

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Hanna Vallner**

**Veebirakendus Spotify esitusloendite  
korrastamiseks**

**Bakalaureusetöö (9 EAP)**

Juhendaja(d):

Jaan Janno

Tartu 2024

## **Veebirakendus Spotify esitusloendite korrastamiseks**

### **Lühikokkuvõte:**

Bakalaureusetöö eesmärk on luua veebirakendus, mis võimaldab kasutajal lootunnuste alusel oma Spotify esitusloendeid hallata. Veebirakenduse loomisel on kasutatud Spotify programmeerimisliidest, Angulari, Node.js käituskeskkonda, Firebase andmebaasi ning Renderi hostimisteenust. Rakendusel on kaks põhifunktsionaalsust: esitusloenditesse lugude lisamine ning nende vahel lugude sorteerimine. Mõlema toimingu juures pakub rakendus kasutajale soovitusi, põhinedes Spotify rakendusliideselt päritud loolutunnuste võrdlusel.

### **Võtmesõnad:**

Veebirakendus, Spotify, Angular, Node.js, Firebase

### **CERCS:**

P175 Informatics, systems theory

## **Web application for organizing Spotify playlists**

### **Abstract:**

The aim of this Bachelor's thesis is to create a web application which enables users to manage their Spotify playlists based on song features. The web application is made with the following technologies: Spotify Web API, Angular, Node.js, Firebase and Render. The application provides two main functionalities: adding songs to playlists and sorting songs between them. During both of the operations, the application gives out suggestions based on the comparison of song features requested from the Spotify Web API.

### **Keywords:**

Web application, Spotify, Angular, Node.js, Firebase

### **CERCS:**

P175 Informaatika, süsteemiteooria

## Sisukord

|                                                         |    |
|---------------------------------------------------------|----|
| Sissejuhatus.....                                       | 4  |
| 1. Teoreetiline taust.....                              | 5  |
| 1.1 Spotify.....                                        | 5  |
| 1.2 Sarnased veebirakendused.....                       | 6  |
| 1.2.1 Sort Your Music.....                              | 7  |
| 1.2.2 MagicPlaylist.....                                | 7  |
| 1.2.3 Skiley.....                                       | 8  |
| 1.3 Nõuded.....                                         | 9  |
| 1.3.1 Funktsionaalsed nõuded (FR).....                  | 9  |
| 1.3.2 Mittefunktsionaalsed nõuded (NFR).....            | 11 |
| 2. Kasutatud tehnoloogiad.....                          | 13 |
| 2.1 Spotify Web API.....                                | 13 |
| 2.1.1 Kasutatud funktsionaalsused.....                  | 14 |
| 2.1.2 Kasutatud tunnused.....                           | 14 |
| 2.2 Angular.....                                        | 15 |
| 2.3 Node.js.....                                        | 16 |
| 2.4 Firebase ja Cloud Firestore.....                    | 17 |
| 2.5 Render.....                                         | 17 |
| 3. Valminud veebirakendus.....                          | 19 |
| 3.1 Kujundus.....                                       | 19 |
| 3.2 Loo ja esitusloendi sobivuse leidmine.....          | 20 |
| 3.3 Rakenduse komponendid.....                          | 21 |
| 3.3.1 Sisselogimise leht.....                           | 22 |
| 3.3.2 Koduleht.....                                     | 23 |
| 3.3.3 Esitusloendite leht.....                          | 23 |
| 3.3.4 Esitusloendi korrastamise leht.....               | 24 |
| 3.3.5 Laulu otsingu leht.....                           | 27 |
| 3.3.6 Loo ja esitusloendite vahelise sobivuse leht..... | 28 |
| 3.3.7 Profiilileht.....                                 | 30 |
| 3.3.8 Tutvustusleht.....                                | 31 |
| 3.4 Võimalikud edasiarendused.....                      | 32 |
| Kokkuvõte.....                                          | 34 |
| Viidatud kirjandus.....                                 | 35 |
| Lisad.....                                              | 37 |

## Sissejuhatus

Muusikatööstuse arengus on alates 1990. aastatest suurt rolli mänginud infotehnoloogia. Infotehnoloogia üks märkimisväärsmaid panuseid muusikatööstusesse on muusika voogedastamine, mille leviala on aastate jooksul üha kasvanud. 2017. aastal ületas muusikatööstuses voogedastuse toodud tulu füüsiliste meediumite oma, ning tulude vahe on seniajani ainult suurenenud. [1]

Enimkasutatuid muusika voogedastusplatvorm on Spotify, millel on üle 615 miljoni kasutaja [2]. Spotify rakenduse üks võimalustest on esitusloendite loomine ja haldamine. Seejuures on Spotifyl mitmesuguseid soovitusi jagavaid lisafunktsionaalsuseid, mis võtavad üldjoontes pakkumiste aluseks lootunnused. Lootunnused on parameetrilised väärtused, mis kirjeldavad lugu kui heliteost. Spotifyl on ka avalik rakendusliides, millega võimaldatakse arendajatel selliste funktsionaalsuste loetelu täiendada.

Bakalaureusetöö eesmärk on luua veebirakendus, mis võimaldab kasutajal lootunnuste alusel oma Spotify esitusloendeid hallata. Veebirakendusel on kaks põhifunktsionaalsust, mis toetavad kasutajat vastavalt kahe tegevuse juures:

1. esitusloendite vahel lugude sorteerimine;
2. esitusloenditesse uute lugude lisamine.

Mõlemad toimingud saab täies mahus teostada loodud veebirakenduses, mille lisaväärtus tuleneb asjakohaste soovitude andmisest. Soovitused põhinevad esitusloendite ning lugude vahelistel sobivustel, kus kindla esitusloendi ja loo sobivuse määrab esitusloendi keskmiste tunnuste ja loo tunnuste sarnasus. Rakenduse soovitusel sõltumatult jääb lõplik otsus esitusloendite haldamisel kasutajale. Kõik kasutatavad metaandmed pärinevad Spotify rakendusliideselt.

Töö on jagatud kolme peatükki. Esimeses peatükis antakse ülevaade töö teoreetilisest taustast, mis hõlmab Spotify rakendust, loodud rakendustega sarnaste veebirakenduste analüüsi, ning loodud rakendusele püstitatud nõudeid. Teises peatükis käsitletakse veebirakenduse loomisel kasutatud tehnoloogiaid: Spotify Web API, Angular, Node.js, Firebase ja Render. Kolmandas peatükis kirjeldatakse valminud veebirakendust: selle kujundust, põhilist meetodit esitusloendi ja loo vahelise sobivuse leidmiseks, erinevaid vaateid ja nende funktsionaalsuseid ning võimalikke edasiarendusi.

## 1. Teoreetiline taust

Peatükis antakse ülevaade Spotify rakendusest ja selle soovitusi jagavast algoritmist, ning loodud veebirakendusega sarnastest rakendustest. Samuti on peatükis välja toodud veebirakendusele püstitatud nõuded, mis on eraldi alapeatükkidena jaotatud funktsionaalseteks ja mittefunktsionaalseteks nõueteks.

### 1.1 Spotify

Spotify<sup>1</sup> on digitaalne voogedastusplatvorm, mille kaudu saab kuulata muusikat, taskuhäälingusaateid ning muid audioformaadis teoseid [2]. Lisaks audiofailide kuulamisele võimaldab Spotify oma lemmikuid erinevatel viisidel talletada, seda näiteks esitusloendeid luues. Spotify loodi 2008. aastal, ning on 2024. aasta seisuga maailma enimkasutatuim voogedastusplatvorm [2].

Spotify üks tugevustest on personaliseeritud lähenemine kasutaja kuulamiskogemuse parendamiseks. Selle tarbeks loob Spotify igast kasutajast individuaalse maitseprofiili (ingl *taste profile*). [3].

Kasutaja maitseprofiili mõjutab eeskätt tema tegevus, näiteks see, mis lugusid ta otsib, kuulab ning esitusloenditesse lisab, ja milliseid hoopis vahele jätab [3]. Täpsemalt on seejuures määravaks lugude tunnused, näiteks tempo ja akustilisus, aga ka populaarsus ning esitaja nimi.

Maitseprofiili põhjal pakub Spotify kasutajale erinevatel viisidel soovitusi järgnevateks kuulamistegevusteks. Järgnevalt on kirjeldatud nelja Spotify funktsionaalsust, mis põhinevad kasutaja maitseprofiilile.

1. Nädala Avastused (ingl *Discover Weekly*) on Spotify loodud esitusloend, millesse kuuluvad lood on valitud igale kasutajale individuaalselt. Lugude valimisel arvestab Spotify algoritm eeskätt kasutaja maitseprofiiliga [4]. Nädala Avastuste esitusloendit uuendatakse iga nädal.
2. Esitusloendi Tark Juhujärjestus (ingl *Smart Shuffle*) on funktsionaalsus, mis täiendab esitusloendi juhuslikus järjestuses esitamise võimalust. Tark Juhujärjestus lisab esitusjärjekorda lugusid, mis esitusloendisse ei kuulu, ent mis

---

<sup>1</sup> Spotify veebirakendus: <https://open.spotify.com>.

oma tunnuste poolest esitusloendiga sobituvad. Seejuures arvestatakse ka kasutaja maitseprofiili [5].

3. Spotify Raadio (ingl *Spotify Radio*) on ajutine esitusloend, mille saab genereerida kindla loo, albumi, esitaja või esitusloendi põhjal. Algoritm valib esitusloendisse kuuluvad lood eeskätt valikuga seotud tunnuste põhjal, ent võtab arvesse ka kasutaja maitseprofiili [6].
4. Päeva Esitusloend (ingl *Daily Mix*) on iga päev uuenev esitusloend, mille lood on valitud kasutaja maitseprofiili põhjal. Erinevalt teistest nimetatud funktsionaalsustest kuuluvad Päeva Esitusloendisse ka sellised lood, mida kasutaja on varasemalt juba kuulanud. Päeva Esitusloendeid võib ühel kasutajal olla kuni kuus: nende arv sõltub sellest, kui mitmekülgne kasutaja muusikamaitse on, sest neid grupeeritakse just keskmiste tunnuste põhja [7].

Kasutaja maitseprofiil on ajas kasutaja tegevusega kohanduv [3]. Seega, kui muutub tema muusikamaitse, suudab rakenduse algoritm sellele reageerida ning sellega oma soovitustes arvestada. Kasutaja saab oma maitseprofiili ka aktiivselt kohandada, näiteks oma maitseprofiilist kinda esitusloendi eemaldamisel või Spotify soovitude tagasisidestamisel.

## **1.2 Sarnased veebirakendused**

Lootunnuste põhjal on loodud mitmeid klassifitseerivaid algoritme, ning nende põhjal rakendusi. Kuna Spotify on üks tuntuimadest muusikaplatvormidest, on paljud neist liidestatud Spotify rakendusega [8].

Kuigi lootunnuste alusel saab koostada klassifitseerimisalgoritme mitmetel erinevatel eesmärkidel, on enamik neist loodud lugude populaarsuse ennustamiseks [9]. Sellegipoolest leidub rakendusi, mille algoritm pakub teistsuguseid võimalusi. Järgnevalt on välja toodud kolm Spotify välisliidesega suhtlevat rakendust, mille tööpõhimõtted sarnanevad töö käigus loodud veebirakenduse omadega.

### 1.2.1 Sort Your Music

Sort Your Music<sup>2</sup> on veebirakendus, mis võimaldab kasutajal oma Spotify esitusloendi lugusid erinevate tunnuste põhjal ümber järjestada. Esitusloendi lugude sorteerimise võimalus on Spotify rakenduses olemas, kuid teistsuguste parameetrite alusel: loo kestvus, esitusloendisse lisamise aeg, loo pealkiri, artisti nimi ja albumi nimetus.

Sort Your Music veebirakenduse abil saab esitusloendi lugusid järjestada erinevate laulutunnuste, näiteks tempo ja energilisuse põhjal. Töös valminud veebirakendusega on Sort Your Music seetõttu sarnane, et võimaldab kasutajal Spotify Web APIlt päritud loo tunnuste põhjal oma esitusloendeid korrastada.

Lisaks lugude endi omadustele saab rakenduses esitusloendeid sorteerida ka lugude populaarsuse järgi, suvaliselt segatuna ning nii, et samade esitajate lood oleksid teineteisest võimalikult eraldatud.

Lisaks Sort Your Music veebirakendusele on veel palju teisigi rakendusi, mis samuti lugude tunnuste alusel Spotify esitusloendeid sorteerida võimaldavad. Nende hulka kuuluvad näiteks Sortlee<sup>3</sup>, Chosic<sup>4</sup> ja Emotionify<sup>5</sup>.

### 1.2.2 MagicPlaylist

MagicPlaylist<sup>6</sup> on veebirakendus, mille abil saab luua Spotify esitusloendeid, mis sobituvad ühe kindla loo või meeleoluga.

Meeleolude (ingl *mood*) alla kuuluvad rakenduses ka mõned muusikažanrid, nagu näiteks räpp, ent samuti on võimalik luua esitusloendeid töö või trenni taustaks, peomeeleolu tekitamiseks või rahuliku ajaveetmise (ingl *chill*) kõrvale kuulamiseks. Enamike meeleolude valiku järel tuleb kasutajal veel täpsustada, millisest žanrist või kitsendatult mis olustikule vastavaid lugusid ta oma esitusloendisse soovib, valides selleks vastava alamkategoria. Näiteks soovides luua esitusloendit trenni taustaks, kuuluvad muuhulgas järgnevate valikute sekka jooksmine, jooga ja jõutrenn.

---

<sup>2</sup> Sort Your Music veebirakendus: <http://sortyourmusic.playlistmachinery.com>.

<sup>3</sup> Sortlee veebirakendus: <https://www.sortspotifyplaylists.com>.

<sup>4</sup> Chosic veebirakendus: <https://www.chosic.com/spotify-playlist-sorter>.

<sup>5</sup> Emotionify veebirakendus: <https://emotionify.nitratine.net>.

<sup>6</sup> MagicPlaylist veebirakendus: <https://magicplaylist.co>.

Ühe kindla looga sobituva esitusloendi moodustamiseks tuleb kasutajal sisestada loo pealkiri kodulehel asuvasse otsinguvälja, ning valida soovitud lugu avanevast rippmenüüst.

Kui kindla meeleolu või looga sobituv esitusloend on loodud, saab kasutaja seda veel mitmeti kohandada. Algselt moodustatud 20 looga esitusloendist saab lugusid ümber järjestada, kustutada ja vahetada. Viimase puhul valib algoritm algsele loole ise asenduse. Lisaks sellele saab muuta ka kogu esitusloendit korraga, seadistades ümber soovitud esitusloendi keskmiseid tunnuseid. MagicPlaylisti kasutatavad tunnused on näiteks tantsulisus, instrumentaalsus, rõõmsameelsus ja populaarsus.

Spotify rakenduses endas on võimalik luua ühe looga sarnaste lugude esitusloend ning kuulata erinevatele meeleoludele vastavaid esitusloendeid, ent neid ei saa tunnuste järgi redigeerida. Seega teeb see omadus MagicPlaylist veebirakenduse pakutavad funktsionaalsused erinevaks Spotify omadest, ning ühtlasi sarnaseks töö käigus valminud veebirakendusega.

### **1.2.3 Skiley**

Skiley<sup>7</sup> on veebirakendus, mis annab kasutajale ülevaate tema kuulamisharjumustest Spotifys, ning võimaldab selle põhjal mitmeti uusi lugusid avastada.

Rakenduse pakutavad võimalused on menüüs jaotatud nelja rühma: kasutaja hetketegevus, statistika, esitusloendid ning avastamine.

Kui kasutaja kuulab rakendust kasutades Spotifys mõnda lugu, on see kuvatud kasutaja hetketegevuse lehel. Sealt leiab lisaks loo üldinfole selle laulusõnad, arvustused, muusikavideo ning kasutajate kommentaarid. Kuulatava loo põhjal on võimalik rakenduselt saada soovitusi teiste, sarnaste lugude kuulamiseks, ning nendest moodustada uus esitusloend.

Statistika alt leiab infot kasutaja enimkuulatud artistide, lugude ja žanrite kohta, seejuures on iga kategooria esindajad kuulamise sageduse põhjal järjestatud. Statistikat saab vaadata viimase paari tunni, 4 nädala, 6 kuu või 12 kuu kohta.

---

<sup>7</sup> Skiley veebirakendus: <https://skiley.net>.



Avastamise lehel on erinevaid võimalusi uute lugude avastamiseks. Näiteks saab tutvuda artistide vähemtuntud lugudega ning luua varasemalt kuulatud lugude põhjal esitusloendeid nendega sarnastest lugudest.

Esitusloendite lehel on esitatud kõik kasutaja esitusloendid, millest igapähega saab teha erinevaid toiminguid: luua sarnaseid, erinevate tunnuste põhjal filtreerida, poolitada ja ümber järjestada. Esitusloendit filtreerides või sarnast luues saab kasutaja valida, mis tunnuseid mis määral seejuures arvesse võtta.

Suures osas on Skiley võimalused sellised, mis on teistsugusel kujul olemas ka Spotifyl. Skiley funktsionaalsuste lisaväärtus tuleneb kasutajale valiku andmisest, mis tunnuseid ning millisel määral oma esitusloendite kogu haldamisel kasutada. Skiley sarnaneb oma võimalustega ka peatükkides 1.2.1 ja 1.2.2 kirjeldatud rakendustega. Skiley kasutamise eelis nende rakendusliidest ees on erinevate toimingute sooritamine ühe rakenduse siseselt, ent miinusena on tarvis selle samas mahus kasutamiseks maksta.

### **1.3 Nõuded**

Peatükis on välja toodud töö käigus loodud veebirakendusele püstitatud nõuded. Seejuures on funktsionaalsed ja mittefunktsionaalsed nõuded eraldi alapeatükkides.

#### **1.3.1 Funktsionaalsed nõuded (FR)**

Järgnevalt on esitatud veebirakenduse funktsionaalsed nõuded, mis on grupeeritud rakenduse erinevate vaadete põhjal. Esitusloendite muutmist käsitlevad nõuded laienevad toimingute tulemustena kasutaja Spotify esitusloenditele. Nõue FR1 tuleneb Spotify seatud eeskirjadest [10].

#### **Üldine**

FR1. Spotifylt päritud metaandmete juures on link Spotify rakendusele.

FR2. Kasutajale antakse teada, kui tema valitud toiming on töös.

FR3. Rakendus põhineb oma arvutuste tegemisel esitusloendi esimestel 50 laulul.

#### **Sisselogimise leht**

FR4. Kasutaja saab veebirakendusse sisse logida oma Spotify kontoandmetega.

## **Koduleht**

FR5. Kasutaja saab lühiülevaate rakenduse kahest põhifunktsionaalsusest.

FR6. Kasutaja saab rakenduse kahest põhifunktsionaalsusest lähtuvalt oma toimingu valida.

## **Esitusloendite leht**

FR7. Kasutaja näeb oma viimase 50 esitusloendi üldinfot.

FR8. Kasutaja saab toodud esitusloendite seast korrastamiseks ükskõik millise välja valida.

## **Esitusloendi korrastamise leht**

FR9. Kasutaja näeb selle esitusloendi üldinfot, mille ta esitusloendite lehel valis.

FR10. Kasutaja saab teada valitud esitusloendi keskmised tunnused.

FR11. Kasutajal on võimalus näha valitud esitusloendi esimese 50 loo üldinfot.

FR12. Kasutaja saab esitusloendist lugusid eemaldada.

FR13. Kasutaja saab kindla loo valimisel teada selle tunnused, sobivuse esitusloendiga ning selle, kas loole leidub kasutaja esitusloendite hulgast paremini sobivam.

FR14. Kui lugu sobiks paremini mõnda teise kasutaja esitusloendisse, saab kasutaja selle valimisel teada parima sobivusega esitusloendi nime, selle keskmised tunnused ja selle sobivuse valitud looga.

FR15. Kui lugu sobiks paremini mõnda teise kasutaja esitusloendisse, on kasutajal võimalus see parima sobivusega esitusloendisse ümber tõsta või senisesse alles jättes lisada.

FR16. Lugude juures on eritähis, kui sellele leidub kasutaja esitusloenditest valituga võrreldes palju sobivam esitusloend.

## **Loo otsingu leht**

FR17. Kasutajal on võimalik otsida lugusid kõigi Spotifys olemasolevate lugude seast.

FR18. Kasutaja saab otsinguvälja sisendi kirjutamisel soovitusi.

FR19. Kasutajal tuleb loo valimiseks see rakenduse soovitude seast valida.

### **Loo ja esitusloendi vahelise sobivuse leht**

FR20. Kasutaja näeb selle loo üldinfot, mille ta otsingu lehel valis.

FR21. Kasutaja saab teavet valitud loo tunnuste kohta.

FR22. Kasutaja näeb oma viimase 50 esitusloendi üldinfot.

FR23. Kasutaja saab valitud lugu piiramatult oma esitusloenditesse lisada.

FR24. Kasutaja saab kindla esitusloendi valimisel teada selle tunnused ning sobivuse valitud looga.

FR25. Kasutajal on võimalus luua uus esitusloend, mis sisaldab üksnes valitud lugu.

### **Profiilileht**

FR26. Kasutaja näeb oma Spotify konto põhilist infot.

FR27. Kasutaja saab rakendusest välja logida.

### **Tutvustusleht**

FR28. Kasutaja saab tutvuda veebirakenduse üldinfoga.

### **1.3.2 Mittefunktsionaalsed nõuded (NFR)**

Järgnevalt on välja toodud rakendusele seatud mittefunktsionaalsed nõuded. Nõuded NFR3-NFR7 tulenevad Spotify seatud eeskirjadest [10].

NFR1. Rakenduse veebilehekülg on interneti vahendusel kättesaadav kõigile.

NFR2. Rakenduse uuendamine GitHubis kajastub automaatselt rakenduse veebilehel.

NFR3. Spotify rakendusliideselt saadud metaandmed on kuvatud ühes Spotify logoga.

NFR4. Spotify rakendusliideselt päritud graafilised elemendid on moonutamata kujul.

NFR5. Iga Spotify rakendusliideselt päritud heliteose või heliteoste kogumiku puhul on kuvatud selle esitaja, albumi, loo ning esitusloendi nimi, kui vastavad andmed on olemas.

NFR6. Spotify logo ning sõnamärk on moonutamata kujul.

NFR7. Spotify värvipaleti roheline toon, värvikoodiga #1DB954, esineb vaid ühetoonilisel taustal.

## 2. Kasutatud tehnoloogiad

Peatükis antakse ülevaade veebirakenduse loomisel kasutatud tehnoloogiatest, millest igaüks on eraldi alampeatükis. Seejuures käsitletakse iga alampeatükis ka seda, mis kujul ja kuidas vastavat tehnoloogiat veebirakenduse loomisel kasutati.

### 2.1 Spotify Web API

Spotify Web API<sup>8</sup> on avaliku ligipääsuga rakendusliides, mis võimaldab kõigil huvilistel tutvuda Spotifysse üleslaetud teoste metaandmetega [11]. Samuti võimaldab Spotify Web API neid kasutaja soovile vastavalt kasutada. See pakub enim huvi arendajatele, kes saavad metaandmetele tuginedes Spotify rakendusliidest kasutades uusi rakendusi luua [11].

Spotify Web API kaudu saab muuhulgas teavet artistide, albumite, lugude ja esitusloendite kohta, ning neid nende erinevate tunnuste põhjal võrrelda. Võimalik on teostada ka erinevaid muutuseid, näiteks redigeerida lugude esitust ning esitusjärjestust ja hallata isiklikke esitusloendeid.

Spotifyga liidestatud rakendustel on kaks võimalikku kvoodirežiimi (ingl *quota mode*): arendusrežiim (ingl *development mode*) ning laiendatud kvoodiga režiim (ingl *extended quota mode*) [12]. Algselt on iga rakendus arendusrežiimis, kus tehtud päringute arvule on karmimad piirid. Samuti on rakenduse kasutajate arvule seatud piirarv, milleks on 25, kusjuures igaüks neist tuleb arendajal eelnevalt lisada lubatud kasutajate loendisse [12]. Töö käigus valminud veebirakendus oli töö avaldamise hetkel arendusrežiimis.

Liideste loomise hõlbustamiseks on Web API kodulehel olemas mitmeid juhiseid ning näiteid. Arendajatele on loodud oma foorum, kuhu saab murede korral pöörduda. Seal pakuvad nõu nii teised arendajad kui ka Spotify töötajad.

Seega on Spotifyga suhtleva veebirakenduse loomise protsess Spotify Web API kaudu muudetud võimalikult kasutajasõbralikuks. Ühtlasi tagab see ligipääsu kõikidele andmestikele ja funktsionaalsustele, mida on esitusloendite korrastamiseks tarvis: kõikide lugude tunnused, kasutaja loodud esitusloendid, nende loomine, muutmine ning kustutamine. Seetõttu on töö jooksul valminud veebirakenduse loomisel kasutatud just Spotify Web API.

### 2.1.1 Kasutatud funktsionaalsused

Spotify Web APIga suhtlemine võimaldab loodud veebirakendusel kasutaja esitusloendite kohta andmeid koguda, neid hallata, ning ühes teiste andmetega veebilehel kuvada. Järgnevalt on välja toodud kasutatud Spotify Web API funktsionaalsused, mis on esitatud neist saadud tulemusena, ning grupeeritud käsu tüübi põhjal.

*GET*-käskude abil saavutatud toimingud:

1. kasutaja autentimine;
2. kasutaja esitusloendite andmete kogumine;
3. esitusloendis sisalduvate lugude andmete kogumine;
4. lugude tunnused andmete kogumine;
5. lugude otsing;
6. kasutaja Spotify konto andmete kogumine.

*POST*-käskude abil saavutatud toimingud:

7. kasutaja autentimisjärgne ümbersuunamine;
8. loo lisamine esitusloendisse;
9. uue esitusloendi loomine.

*DELETE*-käskude abil saavutatud toimingud:

10. loo eemaldamine esitusloendist.

Rakenduses on erinevate funktsionaalsuste saavutamiseks Spotify Web API käske kombineeritud, näiteks saadetakse esitusloendist lugude küsimisel alati päring ka nende tunnuste saamiseks.

Kuna rakendus oli töö esitamise kuupäeval kirjeldatud arendusrežiimis, oli selle päringute saatmine Spotify Web APIle piiratud. Seetõttu oli laulude ja esitusloendite pärimisel jäänud ühe päringu piiresse, mis võimaldab teavet saada kuni 50 esitusloendi ja laulu kohta.

### 2.1.2 Kasutatud tunnused

Iga Spotify loo puhul on Web APIs üles tähendatud 18 kirjeldavat tunnust, mille põhjal saab lugusid omavahel võrrelda. Võrdlustest saab omakorda aimu, milliste tunnuste alusel on kasutaja oma esitusloendeid loonud, ning anda loendite kohandamiseks sobilikke soovitusi.

Iga loo 18st tunnusest on viis sellised, mis sisaldavad väärtustena linke, identifikaatoreid või muutumatuid vaikeväärtuseid, ning kaks sellised, mille puhul tuleks kasutada klassifitseerimist. Ülejäänud 11st tunnusest kasutab rakendus arvutuste tegemisel seitset, mis said määratud rakenduse testimise käigus. Mõneti võeti valikul eeskuju 1.2 peatükis kirjeldatud veebirakendustest. Kõikide tunnuste väärtused on ujukomaarvud, mis varieeruvad vahemikus nullist üheni. Järgnevalt on valitud tunnuseid ükshaaval lähemalt kirjeldatud, tuginedes nende kirjeldusele Spotify Web API lehel [13].

**Akustilisus** (ingl *acousticness*) on tõenäosus, et lugu on akustiline, ehk hinnang, kui kindlalt pole loo esitamisel kasutatud elektroonilisi muusikainstrumente. Mida lähemal tunnuse väärtus ühele on, seda tõenäolisemalt on tegemist akustilise muusikateosega.

**Tantsulisus** (ingl *danceability*) kirjeldab tempo, rütmikuse ja üldise muusikalise ühtluse põhjal, kui võrd sobiv on lugu tantsimiseks. Tunnuse väärtuse kasvades kasvab loo tantsulisus.

**Energia** (ingl *energy*) kirjeldab dünaamilise ulatuse, valjuse, tämbri, entroopia ning kiiruse põhjal loo tunnetuslikku energilisust. Mida energilisem on lugu, seda kõrgem on tunnust kirjeldav väärtus.

**Instrumentaalsus** (ingl *instrumentalness*) kajastab, kui suur osa loost ei sisalda laulmist, vaid üksnes muusikainstrumentide loodud helisid, kus väärtuse 1.0 puhul on tegemist vokaalmuusikaga.

**Otseesitus** (ingl *liveness*) näitab tõenäosust, et pala esitati otse publiku ees, live'is. Mõõdetakse tuvastatud publiku häälte või nende puudumise kaudu loo helifailis.

**Kõnelisus** (ingl *speechiness*) on tunnusena ujukomaarv, mis näitab, kui võrd palju esineb teoses sõnu, mis laulmise asemel pigem öeldakse. Mida kõrgem väärtus on, seda rohkem rääkimisena näivat heli teoses esineb.

**Valents** (ingl *valence*) kirjeldab heliteose rõõmsameelsust. Madalamad arvud tähistavad kurvemaid või muu negatiivsema emotsiooniga lugusid, kõrgemad arvud positiivsema emotsiooniga lugusid.

## 2.2 Angular

Angular<sup>8</sup> on TypeScripti põhine arendusplatvorm, mis toetab veebirakenduste arendamist [14]. Loodud veebirakenduse kliendipoolne süsteem on kirjutatud kasutades Angulari komponente (ingl *components*). Angulari komponendid on alaüksused, mis sätestavad eraldi iga rakenduse lehekülje väljanägemise ning funktsionaalsused. Iga komponent koosneb loodud rakenduse puhul kolmest failist: TypeScript, mis sätestab lehe funktsionaalsused; HTML, mis määrab lehel kuvatava; SCSS, mis paneb paika lehe disaini.

Lisaks komponentidele kasutab loodud veebirakendus Angulari teenuseid (ingl *services*). Teenustes defineeritud meetodeid on lihtsam erinevate komponentide vahel jagada, seega on teenused kasutusel üleliigselt korduva koodi vältimiseks ning arendamise mugavdamiseks [14]. Loodud veebirakenduse puhul sisaldavad teenused meetodeid, mis suhtlevad tagaliidese vahendusel Spotify Web APIga.

Veebilehe eesliides on loodud kasutades just Angulari, sest see toetab struktureeritud lähenemist veebirakenduse loomisele, ning pakub suure jõudlusega eduka ettevõtte tulevikukindlust, sealhulgas rakenduse võimalikule edasiarendamisele. Samuti on paljud 2. peatükis käsitletud sarnased veebirakendused loodud samuti Angulari abil, mis pakkus kinnitust, et taolise rakenduse loomisel on Angular sobilik valik. Rakenduse loomisel on kasutatud Angulari versiooni 17.3, sest see oli arendamise algushetkel uusim väljalase.

## 2.3 Node.js

Node.js<sup>9</sup> on mitmeplatvormiline JavaScripti käituskeskkond, mis toetab arendajaid veebirakenduste ja -serverite loomisel [15]. Veebirakenduste puhul kasutatakse seda üldiselt tagaliidese loomisel, kus toimub suurem osa andmete töötlemisest, mahukamate arvutustest ning suhtlusest erinevate rakendust ülesehitavate osade vahel.

---

<sup>8</sup> Angulari kodulehekül: <https://angular.io>.

<sup>9</sup> Node.js käituskeskkonna kodulehekül: <https://nodejs.org/en>.

Loodud veebirakenduse puhul on Node.js keskkonda kasutades loodud server, mis vahendab eesliidese suhtlust Spotify Web APIga, töötleb ning varustab saadud andmeid. Samuti teostab server suuremaid arvutusi: esitusloendi ja loo omavahelise sobivuse leidmine, väärtuste normaliseerimine, esitusloendi tunnuste keskmiste leidmine, loo jaoks sobivaima esitusloendi leidmine.

Veebirakenduse tagaliides on loodud Node.js käituskeskkonna abil, sest see võimaldab sündmustepõhist asünkroonset lähenemist meetoditele, mis on rakenduse optimaalseks toimimiseks vajalik [16]. Rakenduse loomisel on kasutatud Node.js versiooni 20.11.1, sest see oli veebirakenduse arendamise algushetkel uusim väljalase.

## 2.4 Firebase ja Cloud Firestore

Firebase<sup>10</sup> arenduskeskkond on Google'i tütarettevõtte, mis pakub arendajatele mitmeid erinevaid teenuseid rakenduste loomiseks. Üks põhilisematest pakutavatest teenustest on Cloud Firestore, mida on kasutatud ka loodud veebirakenduse arendamisel.

Cloud Firestore on pilvepõhine NoSQL andmebaas, mis läheneb andmete talletamisele ning haldamisele dokumendipõhiselt. Kõiki andmeid varustatakse dokumentides, mis sisaldavad kirjetena võtmeid ning nende väärtuseid, seejuures võivad väärtused varieeruda sõnedest keerulisemate andmestruktuurideni. Dokumendid on omakorda organiseeritud kogumikesse (ingl *collections*). [17]

Andmebaasi dokumendid sarnanevad ülesehituselt JSON-tüüpi failidega, ning neid saab üldjoontes samamoodi käsitleda [18]. Kuna Spotify Web API tagastab päritud metaandmeid just JSON formaadis, on selline ülesehitus ühtivuses loodud rakenduse üldise töövooga ning seega lihtsalt integreeritav.

Cloud Firestore on ühilduv mitmete erinevate programmeerimiskeelte ja käituskeskkondadega, sealhulgas Node.js keskkonnaga, milles on kirjutatud loodud veebirakenduse tagaliides [17]. Seega on andmebaas sätestatud töötama rakenduse tagaliideses, talletades sealt vajalikke andmeid. Veebirakenduse loomisel on kasutatud Cloud Firestore versiooni 4.6.1, sest see oli rakendusele andmebaasi loomise hetkel uusim väljalase.

---

<sup>10</sup> Firebase keskkonna koduleht: <https://firebase.google.com>.



## 2.5 Render

Render<sup>11</sup> on pilvepõhine platvorm, mis lihtsustab veebirakenduste juurutamist [19]. Render pakub arendajatele mitmesuguseid sellealaseid teenuseid, mille hulka kuuluvad näiteks andmebaasid ja erinevad veebiteenused.

Peamiselt kasutatakse Renderit veebirakenduste domeenile üles seadmiseks ehk hostimiseks, seda ka töö käigus valminud rakenduse puhul. Platvormil on veebirakendusele ligipääs läbi selle GitHubi hoidla (ingl *repository*). Vaikimisi uuendab Render hostitud veebiteenuseid automaatselt iga GitHubi lükatud värskenduse puhul.

Hetkeseisuga on rakenduse veebiteenus Renderis tasuta versiooni peal, mistõttu on selle võimekusel omad piirid. Näiteks läheb veebiteenus 15-minutilise ebaaktiivsuse korral seisundisse, kus uuesti käitamisel kulub laadimiseks kuni minut [20].

---

<sup>11</sup> Renderi koduleht: <https://render.com>.

### 3. Valminud veebirakendus

Töö käigus valminud veebirakenduse nimeks sai Choordify<sup>12</sup>. Veebirakenduse nimetus tuleneb ingliskeelsetest sõnadest *chord* ja *coordinate*, mis tähendavad vastavalt muusikalist kooskõla ja kooskõlastamist. Järelliide *-ify* on lisatud viitena Spotify rakendusele.

Loodud veebirakenduse juures on peetud tähtsaks anda ülevaade parasjagu käsitletavate lugude ja esitusloendite tunnustest. See loob kasutajale parem arusaama sellest, millel põhinedes on erinevad klassifitseerimisotsused tehtud. Sellele tuginedes saab kasutaja tunnuseid uurides ise vabamalt valida, milliseid tunnuseid ning mis määral ta oma esitusloendeid hallates arvestada soovib. Peatükis 1.2 kirjeldatud veebirakendustest andsid kasutajale lootunnustest teada vaid selle esimeses alapeatükis käsitletud rakendused. Kõikide teiste Spotifyga liidestatud rakenduste, ning ka Spotify enda funktsionaalsuste puhul jäävad soovitude tegemiseks kasutatud tunnused kasutajale varjatuks.

Järgnevates alapeatükkides on valminud veebirakendust lähemalt kirjeldatud. Käsitletule alla tulevad veebirakenduse kujundus, põhifunktsionaalsuseid toetava loo ja esitusloendi sobivuse arvutamise meetod ning rakenduse võimalikud edasiarendused. Samuti tuuakse välja veebirakenduse kõik leheküljed, nende põhilised võimalused ning võimaluste meetoodika.

#### 3.1 Kujundus

Rakenduse kujundamisel on eeskätt eeskuju võetud Spotify enda disainist, seda peamiselt kasutajale tuttavliku keskkonna loomiseks. Üldiselt on Spotify kujundust kasutatud pigem stiilinäitena, seda näiteks värvipaleti, kirjastiili ja veebilehe elementide paigutuse ja metaandmete kohta kuvatud sisu suhtes.

Spotify rakenduse disainist eeskuju võttes on järgitud Spotify Web APIga suhtlevatele rakendustele seatud eeskirju, mis kehtivad laiendatud kvoodiga režiimis rakendustele. Loodud veebirakendus nende hulka töö avaldamise kuupäeval ei kuulunud, ent hea tava põhimõttel ning edasiarenduste lihtsustamiseks järgiti eeskirju sellegipoolest. Spotify

---

<sup>12</sup> Choordify lingid:

- Rakenduse veebiteenus Renderis: <https://choordify.onrender.com>
- Rakenduse lähtekood GitHubis: <https://github.com/HannaVallner/Choordify>

seatud eeskirjad on püstitatud rakenduse nõuetena FR1 ja NFR3-NFR7, mis on vastavalt esitatud peatükkides 1.3.1 ja 1.3.2.

Üldiselt on rakenduse disaini hoitud lihtsa ja modernsena. Näiteks on välditud tervet lehte hõlmavaid kerimisribasid, selle asemel on kerimise võimalus lisatud eraldiseisvatele elementidele, kus selleks vajadust on. Sama kujunduse taktikat kasutab ka Spotify.

### 3.2 Loo ja esitusloendi sobivuse leidmine

Loodud rakenduse kahe põhifunktsionaalsuse peamine ülesanne on ühine: loo ja esitusloendi vahelise sobivuse leidmine. Selle jaoks kasutatakse loo tunnuseid ning esitusloendi keskmiseid tunnuseid.

Esitusloendi keskmised tunnused arvutatakse selles olevate lugude põhjal: kõigi lugude tunnused liidetakse kokku ning jagatakse lugude arvuga, ehk rakendatakse aritmeetilise keskmise valemit.

Loo ja esitusloendi vahelise sobivuse arvutamisel kasutatakse Eukleidilise kauguse teoreemi. Järgnevalt on valem esitatud sellisel kujul, nagu seda veebirakendus kasutab.

$$\text{sobivus}(l, p) = \sqrt{(l_a - p_a)^2 + (l_t - p_t)^2 + (l_e - p_e)^2 + (l_i - p_i)^2 + (l_o - p_o)^2 + (l_k - p_k)^2 + (l_v - p_v)^2}$$

Joonis 1. Sobivuse arvutamine Eukleidilise kauguse valemiga.

Joonisel toodud valemil tähistab  $l$  lugu ning  $p$  esitusloendit. Alaindeksitega väärtused esindavad vastavalt loo või esitusloendi kindlaid tunnuseid, kus:

- $a$  on akustilisus;
- $t$  on tantsulisus;
- $e$  on energia;
- $i$  on instrumentaalsus;
- $o$  on otseesitus;
- $k$  on kõnelisus;
- $v$  on valents.

Kõiki tunnuseid on põhjalikumalt kirjeldatud peatükis 2.1.2.

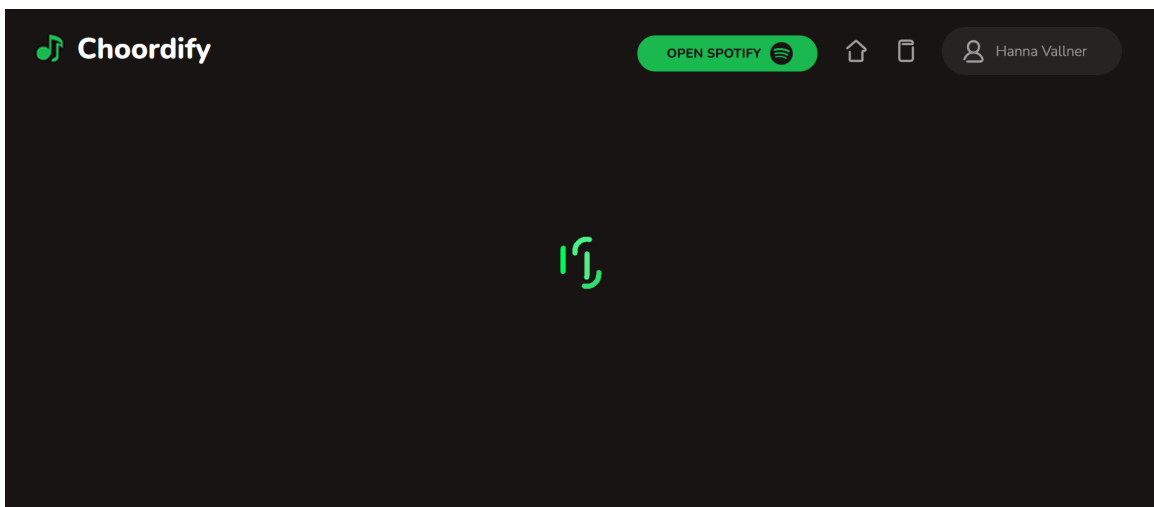
Sobivuse arvutamine viiakse läbi rakenduse tagaliidese meetodis *calculateCompatibility*, kus seda läbi teiste funktsioonide välja kutsutakse.

### 3.3 Rakenduse komponendid

Peatükis kirjeldatakse valminud veebirakenduse kõiki lehekülgi, nende funktsionaalsuseid ning toimimise metoodikat. Kõik leheküljed on loodud eraldi Angulari komponentidena, ehk igal lehel on oma HTML, SCSS ja TypeScript fail.

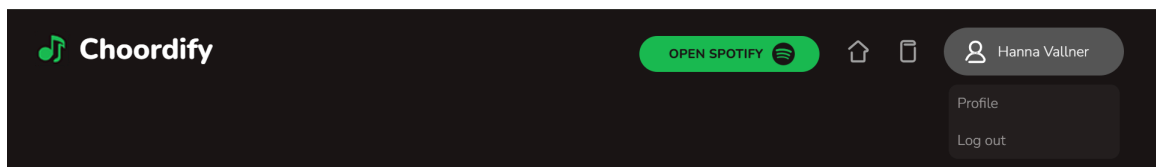
Rakendusel on lisaks eraldiseisvatele lehekülgedele veel kaks komponenti: navigeerimisriba ning laadimisaken. Kokku on erinevaid komponente 11, ent eraldi lehtedena on seega neist üheksa.

Laadimisaken on komponendina kõigi lehtede küljes, mis kasutavad kasutaja konto või esitusloendite andmeid. Seega on laadimisaken profiililehel, esitusloendite lehel, esitusloendi korrastamise lehel ning loo ja esitusloendite sobivuse lehel. Laadimisaken kujutab endast laadimisanimatsiooni, mis katab lehe teisi elemente senikaua, kuni vajalikud andmed on serverist kätte saadud ning lehel kajastamiseks valmis.



Joonis 2. Laadimisaken.

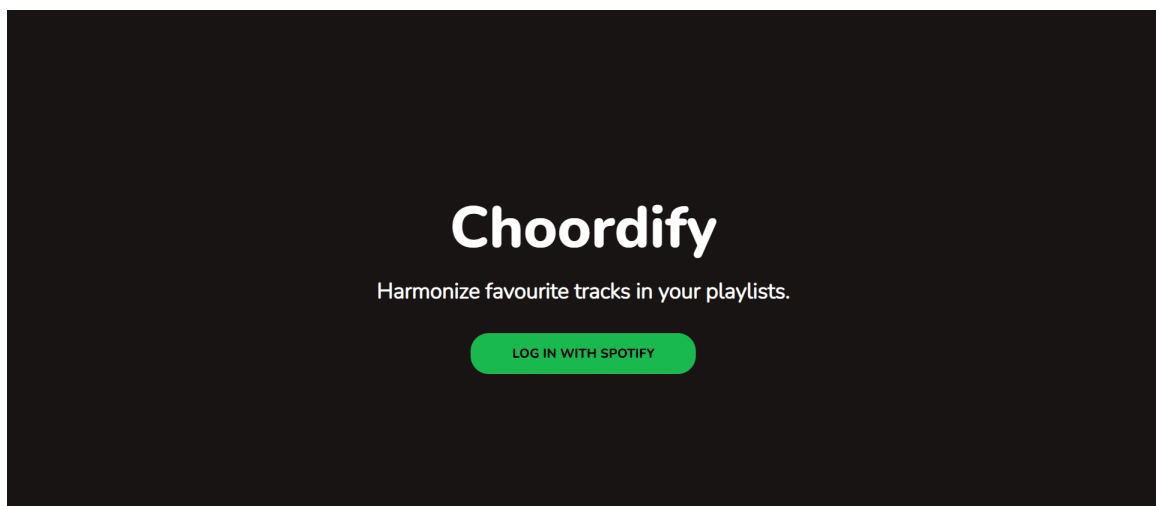
Navigeerimisriba asetseb rakenduse kõikide lehekülgede päises, välja arvatud sisselogimise lehel, ning võimaldab kasutajal mugavalt navigeerida kodu-, profiili- ning tutvustuslehele. Samuti on navigeerimisribal nupp, millele klõpsamisel avaneb Spotify veebirakendus. Kasutaja suunatakse rakenduse kodulehele, kui ta vajutab rakenduse logo või maja kujutisega ikooni peale. Tutvustuslehele pääseb märkmikuga ikoonile klõpsates. Kui vajutada kasutajanimega nupule, avaneb rippmenüü, mille kaudu saab navigeerida profiililehele või rakendusest välja logida.



Joonis 3. Navigeerimisriba avatud menüüga.

### 3.3.1 Sisselogimise leht

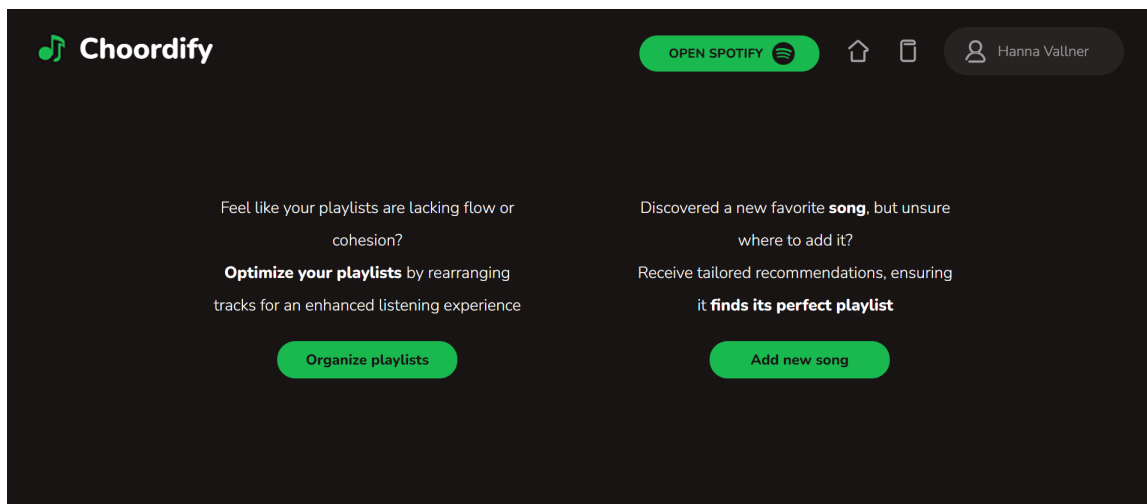
Sisselogimise leht on ühtlasi veebirakenduse maandumisleht, kus ainus võimalik toiming on sisselogimine. Sisselogimise nupule klõpsates suunatakse kasutaja ümber Spotify autentimislehele, kus küsitakse muuhulgas kasutaja nõusolekut juurdepääsuks tema kontoandmetele ja loodud teekidele.



Joonis 4. Sisselogimise leht.

### 3.3.2 Koduleht

Peale autentimist suunatakse kasutaja ümber rakenduse kodulehele, kus saab valida veebirakenduse kahe põhifunktsionaalsuse vahel: esitusloendite vahel lugude sorteerimine, ning uue loo lisamine oma esitusloendite kogumikku. Mõlema jaoks on kodulehel oma kirjeldus ning nupp, mis suunab kasutaja edasi valitud võimalustega lehele.

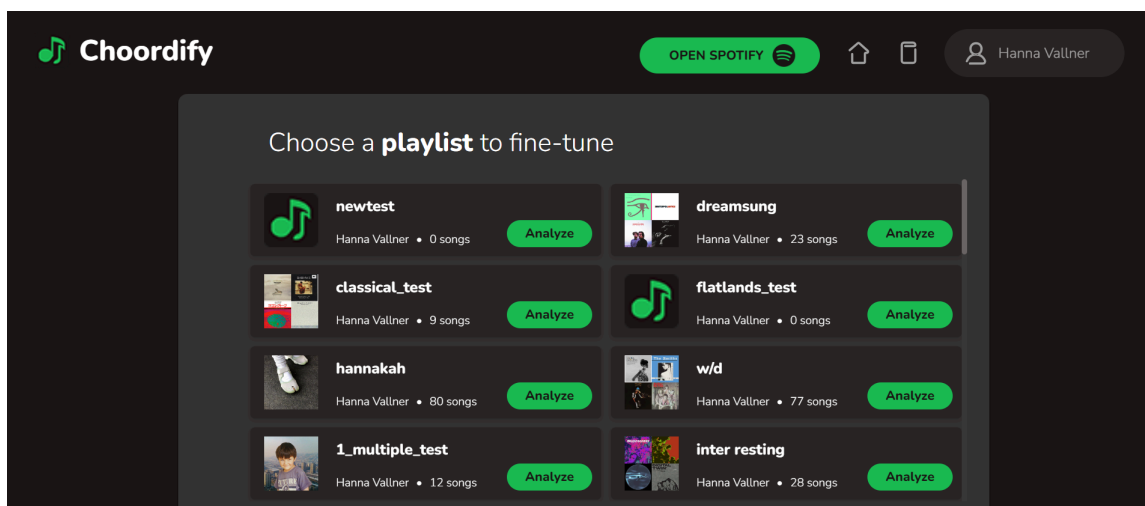


Joonis 5. Koduleht.

Kodulehele jõudes saadab server Spotify Web APIle päringuid mitmete andmebaasis talletavate muutujate algväärtustamiseks. Nende hulka kuuluvad kasutaja esitusloendid, neis leiduvad lood ja lugude tunnused, ning kontoandmed. Iga esitusloendi puhul arvutatakse välja selle lugude tunnuste keskmised, ning seondatakse ühes lugude endiga esitusloendi külge.

### 3.3.3 Esitusloendite leht

Esitusloendite lehele saab liikuda rakenduse kodulehelt, tehes valik olemasolevate esitusloendite korrastamiseks.



Joonis 6. Esitusloendite leht.

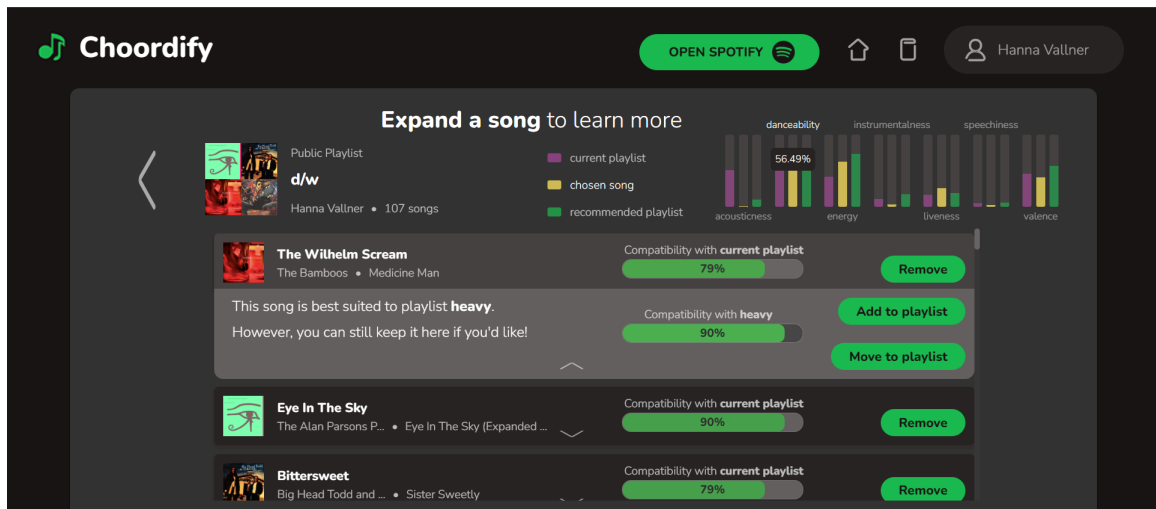
Esitusloendite lehel on keritav loendiboks, mis sisaldab eraldi elementidena kõiki kasutaja esitusloendeid. Iga esitusloendi puhul on näha selle kaanepilt, nimetus, looja kasutajanimi ning sisalduvate lugude arv. Samuti on kõikidel esitusloenditel eraldi nupp, millele klõpsates saab kasutaja esitusloendit ning selles sisalduvaid lugusid lähemalt analüüsida, ning seeläbi selle haldamiseks otsuseid teha. Nupule klõpsamise järel suunatakse kasutaja esitusloendi korrastamise lehele.

### 3.3.4 Esitusloendi korrastamise leht

Kasutaja suunatakse ümber esitusloendi korrastamise lehele, kui ta on esitusloendite lehel korrastamiseks kindla esitusloendi valinud. Lehelt on võimalik esitusloendite vaatele ka tagasi liikuda, klõpsates vasakul ääres olevale noolekesele.

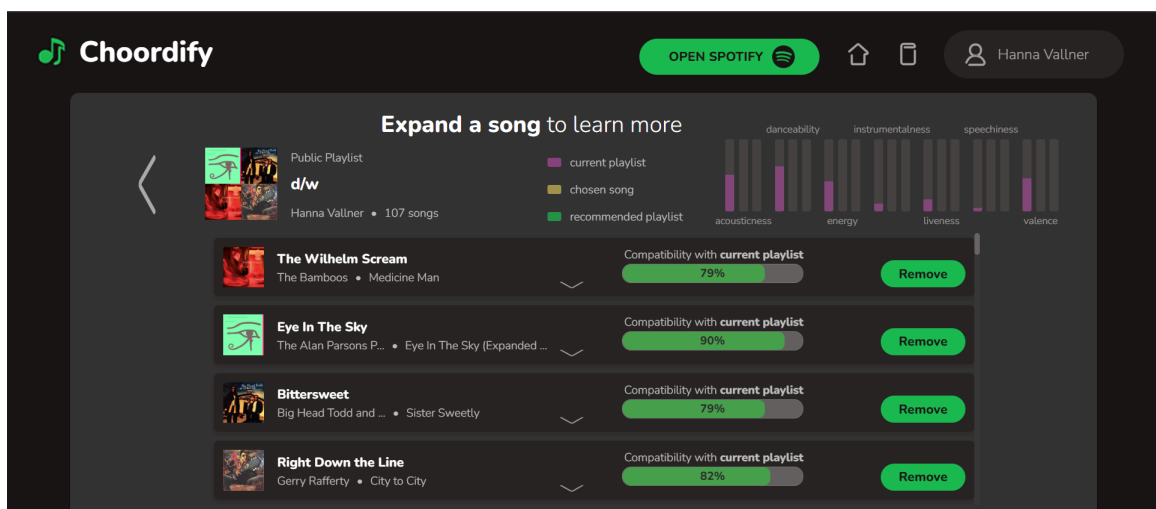
Esitusloendi korrastamise lehe ülaosa kuvab kasutaja poolt valitud esitusloendi üldinfot: kaanepilt, nimetus, lugude arv ning looja nimi. Samuti on lehe ülaosas tulpdiagramm, mis kajastab esitusloendi, loo ja soovitatud esitusloendi keskmiseid tunnuseid, kus on vaikumisi kuvatud vaid esitusloendi tunnused. Tulpdiagrammi juures on ka selle legend. Kui kasutaja kursor on kindla tulba või legendi elemendi peal, muutuvad kõik sama värvitähisega tulbad ning nende legend teistest erksamaks. Kui kursor on kindla tulba

peal, muutub erksamaks tulpa kirjeldava tunnuse nimetus, ning tööriistavihjena ilmub tunnuse protsentuaalne väärtus.



Joonis 7. Esitusloendi korrastamise lehe tulpdigrammi näide.

Selle all on keritava loendina toodud esitusloendi lood, mida on lehel kuni 50. Igal lool on Spotifylt saadud metaandmetest kaanepilt, nimi, esitaja nimi ning albumi nimi. Lisaks on ribadiagrammina välja toodud loo protsentuaalne sobivus esitusloendiga. Sobivuse arvutamist on lähemalt kirjeldatud peatükis 3.2.

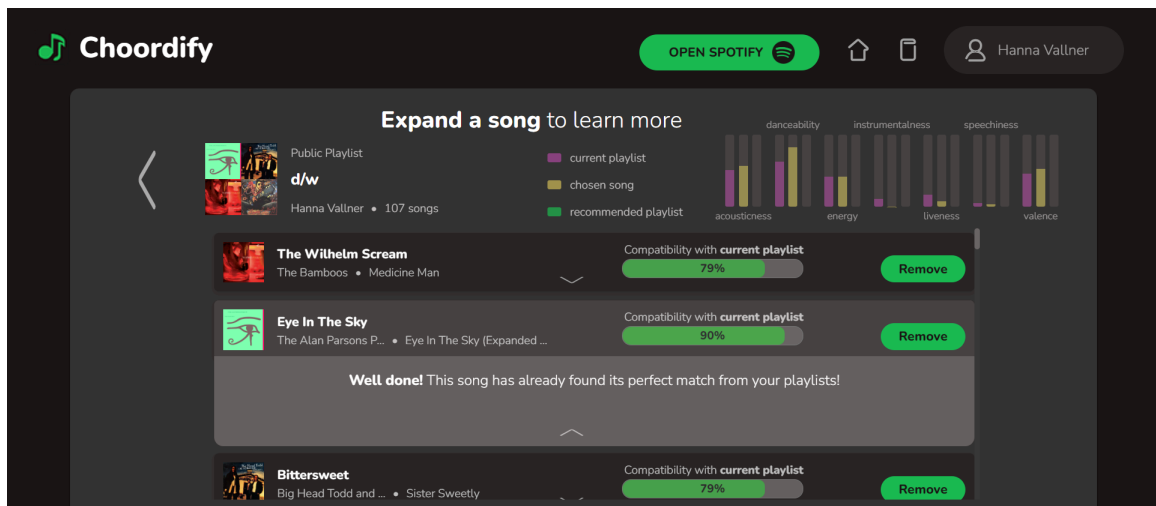


Joonis 8. Esitusloendi korrastamise leht.

Iga esitusloendisse kuuluva loo jaluses on nooleke, millele klõpsamisel laieneb loo infot kajastav loendielement. Samal ajal ilmuvad tulpdigrammil nähtavale valitud loo tunnused. Kui valitud lugu sobiks paremini mõnda teise kasutaja esitusloendisse, ilmuvad tulpdigrammil nähtavale ka vastava esitusloendi tunnused. Laiendatud seisundis saab

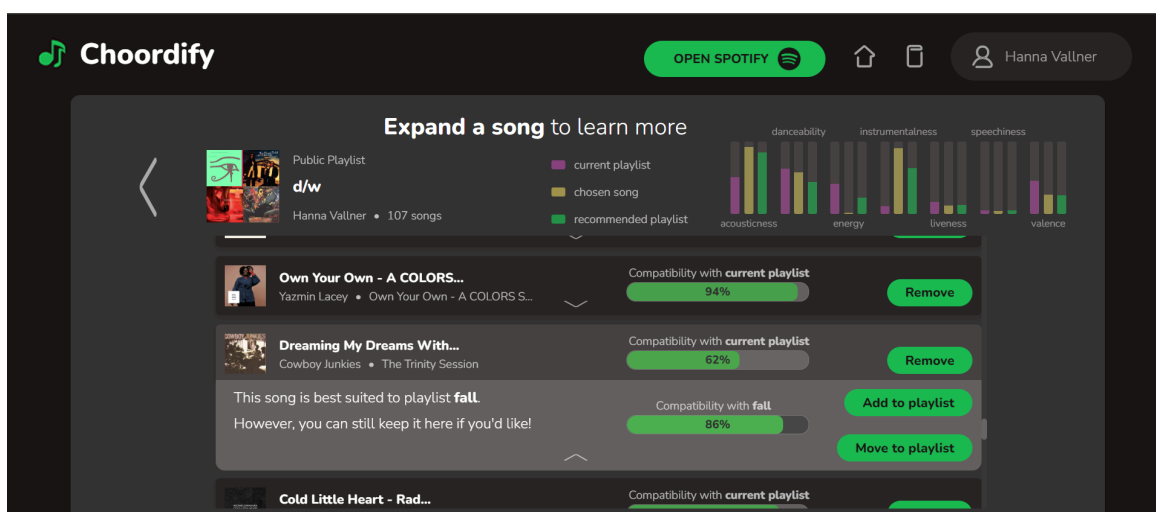


korraga olla vaid üks lugu: ühe avamisel sulgub teine. Laiendatud osas on vastava loo lisainfo, mille täpsem sisu sõltub loo sobivusest selle esitusloendiga, millesse see parasjagu kuulub. Kui esitusloend on loo jaoks kasutaja kõikidest esitusloenditest sobivaim, on seal vaid sellekohane tekst.



Joonis 9. Esitusloendi korrastamise leht parima sobivusega looga.

Kui lugu on oma tunnuste poolest sarnasem mõne teise esitusloendiga, on laiendatud osas välja toodud sobivaima esitusloendi nimi, ning ribadiagrammina ka selle protsentuaalne sobivus. Samuti on seal võimalus lugu esitusloendisse lisada, seda nii algses esitusloendis säilitades kui ka sealt eemaldades. Kui vaadeldava esitusloendi ning sobivaima esitusloendi sobivuse vahe on enam kui 20 protsenti, on loendielementi laiendava noolekese juures eraldi hoiatus.

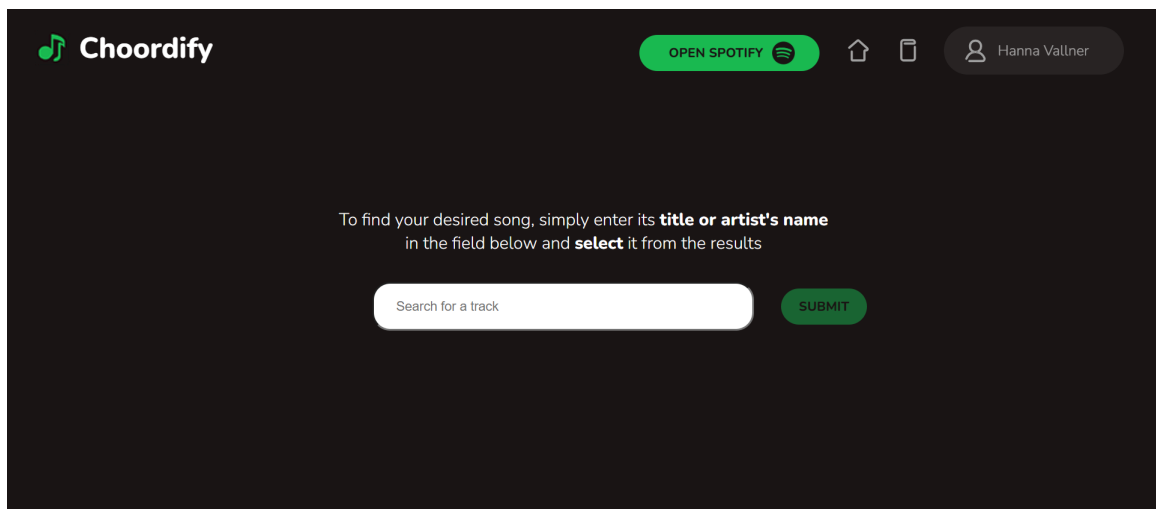


Joonis 10. Esitusloendi korrastamise leht mujale paremini sobiva looga.

Kõiki lugusid saab esitusloendiga sobivusest sõltumatult esitusloendist eemaldada. Sel juhul lugu kuhugi mujale ei lisata. Loo esitusloendist eemaldamise, teise esitusloendi lisamise ning üleviimise puhul ilmub vastava nupu asemele toimingut kirjeldav tekst, mis säilib seal kuni toiming on lõpule viidud. Loo esitusloendist eemaldamisel ning üleviimisel tuleb uuendatud tunnuste ja sobivuste nägemiseks lehte uuendada.

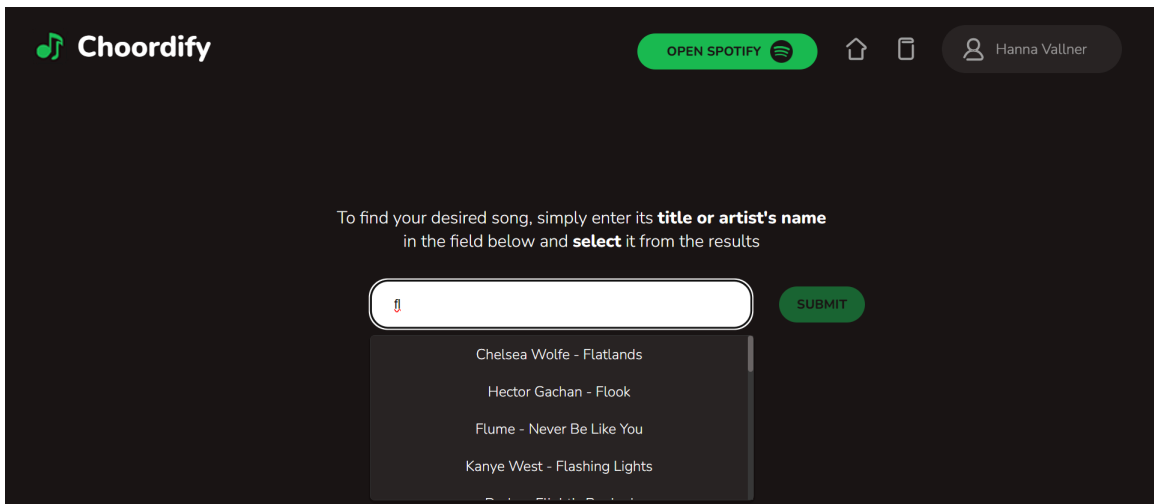
### 3.3.5 Laulu otsingu leht

Laulu otsingu lehele suunatakse kasutaja siis, kui ta teeb kodulehel valiku uue loo lisamiseks oma esitusloendite kogumikku.



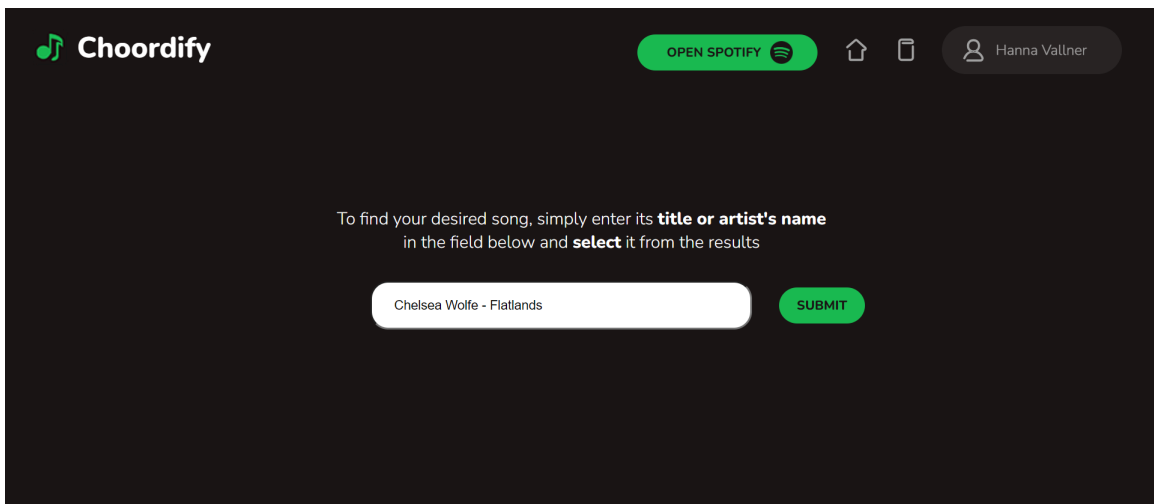
Joonis 11. Laulu otsingu leht.

Lehe keskel on otsinguväli, mis reageerib reaajas kasutaja sisendile. Sisendi põhjal soovitatakse avanevas rippmenüüs lugusid, mille nimi või mille esitaja nimi ühtib sisendiga. Soovituste tegemisel arvestatakse lisaks sisendile kasutaja maitseprofili: tõenäolisemalt soovitatakse lugusid või artiste, mida kasutaja on varem kuulanud. Ühe kindla sisendi kohta antakse vaikimisi 20 soovitusi, mille hulgast filtreeritakse välja duplikaadid.



Joonis 12. Laulu otsingu leht rippmenüüga.

Valikut kinnitavale nupule vajutamiseks tuleb eelnevalt rippmenüüst soovitud lugu valida. Laulu valimisel täitub otsinguväli rippmenüü valikus olnud tekstiga, ning valikut kinnitav nupp muutub klõpsatavaks. Nupu klõpsamisel suunatakse kasutaja edasi loo ja esitusloendite vahelise sobivuse lehele.

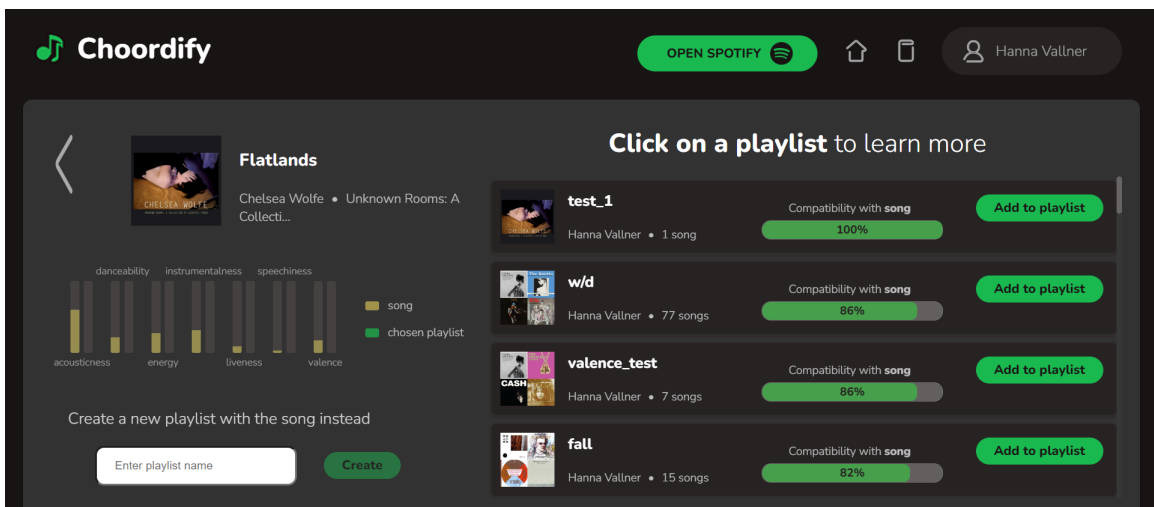


Joonis 13. Laulu otsingu leht valitud looga.

### 3.3.6 Loo ja esitusloendite vahelise sobivuse leht

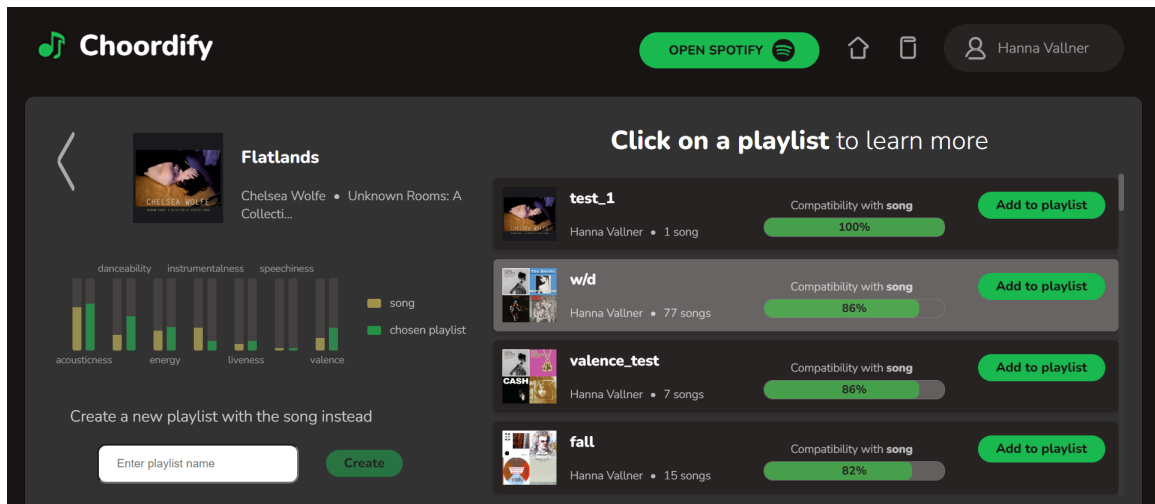
Loo ja esitusloendite vahelise sobivuse lehele suunatakse kasutaja siis, kui ta on loo otsingu lehel soovitud loo välja valinud. Lehelt on võimalik loo otsingu lehele ka tagasi liikuda, klõpsates vasakul ääres olevale noolekesele.

Lehe üleval vasakus ääres on kuvatud valitud loo üldinfo: nimi, loo esitaja ja albumi nimi, ning albumi kaanefoto. Selle all on tulpdiagrammina esitatud loo ja esitusloendi tunnused. Tulpdiagrammi juures on ka selle legend. Vaikimisi on kuvatud vaid loo tunnused. Kui kasutaja kursor on kindla tulba või legendi elemendi peal, muutuvad kõik sama värvitähisega tulbad ning nende legend teistest erksamaks. Kui kursor on kindla tulba peal, muutub erksamaks tulpa kirjeldava tunnuse nimetus, ning tööriistavihjena ilmub tunnuse protsentuaalne väärtus.



Joonis 14. Laulu ja esitusloendite vahelise sobivuse leht.

Lehe paremal pool on kerimisribaga loendis kuvatud kasutaja esitusloendid, millest igaühe puhul on toodud esitusloendi nimi, looja kasutajanimi, lugude arv ning sobivus valitud looga. Sobivuse arvutamist on kirjeldatud peatükis 3.2. Sobivus on kajastatud ribadiagrammina, mille peal on see lisaks protsentuaalsel kujul kirjas. Esitusloendid on loendis sobivuse alusel järjestatud, et kasutajale paremini sobivaimaid esitusloendeid soovitada. Kindlale esitusloendile klõpsates ilmuvad tulpdiagrammil nähtavale selle tunnused.



Joonis 15. Loo ja esitusloendite vahelise sobivuse leht valitud esitusloendiga.

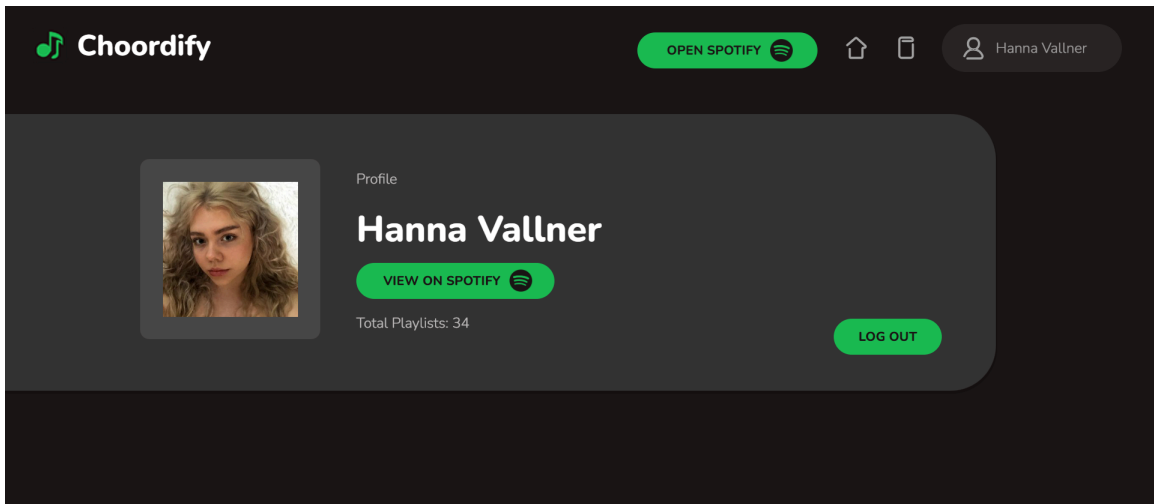
Iga esitusloendi juures on nupp valitud loo esitusloendisse lisamiseks, seega ei pea kasutaja tingimata valima esitusloendit, mis looga kõige paremini sobib. Samuti ei piira rakendus loo lisamist mitmesse esitusloendisse, ega ka ühte esitusloendisse korduvalt. Kui kasutaja klikab loo esitusloendisse lisamise nupul, ilmub nupu asemele tekst, mis teavitab kasutajat, et toiming on sooritamisel. Tekst asendab nuppu senikaua, kuni lugu on esitusloendisse lisatud ning esitusloend on loendis uuendatud. Seejuures uueneb muudatuste põhjal ka valitud esitusloendi sobivus looga.

Lisaks valitud loo lisamisele olemasolevasse esitusloendisse on kasutajal võimalus luua looga hoopis uus esitusloend. Selle jaoks tuleb lehe vasakul all ääres asuvasse tekstivälja kirjutada uue esitusloendi nimi, ning vajutada tekstivälja kõrval olevale nupule. Enne tekstivälja sisendi kirjutamist ei ole nupp klõpsatav. Kui toiming on töös, ilmub tekstivälja ja esitusloendi loomise nupu asemele tekst, mis kasutajat sellest teavitab. Tekst säilib seal senikaua, kuni esitusloend on loodud, ning esitusloendite loend uuendatud.

### 3.3.7 Profiilileht

Profiililehele pääseb rakenduse navigeerimisriba kaudu, klõpsates oma kasutajanimega nupule ning valides avanevast rippmenüüst profiililehe valiku.

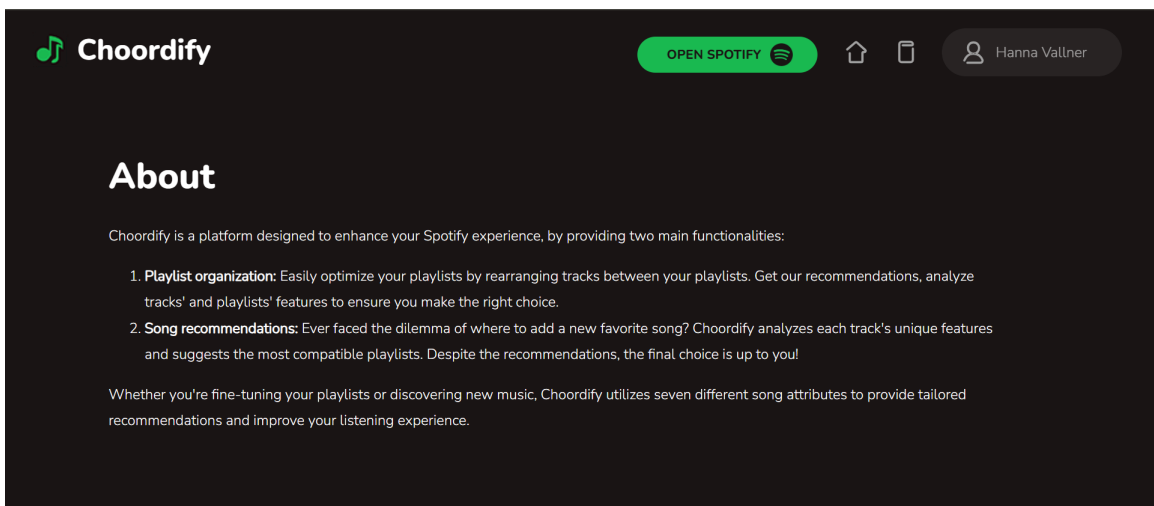
Profiililehel on kuvatud kasutaja nimi, profiilipilt, ning veebirakenduse olemuse tõttu ka esitusloendite arv. Samuti on profiililehel kaks nuppu: üks väljalogimiseks, teine kasutaja profiili avamiseks Spotify lehel. Väljalogimise nupule klõpsamisel suunatakse kasutaja sisselogimise lehele.



Joonis 16. Profiilileht.

### 3.3.8 Tutvustusleht

Rakenduse tutvustuslehele pääseb navigeerimisriba kaudu, klikates märkmiku kujutisega ikoonile.



Joonis 17. Tutvustusleht.

Tutvustuslehel tuuakse lühidalt välja veebirakenduse põhifunktsionaalsused ning kirjeldatakse nende juures tehtavaid soovitusi.

### 3.4 Võimalikud edasiarendused

Loodud veebirakenduse edasiarendamiseks on mitmeid võimalusi, mida bakalaureusetöö aja- ja mahulimiitide tõttu pole veel rakendatud.

Edasiarenduse võimalustest olulisim on Spotify Web APIs rakenduse viimine laiendatud kvoodiga režiimile. Selle jaoks tuleb Spotifyle saata avaldus, kus on veebirakendust põhjalikult kirjeldatud, et tõestada selle vastavust Spotify seatud nõuetele. Töö käigus valminud rakenduse jaoks on laiendatud kvoodiga režiimile üleviimise avaldus esitatud, ent hetkeseisuga pole seda veel vastu võetud. Avaldusele vastuse saamine võtab aega umbes 6 nädalat, mille jooksul vaatab Spotify arendustiim rakenduse üle, kontrollib nõuetele vastavust ning testib töökindlust [12]. Kuna rakendus peab enne avalduse esitamist üldjoontes valmis olema, ning töö käigus loodud rakendus seda piisavalt vara polnud, on see hetkeseisuga veel arendusrežiimis.

Laiendatud kvoodiga režiimis rakenduse eelisena saab seda kasutada piiramatu arv kasutajaid, keda ei pea arendaja eelnevalt lubatud kontode nimistusse lisama. Samuti on selles režiimis rakendusel suurem tehtavate päringute piirarv, mis ühtlasi soodustab mitmeid teisi edasiarendusi.

Loodud rakendus kasutab esitusloendi tunnuste arvutamisel esitusloendi esimese 50 loo tunnuseid, et rakenduse laadimisaeg oleks soovitud piiride raames, ning et tehtud päringute arv ei ületaks limiiti. Seega saab ka esitusloendite vahel laule sorteerida kuni 50 esimese puhul. Samadel põhimõtetel saadetakse päring vaid kasutaja viimase 50 esitusloendi kohta.

Kui rakendus viia üle laiendatud kvoodiga režiimile, saab arvutustes kaasatavate lugude ja esitusloendite piirarvu suurendamist paremini katsetada. Kui testimise tulemustest põhjal otsustada vastavad muudatustes jõustada, tuleb lugude ja esitusloendite arvule siiski kindel limiit seada. Vastasel korral võib mõnel juhul ooteaeg sellegipoolest liiga pikale venida, sest Spotify ei piira kasutaja esitusloendite ega neis olevate lugude hulka.

Alternatiivina on sama eesmärgi saavutamiseks võimalik implementeerida funktsionaalsus, kus kasutaja saab kindlates arvutustes kasutatavad lood ja esitusloendid ise valida. Täielikult töökindel on see võimalus siis, kui kasutaja sisestab soovitud teosed

käsitsi, sest riskimata päringute piirarvu ületamisega pole võimalik kõiki esitusloendeid ega lugusid võimalustena kuvada.

Rakendust saab edasi arendada ka lugude ja esitusloendide sarnasuse leidmise metoodika täiendamisega. Näiteks saab kasutajal võimaldada ise valida, milliseid tunnuseid arvutustes kaasata, ning millisel määral. Lisaks saab võimalike parameetrite hulka täiendada, lisades nii neid, mis lugude tunnuste seast on praegu kõrvale jäetud, kui ka hoopis teisi. Parameetrite hulka saab lisanduvalt arvestada näiteks loo populaarsuse, esitaja nime, albumi nime ja keele. Kuna need tunnused on nominaalsed, tuleb nende kasutamisel klassifitseerida lisaks lugudele ja esitusloenditele ka nende tunnuseid. See suurendaks rakenduse töömahtu piisavalt, et mõistlik oleks selle külge lisada oma andmebaas, kus mõningaid näitajaid hoiustada.

Veebirakenduse jõudluse parendamiseks kuulub selle võimalike edasiarenduste hulka ka tasulise hostimisteenuse või paketi kasutuselevõtt. Sama kehtib rakenduse töövoos kasutatava andmebaasi kohta.

Üldiselt on rakenduse võimalikud edasiarendused pigem olemasolevaid funktsionaalsuseid täiendavad muudatused, ilma milleta on rakenduse eesmärk siiski täidetud.



## Kokkuvõte

Bakalaureusetöö eesmärk oli luua veebirakendus, mis võimaldab kasutajal looutunnuste alusel oma Spotify esitusloendeid hallata. Rakenduse põhifunktsionaalsused olid eesmärgis püstitatuna esitusloenditesse lugude lisamine ning nende vahel lugude sorteerimine. Mõlema toimingu juures pidi rakendus pakkuma lootunnuste võrdlusest lähtuvalt soovitusi.

Enne rakenduse arendamist viidi läbi taustauuring Spotify ning sellega liidestuvate veebirakenduste kohta. Spotify käsitlemisel pandi erilist rõhku selliste funktsionaalsuste uurimisele, mis pakuvad lootunnuste põhjal kasutajale soovitusi. Nii Spotify enda kui sellega liidestuvate veebirakenduste puhul täheldati, et kasutajale jääb piiratud arusaam soovitude aluseks olevatest tunnustest. Seetõttu ei saa kasutaja soovitusalgoritmidega funktsionaalsuste kasutamisel teadvustatud otsuseid teha. Sellest lähtuvalt otsustati loodud veebirakenduse puhul erilist rõhku asetada sellele, et kasutajal oleks täielik arusaam tehtud arvutuste tagamaadest. Töö käigus koostati detailne nimekiri rakendusele seatud nõuetest, mis muuhulgas kehtestavad soovitude läbinähtavuse.

Töö käigus loodi veebirakendus, mis vastab püstitatud eesmärgile ja nõuetele. Lisaks eesmärgiks seatud põhifunktsionaalsustele võimaldab veebirakendus lugusid esitusloenditest kustutada, neid algsest esitusloendist eemaldamata sobivamasse lisada ning luua uusi esitusloendeid. Veebirakenduse arendamisel kasutati järgnevaid tehnoloogiaid: Spotify Web API, Angular, Node.js, Firebase ja Render.

Kuigi loodud veebirakendus sai domeenile üles seatud, oli selle kasutamine töö avaldamise kuupäeval piiratud, sest Spotify ei olnud veel andnud volitust rakendusliidese täisversiooni kasutusele. Rakenduse kasutamiseks peab kasutaja olema arendaja poolt lisatud selle lubatud kasutajate nimistusse. Spotifylt volituse saamisel selline tingimus kaob ning rakendust saavad vabalt kasutada kõik. Samal põhjusel oli karmilt piiratud rakenduse tehtavate päringute arv, millest lähtuvalt saab ligi esitu

## Viidatud kirjandus

- [1] R. A. Rahimi and K. -H. Park, "A Comparative Study of Internet Architecture and Applications of Online Music Streaming Services: The Impact on The Global Music Industry Growth," 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICoICT49345.2020.9166225 (13.05.2024).
- [2] Spotify AB. About Spotify. <https://newsroom.spotify.com/company-info> (07.04.2024).
- [3] Spotify AB. Understanding recommendations on Spotify. <https://www.spotify.com/us/safetyandprivacy/understanding-recommendations> (30.04.2024).
- [4] S. Lindblom. (2015). What made Discover Weekly one of our most successful feature launches to date? <https://engineering.atspotify.com/2015/11/what-made-discover-weekly-one-of-our-most-successful-feature-launches-to-date> (10.05.2024).
- [5] Spotify AB. Smart Shuffle Breathes New Life Into Your Spotify Playlists. <https://newsroom.spotify.com/2023-03-08/smart-shuffle-new-life-spotify-playlists> (10.05.2024).
- [6] Spotify AB. (2024). How does Spotify Radio work? <https://community.spotify.com/t5/FAQs/How-does-Spotify-Radio-work/ta-p/4757060> (10.05.2024).
- [7] Spotify AB. (2018). How Your Daily Mix “Just Gets You”. <https://newsroom.spotify.com/2018-05-18/how-your-daily-mix-just-gets-you> (10.05.2024).
- [8] L. Vardo, J. Jerkić and E. Žunić, "Predicting Song Success: Understanding Track Features and Predicting Popularity Using Spotify Data," 2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2023, pp. 1-6, doi: 10.1109/INFOTEH57020.2023.10094172 (15.11.2023).

- [9] Y. Y. Boon, S. M. Lim, A. Y. Y. Choong and H. Y. Chia, "A Song Classifier for Predicting User Preference Based on Spotify Song Attributes," 2023 13th International Conference on Information Technology in Asia (CITA), Kota Samarahan, Malaysia, 2023, pp. 1-6, doi: 10.1109/CITA58204.2023.10262684 (15.11.2023).
- [10] Spotify AB. Design Guidelines. <https://developer.spotify.com/documentation/design> (03.04.2024).
- [11] Spotify AB. Web API. <https://developer.spotify.com/documentation/web-api> (04.12.2023).
- [12] Spotify AB. Quota modes. <https://developer.spotify.com/documentation/web-api/concepts/quota-modes> (11.04.2024).
- [13] Spotify AB. Web API. <https://developer.spotify.com/documentation/web-api/reference/get-audio-features> (04.12.2023).
- [14] Angular. Introduction to Angular concepts. <https://angular.io/guide/architecture> (16.04.2024).
- [15] Node.js kodulehekülg. <https://nodejs.org/en> (16.04.2024).
- [16] Node.js. About Node.js. <https://nodejs.org/en/about> (16.04.2024).
- [17] Firebase. Cloud Firestore. <https://firebase.google.com/docs/firestore> (13.05.2024).
- [18] Firebase. Cloud Firestore Data model. <https://firebase.google.com/docs/firestore/data-model> (13.05.2024).
- [19] Renderi kodulehekülg. <https://render.com> (13.05.2024).
- [20] Render. Deploy for Free. <https://docs.render.com/free> (13.05.2024).

## **Lisad**

### **I. Litsents**

#### **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Hanna Vallner,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Veebirakendus Spotify esitusloendite korrastamiseks, mille juhendaja on Jaan Janno, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Hanna Vallner*

**15.05.2024**