**Pannonhalmi Főapátság Szegedi SOB Technikuma**

A képen képernyőkép, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Parkolóház Kezelő Rendszer

Szoftverfejlesztő és -tesztelő

**Készítették:** Juhász Szabolcs, Scher János, Major Attila

**Témavezető:** Rédai Dávid

2025

Szeged

**Tartalom**

[A projekt áttekintése 3](#_heading=h.ble4zunxedc8)

[A projekt céljának részletes ismertetése 3](#_heading=h.6beqc97gm87q)

[Tervezési szempontok és alkalmazott módszerek 4](#_heading=h.4zrvp8omuftq)

[SWOT elemzés 7](#_heading=h.j55g5tufow7u)

[Feladatmegosztás a projektcsapaton belül 8](#_heading=h.j1tqq99uqn1k)

[Ütemterv – a projekt előrehaladásának menete 10](#_heading=h.7vgaqrqrd1ml)

[Felhasználói felület és alkalmazott technológiák 13](#_heading=h.543qnkr1817)

[Adatbázisok és eszközök részletes bemutatása 21](#_heading=h.y7ztrpdl0ctp)

[Minőségbiztosítás és tesztelési folyamatok 29](#_heading=h.4h4sti1fk7lw)

[Összegzés 36](#_heading=h.dgqt6dydanl8)

[Felhasznált források 38](#_heading=h.8shacx1g8jpo)

# A projekt áttekintése

A **Parkolóház Kezelő Rendszer** célja egy olyan korszerű, digitális szoftvermegoldás létrehozása volt, amely lehetővé teszi egy modern parkolóház hatékony, automatikus és felhasználóbarát üzemeltetését. A rendszer képes kezelni a felhasználók regisztrációját, a járművek nyilvántartását, a parkolási események rögzítését, valamint a parkolási díjak kiszámítását és megjelenítését. Emellett egy adminisztrációs felületet is biztosít az üzemeltetők számára, amelyen keresztül teljes körű menedzsment végezhető.

# A projekt céljának részletes ismertetése

A fejlesztés célja egy **komplex, modulárisan felépített parkolóház-kezelő rendszer** kialakítása volt, amely képes kiszolgálni különböző felhasználói igényeket, és hosszú távon is skálázható marad. A rendszer főbb funkciói közé tartozik:

* **Felhasználói fiókok kezelése:** regisztráció, bejelentkezés, jogosultságok kezelése.
* **Járműregisztráció:** a felhasználók saját járműveiket rögzíthetik a rendszerben.
* **Parkolási események nyilvántartása:** a parkolóba történő be- és kihajtások naplózása, időbélyeggel.
* **Díjszámítás:** a parkolási idő és a díjtételek alapján automatikus díjszámítás történik.

A rendszer többplatformos elérést biztosít: **webes és mobilfelületet** is kínál, így a felhasználók bármilyen eszközről kényelmesen hozzáférhetnek. A fejlesztés során kiemelt figyelmet fordítottunk az alábbi szempontokra:

* **Modularitás:** a rendszer könnyen bővíthető további funkciókkal.
* **Skálázhatóság:** alkalmas nagyobb parkolóházak kiszolgálására is.
* **Információbiztonság:** biztonságos jelszókezelés, szerepköralapú hozzáférés, adatok titkosítása.

# Tervezési szempontok és alkalmazott módszerek

A rendszer fejlesztését egy előre kidolgozott, **20 hetes fejlesztési terv** mentén végeztük, amely lefedte az igényfelméréstől kezdve a tesztelésen át a dokumentáció elkészítéséig minden fontosabb fázist. A projekt során különös figyelmet fordítottunk a korszerű technológiák és fejlesztési irányelvek alkalmazására, amelyek biztosítják a hosszú távú fenntarthatóságot és a további fejlesztések lehetőségét.

A backend oldal fejlesztéséhez a **.NET 8** platformot választottuk, amely megbízható alapot biztosít a REST API-k kialakításához. Az **Entity Framework Core** ORM használatával elkerültük a manuális SQL lekérdezések írását, miközben megőriztük a teljes kontrollt az adatbázis műveletek felett:

A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A rendszer aszinkron feldolgozást és strukturált hibakezelést alkalmaz, ezzel növelve a teljesítményt és stabilitást:

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A frontend technológiai alapját a **Svelte** képezi, amely a reaktív működésének köszönhetően gyors és letisztult felhasználói élményt nyújt. A keretrendszer használata lehetővé tette a dinamikus komponensek gyors fejlesztését és egyszerű karbantartását.

Az adatkezelés központi eleme egy **MySQL alapú adatbázis**, amelyet az **Aiven felhőszolgáltatás** segítségével hosztoltunk. Ez lehetővé tette a könnyű skálázást, valamint a megbízható és biztonságos elérést különböző környezetekből:

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.



A jelenlegi megvalósításban a felhasználói jelszavak **Base64 kódolással** kerülnek eltárolásra, amely elsősorban a karakterek átalakítására szolgál, de nem nyújt valódi védelmet. Ennek tudatában a jövőbeni fejlesztési terv részeként szerepel egy **kriptográfiailag biztonságos, egyirányú hash-elési algoritmus** – például a **bcrypt** – bevezetése, amely jelentősen növelné a rendszer adatbiztonságát, különösen a jelszavak kezelése terén:

A képen képernyőkép, szöveg, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A fejlesztés során alkalmazott tervezési alapelvek:

* **Szeparált rétegstruktúra:** a backend, frontend és adatbázis külön komponensként lett megtervezve, ami elősegíti a kód újra felhasználhatóságát és a hibakeresés egyszerűségét.
* **Verziókezelés:** a projekt teljes fejlesztése **Git** alapon zajlott, a GitHub-on való együttműködés mellett.
* **Kódolási szabványok:** betartottuk a **C# és JavaScript** nyelvi konvencióit, biztosítva az olvashatóságot és a karbantarthatóságot.
* **Tesztelhetőség:** a rendszer logikájának kialakítása során ügyeltünk az egységtesztelés lehetőségére, amely hosszú távon hozzájárul a minőségbiztosításhoz.

# SWOT elemzés

A projekt elemzése során **SWOT-analízist** alkalmaztunk, amely egy széles körben használt marketing és üzleti tervezési technika. Lehetővé teszi, hogy a rendszer **belső tényezőit** (erősségek, gyengeségek) és **külső környezeti hatásait** (lehetőségek, veszélyek) átlátható módon vizsgáljuk. Az elemzés az alábbi kategóriák mentén történt:

* **S – Strengths (Erősségek)**
* **W – Weaknesses (Gyengeségek)**
* **O – Opportunities (Lehetőségek)**
* **T – Threats (Veszélyek)**

A képen szöveg, képernyőkép, szám, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Feladatmegosztás a projektcsapaton belül

A projektet egy háromfős csapatban készítettük, és mindenki a saját szakterületének megfelelően járult hozzá a munka sikeréhez. Az alábbiakban összegzem, hogyan osztottuk el a feladatokat.

**Juhász Szabolcs – Frontend fejlesztés, Tesztelés, Csapatvezetés**

* Szabolcs volt felelős a felhasználói felület kialakításáért, hogy az könnyen használható és reszponzív legyen. A frontend fejlesztése mellett a tesztelésben is részt vett, hogy biztosítsa, hogy az alkalmazás minden platformon jól működjön. Ő irányította a csapatot is, hogy a projekt zökkenőmentesen haladjon.

**Scher János – Backend fejlesztés, Mobilalkalmazás fejlesztés**

* János volt felelős a backend fejlesztéséért, amely ASP.NET Core technológiával készült. A rendszer logikáját, adatkezelését és biztonságát ő biztosította. Ezen kívül dolgozott a mobilalkalmazás fejlesztésén is, hogy a felhasználók okostelefonon is kényelmesen használhassák a parkolóház rendszert.

**Major Attila – Adatbázis tervezés, Dokumentáció készítés**

* Attila felelt az adatbázis tervezéséért, hogy az jól strukturált és biztonságos legyen. Az ő feladata volt a dokumentáció készítése is, amely segített abban, hogy a projekt átlátható és könnyen karbantartható legyen a jövőben.

**Együttműködés és Kommunikáció**

A csapat minden tagja szorosan együttműködött, és rendszeresen kommunikáltunk, hogy minden feladat a megfelelő ütemben haladjon. A projekt különböző részei, mint a frontend, backend és mobilalkalmazás, folyamatos egyeztetés mellett készültek el. A célunk az volt, hogy egy jól működő és átfogó rendszert hozzunk létre, amely a parkolóházak hatékony kezelését segíti. A Trello ingyenes verzióját használtuk a feladatok nyomon követesére:

A képen szöveg, képernyőkép, Multimédiás szoftver, Számítógépes játék látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Ez a jól szervezett feladatmegosztás segített abban, hogy a projekt sikeresen megvalósuljon, és az elkészült parkolóház-kezelő rendszer megfeleljen a vizsgamunka követelményeinek.

# Ütemterv – a projekt előrehaladásának menete

A projekt fejlesztését 20 hetes ciklusban végeztük. A heti ütemezés az alábbiak szerint történt:

**1. Hét: Projektindítás és Analízis**

A projekt első hetében sor került a csapat összeállítására, a szerepkörök meghatározására, valamint a részletes követelménygyűjtésre a potenciális felhasználókkal. Piackutatás és konkurenciaelemzés is készült, majd kiválasztottuk a szükséges technológiákat. Végül beállítottuk a fejlesztői környezetet, és bevezettük a projektmenedzsment eszközöket (Trello).

**2. Hét: Adatbázis Tervezés**

Elkészítettük a részletes ER diagramot, elvégeztük az adatbázis normalizálását, valamint kidolgoztuk a teljesítmény-optimalizálási terveket. Meghatároztuk a migrációs stratégiákat, valamint elkészítettük a backup/recovery és biztonsági terveket is.- Backup és recovery terv kidolgozása

**3. Hét: Backend Architektúra**

Létrehoztuk az ASP.NET Core 8.0 projektstruktúrát, beállítottuk az Entity Framework Core-t és a MySQL adatbáziskapcsolatot. Megterveztük az API végpontokat, konfiguráltuk a Swagger dokumentációt, valamint a környezeti változók kezelését.

**4. Hét: Frontend Architektúra**

Inicializáltuk a Svelte projektet. Megterveztük a komponens architektúrát és kialakítottuk az API integrációs réteget.

**5. Hét: Felhasználói Autentikáció**

Implementáltuk a cookie alapú hitelesítést, létrehoztuk a regisztrációs és bejelentkezési folyamatokat, valamint az email értesítési rendszert.

**6. Hét: Járműkezelés Rendszer**

Létrehoztuk az Car entitást, implementáltuk a CRUD műveleteket, hozzáadtuk az automatikus logóbetöltést, validáltuk az adatokat, valamint elkészítettük a listázási, szűrési és törlési funkciókat.

**7. Hét: Parkolás Kezelés**

Létrehoztuk a ParkingSpot entitást, kialakítottuk a parkolás indítási/befejezési funkciókat, díjszámítást, foglalási lehetőséget, vizuális megjelenítést és a foglalt/szabad státusz jelzést.

**8. Hét: Adminisztrációs Rendszer**

Implementáltuk az admin jogosultságkezelést, a felhasználók és parkolóhelyek kezelését, valamint a statisztikák megtekintésének lehetőségét.

**9. Hét: Számlázási Rendszer**

Integráltuk az iTextSharp könyvtárat, lehetővé tettük PDF számlák generálását, letöltését, újraküldését, státuszuk kezelését és az előzmények megtekintését.

**10. Hét: Statisztikák és Rijportok**

Kialakítottuk a parkolási előzmények, összesítők, autónkénti és havi statisztikák, bevételi kimutatások és felhasználói aktivitások megjelenítését.

**11. Hét: Email Rendszer**

Integráltuk a MailKit könyvtárat, valamint létrehoztuk a számlák emailes kiküldésének rendszerét.

**12. Hét: Frontend Fejlesztés**

Tovább fejlesztettük a felhasználói felületet: reszponzivitás, vizuális komponensek, állapotkezelés, hibakezelés, betöltési állapotok és visszajelzések kerültek beépítésre.

**13. Hét: API Fejlesztés**

Megvalósítottuk a végpontokat, validációkat, hibakezeléseket, teljesítmény-optimalizálást, biztonsági ellenőrzéseket és frissítettük az API dokumentációt.

**14. Hét: Tesztelés**

Elvégeztük a kézi Swagger tesztelést az API funkciók ellenőrzéséhez.

**15. Hét: Dokumentáció**

Elkészült az API dokumentáció, a telepítési útmutató, valamint a kódhoz kapcsolódó kommentek és leírások.

**16. Hét: Deployment**

Konfiguráltuk a Render és Netlify szolgáltatásokat, beállítottuk az Aiven MySQL kapcsolatot, a környezeti változókat, SSL/TLS-t és CORS-t.

**17. Hét: Teljesítmény Optimalizálás**

Optimalizáltuk a backend és frontend teljesítményét.

**18. Hét: Biztonság**

Adatvédelmi fejlesztéseket végeztünk.

**19. Hét: Stabilizálás**

Elvégeztük a hibakezelést, naplózást, session kezelést, adatbázis stabilizálást, valamint a számlázási rendszer, frontend és biztonság végső simításait.

**20. Hét: Projekt Zárás**

Végső tesztelés, dokumentáció frissítés, valamint a projekt bemutatójához szükséges prezentáció (ppt) elkészítése történt meg.

A projekt során a heti ütemezés szerint haladtunk, és minden modul fejlesztése összhangban történt az előre megtervezett lépésekkel. A backend ASP.NET Core 8 technológiával, a frontend Svelte alapokon készült. A fejlesztés során külön figyelmet fordítottunk a reszponzív felhasználói felületre, valamint a megbízható adatkezelésre.

# Felhasználói felület és alkalmazott technológiák

A felhasználói felület a Svelte keretrendszer segítségével készült. A rendszer reszponzív, azaz asztali és mobil eszközökön egyaránt használható. A dizájn célja az egyszerű kezelhetőség és átláthatóság biztosítása volt, modern, letisztult megjelenéssel. A frontend főbb oldalai és funkciói:

1. **Bejelentkezés és Regisztráció**

* Felhasználói regisztráció email cím és jelszó megadásával
* Bejelentkezés a regisztrált felhasználói adatokkal
* Biztonságos kijelentkezés

A képen szöveg, képernyőkép, Márka, embléma látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **Járműkezelés**

* Új jármű hozzáadása rendszám és márka megadásával
* Járművek listázása és kezelése
* Automatikus márka logók betöltése
* Jármű adatok szerkesztése és törlése

A képen szöveg, képernyőkép, Betűtípus, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **Parkolási Események**

* Parkolóhelyek megtekintése és foglalása
* Aktuális parkolási állapot követése
* Parkolási előzmények megtekintése
* Parkolási díjak és számlák kezelése

A képen szöveg, képernyőkép, tér, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szám, diagram látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **Admin Panel**

* Felhasználók kezelése
* Parkolóhelyek üzemeltetése
* Statisztikák és riportok megtekintése
* Számlák kezelése és újraküldése
* Rendszerbeállítások módosítása

A képen szöveg, szoftver, Számítógépes ikon, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, szoftver, Számítógépes ikon, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, szoftver, Számítógépes ikon, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, Betűtípus, szám, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, szoftver, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **Mobil applikáció**
   * Egyszerű, átlátható felület
   * Könnyű kezelhetőség

|  |  |
| --- | --- |
|  |  |

**A felület főbb jellemzői:**

* Modern, reszponzív dizájn
* Intuitív navigáció
* Valós idejű visszajelzések
* Automatikus email értesítések
* Biztonságos adatkezelés
* Felhasználóbarát hibaüzenetek

A frontend a Netlify szolgáltatáson keresztül van üzemeltetve, amely alapvetően megbízható működést biztosít, azonban időnként lassabb betöltést eredményezhet. A rendszer automatikusan frissül az új verziókkal, és a felhasználók mindig a legfrissebb verziót látják.

# Adatbázisok és eszközök részletes bemutatása

Az alkalmazás MySQL adatbázist használ, amelyet az Aiven szolgáltatáson keresztül üzemeltetünk. Az adatbázis relációs adatmodellre épül, és az Entity Framework Core 8.0 ORM keretrendszer segítségével kezeljük. Az adatbázisban a következő fő entitások találhatók:

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **User (Felhasználó)**

* Azonosító, email, jelszó hash, név, telefonszám, admin jogosultság
* Kapcsolatok: járművek, parkolási események, számlák

1. **Car (Jármű)**

* Azonosító, rendszám, márka, modell, évjárat, parkolási státusz
* Kapcsolatok: felhasználó, parkolóhely

1. **ParkingSpot (Parkolóhely)**

* Azonosító, emelet szám, helyszám, foglaltság
* Kapcsolatok: jármű, parkolási események

1. **ParkingHistory (Parkolási Előzmény)**

* Azonosító, kezdő időpont, záró időpont, díj
* Kapcsolatok: felhasználó, jármű, parkolóhely

1. **Invoice (Számla)**

* Azonosító, számlaszám, összeg, kiállítás dátuma, esedékesség
* Kapcsolatok: felhasználó, parkolási esemény

A képen szöveg, képernyőkép, Betűtípus, Párhuzamos látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Fejlesztői Környezet**

A projekt fejlesztéséhez a következő eszközöket és technológiákat használtuk:

1. **Backend**

* ASP.NET Core 8.0
* Entity Framework Core 8.0
* Pomelo.EntityFrameworkCore.MySql
* Swagger/OpenAPI
* MailKit (email küldés)
* iTextSharp (PDF generálás)

1. **Frontend**

* Svelte

1. **Adatbázis**

* MySQL 8.0
* Aiven üzemeltetés

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **Deployment**

* Render (backend)

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

* Netlify (frontend)

A képen szöveg, szoftver, képernyőkép, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. **Verziókezelés**

* Github

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A rendszer moduláris felépítésű, ami lehetővé teszi a könnyű bővíthetőséget és karbantarthatóságot. Az adatbázis struktúra rugalmas, az API jól dokumentált, és a kód architektúra tiszta, ami megkönnyíti a későbbi fejlesztéseket.

# Minőségbiztosítás és tesztelési folyamatok

**API Tesztelés**

1. **Swagger Integráció**

* A rendszer Swagger/OpenAPI dokumentációt használ
* A Swagger UI elérhető a fejlesztési környezetben: http://localhost:5025/swagger
* Az API végpontok dokumentációja és tesztelési felülete automatikusan generálódik

1. **API Végpont Tesztelés**

* Robot Framework végponttesztelés:

A képen szöveg, képernyőkép, szoftver, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

* Postman tesztelés

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

* A végpontok tesztelése a Swagger UI-on keresztül is történik
* A válaszok és hibakezelések ellenőrzése

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, sor, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szám, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, sor, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szám, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Frontend Tesztelés**

1. **API Integráció Tesztelés**

* Az api.js fájlban részletes hibakezelés és naplózás implementálva
* A kérések és válaszok részletes naplózása fejlesztési célokra
* CORS beállítások tesztelése különböző domainek között

1. **Reszponzív Tesztelés**

* A frontend reszponzív dizájn tesztelése
* CORS beállítások konfigurálva a különböző környezetekhez:
* Lokális fejlesztés: http://localhost:5173
* Netlify deployment: https://parking-garage-app.netlify.app
* Render deployment: https://\*.onrender.com

1. **Automata tesztelés**
   1. Teszteset létrehozása a bejelentkezésre Robot Framework-el:

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, szám, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.  
A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.A képen szöveg, szoftver, Multimédiás szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Környezeti Tesztelés**

1. **Fejlesztői Környezet**

* Lokális fejlesztés: http://localhost:5025
* Adatbázis seed tesztelés
* Környezeti változók kezelése

1. **Produkciós Környezet**

* Render deployment tesztelés
* Netlify deployment tesztelés
* Aiven MySQL kapcsolat tesztelése

**Biztonsági Tesztelés**

1. **Autentikáció**

* Cookie alapú hitelesítés tesztelése
* Jogosultságkezelés ellenőrzése
* Admin felület védelme

1. **Adatbázis Biztonság**

* SSL kapcsolat tesztelése
* Környezeti változók kezelése
* Adatbázis kapcsolat timeout kezelése

A tesztelés főként a Swagger UI-on keresztül és manuális ellenőrzésekkel történik. Néhány automata tesztet is végeztünk, de nem ez volt a jellemző. A frontend oldalon az API hívások részletes naplózása segíti a hibakeresést és tesztelést.

A rendszer reszponzivitását különböző eszközökön is kipróbáltuk (asztali gép, tablet, mobiltelefon). A tesztelésre megkért felhasználóktól kapott visszajelzések alapján folyamatosan javítottuk az élményt és hibákat.

# Összegzés

A **Parkolóház Kezelő Rendszer** projekt során egy olyan alkalmazást hoztunk létre, amely **valóban hasznosítható egy valós parkolóház működtetéséhez**. A rendszer fő erőssége, hogy **egyszerűen használható**, miközben minden szükséges funkciót tartalmaz egy modern parkolóház üzemeltetéséhez.

**Felhasználói oldal**

A felhasználók számára a rendszer **könnyen kezelhető és barátságos**:

* Regisztráció után azonnal hozzáadhatják járműveiket.
* A parkolás teljesen **automatizált**.
* A rendszer **automatikusan számolja a díjakat**, generálja a számlákat, és emailben el is küldi azokat.
* A felhasználók bármikor **megtekinthetik parkolási előzményeiket**.

**Adminisztrációs oldal**

Az adminisztrációs oldal a parkolóház üzemeltetői számára is **hatékony és átlátható kezelőfelületet** biztosít:

* Az adminok **kezelhetik a felhasználókat**, **követhetik a parkolóhelyek foglaltságát**, és **részletes statisztikákat** tekinthetnek meg.
* A számlák **automatikus generálása és küldése** jelentősen csökkenti az adminisztratív terheket.

**Technikai megvalósítás**

A rendszer modern technológiákra épül:

* **Backend**: ASP.NET Core 8.0, amely megbízható és gyors működést biztosít.
* **Frontend**: Svelte és TypeScript, amelyek reszponzív és kellemes felhasználói élményt nyújtanak.
* **Adatbázis**: MySQL, az **Aiven** szolgáltatáson keresztül hosztolva, amely **biztonságos és skálázható adattárolást** kínál.

A rendszer jelenleg sikeresen üzemel a **Render** (backend) és **Netlify** (frontend) platformokon, amelyek lehetővé tették a valódi környezetben történő tesztelést is. A felhasználói visszajelzések alapján a rendszer **megbízhatóan működik**, és **valóban támogatja a parkolóházak üzemeltetését**.

**További fejlesztési lehetőségek**

A jövőben a következő fejlesztések jelenthetnek előrelépést:

* **Rendszerbeállítások felület** az adminok számára.
* **Audit log rendszer** a műveletek követhetősége érdekében.
* **Teljesítményoptimalizálás**.
* **További biztonsági fejlesztések**.

Összességében a projekt sikeresen demonstrálja, hogyan lehet **egy valós üzleti problémát megoldani modern technológiákkal**. A rendszer nemcsak működőképes, hanem valóban hasznosítható egy parkolóház üzemeltetésében, miközben a felhasználók számára is **kiváló élményt** nyújt.

# Felhasznált források

https://learn.microsoft.com/en-us/dotnet/

https://svelte.dev/

https://getbootstrap.com/

https://github.com/

https://www.mysql.com/

https://trello.com/

https://chat.openai.com/