

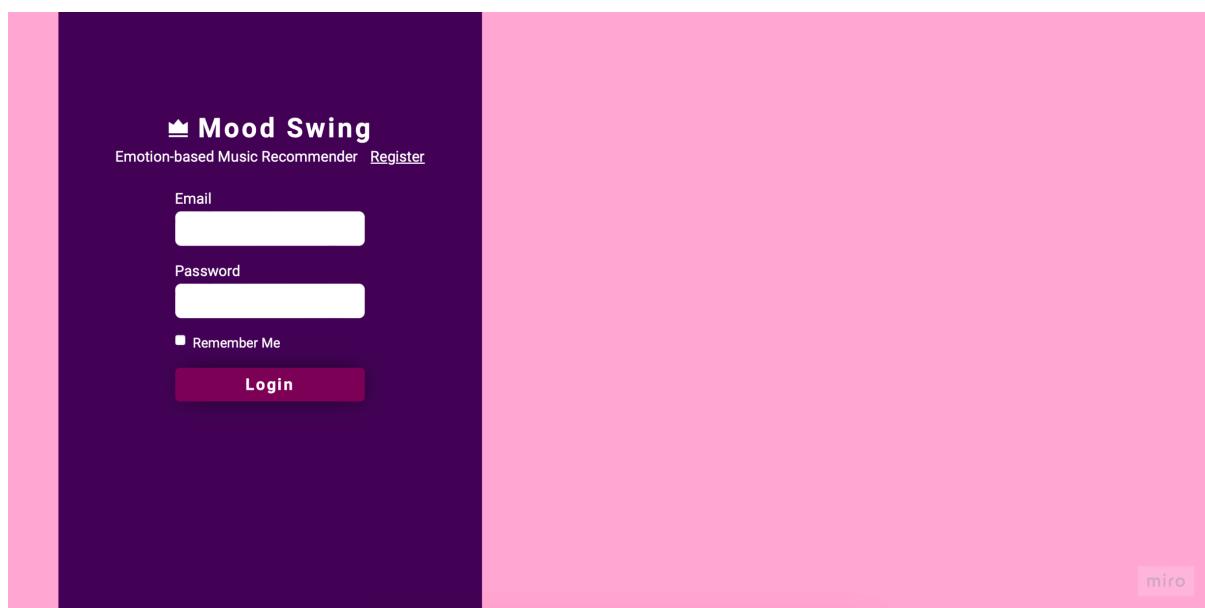
MOOD SWING

Web Application for Sentiment Based Music

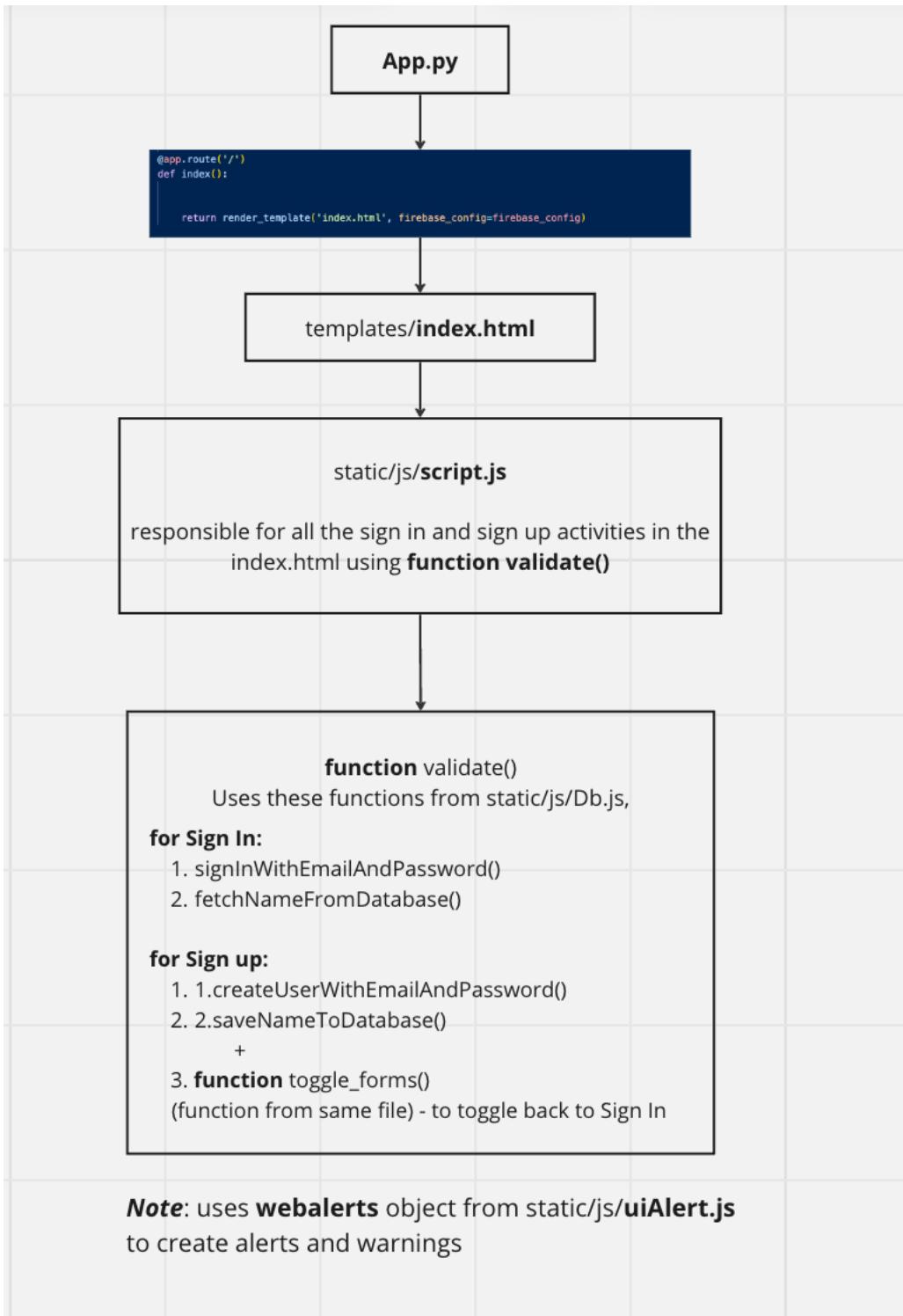
Tech stack:

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

Index Page & Authentication:

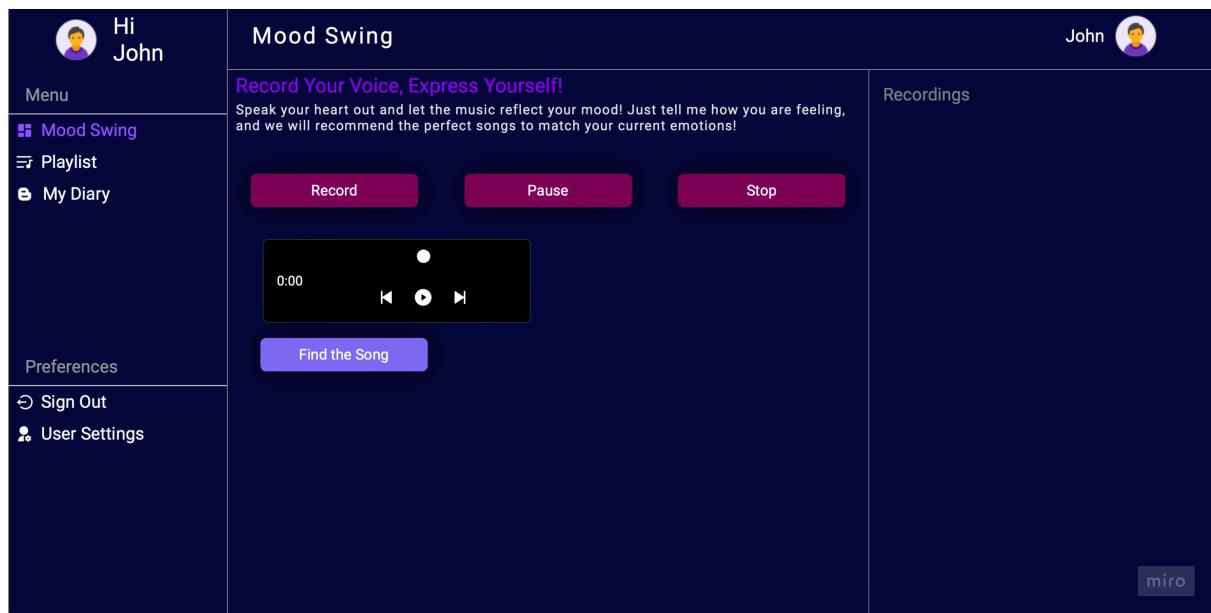


1. Start the App by running **App.py**.
2. Uses the template from **templates/index.html**
3. In that html file will use **static/js/script.js** as a script file.
4. In that Script file there is a function called **validate()** – that will take care of all the **Sign In & Sign Up** activities
5. That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

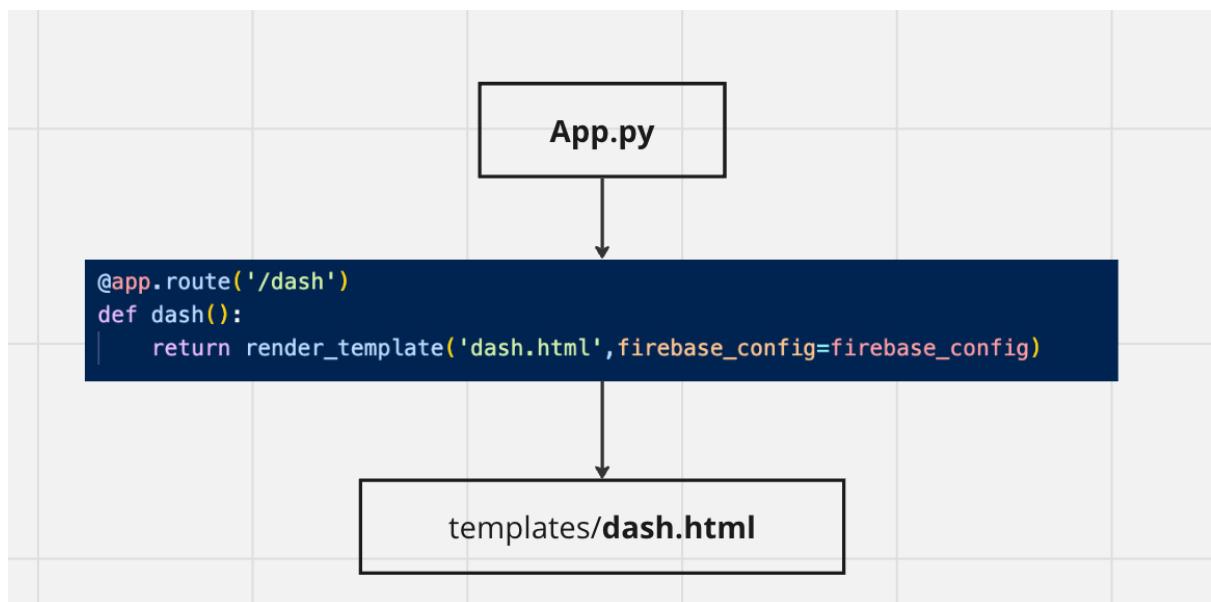


----- END OF Index Page & Authentication -----

Dashboard Page:

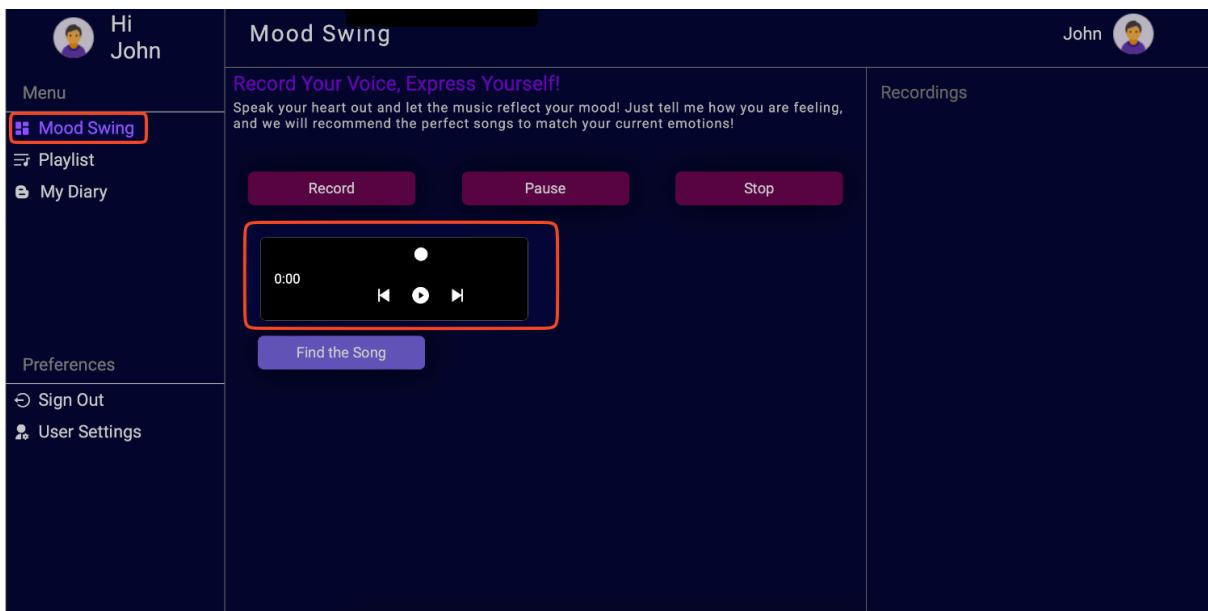


1. After successful sign in **app.py** will render the **dash.html** template from **templates/dash.html**.



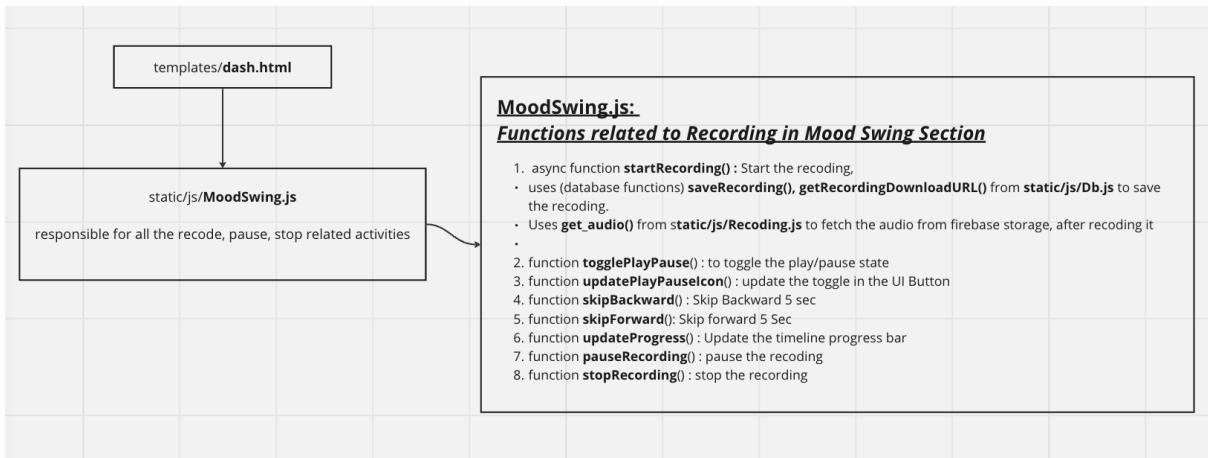
----- END OF Dashboard Page -----

1. Dashboard – Mood Swing

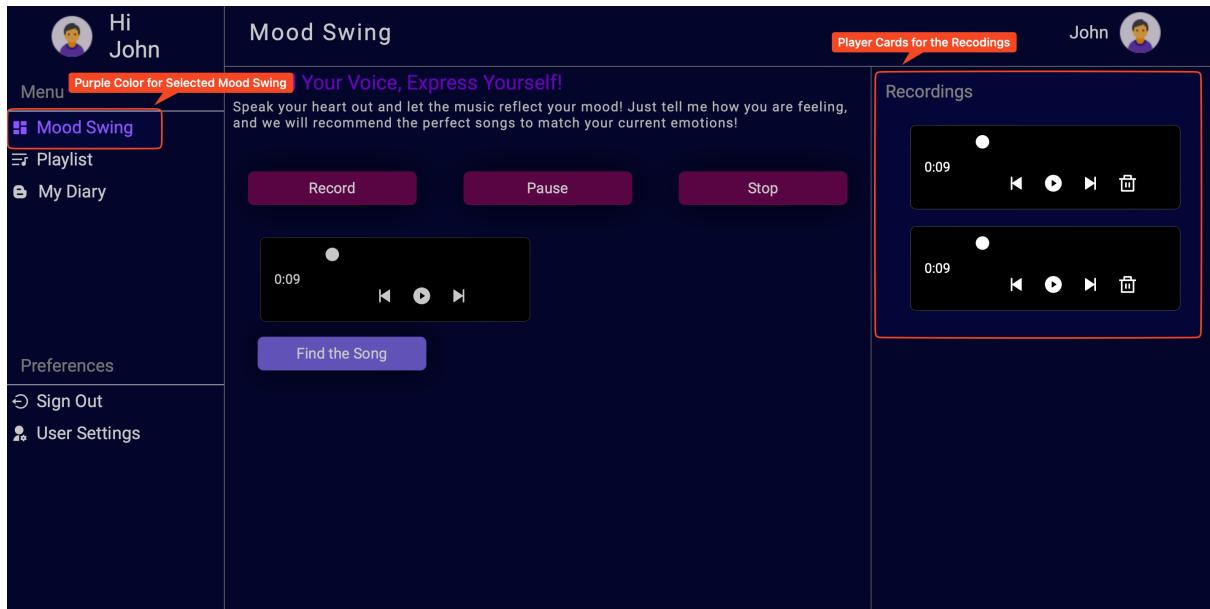


1. dash.html file will use **static/js/MoodSwing.js** as a script file.
2. There are several functions in MoodSwing.js which is responsible for Recording related activities.(refer the block diagram below)

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



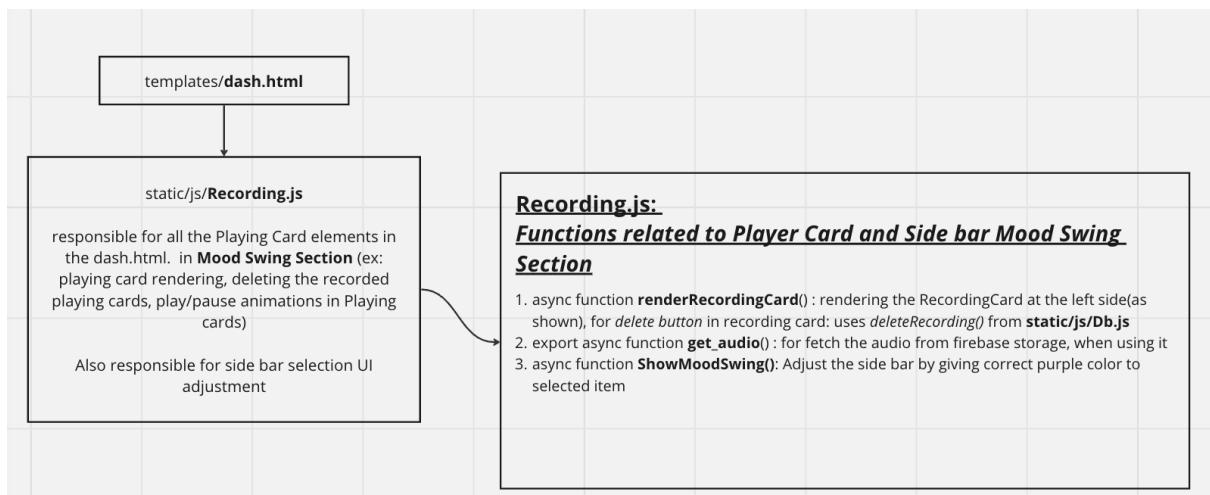
3. After finish recoding it will store in the firebase cloud storage by using `saveRecording()` function from **static/js/Db.js**.
4. Also, recordings will shown in the dashboard as **Playing Cards** as Shown below.

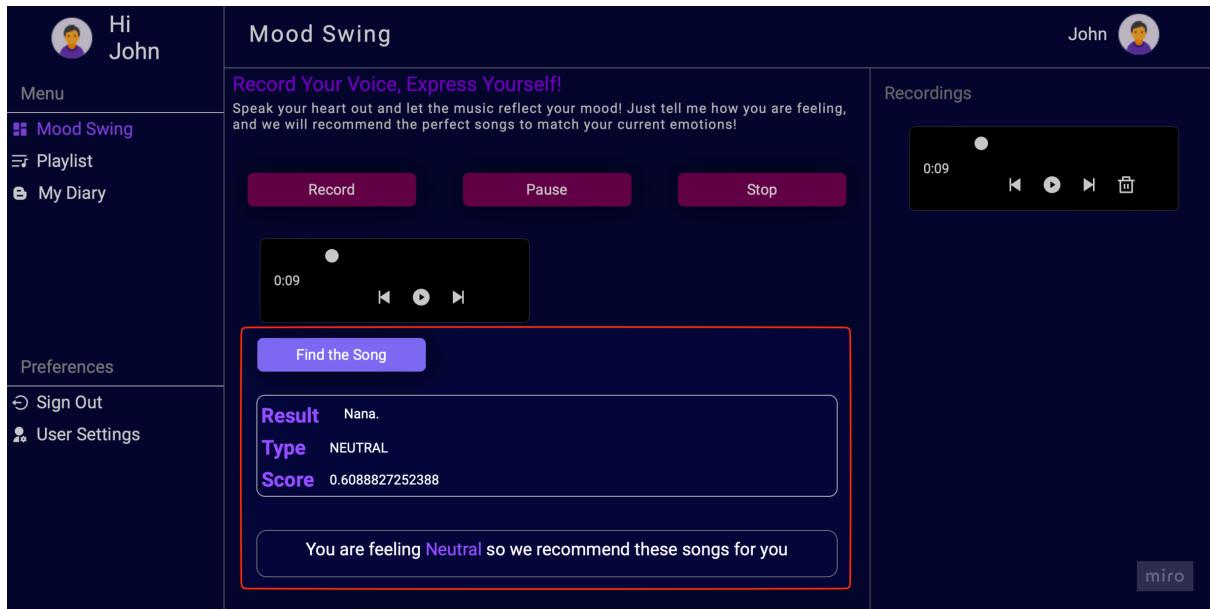


5. dash.html file will use **static/js/Recording.js** as an another script file.
6. There are several functions in **Recording.js** which is responsible for **Player Card** related activities.(refer the block diagram below)

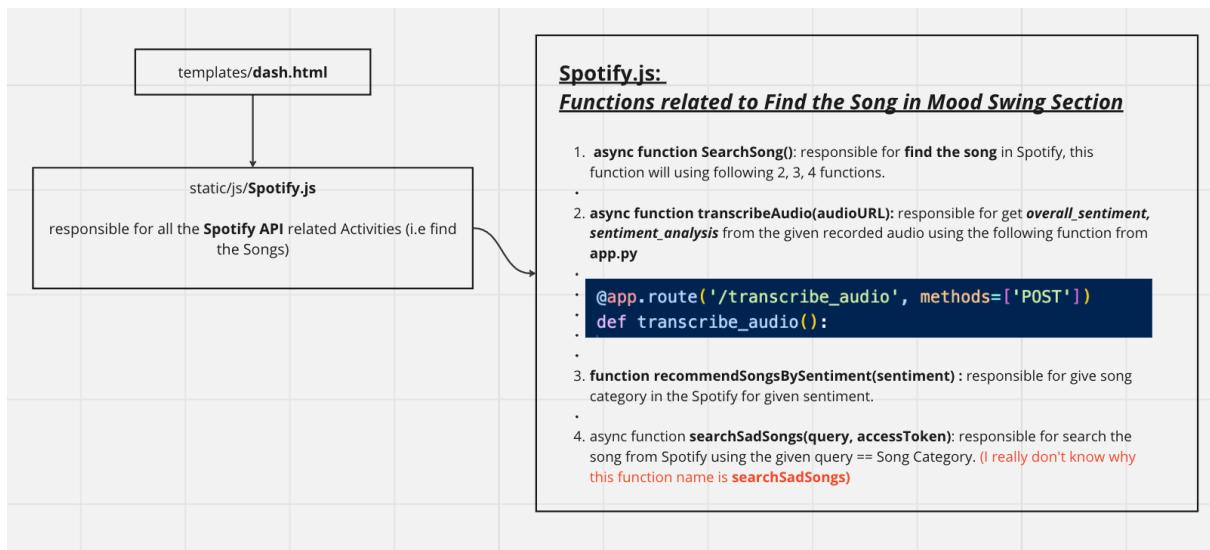
NOTE: Some of those function will use functions from **static/js/Db.js**.

7. 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.





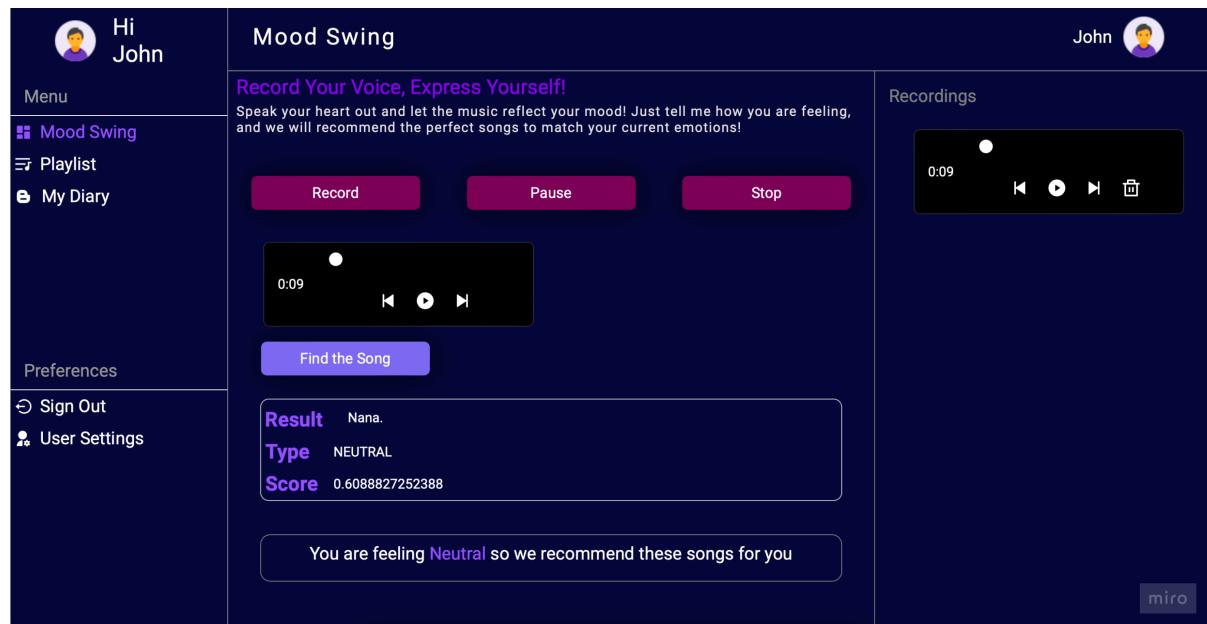
8. After recoding the audio, Using **Find the Song button** we can search the songs which is suitable for our mood in the recoded audio.
9. For that, dash.html file will use **static/js/Spotify.js** as an another script file.
10. There are several functions in **Spotify.js** which is responsible for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



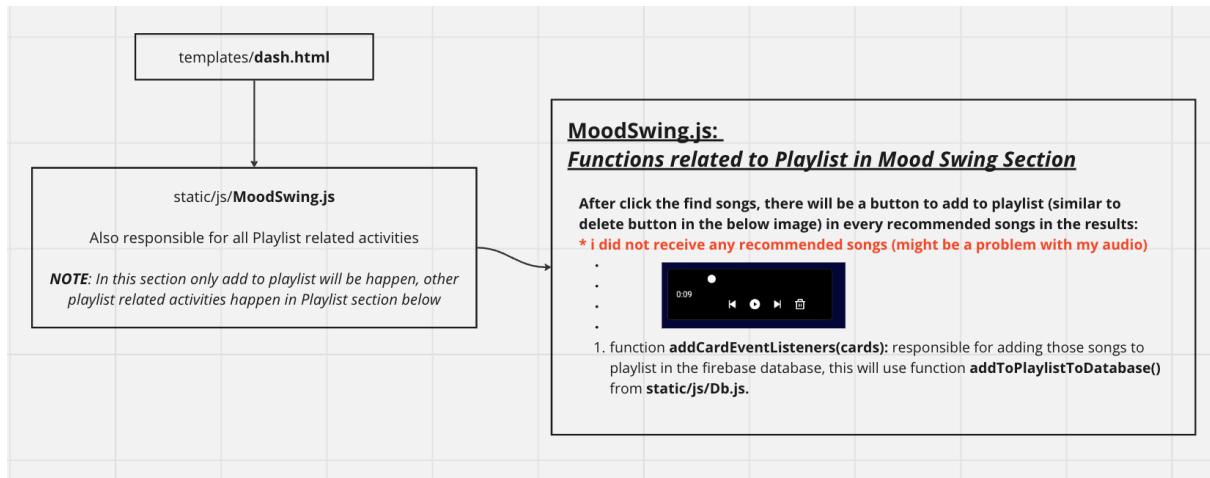
In below NOTE you can get an idea about how sentiment analysis function (**transcribe_audio()**) really works

NOTE: `transcribe_audio()` FUNCTION IN `app.py`

EXPLANATION FOR THE <code>transcribe_audio()</code> FUNCTION IN <code>app.py</code>	
1.	The <code>'transcribe_audio()'</code> function is a Flask route designed to transcribe an audio file specified by its URL (<code>'audio_url'</code>) and perform sentiment analysis on the transcription results using the python's <code>'aai'</code> module.
2.	This module likely encapsulates functionality for audio transcription and sentiment analysis . The input to the <code>'aai'</code> module includes the audio URL (<code>'audio_url'</code>) and a transcription configuration (<code>'TranscriptionConfig'</code>) with sentiment analysis enabled.
3.	The <code>'Transcriber'</code> class is then used to transcribe the audio file based on the provided configuration. The output from this process includes a transcription (<code>'transcript'</code>) along with sentiment analysis results for various segments of the transcribed text.
4.	The function aggregates these results to calculate an overall sentiment (<code>'overall_sentiment'</code>) based on the most frequent sentiment label .
5.	The final output is formatted into a JSON response (<code>'result'</code>) containing the overall sentiment and sentiment analysis details for each segment of text analyzed , which is then returned to the client.
6.	The function also includes error handling to catch and report any exceptions that may occur during the transcription and sentiment analysis process.

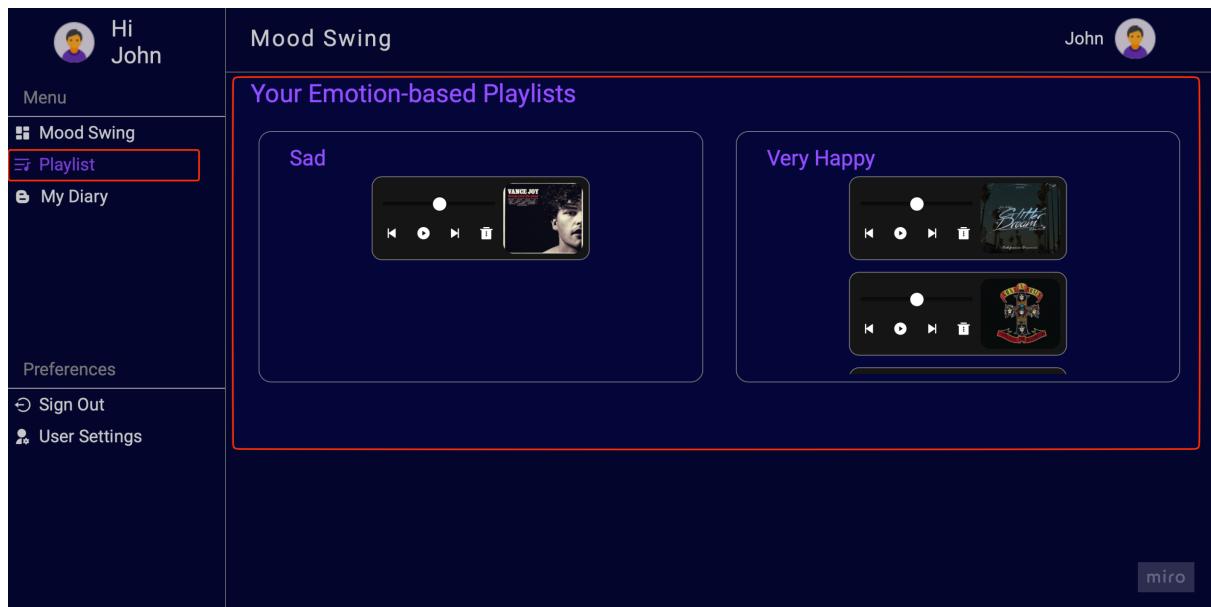


11. After find the recommended songs from Spotify API, we can add them to our own playlist in the firebase database.
12. As Stated, dash.html file will use **static/js/MoodSwing.js** as a script file.
13. There is a function called **addCardEventListeners(Cards)** in the **MoodSwing.js** which is responsible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



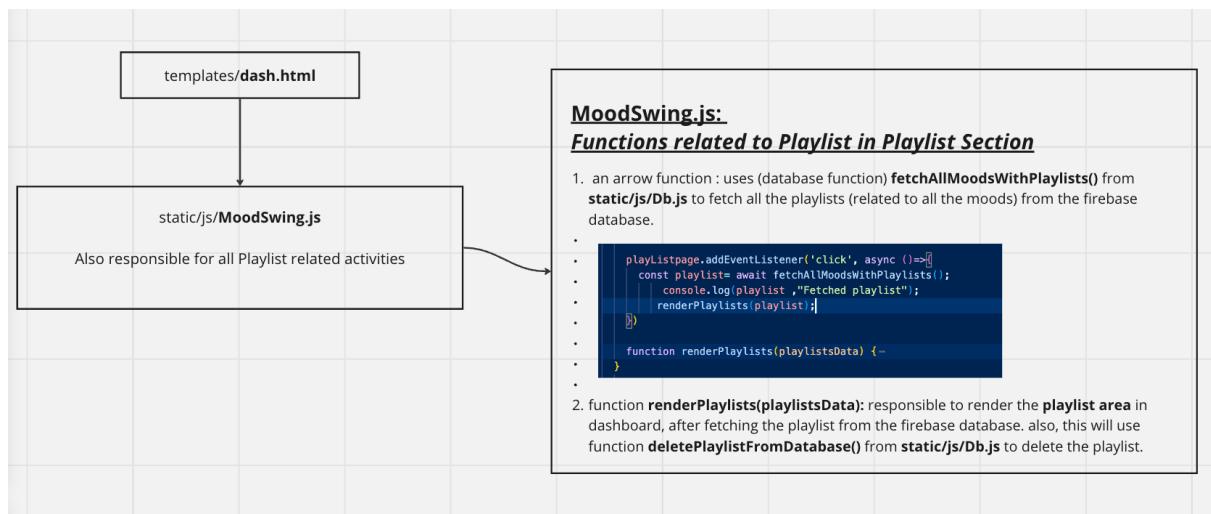
----- END OF Dashboard – Mood Swing -----

2. Dashboard – Playlist



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)



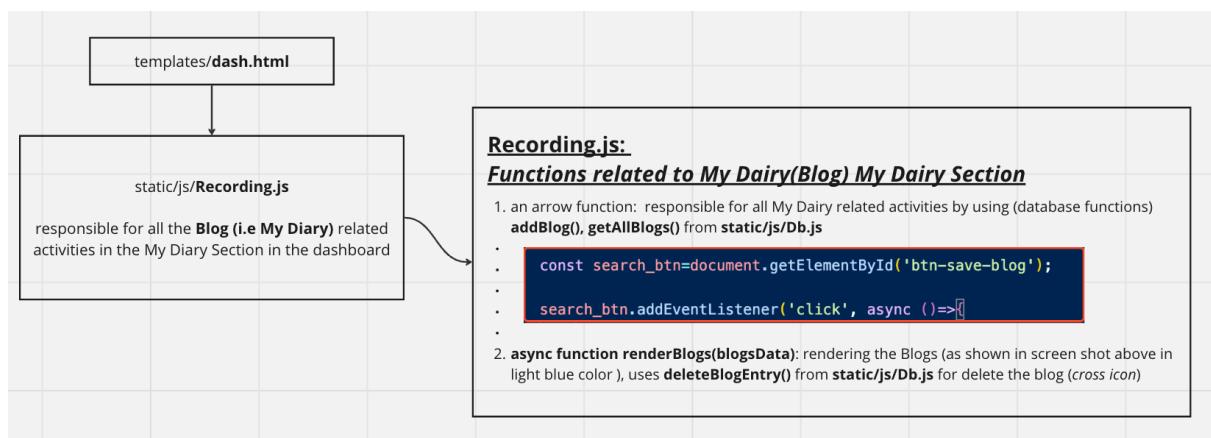
----- END OF Dashboard – Playlist -----

3. Dashboard – My Diary



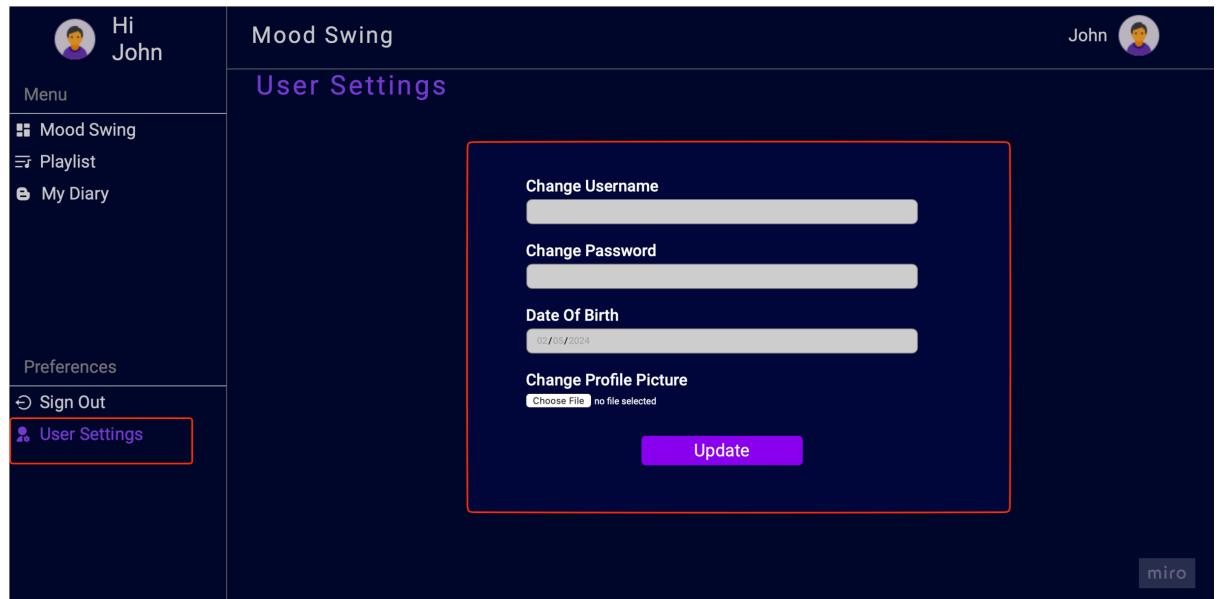
1. As Stated, dash.html file will use **static/js/Recording.js** as a script file.
2. This file is also have functions which is responsible for activities related to My Diary section in the dashboard (i.e add blog to firebase database, delete blog from firebase database, get all the blogs from the firebase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



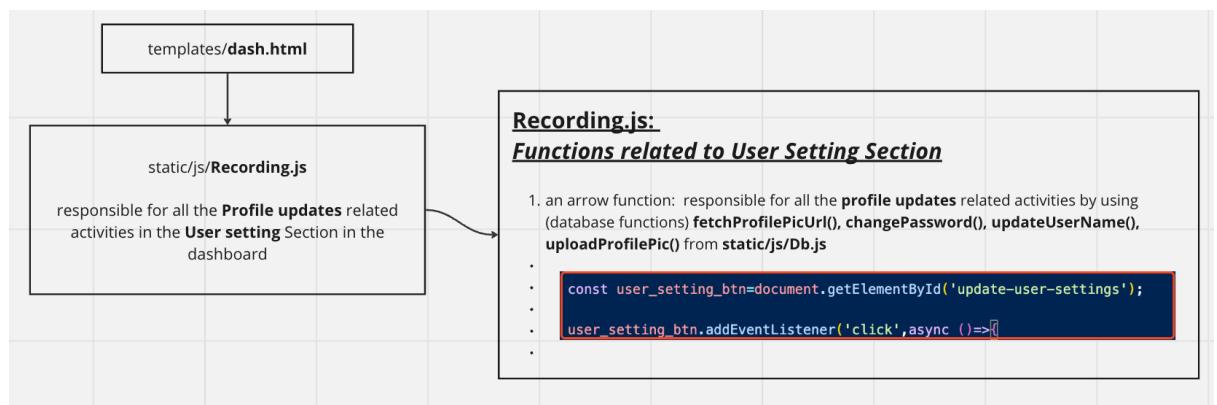
----- END OF Dashboard – My Diary -----

4. Dashboard – User Setting



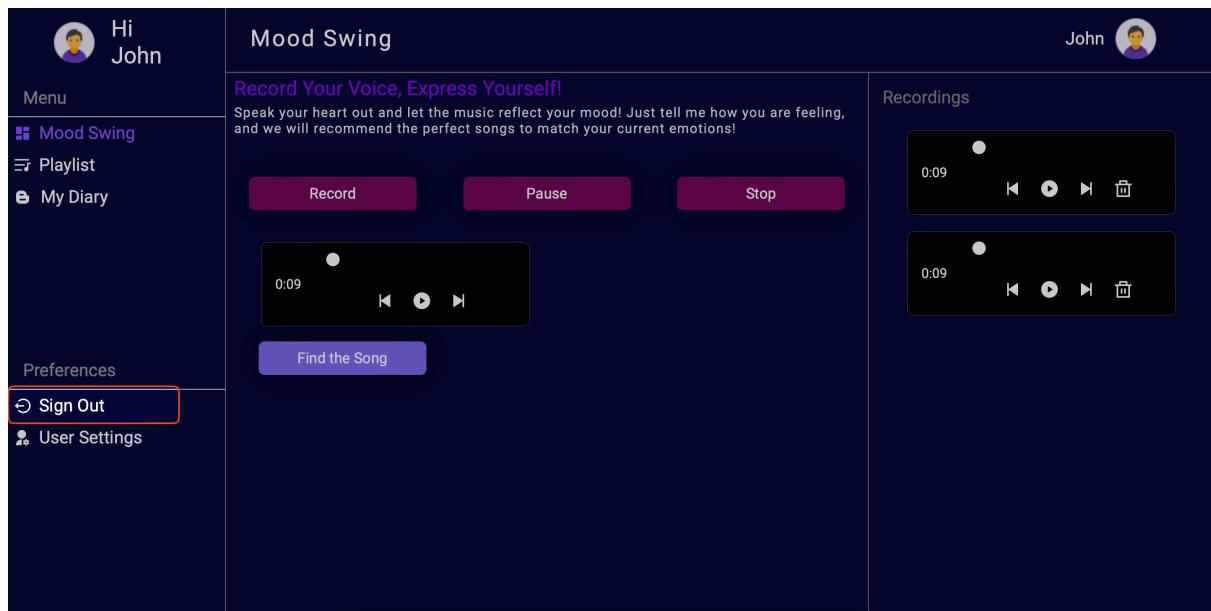
1. As Stated, dash.html file will use **static/js/Recording.js** as a script file.
2. This file is also have functions which is responsible for activities related to User Setting section in the dashboard (i.e change user name, change password, date of birth update, change profile picture).(refer the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

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



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

5. Dashboard – User Setting



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

```
document.getElementById('sign-out').addEventListener('click', ()=>{
  sessionStorage.setItem('Sentiment', null);
  localStorage.setItem('name', null);
  localStorage.setItem('User_id', null);
  window.location.href="/";
})
```

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

Appendix

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

Firebase is used for real time database, storage and authentication

Db.js Functions

```
function writeToDatabase(data)
export function saveNameToDatabase(userId, name)
export function fetchNameFromDatabase(userId)
export async function saveRecording(audioChunks,time)
export function addBlog( userId, content)
export async function getAllBlogs(userId)
export async function deleteBlogEntry(blogEntryId,userId)
export async function changePassword( newPassword)
export async function updateUserName(userId, newName)
export async function uploadProfilePic(userId, file)
export async function fetchProfilePicUrl(userId)
export async function getRecordingDownloadURL(uid, recordingId)
function readFromDatabase()
export function addToPlaylistToDatabase(previewUrl, mood, imageUrl,
songName)
export async function fetchAllMoods()
export async function fetchPlaylistsByMood(mood)
export async function fetchAllMoodsWithPlaylists()
export async function deletePlaylistFromDatabase(userId, playlistsMood,
playlistId)
function writeToRecordings(data)
export async function deleteRecording(userId, recordingId)
export async function fetchDownloadURLsWithIds(userId)

export {
auth,
createUserWithEmailAndPassword,
updateProfile,
signInWithEmailAndPassword,
writeToDatabase,
readFromDatabase,
writeToRecordings }
```