



Vas Vármegyei Szakképzési Centrum  
Nádasy Tamás Technikum és Kollégium

# **PROJEKTFELADAT**

## **ACS (Access Control System)**

**Szilvás Nikolett**

**Balassa Olivér**

**Konzulens:  
Balics Gábor**

**2024**

# Nyilatkozat

Alulírott Szilvás Nikolett, Balassa Olivér Szoftverfejlesztő és -tesztelő szakos technikumi tanulók kijelentjük, hogy az ACS című szakdolgozat feladat kidolgozása a saját munkánk, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtuk fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeink saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásunk szerint hitelesek.

Csepreg, 2024.04.19.

---

hallgató

Csepreg, 2024.04.19.

---

hallgató

# Kivonat

## ACS

Az Access Control System egy forradalmi megoldás a kollégiumi menedzsment területén, amely kifejezetten a diákotthonok specifikus igényeire lett tervezve. Ez a rendszer azon alapszik, hogy egyszerűsítse és gyorsítsa fel a bejelentkezési folyamatot, megszüntetve a korábbi nehézkes eljárásokat, és biztosítva a lakók és a személyzet zökkenőmentes élményét.

Az Access Control System előnye azonban nem merül ki a kényelemben. Lehetővé teszi a diákok nyomon követését, így a személyzet könnyen figyelemmel kísérheti a lakók tartózkodási helyét. Ez nemcsak a biztonságot erősíti, hanem váratlan helyzetekben is nagy segítség, lehetővé téve gyors és hatékony válaszlépéseket vészhelyzetek esetén.

A rendszer képességei azonban még tovább nyúlnak, egy összetett szoftverkomponens beépítésével. Ez a kiváló program rendkívüli módon képes azonosítani a jelenleg az épületben tartózkodó technikai személyzetet. Ez a fajta képesség időt takarít meg, mivel nincs szükség a megfelelő szakember keresésére, ami gyors és hatékony válaszokat tesz lehetővé bármilyen karbantartási kérdésben, legyen az kritikus helyzet vagy látszólag jelentéktelen probléma, mint például egy elveszett zárszercs.

Az Access Control System által nyújtott előnyök egy olyan nyugodt és békés környezetet teremtenek a diákoknak, amelyben teljes mértékben élvezhetik a kollégiumi élet vibráló és gazdag tapasztalatát.

Ezen felül, a rendszer pozitív hatása túlmutat a diákélményen. Az üzemek egyszerűsítésének képessége kiemelkedő hatékonyságot eredményez az intézmény teljes területén. Ez javítja az erőforrások kiosztását és az irányítási gyakorlatokat, végül pedig az egész diákközösség és személyzet javát szolgálja.

Összefoglalva, az Access Control System egy emberi találékonyságot tükröző megoldás. Ez egy paradigmaváltást jelent a kollégiumi menedzsment területén, amely új korszakot hoz a még soha nem látott kényelem, és az általános jólét szempontjából a diákoknak, a személyzetnek és az egész intézménynek.

# Abstract

## ACS

In the ever-evolving realm of dormitory management, a groundbreaking solution has emerged: the Access Control System. This ingenious creation transcends the limitations of conventional access control and record-keeping systems, meticulously crafted to address the specific needs of student housing facilities.

The cornerstone of this system lies in its unwavering commitment to streamlining the check-in process. Gone are the days of cumbersome procedures that hinder both students and staff. The Access Control System ushers in an era of effortless entry, ensuring a smooth and seamless experience for all residents and personnel.

However, the system's brilliance extends far beyond mere convenience. It facilitates student tracking, empowering staff with the ability to effortlessly monitor the whereabouts of residents. This not only bolsters overall security but also proves invaluable in unforeseen circumstances, allowing for swift and effective responses during emergencies.

The system's capabilities extend even further, incorporating a sophisticated software component. This ingenious program boasts the remarkable ability to identify technical staff currently present on the premises. Such a feat eliminates the precious time wasted searching for qualified personnel, ensuring a prompt and efficient response whenever the need for maintenance expertise arises, be it a critical situation or a seemingly mundane issue like a lost locker key.

The benefits of the Access Control System's profound impact fosters a more relaxed and tranquil living environment for students. The system allows residents to fully embrace the vibrant and enriching experience of dormitory life.

Furthermore, the system's positive influence extends beyond the student experience. Its ability to streamline operations promotes remarkable efficiency across the entire institute. This translates to optimized resource allocation and improved management practices, ultimately benefiting the entire student body and staff.

In conclusion, the Access Control System stands as a testament to human ingenuity. It represents a paradigm shift in dormitory management, ushering in an era of unprecedented convenience, and overall well-being for students, staff, and the entire institute.

# Tartalomjegyzék

1.	Felhasználói dokumentáció .....	9
1.1.	Rendszerkövetelmények.....	9
1.2.	Jogosultsági szintek .....	9
1.2.1.	Admin .....	9
1.2.2.	Kollégium vezető.....	9
1.2.3.	Nevelő.....	9
1.2.4.	Diák, Technikai dolgozó .....	9
1.3.	Felhasználói felület.....	10
1.3.1.	Bejelentkezés .....	10
1.3.2.	Kezdőlap.....	10
1.3.3.	Diákok .....	11
1.3.3.1.	Diákok megtekintése .....	11
1.3.3.2.	Új diák felvétele .....	12
1.3.3.3.	Diákok szerkesztése.....	12
1.3.3.4.	Diákok törlése.....	13
1.3.3.5.	Új feljegyzés készítése .....	13
1.3.4.	Dolgozók .....	14
1.3.4.1.	Dolgozók megtekintése .....	14
1.3.4.2.	Új dolgozó felvétele .....	14
1.3.4.3.	Dolgozó szerkesztése.....	15
1.3.4.4.	Dolgozó törlése.....	15
1.3.5.	Beállítások .....	16
1.3.6.	Belépő felület.....	16
2.	Fejlesztői dokumentáció .....	17

2.1.	Adatbázis .....	17
2.1.1.	Az adatbázis felépítése .....	18
2.1.2.	Az adatbázisban található táblák és azoknak jellemzése.....	18
2.2.	Felhasznált programozási nyelvek és technológiák.....	25
2.2.1.	JavaScript .....	25
2.2.2.	Node.js.....	25
2.2.3.	Next.js.....	26
2.2.4.	C# és a .NET 6.....	27
2.2.5.	Entity Framework Core .....	27
2.2.6.	ASP .NET Core Web API .....	28
2.3.	Fejlesztői környezetek és eszközök.....	29
2.3.1.	WebStorm .....	29
2.3.2.	Visual Studio 2022 .....	29
2.3.3.	Rider .....	30
2.3.4.	Postman .....	30
2.3.5.	SQL Server Management Studio 22 .....	30
2.3.6.	SQL Server Profiler 19 .....	31
2.3.7.	DataGrip .....	31
2.3.8.	GitKraken .....	31
2.4.	Tesztelési jegyzőkönyv.....	32
2.4.1.	Lighthouse .....	32
2.4.2.	MSTest.....	32
3.	Irodalomjegyzék .....	34
4.	Ábrajegyzék.....	35
5.	Mellékletek .....	36

5.1.	GitHub Repository.....	36
------	------------------------	----

## Bevezetés

Az elmúlt öt évben kollégistákként megtapasztaltuk, hogy a kollégiumi élet nem mindig egyszerű. A diákok és a nevelők egyaránt szembesülnek kihívásokkal. Számunkra az egyik legnagyobb kihívás a kollégiumból való érkezés és távozás jelentette. Kollégiumi nevelőinkkel való beszélgetésből kiderült, hogy a diákok jelenlétének nyilvántartása számukra nem kis kihívást jelent. Legyen ez a kimenők, illetve edzések nyilvántartása, vagy az általános kimenő végén a kollégiumba visszaért diákok feltárása, illetve nyilvántartása. Ezért úgy döntöttünk, hogy szakdolgozatunk keretében egy olyan weboldalt fejlesztünk ki, amely megkönnyíti a ki- és bejutást a kollégiumba, és javítja a kommunikációt a diákok és a nevelők között.

Szoftverünk célja, hogy automatizálja a beléptetési folyamatot, és valós idejű információt nyújtson arról, hogy ki tartózkodik a kollégiumban. Ez a megoldás nemcsak megkönnyíti a nevelők munkáját, de a diákok számára is kényelmesebb bejutást biztosít. A weboldal segítségével elkerülhetők a félreértések, és javul a kollégium biztonsága.

Célunk, hogy szoftverünk a kollégiumi élet minden résztvevője számára előnyös legyen, és hozzájáruljon a hatékonyabb és zökkenőmentesebb működéshez.



# **1. Felhasználói dokumentáció**

## **1.1. Rendszerkövetelmények**

Projektünk egy webes felület, ebből kifolyólag olyan eszközre van szükség, ami képes kapcsolódni az internethez. Felületünk reszponzívra fejlesztettük ki, így az eszköz lehet asztali számítógép, laptop, táblagép vagy mobil telefon. Az internet elérés mellett a következőkben felsorolt böngészők egyikével kell rendelkeznie: Microsoft Edge, Firefox, Google Chrome, Brave, Safari.

## **1.2. Jogosultsági szintek**

### **1.2.1. Admin**

Az Admin bejelentkezés után látja az összes felhasználót. Joga van dolgozókat felvenni, szerkeszteni és törölni. Emellett a diákokat is tudja kezelni. Az admin rendelkezik belépőkártyával az épülethez. Ennek a segítségével beléphet az épületbe.

### **1.2.2. Kollégium vezető**

A kollégiumvezető a bejelentkezést követően jogosult látni a dolgozókat, illetve elérhetőségeiket. Tud diákot és felvenni, szerkeszteni és törölni. Kezeli tudja a szülő adatait is. Emelet feljegyzéseket tud rögzíteni a diákokról. A kollégium vezető kártyája segítségével mozoghat ki és be az épületből.

### **1.2.3. Nevelő**

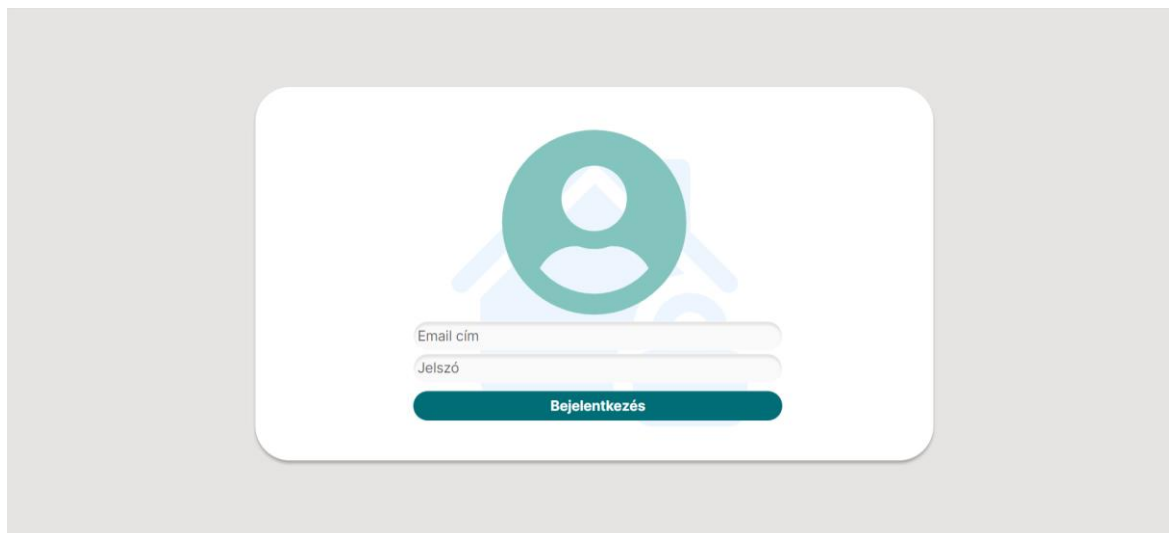
A bejelentkezés után a nevelőnek joga van megtekinteni a diákokat emellett látja a szülő adatait is. Tud feljegyzéseket készíteni. Emellett látja a többi dolgozót és az elérhetőségeiket. Belépőkártyájuk segítségével mozoghat be és ki az intézetből.

### **1.2.4. Diák, Technikai dolgozó**

A diák, technikai dolgozó belépőkártyája segítségével képes be- és kilépni az épületbe. Ezekről a mozgásokról feljegyzés készül.

## 1.3. Felhasználói felület

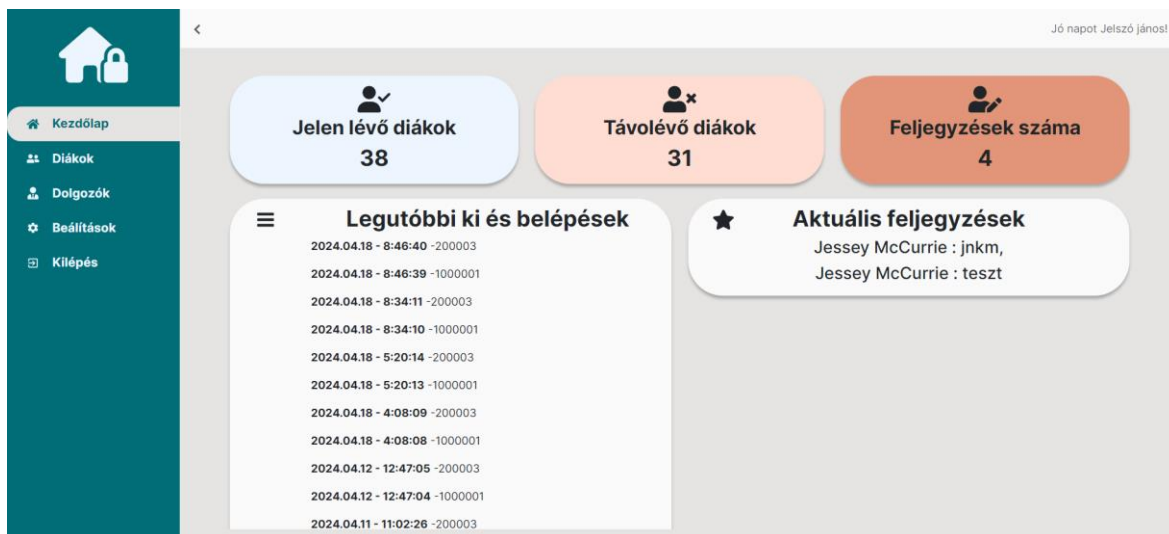
### 1.3.1. Bejelentkezés



1. ábra Bejelentkező oldal

A bejelentkezési joggal rendelkező felhasználók (admin, kollégium vezető, nevelő) itt tudnak belépni regisztrált email címük, illetve jelszavuk segítségével. Sikeres bejelentkezést követően az oldal a kezdőlapra dob át. Ellenkező esetben a felhasználó hiba üzenetet kap.

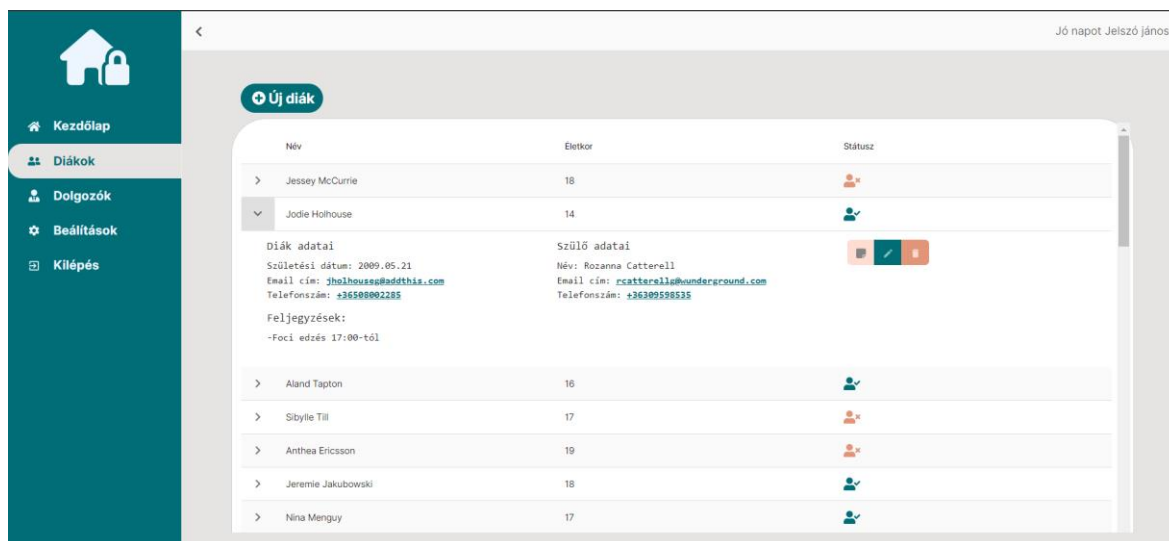
### 1.3.2. Kezdőlap



2. ábra Kezdőlap

A kezdő lapon a felhasználó az első sorban; látja hány diák van jelen az épületbe, mennyi diák van távol, illetve a feljegyzések számát. A következő sorban megtekintheti a legutóbbi tizenöt darab be- és kilépés pontos időpontját, illetve a személy kártyaszámát. Emellett kiírás kerülnek az épen aktuális feljegyzések.

### 1.3.3. Diákok



3. ábra Diákoklap

#### 1.3.3.1. Diákok megtekintése

Mindenki bejelentkező felhasználó számára láthatók a diákok összes adatukkal együtt. A táblázat adott sorának lenyitásakor megtekinthetjük a diák születési idejét, elérhetőségeit, illetve a megadott kapcsolat tartó (szülő) elérhetőségeit. Ezt azért tartottuk fontosnak, hogy vészhelyzet esetén egyszerűen lehessen előkeresni ezen adatokat. Minden diák neve mellett megtalálható egy ikon, ha ott egy kékeszöld pipa található akkor a diák jelen van amennyiben viszont egy narancssárga „x” abban az esetben a diák nem tartózkodik az épületben.

### 1.3.3.2. Új diák felvétele

The screenshot shows a web application interface with a sidebar on the left containing icons for home, users, settings, and a calendar. The main content area displays a table of students. A modal window titled 'Új diák hozzáadása' (Add New Student) is open, allowing for the entry of new student and parent information.

Új diák hozzáadása	
<b>Diák adatai</b>	<b>Szülő adatai</b>
Diák neve	Szülő neve
Diák email cím	Szülő email cím
Diák telefonszáma (+36301234567)	Szülő telefonszám (+36301234567)
éééé. hh. nn.	
<b>Mentés</b>	

4. ábra Új diák felvétele

Ha a felhasználó jogosút (admin, kollégium vezető) képes új diákot felvenni. Ehhez a táblázat felett található új diák felvétele gombra kell kattintani. Ezt követően a felugró ablakban ki kell tölteni mind a diák mind a szülő adatit. Meg kell adni a diák és a kapcsolattartó nevét, telefonszámát, email címét emellett a diák születési dátumát majd a mentés gombbal elmenteni az adatokat.

### 1.3.3.3. Diákok szerkesztése

The screenshot shows the same web application interface as before, but with the 'Diák Szerkesztése' (Edit Student) modal window open. This form allows for editing the details of an existing student and their parent. The background table shows a list of students, with 'Jessey McCurrie' selected.

Diák Szerkesztése	
<b>Diák adatai</b>	<b>Szülő adatai</b>
Jessey McCurrie	Cherey Scamaden
jmccurrie21@usnews.com	cscamadend@cam.ac.uk
+36103076664	+36403644786
<b>Mentés</b>	

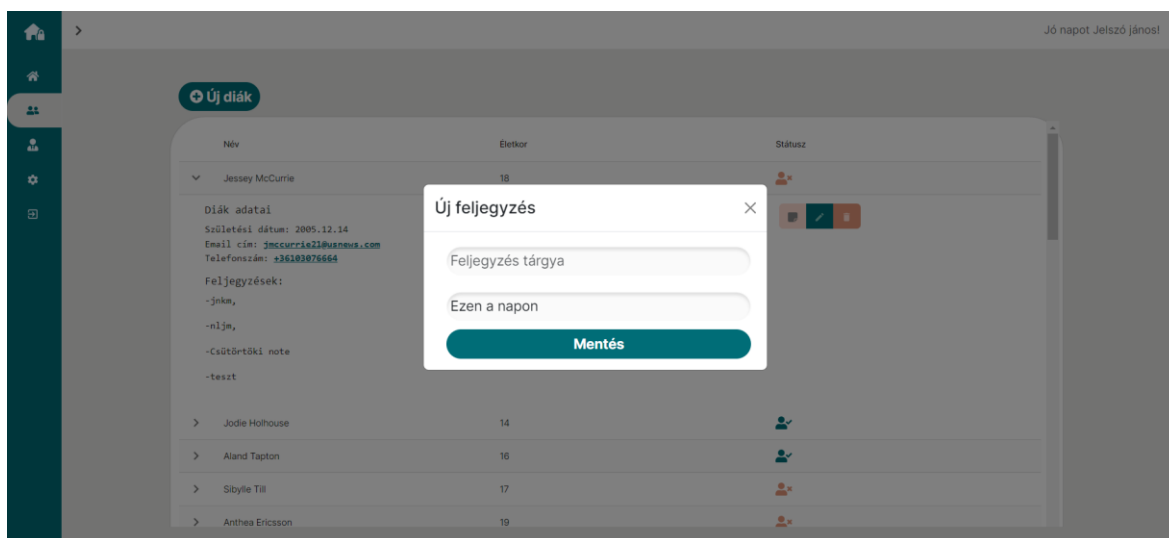
5. ábra Diák szerkesztése

Amennyiben a felhasználó jogosút (admin, kollégium vezető) képes diákat szerkeszteni. Szerkesztés során megtudja változtatni a diák, illetve a kapcsolatartója (szülő) nevét, valamint elérhetőségeit (telefonszám, email cím). A szükséges módosítások során a mentés gombbal tudja a felhasználó menteni a változtatásokat. Ezek a funkciók a táblázat adott sorának lenyitásával válnak láthatóvá.

#### 1.3.3.4. **Diákok törlése**

Amennyiben a felhasználó jogosút (admin, kollégium vezető) képes diákat törölni. A funkció az adott diák lenyitásával elérhető.

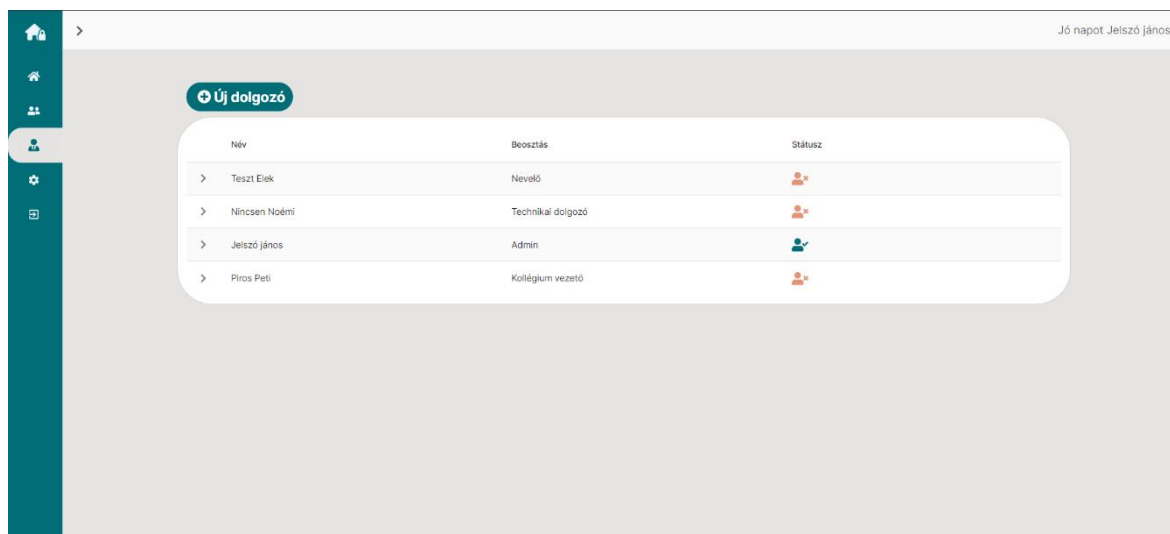
#### 1.3.3.5. **Új feljegyzés készítése**



6. ábra Új feljegyzés

A feljegyzések segítségével tudnak a nevelők egymásnak üzeni adott diákokkal kapcsolatban (pl.: Példa Péter 16:00-19:00-ig kosárlabda edzésen van). Ehhez minden belépési jogosultsággal rendelkező felhasználónak joga van. Az adott diák adatait lenyitva megtalálható ez a gomb melyre, ha a felhasználó rákattint feljön egy felugró ablak, ahol ki kell tölteni a feljegyzés leírását, mely napra szól, illetve mettől meddig.

## 1.3.4. Dolgozók



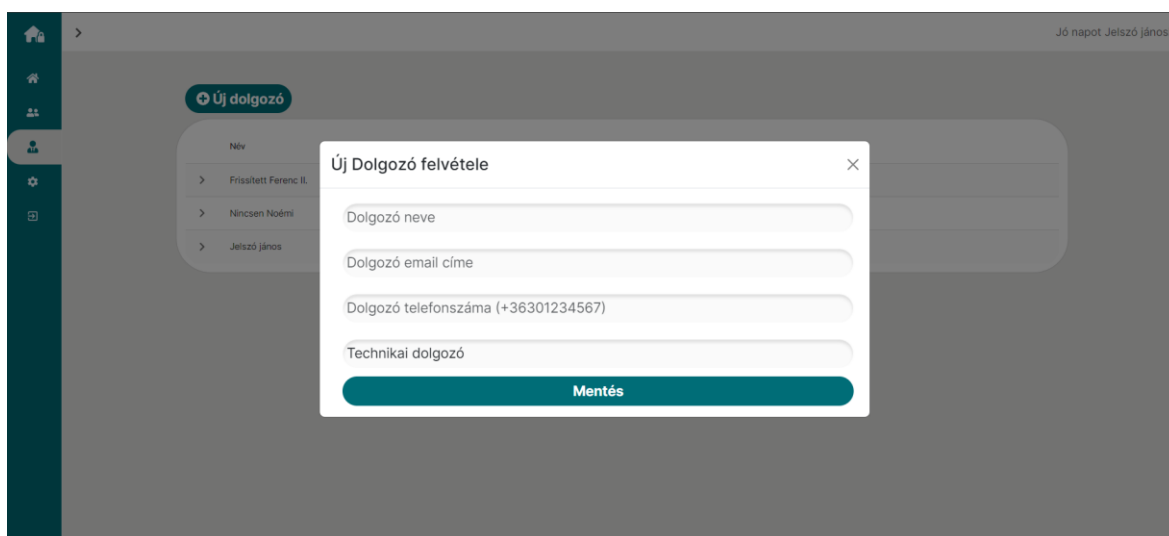
Név	Beosztás	Státusz
> Teszt Elek	Nevelő	
> Nincsen Noémi	Technikai dolgozó	
> Jelszó János	Admin	
> Piros Peti	Kollégium vezető	

7. ábra Dolgozók oldal

### 1.3.4.1. Dolgozók megtekintése

A dolgozók megtekintésére minden bejelentésre jogosult felhasználónak van joga. A dolgozó neve mellett még felvan tüntetve a beosztása, illetve látható a diákokhoz hasonlóan mely dolgozók érhetők (vannak jelen). A lenyíló fülel pedig láthatóvá válnak a dolgozó elérhetőségei.

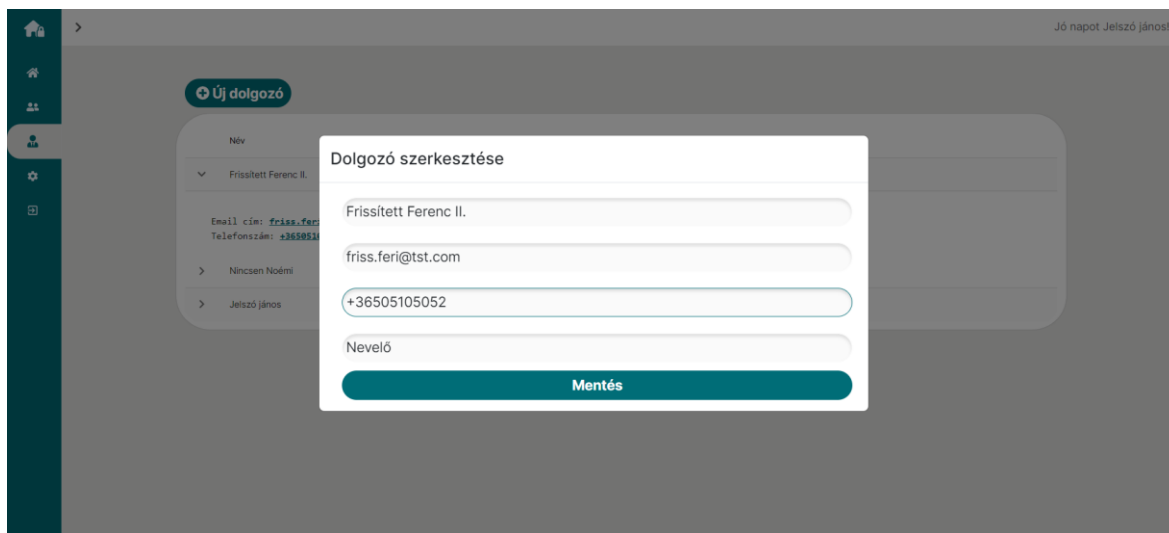
### 1.3.4.2. Új dolgozó felvétele



8. ábra Új dolgozó felvétele

Új dolgozó felvételére csak admin joggal rendelkező felhasználó képes. Meg kell adnia az új felhasználó nevét, elérhetőségeit; email cím telefonszám, illetve a legördülő listából ki kell választania mi az új dolgozó szerepköre (admin, kollégium vezető, nevelő, technikai dolgozó). Ezek a későbbiek módosíthatók például előléptetés esetén.

#### 1.3.4.3. **Dolgozó szerkesztése**



9. ábra Dolgozó szerkesztése

Az admin képes szerkeszteni a dolgozó adatait, többek között a jogosultságát is ami hasznos előléptetés esetén. ezt a legördülő lista segítségével tudja megtenni. Ezen kívül tudja szerkeszteni az elérhetőségeit is; email cím, telefonszám.

#### 1.3.4.4. **Dolgozó törlése**

Ha a felhasználó admin jogosultsággal rendelkezik joga van dolgozót törölni. Ezt a funkció a táblázat adott sorának lenyitásával lesz látható.

### 1.3.5. Beállítások

Jó napot Jelszó jános!

#### Adatok módosítása

Név  
Jelszó Jónás

Telefonszám  
+36703004001

Email cím  
jelszo@teszt.com

Új jelszó  
Új jelszó

Mentés

10. ábra Beállítások oldal

A beállításoknál az éppen bejelentkezett felhasználó tudja módosítani az adatait. Miután végzett a mentés gombbal tudja a módosításait menteni. Mentés nélkül az adatok nem változnak.

### 1.3.6. Belépő felület

2024.04.21 19:09:22

Tartsd a diákigazolványodat a leolvasó alá!

Belépőkód

Belépés/Kilépés

11. ábra Belépő felület

Valós belépőkód megadása után beenged, különben elutasítja a belépést.



## 2. Fejlesztői dokumentáció

### 2.1. Adatbázis

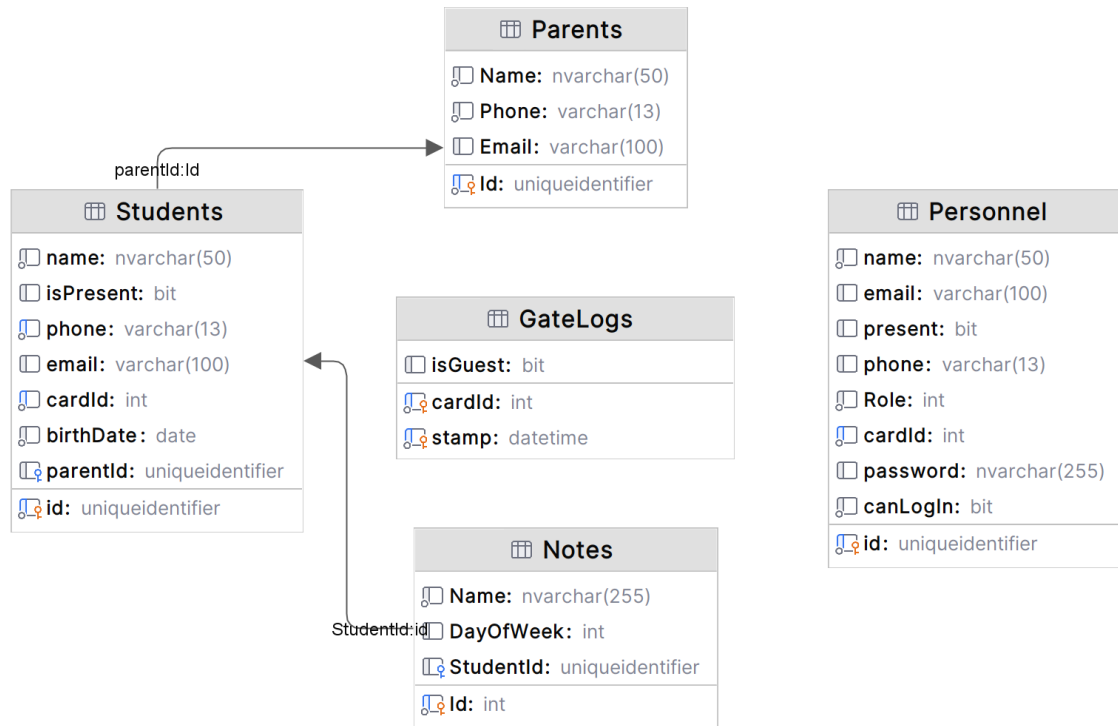
A Microsoft SQL Server 2022 Express az ingyenes, adatbázis-kezelő szoftver legújabb verziója. Ez a robusztus eszköz hatékony adatkezelést, fejlett lekérdezési funkciókat és átfogó biztonsági beállításokat biztosít, miközben könnyen telepíthető és használható.

#### Főbb Jellemzők:

- **Teljes funkcionalitás:** Az Express verzió az SQL Server teljes funkcionalitásának nagy részét magában foglalja, beleértve az adatbázis-létrehozást és -kezelést, a lekérdezéseket, a tárolt eljárásokat, a triggereket és a nézeteket.
- **Fejlett lekérdezési teljesítmény:** Az intelligens lekérdezésfeldolgozás, a lekérdezéstár és a párhuzamos lekérdezésvégrehajtás révén az SQL Server 2022 Express gyorsabb adatlekérdezést és analitikai műveleteket biztosít.
- **Robusztus biztonság:** Az SQL Server 2022 Express a legújabb biztonsági funkciókkal van felszerelve, beleértve a dinamikus adattitkosítást, az Always Encrypted funkciót és a beépített szerepköralapú hozzáférés-vezérlést (RBAC), így megóvjaa adatait a jogosulatlan hozzáféréstől.
- **Egyszerű telepítés és használat:** Az Express verzió telepítése és konfigurálása rendkívül egyszerű, intuitív grafikus felületének köszönhetően. Számos oktatóanyag és online forrás áll rendelkezésre a kezdők számára is.
- **Skálázhatóság:** Az SQL Server 2022 Express skálázható az egyéni munkaállomásoktól a kiszolgálókig, így a vállalkozás növekedésével együtt bővíthető.

### 2.1.1. Az adatbázis felépítése

Az alábbi ábrán látható az adatbázis felépítése:



12. ábra Adatbázis felépítése

### 2.1.2. Az adatbázisban található táblák és azoknak jellemzése

Az adatbázis táblái:

- Students
- Parents
- Notes
- Personnel
- GateLogs

- **Students tábla**
- **Id**
  - Uniqueidentifier
  - Primary key (elsődleges kulcs)
  - Az adott diák azonosításáért felelős
- **Name**
  - Nvarchar típusú, speciális karakterek eltárolására is alkalmas mező
  - Hossza legfeljebb ötven karakter hosszúságú lehet
  - Az adott diáknak a teljes nevét tárolja el
  - Nem vehet fel null értéket
- **IsPresent**
  - Bit típusú
  - Két értéket vehet fel egy, vagy nulla
  - A diák jelenlétének számontartására szolgál
  - Amennyiben a rekord létrehozásakor nem adunk meg explicit értéket ennek a mezőnek, akkor nullás értéket vesz fel, ezzel jelezve azt, hogy az adott diák jelenleg nagy valószínűséggel nem tartózkodik a kollégium területén.
- **Phone**
  - varchar típusú, szöveget tároló mező
  - A telefonszám legfeljebb tizenhárom karakter hosszú lehet
  - Ennek a mezőnek nem lehet *null* az értéke
  - Ebben az oszlopban minden sornak egyedi értékűnek kell lennie
  - Ez a mező tárolja el a diák telefonszámát
- **Email**
  - varchar típusú szöveg tárolására alkalmas mező
  - Ez a mező nem lehet *null* értékű
  - A tanuló e-mail címe legfeljebb száz karakter hosszú lehet

- 
- CardId
  - int típusú, egész szám tárolására alkalmas mező
  - Ezen oszlopban szintúgy, mint sok, ezt megelőzőben sem szerepelhet *null* érték
  - Ebben az oszlopban *is* minden értéknek egyedinek kell lennie
  - Ez a mező felelős egy diák azonosításáért beléptetés szempontjából
- BirthDate
  - date típusú, mező
  - Ennek az oszlopnak szintúgy nem lehet null az értéke
  - Ezen oszlop az adott diák születési dátumát hivatott eltárolni
- ParentId
  - uniqueidentifier típusú mező
  - Az adott diák szülőjének azonosítóját tárolja

#### **Parents tábla**

- Id
  - uniqueidentifier típusú
  - Egy adott szülő azonosítására szolgál
  - Ez a mező a Parents táblának az elsődleges kulcsa
  - Nem vehet fel *null* értéket
- Name
  - nvarchar típusú, speciális karakterek eltárolására is alkalmas mező
  - Hossza legfeljebb ötven karakter hosszúságú lehet
  - Az adott törvényes képviselőnek a teljes nevét tárolja el
  - Nem vehet fel null értéket
- Phone
  - varchar típusú, szöveget tároló mező
  - A telefonszám legfeljebb tizenhárom karakter hosszú lehet
  - Ennek a mezőnek nem lehet *null* az értéke
- Ez a mező tárolja el a törvényes képviselő telefonszámátEmail
  - varchar típusú szöveg tárolására alkalmas mező

- Ez a mező nem lehet *null* értékű
- A törvényes képviselő e-mail címe legfeljebb száz karakter hosszú lehet

### Notes tábla

- Id
  - int típusú
  - Elsődleges kulcs
  - Automatikusan növekedik, a sorok számával együtt
- Name
  - nvarchar típusú speciális karakterek tárolására is alkalmas mező
  - A feljegyzés szövegének tárolására alkalmas mező
  - Legfeljebb kétszázötvenöt karaktert tud eltárolni
  - Kötelezően megadandó mező
- DayOfWeek
  - int értékű, kötelezően megadandó mező
  - A hétnek azon napját tárolja, amelyre a feljegyzés vonatkozik
    - Hétfő = 1
    - Kedd = 2
    - Szerda = 3
    - Csütörtök = 4
    - Péntek = 5
- StudentId
  - uniqueidentifier típusú
  - Idegen kulcs
  - A feljegyzéshez kapcsolódó diák azonosítója
  - Nem vehet fel *null* értéket

## Parents tábla

- Id
  - Uniqueidentifier
  - Primary key (elsődleges kulcs)
  - Az adott kollégiumi alkalmazott azonosításáért felelős
- Name
  - nvarchar típusú, speciális karakterek eltárolására is alkalmas mező
  - Hossza legfeljebb ötven karakter hosszúságú lehet
  - Az adott kollégiumi alkalmazott a teljes nevét tárolja el
  - Nem vehet fel null értéket
- IsPresent
  - Bit típusú
  - Két értéket vehet fel egy, vagy nulla
  - A kollégium alkalmazott jelenlétének számontartására szolgál
  - Amennyiben a rekord létrehozásakor nem adunk meg explicit értéket ennek a mezőnek, akkor nullás értéket vesz fel, ezzel jelezve azt, hogy az adott személy jelenleg nagy valószínűséggel nem tartózkodik a kollégium területén.
- Phone
  - varchar típusú, szöveget tároló mező
  - A telefonszám legfeljebb tizenhárom karakter hosszú lehet
  - Ennek a mezőnek nem lehet *null* az értéke
  - Ebben az oszlopban minden sornak egyedi értékűnek kell lennie
  - Ez a mező tárolja el a kollégiumi alkalmazott telefonszámát
- Email
  - varchar típusú szöveg tárolására alkalmas mező
  - Ez a mező nem lehet *null* értékű
  - A kollégiumi alkalmazott e-mail címe legfeljebb száz karakter hosszú lehet

- CardId
  - int típusú, egész szám tárolására alkalmas mező
  - Ezen oszlopban szintúgy, mint sok ezt megelőzőben sem szerepelhet *null* érték
  - Ebben az oszlopban *is* minden értéknek egyedinek kell lennie
  - Ez a mező felelős egy kollégiumi alkalmazott azonosításáért beléptetés szempontjából
- canLogin
  - bit típusú mező
  - Arra szolgál, hogy egy adott kollégiumi alkalmazott beléphet-e a nevelői kezelő felületbe. Például, egy nevelőnek nincs hozzáférési joga
  - Ha az értéke egy, akkor az adott személy beléphet a rendszerbe, ha nulla, akkor nem léphet be
- Role
  - int típusú mező
  - nem vehet fel *null* értéket
  - Ez jelöli, hogy az adott felhasználó/alkalmazott milyen szerepet tölt be
    - 1 = Adminisztrátor
    - 2 = Kollégiumvezető
    - 3 = Kollégiumi nevelő
    - 4 = Technikai dolgozó
  - Alapértelmezett értéke négy
- Password
  - nvarchar típusú, speciális karakterek eltárolására is alkalmas mező
  - Legfeljebb 255 karakter hosszú lehet
  - Ez tárolja el a felhasználók titkosított jelszavát (már akinek van)

- **GateLogsTábla**
- CardId
  - int típusú mező
  - Nem vehet fel *null* értéket
  - Az be/ki jelentkező személy kártyaszámát tárolja
  - Ez a tábla kompozit elsődleges kulcsának egyik tagja
- Stamp
  - datetime típusú mező
  - A mozgás időpontját tároja el
  - Nem vehet fel *null* értéket
  - Ez a tábla kompozit elsődleges kulcsának másik tagja
- IsGuest
  - bit típusú jelenleg nincs használva a programban



## 2.2. Felhasznált programozási nyelvek és technológiák

### 2.2.1. JavaScript



A JavaScript egy rendkívül sokoldalú és népszerű programozási nyelv, mely eredetileg a weboldalak interaktivitásának növelésére lett tervezve. Azonban a lehetőségei jóval szélesebbek ennél: alkalmas webes alkalmazások fejlesztésére, szerveroldali programozásra, játékfejlesztésre, adatvizualizációra és még számos más területre is.

13. ábra JavaScript  
logó

Az egyik fő előnye, hogy könnyen tanulható, így ideális választás mind a kezdő, mind a tapasztalt fejlesztők számára. Az aktív közösség és a széles körben elérhető források tovább segítik a fejlődést és a problémamegoldást.

Mivel ingyenes és nyílt forráskódú, számos eszköz és keretrendszer áll rendelkezésre JavaScript fejlesztéshez, így nagy szabadságot biztosít a fejlesztőknek az alkalmazások kialakításában és működtetésében.

Összességében a JavaScript egy rendkívül értékes eszköz a modern fejlesztők számára, amely számos lehetőséget kínál a kreativitás kibontakoztatására és a hatékony alkalmazás fejlesztésre.

### 2.2.2. Node.js



14. ábra Node.js logó

Node.js egy olyan szerveroldali JavaScript környezet, amely lehetővé teszi JavaScript alkalmazások futtatását a szerver oldalán. Ez a technológia forradalmasította a szoftverfejlesztést, mivel lehetővé teszi a fejlesztők számára, hogy ugyanazt a nyelvet használják mind a kliensoldali, mind a szerveroldali kód írásához.

Node.js alapján könnyen készíthetünk skálázható, hatékony és gyors alkalmazásokat. Az aszinkron működés lehetővé teszi a hatékony erőforrás-kihasználást, és lehetővé teszi a nagy mennyiségű kérés kezelését anélkül, hogy blokkolná az alkalmazást.

A Node.js nagy előnye, hogy egy nagyon aktív közösség veszi körül, ami folyamatos fejlesztést és támogatást biztosít az új és meglévő projektekhez. Emellett a széles körben elérhető modulok és csomagok gazdag ökoszisztémát képeznek, amelyek segítségével gyorsan és hatékonyan építhetünk komplex alkalmazásokat.

A Node.js lehetővé teszi a fejlesztők számára, hogy teljes mértékben kihasználják a JavaScript nyelv erejét mind a kliens, mind a szerver oldalán, és így lehetővé teszi a modern, gyors és skálázható webalkalmazások készítését.

### 2.2.3. Next.js



15. ábra Next.js  
logó

A Next.js egy elismert keretrendszer a React-hez, amely megkönnyíti a webes alkalmazások fejlesztését. A Next.js szolgáltatásai közé tartozik a szerveroldali renderelés (SSR), statikus weboldal generálás (SSG) és további funkciók, amelyek dinamikus és interaktív weboldalak létrehozását teszik lehetővé. Ezek az oldalak gyorsak, hatékonyak és jól optimalizáltak a keresőmotorok számára.

A főbb jellemzők között említhető a szerveroldali renderelés (SSR), ami gyorsabb betöltési időket és jobb SEO-t eredményez. A statikus weboldal generálás (SSG) lehetővé teszi előre generált, gyorsan betöltődő oldalak készítését. A beépített útvonalkezelés könnyűvé teszi az URL-ek és komponensek kezelését, míg az API-k lehetővé teszik az adatok küldését és fogadását a szerver és a kliens között.

A Next.js a React komponensek támogatásával lehetővé teszi könnyen újrahasználató UI elemek építését. Az aktív közösség folyamatos fejlesztést és támogatást biztosít a fejlesztőknek.

A Next.js előnyei között szerepelnek a gyorsabb betöltési idők és a jobb SEO, valamint a könnyű fejlesztés és a skálázhatóság. Használata ideális dinamikus weboldalak, webes alkalmazások és statikus weboldalak létrehozásához egyaránt.

## 2.2.4. C# és a .NET 6



A C# egy rendkívül sokoldalú, objektumorientált programozási nyelv, amelyet a Microsoft fejlesztett ki. Széles körben alkalmazzák a szoftverfejlesztés területén, beleértve webes, asztali és mobil alkalmazásokat, valamint játékokat is. A C# könnyen tanulható és használható, ugyanakkor hatékony és megbízható is.

### 16. ábra .NET logó

A .NET 6 pedig egy olyan keretrendszer, amelyet szintén a Microsoft fejlesztett ki, és a C# nyelvvel együttműködve biztosít platformfüggetlen fejlesztési környezetet. A .NET 6 számos előnnyel jár a fejlesztők számára, mint például a platformfüggetlenség, a nagy teljesítmény, a gazdag ökoszisztéma és a modern funkciók.

A C# és a .NET 6 kombinációja ideális választás a modern szoftverfejlesztők számára, akik olyan alkalmazásokat szeretnének létrehozni, amelyek bármilyen operációs rendszeren futtathatók, és magas teljesítményt nyújtanak.

A C# és a .NET 6 használati esetei szerteágazóak, beleértve a webes alkalmazásokat, asztali alkalmazásokat, mobil alkalmazásokat, játékokat és felhőalapú alkalmazásokat is. Ezek a technológiák lehetővé teszik a fejlesztők számára, hogy különféle platformokon működő, funkció gazdag és nagy teljesítményű alkalmazásokat hozzanak létre.

## 2.2.5. Entity Framework Core

Az EntityFramework Core (EF Core) egy nyílt forráskódú objektumrelációs leképezési (ORM) keretrendszer, amely a .NET platformhoz készült. Ez a technológia lehetővé teszi a fejlesztők számára, hogy könnyen összekapcsolják objektumorientált kódjukat az adatbázisokkal. Az EF Core segítségével gyorsan és hatékonyan kezelhetik az adatbázis-lekérdezéseket, frissítéseket, beszúrásokat és törléseket anélkül, hogy kézzel kellene SQL-parancsokat írniuk.

A főbb jellemzők között említhető az objektumorientált megközelítés, amely megkönnyíti a fejlesztők számára a munkát és csökkenti a kódolási hibák kockázatát. Az automatikus leképezés lehetővé teszi az objektumok és adatbázis-táblák közötti automatikus átjárást, minimalizálva ezzel a fejlesztői munkát.

Az EF Core széles körű adatbázistámogatást biztosít, beleértve a Microsoft SQL Server-t, MySQL-t, PostgreSQL-t, Oracle-t és SQLite-t is. Emellett a keretrendszer kiterjeszthető, így testreszabható az egyedi igényeknek megfelelően.

Az EF Core használatának előnyei között kiemelhető a növelt termelékenység, kevesebb kód írása és a könnyen karbantartható, objektumorientált megközelítés. A széleskörű támogatásnak köszönhetően a fejlesztők rengeteg forrást és segítséget találnak a tanuláshoz és a fejlesztéshez.

Az EF Core használati esetei sokszínűek, magukban foglalva webes alkalmazásokat, asztali alkalmazásokat, mobil alkalmazásokat és mikroszolgáltatásokat is. Bármilyen alkalmazást fejlesztenek is, amely adatbázisokat használ, az EF Core biztosítja a hatékony és modern megoldást az adatok kezeléséhez.

## **2.2.6. ASP .NET Core Web API**

Az ASP.NET Core Web API a Microsoft által kifejlesztett, nyílt forráskódú keretrendszer, mely modern, RESTful API-k létrehozását teszi lehetővé a .NET platformon. Könnyű súlyú és moduláris felépítésének köszönhetően gyors és hatékony API-kat hozhatunk létre vele, míg RESTful támogatása segítségével az általánosan elfogadott szabványoknak megfelelően tervezhetjük meg az API-kat.

Az ASP.NET Core Web API bővíthető, lehetőséget biztosítva a testre szabásra middleware-ek és szolgáltatások segítségével. Emellett multiplatform jellege lehetővé teszi az alkalmazások futtatását különböző operációs rendszereken, mint például Windows, macOS és Linux.

A gyors fejlesztés és könnyű tanulhatóság mellett az ASP.NET Core Web API tiszta architektúrát támogat, ami jól karbantartható és tesztelhető API-kat eredményez. Nagy teljesítménye és hatékonysága révén alkalmas nagy forgalmú alkalmazások fejlesztésére is. Az ASP.NET Core Web API széleskörű ökoszisztémája pedig gazdag választékot kínál könyvtárak, eszközök és keretrendszerek tekintetében.

A gyakorlati alkalmazások terén az ASP.NET Core Web API ideális választás webes, mobil és IoT alkalmazásokhoz egyaránt. Emellett kiválóan alkalmazható mikró szolgáltatások fejlesztésére is, amelyek API-kat használnak a kommunikációhoz. Ez a keretrendszer széles körű alkalmazásokat támogat, és segítségével könnyen létrehozhatunk modern, hatékony és skálázható API-kat.

## 2.3. Fejlesztői környezetek és eszközök

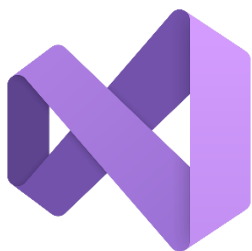
### 2.3.1. WebStorm



16. ábra WebStorm  
logó

A WebStorm egy hatékony IDE a webes fejlesztéshez, mely a JetBrains által kifejlesztett népszerű eszköz. JavaScript, HTML, CSS, TypeScript, Node.js és más technológiák támogatásával rendelkezik. Intelligens kódolási segédeszközökkel, beépített hibakereséssel és profilkészítéssel, valamint verziókövetés integrációval rendelkezik. Ezáltal növeli a termelékenységet, csökkenti a hibákat, segíti a csapatmunkát, lehetővé teszi a rugalmas fejlesztést és testre szabhatóságot biztosít.

### 2.3.2. Visual Studio 2022



17. ábra Visual  
Studio 2022 logó

A Visual Studio 2022 Community Edition egy ingyenes, teljes körű integrált fejlesztői környezet (IDE) Windows platformra. Kiváló eszköz a webes, asztali, mobil és játékfejlesztéshez, támogatva számos népszerű nyelvet, mint például C#, F#, Visual Basic .NET és C++. Beépített hibakereső és profilkészítő eszközökkel, valamint verziókövetés integrációval rendelkezik, ami növeli a termelékenységet és a kód minőségét. A Visual Studio felhasználóbarát felülettel rendelkezik, így könnyen tanulható és használható, illetve számos bővíthető plugin révén testre szabható az egyéni igényeknek

megfelelően. Ideális választás a fejlesztők számára, akik széleskörű alkalmazásokat szeretnének fejleszteni egy erőteljes és sokoldalú fejlesztői környezetben.

### 2.3.3. Rider



18. ábra Rider logó

A JetBrains Rider egy cross-platform .NET IDE, melyet az IntelliJ IDEA alkotói fejlesztenek. Kiváló kódminőség-ellenőrzéssel, gyors fejlesztéssel és modern .NET támogatással rendelkezik, valamint hibakeresés mesterfokon, integrált unit teszteléssel és NuGet csomagkezeléssel teszi hatékonyabbá a fejlesztést. Ajánlott .NET, C# és Xamarin fejlesztőknek, valamint Unity motorral dolgozó játékfejlesztőknek. A Rider felhőszolgáltatások integrációjával és verziókövetési támogatással is rendelkezik, ami zökkenőmentessé teszi a fejlesztési folyamatot. Rugalmas és teljes körű IDE, mely hatékonyabbá teszi a .NET projektek fejlesztését és karbantartását.

### 2.3.4. Postman



19. ábra Postman  
logó

A Postman egy erőteljes API fejlesztői platform, melynek fő jellemzői közé tartozik az API tervezés és tesztelés, gyűjtemények és mappák szervezése, valamint a közös munka lehetősége csapatársakkal. Automatizált dokumentáció generálása és API monitorozási funkciói segítik a hatékony fejlesztést és karbantartást. A könnyen használható felület és ingyenes alapfunkciók teszik népszerűvé mind kezdők, mind tapasztalt fejlesztők körében.

### 2.3.5. SQL Server Management Studio 22

Az SQL Server Management Studio 2022 egy hatékony eszköz a Microsoft SQL Server adatbázisok hatékony kezeléséhez. Széleskörű funkciókat kínál az adatbázis-adminisztrátorok és fejlesztők számára az adatbázisok létrehozásához, kezeléséhez, lekérdezéséhez és optimalizálásához. Az SSMS 22 különösen hasznos a Microsoft SQL Server legújabb kiadásával és az Azure SQL Managed Instance szolgáltatással dolgozók számára.

### 2.3.6. SQL Server Profiler 19

Az SQL Server Profiler 19 kulcsfontosságú eszköz az SQL Server 2019 adatbázisok figyelésére és nyomon követésére. Segítségével könnyen azonosíthatók hibák, optimalizálhatók a lekérdezések, és felügyelhető a biztonság.

Az eszköz lehetővé teszi az adatbázis változásainak nyomon követését és az adatmigráció felügyeletét is, ami fontos biztonsági és teljesítménybeli szempontból egyaránt.

### 2.3.7. DataGrip



20. ábra DataGrip  
logó

A DataGrip egy fejlett, ám könnyen használható adatbázis-eszköz, mely segíti a fejlesztőket az adatbázisok kezelésében és optimalizálásában. Az intelligens kódkiegészítés és a vizuális eszközök segítségével gyorsan és hatékonyan írhatunk és szerkeszthetünk SQL lekérdezéseket. Emellett a verziókövetés integrációval könnyen nyomon követhetjük az adatbázis-sémában történő változásokat. Összességében a DataGrip egy olyan eszköz, mely megkönnyíti az adatbázisokkal kapcsolatos munkafolyamatokat, és lehetővé teszi a gyors és megbízható adatbázis-fejlesztést.

### 2.3.8. GitKraken



21. ábra GitKraken  
logó

A GitKraken a korszerű, vizuális Git kliens, amely kiemeli magát a Git-projektek menedzselésének terén. A rendkívül intuitív felülettel és hatékony funkciókkal a GitKraken segít a projektvezetőknek és fejlesztőcsapatoknak hatékonyabban együttműködni és produktívan dolgozni. Az áttekinthető grafikus felület lehetővé teszi a repo-k könnyű vizualizációját, míg az ágak könnyű kezelése és a beépített hibakeresési eszközök fokozzák a munkavégzés hatékonyságát és minőségét. A GitKraken nem csupán a kezdőknek kínál ideális megoldást, hanem a haladó felhasználóknak is egy teljes körű Git élményt biztosít, mely bővíthető további személyre szabott funkciókkal.

## 2.4. Tesztelési jegyzőkönyv

### 2.4.1. Lighthouse

A Lighthouse, mint ingyenes és nyílt forráskódú weboldal-elemző eszköz, a Google által kifejlesztett szoftver, mely rendkívül alapos áttekintést nyújt a webhelyek teljesítményéről és hozzáférhetőségéről, valamint a keresőoptimalizálásról és a progresszív webalkalmazásokról (PWA). Az általa nyújtott jelentések alapján a webfejlesztők és -tulajdonosok képesek azonosítani a lehetséges problémákat és javítási lehetőségeket a weboldalakon. A Lighthouse könnyen használható a Chrome böngésző fejlesztői eszközeinek részeként, így mindenki számára hozzáférhető és használható.

A tesztet a frontend összes oldalát lefuttattuk. Ezeknek az eredménye a lenti táblázatban található. Az SEO és PWA eredményeket nem vettük figyelembe.

Oldal	Teljesítmény	Hozzáférhetőség	Best Practices
Bejelentkezés	100	100	100
Kezdőlap	97	94	100
Diákok	97	89	100
Személyzet	99	89	100
Beállítások	100	95	100

### 2.4.2. MSTest

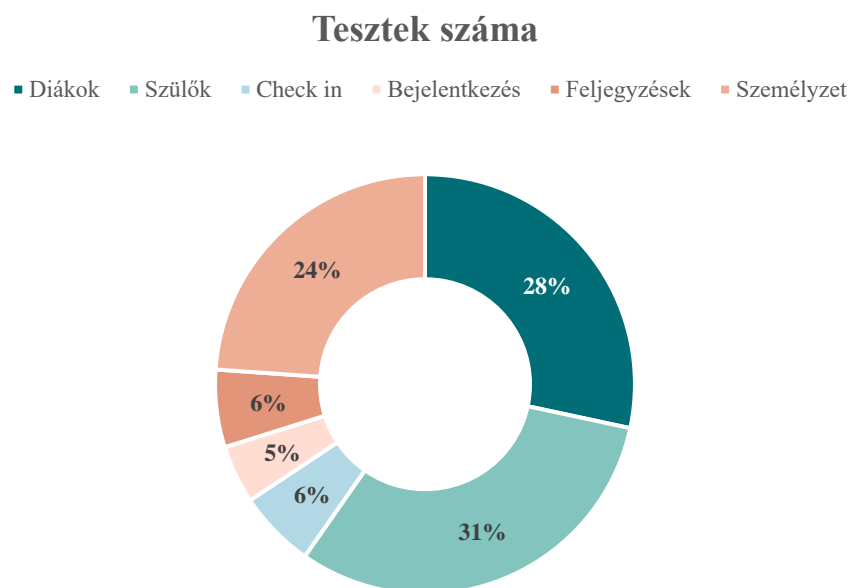
Az MSTest a Microsoft tesztelési keretrendszere, amely kifejezetten a .NET keretrendszerhez lett tervezve. Ez egy rendkívül bővíthető keretrendszer, amely segítségével egységteszteket írhat és futtathatsz .NET-alkalmazásokhoz. Az MSTest szorosan integrálva van a Visual Studio-val, így könnyen és hatékonyan hozhatsz létre, futtathatsz és hibakereshetsz teszteket a fejlesztési környezetben.



A főbb jellemzők közé tartozik az egységtesztek támogatása minden .NET alkalmazási szinten, az automatizált tesztelés lehetősége, valamint a bővíthetőség tesztbővítményekkel. Az MSTest beépített tesztasszerciókat és lehetőséget biztosít a saját asszerciók létrehozására, továbbá támogatja a tesztadatok beállítását és a tesztek csoportokban történő futtatását.

Emellett az MSTest lehetővé teszi a tesztek szűrését és a teszteredmények jelentését HTML, XML vagy egyéni formátumban. Összességében az MSTest egy hatékony és sokoldalú eszköz a .NET alkalmazások teszteléséhez és minőségének biztosításához.

MSTest segítségével több, mint hatvan unit teszt lett készítve, amelyek lefedik a backend legfőbb serviceit. Ezzel biztosítva, hogy azok úgy működnek, mint ahogyan azt elvártuk. A servicek lefedettsége így 81%, illetve a projekt lefedettsége 62% lett. A tesztek 100%-a sikeres eredménnyel tér vissza.



22. ábra Tesztek eloszlása service szerint

### 3. Irodalomjegyzék

Javascript logó: JavaScript Corp., [CC BY-SA 4.0](#), via Wikimedia Commons

Node.js logó: Ryan Dahl, [MIT](#), via Wikimedia Commons

Next.js logó: Vercel, [CC BY-SA 4.0](#), via Wikimedia Commons

.NET logó: Microsoft, CC0, via Wikimedia Commons

WebStorm logó: JetBrains, Public domain, via Wikimedia Commons

Visual Studio 2022 logó: Microsoft, Public domain, via Wikimedia Commons

Rider logó: JetBrains, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Datagrip logó: JetBrains, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

## 4. Ábrajegyzék

1. ábra Bejelentkező oldal .....	10
2. ábra Kezdőlap .....	10
3. ábra Diákoklap.....	11
4. ábra Új diák felvétele.....	12
5. ábra Diák szerkesztése.....	12
6. ábra Új feljegyzés .....	13
7. ábra Dolgozók oldal .....	14
8. ábra Új dolgozó felvétele.....	14
9. ábra Dolgozó szerkesztése.....	15
10. ábra Beállítások oldal .....	16
11. ábra Belépő felület.....	16
12. ábra Adatbázis felépítése .....	18
13. ábra JavaScript logó .....	25
14. ábra Node.js logó.....	25
15. ábra Next.js logó.....	26
16. ábra .NET logó .....	27
17. ábra Visual Studio 2022 logó.....	29
18. ábra Rider logó .....	30
19. ábra Postman logó .....	30
20. ábra DataGrip logó .....	31
21. ábra GitKraken logó .....	31
22. ábra Tesztek eloszlása service szerint.....	33

## 5. Mellékletek

### 5.1. GitHub Repository

<https://github.com/Szilviniki/AccessControlSystem>

