Okos Magántanár Kereső

Vizsgraremek 13. C

Szoftverfejlesztés és tesztelés

BGSZC Logisztikai és kereskedelmi technikum és szakképző iskola

BAracskai Angyalka, MAráczi Péter

2025

# 

Tartalom

[1 Bevezetés 1](#_Toc197200995)

[1.1 Feladat leírás 1](#_Toc197200996)

[1.2 A felhasznált ismeretek 1](#_Toc197200997)

[1.3 A felhasznált szoftverek 1](#_Toc197200998)

[2 Felhasználói dokumentáció 3](#_Toc197200999)

[2.1 A program általános specifikációja 3](#_Toc197201000)

[2.2 Rendszerkövetelmények 3](#_Toc197201001)

[2.2.1 Hardver követelmények 3](#_Toc197201002)

[2.2.2 Szoftver követelmények 4](#_Toc197201003)

[2.3 A program telepítése 4](#_Toc197201004)

[2.4 A program használatának a részletes leírása 8](#_Toc197201005)

[3 Fejlesztői dokumentáció 16](#_Toc197201006)

[3.1 Az alkalmazott fejlesztői eszközök 16](#_Toc197201007)

[3.2 Adatmodell leírása 17](#_Toc197201008)

[3.2.1 Táblák 17](#_Toc197201009)

[3.2.2 Kapcsolatok: 18](#_Toc197201010)

[3.2.3 Jellemzők: 19](#_Toc197201011)

[3.3 Részletes feladatspecifikáció, algoritmusok 19](#_Toc197201012)

[3.4 Tesztelési dokumentáció 21](#_Toc197201013)

[4 Továbbfejlesztési lehetőségek 24](#_Toc197201014)

[5 Önértékelés 25](#_Toc197201015)

[6 Felhasznált irodalom 26](#_Toc197201016)

[7 Ábrajegyzék 27](#_Toc197201017)

# 

# 1 Bevezetés

## Feladat leírás

A projekt célja egy olyan webes alkalmazás kifejlesztése, amely lehetővé teszi diákok számára, hogy magántanárokat találjanak különböző tantárgyakból. A weboldal lehetőséget biztosít keresésre, értékelésre, valamint profilkezelésre mind diákok, mind tanárok számára.

## A felhasznált ismeretek

* HTML, CSS, Bootstrap keretrendszer használata a frontend megvalósításához
* JavaScript a dinamikus viselkedéshez
* C#, Node.js és Express.js a backend fejlesztéséhez
* MySQL adatbázis-kezelő
* SQL injekció elleni védelem és adatbázis biztonsági szabályok alkalmazása

## A felhasznált szoftverek

* Visual Studio Code
* Node.js környezet
* MySQL Workbench
* Postman (teszteléshez)
* Git verziókezelő
* Böngészők (Chrome, Firefox) teszteléshez
* Exprees
* Node.js
* React

# 

# Felhasználói dokumentáció

## A program általános specifikációja

Az Okos Magántanár Kereső Weboldal célja, hogy hatékony platformot biztosítson a tanulni vágyó diákok és magántanárok közötti kapcsolat létrehozására. A rendszer átlátható és felhasználóbarát módon biztosítja a tanárok közötti böngészést, szűrést különböző szempontok alapján (például tantárgy, város, elérhetőség, ár, oktatási forma), valamint lehetővé teszi a tanárok profiljának megtekintését, az előző diákok értékeléseinek elolvasását és közvetlen kapcsolatfelvételt.

A regisztrált diákok üzenetküldő rendszeren keresztül vehetik fel a kapcsolatot a tanárokkal, ezzel elősegítve a gördülékeny időpont-egyeztetést. A tanárok személyes profilja részletes információkat tartalmaz szakterületükről, óradíjukról és oktatási módszerükről. A rendszer célja, hogy gyors, biztonságos és célzott keresést tegyen lehetővé a tanulni vágyók számára, ezzel elősegítve a tudásmegosztást és a személyre szabott tanulási lehetőségeket.

## Rendszerkövetelmények

### Hardver követelmények

Az alkalmazás futtatásához szükséges minimális és ajánlott hardverkövetelmények az alábbiak szerint alakulnak. A program bármely olyan eszközön használható, amely képes Windows 10 vagy újabb operációs rendszert futtatni.

**Minimális rendszerkövetelmények:**

* **Operációs rendszer:** Windows 10 (64-bit)
* **Processzor:** Intel Core i3 vagy AMD Ryzen 3
* **Memória:** 4 GB RAM
* **Tárhely:** Legalább 1 GB szabad lemezterület
* **Képernyőfelbontás:** Minimum 1280x720

**Ajánlott rendszerkövetelmények a zavartalan működéshez:**

* **Operációs rendszer:** Windows 10 vagy újabb (64-bit)
* **Processzor:** Intel Core i5 vagy AMD Ryzen 5 és újabb
* **Memória:** Legalább 8 GB RAM
* **Tárhely:** SSD meghajtó, legalább 5 GB szabad hellyel
* **Képernyőfelbontás:** Full HD (1920x1080) vagy magasabb

### Szoftver követelmények

Az alkalmazás futtatásához szükséges szoftveres környezet az alábbi követelményeket foglalja magában, platformonkénti bontásban.

**Asztali alkalmazás:**

* Telepített **.NET Framework 4.7.2** vagy újabb verzió szükséges
* Operációs rendszer: **Windows 10** vagy újabb

**Webes alkalmazás:**

* Bármely modern böngésző, amely támogatja a **HTML5** szabványt
* Támogatott operációs rendszerek:
  + **Windows 10 vagy újabb**
  + **Linux** (Ubuntu 20.04 vagy újabb)
  + **macOS** 10.15 Catalina vagy újabb

**Szükséges szoftverek a fejlesztéshez vagy futtatáshoz:**

* **Böngésző:** Google Chrome vagy Mozilla Firefox (mindig a legfrissebb verzió ajánlott)
* **Node.js:** 16.x vagy újabb verzió
* **MySQL szerver:** 8.x verzió
* **Git** verziókezelő (opcionálisan, fejlesztéshez)
* **.NET futtatókörnyezet**, ha a háttérfolyamatok C# nyelven íródtak

## A program telepítése

A telepítőcsomag elérhető a SetupTanarKereso/Release mappában, amely két fájlt tartalmaz:

* setup.exe – önálló telepítő
* SetupTanarKereso.msi – Microsoft Installer csomag

Mindkét fájl használható a Windows operációs rendszer alatti telepítéshez.

#### Asztali alkalmazás telepítése

1. Futtassa a SetupTanarKereso.msi fájlt (vagy setup.exe-t), majd kövesse az alábbi lépéseket:
   * Kattintson a **„Next”** gombra. A képen szöveg, elektronika, képernyőkép, szoftver látható

     Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.
   * Válasszon telepítési helyet, vagy hagyja meg az alapértelmezettet.
   * Kattintson az **„Install”** gombra a telepítés indításához.
   * A „Close” gombbal zárd le a folyamatot.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

1. A telepítés után:

* Futtassa az adatbázist létrehozó magantanar.sql fájlt a MySQL rendszerben.
* Ellenőrizze a vizsga.exe.config fájlban az adatbázis csatlakozási adatokat (szerver, port, adatbázisnév, felhasználónév, jelszó).
* Győződjön meg róla, hogy a port helyesen van beállítva (3306 a MySQL szervernek megfelelően).

3. A program indítása

* Indítási lehetőségek:
  + A Start menüből az „Okos Magántanár Kereső” menüpont alatt.
  + Az asztalon megjelenő parancsikonnal (ha létrejött a telepítés során).

A képen szöveg, rajzfilm, poszter, Grafikus tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A képen szöveg, képernyőkép, ember, személy látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

#### Webes alkalmazás telepítése

A webes alkalmazás két részből áll: a backend és a frontend komponensekből.

**Előfeltételek:**

* Telepített Node.js (verzió 16.x vagy újabb)
* Telepített MySQL szerver (verzió 8.x)

**Backend telepítése:**

1. Navigáljon a BACKEND mappába:

cd BACKEND

1. Telepítse a szükséges csomagokat:

npm i

1. Indítsa el a szervert:

node server.js

A szerver alapértelmezés szerint a http://localhost:3000 címen fut.

**Frontend telepítése:**

1. Navigáljon a frontend mappába:

cd frontend

1. Telepítse a szükséges csomagokat:

npm i

1. Indítsa el a fejlesztői szervert:

npm run dev

A frontend alapértelmezés szerint a http://localhost:5173 címen érhető el.

**Alkalmazás elérése:**

Nyissa meg a böngészőt, és navigáljon a http://localhost:5173 címre a frontend megtekintéséhez.

Győződjön meg róla, hogy a backend szerver is fut a http://localhost:3000 címen.

**3. Mire figyeljünk a telepítés során**

* **Rendszergazdai jogosultság:**
  + Ha a telepítő nem indul el, kattints jobb egérgombbal, és válaszd a „Futtatás rendszergazdaként” opciót.
* **.NET-keretrendszer:**
  + A program működéséhez szükséges lehet a Microsoft .NET Framework. Ha nincs telepítve, a hivatalos oldalról letölthető.
* **Tűzfal vagy vírusirtó figyelmeztetés:**
  + Előfordulhat, hogy a vírusirtó figyelmeztet a telepítő futtatására. Ez normális, ha nem aláírt telepítőről van szó.

**4. Hibák és megoldások**

|  |  |  |
| --- | --- | --- |
| **Hiba** | **Ok** | **Megoldás** |
| Nem indul el a telepítő | Jogosultság hiánya | Futtatás rendszergazdaként |
| A program el sem indul | Hiányzó .NET-futtatókörnyezet | A .NET telepítése a Microsoft oldaláról |
| Üres képernyő induláskor | Adatbáziskapcsolat hiba | Fejlesztő felkeresése |

## A program használatának a részletes leírása

**Webes alkalmazás használata**

**1. Kezdőoldal**

A felhasználó elsőként a kezdőoldallal találkozik, amely egy rövid ismertetőt tartalmaz a weboldal céljáról: tanulni vágyó diákok és magántanárok összekapcsolása. A fejlécben található menüpontok segítségével elérhetők a „Bejelentkezés” és „Regisztráció” oldalak. Az oldal reszponzív kialakítású, így mobileszközön is teljes funkcionalitással működik.

**2. Regisztráció**

A regisztráció során a felhasználó kiválaszthatja, hogy diák vagy tanár szerepkörben kíván-e regisztrálni. A kiválasztott szerepkör alapján jelennek meg a kötelező mezők:

* **Diák mezők**: név, e-mail cím, jelszó

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

* **Tanár mezők**: név, cím (irányítószám, város, utca, házszám), e-mail, jelszó, telefonszám, bemutatkozás, díjszabás, tantárgy(ak), bankszámlaszám, adószám, IBAN

A rendszer ellenőrzi az adatok teljességét és az e-mail cím egyediségét. Sikeres regisztráció után a felhasználó visszajelzést kap, és automatikusan átirányításra kerül a bejelentkezési oldalra.

**3. Bejelentkezés**

A regisztrált felhasználó e-mail cím és jelszó megadásával léphet be. A rendszer az azonosító alapján felismeri a szerepkört (diák vagy tanár), és a megfelelő felhasználói főoldalra irányítja. Hibás adatok esetén hibaüzenet jelenik meg: „Hiba történt a belépés során!”.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**4. Felhasználói irányítópult**

A bejelentkezés után a felhasználók saját, szerepkörhöz igazított vezérlőpultot kapnak:

* **Diákok**:
  + Tanárkereső használata
  + Üzenetek kezelése
  + Saját foglalások megtekintése
  + Profiladatok módosítása
* **Tanárok**:
  + Üzenetek megtekintése és válaszadás
  + Beérkező foglalások kezelése
  + Profil szerkesztése (bemutatkozás, tantárgyak, óradíj)
  + Elérhetőség és naptár karbantartása

**5. Tanárkereső**

A tanárkereső egy szűrőfelületet kínál, ahol a diákok kereshetnek tantárgy, város, óradíj és oktatási forma szerint. A találati listában a tanárok rövid profilkártyákon jelennek meg, ahol láthatók a fő adatok és egy „Megtekintés” gomb, amely a részletes tanárprofilra vezet.

**6. Tanárprofil**

A tanár profiloldalán részletes bemutatkozás található, tantárgyak listája, óradíj, címadatok, elérhetőségek, valamint értékelések. Innen lehetőség van közvetlen üzenetküldésre, illetve időpontfoglalásra is. Hitelesített tanároknál külön „Ellenőrzött tanár” ikon jelenik meg.

**7. Üzenetküldés**

A diák üzenetet írhat a tanárnak, amit a rendszer az adatbázisba ment. Az üzeneteket időbélyeggel, feladó és címzett azonosítóval együtt tárolja a rendszer. A tanár bejelentkezés után látja az üzeneteit, és válaszolhat is rájuk. Az üzenetlista egy chathez hasonló megjelenítést kapott.

**8. Időpontfoglalás**

A diák a tanár szabad időpontjai alapján küldhet foglalási kérelmet. A rendszer ellenőrzi, hogy az adott időintervallum szabad-e. A foglalás az események táblába kerül, és e-mail értesítés is generálható (ha be van kapcsolva).

**9. Profilkezelés**

A felhasználó bármikor módosíthatja az adatait. A jelszó, bemutatkozás, telefonszám és elérhetőségi adatok külön felületen frissíthetők. Az inaktívvá tett profilokat az adminisztrátor módosíthatja.

**10. Kijelentkezés**

A kijelentkezés a jobb felső sarokban található „Kijelentkezés” gombbal történik. A session lezárul, és a kezdőoldalra kerül vissza a felhasználó.

**Asztali alkalmazás használata**

Az asztali alkalmazás célja az adminisztrációs funkciók gyors és egyszerű kezelése. A program .NET alapokon készült, és elsősorban a regisztrált felhasználók (tanárok és diákok) valamint a tantárgyak adatainak jóváhagyására, szerkesztésére és karbantartására szolgál.

**1. Indítás és főmenü**

A program indításakor megnyílik egy egyszerű kezdőképernyő, amely három fő kategóriát kínál:

* **Diákok**
* **Tanárok**
* **Tantárgyak**

A felhasználó a kategóriák egyikére kattintva tovább tud lépni a kiválasztott entitás kezelésére szolgáló felületre. Minden egyes kategória külön ablakban (formban) nyílik meg.

**2. Jóváhagyási felületek működése**

A „Diákok” és „Tanárok” kezelőfelületek két külön listát tartalmaznak:

* **Elfogadottak**: azok a felhasználók, akik már aktívak a rendszerben.
* **Várakozók**: új regisztrációk, amelyeket az adminnak jóvá kell hagynia vagy el kell utasítania.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

A „Várakozók” listában egy adott névre kattintva egy új form nyílik meg, ahol a kiválasztott felhasználó adatai részletesen megtekinthetők. Itt az adminisztrátor két gomb segítségével választhat:

* **Elfogadás** – aktiválja a felhasználót, aki azonnal átkerül az „Elfogadottak” listába.
* **Törlés** – eltávolítja a felhasználót, de csak akkor működik, ha az adott rekord nem kapcsolódik más táblához (pl. nincs hozzá tartozó foglalás vagy üzenet).

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az elfogadás vagy törlés után a lista automatikusan frissül, nincs szükség a program manuális újraindítására vagy kézi frissítésre.

**3. Tantárgyak kezelése**

A „Tantárgyak” opció szintén új ablakban nyílik meg. Itt nem szerepel külön elfogadólista, mivel a tantárgyak felvitele manuálisan történik. Az adminisztrátor megadhatja a tantárgy nevét és az óradíjat, majd az „Új hozzáadása” gombra kattintva az adatokat a rendszer menti az adatbázisba.

Jelenleg nincs lehetőség tantárgyak törlésére az alkalmazásból, és nincs automatikus kapcsolás tanárhoz – ezt külön modulon keresztül kell megoldani (vagy SQL-ben manuálisan).

A képen szöveg, képernyőkép, szoftver, képernyő látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**4. Megjelenés testreszabása**

Az asztali alkalmazás minden felületén elérhető egy háttérszín-választó csúszka, amellyel a felhasználó kiválaszthatja, hogy világos (fehér) vagy sötét (fekete) háttérrel szeretné használni az adott ablakot. Ez segít a vizuális kényelem fokozásában, különösen gyenge fényviszonyok között.

Fontos megjegyezni, hogy a színbeállítás csak az adott formra vonatkozik, és nem kerül elmentésre. Új ablak megnyitásakor mindig az alapértelmezett (fehér) háttér jelenik meg.

**5. GitHub elérés**

Az ablak alsó részén elhelyezett GitHub logóra kattintva a program automatikusan megnyitja a fejlesztés során használt GitHub repository-t a felhasználó alapértelmezett böngészőjében. Ez lehetőséget biztosít a forráskód megtekintésére, vagy a projekt további követésére.

**6. Összegzés**

Az asztali alkalmazás a webes felület adminisztratív kiegészítéseként szolgál. Előnye, hogy a jóváhagyások, ellenőrzések és módosítások gyorsan, külön ablakokon keresztül, célzottan végezhetők el. Egyszerűsége és funkcionalitása révén hasznos eszköz az adatkezeléshez és az oktatási platform fenntartásához.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# Fejlesztői dokumentáció

## Az alkalmazott fejlesztői eszközök

A program fejlesztéséhez és dokumentálásához többféle szoftvereszközt és modult használtam fel, amelyek az alábbiakban kerülnek részletezésre:

**Fejlesztői környezetek és nyelvek:**

* **Visual Studio 2019 és 2022:** fő fejlesztői környezetként szolgáltak a C#-os alkalmazás fejlesztéséhez.
* **Visual Studio Code:** frontend és backend kódok szerkesztésére (JavaScript, HTML, CSS).
* **Programozási nyelvek:** JavaScript (frontend és backend), HTML, CSS, C# (asztali alkalmazás rész).
* **Node.js és Express.js:** a backend REST API működtetéséhez.

**Frontend eszközök:**

* **Bootstrap:** a webes felület reszponzív kialakításához.
* **JavaScript natív DOM-kezelés:** a dinamizmus és interakciók kezelésére.

**Adatbázis és kapcsolódó eszközök:**

* **MySQL / MariaDB:** a projekt relációs adatbázis-rendszere.
* **phpMyAdmin:** grafikus kezelőfelület az adatbázis karbantartásához.
* **MySQL Workbench:** lekérdezések írásához és adatmodell vizualizálásához.

**Verziókezelés:**

* **Git:** verziókövetéshez és fejlesztési állapot mentéséhez.
* **GitHub:** a projekt távoli tárhelye és együttműködéshez használt platform.

**Dokumentációkészítéshez használt eszközök:**

* **Microsoft Word:** a vizsgaremek dokumentáció megszerkesztéséhez.
* **draw.io:** az adatmodell és UML-diagramok elkészítéséhez.
* **ChatGPT (OpenAI):** képek szerkesztési ötleteinek kidolgozásához, valamint technikai leírások, szövegek megfogalmazásának támogatásához.

**Használt külső Node modulok:**

* **body-parser:** az érkező HTTP kérések értelmezésére (https://www.npmjs.com/package/body-parser).
* **cors:** CORS-kezeléshez, külső forrásból történő hozzáférések engedélyezésére (https://www.npmjs.com/package/cors).

A fenti szoftvereszközök mind jogtiszta forrásból származnak, és az oktatási projekt során történő felhasználásuk jogszerű. A használt Node modulok nyílt forráskódúak, a projekt céljainak megfelelő licencelésűek.

## Adatmodell leírása

A Magántanár Kereső Weboldal egy relációs adatbázist használ, amely a MariaDB/MySQL adatbázis-kezelő rendszerre épül. Az adatbázis főbb entitásai: diák, tanár, tantárgy, esemény (időpontfoglalás), tanár-tantárgy kapcsoló tábla és üzenetek. Az alábbiakban részletesen bemutatásra kerülnek a táblák szerkezetei és kapcsolataik.

### Táblák

#### diak tábla

diak\_id (int, PK): A diák azonosítója.

d\_nev (varchar): A diák neve.

email (varchar): A diák e-mail címe.

jelszo (varchar): Titkosított jelszó.

aktiv (enum): A fiók aktív állapota ('true' / 'false').

#### tanar tábla

tanar\_id (int, PK): A tanár azonosítója.

t\_nev (varchar): A tanár neve.

iranyitoszam, varos, utca, hazszam (varchar): Címadatok.

email (varchar): E-mail cím.

jelszo (varchar): Titkosított jelszó.

telefonszam (varchar): Kapcsolattartási szám.

dijszabas (varchar): Díjszabás (Ft/óra).

bemutatkozas (text): Rövid leírás.

bszamla, adoszam, IBAN (int/varchar): Pénzügyi adatok.

aktiv (enum): A tanár profiljának státusza.

#### tantargyak tábla

tantargy\_id (int, PK): A tantárgy azonosítója.

tantargy\_nev (varchar): Tantárgy neve.

oradij (int): Óradíj.

#### tanartantargy tábla (kapcsoló tábla)

tanar\_id (int, FK): Hivatkozás a tanar táblára.

tantargy\_id (int, FK): Hivatkozás a tantargyak táblára.

#### uzenetek tábla

uzenetek\_id (int, PK): Üzenet azonosító.

datum (date): Üzenet küldésének dátuma.

diak\_id, tanar\_id (int, FK): Felhasználók közötti kapcsolat.

szoveg (text): Az üzenet tartalma.

#### esemeny tábla (időpontfoglalás)

esemenyek\_id (int, PK): Esemény azonosító.

kezd, veg (datetime): Óra kezdete és vége.

diak\_id, tanar\_id (int, FK): Résztvevők.

tantargy\_id (int, FK): Kapcsolódó tantárgy.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

### Kapcsolatok:

Egy tanár több tantárgyat is taníthat (tanartantargy: több-több kapcsolat).

Egy diák több tanárhoz is fordulhat, ugyanígy egy tanár is több diákkal dolgozhat együtt.

Az esemeny tábla biztosítja az időpontfoglalásokat diákok és tanárok között, és kapcsolódik egy adott tantárgyhoz is.

Az uzenetek tábla rögzíti a kommunikációt tanárok és diákok között.

Adatbázis-kezelés: Az adatbázisban minden elsődleges kulcs auto\_increment értékkel van ellátva, így biztosítva az egyedi azonosítást. Az idegen kulcsok (foreign key) megakadályozzák az inkonzisztens adatbevitel lehetőségét és fenntartják a táblák közötti integritást.

### Jellemzők:

Az adatbázis UTF-8 karakterkódolást használ, így a magyar ékezetes karakterek is megfelelően tárolhatók.

Az aktív mezők lehetővé teszik a fiókok érvényességének egyszerű követését.

A struktúra jól skálázható, új tantárgyak, tanárok, diákok vagy funkciók (pl. értékelések) könnyen hozzáadhatók.

Az adatmodell így biztosítja a weboldal hatékony, gyors és konzisztens működését, miközben támogatja a funkcionális és üzleti logikát.

## Részletes feladatspecifikáció, algoritmusok

Az alkalmazás működését több fő modul és funkció biztosítja. Ezek megvalósításához különböző algoritmusokat és logikai struktúrákat alkalmaztunk. Az alábbiakban részletesen bemutatásra kerül néhány kulcsfontosságú folyamat.

**1. Felhasználó regisztráció**

A regisztrációs űrlapon a felhasználó megadja az e-mail címét, jelszavát, valamint a szükséges személyes adatokat (név, cím, szerepkör: diák vagy tanár). A rendszer ellenőrzi az e-mail egyediségét, majd az adatokat menti az adatbázis megfelelő táblájába (diak vagy tanar).

**Algoritmus:**

***ha e-mail már létezik az adatbázisban:***

hibaüzenet megjelenítése

***különben:***

adatok mentése a kiválasztott szerepkör táblába

sikeres regisztráció üzenet

**2. Bejelentkezés**

A felhasználó megadja az e-mail címét és jelszavát. A rendszer lekérdezi az adatbázist, és ha egyezés van, beállítja a session-t vagy átirányít a megfelelő kezdőfelületre.

**Algoritmus:**

lekérdezés az email és jelszó alapján

ha egyezés:

session beállítása

átirányítás a főoldalra

különben:

hibaüzenet

**3. Tanárkeresés szűrőkkel**

A diák tantárgy, város, ár és értékelés szerint szűrheti a tanárokat. A backend oldalon egy SQL lekérdezés készül a megadott paraméterek szerint.

**SQL példa:**

SELECT \* FROM tanar

WHERE tantargy\_id = ? AND varos = ? AND dijszabas <= ?;

**4. Üzenetküldés**

A diák kiválaszt egy tanárt, és szöveges üzenetet ír neki. Az üzenet tartalma, dátuma, a feladó (diák) és a címzett (tanár) azonosítója bekerül az "uzenetek" táblába.

**Adatbázisművelet:**

INSERT INTO uzenetek (datum, diak\_id, tanar\_id, szoveg)

VALUES (CURDATE(), ?, ?, ?);

**5. Időpontfoglalás**

A tanár és diák megbeszélt időpont alapján foglalást rögzíthetnek, amely az "esemeny" táblában tárolódik kezdési és befejezési időponttal.

**Struktogram vagy UML aktivitásdiagram javasolt ehhez:**

* Ellenőrzés: az adott időintervallum szabad-e
* Foglalás mentése

**6. Adminisztratív jogosultságok (opcionális modul)**

Az admin a felületen keresztül láthatja az összes tanárt és diákot, valamint jogosult lehet tanárok hitelesítésére, profil aktiválására vagy törlésére.

**Példa logika:**

admin belép → listázás → státusz frissítése (aktiv → true/false)

A program belső logikája moduláris, így az egyes funkciók jól elkülöníthetők, bővíthetők és karbantarthatók. Az alkalmazott algoritmusok egyszerűek, de biztosítják a megbízható működést.

## Tesztelési dokumentáció

A fejlesztés során különböző tesztelési módszereket alkalmaztunk (fekete doboz és fehér doboz tesztelés), hogy megbizonyosodjunk a rendszer helyes működéséről. Az alábbiakban bemutatunk három részletes tesztesetet, azok eredményeit és az esetleges hibákat.

**Teszteset 1 – Normál használat (fekete doboz teszt)**

**Esemény:** Diák regisztrációja és bejelentkezése

* **Bemenet:** Név, e-mail, jelszó megadása, szerepkör kiválasztása (diák)
* **Várt eredmény:** Sikeres regisztráció, majd belépés után átirányítás a főoldalra
* **Tényleges eredmény:** A rendszer visszaigazoló üzenetet küldött, és sikeresen bejelentkezett a diák
* **Üzenet:** „Diák regisztráció sikeres!”
* **Teendő:** Nincs további lépés, megfelelő működés

A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Teszteset 2 – Extrém eset: helytelen bejelentkezés (fekete doboz)**

**Esemény:** Felhasználó hibás jelszót ad meg bejelentkezéskor

* **Bemenet:** Helyes e-mail cím, helytelen jelszó
* **Várt eredmény:** Rendszer ne engedje be a felhasználót
* **Tényleges eredmény:** „Hibás e-mail cím vagy jelszó” hibaüzenet megjelenítése
* **Üzenet:** „Hibás bejelentkezési adatok”
* **Teendő:** Újrapróbálkozás helyes adatokkal

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Teszteset 3 – Tanár törlése admin felületen (fehér doboz teszt)**

**Esemény:**

**Bemenet:** Admin kiválasztja a tanárt a listából, majd a „Törlés” gombra kattint

**Feltétel:** A tanár azonosítója nem szerepel idegen kulcsként más adatbázis-táblákban (pl. kurzusok, órarend, hozzárendelések)

**Várt eredmény:**

A tanár rekordja törlésre kerül az adatbázisból

**Megjelenik a visszajelző üzenet:** „Tanár sikeresen törölve”

**Tényleges eredmény:**

A rendszer ellenőrizte az idegen kulcs kapcsolatokat

A tanár törlődött, mivel nem volt hivatkozás más táblában

**Visszajelzés:** „Tanár sikeresen törölve”

**Teendő:** Nincs – megfelelő működés

**Feltárt hibák a tesztelés során:**

1. A regisztrációs űrlapnál nem ellenőrizte az üres mezőket – javítva lett (klienst oldali validációval)
2. Üzenetküldéskor nem frissült automatikusan az oldal – frissítést követően megoldva
3. A tanár törlése csak abban az esetben működik, ha nincs összekapcsolva mással.

**Összegzés**

A rendszer a tesztek alapján stabilan működik a megadott felhasználói tevékenységek mellett. A fekete doboz tesztelés során az inputokra adott válaszokat ellenőriztük, míg a fehér doboz módszert a kommunikáció és adatrögzítés folyamatának nyomon követéséhez alkalmaztuk. Az észlelt hibákat dokumentáltuk és kijavítottuk, a rendszer így megbízhatóan teljesíti a követelményeket.

# 

# Továbbfejlesztési lehetőségek

A jövőbeni fejlesztések során elsődleges cél a felhasználói élmény további javítása. A jelenlegi rendszer mobilbarát, de a reszponzív megjelenés továbbfejlesztésével még rugalmasabbá és minden képernyőméreten ergonomikusabbá tehető az oldal. A fizetési rendszer integrálása lehetővé tenné, hogy a tanárok online óradíjat kérjenek el a diákoktól, ezáltal professzionálisabb platformmá téve az alkalmazást.

Fontos továbbá egy e-mail értesítési rendszer kialakítása, amely automatikusan figyelmeztetést küld a felhasználóknak új üzenetek vagy foglalások esetén. Az admin felület bővítése statisztikai kimutatásokkal, például a regisztrált tanárok és diákok számával, foglalások gyakoriságával, szintén hozzájárulhat a platform hatékonyabb működéséhez.

Egy jövőbeni célkitűzés egy mobilalkalmazás (app) fejlesztése, amely natív élményt biztosítana a felhasználóknak. Az alkalmazásba integrálható lenne egy naptár funkció is, amely lehetővé teszi, hogy a tanárok és diákok időpontokat küldjenek egymásnak, így egyszerűsítve az órák egyeztetését. Emellett javasolt egy tanár-hitelesítési rendszer bevezetése, ahol az adminisztrátor ellenőrzött tanárokat jelölhet meg egy vizuális ikon segítségével, ezzel is növelve a bizalmat a felhasználók között.

# Önértékelés

A vizsgaremek projekt elkészítése során rengeteget tanultunk, fejlődtünk és tapasztalatot szereztünk az együttműködés, a tervezés és a fejlesztés terén. A munka során igyekeztünk a feladatokat megosztani egymás között, figyelembe véve az erősségeinket és érdeklődési körünket.

Angyalka főként az asztali alkalmazás fejlesztéséért volt felelős, ahol sokat fejlődött a C# programozásban, a formok kezelésében, az adatkapcsolatok kialakításában és a telepítőcsomag készítésében. Komoly kihívást jelentett számára a különböző formok közti adatátvitel, valamint az adatbázis-kezelés asztali környezetben, de ezeket önállóan, kitartó munkával sikerült megoldania.

Peti a weboldal fejlesztésére koncentrált, amely magába foglalta a backend és frontend logika elkészítését Node.js, JavaScript és HTML segítségével. A szerveroldali működés megtervezése, a kapcsolódási hibák kezelése és a böngészőben való megjelenítés sok tanulással és próbálkozással járt. A projekt során mégjobban elsajátította az alapvető webszerkesztési és szerverindítási ismereteket is.

Az adatbázis megtervezését, a struktúrák kialakítását, valamint a dokumentáció és a PPT elkészítését közösen végeztük. Igyekeztünk úgy dolgozni, hogy mindenki hozzátegyen a közös munkához, és segítsük egymást, ha valaki elakadt. A munka során gyakoriak voltak a közös ötletelések, új megoldások keresése és egymás támogatása, amit a sikeres együttműködés alapjának tartunk.

Összességében a projekt kivitelezése során nemcsak szakmai tudásunkban fejlődtünk, hanem az együttműködésben, felelősségvállalásban és problémamegoldásban is. Úgy érezzük, hogy egy hasznos, működő alkalmazást hoztunk létre, amelyre büszkék lehetünk, és amely kiváló alapot nyújt további tanulmányainkhoz és szakmai pályafutásunkhoz.

# Felhasznált irodalom

* <https://nodejs.org/>
* <https://getbootstrap.com/>
* <https://expressjs.com/>
* <https://www.mysql.com/>
* <https://github.com/borossandor27/orai_anyagok>
* <https://www.w3schools.com/>
* <https://merlinvizsga.hu/>
* <https://github.com/P3T8/Vizsgaremek>
* <https://www.youtube.com/watch?v=LWoE7P4RHFM>
* <https://www.reddit.com/r/csharp/>

# 

# Ábrajegyzék

* + - 1. Asztali telepítő program next gombja.
      2. Asztali telepítő program befejező gombja.
      3. Asztali app, asztalon megjelenő parancsikonja.
      4. Asztali app megnyíló főoldala.
      5. Webes felületnél diák regisztráció.
      6. Webes felületnél hibás bejelentkezés.
      7. Admin felületnél elfogadott és jóváhagyásra váró diákok nevei.
      8. Admin felületen tanár jóváhagyása/ elutasítása.
      9. Admin felületen új tantárgy hozzáadása.
      10. Admin felületen bezárás gombnál megjelenő exit alert.
      11. Adatbázis kapcsolat.
      12. Sikeres diák regisztráció.
      13. Sikertelen belépés.