🍕 PollákPizza 🍕

A csapattagok: **Huszár Imre, Héja Gábor, Sztojka Milán László**

🛠️ Használt technológiák 🛠️

**Nyelvek: Vue.js, Node.js, CSS/SCSS**

✨ Design ✨

A Design tervezője**: Sztojka Milán**

Használt program: **Figma**

**Telepítési útmutató**

Frontend:

1. **Navigálj a mappába:**

* cd PollakPizza

1. **Inicializáld a projektet:**

* npm init -y

1. **Telepítsd a szükséges csomagokat:**

* npm install

1. **Indítsd el a weboldalt:**

* npm run dev

Backend:

1. **Navigálj a backend mappába:**

* cd PollakPizza/backend

1. **Telepítsd a szükséges csomagokat:**

* npm install

1. **Adatbázis feltöltése:**

* A MySQL-be hozzon létre egy adatbázist **pollakpizza** néven, majd importálja *docs*-ban található *PZdatabase.sql* fájlt

1. **Indítsd el a backendet:**

* node .

Projekt struktúra

Database

Az adatbázis felépítése:

1. pizzas tábla: A pizzák adatait tárolja.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| id | INT | Egyedi pizza azonosító |
| name | VARCHAR(255) | A pizza neve (pl. Margherita) |
| price | INT | A pizza alapára. |
| image | VARCHAR(255) | URL útvonal a pizza képéhez. |

2. size tábla: A pizzaméreteket és az árnövekményeket tartalmazza.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| id | INT | Egyedi méret azonosító |
| size | INT | A pizza mérete (pl. 32). |
| multiPrice | Decimal(10,0) | Méret alapján árnövekmény. |

3. toppings tábla: Az elérhető feltéteket és azok többletköltségeit tárolja.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| id | INT | Egyedi feltét azonosító |
| name | VARCHAR(255) | A feltét neve (pl. Sajt). |
| bonusPrice | INT | A feltétért járó többletköltség |

4. users tábla: A felhasználók adatait tárolja.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| id | INT | Egyedi felhasználói azonosító |
| name | VARCHAR(255) | A felhasználó teljes neve |
| email | VARCHAR(255) | A felhasználó email címe |
| password | VARCHAR(255) | A felhasználó jelszava (titkosított) |
| birthdate | DATE | A felhasználó születési dátum |
| address | VARCHAR(255) | Kiszállítási cím. |
| admin | TINYINT | Admin státuszt jelöl (0/1). |
| phonenumber | VARCHAR(50) | Telefonszám |
| resetPasswordToken | VARCHAR(255) | Jelszó visszaállítási token |
| resetPasswordExpires | BIGINT | A visszaállító token lejárati ideje |

5. orders tábla A leadott pizzarendeléseket kezeli.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| id | INT | Egyedi rendelési azonosító |
| userId | INT | Idegen kulcs a users táblára |
| pizzaId | INT | Idegen kulcs a pizzas táblára |
| pizzaNum | INT(11) | A rendelési pizza darabszáma |
| sizeId | INT | Idegen kulcs a size táblára. |
| address | VARCHAR(255) | A rendelés kiszállítási címe |
| userPhone | VARCHAR(50) | A rendelő telefonszáma |
| finalPrice | INT | A végleges ár |
| status | VARCHAR(50) | A rendelés státusza (pl. Kész). |

6. orderTops tábla: A rendeléshez tartozó feltéteket tárolja. Több a Többhöz kapcsolat az orders és a toppings táblák között.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| orderId | INT | Idegen kulcs az orders táblára |
| toppingId | INT | Idegen kulcs a toppings táblára. |

7. pizzaToppings tábla A pizzákhoz tartozó feltéteket tartalmazza. Több a Többhöz kapcsolat a pizzas és a toppings táblák között.

|  |  |  |
| --- | --- | --- |
| Oszlop | Típus | Leírás |
| pizzaId | INT | Idegen kulcs a pizzas táblára |
| toppingId | INT | Idegen kulcs a toppings táblára. |

Kapcsolatok Egy-Többhöz kapcsolatok:

orders → users

orders → size

orders → pizzas

Több-Többhöz kapcsolatok:

pizzas ↔ toppings a pizzaToppings táblán keresztül.

orders ↔ toppings az orderTops táblán keresztül.

Alapértelmezett Adatok Előre definiált pizzák:

|  |  |  |
| --- | --- | --- |
| Név | Ár(FT) | Kép URL |
| Margherita | 1000 | src/assets/image/margareta.jpg |
| Pepperoni | 1200 | src/assets/image/pepi.jpg |
| Hawaii | 1300 | src/assets/image/hawaii.jpg |

Előre definiált feltétek:

|  |  |
| --- | --- |
| Név | Ár(FT) |
| Sajt | 200 |
| Pepperoni | 300 |
| Ananász | 250 |
| Sonka | 300 |

Pizza-Feltét Kapcsolatok:

|  |  |
| --- | --- |
| Név | Ár(FT) |
| Margherita | Sajt |
| Pepperoni | Sajt, Pepperoni |
| Hawaii | Sajt, Ananász, Sonka. |

Frontend

A frontend Vue.js alapú, és a következő főbb komponensekből áll:

**HomePage.vue:** A főoldal, ahol a felhasználók megtekinthetik az étlapot és rendelhetnek pizzát.

**MenuPage.vue:** Az étlap oldal, ahol a felhasználók böngészhetik a pizzák listáját és rendelhetnek.

**AdminPage.vue:** Az Admin felület, itt lehet majd a pizzákat és a felhasználókat kezelni.

**Footer.vue:** Az oldalak által használt footer.

**ForgotPasswordPage.vue:** Az oldal melyen képes megadni a regisztrált email címét és küldünk önnek a jelszó helyreállító linket

**LoginPage.vue:** A első oldal, amit lát a felhasználó amikor az oldalra lép, itt képes a felhasználó egyszerre belépni, és regisztrálni is.

**Navbar.vue:** A felső navigációs sáv, használatával képes az oldalaink közt a felhasználó mozogni.

**OrderPage.vue:** Az oldal melyen képes a felhasználó rendelést leadni a weboldalon.

**OrdersPage.vue:** Ezen az oldalon képesek lesznek az adminok és dolgozók megfigyelni a bejövő rendeléseket és azoknak státuszát változtatni

**ProfilePage.vue:** Ezen az oldalon a felhasználó, ha szeretné változtathat meg számos adatot melyet mi tárolunk a bejelentkezés, és kiszállítás gyorsaságának érdekében.

**ResetPasswordPage.vue:** Ezt az oldalt a felhasználó abban az esetben tekinti meg ha valamilyen módon elfelejtette jelszavát, az emailben megkapott linkre kattintva itt meg tudja változtatni.

Backend

A backend Node.js és Express alapú, és a következő főbb részekből áll:

**app.js:** Az Express alkalmazás inicializálása és a fő útvonalak beállítása.

**routes:** Az alkalmazás különböző útvonalai.

*auth.js*: Bejelentkezési és regisztrációs útvonalak.

*profile.js*: Felhasználói profil útvonalak.

*pizzas.js*: Pizzák listájának lekérdezése.

**controllers**: Az útvonalakhoz tartozó vezérlők.

*loginController.js:* Bejelentkezési logika.

*registerController***.***js***:** Regisztrációs logika.

**services:** Az üzleti logika és adatbázis műveletek.

*loginService.js***:** Bejelentkezési szolgáltatás**.**

*registerService.js: Regisztrációs szolgáltatás.*

**middleware:** Köztes rétegek, például hitelesítés.

*auth.js:* JWT token hitelesítés.

**models**: Az adatbázis modellek.

*db.js***:** MySQL adatbázis kapcsolat beállítása.

Tesztelés

A projekt teszteléséhez a Jest és supertest könyvtárakat használjuk.

Tesztfájlok

A tesztfájlok a tests mappában találhatók:

**authRoutes.test.js:** Tesztek a bejelentkezési és regisztrációs útvonalakhoz.

**profileRoutes.test.js:** Tesztek a felhasználói profil útvonalakhoz.

**pizzaRoutes.test.js:** Tesztek a pizzák listájának lekérdezéséhez.

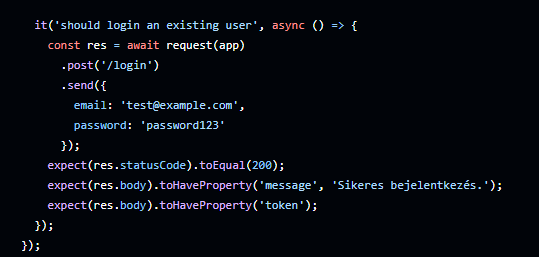
Tesztek futtatása

A tesztek futtatásához használd a következő parancsot a terminálban: **npm test**

Példák

Bejelentkezés

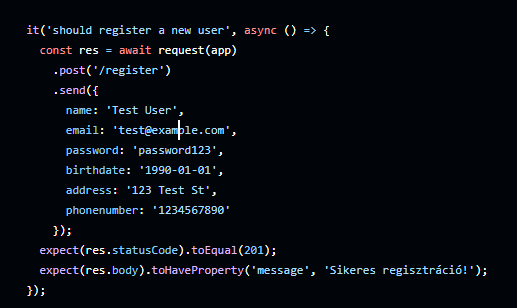
A bejelentkezési útvonal a /login végponton érhető el. A kérésnek tartalmaznia kell az email címet és a jelszót.

Példa kérés: 

Regisztráció

A regisztrációs útvonal a /register végponton érhető el. A kérésnek tartalmaznia kell a felhasználó nevét, email címét, jelszavát, születési dátumát, címét és telefonszámát.

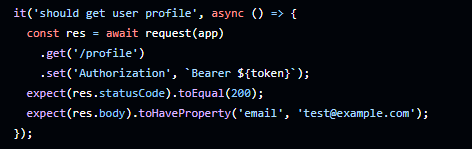
Példa kérés:



Profil lekérés

A felhasználói profil lekérdezése a /profile végponton érhető el.

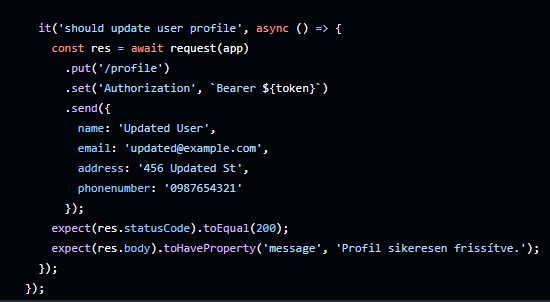
Példa kérés:



Profil frissítése

A felhasználói profil frissítése a /profile végponton érhető el. A kérésnek tartalmaznia kell egy érvényes JWT tokent az Authorization fejlécben, valamint a frissítendő adatokat.

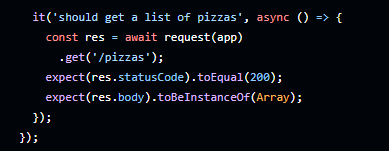
Példa kérés:



Pizzák lekérése

A pizzák listájának lekérdezése a /pizzas végponton érhető el.

Példa kérés:



Hibakeresés

Ha problémák merülnek fel a tesztek futtatása során, ellenőrizd a következőket:

Győződj meg róla, hogy az **adatbázis elérhető** és **megfelelően van beállítva**.

Ellenőrizd, hogy a **szükséges csomagok telepítve** vannak.

Győződj meg róla, hogy a tesztadatok **megfelelően vannak beállítva** és tiszták a tesztek futtatása előtt.