

**ZÁRÓDOLGOZAT**

Készítették:

Ágoston Attila - Lőrincz Loránd László

Konzulens:

Kasza László Róbert

Miskolc

2025.

Miskolci SZC Kandó Kálmán Informatikai Technikum

**SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK**

**CsaposApp**

Vendéglátó egység program

Ágoston Attila – Lőrincz Loránd László

**Bevezetés**

A vendéglátóipar egyre nagyobb mértékben támaszkodik a digitális megoldásokra, különösen az asztalfoglalás és a rendeléskezelés terén. Programunk célja, hogy modern, hatékony és átlátható rendszert biztosítson egy kocsma mindennapi működéséhez, megkönnyítve ezzel a vendégek fogadását, a rendelések kezelését, valamint a személyzet munkáját. A hagyományos, papíralapú vagy szóban történő folyamatokat egy digitális platform váltja fel, amely gyorsabb, pontosabb és megbízhatóbb működést tesz lehetővé.

A rendszer támogatja az előzetes asztalfoglalás lehetőségét, valós idejű rendeléskezelést biztosít a pultosok és felszolgálók számára, és áttekinthető módon jeleníti meg az asztalok állapotát. A fejlesztés során kiemelt figyelmet fordítottunk a felhasználóbarát kezelőfelület kialakítására, a gyors adatbevitelre, valamint arra, hogy a rendszer könnyen bővíthető és skálázható legyen a jövőbeni igények szerint.

A dokumentáció átfogó képet ad a program működéséről, technikai hátteréről, valamint részletesen bemutatja a telepítés, konfigurálás és napi használat lépéseit.

**Fejlesztés során használt főbb technológiák**

**Adatbázis:**

* MySQL
* phpMyAdmin kezelőfelület

**Backend:**

* ASP.NET
* C#
* Entity Framework
* SignalR
* ASP.NET kiegészítő könyvtárak

**Frontend:**

* React
* HTML
* CSS, TailwindCSS
* JavaScript

**Asztali alkalmazás:**

* Electron
* React
* HTML
* CSS
* JavaScript

**Szerver:**

* Docker
* Caddy
* Cloudflare

**Verziókezelés:**

* Git
* Github

**Technológiák rövid bemutatása**

**MySQL**

A **MySQL** egy nyílt forráskódú, relációs adatbázis-kezelő rendszer, amelyet főként strukturált adatok tárolására és kezelésére használnak. A lekérdezésekhez az **SQL** (Structured Query Language) nyelvet használja. Gyors, megbízható, skálázható, és gyakran használják webalkalmazások háttérrendszereként (pl. **WordPress, phpMyAdmin**), ezért is esett erre a rendszerre a választásunk.

**phpMyAdmin**

A **phpMyAdmin** egy nyílt forráskódú, webes felület a **MySQL** és **MariaDB** adatbázisok kezeléséhez. **PHP** nyelven íródott, és lehetővé teszi az adatbázisok létrehozását, módosítását, lekérdezését és karbantartását grafikus felületen keresztül, anélkül hogy **SQL**-parancsokat kellene írni. Könnyen telepíthető, és gyakran használják fejlesztők, rendszergazdák és webtárhely-szolgáltatók körében.

**ASP.NET**

**ASP.NET** egy nyílt forráskódú szerveroldali webfejlesztési keretrendszer, amelyet a **Microsoft** fejlesztett ki a **.NET** platform részeként. Lehetővé teszi dinamikus, interaktív weboldalak, webalkalmazások, API-k és webszolgáltatások létrehozását. A fejlesztők olyan nyelveket használhatnak benne, mint pl. a **C#** vagy **VB.NET** és könnyen integrálható más Microsoft technológiákkal, például **SQL Szerverrel** vagy az **Azure**-rel.

**C#**

A **C#** egy Microsoft által fejlesztett, modern, objektumorientált programozási nyelv a **.NET** platformhoz. Egyszerű, biztonságos és sokoldalú: használható webes, asztali, mobil- és játékalkalmazások fejlesztésére is. Támogatja a modern programozási technikákat, például az aszinkron működést és a LINQ-t. Népszerű a **ASP.NET** és **.NET MAUI** fejlesztésekben.

**Entity Framework**

Az **Entity Framework** a **.NET** platform **ORM** (Object-Relational Mapper) eszköze, amely megkönnyíti az adatbázisokkal való munkát. Lehetővé teszi, hogy az adatokat objektumokként kezeljük, így nem kell közvetlenül **SQL** lekérdezéseket írni – az adatbázis-műveletek **C#** kóddal történnek. Ez a technológia megakadályozza az SQL injection megvalósítását.

**SignalR**

A **SignalR** a Microsoft által fejlesztett könyvtár a **.NET** környezetben, amely lehetővé teszi valós idejű, kétirányú kommunikációt a kliens (pl. böngésző) és a szerver között. Ezt olyan technológiákra építve valósítja meg, mint a **WebSocket, Server-Sent Events** vagy **Long Polling**, és automatikusan kiválasztja a legjobban támogatott megoldást az adott környezethez.

**React**

A **React** egy **JavaScript** alapú nyílt forráskódú könyvtár, amelyet a Meta (Facebook) fejlesztett ki interaktív, gyors és hatékony felhasználói felületek (UI) készítésére. Elsősorban egylapos alkalmazásokhoz (SPA) használják, ahol a tartalom dinamikusan változik anélkül, hogy az egész oldal újratöltődne.

**Electron**

Az Electron és a React kombinálása lehetővé teszi, hogy webes technológiákat (HTML, CSS, JavaScript) használj natív asztali alkalmazások fejlesztésére, miközben a React előnyeit is kihasználhatod, például a komponensalapú felépítést és az állapotkezelést. Ezt a megoldást gyakran alkalmazzák, hogy gyorsan és hatékonyan építsenek multiplatform asztali alkalmazásokat, amelyek a modern webfejlesztési paradigmákra építenek.

**Docker**

A Docker egy nyílt forráskódú platform, amely lehetővé teszi alkalmazások és azok környezetének konténerizálását, azaz az alkalmazások futtatását izolált, hordozható egységekben, úgynevezett konténerekben. A konténerek biztosítják, hogy az alkalmazás minden szükséges függőséggel együtt, egységes környezetben fusson, függetlenül attól, hogy milyen operációs rendszeren fut.

**Caddy**

A Caddy egy modern, könnyen használható, nyílt forráskódú webkiszolgáló, amely különösen a biztonságra és a konfiguráció egyszerűségére összpontosít. Automatikusan kezel SSL/TLS tanúsítványokat, és alapértelmezés szerint HTTPS-t használ, így a webhelyek biztonságos üzemeltetése sokkal egyszerűbbé válik.

**Fejlesztői környezetek bemutatása**

**Visual Studio Code**

A **Visual Studio Code** (VS Code) egy ingyenes, nyílt forráskódú kódszerkesztő, amelyet a Microsoft fejlesztett. Könnyű és gyors, de mégis erőteljes eszközként szolgál webfejlesztéshez, asztali alkalmazások készítéséhez és egyéb programozási feladatokhoz. Támogatja a legtöbb népszerű programozási nyelvet, és bővíthető különféle kiegészítőkkel, mint például kódkiegészítők, hibakereső eszközök és Git integráció. Beépített terminál és verziókezelési lehetőségek segítik a fejlesztést, miközben a testreszabhatóság és a gyors működés lehetővé teszi a fejlesztők számára a hatékony munkát.

**Visual Studio 2022**

A **Visual Studio 2022** egy integrált fejlesztői környezet (IDE) a Microsofttól, amely gyors és erőteljes eszközt kínál asztali, webes, mobil és játékalkalmazások fejlesztéséhez. A 64 bites verzió gyorsabb teljesítményt és nagyobb projektek kezelését teszi lehetővé. A Visual Studio 2022 ideális választás komplex fejlesztési feladatokhoz különböző platformokon.

**Verziókezelés**

**Git**

A Git egy elosztott verziókezelő rendszer, amely lehetővé teszi a fejlesztők számára, hogy nyomon kövessék a kódváltozásokat és együttműködjenek másokkal. A Git a fájlok és a kód történelmét rögzíti, így könnyen visszaállíthatók a korábbi verziók, valamint összevonhatók a különböző fejlesztői ágak.

**Github**

A GitHub egy felhőalapú platform, amely lehetővé teszi a Git verziókezelő rendszer használatát kód tárolására, kezelésére és megosztására. A fejlesztők itt tárolhatják és kezelhetik projektjeiket, miközben együttműködhetnek másokkal, akár nyilvános, akár privát repository-kban. A projekünk Github repository-ja megtekinthető itt: [https://github.com/Solmyr77/CsaposApp](https://github.com/Solmyr77/CsaposApp/)

**Kommunikációs eszközök**

**Discord**

A **Discord** egy ingyenes kommunikációs platform, amelyet elsősorban játékosok és közösségek használnak, de az évek során szélesebb körben is elterjedt, mint eszköz a csoportos beszélgetésekhez és akár munkahelyi együttműködésekhez is. A Discord lehetővé teszi a felhasználók számára, hogy szöveges, hang- és videóhívásokat folytassanak, emellett lehetőség van különféle csatornák létrehozására és szervezésére. A projekt megvalósításának egésze alatt ezen a platformon tartottuk a kapcsolatot.

**Trello**

A **Trello** egy ingyenes webes projekmenedzselő eszköz ami lehetőséget biztosít arra, hogy a csapattagok valós időben frissíthessék a feladatok állapotát, és megosszák a fontos információkat. A platformon oszlopokba rendezett kártyák segítségével menedzselhetkük egyes feladatok állapotát. A projekünk Trello-ja megtekinthető itt: <https://trello.com/b/36hRnRs1/csaposapp> A screenshot of a computer

AI-generated content may be incorrect.

**Github Issues**

A **GitHub Issue** egy eszköz a **GitHub** platformon, amely lehetővé teszi a fejlesztők számára, hogy nyomon kövessék a hibákat, funkciók kérését, vagy bármilyen más típusú feladatot egy projektben. Az Issue egyfajta nyilvántartásként működik, amely segít a csapatoknak az egyes problémák és feladatok szervezésében és nyomon követésében.

**Feladatok megoszlása a csapaton belül**

A feladatok megosztása csapaton belül tökéletesnek bizonyult, két nagyjából egyenlő részre próbáltuk osztani a feladatokat. Az erősségeinket szem előtt tartva választottuk ki az általunk megvalósítani kívánt részeket. Loránd a szerver, adatbázis és a backend létrehozásáért felelt, Attila pedig a webes és az asztali felületekért.

**Design tervek**

**Screens screenshots of a phone

AI-generated content may be incorrect.**

Figmában készült első designterv a weboldalról, mobil nézetben.

A screenshot of a computer

AI-generated content may be incorrect.

Figmában készült kezdetleges asztali alkalmazás látványterv.

**Felhasználói dokumentáció**

**Weboldal bemutatása**

A <https://csaposapp.hu> linken elérhető weboldalunk elsősorban egy mobil készülékekre optimalizált, asztalfoglalásokat és rendeléseket lebonyolító platform, ami letisztult designjával és egyszerű használatával a felhasználók kényelmét szolgálja.

A screenshot of a phone

AI-generated content may be incorrect.**Az oldal működése:**

A screenshot of a phone

AI-generated content may be incorrect.

A fent említett linkre kattintva a bejelentkezési oldalra jutunk, ahol megadhatjuk a fiókunkhoz tartozó felhasználónevet és jelszót. A **“Bejelentkezés”** gombra kattintva hozzáférhetünk a fiókunkhoz.

Ha még nem rendelkezünk fiókkal, a “**Regisztráció”** gombra kattintva létrehozhatunk egyet. Ehhez meg kell adnunk a teljes nevünket, születési évünket, email-címünket, egy felhasználónevet és kétszer a választott jelszót. Fontos, hogy 18 év alattiak nem regisztrálhatnak az oldalra, és a jelszónak tartalmaznia kell egy nagybetűt, egy kisbetűt, egy számot, valamint legalább 8 karakter hosszúnak kell lennie.

A screenshot of a phone

AI-generated content may be incorrect.

Bejelentkezés után az oldal automatikusan a **Felfedezés** felületre navigál, ahol megtekinthetők a platformon elérhető vendéglátóhelyek. A jobb felső sarokban található levél ikonra kattintva elérhetjük az Üzenetek oldalt.

A Felfedezés felirat alatt elhelyezett három gomb segítségével szűrhetjük a megjelenített kocsmákat különböző kategóriák szerint. A Kiemelt egységek szekcióban egy úgynevezett Swiper komponens található, amely automatikus diavetítéssel, néhány másodpercenként váltakozó kártyákkal mutatja be az ajánlott helyeket.

A Swiper komponensben szereplő nagy kártyákra, vagy az alattuk található kis kártyákra kattintva közvetlenül elérhetjük a kiválasztott vendéglátóhely részletes oldalát.

Az oldal alján elhelyezett fő navigációs sáv segítségével könnyedén válthatunk a Felfedezés, Keresés és Profil aloldalak között.

A screenshot of a phone

AI-generated content may be incorrect.A **Keresés** aloldalon lehetőség van vendéglátóhelyek név szerinti keresésére. Az oldal tetején található egy keresősáv, ahol beírhatjuk a keresett hely nevét vagy annak egy részletét. A beírt szöveg alapján az oldal valós időben szűri a találatokat.

Minden találat egy kártyaként jelenik meg, amely tartalmazza a vendéglátóhely nevét és képét – ezekre kattintva közvetlenül elérhető az adott egység profiloldala.

A keresősávban megjelenő **X**-re kattintva törölhetjük a keresőmezőbe írt szöveget.

A screenshot of a phone

AI-generated content may be incorrect.A **Profil** aloldal a felhasználó saját fiókjának kezelésére szolgál. A képernyő tetején megjelenik a felhasználó profilképe és felhasználóneve. Ez alatt található a **Barátok** szekció, ahol a felhasználó ismerősei láthatók profilképes sorban. Az első ikon segítségével új barátokat is hozzáadhatunk.

A **Beállítások** részben két lehetőség érhető el:  
– **Profil szerkesztése**, ahol módosíthatjuk a profilképünket, nevünket vagy más személyes adatokat. (Mellékletben képernyőkép csatolva)  
– **Jelszó módosítása**, amely lehetőséget ad egy új jelszó megadására a fiók biztonságának fenntartása érdekében. (Mellékletben képernyőkép csatolva)

Az oldal alján található a **Kijelentkezés** gomb, amellyel kiléphetünk a fiókunkból.

A screenshot of a phone

AI-generated content may be incorrect.Egy kocsma kiválasztása után a felhasználó egy dedikált információs oldalra érkezik. Az oldal tetején egy fotó látható a helyről, alatta pedig megjelenik a kocsma neve, pontos címe, valamint az értékelés.

Egy **“Nyitva”** címke jelzi az aktuális nyitvatartási állapotot, mellette pedig a zárási idő olvasható. Ez alatt egy leírás olvasható a helyről.

Az események szekcióban a helyhez tartozó események jelennek meg. Egy eseményre kattintva átnavigál az oldal az adott esemény oldalára. (Mellékletben képernyőkép csatolva)

Az alsó részén egy **“Foglalok”** gomb kapott helyet, amely közvetlen átjárást biztosít az asztalfoglalási felületre.

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A foglalási felületen először egy áttekinthető időpontválasztó szekció jelenik meg, ahol az aktuális naptól kezdve a következő 7 napra adhatunk le foglalásokat. A rendelkezésre álló idősávok a nyitvatartási időhöz igazodnak, és 15 perces intervallumokra vannak bontva, így rugalmasan választhatunk érkezési időpontot.

Ezt követően lehetőség van egy szabad asztal kiválasztására az adott napra. Csak a még elérhető asztalok jelennek meg, így biztosan nem történik átfedés más foglalásokkal.

A rendszer lehetőséget biztosít barátok meghívására is – annyi fő hívható meg, amennyi az adott asztal kapacitásával összhangban van. A barátok valós időben értesítést kapnak a meghívásról. (Mellékletben képernyőkép csatolva)

A kiválasztott adatok alapján a felület egy foglalás összegzőt mutat, ahol minden fontos részlet – időpont, asztal, résztvevők – áttekinthető. Ha mindent rendben találunk, a **“Foglalás”** gombra kattintva véglegesíthetjük a foglalást.

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Sikeres foglalás után a főoldalon megjelenik egy **Foglalásaim** szekció, ahol a felhasználó saját foglalásai, valamint az általa elfogadott meghívások is nyomon követhetők. Egy adott foglalásra kattintva egy részletes összegzőt láthatunk, amely tartalmazza a foglalás pontos adatait és annak aktuális státuszát.

A **Lemondás** gomb segítségével a foglalás bármikor visszavonható. A lemondás valós időben frissül, így a meghívott barátok azonnal értesülnek róla, és a vendéglátóhely rendszere is automatikusan naprakész marad.

Amikor elérkezik a foglalás időpontja, a **Kezdés** gombra kattintva a rendszer átirányítja a felhasználót az adott kocsma rendelési felületére, ahol azonnal megkezdhető a fogyasztás és a rendelés leadása.

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

1. Kép: A felhasználó a rendelési felületre érkezve kategóriákra bontva tekintheti meg az adott hely kínálatát. A jobb felső sarokban lévő asztal ikonra kattinta átnavigálhat az **Asztalom** felületre.

2. Kép: Egy termék jobb oldalán található táska ikonra kattintva a képernyő alján egy felugró ablak jelenik meg, ahol beállítható a termék mennyisége, majd egy gomb segítségével hozzáadható a rendeléshez.

3. Kép: A jobb alsó sarokban lévő **"Rendelésem"** vagy a bal alsó sarokban található táska ikonra kattintva megnyílik a rendelés összegző felület. Itt lehetőség van a rendelt tételek mennyiségének módosítására, a termékek eltávolítására, valamint a rendelés véglegesítésére. A leadott rendelések valós időben megjelennek a foglalásban szereplő meghívott barátoknál is, illetve a vendéglátóhely rendszerében is.

A screenshot of a phone

AI-generated content may be incorrect.A screenshot of a phone

AI-generated content may be incorrect.

Az asztalom felület tetején megjelennek a foglalásban résztvevő tagok, valamint egy „Számlám” gomb, amely a Stripe online fizetési rendszerre utaló vizuális elemként szolgál (a funkció jelenleg még nem került implementálásra).

Az alatta elhelyezkedő szekcióban a felhasználó saját rendelései, valamint a barátok által leadott rendelések láthatók időrendi sorrendben. Egy státuszjelző is megjelenik minden rendelés mellett, amely jelzi, hogy az adott rendelés teljesítve lett-e.

A screenshot of a phone

AI-generated content may be incorrect.

**Asztali alkalmazás bemutatása**

Az asztali alkalmazásunk a Point of Sale rendszer szerves része, amely a vendégek által leadott asztalfoglalások és rendelések kezelését végzi. A rendszer lehetővé teszi az éttermi személyzet számára, hogy gyorsan és hatékonyan kövessék nyomon a beérkező kéréseket, miközben az egyszerű, letisztult felület megkönnyíti a használatát és lerövidíti a betanulási időt.

**Backend bemutatása**

A backend rendszerünk központi szerepet játszik az alkalmazás működésében: összeköti a felhasználói felületet (webes és asztali alkalmazást), a Point of Sale rendszert és az adatbázist. Fő feladata az adatok biztonságos fogadása, feldolgozása és továbbítása a különböző komponensek között.

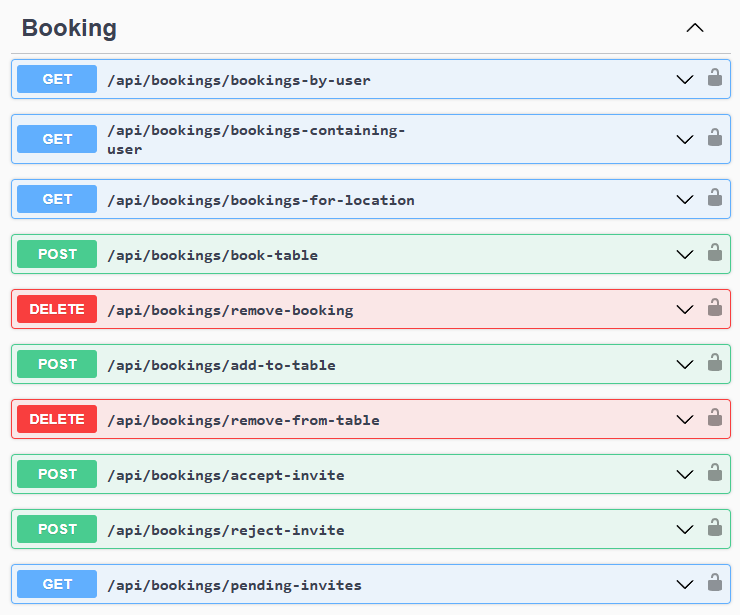
A rendszer backendje több mint 70 különálló REST API végpontot tartalmaz, amelyek lefedik az alkalmazás teljes funkcionalitását – a felhasználókezeléstől kezdve a rendelésfeldolgozáson át az adminisztrációs műveletekig. Ezek a végpontok a Swagger dokumentációs felületen is megtalálhatók, ahol interaktív módon tesztelhetők és áttekinthetők. Az alábbiakban a legfontosabb modulokhoz tartozó végpontok kerülnek bemutatásra példaként.

<https://backend.csaposapp.hu/swagger/index.html>

A rendszer teljes egészében valós időben működik, így a foglalások és meghívások állapotváltozásai azonnal lekövethetők a kliensoldalon is.

**Booking API végpontok**

A képen látható Swagger felület a backend Booking (asztalfoglalási) moduljához tartozó REST API végpontokat mutatja be. Ezek az API-k felelősek az asztalfoglalások kezeléséért, a foglalások létrehozásától kezdve a meghívások kezelésén át egészen azok törléséig. Az API a frontend, az asztali alkalmazás és a backend közötti kommunikáció alapját képezi.



**A Booking végpontok rövid leírása:**

* GET /api/bookings/bookings-by-user  
  Az adott felhasználó által létrehozott foglalásokat adja vissza.
* GET /api/bookings/bookings-containing-user  
  Azokat a foglalásokat listázza, amelyekben a felhasználó résztvevőként szerepel (pl. meghívott vendégként).
* GET /api/bookings/bookings-for-location  
  Egy adott vendéglátóhelyhez tartozó összes foglalást adja vissza.
* POST /api/bookings/book-table  
  Új foglalás létrehozása egy kiválasztott asztalhoz és időponthoz.
* DELETE /api/bookings/remove-booking  
  Meglévő foglalás törlése a rendszerből.
* POST /api/bookings/add-to-table  
  Felhasználó hozzáadása egy már létező foglaláshoz, például meghívás elfogadásakor.
* DELETE /api/bookings/remove-from-table  
  Felhasználó eltávolítása egy foglalásból.
* POST /api/bookings/accept-invite  
  Foglalási meghívás elfogadása.
* POST /api/bookings/reject-invite  
  Meghívás visszautasítása.
* GET /api/bookings/pending-invites  
  A felhasználó függőben lévő meghívásainak lekérdezése.

**Products API végpontok**

A fenti ábra a Products modul Swagger-felületét mutatja, amely az étel- és italkínálat kezelésére szolgáló REST API végpontokat tartalmazza. Ez a modul biztosítja az adminisztrátorok és pultosok számára a termékek (pl. sörök, ételek, italok) listázását, szerkesztését és törlését, valamint lehetőséget ad új elemek felvételére is.



**A Products végpontok rövid leírása:**

* GET /api/products  
  Az összes termék lekérdezése az adatbázisból.
* POST /api/products  
  Új termék létrehozása (név, ár, kategória, leírás stb. megadásával).
* GET /api/products/location/{locationId}  
  Egy adott vendéglátóegységhez tartozó termékek listázása a locationId alapján. Ez lehetővé teszi, hogy különböző helyek egyedi kínálattal rendelkezzenek.
* GET /api/products/{id}  
  Egy konkrét termék adatainak lekérdezése azonosító (id) alapján.
* PUT /api/products/{id}  
  Meglévő termék adatainak módosítása (pl. ár frissítése, leírás szerkesztése).
* DELETE /api/products/{id}  
  Termék törlése az adatbázisból a megadott azonosító alapján.

Ezek a végpontok biztosítják a termékkezelés teljes életciklusát, támogatva a valós idejű kínálatfrissítést és az adminisztrációs feladatokat a rendszerben.

**Auth API végpontok**

A fenti ábra a hitelesítési (Auth) modul Swagger-felületét mutatja, amely a felhasználói azonosítással és jogosultságkezeléssel kapcsolatos műveletekért felelős. Ez a modul biztosítja a biztonságos bejelentkezést, regisztrációt, tokenkezelést és jelszómódosítást.



**Az Auth végpontok rövid leírása:**

* POST /api/auth/login  
  Felhasználó bejelentkezése érvényes felhasználónév és jelszó megadásával. A válaszban JWT token kerül visszaküldésre.
* POST /api/auth/register  
  Új felhasználó regisztrálása a szükséges adatok megadásával (pl. név, jelszó, email).
* POST /api/auth/logout  
  A felhasználó kijelentkeztetése, a szerver oldali session vagy token érvénytelenítése.
* POST /api/auth/refresh-token  
  Lejárt vagy hamarosan lejáró token frissítése új hozzáférési token igénylésével.
* PUT /api/auth/update-password  
  A felhasználó jelszavának módosítása a jelenlegi és az új jelszó megadásával.

Ez a modul gondoskodik az alkalmazás biztonságos használatáról, és a hozzáférési jogosultságok érvényesítéséről minden más végpont eléréséhez.