

SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Zenei Bújócska

DIPLOMADOLGOZAT

Témavezető:
Dr. Iclănzan David Andrei,
Egyetemi docens

Végzős hallgató:
Koncz Krisztián

2023

UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
SPECIALIZAREA INFORMATICĂ



UNIVERSITATEA
SAPIENTIA

De-a v-ați ascunselea muzicală

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Iclănzan David Andrei,
Conferențiar universitar

Absolvent:
Koncz Krisztián

2023

**SAPIENTIA HUNGARIAN UNIVERSITY OF
TRANSYLVANIA
FACULTY OF TECHNICAL AND HUMAN SCIENCES
COMPUTER SCIENCE SPECIALIZATION**



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Musical hide and seek

BACHELOR THESIS

Scientific advisor:
Dr. Iclănzan David Andrei,
Associate professor

Student:
Koncz Krisztián

2023

LUCRARE DE DIPLOMĂ

Coordonator științific:
dr. Iclanzan David

Candidat: **Koncz Krisztián**
Anul absolvirii: 2023

a) Tema lucrării de licență: Proiectarea, dezvoltarea și evaluarea, testarea unui joc social bazat pe preferințele muzicale, ce facilitează interacțiunea și cunoașterea prietenilor prin intermediul muzicii, oferind o experiență distractivă și captivantă în descoperirea și partajarea preferințelor muzicale.

b) Problemele principale tratate:

- Studiul bibliografic: cercetarea și analiza literaturii relevante privind psihologia preferințelor muzicale, impactul muzicii asupra interacțiunilor sociale și teoria jocurilor sociale pentru a obține o înțelegere solidă a fundamentelor teoretice ale jocului bazat pe preferințe muzicale.
- Proiectarea și implementarea unei aplicații web interactive și a algoritmilor necesari pentru a permite utilizatorilor să-și exprime preferințele muzicale, să interacționeze cu alți jucători și să descopere și să partajeze muzica preferată.
- Evaluarea utilizabilității și interfeței cu utilizatorul: realizarea unor teste de utilizabilitate pentru a evalua modul în care utilizatorii interacționează cu aplicația, identificarea posibilelor dificultăți de utilizare și optimizarea interfeței și a funcționalităților pentru a îmbunătăți experiența utilizatorului.
- Evaluarea experienței utilizatorilor: colectarea feedback-ului și opiniilor utilizatorilor privind experiența de joc și satisfacția lor în utilizarea aplicației cu ajutorul unui chestionar.
- Analizarea rezultatelor și identificarea posibilelor îmbunătățiri pentru a crea o experiență mai captivantă și satisfăcătoare.
- Documentarea adecvată a stadiilor de proiectare, implementare și testare a aplicațiilor.

c) Desene obligatorii:

- Schema bloc a sistemului
- Diagrame de proiectare, implementare și testare pentru aplicația software realizată.

d) Softuri obligatorii:

- Aplicație web ce implementează jocul social bazat pe preferințele muzicale.
- Dashboard de analiză statistică și vizualizare a feedbackului obținut din chestionar.

e) Bibliografia recomandată:

Leman, M., Demey, M., Lesaffre, M., Van Noorden, L., & Moelants, D. (2009, August). Concepts, technology, and assessment of the social music game" sync-in-team". In *2009 International Conference on Computational Science and Engineering* (Vol. 4, pp. 837-842). IEEE.

Lehtinen, V., & Liikkanen, L. (2012). The meanings of music sharing in tween life. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems* (pp. 1907-1912).

Hein, E. (2014). Music games in education. *Learning, Education and Games*, 93.

Mansour, E., Sambra, A. V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., ... & Berners-Lee, T. (2016, April). A demonstration of the solid platform for social web applications.

In *Proceedings of the 25th international conference companion on world wide web* (pp. 223-226).

Bell, G. (2009). *Building social Web applications: Establishing community at the heart of your site.* " O'Reilly Media, Inc."

f) Termene obligatorii de consultații: săptămânal, preponderent online

g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca,
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, sala / laboratorul 413

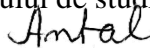
Primit tema la data de: 25.04.2022

Termen de predare: 06.07.2023

Semnătura Director Departament



Semnătura responsabilului
programului de studiu



Semnătura coordonatorului



Semnătura candidatului



Declarație

Subsemnatul/a KONCZ KRISZTIÁN, absolvent(ă) al/a specializării INFORMATICA, promoția 2023 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, TÂRGU MUREȘ
Data: 08.06.2023

Absolvent

Semnătura....Koncz.....

Kivonat

Dolgozatom témája egy webes applikáció fejlesztése, ami segíthet egy kicsit jobban megismerni a régi, de leginkább új ismerőseinket a zenei ízlésükön keresztül.

Az embereknek a zenei ízlése akár sok mindent elárulhat a személyiségükről, valamint, ha tudjuk valakinek a zenei ízlését utána könnyebb beszélgetéseket is kezdeményezni, így ez is segíthet az ismerkedések, barátkozások korai szakaszában.

Egy ilyen applikáció jó jégtörő is lehet más ismerkedős játékok mellett, mert akár vicces is lehet zenéken keresztül megismerni egymást és illetve mi is akár új zenei stílusokat vagy számokat ismerhetünk meg, amikkel lehet, hogy nem találkoztunk volna.

Az online felület miatt akár úgy is tudunk játszani, hogy nem vagyunk egy helyiségben, így akár online oktatás alatt, vagy online munka alatt is jobban megtudjuk ismerni a kollégáinkat.

Rezumat

Tema lucrării mele de licență este dezvoltarea unei aplicații web, care ajută în cunoașterea mai bună a prietenilor vechi și noi prin gustul lor muzical.

Gustul muzical al unei persoane dezvăluie mult despre personalitatea acesteia. Dacă cunoaștem gustul muzical al cuiva e mai ușor să începem o conversație, iar acest lucru ajută foarte mult în stadiile de început ale prieteniiilor.

O astfel de aplicație, împreună cu jocurile de cunoaștere, facilitează voia bună, deoarece este distractiv să ne cunoaștem prin muzică. De asemenea ne vom întâlni cu genuri muzicale și cântece necunoscute până în acel moment.

Interfața online permite joaca și când nu ne aflăm în aceeași încăpere. Putem petrece timp împreună și ne putem cunoaște mai bine și în timpul orelor sau muncii online.

Abstract

Summary The theme of my bachelor's thesis is the development of a web application which helps getting to know old and new acquaintances through their musical taste.

The musical taste of a person reveals a lot about their personality. Furthermore, if we know somebody's musical taste it is easier to start a conversation which is useful in the early stages of a friendship.

Such an application, together with get to know you games, is a very good icebreaker, because it can be funny to get to know somebody through music. In addition, we can get acquainted with musical genres or songs previously unknown to us.

The online interface enables us to play even when we are not in the same location. We can get closer to our colleagues during online classes or online work

Tartalomjegyzék

1. Bevezető	11
1.1. Téma kiválasztása	11
2. Projekt Célja	12
3. Felhasználói követelmények	13
3.1. Use Case Diagram	13
3.2. Felhasznált Technológiák	14
3.2.1. React	14
3.2.2. Chakra UI	14
3.2.3. Express.js	15
3.2.4. Socket.io	15
3.2.5. Typescript	15
3.2.6. Vite	16
3.2.7. Youtube API	16
3.2.8. Git	16
4. Projektmenedzsment Terv	18
4.1. Trello	18
4.2. Git	21
5. Projekt Működése	23
5.1. Szekvencia-Diagram	23
5.2. Kezdőoldal	24
5.3. Lobby (Előszoba)	24
5.4. Gamescene (Játéktér)	25
6. Felhasználói Felület Bemutatása	26
6.1. Kezdőoldal	26
6.2. Lobby (Előszoba)	27
6.3. Gamescene (Játéktér)	28
7. Továbbfejlesztési Lehetőségek	31
7.1. Több API használata	31
7.2. Lejátszási listák helyben való létrehozása	31
7.3. Lejátszási listák módosítása	31
7.4. Lejátszási lista hasonlóság	32

7.5. Google fiókkal való bejelentkezés	32
7.6. Email címmel való regisztráció	32
7.7. Tesztek hozzáadása	32
7.8. Nemmel való szavazás	32
7.9. Lejátszási lista ellenőrzése	32
7.10. 2 játékos mód	32
Összefoglaló	33
Ábrák jegyzéke	35
Irodalomjegyzék	36

1. fejezet

Bevezető

1.1. Téma kiválasztása

Az ötlet onnan jött, hogy elgondolkodtam milyen lenne egy olyan összerázó játék, ami az emberek zenei ízlésére teszi a hangsúlyt.

Egy új kapcsolatban vagy barátságban a közös érdeklődési területek megtalálása fontos szerepet játszik, és a zenei ízlés erre ideális lehetőséget kínál. Az alkalmazás lehetővé teszi, hogy megosszuk a kedvenc előadóinkat, zenekarainkat, albumainkat vagy éppen dalainkat, ezáltal a másik félnek lehetősége van bepillantást nyerni a mi zenei világunkba, ami sokat elárulhat rólunk, illetve mi is jobban megismerhetjük őket.

2. fejezet

Projekt Célja

A Projekt célja egy olyan webes applikáció megvalósítása, ami lehetővé teszi, külön játék szobák létrehozását, azokba több felhasználó csatlakozását, illetve azokban zenei lejátszási listák beküldését, amikből aztán a felhasználók körökre osztva minden körben kapnak egy véletlenszerűen kisorsolt zeneszámot és erről kell eldöntésük, hogy az melyik játékosnak a lejátszási listájából származik.

Amelyik játékos jól tippel, az pontot kap, aki pedig nem találja el, az pontot veszít.

Ezek mellett fontos szempont volt a lehetséges csalási felületek lecsökkentése, ezért is például a backend az adott lejátszási listákból kiszedi a zeneszámokat és körönként csak azok közül küld el egyet, nem az egész lejátszási listát adja át a frontendnek.

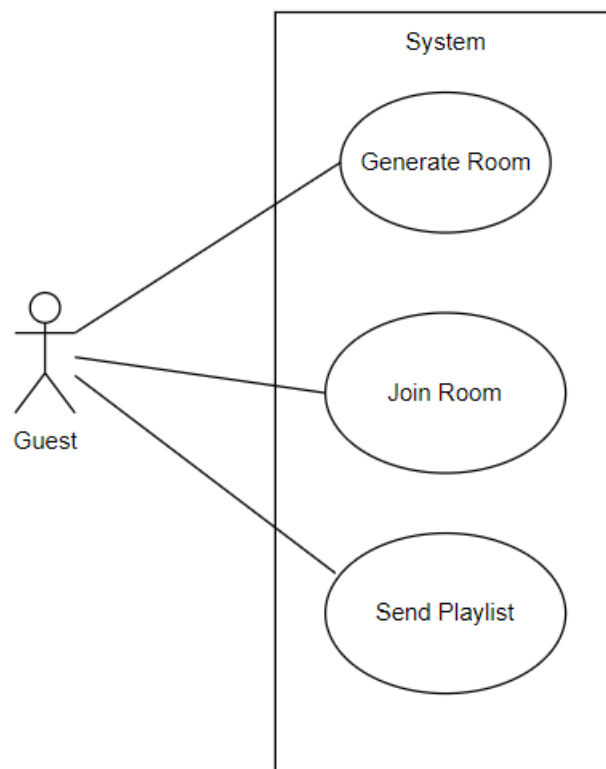
3. fejezet

Felhasználói követelmények

3.1. Use Case Diagram

A webapplikációban egy fajta felhasználói csoport van:

- Egyszerű felhasználó: jogosult lobby létrehozására, lobbyba való csatlakozásra, valamint lejátszási lista küldésére.



3.1. ábra. Use Case Diagram - Guest felhasználó

3.2. Felhasznált Technológiák

A projekt során több technológiát is használtam, a front-end részén a React nevű Javascript könyvtárat Chakra-UI komponens könyvtárral, back-end részén a Node.js-nek az Express.js nevű keretrendszerét. Mind a front-end és mind a back-end részén használtam Typescriptet a biztonságosabb fejlesztés érdekében. A kliens oldalt a szerver oldallal websocketekkel kötöttem össze, amiben a Socket.io segített, így két irányú kapcsolatra lett lehetőség, a kliens oldal tud küldeni és fogadni adatot a szerver oldaltól és a szerver oldal szintén tud küldeni és fogadni adatot a kliens oldaltól.

3.2.1. React

React [\[Reaa\]](#) egy JavaScript könyvtár, amely segít a felhasználói felület (UI) létrehozásában és kezelésében. A React alkalmazása során az alkalmazás felhasználói felülete komponensekre van bontva, amelyek különböző adatokat jelenítenek meg és szolgáltatnak interakciós lehetőségeket a felhasználóval. Az egyes komponensek egyszerűen összeszerelhetők, és újrahasznosíthatók más alkalmazásokban is.

A React használatának néhány előnye:

- Hatékony és gyors: A React hatékonyan kezeli a komponenseket, így az alkalmazások nagyon gyorsak és hatékonyak lehetnek.
- Újrahasznosítható komponensek: A React komponensei újrahasznosíthatók, ami időt és erőforrásokat takarít meg.
- Virtuális DOM: A React virtuális DOM-ot használ, amely lehetővé teszi a változások hatékony kezelését.

3.2.2. Chakra UI

A Chakra UI egy React alapú felhasználói felületi (UI) könyvtár, amely könnyen használható és testreszabható design elemeket kínál az alkalmazások számára. A könyvtár célja, hogy egyszerűvé tegye a fejlesztők számára a felhasználói felületek építését, miközben kiváló minőségű és modern design elemeket biztosítanak az alkalmazások számára.

A Chakra UI számos előnnyel rendelkezik, amelyeket érdemes figyelembe venni, amikor a React alkalmazások front-end részének fejlesztéséről van szó:

- Egyszerű és könnyen használható: A Chakra UI egy egyszerű és könnyen használható könyvtár, amely lehetővé teszi a fejlesztők számára a kód gyors írását és az alkalmazások gyorsabb készítését.
- Könnyen testreszabható: A Chakra UI lehetővé teszi a fejlesztők számára, hogy testreszabják az alkalmazások design elemét a szükségleteiknek megfelelően, miközben megtartják a modern és elegáns megjelenést.
- Jól dokumentált: A Chakra UI részletes dokumentációval rendelkezik, amely segíti a fejlesztőket a könyvtár használatában és a design elemek testreszabásában.

3.2.3. Express.js

Az Express.js [\[exp\]](#) egy Node.js alapú webalkalmazás keretrendszer, amely lehetővé teszi a fejlesztők számára az egyszerű és hatékony webalkalmazások létrehozását. Az Express.js egy nagyon könnyű és rugalmas keretrendszer, amely könnyen testreszabható a szükségletek szerint, és lehetővé teszi az egyszerű webalkalmazások, valamint a nagyobb és bonyolultabb webalkalmazások fejlesztését is.

Az Express.js számos előnnyel rendelkezik, amelyeket érdemes figyelembe venni, amikor a Node.js-alapú webalkalmazások fejlesztéséről van szó:

- **Könnyű és rugalmas:** Az Express.js egy könnyű és rugalmas keretrendszer, amely lehetővé teszi a fejlesztők számára a gyors és hatékony webalkalmazások létrehozását.
- **Nagy és aktív közösség:** Az Express.js rendelkezik egy nagy és aktív fejlesztői közösséggel, amely segíti a fejlesztőket a problémák megoldásában.
- **HTTP függőségek kezelése:** Az Express.js lehetővé teszi a fejlesztők számára, hogy egyszerűen kezeljék az HTTP kéréseket és válaszokat.

3.2.4. Socket.io

A Socket.IO [\[soc\]](#) egy JavaScript könyvtár, amely lehetővé teszi a valós idejű kétirányú kommunikációt a böngésző és a szerver között. A Socket.IO a WebSocket protokollra épül.

A Socket.IO számos előnnyel rendelkezik, amelyeket érdemes figyelembe venni, amikor a valós idejű kommunikációról van szó:

- **Valós idejű kommunikáció:** A Socket.IO lehetővé teszi a valós idejű kommunikációt a böngésző és a szerver között, amely lehetővé teszi a felhasználók számára azonnali visszajelzést és interakciót.
- **Egyszerű használat:** A Socket.IO könnyen használható és könnyen integrálható az alkalmazásokba, így egyszerűen lehet valós idejű kommunikációt létrehozni az alkalmazásokban.
- **Egyszerű hibakeresés:** A Socket.IO lehetővé teszi a fejlesztők számára, hogy egyszerűen hibakeresést végezzenek és problémákat javítsanak a kommunikáció során.

3.2.5. Typescript

TypeScript egy nyílt forrású nyelv, amely a JavaScript kiegészítéseként funkcionál. A TypeScript lehetővé teszi a fejlesztők számára, hogy erőteljesebb és strukturáltabb kódokat írjanak, amelyek könnyebben karbantarthatók és kiterjeszthetők, mint a hagyományos JavaScript.

A TypeScript fő előnyei:

- **Típusbiztonság:** A TypeScript lehetővé teszi a fejlesztők számára, hogy erőteljesebb és strukturáltabb kódokat írjanak, amelyeket a TypeScript fordító a futás előtt ellenőriz, ezáltal csökkentve a hibák számát és növelve a fejlesztési sebességet.

- Jobb karbantarthatóság: A TypeScript lehetővé teszi a fejlesztők számára, hogy szigorúbban szervezzék és strukturálják a kódot, amely javítja a karbantarthatóságot és az olvashatóságot.

3.2.6. Vite

Vite egy gyors és hatékony build tool, amely a modern webalkalmazások fejlesztéséhez készült. A Vite a Vue.js alkalmazásokhoz készült, de használható más keretrendszerrel és JavaScript könyvtárakkal is.

A Vite fő előnyei:

- Gyors build idő: A Vite a fejlesztési szerver használatával rendkívül gyors build időt biztosít, amely lehetővé teszi a változtatások gyors visszajelzését a fejlesztői folyamat során.
- Javított teljesítmény: A Vite az ESM modulokat (ES6 modulok) használja, amelyek gyorsabbak és hatékonyabbak, mint a hagyományos CommonJS modulok.
- Egyszerű beállítás: A Vite nagyon könnyen beállítható, és nincs szükség hosszadalmas konfigurációkra.

3.2.7. Youtube API

Az YouTube API (Application Programming Interface) egy programozási felület, amely lehetővé teszi az alkalmazások számára, hogy az YouTube videómegosztó platformot használják és integrálják saját alkalmazásaikba. Az API-k egyes funkciókat és adatokat tesznek elérhetővé a fejlesztők számára, amelyek segítségével könnyen integrálhatják az YouTube funkcióit az saját alkalmazásukba.

A YouTube API-t számos különböző célra használhatják a fejlesztők, például:

- Videókezelés: A fejlesztők képesek feltölteni, letölteni, frissíteni és lejátszani a videókat az API-n keresztül.
- Keresés: A fejlesztők képesek a kereső funkciót használni az API-n keresztül, hogy keresse meg a videókat.

3.2.8. Git

Git egy nyílt forráskódú verziókezelő rendszer, amelyet a fejlesztők használnak az alkalmazások és projektek változásainak nyomon követésére és kezelésére.

A Git fő előnyei:

- Verziókezelés: A Git lehetővé teszi az alkalmazások és projektek változásainak nyomon követését, így a fejlesztők bármikor visszatérhetnek az előző verziókhoz, ha szükséges.
- Együttműködés: A Git lehetővé teszi a fejlesztők számára, hogy egyszerűen együttműködjenek a projektben, a különböző változtatásokat összevonják és kezeljék.

- Biztonságos adattárolás: A Git tárolja a projekt adatait a fejlesztők számítógépén és a távoli szerveren is, így a projekt adatai mindig biztonságban vannak.

4. fejezet

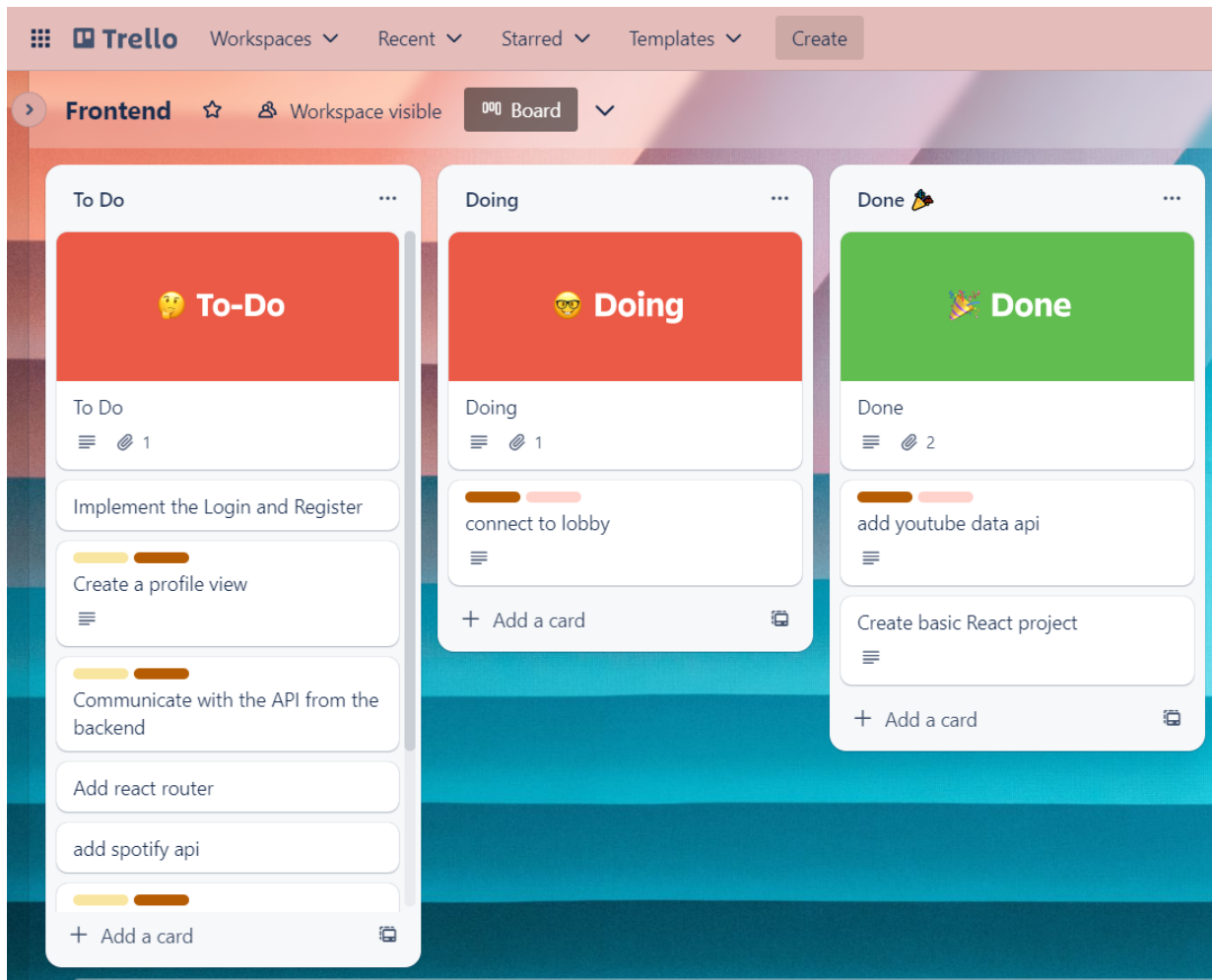
Projektmenedzsment Terv

4.1. Trello

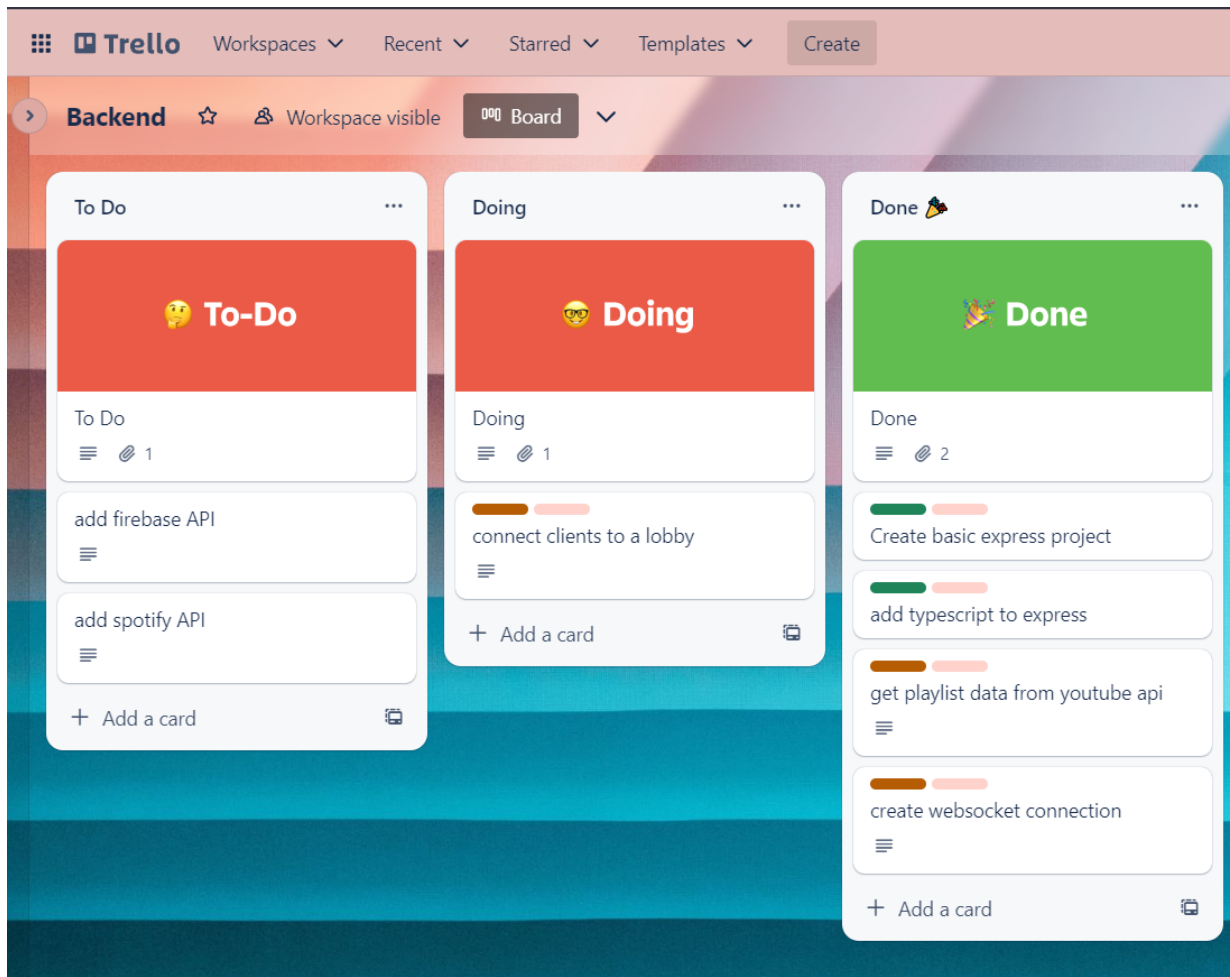
A projektmenedzsmenthez során a Trello nevű projektmenedzsment rendszert használtam. Két Trello Boardot hoztam létre külön a front-endnek és külön a back-endnek. Ezekben próbáltam meg vezetni az elvégzendő (To-Do), jelenleg munka alatt levő (Doing), illetve elvégzett feladatokat (Done), a feladatokat ebbe a három oszlopba rendeztem.

Minden feladatot kettő címkével láttam el: Az egyik jelezte a feladat becsült nehézségét, a másik jelezte a feladat prioritását. Mindkét címkének három-három fajtája volt, a címkének, ami a nehézséget mutatta Difficulty:Easy, Difficulty:Medium, Difficulty:Hard, a prioritási címkének pedig Priority:Low, Priority:Medium, Priority:High. Ez segített a feladatok átlátásában és a következő elvégzendő feladat kiválasztásában. Szükség esetén írtam leírást is a feladathoz, ha nem volt egyértelmű a cím alapján.

A Trello segített a feladatok átlátásában, nyomon követésében. Azért ezt a projektmenedzsment rendszert használtam, mert ezt már használtam korábbi projektnél és eléggé ismert.



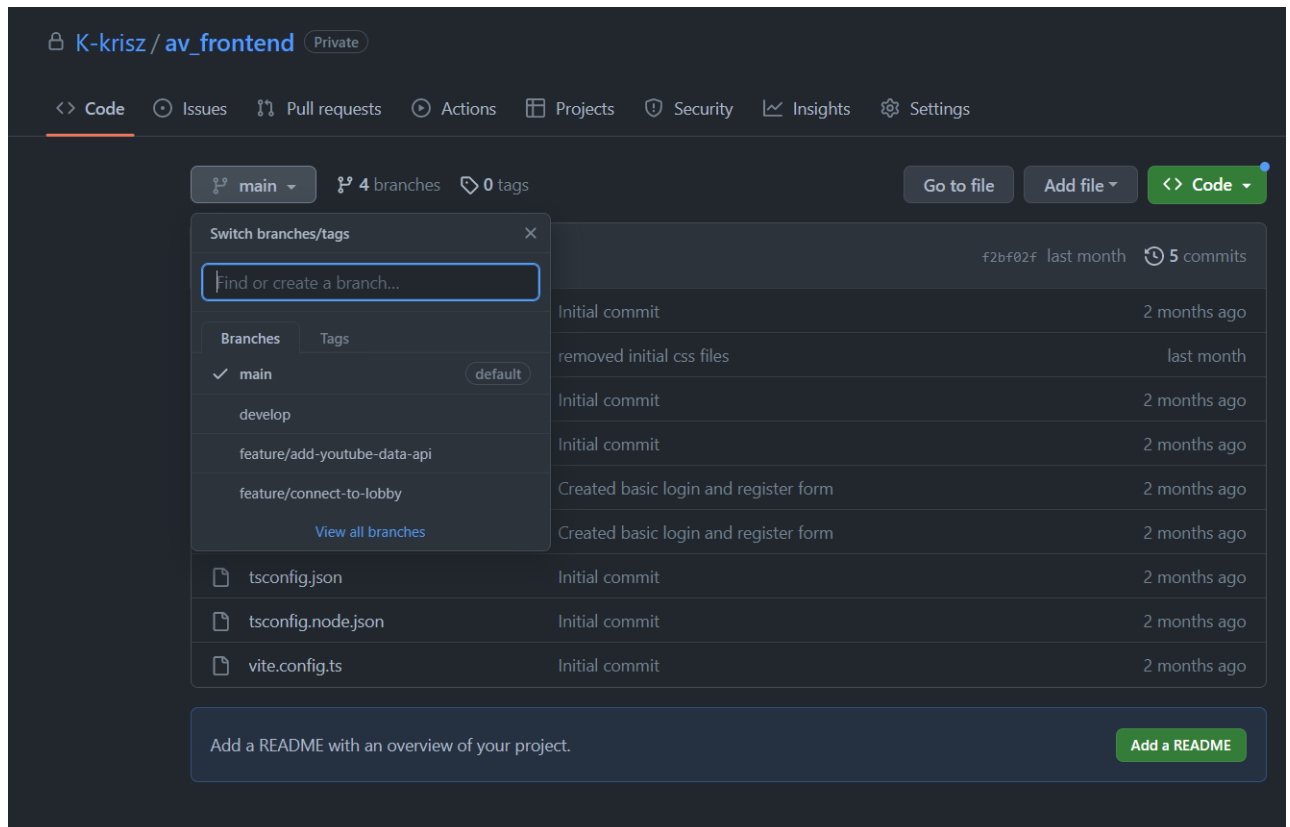
4.1. ábra. Trello Frontend



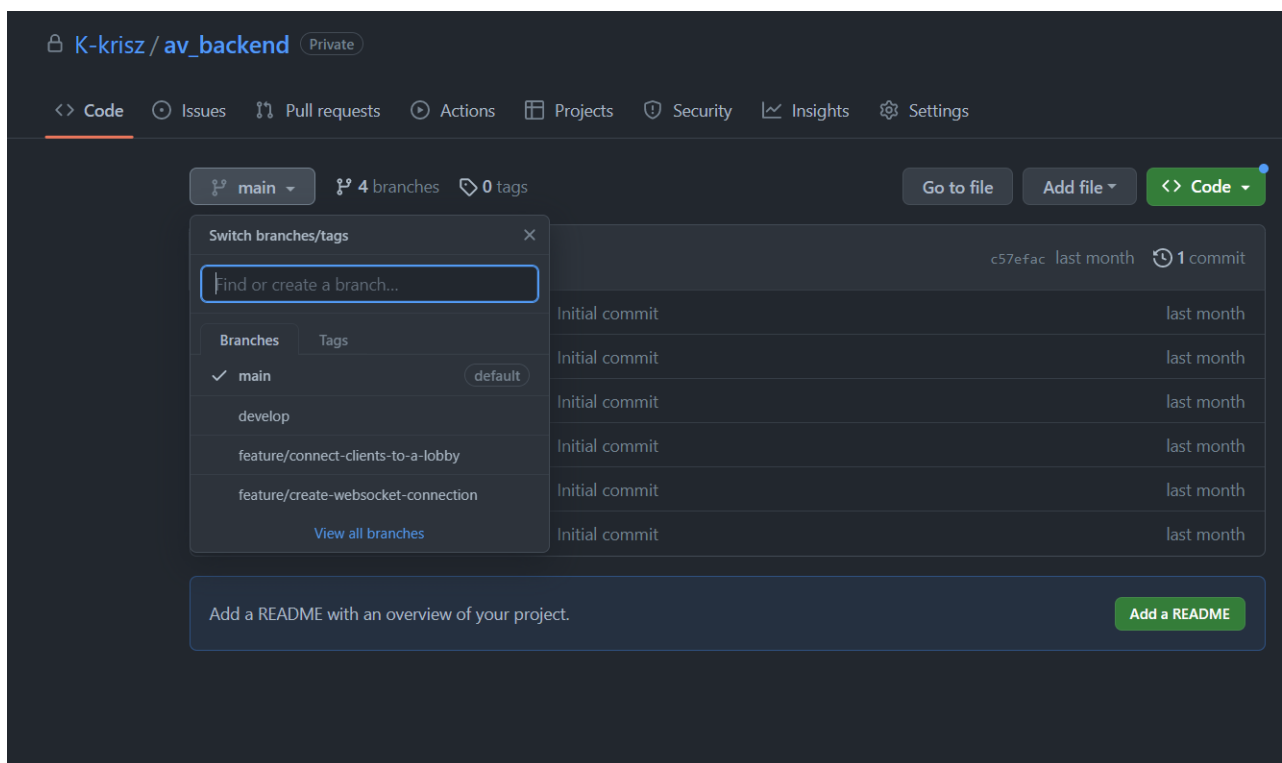
4.2. ábra. Trello Backend

4.2. Git

Verziókövetéshez Gitet használtam és Githubon tároltam a programkódot. A Trellohoz hasonlóan itt is külön kezeltem a front-endet és a back-endet. Kettő alap branchet használtam: main branchet és develop branchet. A main branchen már kiadható verziót tároltam, a developon pedig működő, de még nem teljes verziót. Amikor elkezdtem egy nagyobb feladatot, akkor létrehoztam egy új branchet a developról feature/task-neve névvel, amikor pedig befejeztem a feladatot, akkor mergelem a develop branchre, így minden befejezett task után frissült a develop branch.



4.3. ábra. Github Frontend



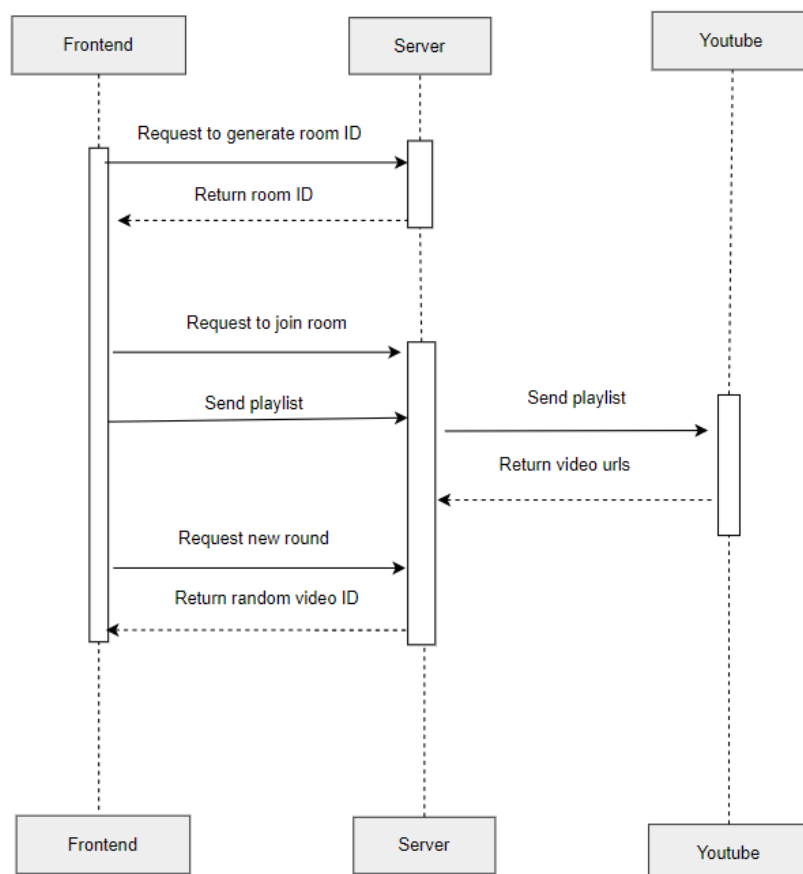
4.4. ábra. Github Backend

5. fejezet

Projekt Működése

Ebben a fejezetben a projekt működéséről lesz szó, milyen részekből áll a weboldal, illetve a weboldal részeinek melyiknek mi a célja.

5.1. Szekvencia-Diagram



5.1. ábra. Szekvencia-Diagram

5.2. Kezdőoldal

Amikor megnyitja a weboldalt a felhasználó, először a kezdőoldallal találkozik, innen van lehetősége szoba azonosítót generálni, illetve szobához csatlakozni.

Ha a felhasználó úgy dönt, hogy ő szeretné létrehozni a szobát, akkor, amikor a "Generate Room" gombra kattint, a frontend websocket kapcsolaton keresztül szól a backendnek, hogy generáljon egy szoba azonosítót, a backend kigenerál egy véletlenszerű szoba azonosítót a `Math.random()` függvény segítségével, majd miután kész van, visszaküldi azt a felhasználónak.

A felhasználó ezt a generált szoba azonosítót megtudja osztani a kívánt személyekkel, akikkel játszani szeretne.

Ahhoz, hogy a felhasználó csatlakozni tudjon egy szobára, amit vagy ő generált, vagy kapott valakitől a "Room id..." mezőbe illesztve és a "Join" gombra kattintva tud.

Miután a "Join" gombra kattintott átirányítódik a lobbyba (előszobába), a `react-router-dom` [reab] könyvtár segítségével.

5.3. Lobby (Előszoba)

A lobby szobába érkezve a a frontend lekéri websocket kapcsolattal a backendtől a szobában levő többi játékost, így meg tudja jeleníteni, hogy kikkel vagyunk jelenleg egy szobában, illetve, mivel websocket kapcsolat nem kell a frontend kéréseket küldjön megadott időközönként, hogy jött-e új felhasználó a szobába, hanem a backend tud szólni a frontendnek, ha változás történt.

Ha a játékos eldöntötte, hogy melyik lejátszási listával szeretne játszani megnyomja a "Send" gombot, ami elküldi a lejátszási lista linket a backendnek, ami a youtube API-n keresztül kieszdi a lejátszási listából a számokat és úgy tárolja a backenden.

Az így kapott youtube videóknak az azonosítóját (ami a videó URL-nek a v paramétere) egy objektumban 5.2 tárolja, amiben található több szoba azonosító (`roomId`), minden szoba azonosítón belül van több felhasználó objektum, amik tartalmazzák az adott felhasználó lejátszási listájából a számoknak az azonosítóját tartalmazó karakterlánc tömböt.

```
export interface roomsUsersSongs {  
  [roomId: string]: {  
    [username: string]: {  
      songUrls: string[]  
    };  
  };  
}
```

5.2. ábra. Typescript Interface

A játék csak azután indul, miután minden lobbyban levő játékos elküldte a lejátszási listáját, ekkor mindenki átirányítódik a játéktérbe (`GameScene`).

5.4. Gamescene (Játéktér)

Amikor a lobbyban való várakozás után átirányítodnak a játékosok a játéktérbe, akkor kezdődik a tippelés, vagyis akkor kell eltalálni kinek a lejátszási listájából van az adott körben megjelenített zeneszám.

A játéktérben láthatjuk a velünk játékban levő felhasználókat, valamint saját magunkat.

Amikor valaki rákattint a "New Round" gombra, akkor a frontend jelez a backendnek, hogy új kört szeretnénk indítani, vagyis küldjön egy új véletlenszerű youtube videót, amit majd a megfelelő szobába vissza is küld. Ez a youtube videó egy iframe segítségével jelenítődik meg, ezzel egyidejűleg pedig küld egy másik kérést szintén a backendnek, hogy indítson el egy időzítőt, ami 10 másodpercről kezd lefele számolni és ezt továbbítja a megfelelő szobának megjelenítésre.

A backendről csak a véletlenszerűen kapott youtube videó azonosítót kapja a frontend, nem az egész lejátszási listának az azonosítóját, így a felhasználók nem tudják kikövetkeztetni, hogy a kapott szám az kitől származik, mivel megjelenített iframe-ben sem az egész lejátszási lista van, illetve a chrome developer toolsban sem lehet megnézni a lejátszási listának az azonosítóját, mivel csak egy videó azonosítót adunk át.

Az a játékos, aki nem szavazott, az nem kap pontot. Minden helyes tippért egy pontot kap a tippelő játékos, illetve minden helytelen tippért egy pontot veszít.

Amint lejárt az idő a frontend küld egy kérést a backendnek, hogy küldje el az eddigi pontszámokat, hogy tudja azt frissíteni és megjeleíteni.

6. fejezet

Felhasználói Felület Bemutatása

Ebben a fejezetben a felhasználói felület bemutatására kerül sor, itt ismerhetjük meg, hogy milyen felülettel találkozhatnak a felhasználók

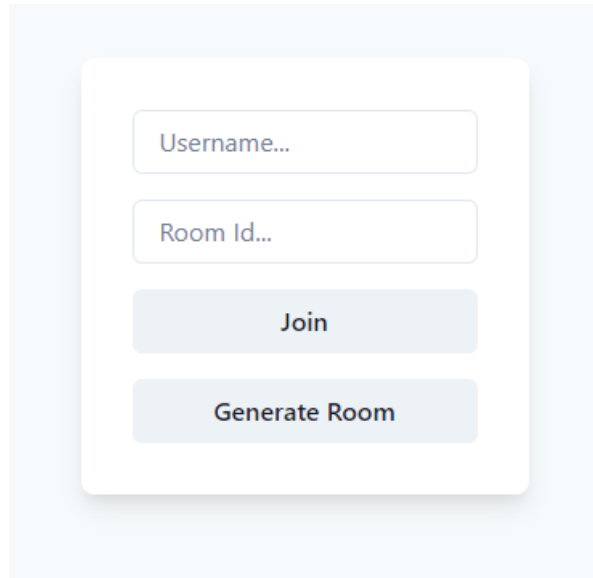
6.1. Kezdőoldal

A kezdőoldalon egy bejelentkezési felület [6.1](#) fogad minket, ahol betudunk jelentkezni guest (vendég) felhasználóként. Ahhoz, hogy el tudjuk kezdeni a játékot először ezen az oldalon meg kell adni a felhasználói nevünket, ami alapján szeretnénk, hogy más játékosok felismerjenek.

Miután megadtuk a felhasználói nevünket meg kell adnunk egy szoba azonosítót is, ez szükséges ahhoz, hogy egy játékba tudjunk kapcsolódni a kívánt személyekkel.

Ha szeretnénk, akkor létre tudunk hozni egy szobát a "Generate Room" gombbal, ami visszaad egy generált szoba azonosítót és ennek segítségével egy játékba tudunk kapcsolódni a többi játékoskal. Ezt az azonosítót ha elküldjük azoknak, akikkel szeretnénk játszani, akkor miután bemásolták a szoba azonosítót a megfelelő mezőbe betudnak csatlakozni a létrehozott szobába a "Join" gomb segítségével.

Miután megnyomtuk a "Join" gombot a weboldal átirányít a Lobby (előszoba) felületre.

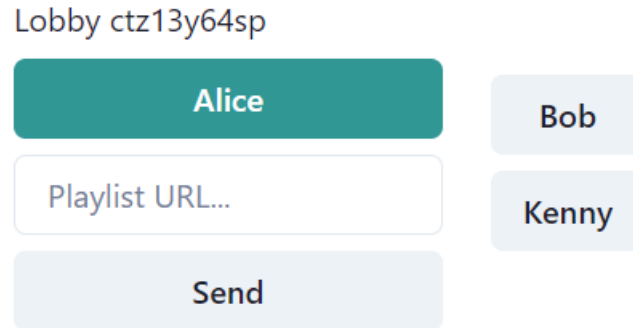
A light blue rectangular form with rounded corners is centered on a slightly darker blue background. Inside the form, there are two white input fields with rounded corners. The first field contains the placeholder text 'Username...' and the second field contains 'Room Id...'. Below these fields are two light blue buttons with rounded corners. The top button is labeled 'Join' and the bottom button is labeled 'Generate Room'.

6.1. ábra. Kezdőoldal

6.2. Lobby (Előszoba)

A lobby szerepe, hogy legyen egy felület, ahol a játékosok előkészíthetik a kívánt lejátsszási listájukat, illetve megvárhatják, amíg minden személy megérkezik a játékba.

Miután valaki belépett a váró szobába láthatja a lobbynak az azonosítóját, hogy, ha szeretné, akkor át tudja küldeni azoknak, akikkel szeretne játszani, valamint még láthatók az eddig bejelentkezett felhasználók a jobb oldalon. A játékos a saját felhasználónevét a lejátsszási lista megadására szolgáló mező felett láthatja, így jól megkülönböztethetően külön van választva a többi felhasználótól, akiknek a nevük is más színnel vannak színezve.



6.2. ábra. Előszoba

6.3. Gamescene (Játéktér)

A játéktérben látható a számláló, a többi játékos, a jelenlegi zeneszám, illetve egy gomb, aminek megnyomásával folytatni lehet a játékot.

Saját magunkat zöld színnel látjuk, illetve alaphoz disabled opcióra van állítva a gomb, ami a minket jelez, hogy ne tudjunk magunkra szavazni, csak a többi játékosra, akik szürke színnel vannak jelölve.

Minden játékos neve mellett ott szerepel, hogy eddig hány ponttal rendelkezik. Ez a szám, akár negatív is lehet.

Amíg a kör tart, vagyis a számláló nem zéró, addig a gomb, amivel új kört lehet kezdeni ("New Round"), az disabledre van állítva, hogy ne legyen kattintható. A kör lejártá után lesz csak újra kattintható.

4

Alice 0

Bob 0

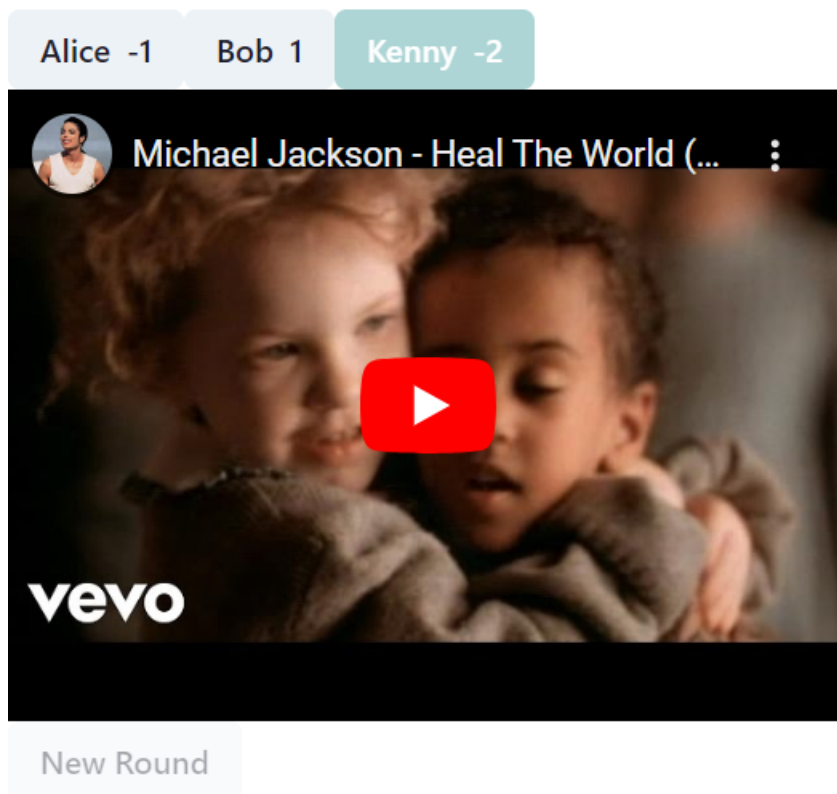
Kenny 0



New Round

6.3. ábra. Játéktér Első körben

8



6.4. ábra. Játéktér pár kör után

7. fejezet

Továbbfejlesztési Lehetőségek

A projekt továbbfejlesztési lehetőségei között szerepel:

7.1. Több API használata

Lehetőség lenne nem csak youtube lejátszási listákkal játszani, hanem lenne még opció spotify vagy soundcloud lejátszási listákat is berakni.

Ha lehetőség lenne több fajta lejátszási listával is játszani attól még nem feltétlen kellene megváltoztatni a frontendet, mivel például a legtöbb zene spotifyon és youtubeon is fent van és egy spotify lejátszási listából kilehetne szedni a zeneszámokat és automatikusan fel lehetne ajánlani a számoknak a youtube megfelelőjét.

Így a frontend is egységes lenne minden lejátszási lista esetében és nem csak spotify premium felhasználók tudnának spotify lejátszási listával játszani, mert a spotify lejátszóval csak akkor enged az API lejátszani egy számot vagy lejátszási listát, ha az adott lista tulajdonosa spotify premium tag.

7.2. Lejátszási listák helyben való létrehozása

Lehetne lejátszási listákat helyben is létrehozni, például a lobbyban vagy egy másik, erre dedikált oldalrészben.

Egy keresőből állna, ami a youtube API-n keresztül működik, rákeres arra, amit beírnak a keresőbe, megjeleníti az eredményeket és lehetőség lenne hozzáadni a kívánt zeneszámot.

7.3. Lejátszási listák módosítása

Az elküldött lejátszási listák módosíthatóak lennének, lehetne azokat bővíteni vagy törölni belőlük, illetve a módosított lejátszási listát le is lehetne menteni, ha a felhasználó be van jelentkezve.

7.4. Lejátszási lista hasonlóság

A játék végén megjelenítődne egy kimutatás, hogy kiknek voltak a leghasonlóbbak a lejátszási listái, vagyis kiknek volt a legközelebb a zenei ízlésük.

7.5. Google fiókkal való bejelentkezés

A felhasználó google fiókkal is bejelentkezhet, így már egyből elérhetőek lennének számára a youtubeos lejátszási listái, mivel ebben az esetben azokat le lehetne kérni a youtube API segítségével.

7.6. Email címmel való regisztráció

Lehetőség lenne saját email címmel regisztrálni, aminek az lenne az előnye a guest felhasználóval szemben, hogy az előzőleg elküldött lejátszási listákat lehetőség lenne elmenteni, így később nem kellene újra a lejátszási lista linket elküldeni, amikor játszani akarnánk, hanem egyből lehetne választani a régebb elküldött lejátszási listák közül egyet.

7.7. Tesztek hozzáadása

A tesztek hozzáadásával könnyebb lenne később több funkciót hozzáadni a projekthez, mivel jóval megkönnyítik a hibák elkerülését.

7.8. Nemmel való szavazás

A nemmel való szavazás lehetőségével nőne a játék komplexitása, illetve élvezhetősége, mivel több fajta szavazási lehetőség lenne, több féle képpen lehetne pontokat szerezni, illetve veszíteni.

7.9. Lejátszási lista ellenőrzése

Amikor lejátszási listát akarnánk kiválasztani, akkor lehetőség lenne megnézni, hogy milyen számok vannak benne, így kisebb lenne az esélye, hogy véletlenül rossz lejátszási listával játszunk.

7.10. 2 játékos mód

Lehetne ketten is játszani, lenne mindkét játékos lejátszási listájához hasonló generált lejátszási lista, ezek a lejátszási listák egy API segítségével generálódnának és azt kellene minden körben kitalálni, hogy az adott zeneszám a generált lejátszási listából van, vagy a másik játékos lejátszási listájából.

Összefoglaló

Dolgozatomban egy webes alkalmazás fejlesztésével foglalkoztam, amelyhez különböző népszerű programozási nyelveket és technológiákat használtam, mint például a React és az Express.js, munkám során sikerült beleásnom magam a websocketek világába, amik úgy gondolom elég hasznosak lesznek számomra a továbbiakban, mivel sok helyzetben hatékonyabbak tudnak lenni, mint a hagyományos HTTP kapcsolatra alapuló technológiák, valamint sikerült tovább gyarapítani az eddigi programozási ismereteimet is.

Repossitory linkek:

https://github.com/K-krisz/av_frontend

https://github.com/K-krisz/av_backend

Köszönetnyilvánítás

Szeretnék köszönetet mondani a vezetőtanáromnak, Dr. Iclanzan Davidnak a segítségéért, illetve köszönöm neki a türelmet és az ötleteket.

Ábrák jegyzéke

3.1. Use Case Diagram - Guest felhasználó	13
4.1. Trello Frontend	19
4.2. Trello Backend	20
4.3. Github Frontend	21
4.4. Github Backend	22
5.1. Szekvencia-Diagram	23
5.2. Typescript Interface	24
6.1. Kezdőoldal	27
6.2. Előszoba	28
6.3. Játéktér Első körben	29
6.4. Játéktér pár kör után	30

Irodalomjegyzék

- [exp] Express.js documentation. <https://expressjs.com/en/5x/api.html>. Hozzáférés dátuma: 2023. július 2.
- [Reaa] React learn. <https://react.dev/learn>. Hozzáférés dátuma: 2023. július 2.
- [reab] React router. <https://www.npmjs.com/package/react-router-dom>. Hozzáférés dátuma: 2023. július 2.
- [soc] Socket.io documentation. <https://socket.io/docs/v4/>. Hozzáférés dátuma: 2023. július 2.