Music Player

# Project Code Link:

Add the link of your project, it can be either OneDrive/Github.

<https://mylambton-my.sharepoint.com/:u:/g/personal/c0821945_mylambton_ca/EetCVvqT5-BJkCe9RhQdnmsByTWL_ZhmjlEng3scVEsG6Q?e=BF9vgS>

# Project Details

I created a Music player web application with two use case scenarios. One is an admin where he can add songs, playlists, and User; the other is a user who can register and log in to the web app, listen to music, and create a playlist. Also, I implemented a machine learning model where based on the age and gender of the User songs suggested to them.

# Team Members:

|  |  |
| --- | --- |
| Student ID | Student Name |
| C082195 | Harmanpreet Singh |

# MongoDB Database Information

Database Name: Music5

|  |  |  |
| --- | --- | --- |
| Collection Name | Collection Details and #of Documents available | Team Member(s) worked on it |
| Music\_playlist | Inside this collection, I store all the information of playlist created by user. It includes Playlist name, Songs (as Array), User. | Harmanpreet Singh |
| Music\_song | Inside this collection, I store all the information of songs created by Admin. It includes Audio, Artist, Audio file, Image url, Genre, time, color | Harmanpreet Singh |
| Music\_user | Inside this collection, I store all the information Users created by themselves or by Admin. It includes Firstname, Lastname, Username, Email, Password, Age, male. | Harmanpreet Singh |

Database backup file: Attach your database backup file here so that it can be restored and used.

<https://mylambton-my.sharepoint.com/:u:/g/personal/c0821945_mylambton_ca/ERZOZPSli9tFjoZhnKpx7nMBJRWE9JG86SkHH5rD99HWTw>

# Python Methods to GET/Insert/Delete/Update Information

|  |
| --- |
| **Method Name, Details, Screen Shot, Author** |
| sAuthor: Harmanpreet Singh  Method: def index(request)  Details: This method fetching User info, all songs and playlist created by that particular user and then putting it inside context array and render it to index.html page  Screenshot of code:Text  Description automatically generated  Screenshot of output window:Graphical user interface, application  Description automatically generated |
| Author: Harmanpreet Singh  Method: def register(request)  Details: This method create new user. It will check whether the request method is POST, if yes, then it will check from existing user that username or email are similar or not. And if doesn’t match anything it will create new user. Also it rendering RegisterForm() and Success method to register.html  Screenshot of code:  A screenshot of a computer  Description automatically generated with medium confidence  Screenshot of output window: |
| Author: Harmanpreet Singh  Method: def playlist\_detail(request, id)  Details: This method used for getting Single playlist data by its playlist id. Once I get playlist info, I run for loop to go through all song id it and compare with all songs collection and then store that specific song list inside the array and finally render it to singleplaylist.html along with user and playlists.  Screenshot of code:Text  Description automatically generated  Screenshot of output window:  Graphical user interface  Description automatically generated with low confidence |
| Author: Harmanpreet Singh  Method: def playlist\_view(request)  Details: The main purpose of this method it to create playlist, where it get all the information from the PlaylistForm provided by the user and then it will check if the method is POST then it will save all the data inside database.  Screenshot of code:A screenshot of a computer  Description automatically generated with medium confidence  Screenshot of output window: |

# User Interface:

UI1:

* Student Name: Harmanpreet Singh
* Details: Home(index.html) Page where user can get all the songs and also have the options for creating new playlist, listening to the older one.
* Screen Shot of Output:

Graphical user interface, application

Description automatically generated

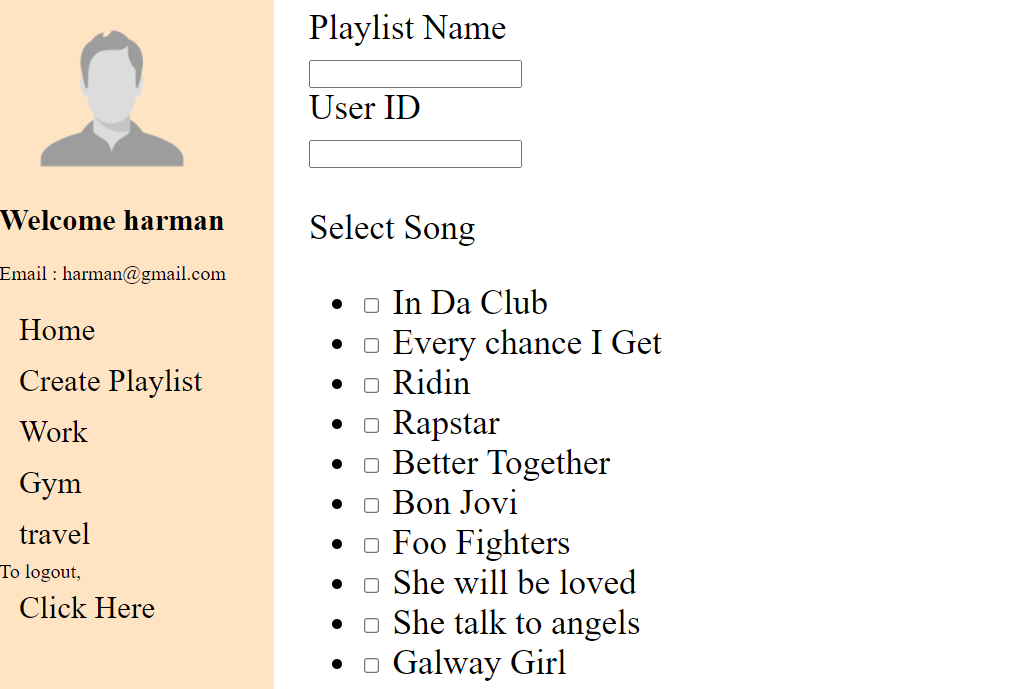
* Screenshot of the Code:

Text

Description automatically generated

UI2:

* Student Name: Harmanpreet Singh
* Details: Create Playlist page is where user create playlist. It consists of form with playlist name, user id which user need to provide along with list of songs from which user can select it.
* Screen Shot of Output:



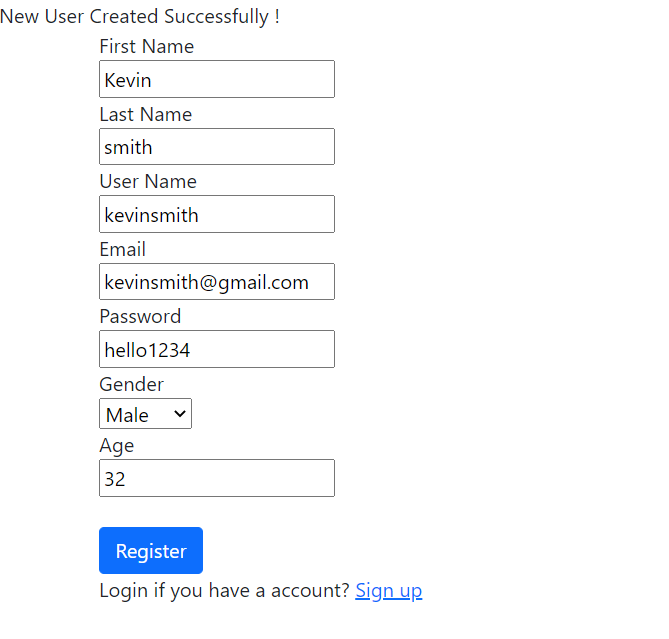
* Screenshot of the Code:

Text

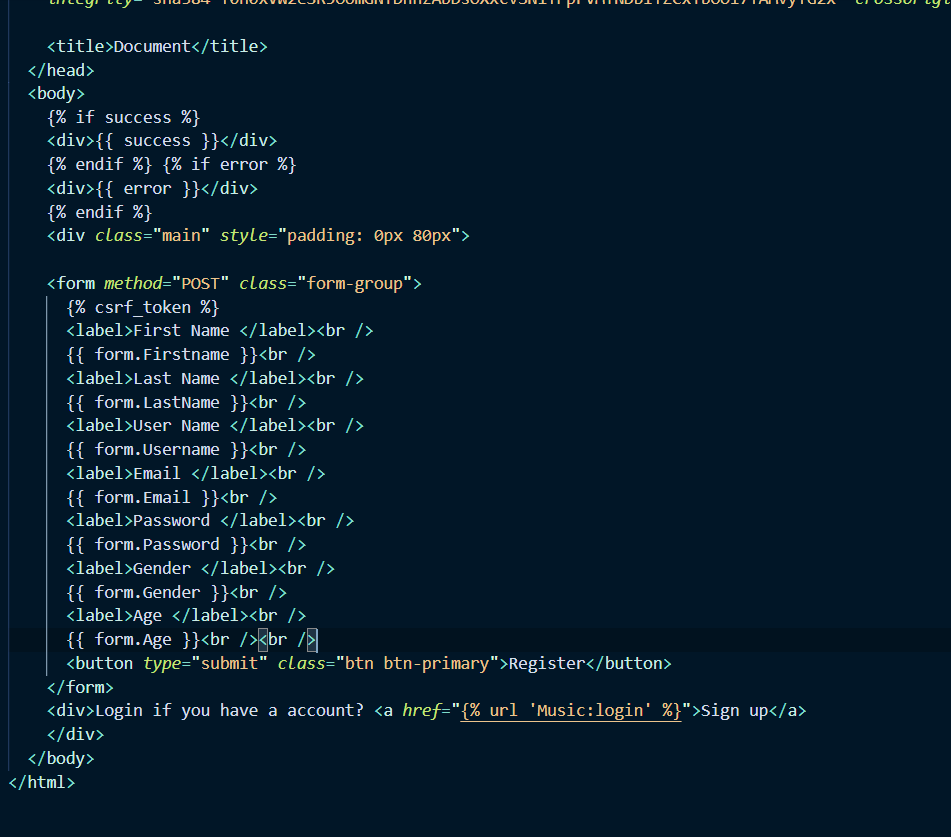
Description automatically generated

UI3:

* Student Name: Harmanpreet Singh
* Details: Register page is where user can register themselves and navigation to login page with login button and then login.
* Screen Shot of Output:



* Screenshot of the Code:



# Implementation of Matplotlib:

UI1:

* Student Name: Harmanpreet Singh
* Details: For Implementation of Matplotlib I used JupyterNotebook. First I’m fetching data from mongo DB, then I’m using panda for putting that data in table form and then finally I’m using Matplotlib for displaying number of songs each user have in bar graph where in Y axis its number of songs and in X axis its name of the artist.
* Screen Shot of Output:

Chart, bar chart

Description automatically generated

* Screenshot of the Code:

Graphical user interface, application, Word

Description automatically generated

UI2:

* Student Name: Harmanpreet Singh
* Details: In this bar chart I’m displaying number of songs by Genre, where Songs is in Y axis and Genre is on X axis
* Screen Shot of Output:

Chart, bar chart

Description automatically generated

* Screenshot of the Code:

Graphical user interface, text, application, Word

Description automatically generated

# Implementation of ML:

For the Machine Learning, I Implemented one model where we provide a user gender and age to the model; In return, it gives the music genre. First, I created the data I wanted to import, as you can see in the picture below.

Graphical user interface, text

Description automatically generated

Then I prepared the data to send it to train which finding any null value and converting genre into numerical since model can’t trained by character, then I trained it using RandomForestClassifier. Once it's trained, I dump the model as forest.pkl and then implemented it in a project inside the login method in Views.py where I'm sending age and gender to the model, and once I got the genre basis on that, I'm suggesting all the songs related to that genre.

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

# How to Submit:

1. Upload your documentation in Moodle (one person from each group). If the file size is too big convert it to PDF.
2. If you are using Github share the github link otherwise upload your code in Onedrive and share with my email address.

# Score Distribution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Total Marks | Student 1 | Student 2 | Student 3 | Student 4 |
| NoSQL Database | 20 |  |  |  |  |
| Python Methods | 25 |  |  |  |  |
| UI Forms and data display | 25 |  |  |  |  |
| Charts/Graphs | 10 |  |  |  |  |
| ML | 10 |  |  |  |  |
| Documentation | 10 |  |  |  |  |
| Total | 100 |  |  |  |  |