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

# 1. Hallgatói nyilatkozat

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

Nacsa Levente Szabolics András Tábor Tünde

# 2. Konzultációs lap

Vizsgázók neve: Nacsa Levente

Szabolics András

Tábor Tünde

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

A projekt nyújtotta szolgáltatások:

* Regisztráció, bejelentkezés
* Hirdetések megosztása
* Hirdetések megtekintése, szűrése, keresése
* Kapcsolódó cikkek megtekintése, szűrése, keresése
* Adminisztráció (Jogosultsági szintek)
* Adatbázis-kezelés

|  |  |  |
| --- | --- | --- |
| Sorszám | A konzultáció időpontja | A konzulens aláírása |
| 1. | 2024.10.17. |  |
| 2. | 2024.11.21. |  |
| 3. | 2024.12.13. |  |
| 4. | 2025.01.16. |  |
| 5. | 2025.02.20. |  |
| 6. | 2025.03.08. |  |

A vizsgaremek beadható: A vizsgaremeket átvettem:  
Vác, 2025. Vác, 2025.

Konzulens A szakképzést folytató  
intézmény felelőse

3. 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)

# 4. Témaválasztás

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

Manapság nagyon sok háztartásban megoldható lenne egy konyhakert vagy akár egy kis ültetvény fenntartása. A Freya’s Garden lehetővé teszi, elősegíti ezeknek működését, megjelenését. A weboldalunkon megtalálhatók inspiráló, segítőkész avagy csak érdekes cikkek. Az app letöltésével és használatával pedig könnyen adományozhatunk, cserélhetünk, adhatunk, vehetünk növényeket és kiegészítőket.

Az ötlet megfogalmazása során nagy szerepet játszott az, hogy egy valós problémát oldjunk meg. A környezetvédelem, és az önfenntarthatóság mind olyan témák, melyek valós problémákkal foglalkoznak, és valamennyire mindenki számára fontosak. Viszont mindezzel foglalkozni kimerítő reménytelen feladat lehet, főleg manapság. Ezért mi kisebb léptékben gondolkoztunk. Tudatos, jóindulatú közösségek megteremtésével egy nagyobb társadalmi változás első, kis lépéseinek adunk lehetőséget. Persze ez nem teljes megoldás, sokkal több faktor sokkal nagyobb léptékben változtatja környezetünket. Annyi viszont biztos, hogy a mindennapjainkban pozitív változást tud előidézni egy olyan kezdeményezés mint a mi projektünk.

# 5. Fejlesztői dokumentáció

## 5.1 Fejlesztői környezet

### 5.1.1 Visual Studio Code

A Visual Studio Code, amelyet általában VS Code-ként is emlegetnek, egy forráskód-szerkesztő, amelyet a Microsoft fejlesztett ki Windows, Linux, macOS és webböngészők számára. A szolgáltatások közé tartozik a hibakeresés támogatása, a szintaktikai kiemelés, az intelligens kódkiegészítés, a kódrészletek, a kódrefaktorálás és a beágyazott verziókezelés a Git segítségével. A felhasználók módosíthatják a témát, a billentyűparancsokat, a beállításokat, telepíthetnek bővítményeket, amelyek funkcionalitást adnak, és beállíthatják a VS-kódot IDE-ként való használatra.

### *.*5.1.2 Visual Studio 2022

A Visual Studio egy integrált fejlesztői környezet (IDE), amelyet a Microsoft fejlesztett ki. Számítógépes programok fejlesztésére használják, beleértve webhelyeket, webes alkalmazásokat, webszolgáltatásokat és mobilalkalmazásokat. A Visual Studio a Microsoft szoftverfejlesztési platformjait használja, beleértve a Windows API-t, a Windows Forms-t, a Windows Presentation Foundation-t (WPF), a Microsoft Store-t és a Microsoft Silverlight-ot. Natív kódot és felügyelt kódot is képes előállítani.

### 5.1.3 phpMyAdmin

A phpMyAdmin egy PHP nyelven írt ingyenes szoftvereszköz, amely a MySQL webes adminisztrációjának kezelésére szolgál. A phpMyAdmin a MySQL és a MariaDB műveletek széles skáláját támogatja. A gyakran használt műveletek (adatbázisok, táblák, oszlopok, relációk, indexek, felhasználók, engedélyek stb. kezelése) végrehajthatók a felhasználói felületen keresztül, miközben továbbra is lehetősége van bármilyen SQL utasítás közvetlen végrehajtására.

### 5.1.4 EchoAPI

Az EchoAPI egy minden az egyben API fejlesztési együttműködési platform, amely olyan alapvető funkciókat kínál, mint az API hibakeresés, tervezés, automatizált tesztelés, stresszteszt és dokumentáció generálás.

### 5.1.5 GIMP

A GNU Image Manipulation Program, közismert mozaikszóval GIMP, egy ingyenes és nyílt forráskódú raszteres grafikus szerkesztő, amely képmanipulációra (retusálásra) és képszerkesztésre, szabad formájú rajzolásra, különböző képfájlformátumok átkódolására és speciálisabb feladatokra használható. Beépülő modulokkal bővíthető, és szkriptelhető. Nem rajzolásra készült, bár egyes művészek és alkotók így használták.

## 5.1.6 XAMPP

Az XAMPP az Apache Friends által kifejlesztett ingyenes és nyílt forráskódú, többplatformos webszerver-megoldási csomag, amely főleg az Apache Server-ből, MariaDB adatbázisból és PHP nyelvű HTTP-tolmácsokból áll. Mivel a legtöbb tényleges webszerver-telepítés ugyanazokat az összetevőket használja, mint a XAMPP, lehetővé teszi az átállást a helyi tesztkiszolgálóról egy élő szerverre.

## 5.2 Használt programozási nyelvek, technológiák, könyvtárak

### 5.2.1 HTML

A HTML (angolul: HyperText Markup Language) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. Az aktuális változata az 5, mely az SGML általános jelölőnyelv egy konkrét alkalmazása.

### 5.2.2 CSS

A Cascading Style Sheets (CSS) egy stíluslapnyelv, amelyet egy jelölőnyelven, például HTML-ben vagy XML-ben (beleértve az XML-dialektusokat, például az SVG-t, MathML-t vagy XHTML-t) írt dokumentumok megjelenítésének és stílusának meghatározására használnak. A CSS a World Wide Web egyik sarokköve, a HTML és a JavaScript mellett.

### 5.2.3 JavaScript

JavaScript amelyet gyakran JS-nek rövidítenek, a HTML és a CSS mellett a World Wide Web programozási nyelve és alaptechnológiája. A webhelyek 99 százaléka JavaScriptet használ a kliensoldalon a weboldalviselkedéséhez. A webböngészőknek van egy dedikált JavaScript motorja, amely végrehajtja az ügyfélkódot. Ezeket a motorokat egyes szervereken és számos alkalmazásban is használják. A legnépszerűbb rendszer a nem-kliensoldali használathoz a Node.js.

### 5.2.4 C#

### 5.2.5 XAML

### 5.2.6 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.

### 5.2.7 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.

### 5.2.8 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.

### 5.2.9 Lavavelx12

### 5.2.10 PHP

### 5.2.11 MAUI

## 5.3 Adatbázis

### 5.3.1 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:

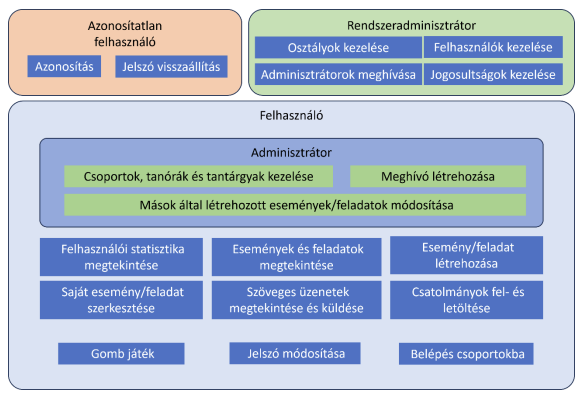
CREATE TABLE `conversations` ( `Id` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL, `Name` varchar(255) NOT NULL, `IsOpen` tinyint(1) NOT NULL, `GroupId` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci; 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;

## 5.4 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

## 5.5 Az alkalmazás felépítése

### 5.5.1 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.

### 5.5.2 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

### 5.5.3 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

### 5.5.4 Dev app

Az asztali alkalmazás a fejlesztők számára lesz elérhető.

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

### 5.5.5 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

### 5.5.6 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

# 6. Felhasználói dokumentáció

## 6.1 A weboldal

## 6.2 Az app

## 6.3 Az asztali app

# 7.Források

#### <https://en.wikipedia.org/wiki/Visual_Studio_Code>

* <https://en.wikipedia.org/wiki/Visual_Studio>
* <https://www.phpmyadmin.net/>
* <https://www.echoapi.com/wiki/docs/introductione>
* <https://en.wikipedia.org/wiki/GIMP>
* <https://hu.wikipedia.org/wiki/HTML>
* <https://en.wikipedia.org/wiki/CSS>
* <https://en.wikipedia.org/wiki/JavaScript>

Az oldalakon angol nyelven megjelenő szöveget a GoogleChrome automatikus fordítórendszerével fordítottuk magyarra.