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

[1. Hallgatói nyilatkozat 2](#_Toc192764694)

[2. Konzultációs lap 3](#_Toc192764695)

[3. Témaválasztás 6](#_Toc192764696)

[4. Fejlesztői dokumentáció 7](#_Toc192764697)

[4.1 Fejlesztői környezet 7](#_Toc192764698)

[4.1.1 Visual Studio Code 7](#_Toc192764699)

[4.1.2 Visual Studio 2022 7](#_Toc192764700)

[4.1.3 phpMyAdmin 7](#_Toc192764701)

[4.1.4 EchoAPI 7](#_Toc192764702)

[4.1.5 GIMP 7](#_Toc192764703)

[4.1.6 XAMPP 7](#_Toc192764704)

[4.2 Használt programozási nyelvek, technológiák, könyvtárak 8](#_Toc192764705)

[4.2.1 HTML 8](#_Toc192764706)

[4.2.2 CSS 8](#_Toc192764707)

[4.2.3 JavaScript 8](#_Toc192764708)

[4.2.4 C# 8](#_Toc192764709)

[4.2.5 XAML 8](#_Toc192764710)

[4.2.6 JSON 8](#_Toc192764711)

[4.2.7 MySQL 9](#_Toc192764712)

[4.2.8 HTTP REST API 9](#_Toc192764713)

[4.2.9 Lavavel 9](#_Toc192764714)

[4.2.10 PHP 9](#_Toc192764715)

[4.2.11 MAUI 9](#_Toc192764716)

[4.3 Adatbázis 10](#_Toc192764717)

[4.3.1 Adatbázis alapadatai 10](#_Toc192764718)

[4.3.2 Táblák 10](#_Toc192764719)

[4.4 A rendszer felépítése 11](#_Toc192764720)

[4.4.1 Marketplace 11](#_Toc192764721)

[4.4.2 Weboldal 11](#_Toc192764722)

[4.4.3 Dev app 11](#_Toc192764723)

[4.4.4 REST API 11](#_Toc192764724)

[4.4.5 Konfiguráció 11](#_Toc192764725)

[4.5 Használati esetmodell, szerepkörök 12](#_Toc192764726)

[4.6 A rendszer működése 14](#_Toc192764727)

[4.6.1 Adatbázis kapcsolat 14](#_Toc192764728)

[4.6.2 Az MVVM architektúra 14](#_Toc192764729)

[4.6.3 Validáció és űrlapok 14](#_Toc192764730)

[4.6.4 Autentikáció 14](#_Toc192764731)

[4.6.5 …. 14](#_Toc192764732)

[4.7 Tesztelés 14](#_Toc192764733)

[4.8 Továbbfejlesztési lehetőségek 14](#_Toc192764734)

[5. Felhasználói dokumentáció 14](#_Toc192764735)

[5.1 A rendszer használata 14](#_Toc192764736)

[5.1.1 MARKETPLACE 14](#_Toc192764737)

[5.1.2 WEB 14](#_Toc192764738)

[5.1.3 DEV 14](#_Toc192764739)

[5.2 A rendszer indítása 14](#_Toc192764740)

[6. Források 14](#_Toc192764741)

[7. Mellékletek 15](#_Toc192764742)

# 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, fenntartását.

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 fontosok. 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 IDE egy kreatív indítópult, amellyel szerkesztheti, hibakeresheti és összeállíthatja a kódot, majd közzétehet egy alkalmazást. A legtöbb IDE által biztosított szabványos szerkesztőn és hibakeresőn túl a Visual Studio fordítókat, kódkiegészítő eszközöket, grafikus tervezőket és még sok más szolgáltatást is tartalmaz a szoftverfejlesztési folyamat javítása érdekében.

### 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.

### Inkscape

Az Inkscape egy ingyenes és nyílt forráskódú vektorgrafikus szerkesztő GNU/Linux, Windows és macOS számára. A funkciók gazdag készletét kínálja, és széles körben használják művészi és műszaki illusztrációkhoz, például rajzfilmekhez, clip art-okhoz, logókhoz, tipográfiához, diagramokhoz és folyamatábrákhoz. Vektorgrafikát használ, hogy éles nyomatokat és korlátlan felbontású rendereléseket tegyen lehetővé, és nincs meghatározott számú pixelhez kötve, mint például a rasztergrafikák.

### 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.

#### MVVM (Model VIew ViewModel) Community Toolkit

A CommunityToolkit.Mvvm csomag (más néven MVVM Toolkit, korábbi nevén Microsoft.Toolkit.Mvvm) egy modern, gyors és moduláris MVVM könyvtár. A .NET Community Toolkit része, és a következő elvekre épül:

### Composer

A Composer egy PHP függőségkezelési eszköz. Lehetővé teszi, hogy deklarálja azokat a könyvtárakat, amelyektől a projektje függ, és kezeli (telepíti/frissíti) azokat az Ön helyett.

## Adatbázis

Adatbázisunk kezeléséhez egy Laravel [REST API](#_REST_API)-t használunk.

*Ahol képet küldünk vagy fogadunk, ott a képet az adatbázisban nem tároljuk, csak az api alap url-jéhez csatolandó feltétlen szükséges elérési útvonalat. A teljes elérési útvonalat a GET request adja vissza.*

Pl.: http://localhost:8069/storage/public/articles/articles\_a2b384ec-a788-45df-9d67-fe0cf2194da3.webp

### Adatbázis alapadatai

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

### Táblák

#### plants

A plants táblában általános adatokat tárolunk különféle növenyekről. A rendszer indulásakor statikus adatokkal töltődik fel. Oszlopai:

* id
  + Elsődleges kulcs
  + Szám
  + Pl.: 3
* name
  + Leíró tulajdonság
  + A növény neve magyarul
  + Szöveg
  + Pl.: Banán
* latin\_name
  + Leíró tulajdonság
  + A növény neve latinul
  + Szöveg
  + Pl.: Musa
* type\_id
  + Idegen kulcs
  + A növény típusának azonosító száma
  + Szám
  + Pl.: 1

A plants táblához csatlakozik a types szótár tábla, mely a növéyn típusokat tárolja. A rendszer indulásakor statikus adatokkal töltődik fel. Oszlopai:

* id
  + Elsődleges kulcs
  + Szám
  + Pl.: 1
* name
  + Leíró tulajdonság
  + A típus neve
  + Szöveg
  + Pl.: gyümölcs

#### users

A users táblában tároljuk a felhasználóink adatait. Dinamikusan feltöltött tábla, regisztráció esetén új rekord kerül bele. Oszlopai:

* id
  + Elsődleges kulcs
  + Szám
  + Pl.: 1
* username
  + Leíró tulajdonság
  + A fiók felhasználóneve
  + Szöveg
  + Pl.: admin
* email
  + Leíró tulajdonság
  + A felhasználó e-mail címe
  + Szöveg (e-mail)
  + Pl.: admin@freyasgarden.com
* city
  + Leíró tulajdonság
  + A felhasználó tartózkodási helye, területe
  + Szöveg
  + Pl.: Vác
* birthdate
  + Leíró tulajdonság
  + A felhasználó születési dátuma
  + Dátum
  + Pl.: 2004-07-20
* password
  + Leíró tulajdonság
  + A felhasználó jelszava
  + (HASH-kóddal titkosított) Szöveg
  + Pl.: $2y$12$9hr1rGGbJoAXnvpjd9AcQuGMLLPXw9xYZy0jQWkMYK9qV8/7Iutw6
* role\_id
  + Idegen kulcs
  + A felhasználó jogi szerepkörének azonosító száma
  + Szám
  + Pl.: 1
* pitcure
  + Leíró tulajdonság
  + A felhasználó profilképének választott képe
  + Elérési útvonalat leíró szöveg
  + Pl.: placeholders/Shovel.png
* description
  + Leíró tulajdonság
  + A felhasználó leírása
  + Szöveg
  + Pl.: Freya’s Garden fejlesztő, paradicsom termesztő
* remember\_token
  + Leíró tulajdonság
  + A felhasználó átmeneti azonosításához szükséges kód
  + Szöveg
  + Pl.: H7GNTScqql

A users táblához csatlakozik a roles szótár tábla, mely a [különféle szerepkörök](#_Használati_esetmodell,_szerepkörök)et tárolja. A rendszer indulásakor statikus adatokkal töltődik fel. Oszlopai:

* id
  + Elsődleges kulcs
  + A szerepkör azonosító száma
  + Szám
  + Pl.: 1
* name
  + Leíró tulajdonság
  + A szerepkör megnevezése
  + Szöveg
  + Pl.: admin

#### user\_plants

A user\_plants tábla tárolja a felhasználók számontartott növényeit, azoknak adatait. Kapcsoló táblaként is szolgál a users, plants, és listings táblák között. Oszlopai:

* id
  + Elsődleges kulcs
  + A számontartott növény(ek) azonosító száma
  + Szám
  + Pl.: 1
* user\_id
  + Idegen kulcs
  + A kapcsolódó felhasználó azonosító száma
  + Szám
  + Pl.: 62
* plant\_id
  + Idegen kulcs
  + A számontartott növény(ek) fajtája
  + Szám
  + Pl.: 90
* stag\_id
  + Idegen kulcs
  + A számontartott növény(ek) stádiumának azonosítója
  + Szám
  + Pl.: 5
* count
  + Leíró tulajdonság
  + A számontartott növény(ek) darabszáma
  + Szám
  + Pl.: 5

A user\_plants táblához csatlakozik a stages szótártábla, mely a különféle növény-életciklus stádiumokat tárolja. A rendszer indulásakor statikus adatokkal töltődik fel. Oszlopai:

* id
  + Elsődleges kulcs
  + A stádium azonosító száma
  + Szám
  + Pl.: 1
* name
  + Leíró tulajdonság
  + A stádium megnevezése
  + Szöveg
  + Pl.: mag

#### listings

A listings táblában tároljuk a marketplace-en közzétett hirdetések adatait. Egy listing létezéséhez szükséges feltétel egy user\_plants létezése. Oszlopai:

* id
  + Elsődleges kulcs
  + A hirdetés azonosító száma
  + Szám
  + Pl.: 1
* user\_plants-id
  + Idegen kulcs
  + A hirdetésben szereplő számontartott növény(ek) azonosító száma
  + Szám
  + Pl.: 4
* title
  + Leíró tulajdonság
  + A hirdetés címe
  + Szöveg
  + Pl.: 2 kiló paradicsom!!!
* description
  + Leíró tulajdonság
  + A hirdetés leírása
  + Szöveg
  + Pl.: Nagyon finomak, érettek 😊
* city
  + Leíró tulajdonság
  + A hirdetéshez releváns terület
  + Szöveg
  + Pl.: Göd
* media
  + Leíró tulajdonság
  + A hirdetéshez csatolt kép(ek)
  + Elérési útvonalt leíró szöveg
  + Pl.: "[\"listings\\\/listings\_6f0c8869-82db-4d23-b605-e...
* price
  + Leíró tulajdonság
  + A meghirdetett ár
  + Szám
  + Pl.: 5000

#### articles

Az articles táblában tároljuk a weboldalon megtalálható cikkek adatait. Oszlopai:

* id
  + Elsődleges kulcs
  + A cikk azonosító száma
  + Szám
  + Pl.: 7
* title
  + Leíró tulajdonság
  + A cikk címe
  + Szöveg
  + Pl.: Receptek répával
* plant\_id
  + Idegen kulcs
  + A cikkhez kapcsolódó növény azonosító száma, nem kötelező
  + Szám
  + Pl.: 34
* author\_id
  + Idegen kulcs
  + A cikk szerzőjének azonosító száma (a users táblára mutat)
  + Szám
  + Pl.: 1
* category\_id
  + Idegen kulcs
  + A cikk kategóriájának azonosító száma
  + Szám
  + Pl.: 2
* description
  + Leíró tulajdonság
  + A cikk rövid leírása
  + Szöveg
  + Pl.: Ezzel az öt recepttel feldobhatod a mindennapjaidat!
* content
  + Leíró tulajdonság
  + A cikk tartalma, markdown-ban tárolva
  + Hosszú szöveg
* source
  + Leíró tulajdonság
  + A cikk forrása, link, nem kötelező
  + Szöveg
  + pl.: http://ankunding.org/officia-deserunt-pariatur-totam-a-ipsam-maiores-minima

Az articles táblához csatlakozik a categories szótártábla, mely a különféle kategóriákat tárolja. A rendszer indulásakor statikus adatokkal töltődik fel. Oszlopai:

* id
  + Elsődleges kulcs
  + A kategória azonosító száma
  + Szám
  + Pl.: 3
* name
  + Leíró tulajdonság
  + A kategória megnevezése
  + Szöveg
  + Pl.: recept

Képek a táblákról, kapcsolatokról közte

## A rendszer felépítése

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 -(továbbiakban *Marketplace*) applikációból áll.

### Marketplace

A Marketplace a fő működtető szerve a projektünknek, regisztrált felhasználóinknak elérhető. Itt hirdetéseket tekinthetünk meg, vagy közzé tehetünk saját hirdetéseket is.

### Weboldal

Weboldalunk fő célja az ismeretterjesztés. Bárki számára elérhető, nem szükséges a regisztráció. Itt különféle cikkeket tekinthetünk meg, innen tölthetjük az asztali és telefonos appokat, emellett az ügyfélszolgálati rész is itt található.

### Dev app

Az asztali alkalmazás a fejlesztők számára lesz elérhető. Itt, a megfelelő jogosultság birtokában, kezelhetjük az adatbázist, valamint statisztikákat, lekéréseket tekinthetünk meg.

### 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. Bizonyos funkciók (POST, PUT, DELETE) csak megfelelő jogosultsággal érhetők el. (lásd [Az API működése](#_Adatbázis_kapcsolat))

elérhető kérések

### Konfiguráció

* client,: readmebe api port stuff

#### A .env fájl

A REST API és a WEB projekt könyvtárában megtalálható .env fájlban sok fontos adat beállítható.

screenshots

## 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.

### Szerepkörök, jogosultságaik

1. Azonosítatlan felhasználó
   1. web app használata
      1. cikkek olvasása, keresése, szűrése
      2. asztali, telefonos app letöltése
   2. Marketplace használata
      1. regisztráció
      2. bejelentkezés
      3. új jelszó kérése (létező fiók esetén)
2. Azonosított felhasználó
   1. Marketplace 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. profil adatok módosítása
         1. felhasználónév
         2. profilkép
         3. leírás
3. Stats
   1. Dev app használata
      1. cikkek kezelése
         1. létrehozás
         2. módosítás
         3. törlés
      2. Statisztikák, lekérések megtekintése
4. Admin
   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

### Szerepkörök halmazábrája

## A rendszer működése

### Adatbázis kapcsolat, Laravel

Az adatbázisunkat a Laravelben elkészített REST API-on keresztül kezeljük. Az API tölti fel adatokkal, valamint az kezeli a weboldalról, és az appokból érkező kéréseket. Ezzel központosított az adatbázis kapcsolat, könnyebben skálázható, módosítható.

#### Migrációk

Az API Laravelben kezelhető migrációkat használ. A migrációk felelősek a táblák létrehozásáért, definiálásáért, valamint a táblák közti kapcsolatok felépítésért.

#### Seederek

A seederek szintén Laravelben kezelhetők, ezek felelősek a [táblák](#_Táblák) feltöltéséért.

### Az MVVM és MVC architektúra

A két egy-egy könyvtár struktúrát takar, melyek segítségével hatékonyabban, rendezettebben, és átláthatóbban tudjuk felépíteni az adott programjainkat.

#### MVVM

(Model View ViewModel) Ezt a Marketplace-ben, és a Dev-appban használjuk

#### MVC

(Model View Controller) Ezt az API-ban\* valamint a weboldalban használjuk

Mindkét esetben hasonló a jelentés, és az elemek funkciói.

* Model

A model-ek feladata az adott objektum vagy típus adatainak, tulajdonságainak leírása, rendszerezése.

* View

A view-k a felhasználó által látott nézetek, túlnyomóan logikai műveletek nélküli frontend fájlok. \*Az API-ban nem használunk view-kat, hiszen annak nincs frontend-része.

* ViewModel/Controller

Ezeknek feladata a logikai műveletek kezelése, a view-k és a Modelek összekapcsolása.

## Tesztelés

### Faker

A Faker egy PHP-könyvtár, amely hamis adatokat generál, ezt kihasználva fel tudtuk tölteni tesztadatokkal a programjainkat.

## Továbbfejlesztési lehetőségek

* letölthető appok
* web forum
* chat ig
* transactions

# Felhasználói dokumentáció

## A rendszer használata

Ide kb egy felhasznaloi tartalomjegyzek kell, nem teljesen reszletes, inkabb atlatható.

### MARKETPLACE

Az app megnyitásakor először regisztráció, avagy már létező felhasználói fiókkal való bejelentkezés szükséges. (lásd [4.5 Használati esetmodell, szerepkörök](#_Használati_esetmodell,_szerepkörök))Enélkül nem lehet használni az app többi funkcióját. Ezután a menüvel, gombokkal, vagy más funkciókkal navigálhatunk az appon belül.

!!SCREENSHOTOK!!

* Profil
  + Személyes adatok megadása, módosítása
    - Lakhely, terület
    - Elérhetőségek
    - E-mail cím
  + jelszó módosítása (weben keresztül)
  + Profilkép beállítása
  + Saját hirdetések megtekintése
* Piac
  + Hirdetések megtekintése (Fogadó oldal)
    - Kártyákon megjelenő hirdetések
      * Kép(ek)
      * Általános leírás, adatok
      * Hirdető megtekintése gomb
    - Szöveges keresősáv
    - Szűrők
* Új hirdetés megosztása
  + Hirdetés adatai
    - Kép(ek)
    - Leírás
    - Név
    - Növény/hirdetett dolog fajtája
* Hirdető megtekintése
  + Publikus profil adatok megtekintése
    - Név
    - Profilkép
    - Lakhely, terület
    - Elérhetőségek
  + Hirdető összes hirdetésének megtekintése

### WEB

A weboldalunk minden oldalán megtalálható a navigációs sáv, aminek segítségével könnyen lehet navigálni az oldalak között. Regisztráció és bejelentkezés nélkül is igénybe vehető.

screenshot of nav

* Főoldal
  + GitHub elérése

screenshot of hero n download stuff

* + Általános információk
    - bemutatkozás
    - elérhetőségek
    - érdekességek
  + friss cikkek

screenshot of friss cikkek n other stuff

* Cikkek
  + Keresés

screenshot of main cikk page

* + Szűrés

screenshot of filters, n pagination

* + Egy cikk megtekintése

screenshot of egy cikk

### DEV

A DEV-app megnyitásakor, a Marketplace-hez hasonlóan bejelentkezés szükséges, itt viszont csak a Stats és az Admin jogosultsággal lehet az alkalmazást igénybe venni. (lásd [4.5 Használati esetmodell, szerepkörök](#_Használati_esetmodell,_szerepkörök))

* Adatok, statisztikák (legalább Stats jogosultság)
  + lekérdezések
  + grafikus kimutatások
* Cikkek hozzáadása, saját cikkek módosítása, törlése (legalább Stats jogosultság)
* Adatbázis kezelés (Admin jogosultság)
  + Adatok törlése, módosítása
    - Felhasználói adatok
    - Hirdetések
    - Cikkek törlése, módosítása

## A rendszer indítása

step by step guide

gitgub

xampp

readme-k n stuff

# Források

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

* <https://visualstudio.microsoft.com/#vs-section>
* <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>
* <https://inkscape.org/about/>
* <https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/#additional-resources>
* <https://getcomposer.org/doc/00-intro.md>
* https://fakerphp.org/

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

# Mellékletek

apidoc

forráskód(ok) (github)