BGSzC Pestszentlőrinci Közgazdasági és Informatikai Szakgimnáziuma

1184 Budapest Hengersor 34.

1. Záró dolgozat

AnimeHUB

Konzulens tanár: Készítette:

név Kovács Pál Roland, Simon Viktor, Lovas Ádám

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

## Feladat leírás

A program egy webalkalmazás, amelynek célja az, hogy összekösse az anime rajongókat egymással. A weboldalon lehetőség van beszélgetni más felhasználókkal, kedvenc animéket hozzáadni a kedvencekhez, kommenteket írni az animékhez, valamint regisztrálni és bejelentkezni a személyes fiókba.

## A felhasznált ismeretek

* React JS.
* Mongoose.
* Bootstrap.
* Node JS

## A felhasznált szoftverek

* Visula Stusio Code,
* MongoDB webfelulete.
* Chrome.

# Felhasználói dokumentáció

## A program általános specifikációja

A webalkalmazásunk képes a felhasználókat egyedi anime véleményekkel és tapasztalatokkal ellátott közösséget összehozni, ami célja az, hogy mindenki megtalálja a saját ízlésének legjobban megfelelőaniméket, az oldalon a felhasználók képesek egymással chatelni és egyéni ízléseket privátban megbeszélni igy tanácsokat és véleményeket nem nyilvánosan is megosztani. Az oldal folytonos frissítések alatt áll és az anime adatbázis folyamatosan nő, az oldalon a felhasználok megtalálják az anime sztoriját röviden és tömören igy eldönthetik, hogy érdekli-e őket egy bizonyos anime, a projekt célja az az hogy össze kösse a lelkes anime felhasználókat.

## Rendszerkövetelmények

A rendszerkövetelménye a weboldalnak igazából minden olyan számítógép, ami internettel rendelkezik és képes egy web böngészőt elfuttatni.

## A program használatának a részletes leírása

A felhasználónak az oldal összes funkciójának a használatához regisztrálnia kell!

Ha nem regisztrált vagy nem jelentkezett be a felhasználó akkor ezek a funkciók érhetők el:

* A felhasználó képes regisztrálni vagy bejelentkezni.
* A felhasználó meg tudja tekinteni az animéket az összes információt, ami az oldalon van hozzá, de nem képes kommenteket írni, de el tudja az összes meglévőt olvasni.
* A felhasználó képes keresni az animék közül egy keresősáv segítségével.

A képen szöveg, képernyőkép, ruházat, személy látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Grafikus tervezés látható

Automatikusan generált leírás

Ha a felhasználó regisztrált és bejelentkezett akkor ezek a funkciók elérhetőek:

* A felhasználó képes kommenteket közzé tenni, válaszolni mások kommentjére.
* A felhasználó képes privát chatelni más felhasználókkal.
* A felhasználó képes animéket menteni, hogy később megnézhesse mit szeretne megnézni.
* A felhasználó természetesen képes minden olyan funkciót elérni, amit a nem regisztrált vagy nem bejelentkezett felhasználó.
* A felhasználó képes a saját fiókja adatai módosítására a weboldalon.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

# Fejlesztői dokumentáció

## Az alkalmazott fejlesztői eszközök, fejlesztői környezet kialakítása

A program fejlesztéséhez ezeket a szoftvereket, illetve rendszereket használtam:

* Visual Studio Code
* Adatbázisnak a Nosql-t választottam és ehez a MongoDB adatbázis kezelőt használtuk.
* A weboldal css keretrendszeréhez a Bootstrap keretrendszer alkalmaztuk.
* Az egyéni stílusok kialakításához saját css-t használtunk.
* A backend részhez mi node.js-t alkalmaztunk ezen belül is express.js-t.
* Mi React \* Vite keretrendszert használtunk, ami valójában egy JavaScript függvény gyűjtemény.
* A szoftver fejlesztése során a projekttagokkal a GitHub verzió kezelő rendszert használtuk.
* A Backend teszteléséhez Thunder Klienst használtuk, mert ez egy beépített funkciója a Visual Stúdiónak.

### Visual Studio Code verziója:

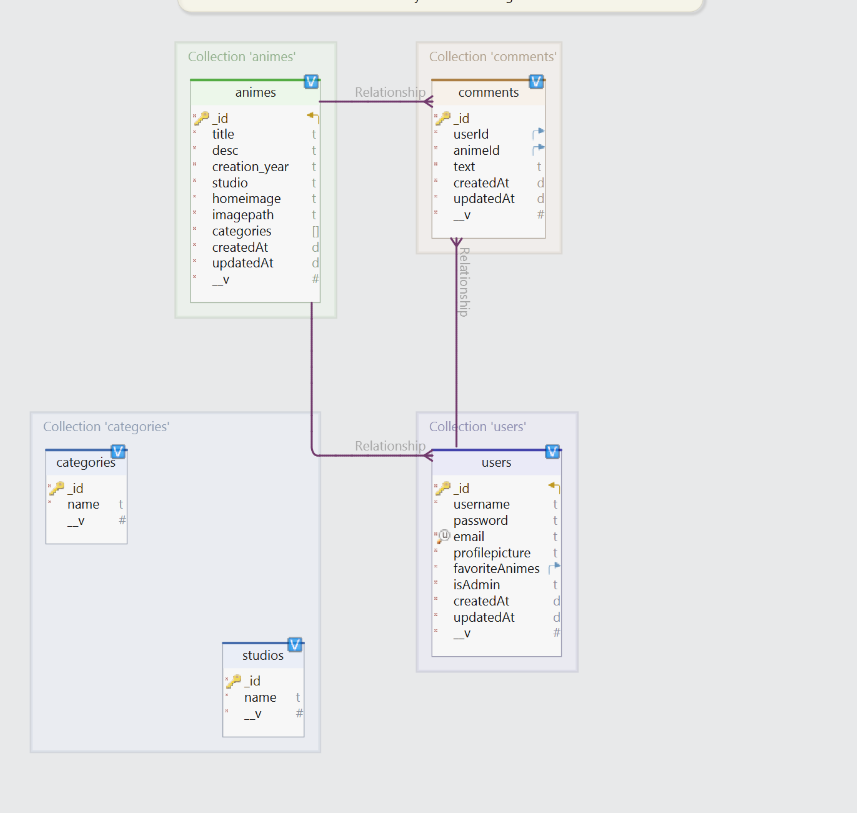
* Version: 1.88.0 (user setup)
* Electron: 28.2.8
* ElectronBuildId: 27744544
* Chromium: 120.0.6099.291
* Node.js: 18.18.2
* V8: 12.0.267.19-electron.0
* OS: Windows\_NT x64 10.0.19045

### A szoftver beuzemeléséhez kell:

* Node.js letöltve.
* npm letöltve megfelelően ez a node.js telepítésével letöltődik.
* Ezek után a githubrol letöltött projekten belül a server mappa termináljában a ’npm install’ parancs után az ’npm start’ parancs lefuttatása.
* Ez után a frontend eléréséhez az animehub-mappa termináljában le kell futtatni az ’npm run dev’ parancsot az ’npm install’ után.

## Adatmodell leírása

* Adatbázis: Anime Gyűjtemény (animeSchema)
* Táblák(collections) leírása:
* animeSchema: Ez a tábla tárolja az animékre vonatkozó információkat.
  + Fields: cím, leírás, készítés éve, stúdió, képek, kategóriák.
* commentsSchema: Ez a tábla tárolja az egyes animékhez fűzött megjegyzéseket.
* Fields: userId, animeId, megjegyzés.
* userSchema: Ez a tábla tárolja a felhasználók adatait.
* Fields: felhasználónév, jelszó, e-mail cím, profilkép, kedvenc animék, isAdmin.
* Kapcsolatok:
* Az animeSchema és a commentsSchema közötti kapcsolat: Az animeId mező az animeSchema tábla egy rekordjára hivatkozik, hogy azonosítsa az animét, amelyhez a megjegyzés tartozik.
* Az userSchema és a commentsSchema közötti kapcsolat: A userId mező az userSchema tábla egy rekordjára hivatkozik, hogy azonosítsa azt a felhasználót, aki a megjegyzést írta.



A képen diagram, képernyőkép, szöveg, kör látható

Automatikusan generált leírás

Ez a diagram megmutatja az USER, ANIME és COMMENT táblák közötti kapcsolatokat. A USER tábla a felhasználókat tárolja, az ANIME tábla az animéket, míg a COMMENT tábla az egyes animékhez fűzött megjegyzéseket. Minden felhasználó több megjegyzést írhat, és minden animéhez több megjegyzés tartozhat. A COMMENT tábla userId és animeId mezői a kapcsolódó felhasználó és animék azonosítójára hivatkoznak.

## Részletes feladatspecifikáció, algoritmusok

• handleSubmit függvény

Feladata: A bejelentkezési adatokat elküldi a szervernek, és kezeli a választ. Paraméterek: e (esemény objektum) Visszatérési érték: Nincs

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

• setToggle függvény: Feladata: Állítja a toggle állapotát. Paraméterek: prevVal (előző állapot) Visszatérési érték: Nincs • Login komponens: Feladata: A bejelentkezés oldal megjelenítése és a bejelentkezési művelet kezelése. Visszatérési érték: JSX elemek.

• Felhasználó kezelése: A kód a useUserContext hook segítségével lekéri a felhasználói kontextust. Ha van bejelentkezett felhasználó, akkor egy "Logout" gombot jelenít meg. Ha nincs bejelentkezve, akkor "Login" és "Register" gombokat jelenít meg.

A képen szöveg, képernyőkép, képernyő látható

Automatikusan generált leírás

• Véletlenszerű anime kiválasztása: A Math.random() metódus segítségével generál egy véletlenszerű számot. Ez a szám szorozva a animes tömb hosszával ad egy véletlenszerű indexet. Ezután az adott indexű anime adatait jeleníti meg az oldalon.

• Anime sorozatok megjelenítése: Az animes tömböt megfordítja, hogy az utolsó bejegyzéseket jelenítse meg először. Az első öt elemet kiválasztja a slice(0, 5) metódussal. Ezeket az elemeket egy map függvénnyel bejárva megjeleníti a kártyák formájában az oldalon.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

• Kedvenc anime sorozatok lekérése: Az axios modul segítségével aszinkron HTTP kérést hajt végre a szerver felé, hogy lekérje a kedvenc anime sorozatokat. Az eredményt a favAnimes állapotba menti, és megjeleníti azokat az oldalon.

• Anime eltávolítása a kedvencekből: A removeAnimeFromFavorites függvény segítségével lehetőség van egy adott anime eltávolítására a kedvencekből. Ez egy újabb aszinkron HTTP kérést indít a szerver felé, hogy eltávolítsa az adott anime-t a kedvencekből. Ezután frissíti a favAnimes állapotot, hogy a megjelenített lista már ne tartalmazza az eltávolított anime-t.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

• Anime adatok lekérése: Az axios modul segítségével aszinkron HTTP kérést hajt végre a szerver felé, hogy lekérje az anime adatokat. A lekérdezett adatokat a newAnimes és animes állapotokba menti el, amelyek a komponens által megjelenített adatokat tárolják.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

• Kategóriák (műfajok) megjelenítése: Az oldalon egy listát jelenít meg az elérhető kategóriákról (műfajokról), amelyek között a felhasználó válogathat. A kategóriák dinamikusan kerülnek megjelenítésre a genres.map függvénnyel, amely a fetchGenres függvény segítségével kerülnek lekérdezésre a szerverről.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

• Keresőmező kezelése: A felhasználó által beírt keresési kulcsszavakat a search állapotban tárolja, és ennek megfelelően szűri az anime adatokat.

• Műfajok alapján szűrés: Amikor a felhasználó kiválaszt egy kategóriát (műfajt), akkor az oldal csak azokat az anime sorozattokat jeleníti meg, amelyek ebbe a kategóriába tartoznak. Ez a filtrerByGenres függvény segítségével történik.

A képen szöveg, képernyőkép, képernyő, Betűtípus látható

Automatikusan generált leírás

• Felhasználói adatok lekérése: Az axios modul segítségével aszinkron módon lekéri a felhasználó adatait a getUserInfo függvénnyel. A lekért adatokat a userInfo állapotban tárolja, hogy azokat megfelelően megjelenítse az oldalon.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

• Kedvencekhez adás: A felhasználó az "Add to favorite" gomb megnyomásával hozzáadhatja az adott animét a kedvenceihez. Az axios.patch metódussal küld egy patch kérést a szerver felé, hogy frissítse a felhasználó kedvenc anime-i közé az új elemet.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

• Hozzászólások megjelenítése: Az oldalon megjeleníti az anime-hoz tartozó hozzászólásokat. A felhasználók itt hozzászólhatnak az anime-hez, és megoszthatják a véleményüket.

• Regisztráció végrehajtása: Az handleSubmit függvényben történik meg az űrlapadatok validációja, majd a szükséges adatok elküldése a szerverre. Ha az űrlapadatok megfelelőek és a felhasználó elfogadta a felhasználási feltételeket, akkor a regisztráció sikeres, és a felhasználót bejelentkezteti.

A képen szöveg, képernyőkép, képernyő látható

Automatikusan generált leírás

• Hibakezelés: Az esetleges hibák (pl.: jelszó nem egyezik meg, felhasználási feltételek elfogadása hiányzik) megjelenítése az űrlap alatt a regisztrációs folyamat során.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

• Felhasználói élmény javítása: Az oldalon lehetőség van a regisztrációhoz külső szolgáltatók (Google, Github, Facebook) fiókjának használatára is.

• Profil műveletek: Az oldalon lehetőség van különböző műveletek végrehajtására, mint például a kedvenc animék megtekintése, profiladatok módosítása, profil törlése vagy az admin felület megnyitása, ha a felhasználó jogosult rá.

A képen szöveg, képernyőkép, szoftver látható

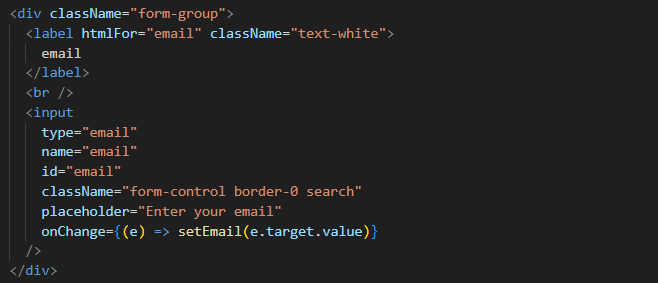
Automatikusan generált leírás

• Navigáció a profil- és animekezelő felületek között: Az oldalsávban található linkek segítségével lehetőség van a profil- és animekezelő felületek közötti navigációra.

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

• Input mezők kezelése és állapotfrissítés: Az input mezők változásait az onChange eseményfigyelők kezelik, amelyek frissítik az állapotot a megfelelő adatokkal.



• Hibakezelés és visszajelzések: A sikeres vagy sikertelen műveletekről a felhasználót visszajelzések informálják. Ezeket a success és error állapotok kezelik.

## Tesztelési dokumentáció

1. Frontend tesztelések:

* A képen szöveg, képernyőkép, szoftver, szám látható

  Automatikusan generált leírásValidációs teszt
* Unit tesztek: ehhez idegen személyeket kértünk meg tesztelésre, hogy a gombok működnek e, és az ehhez hasonló dolgok.
* A képen szöveg, Betűtípus, képernyőkép, szám látható

  Automatikusan generált leírásA képen szöveg, képernyőkép, Betűtípus, Weblap látható

  Automatikusan generált leírásTeljesítmény tesztek

1. Backend tesztek:
   * A képen szoftver, Multimédiás szoftver, Grafikai szoftver, szöveg látható

     Automatikusan generált leírásA képen szöveg, szoftver, Multimédiás szoftver, Grafikai szoftver látható

     Automatikusan generált leírásThunder client tesztek
   * Unit tesztek: szintén más felhasználókkal a regisztráció működik-e, a képek megjelennek e.

# Összefoglalás

## Önértékelés

A projekt készítése során mindent elértünk, amit szerettünk volna terveink szerint, a projekt során sok problémákba ütköztünk például a gombok elhelyezkedése és a komponensek megfelelő importálása, routolás új teknológiája és a bootstrap megfelelő dinamikus kezelése, reszponzivitás problémája, media queryk használata.

A backend programozás során youtube segítségével dolgoztunk, ami megkönnyítette a munkát nagyrészben, de így is voltak problémák például a komment szekcióval.

Az adatbázist 3x terveztük újra mert nem felelt meg a program elvárásainak a legnagyobb problémát az élő beszélgetés jelentette, de ez is megoldásra talált.

Összeségében meg vagyunk elégedve a munkánkkal mert sokat dolgoztunk vele.

## Továbbfejlesztési lehetőségek

* Streaming szolgáltatás
* Online fizetési lehetőség
* Mobil alkalmazás készítése
* A felhasználók legyenek képesek velünk felvenni a kapcsolatot, hogy megoszthassák velünk milyen animéket szeretnének látni az oldalon.

# Felhasznált tartalmak

A képeket ingyenesen felhasználható weboldalrol szedtük bárki elérheti és felhasználhatja őket.

A borítóképeket google képkereső szolgáltatásáról szedtük le, ezeket csak a bemutató idején fogjuk használni mert szerzői jogok védhetik.

A leírásokat chatgpt openai szolgáltatás segítségével írtuk meg és egyéni vélemények alapján, a látott animéket személyes élmény alapján írtuk meg.

A dizájnt azt egy galileo ai online tervező segítségével generáltuk le mert nem volt ötletünk ez miatt 3x kezdtük újra a frontendet.  
  
Az oldalak linkjei:

* <https://stock.adobe.com/>
* <https://pixabay.com/>
* <https://www.google.com/imghp?hl=hu&tab=ri&ogbl>
* <https://www.freepik.com/>
* <https://wallpapers.com/>

# Ábrajegyzék

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