

**Tribhuvan University Faculty of Humanities and Social Science**

**A PROJECT REPORT ON**

**Mood Music Player**

**Submitted to**

**Department of Computer Application**

**Campus Name**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted by

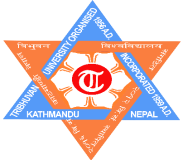
Name: (……….)

Roll no: 21

April-3, 2022

Under the Supervision of

**(……….)**

Supervisor’s certificate

**Tribhuvan University**

**Faculty of Humanities and Social Science**

**Birendra Multiple Campus**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by **(Name)** entitled “**MOOD MUSIC PLAYER”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

**……………..**

**SIGNATURE**

(……)

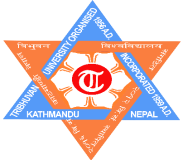
**SUPERVISOR**

Lecturer

Department of Computer Application

Faculty of Humanities and Social Science

Bharatpur-10, Chitwan.

Internal and External Examiner’s Approval

**Tribhuvan University**

**Faculty of Humanities and Social Science**

**College Name**

LETTER OF APPROVAL

This is to certify that this project prepared by (….) entitled “**Mood Music Player”** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **…….……….……….……….…….**  **SIGNATURE of Supervisor**  Mr. (….)  Department of Computer Application  Birendra Multiple Campus  Bharatpur-10, Chitwan. | **…….……….……….……….…………**  **SIGNATURE of HOD/ Coordinator**  Mr. (….)  Department of Computer Application  Birendra Multiple Campus  Bharatpur-10, Chitwan. |
| **….……….……….……….……………**  **SIGNATURE of Internal Examiner**  **Internal Examiner** | **….……….……….……….……………**  **SIGNATURE of External Examiner**  **External Examiner** |

Abstract

A Music Player site which play music on demand via any browser we can listen all sorts of music however, to adjust according to our mood we need to search ,even though there is another problem , we need to know the song name that we are going to listen. That’s where the “**Mood Music Player**” come on handy. Simply register and verify account, take a picture adjusting your face and listen song according to your mood and need more time to listen user will have to subscribe for monthly or yearly using E-Sewa.

Fun Fact, number of songs that are recorded and published is fairly large amount, a user cannot imagine. i.e. No need to search song, no need to remember song in your head a WebApp will manage your mood choice.

Simple Process, with interactive user interface viewing details and listening songs with in one clicks.

All the songs, account and subscription plan are handle via admin using admin panel. Furthermore, User have a choice to review a songs where reviews will further analyzed as the song is the exact match of songs or not.

Acknowledgment

I would like to express my special thanks to Birendra Multiple Campus allowing me to having this project .Moreover, to express my special thanks of gratitude to our supervisor Mr (….) who gave me the golden opportunity to do this wonderful project on the topic **Mood Music Player**, which also helped me in doing a lot of research and I came to know about so many new tools and technologies.

Furthermore, I would like to thank my family who provided me all the resources required to make this project.

Table of content

Contents

[1. Introduction 1](#_Toc102549554)

[1.1 Introduction 1](#_Toc102549555)

[1.2 Problem Statement 1](#_Toc102549556)

[1.3 Objectives 2](#_Toc102549557)

[1.4 Scope and Limitations 2](#_Toc102549558)

[1.4.1 Scope 2](#_Toc102549559)

[1.4.2 Limitations 2](#_Toc102549560)

[1.5 Report Organization 3](#_Toc102549561)

[2. Background Study and Literature Review 4](#_Toc102549562)

[2.1 Background Study 4](#_Toc102549563)

[2.2 Literature Review 4](#_Toc102549564)

[3. System Analysis and Design 5](#_Toc102549565)

[3.1 System Analysis 5](#_Toc102549566)

[3.1.1 Requirement Analysis 5](#_Toc102549567)

[3.1.2 Feasibility Analysis 7](#_Toc102549568)

[3.1.3 Data Modelling (ER-Diagram) 10](#_Toc102549569)

[3.1.4 Process modelling (DFD) 11](#_Toc102549570)

[3.2 System Design 13](#_Toc102549571)

[3.2.1 Architectural Design 14](#_Toc102549572)

[3.2.2 Database Schema Design 14](#_Toc102549573)

[3.2.3 Interface Design (UI Interface / Interface Structure Diagrams) 18](#_Toc102549574)

[3.2.4 Physical DFD 30](#_Toc102549575)

[4. Implementation and Testing 32](#_Toc102549576)

[4.1 Implementation 32](#_Toc102549577)

[4.1.1 Tools Used (CASE tools, Programming languages, Database platforms) 32](#_Toc102549578)

[4.1.2 Implementation Details of Modules (Description of procedures/functions) 32](#_Toc102549579)

[4.2 Testing 33](#_Toc102549580)

[4.2.1 Test Cases for Unit Testing 33](#_Toc102549581)

[4.2.2 Test Cases for System Testing 33](#_Toc102549582)

[5. Conclusion 40](#_Toc102549583)

[5.1 Lesson Learnt / Outcome 40](#_Toc102549584)

[5.2 Conclusion 40](#_Toc102549585)

[5.3 Future Recommendations 40](#_Toc102549586)

List of figures

[Figure 3‑1 Use-Case Diagram Mood Music Player 6](#_Toc102549587)

[Figure 3‑2 Gantt chart 9](#_Toc102549588)

[Figure 3‑3 ER-Diagram of Mood Music Player 10](#_Toc102549589)

[Figure 3‑4 DFD Level-0 of Mood Music Player 11](#_Toc102549590)

[Figure 3‑5 DFD Level-1 Mood Music Player 12](#_Toc102549591)

[Figure 3‑6 System-Design of Mood Music Player 14](#_Toc102549592)

[Figure 3‑7 User Acc’s / Songs Data / User Reviews Database 15](#_Toc102549593)

[Figure 3‑10 Login 18](#_Toc102549594)

[Figure 3‑11 Admin Data Analysis (Dashboard) 18](#_Toc102549595)

[Figure 3‑12 Add Songs 19](#_Toc102549596)

[Figure 3‑13 List of Songs and Delete songs 19](#_Toc102549597)

[Figure 3‑14 Edit Songs 20](#_Toc102549598)

[Figure 3‑19 User Register Form 21](#_Toc102549599)

[Figure 3‑20 User login Interface 21](#_Toc102549600)

[Figure 3‑21 Register Email confirmation and verification 22](#_Toc102549601)

[Figure 3‑22 User Interface without and with subscription 23](#_Toc102549602)

[Figure 3‑23 User Interface sending picture and list of songs which relevant to emotions 24](#_Toc102549603)

[Figure 3‑24 Songs played without and with subscription 25](#_Toc102549604)

[Figure 3‑25 Message for subscription monthly / yearly after minutes listening 26](#_Toc102549605)

[Figure 3‑26 List of Songs 26](#_Toc102549606)

[Figure 3‑27 Play single song and add reviews 27](#_Toc102549607)

[Figure 3‑28 User Reset Password 28](#_Toc102549608)

[Figure 3‑29 Password Reset 29](#_Toc102549609)

[Figure 3‑34 Physical DFD – Admin Panel 30](#_Toc102549610)

[Figure 3‑35 Physical DFD-User Interaction 31](#_Toc102549611)

List of tables

[Table 3‑1 User-Data Table 16](#_Toc102549612)

[Table 3‑2 Songs Name Table 16](#_Toc102549613)

[Table 3‑3 User Reviews Table 17](#_Toc102549614)

[Table 3‑4 Payments Details Table 17](#_Toc102549615)

[Table 3‑5 Check\_analysis\_in\_chart Table 17](#_Toc102549616)

[Table 4‑1 login test 34](#_Toc102549617)

[Table 4‑2 User Password reset via email 35](#_Toc102549618)

[Table 4‑3 User Activation Email 36](#_Toc102549619)

[Table 4‑4 Registration of User with valid email 37](#_Toc102549620)

[Table 4‑5 User Send Image and get the list of songs 38](#_Toc102549621)

[Table 4‑6 Valid Reviews 39](#_Toc102549622)

# Introduction

## Introduction

A Music Player application play a music with handful of playlist. Web application will generate a list of music and play according to user choice.

Mood Music Player is a very handy WebApp where user will capture photo via webcam of computer or android device from using familiar browser and will get the music playlist according to the mood .Furthermore, A user can review the songs that is exactly match or not.

Every Person doesn’t have a time to search music in internet with their mood. Every second new music hit market and get underrated. However, this Web Application idea is to capture a photo with face and give a list of songs which drastically save time. Although, the application need to subscribe for more time to listen and payment is done via E-Sewa as monthly and yearly.

The primary goal of this site is to provide music with mood. Firstly the admin will add songs details with category emotions. Such as happy, sad, angry, surprise, fear, disgust. Secondly, the user will login and upload a picture of current state and then detecting a mood by system it will refer a list of songs.

General speaking, Music player with emotion will take a photo and generate a playlist where user can hear with ease and also post review.

## Problem Statement

This Implementation will have several problem. Analyzing various site there are a lot of Music Player and face emotion detection webapp such as ganna.com, Spotify.com and Emotion detector app, morphcast.com respectively they do have a lot of features searching songs, reviewing it and detection gender, age and many more. However, this feature aren’t integrated to play music implementing a feature like taking picture via browser sending image to server and get relevant mood music.

## Objectives

This project will be conducted to fulfill following major objectives.

## Scope and Limitations

* It shows the correct category and enable the listener to list a song and play songs according to their mood that are available in server.
* To increase efficiency and improve services provided to the customers through better application of technology.
* It is designed to decrease time of user for searching music as per current mood.

### Scope

* Understand and prepare detailed requirement and specifications
* Prepare detailed design specifications of the system
* Develop the system and coding
* Demonstrate a bug free application after suitable modification if needed

### Limitations

* The device must have webcam to take photo.
* The device must contain Speaker.

## Report Organization

Reading the report from introduction to conclusion is quite tedious task therefore, I came forward to write in short and yet fast way to describe from designing, analysis with proper implementation and lastly conclusion.

Firstly, for System Analysis and Design gone thoroughly from what kind of approach and design I have used by showing via screenshot and Table. Moreover, the user register and login using an easy step and also the requirement that website need to run is of minimal cost. There are also the structure of database design which will help system admin to see how data will be passed from one component to another which is very handy.

Secondly, In case of Implementing and testing while designing the ER, DFD and other figure I mostly used drawsql.app and visual-paradigm tool which was an online tool that will provide a free drawing using Browser with easy interface. There was two phase unit and system testing, where unit testing was conducted for three module for admin while inserting data, User module consist of getting account by verifying valid email with valid token. After successful login getting songs details with cover picture, and many other details including reviews of users, the profile module where user can upload profile picture and change password. If some user tend to forgot there password then they can reset password requesting new from email. Finally, the payment module where user will pay subscription plan as per desire monthly/yearly. Which is later forwarded to admin to verify the details before hand over the materials.

Finally, conclusion with how this website could change the perspective of listening a songs with button of click as per emotions. Just by taking a picture and get the list of songs as per emotions .How this come handy in first place and what kind feature this website lack which can be added later so, the website will function better than previous version.

# Background Study and Literature Review

## Background Study

The main objective of “**Mood Music Player**” is to provide an essence of playing music via a simple and yet powerful medium.

Mood Music Player allows you to browse and listen music through endless possibilities, and even offers list of music according to your mood.

What's even more useful from this website is the ability to read reviews which is written by a valid Listener in first place. The reviews will help to calculate the music which is relevant to mood which is listed in emotion based list which is quite helpful for user. You can search songs through artist name, song name and add reviews simultaneously.

## Literature Review

There will be a lot of websites if we google Music Player for all around the globe. However, what interesting is the other website does play music which does have a lots of features. What makes this web application different is taking a photo sending to server and getting songs without any hassle.

Furthermore, while doing some small research in my own university and other couple of store and going through it I came into conclusion mostly the work is done manually means searching songs remembering it which will eventually consume time what if user doesn’t have ample time, here comes the reviews based and emotion base songs.

For Instance, Firstly, Logged in after registration, add a photo capturing via webcam check whether the feature is good or not as trail .If the user experience is good then pay via E-Sewa. A simple step but a Crucial result.

# **System Analysis and Design**

## System Analysis

I used the Waterfall methodology while building this application. This project had fixed specification, ample time and enough resources so Waterfall methodology was used to build this system.

I was working with a fixed requirement so i created fixed milestones and deadlines. Waterfall methodology fitted perfectly with those requirements.

### Requirement Analysis

It is the process of defining the expectations of the users for an application that is to be built or modified. It needed to fulfill functional and non-functional requirement.

#### Functional Requirement

Functional requirements define the capabilities and functions that a system must be able to perform successfully. The functional requirements of this Mood Music Player ordering system include [1].

* The system will enable the admin to add songs and songs details with category.
* The system will enable admin to see the payment details and admin can also BAN user.
* The system will enable the User to take photo and upload and get the list of songs that matches the mood.
* Keeping records of registration of customers.
* The system will display the List of songs and its reviews.
* The user can add reviews in songs.
* The system will provide customer to add profile and change password via email confirmation and many more.
* The System will generate a report base of top emotions relevant songs validating reviews.
* The System will give a trail version for week (for testing purposes now it’s for 8 second).

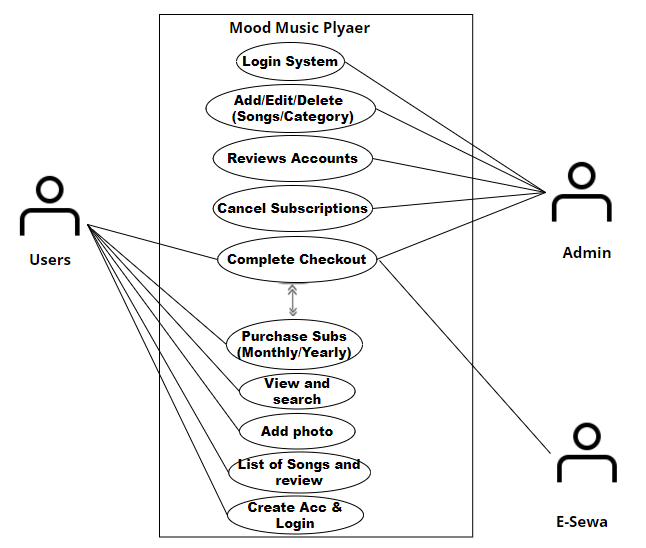


Figure 3‑1 Use-Case Diagram Mood Music Player

#### Non Functional Requirement

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Some of the non-functional requirements include [1]:

* Availability: It will be available online whenever the client needed.
* Performance: It will be fast with good response if the face is cleared. A Single Page App. Where the website will not load even the user change fields which make site faster and better in performance.
* Security: webcam photo will not be publicly available. Furthermore, to get rid of common vulnerability like XSS and cookie hijacking there is HttpOnlyCookie which makes difficult to hijack or get authorized token.
* Usability: Good GUI easy to use.

### Feasibility Analysis

A feasibility study is a test of a system proposal. According to its workability, impacts on the organization, ability to meet user needs and effective use of the resources its main task done during feasibility study are [2].

#### **Technical Feasibility**

This project will be focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the needs of the proposed system. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

**Software Requirements:**

Operating system: Windows or Linux (with Python 3.10.2)

Server: WSGI

Front-end: React

Back-end: Django (Server side code) and My SQLite (Database)

**Hardware requirements:**

Processor: Intel dual core or above

Hardware: Webcam, Speaker

Processor Speed: 1.0GHZ or above

RAM: 4 GB RAM or above

Hard Disk: 50 GB hard disk or above

#### **Economic Feasibility**

Development of this application is highly economically feasible. The organization need not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development.

#### **Operational Feasibility**

The system is operational feasible since the user are familiar with the technologies and hence there is no need to gear up the personnel to use the system. Also the system is very user friendly and easy to use

* Front-end

1. React
2. CSS

* Server

1. Django 4.0.0 ( Python 3.10.2)
2. WSGI Server

* Back end

1. My SQLite
2. Python (3.10.2)

* Framework

1. Bootstrap Framework
2. Material UI

#### **Schedule Feasibility**

Proposed system would be designed as per the time calculated. Likelihood that timeframes can be met and that this adequate to meet organization's needs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Time**  **Phase** | **Jan** | **Feb** | **Mar-Apr** | **May** |
| **Study and analysis** | **1 week** | **1 week** |  |  |
| **Data Fetching and Scrapping** | **1 week** |  |  |  |
| **Coding and Implementation** |  |  | **1 week** |  |
| **Testing** |  |  | **2 week** |  |
| **Documentation** |  |  | **3 week** |  |
| **Reviews** |  |  |  |  |
| **Presentation** |  |  |  | **9 th** |

Figure 3‑2 Gantt chart

### Data Modelling (ER-Diagram)

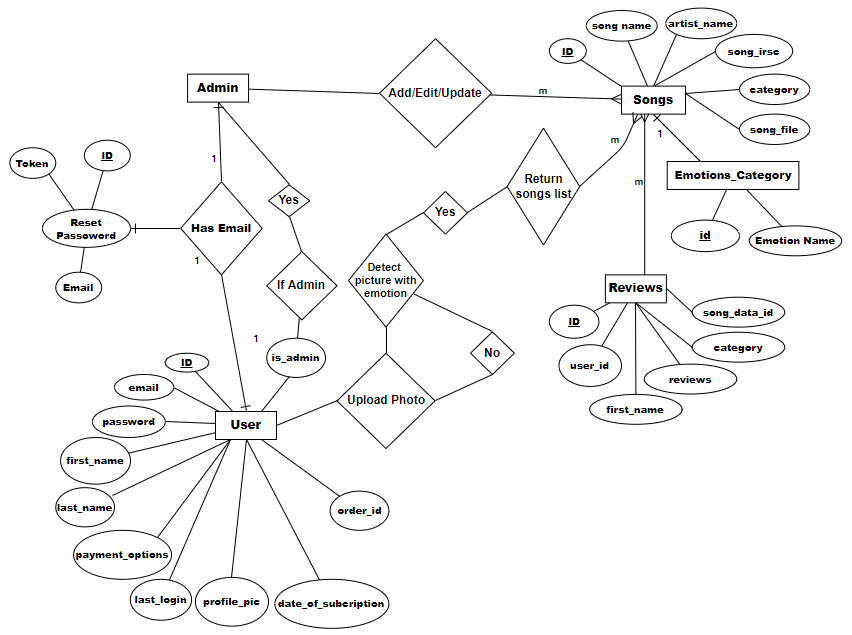


Figure 3‑3 ER-Diagram of Mood Music Player

### Process modelling (DFD)

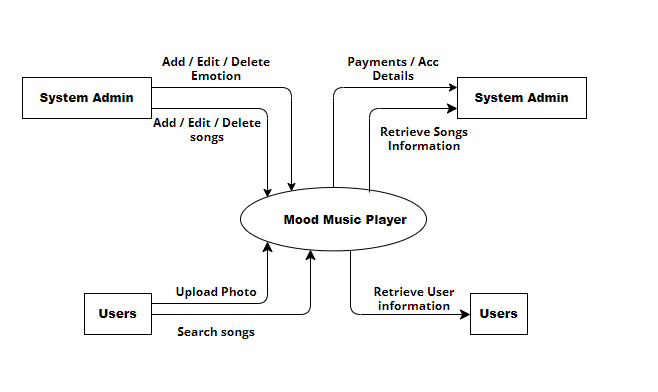
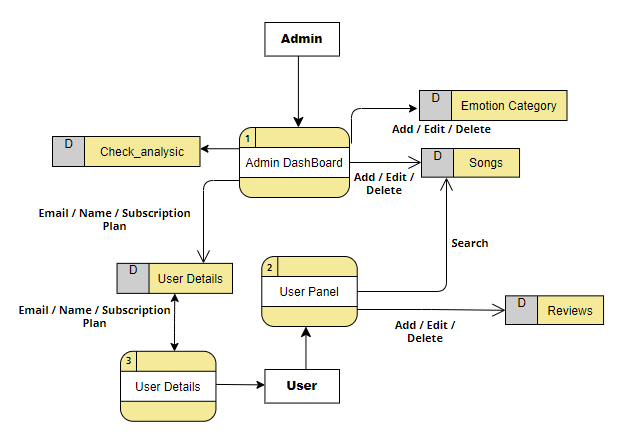


Figure 3‑4 DFD Level-0 of Mood Music Player

Figure 3‑5 DFD Level-1 Mood Music Player



## System Design

System design represent the overall concept on how the system works. In our system there are two modules. They are system module and admin module [3].

**User Module:**

* Only register user can login into the application.
* Those users who are not register must register first by filling the necessary attributes such as name, email, password with verification etc. and then the account will be created by providing user ,email ,name and password.
* User can listen and read the songs and songs name, artist name, cover picture respectively.
* User can also review Songs which will be filter by system and can read Reviews of other users.
* User can use subscription package for monthly and yearly and can listen unlimited for time.

**Admin Module:**

* Admin can Add, Edit and update Songs with cover picture using ISRC number.
* System will automatic take down reviews if they are didn’t have meaning.
* Admin can verify Client who subscribed with all details such as Email, name, days remaining of subscription to end.
* Admin Can revoke Ban user and cancel the subscription.
* The system will analyze the emotion requested by user and generate a report to Admin.
* The system will generate list of relevant emotions base songs analyzing reviews from users.

### Architectural Design

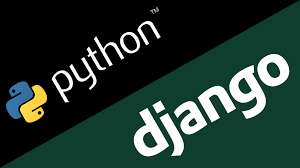
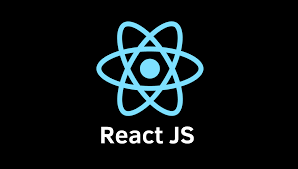


Figure 3‑6 System-Design of Mood Music Player

### Database Schema Design

The purpose of database is to handle information as combined whole system using not repeated technique. A database is a collection of inter-related data stored with minimum redundancy to serve single users quickly and efficiently. The general objective is to make information necessary, quick, inexpensive and flexible for the user.

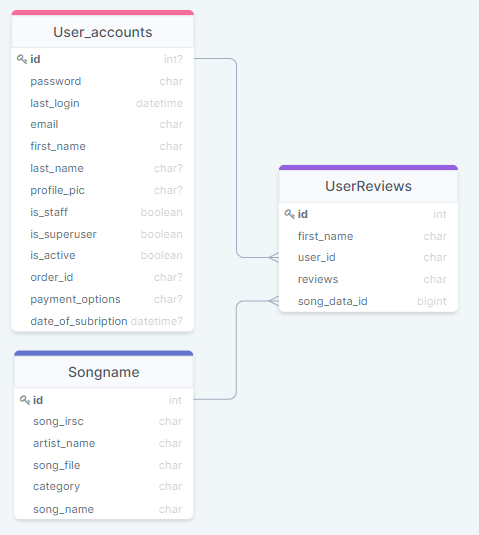


Figure 3‑7 User Acc’s / Songs Data / User Reviews Database

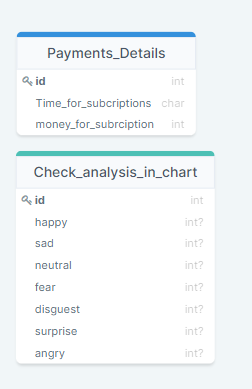


Figure 3‑8 Payment Details / Analysis Database

#### Database Tables:

Table 3‑1 User-Data Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Allow NULL | Constrain |
| ID | Int | 6 | Not | PK |
| first\_name | Varchar | 30 | Not |  |
| last\_name | Varchar | 30 | Not |  |
| email | Varchar | 10 | Not |  |
| password | Varchar | 20 | Not |  |
| profile\_pic | Varchar | 30 | Yes |  |
| last\_login | datetime | 30 | Not |  |
| is\_staff | boolean |  | Not |  |
| is\_superuser | boolean |  | Not |  |
| is\_active | boolean |  | Not |  |
| order\_id | Varchar | 30 | Yes |  |
| payment\_options | Varchar | 30 | Yes |  |
| date\_of\_subcription | datetime |  | Yes |  |

Table 3‑2 Songs Name Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Allow NULL | Constrain |
| ID | Int | 6 | Not | PK |
| artist\_name | Varchar | 50 | Not |  |
| song\_file | Varchar | 50 | Not |  |
| category | Varchar | 30 | Not |  |
| song\_name | Varchar | 100 | Not |  |

Table 3‑3 User Reviews Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Allow NULL | Constrain |
| ID | Int | 100 | Not | PK |
| first\_name | Varchar | 100 | Not |  |
| user\_id | Varchar | 300 | Yes |  |
| reviews | Varchar | 300 | Yes |  |
| song\_data\_id | bigint | 100 | Not |  |

Table 3‑4 Payments Details Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Allow NULL | Constrain |
| Id | Int | 6 | Not | PK |
| Time\_for\_subcriptions | Varchar | 30 | Not |  |
| Money\_for\_subcription | Varchar | 30 | Not |  |

Table 3‑5 Check\_analysis\_in\_chart Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Allow NULL | Constrain |
| Id | Int | 6 | Not | PK |
| happy | Varchar | 30 | Yes |  |
| sad | Varchar | 30 | Yes |  |
| neutral | Varchar | 30 | Yes |  |
| fear | Varchar | 30 | Yes |  |
| disguest | Varchar | 30 | Yes |  |
| surprise | Varchar | 30 | Yes |  |
| angry | Varchar | 30 | Yes |  |

### Interface Design (UI Interface / Interface Structure Diagrams)

All the interface used in my project.

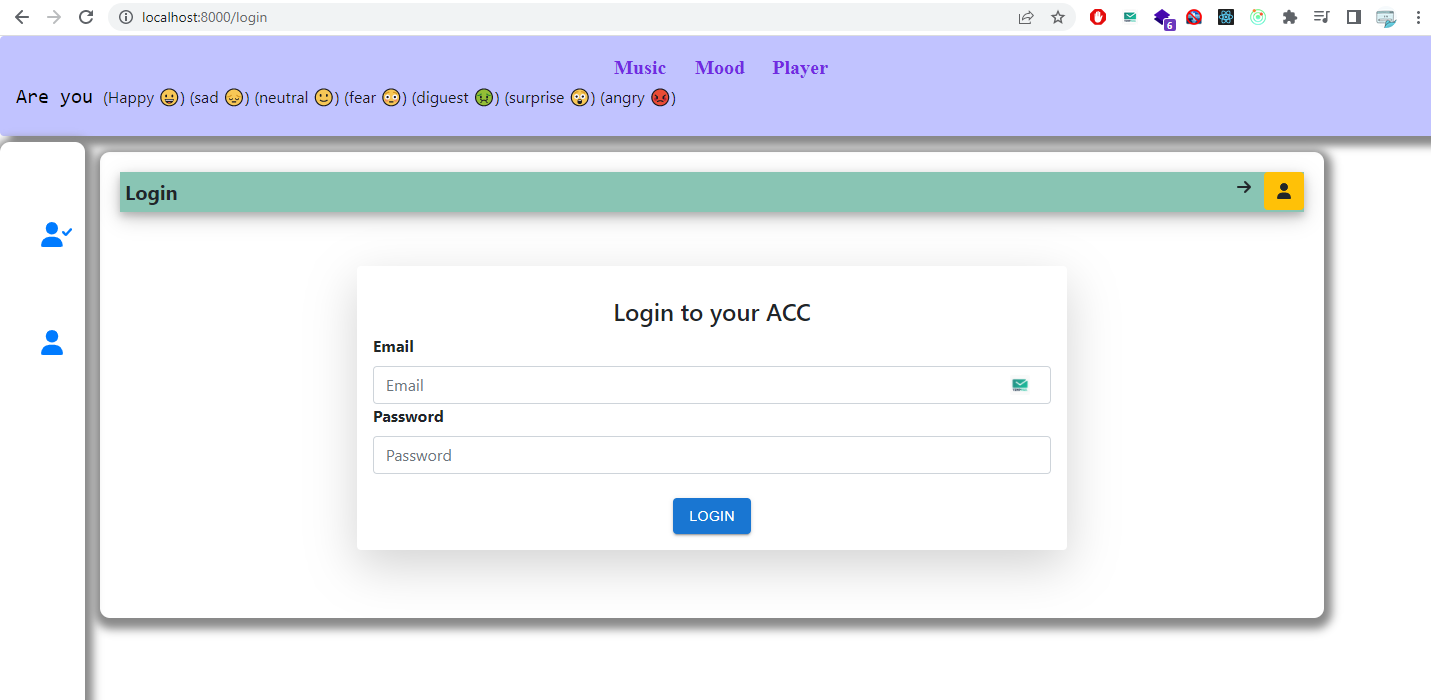


Figure 3‑10 Login

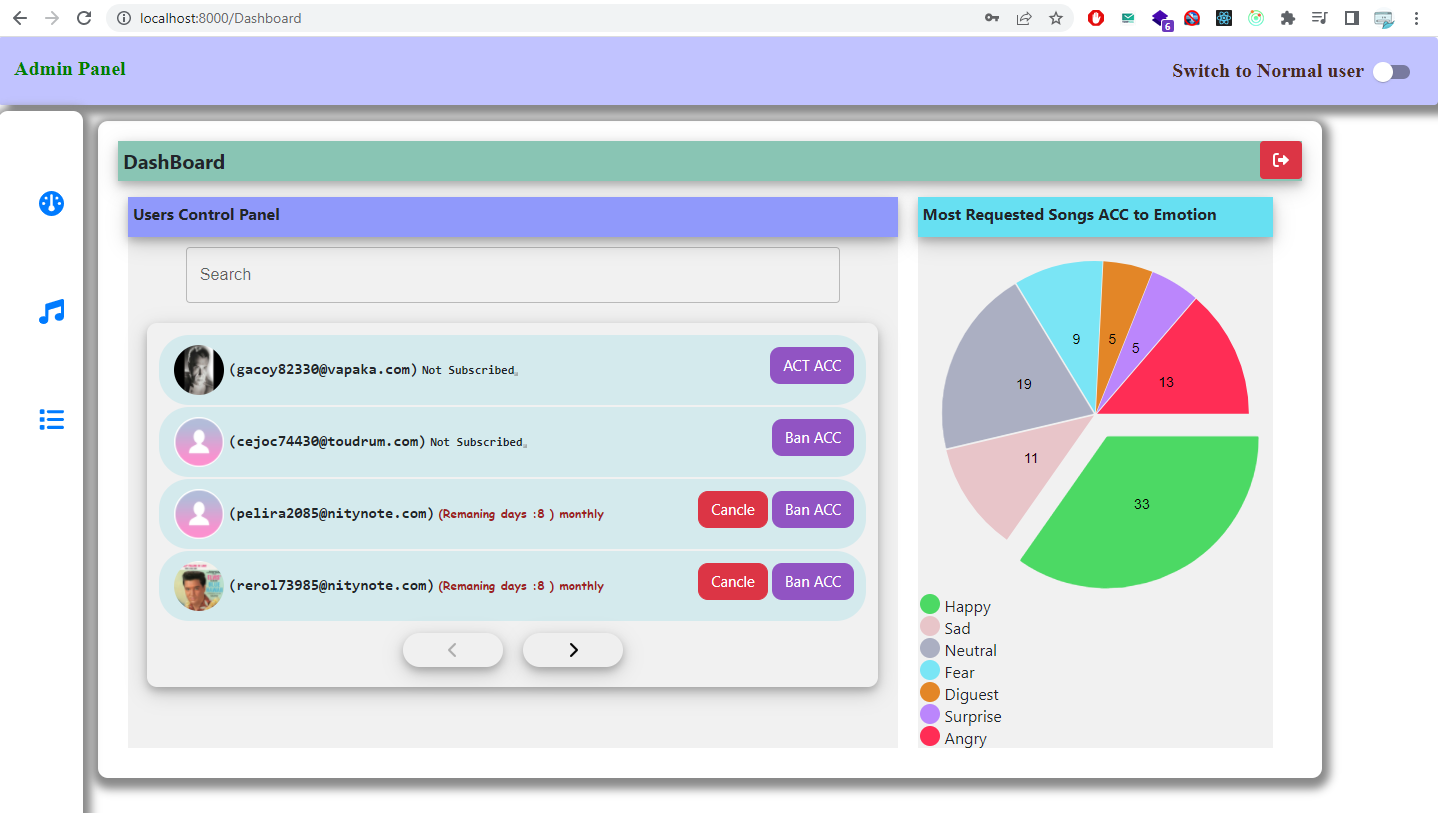


Figure 3‑11 Admin Data Analysis (Dashboard)

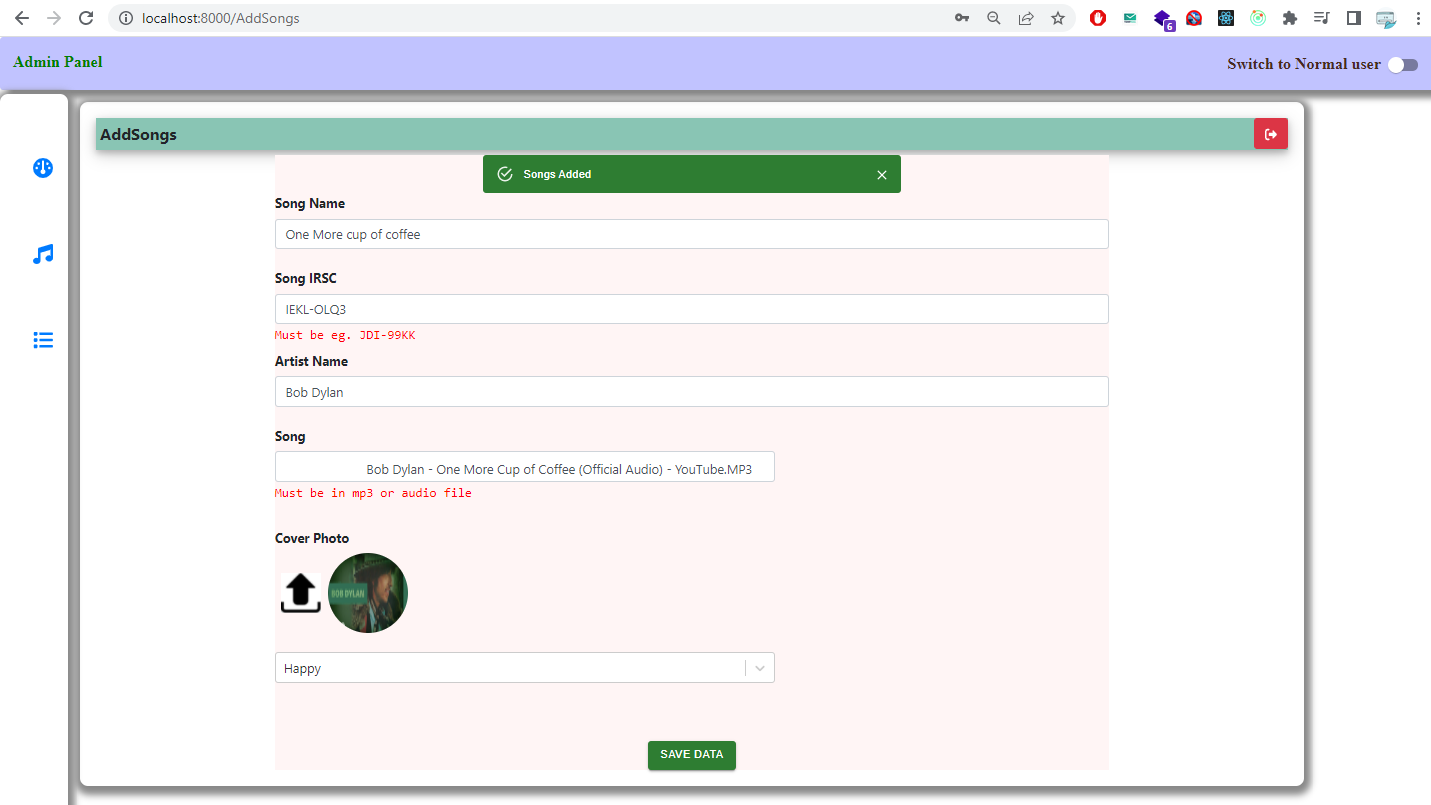


Figure 3‑12 Add Songs

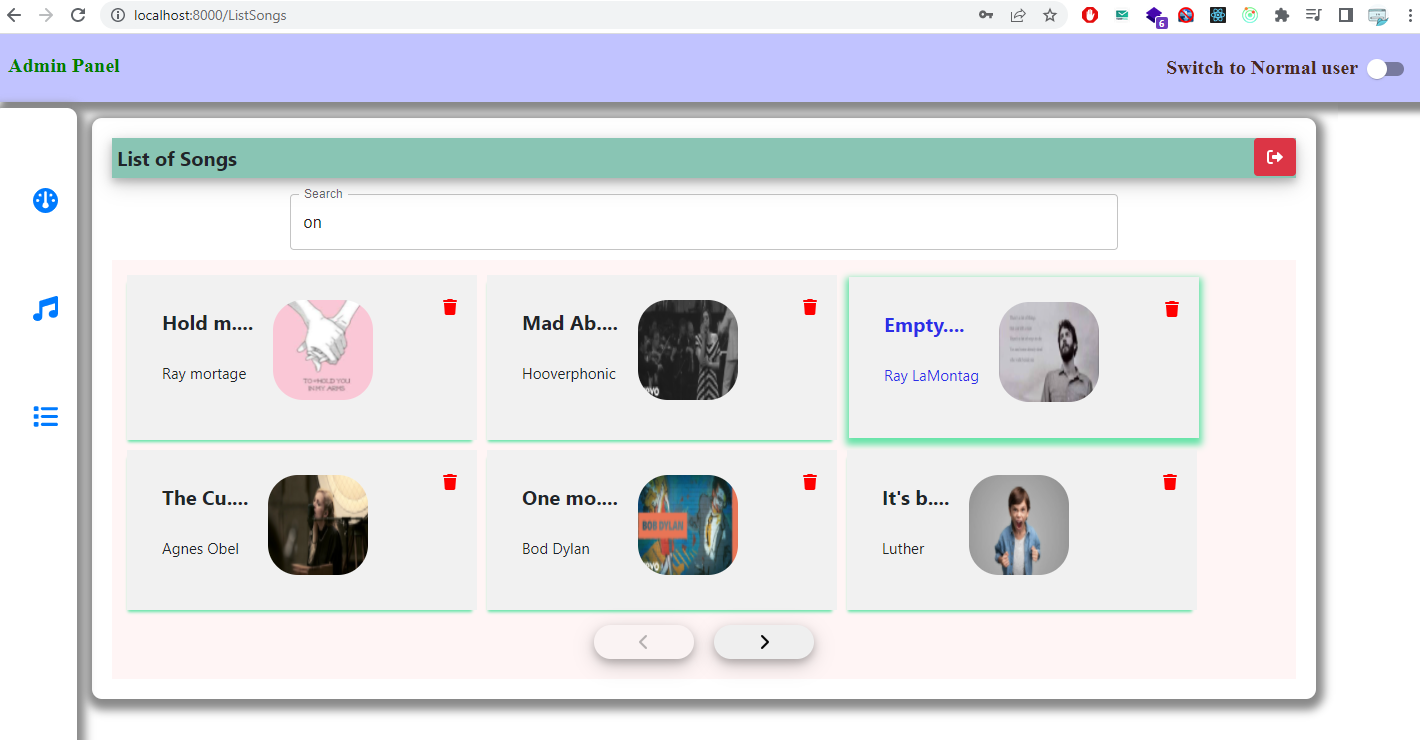


Figure 3‑13 List of Songs and Delete songs

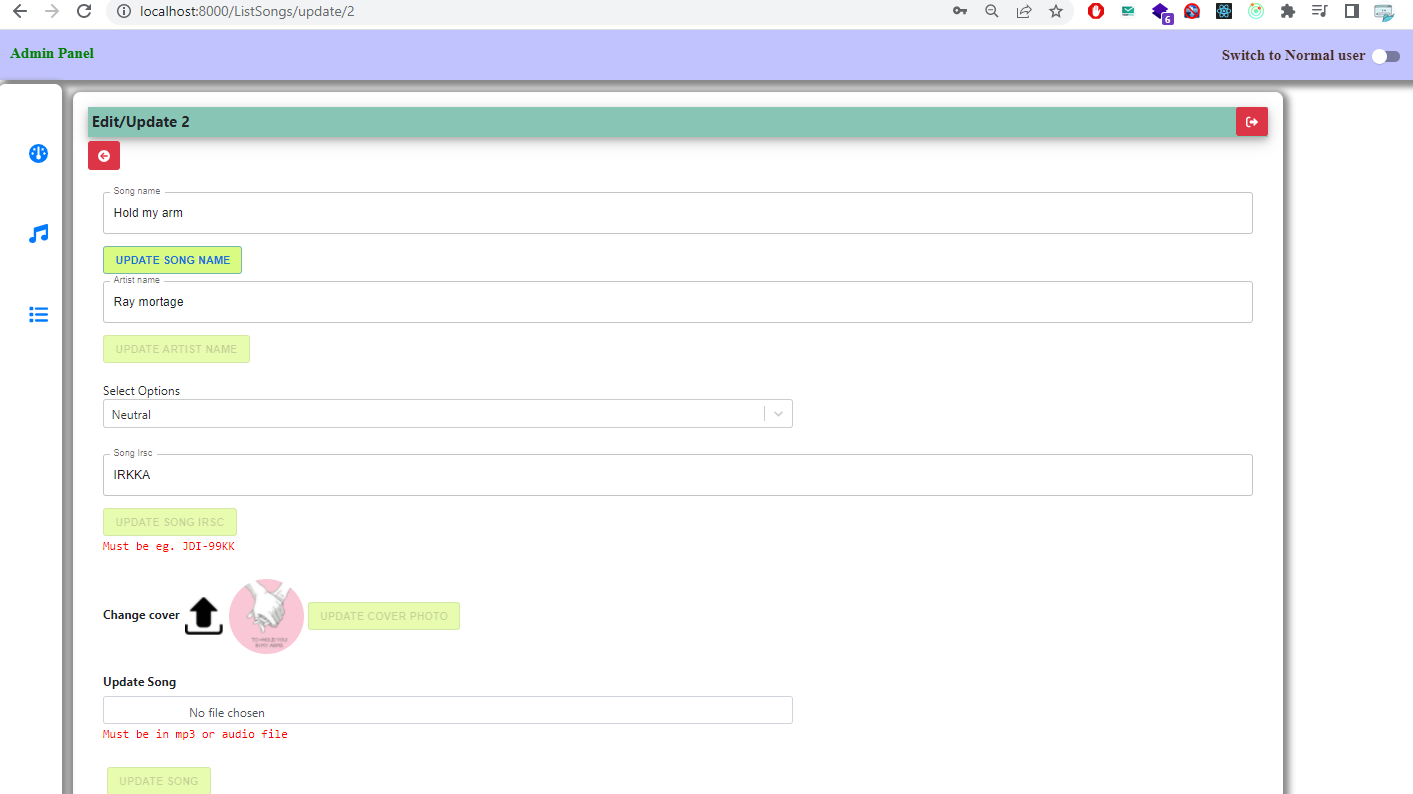


Figure 3‑14 Edit Songs

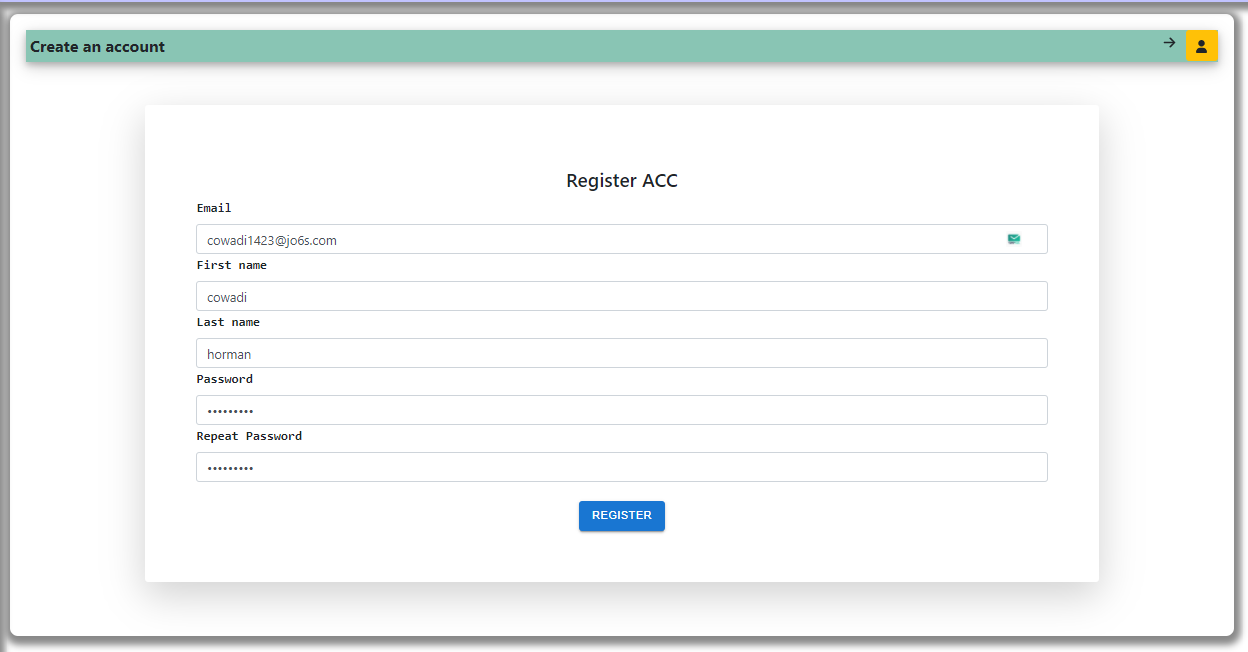


Figure 3‑19 User Register Form

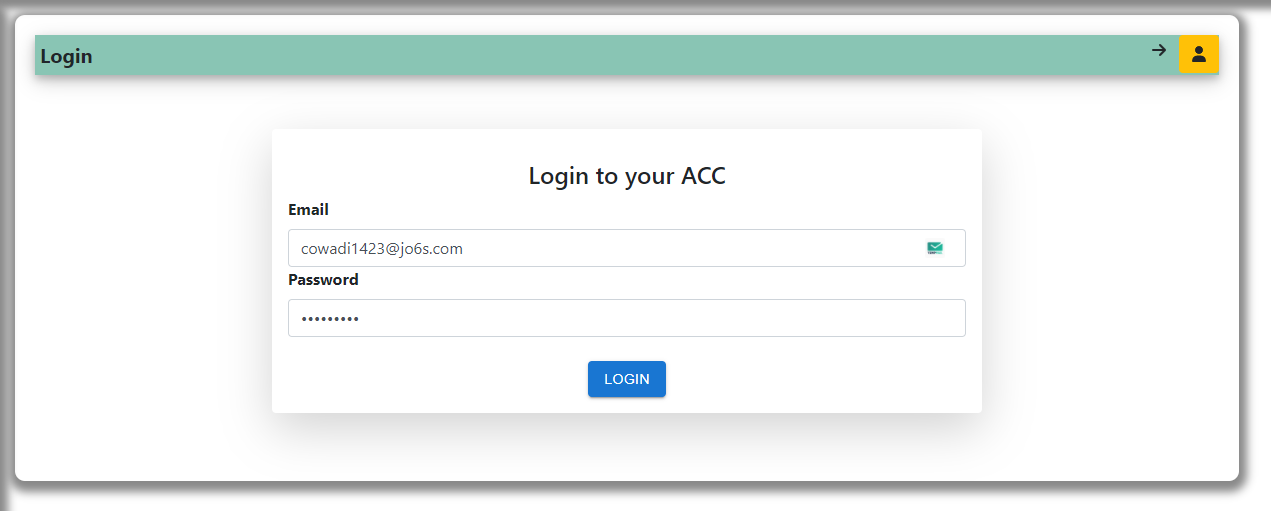


Figure 3‑20 User login Interface

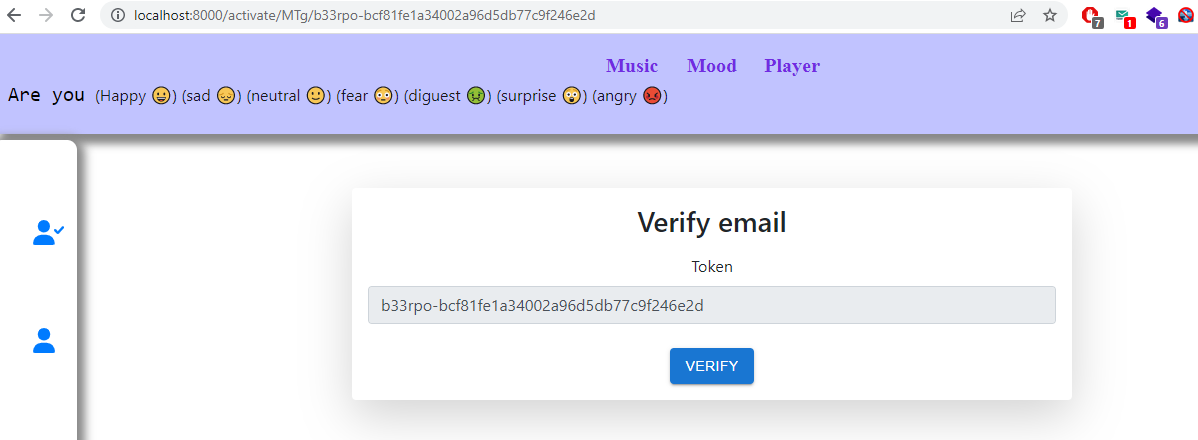
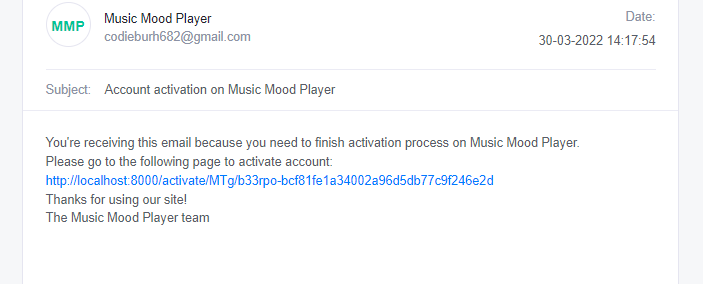


Figure 3‑21 Register Email confirmation and verification

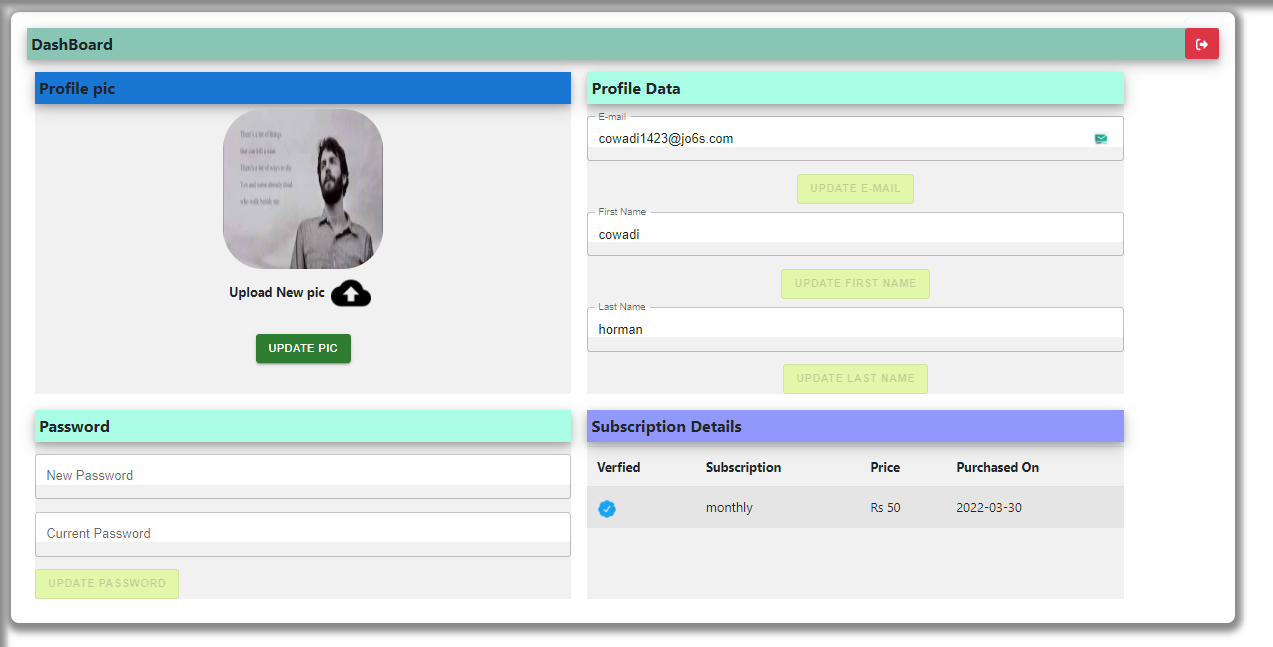
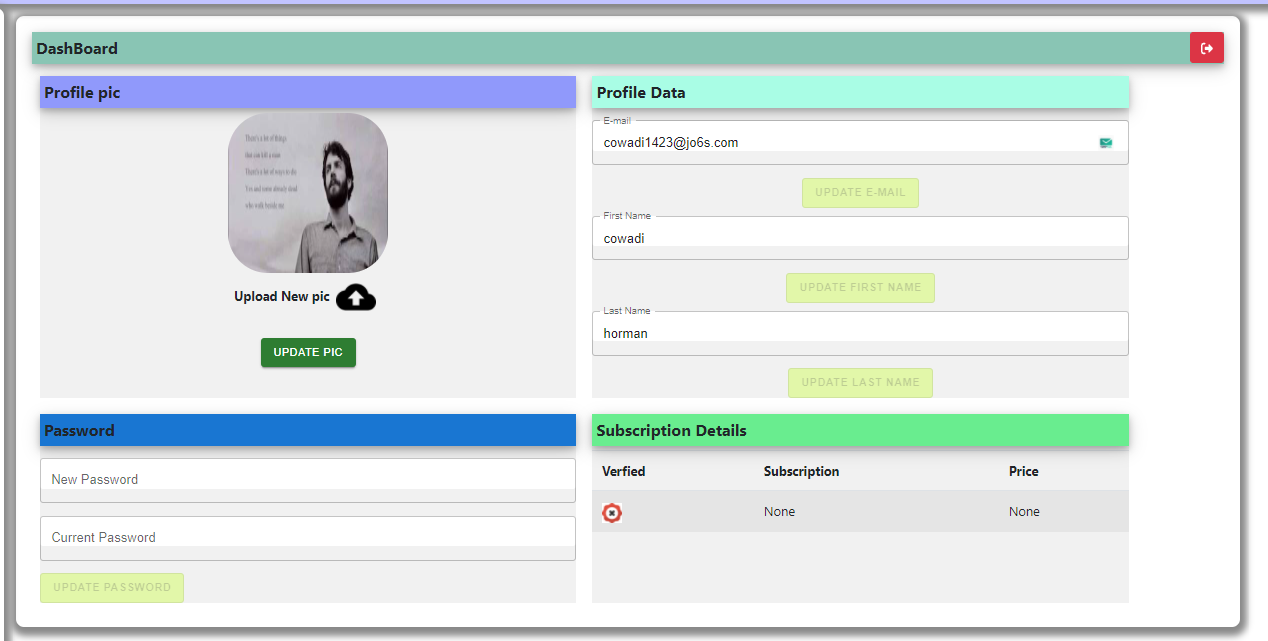


Figure 3‑22 User Interface without and with subscription



Figure 3‑23 User Interface sending picture and list of songs which relevant to emotions

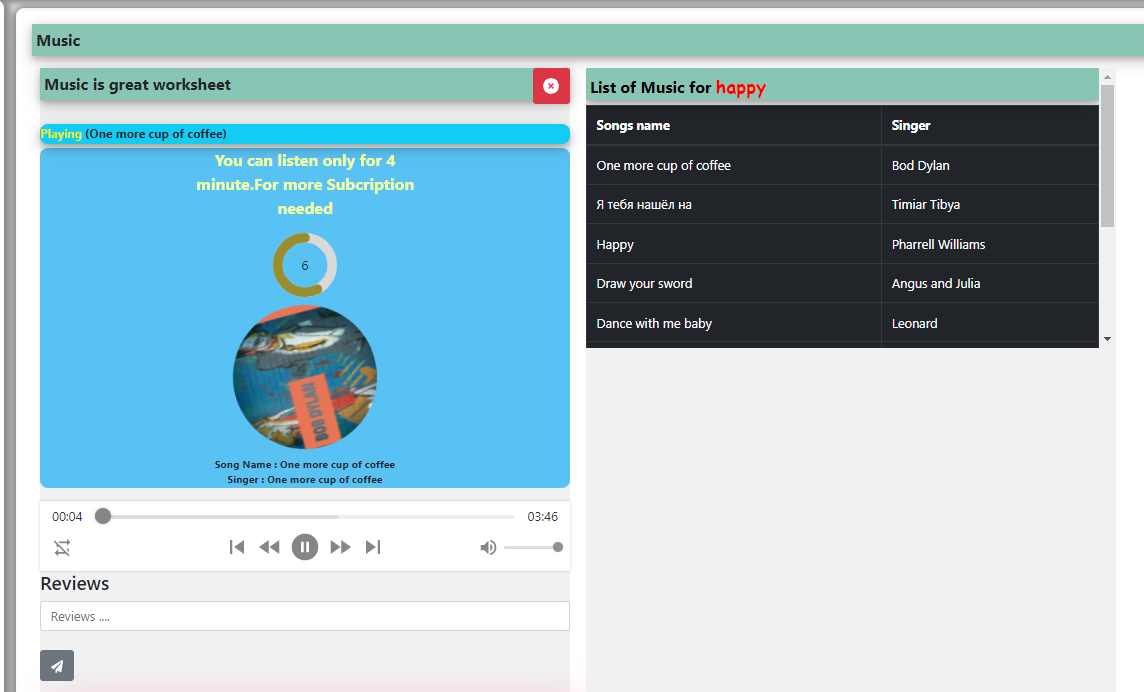
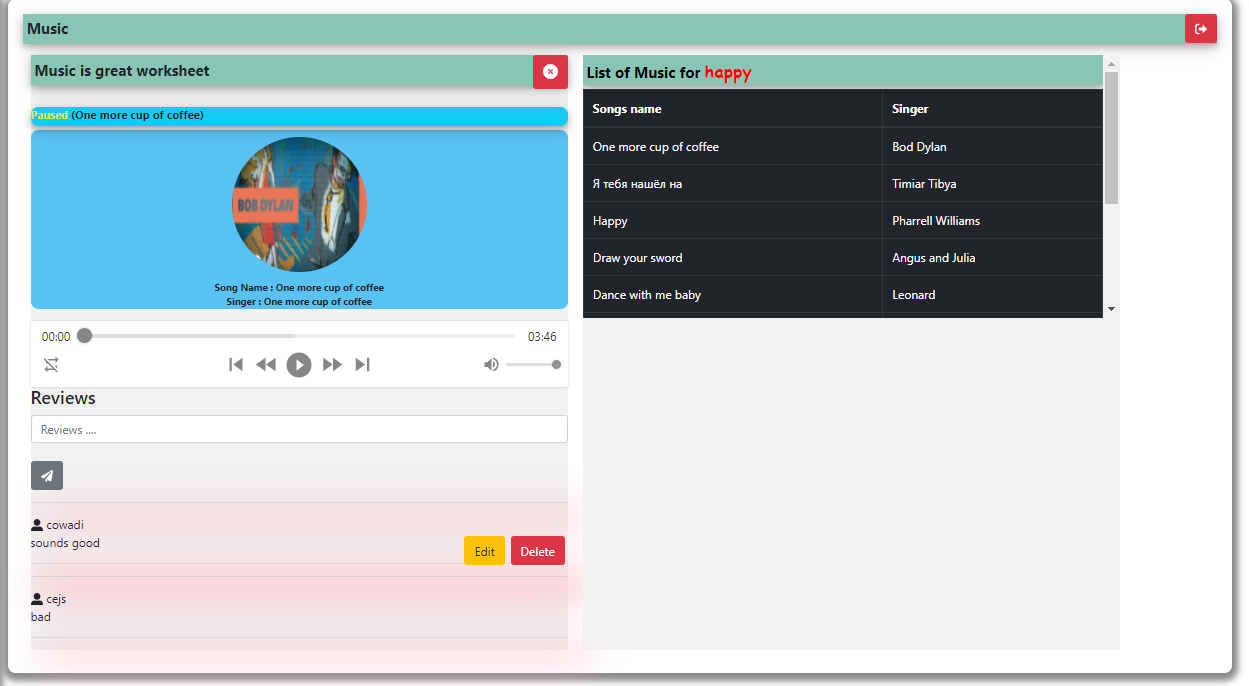


Figure 3‑24 Songs played without and with subscription

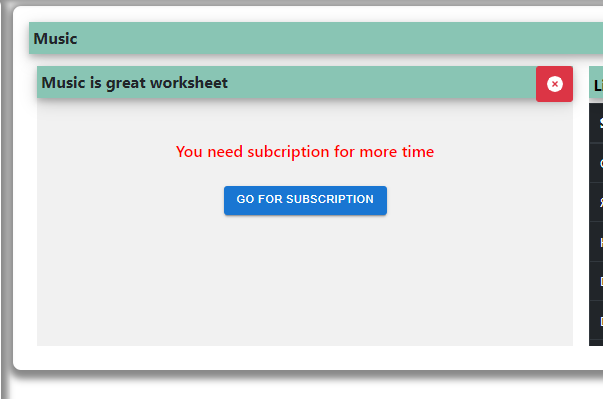


Figure 3‑25 Message for subscription monthly / yearly after minutes listening

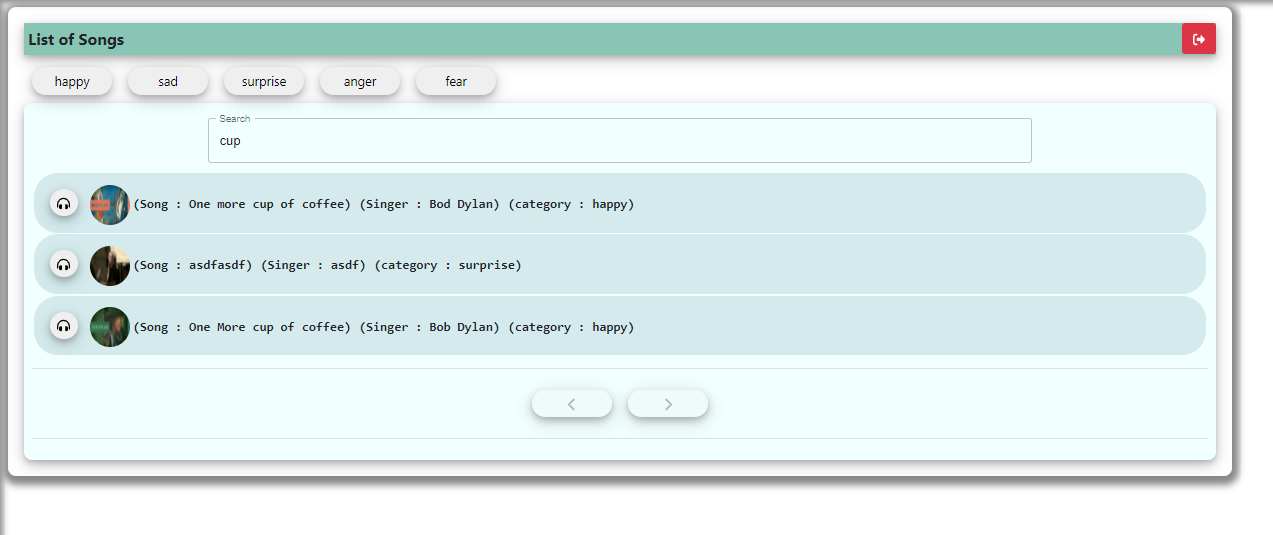


Figure 3‑26 List of Songs

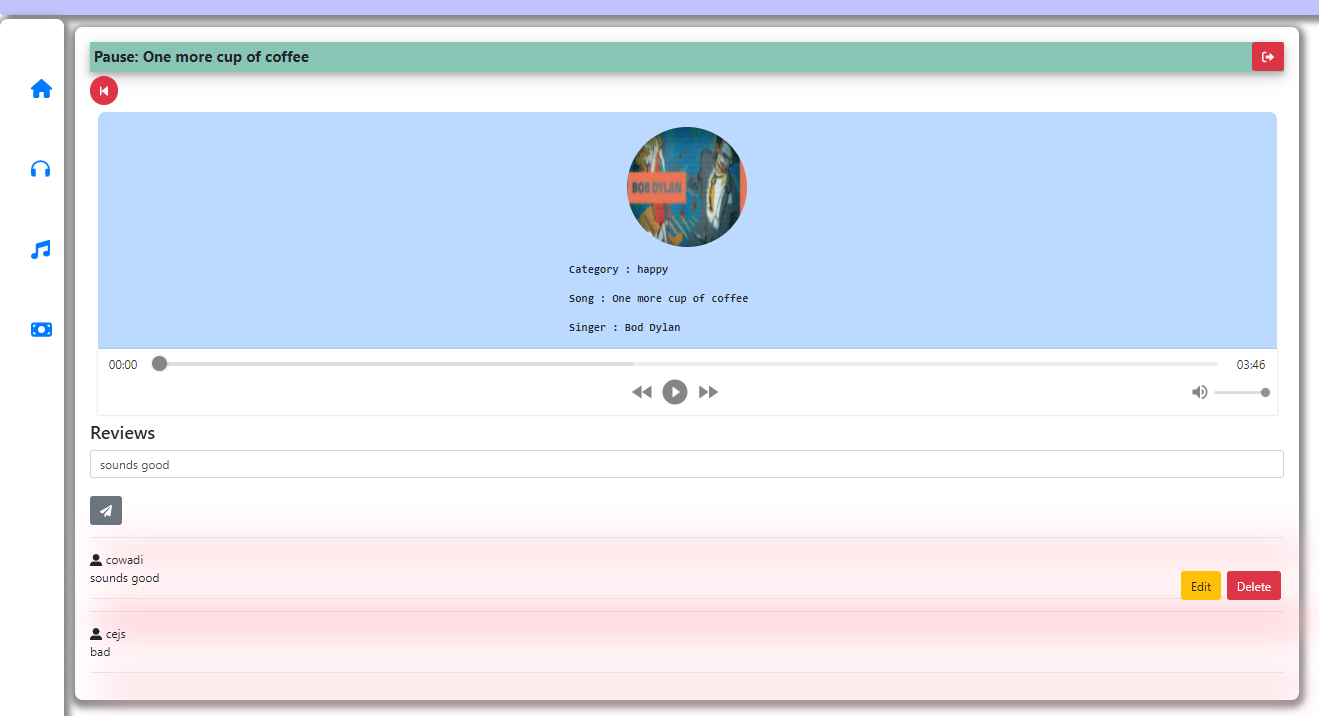


Figure 3‑27 Play single song and add reviews

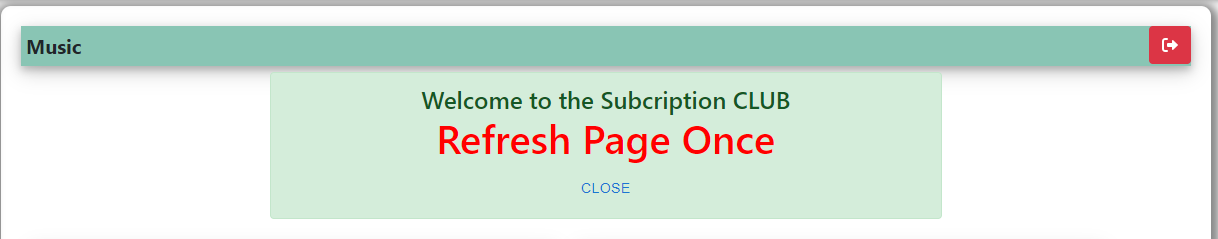
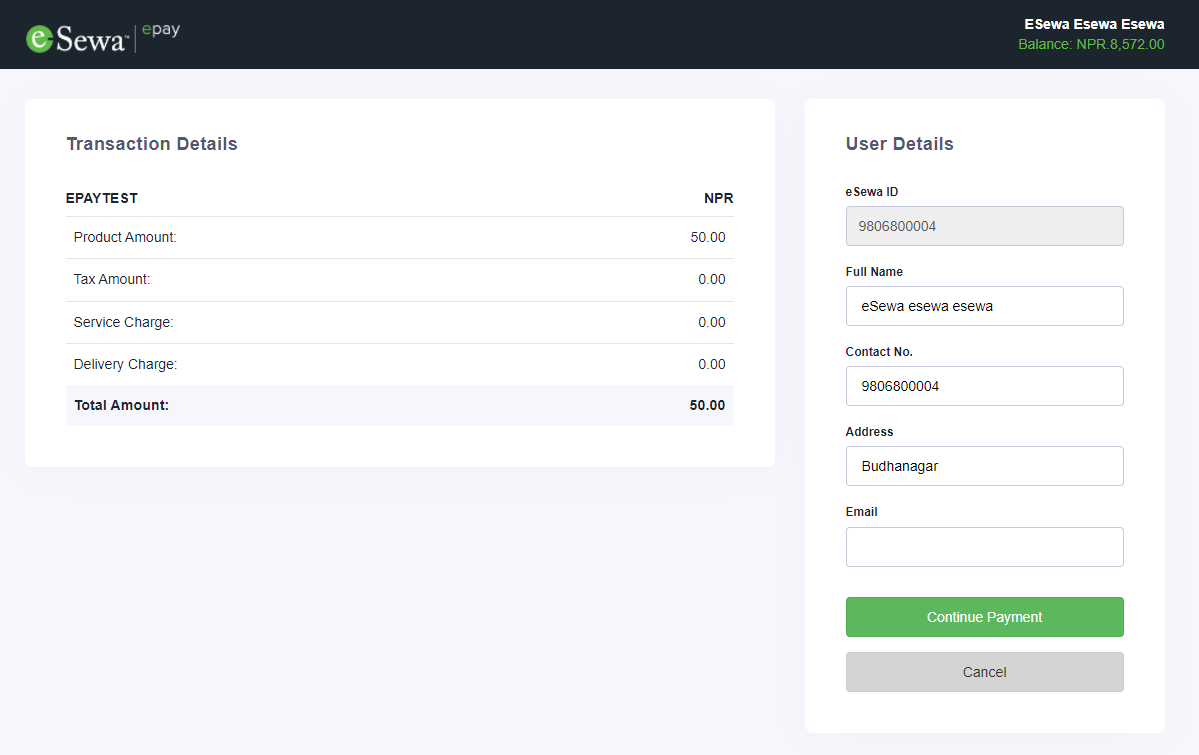
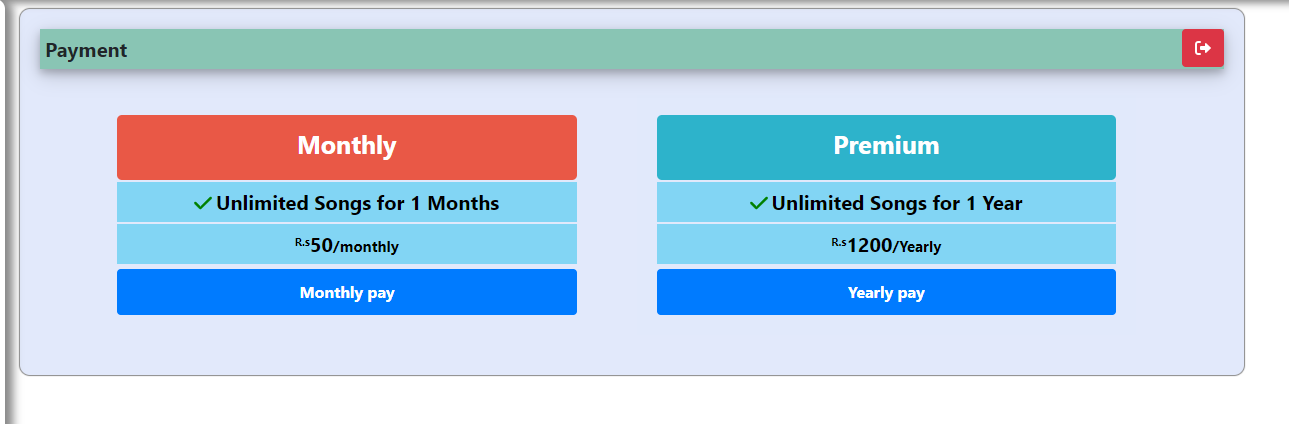


Figure 3‑28 User Reset Password

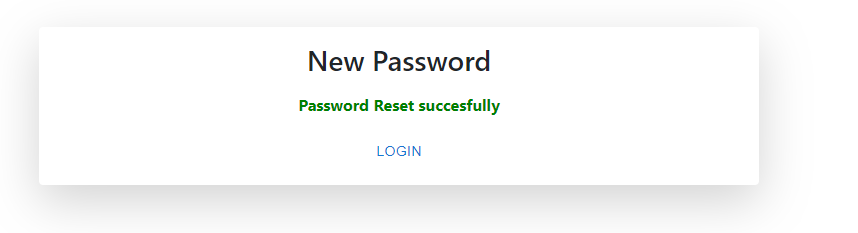
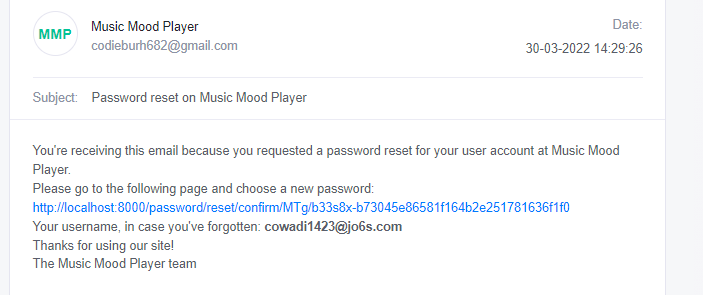
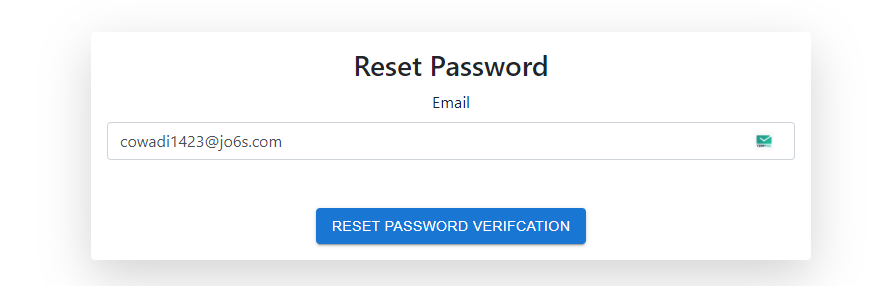


Figure 3‑29 Password Reset

### Physical DFD

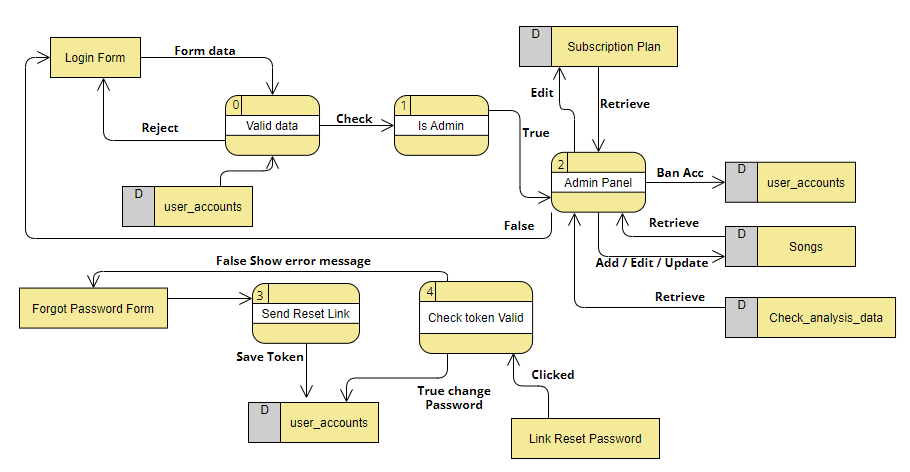
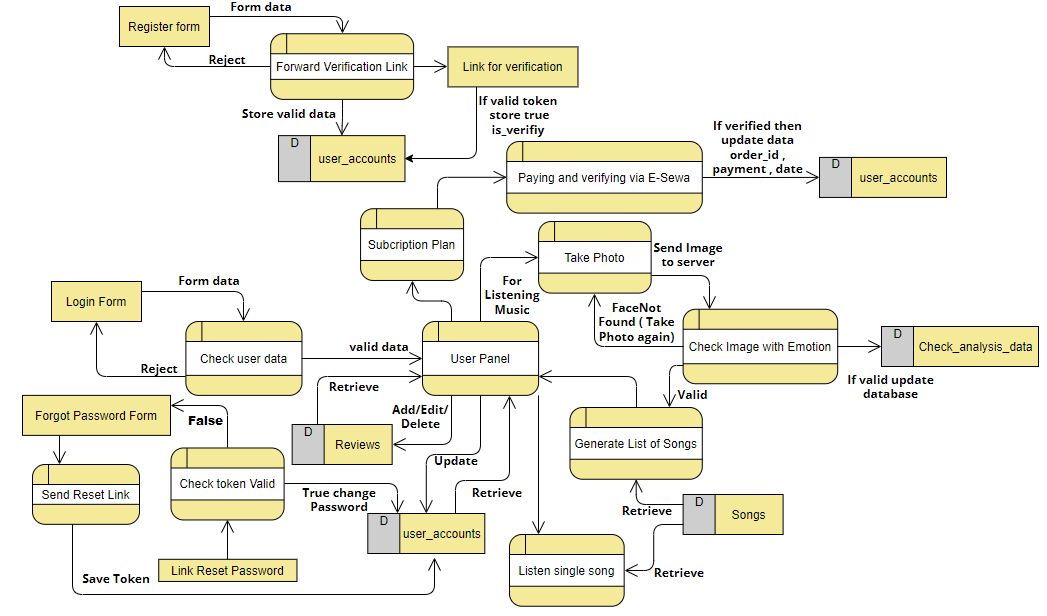


Figure 3‑34 Physical DFD – Admin Panel



0

7

1

9

10

8

5

2

4

3

6

Figure 3‑35 Physical DFD-User Interaction

# Implementation and Testing

## Implementation

### Tools Used (CASE tools, Programming languages, Database platforms)

The Software development life cycle has been used as input in the implementation process. Before coding, I create ER and DFD using drawsql.app and visual-paradigm.com online editor. The actual implementation has been done using Django (Python) as Server Side code and React (JavaScript Library) as front-end code. Django has been used to interact with the backend database. In this implementation, My SQLite has been used as the backend RDBMS. Django processes the inputs or commands given by the user and translates them in the commands understandable to the backend database. The output produced by the backend database is also handled by Django which then displayed on the Browser screen as JSON data. For getting the emotion of user and get sentiment of text (Reviews), DeepFace and TextBlob Library was used respectively.

### Implementation Details of Modules (Description of procedures/functions)

Following are all the modules designed for the Online Shopping System.

#### Admin Module

In this module only admin can have access to login and able to Add Songs details. Admin can also Update, Delete Songs if necessary .Moreover, Admin can see the list of User Subscription with User Details to verify to make purchase successful. Admin can also Ban account if necessary. The report is generated in admin dashboard as how many percentage user are being requesting a certain emotions.

#### Profile Module

In this module a log in user can set his profile picture, username, change password and read the subscription plan. Incase user forgot there password then they can reset it using email verification module.

User can also request a songs capturing a photo with emotions and get list of songs from server.

#### Payment Module

In this module the payment is done. After listening for couples of minute as feature if liked then user can subscribe to the plan and can pay via E-Sewa (Nepali payment service).

## Testing

Testing is a process of running with intent of finding errors in Application. Testing assures the quality of software and represents final review of other phases of software like specification, design, code generation etc. [4].

### Test Cases for Unit Testing

Unit testing emphasizes the verification effort on the smallest unit of web application design i.e. a website component or module. Unit testing is a dynamic method for verification, where code is actually compiled and executed. Unit testing is performed in parallel with the coding phase. Unit testing tests units or modules not the whole website.

I have tested each module of the website individually. As the modules were built up testing was carried out simultaneously, tracking out each and every kind of input and checking the corresponding output until module is working correctly.

The functionality of the modules was also tested as separate units. Each of the three modules was tested as separate units. In each module all the functionalities were tested in isolation.

For admin level, only admin can insert the Mood Music Player details after login. After adding data in database admin can able to edit if there is some error in data.

In profile Module, user can Update his/her profile picture or not. Moreover, user can change password and can login with new password and can see the plan of subscription.

In payment module the user can make payment via esewa which is validated by esewa system and furthermore data are saved in server.

### Test Cases for System Testing

System testing is used to verify, whether the developed system meets the requirements. With minimum hardware and software requirement project i.e. dedicated server containing Debian or Ubuntu or windows with apache with Django ( python needed ) and React and My SQLite install will be sufficient for running the whole website in one go.

Unit testing for each Form

Table 4‑1 login test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Date | Expected Results | Actual Results | Pass/Fail |
| TU01 | Check Customer Login with valid Data | 1.Go to site http://localhost:8000/login  2.Enter email  3.Enter password  4.Click Login | Email=admin@gmail.com  Password=admin | User Should be login into valid E-application | As Expected | Pass |
| TU02 | Check Customer Login with valid Data | 1.Go to site http://localhost:8000/signup  2.Enter email  3.Enter password  4.Click Login | Email =admin@gmail.com  Password=amw | User should not be login into application | As Expected | Pass |

Table 4‑2 User Password reset via email

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Date | Expected Results | Actual Results | Pass/Fail |
| TU03 | Check reset password for valid email and valid token | 1. After getting Email.  2. http://localhost:8000/password/reset/confirm/MTk/b344tw-24eb544422697804b55a45641d728253 | New password=test | Password change | As Expected | Pass |
| TU04 | Check reset password for valid email and valid token | 1. After getting Email.  2. http://localhost:8000/password/reset/confirm/MTk/Invalid\_token | New password=test | Password doesn’t change | As Expected | Pass |

Table 4‑3 User Activation Email

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Date | Expected Results | Actual Results | Pass/Fail |
| TU05 | Check email activation with valid token | 1. After getting Email.  2. http://localhost:8000/activate/MTk/b344lv-8a50a176449d4633b624a3c964cf601f | Click on Verify | E-mail Verifed | As Expected | Pass |
| TU06 | Check email activation with invalid token | 1. After getting Email.  2. http://localhost:8000/activate/MTk/Invalid\_token | Click on Verify | Email will not verified | As Expected | Pass |

Table 4‑4 Registration of User with valid email

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Date | Expected Results | Actual Results | Pass/Fail |
| TU07 | Check Customer Register with valid email and username | 1.Go to site http://localhost:8000/signup  2.Enter app required fields with valid email  4.Click Register | Email=roshan@gmail.com  First name=Roshan  Last name = kc  Password=test@123  Repeat Password=test@123 | User should get mail if email is valid to verify account | As Expected | Pass |
| TU08 | Check Customer Register with invalid email and username | 1.Go to site http://localhost:8000/signup  2.Enter app required fields with valid email  4.Click Register | Email=roshan  First name=Roshan  Last name = kc  Password=test@  Repeat Password=test | User cannot login before verified | As Expected | Pass |

Table 4‑5 User Send Image and get the list of songs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Date | Expected Results | Actual Results | Pass/Fail |
| TU09 | Check whether the image uploaded will send valid emotions songs or not | 1. Take a valid picture and send image in json format as base64  in this url http://localhost:8000/image/uploads/upload/image/  2.Click Submit | {  image\_data=data\_in\_base64  } | Will get the list of songs | As Expected | Pass |
| TU10 | Check whether the not valid image will throw error or not | 1. Take a invalid picture and send image in json format as base64  in this url http://localhost:8000/image/uploads/upload/image/  2.Click Submit | {  image\_data=invalid\_image\_data  } | Will error message | As Expected | Pass |

Table 4‑6 Valid Reviews

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Test Date | Expected Results | Actual Results | Pass/Fail |
| TU11 | Check user reviews with valid data | 1. Go to http://localhost:8000/Songs/songs\_name  .Click Send | {  category: "tmp"  first\_name: "sssk"  reviews: "good"  song\_data: 6  user\_id: 19  } | Will post the reviews | As Expected | Pass |
| TU12 | Check user reviews with invalid data | 1. Go to http://localhost:8000/Songs/songs\_name  .Click Send | {  category: "tmp"  first\_name: "sssk"  reviews:"ksdksdlkdlsk"  song\_data: 6  user\_id: 19  } | Will error message  Not valid reviews | As Expected | Pass |

# Conclusion

## Lesson Learnt / Outcome

It is easy to use, since it’s GUI is very cozy and user friendly. The website is easy to use and interactive making listening music a recreational activity for users. It has been thoroughly tested and implemented. Moreover, the website makes me firmly believe that I can develop other feature related website using simple Django and React with My SQLite database. The requirement for this kind of implementation is minimum and the setup is easy.

On the other hand, there could be some issue if the face isn't properly setup from webcam. i.e., face need to be properly show.

## Conclusion

The ‘**Mood Music Player**’ is designed to provide a web based application that would make listening music easier and more convenient. User don’t have to search songs remembering. Just take a snap and send it, get the lists of songs. Furthermore, songs can be reviews and read reviews of other user. The first phase is trail version given to user for more or less couple of minutes then need to subscribe for more time. Where payment is done via E-Sewa.

## Future Recommendations

Software building is never ending process and continues the life of the software as per the changing needs of the user from time to time. The project is no doubt has been developed keeping in mind easy modification and enhancement that may be required from time to time. However, there are many scopes to modify this software. As because due to shortage of time, I become unable to include many things. The following scope can be added in future, real Time notification can be added where user need to subscribe more or a discount price for who user the website

Frequently can be added

References

|  |  |
| --- | --- |
| [1] | Guru99, "functional-vs-non-functional-requirements.html," 12 6 2020. [Online]. Available: https://www.guru99.com/. |
| [2] | S. Jena, "types-of-feasibility-study-in-software-project-development/," 2021. [Online]. Available: https://www.geeksforgeeks.org/. |
| [3] | indiatimes.com, "definition/systems-design," 2018. [Online]. Available: https://economictimes.indiatimes.com/. |
| [4] | en.wikipedia.org, "wiki/Software\_testing," 23 4 2021. [Online]. Available: https://en.wikipedia.org/. |
| [5] | S. Ramunam, "document/280494," 2018. [Online]. Available: https://www.grin.com/. |