

Product-registry jegyzőkönyv

Csonka Patrik

CMU4ZN

PTI

2024.06.27

Bevezetés

Ez a projekt egy egyszerű webalkalmazás backend és frontend részeinek kódját tartalmazza. A backend Express és MongoDB használatával készült, míg a frontend Angular segítségével van megvalósítva. Az alkalmazás felhasználókezelést és termékkezelést biztosít, beleértve a regisztrációt, bejelentkezést, termékek listázását és hozzáadását.

Backend

A backend az alábbi főbb komponensekből áll:

1. Express alkalmazás létrehozása

A `const express = require('express');` sorral kezdődik az Express alkalmazás létrehozása. Az Express egy minimalista és rugalmas Node.js keretrendszer, amelyet webalkalmazások és API-k készítésére használnak.

```
const express = require('express');
const app = express();
app.use(bodyParser.json());
app.use(cors());
```

2. Middleware beállítása

Az alkalmazás két fő middleware-t használ: body-parser és cors.

```
const bodyParser = require('body-parser');
const cors = require('cors');
app.use(bodyParser.json());
app.use(cors());
```

body-parser: Segítségével az alkalmazás képes lesz JSON formátumú kérések feldolgozására.

cors: Engedélyezi a Cross-Origin Resource Sharing-et, ami lehetővé teszi, hogy a frontend (ami lehet különböző domainen van) hozzáférjen a backendhez.

3. MongoDB kapcsolódás

A MongoDB egy NoSQL adatbázis, amelyet az alkalmazás az adatok tárolására használ. A mongoose csomagot használjuk a MongoDB-hez való csatlakozáshoz és az adatok kezeléséhez.

4. Adatbázis séma és modellek

Két fő modellünk van: User és Product. Mindkettő mongoose séma alapján van definiálva.

```
const UserSchema = new mongoose.Schema({
  username: { type: String, required: true, unique: true },
  password: { type: String, required: true },
});

const ProductSchema = new mongoose.Schema({
  name: { type: String, required: true },
  price: { type: Number, required: true },
  quantity: { type: Number, required: true },
});

const User = mongoose.model('User', UserSchema);
const Product = mongoose.model('Product', ProductSchema);
```

5. JWT autentikáció

A JWT (JSON Web Token) segítségével történik a felhasználók azonosítása. A authenticateJWT middleware ellenőrzi a kérésekhez csatolt tokeneket.

6. Felhasználókezelő végpontok

Két fő végpont van a felhasználók számára: regisztráció és bejelentkezés.

7. Termékkezelő végpontok

A termékek listázása és hozzáadása a következő végpontokon történik, amelyeket csak autentikált felhasználók érhetnek el.

```
app.get('/products', authenticateJWT, async (req, res) => {
  const products = await Product.find();
  res.send(products);
});

app.post('/products', authenticateJWT, [
  check('name').not().isEmpty(),
  check('price').isNumeric(),
  check('quantity').isNumeric()
], async (req, res) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) {
    return res.status(400).json({ errors: errors.array() });
  }

  const { name, price, quantity } = req.body;
  const product = new Product({ name, price, quantity });
  await product.save();
  res.send({ message: 'Product added successfully' });
});
```

8. Szerver indítása

Az alkalmazás a 3000-es porton fut.

```
app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

Frontend

Az Angular frontend komponensei az alábbi fő részekből állnak:

1. AuthService

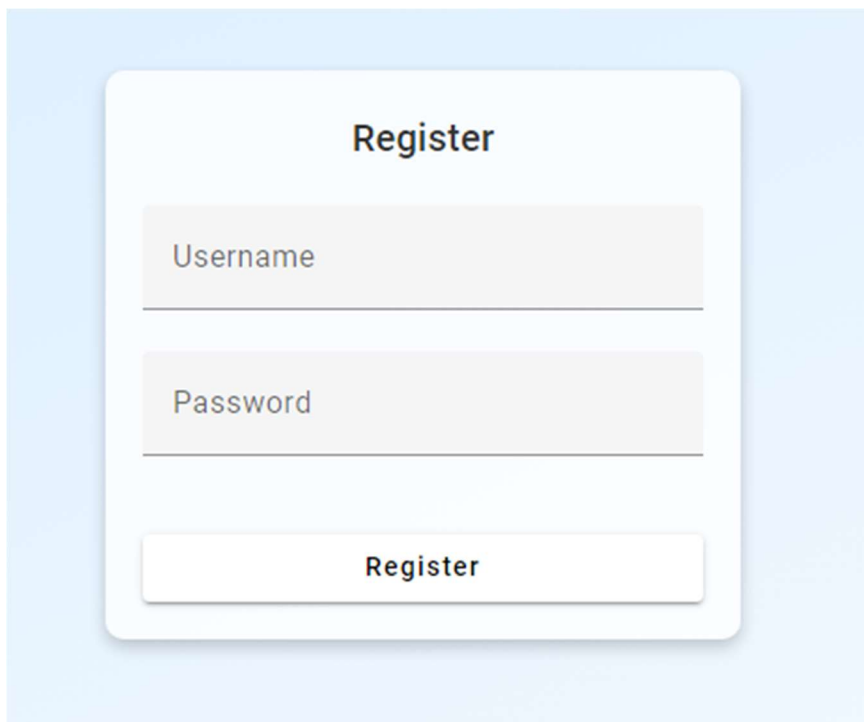
Ez a szolgáltatás kezeli a felhasználók autentikációját, beleértve a regisztrációt, bejelentkezést és a token kezelését.

2. ProductService

Ez a szolgáltatás kezeli a termékekkel kapcsolatos műveleteket, mint például a termékek lekérdezése és hozzáadása.

3. RegisterComponent

A regisztrációs komponens kezeli a felhasználók regisztrációját.

A screenshot of a web application's registration form. The form is titled "Register" and is set against a light blue background. It features two input fields: "Username" and "Password", both with light gray borders and placeholder text. Below these fields is a white "Register" button with a thin gray border. The entire form is enclosed in a white rounded rectangle with a subtle drop shadow.

Register

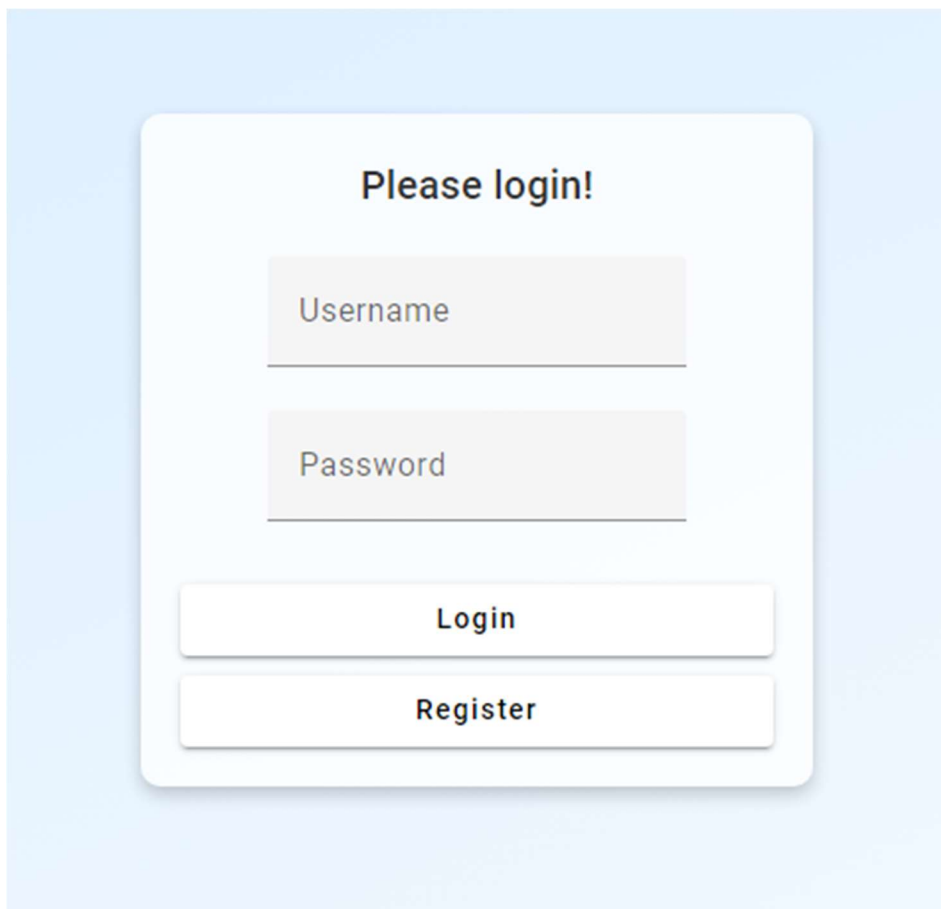
Username

Password

Register

4. LoginComponent

A bejelentkezési komponens kezeli a felhasználók bejelentkezését.



The image shows a login form titled "Please login!". It features two input fields: "Username" and "Password". Below these fields are two buttons: "Login" and "Register". The form is centered on a light blue background.

Please login!

Username

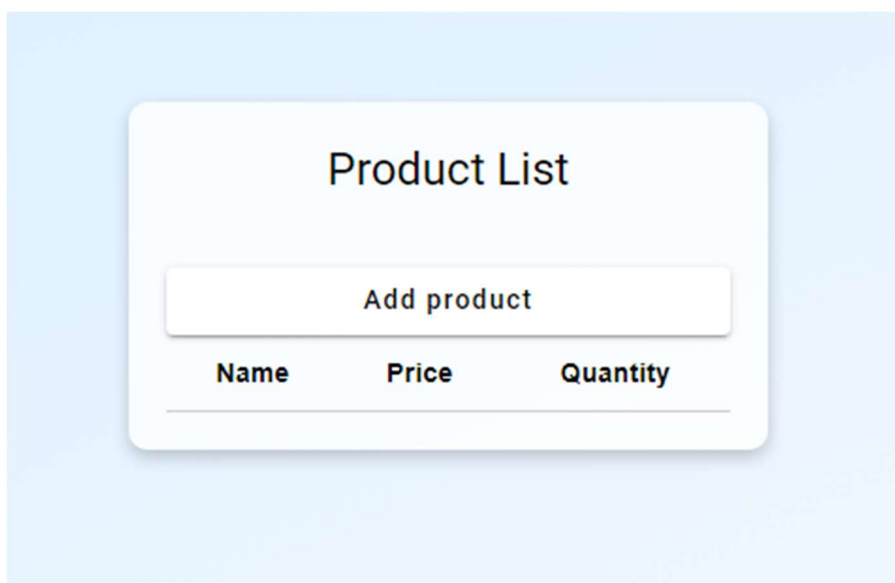
Password

Login

Register

5. ProductListComponent

Ez a komponens listázza a termékeket.



The image shows a product list interface titled "Product List". It features a button labeled "Add product" and a table with three columns: "Name", "Price", and "Quantity". The table is currently empty. The interface is centered on a light blue background.

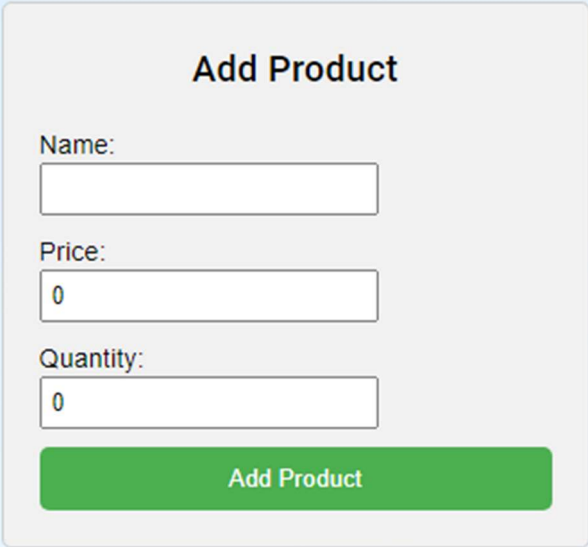
Product List

Add product

Name	Price	Quantity
------	-------	----------

6. ProductAddComponent

Ez a komponens új termékek hozzáadására szolgál.



The image shows a web form titled "Add Product" centered on a light blue background. The form itself is a light gray rectangle with rounded corners. It contains three input fields stacked vertically. The first field is labeled "Name:" and is empty. The second field is labeled "Price:" and contains the number "0". The third field is labeled "Quantity:" and also contains the number "0". Below these fields is a green rectangular button with the text "Add Product" in white.

Összefoglalás

Ez a projekt egy teljes körű webalkalmazás, amely a backend és a frontend komponenseit egyaránt tartalmazza. A backend Express és MongoDB segítségével készült, és biztosítja a felhasználók regisztrációját, bejelentkezését, valamint a termékek kezelését. A frontend Angular segítségével készült, és biztosítja a felhasználók regisztrációját, bejelentkezését, valamint a termékek listázását és hozzáadását.