**PROJEKTMUNKA**

Plant Planet weboldala

**Témavezető:** **Készítette:**

Inántsy-Pap Tamás Dobrai Dávid

Juhászné Kovács Ildikó Tóth Dominik Sándor

Türk Károly

NYÍREGYHÁZA

2025

Tartalom

[Feladat megadása 2](#_Toc196680061)

[1. Szereplők és igényeik 2](#_Toc196680063)

[2. Use-Case Funkció lista 2](#_Toc196680064)

[2.1. Bejelentkezett felhasználó 2](#_Toc196680065)

[2.2. Látogató 3](#_Toc196680066)

[3. Magas szintű rendszerterv 4](#_Toc196680067)

[4. Screenek 5](#_Toc196680068)

[5. Modellek 7](#_Toc196680069)

[6. Alkalmazások kiválasztása 11](#_Toc196680070)

[6.1. Front-end 11](#_Toc196680071)

[6.2. Backend 12](#_Toc196680072)

[6.3. Adatbázis 12](#_Toc196680073)

[6.4. Tesztelés 12](#_Toc196680074)

[7. Routing 13](#_Toc196680075)

[8. Implementáció 16](#_Toc196680076)

[9. Tesztelés 17](#_Toc196680077)

[10. Bevezetés, éles üzemmód 17](#_Toc196680078)

# Feladat megadása

# Egy szolgáltatás, amely segít a növényvásárlásban és ezek gondozásában. Havi előfizetéses rendszerrel működő, modellen alapuló webshop. A növény megvásárlását követően havi instrukciókkal és szükséges kiegészítőkkel látjuk el a felhasználót, az időjárástól függően. A havi előfizetéses modell részeként a felhasználónak nincs más dolga, csak követni az instrukciókat; a szükséges kellékek automatikusan megérkeznek majd az ajtaja elé. Email hírlevél-rendszeren keresztül a felhasználó azonnali információkhoz juthat viharokról vagy más, növényeit fenyegető eseményekről.

# 1. Szereplők és igényeik

**Bejelentkezett felhasználó**

* Legyen képes vásárolni a rögzített termékekből
* A kosár tartalmát megtekinteni, módosítani, megvásárolni.
* Legyen képes saját adatait módosítani

**Látogató**

* Tudjon böngészni
* Legyen képes regisztrálni

# 2. Use-Case Funkció lista

Az alkalmazásban a felhasználókat vagy vásárlókat két csoportra tudjuk osztani. Ezek a csoportok a látogató és a regisztrált vásárló vagy egyszerűen vásárló. A két csoport között a regisztráció tesz különbséget, így hiába a látogató és a vásárló is el tudja végezni ugyanazt a webáruházban, mégis van némi különbség a lehetőségeik között.

## 2.1. Bejelentkezett felhasználó

* Böngészés
* A rögzített termékek között keresési lehetőség
* A rögzített termékek kilistázhatóak kategória alapján
* Vásárlói ’kosár’ használata
* Kijelentkezés

## 2.2. Látogató

* Böngészés
* A rögzített termékek között keresési lehetőség
* A rögzített termékek kilistázhatóak kategória alapján
* Legyen képes regisztrálni (név, email, jelszó)
* Legyen képes bejelentkezni

A látogató és a bejelentkezett vásárló is tud az oldalon a termékek és szolgáltatások között böngészni, termékek között váltani. Mindketten képesek keresni is az oldalon terméket, és tudják a termékeket kategóriánként kilistázni.

A látogatónak, ha vásárlási szándéka van, előbb regisztrálnia kell, hogy a kosárba rakott termékeket megvásárolhassa. A regisztrációkor meg kell adnia bizonyos adatokat szállítással és számlázással kapcsolatban. A vásárló már nem regisztrálhat, mert regisztrált fiókkal rendelkezik.

Regisztráció után a látogató bejelentkezhet. Bejelentkezés után vásárlóként kezeljük a látogatót.

A regisztrált felhasználó tud terméket a kosárba helyezni, termék mennyiségét megadni.

Vásárolni és szolgáltatást igényelni csak a bejelentkezett felhasználó tud, mivel szükségesek a regisztráláskor megadott adatok.

Kijelentkezés lehetősége csak a bejelentkezés után érhető el, ezért kijelentkezni csak a vásárló tud bejelentkezés után bármikor.

# 3. Magas szintű rendszerterv

A Plant Planet alkalmazás fejlesztése az MVC (Model-View-Controller) szemléletmód alapján történt, mely biztosítja a felhasználói felület, az üzleti logika és az adatkezelés elkülönítését. Ez lehetővé teszi a rendszer átláthatóságát, könnyebb karbantartását és jövőbeni bővíthetőségét. A View réteg egy mobilalkalmazás (pl. React Native), amely a növénygondozási funkciókat, közösségi interakciókat és boltmodult jeleníti meg. A Controller réteg egy REST API-n keresztül biztosítja a logikai feldolgozást, míg a Model réteg relációs adatbázisban (MySQL) tárolja a felhasználói, növényi, közösségi és rendelési adatokat. Az architektúra moduláris felépítése lehetővé teszi a jövőbeni funkciók egyszerű hozzáadását.

A képen diagram, vázlat, sor, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

MVC modell

# 4. Screenek

Egy internetes alkalmazás készítésekor fontos több szempontot is figyelembe venni. A végtermék olyan felhasználóknak készül, akikkel közvetlen kapcsolat nem áll fent, így ha a felhasználó úgy érzi nem kapott kellő információt az oldalon, akkor lehet nem fog vásárolni, és vissza sem tér az oldalra. Ezért kiemelkedően fontos az oldal kinézetével és tartalmával kapcsolatban, hogy a látogatók bizalmát elnyerje, könnyen kezelhető és logikus felépítésű legyen. Minden információ egyértelmű legyen, például az elérhetőségek feltüntetése, a termékekről elég információ legyen feltűntetve. Az ilyen alapvető adatok hiánya miatt a vásárló más boltot választhat.

A képen szöveg, képernyőkép, szoftver, Weblap látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Kezdő képernyő terv

A termékek megjelenítésénél a kezdeti szakaszában még bootstrap slider-ben gondolkodott a csapat, de ezt elvetettük hamar, mert látványos volt, de rontotta az átláthatóságát a termékeknek. Végül az oldalon megjelenő termékeket kis dobozokban gondoltuk megjeleníteni, mert így egymás mellett és alatt több termék megjeleníthető és összehasonlítható, valamint az oldal kellően áttekinthető.

A képen szöveg, képernyőkép, szoftver, Webhely látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Termékek megjelenítésének oldal terve

A tervezés szakaszában is egyértelmű volt, hogy a regisztrációs felülethez egy letisztult, egyszerű megjelenésű oldalt képzelünk el.

A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

# 5. Modellek

Az adatok tárolására több modellre is szükség van, úgy mint regisztrált felhasználók, termék kategóriák, termékek, szolgáltatások.

**cart {**

**• id (int)**

**• user\_id (int)**

**• plant\_id (int, nullable)**

**• pot\_id (int, nullable)**

**• price (decimal(10,2))**

**• amount (int)**

**• subscription\_plan\_id (int, nullable)**

**• is\_subscription (boolean)**

**}**

**comments {**

**• id (int)**

**• user\_id (int)**

**• plant\_id (int)**

**• title (String)**

**• comment\_text (Text)**

**• rating (tinyint)**

**• profile\_picture (Text, nullable)**

**• created\_at (timestamp)**

**}**

**order\_item {**

**• id (int)**

**• order\_id (int)**

**• user\_id (int)**

**• plant\_id (int)**

**• pot\_id (int)**

**• subscription\_plan\_id (int)**

**• subscription (boolean)**

**• amount (int)**

**• price (double precision)**

**}**

**orders {**

**• order\_id (int)**

**• name (String)**

**• email (String)**

**• address (String)**

**• city (String)**

**• zipcode (String)**

**• phone\_number (String)**

**• payment\_method (String)**

**• total\_price (decimal(10,2))**

**• order\_date (timestamp)**

**}**

**plants {**

**• id (int)**

**• name (String)**

**• images (JSON)**

**• description (Text)**

**• price (decimal(10,2))**

**• light (String)**

**• water (String)**

**• humidity (String)**

**• temperature (String)**

**• fertilizing (String)**

**• re\_potting (String)**

**• cleaning (String)**

**• propagation (String)**

**}**

**pots {**

**• id (int)**

**• name (String)**

**• image (Text)**

**• price (decimal(10,2))**

**}**

**repot\_reminders {**

**• id (int)**

**• order\_item\_id (int)**

**• email (String)**

**• plant\_name (String)**

**• remind\_at (timestamp)**

**• sent (boolean)**

**}**

**subscription\_plans {**

**• id (int)**

**• name (String)**

**• images (JSON)**

**• description (Text)**

**• type (Enum: 'RANDOM\_PLANT' or 'CARE\_TIPS')**

**• price (decimal(10,2))**

**}**

**user\_subscriptions {**

**• id (bigint)**

**• user\_id (int)**

**• plan\_id (int)**

**• interval\_days (int, nullable)**

**• start\_date (datetime)**

**• next\_trigger\_date (datetime, nullable)**

**• status (Enum: 'ACTIVE', 'PAUSED', 'CANCELLED')**

**}**

**users {**

**• id (int)**

**• name (String)**

**• email (String)**

**• password (String)**

**• profile\_image (Text, nullable)**

**}**

# 6. Alkalmazások kiválasztása

## 6.1. Front-end

* **React**

A React az egyik nyílt forráskódú JavaScript könyvtár. Interaktív felhasználói felületek felépítésére szolgál. Ez egy hatékony, deklaratív és rugalmas könyvtár. A Model-View- Controller (MVC) V, azaz View elemével foglalkozik. Ez nem egy teljes keret, hanem csak egy előlapi könyvtár. Lehetővé teszi összetett felhasználói felületek létrehozását vagy elkészítését, összetevőkként ismert, elkülönített és apró kóddarabok felhasználásával.

(<https://hu.education-wiki.com/9050114-what-is-react>)

* **Javascript**

A JavaScript (röviden JS) egy objektumalapú scriptnyelv, amelyet elterjedten használnak weboldalakon, a web programozási nyelve. A Javascript futásideje egyszálú, ami azt jelenti, hogy egyszerre csak egy darab kódot képes végrehajtani. Ennek a megoldására használunk async függvényeket.

* **Html**

A HTML (Hypertext Markup Language) egy leíró nyelv, melyet weboldalak elkészítésére használhatunk.

* **Bootstrap**

A Bootstrap egy nyílt forráskódú keretrendszer (framework), mely **HTML, CSS, JavaScript technológiákat használ.** Nagyon könnyen, és minimális energia befektetéssel tudunk jól kinéző, bármilyen képernyőméreten szépen megjelenő weboldalakat készíteni a Bootstrap segítségével.

* **axios**

Az Axios egy könyvtár, amelyet HTTP -kérések küldésére használnak a böngészőből Node és Express.js platformon keresztül.

## 6.2. Backend

* **Springboot**

A Java Spring Boot egy olyan, nyílt forráskódú eszköz, amely megkönnyíti a mikroszolgáltatások és webalkalmazások Java-alapú keretrendszerek használatával való létrehozását. A Boot bármely definíciója szerint a beszélgetésnek a Java nyelvvel kell kezdődnie, amely az alkalmazásfejlesztés egyik legnépszerűbb és legszélesebb körben használt fejlesztési nyelve és számítástechnikai platformja. A fejlesztők a világ minden táján a Java tanulásával kezdik meg a kódolás elsajátítását. A rugalmas és felhasználóbarát Java számos alkalmazás esetében a fejlesztők kedvence, a közösségi médiától, a webtől és a játékalkalmazásoktól kezdve a hálózatkezelésig és a nagyvállalati alkalmazásokig.

(https://azure.microsoft.com/hu-hu/resources/cloud-computing-dictionary/what-is-java-spring-boot)

* **bcrypt** – hashelés (jelszó)

A Bcrypt egy népszerű és megbízható módszer a jelszavak kivonására.

* **MySql**

A MySQL egy rendkívül népszerű relációs adatbázis-kezelő rendszer.

## 6.3. Adatbázis

* **MySQL szerver**

A MySQL egy nyílt forráskódú, relációs adatbázis-kezelő rendszer, amely az adatok strukturált tárolását és gyors lekérdezését teszi lehetővé. Leggyakrabban webalkalmazások adatbázisaként használják, például WordPress vagy online boltok mögött. Stabil, skálázható és széles körben támogatott megoldás..

## 6.4. Tesztelés

* **Java Unit tesztek és React teszt**

A Java unit tesztek egyes metódusok helyes működését ellenőrzik különállóan, míg a React tesztek komponensek megjelenését és viselkedését vizsgálják izoláltan.

# 7. Routing

PUT /user/change

|  |  |
| --- | --- |
| tervezett feladat | Felhasználói adatok módosítása |
| bemenet | Személyes adatok |
| kimenet | Módosított személyes adatok |

POST /user/register

|  |  |
| --- | --- |
| tervezett feladat | Regisztrálás |
| bemenet | Adatok megadása |
| kimenet | Regisztrált felhasználó |

POST /user/logout

|  |  |
| --- | --- |
| tervezett feladat | Kijelentkezés |
| bemenet |  |
| kimenet | Kijelentkeztetés |

POST /user/login

|  |  |
| --- | --- |
| tervezett feladat | Bejelentkezés |
| bemenet | Adatok megadása |
| kimenet | Bejelentkezés |

PUT /cart/update

|  |  |
| --- | --- |
| tervezett feladat | Kosár frissítése |
| bemenet |  |
| kimenet | Kosár frissítve |

POST /cart/add

|  |  |
| --- | --- |
| tervezett feladat | Kosárhoz adás |
| bemenet | Hozzáadás |
| kimenet | A termék kosárba helyezése |

GET /cart/view

|  |  |
| --- | --- |
| tervezett feladat | Kosár lekérdezése |
| bemenet |  |
| kimenet | Kosár értéke |

GET /cart/total

|  |  |
| --- | --- |
| tervezett feladat | Kosár teljes tartalma |
| bemenet |  |
| kimenet | Kosár teljes tartalma |

DELETE /cart/remove/{cartItemId}

|  |  |
| --- | --- |
| tervezett feladat | Termék kosárból való eltávolítása |
| bemenet | Eltávolítás |
| kimenet | Sikeresen kikerült a kosárból |

POST /subscriptions/subscribe

|  |  |
| --- | --- |
| tervezett feladat | Előfizetés |
| bemenet | Választott csomag |
| kimenet | Sikeres előfizetés |

GET /subscriptions/plans

|  |  |
| --- | --- |
| tervezett feladat | Csomagok lekérdezése |
| bemenet |  |
| kimenet | Lehetséges csomagok kijelzése |

GET /subscriptions/plans/{id}

|  |  |
| --- | --- |
| tervezett feladat | Adott csomag megtekintése |
| bemenet |  |
| kimenet | Csomag megtekintése |
|  |  |

GET /subscriptions/my-subscriptions

|  |  |
| --- | --- |
| tervezett feladat | Előfizetéseim megtekintése |
| bemenet |  |
| kimenet | Aktív előfizetéseim |

DELETE /subscriptions/cancel/{id}

|  |  |
| --- | --- |
| tervezett feladat | Előfizetés lemondása |
| bemenet | Lemondás |
| kimenet | Sikeres lemondás |

POST /orders

|  |  |
| --- | --- |
| tervezett feladat | Megrendelések |
| bemenet |  |
| kimenet |  |

POST /comments/post

|  |  |
| --- | --- |
| tervezett feladat | Hozzászólás írás |
| bemenet | Hozzászólás megírása |
| kimenet | Sikeres közzététel |

GET /comments/{plantId}

|  |  |
| --- | --- |
| tervezett feladat | Hozzászólások megtekintése |
| bemenet | Adott növény hozászólásai |
| kimenet | Hozzászólások kijelzése |

POST /chat

|  |  |
| --- | --- |
| tervezett feladat | Support Bot |
| bemenet | Support Bot-nak írni |
| kimenet | Sikeres válasz |

GET /pots

|  |  |
| --- | --- |
| tervezett feladat | Cserepek megtekintése |
| bemenet | Adott cserép kiválasztása |
| kimenet | Cserép kiválasztva |

GET /plants

|  |  |
| --- | --- |
| tervezett feladat | Növények felsorolása |
| bemenet |  |
| kimenet | Összes növény |

GET /plant/{id}

|  |  |
| --- | --- |
| tervezett feladat | Adott növény adatai |
| bemenet | Növény kiválasztása |
| kimenet | A növény adatai |

GET /email/send

|  |  |
| --- | --- |
| tervezett feladat | Email küldés |
| bemenet | Email szövege |
| kimenet | Sikeres email küldés |

# 8. Implementáció

Az implementáció a tervezést követő folyamat. A tervezés során megtörténik a feladatokra bontás, valamint az egyes feladatokban érintett funkcionalitás és szerkezet modellezése. Az implementáció során a modellekből készítünk működő, futtatható kódot.   
(https://okt.sed.hu/rf2/gyakorlat/tananyag/implementacio/implementacio/)

# 9. Tesztelés

A felhasználó szempontjából a rendszer futtatásához elegendő bármely olyan operációs rendszer, amelyen futtatható HTML, CSS és Javascript technológiával készült alkalmazás megjelenítésére képes böngésző.

A fejlesztés alatt a rendszer tesztelése desktop eszközön több böngészőben (Google Chrome, Mozilla Firefox) Windows 10/11 illetve Linux operációs rendszer alatt történt. Továbbá mobil eszközön Android operációs rendszer alatt is teszteltük.

A szerveroldalról nézve a rendszernek szüksége van egy olyan platformra, ami MySQL szervereket képes futtatni, a fejlesztés során egy helyi localhost-al oldottuk meg.

A feladat végrehajtása során folyamatosan jelen volt a tesztelés. Már a statikus oldal megtervezésekor és készülésekor tesztelve volt a megjelenés reszponzivitása, az olvashatóság, a láthatóság.

# 10. Bevezetés, éles üzemmód

- Az elkészült munka megtekintéséhez szükséges a MySQL letöltése.

Látogassunk el a következő oldalra:

<https://www.mysql.com/downloads/>, és onnan töltsük le az

operációs rendszerünknek megfelelő MySQL-verziót.

- Telepítsük fel. Hozzunk létre egy plantplanetdb nevű adatbázist.

- Töltse le a Windows telepítőt a Node webhelyről. Miután

kiválasztotta az igényeinek megfelelő verziót, futtassa a telepítőt.

- Indítunk el a Visual Studio Code alkalmazást, a Front

mappát nyitjuk meg, IntelliJ-ben a backend mappát kell megnyitni az

Open Folder menüponttal.

- Visual Studio Code-ban a Terminal-ban npm –i paranccsal

telepítsük az összes modult.

- Visual Studio Code-ban npm run dev paranccsal indítsuk az

alkalmazást

- Ezután a http://localhost:5173 -es címen lehet elérni

az alkalmazást.

A teljes alkalmazás Git repository-ja:

<https://github.com/DobraiDavid/PlantPlanet>