**Baranya Vármegyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

**Vizsgaremek**

Készítették: Németh Noel,

Lőrincz Noel,

Rapcsák János

**Pécs**

**2025**

**Baranya Vármegyei Szakképzési Centrum**

**Simonyi Károly Technikum és Szakképző Iskola**

*Szakma megnevezése: Szoftverfejlesztő és –tesztelő*

*A szakma azonosító száma: 5 0613 12 03*

**Vizsgaremek**

Aliasly

Készítették: Németh Noel,

Lőrincz Noel,

Rapcsák János

**Pécs**

**2025**

**2025**

**EREDETISÉGI NYILATKOZAT**

Alulírottak: Németh Noel, Lőrincz Noel, Rapcsák János

A Baranya Vármegyei SzC Simonyi Károly Technikum és Szakképző Iskola, Szoftverfejlesztő és tesztelő végzős tanulói, büntetőjogi és fegyelmi felelősségünk tudatában nyilatkozunk és aláírásunkkal igazoljuk, hogy a(z)

Aliasly

című vizsgaremek saját, önálló munkánk, az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul vesszük, hogy vizsgaremek esetén plágiumnak számít:

- szószerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;

- tartalmi idézet hivatkozás megjelölése nélkül;

- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentjük, hogy a plágium fogalmát megismertük, és tudomásul vesszük, hogy plágium esetén a vizsgaremekünk visszautasításra kerül.

## Tartalomjegyzék

## Bevezetés

### Projekt bevezető

#### 1.1 Megoldandó feladat és probléma

A modern vállalati környezetekben gyakori, hogy egy dolgozónak több céges fiókja is van. Ezek kezelése nehézkes, időigényes, és biztonsági kockázatot jelenthet, ha a jelszavak vagy hozzáférési jogok nem megfelelően vannak kezelve. A projekt célja egy olyan centralizált fiókkezelő alkalmazás létrehozása, amely segíti a rendszergazdákat a felhasználói adatok biztonságos és átlátható kezelésében.

#### 1.2 Miért ezt választottuk

A vállalati adatvédelem egyre nagyobb hangsúlyt kap, de a meglévő eszközök gyakran túlbonyolítottak vagy költségesek. Saját fejlesztéssel rugalmasan lehet alkalmazkodni a cégek egyedi igényeihez.

A csapatunk számára lehetőség volt a biztonsági protokollok (pl. titkosítás, szerepkör-alapú hozzáférés) és a skálázható architektúra tervezésének gyakorlati alkalmazására.

#### 1.3 Célunk

Egy egyszerű, intuitív felület biztosítása a rendszergazdáknak. Biztonság: AES-256 titkosítás és többfaktoros hitelesítés (MFA) alkalmazása. Felhasználóbarát felület: Intuitív kezelőfelület a rendszergazdák számára. Skálázhatóság: Több ezer felhasználó kezelése kis és közepes vállalatoknál.

### Fejlesztési folyamat és Tapasztalatok

#### 2.1 Mit tanultunk és új ismeretek

Közös döntéshozatal: Rájöttünk, hogy a különböző szakmai háttérrel rendelkező csapattagok (pl. backend vs. frontend fejlesztők) más szempontokat hangsúlyoznak. Például egy biztonsági funkció implementálása során a backend csapat a teljesítményre, míg a frontend a felhasználói élményre fókuszált. Rendszeres design sprint ülésekkel sikerült közös nevezőre jutni.

Feedback kultúra: A napi minimum 15 perces meetingek segítettek a blokkoló problémák gyors feloldásában.

Konfliktuskezelés: Megtanultuk, hogy a technikai nézeteltérések nem személyesek, hanem a projekt minőségét szolgálják.

### Csapat és Munkamegosztás

#### 3.1 Csapatszerepek

Projekt tervezés - Feladatok koordinálása: Németh Noel, Lőrincz Noel, Rapcsák János

Backend fejlesztés - Adatbázis és API fejlesztés: Németh Noel, Rapcsák János

Frontend fejlesztés - Felhasználói felület fejlesztése: Rapcsák János, Lőrincz Noel

Biztonság kivitelezése - Titkosítás és jogosultságok implementálása: Németh Noel

#### 3.2 Projektszervezési eszközök

Github: Verziókövetés és a kódbázis kezelése.

Discord: Kommunikáció és projekt fejlesztésének megbeszélése.

Excel táblázat: Projekt feladatok megtervezése és nyomon követése.

### Műszaki megvalósítás

#### 4.1 Architektúra

Backend: A szerveroldali logikát C# .NET 8.0 keretrendszerrel fejlesztjük.

Frontend: A kliensoldali felület Windows Presentation Foundation (WPF) keretrendszerrel készül. Data Binding és XAML használatával a dinamikus felületek kialakításához.

Adatbázis: Az adatok tárolására MySQL relációs adatbázist használunk, amely megbízható és nagy teljesítményű megoldást nyújt. A phpMyAdmin felület segítségével könnyedén kezelhetők az adatbázisok.

#### 4.2 Biztonsági intézkedések

Adattitkosítás: Az adatok védelmére AES-256 titkosítást használunk.

Hozzáférés-vezérlés: Mesterkulcsos belépőrendszer.

Naplózás: Minden fontos műveletről (pl. bejelentkezések, adatmódosítások).

### Következő lépések

Multi-Faktor Hitelesítés (MFA): A rendszergazdák számára kötelezővé tehető MFA bevezetése.

Integrációs Lehetőségek: Kapcsolódás vállalati HR rendszerekhez a felhasználói adatok automatikus szinkronizálásához.

Mobile Admin Felület: Alkalmazás mobilra, hogy a rendszergazdák akár távolról is kezelhessék a fiókokat.

Felhasználói Önkiszolgáló Portál: Jelszó-változtatás vagy jogosultságkérelmek önálló kezelése.

## Felhasználói dokumentáció

### A program általános specifikációja

### Rendszerkövetelmények

### A program telepítése

### A program használatának a részletes leírása

## Fejlesztői dokumentáció

### Témaválasztás indokolása

### Az alkalmazott fejlesztői eszközök

### Tervezési módszer

### Adatmodell leírása

### Részletes feladatspecifikáció algoritmusok

### Forráskód

### Tesztelési dokumentáció

### Továbbfejlesztési lehetőségek

## Összegzés

### Irodalomjegyzék, forrásmegjelölés