

Budapesti Műszaki Szakképzési Centrum
Petrik Lajos Két Tanítási Nyelvű Technikum
SZOFTVERFEJLESZTŐ ÉS -TESZTELŐ TECHNIKUS SZAKMA

Socks&Guns

Készítette: Balázs Krisztián Márk, Szalai Szabolcs

Budapest, 2022

Budapesti Műszaki Szakképzési Centrum
Petrik Lajos Két Tanítási Nyelvű Technikum

Nyilatkozat

Alulírott **Balázs Krisztián Márk, Szalai Szabolcs** kijelentjük, hogy ez a vizsgaremek saját tudásunk, önálló munkánk terméke.

A vizsgaremek közös részeit pontosan jelöltük.

Balázs Krisztián Márk

Szalai Szabolcs

Budapest, 2022.

Tartalomjegyzék

Bevezetés	1
Témaválasztás	1
Ütemterv	2
Fejlesztői dokumentáció	2
1.1. Backend (közös)	2
1.1.1. Az alkalmazott fejlesztői eszközök	2
1.1.2. Adatmodell leírása	3
1.1.3. Api végpontok	6
1.1.4. Továbbfejlesztési lehetőségek	10
1.2. Mobil (Balázs Krisztián Márk)	11
1.2.1. Az alkalmazott fejlesztői eszközök	11
1.2.2. Felületterv	11
1.2.3. Részletes feladat-specifikáció	12
1.2.4. Tesztelési dokumentáció	19
1.2.5. Továbbfejlesztési lehetőségek	20
1.3. Web (Szalai Szabolcs)	21
1.3.1. Az alkalmazott fejlesztői eszközök	21
1.3.2. Felületterv	22
1.3.3. Részletes feladat-specifikáció	23
1.3.4. Tesztelési dokumentáció	30
1.3.5. Továbbfejlesztési lehetőségek	33
Felhasználói dokumentáció	34
1.4. Mobil (Balázs Krisztián Márk)	34
1.4.1. A program általános specifikációja	34
1.4.2. Rendszerkövetelmények	34
1.4.3. A program telepítésének és konfigurálásának a leírása	35
1.4.4. A program használatának a részletes leírása	40
1.5. Web (Szalai Szabolcs)	41
1.5.1. A program általános specifikációja	41
1.5.2. Rendszerkövetelmények	41

1.5.3. A program telepítésének és konfigurálásának a leírása	42
1.5.4. A program használatának a részletes leírása.....	44
Összegzés	45
Források	47
Ábrajegyzék	48

Bevezetés

Az év elején 2-3 fős csapatokat kellett alkotnunk, aminek az lett a vége, hogy én(Krisztián) és kedves barátom Szabolcs egy csoportba kerültünk. Az elején rengeteg ötlet kavargott a fejünkben. Rengeteg ötletünk volt, hogy csináljunk egy videójátékot vagy egy térképhez hasonló alkalmazást viszont az lett velünk leköszölve, hogy az alkalmazásnak ugyan olyan élményt kell nyújtania telefonon és weben is így a webshop mellett döntöttünk. Egy régebbi projektet ástunk elő, amihez már többször is hozzányúltunk. Ez volt a Socks&Guns.

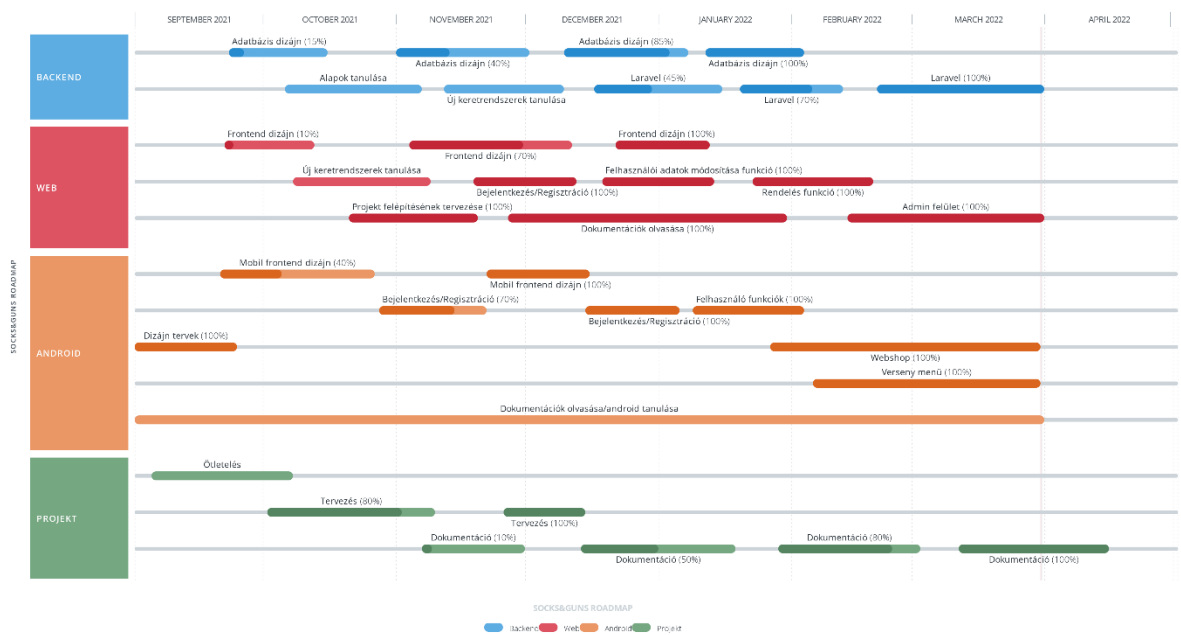
Témaválasztás

A Socks&Guns egy már elkezdett projektünk, amin már több ízben is dolgoztunk különböző iskolai projekteken belül. Az alap elképzelés, hogy egy webshopot építünk fel zoknik és lőfegyverek köré. A zokni részleg egy mindenkinek elérhető, egyszerűen kezelhető webshop lenne, ahol kiválaszthatja a neki megfelelő lábravalót. A fegyver részleg egy privátabb rész lenne, ahol csak a megfelelő dokumentumok beküldése és ellenőrzése után tudna az ember fegyvereket nézegetni. Mivel Magyarországon nem igen vagy egyáltalán nincsen elterjedve a fegyverviselés vagy fegyverbirtoklás ezért erre a témára vonatkozóan nagyon kevés webshop található, nem annyira telített a market, mint egy ruhákat vagy videójátékokat árusító webáruház.

Ütemterv

SOCKS&GUNS ÜTEMTERV - PROJEKT ÜTEMTERV

Sep 1, 2021 - Apr 30, 2022



1. ábra - Ütemterv

Fejlesztői dokumentáció

1.1. Backend (közös)

1.1.1. Az alkalmazott fejlesztői eszközök

Felhasznált programozási nyelvek

1. PHP

Keretrendszerek

1. Laravel

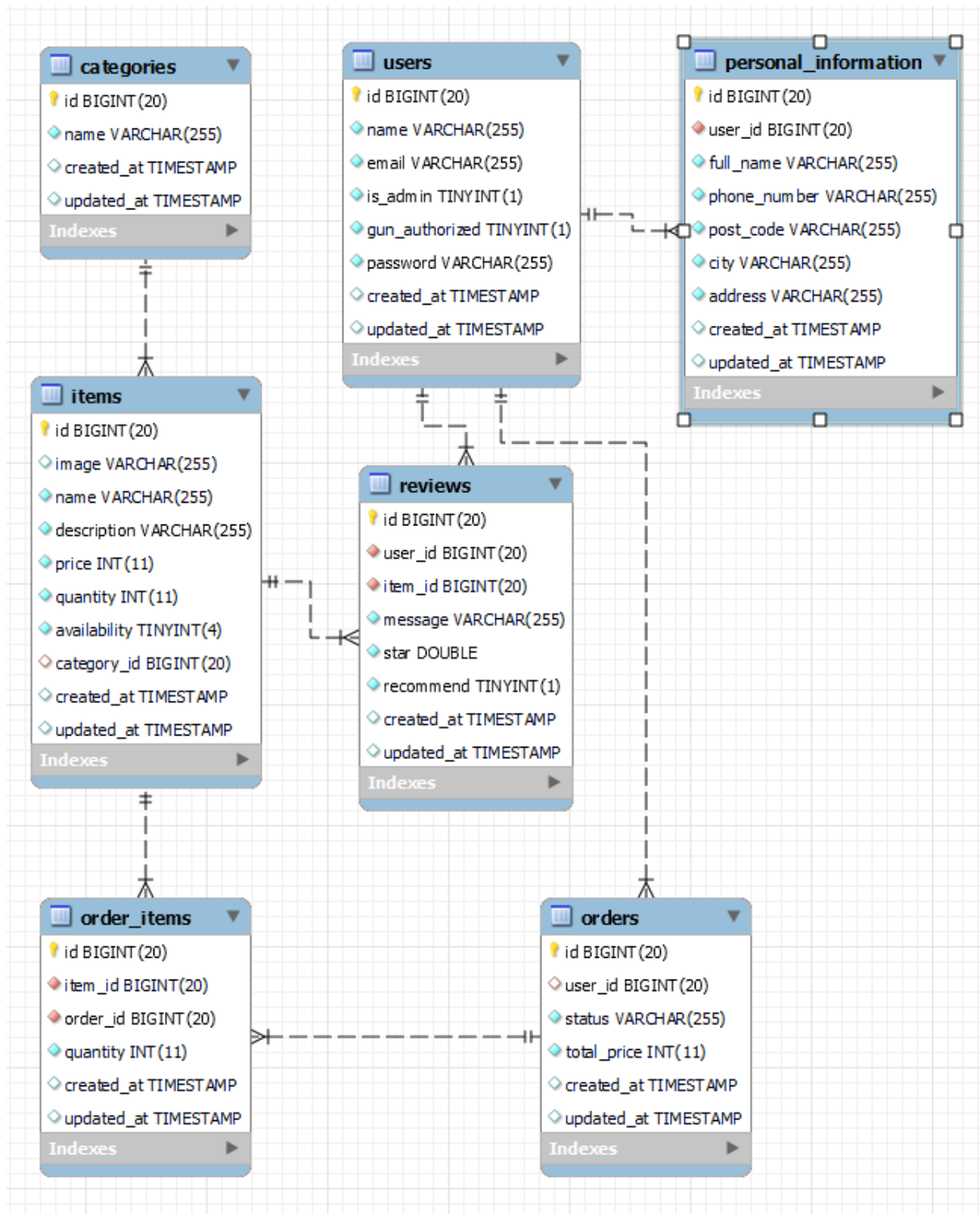
Fejlesztői környezet

1. Visual Studio Code

Bővítmények, függőségek

1. laravel/laravel
2. laravel/sanctum

1.1.2. Adatmodell leírása



users			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Felhasználó azonosítója	AUTO_INCREMENT
name	VARCHAR(255)	Felhasználó neve	Max 255, Min 5 karakter, egyedi
email	VARCHAR(255)	Felhasználó email címe	Max 255 karakter, egyedi
is_admin	TINYINT(1)	Admin jogosultság	0: felhasználó 1: admin
gun_authorized	TINYINT(1)	Vehet e fegyvert a felhasználó	0: nem vehet fegyvert 1: vehet fegyvert
password	VARCHAR(255)	Felhasználó jelszava	Max 255, Min 8 karakter, egyedi
items			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Termék azonosítója	AUTO_INCREMENT
name	VARCHAR(255)	Termék neve	Max 255 karakter
image	VARCHAR(255)	A termék képének az elérési útvonalát tárolja	Nullable, Max 255 karakter
name	VARCHAR(255)	Termék neve	Max 255
description	VARCHAR(255)	Termék leírása	Max 255 karakter
price	INT(11)	Termék ára	Max 11 számjegy
quantity	INT(11)	Termék darabszáma	Max 11 számjegy
availability	TINYINT(4)	Termék elérhetősége	0: nincs raktáron 1: raktáron van
category_id	BIGINT(20)	Kategória idegen kulcsa	Constrained, nullable, onDelete(set null)
categories			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Kategória azonosítója	AUTO_INCREMENT
name	VARCHAR(255)	Kategórianeve	Max 255
personal_information			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Személyes adatok azonosítója	AUTO_INCREMENT

user_id	BIGINT(20)	Felhasználó idegen kulcsa	Constrained, onDelete(cascade)
full_name	VARCHAR(255)	Felhasználó teljes neve	Max 255 karakter
phone_number	VARCHAR(255)	Felhasználó telefonszáma	Max 255 karakter, unique
post_code	VARCHAR(255)	Felhasználó irányítószáma	Max 255 karakter
city	VARCHAR(255)	Felhasználó városa	Max 255 karakter
address	VARCHAR(255)	Felhasználó címe	Max 255 karakter
reviews			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Értékelés azonosítója	AUTO_INCREMENT
user_id	BIGINT(20)	Felhasználó idegen kulcsa	Constrained, onDelete(cascade)
Item_id	BIGINT(20)	Termék idegen kulcsa	Constrained, onDelete(cascade)
message	VARCHAR(255)	Értékelés szövege	Max 255 karakter
star	DOUBLE	Értékelés csillaga	1.0-5.0
recommend	TINYINT(1)	Ajánlás	0: nem ajánlja a terméket 1: ajánlja a terméket
order			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Rendelés azonosítója	AUTO_INCREMENT
user_id	BIGINT(20)	Felhasználó idegen kulcsa	Constrained, nullable, onDelete(set null)
status	VARCHAR(255)	Rendelés státusza	Max 255 karakter
total_price	INT(11)	Rendelés végösszege	Max 11 számjegy
order_item			
Név	Típus	Leírás	Tulajdonságok
id	BIGINT(20)	Rendelt tétel azonosítója	AUTO_INCREMENT
item_id	BIGINT(20)	Termék idegen kulcsa	Constrained, onDelete(cascade)
order_id	BIGINT(20)	Rendelés idegen kulcsa	Constrained, onDelete(cascade)
quantity	INT(11)	Rendelt tétel darabszáma	Max 11 számjegy

1.1.3. Api végpontok

Api url: 127.0.0.1/api

A védett útvonalakhoz a kérés fejlécében a tokenet mindig meg kell adni

Item				
Útvonal	Kérés típusa	Bemenet	Kimenet	Végpont típusa
/items	get	-	összes termék	Publikus
/items/{id}	get	-	létező termék esetén: termék adatai, nem létező termék esetén: Ez a termék nem létezik	Publikus
/items/search/{name}	get	-	létező termék esetén: termék adatai, nem létező termék esetén: Ez a termék nem létezik	Publikus
/items	post	termék neve, leírása, darabszáma, ára, elérhetősége, kategória azonosítója, képe	létrehozott termék adatai	Védett: admin
/items/{id}	put	termék neve, leírása, darabszáma, ára, elérhetősége, kategória azonosítója, képe	Sikeres módosítás	Védett: admin
/items/{id}	delete			Védett: admin

/items/count	get		összes termék száma	Védett: admin
--------------	-----	--	---------------------	---------------

User				
Útvonal	Kérés típusa	Bemenet	Kimenet	Végpont típusa
/user/count	get	-	Összeszámolja az összes user	Publikus
/users	resource	-	Minden ami benne van a user táblában	Publikus
/change_password	put	Új jelszó kétszer	A felhasználó megtudja változtatni a jelszavát	Védett: User
/change_username	put	Új felhasználónév	A felhasználó megtudja változtatni a felhasználónevét	Védett: User
/change_email	put	Új emailcím	A felhasználó megtudja változtatni az emailcímét	Védett: User

Order				
Útvonal	Kérés típusa	Bemenet	Kimenet	Végpont típusa
/order/count	get	-	Megszámolja az ordereket	Védett: Admin
/order/new/count	get	-	Megszámolja a még nem elfogadott rendeléseket	Védett: Admin
/orders/new	get	-	Megjeleníti a még el nem fogadott rendeléseket	Védett: Admin
/orders	get	-	Megjeleníti az összes rendelést	Védett: Admin
/orders/{id}	get	-	Megjelenít egy bizonyos rendelést Id alapján	Védett: Admin

/orders/{order}	put	Status	Rendelések módosítását tudja végrehajtani	Védett: Admin
/orders/new	post	Item Id Darabszám	Felad egy még nem elfogadott rendelést	Védett: Admin
/user/orders	get	-	Lekéri egy bizonyos felhasználó rendeléseit	Védett: User
/orders/{order}	get	-	Törölni tud egy rendelést	Védett: User

Personal Information				
Útvonal	Kérés típusa	Bemenet	Kimenet	Végpont típusa
/personal_information	post	Teljes név Telefonszám Postakód Város Cím	Személyes információt tölt fel	Védett: User
/personal_information	get	-	Az összes személyes információt kéri le	Védett: User
/personal_information/{id}	put	User id Teljes név Telefonszám Postakód Város Cím	Személyes információt módosít id alapján	Védett: User
/personal_information/{id}	delete	User id	Személyes információt töröl id alapján	Védett: User

Authentication				
Útvonal	Kérés típusa	Bemenet	Kimenet	Végpont típusa
/register	post	Név Email Jelszó kétszer	Felhasználó regisztrációt teszi lehetővé	Publikus
/login	post	Email Jelszó	Felhasználó bejelentkezését teszi lehetővé	Publikus

/logout	post	-	Felhasználó kijelentkezését teszi lehetővé	Védett: User
/user	get	-	Éppen bejelentkezett felhasználó adatait adja vissza	Védett: User

1.1.4. Továbbfejlesztési lehetőségek

A backend egy elég stabil konstrukció lett. Közelről sem tökéletes, sőt mi több. Rengeteg fejlesztés és finomítás kell addig amíg azt nem mondhatjuk, hogy a backend makulátlan lett. Meg kell néznünk, hogy mi a felesleges kód vagy funkció, le kell egyszerűsíteni a túlbonyolított kódokat, finomítani kell a különböző api kérések biztonságán és azon, hogy pontosan milyen felhasználók és hogyan érik el. Meg kell néznünk, hogy az sql táblákból mind szükséges vagy megtudjuk oldani szebben és jobban.

Mivel fegyvereket csak úgy nem vásárolhat bárki ezért a következő lépés az alkalmazás fejlesztésében, hogy lehetőséget adjunk a vásárlóknak a fegyvertartási jogunk bemutatásában. Ez egy pdf dokumentum feltöltésével a felhasználó meg is teheti. A dokumentum feltöltése után az admin feladata, hogy megbizonyosodjon a dokumentáció hitelességéről. Ha ez megtörtén és a dokumentáció elfogadásra került, akkor a felhasználó az engedély elfogadásának az időpontjától kezdve fegyvereket is vásárolhat.

Jelenleg az oldalon nincs lehetőség kártyás fizetésre, ezért a későbbiekben egy fizetési rendszer implementálásával fejlesztenénk tovább az oldalt ezzel biztosítva egy jobb vásárlási élményt.

1.2. Mobil (Balázs Krisztián Márk)

1.2.1. Az alkalmazott fejlesztői eszközök

Felhasznált leíró és programozási nyelv

1. XML
2. Java

Keretrendszer

1. Android

Fejlesztői környezet

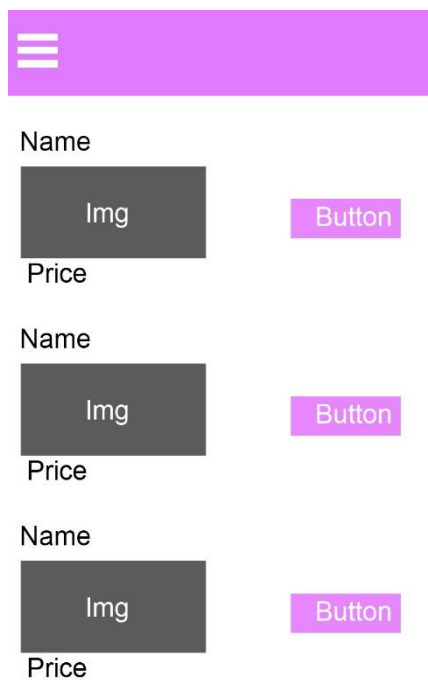
1. Android Studio

Bővítmények függőségek

1. Gson

1.2.2. Felületterv

1.2.2.1. Wireframe



3. ábra - Android Main Terv



2. ábra - Android Menu Terv

Login

Email

Password

Login

Back

4. ábra - Android Login Terv

Register

Name

Email

Password

Password

Register

Back

5. ábra - Android Register Terv

1.2.3. Részletes feladat-specifikáció

1.2.3.1. Classok

Item.java	
dattagok	int id, String name, int price, int quantity, URL image
getId - setId	Az Itemek id-ját tudja vele beállítani vagy lekéréni
getName- setName	Az Itemek nevét tudja vele beállítani vagy lekéréni
getPrice- setPrice	Az Itemek árát tudja vele beállítani vagy lekéréni
getQuantity- setQuantity	Az Itemek mennyiségét tudja vele beállítani vagy lekéréni
getImg- setImg	Az Itemek képét tudja vele beállítani vagy lekéréni

User.java	
dattagok	int id, String name, String email, int is_admin, int gun_authorized
getId - setId	Az Userek id-ját tudja vele beállítani vagy lekérni
getName- setName	Az Userek nevét tudja vele beállítani vagy lekérni
getIs_admin- setIs_admin	Az Userek adminságát lehet vele lekérni vagy megváltoztatni
getGun_authorized- setGun_authorized	Az Userek fegyverviselési engedélyét lehet vele lekérni vagy megváltoztatni

Token.java	
dattagok	String Token
getToken - setToken	Tokenek lekérése vagy megváltoztatása lehetséges ezzel

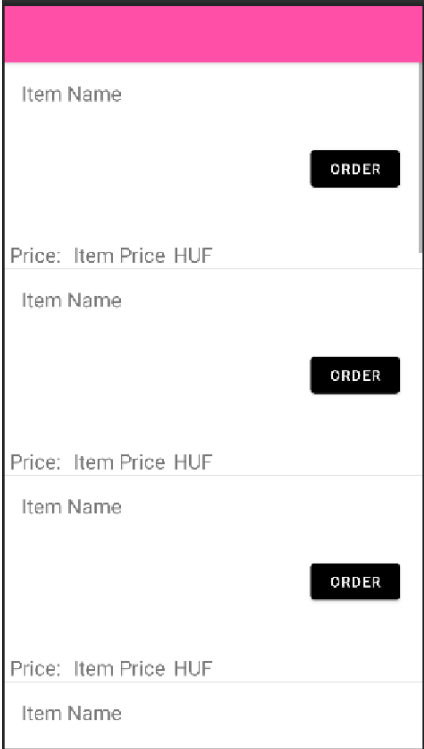
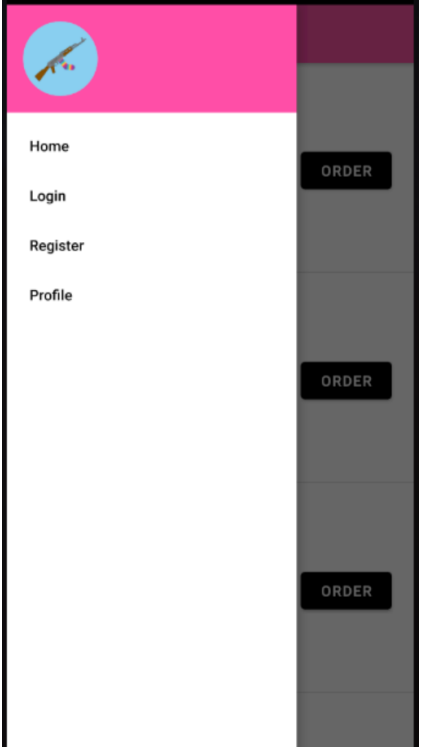
UserResponse.java	
dattagok	User user, String token
getToken - setToken	User tokenek lekérése vagy megváltoztatása lehetséges ezzel
getUser - setUser	User tokenek lekérése vagy megváltoztatása lehetséges ezzel

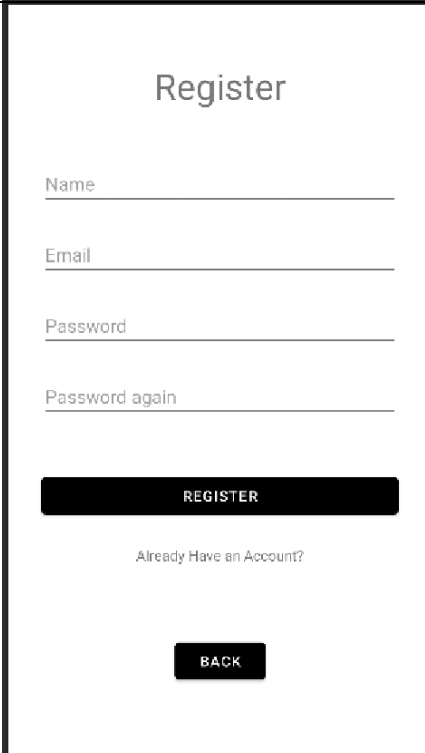
Response.java

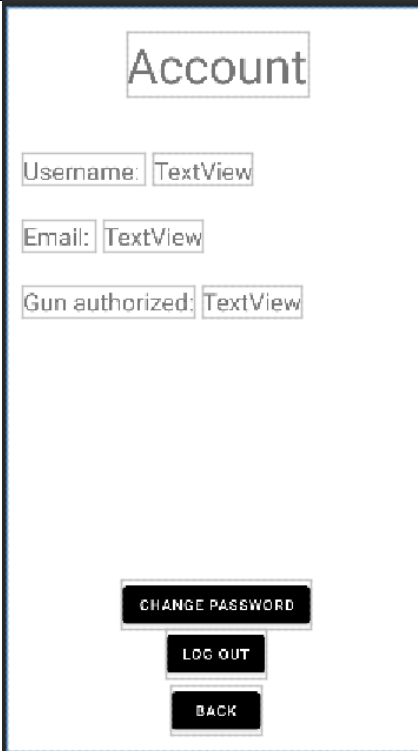
dattagok	int responseCode, String content
getResponseCode - setResponseCode	A műveletek válaszkódját kérem le ezzel. Tesztelésre lett létrehozva
getContent - setContent	A fentebb említett hibakódok hibáinak a pontos kódját kapom vissza ezzel vagy bármilyen responseban szereplő adatot kapok meg, töltök fel

RequestHandler.java	
Response get	Ezen keresztül kapom meg az api válaszokat
Response getWithAuth	Ezen keresztül kapom meg az api válaszokat bearer tokennel
Response post	Ezen keresztül töltök fel mindent az apin keresztül
Response logOutPost	Ezen keresztül töltök fel mindent a bearer tokennel az apin keresztül
Response postWithAuth	Ezen keresztül töltök fel mindent a bearer tokennel és a data-val() az apin keresztül
addRequestBody	Ezen keresztül kérem le a json fileokat
HttpURLConnection setupConnection	Ez segít létrehozni a http connectiont és timeoutol 10 másodpercel az után, hogy elkezdte lekérni az adatokat, ha nem kap adatokat
Response getResponse	Ez segít kiírni a hibáknak a hibakódját

1.2.3.2. Activityk

Activity neve	Activity leírása	Elérhető activityk	Kinézete
MainActivity.java	Ez a főoldal. Itt jelennek meg a termékek egymás alatt, képpel, árral és a termék nevével. Innen egy oldelmenü segítségével lehet eljutni a többi oldalra. Ha az oldalon az order gombra nyomunk, akkor megrendeli azt a terméket, amelyikhez a gomb tartozik	LoginActivity ProfileActivity RegisterActivit y	
SlideMenu	Ezen a menün keresztül érem el a többi alkalmazást	LoginActivity ProfileActivity RegisterActivit y	

LoginActivity.java	Ezen az activity-n lehet bejelentkezni vagy visszalépni a fő oldalra a back gomb segítségével. Bejelentkezés után egyből átdob a főoldalra	MainActivity	 <p>The Login activity UI features a title 'Login' at the top. Below it are two input fields labeled 'Email' and 'Password'. A link 'Forgot password?' is positioned to the right of the password field. A large black button with the text 'BUTTON' is centered below the inputs. At the bottom, there is a link 'Dont have an Account yet?Sign up!' and a black button labeled 'BACK'.</p>
RegisterActivity.java	Ezen az activity-n lehet bejelentkezni vagy a visszalépni a fő oldalra a back gomb segítségével. Ha bejegyezték, akkor visszacsbja a főoldalra	MainActivity	 <p>The Register activity UI features a title 'Register' at the top. Below it are four input fields labeled 'Name', 'Email', 'Password', and 'Password again'. A large black button with the text 'REGISTER' is centered below the inputs. At the bottom, there is a link 'Already Have an Account?' and a black button labeled 'BACK'.</p>

ProfileActivity.java	Ezen az activity-n lehet elérni a felhasználó nevét, emailcímét és azt, hogy a felhasználónak van e fegyverengedély e. Ugyan itt lehet kijelentkezni és a back gomb segítségével visszalépni a főoldalra	MainActivity	
----------------------	--	--------------	--

Main Activity	
getSharedPreferences()	Ezzel kérem le a főoldalon a a preferencst amivel átadom a tokent ami alapján azt tudom ellenőrizni, hogy be van e jelentkezve az illető
getView()	Ezzel tudom megkülönböztetni a ListView list itemjeit
getName()	Ezzel kérem le az Apiból az item nevét
getPrice()	Ezzel kérem le az Apiból az item árát
getImage()	Ezzel kérem le az Apiból az item képét
setText()	Ezzel tudom átállítani a különböző textViewok feliratát. Jelen esetben a névre és az árra
setImageURI(Uri.parse(String.format()))	Ezzel a kód sorral tudom átváltoztatni a az URL típusban bejövő URL-t URI-re amitvel már az android megtudja jeleníteni a képet
doInBackground()	Bármi, amit ebbe a függvénybe írok az a háttérben fog lefutni
RequestHandler.postWithAuth()	A RequestHandler-ben lévő postWithAuth függvényt hívom meg és azt használom az URL-en
getString()	Amihez hozzakötöm annak megkapom a Stringjét
toString()	Legfőképp hibák kiírására használom vagy bárhol, ahol szeretném, hogy átváltoztasson stringbe

Log	Hibák kiíratására használom. Kiíratja a logban a hibakódot vagy a hibát.
Toast	Ezt használom, hogy visszajelzést küldjek a felhasználónak.
onPostExecute()	Megjeleníti a betöltést ameddig a háttérben folyik a számítás. Ezt használom arra, hogy kiírássam az adatokat gsonból
onPreExecute()	Használja a User Interface-et ameddig nem töltött be semmi
onCreate()	Végrehajtja az utasításokat amint betöltött az alkalmazás

Login Activity	
.getToken()	Ezzel kapom meg a felhasználó tokenjét.
finish()	Bezárja azt az Intentet amin éppen rajta van a felhasználó. Jelen esetben arra használom, hogy a bejelentkezés után egyből visszadobjon a főoldalra
backBtn.setOnClickListener()	A BACK gombra egy műveletet tesz ami csak akkor indul el ha a felhasználó rákattint a gombra
.getText().toString().trim()	Ezzel helyezem bele az email változóba az edit text szövegét, változtatom szöveggé és vágom le a végéről a felesleges spacet ha van
.isEmpty()	Ezzel tudom leellenőrizni, hogy az editText üres e
.length()	Ezzel tudom leellenőrizni, hogymilyen hosszú az editText
LoginTask()	Ezzel hívom meg a LoginTask ()-ot ami végrehajtja a felhasználó regisztrálásához szükséges függvényeket

Register Activity	
UserTask()	Ezzel hívom meg a UserTask()-ot ami végrehajtja a felhasználó regisztrálásához szükséges függvényeket

Profile Activity	
.remove("token")	Ezzel a parancsal tudjuk eltávolítani a felhasználó tokenjét a sharedPreferencesből

1.2.4. Tesztelési dokumentáció

1.2.4.1. Regisztráció

Művelet	Bemenet	Várt eredmény	Valós eredmény
Regisztráció új adatokkal	Felhasználónév: testuser2 Email: testuser2@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh1234	Register successful	Sikeres regisztráció, Átirányítás a főoldalra
Regisztráció foglalt email címmel	Felhasználónév: testuser3 Email: testuser2@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh1234	Sikertelen regisztráció, hibaüzenet toastban: Email is already taken	Sikertelen regisztráció, hibaüzenet piros háttérrel: A(z) email már foglalt.
Regisztráció rövid jelszóval	Felhasználónév: testuser3 Email: testuser3@example.com Jelszó: Asdfgh Jelszó újra: Asdfgh	Sikertelen regisztráció, hibaüzenet toastban: The password need to be at least 8 characters long	Sikertelen regisztráció, hibaüzenet piros háttérrel: A jelszónak minimum 8 karakternek kell lennie
Regisztráció nem megegyező jelszavakkal	Felhasználónév: testuser4 Email: testuser4@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh4321	Sikertelen regisztráció, hibaüzenet toastban: The password must match	Sikertelen regisztráció, hibaüzenet piros háttérrel: A két jelszónak meg kell egyeznie
Regisztráció rövid felhasználónévvel	Felhasználónév: test Email: testuser5@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh1234	Sikertelen regisztráció, hibaüzenet toastban: The username need to be a minimum of 5 and a maximum of 20 characters	Sikertelen regisztráció, hibaüzenet piros háttérrel: A felhasználónévnek minimum 5 karakternek kell lennie

1.2.4.2. Bejelentkezés

Művelet	Bemenet	Várt eredmény	Valós eredmény
Bejelentkezés létező felhasználóval	Email: testuser@example.com Jelszó: Asdfgh1234	Login successful	Sikeres bejelentkezés, Átírányítás a főoldalra
Bejelentkezés helytelen adatokkal	Email: testuser420@example.com Jelszó: 12345678	An error occurred while logging in	Sikertelen bejelentkezés, hibaüzenet piros háttérrel: Helytelen felhasználónév vagy jelszó
Bejelentkezés adatok nélkül	Nincs bemenet	An error occurred while logging in	Sikertelen bejelentkezés, hibaüzenet piros háttérrel: Email cím megadása kötelező, Jelszó megadása kötelező

1.2.5. Továbbfejlesztési lehetőségek

Az androidhoz rengeteget terveztem, amivel sajnos nem tudtam elkészülni. Először is szeretném megcsinálni az applikációt sokkal stabilabbra, hogy a túl gyors bejelentkezés vagy a nem megfelelő sorrendben történő bejelentkezés ne tudja befagyasztani, vagy a főoldalon sokkal kellemesebben legyenek elhelyezve a termékek. Terveztem hozzá egy keresőfület, ahol termékek neve vagy kategóriája alapján lehet keresni, de idő hiánya miatt nem tudtam vele elkészülni. Szeretnék beleimplementálni egy kosarat, hogy egyszerre több terméket is lehessen rendelni és ne egyesével kelljen ezt megtenni. Szerettem volna egy plusz funkciót is beletenni, ahol helyet lehetett volna foglalni egy lőtérré és pontokat lehetett volna számolni, hogy ki hol és mit talált el.

1.3. Web (Szalai Szabolcs)

1.3.1. Az alkalmazott fejlesztői eszközök

Felhasznált leíró és programozási nyelvek

1. HTML
2. CSS
3. JavaScript

Keretrendszerek

1. Vue.js
2. Tailwind CSS

Fejlesztői környezet

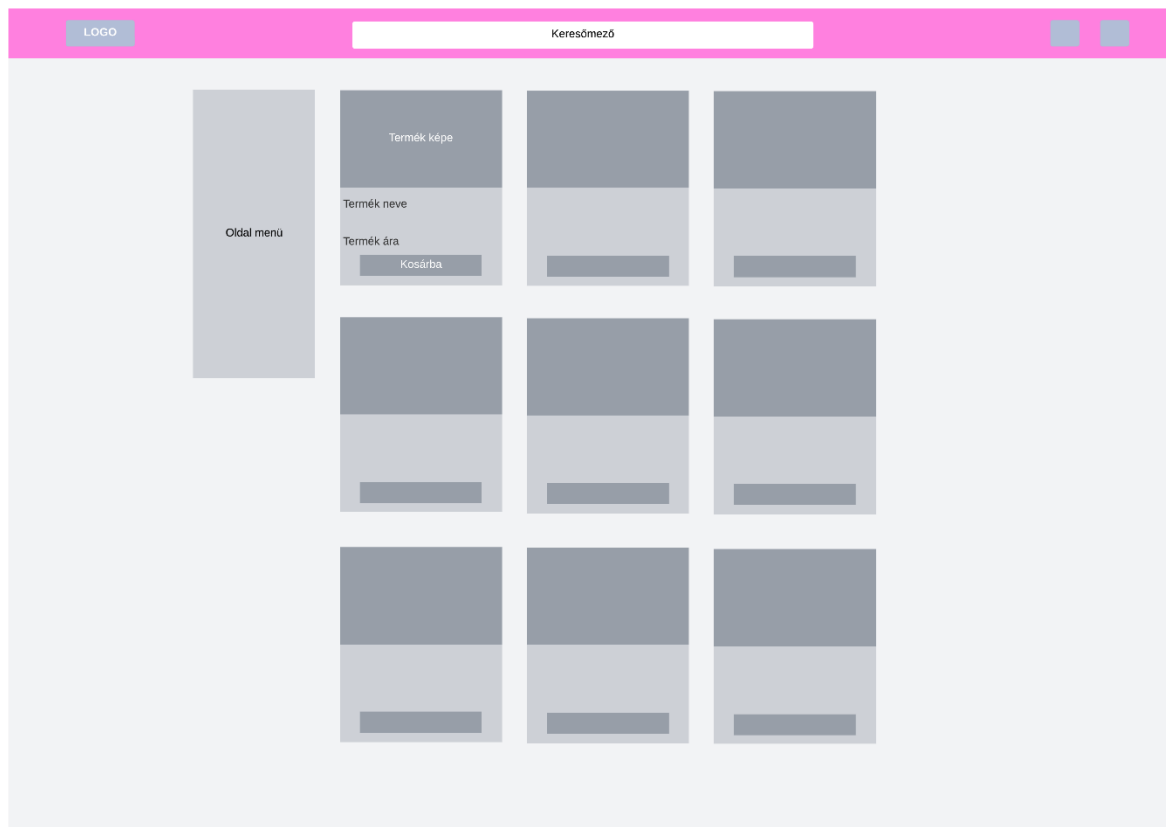
1. Visual Studio Code

Bővítmények, függőségek

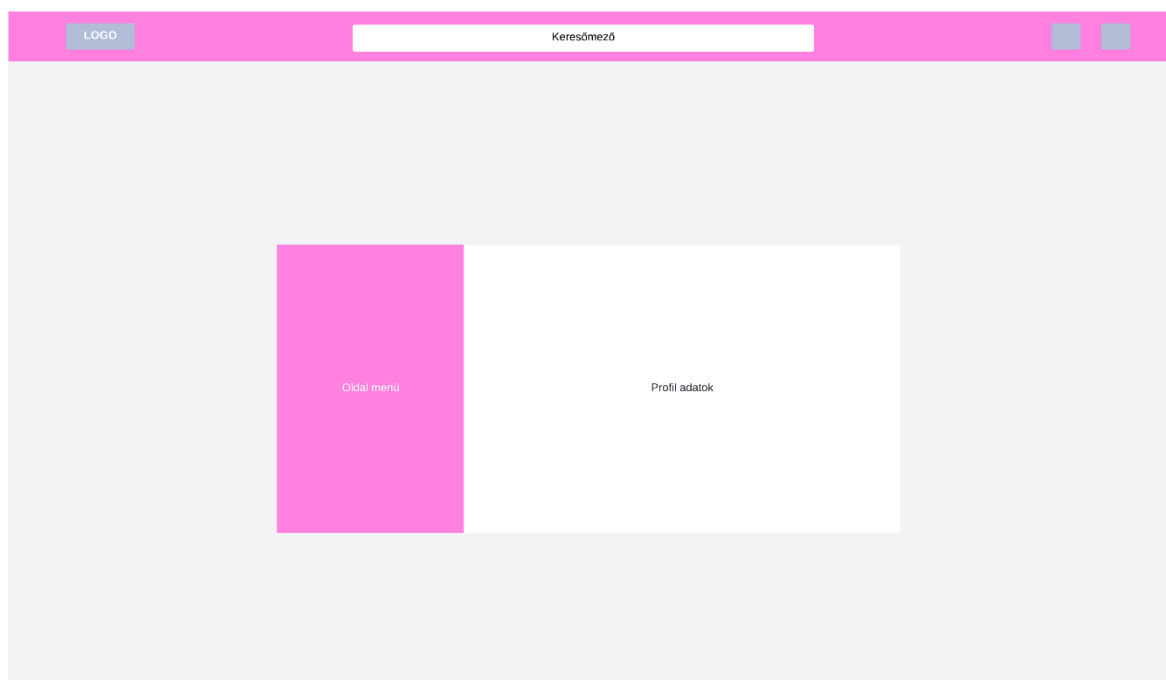
3. Vue Router
4. Vuex
5. Vuelidate
6. Axios
7. Heroicons
8. Vue Language Features (Volar)
9. Tailwind CSS IntelliSense

1.3.2. Felületterv

1.3.2.1. Wireframe



6. ábra - felületterv - főoldal



7. ábra - felületterv - profil menü

1.3.3. Részletes feladat-specifikáció

1.3.3.1. Függvények

products.js		
adattagok	products[]	Összes terméket tárolja
	product[]	Egy terméket tárol
	productCount	Összes termék számát tárolja
getProducts()	Api get kérés segítségével lekéri az összes terméket az adatbázisból és feltölti a products tömböt	
getProduct(id)	Api get kérés segítségével lekér egy darab terméket id alapján az adatbázisból és feltölti a product tömböt	
searchProduct(searchName)	Api get kérés segítségével lekéri azokat terméket az adatbázisból, amelyek nevében szerepel a keresés paramétere és ezekkel feltölti a products tömböt	
getProductCount()	Api get kérés segítségével lekéri a termékek számát	
newProduct(data)	Api post kérés segítségével új rekordot hoz létre az adatbázisban a paraméterként átadott adatokkal	
updateProduct(id, productData)	Api post kérés segítségével meglévő rekordot módosít a termék id -ja alapján	
destroyProduct(id)	Api delete kérés segítségével id , alapján törli a terméket az adatbázisból	

store/index.js		
adattagok	state.user.data	Felhasználó adatait használja
	state.user.token	Tokenet tárolja
	state.cart	Kosárhoz adott termékeket tárolja a böngésző lokális tárhelyében
register	Api post kérés segítségével új felhasználót hoz létre és a setUser függvénnyel elmenti a state.user.data tömbbe és a backend szerver által generált tokenet elmenti a sessionStorage-ba	
login		
logout	Api post kérés segítségével törli a felhasználóhoz tartozó tokenet az adatbázisból és a sessionStorage-ból	
getUser	Api get kérés segítségével lekéri az adatbázisból az éppen bejelentkezett felhasználó adatait	
updateLocalStorage	A cart tömböt JSON formátumba alakítja és elmenti a kosarat a böngésző lokális tárhelyébe	
addToCart	A kosárhoz ad hozzá termékeket, ha a termék már a kosárban van akkor a darabszámot növeli	
decreaseQuantity	A kosárban lévő termék darabszámát csökkenti, ha eléri a nullát a darabszám akkor kitörli a kosárból a terméket	
removeFromCart	Eltávolítja a kosárból a terméket	

emptyCart	Kiüríti az egész kosarat
user	Lekéri a felhasználó adatait
productQuantity	A kosárban lévő termék darabszámát kéri le
cartItems	Visszaadja a kosár tartalmát
cartHasGuns	Visszaadja, hogy van-e fegyver a kosárban
cartItemCount	A kosárban lévő termékek darabszámát adja vissza
cartTotal	A kosárban lévő termékek árának az összegét adja vissza
cartOrder	Elmenti az összes kosárban lévő termék azonosítóját és darabszámát egy orderCart nevű tömbbe , rendelés leadásakor ezt küldjük tovább a szervernek JSON formában

categories.js		
adattagok	categories[]	Összes kategóriát tárolja
	category[]	Egy kategóriát tárol
getCategories()	Api get kérés segítségével lekéri az összes kategóriát az adatbázisból és feltölti a categories tömböt	
getCategory(id)	Api get kérés segítségével lekér egy darab kategóriát id alapján az adatbázisból és feltölti a category tömböt	
newCategory(data)	Api post kérés segítségével új rekordot hoz létre az adatbázisban a paraméterként átadott adatokkal	
updateCategory(id, categoryData)	Api post kérés segítségével meglévő rekordot módosít a kategória id -ja alapján	
destroyCategories(id)	Api delete kérés segítségével, id alapján törli a kategóriát az adatbázisból	

users.js		
adattagok	users[]	Összes felhasználót tárolja
	user[]	Egy felhasználót tárol
	userCount	Összes felhasználó számát tárolja
getUsers()	Api get kérés segítségével lekéri az összes felhasználót az adatbázisból és feltölti a users tömböt	
getUser(id)	Api get kérés segítségével lekér egy darab felhasználót id alapján az adatbázisból és feltölti a user tömböt	
getUserCount()	Api get kérés segítségével lekéri a felhasználók számát	
destroyUser(id)	Api delete kérés segítségével, id alapján törli a kategóriát az adatbázisból	

orderItems.js		
adattagok	orderItems[]	Rendeléshez tartozó összes terméket tárolja
getOrderItems()	Api get kérés segítségével lekéri az adott rendeléshez tartozó összes terméket az adatbázisból és feltölti a orderItems tömböt	

orders.js		
adattagok	orders[]	Összes rendelést tárolja
	order[]	Egy rendelést tárol
	userOrders[]	Bejelentkezett felhasználó rendeléseit tárolja
	orderCount	Rendelések számát tárolja
	newOrderCount	Új rendelések számát tárolja
getOrders()	Api get kérés segítségével lekéri az összes rendelést az adatbázisból és feltölti a orders tömböt	
getOrder(id)	Api get kérés segítségével lekér egy darab rendelést id alapján az adatbázisból és feltölti a order tömböt	
getUserOrders()	Api get kérés segítségével lekéri az éppen bejelentkezett felhasználó összes rendelését és feltölti a userOrders tömböt	
getOrderCount()	Api get kérés segítségével lekéri az összes rendelés darabszámát	
getNewOrderCount()	Api get kérés segítségével lekéri az összes új rendelés darabszámát	
updateOrder(id, orderData)	Api post kérés segítségével meglévő rekordot módosít a rendelés id -ja alapján	
destroyOrder(id)	Api delete kérés segítségével, id alapján törli a rendelést az adatbázisból	

personalInformations.js		
adattagok	pInfo []	Bejelentkezett felhasználó személyes adatait tárolja
	success	Sikeres törlés, módosítás, hozzáadás esetén a response üzenetét tárolja
getPersonalInformation()	Api get kérés segítségével lekéri az éppen bejelentkezett felhasználó személyes adatait az adatbázisból és feltölti a pInfo tömböt	
newPersonalInformation()	Api post kérés segítségével új rekordot hoz létre az adatbázisban a paraméterként átadott adatokkal	
updatePersonalInformation(id, personalInformationData)	Api post kérés segítségével meglévő rekordot módosít a személyes adatok id -ja alapján	
destroyPersonalInformation(id)	Api delete kérés segítségével, id alapján törli a a személyes adatot az adatbázisból	

1.3.3.2. Komponensek

Fő komponensek		
Komponens neve	Komponens leírása	Benne lévő komponensek
App.vue	Ez a legfőbb komponens. Itt jelenik meg az összes komponens dinamikusan, RouterView segítségével	HomePageView.vue LoginView.vue RegisterView.vue UserPageView.vue ProductPageView.vue AdminPageView.vue Cart.vue SearchView.vue
HomePageView.vue	Ez a komponens a főoldal. Itt jelennek meg a termékek kártyás formában	Navbar.vue ProductCard.vue Footer.vue
LoginView.vue	Ezen az oldalon lehet bejelentkezni a webshopba	Login.vue
RegisterView.vue	Ezen az oldalon lehet regisztrálni az oldalra	Register.vue
UserPageView.vue	Ezen az oldalon tudja a felhasználó elérni és szerkeszteni az adatait	Navbar.vue ProfileSideMenu.vue RouterView <ul style="list-style-type: none"> · ProfileInformation.vue · ProfileOrders.vue · ProfileDeliveryAddress.vue Footer.vue
ProductPageView.vue	Ez az oldal jelenít meg egy-egy terméket	Navbar.vue Footer.vue

AdminPageView.vue	Ez az admin menü. Ide csak admin jogosultságú felhasználók léphetnek be. Innen lehet új termékeket, kategóriákat felvenni, törölni és szerkeszteni. Rendelések elfogadására, elutasítására is van lehetőség	AdminSideMenu.vue RouterView <ul style="list-style-type: none"> · AdminDashboard.vue · AdminProductsPanel.vue · AdminNewProduct.vue · AdminEditProduct.vue · AdminCategories.vue · AdminNewCategory.vue · AdminEditCategory.vue · AdminUsers.vue · AdminOrders.vue
CartView.vue	Ez az oldal a kosár	Navbar.vue CartItem.vue Footer.vue
SearchView.vue	Ez az oldal a termékek keresésére szolgál	Navbar.vue SearchProductCard.vue Footer.vue

Mellék komponensek		
Komponens neve	Komponens leírása	Benne lévő függvények
Navbar.vue	Navigációs menü. Innen érhető el a bejelentkezés gomb, bejelentkezett felhasználó esetén pedig a profilhoz vezető gomb és a kijelentkezés gomb. Továbbá a keresés és a kosár oldalt is innen lehet elérni	logout() getUser()
Footer.vue	Footer(élőláb) Ez tartalmazza az oldal nevét és az elkészítés évét	
Login.vue	Ez a komponens a bejelentkezés bonyolítja le	login()
Register.vue	Regisztráció lebonyolításáért felel	register()
CartItem.vue	A kosár különböző elemeinek a megjelenítését hajtja végre	prop-ként kapja meg az adott terméket
SearchProductCard.vue	Ez a komponens jeleníti meg a keresésnek megfelelő termékeket	prop-ként kapja meg a keresett termékeket a SearchView.vue komponensből

ProductCard.vue	Ebben a komponensben jelennek meg a termékek kártyás formában	getProducts()
-----------------	---	---------------

Mellék komponensek		
Komponens neve	Komponens leírása	Benne lévő függvények
ProfileSideMenu.vue	Profil menü. Ebből a komponensből érhetőek el az összes profilhoz tartozó komponensek	
ProfileInformation.vue	Ez a komponens jeleníti meg a felhasználói fiók adatait	getUser()
ProfileOrders.vue	Ez a komponens jeleníti meg a felhasználó rendeléseit	getOrders()
ProfileOrderItem.vue	Megjelenít egy-egy rendelt	prop-ként kapja meg a rendelés adatait a ProfileOrders.vue komponensből
ProfileDeliveryAddress.vue	Ebben a komponensben találhatóak a felhasználó szállítási adatai	getAddress()

Mellék komponensek		
Komponens neve	Komponens leírása	Benne lévő függvények
AdminSideMenu.vue	Admin menü. Ebből a komponensből érhetőek el az összes adminhoz tartozó komponensek	getUser()
AdminDashboard.vue	Ez a komponens egy összesítő oldal. Itt lehet látni a termékek, a felhasználók és a rendelések számát	getProductCount() getUserCount() getOrderCount()
AdminProductsPanel.vue	Ebben a komponensben lehet kilistázni az összes terméket, lehet törölni és a módosítást is innen lehet elérni	getProducts() destroyProduct(id)
AdminNewProduct.vue	Itt hozzá lehet adni új termékeket	getCategories() newProduct()
AdminEditProduct.vue	Ebben a komponensben lehet szerkeszteni	getCategories() getProduct(id) updateProduct(id, newData)
AdminCategories.vue	Ebben a komponensben lehet kilistázni az összes kategóriát, lehet törölni és a	getCategories() destroyCategories(id)

	módosítást is innen lehet elérni	
AdminNewCategory.vue	Itt hozzá lehet adni új kategóriákat	newCategory()
AdminEditCategory.vue	Ebben a komponensben lehet szerkeszteni	updateCategory(id, newData) getCategory(id)
AdminUsers.vue	Ebben a komponensben lehet kilistázni az összes felhasználót, lehet törölni és a módosítást is innen lehet elérni	getUsers() destroyUser(id)
AdminOrders.vue	Ebben a komponensben lehet kilistázni az összes rendelést, el lehet fogadni vagy elutasítani a rendelést	getOrders()
AdminOrderItem.vue	Ez a komponens megjelenít egy-egy rendelést, ahol az admin elfogadhatja vagy elutasíthatja az adott rendelést	updateOrder(id)

1.3.4. Tesztelési dokumentáció

1.3.4.1. Regisztráció

Művelet	Bemenet	Várt eredmény	Valós eredmény
Regisztráció új adatokkal	Felhasználónév: teszt01 Email: test@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh1234	Sikeres regisztráció, Átirányítás a főoldalra	Sikeres regisztráció, Átirányítás a főoldalra
Regisztráció foglalt email címmel	Felhasználónév: teszt02 Email: test@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh1234	Sikertelen regisztráció, hibaüzenet piros háttérrel: A(z) email már foglalt.	Sikertelen regisztráció, hibaüzenet piros háttérrel: A(z) email már foglalt.
Regisztráció rövid jelszóval	Felhasználónév: teszt03 Email: test3@example.com Jelszó: Asdfgh Jelszó újra: Asdfgh	Sikertelen regisztráció, hibaüzenet piros háttérrel: A jelszónak minimum 8 karakternek kell lennie	Sikertelen regisztráció, hibaüzenet piros háttérrel: A jelszónak minimum 8 karakternek kell lennie
Regisztráció nem megegyező jelszavakkal	Felhasználónév: teszt04 Email: test4@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh4321	Sikertelen regisztráció, hibaüzenet piros háttérrel: A két jelszónak meg kell egyeznie	Sikertelen regisztráció, hibaüzenet piros háttérrel: A két jelszónak meg kell egyeznie
Regisztráció rövid felhasználónévvel	Felhasználónév: test Email: test5@example.com Jelszó: Asdfgh1234 Jelszó újra: Asdfgh1234	Sikertelen regisztráció, hibaüzenet piros háttérrel: A felhasználónévnek minimum 5 karakternek kell lennie	Sikertelen regisztráció, hibaüzenet piros háttérrel: A felhasználónévnek minimum 5 karakternek kell lennie

Regisztráció adatok nélkül	Nincs bemenet	Sikertelen regisztráció, hibaüzenet piros háttérrel: Felhasználónév megadása kötelező, Email cím megadása kötelező, Jelszó megadása kötelező	Sikertelen regisztráció, hibaüzenet piros háttérrel: Felhasználónév megadása kötelező, Email cím megadása kötelező, Jelszó megadása kötelező
-------------------------------	---------------	---	---

1.3.4.2. Bejelentkezés

Művelet	Bemenet	Várt eredmény	Valós eredmény
Bejelentkezés létező felhasználóval	Email: testuser@example.com Jelszó: TestPassword01	Sikeres bejelentkezés, Átírányítás a főoldalra	Sikeres bejelentkezés, Átírányítás a főoldalra
Bejelentkezés helytelen adatokkal	Email: test137@example.com Jelszó: 12345678	Sikertelen bejelentkezés, hibaüzenet piros háttérrel: Helytelen felhasználónév vagy jelszó	Sikertelen bejelentkezés, hibaüzenet piros háttérrel: Helytelen felhasználónév vagy jelszó
Bejelentkezés adatok nélkül	Nincs bemenet	Sikertelen bejelentkezés, hibaüzenet piros háttérrel: Email cím megadása kötelező, Jelszó megadása kötelező	Sikertelen bejelentkezés, hibaüzenet piros háttérrel: Email cím megadása kötelező, Jelszó megadása kötelező

1.3.4.3. Termék hozzáadása admin felületen

Művelet	Bemenet	Várt eredmény	Valós eredmény
Termék hozzáadása helyes adatokkal	Termék neve: Piros zokni Kategória: zokni Ár: 3200 Mennyiség: 10000	Sikeres termék hozzáadás, Átírányít a termékeket listázó oldalra	Sikeres termék hozzáadás, Átírányít a termékeket listázó oldalra

1.3.4.4. Termék törlése admin felületen

Művelet	Bemenet	Várt eredmény	Valós eredmény
Termék törlésekor felugró ablak elfogadása	Nincs bemenet	A kiválasztott termék törlődik, Termék lista frissülése után eltűnik a törölt termék	Termék lista frissülése után eltűnik a törölt termék
Termék törlésekor felugró ablak elutasítása	Nincs bemenet	Termék nem törlődik	Termék nem törlődik

1.3.4.5. Kategória hozzáadása admin felületen

Művelet	Bemenet	Várt eredmény	Valós eredmény
Kategória hozzáadása helyes adatokkal	Kategória neve: pisztoly	Sikeres kategória hozzáadás, Átírányít a kategóriákat listázó oldalra	Sikeres kategória hozzáadás, Átírányít a kategóriákat listázó oldalra
Kategória adat nélkül	Nincs bemenet	Sikertelen kategória hozzáadás, Hibaüzenet piros háttérrel: Kategória nevét kötelező megadni,	Sikertelen kategória hozzáadás, Hibaüzenet piros háttérrel: Kategória nevét kötelező megadni,

1.3.5. Továbbfejlesztési lehetőségek

Hiába érezhető úgy, hogy egy alkalmazás tökéletes, továbbfejlesztési lehetőségek mindig vannak. Ez ennél az alkalmazásnál sincs másképp.

Mivel ez egy fegyvereket is árusító webshop, a későbbiekben szeretnénk lehetőséget biztosítani ezen termékek kipróbálásához. Ezt egy lőtér létrehozásával lehetne megoldani, ahol nem csak a vásárlás előtti tesztet lehetne megejteni, hanem élménylövészetre is lenne lehetőség. Az alkalmazásban egy pálya-, időpontfoglalási rendszer állna rendelkezésre, amellyel, egy letisztult és egyszerűen kezelhető felületen lehetne kiválasztani a vásárlónak megfelelő tevékenységet és időpontokat.

Felhasználói dokumentáció

1.4. Mobil (Balázs Krisztián Márk)

1.4.1. A program általános specifikációja

Miután megnyitottuk az alkalmazást a főoldalon lehetőségünk van a termékek között keresni. Ha elhúzzuk a képernyő bal oldalát jobbra vagy a bal felső sarokban a három csíkra kattintunk akkor lenyílik az oldalmenü.

Ebben az oldalmenüben választhatjuk ki, hogy az alkalmazás mely részére szeretnénk eljutni.

Az első Home menüpont újra tölti az oldalt és újra listázza ezzel együtt.

A második Login menüpontra tudunk bejelentkezni meglévő felhasználóval

A negyedik Register menüpontra tudunk új felhasználót hozzáadni a rendszerhez

Az ötödik Profile menüpontra tudjuk megtekinteni a jelenleg bejelentkezett felhasználónak az adatait és szintén ezen az oldalon tudunk a felhasználóból kijelentkezni

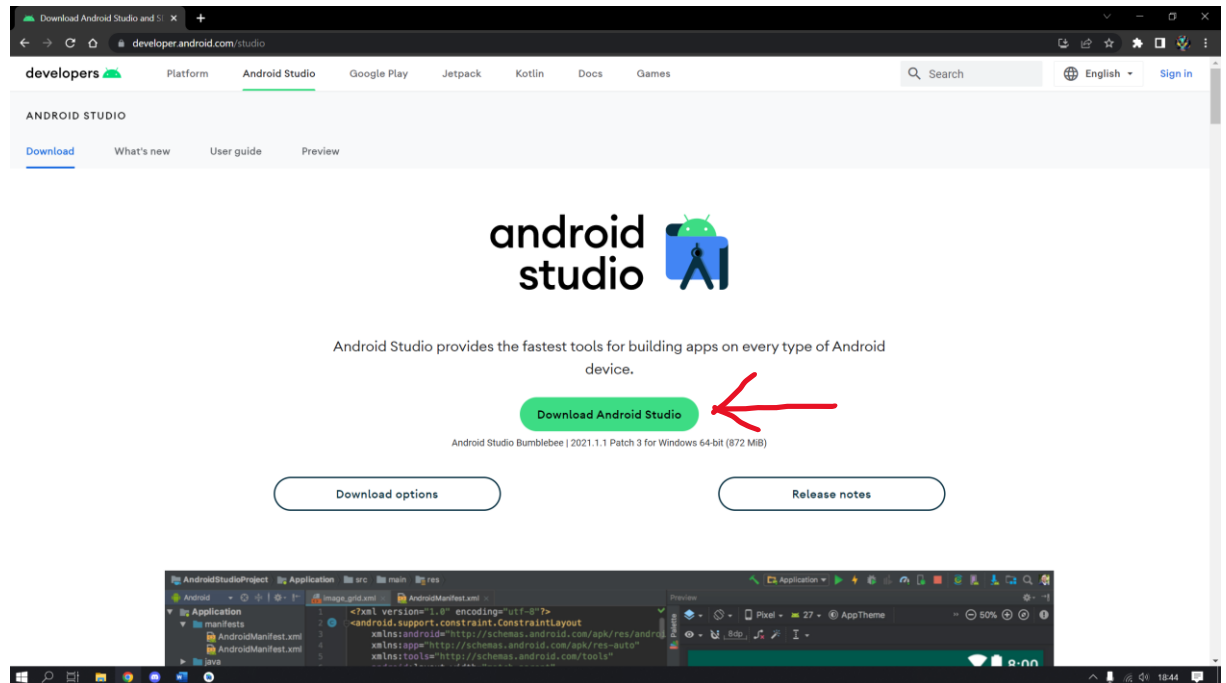
1.4.2. Rendszerkövetelmények

Ajánlott rendszerkövetelmények

1. Android 7.0 (Nougat) vagy a-felett

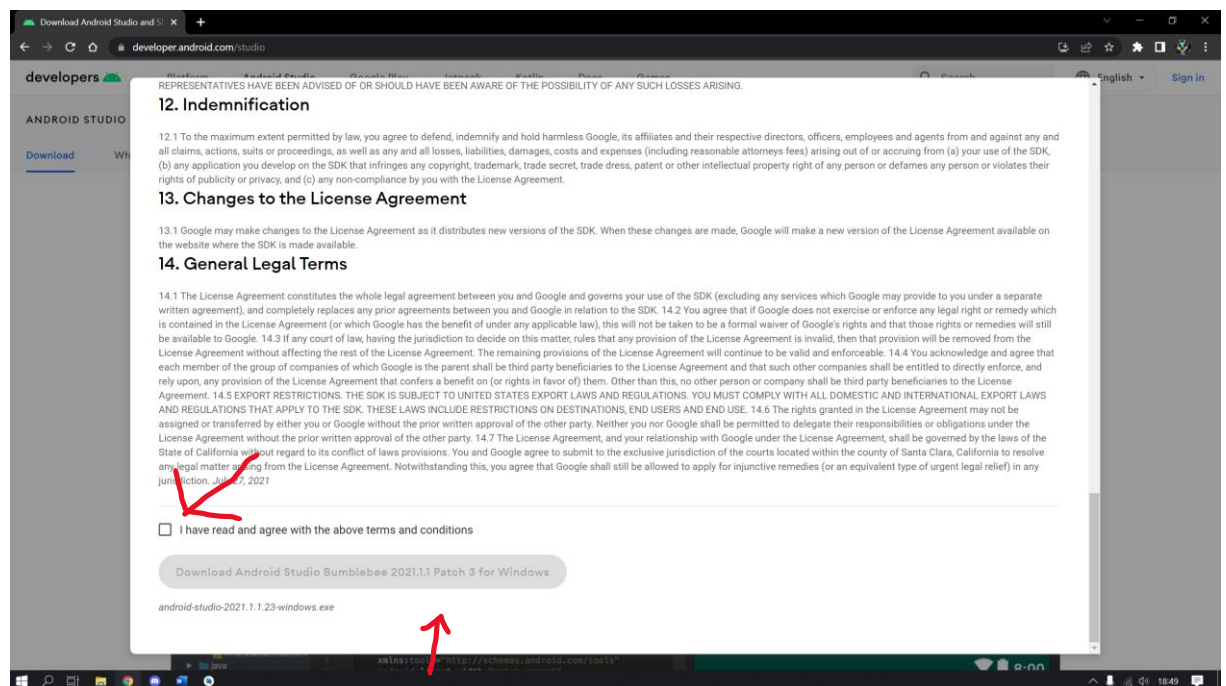
1.4.3. A program telepítésének és konfigurálásának a leírása

Ahhoz, hogy az android stудиot letölthessük először is fel kell mennünk az [android studio](https://developer.android.com/studio) weboldalára és rányomni a nagy zöld gombra.



8. ábra - Android Studio weblap

Miután elolvastuk a felhasználási feltételeket és kipipáljuk a négyzetet újfent rákattintunk a nagy zöld gombra. Elfogja kezdeni letölteni az android studio telepítőjét.

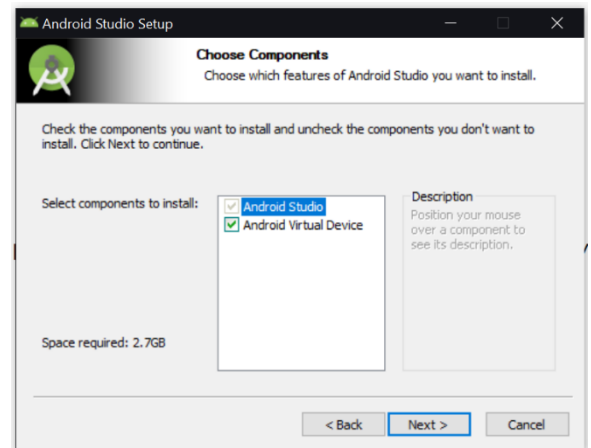


9. ábra - Android Studio felhasználási feltételek

Miután letöltötte nyissuk meg. Az első két oldalon nyomjunk a next gombra.

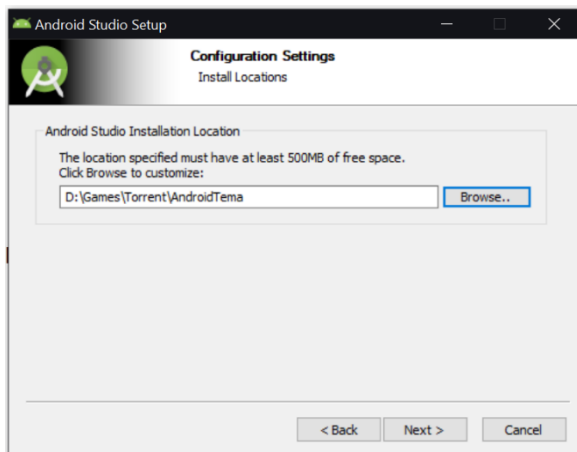


10. ábra - Android Studio Setup első oldal

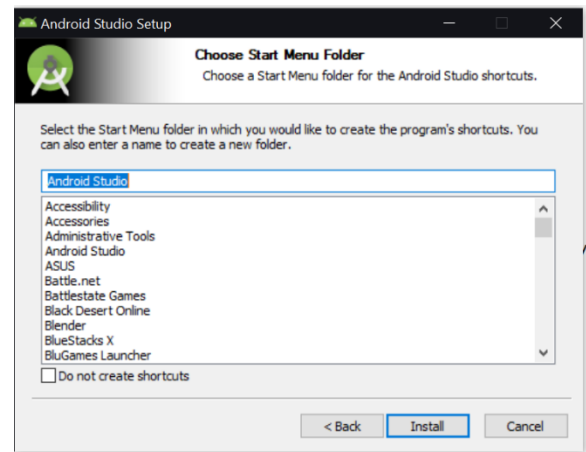


11. ábra - Android Studio Setup második oldal

A harmadik oldalon válassza ki, hogy hova szeretné telepíteni az alkalmazást majd nyomjon rá a next gombra. A következő oldalon semmi teendő nincs csak nyomjon rá az Install gombra és telepítse az alkalmazást.

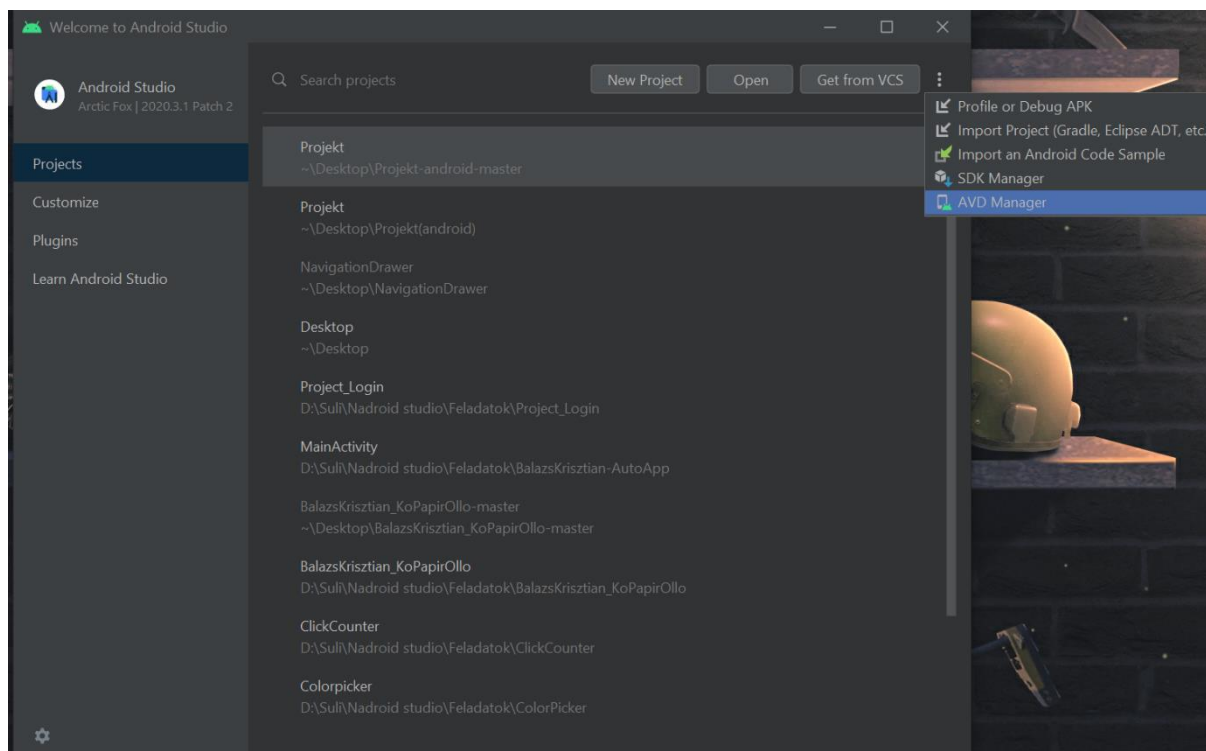


13. ábra - Android Studio Setup harmadik oldal



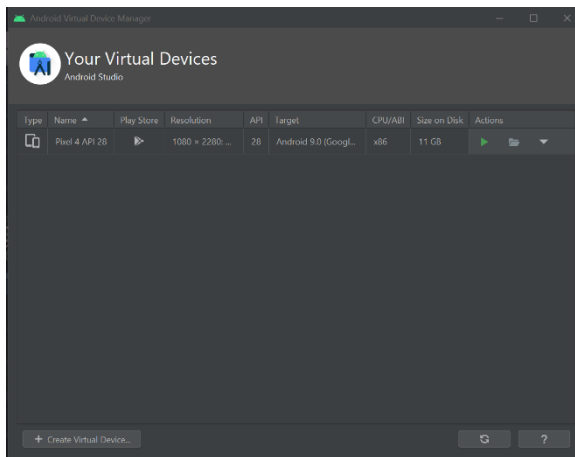
12. ábra - Android Studio Setup negyedik oldal

Miután letelepített az alkalmazás nyissa meg és kattintson rá a három kis pöttyre jobb fölül. A lenyíló menüben kattintson rá az AVD managerre.

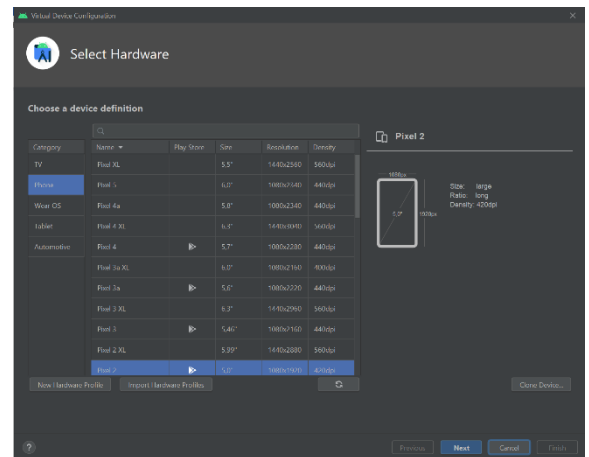


14. ábra - Android Studio AVD Manager

Kattintson rá a bal alsó sarokban a „+ Create Virtual Device...” gombra. Ha van preferált telefon típus, amit szeretne használni akkor azt válassza ki, egyébként nyomjon rá a kék next gombra.

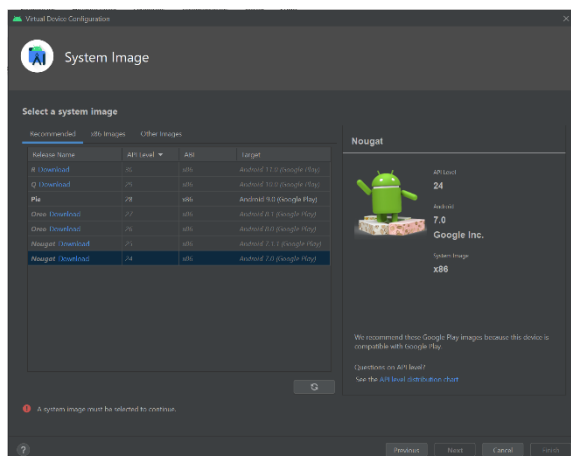


16. ábra - Android Studio emulátor választó

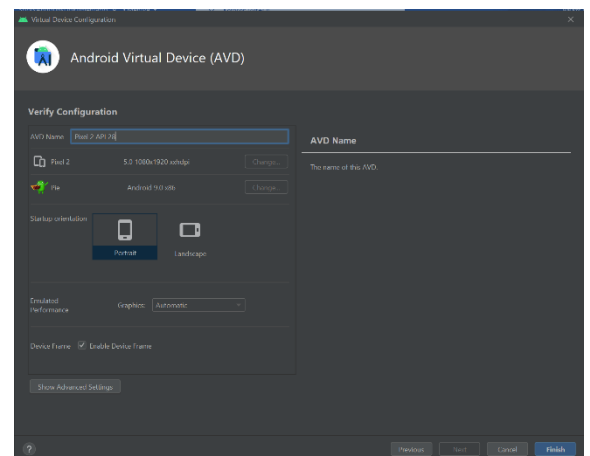


15. ábra - Android Studio telefon választó

A System image oldalon belül válasszon ki és/vagy töltsön le egy tetszőleges operációsrendszer disztribúciót majd nyomjon a kék next gombra.

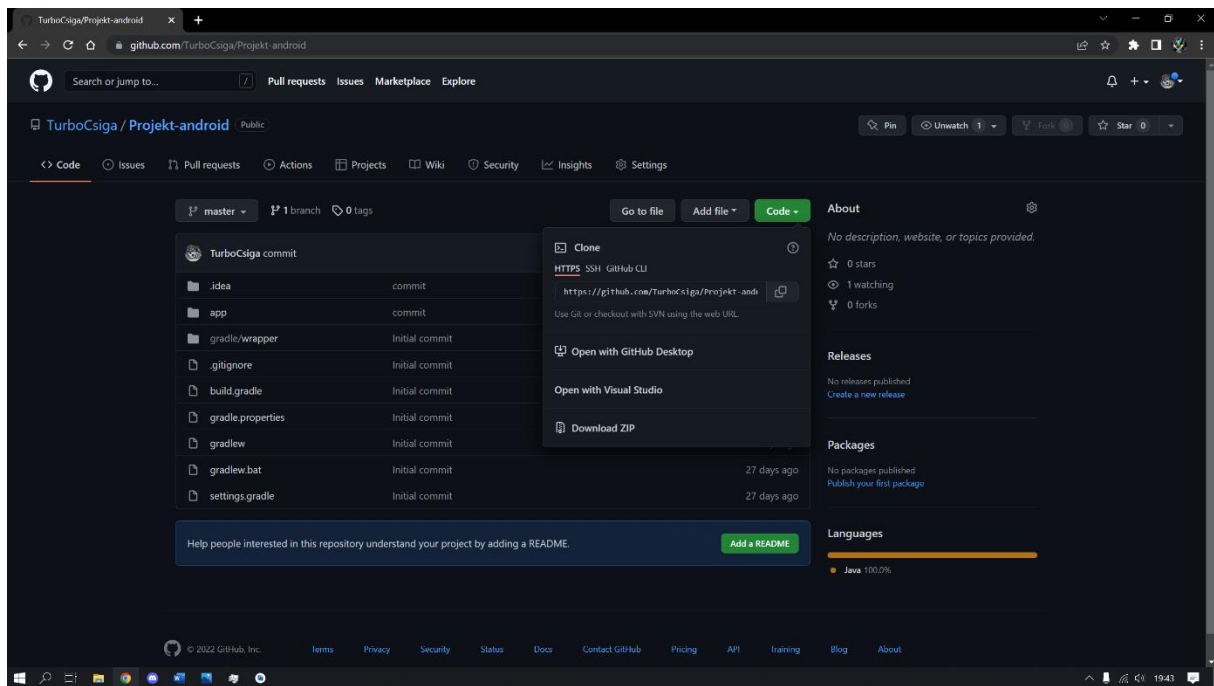


16. ábra - Android Studio operációs rendszer választó



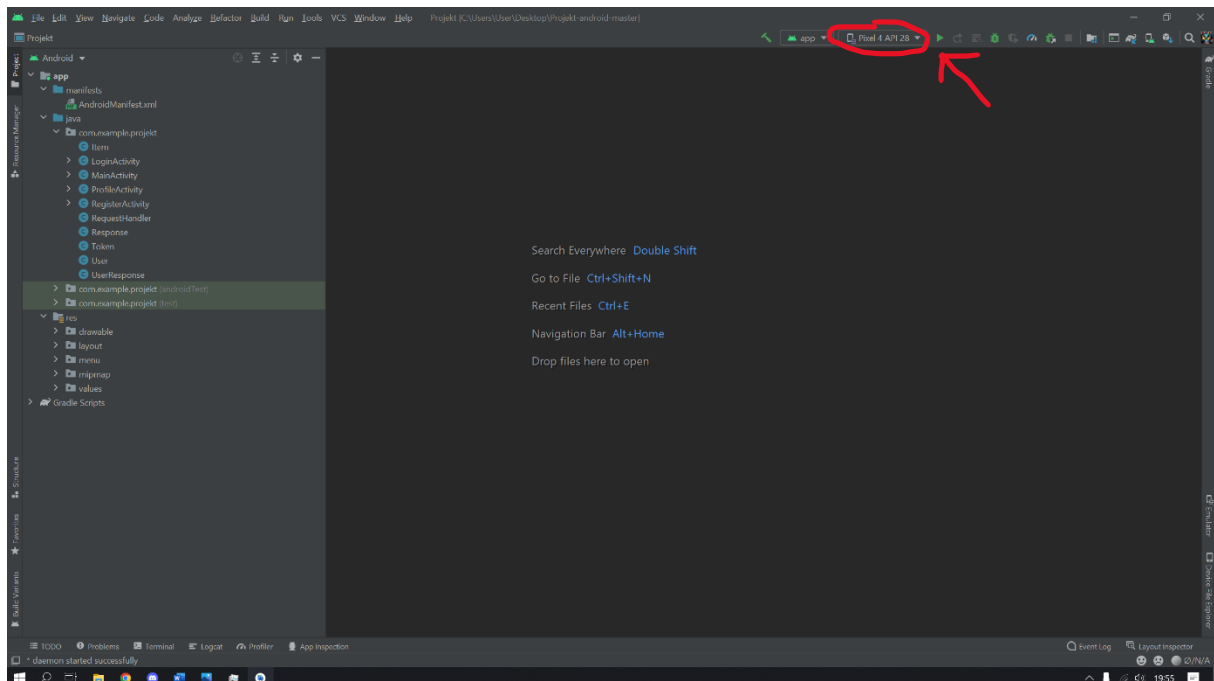
18. ábra - Android Studio AVD finish

Az utolsó oldalon elnevezheti az emulátort, ha szeretné és utána nyomjon rá a kék finish gombra.



19. ábra - Github Android Projekt repository

Miután készen lett az android studio és az emulátor telepítésével már csak az alkalmazást kell letöltenie. Kattintson rá [erre a linkre](#) ami elviszi a github repomba. Itt nyomjon rá a zöld code gombra és nyomjon rá a „Download ZIP” menüpontra. Ezt helyezze el egy önnek tetsző helyen és telepítse ki. Kitelepítés után csak bele kell húzni a mappát az android stuioiba.



20. ábra - Android Studio applikáció elindítása

Főnt válassza ki az emulátort majd nyomjon rá a zöld háromszögre. Ez elfogja indítani az emulátort, valamint az alkalmazást.

1.4.4. A program használatának a részletes leírása

Az oldalmenün keresztül elnavigálunk a Register menüponthoz, ahol is létrehozunk egy új felhasználót. Ezután szintén az oldalmenüből eljutunk a login menüponthoz, ahol az újonnan létrehozott felhasználónkba belépünk. A főoldalon kiválasztjuk a nekünk tetsző terméket és az order gomb megnyomásával megrendeljük azt. Ez után megnyitjuk a profile menüpontot és kijelentkezünk a felhasználónkból

1.5. Web (Szalai Szabolcs)

1.5.1. A program általános specifikációja

Regisztráció vagy bejelentkezés után lehetőség van felhasználói profil szerkesztésére. A profil menüben meg lehet változtatni a felhasználónevet, email címet, és a jelszót. Továbbá lehetőség van rendelések állapotának a megnézésére és a szállítási adatok hozzáadására vagy szerkesztésére.

A termékek oldalain vagy a főoldalon a kosárhoz ad gomb segítségével hozzáadhatjuk az adott terméket a kosárhoz, amit a kosár oldalon meg is tekinthetünk. A kosárban lehetőség van a termékek mennyiségének növelésére, csökkentésére, illetve el is távolíthatjuk a terméket a kosárból.

Rendelés leadása után a profil oldalon a rendelések menüponthoz tekinthető meg az összes eddig leadott rendelés.

1.5.2. Rendszerkövetelmények

Ajánlott rendszerkövetelmények

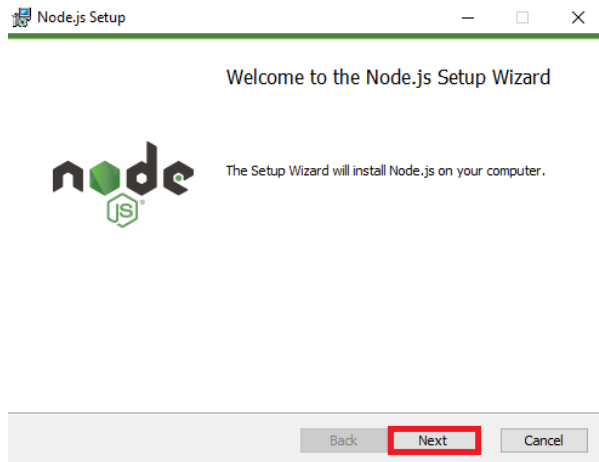
1. Windows 10, Windows 11
2. Legalább 16.14.2 verziójú [Node.js](#)

Ajánlott böngésző verziók

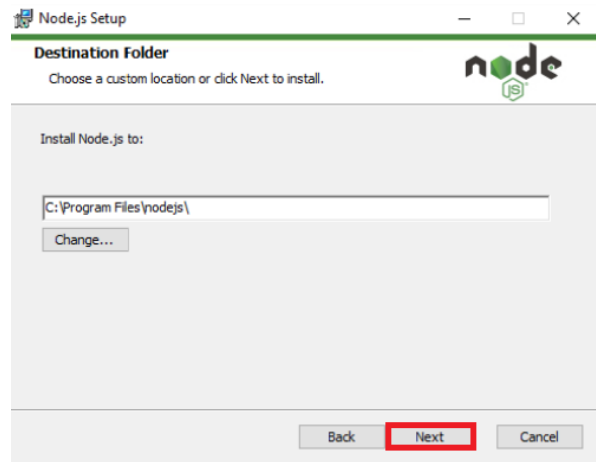
1. Chrome: legújabb
2. Edge: legújabb
3. Safari: legújabb
4. Opera: legújabb

1.5.3. A program telepítésének és konfigurálásának a leírása

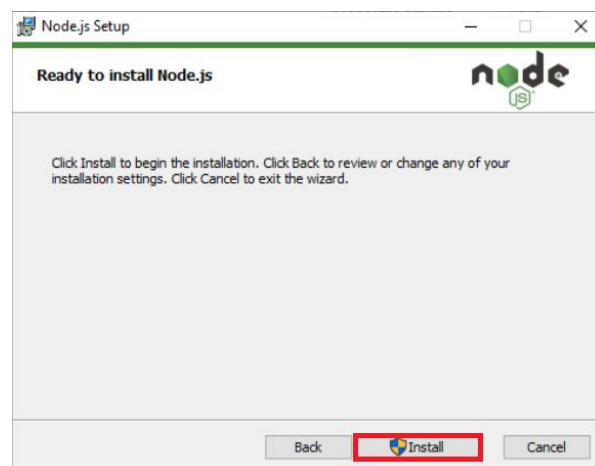
Ahhoz, hogy a program telepítését végbe lehessen vinni, telepíteni kell a [Node.js](#) legalább 16.14.2 verzióját.



22. ábra - Node telepítő első oldal

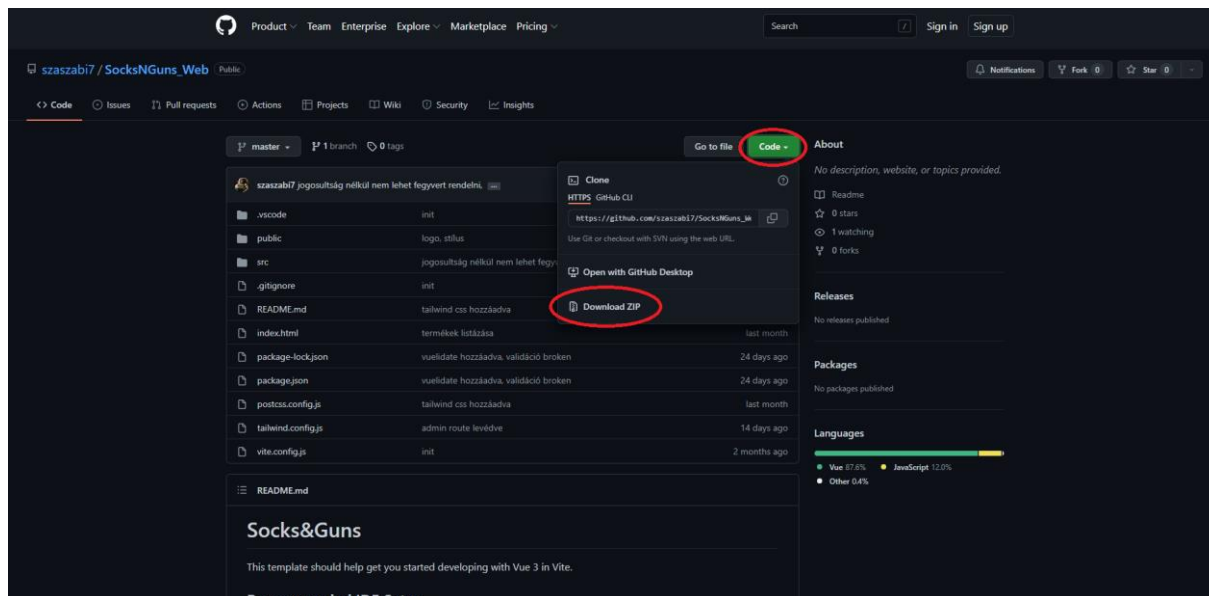


21. ábra - Node telepítő második oldal



23. ábra - Node telepítő harmadik oldal

A Node.js telepítése után le kell tölteni a program forráskódját [GitHub](#)-ról. Kattintsunk a zöld színű Code gombra, majd nyomjunk a Download Zip gombra.



24. ábra - Github Vue Projekt repository

Ha ezt megtettük akkor a letöltött zip fájlt ki kell csomagolni. A kicsomagolt mappában indítsuk el az **install.bat** fájlt. Ha a telepítés befejeződött indítsuk el a **run.bat** fájlt, ez automatikusan elindítja az alkalmazást. A futó alkalmazást a terminálban megjelenő linkről tudjuk elérni.

1.5.4. A program használatának a részletes leírása

A weboldal megnyitásakor egy kevés időnyi töltés után megjelenik a főoldal, ahol az összes termék található.

Az oldal tetején látható a navigációs sáv. Ezen a sávon, a bal oldalon látható az oldal logója és a neve, ha rányomunk akkor a főoldalra irányít át. Középen található a két fő kategória. Ezen elemekre kattintásakor az adott kategóriához tartozó termékek jelennek meg a főoldalon. A navigációs sáv jobb oldalán egy kereső ikon, egy kosár ikon és egy profil ikon található. A kereső ikon átirányít a kereső oldalra, ahol név alapján lehet a termékekre rákeresni. A következő a kosár ikon, ami a kosár oldalát hozza be. Itt található az összes olyan termék, amely hozzá lett adva a kosárhoz. A kosártól jobbra a profil ikon látható. Ez az ikon, ha még nem jelentkezünk be akkor a bejelentkezés oldalra irányít át. Ha a felhasználó be van jelentkezve akkor a saját profil oldalára irányít az oldal. Továbbá, ha bejelentkezett a felhasználó akkor a profil ikontól jobbra megjelenik egy kijelentkezés ikon.

Egy adott termék nevére vagy képére kattintva megjeleni a terméknek az oldala, ahol a termék nevét, kategóriáját, árát és egy a leírását láthatjuk.

A profil oldalon az éppen bejelentkezett felhasználóhoz tartozó adatokat lehet módosítani. Email címet, felhasználónevet és jelszót is lehet változtatni. A szállítási adatok menüpontban lehetőség van megadni, szerkeszteni vagy törölni a felhasználó címét. A rendelések menüpontban látható az összes korábbi rendelés és annak státusza, teljes összege és a rendelés leadásának dátuma.

Összegzés

Szalai Szabolcs:

Sok órányi munka után elmondhatom, hogy ez egy teljesen új és érdekes élmény volt. Számos új technológiával, és a problémák gyors és hatékony megoldásának módjával ismerkedtem meg.

Kezdetben kicsit nehéz volt csapatban dolgozni, de az idő előrehaladtával ez egyre könnyebbé vált. A projekt webes részét én csináltam, a backend részét pedig csapatban készítettük. A projekt elkészítése nem volt egyszerű, mert nem volt sok tapasztalatom, és még soha nem csináltam ehhez hasonlót. Az iskolában tanult technológiákkal, és dokumentációk olvasgatásával, kezdtem megérteni az alapokat. Innentől kezdve sokkal kisebb erőfeszítéssel tudtam megoldani nehezebb feladatokat is. Sokkal gyorsabban találtam megoldást azokra a problémákra, amelyekre korábban csak hosszú órákon át tartó kutatás után tudtam megoldani. Úgy gondolom, hogy ez a projektmunka segített abban, hogy jobb programozóvá, csapatjátékosná váljak, és remélem, hogy a jövőben is részt vehetek ilyen érdekes és sok tapasztalatot, tudást adó projekteken.

Bár nem tudtam megszerezni azt a tapasztalatot, amivel a legjobb lehetek, de úgy gondolom, hogy ez a projekt megadta nekem azokat az alapokat, amelyekre a jövőben építkezni tudok.

Balázs Krisztián:

Ez a projektmunka ezer és egy akadályt görgetett elém mégis úgy érzem sikeresen áttudtam evickélni rajtuk.

Számomra az androidra való fejlesztés egy nagyon új és idegen élmény volt. Ahogy haladtunk előre a tananyaggal egyre több és több akadályt tudtam magam alá gyűrti mégsem érzem úgy, hogy ezt a nyelvet vagy fejlesztői környezetet teljesen el tudtam volna sajátítani. Még nagyon sok munka és akadály áll előttem míg azt tudom mondani magamról, hogy el tudtam sajátítani az android csínját-bínját, de szorgosan haladok célom felé és nem adom fel. Talán még így is egy sokkal nagyobb akadály volt számomra a csapatmunka ilyes formája. Soha nem hittem volna, hogy egy alkalmazás fejlesztéséhez ennyire összehangolt és kommunikatív csapatmunka kell. Rengeteg időt töltöttünk az „írásztalnál” tervezéssel mire

megtudtuk valósítani a backendet. Egy nagy kihívás volt a backend mivel még nem dolgoztam api-kezeléssel egyáltalán, egy ekkor projektbe így egy kicsit nehéz és félelmetes volt belevágni.

Úgy érzem a projektmunka alatt rengeteget fejlődtem, mint programozó és mint csapatjátékos.

Források

- <https://laravel.com/docs/9.x>
- <https://vuejs.org/>
- <https://tailwindcss.com/>
- <https://heroicons.com/>
- <https://vuex.vuejs.org/>
- <https://router.vuejs.org/>
- <https://vuelidate.js.org/>
- <https://axios-http.com/docs/intro>
- <https://stackoverflow.com/>
- <https://developer.android.com/docs>
- <https://github.com/google/gson>

Ábrajegyzék

1. ábra - Ütemterv	2
2. ábra - Android Menu Terv	11
3. ábra - Android Main Terv	11
4. ábra - Android Login Terv	12
5. ábra - Android Register Terv	12
6. ábra - felületterv - főoldal	22
7. ábra - felületterv - profil menü.....	22
8. ábra - Android Stusio weblap	35
9. ábra - Android Studio felhasználási feltételek	35
10. ábra - Android Studio Setup első oldal	36
11. ábra - Android Studio Setup második oldal.....	36
12. ábra - Android Studio Setup negyedik oldal.....	36
13. ábra - Android Studio Setup harmadik oldal	36
14. ábra - Android Studio AVD Manager.....	37
15. ábra - Android Studio telefon választó	38
16. ábra - Android Studio emulátor választó	38
17. ábra - Android Studio operációs rendszer választó	38
18. ábra - Android Studio AVD finish.....	38
19. ábra - Github Android Projekt repository	39
20. ábra - Android Studio applikáció elindítása	39
21. ábra - Node telepítő második oldal	42
22. ábra - Node telepítő első oldal	42
23. ábra - Node telepítő harmadik oldal	42
24. ábra - Github Vue Projekt repository	43