



**BAJAI SZAKKÉPZÉSI CENTRUM**  
**KALOCSAI DÓZSA GYÖRGY TECHNIKUM ÉS KOLLÉGIUM**  
**6300 KALOCSA, Asztrik tér 5-7.**  
**OM azonosító: 203028**  
**Tel.: 36-78-461-600      Fax.: 36-78-461-178**  
**E-mail: [info@dozsakalocsa.hu](mailto:info@dozsakalocsa.hu)   [www.dozsakalocsa.hu](http://www.dozsakalocsa.hu)**

---



**Szoftverfejlesztő és -tesztelő**

## **Dokumentáció**

**Képzőhely neve és címe: Bajai SZC Kalocsai Dózsa György Technikum és Kollégium**

**Vizsgázók nevei: Kovács Sándor, Szeghalmi László, Murányi Marcell**

**Vizsga helyszíne: 6300 Kalocsa, Martinovics utca 2-4.**

**2025.**

# Tartalom

Nyilatkozatok .....	6
Bevezetés.....	7
Zárodolgozat témájának megindoklása, projekt célja.....	7
Célközönség.....	7
Fejlesztői dokumentáció .....	8
Rendszerkövetelmény(frontend) .....	8
Google Chrome .....	8
Mozilla Firefox.....	8
Microsoft Edge .....	9
Safari .....	10
Opera .....	10
Rendszerkövetelmény(backend).....	11
XAMPP .....	11
Visual Studio Code.....	11
Szoftverek, amiket használtunk .....	12
Windows 10 .....	12
Visual Studio Code.....	12
XAMPP .....	12
Node.js .....	13
Npm .....	13
Programozási nyelvek, amiket használtunk.....	14
HTML.....	14
CSS .....	14
Javascript (JS) .....	14

MySQL.....	15
EJS .....	15
Adatbázis szerkezete .....	16
Az adatbázis táblái.....	16
Products tábla .....	17
Orders tábla .....	18
Users tábla .....	19
Algoritmusok a weboldalon .....	20
Regisztráció(controller) .....	20
Regisztráció(router).....	22
Bejelentkezés(controller).....	23
Bejelentkezés(router) .....	25
Tesztelés.....	26
Hibás kód .....	26
Helyes kód.....	27
Kinek mi volt a feladata a projektben.....	28
Szeghalmi László.....	28
Murányi Marcell .....	28
Kovács Sándor .....	28
Továbbfejlesztési lehetőségek a weboldalon .....	29
Felhasználói dokumentáció .....	30
Rendszerkövetelmények.....	30
Google Chrome .....	30
Mozilla Firefox.....	31
Microsoft Edge .....	31

Safari .....	32
Opera .....	32
Program telepítése .....	33
A webshop bemutatása .....	33
Név.....	35
Email .....	35
Telefonszám .....	36
Jelszó.....	36
Email .....	37
Jelszó.....	37
Leírás oldal .....	40
Kosár oldal.....	40
Köszönetnyilvánítás.....	42

1. ábra Adatbázis szerkezete .....	16
2. ábra Products tábla .....	17
3. ábra Orders tábla .....	18
4. ábra Users tábla .....	19
5. ábra Webshop főoldal .....	33
6. ábra Regisztrációs gomb.....	34
7. ábra Regisztrációs oldal .....	35
8. ábra Bejelentkezés gomb .....	36
9. ábra Bejelentkezés oldal.....	37
10. ábra Bejelentkezett főoldal .....	38
11. ábra Webshop gomb .....	38
12. ábra Webshop oldal .....	39
13. ábra Termék ikon .....	39
14. ábra Termék leírása .....	40
15. ábra Kosár .....	40
16. ábra Hamburger menü .....	41
17. ábra Profil.....	41

# Nyilatkozatok

Alulírottak Kovács Sándor (2005.07.01. Dunaújváros), Szeghalmi László (2005.11.24. Kalocsa) és Murányi Marcell (2006.01.10. Baja) jelen nyilatkozat aláírásainkkal kijelentjük, hogy a Szoftverfejlesztő és –tesztelő szakmai vizsgára készített program és dokumentáció saját munkánk eredménye. Nyilatkozunk, hogy az egyes dokumentumokban foglalt tények és adatok a megadott forrásmunkák felhasználásán alapulnak.

Tudomásul vesszük, hogy amennyiben a dokumentációmban a vizsgabizottság tagjai olyan dokumentumot találnak, amely problémaorientált feladat esetén 20%-nál nagyobb mértékben nem önálló munkánk eredménye vagy leíró/összegző feladat esetén nem saját kutatásunk eredménye, és amelyből a források kiválasztásának gondolatmenten vagy relevanciája nem követhető, valamint nem tartalmazza a pontos forrásmegjelöléseket, abban az esetben a vizsgabizottság tagjai a dokumentációt nem fogadják el. Ezen pont alól kivételt képeznek a csoportos projektek, komplex feladatok és esettanulmányok, melyek keletkezési körülményeit a bevezető tartalmazza.

Jelen nyilatkozat aláírásával tudomásul vesszük, hogy amennyiben bizonyítható, hogy a dokumentációt nem magunk készítettük, a dokumentáció bemutatása nem kezdhető meg, így ez a vizsgarész elégtelen értékelést kap.

Hozzájárulunk ahhoz, hogy a vizsgabizottság tagjai, jegyzője és a *Vizsgaszervező* munkatársai a dokumentáció tartalmát megismerhessék.

Kijelentjük, hogy a dokumentációra kapott értékelésünket és dokumentációnkat további felhasználásra sem kinyomtatva sem pedig elektronikus úton nem továbbítom senkinek.

Kalocsa 2025.03.15.

Aláírások

# Bevezetés

## Záródolgozat témájának megindoklása, projekt célja

A mai világban az emberek ki vannak téve az egészségtelen ételeknek ezáltal egyre jobban szükségünk van az egészséges és mozgás dús életmódra. Ebben a szellemben a konditermek jelentős szerepet töltenek be a rendszeres testmozgás biztosításában, valamint az egészség megőrzésében. A vizsgaremekünk célja, hogy egy olyan weboldalt fejlesszünk, ami egy konditerem termékeit és az egészséges életmód fontos szükségleteit ajánlja a vásárlóknak és hasznos információkkal látja el őket. A témaválasztásunk alapja, hogy mi magunk is rendszeresen járunk edzeni, és ezt a tudásunkat akarjuk ezzel a weboldallal bemutatni. Az oldal nem csak a konditerem szolgáltatásait mutatja be, hanem hasznos funkciókkal segíti a felhasználókat, mint például órarendek, edzői profilok és online bejelentkezési lehetőségek.

A projekt szakmai célja a modern webfejlesztési technológiák alkalmazásával egy felhasználóbarát, esztétikus és könnyen navigálható weboldal elkészítése. Ezzel nemcsak a felhasználói élményt kívánjuk növelni, hanem a konditerem népszerűségét is elősegíteni, hogy minél több embert érjenek el a fontos információk.

## Célközönség

Kinek ajánljuk a projektmunkánkat?

Azoknak az embereknek, akik szeretnék egy kiegyensúlyozottabb életmódra váltani, mozogni, egészségesen táplálkozni és még segítséget is kérnének tapasztalt edzőktől, akik szeretnék belemélyedni például a testépítés rejtelmeibe vagy éppen bármilyen sportot ki akarnak próbálni, akár hosszútávon még űzni is.

# Fejlesztői dokumentáció

## Rendszerkövetelmény(frontend)

(klienskövetelmény)

### *Google Chrome*

Operációs rendszerek				
Windows	macOS	Linux	Android	iOS
<b>7-11(32 és 64bit)</b>	10.11 vagy újabb	legtöbb verziója alkalmas	4.4 vagy újabb	12.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 350MB szabad tárhely szükséges

### *Mozilla Firefox*

Operációs rendszerek				
Windows	macOS	Linux	Android	iOS
<b>7-11(32 és 64bit)</b>	10.12 vagy újabb	legtöbb verziója alkalmas	5.0 vagy újabb	13.0 vagy újabb



Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 200MB szabad tárhely szükséges

### *Microsoft Edge*

Operációs rendszerek				
Windows	macOS	Linux	Android	iOS
<b>7-11(64bit)</b>	10.12 vagy újabb	legtöbb verziója alkalmas	4.4 vagy újabb	12.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 250MB szabad tárhely szükséges

## *Safari*

Operációs rendszerek		
Windows	macOS	iOS
<b>7-11(64bit)</b>	10.12 vagy újabb	11.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 200MB szabad tárhely szükséges

## *Opera*

Operációs rendszerek			
Windows	macOS	Linux	Android
<b>7-11(32 és 64bit)</b>	10.12 vagy újabb	legtöbb verziója alkalmas	5.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 200MB szabad tárhely szükséges

# Rendszerkövetelmény(backend)

(szerverkövetelmény)

## *XAMPP*

- **Operációs rendszer**
  - **Windows:** 7, 8, 10, 11 (32 és 64 bit)
  - **Linux:** bármely verziója alkalmas a XAMPP futtatásához
  - **macOS:** (macOS 10.12 vagy újabb)
- **Hardver**
  - **Processzor:** Minimum 1Ghz
  - **RAM:** Legalább 2GB, de inkább legyen 4GB-nál nagyobb
  - **Tárhely:** Nem kevesebb mint 400MB-os szabad tárhely csak XAMPP-hoz

## *Visual Studio Code*

- **Operációs rendszer**
  - **Windows:** 7, 8, 10, 11 (64 bit)
  - **Linux:** legtöbb verzió alkalmas a Visual Studio Code futtatásához
  - **macOS:** (macOS 10.11 vagy újabb)
- **Hardver**
  - **Processzor:** Legalább 1Ghz, de erősebb processzor ajánlott
  - **RAM:** Minimum 1GB, de ajánlott a 2GB-nál nagyobb
  - **Tárhely:** Legyen minimum 200MB-os szabad tárhely
- **Szoftver**
  - **.NET Framework(Windows):** Legalább 4.5 vagy újabb verzió szükséges

## Szoftverek, amiket használtunk

### *Windows 10*



A Windows 10 az az operációs rendszer, ami a legismertebb és a legfelkapottabb, jelenleg már nem a legújabb, de talán a legmegbízhatóbb. Más operációs rendszert nem használtunk mivel a Windows 10-hez szoktunk hozzá.

### *Visual Studio Code*



Egy elég felkapott kódszerkesztő szoftver és ez sem véletlen, mivel nagy segítséget nyújt a kódok írásában. Sok mintával mutatja az utat a megfelelő program felé és ezt az egyet használtuk mivel ezt tanultuk a tanárainktól.

### *XAMPP*



Lehet, hogy nem a XAMPP a legjobb adatbázis kezelő, de mivel mi ezt ismerjük a legjobban mert már lassan 3 éve ezzel dolgozunk ezáltal erre esett a választásunk.

## *Node.js*



A Node.js egy nyílt forráskódú, JavaScript futtatókörnyezet, amely a JavaScript-et szerver oldalon is tudja futtatni. Eredetileg azért jött létre, hogy lehetővé tegye a JavaScript alkalmazások fejlesztését szerver oldalon, ezáltal a JavaScript használható mind frontend és backend programozásra is. A Node.js-t lehet használni RESTful API-k készítésére és fejlesztésére meg skálázható webszolgáltatások létrehozására is.

## *Npm*



Az npm (Node Package Manager) egy csomagkezelő, amit a Node.js-ben használunk. Feladata a JavaScript könyvtárak és modulok kezelését leegyszerűsíteni a felhasználó számára. Az npm lehetővé teszi a fejlesztők számára, hogy könnyedén telepítsenek, frissítsenek, kezeljenek és osszanak meg csomagokat (modulokat), amelyek más fejlesztők által létrehozott kódokat tartalmaznak. Az npm egyúttal egy online csomagregisztrációs adatbázis is, ahol több ezer, a világ minden tájáról származó csomagot találhatunk.

## Programozási nyelvek, amiket használtunk

### *HTML*



HyperText Markup Language, vagyis HTML, egy leíró nyelv, amelyet weboldalak készítésére és struktúrájuk meghatározására használnak. A HTML segítségével a fejlesztők különböző elemeket, mint szövegek, képek és linkek, helyezhetnek el a weboldalon. A HTML dokumentumok címkékből, azaz tag-ekből állnak, melyek segítségével a böngészők értelmezik és megjelenítik a tartalmat. A HTML alapvető technológia a webfejlesztésben, és más eszközökkel, mint CSS és JavaScript, együttműködve hozzájárul a modern weboldalak létrehozásához.

### *CSS*



Cascading Style Sheets egy stíluslap nyelv, amely a HTML weboldalak kinézetének formázását segíti. Segítségével meghatározhatjuk a betűszíneket, háttérképeket, margókat és egyéb dizájnelemeket. A CSS lehetővé teszi, hogy a weboldalak megjelenését elkülönítsük a tartalomtól, így könnyebbé válik a karbantartás és a dizájn módosítása.

### *Javascript (JS)*



A JavaScript egy magas szintű programozási nyelv, amely weboldalak fejlesztését segíti, és lehetővé teszi a dinamikus elemek hozzáadását, például animációkat. A JavaScript segítségével interaktív funkciókat, választható tartalmakat és más látványos elemeket építhetünk be a weboldalakba, így javítva a felhasználói élményt.

## MySQL



A MySQL nem programozási nyelv, hanem egy relációs adatbázis-kezelő rendszer, amelyet adatbázisok és webalkalmazások kezelésére használnak. A MySQL segítségével adatokat tárolhatunk és kérdezhetünk le, így lehetségessé téve dinamikus weboldalak és alkalmazások működését. Az ilyen típusú rendszerek futtatásához gyakran alkalmazzuk a XAMPP-ot, ami egy ingyenes, nyílt forráskódú programcsomag, amely tartalmazza a MySQL-t és más szükséges eszközöket, mint a Apache webservert, így segít a fejlesztésben és tesztelésben.

## EJS



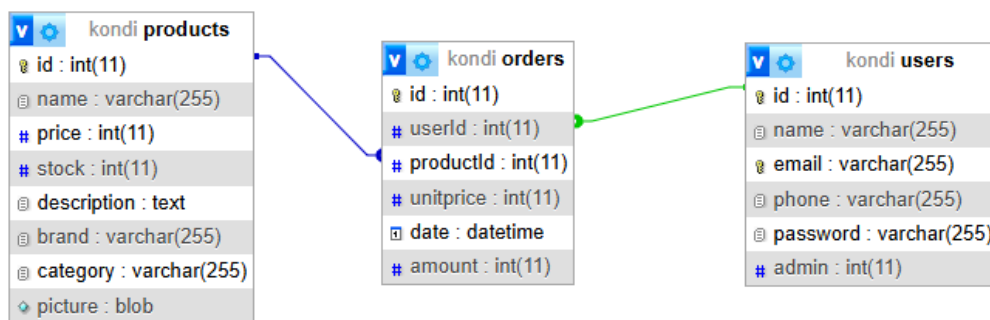
Az EJS egy egyszerű sablonnyelv, amely lehetővé teszi HTML kód generálását sima JavaScript segítségével. Nincs vallásos hozzáállás a dolgok megszervezéséhez. Nincs iteráció és vezérlési áramlás újra feltalálása. Csak sima JavaScript.<sup>1</sup>

---

<sup>1</sup> Forrás: <https://ejs.co/>

## Adatbázis szerkezete

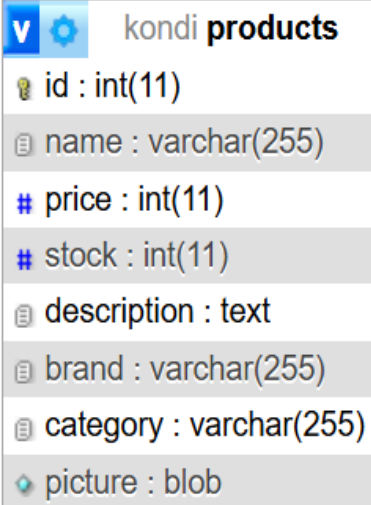
### *Az adatbázis táblái*



1. ábra Adatbázis szerkezete



## *Products tábla*



id : int(11)
name : varchar(255)
price : int(11)
stock : int(11)
description : text
brand : varchar(255)
category : varchar(255)
picture : blob

2. ábra Products tábla

**id(Termékek azonosítója):** Ezen mező az adott termék azonosítója, minden hozzáadott termékkel arányosan növekszik. Ez segít megkülönböztetni, azaz azonosítani a termékeket.

**name (Termékek neve):** Ez a mező felelős a táblában tárolt termékek neveinek a tárolására, hogy meglehessen őket különböztetni egymástól.

**price (Termékek ára):** A mező a termékek árát tárolja magyar valutában, az árakat próbáltuk valósághűre csinálni a vizsga időpontjához képest. Ez a mező nagyon fontos az adatbázisban.

**stock (Termékek mennyisége):** Ez a mező a termékek mennyiségét tárolja például, hogy hány darab elérhető még az adott áruból.

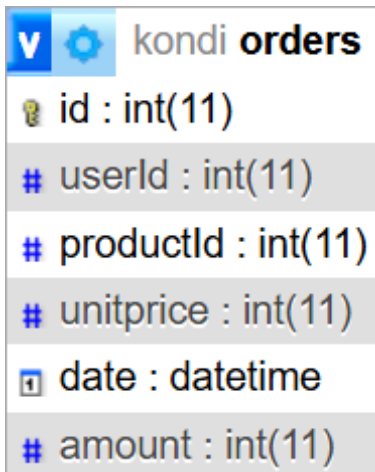
**description (Termékek leírása):** Ezen mező hosszabb szöveges tartalmakra lett kitalálva, amibe bele tudjuk írni a termékek leírásait például tápanyag tartalmát tömegét egyéb hasznos információkat.



**brand (Termékek márkája):** Ez a mező az adott termék márkáját különbözteti meg, így tud például a felhasználó alaposabban rákeresni a dolgokra.

**category (Termékek típusai):** A mező a termékek kategóriáját vagy típusát különíti el így tudunk rákeresni a táplálék kiegészítőkre a felszerelésekre és az edzéstervekre is külön.

**picture (Termék képe):** Ez a mező a termékek képét kívánja eltárolni az adatbázisban és megjeleníteni a webshop fiúlnél. Azért rakunk a termékekhez képet, hogy a vásárló a kiválasztás előtt, látja, hogy néz ki a megvásárolni kívánt terméke.

### *Orders tábla*



	id : int(11)
#	userId : int(11)
#	productId : int(11)
#	unitprice : int(11)
	date : datetime
#	amount : int(11)

3. ábra Orders tábla

**id(Rendelések azonosítója):** Ez a mező az adott rendelés azonosítója, minden plusz rendeléssel arányosan növekszik. Ez segít megkülönböztetni, azaz azonosítani a különböző rendeléseket.

**userId (Vásárlók azonosítója):** Ez a mező az adott vásárlót azonosítja minden egyes vásárlóval arányosan növekszik. Ez segít megkülönböztetni a vásárlókat.

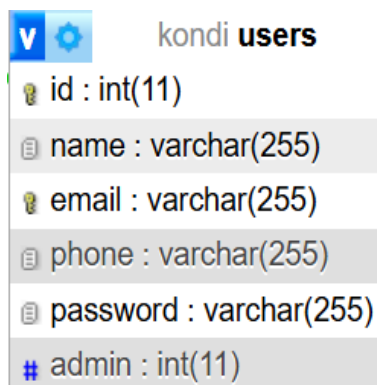
**productId(Termékek azonosítója):** Ez a mező az adott termék azonosítója minden hozzáadott termékkel arányosan növekszik. Ez segít megkülönböztetni azaz azonosítani termékeket. A termékek tábla id azonosítójának az idegen kulcsa.







**unitprice (Rendelések összára):** Ez a mező felelős a rendelések összárának a tárolására. Így könnyebben tud vele számolni az algoritmus. A mező adata a vásárló számára is látható a rendelési kosárban

**date (Rendelés ideje):** Ez a mező felelős a leadott rendelések időpontjának a tárolására. Ez az adat fontos a vásárló számára is de még fontosabb a weboldalt üzemeltetőknek.

**amount (Rendelések mennyisége):** Ez a mező a rendelések mennyiségéért felel így tudjuk tárolni a rendelések számát és ez miatt elengedhetetlen a unitprice mező kivitelezéséhez is.

## Users tábla



	id : int(11)
	name : varchar(255)
	email : varchar(255)
	phone : varchar(255)
	password : varchar(255)
	admin : int(11)

4. ábra Users tábla

**id(Vásárlók azonosítója):** Ez a mező felelős a vásárlók azonosításáért minden egyes vásárlóval arányosan növekszik az értéke. Ez segít megkülönböztetni azaz azonosítani a vásárlókat.

**name (Vásárlók nevei):** Ez a mező felelős az adott vásárló nevének a tárolására. A mező adatait a felhasználó tölti fel a regisztráció során

**email (Vásárlók email címe):** Ez a mező felelős az adott vásárló email címének a tárolásáért. Ezt a mezőt is a vásárló tölti meg a regisztráció idejében.

**phone (Vásárlók telefonszámai):** Ez a mező az adott vásárló telefonszámának a tárolását végzi el. Szintén a vásárló tölti fel mikor regisztrál a weboldalra.

**password (Vásárlók jelszavai):** Ez a mező az adott vásárló által használt jelszónak a tárolását végzi el. Ugyanúgy a vásárló adja meg a mező adatait a regisztráció során.

**admin:** Ez a mező az admin felhasználókat különíti el a vásárlóktól. Az admin értéke 1 akkor jogosult, hogy hozzáférjen a webshop oldalon belül egy operátor fülhöz, amivel termékeket tud hozzáadni és eltávolítani a weboldalhoz viszont, ha az admin értéke 0 akkor csak vásárolni tud termékeket, és nem jelenik meg neki az operátor fül a webshop jobb szélén.

## Algoritmusok a weboldalon

### *Regisztráció(controller)*

```
const Users = require('../models/users');

exports.getRegisterPage = (req, res) => {
  res.render('register');
};
```

A regisztrációs controllerünkben az első sor oldotta meg a Users modell importálását. A másik kettő pedig a regisztrációs oldal megjelenítéséért felel GET kéréssel.

```
exports.registerUser = async (req, res) => {
  try {
    const { name, email, phone, password } = req.body;
```

Ez a POST kérés dolgozza fel a regisztráció során megadott adatokat. A nevet, email címet, telefonszámot és a jelszót.

```
if (!name || !email || !phone || !password) {
  return res.status(400).send('Minden mezőt ki kell tölteni!');
}
```

A szöveg egy if-ben helyezkedik el, ami feltételes ellenőrzést hajt végre. Minden mező kitöltését ellenőrzi, így nem lesznek üres adatok az adatbázisban.

```
const existingUser = await Users.findOne({ where: { email } });
if (existingUser) {
  return res.status(400).send('Ez az email cím már regisztrálva van!');
}
```

Ebben a kódban azt ellenőrizzük, hogy az email cím vagy a felhasználó létezik-e, ha nem akkor 400-as hibát kapunk és kiírja a képernyőre a felhasználónak, hogy az email címe már fellelhető az adatbázisunkban.

```
await Users.create({
  name,
  email,
  phone,
  password,
});
```

A fent lévő kódban a vásárlót adatait hozzuk létre az adatbázisban. A vásárló nevét, email címét, telefonszámát és a jelszavát viszont nem titkosítva, mivel nem ad hozzá olyan sokat a projektmunkánkhoz.

```
} catch (err) {
  console.error('Hiba:', err);
  res.status(500).send('Hiba történt a regisztráció során.');
}
```

Az alábbi catch blokk a hibákat jelzi a felhasználónak, ha történik valami nem odavaló a regisztráció során. Ha hibát talál akkor küld a felhasználónak egy 500-as általános hibaüzenetet.

## *Regisztráció(router)*

```
const express = require('express');  
const regController = require('../controllers/regController');  
const router = express.Router();
```

A routerrel a regisztrációs controller importálásával lehetővé teszi az oldal műveleteinek kezelését.

```
router.get('/register', regController.getRegisterPage);  
router.post('/register', regController.registerUser);  
module.exports = router;
```

Az első sor a regisztrációs oldal megjelenítéséért felel GET kéréssel. A második pedig az adatok feldolgozását segíti POST kéréssel. A `module.exports = router;` pedig azért felelős hogy a router objektumot exportálja.

## *Bejelentkezés(controller)*

```
const Users = require('../models/users');

exports.getLoginPage = (req, res) => {
  res.render('login', {
    title: 'Bejelentkezés',
    loginMessage: 'Kérlek, jelentkezz be!'
  });
};
```

Az első sor a Users modell importálása. A képen látható maradék kódok a bejelentkezési oldal megjelenítéséért felel még egy üzenetet is küld a vásárlónak a loginMessage paranccsal.

```
exports.loginUser = async (req, res) => {
  try {
    console.log(req.session);

    const { username, password } = req.body;
```

A fenti kód a bejelentkezés feldolgozását csinálja meg, ami a felhasználónévből(email) és a jelszóból áll.

```
const user = await Users.findOne({ where: { email: username } });

if (!user) {
  return res.render('login', {
    loginMessage: 'Nincs ilyen felhasználó!'
  });
}
```

A user constructor megkeresi a beírt felhasználót és ha létezik akkor tovább megy a jelszó feldolgozására viszont, ha nem akkor kiír egy loginMessageet hogy „Nincs ilyen felhasználó!”.

```

if (user.password === password) {
  req.session.user = {
    username: user.name,
    email: user.email,
    admin: user.admin
  };

  if (user.admin === 1) {
    console.log('Admin felhasználó jelentkezett be');
  }

  return res.redirect('/');
} else {
  return res.render('login', {
    loginMessage: 'Hibás jelszó!'
  });
}

```

A felső if else elágazásnál azt ellenőrizzük, hogy a jelszó helyes-e. Ha helyes a jelszó akkor elmentjük a sessionbe a felhasználó nevet és az email címet. A return kód sorral pedig átirányítjuk a főoldalra. Viszont, ha hamis a jelszó akkor kapunk egy hiba üzenetet, hogy „Hibás jelszó!”.

```

} catch (err) {
  console.error(err);
  res.status(500).send('Hiba történt a bejelentkezés során.');
```

A következő sor hibát jelez a bejelentkezés során 500-as hibakóddal.



```

exports.getIndexPage = (req, res) => {
  res.render('index', {
    headerText: 'Üdvözlünk!',
    imageSrc: '/img/logo.png',
    loginHref: req.session.user ? '/logout' : '/login',
    loginText: req.session.user ? 'Kijelentkezés' : 'Bejelentkezés',
    loggedInUser: req.session.user || null
  });
};

```

A felső képen lévő kód részlet a bejelentkezés után új fület hoz létre, ami webshop névre hajaz. A webshop fülön keresztül lehet a termékeket megvásárolni a bejelentkezett felhasználóknak. A bejelentkezés után megjelenik a profil és a kijelentkezés opció is a weboldalon.

### *Bejelentkezés(router)*

```

const express = require('express');
const router = express.Router();
const loginController = require('../controllers/loginController');

router.get('/login', loginController.getLoginPage);

router.get('/logout', (req, res) => {
  req.session.destroy();
  res.redirect('/login');
});

router.post('/auth', loginController.loginUser);

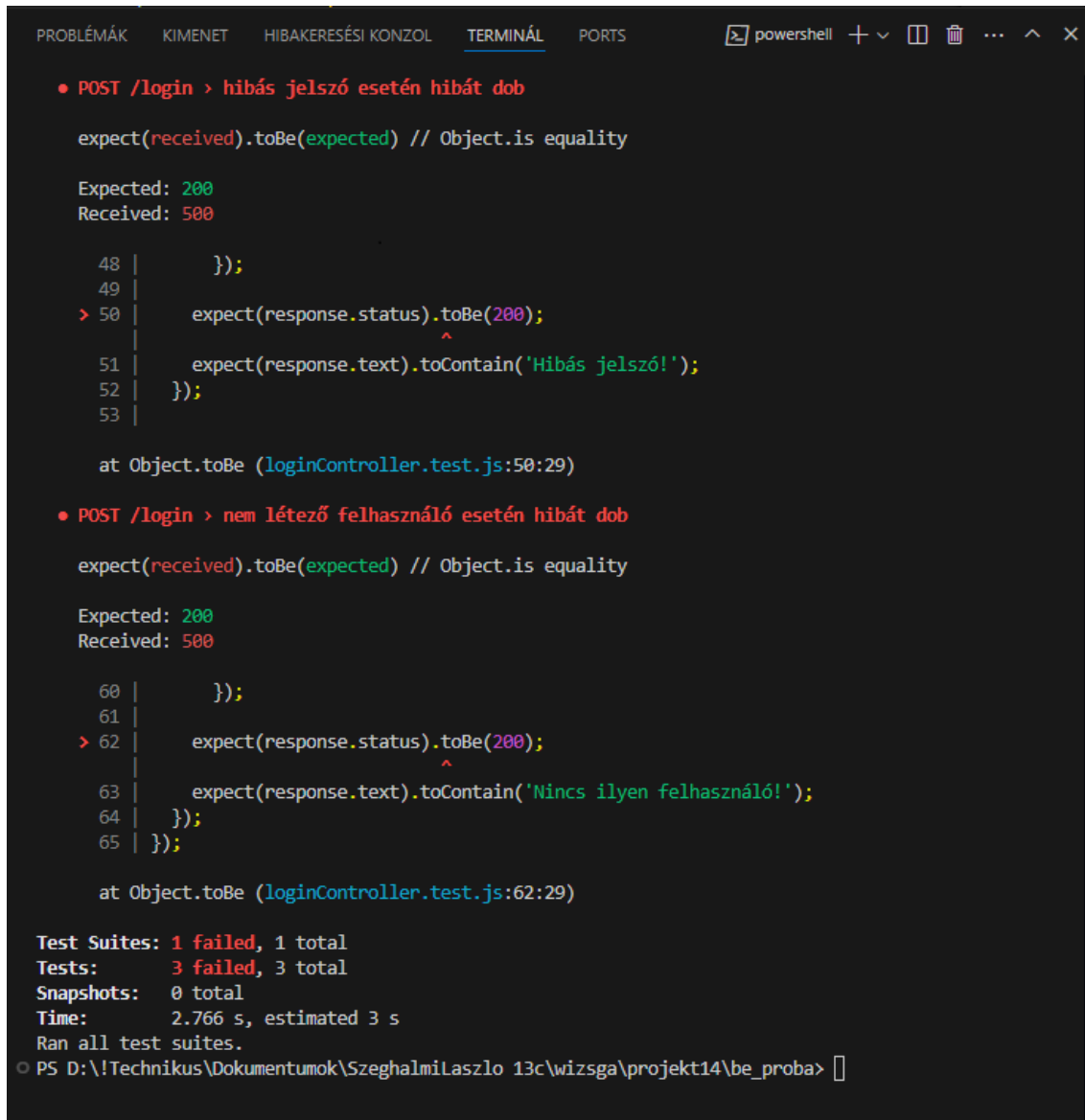
module.exports = router;

```

A bejelentkezés router felelős a bejelentkezések és a regisztrációs kérések kezeléséért. A `router.get('/login')` sor a bejelentkezés útvonalát szolgálja ki. A `router.get('/logout')` törli a sessiont és utána visszairányít a bejelentkezési weboldalra. Bejelentkezés feldolgozását a `router.post` végzi el, ugye POST kéréssel.

# Tesztelés

## Hibás kód



```
PROBLÉMÁK  KIMENET  HIBAKERESÉSI KONZOL  TERMINÁL  PORTS  powershell + - [] [X] ... ^ X

• POST /login > hibás jelszó esetén hibát dob

expect(received).toBe(expected) // Object.is equality

Expected: 200
Received: 500

48 |     });
49 |
> 50 |     expect(response.status).toBe(200);
    |                               ^
51 |     expect(response.text).toContain('Hibás jelszó!');
52 |   });
53 |

at Object.toBe (loginController.test.js:50:29)

• POST /login > nem létező felhasználó esetén hibát dob

expect(received).toBe(expected) // Object.is equality

Expected: 200
Received: 500

60 |     });
61 |
> 62 |     expect(response.status).toBe(200);
    |                               ^
63 |     expect(response.text).toContain('Nincs ilyen felhasználó!');
64 |   });
65 |

at Object.toBe (loginController.test.js:62:29)

Test Suites: 1 failed, 1 total
Tests:       3 failed, 3 total
Snapshots:   0 total
Time:        2.766 s, estimated 3 s
Ran all test suites.
PS D:\!Technikus\Dokumentumok\SzeghalmiLaszlo 13c\wizsga\projekt14\be_proba> []
```

A tesztelésünk a bejelentkezést vette figyelembe, aminél volt olyan, hogy helyes volt a felhasználó volt olyan, amikor a jelszó volt a hibás és volt olyan is mikor nem létezett a felhasználó. Elsőként egyik se ment át a teszten.

## Helyes kód



```
66 }  
67 }  
  
PROBLÉMÁK KIMENET HIBAKERESÉSI KONZOL TERMINÁL PORTS  
powershell + - ... ^ x  
  
at Query.run (node_modules/sequelize/src/dialects/mysql/query.js:46:25)  
at node_modules/sequelize/src/sequelize.js:650:28  
at MySQLQueryInterface.showIndex (node_modules/sequelize/src/dialects/abstract/query-interface.js:600:12)  
at Function.sync (node_modules/sequelize/src/model.js:1413:29)  
at Sequelize.sync (node_modules/sequelize/src/sequelize.js:825:9)  
  
Test Suites: 1 failed, 1 total  
Tests: 3 passed, 3 total  
Snapshots: 0 total  
Time: 2.74 s, estimated 3 s  
Ran all test suites.  
PS D:\Technikus\Dokumentumok\SzeghalmiLaszlo 13c\vizsga\projekt14\be_proba>
```

A következő tesztelésünknel kijavítottuk a hibákat, viszont egy probléma fent maradt, ami azért jelenhetett meg mivel egy táblát nem használtunk, ami az Orders tábla névre hallgat. A projektünknel ezért az továbbfejlesztési lehetőség marad az Orders tábla használata.

## **Kinek mi volt a feladata a projektben**

### ***Szeghalmi László***

Az ő feladata a weboldalon a frontend és a backend volt ő csinálta meg például a webshopnak az alapját a mappaszerkezetet az adatbázist, hogy összelegyen kapcsolva az egész weboldallal.

### ***Murányi Marcell***

A feladata neki is a frontend meg a backend volt, amin ketten dolgoztak a legtöbbet. A bejelentkezés nagy részét neki köszönhetjük ahogyan a profilt is meg a kosarat ketten hozták össze nagyrészt.

### ***Kovács Sándor***

Feladata a dokumentáció meg a frontend volt a dokumentáció nagy részét ő írta némi segítséget kapva a többiektől. A frontend külső tulajdonságait ő véglegesítette úgy, hogy az alap már megvolt.

## **Továbbfejlesztési lehetőségek a weboldalon**

Elsősorban a rendelés gomb megnyomásával olyan mezőket lehetne létrehozni, amik bekérlik a felhasználó szállítási adatait, hogy alkalmasabb legyen egy valósághűbb rendelés végrehajtásához.

Ezzel egyhuzamban létre lehetne hozni egy új táblát és meglehetne oldani, hogy regisztrálás nélkül is lehessen rendelni, amivel átkéne alakítani a főoldalt úgy, hogy mindenki számára látható legyen a webshop fül.

A következő ötlet, ami előjött hogy egy másik menüpontra lehetne olyat, hogy egészséges táplálkozás, ami egy új oldalt nyitna meg ahol szakemberek tanácsait lehetne olvasni. Fórumnak is megfelelő lenne kommentekkel értékelésekkel.

Lehetne még egy fül a menüpontra amin oktató videók vannak edzésekről meg ételek elkészítéséről. Ezek a videók segítséget nyújtanának a látogatóknak, hogy egészségesebb életmódra váltsanak. Kezdő löketet biztosítana az újoncoknak, akik mozgással telibb életet szeretnének űzni.

# Felhasználói dokumentáció

## Rendszerkövetelmények

### *Google Chrome*

Operációs rendszerek				
Windows	macOS	Linux	Android	iOS
<b>7-11(32 és 64bit)</b>	10.11 vagy újabb	legtöbb verziója alkalmas	4.4 vagy újabb	12.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 350MB szabad tárhely szükséges

## ***Mozilla Firefox***

Operációs rendszerek				
Windows	macOS	Linux	Android	iOS
<b>7-11(32 és 64bit)</b>	10.12 vagy újabb	legtöbb verziója alkalmas	5.0 vagy újabb	13.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 200MB szabad tárhely szükséges

## ***Microsoft Edge***

Operációs rendszerek				
Windows	macOS	Linux	Android	iOS
<b>7-11(64bit)</b>	10.12 vagy újabb	legtöbb verziója alkalmas	4.4 vagy újabb	12.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 250MB szabad tárhely szükséges

## *Safari*

Operációs rendszerek		
Windows	macOS	iOS
<b>7-11(64bit)</b>	10.12 vagy újabb	11.0 vagy újabb

Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 200MB szabad tárhely szükséges

## *Opera*

Operációs rendszerek			
Windows	macOS	Linux	Android
<b>7-11(32 és 64bit)</b>	10.12 vagy újabb	legtöbb verziója alkalmas	5.0 vagy újabb

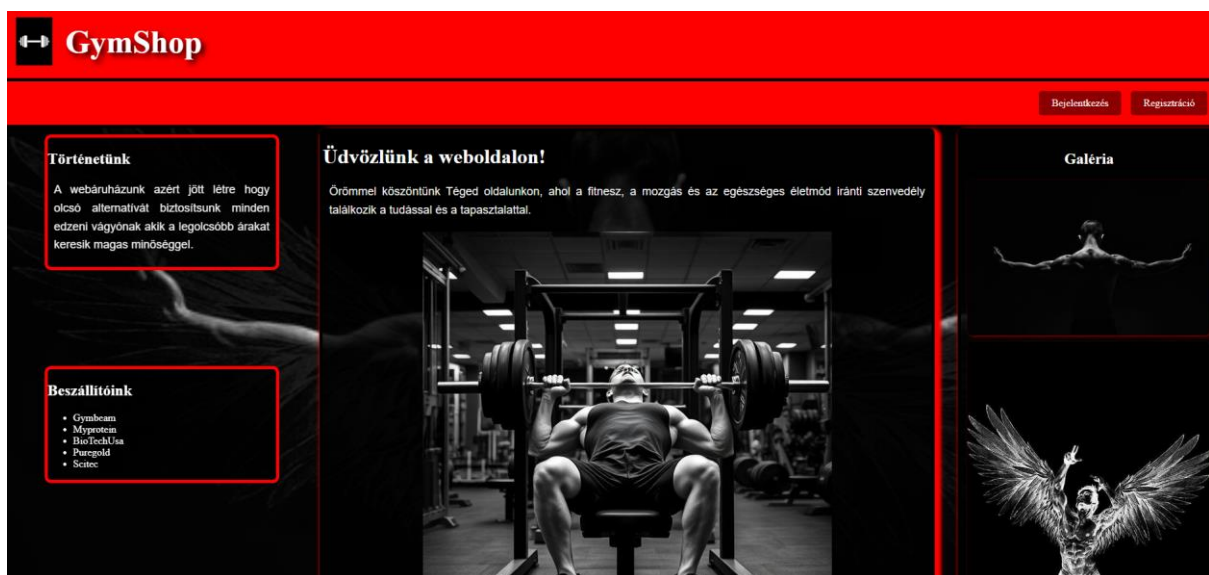
Hardver		
Processzor	RAM	Tárhely
<b>minimum 1GHz</b>	Legalább 2GB	minimum 200MB szabad tárhely szükséges



## Program telepítése

A letöltött .exe fájlt indítsa el és már tudja is használni a weboldalunkat minden funkciójával.

## A webshop bemutatása



5. ábra Webshop főoldal

Az webshop látogatását követően ez az oldal fogadja a leendő vásárlót.



6. ábra Regisztrációs gomb

A főoldal jobb oldalán található a bejelentkezés és a regisztrációs gomb. A regisztrációs gomb megnyomásával tud a felhasználó létre hozni egy új fiokót amit követően majd be tud jelentkezni az éppen létrehozott fiókjába és be tud majd jelentkezni a bejelentkezés gomb segítségével. A gomb viszont elsőnek a regisztrációs oldalra fog bedobni.



**Regisztráció**

Név:

Email:

Telefonszám:

Jelszó:

REGISZTRÁCIÓ

7. ábra Regisztrációs oldal

### *Név*

A regisztrációs oldal tartalmaz egy név mezőt, ahova a felhasználónak egy olyan nevet kell beírnia, amit szívesen látna a weboldal jobb felső sarkában a bejelentkezést követően.

### *Email*

Az email részhez a jövőbeli felhasználónak egy létező email címet kell begépelnie. Olyan emailt kell begépelnie a felhasználónak, amit szívesen használ arra is hogy értesítéseket kap az újdonságokról vagy az éppen megrendelt termékének vagy szolgáltatásának információit szeretné megnézni.

## ***Telefonszám***

A telefonszám mezőhöz a regisztrálni kívánó személynek egy létező mobilszámot kell hozzáadnia amire szeretné kapni az esetleges rendeléssel járó sms-eket vagy hogy a weboldal üzemeltetői, vészhelyzet esetén feltudják hívni a felhasználót.

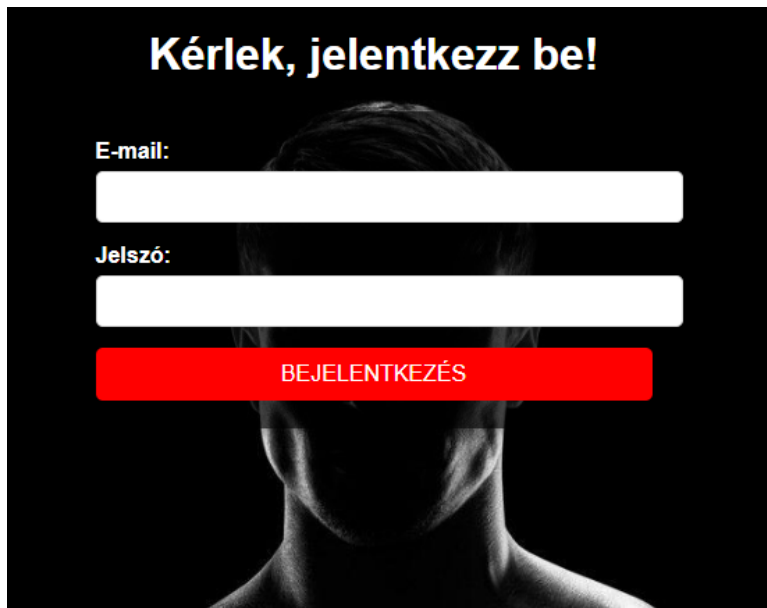
## ***Jelszó***

A felhasználónak egy szabadon választott jelszót kell beírnia a mezőbe. A jelszónak olyan szövegnek kell lennie, amit a leendő felhasználó nem nagyon tud elfelejteni.



8. ábra Bejelentkezés gomb

A regisztráció gombtól bal oldalra található a bejelentkezés gomb, amit, ha megnyom a jövőbeli felhasználó akkor átdobja egy másik oldalra, ahova a regisztráció során beírt adatokat kell begépelnie. Ezek után tud majd rendeléseket leadni és szolgáltatásokat vásárolni.



**Kérlek, jelentkezz be!**

E-mail:

Jelszó:

**BEJELENTKEZÉS**

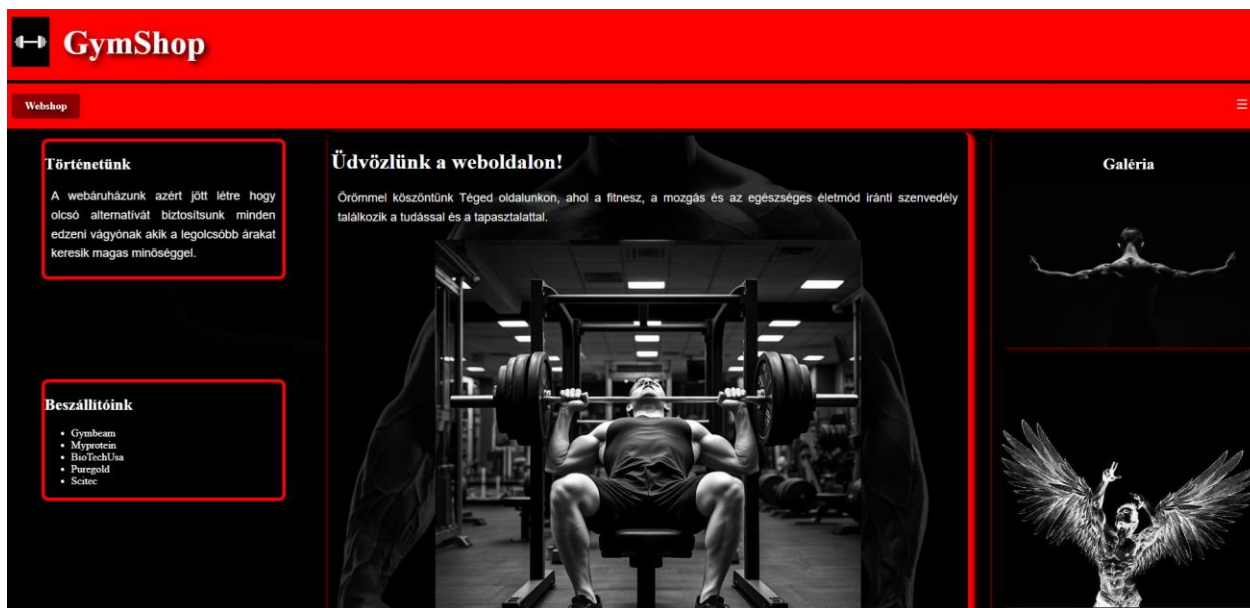
9. ábra Bejelentkezés oldal

### ***Email***

Az email mező arra szolgál, hogy a felhasználó által már korábban regisztrált email címmel tudja az oldal azonosítani, hogy ki szeretne bejelentkezni.

### ***Jelszó***

A jelszó mező azt a célt szolgálja, hogy az előbb beírt email címhez megerősítse a felhasználó, hogy ténylegesen az ő tulajdonát képezi.



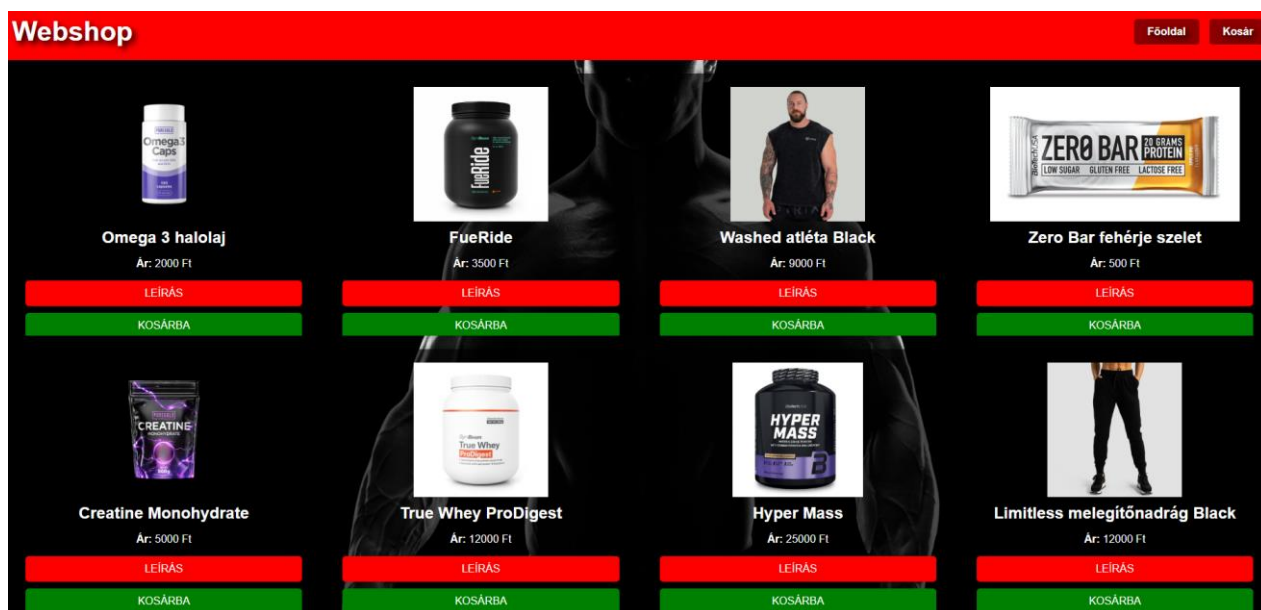
10. ábra Bejelentkezett főoldal

A felhasználót ez az oldal fogja várni bejelentkezés után a bekeretezett részen található a webshop gomb.



11. ábra Webshop gomb

A webshop gomb csak a felhasználó bejelentkezése után jelenik meg amin keresztül lehet megtekinteni az oldalon fellelhető termékeket és megvásárolhatja őket.



12. ábra Webshop oldal

A felhasználót webshop gomb megnyomása után ez a felület fogadja, ahol tudja megvásárolni a termékeket. Az oldal megjeleníti a termékeket azok árait és a leírás menüben plusz információkat lehet olvasni az adott árurol.



13. ábra Termék ikon

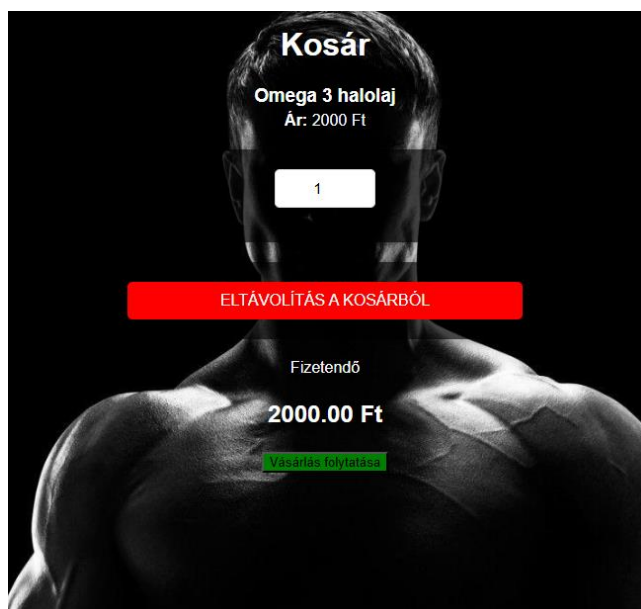
## *Leírás oldal*



14. ábra Termék leírása

A leírás oldal láthatóan tartalmazza a termék nevét, márkáját, kategóriáját a készleten lévő árú mennyiségét az árát és még egy leírást is lehet róla olvasni.

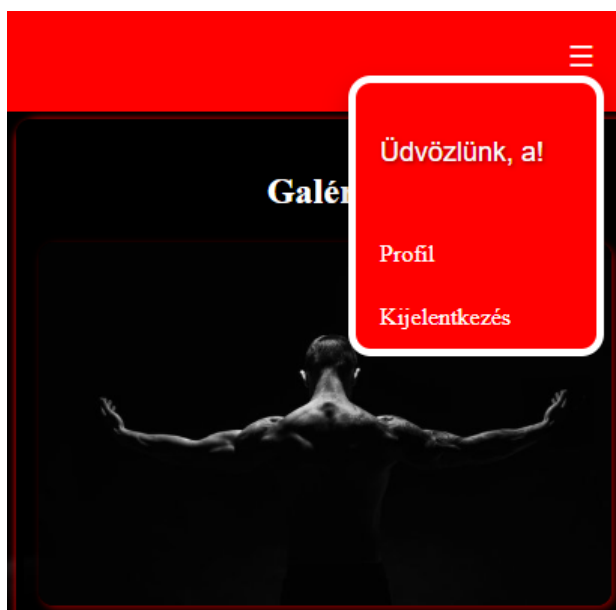
## *Kosár oldal*



15. ábra Kosár

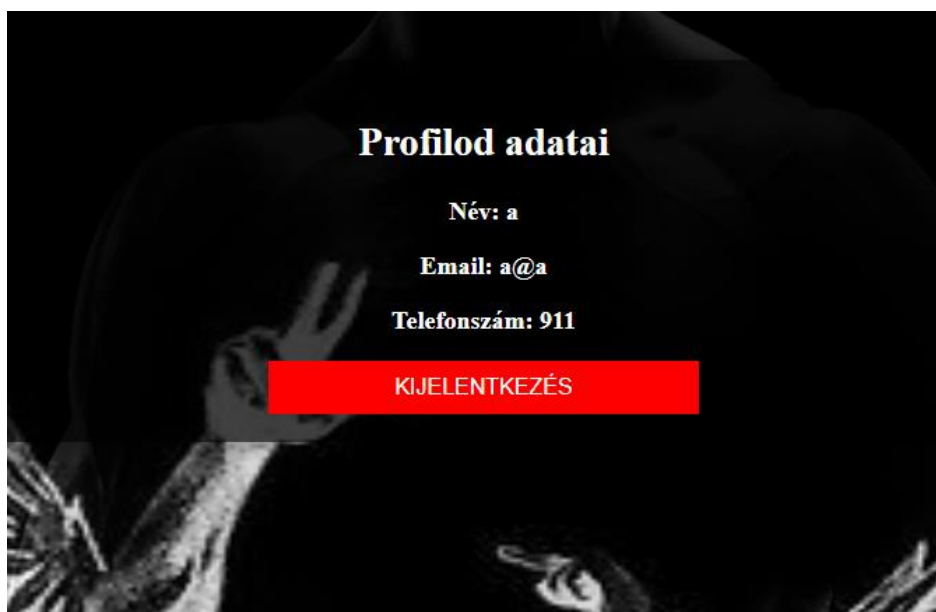
A kosár oldalon tudja a vásárló a megrendelni kívánt termékét megvásárolni. Kitudja választani a mennyiséget és az oldal a mennyiséggel arányosan kiírja az árát.





16. ábra Hamburger menü

Miután a felhasználó visszalépett a webshop oldalról a főoldal felületére utána a jobb oldalon megtalálhat egy hamburger ikon mögé rejtett profil és kijelentkezés füleket.



17. ábra Profil

A profil fülön tudja a felhasználó megnézni a regisztráció során beírt adatait, ami a nevet, emailt és a telefonszámot takarja. Ha a kijelentkezés gomra rá kattint a felhasználó mind a profil oldalon belül és a hamburger ikonon belül kijelentkezik a jelenlegi fiókjából és akár bejelentkezhethet egy másikba is.

# Köszönetnyilvánítás

Szeretnénk kifejezni őszinte köszönetünket mindazoknak, akik hozzájárultak a projektünk sikeres megvalósításához.

Először is, hálásak vagyunk tanárainknak, akik folyamatos támogatásával, iránymutatásaikkal és értékes visszajelzéseivel segítette munkánkat a projekt során.

Munkánk során mindannyian hozzájárultunk a tervezéshez, fejlesztéshez és teszteléshez, az együttműködésünk eredményeként egy sikeres projektet valósítottunk meg.

Végül szeretnénk megköszönni családjainknak és barátainknak a támogatást és a türelmet, amelyet a projekt során nyújtottak.

Hálásak vagyunk, hogy mindezek a segítségek lehetővé tették számunkra, hogy értékes tapasztalatokat szerezzünk a webszerkesztés területén.