Okos Magántanár Kereső

BGSZC Logisztikai és kereskedelmi technikum és szakképző iskola

Vizsgraremek projekt 13. évfolyam Szoftverfejlesztés és Tesztelés

BAracskai Angyalka, MAráczi Péter

2025

1 bevezetés

* 1. feladat leírás
  2. felhasznált ismeretek
  3. felhasznált szoftverek

1. felhasználói dokumentáció
   1. program általános specifikációja
   2. rendszerkövetelmények
      1. hardver követelmények
      2. szoftver követelmények
   3. program telepítése
   4. program használatának leírása
      1. asztali leírás
      2. webes leírás
2. fejlesztői dokumentáció
   1. alkalmazott fejlesztői eszközök
   2. adatmodell leírása

3.2.1.1 diak tábla

3.2.1.2 tanar tábla

3.2.1.3 tantargyak tábla

3.2.1.4 tanartantargy tábla

3.2.1.5 uzenetek tábla

3.2.1.6 esemeny tábla

3.2.3 kapcsolatok

3.2.4 jellemzők

* 1. részletes feladatspecifikáció, algoritmusok
  2. tesztelési dokumentáció

1. továbbfejlesztési lehetőségek
2. felhasznált irodalom
3. ábrajegyzék

# 1 Bevezetés

## 1.1 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, üzenetküldésre, értékelésre, valamint profilkezelésre mind diákok, mind tanárok számára.

## 1.2 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
* bcrypt algoritmus a jelszavak titkosításához
* SQL injekció elleni védelem és adatbázis biztonsági szabályok alkalmazása

## 1.3 A felhasznált szoftverek

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

# 

# 2 Felhasználói dokumentáció

## 2.1 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.

## 2.2 Rendszerkövetelmények

### 2.2.1 Hardver követelmények

### Minimum követelmény:

### Processzor: Intel Core i3 vagy AMD Ryzen 3

### Memória (RAM): 4 GB

### Tárhely: minimum 1 GB szabad lemezterület

### Képernyő: 1280x720 vagy jobb felbontás

### Ajánlott követelmény:

### Processzor: Intel Core i5 vagy AMD Ryzen 5 és újabb

### Memória (RAM): legalább 8 GB

### Tárhely: SSD meghajtó, 5 GB szabad hely

### Képernyő: Full HD (1920x1080) felbontás vagy jobb

### 2.2.2 Szoftver követelmények

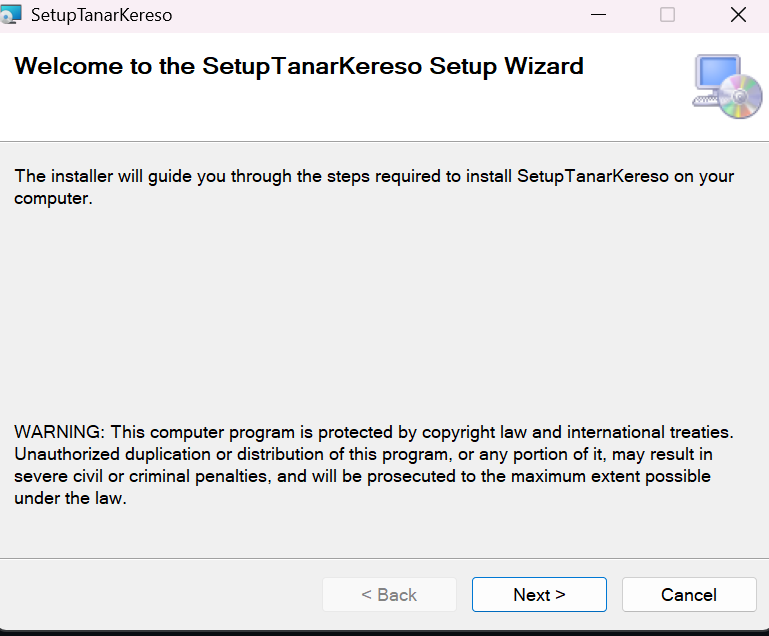
* Operációs rendszer:
  + Windows 10 vagy újabb
  + Linux (Ubuntu 20.04+)
  + macOS 10.15 Catalina vagy újabb
* Szükséges szoftverek:
  + Böngésző: Google Chrome / Mozilla Firefox (legfrissebb verzió)
  + Node.js (16.x vagy újabb verzió)
  + MySQL szerver (8.x verzió)
  + Git verziókezelő (opcionális, fejlesztéshez)
  + .NET futtatókörnyezet (ha a backend C#-os elemeket is tartalmaz)### 2.3 A program telepítése

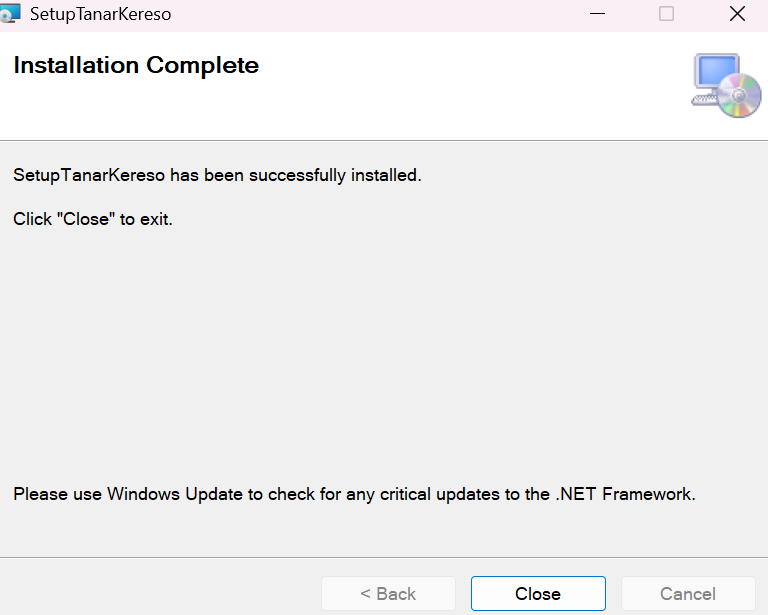
## 2.3 A program telepítése

A program telepítőcsomagja elérhető a SetupTanarKereso/Release mappában, ahol két fájl található: egy .exe (setup.exe) és egy .msi(SetupTanarKereso.msi). Bármelyik használható a telepítéshez Windows rendszeren.

**1. Telepítés lépései**

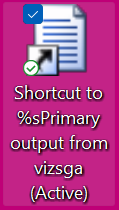
* Navigálj a Vizsgaremek/SetupTanarKereso/Release mappába.
* Kattints duplán a TanarKeresoSetup.exe fájlra a telepítés indításához.
  + Ha ez nem működik, a .msi fájl is használható ugyanígy.
* A megjelenő telepítőablakban:
  + Kattints a „Next” gombra.
  + Válaszd ki a telepítési mappát, vagy hagyd az alapértelmezett beállítást.
  + Kattints az „Install” gombra a telepítés indításához.
  + A „Finish” gombbal zárd le a folyamatot.



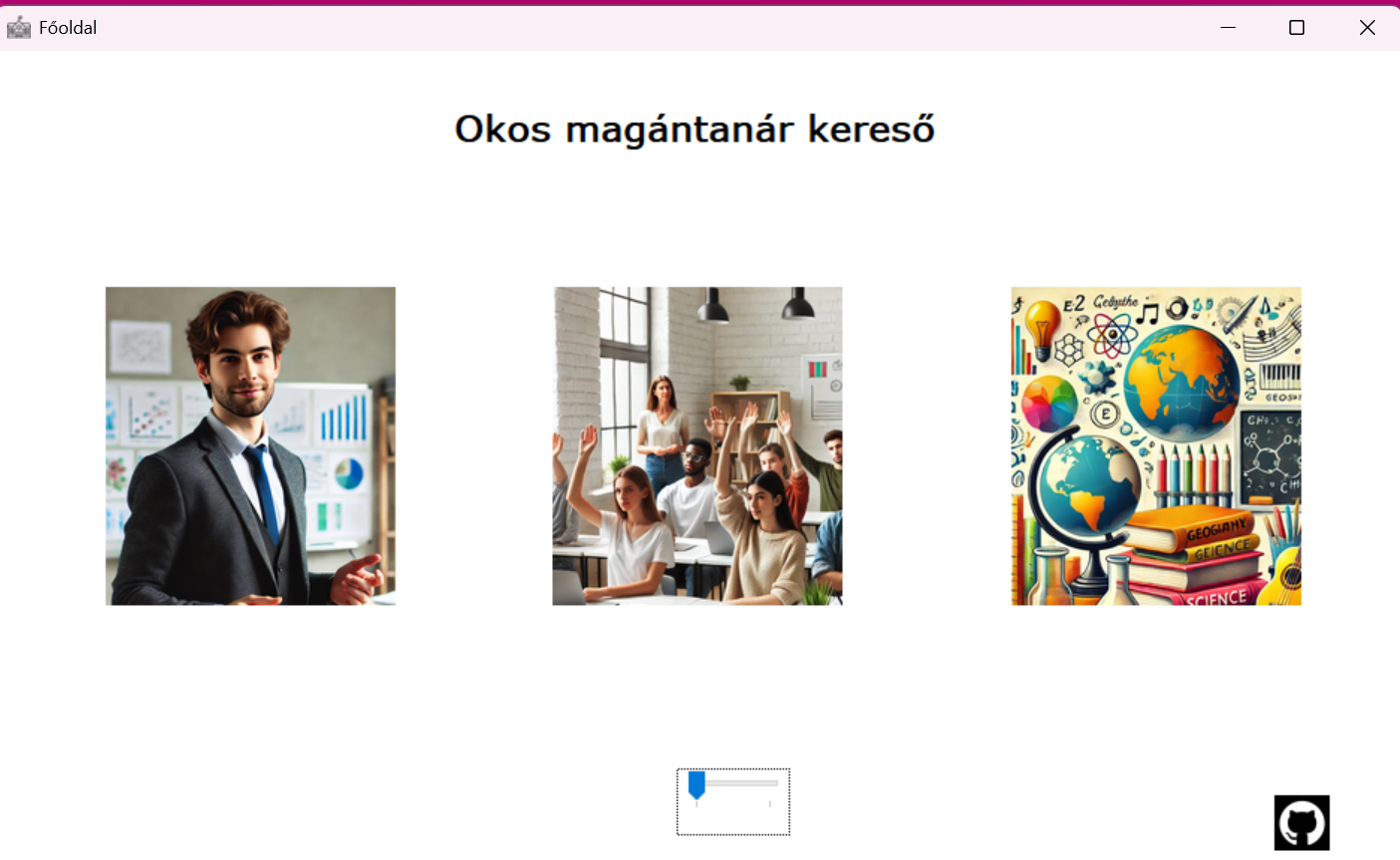


**2. A program indítása**

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



* Az első indítás után megjelenik a kezdőfelület:
  + Regisztráció diák vagy tanár típusban.
  + Bejelentkezés meglévő fiókkal.



**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 |

## 2.4 A program használatának a részletes leírása

Mindenre kiterjedő, részletes leírás a program használatáról. Alapszabályok:

* Amit leprogramoztál, azt a dokumentációban írd is le, ne legyenek eltitkolt funkciók.
* Minden pontosan, „szájbarágósan” legyen leírva. A dokumentáció alapján a teljesen kezdő, vagy laikus felhasználóknak is használniuk kell tudni a programot.
* A stílus legyen pontos és közérthető, vedd figyelembe, hogy a felhasználói dokumentáció nem szakembereknek készül.
* Ugyanakkor kerüld a laza stílust: rövidítések, smilie-k, szleng kizárva.
* Alkalmazz ábrákat, screenshot-okat , de a ne legyen túlzott a képek aránya a szöveghez képest. Kb. 2-3 oldalanként egy ábra megfelelő.

Ajánlott terjedelem: 10-15 oldal, ábrákkal együtt.

# 3 Fejlesztői dokumentáció

## 3.1 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.

## 3.2 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.

### 3.2.1Táblák

#### 3.2.1.1 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').

#### 3.2.1.2 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.

#### 3.2.1.3 tantargyak tábla

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

tantargy\_nev (varchar): Tantárgy neve.

oradij (int): Óradíj.

#### 3.2.1.4 tanartantargy tábla (kapcsoló tábla)

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

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

#### 3.2.1.5 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.

#### 3.2.1.6 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.

### 3.2.2 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.

### 3.2.3 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.

## 3.3 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.

## 3.4 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:** „Sikeres regisztráció!”
* **Teendő:** Nincs további lépés, megfelelő működés

**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

**Teszteset 3 – Teljes kommunikációs teszt (fehér doboz)**

**Esemény:** Diák üzenetet küld tanárnak, majd tanár válaszol

* **Bemenet:** Diák kiválasztja a tanárt, üzenetet ír, tanár belép, és válaszol
* **Várt eredmény:** Mindkét üzenet megjelenik a beszélgetés ablakban
* **Tényleges eredmény:** Az üzenetek sorrendben, időbélyeggel együtt megjelentek
* **Üzenet:** „Üzenet elküldve”
* **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 kereső szűrési feltételei nem működtek egyszerre – SQL lekérdezés módosításával javítva

**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.

# 4 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.

# 5 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>

# 6 Ábrajegyzék

[1. kép Ez a módosító ablak a programban 5](file:///C:\Users\barac\Downloads\ZD_minta.docx#_Toc63250394)