Branti

A music streaming platform

Mid term progress report

Siddhant Kumar and Rishabh Raj

March 2022

1 Introduction

Branti is a music streaming platform that provides creators with a streamlined process of getting their music released while also giving listeners an enjoyable experience.

It is being developed by **Siddhant Kumar (1912505)** and **Rishabh Raj (1912010)** under the supervision of **Rahul Gautam**.

2 Progress

The progress on **Branti** has been great so far. We have roughly created an SRS document that outlines the features that we are planning to put in place in this project. We have also divided tasks between us.

Rishabh Raj is working on the front-end of **Branti** which includes HTML, CSS, and JavaScript. His responsibilities include implementing the User Interface and building the media player.

I (Siddhant Kumar) will be working on the backend of **Branti** that is being built using the Ruby on Rails framework. I will handle the database modeling, media upload and provide other necessary data required on the front-end.

2.1 Frontend Progress

The front-end of our project consists of a single page that will show available songs, a media player, playlists, and modal windows for adding/editing custom playlists.

The designs are ready for the home page. It is available in the documentation of the code repository (`gitlab.com/siddhantCodes/branti`).

Furthermore, Rishabh Raj has worked on the static HTML and CSS code of the home page and it is almost complete.

2.2 Backend Progress

- The Gitlab project has been created.
- Some documentation on how to setup the database of the project (`gitlab.com/siddhantCodes/branti`).
- Database setup (postgresql) is done.
- Database modeling and relations have been created.
- Authentication (email-password and Google OAuth sign-in) has been implemented.
- Test runner and framework (Rspec) is installed.
- Tests for existing models are in progress.

3 Pending Tasks

- Tests for models and associations between database models.
- Media upload (.webm music, thumbnails and user avatar).
- Controller tests.
- E2E tests (Optional).
- Documentation.

4 Future Goals

Once the MVP is completed with basic media playback, we would like to improve it further by adding some planned features like:

- Upload in chunks to support parallel upload and resume capability of songs.
- Chunked streaming (http2?) to serve large and high quality audio files.
- Subtle animations and new standardised UI components

4.1 Possible To dos

- Model tests.
- Media upload.
- Controller tests.
- Documentation.

Supervisor's Signature