

Szerzők

Nevek: Móricz János Milán, Lévai Levente József

Email-ek:

levlev324@gmail.com

moriczj10@gmail.com

Projekt vezető: Móricz János Milán

Tartalom

Szereplők és igényeik-----	1.oldal
Use-case funkciólista-----	1-5.oldal
Magasszintű rendszerterv-----	6-7.oldal
Screenek-----	7.oldal
Adatbázis modell-----	7-8.oldal
Alkalmazások kiválasztása-----	8.oldal
Routing-----	9-10.oldal
Implementáció-----	10.oldal
Tesztelés-----	10.oldal
Feladat-----	11-12.oldal
Környezet-----	13.oldal
Fejlesztés menete-----	13-16.oldal
Fejlesztési lehetőségek-----	17.oldal

Fejlesztői dokumentáció

Szereplők és igényeik:

Látogató: Regisztráció nélkül megtudja tekinteni az oldalon található termékeket és azok részletes leírását, és keresni a termékek között. Tud regisztrálni és azt követően bármikor bejelentkezni a megadott adataival.

Látja az oldalon elhelyezett bejegyzéseket (Híreket).

Felhasználó: Rendelkezik minden joggal, amivel a Látogató is.

Fiókjából ki tud jelentkezni.

Adott terméket kosárba tudja rakni és étékelést tud írni a termékre. A kosárból tovább tud lépni a vásárlásra, ahol az adatai megadása után megrendelheti a termékeket és sikeres vásárlás esetén visszaigazoló üzenetet ad.

Profilja oldalán látja a leadott rendeléseit.

Admin: Az admin tudja szerkeszteni a termékeket, bejegyzéseket és felhasználókat.

Tudja törölni a termékeket, bejegyzéseket, felhasználókat, rendeléseket és értékeléseket.

Tud új termékeket és bejegyzéseket létrehozni.

A bejegyzéseknél látja, hogy melyik admin írta őket. Fiókjából ki tud jelentkezni.

Use-case funkciólista:

Regisztráció:

POST: /register

Bemenő paraméterek: username, email, password.

Kimenő paraméterek: 201-es status response, a felhasználónév vagy email foglalt.

Bejelentkezés:

POST: /login

Bemenő paraméterek: username, password

Kimenő paraméterek: Az adott felhasználó tokennel JSON formátumban, „rossz felhasználónév” vagy „rossz jelszó”.

Termékek:

GET: /products

Bemenő paraméterek: null.

Kimenő paraméterek: Termékek listája JSON formátumban, „Not found any product”.

Egy termék:

GET: /products/product/:id Bemenő paraméterek: null.

Kimenő paraméterek: Egy termék JSON formátumban, „Not found any product”.

Termék törlése:

DELETE: /deleteProduct/:id A termék keresése.

Kimenő paraméterek: „Deleted a product”, „Not deleted any product”.

Termék létrehozása:

POST: /products

Bemenő paraméterek: cost, category, name, description, image.

Kimenő paraméterek: „Added a product”, „Product not added”.

Termék szerkesztése:

PUT: /products

Bemenő paraméterek: cost, category, name, description, image. Kimenő paraméterek: „Product updated”, „Product not updated”.

Adott termék értékelése:

GET: /productReviews/:id Bemenő paraméterek: null.

Kimenő paraméterek: Adott termék értékelése JSON formátumban, „Not found any review for this product”.

Termék értékelése:

POST: /review

Bemenő paraméterek: user_id, product_id, description, rating, username.

Kimenő paraméterek: "Review added", "Review not added".

Értékelés törlése:

DELETE: /deleteReview/:id

Az értékelés keresése.

Kimenő paraméterek: "Deleted a review", "Not deleted any review".

Belyegyzések:

GET: /entries

Bemenő paraméterek: null.

Kimenő paraméterek: Belyegyzések listája JSON formátumban, "Not found any entry".

Egy belyegyzés:

GET: /entries/:id

Bemenő paraméterek: null.

Kimenő paraméterek: Az adott bejegyzés JSON formátumban, "Not found any entry".

Belyegyzés törlése:

DELETE: /deleteEntries/:id

Belyegyzés keresése.

Kimenő paraméterek: "Deleted an entry", "Not deleted any entry".

Belyegyzés létrehozása:

POST: /entries

Bemenő paraméterek: user_id, username, title, description.

Kimenő paraméterek: "Entry added", "Entry not added".

Belyegyzés szerkesztése:

PUT: /entries

Bemenő paraméterek: title, description.

Kimenő paraméterek: "Entry updated", "Entry not updated".

Vásárlás leadása:

POST: /checkout

Bemenő paraméterek: contactInfo, items, user_id, username, email, totalCost, itemName.

Kimenő paraméterek: "Checkout confirmed", "Checkout failed".

Rendelések:

GET: /orders

Bemenő paraméterek: null.

Kimenő paraméterek: Rendelések JSON formátumban, "Not found any order".

Rendelés törlése:

DELETE: /deleteOrder/:id

Bemenő paraméterek: null.

Kimenő paraméterek: "Deleted an order", "Not deleted any order".

Felhasználók:

GET: /users

Bemenő paraméterek: null.

Kimenő paraméterek: Felhasználók listája JSON formátumban, "Not found any user".

Egy felhasználó:

GET: /users/:id

Bemenő paraméterek: null.

Kimenő paraméterek: Egy felhasználó JSON formátumban, "Not found the user".

Felhasználó szerkesztése:

PUT: /users

Bemenő paraméterek: username, password, email, role.

Kimenő paraméterek: "User updated", "User not updated".

Felhasználó törlése:

DELETE: /deleteUser/:id

Felhasználó keresése.

Kimenő paraméterek: "Deleted a user", "Not deleted any user".

Felhasználó rendelései:

GET: /usersOrder/:id

Bemenő paraméterek: null.

Kimenő paraméterek: Az adott felhasználó rendelései JSON formátumban, "Not found any order for this user".

Magasszintű rendszerterv:

Backend: Használt technológia Node.js

Azért választottuk a node.js-t mert sok lehetőséget tartalmaz, könnyen átlátható és a modulos felépítése miatt tökéletes a projektünkhöz.



Frontend: Használt technológia React.js

Azért választottuk a react.js-t mert könnyen össze lehet kötni a node.js-es backenddel, könnyen használható modulokat tartalmaz, a stílushoz használható bootstrap modult tartalmaz és könnyű a routingot kezelni.



Adatbázis: Használt technológia MySQL adatbázis

Azért választottuk a MySQL-t mert egy még kezdők számára is könnyű adatbázist készíteni bene, letisztult és könnyen használható.



Böngésző követelmény: A legfrissebb verziójú Google Chrome, MozillaFirefox és Safari böngészők.



Screenek:

Screenek.pdf-ben mellékelve

Adatbázis modell:

```
entries{
    id(PK, AUTO_INCREMENT, int)
    user_id(FK, int)
    username(varchar)
    title(varchar)
    description(varchar)
    image(mediumtext)
}
```

```
orders{
    id(PK, AUTO_INCREMENT, int) user_id(FK, int)
    username(varchar)
    email(varchar)
    contactInfo(longtext)
    items(longtext)
    itemName(varchar)
    totalCost(int)
}
```

```
product{
    id(PK, AUTO_INCREMENT, int) cost(int) category(varchar)
    name(varchar)
    description(varchar)
    image(mediumtext)
}
```

```
reviews{
    id(PK, AUTO_INCREMENT, int) user_id(FK, int)
    product_id(FK, id)
    username(varchar)
    rating(int)
    description(varchar)
}
```

```
user{
    id(PK, AUTO_INCREMENT, int)
    username(varchar)
    email(varchar)
    password(varchar)
    role(varchar)
}
```

Alkalmazások kiválasztása:

Visual Studio Code

XAMPP

Routing:

`app.post('/products',` Formok kitöltése és validálás után gombnyomásra létrehoz egy terméket)

`app.get('/products',` Kilistázza az összes terméket)

`app.put('/products/:id',` Formokban szereplő értékek átírása és validálás után gombnyomásra szerkeszti az adott terméket)

`app.get('/products/product/:id',` Kilistázza az adott terméket)

`app.delete('/deleteProduct/:id',` Gombnyomásra törli az adott terméket)

`app.post('/review',` Formok kitöltése és validálás után gombnyomásra hozzá ad egy értékelést az adott termékhez)

`app.get('/productReviews/:id',` Lekéri a termék értékeléseit) -----

`app.delete('/deleteReview/:id',` Gombnyomásra törli az adott értékelést)

`app.post('/entries',` Formok kitöltése és validálás után gombnyomásra létrehoz egy bejegyzést)

`app.put('/entries/:id',` Formokban szereplő értékek átírása és validálás után gombnyomásra szerkeszti az adott bejegyzést)

`app.get('/entries',` Kilistázza az összes bejegyzést)

`app.get('/entries/:id',` Kilistázza az adott bejegyzést)

`app.delete('/deleteEntries/:id',` Gombnyomásra törli az adott bejegyzést)

`app.post('/register',` Formok kitöltése és validálás után gombnyomásra létrehoz egy felhasználót)

`app.post('/login',` Formok kitöltése és ellenőrzése után gombnyomásra beléptet egy felhasználót)

`app.post('/checkout', Formok kitöltése és validálás után gombnyomásra leadja az adott felhasználó rendelését)`

`app.get('/orders', Kilistázza az összes rendelést)`

`app.delete('/deleteOrder/:id', Gombnyomásra törli az adott rendelést)`

`app.get('/users', Kilistázza az összes felhasználót)`

`app.get('/users/:id', Kilistázza az adott felhasználót)`

`app.put('/users/:id', Formokban szereplő értékek átírása és validálás után gombnyomásra szerkeszti az adott felhasználót)`

`app.delete('/deleteUser/:id', Gombnyomásra törli az adott felhasználót)`

`app.get('/usersOrder/:id', Kilistázza az adott felhasználó rendeléseit)`

Implementáció:

Forráskód: <https://github.com/MoriczJanos113C/TechShop>

Tesztelés:

Node: Postman tesztek mellékelve.

React: Axios és form tesztelés.

Feladat:

Adminisztrátori felület:

Látjuk az adatbázisban szereplő felhasználókat, akiket lehet törölni vagy módosítani, nevük alapján keresni.

Termékeket lehet törölni, módosítani vagy újat felvenni, nevük alapján keresni.

Az adott termékhez kapcsolódó véleményt lehet törölni.

Rendeléseket lehet törölni, az adott felhasználóneve alapján keresni köztük.

Bejegyzéseket lehet hozzáadni, törölni, módosítani.

Bejelentkezési oldal:

Minden felhasználó először ide kerül, ahol be kell jelentkeznie. A felhasználó dönthet, hogy bejelentkezik vagy folytatja bejelentkezés nélkül.

Regisztrációs oldal:

A felhasználó regisztrálhat, vagy folytathatja regisztráció nélkül.

Kezdőlap:

A felhasználó ide kerül mind bejelentkezéssel és bejelentkezés nélkül is.

Az oldal alapjáraton megjeleníti az összes terméket, ami elérhető a webshopban.

Termékek neve alapján lehet keresni az oldalon.

Termék oldal:

Alap információk kiírása a termékről, mint ár, kép és részletes leírás.

A felhasználó itt dönthet úgy, hogy kosárba rakja a terméket. A felhasználó tudja értékelni és szöveges véleményt adni az adott terméket.

Kosár:

A vásárolni kívánt termékek először ide kerülnek. El lehet távolítani a nem kívánt termékeket.

Vásárlási felület:

Személyes adatok kitöltése.

Sikeres vásárlás esetén át navigálja az adott felhasználó profil oldalára.

Profil oldal:

A felhasználó rendelésinek ki listázása.

Bejegyzések:

A felhasználó itt tekintheti meg a különböző informatikai cikkeket.

Elérhetőségek oldal:

A felhasználó ezen az oldalon található telefonszámok és email címek alapján tud kapcsolatba lépni az ügyfélszolgálattal.

Környezet:

Visual Studio Code, XAMPP

Fejlesztés menete:

2022.02.20.: A fejlesztés kezdete. React alkalmazás elkészítve, register/login oldal elkészítve.

A nap egy későbbi pontján a register és login oldalak külön lettek szedve, két külön felületre. TechShop adatbázis elkészítve.

2022.02.22.: Footer, navbar placeholderre, product és home page minták elkészítve.

Ismeretlen hiba miatt egy extra package.lock.json fájl került a projektbe. Ez törlésre került.

Adatbázissal való összekötés megvalósítva.

2022.02.23.: Termék oldalra a termékek még nem jelenítődnek meg id szerint.

Javítás: A termékek megjelennek id szerint.

Kosár felület elkészítve, de még nem kapja meg a termékeket.

2022.02.24.: Product page elkészítve. Egy termék leírást tartalmazza ez az oldal mit a főoldalról lehet elérni az adott termékre kattintva. Utólag fölöslegesnek gondolt fájlok törlésre kerültek.

2022.02.28.: Kosár tovább fejlesztve. Továbbra se úgy működik ahogy kéne.

2022.03.01.: A product page sikeresen megjeleníti az odatartozó terméket.

Bejelentkezés és regisztrálás működőképes.

2022.03.03.: Profile page elkészítve.

A felhasználó már a user_id-jával lép be az oldalra.

Vásárlás oldal elkészítve, ahol a felhasználónak meg kell adnia az adatait.

Route-ok átalakítva, mindegyik működik. Profile page törlésre került.

2022.03.06.: Közös megbeszélés alapján egy teljes átalakítást kezdeményeztünk az egész projectben. Az előzőleg megírt kódokat alapul véve minden eddig létező oldalt javítottunk, kosár és profil oldalak újra lettek kezdve. Footert eldobtuk.

2022.03.07.: Route-ok külön lettek szedve egy külön Router.js-be. Az EditProduct.js, LoginPage.js és LogoutPage.js elkészítve. Termék oldalon a routing frissítve lett. Index.js-ben adatbázis kérések megvalósítva, JWT tokenes validáció elkezdése.

2022.03.07.: JWT tokenes validáció folytatása, még nem működik. Bcrypt kipróbálása.

2022.03.09.: Userek lekezelve. Orderhez backend még nem lett lekezelve adatbázis tábla hiány miatt. Javítások, hogy ne akadjon el a backend. JWT token működik userrel. ProductPage.js elkezdve, hozzá tartozó route-val együtt.

2022.03.10.: User és admin a roleok működnek. ProductPage.js befejezve. Update, delete kérések frontend és backend részen is működnek.

2022.03.14.: CSS hozzárendelve az oldalakhoz.

2022.03.15.: Vásárlás lebonyolítása kész, productpage javítása, értékelés és véleményírás előkészítve.

2022.03.16.: ProductPage.js-hez értékelés és véleményírás elkészítve, frontendes hiba miatt nem működik. További Adatbázis kérések megírása az index.js-ben.

2022.03.17.: AppRouter.js, Create.ProductPage.js, EditProduct.js, EditUser.js és index.js tovább fejlesztve.

2022.03.20.: User szerkesztése és törlése működik már, összes user oldal kész, véleményeket lehet már törölni és lehet látni is, hogy kiírta a véleményt, gombbal lehet törölni amit csak admin lát, rendelésekhez adott user aki rendelte hozzáadva azaz felmegy adatbázisba, összes rendelés oldala kész az admin részen, profil oldal kész, rendeléseket még nem lehet látni az adott userhez, -confirmation page törölve, most már az adott usert egyből a profiljára irányítja és majd ott kap visszajelzést, hogy kész van a rendelése.

Az oldalak, ahol automatikus frissítésre van szükség hozzá lettek adva, kisebb hibák javításra kerültek.

A rendeléseket megjeleníti, de nem kapja meg az adatokat az adatbázisból.

2022.03.21.: Rendeléseket rendesen ki írja, de még minden felhasználó látja az összes felhasználó rendelését. Kijelentkezésnél a kosár automatikusan kiürül.

A felhasználó regisztrációnál emailt kap, rendelésnél át adja az email címet egy külön oszlopban az adatbázisban.

Stílus tovább fejlesztve.

2022.03.22.: Utolsó feltöltés által okozott hibák visszafordítása.

2022.03.23.: Rendelésnél a termékek nevei és teljes összege bekerül az adatbázisba.

Form validation megcsinálva.

2022.03.24.: Edit user/product formokban már megjelennek az értékek. Regisztrációnál a form validation működik.

2022.03.26.: CSS tovább fejlesztve.

2022.03.27.: Formok validációja kész, bejegyzés oldal és szerkesztése kész.

Termék oldalon szerkesztésnél nem tartja meg a képet.

Bejegyzések szerkesztésének javítása.

EditProduct.js és index.js javítva.

2022.03.28.: Users/entries edit oldalakon formdata megszüntetve mert ezekhez nincs kép.

Fromokban már ékezetet is elfogad.

Ha a szerkesztésnél nem adunk új nevet a terméknek akkor undefined lesz.

Register backend lekezelve. Login form validálva.

Navigálás nem jó a login registernél, de a validálás működik.

2022.03.29.: Register/login formok validálva.

Termék szerkesztésnél már nem lesz undefined és a kép is marad.

Specifikáció alapján refaktorálások megvalósítva.

2022.04.15.: Számos kisebb hiba kijavítva.

2022.04.22.: Backend hibák javítva, Postman Collection kész a nodejshez

2022.04.23.: Validálással ellátott részekben megengedett karakterek kibővítve.

Postmanben POST és PUT javítva.

2022.04.24.: Adatbázis új verziójának adatokkal való feltöltése.

Képek felrakása.

Console.log-ok eltávolítása backend és frontend oldalakon,

Approuterben a route-ok beállítása és css javítások.

2022.04.25.: Végleges adatbázis és postman collection. Regisztráció hiba javítva, ha olyan felhasználó akar regisztrálni, aki már létezik.

2022.04.26.: Profile page bug javítása, kommentek hozzáadása.

2022.04.27.: React axios tesztek írása.

Fejlesztési lehetőségek:

- Session hozzáadása.
- Kosár tartalmaznának adatbázisba való mentése.
- Adott termékeket checkbox-al szűrni kategóriák szerint.
- Profil oldalon a felhasználó új jelszót tud kérni.
- Bejelentkezés oldalon „Elfelejtettem a jelszavam” opció hozzáadása.
- Live chat bot.
- id-k kódolása.