**BZSH LELTÁR PROGRAM**

**dokumentáció**

Fejezetek:

1. Bevezetés
2. Fejlesztői dokumentáció
3. Felhasználói dokumentáció
4. Tesztelés
5. Záró rész

Bevezetés:

A következő dokumentáció kívánja bemutatni a BZSH Leltár programját, annak specifikációit, elkészültét.

A program back-end szoftverét Express.js keretrendszer szolgáltatja, MySQL adatbázis környezetben.

Az alkalmazás front-end szoftver része Angular keretrendszerben íródott.

Az alkalmazás célja, hogy egy felhasználóbarát felületet biztosítson, amely megfelel az iskola alapvető igényeinek, és támogatja a hatékony leltározást.

Fejlesztői dokumentáció:

Használt eszközök, technológiák:

Visual Studio Code

Postman, JMETER, Swagger

MySQL adatbázis környezet – MySQL 8.0

Express.js 4.21.2

BACK-END

Adatbázis felépítés:

Táblák:

**item\_name:**

A tábla célja, hogy egy listát biztosítson az iskolában fellelhető és leltárazandó tárgyakból.

Oszlopai:

ID: Azonosító a tárgyaknevekhez. – integer

ITEM: A tárgy neve. – string

**item:**

Felhasználva az item\_name táblát egyedi azonosítást biztosít minden tárgy számára így különbséget téve pl. laptop és laptop közt.

Oszlopai:

ID: Azonosító a tárgyakhoz. – integer

ITEMNAMEID: Idegenkulcs ami hivatkozza a tárgynevet. – integer

**storage\_place:**

A tábla célja, hogy egy listát biztosítson az iskolában fellelhető tárolókból.

Oszlopai:

ID: Azonosító a tárolókhoz. – integer

STORAGE: A tároló neve. – string

**storage\_conn:**

A tábla célja, hogy összeköttetést biztosítson tárgyak és tárolók között, ehhez felhasználva az item és storage\_place táblák adatait.

Oszlopai:

ID: Azonosító – integer

ITEMID: Idegenkulcs, ami hivatkozza a tárgyat. – integer

STORAGEPLACEID: Idegenkulcs, ami hivatkozza a tárolót. – integer

**user:**

Eltárolja a felhasználókat és tulajdonságait.

Oszlopai:

ID: Azonosító a felhasználókhoz. – integer

USEREMAIL: Tartalmazza a felhasználóhoz tartozó emailt. - string

USERPASSWORD: Tartalmazza a felhasználóhoz tartozó jelszót, egy hash formájában. - string

ISADMIN: Igaz/Hamis értékpár jelzi, hogy a felhasználó admin vagy felhasználói szerepkörrel rendelkezik-e. -boolean

ISFIRSTLOGIN: Igaz/Hamis értékpár, a regisztrációs műveletek egyediségéből adódóan ezen érték alapján a felhasználó jelszócseréjét kötelezi. -boolean

ISDISABLED: Igaz/Hamis értékpár jelzi, hogy a felhasználó állapota aktív vagy tiltott. – boolean

CREATEDAT: Időbélyeg, a létrehozás idejét rögzíti. – timestamp

UPDATEDAT: Időbélyeg, a felhasználóval kapcsolatos változások idejét rögzíti. – timestamp

CREATEDBY: Idegenkulcs, ami hivatkozza létrehozó személyt. – integer

**logs:**

A logs tábla rögzíti minden tárgyakkal kapcsolatos művelet egyenként történő rögzítését.

Oszlopai:

ID: Azonosító – integer

ITEMID: Idegenkulcs, ami hivatkozza a tárgyat. – integer

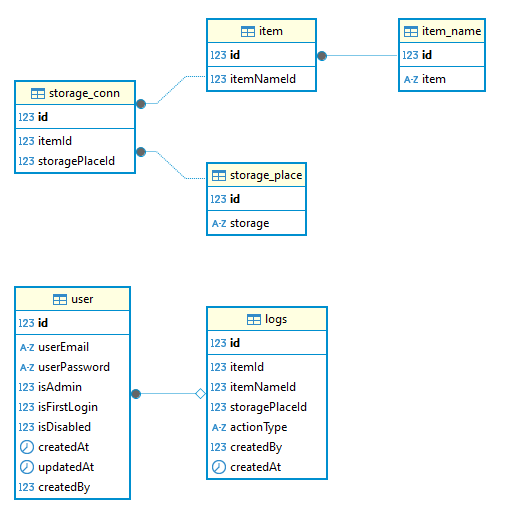
ITEMNAMEID: Idegenkulcs ami hivatkozza a tárgynevet. – integer

STORAGEPLACEID: Idegenkulcs, ami hivatkozza a tárolót. – integer

ACTIONTYPE: Enum lista ('ADD', 'UPDATE', 'DELETE'), a művelet milyenségét jelzi. – enum

CREATEDBY: Idegenkulcs, ami hivatkozza létrehozó személyt. – integer

CREATEDAT: Időbélyeg, a létrehozás idejét rögzíti. – timestamp



Mappa struktúra

BZSHStorage

│

└──BACKEND

│

├──associations

│ └──associations.js

│

├──models

│ ├──ItemModel.js

│ ├──ItemNameModel.js

│ ├──LogModel.js

│ ├──StorageConnModel.js

│ ├──StoragePlaceModel.js

│ └──UserModel.js

│

├──router

│ └──IeltarRouter.js

│

├──services

│ ├──auth

│ │ └──User.service.js

│ │

│ ├──log

│ │ ├──ExcelLog.service.js

│ │ └──Logs.service.js

│ ├──storage

│ │ ├──Item.service.js

│ │ ├──ItemName.service.js

│ │ ├──Storage.service.js

│ │ └──StorageConn.service.js

│ ├──Storage.service.js

│ └──StorageConn.service.js

│

├──.gitignore

├──config.js

└──index.js

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Végpont | Metódus | Leírás | Elérés |
| 1. | /storagePlace | GET | Tárolók listázása. | felhasználó |
| 2. | /storagePlace | POST | Tároló felvétele. | admin |
| 3. | /storagePlace/{id} | DELETE | Tároló törlése azonosító alapján. | admin |
| 4. | /itemName | GET | Tárgynevek listázása. | felhasználó |
| 5. | /itemName | POST | Tárgynevek felvétele. | admin |
| 6. | /itemName/{id} | DELETE | Tárgynév törlése azonosító alapján. | admin |
| 7. | /item | GET | Tárgyak listázása. | admin |
| 8. | /item | POST | Tárgyak felvétele. | felhasználó |
| 9. | /item | PUT | Tárgyak áthelyezése. | admin |
| 10. | /item | DELETE | Tárgyak törlése. | admin |
| 11. | /users | GET | Felhasználók listázása. | admin |
| 12. | /disable | PATCH | Felhasználók letiltása | admin |
| 13. | /enable | PATCH | Felhasználók feloldása. | admin |
| 14. | /demote | PATCH | Admin jog elvétele. | admin |
| 15. | /promote | PACTH | Admin jog kiosztása. | admin |
| 16. | /register | POST | Felhasználó rögzítése. | admin |
| 17. | /login | POST | Bejelentkezés. | felhasználó |
| 18. | /firstLogin | PUT | Bejelentkezés és jelszóváltás. | felhasználó |
| 19. | /passwordChange | PUT | Jelszóváltás. | felhasználó |
| 20. | /log | POST | Műveleti napló letöltés. | admin |

Végpont kifejtés:

1. A GET http metódusú /storagePlace végpont célja a rögzített tárolóhelyek listázása a felhasználó számára, hogy azokból választva megelőzzük a tárgyrögzítés során előforduló hibákat.
2. A POST http metódusú /storagePlace végpont célja a tárolóhelyek rögzítése.
3. A DELETE http metódusú /storagePlace/{id} végpont célja a tárolóhelyek törlése esetleges hibás rögzítés vagy tárolóhely megszűnés esetén.
4. A GET http metódusú /itemName végpont célja a rögzített tárgynevek listázása a felhasználó számára, hogy azokból választva megelőzzük a tárgyrögzítés során előforduló hibákat.
5. A POST http metódusú /itemName végpont célja a tárgynevek rögzítése.
6. A DELETE http metódusú /itemName{id} végpont célja a tárgynevek törlése esetleges hibás rögzítés vagy tárgynevek megszűnése esetén.
7. A GET http metódusú /item végpont célja a rögzített tárgyak listázása.
8. A POST http metódusú /item végpont célja új tárgyak rögzítése.
9. A PUT http metódusú /item végpont célja a rögzített tárgyak termek közti áthelyezését végzi.
10. A DELETE http metódusú /item végpont célja a már rögzített tárgyakból való törlés selejtezett tárgyak esetén.
11. A /users végpont a felhasználókat listázza.
12. A /disable végpont a felhasználók letiltását kezeli.
13. A /enable végpont a letiltott felhasználók feloldását kezeli.
14. A /demote végpont az admin felhasználók jogának elvételét kezeli.
15. A /promote végpont az felhasználók adminná promótálását kezeli.
16. A /register végpont az admin felhasználó számára nyújt lehetőséget vele egyenrangú admint vagy kevesebb jogkörrel rendelkező felhasználót felvennie. A végpont e-mail alapján küld egy random jelszót az új felhasználónak.
17. A /login végpont végzi a bejelentkeztetést.
18. A /firstLogin végpont az új felhasználók számára és elfelejtett jelszó esetén esetén kötelező jelszóváltást alkalmaz.
19. A /passwordChange végpont kezeli a jelszóváltást a felhasználóknak, e-mail alapján.
20. A /log végpont egy Excel alapú listázást végez, ami letölthetővé válik. Az Excelben rögzülnek a tárgyakhoz tartozó tulajdonságok, műveletek és a műveletet elvégző felhasználó.

Request bodyk:

|  |  |  |
| --- | --- | --- |
| 1. | /storagePlace | nincs |
| 2. | /storagePlace | **{**  **"storage": "Main Warehouse"**  **}** |
| 3. | /storagePlace/{id} | nincs |
| 4. | /itemName | nincs |
| 5. | /itemName | **{**  **"item": "Sample Item Name"**  **}** |
| 6. | /itemName/{id} | nincs |
| 7. | /item | nincs |
| 8. | /item | **{**  **"itemNameId": 1,**  **"storagePlaceId": 2,**  **"quantity": 3**  **}** |
| 9. | /item | **{**  **"itemIdList": [**  **1, 2, 3**  **],**  **"newStoragePlaceId": 4**  **}** |
| 10. | /item | **{**  **"itemIdList": [**  **1, 2, 3**  **]**  **}** |
| 11. | /users | nincs |
| 12. | /disable | **{**  **"userEmail": "string"**  **}** |
| 13. | /enable | **{**  **"userEmail": "string"**  **}** |
| 14. | /demote | **{**  **"userEmail": "string"**  **}** |
| 15. | /promote | **{**  **"userEmail": "string",**  **}** |
| 16. | /register | **{**  **"userEmail": "string",**  **"isAdmin": false**  **}** |
| 17. | /login | **{**  **"userEmail": "string",**  **"userPassword": "string"**  **}** |
| 18. | /firstLogin | **{**  **"userPassword": "string"**  **}** |
| 19. | /passwordChange | **{**  **"userEmail": "string"**  **}** |
| 20. | /log | **{**  "itemNameId"**: 0,**  "storagePlaceId"**: 0,**  "createdBy"**: 0,**  "fromDate"**: "string",**  "toDate"**: "string"**  **}** |

Funkciók és szervizek:

**AUTH**:

**User.service:**

**getUser:**

Feladat:

Az adatbázisban rögzített felhasználók lekérdezése és listázása.

Válasz: user objektum

Bemenő paraméterei:

nincs

Hibakezelés:

Hiba esetén "Failed to fetch user" hibaüzenettel válaszol.

**createUser:**

Feladat:

Felhasználók rögzítése az adatbázisba.

Válasz: "User created successfully"

Bemenő paraméterei:

userEmail – string, userPassword – string, isAdmin – boolean

Hibakezelés:

Hiba esetén "Failed to create user" hibaüzenettel válaszol.

**updateUser:**

Feladat:

Az adatbázisban rögzített felhasználók jelszócseréje.

Válasz: "Password changed'

Bemenő paraméterei:

userEmail – string, newPassword – string

Hibakezelés:

Hiba esetén "Failed to update user" hibaüzenettel válaszol.

**deleteAdminRole:**

Feladat:

Az adatbázisban rögzített felhasználók admin jogkörének elvétele.

Válasz: "User demoted from admin role"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error demoting user" hibaüzenettel válaszol.

**addAdminRole:**

Feladat:

Az adatbázisban rögzített felhasználók jelszócseréje.

Válasz: "User promoted to admin role"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error promoting user" hibaüzenettel válaszol.

**disableUser:**

Feladat:

Az adatbázisban rögzített felhasználók letiltása.

Válasz: " User disabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén " Error disabling user " hibaüzenettel válaszol.

**enableUser:**

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

**Utilfunctions.utils:**

checkPassword:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    validateAdmin:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    checkRequiredFields:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    authMiddle:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    sendEmail:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    genPassword:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    firstLoginFalse:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

    validatePassword:

Feladat:

Az adatbázisban rögzített felhasználók feloldása.

Válasz: "User enabled"

Bemenő paraméterei:

userEmail – string

Hibakezelés:

Hiba esetén "Error enabling user" hibaüzenettel válaszol.

FRONT-END:

**Felhasználói kézikönyv:**

Felhasználók számára:

A felhasználói jogkörrel bíró felhasználók egy limitált felületet láthatnak, amit egy bejelentkezés előz meg.

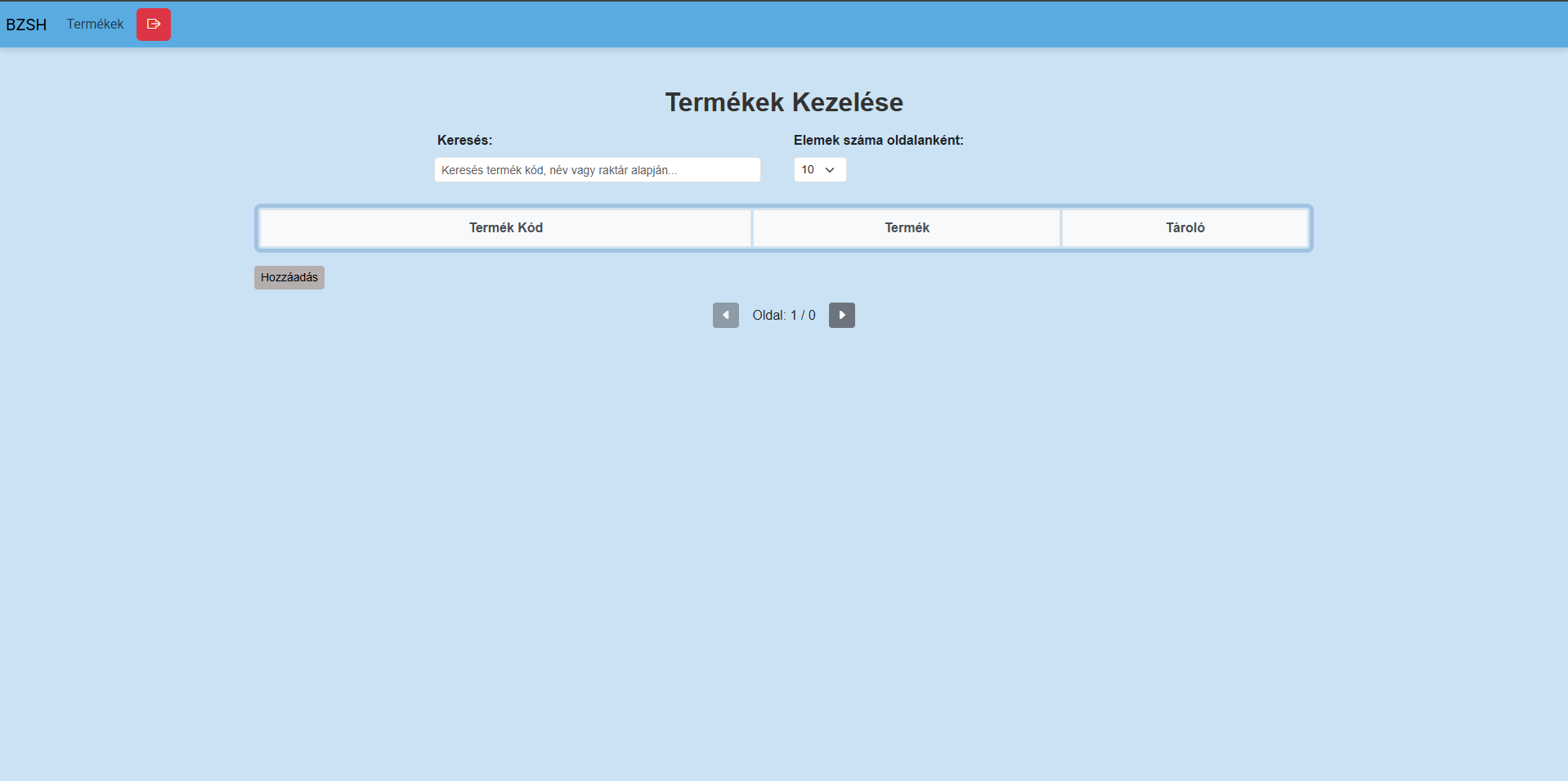
Ahogy az alábbi képen látható a bejelentkezéshez a felhasználónak 2 adatra van szüksége. Az e-mail címre, amin keresztül az admin rögzítette a felhasználót és az e-mail címre kiküldött jelszóra.



Ezt követően a felhasználó egy jelszóváltásra köteles.



A sikeres jelszóváltás után az új jelszóhasználatával ismételt belépést követel meg az oldal, mely, ha sikeres az alábbi képen látható felületre irányít.



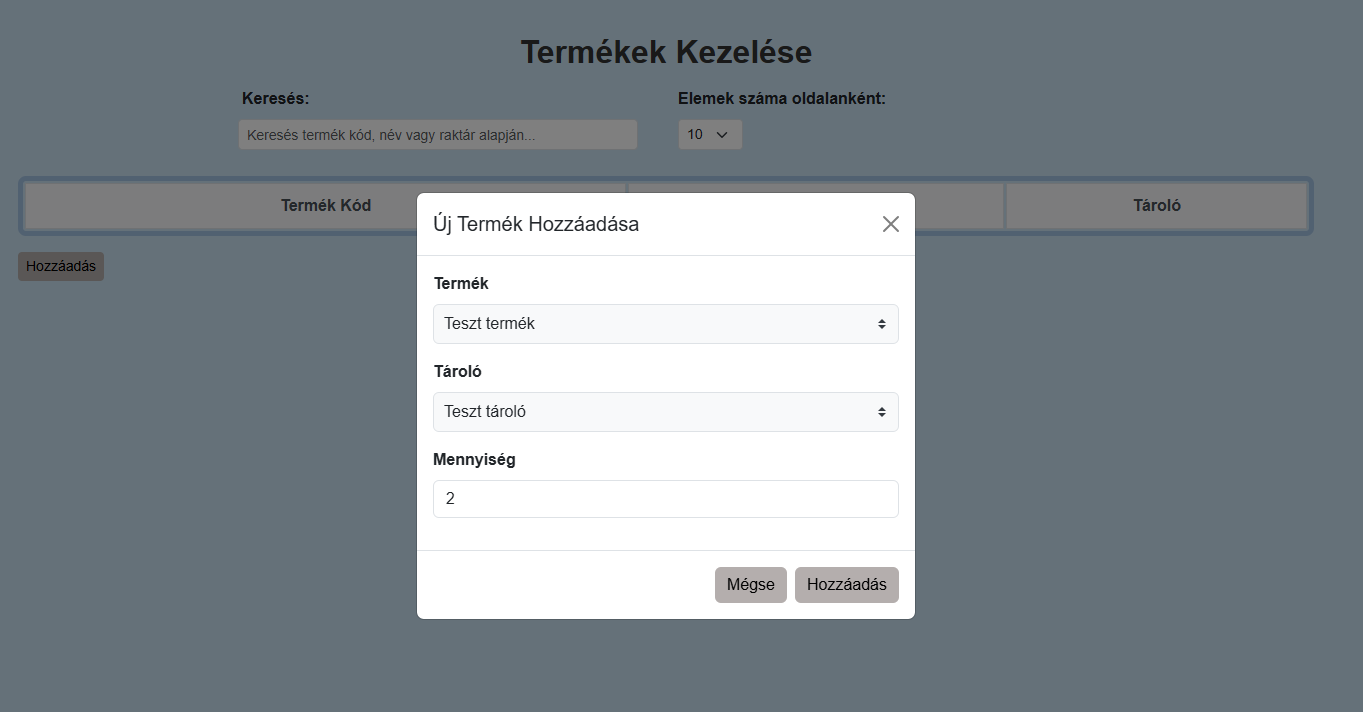
A jogosultságok limitálásából adódóan a felhasználó csak megtekinteni és rögzíteni tudja a leltárazandó tárgyakat.

A hozzáadás gombbal új egyedi tárgyat rögzíthetnek a felhasználók.



A rögzítés során a felhasználó egy listából elérhető termeket és tárgyneveket választhatja illetve megadhatja, hogy egy teremben hány darab van tárolva egy adat tárgyfajtából.

A Hozzáadással megerősíti és rögzíti a termékeket, a Mégse gombbal pedig kilép a folyamatból.

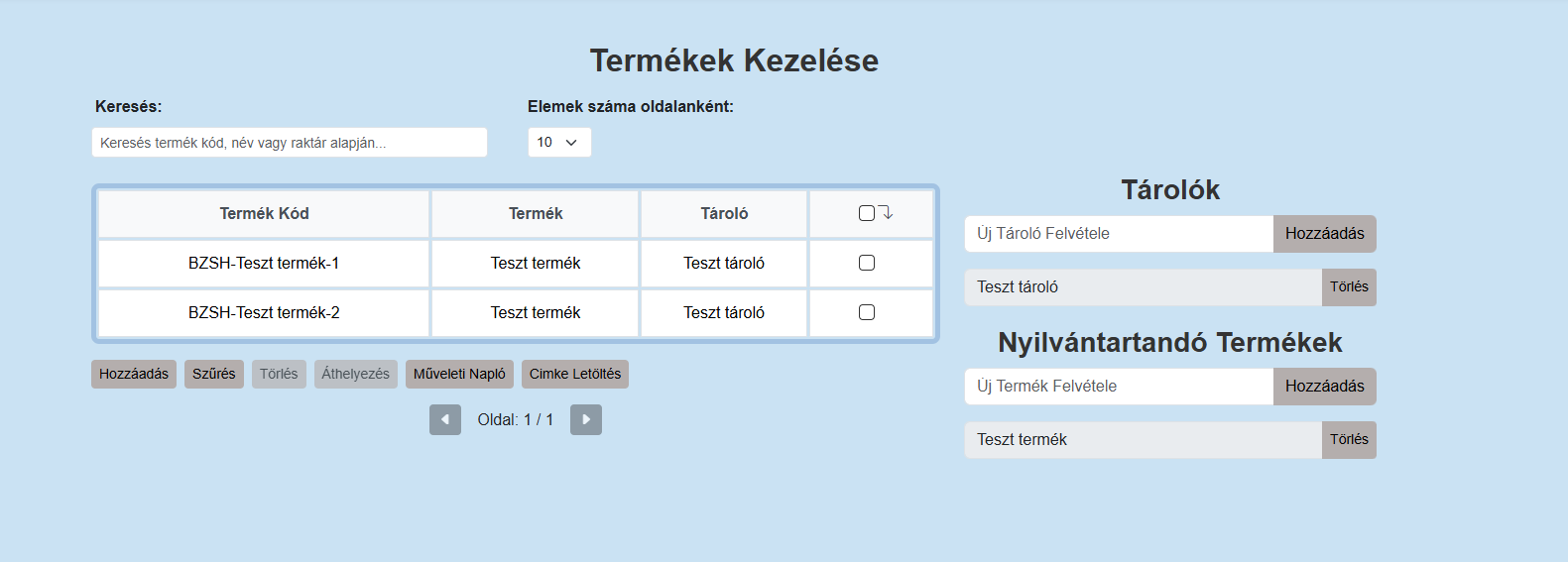


A folyamat végén a felületen láthatóak a rögzült tárgyak.

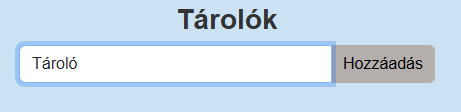
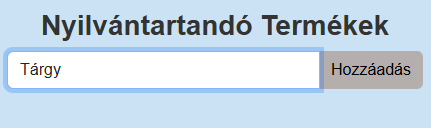


Admin felhasználók számára:

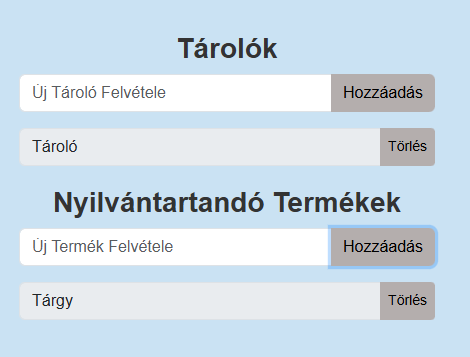
A különleges jogokkal nem rendelkező felhasználók által is elérhető felület az adminok számára sok funkcióval kibővül.



Az admin rögzíthet tárolókat és tárgyakat, ezáltal bővítve a listát, melyből az egyedi tárgy rögzítés során választhat a rögzítő, a listában szereplő tárolók és tárgyak nem törölhetők a listából, amíg bármely egyedileg rögzített tárgy használja valamely tulajdonságot.

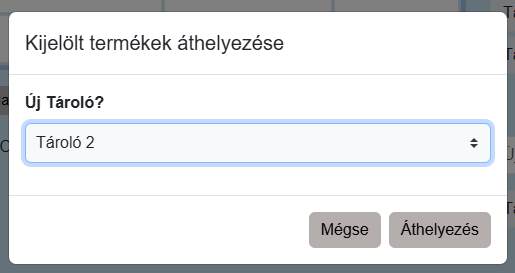
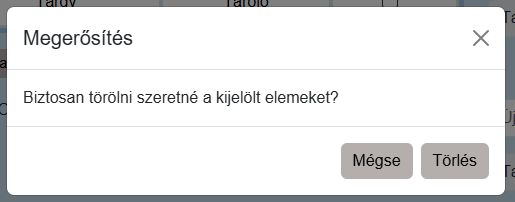
 

A sikeres rögzítés után:



A kijelölt tárgyak törölhetők és áthelyezhetők más tárolókba. Egyszerre több tárgy is kijelölhető és ugyanazon művelet alkalmazható rajtuk.



A rögzített tárgyakon elvégzett műveletek a háttérben rögzülnek, melyek egy letölthető excel állományban elérhetők az admin felhasználók számára. A napló letöltéséhez meghatározott szűrők használhatók.

Szűrők:

Tárgy: Kiválasztható hogy, mely konkrét tárgytípuson (pl. Laptop) szeretnénk a műveleteket lekérdezni. Egyszerre a szűrés egy tárgytípus megadását engedi, amennyiben szűrő üres, úgy minden tárgytípus bekerül.

Tároló: Kiválasztható hogy, mely konkrét tárolóban tárolt termékeken (pl. 2-es tároló) szeretnénk a műveleteket lekérdezni. Egyszerre a szűrés egy tároló megadását engedi, amennyiben szűrő üres, úgy minden tárolóban tárolt termék bekerül.

Felhasználó: Egy adott felhasználó által végzett műveleteket kérdez le. Egyszerre a szűrés egy felhasználó megadását engedi, amennyiben szűrő üres, úgy minden felhasználó művelete bekerül.

Kezdő és záró dátum: A tárgyak rögzítésének időpontját a felület elmenti így lehetőség van célzott periódusokban történt műveletek megtekintésére.

