Online Music App- Zoomy

Group Members:

Rutuja Jagtap

Sahil Wadekar

Aditya Patil

Guided By: Ashish Patwardhan

Given Problem Statement:

To create an Online music application UI that will serve as an on the go entertainment service for the user. The music app will provide basic features like users profile creation, authentication, media player, music browsing and discovery.

The detailed features are as follows:

1. Signup.
2. Login.
3. Browse Songs.
4. Search Songs by their title.
5. Display Searched results.
6. Display filtered Search results.
7. Display their profile.
8. Show all users profile info.

Scope And Objective:

The project's scope involves the creation of a Single Page Application (SPA) that offers an interactive user experience. The SPA will be divided into distinct sections, each of which will be dynamically loaded upon user interaction, such as button clicks or scrolling. The application will encompass a range of interactive elements, including forms, buttons, links, and lists.

The SPA will primarily feature 4 key components:

User Authentication:

The project will introduce a user authentication system, enabling users to create

accounts, log in, and log out. This ensures that only authenticated users can access

the app.

User Profile Management:

Authenticated users will have access to a personalised user profile section. Here,

they can update their personal information, including contact details and

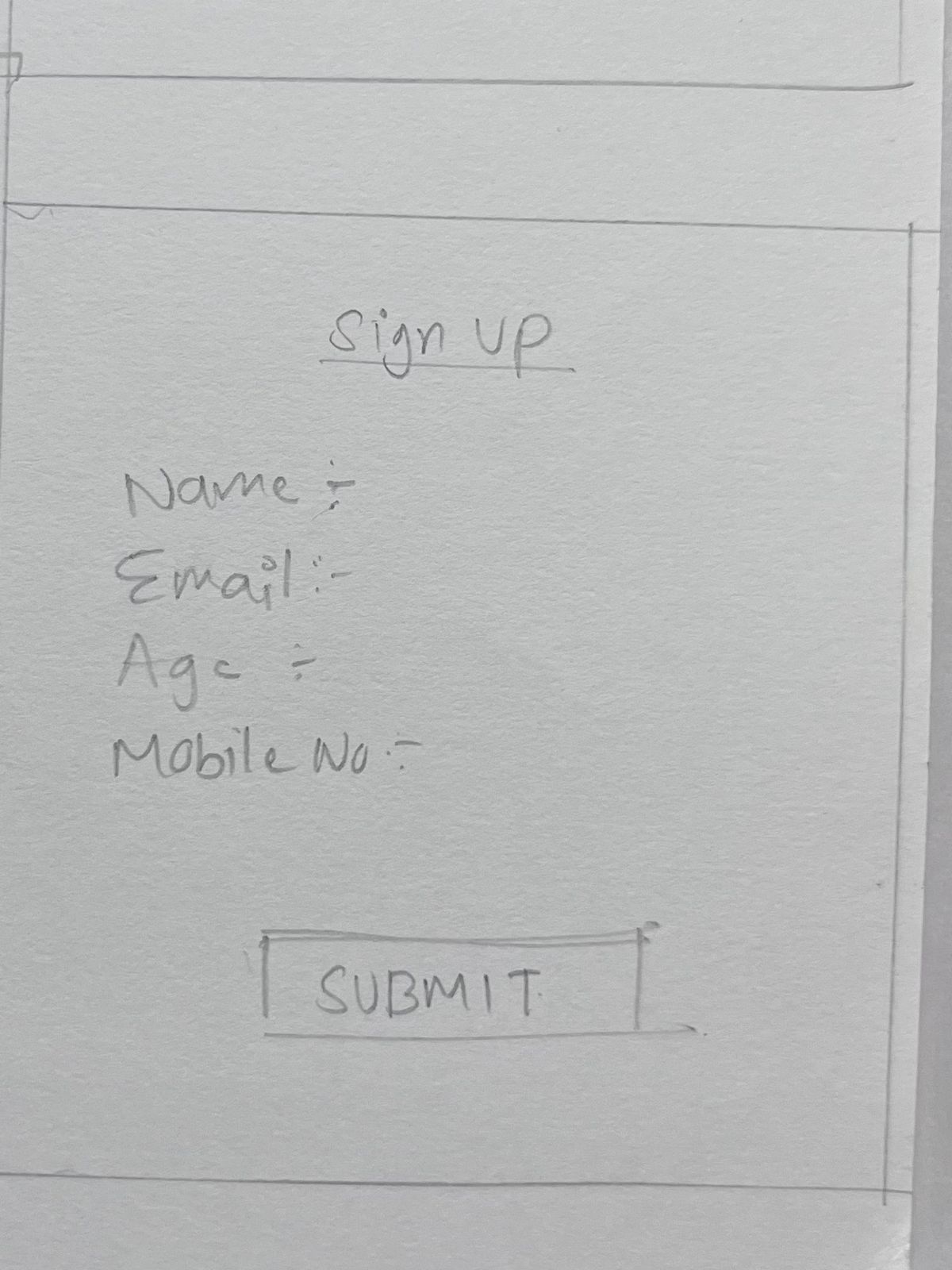
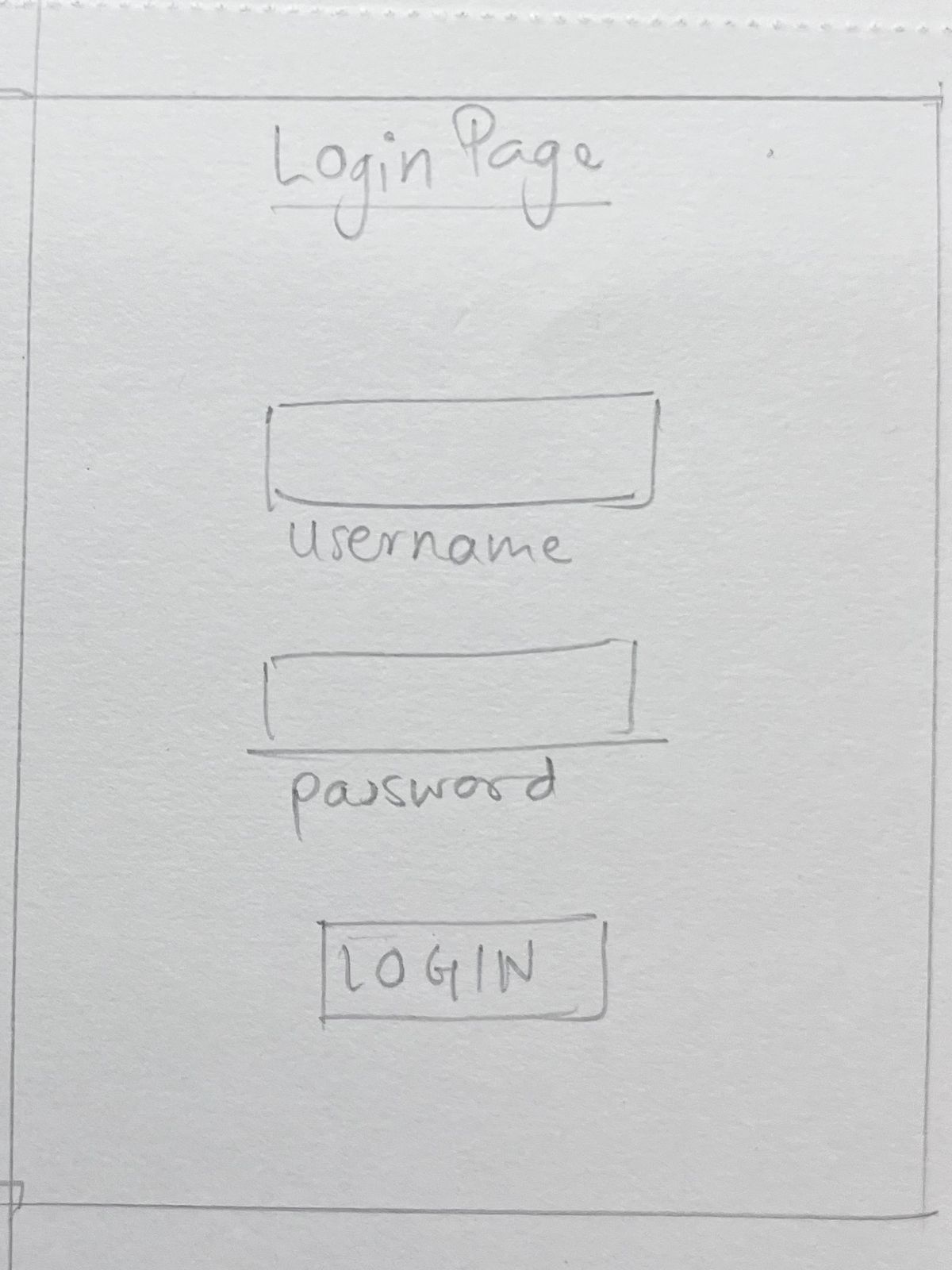
Preferences.

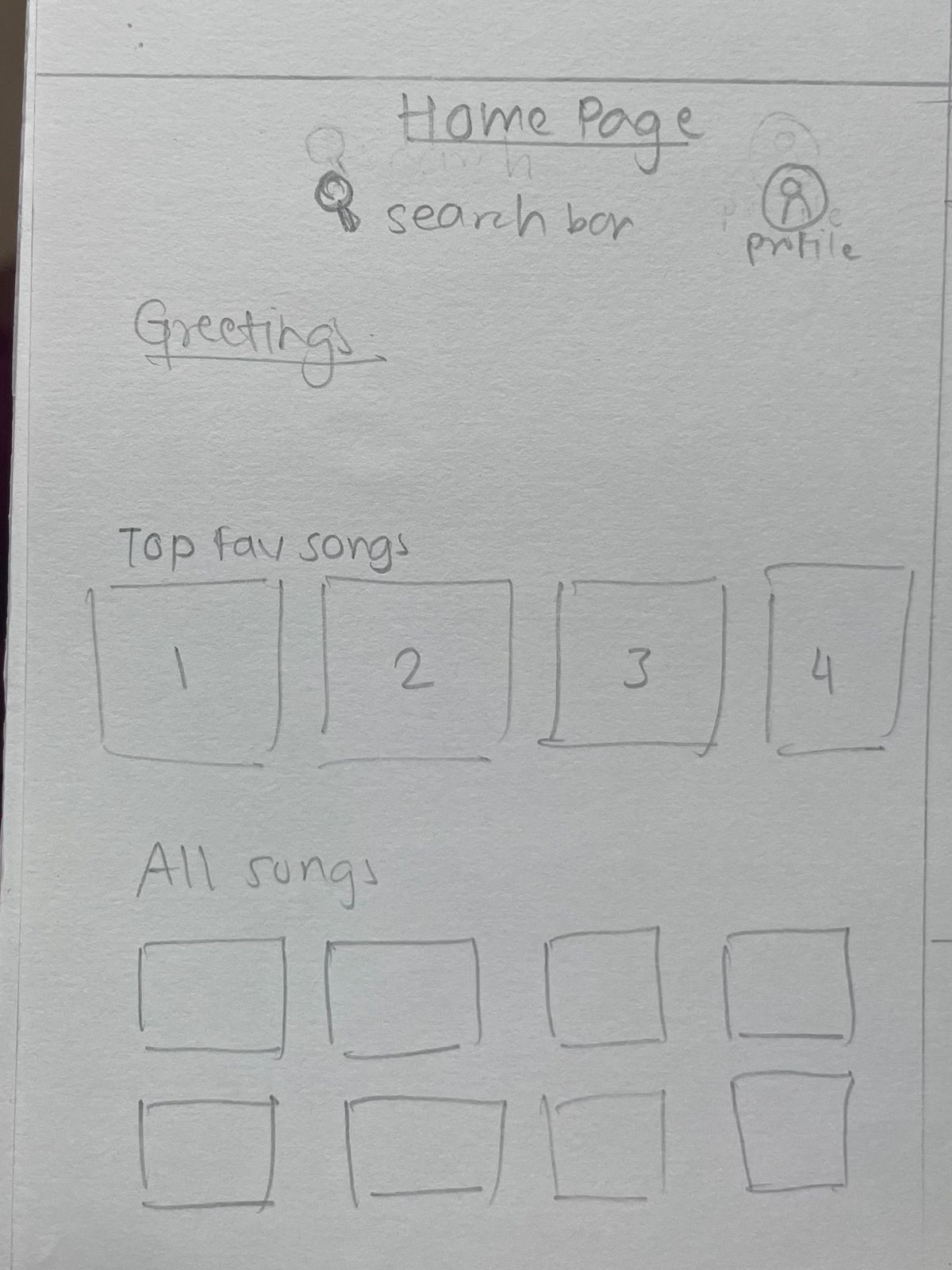
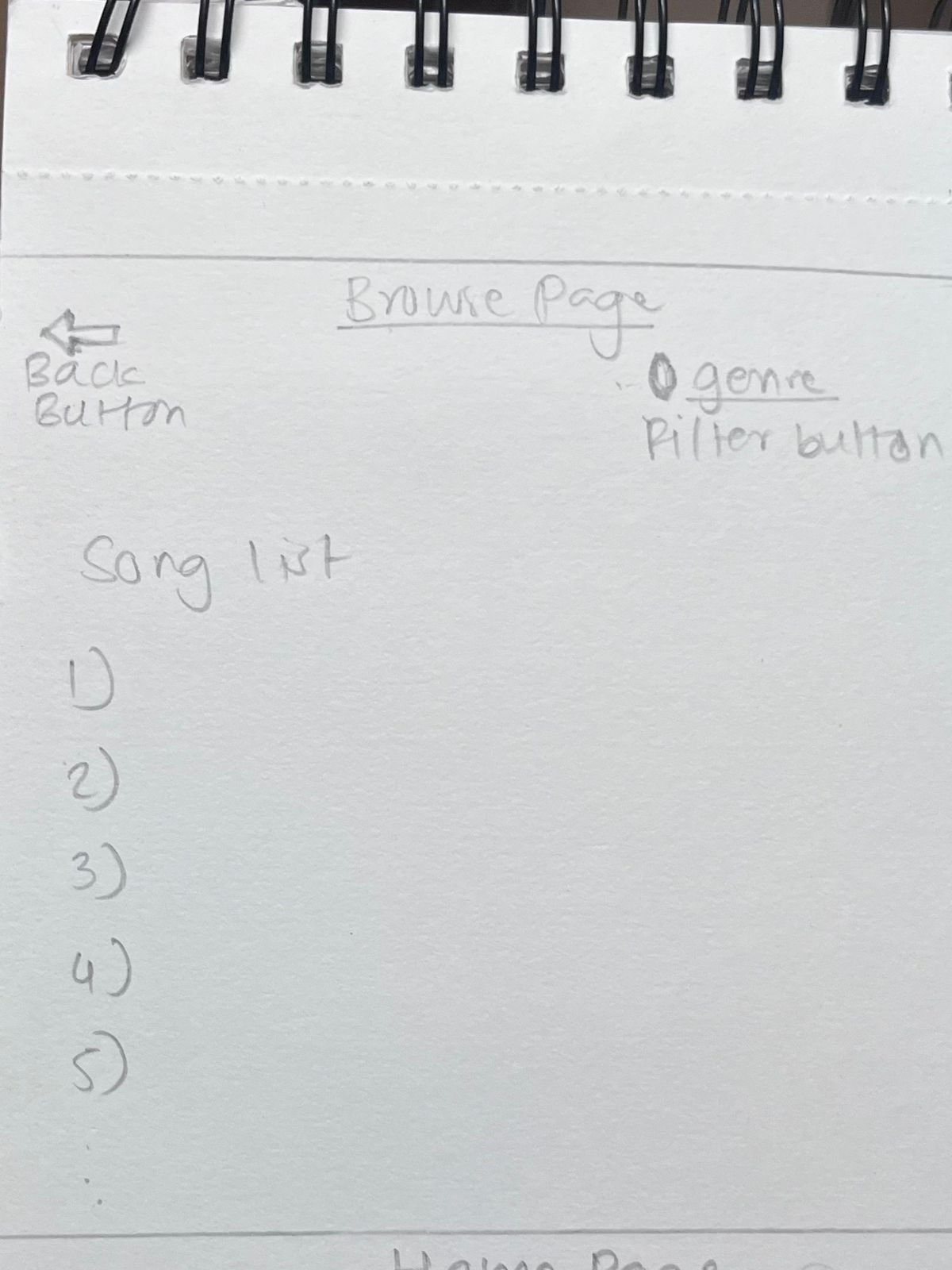
Media player:

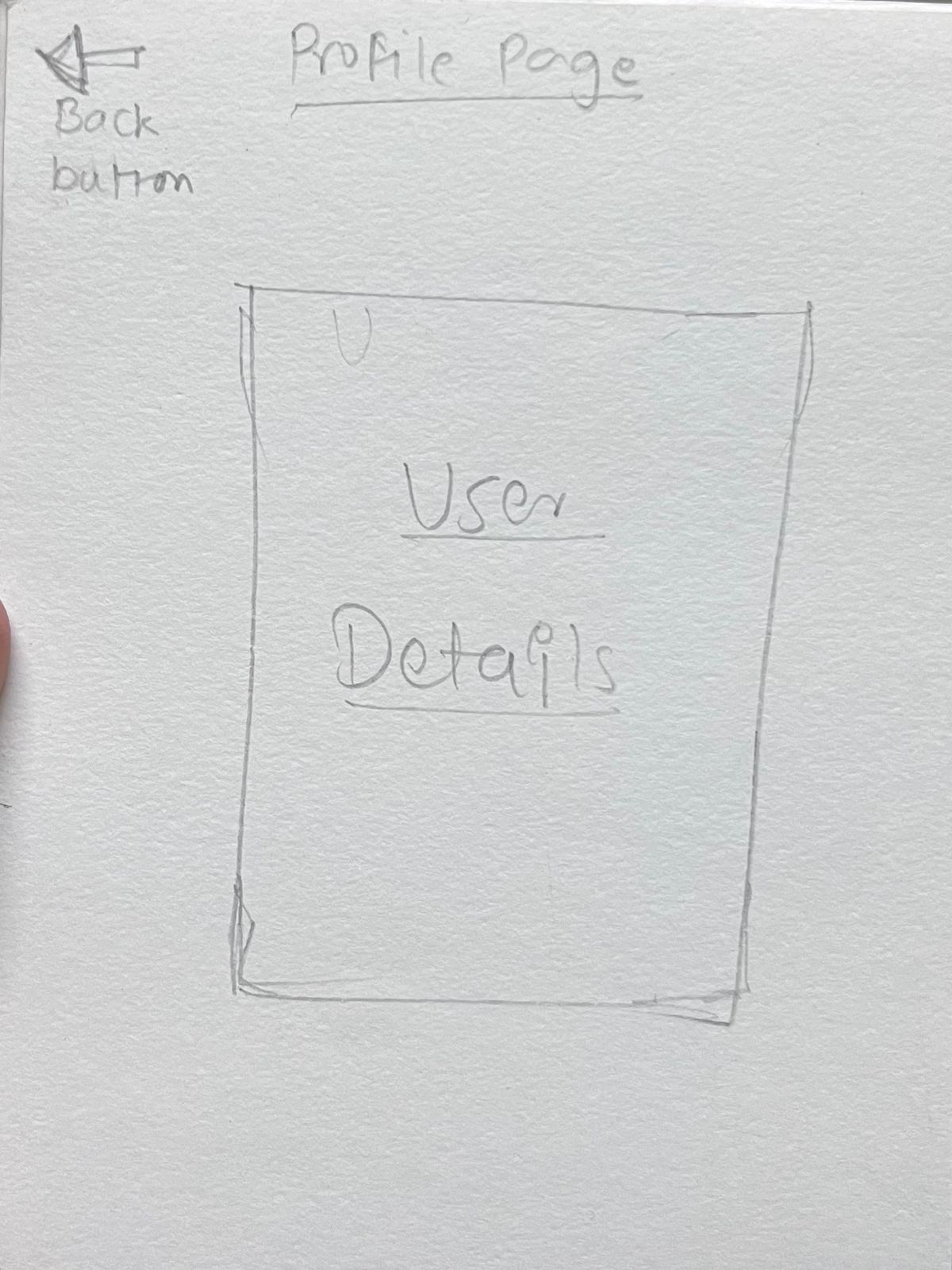
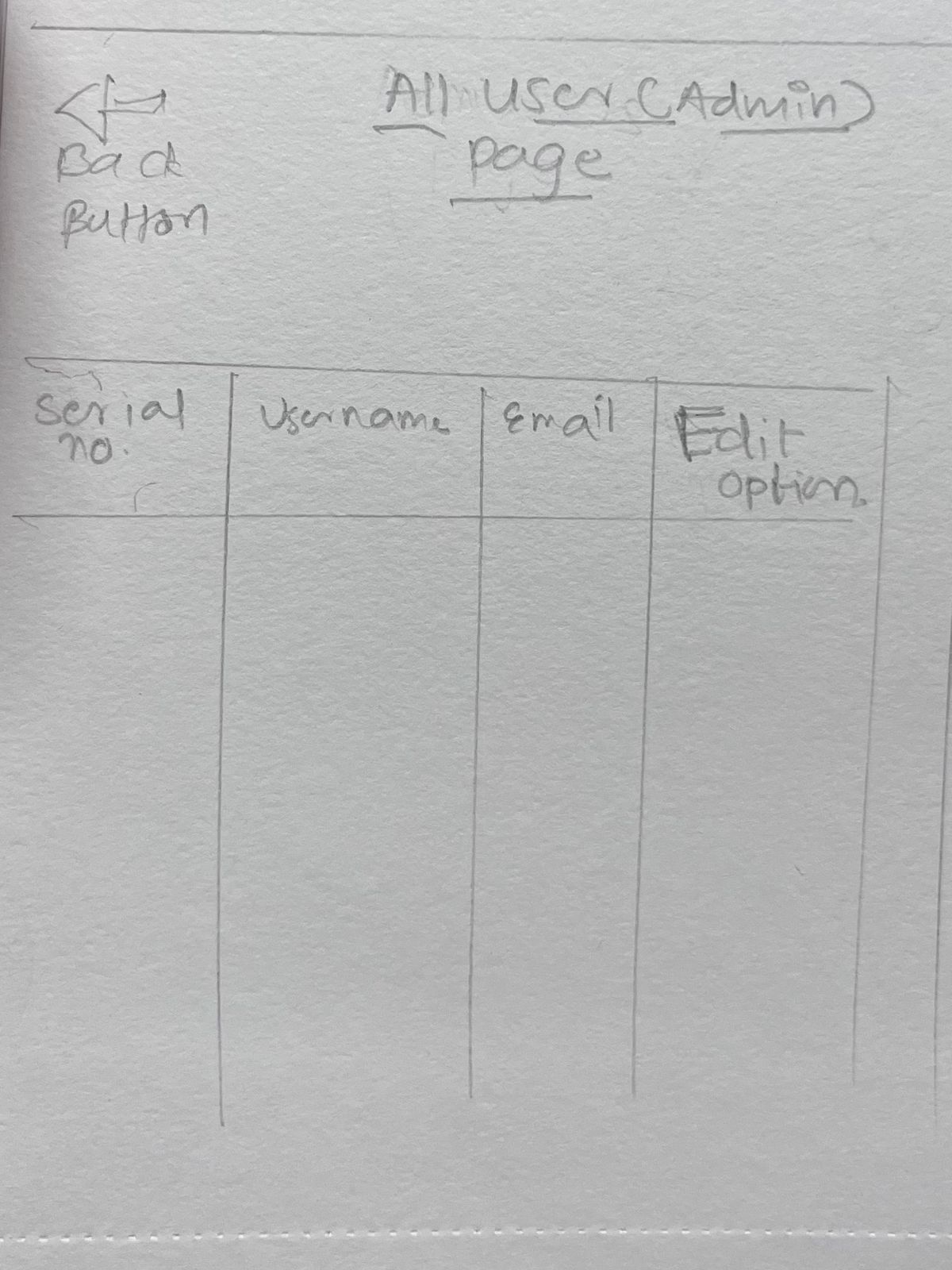
The logged in user will allowed to play the music in the app right in the browser. The user can save the music to the profile as liked song.

All User info:

Only a admin user will be privileged to access the all user info page where a table will be shown displayed with basic details of all the users.

Concepts Applied

1.SPA (Single Page Application): When a user visits the music app, As an SPA, the entire application loads as a single page, offering a smooth and uninterrupted browsing experience.

2. Module: The application is divided into modules, such as "MusicPlayerModule" and "MusicBrowseModule." These modules encapsulate related functionalities and components.

3. Controllers: Within the "MusicPlayerModule," the "MusicPlayerController" manages interactions related to the music playing and browsing, such as fetching and displaying song data.

4. Views: The "HomeView" is a component of the application's interface managed by the "HomeViewController." It displays the list of available songs and playlists with their details and images.

5. Include: The "MusicBrowseModule" may include a navigation bar template, and other common templates using the include directive, ensuring consistent design across the application.

6. Filters: A "GenreFilter" filter could be applied to the various songs genres displayed in the "MusicBrowseView," allowing users to filter songs based on their tastes.

7. Forms: Users can sign up and login. The ”LoginForm” and “SignupForm” allows users to input their details, including name, email, and contact number.

8. Form Events: Upon submitting the "SignupForm," a "Submit" event is triggered, which returns a valid message or else if

9. Form Validations: Before the "SignupForm" and “LoginForm” is submitted, client-side form validations ensure that mandatory fields are filled correctly, preventing incomplete or erroneous data from being submitted.

10.Data Binding: As users input data in the "SearchBox" the two-way data binding updates the “SearchString” display in real-time, providing immediate feedback to users about their input.

11. $http and Ajax: When the user submits the "SearchBox” the $http service sends an Ajax request to the local server to fetch the json containing the list of songs.

12.Use of Services: The heck"MediaPlayerService," a custom service, handles the media that is to be loaded in the “MediaPlayerView”. It encapsulates HTTP requests, making the code more modular and maintainable.

Work allocation

| **Work** | **Status** | **Assigned to** |
| --- | --- | --- |
| Documentation | Completed | Sahil, Aditya, Rutuja |
| Wireframes | Completed | Sahil |
| Environment Setup | In progress | Rutuja |
| SPA Template | In progress | Aditya |
| Views - Login | In progress | Sahil |
| Views - Signup | Not started | Rutuja |
| Views - Home | Not started | Aditya |
| Views - Browse Music | Not started | Sahil |
| Views - User Profile | Not started | Rutuja |
| Views - All users | Not started | Aditya |
| Json Data | Not started | Sahil, Rutuja |
| CSS Graphics | Not started | Aditya, Sahil |
| Testing | Not started | Rutuja, Sahil, Aditya |

Signatures

Aditya Patil: \_\_\_\_\_\_\_\_\_\_\_\_\_

Sahil Wadekar: \_\_\_\_\_\_\_\_\_\_\_\_\_

Rutuja Jagtap: \_\_\_\_\_\_\_\_\_\_\_\_\_