VÁCI SZAKKÉPZÉSI CENTRUM  
BORONKAY GYÖRGY  
MŰSZAKI TECHNIKUM ÉS GIMNÁZIUM

VIZSGAREMEK  
Freya’s Garden – Növény csere-bere

Nacsa Levente  
Szabolics András  
Tábor Tünde  
2024-2025.

VÁCI SZAKKÉPZÉSI CENTRUM  
BORONKAY GYÖRGY  
MŰSZAKI TECHNIKUM ÉS GIMNÁZIUM  
VIZSGAREMEK  
Freya’s Garden – Növény csere-bere

Konzulens: Kovács László Készítette: Nacsa Levente  
Gyombolai Ferenc Péter Szabolics András  
 Tábor Tünde

# Hallgatói nyilatkozat

Alulírott, ezúton kijelentem, hogy a vizsgaremek saját, önálló munkám, és korábban még sehol nem került publikálásra.

Nacsa Levente Szabolics András Tábor Tünde

# Konzultációs lap

Vizsgázók neve: Nacsa Levente,

Szabolics András

Tábor Tünde

Vizsgaremek címe: Freya’s Garden – Növény csere-bere

A projekt nyújtotta szolgáltatások:

• Események, feladatok nyomon követése

• Erőforrások (fájlok, ötletek, információk) megosztása a közösséggel

• Regisztráció / Felhasználók nyilvántartása és azonosítása

• Meghívó küldése

• Adminisztráció (Admin, Stats, User)

• E-mailek (visszaigazolás, jelszó módosítás)

(táblázat)

Sorszám A konzultáció időpontja A konzulens aláírása

1. 2023.10.17.

2. 2023.11.21.

3. 2023.12.13.

4. 2024.01.16.

5. 2024.02.20.

6. 2024.03.08.

A vizsgaremek beadható: A vizsgaremeket átvettem:

Vác, 2025. Vác, 2025.

Konzulens

..............................................

A szakképzést folytató

intézmény felelőse

Tartalomjegyzék

[Hallgatói nyilatkozat 2](#_Toc191022941)

[Konzultációs lap 3](#_Toc191022942)

[Témaválasztás 6](#_Toc191022943)

[Fejlesztői dokumentáció 7](#_Toc191022944)

[Fejlesztői környezet 7](#_Toc191022945)

[Visual Studio Code 7](#_Toc191022946)

[Visual Studio 2022 7](#_Toc191022947)

[phpMyAdmin 7](#_Toc191022948)

[PostMan 7](#_Toc191022949)

[…etc 7](#_Toc191022950)

[XAMPP? 7](#_Toc191022951)

[Használt programozási nyelvek, technológiák, könyvtárak 7](#_Toc191022952)

[HTML 7](#_Toc191022953)

[CSS 7](#_Toc191022954)

[JavaScript 7](#_Toc191022955)

[C# 7](#_Toc191022956)

[XAML 7](#_Toc191022957)

[JSON 7](#_Toc191022958)

[JSON Web Token 7](#_Toc191022959)

[MySQL 8](#_Toc191022960)

[HTTP REST API 8](#_Toc191022961)

[Lavavel 8](#_Toc191022962)

[PHP 8](#_Toc191022963)

[MAUI 8](#_Toc191022964)

[Adatbázis 8](#_Toc191022965)

[Adatbázis alapadatai 8](#_Toc191022966)

[(kép a táblákról, kapcsolatokról) 9](#_Toc191022967)

[!Táblák – itt nekünk kérdés hogy kell e sql (valószínűleg export phpmyadminrol oszt csá) 10](#_Toc191022968)

[Használati esetmodell, szerepkörök 10](#_Toc191022969)

[Az alkalmazás felépítése 12](#_Toc191022970)

[Az (projekt) alkalmazás részei (egységei) 12](#_Toc191022971)

[Konfiguráció 12](#_Toc191022972)

[A rendszer indulása 13](#_Toc191022973)

[Adatbázis kapcsolat 13](#_Toc191022974)

[Validáció és űrlapok 13](#_Toc191022975)

[Autentikáció 13](#_Toc191022976)

[…. 13](#_Toc191022977)

[ezek csak lehetséges pontok, igazából itt kéne elmagyarázni a kód és a program működését (pl különböző mappák, classek, mi generál mit) 13](#_Toc191022978)

[Tesztelés 13](#_Toc191022979)

[Továbbfejlesztési lehetőségek 13](#_Toc191022980)

[Felhasználói dokumentáció 13](#_Toc191022981)

# Témaválasztás

A Freya’s Garden célja egy önfenntartó, egészséges közösség alapítása és fenntartása az otthoni kertek működéséért a városokban.

„Tavaly augusztus óta egy önkéntességen alapuló ifjúsági szervezet budapesti (Budapest és Pest megye) körzetében tevékenykedem, mint „Önkéntesekért felelős körzetkoordinátor”, a szervezet középiskolai csereprogramokat bonyolít le **világszinten. Az elmúlt egy-két éves időszakban történt egy generációváltás, önkénteseink nagyobb része már nem tud aktívan részt venni a szervezet életében, mert felsőfokú akadémiai céljaikra vagy karrierjükre kell már fókuszálniuk, azonban** az új érdeklődők bevonása folyamatosan történik. Ebben az időszakban kerültem a körzetvezetői pozícióba, és legfontosabb feladatom az önkéntesek motiválása, elismerése. Az év végén oklevéllel történő elismerés ötlete más körzetek tapasztaltabb vezetőitől érkezett, egy-egy körzetnél ez már jól bevált hagyománynak mondható, azonban Budapest és Pest megye területén eddig nem volt bevett szokás. Ezen ötlet megvalósításához szükségünk van egy módszerre, amely oklevelek tömeges kiállítását teszi lehetővé, emellett az önkénteseket emailben tájékoztatja oklevelük megszerzéséről, ha valamelyikükkel esetleg nem tudnánk személyesen találkozni. Továbbá szempont volt az is, hogy az önkéntesek könnyedén vissza tudják keresni már kiállított okleveleiket, illetve az is, hogy a körzetvezetés is egyszerűen nyilván tudja tartani az elismeréseket. A projekt azonban túlmutat az önkéntesek elismerésén, hiszen ez a motivációs módszertan számos szituációban használható, legyen szó oktatási intézményekről vagy akár munkáltatói belső képzésekről. A projekt célja tehát egy olyan program, amely képes okleveleket kiállítani és azokat nyilvántartásba szervezni, amely a megfelelő jogosultságokkal böngészhető. Az ilyesfajta elismerések azonban nemcsak motivációs erővel bírnak, hanem bizonyos helyzetekben a külvilág számára is információkat hordozhatnak, amelybőlkönnyen hasznot kovácsolhatunk. Gondoljunk csak bele – például egy nyelvtanfolyamról bemutatott tanúsítvány jól mutathat az önéletrajzunkban vagy LinkedIn profilunkon. Minél több ilyen apró teljesítményt rögzítünk szakmai önéletrajzunkban, annál nagyobb valószínűséggel érhetünk el sikereket a munkáltatók meggyőzésében. Összegezve a projekt célja egy újfajta elismerési kultúra népszerűsítése, amely nemcsak a további tanulásra és sikerek elérésére ösztönzi az embereket, hanem a készségek és teljesítményeket kiemelését is lehetővé teszi a munkavállalókat keresők számára. A mai technológiai adottságok mellett nem is volt kérdéses az alkalmazás szerkezete, **mindenképpen webes technológiákban volt érdemes gondolkodni, hiszen ezek** platformfüggetlenül nyújtanak azonos felhasználói élményt.”

# Fejlesztői dokumentáció

## Fejlesztői környezet

### Visual Studio Code

„A Visual Studio Code egy nyílt forrású, ingyenesen elérhető kódszerkesztő alkalmazás, amelyet a Microsoft fejlesztett ki. Támogatja többek között a szintaxis kiemelést, a kódfordítást, a hibakeresést és a Git integrációt. Emellett rengeteg bővítmény érhető el hozzá, amelyekkel testre szabható és kiegészíthető a funkcionalitása. Az alkalmazás keresztplatformos, és támogatja a Windows, macOS és Linux rendszereket. A Visual Studio Code egyre népszerűbb fejlesztői eszköz a web- és az alkalmazásfejlesztés területén.”

### Visual Studio 2022

### phpMyAdmin

### PostMan

### …etc

## XAMPP?

## Használt programozási nyelvek, technológiák, könyvtárak

HTML

Az HTML (HyperText Markup Language) egy leíró nyelv, amelyet a weboldalak létrehozásához használnak. Az HTML alapvetően a weboldalak strukturális elemeit definiálja, és a böngészők megfelelő megjelenítése érdekében használják. Az HTML használatával a fejlesztők képesek az oldal tartalmát, címsorait, fejlécét, láblécét, képeit, videóit és más multimédiás tartalmait definiálni.

### CSS

A CSS (Cascading Style Sheets) egy stílusleíró nyelv, amely lehetővé teszi a weboldalak megjelenésének testreszabását. A CSS segítségével a fejlesztők definiálhatják a weboldalak eleminek stílusát, például a betűméretet, a betűtípust, a színt, a háttérszínt és a marginokat. A CSS megkönnyíti az oldalak egyedi megjelenésének létrehozását és a tartalom és a megjelenés elkülönítését, ami növeli a karbantarthatóságot és a skálázhatóságot.

### JavaScript

A JavaScript egy dinamikus, interpretált, magas szintű programozási nyelv, amely weboldalakon történő interaktív műveletek végrehajtására használatos. A JavaScript elengedhetetlen az internetes alkalmazások és weboldalak fejlesztésében, és számos webes keretrendszer és könyvtár, mint például az Angular, a React és a Vue.js épül rá. A JavaScript lehetővé teszi a dinamikus tartalom létrehozását, az adatok lekérdezését, a felhasználói interakciók kezelését, az animációk és effektek létrehozását. A JavaScript nem csak a böngészőkben használható, hanem a szerveroldalon is, például a Node.js segítségével.

### C#

### XAML

### JSON

### JSON Web Token

A JWT (JSON Web Token) egy nyílt szabvány, amely lehetővé teszi a biztonságos információcserét tokenek segítségével az internetes alkalmazások között. A tokenek tartalmazzák az azonosítást és az engedélyeket, és digitálisan alá vannak írva, hogy biztosítsák az adatok valódiságát és érvényességét. A JWT-k nagyon elterjedtek az internetes alkalmazásokban, mivel könnyen használhatók és biztonságosak. A JWT-k több adatot tartalmazhatnak, mint például az időtúllépési időt vagy a felhasználó szerepkörét, amelyeket az alkalmazás használhat a felhasználói munkamenetek kezeléséhez és az engedélyek felügyeletéhez. A JWT-knek három része van: a fejléc, a token test és az aláírás. A fejléc tartalmazza a token típusát és az algoritmust, amelyet az aláírás létrehozásához használnak. A token test tartalmazza az információkat, amelyek azonosítják a felhasználót és az engedélyeit. Az aláírás az információk digitális aláírására szolgál, és biztosítja, hogy a token tartalma ne módosuljon vagy hamisításra kerüljön. A JWT-knek számos előnye van, például, hogy könnyen használhatók, biztonságosak és hordozhatók az alkalmazások között.

### MySQL

Az MySQL egy ingyenes, nyílt forráskódú relációs adatbázis-kezelő rendszer, amelyet a legtöbb operációs rendszerre telepíthetünk. A MySQL könnyen használható, széles körben elterjedt és nagyon skálázható, ami azt jelenti, hogy nagy adatmennyiségeket is képes hatékonyan kezelni. Az adatok egyszerű kezelése és a megbízhatósága miatt a MySQL széles körben használják az üzleti alkalmazásokban, weboldalakban, mobilalkalmazásokban, játékokban és még sok más területen. A MySQL támogatja az ACID-képességeket, amelyek biztosítják az adatok integritását és konzisztenciáját.

### HTTP REST API

A REST (Representational State Transfer) egy elterjedt architektúra az alkalmazás-programozási interfészek (API-k) tervezéséhez és megvalósításához. A REST API-k az erőforrásokat (adatokat, szolgáltatásokat) ábrázolják, és az HTTP protokollt használják a kliens és a szerver közötti kommunikációhoz. A REST API-k állapotmentesek, azaz a kliens kérésének tartalmaznia kell az összes szükséges információt. A REST API-k népszerűségét az egyszerűség, az egységes interfész és a skálázhatóság adja, amely lehetővé teszi a nagyobb terhelések kezelését.

### Lavavel

### PHP

### MAUI

## Adatbázis

### Adatbázis alapadatai

Adatbázis szerver: MySQL

Adatbázis neve: certifio

Illesztés: utf8mb4\_general\_ci

Motor: InnoDb SQL

parancs a létrehozásához:

CREATE DATABASE IF NOT EXISTS `certifio` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci;

Az adatbázisszerkezet a következő oldalon látható.

### (kép a táblákról, kapcsolatokról)

### !Táblák – itt nekünk kérdés hogy kell e sql (valószínűleg export phpmyadminrol oszt csá)

#### A „conversationuser” tábla



Many-to-many kapcsolótábla, felhasználókat rendel kommunikációs csatornákhoz („beszélgetésekhez”).

(van ahol ilyen a leírás: Adatvédelmi kulcsokat tárol az ASP.NET Core keretrendszer számára.)

SQL parancs:

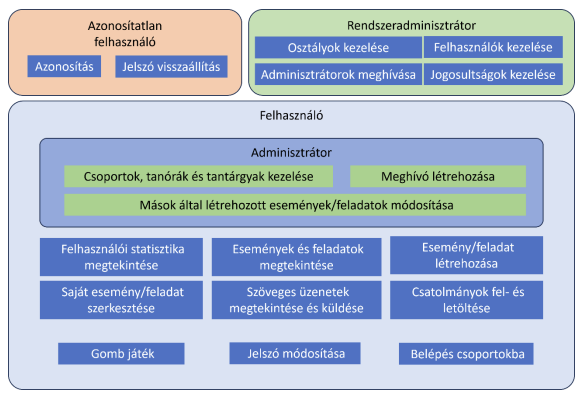
CREATE TABLE `conversations` ( `Id` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL, `Name` varchar(255) NOT NULL, `IsOpen` tinyint(1) NOT NULL, `GroupId` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci; ALTER TABLE `conversations` ADD PRIMARY KEY (`Id`), ADD KEY `IX\_Conversations\_GroupId` (`GroupId`); ALTER TABLE `conversations` ADD CONSTRAINT `FK\_Conversations\_Groups\_GroupId` FOREIGN KEY (`GroupId`) REFERENCES `groups` (`Id`) ON DELETE CASCADE; CREATE TABLE `conversationuser` ( `ConversationsId` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL, `MembersId` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci; ALTER TABLE `conversationuser` ADD PRIMARY KEY (`ConversationsId`,`MembersId`), ADD KEY `IX\_ConversationUser\_MembersId` (`MembersId`); ALTER TABLE `conversationuser` ADD CONSTRAINT `FK\_ConversationUser\_Conversations\_ConversationsId` FOREIGN KEY (`ConversationsId`) REFERENCES `conversations` (`Id`) ON DELETE CASCADE, ADD CONSTRAINT `FK\_ConversationUser\_Users\_MembersId` FOREIGN KEY (`MembersId`) REFERENCES `users` (`Id`) ON DELETE CASCADE;

## Használati esetmodell, szerepkörök

A programban a felhasználók szerepkörei meghatározott jogosultságokkal rendelkeznek.Az ilyen szerepkörök megfelelően strukturálják a felhasználói jogosultságokat, így biztosítva az adatbiztonságot és a jogosulatlan hozzáférés megakadályozását. A jogosultságok hierarchikusan egymásra épülnek, azaz a legalább második szintű jogosultsági szinttől elmondható, hogy minden jogosultsági szint rendelkezik az őt megelőző szint tulajdonságaival.

1. Az azonosítatlan felhasználó jogosultságai:
   1. afsasdfsd
   2. asdfs
   3. asf
2. Az azonosított felhasználó jogosultságai:
   1. sdgsdg
   2. asf
3. A ’Stats’ joggal rendelkező felhasználó jogosultságai:
   1. sdfgsdf
   2. sdffdg
4. Az ’Admin’ joggal rendelkező felhasználó jogosultságai:
   1. fkasédfka
   2. asdfsdf

Az egyes engedélyeket a következő ábrák segítségével szemléltethetjük.

pl  




3. ábra A jogosultságok

## Az alkalmazás felépítése

### Az (projekt) alkalmazás részei (egységei)

A projekt egy Laravel REST API-ból, egy Laravel weboldalból (továbbiakban *Weboldal*), egy MAUI fejlesztőknek szánt – elsősorban asztali - applikációból (továbbiakban *Dev app*) és egy MAUI felhasználóknak szánt - elsősorban mobil - (marketplace) applikációból áll (továbbiakban *Mobil app*).

Mind a weboldal, a DEV és a MOBIL app a Laravel REST API segítségével kommunikál az adatbázissal.

## Mobil app

A mobil applikáció a fő működtető szerve a projektünknek, regisztrált felhasználóinknak elérhető.

Funkcióspecifikáció: (ez forgatókönyv vagy funkcióterv)  
Az applikációban három aloldal fogad.

1. Profil
   * ha nincs bejelentkezve, bejelentkezés/regisztráció screen
   * ha be van jelentkezve:
     + Személyes adatok megadása, módosítása (leíró dolgok)
       - Lakhely vagy terület
       - Elérhetőségek
     + jelszó/email módosítás
     + dev jog kérés ig
2. Piac – gridben piciben hirdetések screen (
   * Új hirdetés megosztása gomb
     + Eladás, csere vagy adomány
     + Kép, leírás
   * Hirdetések megtekintése
     + ez a kijelző fogad, gridben kicsiben a hirdetések fő adatai
       - (termék neve, ár?, település pl)
     + Kereső bar (szöveges)
     + szűrés felül
       - szűrő címkék kategóriák alapján pl kiegészítő, termés,
       - tovbbi szűrő gomb
     + Hirdetők elérhetősége
     + Saját hirdetések kezelése
3. Növények (táblázat)
   * Egy növény (táblázatból)
     + Képek feltöltése
     + Cikkek könnyű elérése
     + Locsolás/gondozás nyomon követés
     + Eladás/hirdetés gomb

### Weboldal

Weboldalunk fő célja az applikácók letöltése és a cikkek elérése. Bárki számára elérhető, nem szükséges a regisztráció.

* Bemutatkozás, általános információk
  + Fő cél
  + Készítők
* Elért eredmények
  + Statisztikák
  + Példák
* Képek
  + Hangulatteremtés, inspiráció
* Tippek, oktatóanyagok, cikkek
  + Növénygondozási útmutatók, javaslatok
  + Receptek
  + Témához kapcsolódó cikkek, érdekességek

### Dev app

Az asztali alkalmazás a fejlesztők számára lesz elérhető.

* Adatok, statisztikák
  + lekérdezések
  + grafikus kimutatások
* Adatbázis kezelés
  + Adatok törlése, módosítása
    - Felhasználói adatok
    - Cikkek

### REST API

Mivel projektünk minden főbb része az adatbázissal dolgozik egy saját API használata általánosítja, kezelhetőbbé teszi a megvalósítást. Az API publikus, közösség számára is elérhető lesz. Bizonyos funckiók (POST, PUT, DELETE) csak megfelelő jogosultsággal érhetők el.

elérhető kérések

### Konfiguráció

A rendszer a konfiguráció tárolásához egy JSON fájlt használ. (appsettings.json) Ez kezdetben még nem létezik, a konfigurációs fázisban hozza létre a rendszer. A konfigurációs fájlban az alábbi beállítások adhatók meg:

• Jwt o Key: Titkos kulcs (a rendszer automatikusan generál egy 128 karakterből álló hexadecimális kulcsot) o Audience, Issuer: A többi mezőre nincs szükség, csak komplexebb authorizációs rendszerek esetén

• FileStorageDir: Feltöltött fájlok tárolásához használt mappa útvonala

• ConnectionStrings

## A rendszer indulása

### Adatbázis kapcsolat

#### Az MVVM architektúra

### Validáció és űrlapok

### Autentikáció

### ….

### ezek csak lehetséges pontok, igazából itt kéne elmagyarázni a kód és a program működését (pl különböző mappák, classek, mi generál mit)

## Tesztelés

## Továbbfejlesztési lehetőségek

# Felhasználói dokumentáció

A program rövid ismertetése