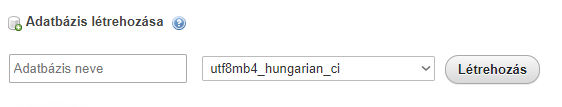
Szerveroldali megfeleléshez az ajánlott követelmény:

* MySQL 10.4
* PHP 8
* 500MB tárhely
* Microsoft Server 2019, Ubuntu 20.10, Linux 10

### 3.4.2 Adatbázis létrehozása

Az adatbázis létrehozásához a phpMyAdmin 5.1.1 verzióját, az adatbázisok tábláinak a kezelésére a MariaDB 10.4.11 használtam. Az adatbázisom neve: „*dikochan*”.

A XAMPP Control Panel elindításával, és az Apach, MySQL -l a „start” action kiadásával elindul a szerverünk. A MySQL-nél az „Admin” -ra kattintva megnyílik a phpMyAdmin ablak.

 Az adatbázis létrehozását a phpMyAdmin -n belül bal-felül az „Új” gombra kattintva hozhatunk létre.

2. ábra Adatbázis létrehozása

### 3.4.3 Alkalmazás telepítése

Az GitHubról letöltött fájlt a XAMPP/htdocs mappába kell kicsomagolni és telepíteni.

### 3.4.4 A Backend beállítása és elindítása

Az elindításhoz a terminálban a következő parancsokat kell kiadni: (elötte ellenőrizni kell, hogy a „env” fileban a DB neve egyezik e a létrehozott adatbázis nevével)

* composer install
* npm install
* php artisan key:generate
* php artisan migrate
* php artisan db:seed
* php artisan serve (a server elindításához kiadandó parancs)

A backend kezdőlap címe: [*https://localhost:8000*](https://localhost:8000)*.*

A seederbe automatikusan betöltésre kerül 2 felhasználó, egy admin és egy sima felhasználó, ahol a sima felhasználóhoz beállítottam 2 példa commission-t. Ennek az adminnak a következők az adatai: email -> admin@dikochan.com password: admin123. A példa felhasználó is jelenleg ugyan ezzel a jelszóval rendelkezik.

## 3.5 Frontend megvalósítása React keretrendszer használatával

### 3.5.1 A React alkalmazás felépítése és az adatkérések megvalósítása

Az artist bemutató oldalam Frontend-je React alkalmazásban készült el. Az oldal nagy hangsúlyt fektet arra, hogy főként bemutató oldalként funkcionáljon és mellékfunkciója, hogy egy bejelentkezéshez/authentikációhoz kötött megrendelés felülettel rendelkezzen. Különböző oldalak között lehet navigálni.

A főbb funkciókat saját komponensekkel láttam el, ezeket a komponenseket az app.js ben fűztem össze és adtam meg a Routokat attól függően, hogy melyik oldal melyik komponensekre hivatkozzon. Működésükhöz szükséges az src/api könyvtárban belül megtalálható osztályok használata:

* **api.js:** Ez a file tartalmazza a különböző authentikációs és egyéb műveteket. az alábbi metódusokkal rendelkezik:
  + **componentDidMount()**
  + **fetchApi()**
  + **login**
  + **signup**
  + **commission**
  + **getCommission**
  + **handleLogout**
* **LoginRequired.js:** Ellenőrzi, hogy a felhasználó be van e jelentkezve.
* **Guest.js:** Ellenőrzi, hogy a felhasználó NE legyen bejelentkezve.

### 3.5.2 bejelentkezés nélkül használható komponensek

**Artworks:** Az képek betöltéséért felel, itt hívom be az összes képet, amit a mappába feltöltöttem.

**Cards:** A navbar alatt megtalálható 4 képből álló kitöltősáv. Főként a Design része.

**Contactme:** Egy rövid bemutatkozó szöveggel rendelkezik, mellette egy GIF-l.

**Footer:** A footer szerepe egy köszönő résszel, és néhány további platformom linkjeivel rendelkezik, itt könnyedén tud bárki navigálni.

**Navbar:** az aloldalak navigálását végzi, itt zajlik a legtöbb művelet:

* handleSignup
* handleLogin
* toggleNav

A bejelentkező és regisztráció mezőt Modalokkal oldottam meg.

**Welcome:** A Home oldalon szerepel, üdvözlő oldalként

**Commissions:** CSAK a szabályzat leírásos része jelenik meg a nem bejelentkezett felhasználók felé

### 3.5.3 Bejelentkezéshez kötött komponensek

**Commissions:** A bejelentkezett felhasználók itt láthatnak egy megrendelés leadáshoz szükséges form-t. A korábban az api-ban létrehozott loginrequired.js meghívásával ellenőrzi hogy a felhasználó be van e jelentkezve.

**CommissionTables:** kilistázza táblába az éppen bejelentkezett felhasználónak a korábban leadott megrendeléseit. Ha a bejelentkezett felhasználó az 1-s userType-l rendelkezik, akkor az összes eddig leadott megrendelést láthatja.

## 3.6 Tesztelési dokumentáció

Az oldalam tesztelését a forrás kód ismerete mellett végeztem el, és minden egyes elkészült résznél tesztelt. Modulteszteket végeztem az egy-egy elkészült funkciónál, minden egyes alkalommal amikor elkészült egy-egy rész.

A felhasználó a megrendelés leadásakor pozitív egész számot tud leadni, viszont egy bizonyos értéknél többet nem tud megadni.  
 Authentikációs tesztet végeztem amikor a felhasználó regisztrációnál vagy bejelentkezésnél próbálkozik belépni hibás adatokkal. Ha regisztrációnál a két jelszó nem egyezik, egy alert-et dob fel, hogy helytelen a két jelszó nem egyezik.   
 Bejelentkezett felhasználó típusát, mint userType alkalmaztam és első soron azt veszi figyelembe, amikor a megrendeléseket felsorolja. Ha az adott felhasználónap a típusa a 0-val rendelkezik akkor csak azokat az adott megrendeléseket sorolja fel, amiket az adott felhasználó adott meg. Viszont az 1-s típusú felhasználó az összes megrendelést látja.

## 3.7 Ismert hibák

Szeretnék felsorolni néhány hibát, amik elsősorban hiányosságok, kimaradt részek. Bejelentkezésnél nem ad üzenetet arra, hogy sikeres a bejelentkezés csak az elem vizsgálatakor, és csak annyi különbséget veszünk észre, hogy a bejelentkezés, regisztrációs gomb átvált, kijelentkezés-re. Az ablak nem záródik be, nekünk kell bezárni. Erre a hibára elsőnek azt a megoldást alkalmaztam, hogy újratöltöttem az oldalt, viszont fellépet egy hiba, ami közben történt és emiatt inkább ki vettem ezt a funkciót a kódból.

Átméretezésnél, amikor a navbar-t kinyitjuk összecsukott állapotáról, az oldal neve elugrik, illetve a jobb oldalon lévő bejelentkezés, regisztrációs gomb a bal alsó sarokba kerül.

Néhány alertet jelez, sajnos nem jöttem rá, hogy mi okozza pontosan és a megoldásához nem volt elegendő rendelkezésre álló időm, ezért azokat bent hagytam.

Az admin avagy userType = 1-l rendelkező felhasználók nem rendelkeznek az oldalon a megrendelések állapotának átállításával, ezeket kézzel lehet a táblában átállítani.

## 3.8 Továbbfejlesztési lehetőségek

Sajnos rendelkezik számos szépséghibával, kommentekkel, amik azért maradtak bent mert nem mertem kitörölni, illetve adott fázisok tesztelésére használtam (ezeket leredukáltam, amennyire tudtam).

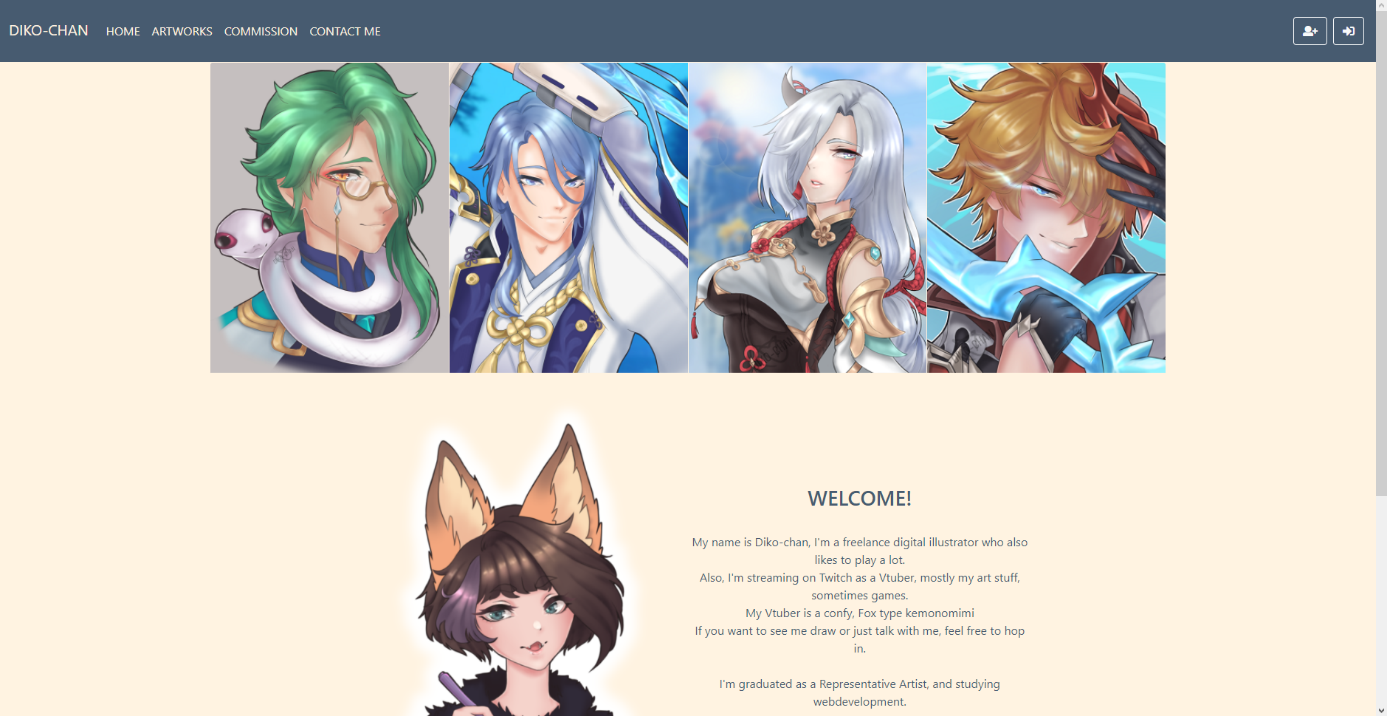
Néhány funkció kimaradt, amit szerettem volna még bele tenni, de sajnos idő hiányában és egyéb okok miatt erre nem került sor. Felsorolnék néhányaz ezek közül:

* az 1-s userType-l rendelkező felhasználók tudják változtatni a megrendelések állapotát. ez a useType az adminokat jelöli.
* jelenleg a regisztrációnál alapvetően 0-s, sima felhasználókat vesz fel, és ahhoz, hogy admin besorolást kapj, kézzel kell a táblában ezt átállítani. Erre kelleni valami megoldást találni
* Ne a frontend oldalon történjen, hanem adatbázisba feltöltött fileokkal lehessen az artworksnél a képeket betölteni, ennek megvalósítását elkezdtem, de sajnos nem sikerült befejeznem.
* A bejelentkezés oldalon pótolni a jelszóemlékeztető kérésének a lehetőségét.

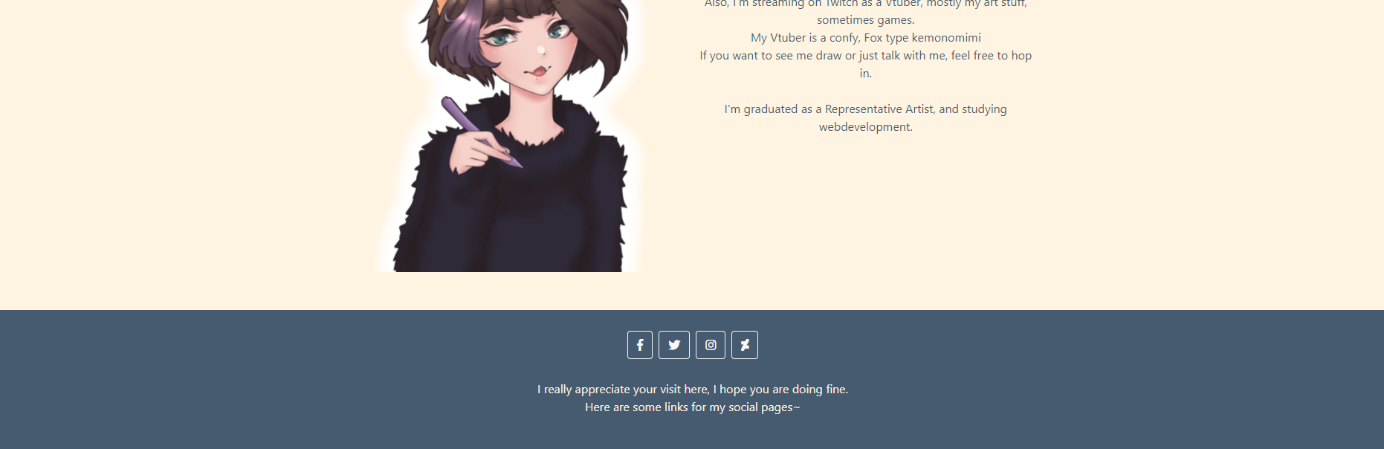
# *Felhasználói dokumentáció*

Ez az Artist oldal egy művész munkásságait mutatja be, és regisztráció megléte után megrendelést is le lehet adni. Ezt az oldalt valós, saját munkákkal töltöttem fel.

## 4.1 A program általános specifikációja és használatának részletes leírása

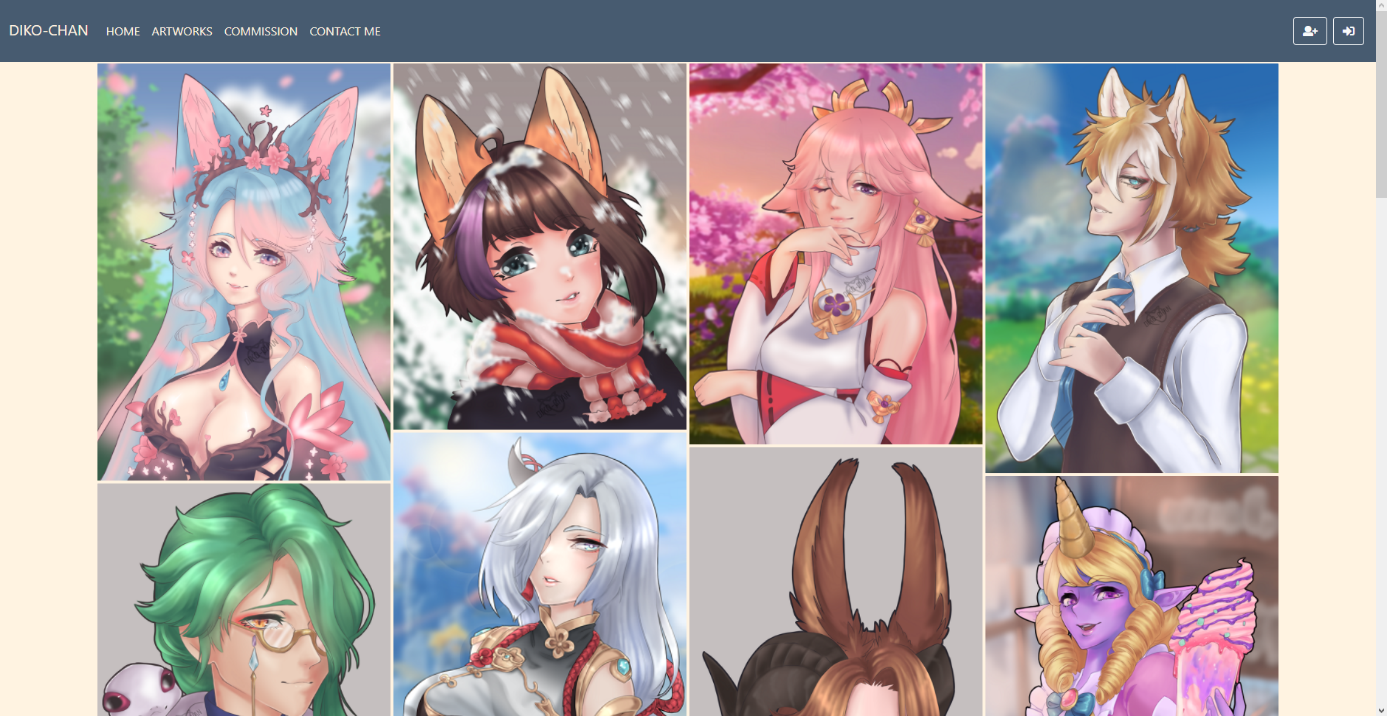
A főoldal megnyitásakor egy köszöntő üzenetet látunk, néhány rajzot bemutatva. Az oldal teljes tartalma nem fér ki egy kijelzőre ezért lejjebb kell görgetni. Felül egy navigációs sávot látunk az alkotó nevével és a további oldalak továbblépési funkciójával. Ennek a navigációs sávnak a jobboldalán tudunk az oldalra belépni, regisztrálni. A 3. ábrán láthatunk erre példát.

3. ábra Home page teteje

Ezen az oldalon amikor lejjebb görgetünk láthatunk egy footer-t. Ez a footer szolgál arra, hogy át tudjunk navigálni a további platformokra, ahol elérhető az adott művész, jelen esetben a saját platformjaimat adtam meg, létező oldalakra irányít át minket. Erre példát a 4. ábrán láthatunk, viszont ez a footer az összes oldalon megtalálható, és nem tesz különbséget bejelentkezett és nem bejelentkezett felhasználó között.

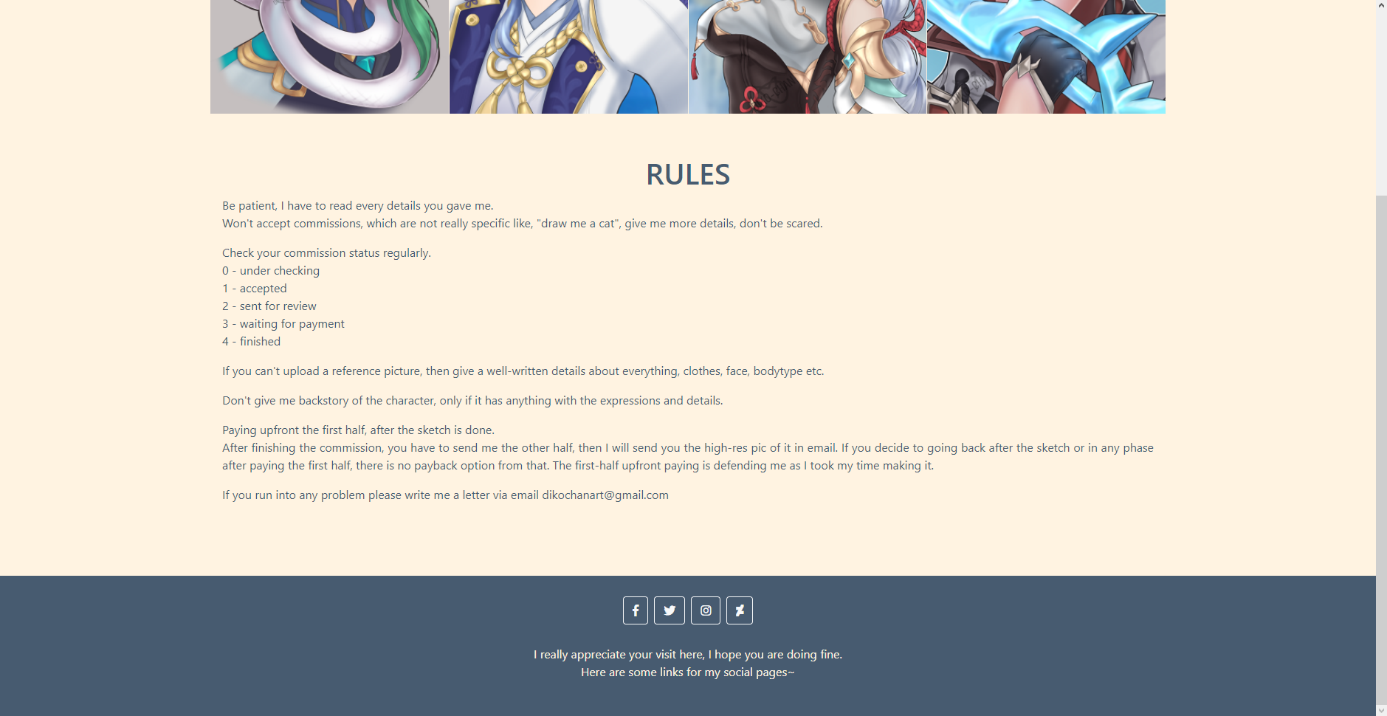
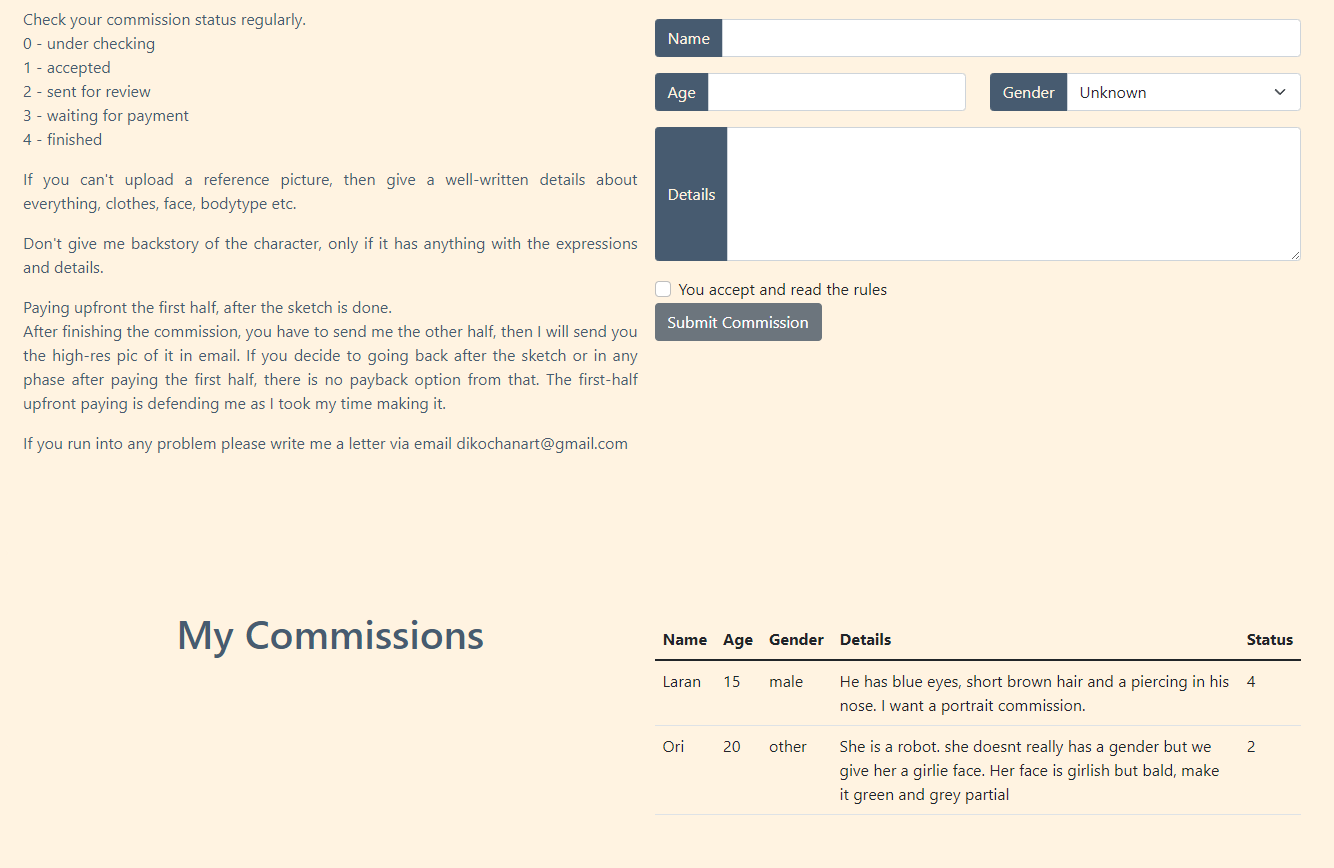
4. ábra Home page alja

A következő oldalunk az ARTWORKS oldal (5. ábra). Ezen az oldalon ugyan úgy megtalálható a navbar, illetve a footer. Ezen az oldalon egymás mellett és alatt találhatóak meg az eddigi rajzok egy része, nem az összes van ide feltöltve, mert az nagyon sok lenne.



5. ábra ARTWORKS oldal

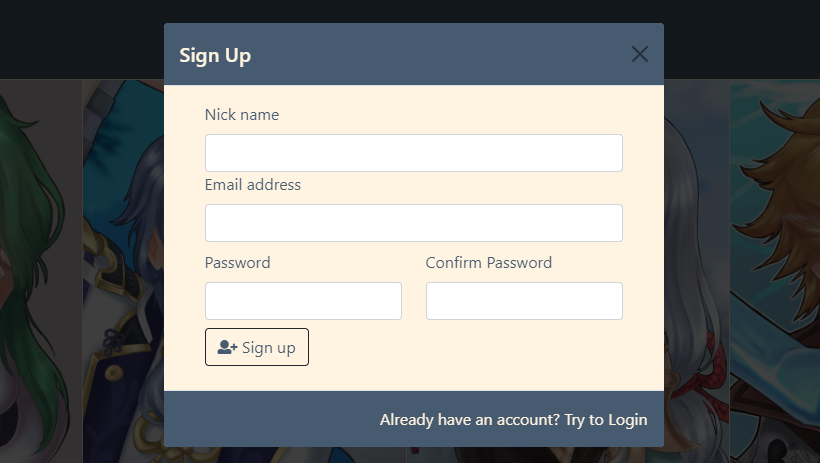
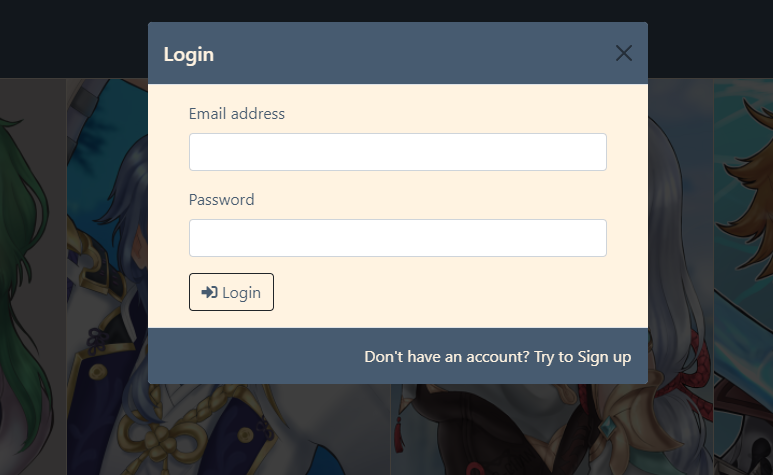
Ezt következő a COMMISSION oldal, itt két részre oszlik, van egy, amit csak a bejelentkezett felhasználók és a nem bejelentkezett felhasználók láthatnak. A nem bejelentkezett felhasználók csak a szabályzatot tudják elolvasni. Viszont a bejelentkezett felhasználók már plusz felületekkel gazdagodnak, számukra elérhető a rendelés leadás opció is, illetve alatta kilistázza, hogy adott e le már megrendelést, és ha igen annak a részleteit. (6. és 7. ábra)



6. ábra Commission oldal a bejelentkezett felhasználónak

7. ábra Commission oldal nem bejelentkezett felhasználóknak

Az utolsó oldalon, a CONTACT ME fül, egy részletesebb leírással szolgál, hogy miként elérhető a művész mivel foglalkozik és általában egy plusz email elérhetőséggel van ellátva.

 Az Navigációs fülön megtalálható bejelentkezés és regisztráció felületről az illusztrációkat a 8. illetve a 9. ábrán láthatunk.

8. ábra Bejelentkezési felület

9. ábra Regisztrációs felület

A regisztráció után automatikusan belépteti a felhasználót az oldal.

# Összegzés

Már nagyon régóta szerettem volna megvalósítani egy saját művész oldalt, előre kitűzött célok vezéreltek, ugyan ezek közben változtak ahogy haladt a projekt és ahogy megtudtam milyen funkciókkal kell ellátni. Valószínűleg, ha elölről kellene kezdenem másként állnék neki most, hogy jobban megismerkedtem a Reactal és a Laravellel. Nem sikerült minden funkcióval ellátnom, mint amivel terveztem, de így is eléggé meg vagyok elégedve azzal amit eddig sikeresen elértem. Rengeteg kihívást jelentett az új keretrendszerek megismerése, megértése.

A fejlesztés során kiemelt figyelmet szerettem volna fordítani, hogy könnyen kezelhető legyen, hiszen megrendeléseket már kamaszok, illetve idős emberek is szeretnek leadni, ezért minden korosztály számára könnyedén értelmezhetőre próbáltam készíteni.

Ezen vizsgaremek készítésekor nagyon sokszor akadtam el, de a tanáraim és szaktársaim is nagyon segítőkészek voltak, és nem csak elméleti kérdéseimben, hanem megannyi hibámban segítséget nyújtottak. Nagyon hálás vagyok a segítségeiknek és a türelmüknek, amit felém nyújtottak.

# Forrásjegyzék, irodalomjegyzék

A következő oldalakat használtam fel az oldalam elkészítéséhez.

<https://laravel.com>

<https://reactjs.org>

<https://www.w3schools.com>

<https://stackoverflow.com>

<https://getbootstrap.com>

<https://laracast.org>

Az oldalon szerepló ikonokhoz a fontAwesom oldalon talált ikonokat importáltam be.

<https://fontawesome.com>

Az oldalon szereplő összes további kép, munka, rajzot én készítettem. Ezek a react-n belül lettek beimportálva. Vízjelemmel vannak ellátva, illetve kisebb formátumban vannak elmentve, így nem érzem feszélyezve, ha ezeket véletlen letöltené bárki is. Nagy file formátumú változatukat az artstation oldalamon ugyan úgy elérhették eddigis.

# Ábrajegyzék

[1. ábra adatbázis-kapcsolati ábra 6](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413938)

[2. ábra Adatbázis létrehozása 10](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413939)

[3. ábra Home page teteje 14](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413940)

[4. ábra Home page alja 15](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413941)

[5. ábra ARTWORKS oldal 15](#_Toc102413942)

[6. ábra Commission oldal a bejelentkezett felhasználónak 16](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413943)

[7. ábra Commission oldal nem bejelentkezett felhasználóknak 16](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413944)

[9. ábra Bejelentkezési felület 17](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413945)

[8. ábra Regisztrációs felület 17](file:///D:\Orsi\Suli\dokumentáció.docx#_Toc102413946)