Vizsgaremek dokumentáció

Készítették:

Réti-Nagy Tamás

Handbauer Péter

Felhasználói dokumentáció

Az általunk készített szoftver egy általános raktárkészlet kezelő alkalmazás, amelyben egyszerűen megtalálhatóak a termékek raktárokon átívelően.

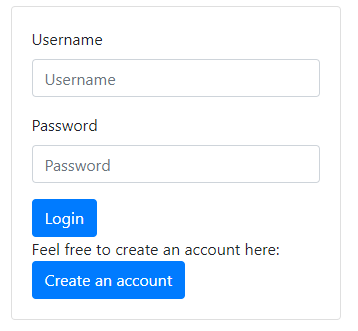
Az alkalmazásban lehetőségünk van raktárakat, polcrendszereket, polcokat, illetve termékeket kezelni, ezeket a termékeket polcokra helyezni, levenni a polcokról, illetve keresni a termékek között, megnézni a termékek elérhetőségét.

Felhasználói felület

Raktárkészlet kezelő alkalmazás

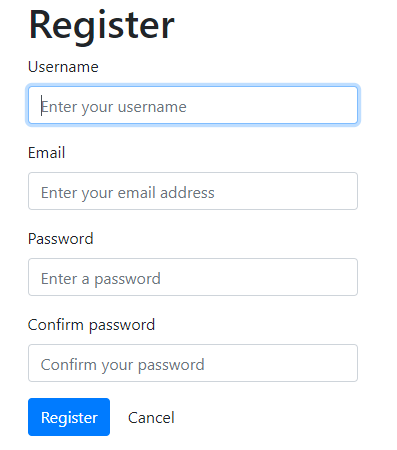
Bejelentkezés/Regisztráció/Kijelentkezés

Az alkalmazás használatához minden esetben szükség van felhasználói fiókra, illetve bejelentkezésre. Az alkalmazás megnyitásakor automatikusan a bejelentkezési felületre kerülünk.



Ezen a felületen a megfelelő bejelentkezési adatok megadása után a „Login” gombra kattintva ezek ellenőrzésre kerülnek, és ha sikeres az azonosítás, akkor visszakerülünk az alkalmazás felületére.

Ha nem rendelkezünk felhasználói fiókkal, akkor a „Create an account” gombra kattintva létrehozhatunk egyet:



A regisztrációhoz tartozó mezők követelményei:

Email:

– Szintaktikailag érvényes e-mail címet kell megadni.

Jelszó:

– Legalább 6 karakter.

– Legalább egy alfanumerikus karakter (A-Z, 0-9).

– Legalább egy nem alfanumerikus karakter/szimbólum (például , %).

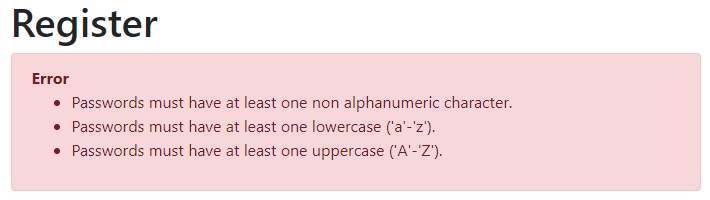
– Legalább egy kisbetűs karakter.

– Legalább egy nagybetűs karakter.

A mezők kitöltése után kattintsunk a "Register" gombra.

• Sikeres regisztráció esetén a bejelentkezési felületre kerülünk, itt már be tudunk jelentkezni a frissen készített felhasználónkkal.

• Sikertelen regisztráció esetén visszakerülünk a regisztrációs felületre, ahol megtudhatjuk a sikertelenség okát:



A Cancel gombra kattintás esetén visszatérünk az alkalmazásunkhoz.

Alkalmazott technológiák

Frontend

* Angular

Backend

A felhasználókezelést végző szerver:

* .NET 6
* Entity Framework Core: A backend alkalmazás és az adatbázis közötti kapcsolat megvalósítására.
* IdentityServer4

Az alkalmazást kiszolgáló WebAPI:

* .NET 6
* Entity Framework Core: A backend alkalmazás és az adatbázis közötti kapcsolat megvalósítására.

Adatbázis:

* MSSQL Server 2022
* SQL Server Management Studio 19 - Az adatbázisok kezelését/monitorozását segítő eszköz.

A WebAPI kód felépítése:

Controllers

A WebAPI Endpointjainak gyűjtője, ez a belépési pont az Angular alkalmazás felől

Dtos

Az endpointokon érkező és kimenő adatok adatszerkezetei.

Entities

Az adatbázis entitásainak leképezése a WebAPI oldalán.

Data

Az adatbázis felépítése, ileltve entitásai kapcsolatainak leképezése a WebAPI oldalán

Az adatbázis kezdeti adatainak feltöltését segítő DatabaseSeed megvalósítása.

– Ha nem létezik az adatbázis, akkor létrehozza azt, és migrálja is a legfrissebb verzióra.

– Feltölti az adatbázist minta adatokkal.

Repositories

Repository Design Patternt megvalósító osztályok, amelyek az adatbázisban lévő adatok eléréséhez nyújtanak segítséget.

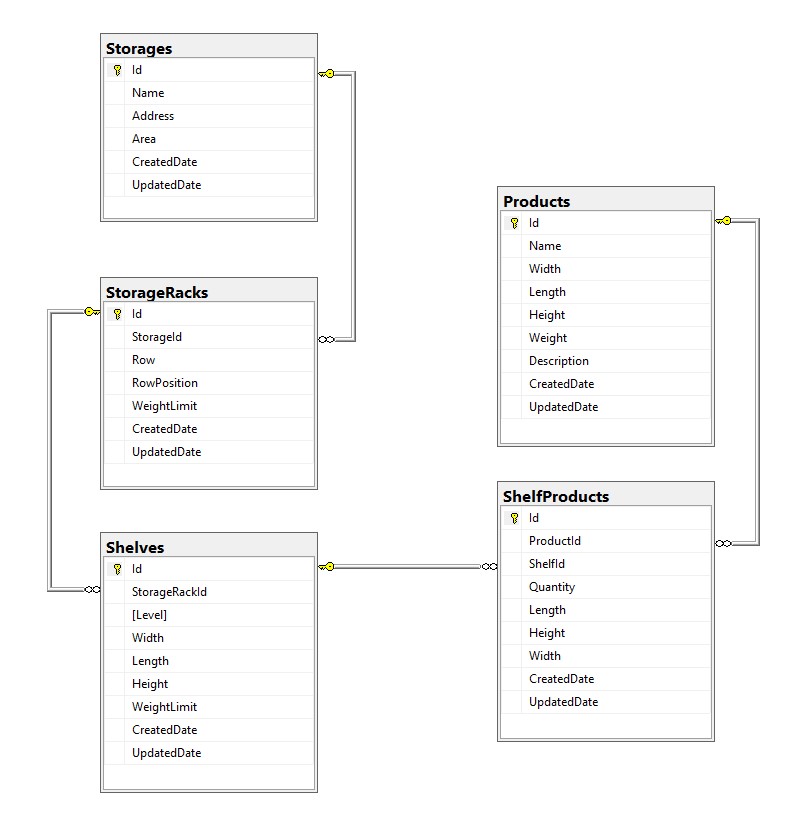
Services

A Repository-k és a Controller-ek közötti réteg, amely a modellek átalakítását, üzleti logikát tartalmaz.

Migrations

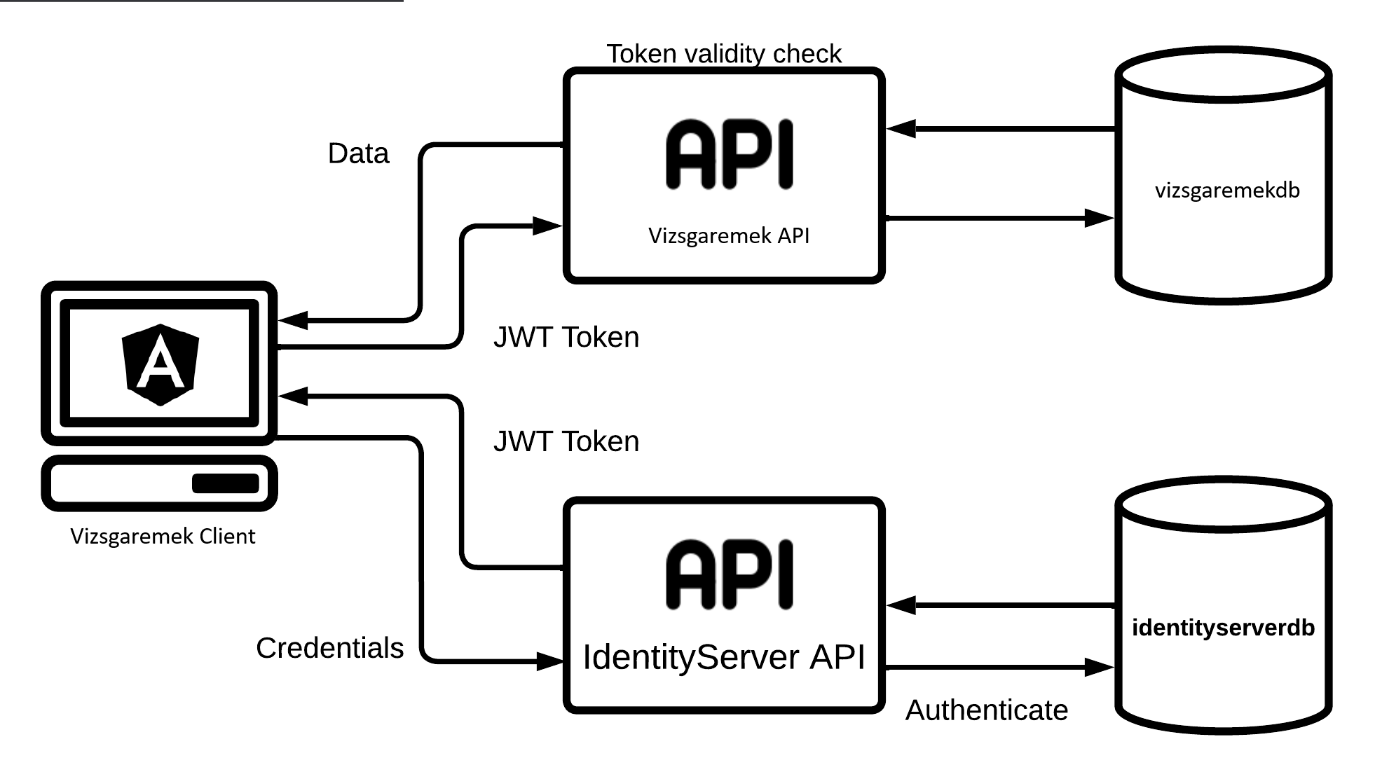
Az adatbázist felépítő migrációkat tartalmazza

Az adatbázis felépítése:



Az adatbázis modelljeinek kapcsolata.

Szoftver architectúra:



Az architectúra tervből látszik az adatok áramlása. A felhasználói felületen először azonosítani kell magunkat a bejelentkezéssel. Ez a kérés az IdentityServer felé irányul, ami ellenőrzi, hogy hitelesek-e a belépési adataink. Ha sikeres a bejelentkezés, akkor egy tokent ad vissza az Angular alkalmazásnak, és eltárolásra kerül.

Amikor az Angular alkalmazás a Vizsgaremek API felé intéz hívást, akkor a request headerjében küldi a tokent is. Az API, ha az endpoint védve van az [Authorize] attribute-tal, ellenőrzi a token hitelességét.

• Hiteles token esetén a WebAPI biztosítja az adatok elérését.

• Hibás token esetén a WebAPI 401 - Unauthorized hibakóddal válaszol.

WebAPI által biztosított végpontok:

ProductController:

GET /api/Product – Az összes Product lekérése

POST /api/Product – Új Product hozzáadása

Minta request body:

**{**

**"id": 0,**

**"name": "string",**

**"width": 0,**

**"length": 0,**

**"height": 0,**

**"weight": 0,**

**"description": "string"**

**}**

GET /api/Product/{id} – Egy termék lekérdezése „id” alapján

PUT /api/Product/{id} – Egy termék frissítése „id” alapján. A request bodyban szükséges megadni a frissítendő adatokat:

**{**

**"id": 0,**

**"name": "string",**

**"width": 0,**

**"length": 0,**

**"height": 0,**

**"weight": 0,**

**"description": "string"**

**}**

DELETE /api/Product/{id} – Egy termék törlése „id” alapján

GET /api/Product/{id}/ProductShelves – Megadja, hogy a termék mely polcokon van jelen

GET /api/Product/Search – Megadott string alapján keres a termékek között, és visszaadja azokat, amelyek megfelelnek a keresési feltételnek.