**MOOD SWING**

**Web Application for Sentiment Based Music**

**Tech stack:**

1. **Python Flask**
2. **Firebase Real time database, Storage, and authentication**
3. **JavaScript, HTML, CSS**

**Index Page & Authentication:**

**A screenshot of a computer

Description automatically generated**

1. Start the App by running **App.py.**
2. Uses the template from **templates/index.html**
3. In that html file will use **static/js/script.js** as a script file.
4. In that Sricpt file there is a function called **validate() –** that will take care of all the **Sign In & Sign Up** activities
5. That validate function will use some functions from **static/js/Db.js,** to connect with firebase authentication.(refer the block diagram below)

**A screenshot of a computer flowchart

Description automatically generated**

----------------- END OF **Index Page & Authentication** ------------------

**Dashboard Page:**

**A screenshot of a computer

Description automatically generated**

1. After successful sign in **app.py** will render the **dash.html** template from **templates/dash.html.**

**A screenshot of a computer program

Description automatically generated**

------------------------- END OF **Dashboard Page** ------------------------

1. **Dashboard – Mood Swing**

A screenshot of a video recording

Description automatically generated

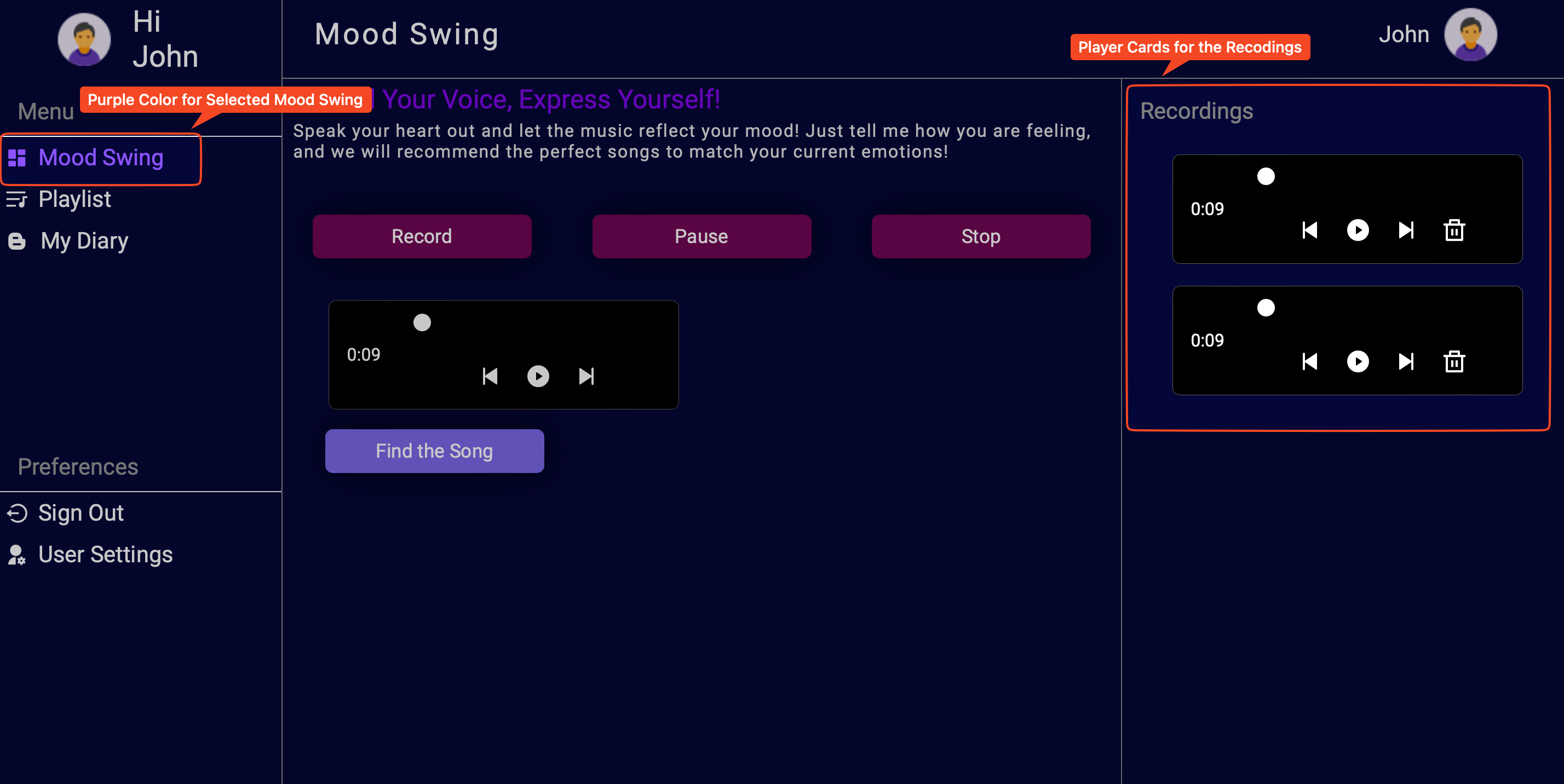
1. dash.html file will use **static/js/MoodSwing.js** as a script file.
2. There are several functions in MoodSwing.js which is responsible for Recording related activities.(refer the block diagram below)

NOTE: Some of those function will use functions from **static/js/Db.js and static/js/Recording.js**

A screenshot of a computer

Description automatically generated

1. After finish recoding it will store in the firebase cloud storage by using saveRecording() function from **static/js/Db.js.**
2. Also, recodings will shown in the dashboard as **Playing Cards** as Shown below.



1. dash.html file will use **static/js/Recording.js** as an another script file.
2. There are several functions in **Recording**.**js** which is responsible for **Player Card** related activities.(refer the block diagram below)

**NOTE**: Some of those function will use functions from **static/js/Db.js.**

1. Also there is a function in **Recording**.**js** which is responsible for **adjust the side bar** selection by giving a separate color as shown in the above screen shot.

A screen shot of a computer

Description automatically generated

A screenshot of a video chat

Description automatically generated

1. After recoding the audio, Using **Find the Song button** we can search the songs which is suitable for our mood in the recoded audio.
2. For that, dash.html file will use **static/js/Spotify.js** as an another script file.
3. There are several functions in **Spotify**.**js** which is responsible for **Spotify API** related activities which is Find the Correct song for our current mood.(refer the block diagram below)

A screenshot of a computer

Description automatically generated

In below NOTE you can get an idea about how sentiment analysis function (**transcribe\_audio ()**) really works

NOTE: **transcribe\_audio () FUNCTION IN app.py**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. After find the recommended songs from Spotify API, we can add them to our own playlist in the firebase database.
2. As Stated, dash.html file will use **static/js/MoodSwing.js** as a script file.
3. There is a function called **addCardEventListners(Cards)** in the **MoodSwing.js** which is responsible for add songs to playlist in the firebase database. .(refer the block diagram below)

A screenshot of a computer

Description automatically generated

----------------- END OF **Dashboard – Mood Swing** --------------------

1. **Dashboard – Playlist**

A screenshot of a music player

Description automatically generated

1. As Stated, dash.html file will use **static/js/MoodSwing.js** as a script file.
2. This file is also have functions which is responsible for activities related to playlist section in the dashboard (i.e delete playlist, render playlist).(refer the block diagram below)

NOTE: add playlist also done in the **MoodSwing.js** (explained in the last part in **dashboard - mood swing** above)

A screenshot of a computer program

Description automatically generated

-------------------- END OF **Dashboard – Playlist** ------------------------

1. **Dashboard – My Diary**

A screenshot of a computer

Description automatically generated

1. As Stated, dash.html file will use **static/js/Recording.js** as a script file.
2. This file is also have functions which is responsible for activities related to My Diary section in the dashboard (i.e add blog to firebase database, delete blog from firebase database, get all the blogs from the firebase database, render all blogs in the webpage).(refer the block diagram below)

NOTE: **adding blogs, get all blogs** will happen inside **arrow function**, **delete and render** will happen inside the **renderBlog()** function.

A screenshot of a computer

Description automatically generated

-------------------- END OF **Dashboard – My Diary** -----------------------

1. **Dashboard – User Setting**

**A screenshot of a computer screen

Description automatically generated**

1. As Stated, dash.html file will use **static/js/Recording.js** as a script file.
2. This file is also have functions which is responsible for activities related to User Setting section in the dashboard (i.e change user name, change password, date of birth update, change profile picture).(refer the block diagram below)

NOTE: arrow function will use functions from **static/js/Db.js**

**A screen shot of a computer

Description automatically generated**

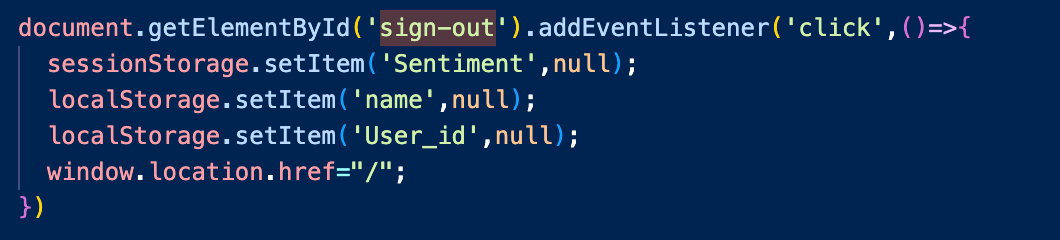
----------------- END OF **Dashboard – User Setting** ---------------------

1. **Dashboard – User Setting**

A screenshot of a computer

Description automatically generated

Sign out will be taken care by the **static/js/MoodSwing.js.**



----------------- END OF **Dashboard – Sign out** ---------------------

**Appendix**

**(Function name will represent the functoriality of each function.)**



1. **Spotify API Connection:**

**In script/Spotify.js,**

There is **two function** responsible for this:

1. **function authenticate()**

A blue screen with colorful text

Description automatically generated

This function is responsible for initiating the Spotify authentication process.

It constructs a Spotify authorization URL (https://accounts.spotify.com/authorize) with specific parameters:

* **client\_id:** The Spotify client ID (obtained from your Spotify Developer Dashboard).
* **redirect\_uri**: The URL to which Spotify will redirect the user after authentication. This URL should match one of the redirect URIs configured in your Spotify application settings.
* **scope**: The required scopes (permissions) for accessing user data. In this case, the function requests permission to read the user's private information (e.g., profile and email).

After constructing the authorization URL, the function redirects the user to this URL using **window.location.href,** initiating the Spotify authentication flow.

1. **function getAccessTokenFromHash()**

A blue screen with text on it

Description automatically generated

After successful authentication and redirect back to your application's redirect\_uri, call getAccessTokenFromHash() to extract the access token from the URL hash fragment and store/use it for making API requests to Spotify on behalf of the authenticated user.

1. **Firebase Initialization:**

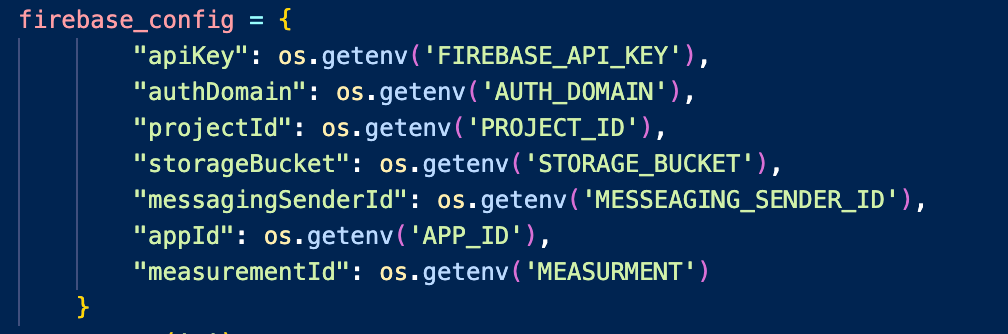
Following script block will be found in both **template/index.html & template/dash.html**

A screen shot of a computer code

Description automatically generated

* This script block initializes a **JavaScript object** named **firebaseConfig** with Firebase configuration parameters.
* The values of these parameters are placeholders

**({{ firebase\_config.<param> }})** that will be dynamically replaced by values that defined/read from **app.py** (code block from app.py shown below)



**NOTE: below 3 parts are not important for the project.**

1. **Explanation for taostify.js:**

* **node modules/toastify-js 🡪 File is directly download from** <https://github.com/apvarun/toastify-js>

\*\*\*this code has not been used in project

A screenshot of a computer program

Description automatically generated

1. **Explanation for dotenv:**

* \*\*\*node module/dotenv file also not used in the project code
* But below part of the code in **app.py** uses **dotenv python module** *(not the one from node module/dotenv)*,
* but **we don’t have a .env** file to configure the environment variable, **so we can skip that part in the code**.
* If you're working with a Flask application and you want to configure it without a .env file, you can directly set environment variables in your operating system.

A screenshot of a computer

Description automatically generated

1. **Explanation for config.js:**

* **config/config.js used to load environment variables from a .env** file into the Node.js environment using the dotenv package, and then export those variables as part of a configuration object.
* \*\*\* Since we don’t have .env file in the project config part also won’t be a part of the project.

A screen shot of a computer program

Description automatically generated