



STREAMX

A Technical report of Project submitted to Asutosh College, University of Calcutta
in partial fulfilment of the requirements
for the Degree of

BACHELOR OF VOCATIONAL in SOFTWARE DEVELOPMENT

SUBMITTED BY:

Saumili Dutta

Bistirno Shil

Srinjon Bhattacharjee

Monalisa Roy

Under the guidance of Prof. Madhurima Khan

SESSION : 2024

SEMESTER - VI

Acknowledgement

I would like to extend my sincere gratitude to my project guide, Prof. Madhurima Khan, for her continuous support and invaluable guidance throughout this project. I would like to thank my team member for their help, which were crucial in bringing this project to fruition.

Preface

This document outlines the development of a music streaming application designed for personal use. Inspired by the success of platforms like Spotify in curating personalized playlists based on user moods and preferences, our goal is to create an intuitive and engaging music experience. This application allows users aged 10 and above to explore and enjoy music tailored to their tastes, providing a seamless blend of discovery and enjoyment. Through this project, we seek to combine the joy of music discovery with the convenience of modern streaming technology.

INDEX

- Synopsis
- Introduction
- Scope and Objective
- Definition Of Problem
- System Analysis and System Planning
- Project Description
- Project Architecture
- Software development methodology
- Life cycle of project
- Data Flow Diagram
- Use Case Diagram
- Graphical abstract
- ERD
- Details of software and hardware used
- Cost benefit analysis
- Coding
- Database
- Project screenshot
- Test Case
- Result and Discussion
- Conclusion
- Future work
- Bibliography

Synopsis

This project aims to develop a comprehensive music streaming application to revolutionize how users access and enjoy music. Traditional music listening methods, such as radio and personal music libraries, often lack the flexibility and personalization that modern users desire. Our solution leverages advanced technology to provide a seamless, high-quality streaming experience. By integrating features such as personalized playlists, robust search and discovery tools, the application will enhance user engagement and satisfaction. The application ensures users can effortlessly discover new artists and genres that match their unique preferences and moods. This project seeks to transform the music listening experience, making it more dynamic, personalized, and enjoyable for all users.

Introduction

Music streaming applications are essential for enhancing user experience in the digital music landscape. Traditional discovery methods, such as radio and personal recommendations, often fall short in terms of personalization and immediacy. This project introduces a music streaming application designed to provide customized music suggestions, improve user engagement, and deliver an intuitive and enjoyable music listening experience. By leveraging a user-friendly interface, our application aims to offer personalized playlists and music discovery, catering to the unique tastes and preferences of each user.

Scope and Objective

This project encompasses the development of a comprehensive music streaming application. The functionalities include user account management, music library access, playlist creation, search and discovery features, and seamless music playback.

User Account Management:

- Implement user registration, login, and profile management features

Music Library Access:

- Provide access to an extensive and diverse music library
- Include options for browsing by artist, album, and podcasts

Search and Discovery:

- Develop robust search functionality to help users find specific tracks, albums, or artists quickly.

Seamless Music Playback:

- Ensure high-quality audio streaming with minimal buffering.
- Implement features like going back and moving forward by 10s, choosing to play the song from somewhere specific

Streamlined Music Discovery:

- Provide curated playlists and personalized recommendations to keep the user experience fresh and exciting.

By focusing on these objectives, the project aims to deliver a user-friendly and engaging music streaming application that caters to the diverse needs and preferences of music enthusiasts.

Definition Of Problem

The primary problem addressed by this project is the challenge users face in managing and enjoying their music streaming experience efficiently. Users often encounter difficulties in navigating large music libraries, creating personalized playlists, and maintaining uninterrupted, high-quality playback. The vast amount of available music and inadequate user interface design can lead to a suboptimal user experience.

- User Frustration: Users struggle to navigate through extensive music libraries, leading to frustration and a subpar user experience.
- Complexity in Creation: Creating and managing playlists can be cumbersome and unintuitive, discouraging users from curating their own music collections.
- Inconsistent Quality: Users experience interruptions and buffering, leading to frustration and decreased enjoyment of the streaming service.
- Generic Recommendations: Broad algorithms fail to capture unique user preferences, leading to generic and unsatisfactory music recommendations.
- Missed Discovery Opportunities: Users miss out on discovering new artists and genres they would enjoy, limiting their exposure to diverse music.
- Reduced Engagement: Difficulty in navigating the app and finding enjoyable music decreases user engagement and satisfaction with the platform.

System Analysis and System Planning

Music streaming apps have revolutionized how people consume music, offering a vast library of songs on demand. To develop an effective music streaming app, it is crucial to analyze the needs of potential users, market trends, and the existing competition.

Planning involves defining the project scope, objectives, and deliverables. The following steps are critical:

- Identify Requirements: User needs, functionalities, and performance requirements
- Define Objectives: High-quality audio streaming, user-friendly interface, and robust recommendation system
- Project Milestones: Initial research, design, development, testing, deployment, and maintenance.

Project Description

The music streaming application is designed to revolutionize the way users interact with and consume music online. This project aims to provide a seamless, high-quality music streaming experience accessible via web browsers. Users can explore a vast library of songs, view playlists, and like it to store in the library. The application combines robust backend services, and an intuitive frontend interface to deliver a user-friendly and engaging experience.

Project Architecture

The architecture consists of the following key components:

Client-Side (Frontend):

- User Interface: Developed using HTML, CSS and JS, providing a responsive and interactive user experience.
- Authentication: Implemented secure user authentication using username and password, managed through XAMPP's MySQL database.

Server-Side (Backend):

- API Gateway: Managed by a Django application, handling requests from the frontend and routing them to appropriate services. The `requests` library is used for outbound HTTP requests to integrate with third-party APIs and external services.
- Business Logic: Implemented in Django, managing core functionalities such as user management, playlist creation, and music recommendations.
- Database: Managed using MySQL from XAMPP, ensuring efficient storage and retrieval of structured data.
- Music Streaming: Integration with third-party APIs like Spotify scraper API and You Tube for music content, ensuring efficient and seamless music playback.

Software development methodology

Softwares can be developed using several SDLC models such as Waterfall Model, Incremental Model, Iterative Model, Prototype model, Agile Model. The phases of development is: requirements analysis, system design, implementation, integration and testing, deployment, and maintenance.

Waterfall model:

The Waterfall model is a linear and sequential approach where each phase must be completed before the next phase begins. This model follows a strict order. Once a phase is completed, it is typically not revisited.

Incremental Model:

The Incremental model divides the project into smaller modules. Each module goes through the phases of development. These increments are developed and delivered in successive versions until the complete system is finished. This allows for parts of the system to be available and functional earlier in the development process.

Iterative Model:

The Iterative model focuses on refining and improving the software through repeated cycles (iterations). Each iteration involves the phases of development. Feedback from each iteration is used to make improvements and additions in subsequent iterations until the final product meets all requirements.

Prototype Model:

The Prototype model involves creating prototypes based on initial requirements and then presented to customers for feedback. This feedback is used to refine the requirements and improve the prototypes until a final version is agreed upon, which is then fully developed.

Agile Model:

The Agile model combines elements of both the incremental and iterative models.

For this project, we have used Incremental model as we have divided the project into small increments and developed and tested each increments and then merged all into the application.

Life cycle of project

The life cycle of the project includes the following phases:

- Requirement Analysis
- Doing research to gather requirements.

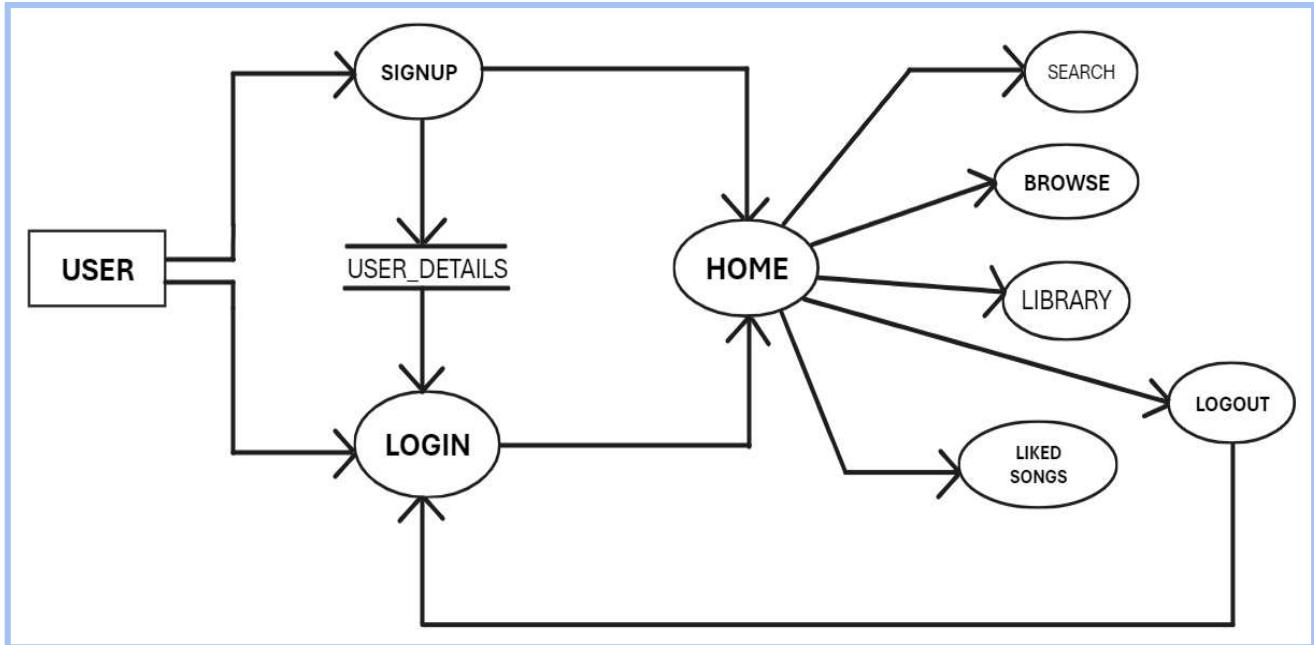
- Design
- Creating the layout for the user interface.
- Designing the system architecture and database schema.

- Dividing into increments
- Divided the whole application into smaller modules(increments): home, library, liked songs, search, artist profile, login/signup/database

- Implementation and Development
- Frontend development using HTML, CSS, JS for a dynamic and responsive UI.
- Backend development using Django for robust server-side logic.
- Integrating third-party API for music content.
- Using XAMPP to manage the database
- Developing each increment

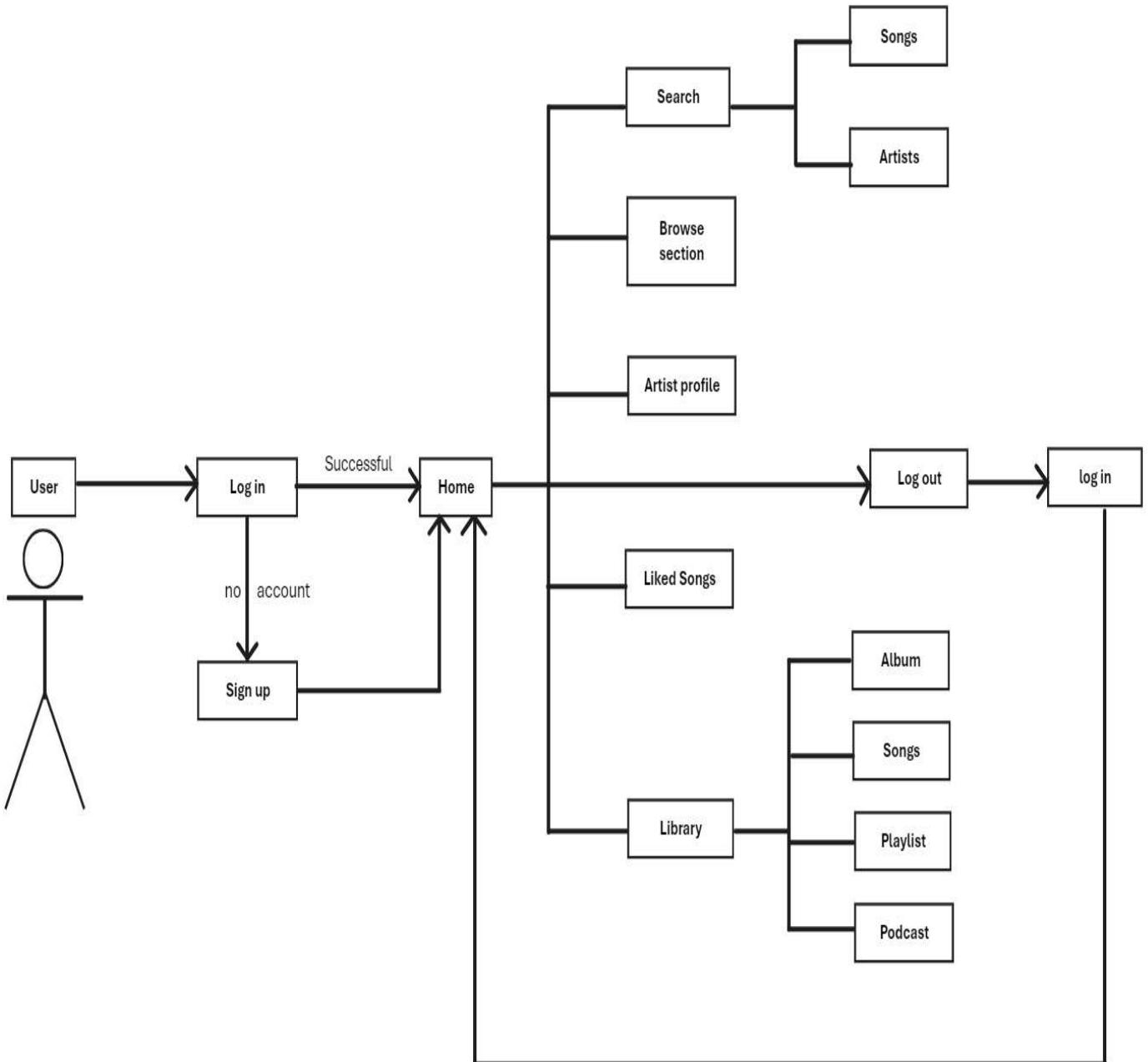
- Testing
- Unit testing of individual components.
- Integration testing to ensure seamless interaction between modules.

Data Flow Diagram

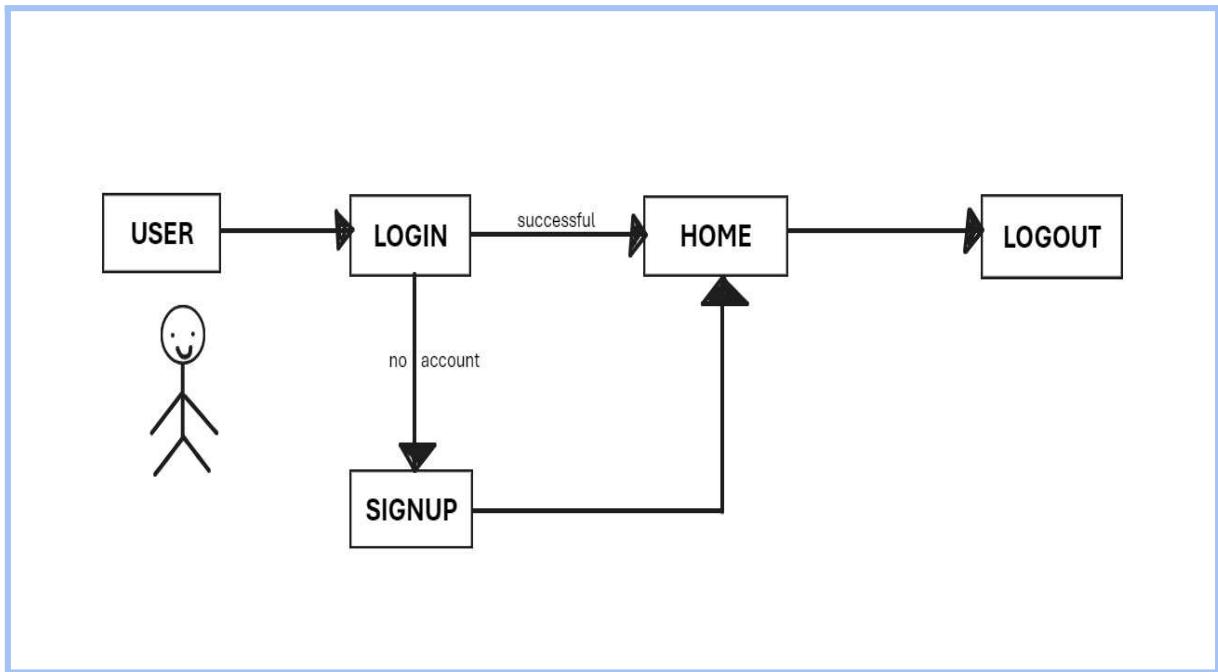


Level 1

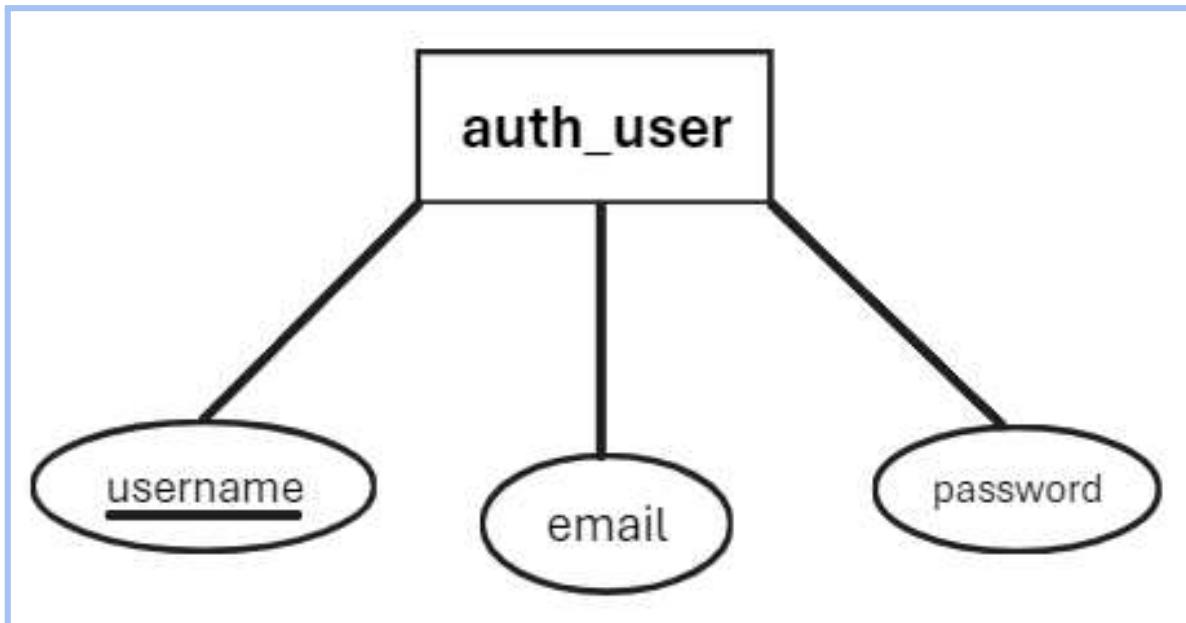
Use Case Diagram



Graphical abstract



Entity Relationship Diagram



Details of software and hardware used

The softwares used in this project are:

- Python interpreter

The project uses Django, a Python framework for backend development and thus it is necessary to have a Python interpreter to execute the Python code. For this project, Python's 3.12.3 version has been used.

- Visual Studio

An IDE is the integral part of the project. To edit the HTML, CSS and JS code notepad could have used, to edit the Python code Pycharm IDE could have used and to run the Django commands Command Prompt could have used. Instead of using three different software, we have used Visual Studio code which is a lightweight but powerful code editor developed by Microsoft which has integrated terminal and extensions to debug and run codes of different languages. For this project, Visual Studio code's 1.88.0 version has been used.

- XAMPP

To store the user details upon registration and match with the details while logging in, a database is needed and for that we have used Xampp which is a free and open-source cross-platform web server solution stack.

- Adobe Photoshop CS6

To design the various cover photos and logos for our application, we have used Photoshop CS6

- Microsoft Whiteboard

To draw and design the primary interface, we have used Microsoft Whiteboard. Using this software has a pro and that it has easy to understand features

The hardware required for this project are as follows:

- Processor: Intel Core i3 10th Gen
- Memory (RAM): 8 GB
- Storage: 1 TB HDD
- Operating System: Windows 11

Cost benefit analysis

The application relies heavily on the Spotify Scraper API to function properly, for which there is a subscription required of \$24 which is equivalent to Rs. 2010/- .

Coding

Coding implementation

streamX

- asgi.py
- settings.py
- urls.py
- wsgi.py

templates

- index.html
- library.html
- likedsongs.html
- login.html
- music.html
- profile.html
- search.html
- signup.html

static

- index.css
- login.css
- music.css
- profile.css
- search-style.css
- signup.css

music

- apps.py
- urls.py
- views.py

manage.py

requirements.txt

I. STREAMX

asgi.py

.....

ASGI config for streamX project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see

<https://docs.djangoproject.com/en/5.0/howto/deployment/asgi/>

.....

```
import os
```

```
from django.core.asgi import get_asgi_application
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'streamX.settings')
```

```
application = get_asgi_application()
```

settings.py

.....

Django settings for streamX project.

Generated by 'django-admin startproject' using Django 5.0.6.

For more information on this file, see

<https://docs.djangoproject.com/en/5.0/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/5.0/ref/settings/>

.....

```
from pathlib import Path
```

```
import os
```

```
# Build paths inside the project like this: BASE_DIR / 'subdir'.
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```

SECRET_KEY = 'django-insecure-f*(xrvc_oqg@xt!a+m)9!($+-314be2ysj7h%eu0vw=dj)@'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'music',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'streamX.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR, 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {

```

```

'context_processors': [
    'django.template.context_processors.debug',
    'django.template.context_processors.request',
    'django.contrib.auth.context_processors.auth',
    'django.contrib.messages.context_processors.messages',
],
},
},
]
]

WSGI_APPLICATION = 'streamX.wsgi.application'

# Database
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'streamX',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

# Password validation
# https://docs.djangoproject.com/en/5.0/ref/settings/#auth-passwordValidators
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
]

```

```

    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]

# Internationalization
#https://docs.djangoproject.com/en/5.0/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True

# Static files (CSS, JavaScript, Images)
STATIC_URL = 'static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static'),
]

# Default primary key field type
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

urls.py

.....

URL configuration for streamX project.

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/5.0/topics/http/urls/>

Examples:

Function views

1. Add an import: from my_app import views

2. Add a URL to urlpatterns: path("", views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home

2. Add a URL to urlpatterns: path("", Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

.....

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('music.urls'))
]
```

wsgi.py

.....

WSGI config for streamX project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see

<https://docs.djangoproject.com/en/5.0/howto/deployment/wsgi/>

.....

```
import os
from django.core.wsgi import get_wsgi_application
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'streamX.settings')
application = get_wsgi_application()
```

II. TEMPLATES

index.html

```

{%- load static %}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>streamX</title>

    <link rel="stylesheet" type="text/css" href="{% static 'index.css' %}" />

</head>

<body style="margin:0; padding:0; background:#121212; font-family:Arial, sans-serif;
color:#fff;">

<div style="margin-left:240px; padding:20px;">

    <header class="header-container">

        <form action="/" method="POST">

            {% csrf_token %}

            <div class="search-bar" style="width: 390px;">

                <input type="search" name="search_query" placeholder="Search for..." aria-label="Search">

                <button type="submit" aria-label="Search">
                    
                </button>

            </div>

        </form>

        <div class="search-title">Top Artists</div>

    </header>

<main style="padding-top:20px;">

    <section style="margin-top:0px; padding-left: 10px; white-space:nowrap;">
```

```

<!-- Artist Card -->
<div class="artist-card">
  <a href="/profile/61JrslREXq98hurYL2hYoc/" style="text-decoration: none;">
    
    <div class="artist-info">
      <div class="artist-name">Lata Mangeshkar</div>
      <div class="artist-title">Artist</div>
    </div>
  </a>
</div>

<div class="artist-card">
  <a href="/profile/OGF4shudTAFv8ak9eWdd4Y/" style="text-decoration: none;">
    
    <div class="artist-info">
      <div class="artist-name">Kishore Kumar</div>
      <div class="artist-title">Artist</div>
    </div>
  </a>
</div>

<div class="artist-card">
  <a href="/profile/2ae6PxICSOZHvjqiCcg0n8/" style="text-decoration: none;">
    
    <div class="artist-info">
      <div class="artist-name">S.P. Balasubramanyam</div>
      <div class="artist-title">Artist</div>
    </div>
  </a>
</div>

```

```

</div>

<div class="artist-card">
  <a href="/profile/1dVygo6tRFXC8CSWURQJq2/" style="text-decoration: none;">
    
    <div class="artist-info">
      <div class="artist-name">Sonu Nigam</div>
      <div class="artist-title">Artist</div>
    </div>
  </a>
</div>

<div class="artist-card">
  <a href="/profile/OoOet2f43PA68X5RxKobEy/" style="text-decoration: none;">
    
    <div class="artist-info">
      <div class="artist-name">Shreya Ghoshal</div>
      <div class="artist-title">Artist</div>
    </div>
  </a>
</div>

<div class="artist-card">
  <a href="/profile/6KImCVD70vtloJWnq6nGn3/" style="text-decoration: none;">
    
    <div class="artist-info">
      <div class="artist-name">Harry Styles</div>
      <div class="artist-title">Artist</div>
    </div>
  </a>
</div>

```

```

    </div>
</a>
</div>

<div class="artist-card">
    <a href="/profile/6eUKZXaKkcvIHOKu9w2n3V/" style="text-decoration: none;">
        
            <div class="artist-info">
                <div class="artist-name">Ed Sheeran</div>
                <div class="artist-title">Artist</div>
            </div>
        </a>
    </div>
</section>

<!-- "Browse All" Section Title -->
<div class="browse-all-title">Browse all</div>

<!-- "Browse All" Section Content -->
<div class="browse-all">
    <!-- Category Card -->
        <div class="category-card">
            
            <div class="category-title">Podcasts</div>
        </div>

        <div class="category-card">
            
            <div class="category-title">Audiobooks</div>
        </div>

        <div class="category-card">

```

```

<div class="category-title">Live Events</div>
</div>

<div class="category-card">

<div class="category-title">Made For You</div>
</div>

<div class="category-card">

<div class="category-title">New Releases</div>
</div>

<div class="category-card">

<div class="category-title">2024 in Music</div>
</div>
</div>

<div class="browse-all">
<!-- Category Card --&gt;
&lt;div class="category-card"&gt;
&lt;img src="{% static 'assets/2024podcast.jpeg' %}" alt="2024 in podcasts"&gt;
&lt;div class="category-title"&gt;2024 in podcasts&lt;/div&gt;
&lt;/div&gt;

&lt;div class="category-card"&gt;
&lt;img src="{% static 'assets/merch.jpeg' %}" alt="Merch"&gt;
&lt;div class="category-title"&gt;Merch&lt;/div&gt;
&lt;/div&gt;

&lt;div class="category-card"&gt;
&lt;img src="{% static 'assets/christmashits.jpeg' %}" alt="Merry Christmas"&gt;</pre>
```

```
<div class="category-title">Merry Christmas</div>
</div>

<div class="category-card">
  
  <div class="category-title">Pop</div>
</div>

<div class="category-card">
  
  <div class="category-title">Mood</div>
</div>

<div class="category-card">
  
  <div class="category-title">Hip-Hop</div>
</div>
</div>

<div class="browse-all">
  <!-- Category Card -->
  <div class="category-card">
    
    <div class="category-title">Podcast Charts</div>
  </div>

  <div class="category-card">
    
    <div class="category-title">Podcast New Releases</div>
  </div>

  <div class="category-card">
    
    <div class="category-title">Video Podcasts</div>
  </div>
</div>
```

```

</div>

<div class="category-card">
  
  <div class="category-title">Fiction & Literature</div>
</div>

<div class="category-card">
  
  <div class="category-title">Charts</div>
</div>

<div class="category-card">
  
  <div class="category-title">Workout</div>
</div>
</div>
</main>
</div>

<div class="sidebar">
  <!-- streamX Logo -->
  <div class="logo-container">
    
  </div>

  <!-- Welcome User -->
  <div class="sidebar-section">
    <span style="margin-left: 14px;" class="sidebar-link-icon">👤 </span>
    <b>Welcome, {{user.username}}</b>
    <br />
    <br />
  </div>

```

```

<a href="/logout">
  <span class="sidebar-link-icon"></span>
  <b>Logout</b>
</a>
</div>

<!-- sidebar section -->
<div class="sidebar-section">
  <a href="/" class="sidebar-link active">
    <span class="sidebar-link-icon"></span>
    Home
  </a>
  <a href="/search/" class="sidebar-link">
    <span class="sidebar-link-icon"></span>
    Search
  </a>
  <a href="/library/" class="sidebar-link">
    <span class="sidebar-link-icon"></span>
    Your Library
  </a>
  <a href="/likedsongs/" class="sidebar-link">
    <span class="sidebar-link-icon"></span>
    <span class="sidebar-link-text">Liked Songs</span>
  </a>
</div>
</div>
</body>
</html>

```

library.html

```
{% load static %}
```

```
<!DOCTYPE html>
```

```

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Library</title>
    <link rel="stylesheet" type="text/css" href="{% static 'profile.css' %}" />
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.1.4/dist/tailwind.min.css"
rel="stylesheet">
  </head>

  <body>
    <!-- Sidebar -->
    <div class="sidebar">
      <!-- streamX Logo -->
      <div class="logo-container">
        
      </div>

      <!-- sidebar section -->
      <div class="sidebar-section">
        <a href="/" class="sidebar-link">
          <span class="sidebar-link-icon">🏠 </span>
          Home
        </a>
        <a href="/search" class="sidebar-link">
          <span class="sidebar-link-icon">🔍 </span>
          Search
        </a>
        <a href="/library/" class="sidebar-link">
          <span class="sidebar-link-icon">🎵 </span>
          Your Library
        </a>
        <select class="sidebar-link2">
          <option value="songs">All</option>

```

```

<option value="songs">Songs</option>
<option value="songs">Album</option>
<option value="playlist">Playlist</option>
<option value="podcast">Podcast</option>
</select>
</div>
</div>

<!-- Main Content -->
<div class="flex-1">

<!-- Favourite Tracks Section -->
<div class="p-5">
  <h2 class="text-2xl font-bold mb-4">Songs</h2>
  <div class="divide-y divide-gray-800">

    <!-- Single Row -->
    <div class="grid grid-cols-3 items-center py-3">
      <div class="flex items-center space-x-4">
        
        <div>
          <a href="/music/OcUdmruT6kjxa5KpteSdw/" style="text-decoration: none; color: white;">
            <p class="text-base font-medium">Nahin Milta</p>
            <p class="text-xs text-gray-400">Bayaan</p>
          </a>
        </div>
      </div>
      <div class="justify-self-end text-gray-400">2019</div>
      <div class="justify-self-end text-gray-400">4:45</div>
    </div>

    <div class="grid grid-cols-3 items-center py-3">
      <div class="flex items-center space-x-4">

```

```


<div>
  <a href="/music/1ZMiCix7XSAbfAJIEZWMCP/" style="text-decoration: none; color: white;">
    <p class="text-base font-medium">Falling</p>
    <p class="text-xs text-gray-400">Harry Styles</p>
  </a>
</div>
</div>
<div class="justify-self-end text-gray-400">2019</div>
<div class="justify-self-end text-gray-400">4:02</div>
</div>

<div class="grid grid-cols-3 items-center py-3">
  <div class="flex items-center space-x-4">
    
    <div>
      <a href="/music/4x7fVJSYQvwaaJUqVnFGPa/" style="text-decoration: none; color: white;">
        <p class="text-base font-medium">Rangeela Re</p>
        <p class="text-xs text-gray-400">Lata Mangeshkar</p>
      </a>
    </div>
  </div>
  <div class="justify-self-end text-gray-400">1970</div>
  <div class="justify-self-end text-gray-400">6:63</div>
</div>
<a href="/likedsongs/" style="color:rgb(110, 193, 241);">See more liked songs...</a>
</div>
</div>

<!-- Favourite Albums Section -->
<div class="p-5">

```

```

<h2 class="text-2xl font-bold mb-4">Albums</h2>
<div class="divide-y divide-gray-800">

    <!-- Single Row -->
    <div class="grid grid-cols-3 items-center py-3">
        <div class="flex items-center space-x-4">
            
            <div>
                <a href="#" style="text-decoration: none; color: white;">
                    <p class="text-base font-medium">Hum Aapke Hain Koun</p>
                </a>
            </div>
        </div>
        <div class="justify-self-end text-gray-400">1994</div>
        <div class="justify-self-end text-gray-400">15 songs</div>
    </div>

    <div class="grid grid-cols-3 items-center py-3">
        <div class="flex items-center space-x-4">
            
            <div>
                <a href="#" style="text-decoration: none; color: white;">
                    <p class="text-base font-medium">Rockstar</p>
                </a>
            </div>
        </div>
        <div class="justify-self-end text-gray-400">2011</div>
        <div class="justify-self-end text-gray-400">14 songs</div>
    </div>
    <a href="#" style="color:rgb(110, 193, 241);">See more albums...</a>
</div>
</div>

```

```

<!-- Favourite Playlists Section -->
<div class="p-5">
  <h2 class="text-2xl font-bold mb-4">Playlists</h2>
  <div class="divide-y divide-gray-800">

    <!-- Single Track Row -->
    <div class="flex justify-between items-center py-3">
      <div class="flex items-center space-x-4">
        
        <div>
          <a href="#" style="text-decoration: none; color: white;">
            <p class="text-base font-medium">Dance Mix</p>
          </a>
        </div>
      </div>
      <div class="text-gray-400">60 items</div>
    </div>

    <div class="flex justify-between items-center py-3">
      <div class="flex items-center space-x-4">
        
        <div>
          <a href="#" style="text-decoration: none; color: white;">
            <p class="text-base font-medium">Focus Ragas</p>
          </a>
        </div>
      </div>
      <div class="text-gray-400">37 items</div>
    </div>
    <a href="#" style="color:rgb(110, 193, 241);">See more playlists...</a>
  </div>
</div>

```

```

<!-- Favourite Podcasts Section -->
<div class="p-5">
  <h2 class="text-2xl font-bold mb-4">Podcasts</h2>
  <div class="divide-y divide-gray-800">

    <!-- Single Row -->
    <div class="flex justify-between items-center py-3">
      <div class="flex items-center space-x-4">
        
        <div>
          <a href="#" style="text-decoration: none; color: white;">
            <p class="text-base font-medium">Gita for the Young and Restless</p>
          </a>
        </div>
        </div>
        <div class="text-gray-400">11 episodes</div>
      </div>
    </div>

    <div class="flex justify-between items-center py-3">
      <div class="flex items-center space-x-4">
        
        <div>
          <a href="#" style="text-decoration: none; color: white;">
            <p class="text-base font-medium">Death, Lies and Cyanide</p>
          </a>
        </div>
        </div>
        <div class="text-gray-400">10 episodes</div>
      </div>
      <a href="#" style="color:rgb(110, 193, 241);">See more podcasts...</a>
    </div>
  </div>
</div>

```

```

    </div>
</div>
</div>
</body>
</html>
```

likedsongs.html

```

{% load static %}

<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liked Songs</title>
    <link rel="stylesheet" type="text/css" href="{% static 'profile.css' %}" />
</head>
<body>

    <!-- Sidebar -->
    <div class="sidebar">
        <!-- streamX Logo -->
        <div class="logo-container">
            
        </div>

        <!-- sidebar section -->
        <div class="sidebar-section">
            <a href="/" class="sidebar-link">
                <span class="sidebar-link-icon">🏠</span>
                Home
            </a>
            <a href="/search" class="sidebar-link">
                <span class="sidebar-link-icon">🔍</span>
            </a>
        </div>
    </div>
```

```

Search
</a>
<a href="/library/" class="sidebar-link">
  <span class="sidebar-link-icon">🎵 </span>
  Your Library
</a>
<a href="/likedsongs/" class="sidebar-link">
  <span class="sidebar-link-icon">❤️ </span>
  Liked Songs
</a>
</div>
</div>

<!-- Main Content -->
<div class="flex-1">
  <!-- Artist Header -->
  <div class="relative">
    
  </div>

  <!-- Popular Tracks Section -->
  <div class="p-5">
    <h2 class="text-2xl font-bold mb-4">Liked Songs</h2>
    <div class="divide-y divide-gray-800">

      <!-- Single Row -->
      <div class="flex justify-between items-center py-3">
        <div class="flex items-center space-x-4">
          
          <div>
            <a href="/music/0cUdmruT6kjxa5KpteSdwc/" style="text-decoration: none; color: white;">
              <p class="text-base font-medium">Nahin Milta</p>
              <p class="text-xs text-gray-400">Bayaan</p>

```

```

        </a>
    </div>
</div>
<div class="text-gray-400">2019</div>
<div class="text-gray-400">4:45</div>
</div>

<div class="flex justify-between items-center py-3">
    <div class="flex items-center space-x-4">
        
        <div>
            <a href="/music/5XmKj1B7evELEXCUVcx9k0/" style="text-decoration: none; color: white;">
                <p class="text-base font-medium">Farda</p>
                <p class="text-xs text-gray-400">Bayaan</p>
            </a>
        </div>
    </div>
    <div class="text-gray-400">2019</div>
    <div class="text-gray-400">4:29</div>
</div>

<div class="flex justify-between items-center py-3">
    <div class="flex items-center space-x-4">
        
        <div>
            <a href="/music/21dj07ljkQC7X5ZvKE4fx4/" style="text-decoration: none; color: white;">
                <p class="text-base font-medium">Naam Goom Jayega</p>
                <p class="text-xs text-gray-400">Lata Mangeshkar and Bhupinder Singh</p>
            </a>
        </div>
    </div>
</div>

```

```

<div class="text-gray-400">1977</div>
<div class="text-gray-400">5:40</div>
</div>

<div class="flex justify-between items-center py-3">
  <div class="flex items-center space-x-4">
    
    <div>
      <a href="/music/3J5XaAaG7lcPawT8bpnaIW/" style="text-decoration: none; color: white;">
        <p class="text-base font-medium">Luka Chuppi</p>
        <p class="text-xs text-gray-400">Lata Mangeshkar and A.R. Rahman</p>
        </a>
      </div>
    </div>
    <div class="text-gray-400">2006</div>
    <div class="text-gray-400">4:12</div>
  </div>
</div>

<div class="flex justify-between items-center py-3">
  <div class="flex items-center space-x-4">
    
    <div>
      <a href="/music/5SNISinXkLIdkP8eBGWfO/" style="text-decoration: none; color: white;">
        <p class="text-base font-medium">Aa chal ke tujhe</p>
        <p class="text-xs text-gray-400">Kishore Kumar</p>
        </a>
      </div>
    </div>
    <div class="text-gray-400">1964</div>
    <div class="text-gray-400">5:09</div>
  </div>
</div>

```

```

<div class="flex justify-between items-center py-3">
  <div class="flex items-center space-x-4">
    
    <div>
      <a href="/music/2UJ8Fg20zeJiC8WQEEE9uQ/" style="text-decoration: none; color: white;">
        <p class="text-base font-medium">Jaat Kahan Ho</p>
        <p class="text-xs text-gray-400">Kesarbai Kerkar</p>
      </a>
    </div>
  </div>
  <div class="text-gray-400">1944</div>
  <div class="text-gray-400">3:39</div>
</div>

<div class="flex justify-between items-center py-3">
  <div class="flex items-center space-x-4">
    
    <div>
      <a href="/music/0iXLwnLmLwn9y54JtBTNxY/" style="text-decoration: none; color: white;">
        <p class="text-base font-medium">Main Agar Kahoon</p>
        <p class="text-xs text-gray-400">Sonu Nigam and Shreya Ghoshal</p>
      </a>
    </div>
  </div>
  <div class="text-gray-400">2007</div>
  <div class="text-gray-400">5:29</div>
</div>

<div class="flex justify-between items-center py-3">
  <div class="flex items-center space-x-4">
    
    <div>

```

```

<a href="/music/2ILhVeTkbDAYEvesCCBsNo/" style="text-decoration: none; color: white;">
    <p class="text-base font-medium">Tumhi Dekho Na</p>
    <p class="text-xs text-gray-400">Sonu Nigam and Alka Yagnik</p>
</a>
</div>
</div>
<div class="text-gray-400">2006</div>
<div class="text-gray-400">4:56</div>
</div>

<div class="flex justify-between items-center py-3">
    <div class="flex items-center space-x-4">
        
        <div>
            <a href="/music/0fryaPZ0trTqvVKn7AZel9/" style="text-decoration: none; color: white;">
                <p class="text-base font-medium">Gali Mein Aaj Nikla</p>
                <p class="text-xs text-gray-400">Alka Yagnik</p>
            </a>
        </div>
    </div>
    <div class="text-gray-400">1998</div>
    <div class="text-gray-400">3:30</div>
</div>

</div>
</div>
</div>
</body>
</html>

```

login.html

```
{% load static %}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login</title>
<link rel="stylesheet" type="text/css" href="{% static 'login.css' %}">
</head>

<body>
<div class="container">
<!-- streamX logo -->
<div class="logo">

<h1>streamX</h1>
</div>

<div>
<style>
h5 {
    color: red;
    font-size: 15px;
    text-align: center;
}
</style>
{% for message in messages %}
<h5>{{message}}</h5>
{% endfor %}
</div>

<form method="POST" action="">
    {% csrf_token %}
```

```

<div class="input-group">
  <input type="text" name="username" placeholder="Username">
</div>
<div class="input-group">
  <input type="password" name="password" placeholder="Password">
</div>
<button type="submit">Sign In</button>
<p class="signup-text">Don't have an account? <a href="/signup">Sign up</a></p>
</form>

<p class="footer-text">streamX © 1.0.1</p>
</div>
</body>
</html>

```

music.html

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Music player</title>
  <link rel="stylesheet" type="text/css" href="{% static 'music.css' %}">
</head>

<body>
  <!-- Sidebar remains unchanged -->
  <div class="sidebar">
    <!-- streamX Logo -->
    <div class="logo-container">
      
    </div>

```

```

<!-- sidebar section -->
<div class="sidebar-section">
  <a href="/" class="sidebar-link">
    <span class="sidebar-link-icon">🏠 </span>
    Home
  </a>

  <a href="/search/" class="sidebar-link">
    <span class="sidebar-link-icon">🔍 </span>
    Search
  </a>
  <a href="/library/" class="sidebar-link">
    <span class="sidebar-link-icon">🎵 </span>
    Your Library
  </a>
  <a href="/likedsongs/" class="sidebar-link">
    <span class="sidebar-link-icon">❤️ </span>
    Liked Songs
  </a>
</div>
</div>

<!-- Main content area where we will add the music player -->
<div class="music-player">
  <!-- Music Player Interface -->
  <div class="player-container">
    
    <div class="song-info">
      <h3 class="song-title">{{track_name}}</h3>
      <h4 class="artist-name">{{artist_name}}</h4>
    </div>
  </div>

```

```

<div class="player-controls">
    <button id="prev" class="control-btn">&#9198;</button> <!-- previous button -->
    <button id="play-pause" class="control-btn play-btn">&#9654;</button> <!--
play/pause button -->
    <button id="next" class="control-btn">&#9197;</button> <!-- next button -->
</div>
<div class="progress-bar" id="progress-bar">
    <div class="progress" id="progress"></div>
</div>
<div class="time-stamps">
    <span id="current-time">0:00</span>
    <span id="total-duration">{{duration_text}}</span>
</div>
</div>
<audio id="audio-player" src="{{audio_url}}"></audio>
</div>

<script>
    const audioPlayer = document.getElementById('audio-player');
    const playPauseBtn = document.getElementById('play-pause');
    const progressBar = document.getElementById('progress-bar');
    const progress = document.getElementById('progress');
    const currentTimeElement = document.getElementById('current-time');
    const totalDurationElement = document.getElementById('total-duration');

    playPauseBtn.addEventListener('click', () => {
        if (audioPlayer.paused) {
            audioPlayer.play();
            playPauseBtn.innerHTML = '&#10074;&#10074;'; // change to pause icon
        }
        else {
            audioPlayer.pause();
            playPauseBtn.innerHTML = '&#9654;'; // change to play icon
        }
    });
</script>

```

```

    }

});

audioPlayer.addEventListener('timeupdate', () => {
  const progressPercentage = (audioPlayer.currentTime / audioPlayer.duration) * 100;
  progress.style.width = `${progressPercentage}%`;

  //Update the current time text
  const currentMinutes = Math.floor(audioPlayer.currentTime / 60);
  const currentSeconds = Math.floor(audioPlayer.currentTime - currentMinutes * 60);
  currentTimeElement.textContent =
`${currentMinutes}:${currentSeconds.toString().padStart(2, '0')}`;
});

progressBar.addEventListener('click', (e) => {
  const progressBarWidth = progressBar.clientWidth;
  const clickX = e.offsetX;

  const duration = audioPlayer.duration;
  audioPlayer.currentTime = (clickX / progressBarWidth) * duration;
});

//move forward by 10 seconds
document.getElementById('next').addEventListener('click', () => {
  audioPlayer.currentTime = Math.min(audioPlayer.duration, audioPlayer.currentTime + 10);
});

//move backward by 10 seconds
document.getElementById('prev').addEventListener('click', () => {
  audioPlayer.currentTime = Math.max(0, audioPlayer.currentTime - 10);
});

</script>
</body>
</html>

```

profile.html

```

{%- load static %}

<!DOCTYPE html>

<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>{{name}} Profile</title>
        <link rel="stylesheet" type="text/css" href="{% static 'profile.css' %}" />
        <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.1.4/dist/tailwind.min.css"
        rel="stylesheet">
        <style>
            .artist-header {
                background-image: url('assets/bayaan.jpg');
                background-size: cover;
                background-position: center;
                height: 60vh;
            }
        </style>
    </head>

    <body class="bg-black text-white flex">
        <!-- Sidebar -->
        <div class="sidebar">
            <!-- streamX Logo -->
            <div class="logo-container">
                
            </div>
            <!-- sidebar section -->
            <div class="sidebar-section">
                <a href="/" class="sidebar-link">

```

```

<span class="sidebar-link-icon">🏠 </span>
Home
</a>
<a href="/search" class="sidebar-link">
  <span class="sidebar-link-icon">🔍 </span>
Search
</a>
<a href="/library/" class="sidebar-link">
  <span class="sidebar-link-icon">🎵 </span>
Your Library
</a>
<a href="/likedsongs/" class="sidebar-link">
  <span class="sidebar-link-icon">❤️ </span>
Liked Songs
</a>
</div>
</div>

<!-- Main Content --&gt;
&lt;div class="flex-1"&gt;
  &lt;!-- Artist Header --&gt;
  &lt;div class="relative"&gt;
    &lt;!-- Increased height with inline style --&gt;
    &lt;img src="{{headerUrl}}" alt="{{name}}" style="height: 600px; object-fit: cover; width: 100%;"&gt;
    &lt;div class="absolute bottom-0 p-5 w-full" style="background: rgba(0, 0, 0, 0.5);"&gt;
      &lt;div class="flex items-center"&gt;
        &lt;svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 text-blue-500" viewBox="0 0 24 24" fill="currentColor"&gt;
          &lt;!-- Your SVG path for the verified icon --&gt;
          &lt;path d="M12 0C5.373 0 0 5.373 0 12S5.373 12 12 12 12-5.373 12-12S18.627 0 12 0zm0 18.1c-3.374 0-6.1-2.726-6.1-6.1 0-3.374 2.726-6.1 6.1-6.1 3.374 0 6.1 2.726 6.1 6.1 0 3.374-2.726 6.1-6.1 6.1zm3.95-8.95l-4.4 4.4-2.1-2.1-1.45 1.45 3.55 3.55 5.85-5.85-1.45-1.45z"/&gt;
      &lt;/div&gt;
    &lt;/div&gt;
  &lt;/div&gt;
&lt;/div&gt;
</pre>

```

```

</svg>
<span class="text-sm text-gray-300 ml-1">Verified Artist</span>
</div>
<h1 class="text-5xl font-bold text-white mt-2">{{name}}</h1><br />
<p class="text-gray-300">{{monthlyListeners}} monthly listeners</p>
</div>
</div>

<!-- Popular Tracks Section -->
<div class="p-5">
  <h2 class="text-2xl font-bold mb-4">Songs</h2>
  <div class="divide-y divide-gray-800">
    <!-- Single Track Row -->
    {% for track in topTracks %}
      <div class="grid grid-cols-3 items-center py-3">
        <div class="flex items-center space-x-4">
          
          <div>
            <a href="/music/{{track.id}}">
              <p class="text-base font-medium">{{track.name}}</p>
              <p class="text-xs text-gray-400">{{name}}</p>
            </a>
          </div>
        </div>
        <div class="justify-self-end text-gray-400">{{track.playCount}}</div>
        <div class="justify-self-end text-gray-400">{{track.durationText}}</div>
      </div>
    {% endfor %}
  </div>
</div>
</body>
</html>

```

search.html

```

{%- load static %}

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search</title>
    <link rel="stylesheet" type="text/css" href="{% static 'search-style.css' %}" />
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.1.4/dist/tailwind.min.css"
rel="stylesheet">
    <style>
      .artist-header {
        background-image:
          url('https://cdn.punchng.com/wp-content/uploads/2023/05/20120536/Rema.jpeg');
        background-size: cover;
        background-position: center;
        height: 60vh;
      }
    </style>
  </head>

  <body class="bg-black text-white flex">
    <!-- Sidebar -->
    <div class="sidebar">
      <!-- streamX Logo -->
      <div class="logo-container">
        
      </div>
      <!-- sidebar section -->
      <div class="sidebar-section">
        <a href="/" class="sidebar-link">

```

```

<span class="sidebar-link-icon">🏠 </span>
Home
</a>
<a href="/search" class="sidebar-link active">
  <span class="sidebar-link-icon">🔍 </span>
Search
</a>
<a href="/library/" class="sidebar-link">
  <span class="sidebar-link-icon">🎵 </span>
Your Library
</a>
<a href="/likedsongs/" class="sidebar-link">
  <span class="sidebar-link-icon">❤️ </span>
Liked Songs
</a>
</div>
</div>

```

```

<!-- Main Content -->
<div class="flex-1">
<br>
<br>

<!-- Search Box -->
<header class="header-container">
<form action="" method="POST">
  {% csrf_token %}
  <div class="search-bar" style="width: 390px;">
    <input type="search" name="search_query" placeholder="Search" aria-label="Search">
    <button type="submit" aria-label="Search">
      🔎
    </button>
  </div>
</form>
</header>

```

```
</div>
</form>
</header>

<!-- Popular Tracks Section -->
<div class="p-5">

<br>
<h2 class="text-2xl font-bold mb-4">Search Results {{search_results_count}} results</h2>
<div class="divide-y divide-gray-800">
<!-- Single Track Row -->
{% for track in track_list %}
<div class="flex justify-between items-center py-3">
<div class="flex items-center space-x-4">

<div>
<a href="/music/{{track.trackid}}">
<p class="text-base font-medium">{{track.track_name}}</p>
<p class="text-xs text-gray-400">{{track.artist_name}}</p>
</a>
</div>
</div>
<div class="text-gray-400">{{track.duration}}</div>
</div>
{% endfor %}
</div>
</div>
</body>
</html>
```

signup.html

```

{%- load static %}

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Signup</title>
    <link rel="stylesheet" type="text/css" href="{% static 'signup.css' %}">
  </head>

  <body>
    <div class="container">
      <div class="logo">
        
        <h1>streamX</h1>
      </div>
      <div>
        <style>
          h5 {
            color: red;
            font-size: 15px;
            text-align: center;
          }
        </style>
        {% for message in messages %}
          <h5>{{message}}</h5>
        {% endfor %}
      </div>

      <form action="" method="POST" class="bg-black">
        {% csrf_token %}
        <div class="input-group">

```

```
<input type="text" name="username" placeholder="Username">
</div>
<div class="input-group">
  <input type="email" name="email" placeholder="Email">
</div>
<div class="input-group">
  <input type="password" name="password" placeholder="Password">
</div>
<div class="input-group">
  <input type="password" name="password2" placeholder="Repeat Password">
</div>
<button type="submit">Register</button>
<p class="signin-text">Already have an account? <a href="/login">Sign in</a></p>
</form>
<p class="footer-text">streamX &copy; 1.0.1</p>
</div>
</body>
</html>
```

III. STATIC

index.css

```
/* General Reset */  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
    font-family: 'Arial', sans-serif;  
}
```

```
body {  
    color: #fff;  
    background-color: #121212;  
    display: flex;  
    height: 100vh;  
}
```

```
/* Sidebar Styles */  
.sidebar {  
    background-color: #000;  
    width: 240px;  
    padding: 20px;  
    display: flex;  
    flex-direction: column;  
}
```

```
/* Content Styles */  
.content {  
    flex-grow: 1;  
    overflow-y: auto;  
}
```

```
/* Header Styles */
header {
    background-color: #181818;
    padding: 20px;
    display: flex;
    justify-content: space-between;
}

/* Section Styles */
section {
    margin: 10px;
    display: flex;
}

/* Card Styles */
.card {
    display: inline-block;
    background-color: #282828;
    margin-right: 20px;
    border-radius: 4px;
    width: 200px;
    height: 200px;
    background-size: cover;
    background-position: center;
    position: relative;
}

.card .title {
    position: absolute;
    bottom: 10px;
    left: 10px;
    color: white;
    font-size: 16px;
}
```

```
/* Scrollbar Styles */
::-webkit-scrollbar {
    width: 12px;
}

::-webkit-scrollbar-track {
    background: #0e0e0e;
}

::-webkit-scrollbar-thumb {
    background-color: #ffffff;
    border-radius: 20px;
    border: 3px solid #0e0e0e;
}

/* Navigation Sidebar Styles */
.sidebar {
    color: #b3b3b3;
    padding: 25px 10px;
    font-size: 14px;
}

.sidebar a {
    text-decoration: none;
    color: inherit;
    display: block;
    padding: 5px 15px;
    margin: 10px 0;
    border-radius: 4px;
    transition: background-color 0.3s ease;
}

.sidebar a:hover,
.sidebar a.active {
```

```
background-color: #282828;  
}
```

```
/* Search Bar Styles */  
.search-bar {  
    background-color: #fff;  
    border-radius: 4px;  
    padding: 10px 15px;  
    margin: 20px 0;  
}
```

```
/* Main Content Tiles */
```

```
.tile {  
    display: inline-flex;  
    flex-direction: column;  
    width: 150px;  
    margin: 0 10px 20px;  
    text-align: center;  
    color: #fff;  
}
```

```
.tile img {  
    width: 100%;  
    border-radius: 8px;  
    margin-bottom: 8px;  
}
```

```
.tile .title {  
    font-size: 16px;  
    font-weight: 600;  
    overflow: hidden;  
    text-overflow: ellipsis;  
    white-space: nowrap;  
}
```

```
.tile .subtitle {  
    font-size: 12px;  
    color: #b3b3b3;  
}  
  
/* Main Content Section Heading */  
.content h2 {  
    color: #fff;  
    font-size: 24px;  
    margin-bottom: 20px;  
}  
  
/* Scrollable Container for Tiles */  
.scrollable-row {  
    display: flex;  
    overflow-x: auto;  
    padding-bottom: 20px;  
}  
  
.scrollable-row::-webkit-scrollbar {  
    height: 8px;  
}  
  
.scrollable-row::-webkit-scrollbar-thumb {  
    background-color: #b3b3b3;  
    border-radius: 10px;  
}  
  
.player .song-info {  
    display: flex;  
    align-items: center;  
}
```

```
.player img {  
    width: 50px;  
    height: 50px;  
    margin-right: 15px;  
}
```

```
.player .song-title {  
    font-size: 16px;  
    font-weight: bold;  
}
```

```
.player .artist-name {  
    font-size: 14px;  
}
```

```
.artist-card {  
    width: 230px;  
    height: 255px;  
    background-color: #181818;  
    border-radius: 10px;  
    padding: 16px;  
    box-shadow: 4px 8px rgba(0,0,0,0.5);  
    text-align: center;  
    overflow: hidden;  
    margin: 5px 20px;  
    display: flex;  
    flex-direction: column;  
}
```

```
.artist-card img {  
    margin-top: 10px;  
    width: 120px;  
    height: 120px;  
    border-radius: 50%;
```

```
    object-fit: cover;  
    margin-bottom: 8px;  
}  
  
}
```

```
.artist-card .artist-name {  
    font-size: 18px;  
    color: #fff;  
    margin-top: 8px;  
}
```

```
.artist-card .artist-title {  
    font-size: 13px;  
    color: #b3b3b3;  
}
```

```
.close-btn {  
    position: absolute;  
    top: 0;  
    right: 0;  
    background: none;  
    border: none;  
    color: #fff;  
    font-size: 24px;  
    padding: 16px;  
}
```

```
.header-container {  
    background: #121212;  
    padding: 20px;  
    display: flex;  
    flex-direction: column;  
}
```

```
.search-bar {
```

```
display: flex;  
align-items: center;  
justify-content: space-between;  
background-color: #2a2a2a;  
border-radius: 20px;  
padding: 10px;  
margin: 0 0px;  
}  
  
.search-bar input[type="search"] {  
flex-grow: 1;  
background-color: transparent;  
border: none;  
padding-left: 20px;  
padding-right: 20px;  
color: #fff;  
font-size: 18px; /* Increase font size for bigger appearance */  
}  
  
.search-bar button {  
background: none;  
border: none;  
color: #fff;  
font-size: 24px;  
padding: 0 10px;  
cursor: pointer;  
}  
  
.search-title {  
color: #fff;  
font-size: 22px;  
padding-top: 20px;  
}
```

```
/* Sidebar Styles */
.sidebar {
    background: #000;
    width: 250px;
    height: 100vh;
    position: fixed;
    overflow: auto;
    padding: 9px;
    display: flex;
    flex-direction: column;
}

/* Sidebar Bottom Section Styles */
.sidebar-section {
    background: #181818;
    border-radius: 10px;
    padding: 15px;
    margin-bottom: 9px;
    overflow: hidden;
}

.sidebar-link {
    color: #b3b3b3;
    text-decoration: none;
    display: flex;
    align-items: center;
    margin-bottom: 10px;
    transition: color 0.3s;
}

.sidebar-link:hover,
.sidebar-link.active {
    color: #fff;
```

```
}
```

```
.sidebar-link-icon {  
    margin-right: 15px;  
    display: inline-block;  
    width: 24px; /* Adjust size as needed */  
    text-align: center;  
}
```

```
.sidebar-link-text {  
    flex-grow: 1;  
}
```

```
.sidebar-link-subtext {  
    margin-left: 10px;  
    background-color: #b3b3b3;  
    color: #000;  
    border-radius: 4px;  
    padding: 2px 6px;  
    font-size: 12px;  
    text-transform: uppercase;  
}
```

```
/* Additional styles */
```

```
.browse-all {  
    display: grid;  
    grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));  
    gap: 95px;  
    padding: 10px;  
    background-color: #121212; /* or any other color you prefer */  
}
```

```
.category-card {  
    display: flex;
```

```
flex-direction: column;  
align-items: center;  
justify-content: center;  
padding: 10px;  
border-radius: 8px;  
background: #181818;  
color: white;  
text-align: center;  
box-shadow: 0 4px 8px rgba(0,0,0,0.5);  
height: 240px;  
width: 220px;  
}  
  
.category-card img {  
width: 100%;  
height: auto;  
margin-top: 12px;  
margin-bottom: 10px;  
border-radius: 4px;  
}  
  
.category-title {  
font-size: 14px;  
font-weight: bold;  
margin-bottom: 12px;  
}  
  
/* Add more specific styles for badges if needed */  
.browse-all-title {  
font-size: 24px;  
font-weight: bold;  
color: #fff;  
margin-left: 20px; /* Align with the padding of the grid */  
padding-top: 20px; /* Space above the title */
```

```
}
```

```
/* Artist Header Styles */
```

```
.artist-header {  
    padding: 20px 0;  
    border-bottom: 1px solid #282828;  
}
```

```
.artist-name {
```

```
    font-size: 48px;  
    font-weight: 700;  
    margin-bottom: 5px;
```

```
}
```

```
.listeners {
```

```
    font-size: 18px;  
    color: #b3b3b3;
```

```
}
```

```
/* Verified Artist Icon */
```

```
.verified-artist img {  
    display: block;  
    margin-left: 10px;
```

```
}
```

```
/* Songs List Styles */
```

```
.artist-songs ul {  
    padding: 20px 0;
```

```
}
```

```
.song-item {
```

```
    padding: 10px 0;  
    border-bottom: 1px solid #282828;
```

```
}
```

```
.song-item .song-title {  
    font-size: 24px;  
    font-weight: 500;  
}  
  
.song-item .play-count {  
    font-size: 18px;  
    color: #b3b3b3;  
    margin-right: 20px;  
}  
  
.song-item .song-duration {  
    font-size: 18px;  
    color: #b3b3b3;  
}
```

login.css

```
body {  
    background-color: #000;  
    color: #fff;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
  
    margin: 0;  
    font-family: Arial, sans-serif;  
}
```

```
.container {  
    width: 100%;  
    max-width: 320px;  
    padding: 0 16px;
```

```
  box-sizing: border-box;  
}
```

```
.logo {  
  text-align: center;  
  margin-bottom: 40px;  
}
```

```
.logo img {  
  width: 80px;  
  height: 80px;  
}
```

```
.logo h1 {  
  font-size: 32px;  
  font-weight: bold;  
  margin-top: 8px;  
}
```

```
form {  
  background-color: #000;  
}
```

```
.input-group {  
  margin-bottom: 16px;  
}
```

```
.input-group input {  
  width: 100%;  
  padding: 8px;  
  border-radius: 4px;  
  border: 1px solid #444;  
  background-color: #333;  
  color: #fff;
```

```
}
```

```
button {  
    width: 100%;  
    padding: 12px;  
    background-color: #1DB954;  
    border: none;  
    border-radius: 4px;  
    color: #fff;  
    font-size: 16px;  
    font-weight: bold;
```

```
    cursor: pointer;  
    margin-bottom: 16px;  
}
```

```
button:hover {  
    background-color: #1AA34A;  
}
```

```
.signup-text {  
    text-align: center;  
    font-size: 12px;  
    color: #999;  
}
```

```
.signup-text a {  
    color: #1DB954;  
    text-decoration: none;  
}
```

```
.footer-text {  
    text-align: center;
```

```
font-size: 12px;  
color: #777;  
margin-top: 16px;  
}
```

music.css

```
body {  
    margin: 0;  
    padding: 0;  
    background: #121212;  
    font-family: Arial, sans-serif;  
    color: #fff;  
}
```

```
.logo-container img {  
    width: 100%;  
    height: auto;  
    padding: 0 20px;  
    box-sizing: border-box;  
}
```

```
.music-player {  
    margin-left: 240px;  
    padding: 20px;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: calc(100vh - 40px);  
}
```

```
/* Music Player Container */  
.music-player {  
    flex-grow: 1;
```

```
padding: 20px;  
}  
  
/* Music Player Interface */  
.player-container {  
    max-width: 400px;  
    text-align: center;  
    padding: 20px;  
}  
  
.album-cover {  
    border-radius: 10px;  
    margin-bottom: 5px;  
}  
  
.song-info {  
    margin-bottom: 30px;  
}  
  
.song-title {  
    color: #fff;  
    font-size: 2.5em;  
    margin-bottom: 0.3em;  
}  
  
.artist-name {  
    color: #b3b3b3;  
  
    font-size: 1.5em;  
}  
  
.player-controls {  
    margin-bottom: 20px;  
}
```

```
.control-btn {  
    background: none;  
    border: none;  
    color: #fff;  
    font-size: 2em;  
    margin: 0 20px;  
    cursor: pointer;  
}
```

```
.play-btn {  
    color: rgb(88, 161, 245);  
    font-size: 3em;  
}
```

```
.progress-bar {  
    width: 100%;  
    background: #404040;  
    border-radius: 2px;  
    height: 5px;  
    position: relative;  
  
    margin-bottom: 10px;  
}
```

```
.progress {  
    background: #23b9f5;  
    width: 1%;  
    height: 100%;  
    border-radius: 2px;  
}
```

```
.time-stamps {  
    display: flex;
```

```
justify-content: space-between;
font-size: 1.2em;
}

/* Sidebar Styles */
.sidebar {
    background-color: #000;
    width: 240px;
    padding: 20px;
    display: flex;
    flex-direction: column;
}

section {
    margin: 20px;
}

/* Navigation Sidebar Styles */
.sidebar {
    color: #b3b3b3;
    padding: 25px 10px;
    font-size: 14px;
}

.sidebar a {
    text-decoration: none;
    color: inherit;
    display: block;
    padding: 5px 15px;
    margin: 10px 0;
    border-radius: 4px;
    transition: background-color 0.3s ease;
}
```

```
.sidebar {  
    background: #000;  
    width: 250px;  
    height: 100vh;  
    position: fixed;  
    overflow: auto;  
    padding: 9px;  
    display: flex;  
    flex-direction: column;  
}  
  
/* Sidebar Bottom Section Styles */  
.sidebar-section {  
    background: #181818;  
    border-radius: 10px;  
    padding: 15px;  
    margin-bottom: 9px;  
    overflow: hidden;  
}  
  
.sidebar-link {  
    color: #b3b3b3;  
    text-decoration: none;  
    display: flex;  
    align-items: center;  
    margin-bottom: 10px;  
    transition: color 0.3s;  
}  
  
.sidebar-link:hover,  
.sidebar-link.active {  
    color: #fff;  
}
```

```
.sidebar-link-icon {
    margin-right: 15px;
    display: inline-block;
    width: 24px; /* Adjust size as needed */
    text-align: center;
}
```

```
.sidebar-link-text {
    flex-grow: 1;
}
```

```
.sidebar-link-subtext {
    margin-left: 10px;
    background-color: #b3b3b3;
    color: #000;
    border-radius: 4px;
    padding: 2px 6px;
    font-size: 12px;
    text-transform: uppercase;
}
```

profile.css

```
.artist-overlay {
    background: rgba(0, 0, 0, 0.7);
    height: 100%;

    display: flex;
    flex-direction: column;
    justify-content: end;
    padding: 2rem;
}
```

```
.artist-info {
```

```
color: rgb(255, 255, 255);
}

.verified-badge {
  color: #1DB954;
}

/* Responsive Design */
@media (min-width: 768px) {
  .sidebar {
    flex-basis: 250px;
    flex-shrink: 0;
  }

  .main-content {
    flex-grow: 1;
  }
}

/* Sidebar Styles */
.sidebar {
  background-color: #000;
  width: 240px;
  padding: 20px;
  display: flex;
  flex-direction: column;
}

/* Navigation Sidebar Styles */
.sidebar {
  color: #b3b3b3;
  padding: 25px 10px;
  font-size: 14px;
}
```

```
.sidebar a {  
    text-decoration: none;  
    color: inherit;  
    display: block;  
    padding: 5px 15px;  
  
    margin: 10px 0;  
    border-radius: 4px;  
    transition: background-color 0.3s ease;  
}  
  
.sidebar a:hover,  
.sidebar a.active {  
    background-color: #282828;  
}  
  
/* Sidebar Bottom Section Styles */  
.sidebar-section {  
    background: #181818;  
    border-radius: 10px;  
    padding: 15px;  
    margin-bottom: 9px;  
    overflow: hidden;  
}  
  
.sidebar-link {  
    color: #b3b3b3;  
    text-decoration: none;  
    display: flex;  
    align-items: center;  
    margin-bottom: 10px;  
    transition: color 0.3s;  
}
```

```
.sidebar-link:hover,  
.sidebar-link.active {  
    color: #fff;  
}  
  
.sidebar-link-icon {  
    margin-right: 15px;  
    display: inline-block;  
    width: 24px; /* Adjust size as needed */  
    text-align: center;  
}  
  
.sidebar-link-text {  
    flex-grow: 1;  
}  
  
.sidebar-link-subtext {  
    margin-left: 10px;  
    background-color: #b3b3b3;  
    color: #000;  
    border-radius: 4px;  
    padding: 2px 6px;  
    font-size: 12px;  
    text-transform: uppercase;  
}  
  
/* updated styles tailwind and css */  
body {  
    background-color: black;  
    color: white;  
    display: flex;  
}  
  
.artist-header {
```

```
background-image:  
url('https://cdn.punchng.com/wp-content/uploads/2023/05/20120536/Rema.jpeg');  
background-size: cover;  
background-position: center;  
height: 60vh;  
}  
  
.sidebar {  
width: 250px;  
background-color: #111;  
padding: 20px;  
box-sizing: border-box;  
}  
.logo-container img {  
width: 100%;  
height: auto;  
}  
  
.sidebar-section {  
margin-top: 20px;  
}  
  
.sidebar-link {  
display: block;  
  
color: white;  
text-decoration: none;  
padding: 10px 0;  
font-size: 16px;  
}  
  
.sidebar-link:hover {  
background-color: #333;  
}
```

```
.sidebar-link2 {  
    display: block;  
    color: white;  
    background: #000;  
    text-decoration: none;  
    padding: 10px 0;  
    font-size: 16px;  
}  
  
}
```

```
.sidebar-link2:hover {  
    background-color: #333;  
}  
  
}
```

```
.sidebar-link-icon {  
    margin-right: 10px;  
}  
  
}
```

```
.flex-1 {  
    flex: 1;  
    display: flex;  
    flex-direction: column;  
}  
  
}
```

```
.relative {  
    position: relative;  
}  
  
}
```

```
.relative img {  
    height: 600px;  
    object-fit: cover;  
    width: 100%;  
}  
  
}
```

```
.relative .absolute {  
  position: absolute;  
  bottom: 0;  
  padding: 20px;  
  width: 100%;  
  background: rgba(0, 0, 0, 0.5);  
}  
  
.relative .absolute .flex {  
  display: flex;  
  align-items: center;  
}  
  
.relative .absolute svg {  
  height: 24px;  
  width: 24px;  
  color: #3b82f6; /* Tailwind's blue-500 color */  
}  
  
.relative .absolute .text-sm {  
  font-size: 14px;  
  margin-left: 5px;  
}  
  
.relative .absolute .text-5xl {  
  font-size: 3rem;  
  font-weight: bold;  
  margin-top: 10px;  
}  
  
.relative .absolute .text-gray-300 {  
  color: #d1d5db; /* Tailwind's gray-300 color */  
}
```

```
.p-5 {  
  padding: 20px;  
}  
  
.text-2xl {  
  font-size: 1.5rem;  
  font-weight: bold;  
  margin-bottom: 16px;  
}  
  
.divide-y > :not([hidden]) ~ :not([hidden]) {  
  border-top-width: 1px;  
  border-color: #374151; /* Tailwind's gray-800 color */  
}  
  
.flex {  
  display: flex;  
}  
  
.justify-between {  
  justify-content: space-between;  
}  
  
.items-center {  
  align-items: center;  
}  
  
.py-3 {  
  padding-top: 12px;  
  padding-bottom: 12px;  
}  
  
.space-x-4 > :not([hidden]) ~ :not([hidden]) {  
  margin-left: 16px;
```

```
}
```

```
.w-12 {
  width: 48px;
}
```

```
.h-12 {
  height: 48px;
}
```

```
.text-base {
  font-size: 1rem;
  font-weight: medium;
}
```

```
.text-xs {
  font-size: 0.75rem;
}
```

```
.text-gray-400 {
  color: #9ca3af; /* Tailwind's gray-400 color */
}
```

```
.text-green-500 {
  color: #10b981; /* Tailwind's green-500 color */
}
```

search-style.css

```
.artist-overlay {
  background: rgba(0, 0, 0, 0.7);
  height: 100%;
  display: flex;
  flex-direction: column;
```

```
justify-content: end;
padding: 2rem;
}

.artist-info {
color: white;
}

.verified-badge {
color: #1DB954;
}

/* Responsive Design */
@media (min-width: 768px) {
.sidebar {
flex-basis: 250px;
flex-shrink: 0;
}

.main-content {
flex-grow: 1;
}
}

/* Sidebar Styles */
.sidebar {
background-color: #000;
width: 240px;
padding: 20px;
display: flex;
flex-direction: column;
}

/* Navigation Sidebar Styles */
```

```
.sidebar {  
    color: #b3b3b3;  
    padding: 25px 10px;  
    font-size: 14px;  
}  
  
.sidebar a {  
    text-decoration: none;  
    color: inherit;  
    display: block;  
    padding: 5px 15px;  
    margin: 10px 0;  
    border-radius: 4px;  
    transition: background-color 0.3s ease;  
}  
  
.sidebar a:hover,  
.sidebar a.active {  
    background-color: #282828;  
}  
  
/* Sidebar Styles */  
/* Sidebar Bottom Section Styles */  
.sidebar-section {  
    background: #181818;  
    border-radius: 10px;  
    padding: 15px;  
    margin-bottom: 9px;  
    overflow: hidden;  
}  
  
.sidebar-link {  
    color: #b3b3b3;  
    text-decoration: none;
```

```
display: flex;
align-items: center;
margin-bottom: 10px;
transition: color 0.3s;
}

.sidebar-link:hover,
.sidebar-link.active {
color: #fff;
}

.sidebar-link-icon {
margin-right: 15px;
display: inline-block;
width: 24px; /* Adjust size as needed */
text-align: center;
}

.sidebar-link-text {
flex-grow: 1;
}

.sidebar-link-subtext {
margin-left: 10px;
background-color: #b3b3b3;
color: #000;
border-radius: 4px;
padding: 2px 6px;
font-size: 12px;
text-transform: uppercase;
}

/* Search Bar Styles */
.search-bar {
```

```
background-color: #fff;  
border-radius: 4px;  
padding: 10px 15px;  
margin: 20px 0;  
}
```

```
.search-bar {  
display: flex;  
align-items: center;  
justify-content: space-between;  
background-color: #2a2a2a;  
border-radius: 20px;  
padding: 10px;  
margin: 0 0px;  
}
```

```
.search-bar input[type="search"] {  
flex-grow: 1;  
background-color: transparent;  
border: none;  
padding-left: 20px;  
padding-right: 20px;  
color: #fff;  
font-size: 18px; /* Increase font size for bigger appearance */  
}
```

```
.search-bar button {  
background: none;  
border: none;  
color: #fff;  
font-size: 24px;  
padding: 0 10px;  
cursor: pointer;  
}
```

```
.search-title {  
    color: #fff;  
    font-size: 22px;  
    padding-top: 20px;  
}
```

signup.css

```
body {  
    background-color: #000;  
    color: #fff;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
    margin: 0;  
    font-family: Arial, sans-serif;  
}
```

```
.container {  
    width: 100%;  
    max-width: 320px;  
    padding: 0 16px;  
    box-sizing: border-box;  
}
```

```
.logo {  
    text-align: center;  
    margin-bottom: 40px;  
}
```

```
.logo img {  
    width: 80px;
```

```
height: 80px;  
}  
  
.logo h1 {  
    font-size: 32px;  
    font-weight: bold;  
    margin-top: 8px;  
}  
  
form {  
    background-color: #000;  
}  
  
.input-group {  
    margin-bottom: 16px;  
}  
  
.input-group input {  
    width: 100%;  
    padding: 8px;  
    border-radius: 4px;  
    border: 1px solid #444;  
    background-color: #333;  
    color: #fff;  
}  
  
button {  
    width: 100%;  
    padding: 12px;  
    background-color: #1DB954;  
    border: none;  
    border-radius: 4px;  
    color: #fff;  
    font-size: 16px;
```

```
font-weight: bold;  
cursor: pointer;  
margin-bottom: 16px;  
}  
  
button:hover {  
background-color: #1AA34A;  
}  
  
.signin-text {  
text-align: center;  
font-size: 12px;  
color: #999;  
}  
  
.signin-text a {  
color: #1DB954;  
text-decoration: none;  
}  
  
.footer-text {  
text-align: center;  
font-size: 12px;  
color: #777;  
margin-top: 16px;  
}
```

IV. MUSIC

apps.py

```
from django.apps import AppConfig  
class MusicConfig(AppConfig):  
    default_auto_field = 'django.db.models.BigAutoField'  
    name = 'music'
```

urls.py

```
from django.urls import path  
from . import views  
urlpatterns = [  
    path('', views.index, name='index'),  
    path('login/', views.login, name='login'),  
    path('signup/', views.signup, name='signup'),  
    path('logout/', views.logout, name='logout'),  
    path('music/<str:pk>/', views.music, name='music'),  
    path('profile/<str:pk>/', views.profile, name='profile'),  
    path('search/', views.search, name='search'),  
    path('library/', views.library, name='library'),
```

```

path('likedsongs/', views.likedsongs, name='likedsongs'),
path('latamangeshkar/', views.latamangeshkar, name='latamangeshkar'),
]

```

views.py

```

from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from django.contrib import messages
from django.contrib.auth.models import User, auth
from django.contrib.auth.decorators import login_required
import requests
from bs4 import BeautifulSoup as bs
import re

# Create your views here
def top_artists():
    url = "https://spotify-scraper.p.rapidapi.com/v1/chart/artists/top"
    #querystring = {"type": "weekly"}
    headers = {
        "x-rapidapi-key": "b44af34dffmsh9f25514dbp10f665jsnc58529c23e0c",
        "x-rapidapi-host": "spotify-scraper.p.rapidapi.com"
    }
    response = requests.get(url, headers=headers)
    response_data = response.json()

    artists_info = []

    if 'artists' in response_data:
        for artist in response_data['artists']:
            name = artist.get('name', 'No name')
            avatar_url = artist.get('visuals', {}).get('avatar', {}).get('url', 'No URL')

```

```

    artist_id = artist.get('id', 'No ID')
    artists_info.append((name, avatar_url, artist_id))
return artists_info

def top_tracks():
    url = "https://spotify-scraper.p.rapidapi.com/v1/chart/tracks/top"
    #querystring = {"type":"weekly"}
    headers = {
        "x-rapidapi-key": "b44af34dffmsh9f25514dbp10f665jsnc58529c23e0c",
        "x-rapidapi-host": "spotify-scraper.p.rapidapi.com"
    }
    response = requests.get(url, headers=headers)
    data = response.json()
    track_details = []

    if 'tracks' in data:
        shortened_data = data['tracks'][:18]
        # id, name, artist, cover url
        for track in shortened_data:
            track_id = track['id']
            track_name = track['name']
            artist_name = track['artist'][0]['name'] if track['artist'] else None
            cover_url = track['album']['cover'][0]['url'] if track['album']['cover'] else None
            track_details.append({
                'id': track_id,
                'name': track_name,
                'artist': artist_name,
                'cover_url': cover_url
            })
    else:
        print('No tracks found in the response')
return track_details

```

```

def get_audio_details(query):
    url = "https://spotify-scraper.p.rapidapi.com/v1/track/download"
    querystring = {"track": query}
    headers = {
        "x-rapidapi-key": "b44af34dffmsh9f25514dbp10f665jsnc58529c23e0c",
        "x-rapidapi-host": "spotify-scraper.p.rapidapi.com"
    }
    response = requests.get(url, headers=headers, params=querystring)

    audio_details = []

    if response.status_code == 200:
        response_data = response.json()

        if 'youtubeVideo' in response_data and 'audio' in response_data['youtubeVideo']:
            audio_list = response_data['youtubeVideo']['audio']
            if audio_list:
                first_audio_url = audio_list[0]['url']
                duration_text = audio_list[0]['durationText']
                audio_details.append(first_audio_url)
                audio_details.append(duration_text)
            else:
                print("No audio found")
        else:
            print('No audio key found')
    else:
        print('Error in fetching audio details')

    return audio_details

```

```

def get_track_image(track_id, track_name):
    url = 'https://open.spotify.com/track/' + track_id
    r = requests.get(url)

```

```

soup = bs(r.content)

image_links_html = soup.find('img', {'alt': track_name})

if image_links_html:

    image_links = image_links_html['srcset']

else:

    image_links = ""

match = re.search(r'https://i.scdn.co/image/[a-zA-Z0-9]+640w', image_links)

if match:

    url_640w = match.group().rstrip(' 640w')

else:

    url_640w = ""

return url_640w


def music(request, pk):

    track_id = pk

    url = "https://spotify-scraper.p.rapidapi.com/v1/track/metadata"

    querystring = {"trackId": track_id}

    headers = {

        "x-rapidapi-key": "b44af34dffmsh9f25514dbp10f665jsnc58529c23e0c",
        "x-rapidapi-host": "spotify-scraper.p.rapidapi.com"
    }

    response = requests.get(url, headers=headers, params=querystring)

    if response.status_code == 200:

        data = response.json()

        #track_name, artist_name

        track_name = data.get('name')

        artists_list = data.get('artists', [])

        first_artist_name = artists_list[0].get('name') if artists_list else "Unknown"

        audio_details_query = track_name + first_artist_name

        audio_details = get_audio_details(audio_details_query)

```

```

audio_url = audio_details[0]
duration_text = audio_details[1]

track_image = get_track_image(track_id, track_name)

context = {
    'track_name': track_name,
    'artist_name': first_artist_name,
    'audio_url': audio_url,
    'duration_text': duration_text,
    'track_image': track_image,
}

return render(request, 'music.html', context)

@login_required(login_url='login')

def index(request):
    artists_info = top_artists()
    top_track_list = top_tracks()
    return render(request, 'index.html')

def search(request):
    if request.method == 'POST':
        search_query = request.POST['search_query']
        url = "https://spotify-scraper.p.rapidapi.com/v1/search"
        querystring = {"term":search_query,"type":"track"}
        headers = {
            "x-rapidapi-key": "b44af34dffmsh9f25514dbp10f665jsnc58529c23e0c",
            "x-rapidapi-host": "spotify-scraper.p.rapidapi.com"
        }
        response = requests.get(url, headers=headers, params=querystring)
        track_list = []

    if response.status_code == 200:
        data = response.json()

```

```

search_results_count = data['tracks']['totalCount']
tracks = data['tracks']['items']
for track in tracks:
    track_name = track['name']
    artist_name = track['artists'][0]['name']
    duration = track['durationText']
    trackid = track['id']

    if get_track_image(trackid, track_name):
        track_image = get_track_image(trackid, track_name)
    else:
        track_image =
'https://imgv3.fotor.com/images/blog-richtext-image/music-of-the-spheres-album-cover.jpg'

    track_list.append({
        'track_name': track_name,
        'artist_name': artist_name,
        'duration': duration,
        'trackid': trackid,
        'track_image': track_image,
    })
else:
    search_results_count = 0

context = {
    'search_results_count': search_results_count,
    'track_list': track_list,
}
return render(request, 'search.html', context)

else:
    return render(request, 'search.html')

```

```

def login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']

        # check if user exists
        user = auth.authenticate(username=username, password=password)

        if user is not None: #user has an account
            #log user in
            auth.login(request, user)
            return redirect('/')

        else:
            messages.info(request, 'Invalid credentials')
            return redirect('login')

    return render(request, 'login.html')

def signup(request):
    if request.method == 'POST':
        email = request.POST['email']
        username = request.POST['username']
        password = request.POST['password']
        password2 = request.POST['password2']

        if password == password2:
            # print('Password matched')
            if User.objects.filter(email=email).exists():
                messages.info(request, 'Email is taken')
                return redirect('signup')

            elif User.objects.filter(username=username).exists():
                messages.info(request, 'Username is taken')

```

```

        return redirect('signup')

    else:
        user = User.objects.create_user(username=username, email=email,
password=password)
        user.save()

        # log user in
        user_login = auth.authenticate(username=username, password=password)
        auth.login(request, user_login)
        return redirect('/')

    else:
        messages.info(request, 'Password is not same')
        return redirect('signup')

else:
    return render(request, 'signup.html')




@login_required(login_url='login')
def logout(request):
    auth.logout(request)
    return redirect('login')


def profile(request, pk):
    artist_id = pk
    url = "https://spotify-scraper.p.rapidapi.com/v1/artist/overview"
    querystring = {"artistId":artist_id}
    headers = {
        "x-rapidapi-key": "b44af34dffmsh9f25514dbp10f665jsnc58529c23e0c",
        "x-rapidapi-host": "spotify-scraper.p.rapidapi.com"
    }
    response = requests.get(url, headers=headers, params=querystring)

```

```

if response.status_code == 200:
    data = response.json()
    name = data['name']
    monthly_listeners = data['stats']['monthlyListeners']
    header_url = data['visuals']['header'][0]['url']

    top_tracks = []

    for track in data['discography']['topTracks']:
        trackid = str(track['id'])
        trackname = str(track['name'])

        if get_track_image(trackid, trackname):
            trackimage = get_track_image(trackid, trackname)
        else:

            trackimage =
            'https://imgv3.fotor.com/images/blog-richtext-image/music-of-the-spheres-album-cover.jpg'

        track_info = {
            'id': track['id'],
            'name': track['name'],
            'durationText': track['durationText'],
            'playCount': track['playCount'],
            'track_image': trackimage
        }

        top_tracks.append(track_info)

    artist_data = {
        'name': name,
        'monthlyListeners': monthly_listeners,
        'headerUrl': header_url,

```

```

        'topTracks': top_tracks,
    }
else:
    artist_data = {}
return render(request, 'profile.html', artist_data)

```

```

def library(request):
    return render(request, 'library.html')
def likedsongs(request):
    return render(request, 'likedsongs.html')
def latamangeshkar(request):
    return render(request, 'latamangeshkar.html')

```

V. manage.py

```

#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'streamX.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

```

VI. requirements.txt

asgiref==3.8.1

beautifulsoup4==4.12.3

certifi==2024.6.2

charset-normalizer==3.3.2

Django==5.0.6

idna==3.7

mysqlclient==2.2.4

requests==2.32.3

soupsieve==2.5

sqlparse==0.5.0

tzdata==2024.1

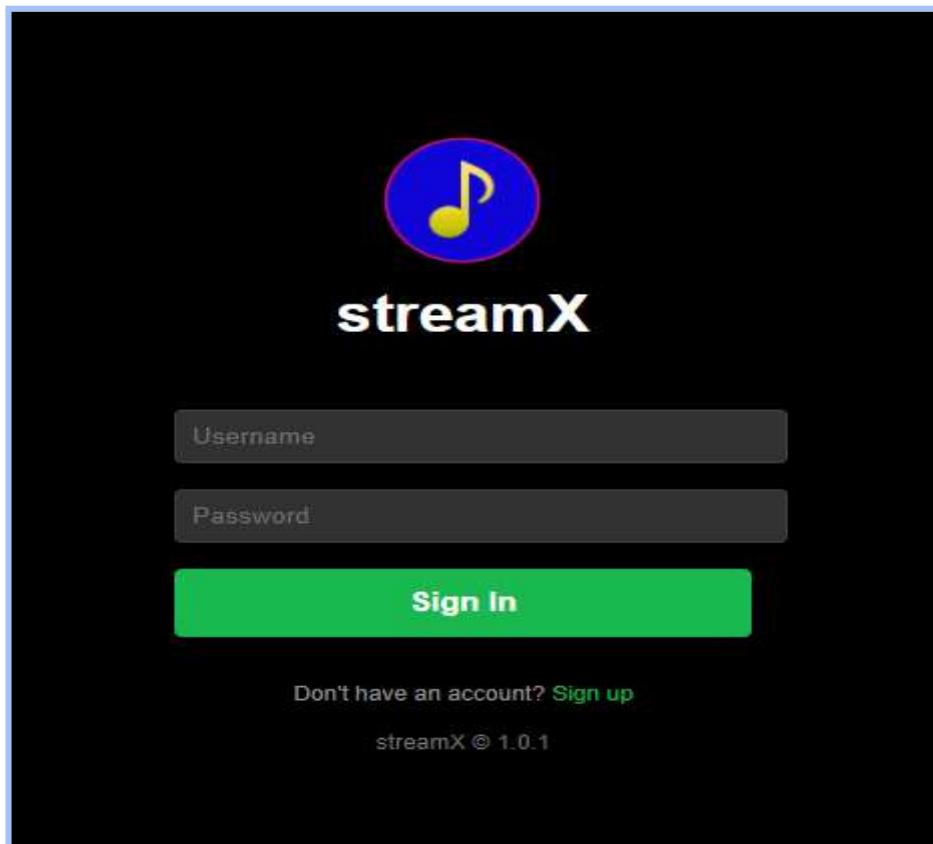
urllib3==2.2.2

Database

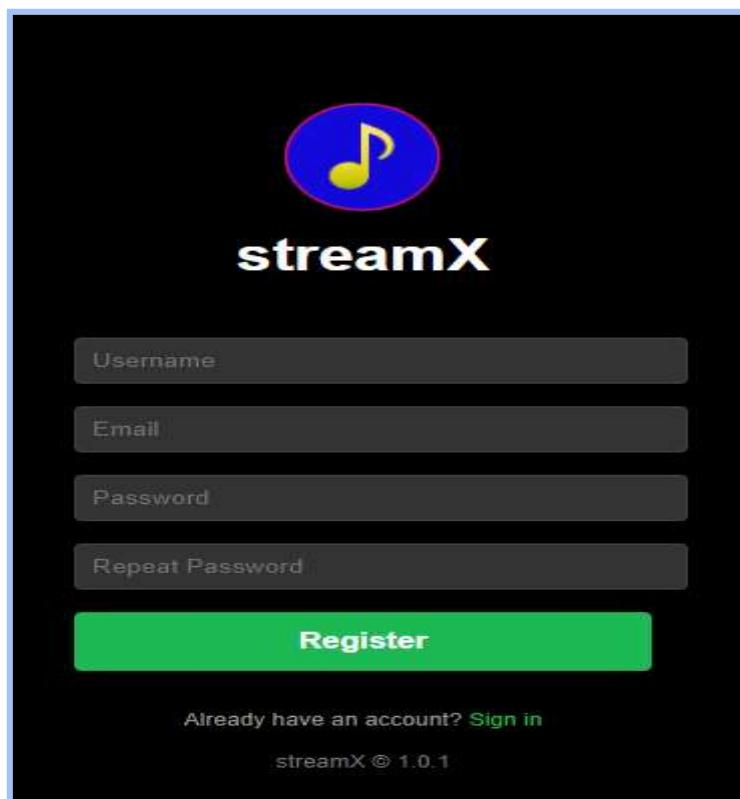
id	password	username	email
1	pbkdf2_sha256\$720000\$mmU0OvLdDeb1Ws9o5J7Dve\$5OLzb0WyR9Y1Xv5Mh2ne3b0DYUocr2Fwsg3VLc4rpI8=	edy	edy111@yahoo.com
2	pbkdf2_sha256\$720000\$ic0PdHG7f72feP8RT8u9Os\$Jj9HX7N29JksBliwygwsjCdR7iolrNY9Io+znfvEUY4=	bistirno	bistirno7@gmail.com
3	pbkdf2_sha256\$720000\$lc9EdbW5RC0fNMSCknLJ98\$+cJHzgerdvlyIMcsiXqBzkewjUBQdrFU9C/DTxq10ms=	rdc78	rdc789@gmail.com
4	pbkdf2_sha256\$720000\$wDMOgezAXZhRD7u3QTqxML\$HGmXIcn2AJqtk+h1bZUq+UKsDIN80hrcrcXNb2WqNSM=	bacillusthuringesis	bt1469@gmail.com
5	pbkdf2_sha256\$720000\$ikjYIFORUdyfIL7BzkzFwE\$iwccgDRvUDSFKgTXBiGJ7zMcmxu2J8u/YQz3pRISs8=	aumii	aumii.123@gmail.com

Project screenshot

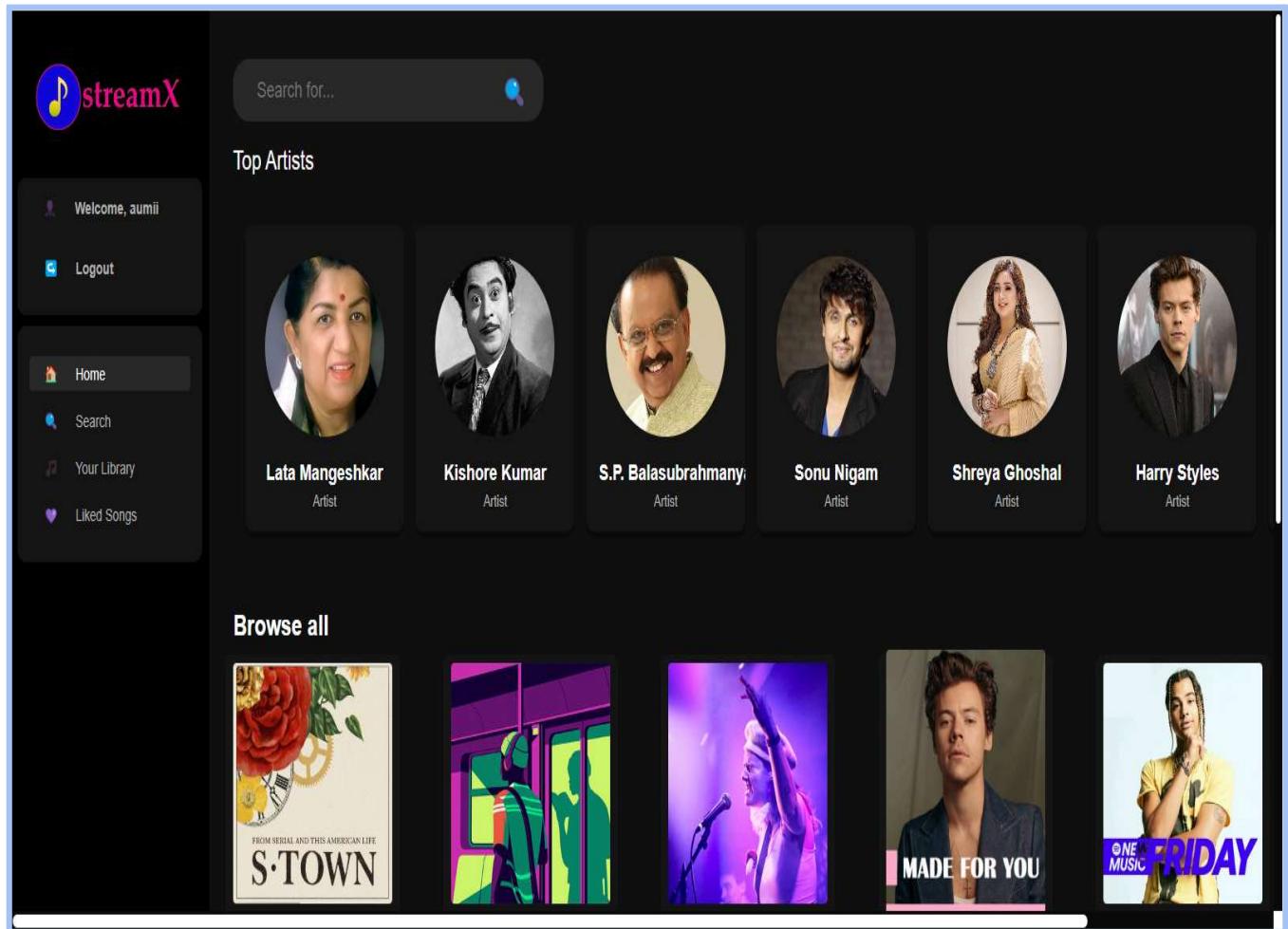
- LOGIN



- SIGNUP

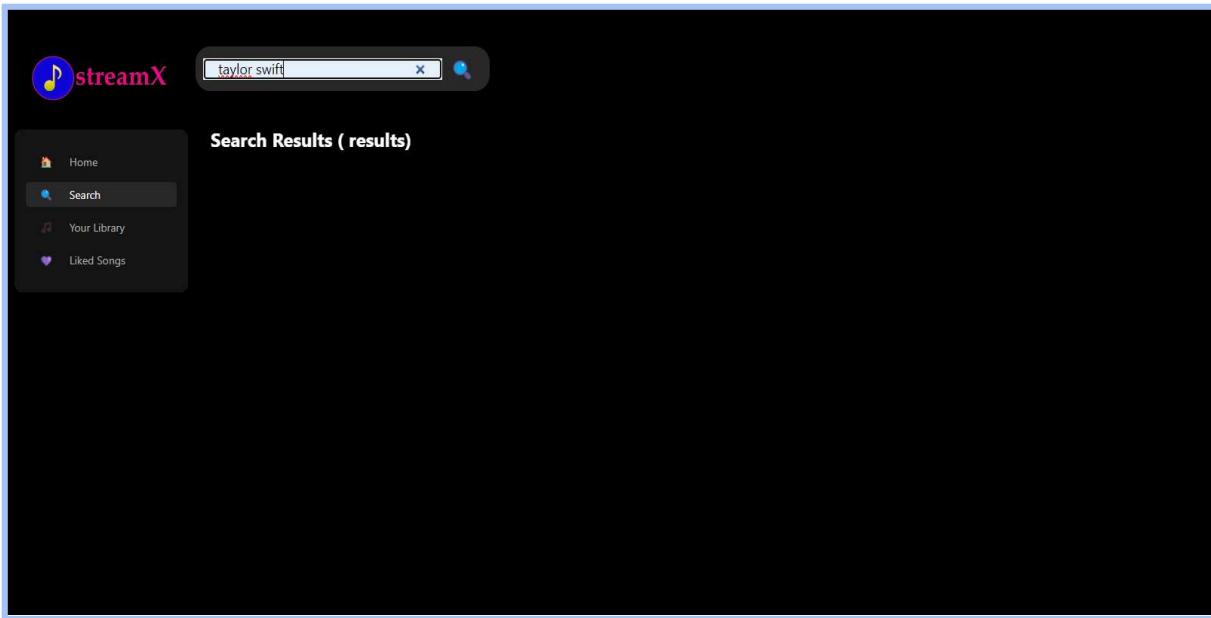


• HOME



The screenshot shows the StreamX mobile application's home screen. On the left is a vertical navigation sidebar with a dark background. At the top of the sidebar is the StreamX logo, which consists of a blue circle with a white musical note icon followed by the word "streamX" in white. Below the logo are five menu items: "Welcome, aumii" (with a user icon), "Logout" (with a sign-out icon), "Home" (selected, indicated by a grey background and a house icon), "Search" (with a magnifying glass icon), "Your Library" (with a music note icon), and "Liked Songs" (with a heart icon). To the right of the sidebar is the main content area. At the top of the main area is a search bar with the placeholder "Search for..." and a magnifying glass icon. Below the search bar is a section titled "Top Artists" featuring six artist profiles arranged in two rows of three. Each profile includes a circular portrait, the artist's name, and the word "Artist" below it. The artists listed are Lata Mangeshkar, Kishore Kumar, S.P. Balasubrahmanyam, Sonu Nigam, Shreya Ghoshal, and Harry Styles. Below this section is another titled "Browse all" which displays five thumbnail images representing different content: "S.TOWN" (a red rose and gears), "EAT" (a colorful graphic design), "MADE FOR YOU" (Harry Styles), and "FRIDAY" (Wiz Khalifa).

● SEARCH



A screenshot of the StreamX application interface, identical to the one above but with a different search term. The search bar now contains "taylor swift". The main content area is titled "Search Results (892 results)" and lists eight songs by Taylor Swift, each with a small album cover thumbnail, the song title, the artist name, and the duration. The songs listed are: Shake It Off, Fortnite (feat. Post Malone), Cruel Summer, You Belong With Me, Anti-Hero, I Can Do It With a Broken Heart, Bad Blood, and another instance of I Can Do It With a Broken Heart.

● LIBRARY

Songs

	Nahin Milta Bayaan	2019	4:45
	Falling Harry Styles	2019	4:02
	Rangeela Re Lata Mangeshkar	1970	6:63

See more liked songs...

Albums

	Hum Aapke Hain Koun	1994	15 songs
	Rockstar	2011	14 songs

See more albums...

Playlists

	Dance Mix	60 items
	Focus Ragas	37 items

See more playlists...

Focus Ragas

Albums

	Hum Aapke Hain Koun	1994	15 songs
	Rockstar	2011	14 songs

See more albums...

Playlists

	Dance Mix	60 items
	Focus Ragas	37 items

See more playlists...

Podcasts

	Gita for the Young and Restless	11 episodes
	Death, Lies and Cyanide	10 episodes

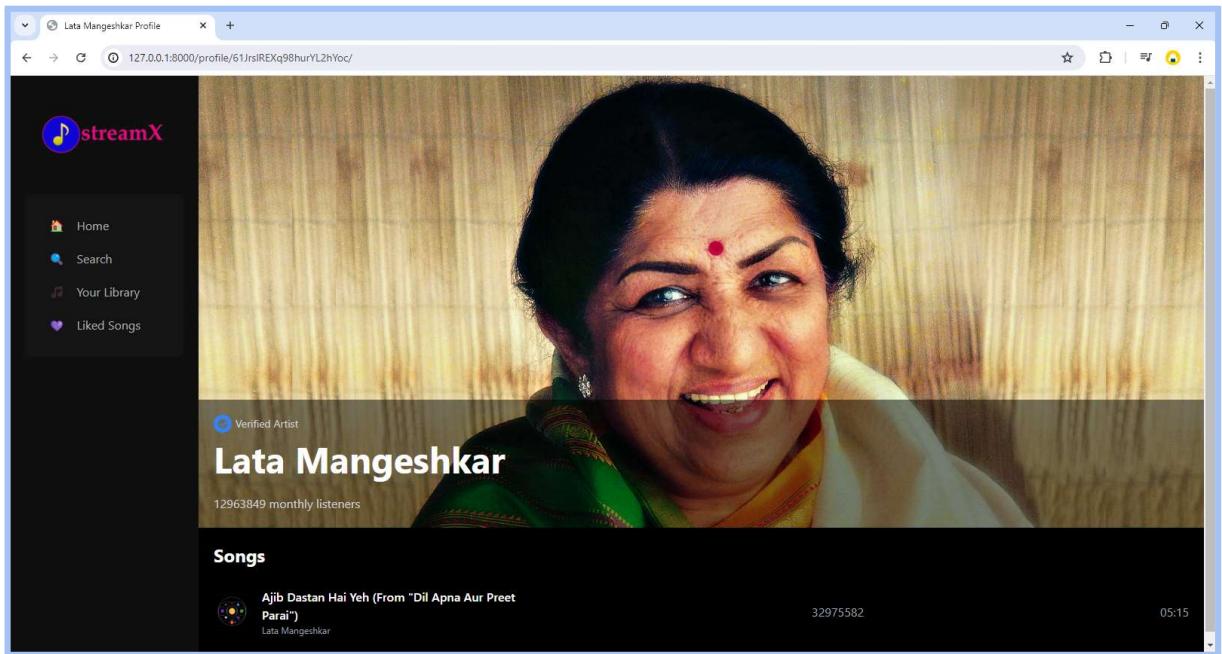
See more podcasts...

• LIKED SONGS

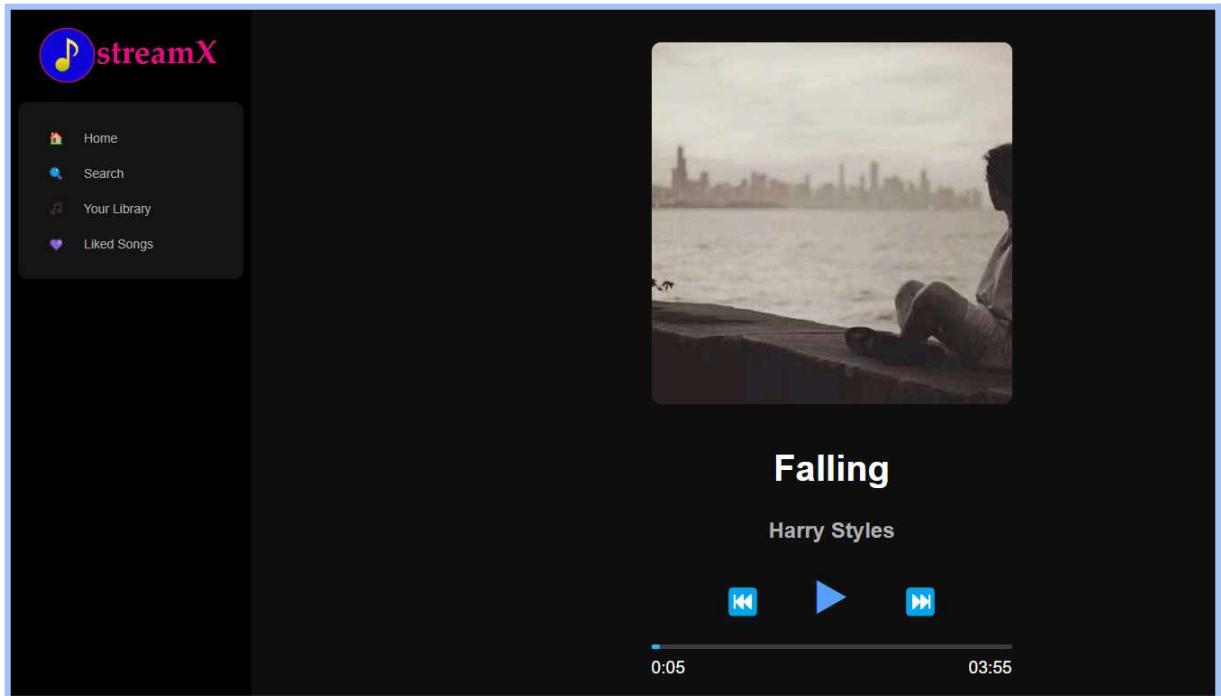


Liked Songs			
Nahin Milta	Bayaan	2019	4:45
Farda	Bayaan	2019	4:29
Naam Goom Jayega	Lata Mangeshkar and Bhupinder Singh	1977	5:40
Luka Chuppi	Lata Mangeshkar and A.R. Rahman	2006	4:12
Aa chal ke tujhe	Kishore Kumar	1964	5:09
Jaat Kahan Ho	Kesarbai Kerkar	1944	3:39
Main Agar Kahoon	Sonu Nigam and Shreya Ghoshal	2007	5:29
Tumhi Dekho Na	Sonu Nigam and Alka Yagnik	2006	4:56
Gali Mein Aaj Nikla	Alka Yagnik	1998	3:30

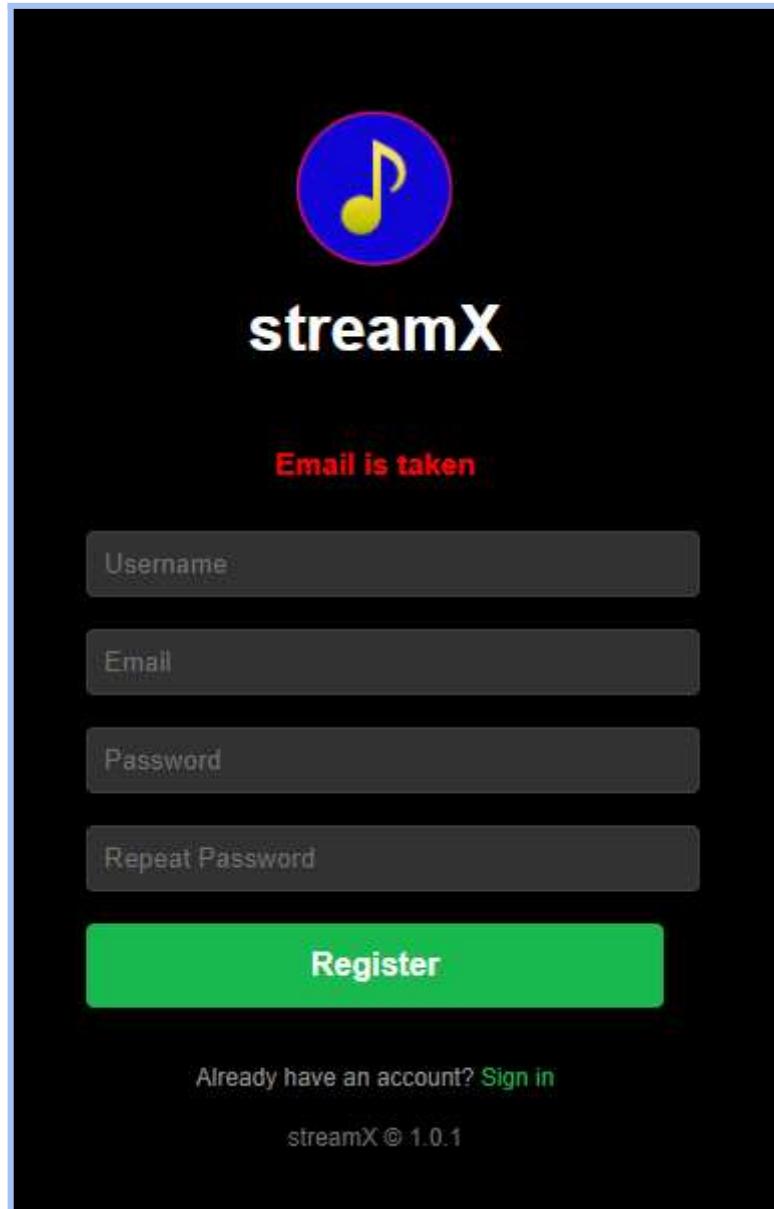
• ARTIST PROFILE



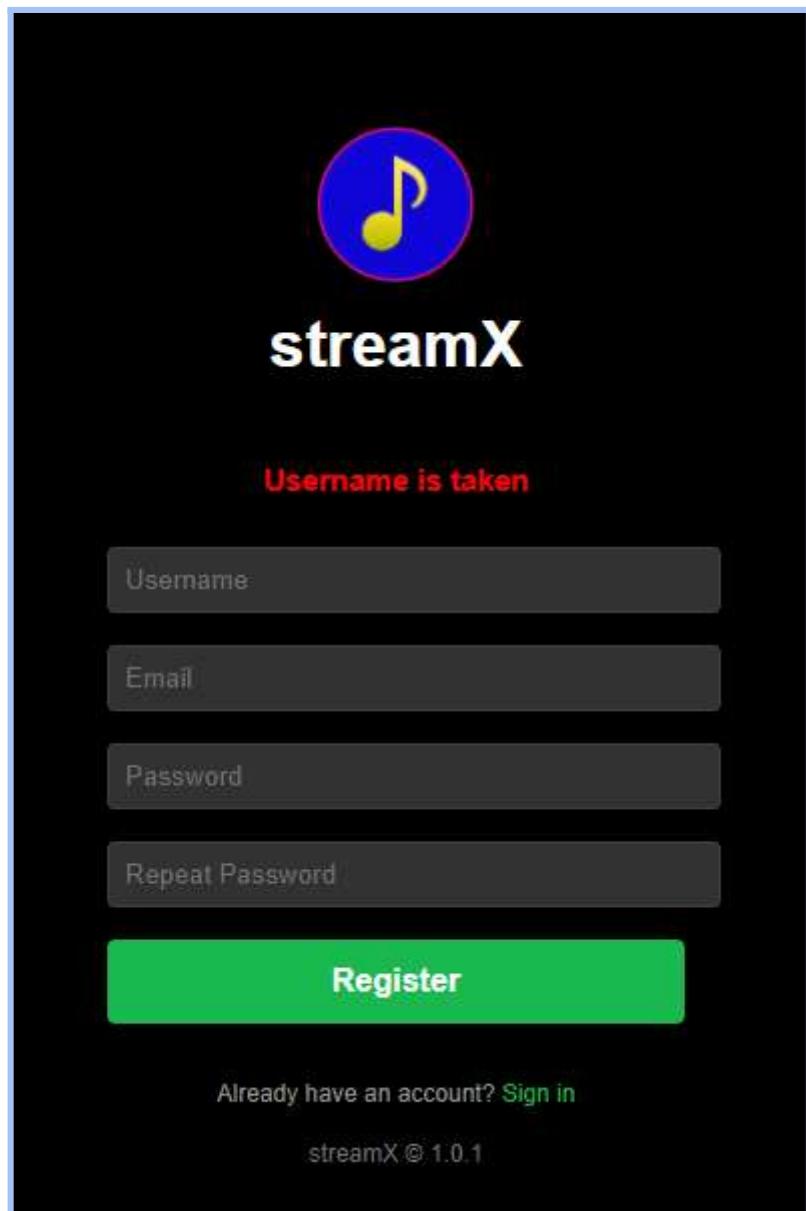
• MUSIC PAGE



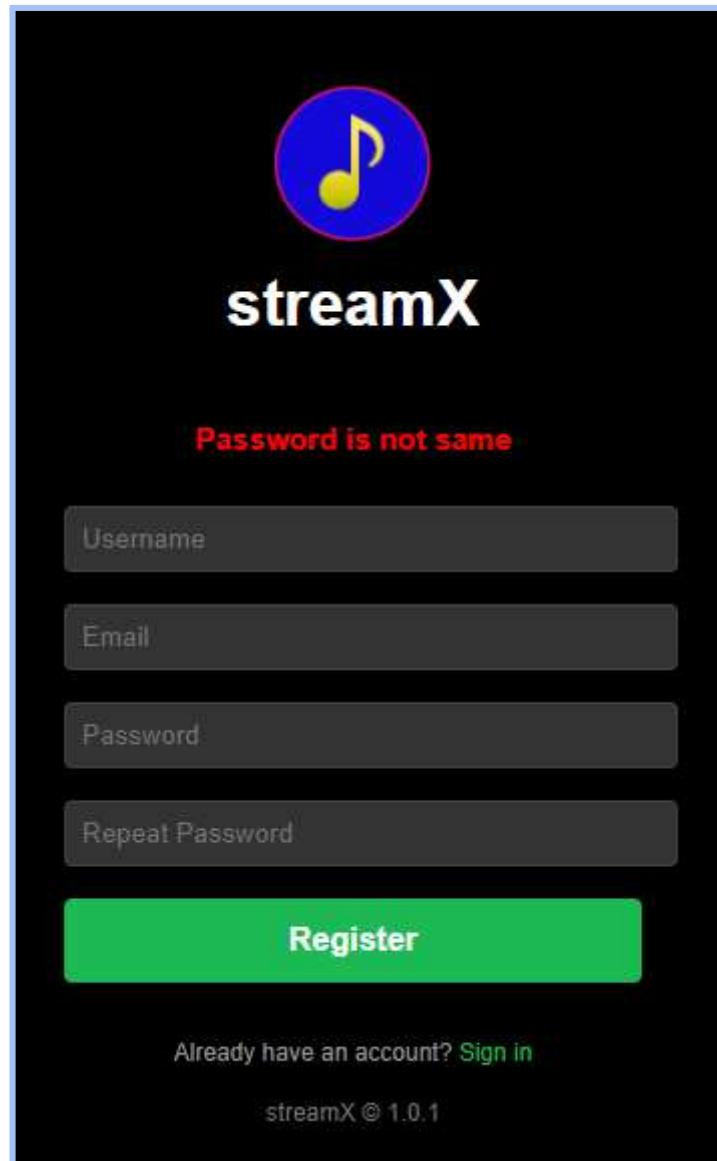
- ERRORS



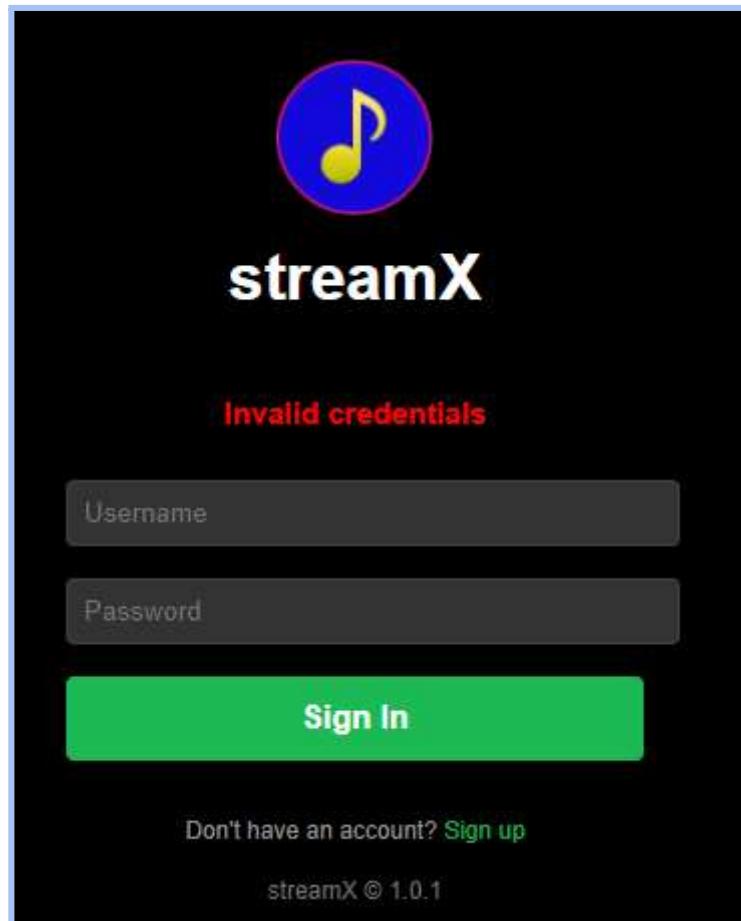
Email used to create account has already been used



Username used to create account has already been used



Password used to create account is not the same in password and repeat password fields



Either of username or password is wrong and different than the ones used to create account

Test Case

- Click on link, takes to login page
 - If login successful, takes to home page
 - If login credentials wrong, gives an error message
 - If there is no account, go to sign up page
- Create a new account
 - If password and repeat password not same, error message is displayed
 - If same, it does a check to see if the username and email is already registered
 - If email is already used then error message is displayed
 - If username is already used then error message is displayed
 - If not used for other account, account successfully created and redirected to home page
 - If an account exists, go to login page
- Home page
 - User is welcomed using username
 - Artist profiles are displayed on home page
 - Songs can be searched in the search bar
- Sidebar connects all the pages by redirecting to the specific page
- Library displays all the first few items from users favourite songs, albums, playlists and podcasts
- Liked songs contains all the songs liked by the user
- Artist profile contains the information regarding the artist, such as picture, name, monthly listeners and songs
- Search page can be used to search for songs and the total number of search results is also displayed
- Songplay area plays the songs, user can move forward or go back by 10 seconds or play from a certain part

Result and Discussion

The music streaming app successfully allows users to stream music, view artist profile, visit library and play songs from lists of liked songs. The app's performance is satisfactory, with a smooth user interface and minimal latency.

Conclusion

The music streaming app project demonstrates the effective use of modern technologies to deliver a high-quality user experience. With continuous enhancements and user feedback, the app can achieve significant market penetration and user satisfaction.

Future Work

There are a lot improvements we want to make in the projects as part of future work.

- API to fetch the top artists
- Add a button to create playlists by user and sharing option with friends and family
- Algorithms to get more personalised recommendations.
- Audio input integration to identify the song based on humming by an user
- Deploying the application

We want to do a more extensive research to make these work and make this application different from other existing music streaming applications.

Bibliography

- HTML and CSS:
<https://youtu.be/G3e-cpL7ofc?si=vKIkJ1JqCD-OlwLE>
- Python: <https://youtu.be/kqtD5dpn9C8?si=ufQR8pb8jesVI5Le>
- SQL: https://youtu.be/7S_tz1z_5bA?si=GtJ80meQUN2k3b7q
- Learning API: Postman Student Expert programme
- Django installation:
<https://www.youtube.com/watch?v=weAUmhABjBc>
- Database connection:
<https://www.youtube.com/watch?v=lEonn27cUfo>
- Adding static files:
<https://docs.djangoproject.com/en/5.0/howto/static-files/>