****

**ZÁRÓDOLGOZAT**

Készítették:

Kalina Tamás – Homovics Márton

Konzulens:

Horváth István

Miskolc

2025.

Miskolci SZC Kandó Kálmán Informatikai Technikum

Miskolci Szakképzési Centrum

SZOFTVERFEJLESZTŐ- ÉS TESZTELŐ SZAK

Edzőterem weboldal

online vásárló és motiváló oldal

Kalina Tamás – Homovics Márton

2024-2025

Tartalomjegyzék

[Záródolgozat tartalmi követelmények 1](#_Toc161991981)

[Forrásjegyzék 3](#_Toc161991982)

[Vizsgaremek benyújtása 4](#_Toc161991983)

[A vizsgaremek bemutatója 5](#_Toc161991984)

[Irodalomjegyzék 6](#_Toc161991985)

[Mellékletek I](#_Toc161991986)

[Index.js: II](#_Toc161991987)

[index.html: III](#_Toc161991988)

Bevezetés

Sokat agyaltunk mit szeretnénk pontosan de végül azt választottuk amiket szeretünk, az edzést, a ruhákat és az Opium stílust. Ezért úgy gondoltuk egybe vegyítjük a hármat hiszen ilyet még nem nagyon láttunk sehol a világon.

Nem mindennapi úgymond Opium stílust a rapper Playboi Carti és csapatárol inspiráltuk, az Opium az előbb említett rappernek a zenekiadó csapatának a neve, itt nem a drogra kell gondolni amikor kimondjuk a nevét, csak az interneten 2023ban így terjedt el a stílusuk neve.



Programozási nyelvek és technológiák

Sokat agyaltunk mit szeretnénk pontosan de végül azt választottuk amiket szeretünk, az edzést, a ruhákat és az Opium stílust. Ezért úgy gondoltuk egybe vegyítjük a hármat hiszen ilyet még nem nagyon láttunk sehol a világon.



MySQL adatbázis (Database)

A projekt kezdeti lépése az adatbázis megtervezése volt. De mit is jelent pontosan az adatbázis fogalma? Az adatbázis egy olyan rendezett információgyűjtemény, amelyet elektronikus formában, számítógépes rendszerekben tárolnak. Ezeket az adatokat egy ún. adatbázis-kezelő rendszer (DBMS) irányítja, amely összekapcsolja a felhasználót az adatbázissal, és lehetővé teszi annak kezelését.

A DBMS, az adatbázis, valamint a hozzá tartozó alkalmazások együtt alkotják az úgynevezett adatbázisrendszert – ezt gyakran egyszerűen csak „adatbázisnak” nevezzük.

A leggyakrabban alkalmazott adatbázis-típusokban az információk táblák formájában, sorokba és oszlopokba rendezve kerülnek tárolásra. Ez a struktúra lehetővé teszi, hogy az adatok könnyen hozzáférhetők, módosíthatók, frissíthetők és ellenőrizhetők legyenek, valamint támogatja a hatékony adatfeldolgozást és lekérdezést.

### Mi az SQL? (Strukturált lekérdező nyelv)

.

Az SQL (Structured Query Language) egy speciális programozási nyelv, amelyet szinte minden relációs adatbázis használ az adatok lekérdezésére, módosítására, kezelésére, valamint a jogosultságok szabályozására. Az SQL nyelvet az 1970-es években fejlesztette ki az IBM, az Oracle közreműködésével, amely később az ANSI szabványosításához vezetett. Azóta több vállalat – köztük az IBM, az Oracle és a Microsoft – továbbfejlesztette saját verzióit, így napjainkban számos változat létezik belőle. Bár az SQL továbbra is az egyik legszélesebb körben alkalmazott nyelv, egyre több új nyelv is megjelenik a piacon.

**Relációs adatbázis:** Az ilyen típusú adatbázisokban az információk táblázatos formában, sorokba és oszlopokba rendezve kerülnek tárolásra. Ez a megközelítés biztosítja a strukturált adatok gyors és rugalmas kezelését, lekérdezését. A relációs adatbázisok már az 1980-as évektől kezdve meghatározó szerepet játszanak a piacon, és napjainkban is a legelterjedtebb megoldásnak számítanak – noha léteznek alternatívák is, mint például az objektumorientált, elosztott vagy felhőalapú adatbázisok. A legtöbb fejlesztési környezet és nyelv azonban továbbra is elsősorban relációs adatbázisokkal dolgozik.

### phpMyAdmin és XAMPP

### C:\Users\Tanulo\Desktop\XAMPP.jpg

Projektünk során az adatbázis kezeléséhez a **phpMyAdmin** nevű szoftvert használtuk, amelyhez a **XAMPP** elnevezésű vezérlőpanel szolgált technikai alapként.

#### Mi az a XAMPP?

Bár elsőre csupán véletlenszerű betűkombinációnak tűnhet, a **XAMPP** valójában egy ingyenesen elérhető, nyílt forráskódú webszerver-csomag. Ez a szoftvercsomag lehetővé teszi a PHP-kódok futtatását, valamint az SQL-alapú adatbázisok kezelését. A phpMyAdmin felület eléréséhez elengedhetetlen, hogy először az **Apache** és a **MySQL** szolgáltatásokat elindítsuk a XAMPP vezérlőpanelen, majd az **"Admin"** gombra kattintva hozzáférjünk az adatbázis-kezelő felülethez.

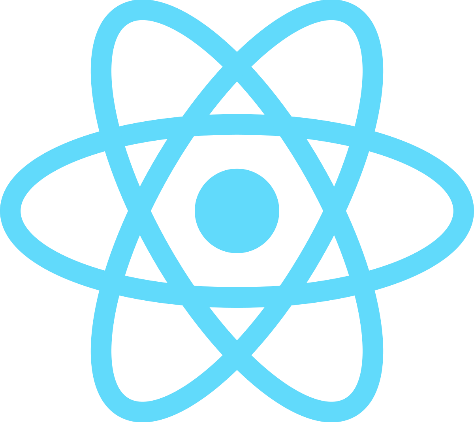
#### Mi az a phpMyAdmin?

A **phpMyAdmin** szintén egy nyílt forráskódú, ingyenesen használható webes alkalmazás, amely a legnépszerűbb eszközök közé tartozik SQL-alapú adatbázisok kezelésére. Böngészőből érhető el, így platformfüggetlen, és bármilyen operációs rendszeren használható, ahol van hozzáférés a szerverhez.

A phpMyAdmin felületén egyszerűen hozhatunk létre új adatbázisokat, táblákat és listákat, valamint lehetőség van fájlok importálására és exportálására különböző formátumokban. Az összetett SQL lekérdezések megírása mellett kattintásalapú módon is végezhetünk adatmanipulációt, ami gyors és felhasználóbarát működést tesz lehetővé.

A szoftver többféle operációs rendszerrel kompatibilis – így nemcsak Windows, hanem számos Linux disztribúció alatt is zavartalanul használható.

REACT-FRONTEND



### Hogyan működik a React?

A **ReactJS** egy olyan JavaScript-alapú könyvtár, amely a komponensalapú fejlesztés elvét követi. Ez azt jelenti, hogy az alkalmazások kisebb, újrahasznosítható részekből (komponensekből) épülnek fel, és ezek egyirányú adatfolyam mentén kommunikálnak egymással. A React egyik alaptézise az állapotkezelés és az adatok megváltoztathatatlansága (immutabilitás). A komponensek szülő-gyermek viszonyban állnak, és szigorú hierarchiába rendeződnek. Minden komponens egy önálló, jól elkülönített kódrészlet, amelyet a tiszta kód (Clean Code) elveinek megfelelően hozunk létre.

### Miért népszerű a React a fejlesztők körében?

A React számos olyan előnyt kínál, amely miatt sok fejlesztő első választása:

* **Egyszerű és fokozatos tanulási görbe** – A fejlesztés során a kisebb, egyszerűbb komponensekből könnyen át lehet térni a bonyolultabb struktúrákra.
* **Rugalmasság** – A React nem ír elő szigorú szabályrendszert, így a fejlesztők szabadon választhatják meg az alkalmazott technológiák kombinációját.
* **Újrafelhasználható komponensek** – A kód moduláris, így egyes elemek könnyen módosíthatók anélkül, hogy a teljes alkalmazásra hatással lennének.
* **Deklaratív programozás** – A fejlesztő megadja, hogyan nézzen ki az adott elem az állapot alapján, a React pedig automatikusan frissíti azt, ha változás történik.
* **Univerzális felhasználhatóság** – A React kód használható webes, mobilos és cross-platform alkalmazások fejlesztésére is („Learn once, write anywhere” – ahogy a Facebook szlogenje is mondja).
* **Frontend és backend integráció** – A React kód akár backend rendszerekbe is beágyazható, különösen akkor, ha más eszközkészletekkel, például API-kkal együtt használjuk.

Mi az a Frontend?

A **frontend** az a része egy weboldalnak vagy webes alkalmazásnak, amit **a felhasználó lát és használ**. Ide tartoznak a gombok, szövegek, képek, űrlapok, menük – minden, amit egy böngésző megjelenít.

Egyszerűen szólva: **a frontend a felület**, amin keresztül a felhasználó kapcsolatba lép a webes alkalmazással.

**Modern frontend eszközök**

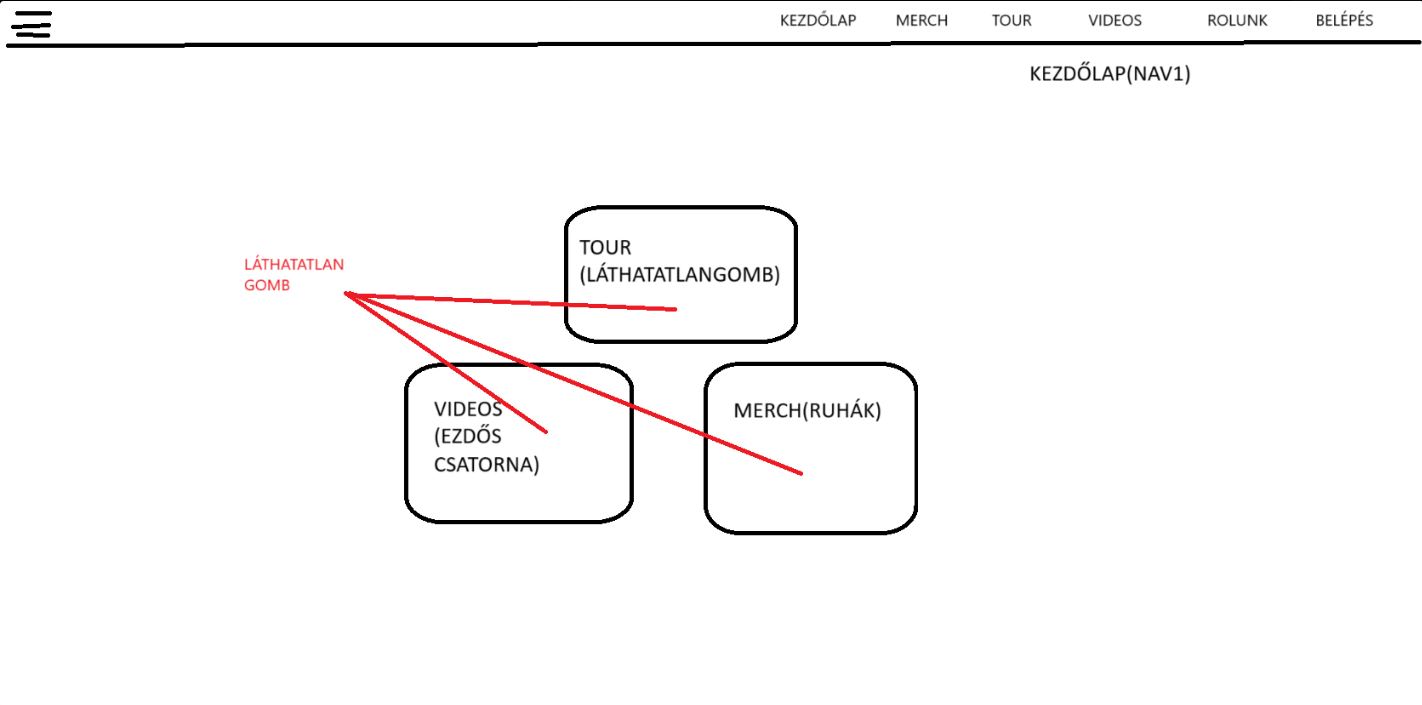
A mai webes frontend fejlesztés sokszor keretrendszerekkel történik, például:

* **React** – nagyon népszerű, komponens-alapú megközelítést használ
* **Vue.js**
* **Angular**

Ezek segítenek abban, hogy a nagyobb projekteket könnyebben, strukturáltabban lehessen kezelni.

Frontend kezdeti tervelés

Tervezés az nélkülözhetetlen egy ilyen projectnél,kell egy látás és kreativitás a megvalósításhoz.A frontenden dolgozó személynek ez is egy fontos feladata, hiszen amit a publikus ember lát az a frontend.Ha az az adott témához nem illik,gagyi,átláthatatlan akkor a látogató ki foglépni az oldalról.Ezért célszerű egy szem kecsegtető kreatív főoldalt kitervelni,később megvalósítani.



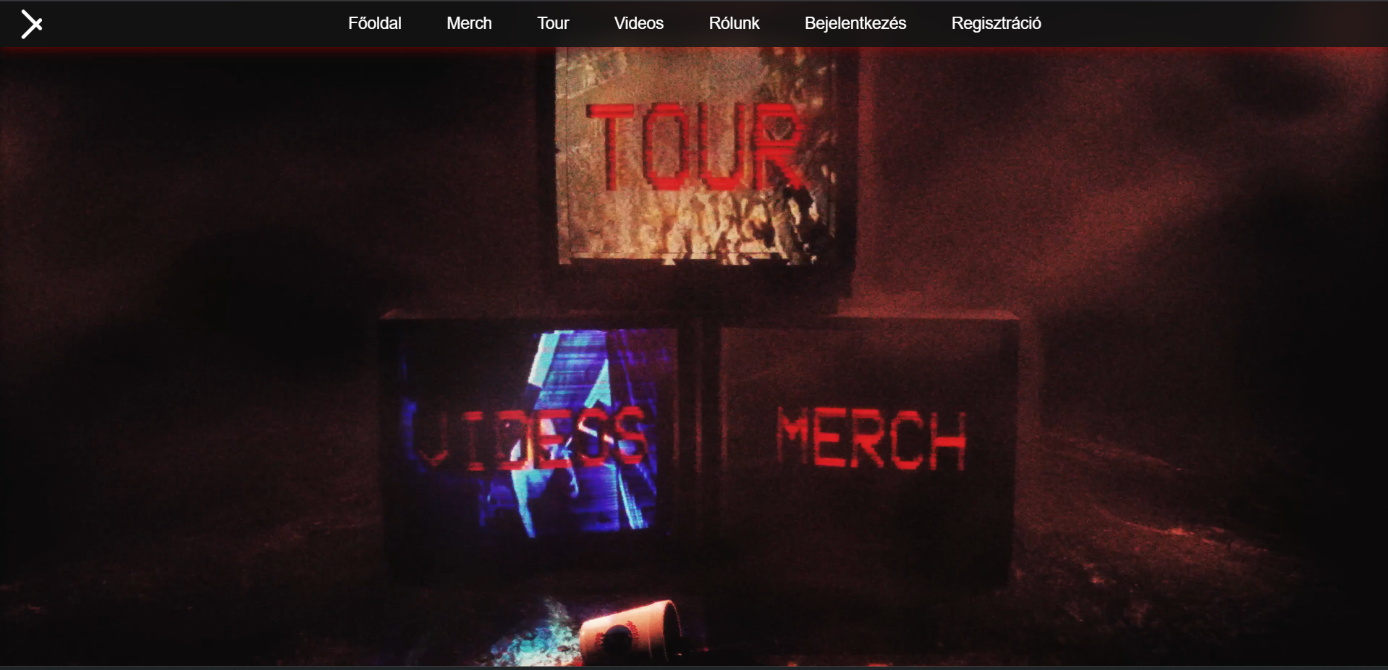
Működési módszerek

A weboldalon a design nagyrésze ami kiemeli az egészet az a háttér videók,ennek működtetéséhez szükségesek láthatatlan gombok amik elvannak sok helyen helyezve a képernyőn.



Final kész kezdőlap

Fontos hogy a kivitelezésnél csiszoljuk a lehető legszebb legjobbra a kinézetét és személy szerint nézegessük napi többször a sablont vagy a félkész főoldalt hátha valami nem tetszik finomításra szorul vagy egy jobb ötleted támad kivitelezésre,győződj meg róla,hogy a legjobb munkád készül el a publikálásra



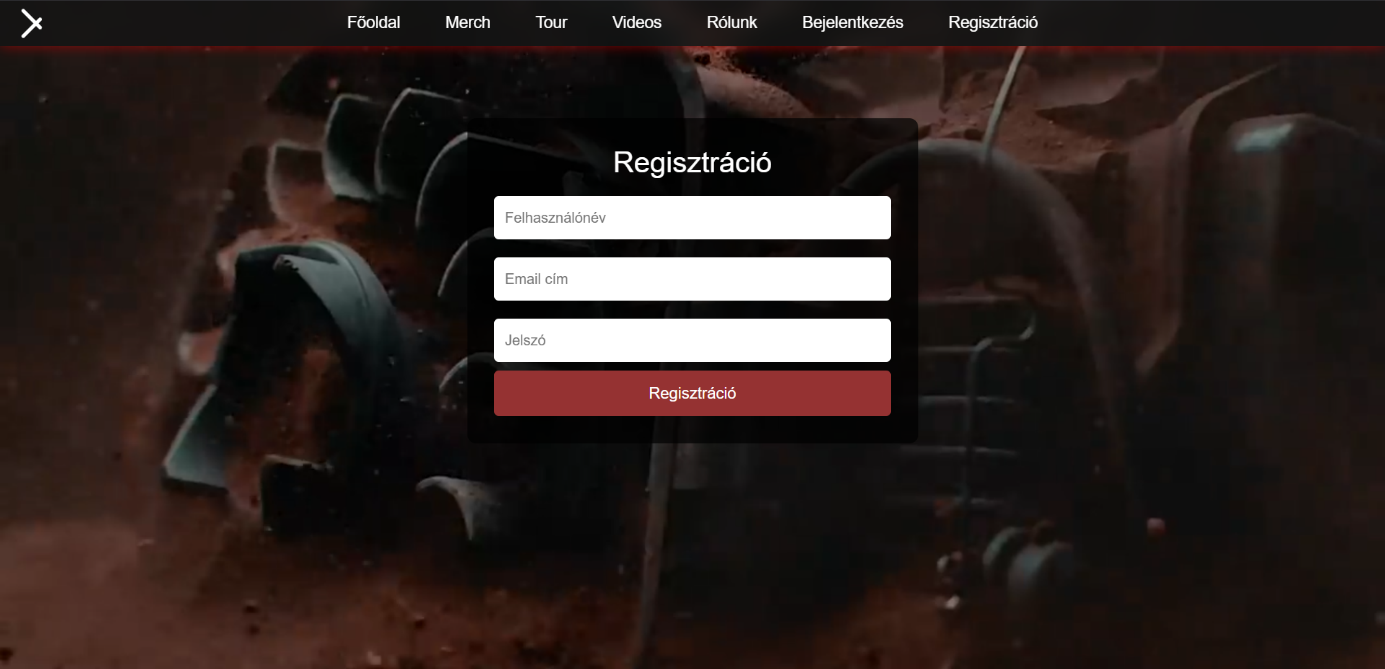
Navbar

A navbár handleNavigation el navigál Főoldal,Mech,Tour,Videos,Rólunk,Bejelentkezés,Regisztráció felületekhez.A bal oldalon található menu-button egyedi animációval és kinézettel rendelkezik.Nem hibából néz ki úgy mint egy relációsjel hanem úgy gondoltuk minden egyes dolog egyedibb legyen a megszokottnál, tűnjön ki a megszokottnál.Menu-buttonra kattintva eltűnnek az opciók és megjelennek,saját animációval.



Bejelentkezés,Regisztráció

A backend segítsége nélkül nem működne.Swagger működteti és adatbázis tárolja.Egyszerű regisztrációs felület semmi bonyolítás.Szép design.



Merch

Merch felületen tud a weboldal látogatója vásárolni,nézelődni a saját termékeink közül.Ez is a backend segítségével működik.Amikor kiválasztod a terméket az bekerül a kosárba ahonnan utána ki is tudod venni ha meggondoltad magad.

### Backend

### C:\Users\Tanulo\Desktop\backend-is.png

**(Web API, ASP.NET Core)**

#### Mi az API és mire használjuk?

Az **API** (Application Programming Interface – magyarul alkalmazásprogramozási felület) egy olyan megoldás, amely lehetővé teszi különböző szoftverek számára, hogy adatokat osszanak meg egymással, vagy meghatározott műveleteket hajtsanak végre egymás között. Bár az API-k már jóval az internet és a PC-k térhódítása előtt is léteztek, az online világ robbanásszerű fejlődése a 2000-es évektől tette őket nélkülözhetetlenné.

Egyszerűen megfogalmazva: az API-k olyan „kapuk”, amelyeken keresztül az egyik alkalmazás utasítást adhat egy másiknak – például adatlekérdezésre, frissítésre vagy egy funkció elindítására.

Fontos megjegyezni, hogy az API **nem egy programozási nyelv**, és **nem is kötődik egy konkrét nyelvhez** – bármilyen környezetben és technológiában kialakítható, majd integrálható a megfelelő szoftverbe. Emiatt gyakorlatilag végtelen számú programban, alkalmazásban és platformon használható.

#### Fejlesztői előnyök

Az API-k használata rengeteg előnyt kínál a fejlesztőknek:

* **Gyorsítja a munkát:** Nem kell minden funkciót a nulláról megírni – elegendő egy már meglévő, jól működő API-t beilleszteni a rendszerbe.
* **Moduláris felépítést tesz lehetővé:** A fejlesztők önálló komponensként tudják kezelni az API-t, így könnyen karbantartható és továbbfejleszthető.
* **Időt és energiát spórol:** Az API használata lerövidíti a fejlesztési ciklust, miközben javítja a kód strukturáltságát.

#### Felhasználói előnyök

Az API-k nemcsak a fejlesztői oldalt támogatják, hanem a felhasználói élményt is jelentősen javítják:

* **Egyszerűség és központosítás:** Több funkció – például vásárlás, időpontfoglalás vagy kommunikáció – egyetlen webes felületen elérhetővé válik.
* **Platformfüggetlenség:** Az API integrálható különböző operációs rendszerekbe (Windows, Linux, stb.), így nem jelent problémát, ha a felhasználók eltérő környezetből érkeznek.
* **Modern szolgáltatások összekapcsolása:** A weboldal könnyedén összeköthető külső szolgáltatásokkal (például Google Naptár, Messenger, WhatsApp), így még kényelmesebbé válik a használat.

### A backend szerepe és feladatai

A **backend** a rendszer azon része, amely a felhasználó számára nem látható, mégis kulcsfontosságú a működés szempontjából. Ez a háttérrendszer végzi el a program működéséhez szükséges számításokat, kezeli az adatokat, és kapcsolatot biztosít a kliens (frontend) és a szerver között.

Elsődleges feladata, hogy a frontendről érkező adatokat gyorsan feldolgozza, majd az eredményeket visszajuttassa a felhasználói felületre. Emellett a backend kezeli az adatbázist is: gondoskodik az adatok biztonságos tárolásáról, azok frissítéséről, módosításáról vagy éppen törléséről.

A backend fejlesztés során úgynevezett **végpontokat (endpointokat)** hozunk létre, amelyek lehetővé teszik az adatcserét a kliens és a szerver között. Ezek a végpontok irányítják például a mentési, törlési vagy lekérdezési műveleteket. Fontos szempont az is, hogy a rendszer megfelelően skálázható és gyors legyen, így a felhasználók mindig gördülékeny élményben részesüljenek.

Swagger

A swagger segítségével tud nyújtani regisztrációt,bejelentkezést,Merch kártyának megvalósítását.Lehet ott tesztelni és valósítani.

Backend tapasztalat

A program gerince, Backend, nagyon fontos hogy tökéletesen hibátlanul legyen megírva. Minden sor után ugyan az az érzés fogott el mint kiskoromba mikor találtam valami új dolgot.

Fontos volt számomra hogy mindent magamtól írjak, és hogy saját tudásomat erősítsem.

Ez nekem egy új élmény volt amit sose fogok elfelejteni, és mindenkinek ajánlom azt hogy egy szobába üljön több órát és csak a számitógépet nyomja minte egy igazi szocializálódott ember!

Kivitelezés

Mi eltérő módon kezdtük az írást, először a frontendel kezdtük amit nem bantunk meg, nekünk a felig kész frontendre könnyebben tudtunk írni backendet.

### Projektmenedzsment – Trello

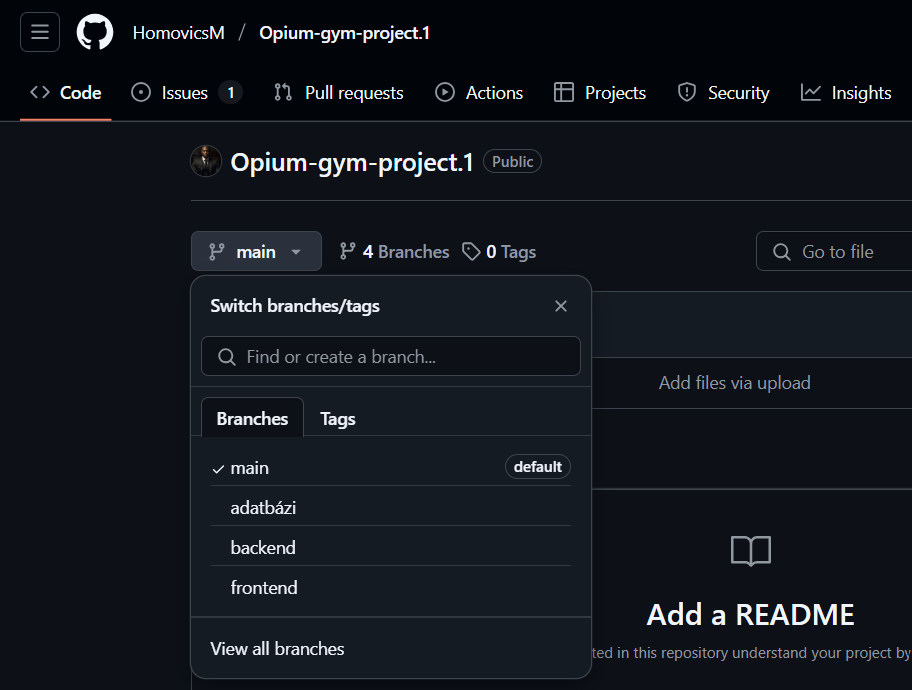
### C:\Users\Tanulo\Desktop\trelllo085914.png

A csapatmunka hatékony szervezése érdekében a **Trello** alkalmazást használtuk projektmenedzsment eszközként. Ez egy rendkívül egyszerűen kezelhető, vizuális alapú felület, ahol létrehozhattuk saját **munkaterünket**, ezen belül **táblákat**, listákat és kártyákat is, hogy átláthatóan tudjuk követni a fejlesztési folyamatot.

A Trello segítségével mindig pontosan láttuk, mely feladatok vannak tervezés alatt, melyek készülnek éppen, és melyekkel végeztünk már. A **Scrum módszertan** alapelveit szem előtt tartva próbáltuk strukturálni a munkát. A felület rendkívül hasznosnak bizonyult: teljesen ingyenes, felhasználóbarát, gyorsan elsajátítható és könnyen testreszabható.

A Trello.com oldalon bárki számára elérhető, azonban saját projektjeinket priváttá tehettük, így csak az általunk meghívott tagok férhettek hozzá és dolgozhattak benne együtt velünk.

Github



Mi az a GitHub?

A **GitHub** egy webalapú platform, amelyet a szoftverfejlesztők világszerte használnak a kódjaik tárolására, megosztására és együttműködésre más fejlesztőkkel. Lényegében egy „közösségi hálózat” programozóknak, de a célja nem a posztolás, hanem a **kód verziókezelése**.

**Mire használják?**

* **Verziókövetésre**: A GitHub a **Git** nevű verziókezelő rendszerre épül, ami lehetővé teszi, hogy a kód változásait nyomon kövessük, visszamenjünk korábbi verziókra, vagy épp párhuzamosan dolgozzunk több fejlesztési ágon (branch).
* **Közös munkára**: Több fejlesztő dolgozhat egyszerre ugyanazon a projekten, egymás kódját át tudják nézni (pull request), és jóváhagyás után beemelhetik a fő kódba.
* **Projektmenedzsmentre**: Lehetőség van hibák (issues), feladatok (tasks) kezelésére, és automatizált munkafolyamatokra is.
* **Publikálásra és portfólióépítésre**: A nyilvános repókban megoszthatod a saját projektjeidet, amit mások megnézhetnek, sőt akár hozzájárulhatnak is.

**Mikor hasznos?**

* Ha csapatban fejlesztesz weboldalt, appot vagy bármilyen programot.
* Ha dokumentálni szeretnéd a saját fejlődésedet vagy portfóliót építesz.
* Ha biztonságosan akarod tárolni a kódod és bármikor vissza akarsz tudni térni egy korábbi verzióhoz.

Források

Segítséget nyújtott a <https://chatgpt.com/>

Inspirációnk <https://www.homixidegang.com/#/>

Inspirációnk <https://www.kencarson.xyz/>

Inspirációnk <https://www.destroylonely.net/#/>

Swagger <https://swagger.io/>

Xampp <https://www.apachefriends.org/hu/index.html>

React <https://react.dev/>

W3Schools <https://www.w3schools.com/react/default.asp>

OPIUM GYM



AKI MER AZ NYER