## BGSZC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Finomságok a mi mesterségünk

Receptek4You

Konzulens tanár: Bencze István Készítette: Név: Horváth Ákos, Varju Richárd Dániel

Tartalom

[1 Bevezetés 3](#_Toc85723173)

[1.1 Feladat leírás 3](#_Toc85723174)

[1.2 A felhasznált ismeretek 3](#_Toc85723175)

[1.3 A felhasznált szoftverek 3](#_Toc85723176)

[2 Felhasználói dokumentáció 4](#_Toc85723177)

[2.1 A program általános specifikációja 4](#_Toc85723178)

[2.2 Rendszerkövetelmények 4](#_Toc85723179)

[2.2.1 Hardver követelmények 4](#_Toc85723180)

[2.2.2 Szoftver követelmények 4](#_Toc85723181)

[2.3 3. A program telepítése 4](#_Toc85723182)

[2.4 A program használatának a részletes leírása 5](#_Toc85723183)

[3 Fejlesztői dokumentáció 6](#_Toc85723184)

[3.1 Az alkalmazott fejlesztői eszközök 6](#_Toc85723185)

[3.2 Adatmodell leírása 6](#_Toc85723186)

[3.3 Részletes feladatspecifikáció, algoritmusok 6](#_Toc85723187)

[3.4 Tesztelési dokumentáció 7](#_Toc85723188)

[4 Összefoglalás 8](#_Toc85723189)

[4.1 Önértékelés 8](#_Toc85723190)

[4.2 Továbbfejlesztési lehetőségek 8](#_Toc85723191)

[5 Felhasznált irodalom 9](#_Toc85723192)

[6 Ábrajegyzék 10](#_Toc85723193)

# Bevezetés

* A felhasználói dokumentáció célja, hogy segítséget adjon a leendő felhasználónak a program telepítésével és használatával kapcsolatos minden probléma megoldásában, segítse a program kezelésének az elsajátítását.

## Feladat leírás

* Mestermunka – A weboldal célja, hogy a felhasználók saját receptjeiket oszthassák meg az oldalon kategóriák, érzékenységek valamit preferenciák kiválasztása szerint, valamint elmenthessék maguknál. Először egy nyitó oldal fog megjelenni egy pár másodpercre, ami átirányítja a felhasználót a főoldalra. A főoldalon láthatja a már felrakott recepteket és a bejelentkezés lehetőségét. Az oldal főbb lehetőségeit és funkciót, ahhoz, hogy láthassa és használhassa, (mint például a receptfeltöltést) először regisztrálni kell, majd be kell jelentkezni, eközben el kell fogadnia az általános szabályokat és ezzel aláírja, hogy tudomásul vette őket. A regisztráció és a bejelentkezés után a látogatónak kétféle felhasználói joga lehet. Az egyik a sima felhasználói felület, ami receptfeltöltésére, saját profiljának megtekintésére, egy adott recept keresésére (a menüben található kereső mezővel lehetséges a receptkeresés) és már az előre feltett receptek megtekintésére ad jogosultságot. A másik felhasználói jog az admin, aki jóváhagyhatja a receptek feltöltését vagy elutasíthatja azt, valamint megtekintheti az oldalra beregisztrált felhasználok adatait, valamint azt, hogy hány receptet töltött fel az adott felhasználó, ha valami gyanúsat vagy trollkodásra utaló jelet talál az adott felhasználó által, akkor törölheti a profilját emellett ő adhat admin jogot egy adott felhasználónak, továbbá mindenre lehetősége, amire egy sima felhasználónak. Az oldal egyik fő célja az, hogy a felhasználók ne tehessenek fel olyan receptet, aminek hozzávalói nem illenek össze, esetleg rossz mennyiséget használtak vagy rossz hozzávalót írtak bele a receptbe (netán egy receptet többször is megpróbáltak feltölteni). Ezeknek a vizsgálatát egy specialista (egy séf, aki magas végzettséggel rendelkezik és mind a mai napig ebben a szakmában tevékenykedik) és mi, a weboldal készítői fogjuk végezni. A weboldal másik fő célja, hogy az ételérzékenyek és a preferencia szerint étkezők is megtalálják a számukra legideálisabb recepteket. Ezek vizsgálatában pedig kikértük több ételérzékenységben szenvedő, valamint választott táplálkozás szerint étkező emberek véleményét és akaratát, hogy a számukra a lehető legideálisabb recepteket láthassák (köztük én is egy ilyen ételérzékenységben szenvedek).

## A felhasznált ismeretek

**Programozási ismeretek**

* Algoritmusok (keresési algoritmusok, rendezési algoritmusok)
* Adatstruktúrák (tömbök, listák, sorok)
* Objektumorientált programozás (OOP) (osztályok, objektumok)
* Funkcionális programozás (magas szintű függvények, alacsony színtű függvények)
* Hibakezelés (try-catch blokkok kivételkezelés, hibakódok)
* Tesztelés (unit tesztelés, integrációs tesztelés)

**Webfejlesztési ismeretek**

* HTML/CSS (HTML5, CSS3, Flexbox, Grid layout)
* JavaScript (DOM manipuláció, eseménykezelés, AJAX, JSON)
* Frontend keretrendszerek (React, Bootstrap)
* Backend fejlesztés (Node.js)
* API fejlesztés (RESTful API)
* Weboldal fejlesztés (page speed, responsive design)
* Webbiztonság (HTTPS)
* Webalkalmazás architektúrák (MVC)

**Adatbázis-kezelés**

* Relációs adatbázisok (MySQL)
* Adatbázis-tervezés (normálformák, relációs modellezés, idegen kulcsok)

**Hálózatok és rendszerek**

* Adatbázis-tervezés (normálformák, relációs modellezés, idegen kulcsok)
* Webes protokollok (HTTPS)
* Cloud computing (Google Cloud)

**Operációs rendszerek**

* Windows operációs rendszer (Windows 10, PowerShell)
* Fájlkezelés és fájlrendszerek (NTFS, FAT32)
* Hibakeresés és naplózás (event viewer, console log, tesztdokumentáció)

**Operációs rendszerek**

* Verziókezelés (Git, GitHub)
* Kódminőség és karbantartás (kód tesztelés)
* Dokumentáció és API tervezés (ER, UML, OpenAPI, Postman)

**Adattudomány és analitika**

* ETL folyamatok (adatgyűjtés, adattranszformáció, adatbetöltés)

**Szoftverfejlesztési eszközök és keretrendszerek**

* Integrált fejlesztői környezetek (IDE) (Visual Studio)
* API tesztelés (Postman)
* Unit tesztelés (JUnit)

## A felhasznált szoftverek

* Irodai alkalmazások: Microsoft Office (Word, PowerPoint), WPS Office, Google Workspace (Docs, Sheets, Drive, Gmail)
* Programozás és fejlesztés: Visual Studio, Git, GitHub, Postman (API tesztelés)
* Projektmenedzsment és kollaborációs eszközök: Trello, Microsoft Project, Discord, Microsoft Teams
* Adatbázis-kezelő rendszerek: MySQL, (phpMyAdmin)
* Webes fejlesztés: React, Bootstrap
* Kommunikációs eszközök: Microsoft Teams, Discord, Messenger (Facebook)
* Cloud alapú tárolás és megosztás: Google Drive, OneDrive
* Antivírus és biztonsági szoftverek: Avast, McAfee, Windows Defender
* Videókonferencia eszközök: Microsoft Teams
* Mesterséges intelligencia: ChatGPT

# Felhasználói dokumentáció

## A program általános specifikációja

* Program főbb funkciói és jellemzői:

1. Regisztráció és bejelentkezés
2. Bejelentkezés során az általános szabályok elfogadása (ez szükséges a továbblépéshez)

**Felhasználói felület:**

1. Recept feltöltése kategóriák kiválasztásával
2. Recept elmentése a felhasználó profiljába a kiválasztott kategóriákkal
3. Receptek név alapú keresése
4. A felhasználó saját profiljának megtekintése
5. A felhasználó a profilján keresztül képes saját felhasználónevének módosítására

**Admin felület:**

1. Adott recept elfogadása (és ezáltal felkerül az oldalra)
2. Adott recept elutasítása (és ezáltal nem kerül fel az oldalra, az admin törölheti)
3. Egy adott felhasználó profiljának megtekintése
4. Egy adott felhasználó profiljának törlése
5. Az oldal karbantartása
6. Admin jogi felhatalmazás az általa kiválasztott felhasználóknak, valamint annak visszavonása

**Könnyű navigáció**

1. Egyszerű és felhasználóbarát a weboldal kialakítása.
2. A menü világos, jól felépített, de mégis egyszerű struktúrával rendelkezik, és minden kategória könnyen elérhető.

**Reszponzív dizájn**

1. Az oldal automatikusan alkalmazkodik az eszközhöz, legyen az okostelefon, tablet vagy asztali számítógép. Bárhol és bármikor kényelmesen böngészhetnek anélkül, hogy zavaró lenne a dizájn.

**Szép és egyszerű megjelenés**

1. Képek és gombok segítenek a receptek bemutatásában, amelyek inspirálnak, és megmutatják, hogyan néz ki a kész étel.
2. Az ételfotók színesek, részletesek.

**Recept keresés és szűrés**

1. Gyorsan megtalálható minden, amit keresnek, például étkezési típus (preferencia), alapanyagok, allergén szempontok alapján. A szűrők könnyen használhatóak, így gyorsan elérhetők az igényeiknek megfelelő receptek.

**Lépésről lépésre útmutatók**

1. A receptek könnyen követhetőek, és világos utasításokkal vannak ellátva. A képek segítenek megérteni a főzés folyamatát, különösen azoknak, akik újabbak a főzés világában.

## Rendszerkövetelmények

## 3. A program telepítése

1. **Szoftverkövetelmények**
2. **Frontend (React):**

* React.js (v18 vagy újabb)
* Tailwind CSS a stílushoz
* Framer Motion az animációkhoz
* React Router DOM az oldalak közötti navigációhoz
* Axios vagy Fetch API az adatok lekéréséhez
* Vite vagy Webpack a fejlesztési környezethez

1. **Backend (Node.js + Express):**

* Node.js (v18 vagy újabb)
* Express.js szerveroldali útvonalakhoz
* JWT vagy Session alapú hitelesítéshez
* bcrypt jelszó hasheléshez
* CORS és egyéb biztonsági beállítások

1. **Adatbázis:**

* MySQL (v8.0+)
* Sequelize vagy Prisma ORM (opcionális, de javasolt)
* Táblák: Users, Recipes, Favorites, PolicyAccepted

1. **Egyéb szoftverek és eszközök:**

* Git + GitHub (verziókövetés)
* Postman (API teszteléshez)
* Docker (konténerizálás – nem kötelező, de hasznos)
* Visual Studio Code vagy más IDE

1. **Hardverkövetelmények**
2. **Fejlesztői gép:**

* CPU: Legalább 4 magos (pl. Intel i5 / Ryzen 5)
* RAM: Min. 8 GB, ajánlott 16 GB
* Tárhely: 50+ GB SSD (projektfájlokhoz, node\_modules, képekhez)
* Monitor: 1080p vagy jobb felbontás
* Internetkapcsolat: Stabil, 25 Mbps vagy jobb

1. **Webszerver (ha nem felhőben):**

* CPU: 2+ magos (VPS esetén is elég)
* RAM: Minimum 4 GB (optimális: 8 GB)
* Tárhely: 50–100 GB SSD
* Hálózat: Min. 1 Gbps kapcsolat

1. **Rendszerkövetelmények**
2. **Funkcionális elemek:**

* Felhasználói regisztráció / bejelentkezés (JWT/session alapú)
* Receptfeltöltés képpel és leírással
* Receptek mentése (kedvencek)
* Receptek böngészése
* Szabályzat elfogadása a belépés előtt
* Animált belépőképernyő (React + Framer Motion)
* Reszponzív dizájn mobil/tablet/desktop eszközökre

1. **Nem-funkcionális követelmények:**

* Biztonságos adatkezelés (jelszó hash, XSS, CORS)
* Gyors válaszidő (REST API-k optimalizálása)
* Magas rendelkezésre állás (CI/CD és backup támogatás)
* Bővíthetőség: további modulok könnyű integrálása

1. **Deploy lehetőségek:**

* Frontend: Vercel / Netlify / saját VPS
* Backend + DB: Railway / DigitalOcean / Hetzner / Docker konténer
* SSL: Let's Encrypt

## A program használatának a részletes leírása

* A weboldal részletes leírása (működése, használata):

**Az oldalon található funkciók:**

**1. Bejelentkezés és Fiók Létrehozása**

* A fiók létrehozása egy nyitó oldal vezeti majd a főoldalra, ahol a felhasználó láthatja a már az oldalra felvitt recepteket és a bejelentkezés lehetőségét. A menü főbb funkciói és lehetőségei csak a regisztráció és a bejelentkezés, valamint az általános szabályzatok elfogadása után jelenik meg.

**Innen kezdetét veheti:**

#### **1.1. Fiók létrehozása**

* Amikor először látogatja meg a weboldalt, az alábbi lépéseket kell követni a fiók létrehozásához:

1. Nyissa meg a **"Regisztráció"** oldalt a kezdőképernyőn.
2. Töltse ki a szükséges adatokat:
   1. **Felhasználónév**: A felhasználó által kitalált, megadott név.
   2. **Email cím**: Az email cím, amellyel a fiókot regisztrálja.
   3. **Jelszó**: A jelszó, amely biztosítja a fiók biztonságát.
3. Kattintson a **"Nincs fiókod? Regisztráció"** gombra.

A sikeres regisztráció után kiírja a szerver a weboldalon, hogy **„Sikeres regisztráció”**.

#### **1.2. Bejelentkezés**

* A regisztráció után a bejelentkezéshez kövesse az alábbi lépéseket:

1. A főoldalon a regisztráció után automatikusan feldobja a bejelentkezés lehetőségét, persze, ha újra regisztrálni szeretne, ismét csak rákattint a **„Nincs fiókod? Regisztráció”** gombra. Viszont amennyiben nem kíván ismét regisztrálni és visszament a regisztrációs menühöz, akkor kattintson a **„Már van fiókod? Bejelentkezés”** gombra kattint, adja meg a regisztrált email címet és jelszót.
2. A **„Bejelentkezés”** gombra kattintás előtt felugrik egy szabályzati leírás azzal kapcsolatban, hogy mikre van lehetősége az oldalon és lehetőség szerint mit ne tegyen a weboldalon. A továbbjutáshoz szükséges ennek elfogadása, amennyiben nem kívánja elfogadni, akkor kattintson a **„Kilépek”** gombra, viszont így nem fog tudni bejelentkezni. Amennyiben elolvasta és elfogadja a használati szabályzatot, annyiban kattintson az **„Elfogadom”** gombra és meg is kezdheti a bejelentkezést.
3. Kattintson a **"Bejelentkezés"** gombra, és belépteti a weboldalra, ha minden adat megegyezik, egyébként kiírja a szerver az oldalon, hogy mi lehet a gond (pl.: **„hibás jelszó”**). Ha kihagyja az egyik mezőt, akkor az oldal kéri annak kitöltésére. Ha az email címből kihagyja a **„@” -**ot, arra is figyelmezteti, hogy hiányzik. Ha nem létezik az email, akkor arra is figyelmezteti az oldal.

### 2. Navigáció és Menü

* A főoldalon található menü segít eligazodni a weboldalon. Az alábbiakban bemutatjuk a menü főbb pontjait:

#### **2.1. Receptek**

1. Az oldal legfontosabb része a receptgyűjtemény, amelyet a **"Receptek"** menüpont alatt talál.
2. A receptek kategóriák szerint rendezhetők (pl.: konyha szerint, preferencia szerint, érzékenység és étkezési idő szerint)
3. A recept felvitelekor tudja ezeket a lehetőségeket kiválasztani és feltölteni a receptet.
4. A recept feltöltése azonban nem történik meg azonnal, hanem egy **„Admin”** felületre kerül, ahol az erre felhatalmazott személy egy hosszú ellenőrzés után eldöntheti, hogy engedélyezi ennek a feltételét vagy sem (pl.: ha talál benne valami oda nem illőt vagy már van fent egy ilyen recept), ezeknek a recepteknek a vizsgálatához egy profi szakértőnek a folyamatos segítségét kértük.

#### **2.2. Keresés**

1. A képernyő tetején található keresőmező segítségével gyorsan megtalálhatja a kívánt receptet.
2. Írja be a keresett étel nevét (pl.: „Tojásos Bruschetta” vagy „Mac and Cheese”).
3. A rendszer azonnal megjeleníti az összes releváns találatot (a keresés a nem pontos recept nevére is találatot ad, ha például elgépel egy ékezetet).

#### **2.3. Felhasználói profil**

1. A **„Profil”** menüpont alatt tekintheti meg saját fiókadatait, elmentett kedvenc receptjeit, valamint módosíthatja a beállításokat.
2. Itt található lehetőség a jelszó megváltoztatására is.

### 3. Recept Megtekintése és Mentése

#### **3.1. Recept megtekintése**

1. A keresett recept/nem keresett recepteknél található 3 gomb: a **„Több”** gombra kattintva megismerheti az adott recept elkészítési menetét, a **„Bővebb információ”** gombra kattintva megtud róluk minden fontos információt (pl.: A hozzávalóik nevét, a mennyiségét és a mértékegységét, az adott étel minden érzékenységgel rendelkezik és, hogy minden preferencia étkezésnek felel meg (itt is van olyan opció hogy egyik sem szerepel benne), a **„Mentés”** gombra kattintva eltudja menteni a profiljában azokat a receptek, amik tetszenek vagy amiket kipróbálna(ha elmentett egy receptet, azt a weboldal jelzi számunkra) - (egyszerre több receptet is elmenthet és csak azok jelennek meg amiket elmentett, a profilban pedig eltávolíthatja az **„Eltávolítás”** gombra kattintva azt a receptet amit véletlenül tett be vagy csak egyszerűen nem szeretné már ott látni. A profilban is megtekinthetőek az információk a recepttel kapcsolatban a **„Több”** és a **„Bővebb információ”** gombokra kattintva.
2. Az oldal tartalmazza a recept összetevőit, az elkészítés lépéseit, valamint szükséges időt.
3. A recept oldalon videók és képek segítik a főzést, lépésről lépésre bemutatva az étel elkészítését.

#### **3.2. Recept mentése**

1. Ha megtetszett egy recept, mentheti azt a **„Mentés”** gombra kattintva.
2. Az elmentett recepteket a **"Profil"** menüpont alatt találja.

#### **3.2. Admin felület és kezelése**

1. Az admin képes megvizsgálni a felvitt receptek részleteit a **„Több”** gombra kattintás segítségével és
2. A képeknél felül található egy csúszka, ennek segítségével engedélyezheti a recept felöltését, ha nem szeretné engedélyezni, akkor nincsen a csúszkával semmi dolga (nem nyúl hozzá)
3. Ha az admin törölni kívánja az adott receptet, akkor ezt megteheti azzal, ha a **„Törlés”** gombra kattint.
4. Az admin megtekintheti a beregisztrált felhasználók adatait és azt, hogy hány receptet töltött fel az adott felhasználó
5. Az adminnak lehetősége van a felhasználók profiljának törlésére
6. Az adminnak lehetősége van az általa kiválasztott profiloknak admin jogot biztosítani, valamint vissza is vonhatja azt

### 4. Közösségi Funkciók

#### **4.1. Értékelés és Vélemények**

1. Minden recept alatt található egy értékelő rendszer, ahol 1-5 csillag között adhatja meg a recept minőségét.
2. Az értékeléshez hozzáfűzhet saját véleményt is. Írja le tapasztalatait, és segítse más felhasználókat a döntésben.

# Fejlesztői dokumentáció

* A fejlesztői dokumentáció célja, hogy a segítse program logikájának, illetve a program kódjának a megértését, illetve a program továbbfejlesztését. Szakemberek számára készül, elvárás tehát a pontosság és a szakmai jellegű stílus. Ezzel célunk *“segíteni a programok és rendszerek működésének és használatának megértésében, valamint a fejlesztők közötti hatékony együttműködésben.”* - Idézet forrása: [06. óra - Rendszerfejlesztés II.](https://okt.inf.szte.hu/rf2/tananyag/06.html)

## Az alkalmazott fejlesztői eszközök

* A felhasznált eszközök (fejlesztői környezet): Visual Studio Code
* Programozási nyelvek:

1. Html (hiperszöveges jelölőnyelv+leíró nyelv)
2. CSS (rgymásra épülő stíluslapok +stílusleírónyelv)
3. JSON (egyszerűen olvasható szöveg alapú szabvány)
4. JavaScript (prototípus-apaú szkriptnyelv)
5. JSX (szintaxis kiterjesztés)

* Dokumentáció készítéséhez felhasznált eszközök:

1. Word (szövegszerkeztő és dokumentum készítő)
2. Képmetsző (képek kimásolása és elmentése)
3. Paint (képek szerkeztése)

* Diagramkészítő szoftver: SmartDraw (Egységes vizuális együttműködési alkalmazás) - ER, UML
* Verziókezelő szoftver: Git (nyílt forráskódú) - GitHub (Git segítségével verziókövetést nyújt)
* Adatbázis-kezelő rendszer: MySql (SQL-alapú relációs) - phpMyAdmin (nyílt forráskódú webes alapú eszköz)
* Keretrendszer:

1. React (nyílt forráskódú JavaScript könyvtár - webalkalmazás)
2. Bootstrap (nyílt forráskódú - UI/UX)

* Más által kifejlesztett modul (Adatbázis és web összeköttetése - videós segítség, aminek forrása: <https://www.youtube.com/watch?v=Q3ixb1w-QaY> ) - publikusan megosztott videó, ami direkt erre a célra lett elkészítve(példa videó) - A szöveg leírása : “Hogyan csatlakoztassuk a React JS-t a MySQL adatbázishoz a Node.JS/Express.js használatával. Hogyan implementáljunk egy szervert ReactJS és MySQL alkalmazásokhoz. ma bemutatok egy nagyon egyszerű React alkalmazást, amely integrálva van a MySql DB-vel egy egyszerű alkalmazáson keresztül. React Node.js MySQL oktatóanyag kezdőknek. Csomópont a React és a MySQL oktatóanyaggal.”.

A videó készítője: <https://www.youtube.com/@codewithyousaf>

## Adatmodell leírása

Leírás az adatbázisunk elképzelésének és megvalósításának rövid menetéről

* Olyan adatbázist szeretnénk létrehozni, amely hozzávalókat és különböző felhasználok adatait, receptjeit tartalmazza.
* Tároljuk a felhasználó e-mail/telefonszámát, jelszavát, étrendjét (pl.: speciális étrend (gluténmentes…stb.) a hozzávalókat és a recepteket (ételek receptjeit) - (a recept hozzávalóit), valamint az ételek konyhai honlétét (pl.: ez az étel xy nemzetiségű konyháról származik).
* Feltételezzük, hogy egyféle receptet több felhasználó is feltehet (más-más elkészítési módszerrel, illetve hozzávalókkal és más névvel), valamint egy hozzávaló több receptben is benne lehet. Ha nincs kikötés az ételre vonatkozóan (pl.: laktózmentes…stb.), akkor bármilyen ételt választhat a felhasználó(elkészítésre). Az ételt egyértelműen azonosítja az, hogy milyen fogás (pl.: desszert, főétel…stb. - napszak) Egy adott nemzetiségű konyhán csak azonos nemzetiségű ételek kerülhetnek fel.
* Egy főzős weboldal, ahol hozzávalók kiválasztásával több recept megjelenne, amik különféle kategóriák szerint lennének megtalálhatók.
* A felhasználó saját receptjeit tudná elmenti és tárolni, illetve publikálni, ha az admin engedélyezi a számára. (A mestermunka fő kritériuma egy életbeli probléma megoldása à miénk célja, hogy szűrni tudjuk azokat a felhasználókat, akik mindenféle össze nem illő hozzávalókkal kreált receptet szeretnének közhírré tenni
* Az adatbázis “alaprajza” - végleges papíralapú elképzelésünk

RECEPTEK (receptek\_neve, keszites, Receptek\_id, kep, konyha\_osszekoto, napszak\_osszekoto)

1. receptek\_neve(varchar(100), NOT NULL)
2. keszites(text)
3. Receptek\_id(autoincrement)
4. kep(varchar(50), NOT NULL)
5. konyha\_osszekoto(int(10), NOT NULL)
6. napszak\_osszekoto(int(10), NOT NULL)

HOZZAVALOK (Hozzavalok\_id, hozzavalok\_neve)

1. Hozzavalok\_id(autoincrement)
2. hozzavalok\_neve(varchar(30), NOT NULL)

REGISZTRACIO (admin, felhasznalo\_id, e-mail, jelszo, felhasznalonev, szabalyzat, like)

1. admin(int(1), NOT NULL)
2. felhasznalo\_id(autioncrement)
3. e-mail(varchar(100), NOT NULL)
4. jelszo(varchar(255), NOT NULL)
5. felhasznalonev(varchar(30), NOT NULL)
6. szabalyzat(int(11), NOT NULL)
7. like (int(11), NOT NULL)

PREFERENCIA (etkezes, etkezes\_id)

1. etkezes(varchar(30), NOT NULL)
2. etkezes\_id(autoincrement)

KONYHA (nemzetiseg, konyha\_id)

1. nemzetiseg(varchar(30), NOT NULL)
2. konyha\_id(autoincrement)

OSSZEKOTO (hozzavalok\_id, receptek\_id, ervenyes, etrend\_id, mertekegyseg\_id, preferencia\_id)

1. hozzavalok\_id(int(11), NOT NULL)
2. receptek\_id(int(11), NOT NULL)
3. ervenyes(int(1), NOT NULL)
4. etrend\_id(int10), NOT NULL)
5. mertekegyseg\_id(int(10), NOT NULL)
6. preferencia\_id(int(11), NOT NULL)

NAPSZAK (idoszak, napszak\_id)

1. idoszak(varchar(20), NOT NULL)
2. napszak\_id(autoincrement)

SAJAT\_RECEPTEK (recept, profil)

1. recept(int(10), NOT NULL)
2. profil(int(10), NOT NULL)

MERTEKEGYSEG (mennyiseg, mertekegyseg, Mertekegyseg\_id)

1. mennyiseg(int(20), NOT NULL)
2. mertekegyseg(varchar(30), NOT NULL)
3. Mertekegyseg\_id(autoincrement)

ERZEKENYSEGEK (erzekenyseg, erzekenyseg\_id)

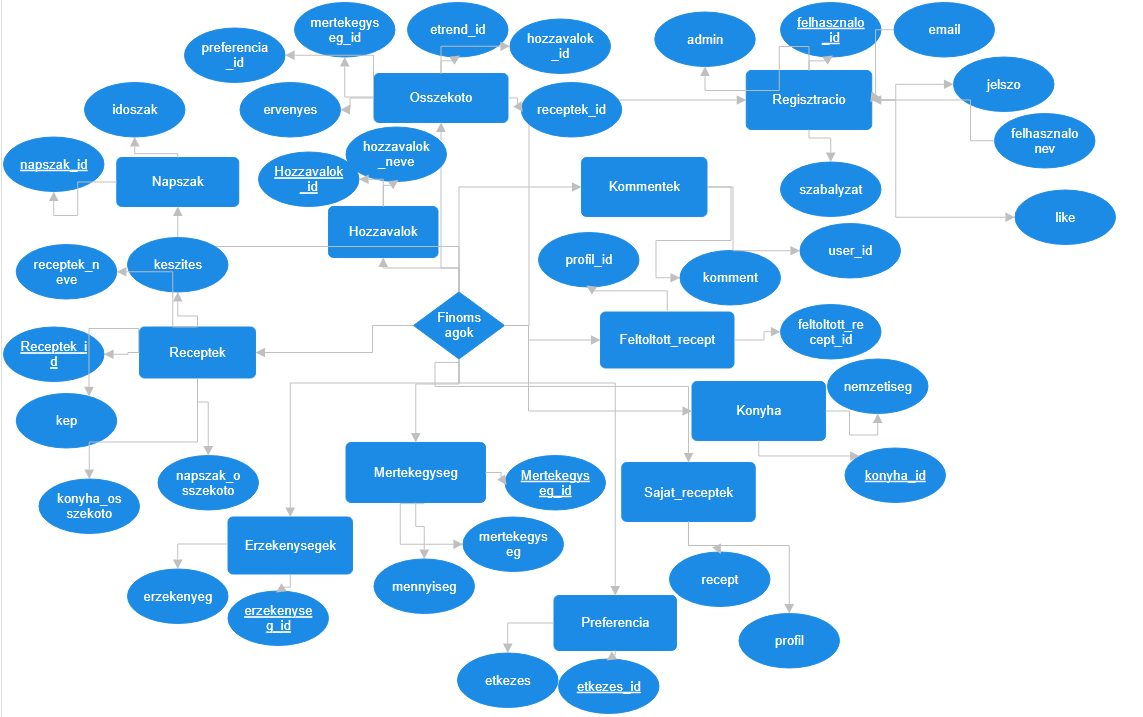
1. erzekenyseg(varchar(30), NOT NULL)
2. erzekenyseg\_id(autoincrement)

FELTOLTOTT\_RECEPT (profil\_id, feltoltott\_recept\_id)

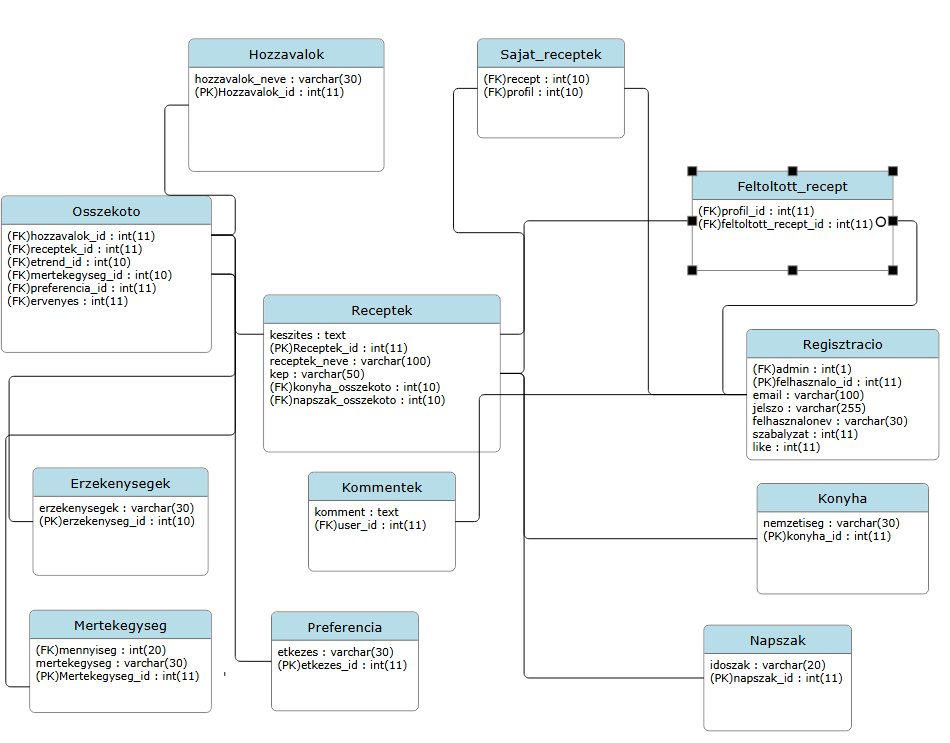
1. profil\_id(int(11), NOT NULL)
2. feltoltott\_recept\_id(int(11), NOT NULL)

KOMMENTEK (komment, user\_id)

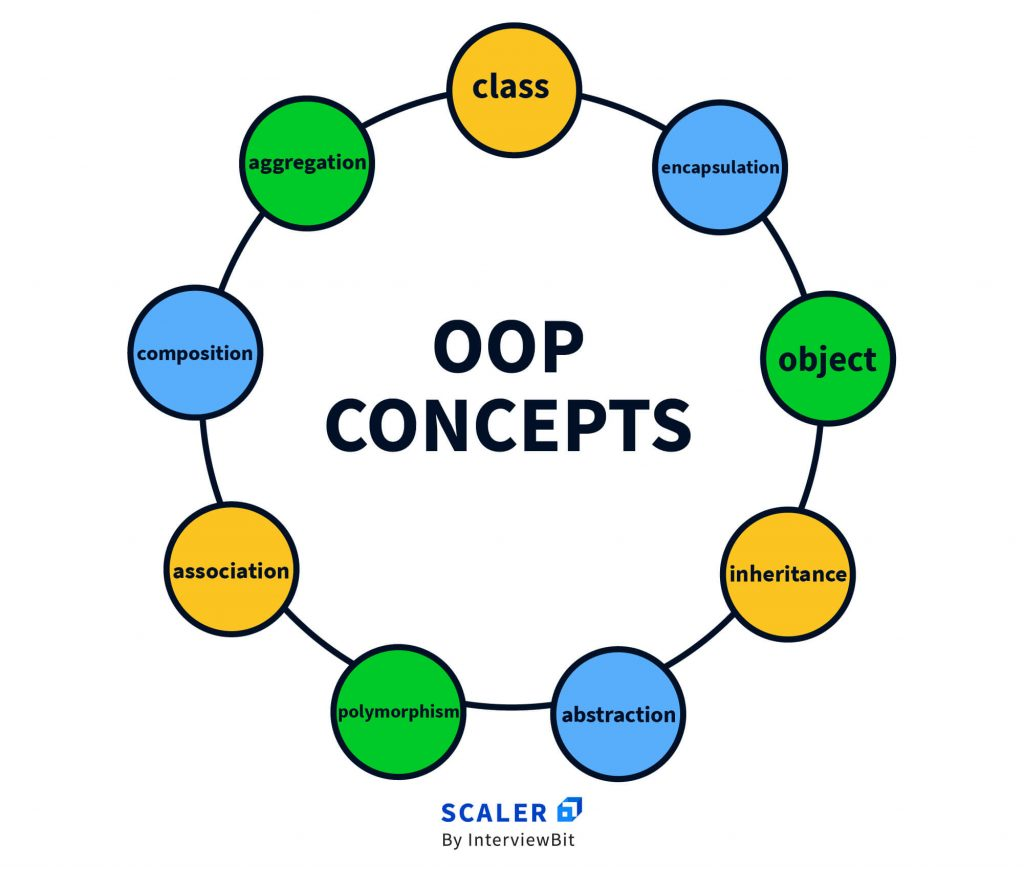
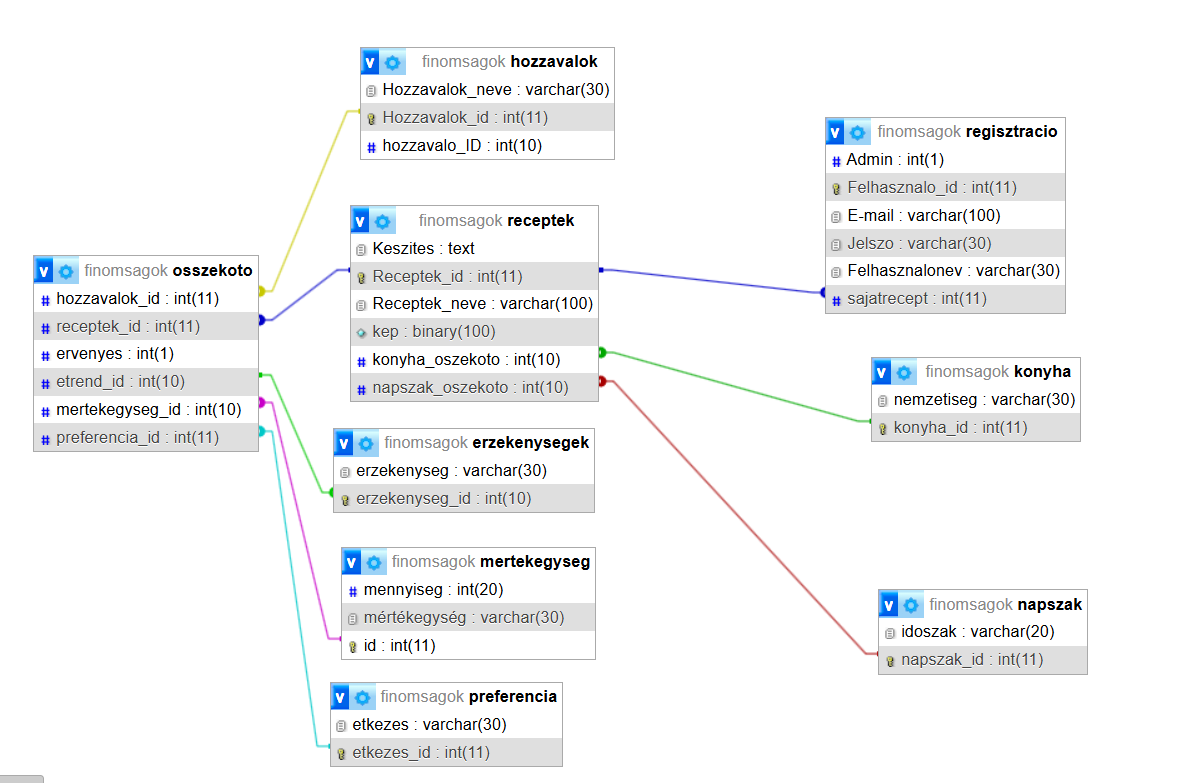
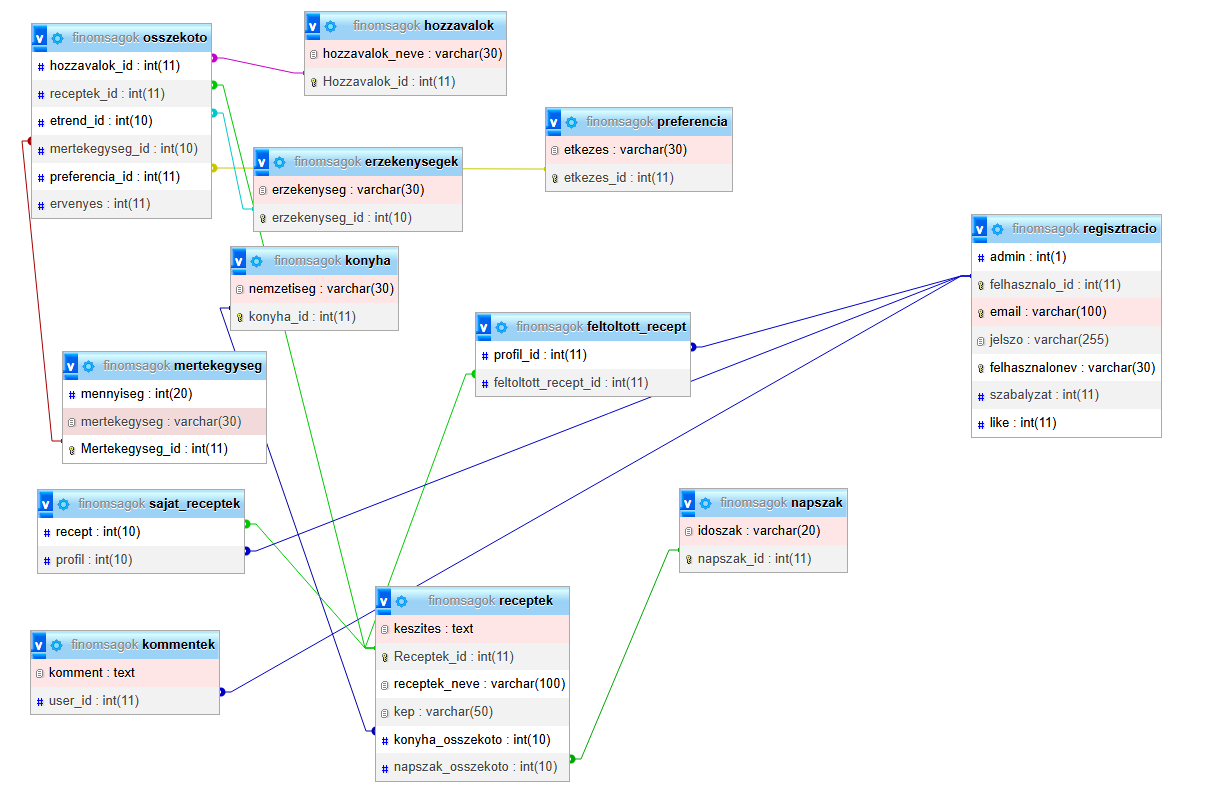
1. komment(text), NOT NULL)
2. user\_id(int(11), NOT NULL)

A feladatban XAMPP és Apache szerverrel futott MySql adatbáziskezelő rendszert használtunk adatbázisunk elkészítésére. Emellett alkalmaztunk phpMyAdmin nyílt forráskódú adatbázis-kezelési eszközt a MySql számára. Ezen a webes felületen kezeltük az adatbázisunkat.

* Ez az ER (egyed-kapcsolat modell) diagram a “smartdraw” diagramkészítőben készült. Egyedtípusokból áll, és meghatározza az egyedek között létező kapcsolatokat. Láthatóak rajta a táblák, a táblák rekordjai és az elsődleges kulcsok. Található rajta egy összekötő tábla, aminek célja, hogy ne sérüljön a táblák közti integritás. Rengeteg tábla a normálformák megfelelésének érdekében jött létre (hogy megfeleljenek az 1/2/3-as normálformának).



* OOP megvalósításhoz létrehoztunk egy UML osztálydiagramot, amit “smartdraw” diagramkészítőben készítettünk. Az adatbázis kialakításának és funkcionalitásának meghatározására szolgál. Osztályokat, objektumokat és a köztük lévő kapcsolati képeket ábrázolja.



* Itt található az adatbázisunk (phpMyAdmin felületi kialakítással). Az adatokat elektronikus tárolja és kezeli. Az adatbázis tartalmaz 11 táblát, található benne 1: N, N:N kapcsolat.

## Részletes feladatspecifikáció, algoritmusok

1. **AboutSection komponens**
2. **Fő funkciók:**

* **toggleAboutText**:
  + **Leírás**: A "Rólunk" szöveg megjelenítését váltja. Ha az információk megjelenítése van, akkor elrejti, ha pedig rejtett, akkor megjeleníti.
  + **Paraméterek**: Nincs.
  + **Visszatérési érték**: Nincs, de a showAbout állapotot módosítja.
* **handleLike**:
  + **Leírás**: A "Like" gombot kezeli, növeli a lájkok számát.
  + **Paraméterek**: Nincs.
  + **Visszatérési érték**: Nincs, de a likes állapotot növeli.
* **handleFeedbackChange**:
  + **Leírás**: A felhasználó által írt visszajelzés szövegét frissíti.
  + **Paraméterek**: e - az esemény, amely a textarea mező változását jelzi.
  + **Visszatérési érték**: Nincs, de a feedback állapotot frissíti.
* **handleFeedbackSubmit**:
  + **Leírás**: A visszajelzést elküldi. Ha van beírt szöveg, új visszajelzés objektumot hoz létre, és hozzáadja a submittedFeedback tömbhöz.
  + **Paraméterek**: e - az esemény, amely a form küldését jelzi.
  + **Visszatérési érték**: Nincs, de a submittedFeedback állapotot frissíti.

1. **2. Admin komponens**
2. **Fő funkciók:**

* **deleteRecipe**:
  + **Leírás**: Eltávolít egy receptet az adatbázisból.
  + **Paraméterek**: recipeId - a recept azonosítója.
  + **Visszatérési érték**: Nincs, de az összes állapotot frissíti, eltávolítva a receptet.
* **deleteUser**:
  + **Leírás**: Eltávolít egy felhasználót és annak receptjeit az adatbázisból.
  + **Paraméterek**: userId, recipeId - a felhasználó és recept azonosítója.
  + **Visszatérési érték**: Nincs, de a users állapotot frissíti, eltávolítva a felhasználót.
* **toggleAdminStatus**:
  + **Leírás**: Az admin státuszt váltja a felhasználónál.
  + **Paraméterek**: userId, currentStatus - a felhasználó azonosítója és az aktuális admin státusz.
  + **Visszatérési érték**: Nincs, de a users állapotot frissíti a felhasználó admin státuszával.
* **toggleRecipeStatus**:
  + **Leírás**: Váltja a recept státuszát (aktív vagy inaktív).
  + **Paraméterek**: recipeId, currentStatus - a recept azonosítója és az aktuális státusz.
  + **Visszatérési érték**: Nincs, de az osszes állapotot frissíti a recept státuszával.

1. **3. BackgroundBubbles komponens**
2. **Fő funkciók:**

* **generateRandomPosition**:
  + **Leírás**: Véletlenszerű pozíciókat generál a buborékok számára.
  + **Paraméterek**: Nincs.
  + **Visszatérési érték**: A véletlenszerű pozíciót tartalmazó objektum (top, left).

1. **Home komponens**
2. **Leírás:**

A Home komponens a főoldali receptlistát és a felhasználói interakciókat kezeli. Az alábbi funkciókat tartalmazza:

* **Recept keresés és listázás**
* **Recept részletes információjának megjelenítése**
* **Recept mentése a felhasználó profiljához**

1. **Főbb metódusok:**

* **useEffect()**
  + **Leírás**: Beállítja a felhasználói adatokat (ha be van jelentkezve), betölti a mentett recepteket és a recept adatokat egy API hívás segítségével.
  + **Paraméterek**: kereses (a keresett kifejezés)
  + **Visszatérés**: Nincs (side-effectet végez).
* **loadSavedRecipes()**
  + **Leírás**: Betölti a felhasználó által mentett recepteket az API-ból.
  + **Paraméterek**: userId (a felhasználó ID-ja)
  + **Visszatérés**: Nincs (a mentett receptek állapotát frissíti).
* **saveRecipe()**
  + **Leírás**: Ment egy receptet a felhasználóhoz, ha be van jelentkezve.
  + **Paraméterek**: recipeId (a recept ID-ja)
  + **Visszatérés**: Nincs (alert és állapot frissítés történik).
* **filteredRecipes** (computed property)
  + **Leírás**: A receptlistát szűri a keresett kifejezés alapján.
  + **Paraméterek**: Nincs.
  + **Visszatérés**: A szűrt receptlista.

1. **Algoritmusok:**

* **Szűrés (Keresés)**:

plaintext

MásolásSzerkesztés

Input: recept lista, keresési kifejezés

Output: szűrt recept lista

1. Iterálj végig a recept listán.

2. Minden recept neve tartalmazza a keresett kifejezést?

3. Ha igen, tedd hozzá az eredményekhez.

4. Visszaadjuk a szűrt listát.

1. **2. LandingPage komponens**
2. **Leírás:**

A LandingPage komponens egy animált bevezető oldalt jelenít meg, amely időzítve váltogatja a szövegeket, és egy gombbal navigál a főoldalra.

1. **Főbb metódusok:**

* **useEffect()**
  + **Leírás**: A bevezető oldal betöltésekor váltogatja a megjelenített szövegeket és kezeli a betöltési időzítőt.
  + **Paraméterek**: Nincs.
  + **Visszatérés**: Nincs (side-effectet végez).
* **setInterval()** és **setTimeout()** (időzítő metódusok)
  + **Leírás**: A setInterval váltogatja a szövegeket, a setTimeout pedig a betöltési animáció végét jelzi.
  + **Paraméterek**:
    - setInterval: A szövegek közötti váltás ideje (2000 ms).
    - setTimeout: A betöltés ideje (4500 ms).
  + **Visszatérés**: Nincs.

1. **Algoritmusok:**

* **Szövegváltás**:

plaintext

MásolásSzerkesztés

Input: szövegek listája

Output: aktuális szöveg

1.Elkezdi az időzítőt, amely minden 2 másodpercben változtatja a szöveget.

2. Miután 4.5 másodperc eltelt, átirányítja a felhasználót a főoldalra.

3. A szövegek indexét ciklikusan frissíti.

 **Bejelentkezés és regisztráció:**

* A felhasználók bejelentkezhetnek vagy regisztrálhatnak a felhasználónevük, e-mail címük és jelszavuk megadásával.
* A regisztráció során a felhasználók számára lehetőség van a felhasználónév módosítására.
* A bejelentkezett felhasználók számára a rendszer figyeli, hogy elfogadták-e a szabályzatot. Ha nem, akkor a rendszer felugró ablakban kéri a szabályzat elfogadását.
* Ha a szabályzatot nem fogadják el, akkor nem folytathatják a használatot.

** Profil oldal:**

* A profiloldalon a felhasználók megtekinthetik és szerkeszthetik a profiljukat, például módosíthatják a felhasználónevüket.
* Az oldalon megjelennek a felhasználó által mentett és létrehozott receptek.
* A receptek alatt találhatóak olyan funkciók, mint a recept bővebb információinak megtekintése vagy egy recept eltávolítása a mentett listából.

 **Receptek kezelése:**

* A felhasználók hozzáadhatják, szerkeszthetik és törölhetik a receptjeiket.
* A receptekhez tartozó információk, mint például hozzávalók és étkezési típusok, az egyes receptek alatt láthatóak.

 **Frissítés és animációk:**

* Az alkalmazás a Framer Motion könyvtárat használja, hogy animációkat és átmeneteket biztosítson a felhasználói élmény javítása érdekében.

1. **Recept komponens**

* **Célja:** A Recept komponens lehetővé teszi a felhasználó számára, hogy új receptet töltsön fel a rendszerbe. A felhasználó megadhatja a recept nevét, leírását, hozzávalókat, azok mennyiségét és mértékegységét, a recepthez tartozó képet, valamint kategorizálhatja azt különböző szűrők alapján (nemzetiség, napszak, érzékenység, preferencia).

1. **Metódusok:**

* **useState használata:**
  + recipeName, description, preferences, sensitivity, image, ingredients: A felhasználói adatokat és a recepthez tartozó információkat tároló állapotok.
  + ingredientOptions, nationalityOptions, dayTimeOptions, sensitivityOptions, preferenceOptions: Az API-ból származó beállítási lehetőségeket tartalmazó állapotok.
  + selectedNationality, selectedDayTime: A felhasználó által kiválasztott nemzetiség és napszak.
  + user: A bejelentkezett felhasználó adatai.
* **Függvények:**
  + **useEffect()**: Betölti az adatokat az API-ból és beállítja az opciókat (hozzávalók, nemzetiségek, napszakok, érzékenység, preferenciák).
  + **handleIngredientChange(index, selectedOption)**: Frissíti az hozzávalókat, amikor egy új hozzávalót választ a felhasználó.
  + **handleInputChange(index, field, value)**: A hozzávalók mezőinek módosítása.
  + **addIngredient()**: Új hozzávaló mező hozzáadása.
  + **removeIngredient(index)**: Hozzávaló mező eltávolítása.
  + **validateData()**: Ellenőrzi, hogy minden kötelező mező kitöltésre került-e.
  + **handleSubmit(e)**: Az űrlap elküldése, adatküldés az API-hoz, validálás után.

1. **NavBar komponens**

* **Célja:** A NavBar komponens biztosítja a navigációt az alkalmazásban, és tartalmazza a keresőmezőt, illetve a bejelentkezési, adminisztrátori és profil linkeket.

1. **Metódusok:**

* **handleSearch(e)**: A keresési szót állítja be a felhasználó által beírt érték alapján, és továbbítja a setKereses függvénynek.
* **useState használata:**
  + menuOpen: A hamburger menü állapotának nyitott/zárt állapotát tárolja.
  + searchValue: A keresési kifejezés tárolása.

1. **2. Algoritmusok és logikai folyamatok**
2. **Recept hozzáadása:**
3. **Adatok validálása:** Ellenőrzi, hogy a kötelező mezők ki vannak-e töltve (pl. recept neve, hozzávalók, kép stb.).
4. **Adatküldés:** Ha minden validált adat rendben van, a recept adatait elküldi az API-ra egy POST kérés formájában, formData segítségével.
5. **Hozzávalók kezelése:** A hozzávalók tömböt dinamikusan lehet bővíteni (hozzáadni új mezőt) vagy módosítani (mennyiség, mértékegység). Az új hozzávalók feldolgozása a handleIngredientChange és handleInputChange metódusokkal történik.
6. **Keresés:**
7. A felhasználó által beírt keresési szót a NavBar komponensben egy input mező figyeli.
8. A keresett érték a setKereses függvény segítségével átadásra kerül a szülő komponensnek, ahol tovább szűrheti az adatokat.
9. **App komponens:**

* **Függvény leírása**: Az App komponens az alkalmazás fő komponenst képviseli, amely a navigációs sávot, a felhasználó bejelentkezését és a különböző oldalak közötti navigálást kezeli.
* **Állapotok**:
  + isLoggedIn: A felhasználó bejelentkezési állapota.
  + kereses: A keresési kifejezés, amelyet az alkalmazásban használnak.
* **Metódusok**:
  + useEffect: Két useEffect hook-ot használunk:
    1. Az első arra szolgál, hogy a helyi tárolóból beolvassa a felhasználó adatait és beállítja a isLoggedIn állapotot.
    2. A második arra figyel, hogy a navigációs sáv látszódjon vagy ne látszódjon a főoldalon kívüli oldalak esetében.
  + handleLogout: A felhasználó kijelentkezését kezeli, eltávolítja a helyi tárolóból a felhasználó adatokat.
* **Jellemzők**:
  + **Navigációs viselkedés**: Az App komponens a Routes komponens segítségével rendezi el az alkalmazás különböző oldalait. Ha a felhasználó nincs bejelentkezve, a profil, recept és admin oldalakat a Login oldalra irányítja.
  + **Változók**:
    1. kereses: A keresési adatokat átadja a Home oldalnak.

1. **Routes és Route komponensek:**

* **Függvény leírása**: A Routes komponens tartalmazza az összes alkalmazásbeli útvonalat, míg a Route komponensek a különböző oldalak renderelését kezelik. Az alkalmazás a következő oldalakat tartalmazza:
  + /: A főoldal (LandingPage).
  + /home: A kezdő oldal (Home).
  + /login: A bejelentkezési oldal (Login).
  + /profile: A felhasználó profiloldala (Profile).
  + /recept: A receptoldal (Recept).
  + /admin: Az adminisztrációs felület (Admin).

1. **NavBar komponens:**

* **Függvény leírása**: A navigációs sáv, amely megjeleníti az oldalakat és lehetőséget ad a kijelentkezésre.
* **Metódusok**:
  + setKereses: A keresési kifejezés frissítésére szolgál.

1. **Login, Profile, Recept, és Admin komponensek:**

* **Függvények leírása**:
  + A Login oldal kezeli a felhasználó bejelentkezését.
  + A Profile, Recept, és Admin oldalak az isLoggedIn állapotot ellenőrizve biztosítják, hogy a felhasználó csak akkor férhessen hozzá a tartalomhoz, ha be van jelentkezve.

1. **2. Algoritmizálható részek:**

Az alkalmazás kulcsfontosságú algoritmikus része az, hogy a felhasználó bejelentkezési állapota alapján az oldalakat dinamikusan rendereljük, és hogy a felhasználó kijelentkezésekor eltávolítjuk a tárolóból a felhasználói adatokat. Ez az alábbi logikával valósul meg:

1. **Bejelentkezés ellenőrzés**:
   * Ha a felhasználó adatai a helyi tárolóban vannak, állítsd be az isLoggedIn értéket igazra.
2. **Navigáció ellenőrzés**:
   * Ha a felhasználó nem jelentkezett be, irányítsd át a Login oldalra.
   * Ha be van jelentkezve, akkor jelenítsd meg a profil és recept oldalt.
3. **Kijelentkezés**:
   * A handleLogout függvény eltávolítja a felhasználói adatokat a helyi tárolóból, és beállítja az isLoggedIn értékét hamisra.

* **Bejelentkezés/Profil oldal**:
  + **Feltétel**: Van-e user a helyi tárolóban?
    - Igen: Állítsd be isLoggedIn-et igazra.
    - Nem: Térj vissza a Login oldalra.
* **Logout**:
  + **Feltétel**: Kattintott-e a kijelentkezés gombra?
    - Igen: Távolítsd el a felhasználót a helyi tárolóból, állítsd be isLoggedIn-t hamisra.

1. **3. Fontosabb kódrészletek magyarázata:**

* **Navigációs sáv kezelése**: A useEffect figyeli a location.pathname változást, és dinamikusan hozzáadja vagy eltávolítja a with-navbar osztályt a body elemhez, hogy csak azokon az oldalakon legyen a navigációs sáv, amelyek nem a főoldal (/).
* **Bejelentkezés és állapotkezelés**: A localStorage.getItem("user") hívás ellenőrzi, hogy a felhasználó be van-e jelentkezve, és ennek megfelelően állítja be az isLoggedIn állapotot.

Ez a megközelítés biztosítja a dinamikus és biztonságos navigációt az alkalmazásban, figyelembe véve a felhasználó bejelentkezési állapotát.

1. **Példa Pseudo-kódok:**
2. **Általános működés**

MásolásSzerkesztés

Amikor a komponens betöltődik:

- Lekérdezi az összes szükséges opciót az adatbázisból:

- hozzávalók, konyhák, napszakok, érzékenységek, preferenciák

- Beolvassa a felhasználót a localStorage-ből

1. **Űrlapmezők (state-ek):**

MásolásSzerkesztés

- recept neve

- leírás

- preferencia

- érzékenység

- kép

- hozzávalók (több lehet, darabonként: id, mennyiség, mértékegység)

- kiválasztott konyha

- kiválasztott napszak

1. **Űrlap műveletek**

MásolásSzerkesztés

Felhasználó módosít:

- beírja a recept nevét, leírását

- kiválaszt képet

- hozzávalókat ad hozzá, vagy töröl belőlük

- beállítja konyhát, napszakot, érzékenységet, preferenciát

Amikor a 'Recept feltöltése' gombra kattint:

- Ellenőrzi, hogy minden ki van-e töltve

- Ha nincs kép, vagy valami hiányzik: figyelmeztet

- Összeállít egy `FormData`-t a mezőkből

- POST kérést küld a szervernek (backend API-nak)

- Sikeres esetben: kiírja, hogy sikeres feltöltés

1. **Adatküldés**

MásolásSzerkesztés

FormData-t küld:

- recept neve, leírás

- nemzetiségId, napszakId

- preferencia, érzékenység

- kép

- hozzávalók tömbje

- userId (a localStorage alapján)

1. **Stílus és Select mezők**

MásolásSzerkesztés

A dropdown mezők egyedi sötét témát használnak (React Select)

- fekete háttér, fehér szöveg

- fókuszban kék keret

1. **Mentett receptek szűrése csak a bejelentkezett felhasználónak**
2. **1. Adatok lekérése localStorage-ból**

css

MásolásSzerkesztés

Ha van elmentett recept a localStorage-ben:

Olvasd be az összes receptet (mint tömb)

Egyébként:

Használj üres tömböt

1. **2. Szűrés a bejelentkezett felhasználó emailje alapján**

less

MásolásSzerkesztés

Ha van bejelentkezett felhasználó (currentUser) ÉS van email címe:

Végigmegyünk az összes recept listán:

Ha a recept.user megegyezik a currentUser.email-lel:

Meghagyjuk a receptet

Egyébként:

Eldobjuk

Az így kiszűrt recepteket elmentjük egy új állapotba (filteredRecipes)

Különben (nincs bejelentkezett felhasználó):

filteredRecipes legyen üres lista

1. **Frissítés, ha a felhasználó változik**

less

MásolásSzerkesztés

Amikor a currentUser megváltozik (pl. bejelentkezik valaki új):

Futtasd le újra a fenti szűrést

1. **Admin komponens pszeudokód:**

plaintext

MásolásSzerkesztés

Kezdéskor:

- Hozz létre változókat: keresés, összes recept, bővebb infó ID, táblázat megjelenítés, leírások, felhasználók megjelenítése, felhasználók listája

useEffect:

- Ha NEM felhasználókat mutat:

→ Kérd le az összes receptet az API-tól keresés alapján

→ Kérd le a hozzávalók leírásait

- Ha felhasználókat kell mutatni:

→ Kérd le a felhasználói statisztikákat

Recept törlése:

- Küldj DELETE kérést a recept azonosítójával

- Töröld ki a helyi listából is

Felhasználó törlése:

- Küldj DELETE kérést a felhasználó azonosítójával és egy recept ID-val

- Töröld ki a listából

Admin státusz váltása:

- Küldj POST kérést userId és új státusszal

- Frissítsd a helyi listában az adott user admin mezőjét

Recept érvényességének kapcsolása:

- Állítsd át a státuszt helyileg

- Küldd el az új státuszt a backendnek

- Ha hiba van, állítsd vissza az eredeti értékre

Megjelenítés:

- Ha nem felhasználók jelennek meg:

→ Mutasd a keresőmezőt

→ Receptkártyák (kép, név, szöveg, bővebb leírás gomb, hozzávalók táblázata, státusz kapcsoló, törlés)

- Ha felhasználók jelennek meg:

→ Mutasd a nevüket, emailt, receptszámot, admin státuszt

→ Gomb: törlés vagy admin jog kapcsolása

Alul: Gomb felhasználók és receptek váltására

1. **BackgroundBubbles komponens pszeudokód:**

plaintext

MásolásSzerkesztés

- Hozz létre ikonokat (pl. répa, hal, pizza stb.)

- 20 véletlenszerű buborékot generálj:

→ Véletlen pozíció (top, left)

→ Véletlen méret (30-70px)

→ Véletlen szín (HSL)

→ Framer Motion animáció: lebegés felfelé végtelenítve

→ Használj egy ikon minden buborékhoz

- Térj vissza a buborékokkal körberakott konténerrel

1. **AboutSection komponens:**

plaintext

MásolásSzerkesztés

- Állapotok:

→ Megjelenjen-e a rólunk szöveg

→ Like-ok száma

→ Vélemény szövege

→ Beküldött vélemények listája

→ Kiválasztott csillag értékelés

Műveletek:

- toggleAboutText:

→ Kapcsold be vagy ki a rólunk szöveg megjelenítését

- closePopup:

→ Rólunk ablak bezárása

- handleLike:

→ Like-ok számát növeld 1-gyel

- handleFeedbackChange:

→ Vélemény szöveg frissítése gépeléskor

- handleFeedbackSubmit:

→ Ne engedd az üres beküldést

→ Hozz létre egy objektumot (szöveg, értékelés, dátum)

→ Add hozzá a visszajelzésekhez

→ Ürítsd ki az inputot

Megjelenítés:

- Elérhetőségek felsorolása

- Reklám szöveg megjelenítése

## Tesztelési dokumentáció

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tesztesetek** | **Sor típus**  **TesztEset** | **Teszt lépés leírása** | **Elvárt eredmény** | **Hibás Eredmény** | **Megjegyzés** |
| React alapú regisztráció és bejelentkezés egy adott porton futó szerveren | >TesztEset | Regisztrálás és bejelentkezés (ezek során az adatok egyeznek-e, belekerültek-e az adatbázisba, a jelszó titkosítása megtörtént-e) | A felhasználónak egy sikeres regisztráció és egy sikeres bejelentkezés után megjelentek a főbb menüpontok a menüben | Hibás eredmény – A regisztráció sikeresnek bizonyult viszont a bejelentkezés során folyton azt írta ki, hogy **„hibás jelszó”** (ez több problémát is felvetett, rossz a tikosított bemeneti karakterlánc, a titkosítás nem történt meg helyesen és ezáltal a jelszó nem az, amit megadtunk neki…stb.) | Javítás - A titkosítás megtörtént, csak a karakterlánc hossza nem volt megfelelő (nekünk varchar(30) volt, de (255) -re volt szükség, hogy a titkosítás teljes mértékben működjön, így gyakorlatilag nem tudta eltárolni a teljes általunk megadott jelszót, csak egy részét)) |
| Bejelentkezés és regisztrációs blokk testre szabása | >TesztEset | Bejelentkezés és regisztrációs blokknak a kinézete | Előszőr a bejelentkezés/regisztráció menüpontra kattintva dobja fel a regisztrációt, egy sikeres regisztráció után, egy animáció segítségével jelenjen meg a bejelentkezés blokk | Hibás eredmény – A regisztráció menü megjelent (de nem az elvárt kinézettel), viszont, ha regisztráltál, nem működött jól az animáció (el kellett volna tűnnie regisztrációnak és megkellet volna jelennie a bejelentkezésnek, ehelyett megjelent a regisztráció és miután rákattintottál a bejelentkezésre, ott maradt a regisztráció és rácsúszva megjelent a bejelentkezés is) | Javítás - Rosszul voltak megadva a méretezések a weboldalon, emiatt lement az animáció, de nem tudta gyakorlatilag **„végig csinálni”** mert a méretek szerint dolgozott |
| Nyitó oldal elkészítése | >TesztEset | Alapkoncepciónk az volt, hogy a főoldal lesz az első rész, amit a felhasználó látni fog, végül csináltunk egy nyitó oldalt, ami a weboldal első elindításánál fog megjelenni és az irányítja át a felhasználót automatikusan a főoldalra | A nyitó oldal összes animációja, konfigurációja, betöltése és szövegei kiíratása megtörténjen, aztán automatikusan irányítson át a főoldalra | Hibás eredmény - Rosszul volt navigálva. Mindig akkor hozta be, amikor rámentünk a főoldalra és nekünk az kellet, hogy csak akkor jelenjen meg, mikor legelőször betölt a felhasználónak a weboldal | Javítás – A nyitó oldal lett alapra betéve és onnan egy **„kereses”** függvény segítségével irányítja át a felhasználót a főoldalra   <Route path="/" element={<LandingPage />} />          <Route path="/home" element={<Home kereses={kereses} />} /> |
| Weboldalunk tesztelése egy osztálytársunk által | >TesztEset | Fekete dobozos tesztelés (Nem ismerte az algoritmusokat, a kódsorokat, a funkciókat…semmit) | A felhasználó akárhova kattint, minden funkció működjön, minden oldal jelenjen meg megfelelően, ne omoljon össze a weboldal és annyiszor használhasson egy funkciót amennyiszer szeretné (nem minden funkciót érdemes többször használni, bizonyos funkciók limitálva vannak) | Hibás eredmény – Osztálytársunk fényt derített arra, ha megpróbált bejelentkezni vagy regisztrálni összeomlott a weboldal. A regisztráció és a bejelentkezés kezelése fel volt cserélve, ezért akárhányszor megpróbált beregisztrálni a rendszer szerint bejelentkezett (megcsinálta, csak nem tudta mit kezdjen az információkkal és hol tárolja őket) és amikor megpróbált bejelentkezni, hivatalosan akkor regisztrált | Javítás – Megcseréltük a sorrendjüket |
| Weboldalunk tesztelése egy tanárunk által | >TesztEset | Fehér dobozos tesztelés (már jó párszor rákérdezett és belenézett a kódsorainkba és a különböző összecsatolt elemekbe) | A felhasználó akárhova kattint, minden funkció működjön, minden oldal jelenjen meg megfelelően, ne omoljon össze a weboldal és annyiszor használhasson egy funkciót amennyiszer szeretné (nem minden funkciót érdemes többször használni, bizonyos funkciók limitálva vannak) | (Hibát konkrétan nem talált, véleményt mondott, hogy mivel tehetnénk szebbé, kreatívabbá és még érdekesebbé a weboldalt. Mondott pár alap funkciót, mint például a nyitó oldal, a színek változtatása…stb.) | - |

# Összefoglalás

## Önértékelés

* A kitűzött céljaink elérése:

1. Egy olyan főzős weboldal, aminek a megtekintése során ne lehessen mindent látni, csak, ha regisztrál és bejelentkezik a felhasználó.
2. A regisztráció és bejelentkezés után jelenjenek meg a főbb menük és egy profil („Admin” (de, csak adminként van bejelentve), „Receptek” (receptfeltöltés), „Profil” (saját profil és adatai), „Logout” (kijelentkezés))
3. Mindegyik menü funkcionalitásai az eltervezett, elvárt módon működjenek
4. A dokumentáció és a prezentáció tökéletesen tükrözze és leírja a weboldalunk által kitűzött célokat, a weboldal funkcióit és minden fontos részletét

* A feladat során miben fejlődtem (mit tanultunk meg, hogyan alkalmaztuk, a felmerült problémák és megoldásaik):
* Varju Richárd Dániel:
* Az én véleményem és tapasztalataim szerint az embernek egy ilyen jelentőségű feladat elvégzése egyszerre egy jó cél és stressz. Rendkívül sok mindenben fejlődhetsz és megannyi érdekességet tudhatsz meg a feladatod elvégzése során, viszont alaposan át kell gondolni minden lépést, mielőtt belekezdenél a feladataidba, mert lehet, hogy nem előre látott problémákkal találkozol, amik megnehezítik a további folyamatokat.
* Nekem a legnagyobb problémáim a bejelentkezéssel és a regisztrációval voltak. A problémák közé tartozott például, hogy nem működött a regisztráció, ha működött, akkor nem kerültek fel az adatbázisba az adatok, ha felkerültek, nem azok az adatok kerültek fel, amiket megadtam…stb.
* A csapatmunka volt a legjobb az egészben szerintem. A csapattársammal még ha nem is mindenben értettünk egyet, akkor is megtudtunk mindent beszélni egymással (meghallgattuk egymást és közös döntésre jutottunk a munka folytatásával kapcsolatban). A cél az mindig közös volt és mindkettőnknek ugyanolyan prioritást jelentett.
* A problémákat mindig közösen oldottuk meg. A céljaink mindig bővültek, időről időre, ahogy kész lettünk egy-egy újabb kitűzött céllal. Kikértük mások véleményét a munkánkról és tanácsaikat, ötleteiket megpróbáltuk a nekünk lehető legérthetőbb módon megvalósítani.
* Horváth Ákos:
* Nekem a személyes véleményem és tapasztalatom a mestermunkánkkal kapcsolatban az, hogy szerintem sokat tud fejlődni az ember, ha belefog egy ilyen munkába és hogy sokkal többet tud tanulni mintha tanítanának (alkalmazkodtunk a szituációkhoz és ha valami problémába ütköztünk, problémáink megoldásának keresése közben minden kérdésünkre választ kaptunk), igaz azért egy néhányszor szükségünk volt segítségre, de összeségében egy nagyon jó tanuló lecke volt mindkettőnk számára.
* Nekem személyesen legtöbbször a dizájnnal gyűlt meg a bajom, mivel az összes oldal szerkezete bomlott egy oldal miatt. Ha egy oldalnak a dizájnját módosítottam és egy másik oldalnak pont ugyan az volt az azonosítója véletlenül, akkor nem működtek a stílusok, ezek voltak a legnagyobb nehézségeim, de néha az oldal funkcióival is voltak problémák, de nem olyan gyakran, mint a stílusokkal.
* Szerintem a csapatmunka eszméletlen jól alakult, sok mindent megbeszéltünk és legtöbb hibára vagy esetleges kérdésekre együtt kerestünk megoldást. Mindenről kommunikáltunk személyesen vagy olyan programokon keresztül, amin lehetőségünk volt írni vagy megbeszélni a dolgokat (például: Discord vagy Messenger).
* A céljaink nagy részét szerintem elértük, és talán még egy kicsivel többet is „mert, ha már van időnk akkor használjuk ki rendesen és tegyük jóbbá az oldalt" címszóval. A legnagyobb fejlődési előre lepésem a React keretrendszerben történt a jsx működésével kapcsolatban. A keretrendszer lehetőségeinek fel-és kihasználásban és szerintem ezeket a jövőben képes leszek felhasználni.

## Továbbfejlesztési lehetőségek

#### **3.9. Szűrők**

1. Ha speciális szempontok szerint keres, használja a **szűrő** opciót. A szűrők segítségével például diétás vagy allergénmentes recepteket találhat.
2. A szűrők között szerepelnek:
   1. **Étel típus** (pl. vegetáriánus, gluténmentes).
   2. **Elkészítési idő**.
   3. **Hozzávalók** (pl. alapanyagok szerint).
      1. A szűrő kiválasztása után a találatok azonnal frissülnek, és csak azok a receptek jelennek meg, amelyek megfelelnek az Ön által megadott kritériumoknak.

#### **4.2. Közösségi Megosztás**

1. A weboldal lehetőséget biztosít arra, hogy kedvenc receptjeit megossza másokkal a közösségi médián. Használja a **"Megosztás"** gombokat, hogy az adott receptet elküldhesse a Facebookon, Instagramon vagy Pinteresten.

### 5. Kedvezmények és Akciók

#### **5.1. Webshop és Akciók**

* A weboldalon található webshopban főzéshez szükséges eszközöket és alapanyagokat vásárolhat. Az oldal rendszeresen kínál kedvezményeket, amelyek a **"Kedvezmények"** menüpont alatt elérhetők. Itt különböző akciókat találhat, amelyek a felhasználók számára előnyösek.

### 6. Gyakran Ismételt Kérdések (GYIK)

#### **6.1. Hogyan tudok egy receptet módosítani?**

* A weboldalon nem lehet közvetlenül módosítani egy receptet, de ha szeretne valamit hozzáadni, vagy megosztani a saját variációját, ezt a **"Kommentek"** szekcióban teheti meg.

#### **6.2. Miért nem találok egy receptet?**

* Ha egy receptet nem talál a keresőben, próbáljon meg más kulcsszavakkal keresni, vagy ellenőrizze a szűrők beállításait.

#### **6.3. Hogyan használhatom a vásárlási listát?**

* A vásárlási lista a **"Kedvenc"** receptek oldalán található. Itt bármely recepthez hozzáadhatja az összes hozzávalót a vásárlási listához, amelyet később kinyomtathat, vagy mobiltelefonján elérhet.

### 7. Segítségkérés és Támogatás

* Ha bármilyen kérdése van, vagy segítségre van szüksége, lépjen kapcsolatba ügyfélszolgálatunkkal a **"Kapcsolat"** menüpont alatt található elérhetőségek egyikén.

# Felhasznált irodalom

[phpmyadmin | Nyílt forráskódú adatbáziskezelő eszköz](https://products.containerize.com/hu/database-management/phpmyadmin/) - utolsó megtekintés (2025.02.11., kedd – 9:35)

[phpMyAdmin – Wikipédia](https://hu.wikipedia.org/wiki/PhpMyAdmin) - utolsó megtekintés (2025.02.11., kedd – 9:20)

[ER-modell – Wikipédia](https://hu.wikipedia.org/wiki/ER-modell) - utolsó megtekintés (2025.02.11., kedd – 8:14)

[Mi a React? - Hogyan működik - Hogyan kell használni Használat és előnyei | Szoftverfejlesztés 2025](https://hu.education-wiki.com/9050114-what-is-react) - utolsó megtekintés - (2025.02.11., kedd – 8:35)

[HTML – Wikipédia](https://hu.wikipedia.org/wiki/HTML) - utolsó megtekintés - (2025.02.11., kedd – 8:12)

[Word I. Bevezető. Alapfogalmak - PDF Ingyenes letöltés](https://docplayer.hu/6435625-Word-i-bevezeto-alapfogalmak.html) - utolsó megtekintés (2025.02.11., kedd – 9:05)

[JavaScript – Wikipédia](https://hu.wikipedia.org/wiki/JavaScript) - utolsó megtekintés (2025.02.11., kedd – 8:10)

[Mi az a phpMyAdmin és mire használható?](https://kiszervezettmarketing.hu/weboldal-keszites/phpmyadmin-hasznalata/#:~:text=A%20phpMyAdmin%20egy%20n%C3%A9pszer%C5%B1%2C%20ingyenes%20%C3%A9s%20ny%C3%ADlt%20forr%C3%A1sk%C3%B3d%C3%BA,webes%20kezel%C3%A9s%C3%A9re%20fejlesztettek%20ki.%20PHP%20programoz%C3%A1si%20nyelven%20%C3%ADrt%C3%A1k.) - utolsó megtekintés (2025.02.11., kedd – 8:50)

[JSX bemutatása – React](https://hu.legacy.reactjs.org/docs/introducing-jsx.html) - utolsó megtekintés (2025.02.11., kedd – 9:45)

[GitHub – Wikipédia](https://hu.wikipedia.org/wiki/GitHub) - utolsó megtekintés (2025.02.11., kedd – 10:37)

<https://www.youtube.com/watch?v=Q3ixb1w-QaY> - utolsó megtekintés (2025.02.11., kedd – 10:36)

[google fordító - Keresés](https://www.bing.com/search?q=google+ford%C3%ADt%C3%B3&gs_lcrp=EgRlZGdlKgkIABBFGDsY-QcyCQgAEEUYOxj5BzIGCAEQLhhAMgYIAhBFGDsyBggDEEUYOTIGCAQQLhhAMgYIBRAAGEAyBggGEC4YQDIGCAcQABhAMgYICBAAGEAyCAgJEOkHGPxV0gEIMTIxNGowajGoAgCwAgE&FORM=ANAB01&PC=U531) - utolsó megtekintés (2025.02.11., kedd – 10:25)

[Első magyar receptgyűjtemény és gasztro portál | Mindmegette.hu](https://www.mindmegette.hu/) - utolsó megtekintés (2025.02.18., kedd – 8:40)

[Sertéspörkölt Nokedlivel | Pörköltrecept.hu](https://porkoltrecept.hu/sertesporkolt-nokedlivel/) - utolsó megtekintés (2025.02.18., kedd – 8:05)

[Enchiladas: hagyományos fogás egyenesen Mexikóból - Recept | Femina](https://femina.hu/recept/enchiladas_recept/) - utolsó megtekintés (2025.02.18., kedd – 8:20)

[Gyors és egyszerű sajt Tamales Recept](https://hu.hiloved.com/gyors-es-egyszeru-sajt-tamales-recept/) - utolsó megtekintés (2025.02.18., kedd – 8:15)

[Pane e Nutella, con questa ricetta avrai una colazione speciale e sfiziosa](https://www.mammastyle.it/2021/03/21/pane-e-nutella-con-questa-ricetta-avrai-una-colazione-speciale-e-sfiziosa/) - utolsó megtekintés (2025.02.18., kedd – 8:11)

[Lasagne recept | Mindmegette.hu](https://www.mindmegette.hu/recept/lasagne) - utolsó megtekintés (2025.02.18., kedd – 8:46)

[Sajtszószos gnocchi recept | Mindmegette.hu](https://www.mindmegette.hu/recept/gnocchi-sajtszosszal) - utolsó megtekintés (2025.02.18., kedd – 8:53)

[A hét Ájurvéda receptje: Upma, egy Ájurvédikus reggeli recept a hideg évszakokra… | Bhakti Kutir](https://bhaktikutir.hu/a-het-ajurveda-receptje-upma-egy-ajurvedikus-reggeli-recept-a-hideg-evszakokra/) - utolsó megtekintés (2025.02.18., kedd – 8:57)

[Aloo gobi | Vidék Íze](https://videkize.hu/recept/aloo-gobi/) - utolsó megtekintés (2025.02.18., kedd – 9:02)

[Indiai lencsefőzelék sütőben készítve (dal makhani) | Chili és Vanília](https://chiliesvanilia.hu/2016/12/29/indiai-lencsefozelek-sutoben-keszitve-dal-makhani/) - utolsó megtekintés (2025.02.18., kedd – 9:06)

[Klasszikus rántotta | Mindmegette.hu](https://www.mindmegette.hu/recept/rantotta) - utolsó megtekintés (2025.02.18., kedd – 9:10)

[Klasszikus töltött káposzta | Mindmegette.hu](https://www.mindmegette.hu/recept/klasszikus-toltott-kaposzta) - utolsó megtekintés (2025.02.18., kedd – 9:12)

[Gofri - amerikai palacsinta - Trendi Receptek](https://trendireceptek.hu/gofri-amerikai-palacsinta/) - utolsó megtekintés (2025.02.18., kedd – 9:21)

[Klasszik amerikai hamburger recept | Street Kitchen](https://streetkitchen.hu/szendvicsek-es-burgerek/klasszik-amerikai-hamburger/) . Utolsó megtekintés (2025.02.18., kedd – 9:24)

[A régi jó BBQ-oldalas bevált receptje: szinte szétolvad a szádban - Recept | Femina](https://femina.hu/recept/bbq-oldalas-recept/) - utolsó megtekintés (2025.02.18., kedd – 9:30)

[A legfinomabb fajitas | Mindmegette.hu](https://www.mindmegette.hu/recept/legfinomabb-fajitas) - utolsó megtekintés (2025.02.25., kedd – 8:29)

[Torta di mele, avagy az olaszok almás pitéje | Marie Claire- utolsó](https://marieclaire.hu/gasztro/2023/08/17/almas-pite-recept-torta-di-mele/) megtekintés (2025.02.25., kedd – 9:16)

[Zöldséges frittata receptje | Mindmegette.hu](https://www.mindmegette.hu/recept/zoldseges-frittata) - utolsó megtekintés (2025.02.26., szerda – 8:51)

<https://www.youtube.com/watch?v=jwEbw0zJqiY>

[](https://www.youtube.com/watch?v=jwEbw0zJqiY)

- utolsó megtekintés (2025.02.27., csütörtök - 11:28)

<https://chatgpt.com/c/67c58903-10f4-8004-8e26-7a31bc2898e3> - utolsó megtekintés (2025.03.03., hétfő - 12:03)

[PowerPoint Slide Zoom Tutorial 🔥Free Template🔥](https://www.youtube.com/watch?v=eog6CwB5wUs&ab_channel=OneSkillPowerPoint) – utolsó megtekintés (2025.03.23., szerda – 10:33)

[Create DYNAMIC Animated PowerPoint Slides Like a PRO! 🔥](https://www.youtube.com/watch?v=irL_9tkGgP8&t=527s) – utolsó megtekintés (2025.03.30., szerda – 12:55)

# Ábrajegyzék

[1. kép Ez a módosító ablak a programban 5](#_Toc63250394)