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:

* 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

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.

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.

# Fejlesztői dokumentáció

## Fejlesztői környezet

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

## Használt programozási nyelvek, technológiák, könyvtárak

### 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.

### 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.

### 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.

### C#

A C# egy általános célú, magas szintű programozási nyelv, amely több paradigmát is támogat. A C# magában foglalja a statikus gépelést erős gépelést, lexikális hatókörű, felszólító, deklaratív, funkcionális, általános, objektum-orientált (osztály-alapú) és komponens-orientált programozási tudományágat.

### XAML

Az XAML egy deklaratív jelölőnyelv. A .NET programozási modellhez hasonlóan az XAML leegyszerűsíti a felhasználói felület létrehozását egy .NET-alkalmazáshoz.

### JSON

A JSON (JavaScript Object Notation) egy könnyű adatcsere-formátum. Az ember számára könnyű írni és olvasni. A gépek könnyen értelmezhetők és generálhatók. A JavaScript programozási nyelv szabvány ECMA-262 3. kiadás – 1999. december egy részhalmazán alapul. A JSON egy szövegformátum, amely teljesen nyelvfüggetlen, de olyan konvenciókat használ, amelyek a C nyelvcsalád programozói számára ismertek, beleértve a C, C++, C#, Java, JavaScript, Perl, Python és sok más nyelvet.

#### JSON Web Token

A JSON Web Token (JWT) egy kompakt, URL biztonságos eszköz a két fél között átadandó követelések megjelenítésére. A JWT-ben található követelések JSON-objektumként vannak kódolva, amely egy JSON Web Signature (JWS) struktúra hasznos terheléseként vagy egy JSON Web Encryption (JWE) struktúra egyszerű szövegeként használatos, lehetővé téve a követelések digitális aláírását vagy az integritás védelmét üzenet-hitelesítő kóddal (MAC) és/vagy titkosítással.

### MySQL

A MySQL szoftver egy nagyon gyors, többszálú, többfelhasználós és robusztus SQL (Structured Query Language) adatbázis-kiszolgálót biztosít. A MySQL Server kritikus fontosságú, nagy terhelésű termelési rendszerekhez, valamint tömegesen telepített szoftverekbe való beágyazáshoz készült. Az Oracle az Oracle Corporation és/vagy leányvállalatainak bejegyzett védjegye.

### HTTP REST API

A REST API a REpresentational State Transfer API rövidítése. Ez egyfajta API (Application Programming Interface), amely lehetővé teszi a kommunikációt a különböző rendszerek között az interneten keresztül. A REST API-k úgy működnek, hogy kéréseket küldenek és válaszokat fogadnak, jellemzően JSON formátumban, az ügyfél és a szerver között. A REST API-k HTTP-módszereket (például GET, POST, PUT, DELETE) használnak az erőforrásokon végrehajtható műveletek meghatározásához. Ezek a módszerek igazodnak a CRUD (Létrehozás, Olvasás, Frissítés, Törlés) műveletekhez, amelyek az erőforrások weben történő manipulálására szolgálnak

### Lavavel

A Laravel egy webalkalmazás-keretrendszer kifejező, elegáns szintaxissal. A webes keretrendszer struktúrát és kiindulópontot biztosít az alkalmazás létrehozásához, lehetővé téve, hogy valami csodálatos létrehozására összpontosítson, miközben a részleteket izzadjuk.

### PHP

A PHP , amely a PHP: Hypertext Preprocessor rövidítése , egy széles körben használt nyílt forráskódú, általános célú szkriptnyelv, amely különösen alkalmas webfejlesztésre, és beágyazható a HTML-be. Szintaxisa C-re, Java-ra és Perl-re támaszkodik.

### MAUI

A .NET többplatformos alkalmazás felhasználói felülete (.NET MAUI) egy platformfüggetlen keretrendszer natív mobil-és asztali alkalmazások C# és XAML használatával történő létrehozásához.

## Adatbázis

### Adatbázis alapadatai

* Adatbázis szerver: MySQL
* Adatbázis neve: freyas\_garden
* Illesztés: utf8mb4\_unicode\_ci
* Motor: InnoDb SQL
* parancs a létrehozásához:
* 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:

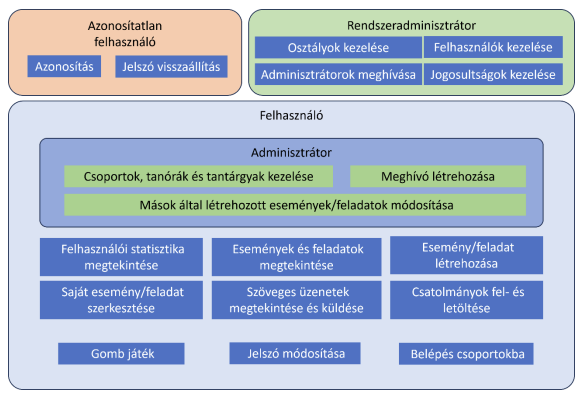
CREATE TABLE `conversations` ( `Id` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL, `Name` varchar(255) NOT NULL, `IsOpen` tinyint(1) NOT NULL, `GroupId` char(36) CHARACTER SET ascii COLLATE ascii\_general\_ci NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci; 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 programjainkban a felhasználók különböző szerepkörökre vannak osztva, ezekkel a szerepkörökkel különböző jogosultságok járnak. A szerepkörök hierarchikusan épülnek egymásra, tehát egy magasabb szintű szerepkör rendelkezik az alatta lévő szerepkör jogosultságaival.

1. Az azonosítatlan felhasználó jogosultságai:
   1. web app használata
      1. cikkek olvasása, keresése, szűrése
      2. asztali, telefonos app letöltése
   2. Mobil app használata
      1. regisztráció
      2. bejelentkezés
      3. új jelszó kérése (létező fiók esetén)
2. Az azonosított felhasználó jogosultságai:
   1. Mobil app használata
      1. hirdetések megtekintése, keresése, szűrése
         1. egy hirdetés kiemelt megtekintése, hirdető adatok, elérhetőségek megtekintése
      2. új hirdetés közzététele
      3. saját hirdetések módosítása, törlése
      4. megtekintett hirdetések elmentése, elmentett hirdetések megtekintése
      5. saját növények kezelése
      6. profil adatok módosítása
         1. felhasználónév
         2. profilkép
         3. leírás
3. A ’Stats’ joggal rendelkező felhasználó jogosultságai:
   1. Dev app használata
      1. cikkek kezelése
         1. létrehozás
         2. módosítás
         3. törlés
      2. Statisztikák, lekérések megtekintése
4. Az ’Admin’ joggal rendelkező felhasználó jogosultságai:
   1. Dev app használata
      1. Felhasználók kezelése
         1. összes fiók megtekintése
         2. felhasználói adatok kezelése
            1. alapvető adatok módosítása
            2. új jelszó igénylése (felhasználói e-mailre)
            3. fiók törlése
         3. felhasználó hirdetéseinek kezelése
            1. módosítás
            2. törlés

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 weboldal

## Az app

## Az asztali app

# 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/introduct ion](https://www.echoapi.com/wiki/docs/introduct%20%20ion)
* <https://en.wikipedia.org/wiki/GIMP>
* <https://hu.wikipedia.org/wiki/HTML>
* <https://en.wikipedia.org/wiki/CSS>
* <https://en.wikipedia.org/wiki/JavaScript>
* <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>
* <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-9.0>
* <https://www.json.org/json-en.html>
* <https://www.rfc-editor.org/info/rfc7519>
* <https://dev.mysql.com/doc/refman/8.0/en/introduction.html>
* <https://www.geeksforgeeks.org/rest-api-introduction/>
* <https://laravel.com/docs/12.x#meet-laravel>
* <https://www.php.net/manual/en/preface.php>

Az oldalakon az idegen nyelven megjelenő szövegeket a GoogleChrome automatikus fordítórendszerével fordítottuk magyarra.