

Online Music Streaming Platform

Under Guidance Of
Mr. Joyjit Guha Biswas
Subject Matter Expert (PHP)

Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)
CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

A
Project Report
Submitted In Partial Fulfillment of The
Requirements for The Award Of the
Bachelor of Technology
Project Carried Out At



Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)
CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

Submitted By

**Sangram Banerjee
Sukanya Das
Priyan Mukherjee
Reetom Chakraborty**

**Department
Of
Computer Science and Engineering
(Artificial Intelligence and Machine Learning)**

**Future Institute of Technology,
240 Garia Boral Main Road
Kolkata–700154 | West Bengal | India**

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information of approval in any respect will be summarily rejected.)

1. Name of the Student with Group: 1: Sangram Banerjee
 2: Sukanya Das
 3: Priyan Mukherjee
 4: Reetom Chakraborty
2. Title of the Project : **Music Streaming Platform**
3. Name and Address of the Guide : **Mr. Joyjit Guha Biswas**
Sr. Subject Matter Expert & Technical Head (PHP) Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064
4. Educational Qualification of the Guide: Ph.d* M.tech* B.E*/B.Tech * MCA* M.Sc*
5. Working and Teaching experience of the Guide: Years
6. Software used in the Project:
a. Visual Studio Code Platform
b. Python Django Server
c. DB Browser (SQLite)

1.
2.
3.
4.

Signature of the Student

Date:

For Office Use Only

Approved

Not Approved

Signature of the Guide

Date:

Name: Mr. Joyjit Guha Biswas

Subject Matter Expert.

Signature, Designation, Stamp of the Project
Proposal Evaluator

Project Responsibility Form

ONLINE MUSIC STREAMING PLATFORM

SERIAL NUMBER	NAME OF MEMBER	RESPONSIBILITY
1	SANGRAM BANERJEE	PROJECT CO-ORDINATOR & MAIN WEB FRAME BACKEND DEVELOPMENT
2	SUKANYA DAS	DEVELOPER & PROGRAMMER
3	PRIYAN MUKHERJEE	UI DESIGNER, DOCUMENTATION, POWERPOINT-PRESENTATION
4	REETOM CHAKRABORTY	WEBPAGE DESIGN & CONTENT MANAGER

Self-Certificate

This is to certify that the dissertation/project proposal entitled "**Online Music Streaming Platform**" is done by us, is an Authentic work carried out for the partial fulfillment of the requirements for the award of the certificate of **Bachelor of Technology** under the guidance of **Mr. Joyjit Guha Biswas**. The matter embodied in this project work has not been submitted earlier for award of any certificate to the best of our knowledge and belief.

Name of the Student:

- Sangram Banerjee**
- Sukanya Das**
- Priyan Mukherjee**
- Reetom Chakraborty**

Signature of the Student:

- a.
- b.
- c.
- d.

Certificate by Guide

This is to certify that this project entitled "**Online Music Streaming Platform**" submitted in partial fulfillment of the certificate of Bachelor of Computer Science and Engineering (Artificial Intelligence & Machine Learning) through **Ardent Computech Pvt Ltd**, done by the Group Members

- Sangram Banerjee**
- Sukanya Das**
- Priyan Mukherjee**
- Reetom Chakraborty**

is an authentic work carried out under my guidance & best of our knowledge and belief.

- a.
- b.
- c.
- d.

Signature of the students

Date:

Signature of the Guide

Date:

Certificate of Approval

This is to certify that this proposal of Minor project, entitled "**Online Music Streaming Platform**" is a record of bona-fide work, carried out by:

1. Sangram Banerjee
2. Sukanya Das
3. Priyan Mukherjee
4. Reetom Chakraborty

under my supervision and guidance through the Ardent ComputechPvt Ltd. In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the **Future Institute of Technology** as per regulations of the *Ardent®*. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for Bachelor of Technology.

Guide/Supervisor

Mr. Joyjit Guha Biswas
Subject Matter Expert & Technical
Head (Php) Ardent Computech Pvt Ltd
(An ISO 9001:2008 Certified) CF-137, Sector
- 1, Salt Lake City, Kolkata - 700 064

External Examiner(s)

Head of the Department
Department of Computer Science Engineering
(Artificial Intelligence & Machine Learning)
Boral, W.B (Affiliated to MAKAUT, WB)



MSP

TABLE OF CONTENTS

<u>S.NO.</u>	<u>Name of the Topic</u>	<u>Page No.</u>
1.	Company Profile	10
2	Introduction	12-14
2.a	Objective	13
2.b	Scope	14
3	System Analysis	15-26
3.a	Identification of Need	16
3.b	Feasibility Study	17
3.c	Work flow	18-21
3.d	Study of the System	21
3.e	Input & Output	22
3.f	Software Requirements Specification (SRS)	23-24
3.g	Software Engineering Paradigm Applied	25-26
4	System Design	27-50
4.a	Data Flow Diagram (DFD)	28-32
4.b	Sequence Diagram	33-35
4.c	Entity-Relationship Diagram	36-39
4.d	Use-case Diagram	40-42
4.e	Modularization Details	43
4.f	Database Design	44-50
5	Output Screen	51-122
5.a	User Interface Design	52
5.b	Snapshots	53-122
6	Implementation & Testing	123-131
6.a	Introduction	123
6.b	Objectives of Testing	124

6.c	Test Cases	125-136
6.d	White Box Testing	137
6.e	Black Box Testing	137
6.f	Output Testing	138
6.g	Goal of Testing	138
6.h	Integration Test Reports	139
7	System Security Measures	140-141
7.a	Database security measures	141
7.b	System security measures	141
7.c	Limitations	141
8	Conclusion	142
9	Future Scope & Further Enhancements	142-143
10	Bibliography/References	143

1. ARDENT COMPUTECH PVT.LTD.

Ardent Computech Private Limited is an ISO 9001-2008 certified Software Development Company in India. It has been operating independently since 2003. It was recently merged with ARDENT TECHNOLOGIES.

Ardent Technologies

ARDENT TECHNOLOGIES is a Company successfully providing its services currently in UK, USA, Canada and India. The core line of activity at ARDENT TECHNOLOGIES is to develop customized application software covering the entire responsibility of performing the initial system study, design, development, implementation and training. It also deals with consultancy services and Electronic Security systems. Its primary clientele includes educational institutes, entertainment industries, resorts, theme parks, service industry, telecom operators, media and other business houses working in various capacities.

Ardent Collaborations

ARDENT COLLABORATIONS, the Research Training and Development Department of ARDENT COMPUTECH PVT LTD is a professional training Company offering IT enabled services & industrial trainings for B-Tech, MCA, BCA, MSc and MBA fresher's and experienced developers/programmers in various platforms. Summer Training / Winter Training / Industrial training will be provided for the students of B. TECH, M. TECH, MBA and MCA only. Deserving candidates may be awarded stipends, scholarships and other benefits, depending on their performance and recommendations of the mentors.

Associations

Ardent is an ISO 9001:2008 company.

It is affiliated to National Council of Vocational Training (NCVT), Directorate General of Employment & Training (DGET), Ministry of Labor & Employment, and Government of India.

MSP

(MUSIC STREAMING PLATFORM)

2. INTRODUCTION

In digital age that we live, it has been music streaming platforms which had made a huge change in the way we listen to music, making all the songs by all artists easily accessible and available virtually instantly. Our web project goal is to develop a streaming music service of topnotch quality which both fits in with and caters for the music needs of a modern audiophile. Not only that with user experience, functionality and scalability of the platform in mind, it will give a smooth and immersive music listening experience to users across the globe.

Key Features:

- **Intuitive User Interface:** The platform will possess clean and user-friendly design, where users could go through their preferred songs, albums, as well as playlists within seconds.
- **Extensive Music Library:** By extending a hand to major record labels and independent musicians, our platform will provide people with a large variety of songs, albums, and lists of music which cross a wide range of genres and languages.
- **Seamless Playback Experience:** The audience will be fully involved in the music process since no interruptions will occur with the awesome high-quality audio streaming and smooth transitions from tracks.
- **Artist Promotion and Discovery:** We will open up the way for the new musicians to performance their music and to interact with the worldly audience this that will aid in the development of a dynamic music domain of music creators and devotees.

With a dedicated team of developers, designers, and content creators, we are committed to delivering a state-of-the-art music streaming platform that redefines the way people experience and engage with music online.

2a. OBJECTIVE

Our key goal in this web development project is to develop a platform with the highest quality and extraordinary user's experience which provides the best possible listening for every music fan on the planet.

The objective of the project is to implement the online music website that will allow user to browse, search, listen, and download all the songs that they want. The user interface will have all the necessary songs, language and user settings for betterment of the user. User can create playlist, add songs to it and remove when they need and sorting it on their own. The main aim will implement a whole user-friendly UI for the user.

- Enable unlimited access to the vast music collection where you can find songs, collections and mixes of artists from different genres and languages.
- Promote clear and seamless audio streaming without any lags, with a maximized performance for multiple devices.
- Improve user interaction and appeal through the development of user-friendly navigation, minimalistic design and interactive features.

We want to accomplish our project by addressing the following issues: to fundamentally, change the interactions of people with music via the internet into engagement, involvement and emotional resonance using the music streaming platform, that will be relevant to all kinds of users.

2b. SCOPE

Purpose: The Purpose of this project is to provide an automated Django web application and to explore the capabilities and provide a convenient website to user.

Scope: The scope of the project is as follows:

The main scope is to provide the right platform to music listener out in the digital market. In future if website is successful and user are increasing and system administrator is been done properly. we can develop the app for the user and the betterment of music listener.

SYSTEM ANALYSIS

3a. IDENTIFICATION OF NEED

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system studies the minutest detail and gets analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The System is viewed as a whole and the input to the system are identified. The outputs from the organization are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and Decisional variables, analysis and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem area are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solution is given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

3b. FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work.

Effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A Feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various area that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

3c. WORK FLOW

This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

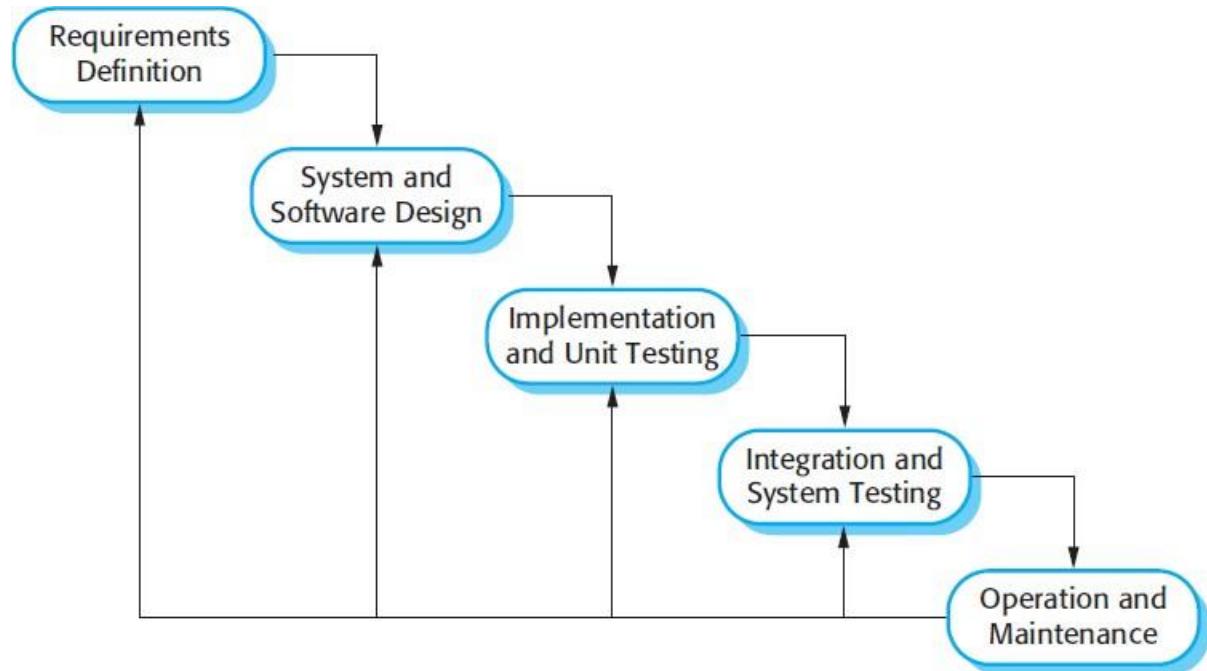
Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

Waterfall Model design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of

each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

3d. STUDY OF THE SYSTEM

Modules:

The modules used in this software are as follows:

- **USER REGISTRATION:** New user can register here to access the webpage.
- **USER LOGIN:** This module is for USER. Logged in user can directly access the music streaming website and can avail the underlying facilities of the webpage.
- **ADMIN REGISTRATION:** Registers new admins to the portal.
- **ADMIN LOGIN:** This module is for ADMIN. Logged in user can directly access the music streaming website and can avail the underlying facilities of the webpage.
- **ADMIN DASHBOARD:** Admin can access to various features such as **Edit User Info, Edit Admin Info, Admin Verification, Playlist Creation, Track Uploading.**

3e. INPUT AND OUTPUT

The main inputs, outputs and the major function the details are:**INPUT**

- User can login using username and password.
- Admin can login using admin-name and password.
- Admin can edit & delete User Information.
- Admin can also edit & delete their Own Admin Information.
- Admin can register other eligible admins through proper verification.
- User can add Playlists.
- Admins can also add Playlists.
- Admin can upload Tracks.
- Admin can add Events to the webpage which will be referred to aforementioned BookMyShow Events page.

OUTPUT

- User can directly access the music streaming website and can avail the underlying facilities of the webpage.
- Admin can view the details of other user, other admins, admins for verification, uploaded tracks & playlists, uploaded events in the website.

3f. SOFTWARE REQUIREMENT SPECIFICATIONS

Software Requirements Specification provides an overview of the entire project. It is a description of a software system to be developed, laying out functional and nonfunctional requirements. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive therequirements, we need to have clear and thorough understanding of the project to be developed. This is prepared after the detailed communication with project team and the customer.

The developer is responsible for: -

- ✓ Developing the system, which meets the SRS and solving all therequirements of the system?
- ✓ Demonstrating the system and installing the system at client's location after acceptance testing is successful.
- ✓ Submitting the required user manual describing the systeminterfaces to work on it and also the documents of the system.
- ✓ Conducting any user training that might be needed for using thesystem.
- ✓ Maintain the system for a period of one year after installation.

HARDWARE REQUIREMENTS:

1. Server:

- Processor: Multi-core processor (e.g., Intel Core i5 or higher) for handling concurrent requests.
- Memory (RAM): At least 4GB RAM, although more RAM is recommended for better performance, especially if dealing with large datasets or high traffic.
- Storage: Sufficient disk space for storing media files, database, and other project files. SSD storage is preferable for faster read/write operations.

2. Network Infrastructure:

- High-speed internet connection for development, deployment, and streaming purposes.
- Adequate network bandwidth to handle concurrent audio streaming requests efficiently.

SOFTWARE REQUIREMENTS:

1. Operating System:

- WINDOWS OS (XP/2000/2003 or 2000 Server/Vista or 7)

2. Python:

- Python 3: Django framework is compatible with Python 3 versions

3. Django & Dependencies:

- Django: Install Django framework using pip, the Python package manager (pip install Django).

4. Development Tools:

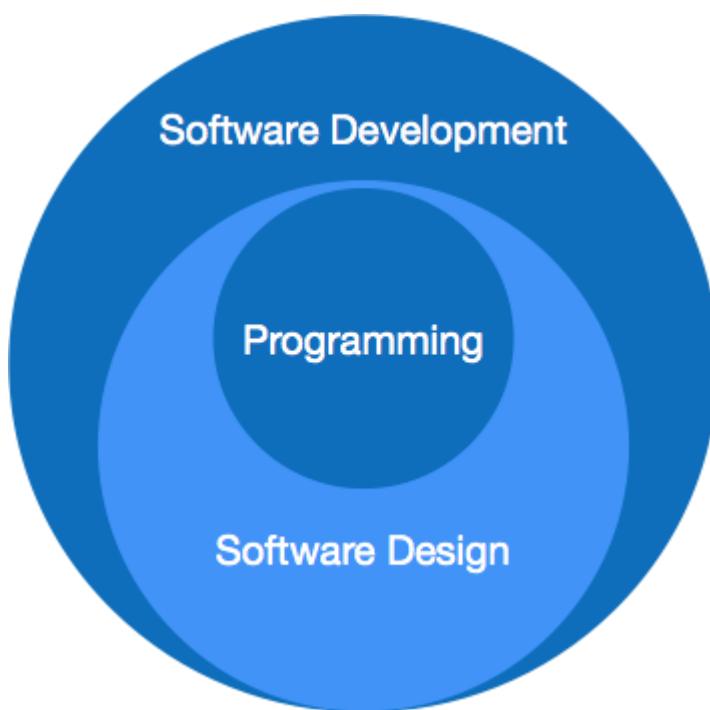
- Integrated Development Environment (IDE): Visual Studio Code Platform

5. Database:

- DB Browser (SQLite): Django supports multiple database backends.

3g. SOFTWARE ENGINEERING PARADIGM APPLIED

Software paradigms refer to the methods and steps, which are taken while designing the software. There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand. These can be combined into various categories, though each of them is contained in one another.



Programming paradigm is a subset of Software design paradigm which is further a subset of Software development paradigm.

There are two levels of reliability. The first is meeting the right requirement. A carefully and through systems study is needed to satisfy this aspect of reliability. The second level of systems reliability involves the actual working delivered to the user. At this level, the systems

reliability is interwoven with software engineering and development. There are three approaches to reliability.

1. Error avoidance: Prevents errors from occurring in software.
2. Error detection and correction: In this approach errors are recognized whenever they are encountered and correcting the error by effect of error of the system does not fail.
3. Error tolerance: In this approach errors are recognized whenever they occur, but enables the system to keep running through degraded performance or applying values that instruct the system to continue process.

Maintenance:

The key to reducing need for maintenance, while working, if possible, to do essential tasks.

1. More accurately defining user requirement during system development.
2. Assembling better systems documents.
3. Using some effective methods for designing, processing, and login and communicating information with project team members.
4. Making better use of existing tools and techniques.

SYSTEM DESIGN

4a. DATA FLOW DIAGRAM

A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A **DFD** is often used as a preliminary step to create an overview of the system, which can later be elaborated.

DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report.

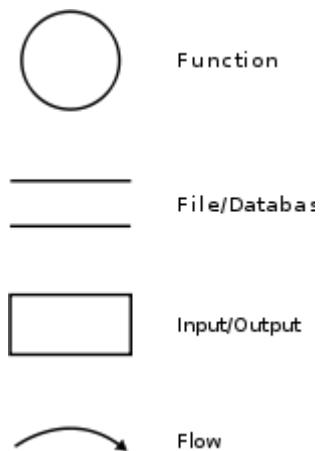
How any system is developed can be determined through a data flow diagram model.

In the course of developing a set of *leveled* data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to identify the transaction data in the data model.

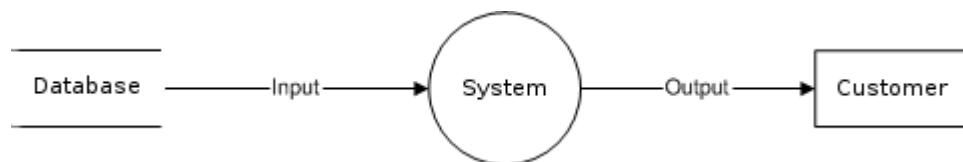
Data flow diagrams can be used in both Analysis and Design phase of the SDLC.

There are different notations to draw data flow diagrams, defining different visual representations for processes, data stores, data flow, and external entities.^[6]

DFD NOTATION



DFD EXAMPLE



Steps to Construct Data Flow Diagram: -

Four Steps are generally used to construct a DFD.

- Process should be named and referred for easy reference. Each name should be representative of the reference.
- The destination of flow is from top to bottom and from left to right.
- When a process is distributed into lower-level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram: -

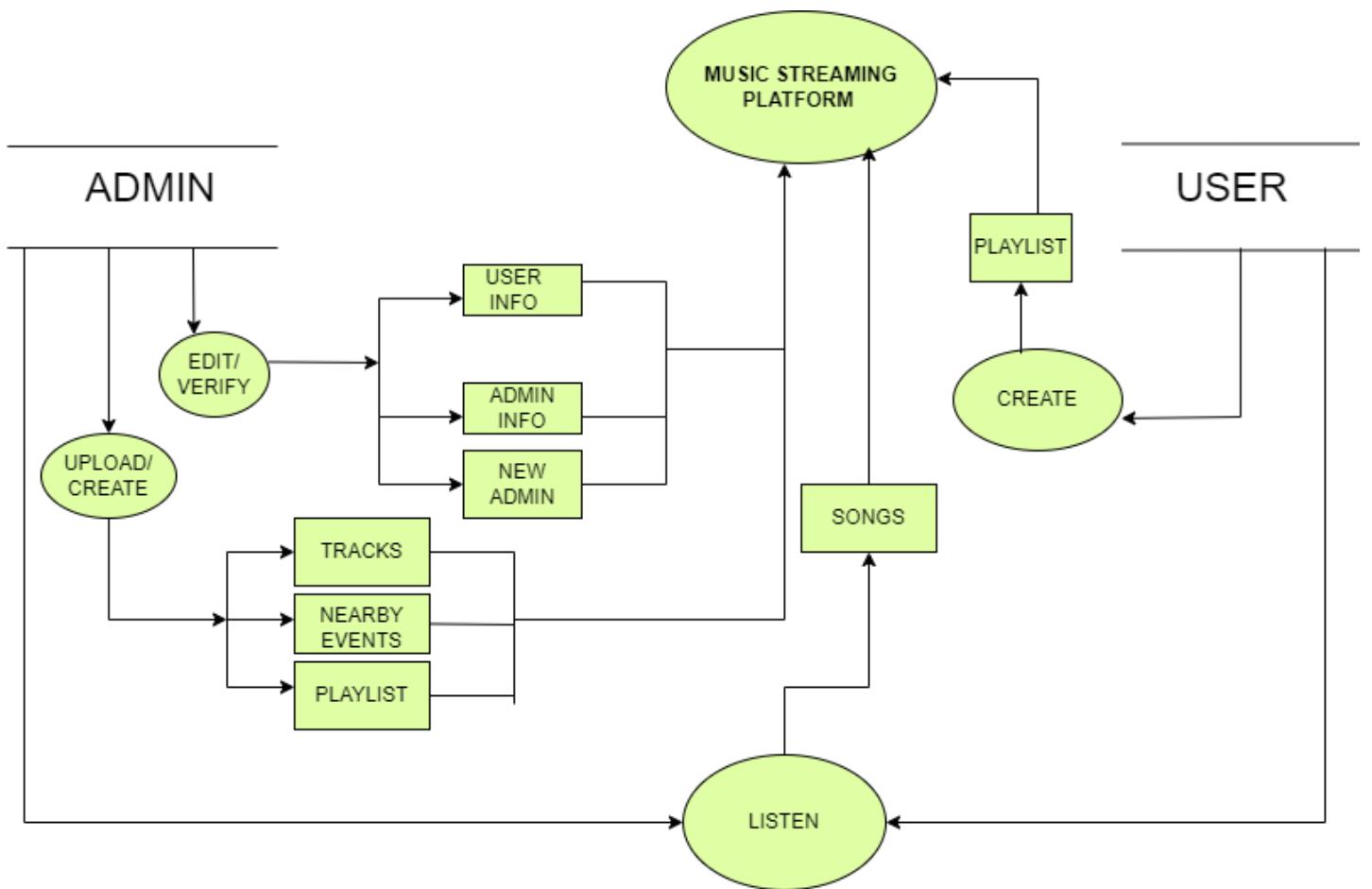
- Arrows should not cross each other.
- Squares, Circles, Files must bear a name.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

DATA FLOW DIAGRAM

LEVEL-0 DFD DIAGRAM:



LEVEL-1 DFD DIAGRAM:



4b. SEQUENCE DIAGRAM

A **Sequence diagram** is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

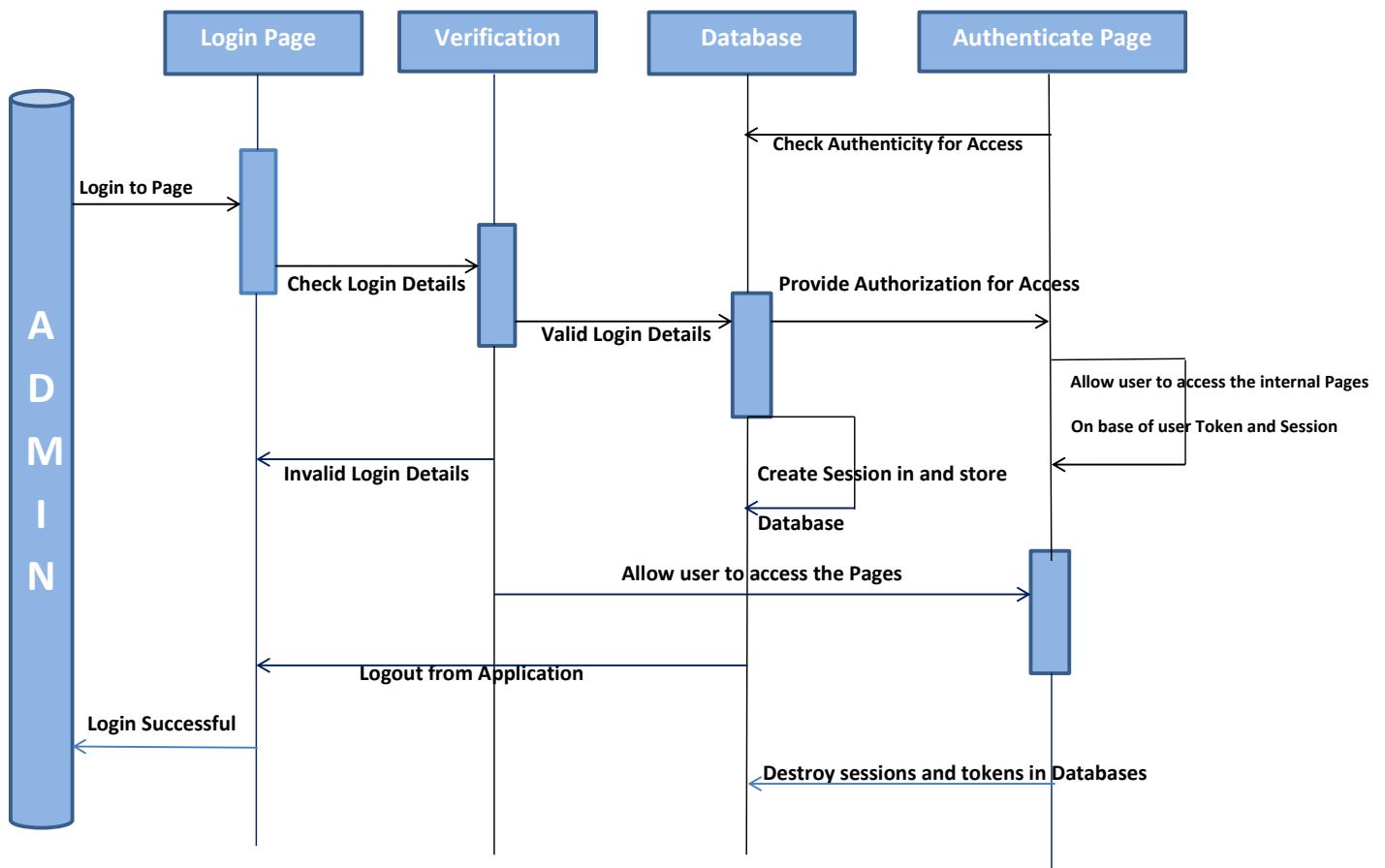
A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.

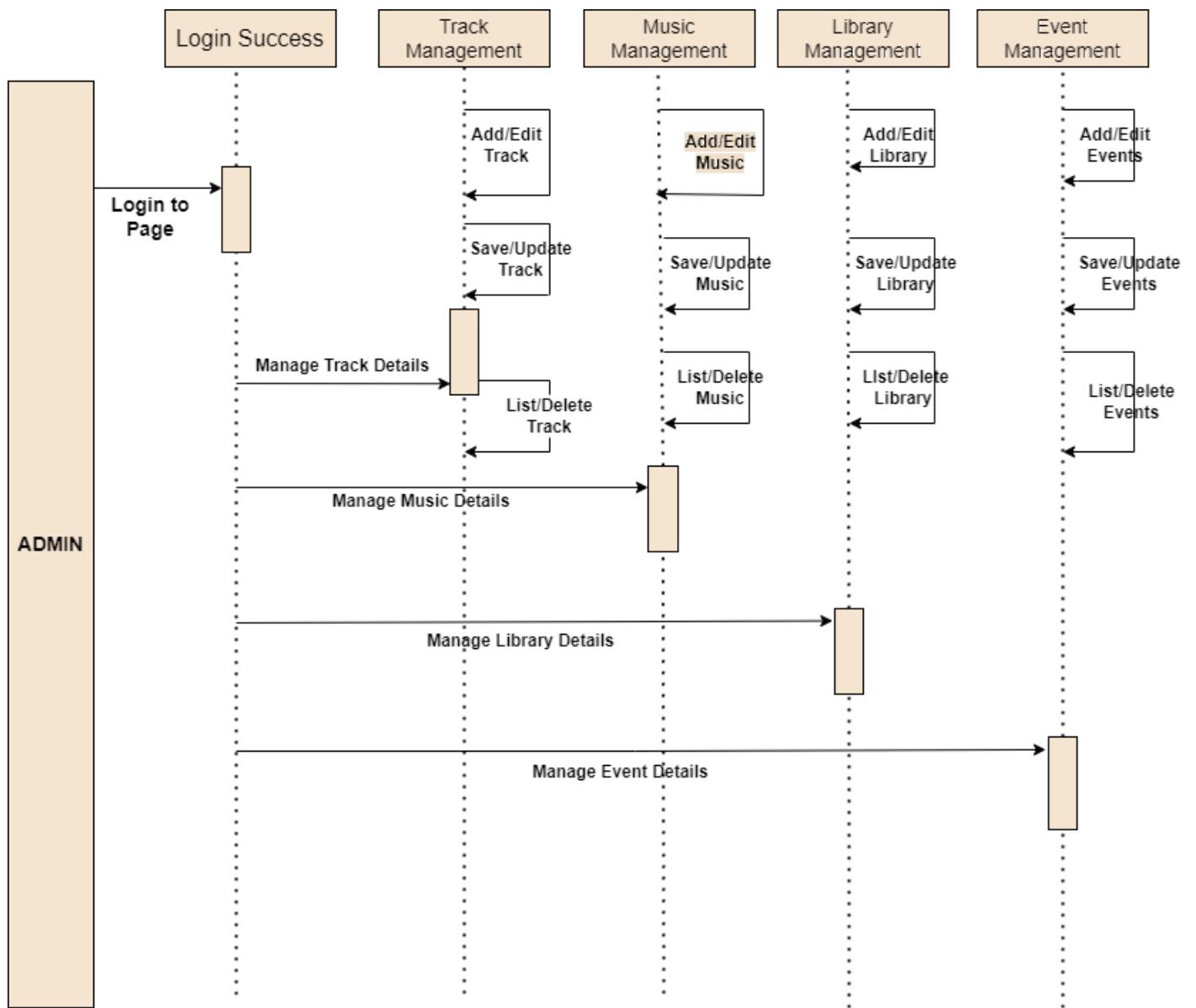
Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

The following nodes and edges are typically drawn in a UML **sequence diagram**: lifeline, execution-specification, message, fragment, interaction, state invariant, continuation, destruction occurrence.

LOGIN SEQUENCE DIAGRAM:



UML SEQUENCE DIAGRAM:



4c. ENTITY RELATIONSHIP DIAGRAM

In software engineering, an **entity–relationship model(ER model)** is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational. The main components of ER models are entities (things) and the relationships that can exist among them.

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data. The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers are the physical implementation of the relationships.

The three schema approach to software engineering uses three levels of ER models that may be developed.

Conceptual data model:

The conceptual ER model normally defines master reference data entities that are commonly used by the organization. Developing an enterprise-wide conceptual ER model is useful to support documenting the data architecture for an organization. A conceptual ER model may be used as the foundation for one or more logical data models. The purpose of the conceptual ER model is then to establish

structural metadata commonality for the master data entities between the set of logical ER models. The conceptual data model may be used to form commonality relationships between ER models as a basis for data model integration.

Logical data model:

The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the relationships between these data entities are established. The logical ER model is however developed independent of technology into which it can be implemented.

Physical data model:

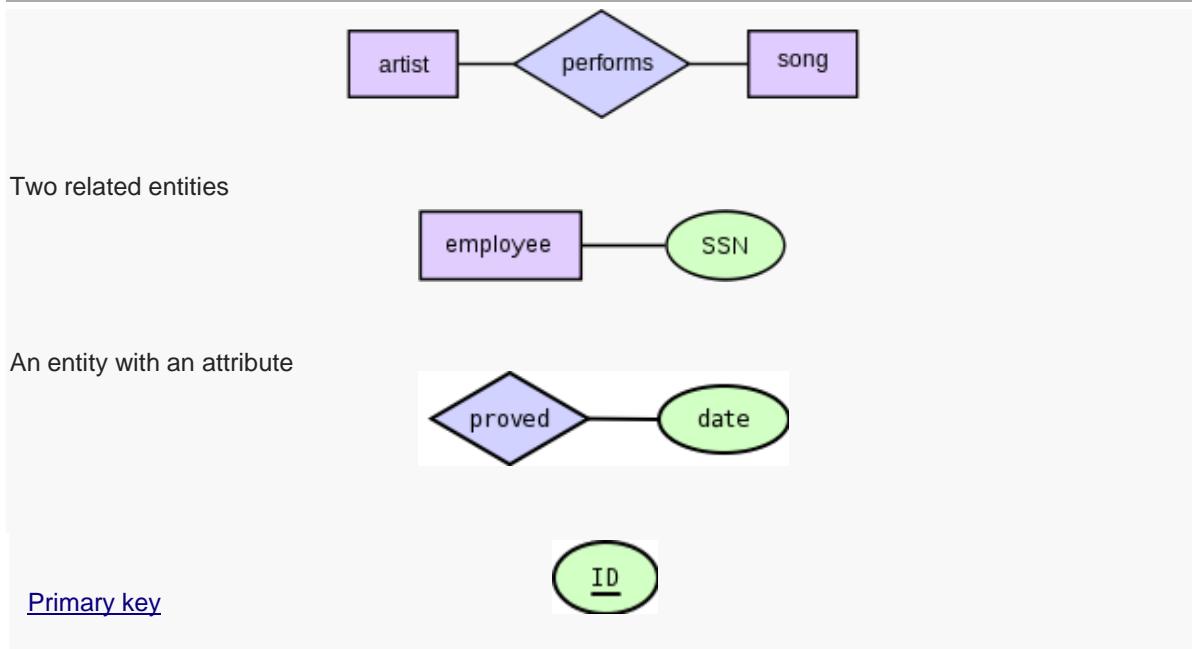
One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database and each physical ER model is technology dependent since each database management system is somewhat different.

The physical model is normally instantiated in the structural metadata of a database management system as relational database objects such as database tables, database indexes such as unique key indexes, and database constraints such as a foreign key constraint or a commonality constraint. The ER model is also normally used to design modifications to the relational database objects and to maintain the structural metadata of the database.

The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational

model; this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design". It is also used in database management system.

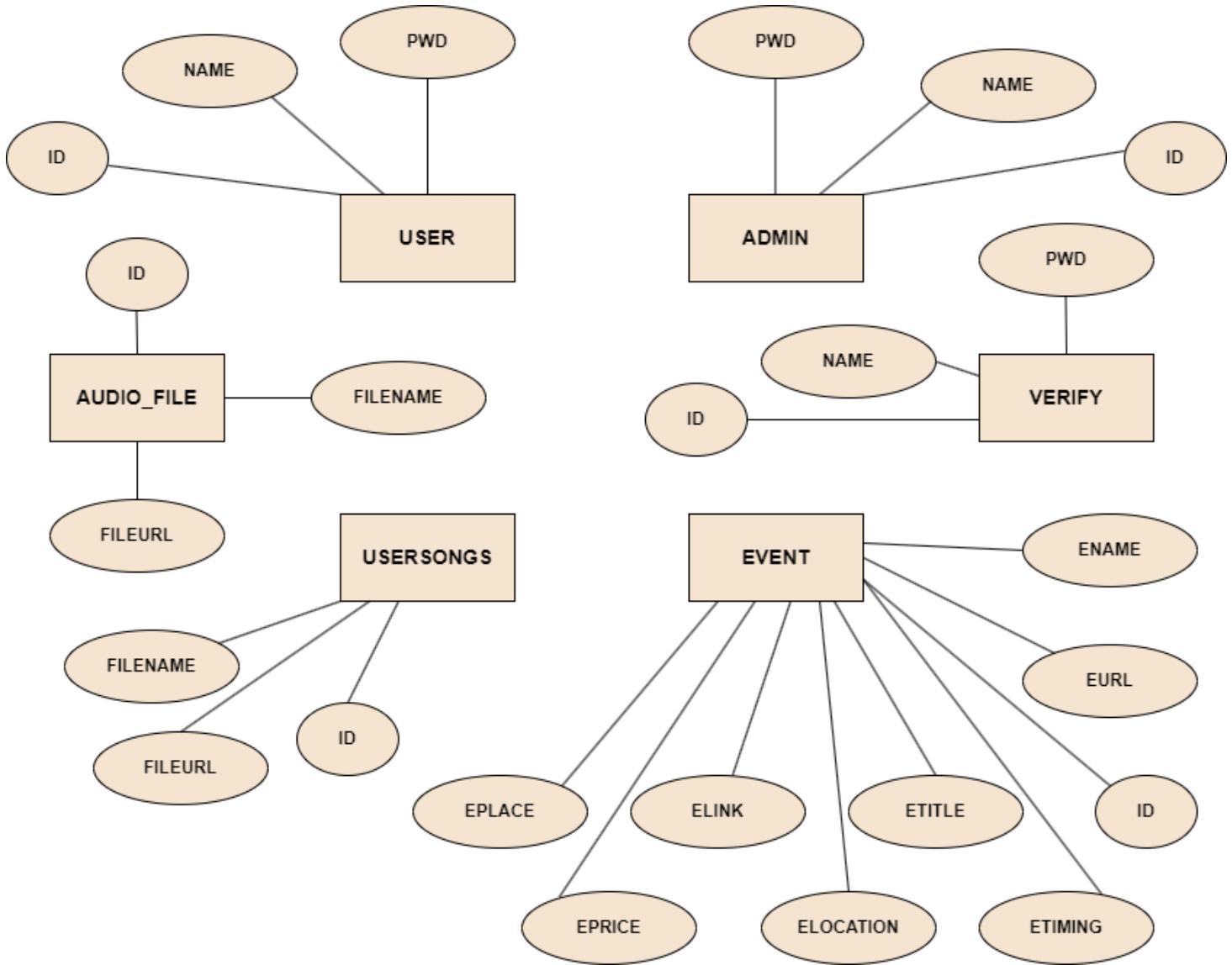
Entity–relationship modeling



Cardinality constraints are expressed as follows:

- A double line indicates a *participation constraint*, totality or subjectivity: all entities in the entity set must participate in *at least one* relationship in the relationship set;
- an arrow from entity set to relationship set indicates a key constraint, i.e. injectivity: each entity of the entity set can participate in *at most one* relationship in the relationship set;
- A thick line indicates both, i.e. bijectivity: each entity in the entity set is involved in *exactly one* relationship.
- An underlined name of an attribute indicates that it is a key: two different entities or relationships with this attribute always have different values for this attribute.

ER-DIAGRAM



4d. USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So, use case diagrams are consists of actors, use cases and theirrelationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So, to model the entire system numbers of use case diagrams are used.

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So, we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So, in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

How to draw Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. So, when the requirements of a system are analyzed, the functionalities are captured in use cases.

So, we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

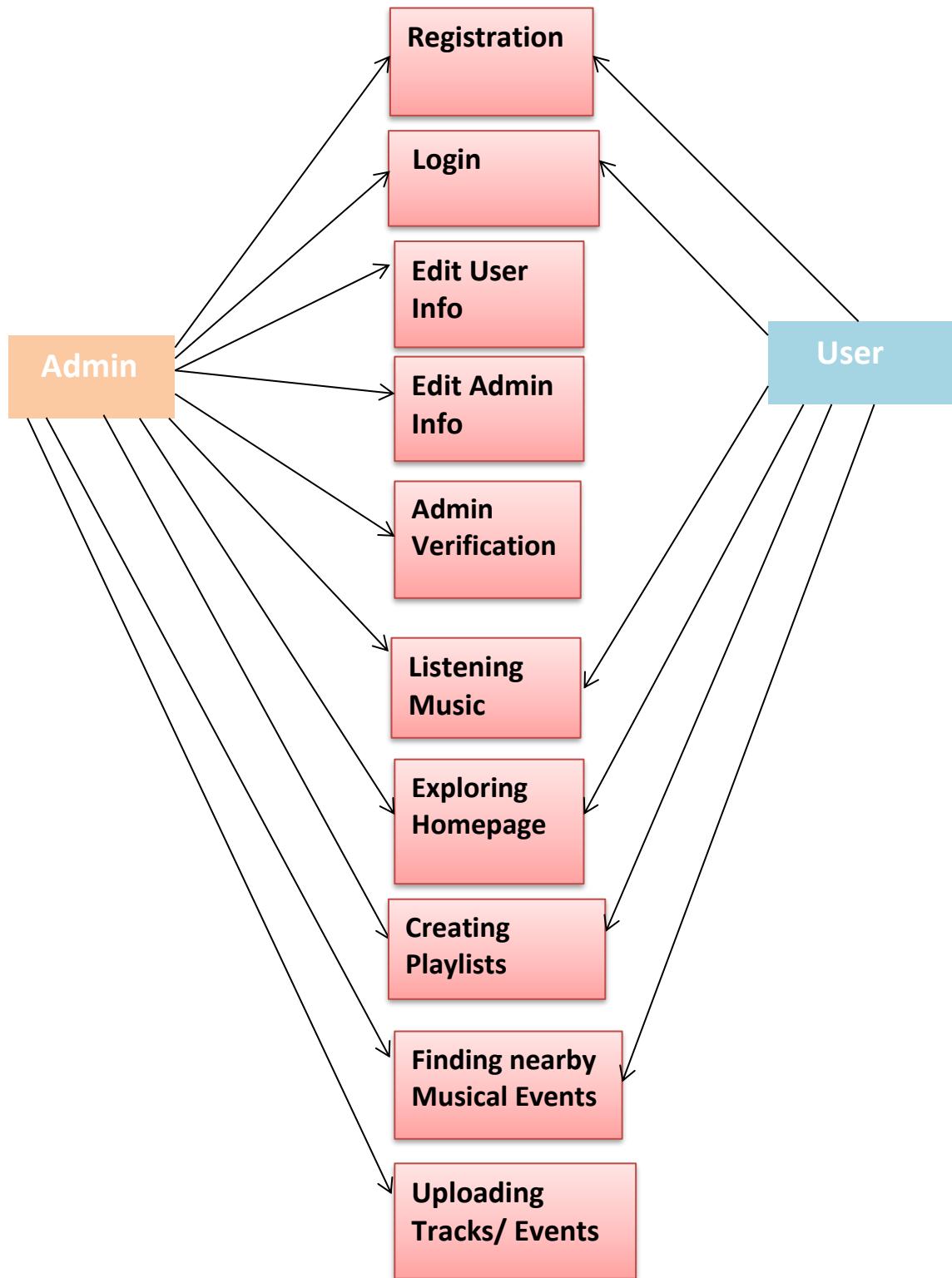
The actors can be human user, some internal applications or may be some external applications. So, in a brief when we are planning to draw a use case diagram, we should have the following items identified.

- Functionalities to be represented as a use case
- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. So, after identifying the above items we have to follow the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. So, the name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships. Because the main purpose of the diagram is to identify requirements.
- Use note whenever required to clarify some important points.

USE CASE DIAGRAM



4.e MODULARIZATION DETAILS

As Modularization has gained increasing focus from companies outside its traditional industries of aircraft and automotive, more and more companies turn to it as strategy and product development tool. I intend to explain the importance aspects of modularization and how it should be initiated within a company. After determining the theoretical steps of modularization success described in literature, I intend to conduct a multiple case study of companies who have implemented modularization in order to find how real-world modularization was initiated and used to improve the company's competitiveness. By combining theory and practical approach to modularization I will derive at convergence and divergence between theoretical implementation to modularization and real-world implementation to modularization. This gives a valuable input for both implantations in companies as well as new aspects to be further.

DATA INTEGRITY AND CONSTRAINTS:

Data integrity is normally enforced in a database system by a series of integrity constraints or rules. Three types of integrity constraints are an inherent part of the relational data model: entity integrity, referential integrity and domain integrity:

1. Entity integrity concerns the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.
2. Concerns the concept of a foreign key. The referential integrity rule states that any foreign-key value can only be in one of two states. The usual state of affairs is that the foreign-key value refers to a primary key value of some table in the database. Occasionally, and this will depend on the rules of the data owner, a foreign-key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown.
3. Domain integrity specifies that all columns in a relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type.

4f. DATABASE DESIGN

A database is an organized mechanism that has capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is two level processes. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called information Level design and it is taken independent of any individual DBMS.

In the following snapshots we display the way we have used SQL Server as the back-end RDBMS for our project and the various entities that have been used along with their table definition and table data.

DATA DICTIONARY

Registered Admin table:

The screenshot shows the 'admin' table in DB Browser for SQLite. The table has three columns: id, name, and pwd. The data is as follows:

	id	name	pwd
1	20	Cristiano Ronaldo	12345
2	21	Luuk De Jong	456789
3	22	Oliver Giroud	789898787
4	23	Allisson Becker	456789878
5	24	Andrew Tate	123456789

Registered User table data:

The screenshot shows the 'user' table in DB Browser for SQLite. The table has three columns: id, name, and pwd. The data is as follows:

	id	name	pwd
1	2	Haaland	123456
2	4	Kvaratskhelia	79865454
3	10	Ameen Al Dakhil	457897854
4	11	Phil Foden	123
5	13	Daniel Munoz	4567897
6	14	Bruno Fernandez	123456789

Verify table:

The screenshot shows the DB Browser for SQLite interface with the database file song.db.sqlite3 open. The 'verify' table is selected, displaying one row of data: id 1, name 13 Sukanya Das, and pwd 0000. The 'Edit Database Cell' panel on the right shows the current value '1'. The 'Remote' panel shows a list of files with the first item being 'Name'.

id	name	pwd
1	13 Sukanya Das	0000

Audio-files table:

The screenshot shows the DB Browser for SQLite interface with the database file song.db.sqlite3 open. The 'audio_files' table is selected, displaying one row of data: id 1, filename backing-track-november-rain-guns-n..., and fileurl upload/backing-track-november-rain.... The 'Edit Database Cell' panel on the right shows the current value '1'. The 'Remote' panel shows a list of files with the first item being 'Name'.

id	filename	fileurl
1	backing-track-november-rain-guns-n...	upload/backing-track-november-rain...

Authentication permission table:

The screenshot shows the DB Browser for SQLite interface with the auth_permission table selected. The table has four columns: id, content_type_id, codename, and name. The data consists of 25 rows, each representing a permission entry. The 'name' column contains descriptive text such as 'Can add log entry', 'Can change log entry', etc.

	id	content_type_id	codename	name
1	1	1	add_logentry	Can add log entry
2	2	1	change_logentry	Can change log entry
3	3	1	delete_logentry	Can delete log entry
4	4	1	view_logentry	Can view log entry
5	5	2	add_permission	Can add permission
6	6	2	change_permission	Can change permission
7	7	2	delete_permission	Can delete permission
8	8	2	view_permission	Can view permission
9	9	3	add_group	Can add group
10	10	3	change_group	Can change group
11	11	3	delete_group	Can delete group
12	12	3	view_group	Can view group
13	13	4	add_user	Can add user
14	14	4	change_user	Can change user
15	15	4	delete_user	Can delete user
16	16	4	view_user	Can view user
17	17	5	add_contenttype	Can add content type
18	18	5	change_contenttype	Can change content type
19	19	5	delete_contenttype	Can delete content type
20	20	5	view_contenttype	Can view content type
21	21	6	add_session	Can add session
22	22	6	change_session	Can change session
23	23	6	delete_session	Can delete session
24	24	6	view_session	Can view session
25	25	7	add_user	Can add user

Authentication User table:

The screenshot shows the DB Browser for SQLite interface with the auth_user table selected. The table has ten columns: id, password, last_login, is_superuser, username, last_name, email, is_staff, is_active, and date_joined. The data shows one row with all fields set to NULL.

	id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
1											

SQLite Sequence table:

The screenshot shows the DB Browser for SQLite interface with the database file db.sqlite3 open. The main window displays the contents of the sqlite_sequence table. The table has two columns: name and seq. The data shows various Django model names and their corresponding sequence values.

name	seq
django_migrations	27
django_admin_log	0
django_content_type	12
auth_permission	48
auth_group	0
auth_user	0
verify	0
user	2
admin	1
audio_files	2
event	3

User songs table:

The screenshot shows the DB Browser for SQLite interface with the database file db.sqlite3 open. The main window displays the contents of the usersongs table. The table has three columns: id, filename, and fileurl. The data shows three entries. An edit dialog is open over the first row, showing the value '1' in the cell. The dialog also displays information about the data type (Text / Numeric) and character count (1 character(s)).

id	filename	fileurl
1	backing-track-november-rain-guns-n...	upload/backing-track-november-rain...
2	backing-track-november-rain-guns-n...	upload/backing-track-november-rain...
3	WhatsApp Audio 2024-01-30 at ...	upload/WhatsApp Audio 2024-01-30 ...

Events table:

The screenshot shows the DB Browser for SQLite interface with the 'event' table selected. The table has columns: id, ename, eurl, etitle, elink, eplace, eprice, elocation, and etiming. There are two rows of data:

	id	ename	eurl	etitle	elink	eplace	eprice	elocation	etiming
1	1	sanam.jpg	eventupload/sanam.jpg	American Pride Soda Music Festival - ...	https://in.bookmyshow.com/events/...	Kolkata, WB 799 onwards	Nicco Park	Sat, 30 Mar 2024, 4:00 PM IST	
2	2	musicshow.png	eventupload/musicshow.png	TRIBUTE TO BRYAN ADAMS AND B...	https://in.bookmyshow.com/events/...	Kolkata, WB 499 onwards	Hard Rock Cafe	Wed, 27 Mar 2024, 9:00 PM IST	

Django Migrations:

The screenshot shows the DB Browser for SQLite interface with the 'django_migrations' table selected. The table has columns: id, app, name, and applied. There are 25 rows of data:

	id	app	name	applied
1	1	contenttypes	0001_initial	2024-03-25 18:49:05.163314
2	2	auth	0001_initial	2024-03-25 18:49:05.222327
3	3	admin	0001_initial	2024-03-25 18:49:05.263105
4	4	admin	0002_logentry_remove_auto_add	2024-03-25 18:49:05.306053
5	5	admin	0003_logentry_add_action_flag_choc...	2024-03-25 18:49:05.321135
6	6	contenttypes	0002_remove_content_type_name	2024-03-25 18:49:05.383678
7	7	auth	0002_alter_permission_name_max_l...	2024-03-25 18:49:05.415387
8	8	auth	0003_alter_user_email_max_length	2024-03-25 18:49:05.447116
9	9	auth	0004_alter_user_username_opts	2024-03-25 18:49:05.472886
10	10	auth	0005_alter_user_last_login_null	2024-03-25 18:49:05.506376
11	11	auth	0006_require_contenttypes_0002	2024-03-25 18:49:05.514724
12	12	auth	0007_alter_validators_add_error_me...	2024-03-25 18:49:05.545996
13	13	auth	0008_alter_user_username_max_len...	2024-03-25 18:49:05.580610
14	14	auth	0009_alter_user_last_name_max_len...	2024-03-25 18:49:05.619469
15	15	auth	0010_alter_group_name_max_length	2024-03-25 18:49:05.654140
16	16	auth	0011_update_proxy_permissions	2024-03-25 18:49:05.674824
17	17	auth	0012_alter_user_first_name_max_le...	2024-03-25 18:49:05.713933
18	18	play	0001_initial	2024-03-25 18:49:05.730227
19	19	play	0002_alter_user_table	2024-03-25 18:49:05.745352
20	20	play	0003_admin	2024-03-25 18:49:05.759225
21	21	play	0004_verify	2024-03-25 18:49:05.771963
22	22	play	0005_verify_pwd	2024-03-25 18:49:05.791926
23	23	play	0006_audiofile_kksongs_rockfile_user...	2024-03-25 18:49:05.807981
24	24	play	0007_event_delete_kksongs_delete...	2024-03-25 18:49:05.832366
25	25	sessions	0001_initial	2024-03-25 18:49:05.854854

Django Content Types:

The screenshot shows the DB Browser for SQLite interface with the database 'D:\Django Project\ Django Project Final\ Django Project Final\song\db.sqlite3' open. The 'django_content_type' table is selected, displaying 12 rows of data. The columns are 'id', 'app_label', and 'model'. The data includes entries for admin, auth, contenttypes, sessions, play, verify, audiofile, usersongs, and event.

	id	app_label	model
1	1	admin	logentry
2	2	auth	permission
3	3	auth	group
4	4	auth	user
5	5	contenttypes	contenttype
6	6	sessions	session
7	7	play	user
8	8	play	admin
9	9	play	verify
10	10	play	audiofile
11	11	play	usersongs
12	12	play	event

Django Admin Logs:

The screenshot shows the DB Browser for SQLite interface with the database 'D:\Django Project\ Django Project Final\ Django Project Final\song\db.sqlite3' open. The 'django_admin_log' table is selected, displaying 0 rows of data. The columns are 'id', 'object_id', 'object_repr', 'action_flag', 'change_message', 'content_type_id', 'user_id', and 'action_time'.

	id	object_id	object_repr	action_flag	change_message	content_type_id	user_id	action_time

OUTPUT SCREEN

5a. USER INTERFACE DESIGN

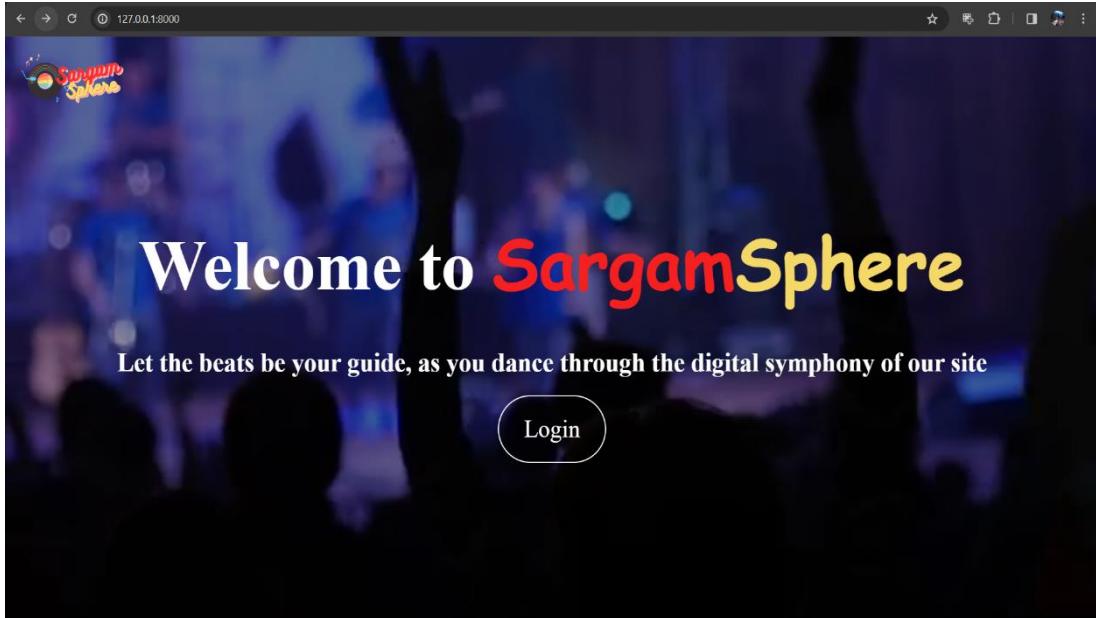
User interface design (UID) or **user interface engineering** is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether that be software design, user research, web design, or industrial design.

SNAPSHOTS

- Welcome page Output:



- Welcome page Source-Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  {% load static %}
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome to Sargamsphere</title>
  <style>
    * {
      padding: 0;
      margin: 0;
      box-sizing: border-box;
      font-family: 'Poppins';
    }
    .logo {
      position: fixed;
      top: 20px; /* Adjust as needed */
      left: 20px; /* Adjust as needed */
      z-index: 1000;
      max-width: 150px; /* Adjust the maximum width of the logo */
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="sargam">
      Welcome to SargamSphere
      Let the beats be your guide, as you dance through the digital symphony of our site
    </div>
    <div class="sphere">
      SargamSphere
    </div>
    <div class="content">
      <button>Login</button>
    </div>
  </div>
</body>
</html>
```

```

        margin-bottom: 50px;
    }

.content h2 {
    font-size: 35px;
    color: #fff;
    margin-bottom: 50px;
}

.content a {
    font-size: 33px;
    color: #fff;
    text-decoration: none;
    border: 2px solid #fff;
    padding: 25px 35px;
    border-radius: 50px;
}

.content a:hover{
    background-color: #f62020;
    color: #000;
    border-color: #f62020;
    transition: 1s;
}

.background-clip{
    position: absolute;
    right: 0;
    bottom: 0;
    z-index:-1;
    transition: 1s;
}

@media (min-aspect-ratio: 16/9){
    .background-clip{
        width: 100%;
        height: auto;
    }
}

@media(max-aspect-ratio: 16/9){
    .background-clip{
        width: auto;
        height: 100%;
    }
}

</style>
</head>
<body>
<div class="logo"></div>
<div class="container">
    <video autoplay loop muted plays-inline
    class="background-clip">
        <source src="{% static '/video/1.mp4' %}"
        type="video/mp4">
    </video>

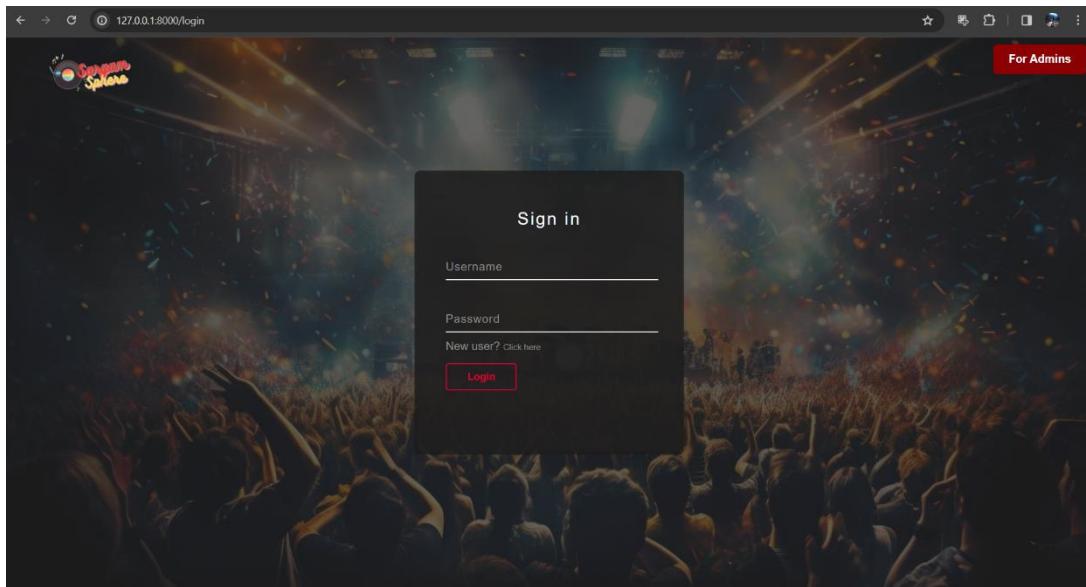
```

```

<div class="content">
    <h1>Welcome to <span
    class="sargam">Sargam</span><span
    class="sphere">Sphere</span></h1>
    <h2>Let the beats be your guide, as you dance
    through the digital symphony of our site</h2>
    <a href="login" class="login-button">Login</a>
</div>
</div>
</body>
</html>

```

- USER login page Output:



- USER login page Source-Code:

```

<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
        href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'
    >
    <title>Login Page</title>
<style>
    * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        font-family: 'Poppins', sans-serif;
    }
    body{
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
    }
    .box form{
        position: absolute;
        inset: 4px;
        background: rgba(34, 34, 34, 0.5);
        padding: 50px 40px;
        border-radius: 8px;
        z-index: 2;
        display: flex;
        flex-direction: column;
    }
    .box form h2{
        color: #fff;
        font-weight: 500;
        text-align: center;
        letter-spacing: 0.1em;
    }
    .box form .inputbox
    {
        position: relative;
        width: 300px;
        margin-top: 25px;
    }
    .box form .inputbox input{
        position: relative;
        width: 100%;
        padding: 20px 10px 10px;
        background: transparent;
        outline: none;
        border: none;
        box-shadow: none;
        color: #23242a;
        font-size: 1em;
    }

```

```

letter-spacing: 0.05em;
transition: 0.5s;
z-index: 10;
}
.box form .inputbox span{
position: absolute;
left: 0;
padding: 20px 0px 10px;
pointer-events: none;
color: #8f8f8f;
font-size: 1em;
letter-spacing: 0.05em;
transition: 0.5s;
}
.box form .inputbox input:valid ~ span,
.box form .inputbox input:focus ~ span
{
color:#fff;
font-size: 0.75em;
transform: translateY(-34px);
}
.box form .inputbox i
{
position: absolute;
left: 0;
bottom: 0;
width: 100%;
height: 2px;
background: #fff;
border-radius: 4px;
overflow: hidden;
transition: 0.5s;
pointer-events: none;
}
.box form .inputbox input:valid ~ i,
.box form .inputbox input:focus ~ i
{
height: 44px;
}
.box form .new-user{
display: flex;
justify-content:space-between;
color: #8f8f8f;
margin-top: 10px;
margin-bottom: 5px;
}
.box form .new-user a{
margin: 10px 0;
font-size: 0.75em;
color: #8f8f8f;
text-decoration: none;
}

```

```

}
.box form .new-user a:hover,
.box form .new-user a:nth-child(2)
{
color:aliceblue ;
}
.box form input[type="submit"] {
outline: none;
padding: 9px 25px;
background: #d80032;
cursor: pointer;
font-size: 0.9em;
border-radius: 4px;
font-weight: 600;
width: 100px;
margin-top: 10px;
transition: background-color 0.5s, color 0.5s; /* Add transition for smooth effect */
border: 2px solid #d80032;
}
.box form input[type="submit"]:hover {
background: transparent; /* Change background to transparent on hover */
color: #d80032; /* Change text color to white on hover */
/*border-color: transparent; */ /* Change border color to transparent on hover */
}
.error-message {
color: #d80032;
margin-top: 20px;
}
.banner {
width: 100%;
height: 100vh;
position: relative;
overflow: hidden;
}
.slider {
width: 100%;
height: 100%;
position: absolute;
top: 0;
left: 0;
display: flex;
align-items: center;
justify-content: center;
z-index: 1; /* Set a lower z-index */
}
.slider img {
width: 100%;

```

```

height: auto;
position: absolute;
opacity: 0;
transition: opacity 2s ease-in-out;
}
.slider img.active {
  opacity: 1;
}
.overlay {
  width: 100%;
  height: 100vh;
  background: linear-gradient(rgba(39, 38, 38, 0.7), rgba(39, 38, 38, 0.7));
  position: absolute;
  top: 0;
}
.navbar {
  width: 92%;
  display: flex;
  align-items: center;
  margin: 20px auto;
}
.logo {
  flex-basis: 15%;
}
.logo img {
  width: 120px;
  cursor: pointer;
}
.box{
  position: absolute; /* Changed position to absolute */
  top: 50%; /* Center vertically */
  left: 50%; /* Center horizontally */
  transform: translate(-50%, -50%); /* Center the box */
  width: 380px;
  height: 400px;
  background: rgba(28, 28, 28, 0.8);
  border-radius: 8px;
  z-index: 2; /* Set a higher z-index */
  padding: 20px; /* Added padding for better appearance */
  overflow: hidden;
}
.admin-button {
  position: fixed;
  top: 10px;
  right: 10px;
  padding: 10px 20px;
  background-color: #8B0000;
  color: #fff;
  border: 2px solid #8B0000;
  border-radius: 5px;
  text-decoration: none;
  cursor: pointer;
  font-family: Arial, sans-serif;
  font-size: 16px;
  font-weight: bold;
  transition: background-color 0.5s, color 0.5s,
border-color 0.5s;
}
.admin-button:hover {
  background-color: transparent;
  border-color: #FF4500;
}
.admin-button span {
  display: inline-block;
  vertical-align: middle;
}
.admin-button:focus,
.admin-button:hover {
  outline: none;
}

```

</style>

</head>

<body>

<div class="banner">

<div class="slider">

</div>

<div class="overlay" style="z-index: 1;">

<div class="navbar" style="z-index: 2;">

<div class="logo">

</div>

</div>

</div>

<div class="box" style="z-index: 3;">

```

<span class="borderline"></span>
<form action="log" method="GET">
    <h2>Sign in</h2>
    {% if error_message %}
        <p class="error-message">{{ error_message }}</p>
    {% endif %}
    <div class="inputbox">
        <input type="text" name="a1" required="required">
        <span>Username</span>
        <i class='bx bx-user'></i>
    </div>
    <div class="inputbox">
        <input type="password" name="a2" required="required">
        <span>Password</span>
        <i class='bx bx-lock-alt'></i>
    </div>
    <div class="new-user">
        <p>New user? <a href="register1">Click here</a></p>
    
```

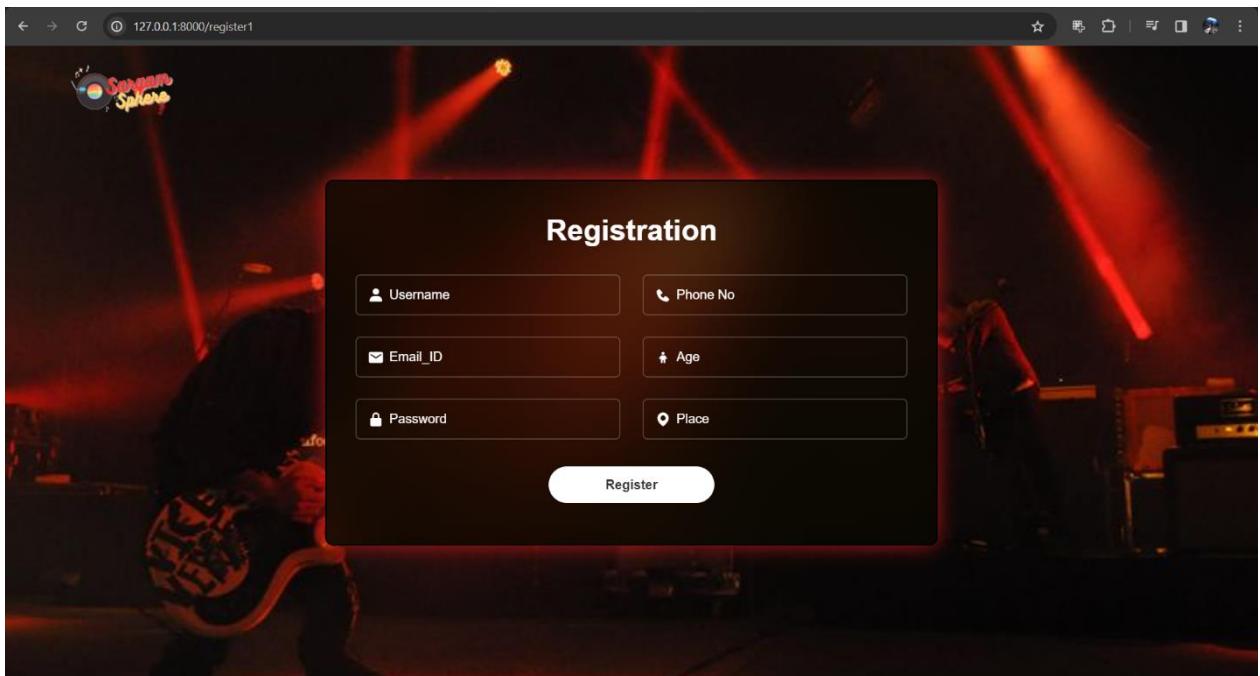
```

</div>
</form>
</div>
<a href="login2" class="admin-button" style="z-index: 2;">For Admins</a>
<script>
    const images =
        document.querySelectorAll('.slider img');
    let currentImageIndex = 0;

    function changeImage() {
        images[currentImageIndex].classList.remove('active');
        currentImageIndex = (currentImageIndex + 1) % images.length;
        images[currentImageIndex].classList.add('active');
    }
    setInterval(changeImage, 4000); //
</script>
</body>
</html>

```

- USER/ADMIN Registration page Output:



• USER/ADMIN Registration page Source-Code:

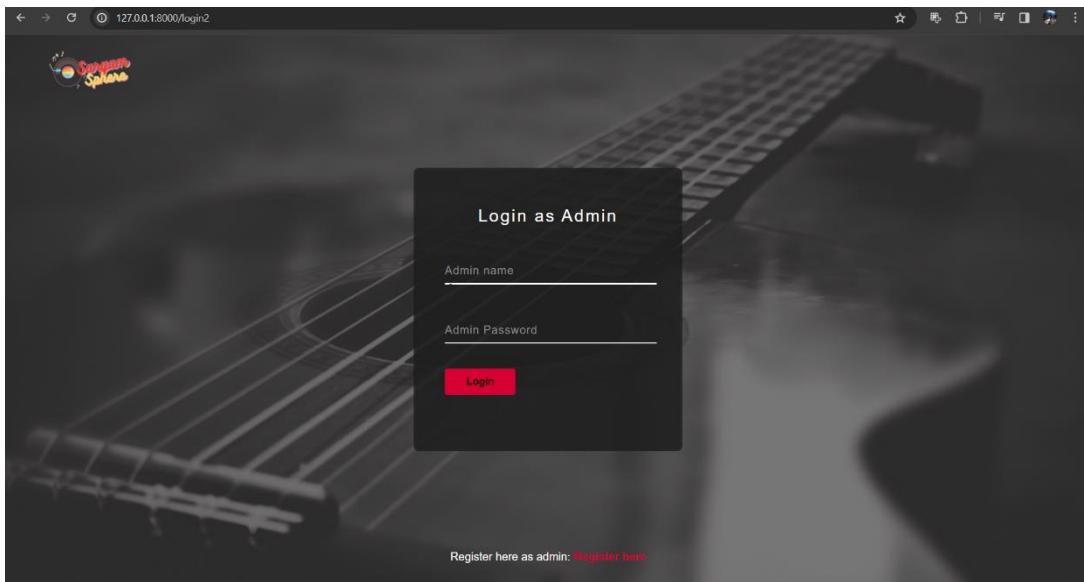
```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css" integrity="sha512-
DTOQO9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQ
eAIfl+Vegovlnnee1c9QX4TctnWMn13TZye+giMm8e2Lw
A==" crossorigin="anonymous" referrerPolicy="no-
referrer" />
    <script
src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></
script>
    <link
href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.c
ss' rel='stylesheet'>
    <link rel="stylesheet" href="{% static 'css/style3.css'
% }">
    <title>User Registration Page</title>
    <title>Admin Registration Page</title> #for Admins
</head>
<style>
    * {
        margin:0;
        padding: 0;
        box-sizing: border-box;
        font-family: "Poppins",sans-serif;
    }
    body{
        display:flex;
        justify-content:center;
        align-items:center;
        min-height:100vh;
        background-size: cover;
        background-position: center;
    }
</style>
<body id="bg" style="background-image: url('{% static
"img/background/bg13.jpg" % }');">
    <img src='{% static "img/logo/logo.png" % }' alt="Logo"
class="logo">
    <div class="wrapper">
        <h1>Registration</h1>
        {% if error_message %}
            <p class="error-message">{{ error_message }}</p>
        {% endif %}
        <form action="rgs1">
            <div class="input-box">
                <div class="input-field">
                    <input type="text" name="a1"
placeholder="Username" required>
                        <i class='bx bxs-user'></i>
                    </div>
                <div class="input-field">
                    <input type="number" placeholder="Phone
No" required>
                        <i class='bx bxs-phone'></i>
                    </div>
                <div class="input-box">
                    <div class="input-field">
                        <input type="email"
placeholder="Email_ID" required>
                            <i class='bx bxs-envelope'></i>
                        </div>
                    <div class="input-field">
                        <input type="number" placeholder="Age"
required>
                            <i class='bx bx-child'></i>
                        </div>
                    <div class="input-box">
                        <div class="input-field">
                            <input type="password" name="a2"
placeholder="Password" required>
                                <i class='bx bxs-lock-alt'></i>
                            </div>
                        <div class="input-field">
                            <input type="text" placeholder="Place"
required>
                                <i class='bx bxs-map'></i>
                            </div>
                        <button type="submit" class="btn">
                            Register
                        </button>
                    </div>
                </div>
            </div>
        </form>
    </div>
</body>
</html>

```

- ADMIN login page Output:



- ADMIN login page Source-Code:

```
<!DOCTYPE html>
{%
  load static %
}
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
  <title>For Admins</title>
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
  }
  body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
  }
  .box form{
    position: absolute;
    inset: 4px;
    background: rgba(34, 34, 34, 0.5);
    padding: 50px 40px;
    border-radius: 8px;
    z-index: 2;
    display: flex;
    flex-direction: column;
  }
  .box form h2{
    color: #fff;
    font-weight: 500;
    text-align: center;
    letter-spacing: 0.1em;
  }
  .box form .inputbox{
    position: relative;
    width: 300px;
    margin-top: 35px;
  }
  .box form .inputbox input{
    position: relative;
    width: 100%;
    padding: 20px 10px 10px;
    background: transparent;
    outline: none;
  }
</style>
```

```

border: none;
box-shadow: none;
color: #23242a;
font-size: 1em;
letter-spacing: 0.05em;
transition: 0.5s;
z-index: 10;
}
.box form .inputbox span{
position: absolute;
left: 0;
padding: 20px 0px 10px;
pointer-events: none;
color: #8f8f8f;
font-size: 1em;
letter-spacing: 0.05em;
transition: 0.5s;    }
.box form .inputbox input:valid ~ span,
.box form .inputbox input:focus ~ span{
color:#fff;
font-size: 0.75em;
transform: translateY(-34px);    }
.box form .inputbox i{
position: absolute;
left: 0;
bottom: 0;
width: 100%;
height: 2px;
background: #fff;
border-radius: 4px;
overflow: hidden;
transition: 0.5s;
pointer-events: none;
}
.box form .inputbox input:valid ~ i,
.box form .inputbox input:focus ~ i{
height: 44px;
}
.box form input[type="submit"] {
outline: none;
padding: 9px 25px;
background: transparent;
color: #d80032;
cursor: pointer;
font-size: 0.9em;
border-radius: 4px;
font-weight: 600;
width: 100px;
margin-top: 35px;
transition: background-color 0.5s, color 0.5s; /*
Add transition for smooth effect */
border: 2px solid #d80032;
}
.box form input[type="submit"]:hover {
background: #d80032; /* Change background to
transparent on hover */
color: black; /* Change text color to white on
hover */
/*border-color: transparent; *//* Change border
color to transparent on hover */
}
.error-message {
color: #d80032;
margin-top: 20px;
}
.register {
position: absolute;
bottom: 20px; /* Adjust positioning as needed */
left: 50%; /* Center horizontally */
transform: translateX(-50%); /* Center horizontally */
text-align: center;
width: 100%;
z-index: 3; /* Ensure it's above other elements */
}
.register p {
font-size: 1em;
color: #fff;
margin: 10px 0;
}
.register a {
color: #d80032;
text-decoration: none;
font-weight: bold;
transition: background-color 0.5s, color 0.5s;;
}
.register a:hover {
color: #fff; /* Change text color on hover */
}
.banner {
width: 100%;
height: 100vh;
position: relative;
overflow: hidden;
}
.slider {
width: 100%;
height: 100%;
position: absolute;
top: 0;
left: 0;
display: flex;
align-items: center;
justify-content: center;
z-index: 1; /* Set a lower z-index */
}
.slider img {
width: 100%;
height: auto;
position: absolute;

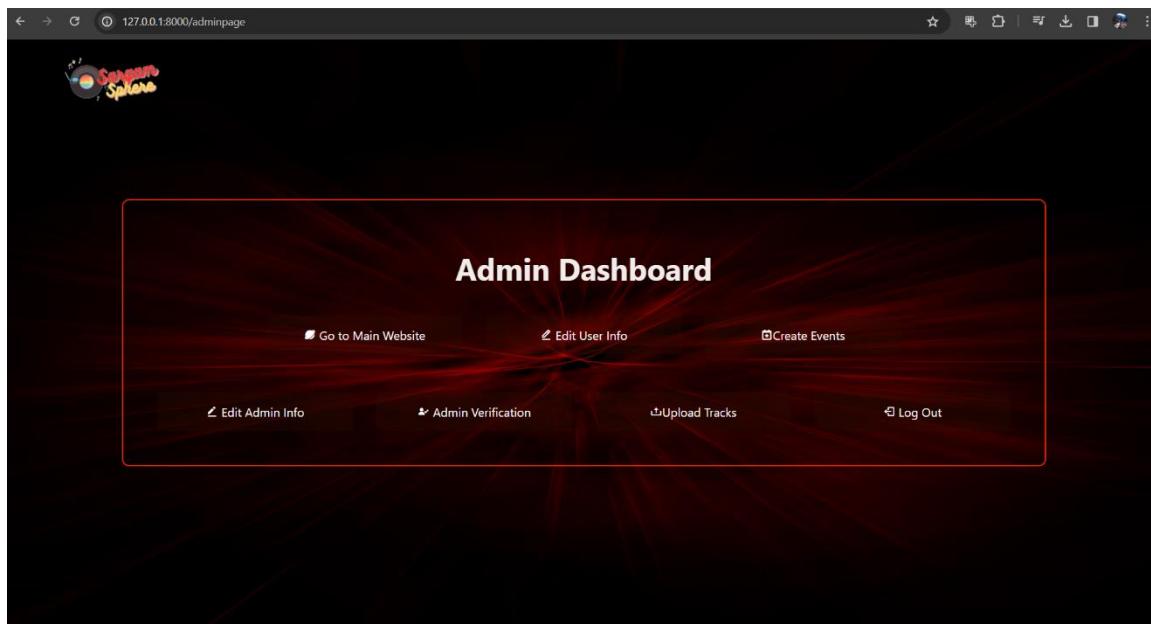
```

```

    opacity: 0;
    transition: opacity 2s ease-in-out;      }
.slider img.active {
    opacity: 1;      }
.overlay {
    width: 100%;
    height: 100vh;
    background: linear-gradient(rgba(61, 60, 60, 0.7),
rgba(61, 60, 60, 0.7));
    position: absolute;
    top: 0;
}
.navbar {
    width: 92%;
    display: flex;
    align-items: center;
    margin: 20px auto;
}
.logo {
    flex-basis: 15%;      }
.logo img {
    width: 120px;
    cursor: pointer;
}
.box{
    position: absolute; /* Changed position to absolute */
    top: 50%; /* Center vertically */
    left: 50%; /* Center horizontally */
    transform: translate(-50%, -50%); /* Center the
box *//
    width: 380px;
    height: 400px;
    background: rgba(28, 28, 28, 0.8);
    border-radius: 8px;
    z-index: 2; /* Set a higher z-index */
    padding: 20px; /* Added padding for better
appearance */
    overflow: hidden;      }
</style>
</head>
<body>
    <div class="banner">
        <div class="slider">
            
                
                
                
                
        </div>
        
    </div>
    <div class="overlay" style="z-index: 1;">
        <div class="navbar" style="z-index: 2;">
            <div class="logo">
                
            </div>
            </div>
        </div>
        <div class="box" style="z-index: 3;">
            <span class="borderline"></span>
            <form action="log2" method="GET">
                <h2>Login as Admin</h2>
                {% if error_message %}>
                    <p class="error-message">{{ error_message }}</p>
                {% endif %}>
                <div class="inputbox">
                    <input type="text" name="a1" required="required">
                    <span>Admin name</span>
                    <i class='bx bx-user'></i>
                </div>
                <div class="inputbox">
                    <input type="password" name="a2" required="required">
                    <span>Admin Password</span>
                    <i class='bx bx-lock-alt'></i>
                </div>
                <input type="submit" value="Login">
            </form>
        </div>
        <div class="register">
            <p>Register here as admin: <a href="register2">Register here</a></p>
        </div>
        <script>
            const images = document.querySelectorAll('.slider img');
            let currentImageIndex = 0;
            function changeImage() {
                images[currentImageIndex].classList.remove('active');
                currentImageIndex = (currentImageIndex + 1) %
                images.length;
                images[currentImageIndex].classList.add('active');
            }
            setInterval(changeImage, 4000);
        </script>
    </div>
</body>
</html>

```

- ADMIN Dashboard Output:



- ADMIN Dashboard Source-Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    {%- load static %}
    <meta charset="UTF-8">
    <meta name="viewport"
    content="width=device-width, initial-
    scale=1.0">
    <link
    href='https://unpkg.com/boxicons@2.1.
    4/css/boxicons.min.css'
    rel='stylesheet'>
    <link rel="shortcut icon" href="{%
    static 'img/logo/favicon.ico' %}">
    <title>Admin Dashboard</title>
    <style>
        body {
            font-family: 'Segoe UI',
            Tahoma, Geneva, Verdana, sans-serif;
            background-image: url('{%
            static 'bg23.jpg' %}');
            background-repeat: no-repeat;
            background-size: cover;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
            color: #fff;
        }
        .button-container {
            text-align: center;
            border: 2px solid #ff2600;
            border-radius: 10px;
            padding: 30px;
            background: transparent;
            box-shadow: 0 0 20px rgba(0,
            0, 0, 0.3);
        }
        .button-group {
            display: flex;
            justify-content: center;
            align-items: center;
            flex-wrap: wrap;
            margin-top: 20px;
        }
        .button {
            padding: 15px 30px;
            background-color: white;
            color: #ff2600;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            text-decoration: none;
            display: inline-block;
            margin: 15px;
            font-size: 16px;
            transition: background-color
            0.3s ease-in-out, color 0.3s ease-in-
            out;
            width: 200px; /* Set a fixed
            width for the buttons */
        }
        .button:hover {
            background-color: #7e21103a;
            color: whitesmoke;
        }
        h2 {
            color: #f0ebedb;
            margin-bottom: 20px;
        }
    </style>

```

```

        font-size: 40px;
    }
    .button {
        background-color:rgba(42, 12,
4, 0.321);
        color: #fff;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        text-decoration: none;
        transition: background-color
0.3s ease-in-out; /* Added transition
effect */
    }
    .button:hover {
        background-color: #722013; /**
Change background color on hover */
    }
    .logo {
        position: absolute;
        top: 20px;
        left: 80px;
        width: 100%;
        max-width: 130px;
    }

```

</style>

</head>

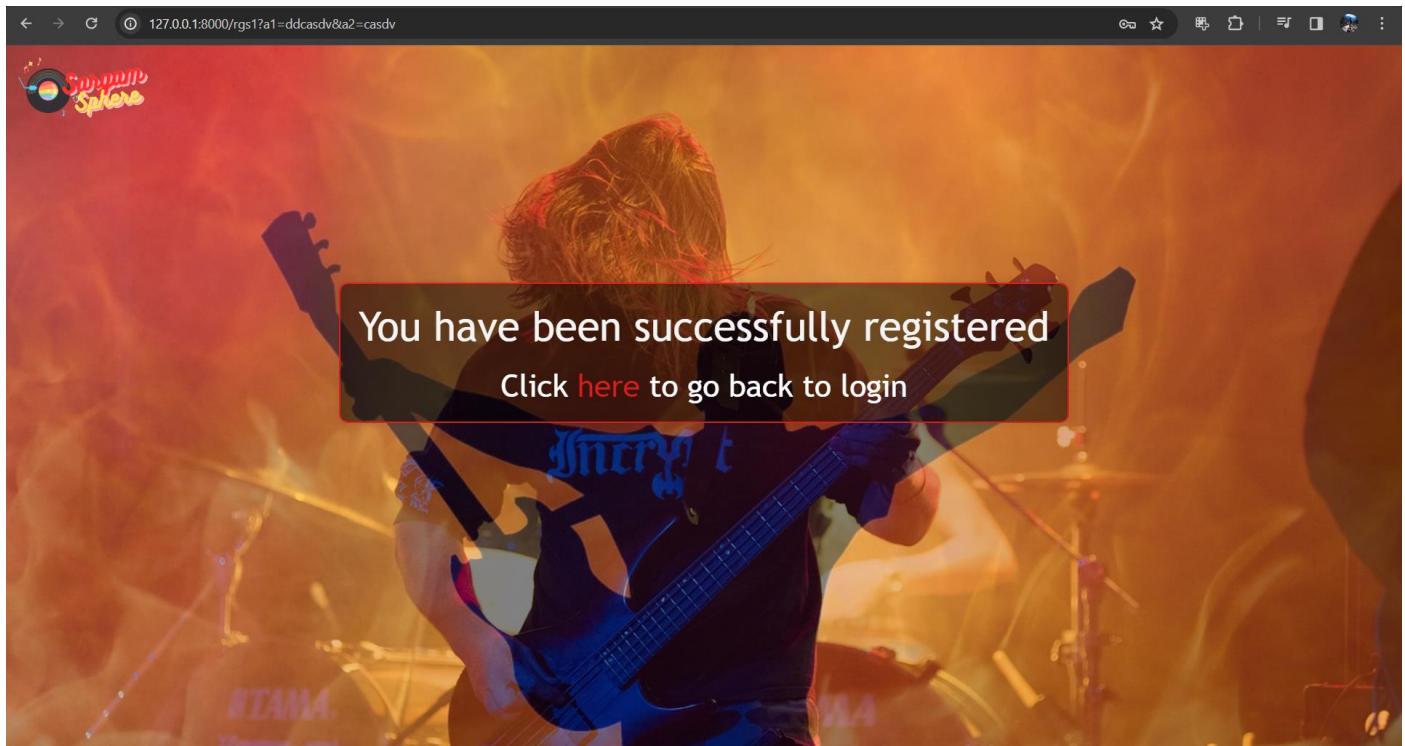
<body>


```

<div class="button-container">
    <h2>Admin Dashboard</h2>
    <div class="button-group">
        <a href="index"
class="button"><i class='bx bxs-
planet'></i> Go to Main Website</a>
        <a href="show"
class="button"><i class='bx bx-edit-
alt' ></i> Edit User Info</a>
        <a href="eventupload"
class="button"><i class='bx bx-
calendar-star'></i>Create Events</a>
    </div>
    <div class="button-group">
        <a href="show2"
class="button"><i class='bx bxs-edit-
alt' ></i> Edit Admin Info</a>
        <a href="show3"
class="button"><i class='bx bxs-user-
check'></i> Admin Verification</a>
        <a href="upld"
class="button"><i class='bx bx-
upload'></i>Upload Tracks</a>
        <a href="/" class="button"><i
class='bx bx-log-out'></i> Log
Out</a>
    </div>
</div>
</body>
</html>

```

- USER Successful Registration page Output:



• USER Successful Registration page Source-Code:

```

<!DOCTYPE html>
<html lang="en">
<head>
{%- load static %}
<meta charset="UTF-8">
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
<title>Registration Success</title>
<style>
    body {
        font-family: 'Trebuchet MS',
'Lucida Sans Unicode', 'Lucida Grande',
'Lucida Sans', Arial, sans-serif;
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        animation: backgroundTransition
15s infinite; /* Adjust the duration as
needed */
    }
    @keyframes backgroundTransition {
        0% {
            background-image: url('{{
static 'rc50.jpg' }}');
            background-repeat: no-repeat;
            background-size: cover;
        }
        33% {
            background-image: url('{{
static 'rc60.jpg' }}');
            background-repeat: no-repeat;
            background-size: cover;
        }
        66% {
            background-image: url('{{
static 'rc70.jpg' }}');
            background-repeat: no-repeat;
            background-size: cover;
        }
        100% {
            background-image: url('{{
static 'rc49.jpg' }}');
            background-repeat: no-repeat;
            background-size: cover;
        }
    }
    .container {
        text-align: center;
        background-color: rgba(0, 0, 0,
0.5); /* Transparent background */
        border: 2px solid #e61f1feb;
        border-radius: 10px;
        padding: 20px;
        box-shadow: 0 0 10px rgba(0, 0, 0,
0.1);
        animation: slide-in 2s ease-in-out
forwards;
        position: relative;
        top: -50px;
    }
    .logo {
        position: absolute;
        top: 10px;
        left: 10px;
        max-width: 150px; /* Adjust the
max-width as needed */
        max-height: 150px; /* Adjust the
max-height as needed */
        z-index: 1; /* Ensure the logo is
above other elements */
    }
    .message {
        font-size: 45px;
        margin-bottom: 20px;
        color: whitesmoke;
    }
    .small-text {
        font-size: 34px;
        color: white;
    }
    a {
        color: #e61f1feb;
        text-decoration: none;
        transition: color 0.3s ease-in-
out;
    }
    a:hover {
        text-decoration: underline;
        color: #1a5276;
    }
    @keyframes slide-in {
        from {
            transform: translateY(-100%);
            opacity: 0;
        }
        to {
            transform: translateY(0);
            opacity: 1;
        }
    }
</style>
</head>
<body>
```

```


<div class="container">
    <div class="message">You have been
successfully registered</div>
    <div class="small-text">Click <a

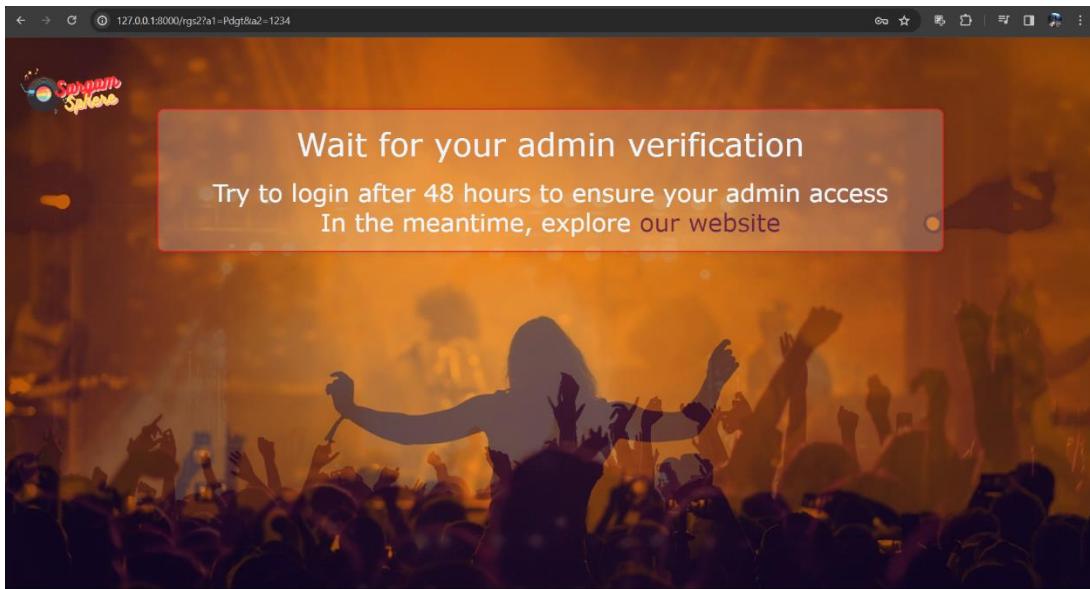
```

```

href='login'>here</a> to go back to
login</div>
</div>
</body>
</html>

```

- ADMIN VERIFICATION page Output:



- ADMIN VERIFICATION page Source-Code:

```

<!DOCTYPE html>
<html lang="en">
<head>
{% load static %}
<meta charset="UTF-8">
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
<title>Admin Verification</title>
<style>
    body {
        font-family: Verdana, Geneva,
        Tahoma, sans-serif;
        text-align: center;
        margin-top: 100px;
        background-size: cover;
        animation: changeBackground 12s
infinite; /* Increased duration to 12s */
        position: relative; /* Added
position relative to establish a stacking
context */
    }
    @keyframes changeBackground {
        0%, 100% {
            background-image: url('{%
static 'rc49.jpg' %}');
            background-repeat: no-repeat;

```

```

background-size: cover;
        }
        33% {
            background-image: url('{%
static 'rc45.jpg' %}');
            background-repeat: no-repeat;
            background-size: cover;
        }
        66% {
            background-image: url('{%
static 'bg28.jpg' %}');
            background-repeat: no-repeat;
            background-size: cover;
        }
    }
    .container {
        border: 2px solid #e61f1feb;
        border-radius: 10px;
        padding: 20px;
        width: 70%;
        margin: 0 auto;
        background-color: rgba(255, 255,
255, 0.2);
        box-shadow: 0 0 10px rgba(0, 0, 0,
0.1);
        overflow: hidden;
    }

```

```

.logo {
    position: absolute;
    top: -60px;
    left: 10px;
    max-width: 150px; /* Adjust the
max-width as needed */
    max-height: 150px; /* Adjust the
max-height as needed */
}
.message,
.small-text {
    opacity: 0;
    animation: slide-in 2s ease-in-out
forwards;
}
.message {
    font-size: 45px;
    margin-bottom: 20px;
    color: aliceblue;
}
.small-text {
    font-size: 34px;
    color: aliceblue;
}
a {
    color: rgba(73, 10, 73, 0.756);
    text-decoration: none;
    transition: color 0.3s ease-in-
out;
}
a:hover {
    text-decoration: underline;
    color: #1a5276;
}
@keyframes slide-in {
    from {
        transform: translateY(-100%);
        opacity: 0;
    }
    to {
        transform: translateY(0);
        opacity: 1;
    }
}

```

</style>

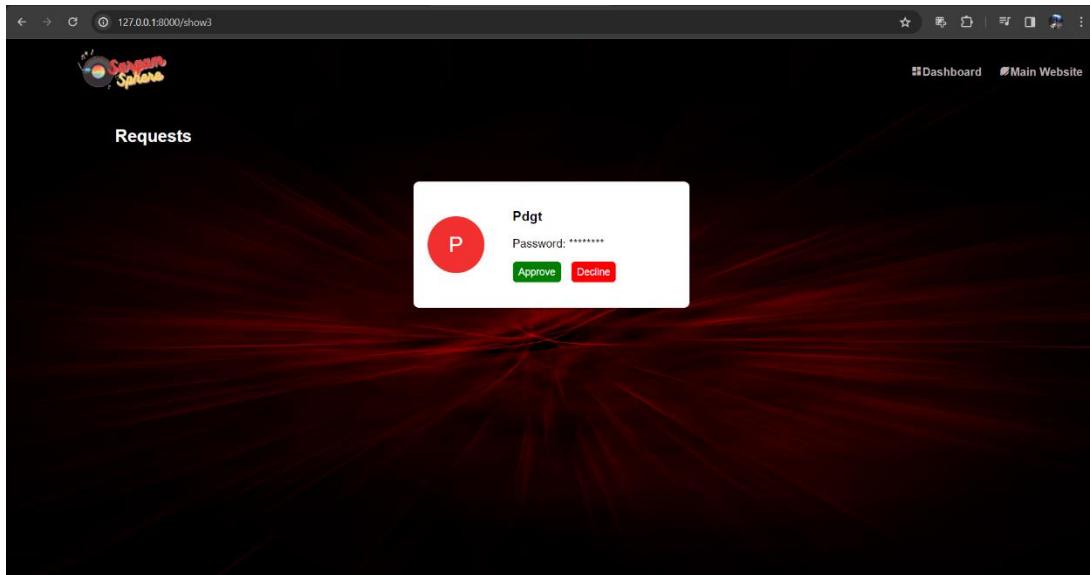
</head>

<body>

-
- <div class="container">
- <div class="message">Wait for your admin verification</div>
- <div class="small-text">Try to login after 48 hours to ensure your admin access</div>
- <div class="small-text">In the meantime, explore our website</div>
- </div>

</body>

- NEW ADMIN REQUEST PERMISSION page Output:



- NEW ADMIN REQUEST PERMISSION page Source-Code:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Request Management</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ09RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAlftl+Vegovlnnee1c9QX4TctnWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-referrer" />
    <script
        src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
    <link rel="stylesheet" href="{% static 'css/style2.css' %}">
    <link rel="stylesheet" href="{% static 'css/stylenav.css' %}">
    <link
        href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
</head>
```

```
<body id="bg" style="background-image: url('{% static "img/background/bg11.jpg" %}');">
    <!-- Navigation Bar -->
    <nav class="navbar">
        <!-- Logo -->
        <a class="navbar-brand" href="#home">
            <img class="img-responsive" src='{% static "img/logo/logo.png" %}' alt="" />
        </a>

        <div class="ml-auto">
            <a class="nav-link" href="adminpage"><i class='bx bxs-dashboard'></i>Dashboard</a>
            <a class="nav-link" href="index"><i class='bx bxs-planet'></i>Main Website</a>
        </div>
    </nav>
    <!-- End of Navigation Bar -->
    <!-- Request Information Section -->
    <div class="container">
        <h2 class="heading-white">Requests</h2>
        <div class="user-list">
            <!-- Request cards -->
            <!-- Loop through admins and display Request cards -->
        </div>
    </div>
```

```

{% for i in v %}
<div class="user-card">
  <div class="profile-picture">
    {{ i.name|slice:"1" }}
  </div>
  <div class="user-details">
    <h3>{{i.name}}</h3>
    <p>Password: *****</p>
    <p>
      <form action="accept_admin/{{i.id}}" method="get">
        <button type="submit" class="btn btn-approve">Approve</button>
      </form>
      <form action="del3/{{i.id}}" method="get">
        <button type="submit" class="btn btn-delete">Decline</button>
      </form>
    </p>
  </div>
<% endfor %>
<!-- End of Request cards -->
</div>
</div>
<!-- End of Request Information Section -->
</body>
</html>

```

- USER/ADMIN INFO page Output:

Users					
	Haaland	Password: *****			
	Kvaratskhelia	Password: *****			
	Phil Foden	Password: *****			
	Daniel Munoz	Password: *****			
	Virat	Password: *****			
	ddcasdv	Password: *****			

Admins			
	Cristiano Ronaldo	Password: *****	
	Luuk De Jong	Password: *****	
	Virat	Password: *****	

• USER INFO page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Management</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAlftI+Vegovlnee1c9QX4TctnWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-referrer" />
    <script
    src="https://unpkg.com/sweetalert/dist/sweetalert.m
    in.js"></script>
    <link rel="stylesheet" href="{% static 'css/style2.css' %}">
    <link rel="stylesheet" href="{% static
    'css/stylenav.css' %}">
    <link
    href="https://unpkg.com/boxicons@2.1.4/css/boxicon
    s.min.css" rel='stylesheet'>

</head>
<body id="bg" style="background-image: url('{% static
    'img/background/bg7-transformed.png" %}');">
    <!-- Navigation Bar -->
    <nav class="navbar">
        <!-- Logo -->
        <a class="navbar-brand" href="#home">
            <img class="img-responsive" src='{% static
                "img/logo/logo.png" %}' alt="" />
        </a>
        <div class="ml-auto">
            <a class="nav-link" href="adminpage"><i
            class='bx bxs-dashboard'></i>Dashboard</a>
            <a class="nav-link" href="index"><i
            class='bx bxs-planet'></i>Main Website</a>
        </div>
        <!-- Search Bar -->
        <div class="search-bar">
            <form method=POST action="{% url
            'search_user' %}">
                {% csrf_token %}
                <input type="text" name="query" id="search" placeholder="Search Users...">
                <button type="submit"><i class="fa fa-search" aria-hidden="true"></i></button>
            </form>
        </div>
    </nav>
    <!-- End of Navigation Bar -->

    <div class="container">
        <!-- User Information Section -->
        <h2 class="heading-white">Users</h2>
        <div class="user-list">
            <!-- User cards -->
            <!-- Loop through users and display user cards - ->
            {% for i in u %}
            <div class="user-card">
                <div class="profile-picture">
                    {{ i.name | slice:'1' }}
                </div>
                <div class="user-details">
                    <h3>{{i.name}}</h3>
                    <p>Password: *****</p>
                    <p>
                        <a href="edit/{{i.id}}">
                            <button class="btn btn-edit">
                                <i class="fa fa-edit"></i>
                            </button>
                        </a>
                    <!-- Add SweetAlert confirmation dialog - ->
                    <button class="btn btn-delete" onclick="confirmDelete('{{i.id}}', '{{i.name}}')">
                        <i class="fa fa-trash"></i>
                    </button>
                </p>
            </div>
            {% endfor %}
            <!-- End of user cards -->
        </div>
    </div>
    <!-- End of User Information Section -->
    <!-- JavaScript code for SweetAlert confirmation -->
    <script>
        function confirmDelete(adminId, adminName) {
            swal({

```

```

        title: "Are you sure?",
        text: "Once deleted, you will not be able to
recover " + adminName + "'s account!",
        icon: "warning",
        buttons: true,
        dangerMode: true,
    })
.then((willDelete) => {
    if (willDelete) {
        // If confirmed, proceed with the deletion
        window.location.href = "del/" + adminId;
    }
});

```

• ADMIN INFO page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>Admin Management</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css" integrity="sha512-
DTOQ09RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQ
eAlftI+Vegovlnnee1c9QX4TctnWMn13TZye+giMm8e2L
wA==" crossorigin="anonymous" referrerPolicy="no-
referrer" />
    <script
src="https://unpkg.com/sweetalert/dist/sweetalert.m
in.js"></script>
    <link rel="stylesheet" href="{% static 'css/style2.css'
%}">
    <link rel="stylesheet" href="{% static
'css/stylenav.css' %}">
    <link
href='https://unpkg.com/boxicons@2.1.4/css/boxicon
s.min.css' rel='stylesheet'>
</head>
<body id="bg" style="background-image: url('{% static
"img/background/bg7-transformed.png" %}');">
    <!-- Navigation Bar -->
    <nav class="navbar">
        <!-- Logo -->
        <a class="navbar-brand" href="#home">
            <img class="img-responsive" src='{% static
"img/logo/logo.png" %}' alt="" />
        </a>
        <div class="ml-auto">
            <a class="nav-link" href="adminpage"><i
class='bx bxs-dashboard'></i>Dashboard</a>
            <a class="nav-link" href="index"><i class='bx
bxs-planet'></i>Main Website</a>
        </div>

```

```

<!-- Search Bar -->
<div class="search-bar">
    <form method=POST action="{% url
'search_admin' %}">
        {% csrf_token %}
        <input type="text" name="adminquery"
id="search" placeholder="Search Admins...">
        <button type="submit"><i class="fa fa-
search" aria-hidden="true"></i></button>
    </form>
</div>
</nav>
<!-- End of Navigation Bar -->

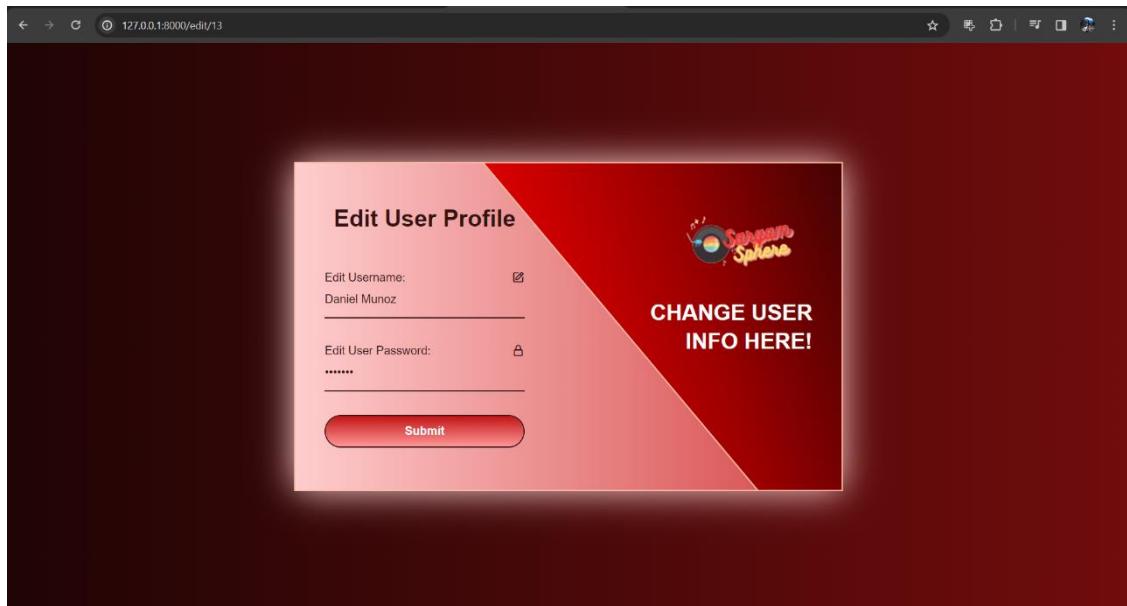
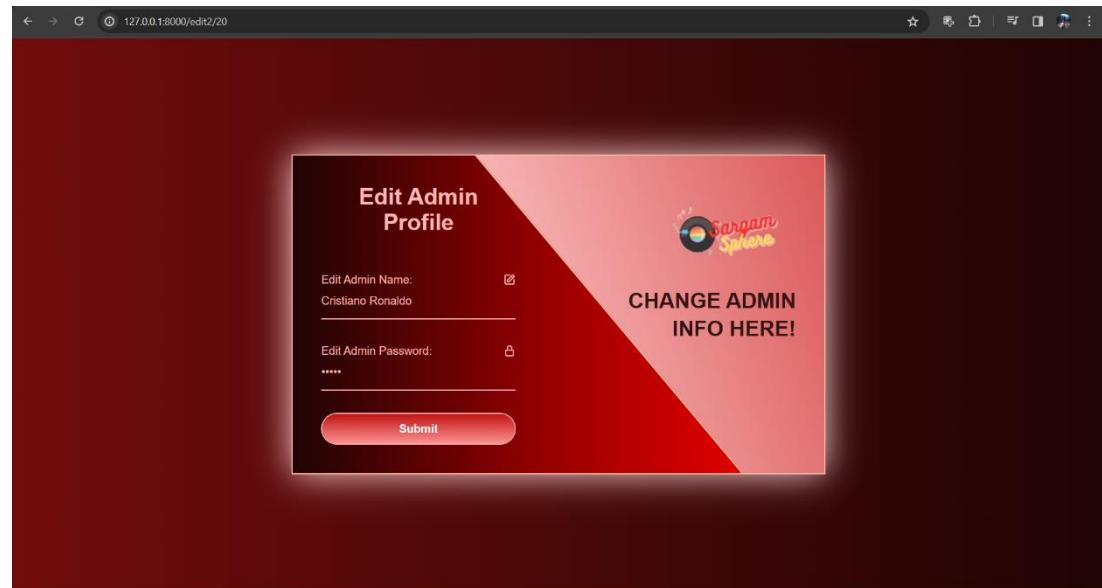
<!-- Admin Information Section -->
<div class="container">
    <h2 class="heading-white">Admins</h2>
    <div class="user-list">
        <!-- User cards -->
        <!-- Loop through admins and display admin
cards -->
        {% for i in a %}
            <div class="user-card">
                <div class="profile-picture">
                    {{ i.name | slice:"1" }}
                </div>
                <div class="user-details">
                    <h3>{{i.name}}</h3>
                    <p>Password: *****</p>
                    <p>
                        <a href="edit2/{{i.id}}">
                            <button class="btn btn-edit">
                                <i class="fa fa-edit"></i>
                            </button>
                        </a>
                    <!-- Add SweetAlert confirmation dialog -
->
                        <button class="btn btn-delete"
onclick="confirmDelete('{{i.id}}', '{{i.name}}')">
                            <i class="fa fa-trash"></i>
                        </button>
                    
```

```

        </p>
        </div>
    </div>
    {% endfor %}
    <!-- End of admin cards -->
</div>
</div>
<!-- End of Admin Information Section -->
<!-- JavaScript code for SweetAlert confirmation -->
<script>
function confirmDelete(adminId, adminName) {
    swal({
        title: "Are you sure?",
        text: "Once deleted, you will not be able to
recover " + adminName + "'s account!",
        icon: "warning",
        buttons: true,
        dangerMode: true,
    })
    .then((willDelete) => {
        if (willDelete) {
            // If confirmed, proceed with the deletion
            window.location.href = "del2/" + adminId;
        }
    });
}
</script>
</body>
</html>

```

- USER/ADMIN INFO EDIT page Output:



• USER INFO EDIT page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content="width=device-
width,initial-scale=1.0">
    <title>Login and Registration Form in HTML and CSS
| Codehal</title>
    <link rel="stylesheet" href="style.css">
    <link
href='https://unpkg.com/boxicons@2.1.4/css/boxicon
s.min.css' rel='stylesheet'>
</head>
<style>
*<{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: linear-gradient(to right, #1f0505,
#710d0d);
}
.wrapper{
    position: relative;
    width: 750px;
    height: 450px;
    background: linear-gradient(to right, #ffcece,
#d44545);
    border: 2px solid #ffbda5;
    box-shadow: 0 0 50px #fed5d5;
    overflow: hidden;
}
.wrapper .form-box {
    position: absolute;
    top: 0;
    width: 50%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
}
.wrapper .form-box.login{
    left: 0;
    padding: 0 60px 0 40px;
}
.form-box h2 {
    font-size: 32px;
    color: #341313;
    text-align: center;
}
.form-box .form-group{
    position: relative;
    width: 100%;
    height: 50px;
    margin: 50px 0;
}
.form-group input {
    width: 100%;
    height: 100%;
    background: transparent;
    border: none;
    outline: none;
    border-bottom: 2px solid #341313;
    padding-right: 23px;
    font-size: 16px;
    color:#341313;
    font-weight: 500;
}
.form-group label {
    position: absolute;
    top: 25%;
    left: 0;
    transform: translateY(-50%);
    font-size: 16px;
    color: #341313;
    pointer-events: none;
}
.form-group i{
    position: absolute;
    top: 25%;
    right: 0;
    transform: translateY(-50%);
    font-size: 18px;
    color:#341313;
}
.submit-btn{
    position: relative;
    width: 100%;
    height: 45px;
    background: transparent;
    border: 2px solid #341313;
    outline: none;
    border-radius: 40px;
    cursor: pointer;
}

```

```

font-size: 16px;
color: #fff;
font-weight: 600;
z-index: 1;
overflow: hidden;
}
.submit-btn::before {
content: "";
position: absolute;
top: -100%;
left: 0;
width: 100%;
height: 300%;
background: linear-gradient(#290808, #c21414,
#ff9696, #6f0a0a);
z-index: -1;
transition: .5s;
}
.submit-btn:hover::before{
top: 0;
}
.wrapper .info-text{
position: absolute;
top: 0;
width: 50%;
height: 100%;
display: flex;
flex-direction: column;
justify-content: center;
}
.wrapper .info-text.edit{
right:0;
text-align: right;
padding: 0 40px 30px 60px;
}
.logo{
position:static;
padding-left: 100px;
padding-right: 20px;
padding-bottom: 35px;
width: 100%;
}
.info-text h2 {
font-size: 30px;
color: #fff;
padding-bottom: 85px;
line-height: 1.3;
text-transform: uppercase;
}

```

```

}
.wrapper .bg-animate {
position: absolute;
top: -15px;
right: 0;
width: 850px;
height: 600px;
background: linear-gradient(45deg,
#ff0000,#9c0000,#1f0505);
border-bottom: 3px solid #ffbda5;
transform: rotate(10deg) skewY(40deg);
transform-origin: bottom right;
}
</style>
<body>
<div class="wrapper">
<span class="bg-animate"></span>
<div class="form-box login">
<h2>Edit User Profile</h2>
<form action="../edcode/{{u.id}}" method="GET">
<div class="form-group">
<label for="name">Edit Username:</label><br>
<input type="text" id="name" name="a1" value="{{u.name}}>
<i class='bx bx-edit' ></i>
</div>
<div class="form-group">
<label for="password">Edit User Password:</label><br>
<input type="password" id="password" name="a2" value="{{u.pwd}}>
<i class='bx bx-lock-alt' ></i>
</div>
<button type="submit" class="submit-btn">Submit</button>
</form>
</div>
<div class="info-text edit">

<h2>Change User Info here!</h2>
</div>
</div>
</body>
</html>

```

- ADMIN INFO EDIT page Source-Code:

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content="width=device-
width,initial-scale=1.0">
    <title>Login and Registration Form in HTML and CSS
| Codehal</title>
    <link rel="stylesheet" href="style.css">
    <link
href='https://unpkg.com/boxicons@2.1.4/css/boxicon
s.min.css' rel='stylesheet'>
</head>
<style>
*<{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: linear-gradient(to left, #1f0505,
#710d0d);
}
.wrapper{
    position: relative;
    width: 750px;
    height: 450px;
    background: linear-gradient(to left,
#ff0000,#9c0000,#1f0505);
    border: 2px solid #ffbda5;
    box-shadow: 0 0 50px #fed5d5;
    overflow: hidden;
}
.wrapper .form-box {
    position: absolute;
    top: 0;
    width: 50%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
}

```

```

    }
.wrapper .form-box.login{
    left: 0;
    padding: 0 60px 0 40px;
}
.form-box h2 {
    font-size: 32px;
    color: #ffb7b7;
    text-align: center;
}
.form-box .form-group{
    position: relative;
    width: 100%;
    height: 50px;
    margin: 50px 0;
}
.form-group input {
    width: 100%;
    height: 100%;
    background: transparent;
    border: none;
    outline: none;
    border-bottom: 2px solid #ffb7b7;
    padding-right: 23px;
    font-size: 16px;
    color: #ffb7b7;
    font-weight: 500;
}
.form-group label {
    position: absolute;
    top: 25%;
    left: 0;
    transform: translateY(-50%);
    font-size: 16px;
    color: #ffb7b7;
    pointer-events: none;
}
.form-group i{
    position: absolute;
    top: 25%;
    right: 0;
    transform: translateY(-50%);
    font-size: 18px;
    color: #ffb7b7;
}
.submit-btn{
    position: relative;
    width: 100%;
    height: 45px;
}

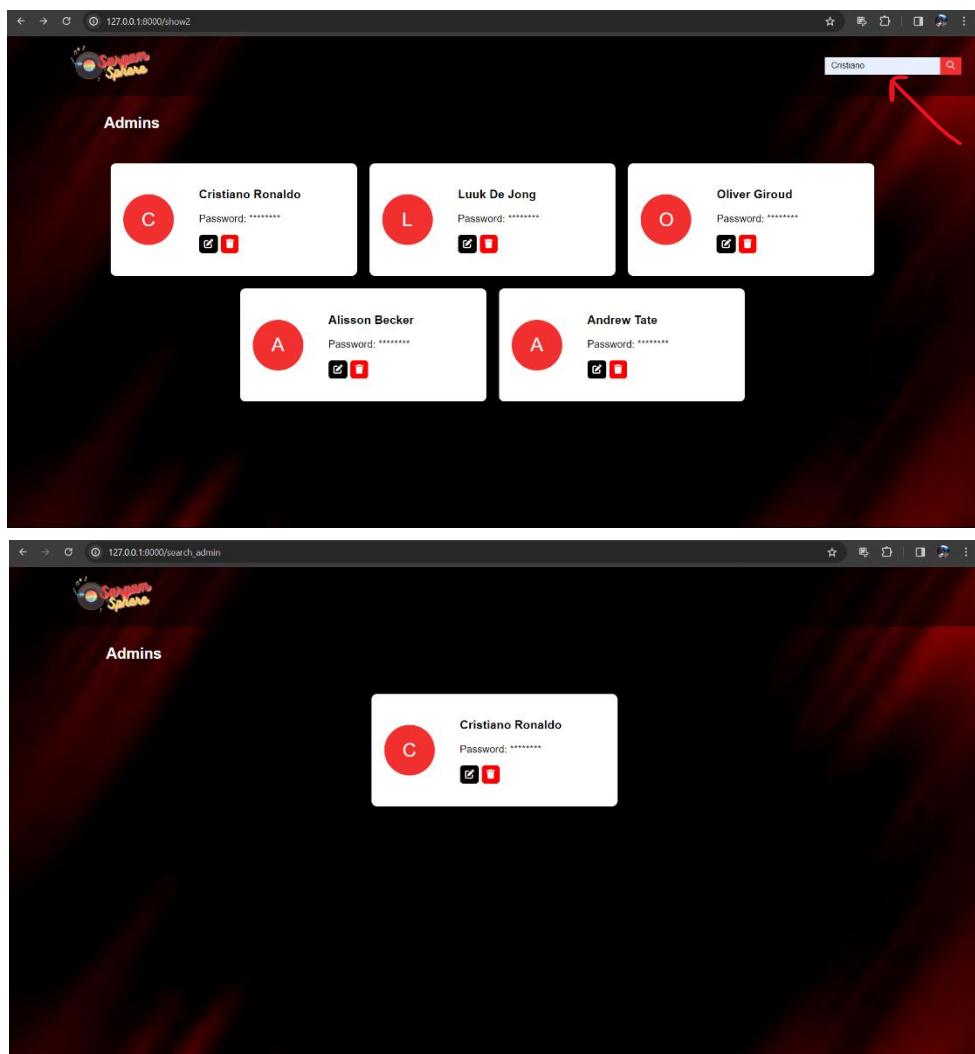
```

```

background: transparent;
border: 2px solid #ffb7b7;
outline: none;
border-radius: 40px;
cursor: pointer;
font-size: 16px;
color: #fff;
font-weight: 600;
z-index: 1;
overflow: hidden;
}
.submit-btn::before {
  content: "";
  position: absolute;
  top: -100%;
  left: 0;
  width: 100%;
  height: 300%;
  background: linear-gradient(#290808, #c21414,
#ff9696,#f0a0a0);
  z-index: -1;
  transition: .5s;
}
.submit-btn:hover::before{
  top: 0;
}
.wrapper .info-text{
  position: absolute;
  top: 0;
  width: 50%;
  height: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
}
.wrapper .info-text.edit{
  right:0;
  text-align: right;
  padding: 0 40px 30px 60px;
}
.logo{
  position:static;
  padding-left: 100px;
  padding-right: 20px;
  padding-bottom: 35px;
  width: 100%;
}
.info-text h2 {
  font-size: 30px;
  color: #341313;
  padding-bottom: 85px;
  line-height: 1.3;
  text-transform: uppercase;
}
.wrapper .bg-animate {
  position: absolute;
  top: -15px;
  right: 0;
  width: 850px;
  height: 600px;
  background: linear-gradient(45deg, #ffcece,
#d44545);
  border-bottom: 3px solid #ffbda5;
  transform: rotate(10deg) skewY(40deg);
  transform-origin: bottom right;
}
</style>
<body>
<div class="wrapper">
  <span class="bg-animate"></span>
  <div class="form-box login">
    <h2>Edit Admin Profile</h2>
    <form action="../edcode2/{{a.id}}"
method="GET">
      <div class="form-group">
        <label for="name">Edit Admin
Name:</label><br>
        <input type="text" id="name" name="a1"
value="{{a.name}}">
        <i class='bx bx-edit' ></i>
      </div>
      <div class="form-group">
        <label for="password">Edit Admin
Password:</label><br>
        <input type="password" id="password"
name="a2" value="{{a.pwd}}">
        <i class='bx bx-lock-alt' ></i>
      </div>
      <button type="submit" class="submit-
btn">Submit</button>
    </form>
  </div>
  <div class="info-text edit">
    <!-- Add your logo here -->
    
    <h2>Change Admin Info here!</h2>
  </div>
</div>
</body>
</html>

```

- ADMIN SEARCH page Output:



- ADMIN SEARCH page Source-Code:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Management</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ9RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAIfI+Vegovlnee1c9QX4TctnWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-referrer" />
    <script
        src="https://unpkg.com/sweetalert/dist/sweetalert.m
```

```
in.js"></script>
    <link rel="stylesheet" href="{% static 'css/style2.css' %}">
</head>
<body id="bg" style="background-image: url('{% static "img/background/bg7-transformed.png" %}');">
    <!-- Navigation Bar -->
    <nav class="navbar">
        <!-- Logo -->
        <a class="navbar-brand" href="#home">
            <img class="img-responsive" src='{% static "img/logo/logo.png" %}' alt="" />
        </a>
    </nav>
    <!-- End of Navigation Bar -->
    <div class="container">
        <!-- User Information Section -->
```

```

<h2 class="heading-white">Admins</h2>
<div class="user-list">
    <!-- User cards -->
    <!-- Loop through users and display user cards ->
    {% if admins %}
        {% for admin in admins %}
            <div class="user-card">
                <div class="profile-picture">
                    {{ admin.name|slice:"1" }}
                </div>
                <div class="user-details">
                    <h3>{{admin.name}}</h3>
                    <p>Password: *****</p>
                    <p>
                        <a href="edit/{{admin.id}}">
                            <button class="btn btn-edit">
                                <i class="fa fa-edit"></i>
                            </button>
                        </a>
                        <a href="del/{{admin.id}}">
                            <button class="btn btn-delete">
                                <i class="fa fa-trash"></i>
                            </button>
                        </a>
                    </p>
                </div>
            </div>
        {% endfor %}
    {% else %}
        <h2 class="heading-white">NO ADMINS FOUND!!!</h2>
        {% endif %}
        <!-- End of user cards -->
    </div>
    <!-- End of User Information Section -->
</body>
</html>

```

- USER SEARCH page Output:

The screenshots illustrate the User Search feature. The top one shows a list of users with their details and a search bar. The bottom one shows the results of a search for 'Haaland'.

User	Initial	Password
Haaland	H	*****
Kvaratskhelia	K	*****
Ameen Al Dakhil	A	*****
Phil Foden	P	*****
Daniel Munoz	D	*****
Bruno Fernandez	B	*****

- USER SEARCH page Source-Code:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Management</title>
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ09RWCH3ppGqcWaEA1BIZOC6xalwEsw9c2QQeAlftl+Vegovlhee1c9QX4TctnWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-referrer" />
    <script
        src="https://unpkg.com/sweetalert/dist/sweetalert.m
        in.js"></script>
    <link rel="stylesheet" href="{% static 'css/style2.css'
        %}">
```

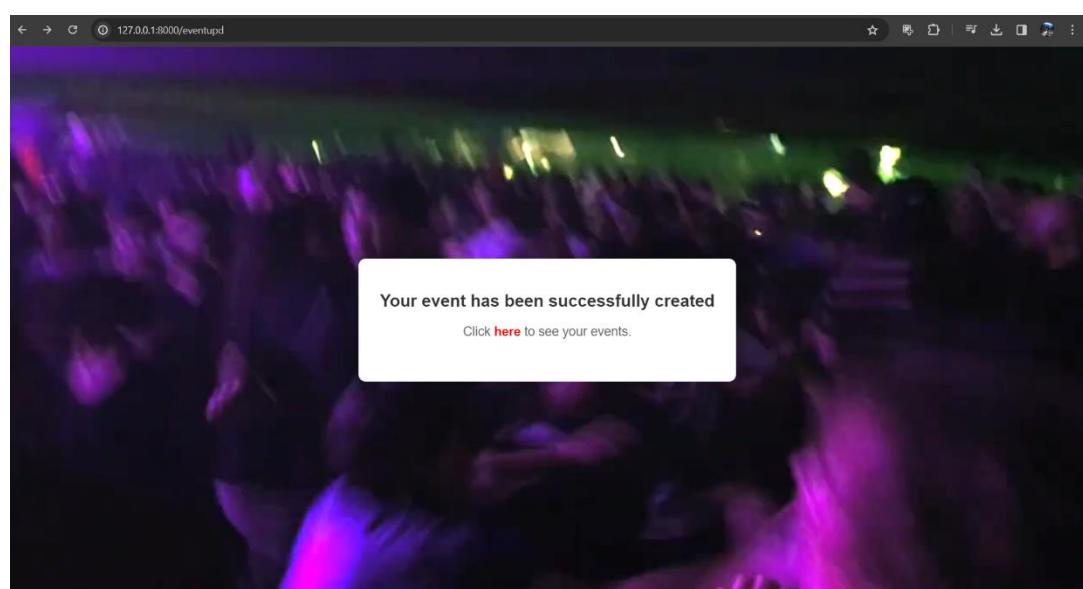
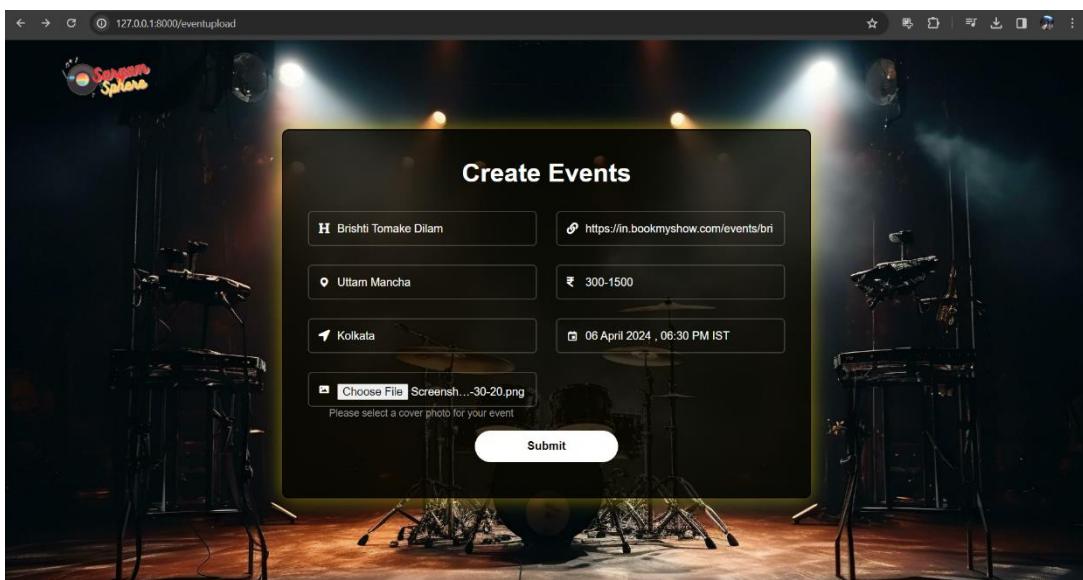
```
</head>
<body id="bg" style="background-image: url('{% static
    "img/background/bg7-transformed.png" %}');">
    <!-- Navigation Bar -->
    <nav class="navbar">
        <!-- Logo -->
        <a class="navbar-brand" href="#home">
            <img class="img-responsive" src='{% static
                "img/logo/logo.png" %}' alt="" />
        </a>
    </nav>
    <!-- End of Navigation Bar -->
    <div class="container">
        <!-- User Information Section -->
        <h2 class="heading-white">Users</h2>
        <div class="user-list">
            <!-- User cards -->
            <!-- Loop through users and display user cards ->
            {% if users %}
```

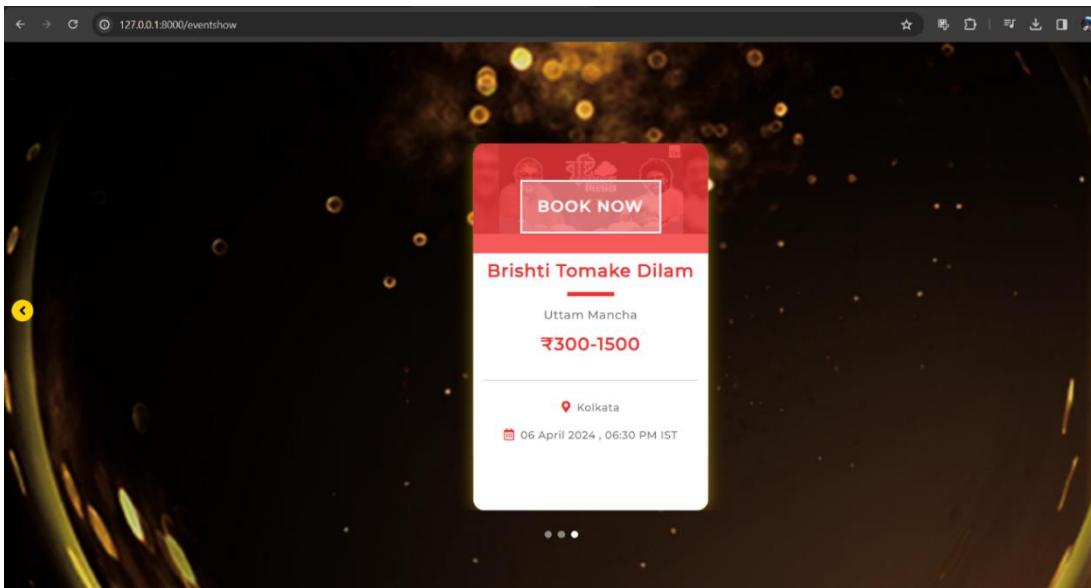
```

{%
for user in users %}
<div class="user-card">
<div class="profile-picture">
{{ user.name|slice:"-1" }}
</div>
<div class="user-details">
<h3>{{user.name}}</h3>
<p>Password: *****</p>
<p>
<a href="edit/{{user.id}}">
<button class="btn btn-edit">
<i class="fa fa-edit"></i>
</button>
</a>
<a href="del/{{user.id}}">
<button class="btn btn-delete">
<i class="fa fa-trash"></i>

```

- EVENTS PAGE:





• EVENT UPLOAD page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css" integrity="sha512-DTOQ09RWCH3ppGqcWaEA1BIZOC6xxalwEsw9c2QQeAIftI+Vegovlnee1c9QX4TctnWMn13TZye+giMm8e2LwA==" crossorigin="anonymous" referrerPolicy="no-referrer" />
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
    <link
        href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
    <link rel="stylesheet" href="{% static 'css/style4.css' %}">

    <title>Event upload</title>
</head>
<style>
    * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        font-family: "Poppins", sans-serif;
    }

```

```

body{
    display:flex;
    justify-content:center;
    align-items:center;
    min-height:100vh;
    background-size: cover;
    background-position: center;
}
.file-instruction {
    position: absolute;
    bottom: -17px;
    left: 30px;
    font-size: 14px;
    color: #999;
}
</style>
<body id="bg" style="background-image: url('{% static 'img/background/ee.jpg' %}');">
    <img src='{% static "img/logo/logo.png" %}' alt="Logo" class="logo">
    <div class="wrapper">
        <h1>Create Events</h1>
        <form action="eventupd" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            <div class="input-box">
                <div class="input-field">
                    <input type="text" placeholder="Event name" name="a2">
                    <i class="fa fa-header"></i>
                </div>
                <div class="input-field">
                    <input type="text" placeholder="Bookmyshow link" name="a3">

```

```

        <i class="fa fa-link"></i>
    </div>
</div>
<div class="input-box">
    <div class="input-field">
        <input type="text" placeholder="Place"
name="a4">
            <i class='bx bxs-map'></i>
        </div>

        <div class="input-field">
            <input type="text" placeholder="Price"
name="a5">
                <i class="fa fa-inr"></i>
            </div>
        </div>
        <div class="input-box">
            <div class="input-field">
                <input type="text"
placeholder="Location" name="a6">
                    <i class="fa fa-location-arrow" ></i>
            </div>

```

```

<div class="input-field">
    <input type="text" placeholder="Date &
Time" name="a7">
        <i class='bx bx-calendar-event'></i>
    </div>
</div>
<div class="input-box">
    <div class="input-field">
        <input type="file" name="a1" class="file-
input">
            <i class='bx bxs-image'></i>
        <p class="file-instruction">Please select
a cover photo for your event</p>
    </div>
</div>
<button type="submit" class="btn"
value="Upload"> Submit</button>
</form>
</div>
</body>
</html>

```

● EVENT UPLOADED SUCCESSFUL page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <link rel="shortcut icon" href="{% static
'img/logo/favicon.ico' %}">
    <title>Event Upload Success</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f2f2f2;
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
    }
    .container {
        text-align: center;
        padding: 30px;
        background-color: #fff;
        border-radius: 10px;
        box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
    }

```

```

h1 {
    color: #333;
    font-size: 24px;
    margin-bottom: 20px;
}
p {
    color: #666;
    font-size: 18px;
    margin-bottom: 30px;
}
a {
    text-decoration: none;
    color: #FF0000;
    font-weight: bold;
}
a:hover {
    text-decoration: underline;
}
.background-clip{
    position: absolute;
    right: 0;
    bottom: 0;
    z-index:-1;
    transition: 1s;
}
@media (min-aspect-ratio: 16/9){
    .background-clip{
        width: 100%;

```

```

        height: auto;
    }
}

@media(max-aspect-ratio: 16/9){
    .background-clip{
        width: auto;
        height: 100%;
    }
}

```

</style>

</head>

<body>

<div class="container">

```

<video autoplay loop muted plays-inline
class="background-clip">
    <source src="{% static '/video/3.mp4' %}"
type="video/mp4">
</video>
<h1>Your event has been successfully
created</h1>
<p>Click <a href="eventshow">here</a> to see
your events.</p>
</div>
</body>
</html>

```

• EVENTS page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <link
        href="https://fonts.googleapis.com/css2?family=Mon
        tserrat:wght@500&display=swap" rel="stylesheet">
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-
        awesome/5.14.0/css/all.min.css">
    <link rel="stylesheet" href="{% static
        'css/styleevent.css' %}">
    <link rel="shortcut icon" href="{% static
        'img/logo/favicon.ico' %}">
    <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/swiper@9/swiper-
        bundle.min.css">
    <title>Events Ticket Card</title>
<style>
    /* Additional CSS for the swiper container */
    .swiper-container {
        position: relative; /* Ensure the navigation
        arrows position correctly */
        width: 100%;
        padding-top: 50px; /* Adjust as needed based
        on your design */
        padding-bottom: 50px; /* Adjust as needed
        based on your design */
    }
    .swiper-slide {
        background-size: cover;
        background-position: center;
        width: calc(100% / 3 - 40px); /* Adjust the
        width of each slide */
    }

```

```

        height: 500px; /* Adjust the height of each
        slide as needed */
        margin-right: 20px; /* Adjust the margin
        between slides */
        border-radius: 15px;
        overflow: hidden;
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        box-sizing: border-box; /* Include padding and
        border in the element's total width and height */
    }
    .swiper-button-prev,
    .swiper-button-next {
        position: absolute;
        top: 50%;
        transform: translateY(-50%);
        width: 30px;
        height: 30px;
        background-color: gold; /* Set background
        color to gold */
        border-radius: 50%; /* Make it a circle */
        cursor: pointer;
        z-index: 10;
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .swiper-button-prev:after {
        content: '\f104'; /* Unicode for right arrow
        ('\f054' for Font Awesome 5) */
        font-family: 'Font Awesome 5 Free'; /* Font
        Awesome for arrow */
        font-weight: 900;
        color: black;
        font-size: 20px;
    }

```

```

}
.swiper-button-next:after{
  content: '\f054'; /* Unicode for right arrow
  '\f054' for Font Awesome 5 */
  font-family: 'Font Awesome 5 Free'; /* Font
Awesome for arrow */
  font-weight: 900;
  color: black;
  font-size: 16px;
}
.swiper-button-prev {
  left: 10px;
}
.swiper-button-next {
  right: 10px;
}
.swiper-button-disabled {
  visibility: hidden; /* Hide the disabled button */
}
.swiper-pagination {
  position: absolute;
  bottom: 10px;
  left: 0;
  width: 100%;
  text-align: center;
}
.swiper-pagination-bullet {
  width: 10px;
  height: 10px;
  background-color: #fff;
  opacity: 0.5;
  border-radius: 50%;
  display: inline-block;
  margin: 0 5px;
  cursor: pointer;
  transition: opacity 0.2s ease;
}
.swiper-pagination-bullet-active {
  opacity: 1;
}
</style>
</head>
<body id="bg" style="background-image: url('{{ static
"img/background/bgevent2.jpg" }}');">
  <!-- Swiper container -->
  <div class="swiper-container">
    <div class="swiper-button-prev swiper-button-
disabled"></div>
    <div class="swiper-button-next"></div>
    <div class="swiper-wrapper">
      <!-- Your event cards (items) go here -->
      {% for i in u %}
        <div class="swiper-slide">
          <div class="item-container">
            <div class="img-container">
              
            </div>
            <div class="body-container">
              <div class="overlay"></div>
              <div class="event-info">
                <p class="title">{{ i.etitle }}</p>
                <div class="separator"></div>
                <p class="info">{{ i.eplace }}</p>
                <p class="price">${{ i.eprice }}</p>
              </div>
            </div>
            <div class="additional-info">
              <p class="info">
                <i class="fas fa-map-marker-
alt"></i>
                {{ i.elocation }}</p>
              <p class="info">
                <i class="far fa-calendar-alt"></i>
                {{ i.etiming }}</p>
              </div>
            </div>
            <a class="action" href="{{ i.elink }}"
target="_blank">Book Now</a>
          </div>
        </div>
      <% endfor %>
      <!-- End of event cards -->
    </div>
    <!-- Add pagination if needed -->
    <div class="swiper-pagination"></div>
  </div>
  <!-- Swiper JS -->
  <script
src="https://cdn.jsdelivr.net/npm/swiper@9/swiper-
bundle.min.js"></script>
  <script>
    // Initialize Swiper
    var swiper = new Swiper(".swiper-container", {
      slidesPerView: 1,
      spaceBetween: 30,
      loop: true,
      navigation: {
        nextEl: ".swiper-button-next",
        prevEl: ".swiper-button-prev",
      },
      pagination: {
        el: ".swiper-pagination",
        clickable: true,
      },
      breakpoints: {
        // Adjust slide width for different screen sizes if
        needed
      }
    });
  </script>

```

```

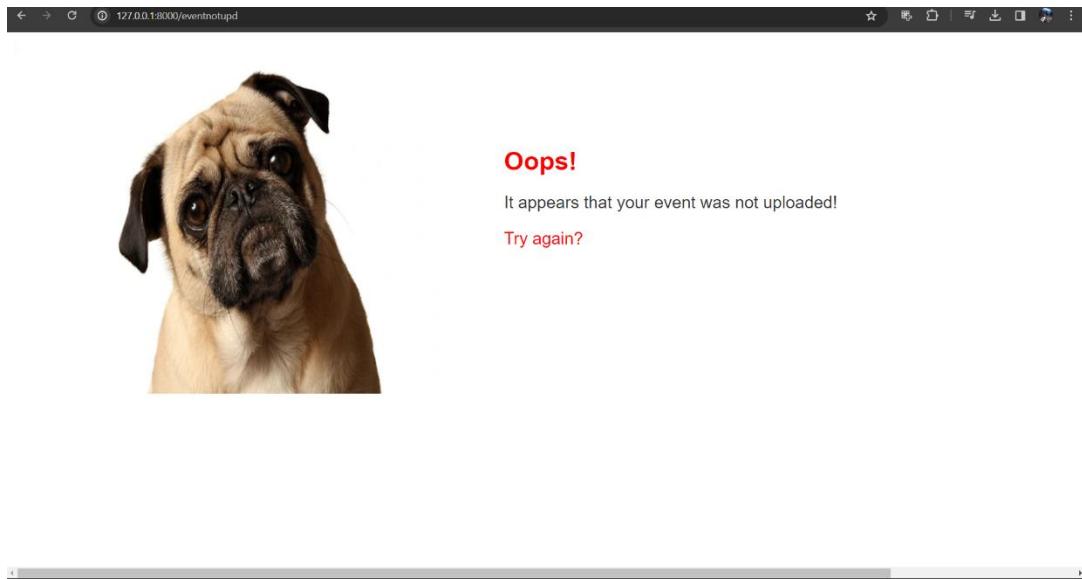
768: {
  slidesPerView: 1,
  spaceBetween: 0
}
},
on: {
  slideChange: function () {
    // Check if it's the first slide and disable the
    prev button
    if (this.isBeginning) {
      document.querySelector('.swiper-button-
prev').classList.add('swiper-button-disabled');
    } else {
      document.querySelector('.swiper-button-
prev').classList.remove('swiper-button-disabled');
    }

    // Check if it's the last slide and disable the next
    button
    if (this.isEnd) {
      document.querySelector('.swiper-button-
next').classList.add('swiper-button-disabled');
    } else {
      document.querySelector('.swiper-button-
next').classList.remove('swiper-button-disabled');
    }
  }
}

// Manually update pagination bullet if looping
from last to first slide
if (this.isEnd && this.isBeginning) {
  var bullets =
document.querySelectorAll('.swiper-paginatio-
bullet');
  bullets.forEach(function(bullet) {
    bullet.classList.remove('swiper-paginatio-
bullet-active');
  });
  bullets[0].classList.add('swiper-paginatio-
bullet-active');
}
}
});
</script>
</body>
</html>

```

- EVENTS NOT UPLOADED SUCCESSFULLY page Output:



- EVENTS NOT UPLOADED SUCCESSFULLY page

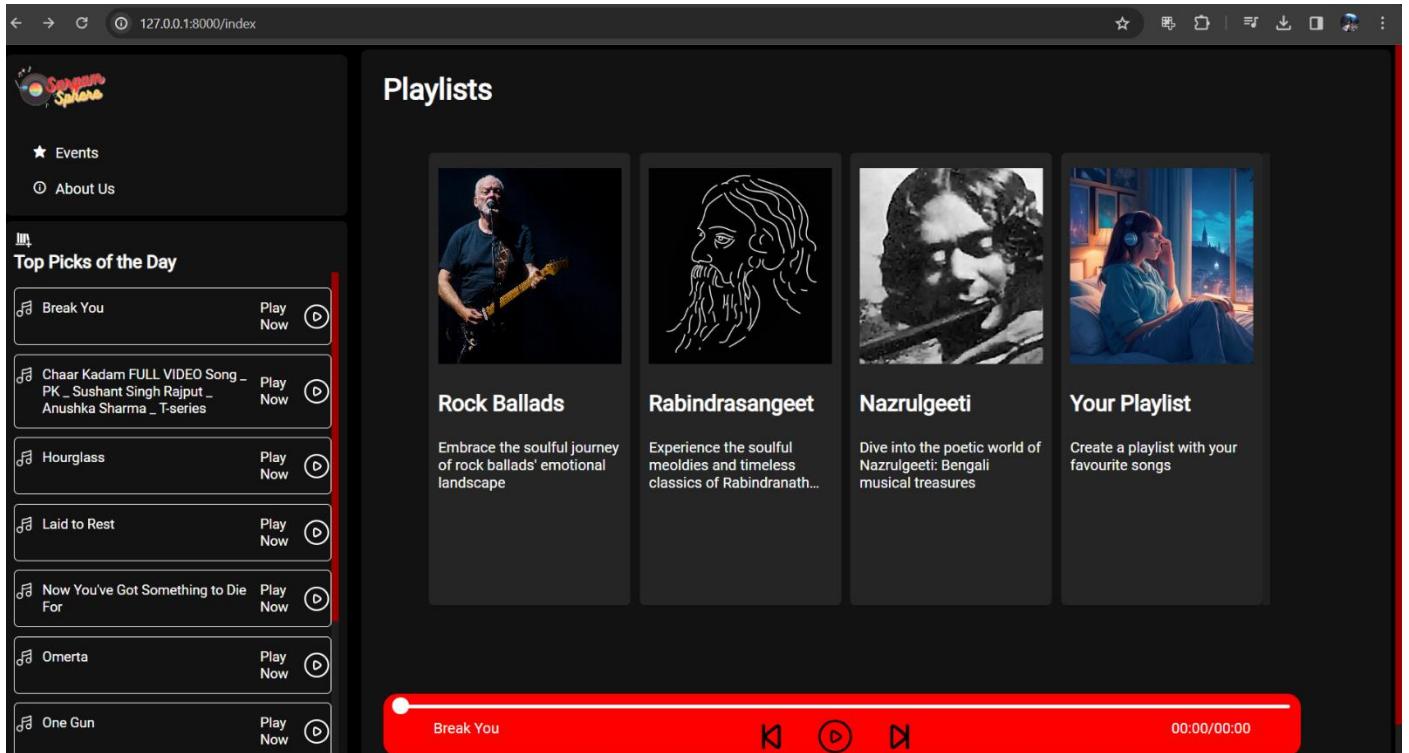
Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
    <title>Oops!</title>
    <style>
        .left{
            width: 25vw;
            padding: 5px;
        }
        .left img {
            height: 500px;
            width: 600px;
        }
        .right{
            width: 75vw;
            position: absolute;
            margin-left: 700px;
            margin-top: 130px;
            font-family: Arial, sans-serif; /* Change font family */
        }
        .right h1 {
            color: #FF0000; /* Change color of h1 */
            font-size: 36px; /* Change font size of h1 */
            font-weight: bold; /* Make h1 bold */
            margin-bottom: 20px; /* Add some margin below h1 */
        }
        .right p {
            font-size: 24px; /* Change font size of paragraphs */
            color: #333; /* Change color of paragraphs */
            margin-bottom: 10px; /* Add some margin below paragraphs */
        }
        .right a {
            font-size: 24px; /* Change font size of anchor */
            color: #FF0000; /* Change color of anchor */
            text-decoration: none; /* Remove underline from anchor */
        }
        .right a:hover {
            text-decoration: underline; /* Add underline on hover */
        }
    </style>
</head>
<body>
    <div class="right">
        <h1>Oops!</h1>
        <p>It appears that your event was not uploaded!</p>
        <p><a href="eventupload">Try again?</a></p>
    </div>
    <div class="left">
        
    </div>
</body>
</html>

```

• Music Streaming Platform Main Webpage Output:



• Music Streaming Platform Main Webpage Source Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{% static "css/style.css" %}">
    <link rel="stylesheet" href="{% static "css/utility.css" %}">
    <title> SargamSphere</title>
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicon-s.min.css' rel='stylesheet'>
</head>
<style>
    /* Add button styles */
    .add-button {
        top: 10px;
        right: 20px;
        background-color: transparent;
        color: white;
        border: none;
        padding: 10px 20px;
        border-radius: 5px;
        text-decoration: none;
        cursor: pointer;
        transition: background-color 0.3s ease;
        display: inline-block;
        width: 315px;
    }
    .add-button:hover {
        background-color: #F23030;
    }
    .add-button:focus {
        outline: none;
    }
    .add-button i {
        margin-right: 5px;
    }
    .aboutus-button {
        top: 10px;
        right: 20px;
        background-color: transparent;
        color: white;
        border: none;
        padding: 10px 20px;
        border-radius: 5px;
        text-decoration: none;
    }

```

```

cursor: pointer;
transition: background-color 0.3s ease;
display: inline-block;
width: 315px;
}
.aboutus-button:hover {
background-color: #F23030;
}
.aboutus-button:focus {
outline: none;
}
.aboutus-button i {
margin-right: 5px;
}

```

</style>

<body>

- [Events](#)
- [About Us](#)

Top Picks of the Day

-

Playlists

Embrace the soulful journey of rock ballads' emotional landscape

Experience the soulful meoldies and timeless classics of Rabindranath Tagore

```

</a>
<a href="nazrulgeeti" style="text-decoration: none; color: inherit;">
    <div class="card ">
        <div class="play">
            <svg
                xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="40" height="40">
                <circle cx="12" cy="12" r="12" fill="red" />
                <path d="M18.8906 12.846C18.5371 14.189 16.8667 15.138 13.5257 17.0361C10.296 18.8709 8.6812 19.7884 7.37983 19.4196C6.8418 19.2671 6.35159 18.9776 5.95624 18.5787C5 17.6139 5 15.7426 5 12C5 8.2574 5 6.3861 5.95624 5.42132C6.35159 5.02245 6.8418 4.73288 7.37983 4.58042C8.6812 4.21165 10.296 5.12907 13.5257 6.96393C16.8667 8.86197 18.5371 9.811 18.8906 11.154C19.0365 11.7084 19.0365 12.2916 18.8906 12.846Z" fill="white" />
            </svg>
        </div>
        
    </div>
    <h2> Nazrulgeeti </h2>
    <p> Dive into the poetic world of Nazrulgeeti: Bengali musical treasures </p>
    </a>
    <a href="userplaylistshow" style="text-decoration: none; color: inherit;">
        <div class="card ">
            <div class="play">
                <svg
                    xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="40" height="40">
                    <circle cx="12" cy="12" r="12" fill="red" />
                    <path d="M18.8906 12.846C18.5371 14.189 16.8667 15.138 13.5257 17.0361C10.296 18.8709 8.6812 19.7884 7.37983 19.4196C6.8418 19.2671 6.35159 18.9776 5.95624 18.5787C5 17.6139 5 15.7426 5 12C5 8.2574 5 6.3861

```

- About-Us Webpage Output:

About SargamSphere

Welcome to SargamSphere, where music meets innovation and creativity intertwines with technology. We are more than just a music streaming platform; we are a community of music enthusiasts, creators, and listeners united by our love for melodies, rhythms, and harmonies from around the world. At SargamSphere, we believe that music has the power to transcend boundaries, evoke emotions, and bring people together like nothing else can. Our mission is to provide a seamless and immersive musical experience that caters to the diverse tastes and preferences of our global audience.

Our Journey

SargamSphere was born out of a shared passion for music and a desire to revolutionize the way people discover, enjoy, and engage with their favorite tunes. Founded in 2024, our journey began with a vision to create a platform that not only offers unlimited access to a vast library of songs but also fosters a sense of belonging and connection among music lovers worldwide.

Passionate about blending design and functionality to resonate with every beat and melody. Let's build an immersive sonic oasis together

-- **Sangam Banerjee**
Director

Architecting and sculpting robust backend infrastructures to orchestrate smooth, scalable music streaming services & lot more , ensuring every line of code hits the right note

-- **Sukanya Das**
Developer-Producer

Crafting captivating interfaces that are aesthetically pleasing and harmonizes usability, ensuring every pixel sings in perfect rhythm with the user's experience.

-- **Priyan Mukherjee**
Artist

Curating a symphony of engaging content to enrich your music streaming journey, harmonizing diverse voices and genres into a seamless listening experience.

-- **Rector Chakraborty**
Composer

• About-Us Webpage Source-Code:

```
{% load static %}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible"  
content="IE=edge">  
    <meta name="viewport" content="width=device-  
width,initial-scale=1.0">  
    <title>About Us</title>  
    <link rel="stylesheet" href="{% static 'style.css' %}">  
    <link rel="shortcut icon" href="{% static  
'img/logo/favicon.ico' %}">  
    <link  
href="https://unpkg.com/boxicons@2.1.4/css/boxico  
ns.min.css" rel="stylesheet">  
</head>  
<style>  
    * {  
        margin: 0;  
        padding: 0;  
        box-sizing: border-box;  
        font-family: 'poppins', sans-serif;  
    }  
    body {  
        color: #eddede;  
    }  
.header {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    padding: 20px 10%;  
    background: transparent;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    z-index: 100;  
}  
.navbar a {  
    display: inline-block;  
    font-size: 18px;  
    color: #fff;  
    text-decoration: none;  
    font-weight: 500;  
    margin-left: 25px;  
    transition: .3s;  
}  
.navbar a:hover {  
    color: #8ef;  
}  
.home {  
    position: relative;  
    width: 100%;  
    justify-content: space-between;  
    height: 100vh;  
    background-size: cover;  
    background-position: center;  
    display: flex;  
    align-items: center;  
    padding: 70px 10% 0;  
}  
.home-content {  
    max-width: 600px;  
    position: absolute;  
    top: 50%;  
    left: 10%;  
    transform: translateY(-50%);  
    text-align: left;  
}  
.home-content h3 {  
    font-size: 24px;  
    font-weight: 700;  
    line-height: 1.4;  
    text-align: justify;  
    opacity: 0;  
    animation: slideBottom 1s ease forwards;  
    animation-delay: .4s;  
}  
.home-content p {  
    font-size: 18px;  
    line-height: 1.6;  
    text-align: justify;  
    opacity: 0;  
    animation: slideTop 1s ease forwards;  
    animation-delay: .4s;  
}  
.home-sci {  
    position: absolute;  
    bottom: 50px;  
    left: 50%;  
    transform: translateX(-50%);  
}  
.home-sci a {  
    display: inline-flex;  
    justify-content: center;  
    align-items: center;  
    width: 40px;  
    height: 40px;  
    background: transparent;  
    border: 2px solid rgb(255, 176, 176);  
    border-radius: 50%;  
    font-size: 20px;  
    color: rgb(255, 176, 176);
```

```

text-decoration: none;
margin: 0 15px;
}
.home-sci a:hover {
background: rgb(120, 0, 0);
color: rgb(255, 176, 176);
box-shadow: 0 0 20px rgb(255, 255, 255);
}
.logo {
position: absolute;
top: 10px;
left: 10px;
z-index: 1000;
max-width: 150px;
}
.container {
width: 100%;
height: 100vh;
position: fixed;
top: 0;
left: 0;
z-index: -1;
}
.background-clip {
width: 100%;
height: 100%;
object-fit: cover;
}
.sargam {
color: #f62020;
font-family: cursive;
}
.sphere {
color: #f4d868;
font-family: cursive;
}
@keyframes slideTop{
0%{
transform: translateY(100px);
opacity: 0;
}
100%{
transform: translateY(0px);
opacity: 1;
}
}
@keyframes slideBottom{
0%{
transform: translateY(-100px);
opacity: 0;
}
100%{
transform: translateY(0px);
opacity: 1;
}
}

```

```

}
.about {
display: grid;
grid-template-columns: auto 1fr;
align-items: center; /* Align items vertically */
gap: 1.5rem;
}
.about-img {
display: flex;
align-items: center; }
.about-text {
margin-left: 20px;
font-style: italic; }
.about-text h2 {
font-size: 20px;
align-items: left;
margin-right: 750px;
text-align: justify;
margin-bottom: 10px;
}
.about-sci {
margin-top: 10px;
}
.about-sci a {
display: inline-flex;
justify-content: center;
align-items: center;
width: 40px;
height: 40px;
background: transparent;
border: 2px solid rgb(255, 176, 176);
border-radius: 50%;
font-size: 20px;
color: rgb(255, 176, 176);
text-decoration: none;
margin-right: 15px;
}
.about-sci a:hover {
background: rgb(120, 0, 0);
color: rgb(255, 176, 176);
box-shadow: 0 0 20px rgb(255, 255, 255);
}
::-webkit-scrollbar {
width: 8px; /* Width of the scrollbar */
}
::-webkit-scrollbar-track {
background: #1f1f1f;
}
::-webkit-scrollbar-thumb {
background-color: #990000;
}
::-webkit-scrollbar-thumb:hover {
background-color: #770000; /* Darker red

```

```

color on hover */
}

</style>
<body>
  <div class="logo">
    
  </div>
  <div class="container">
    <video autoplay loop muted plays-inline
  class="background-clip">
      <source src="{% static '/video/5.mp4' %}"
  type="video/mp4">
    </video>
  </div>
  <section class="home">
    <div class="home-content">
      <h3>About <span
  class="sargam">Sargam</span><span
  class="sphere">Sphere</span></h3>
      <br>
      <p>Welcome to <span
  class="sargam">Sargam</span><span
  class="sphere">Sphere</span>, where music meets
  innovation
      and creativity intertwines with technology.
      We are more than just a music streaming
  platform; we are a community of music enthusiasts,
  creators,
      and listeners united by our love for melodies,
  rhythms, and harmonies from around the world.
      At <span
  class="sargam">Sargam</span><span
  class="sphere">Sphere</span>, we believe that music
  has the power
      to transcend boundaries, evoke
  emotions, and bring people together like nothing else
  can.
      Our mission is to provide a seamless and
  immersive musical experience that caters to
  the diverse tastes and preferences of our global
  audience. </p>
      <br>
      <h3>Our Journey</h3>
      <br>
      <p><span class="sargam">Sargam</span><span
  class="sphere">Sphere</span> was born out of a
  shared passion for music
      and a desire to revolutionize the way people
  discover,
      enjoy, and engage with their favorite tunes.
  Founded in 2024, our journey began with a vision to
      create a platform that not only offers
  unlimited access to a vast library of songs but also
  fosters a

```

sense of belonging and connection among
 music lovers worldwide.</p>

</div>
 <div class="home-sci">
 <i class='bx bxl-facebook'></i>
 <i class='bx bxl-instagram'></i>
 <i class='bx bxl-whatsapp'></i>
 </div>
 </section>
 <section class="about" id="about">
 <div class="about-img">

 </div>
 <div class="about-text">
 <h2>Passionate about blending design and
 functionality to resonate
 with every beat and melody. Let's build an
 immersive sonic oasis together</h2>
 <p>-- Sangram Banerjee

 Director
 </p>
 <div class="about-sci">
 <a
 href="https://www.facebook.com/profile.php?id=100
 077604998446"><i class='bx bxl-facebook'></i>
 <a
 href="https://www.instagram.com/nocturne_misanth
 ropia/"><i class='bx bxl-instagram'></i>
 </div>
 <div class="about-img">

 </div>
 <div class="about-text">
 <h2>Architecting and sculpting robust backend
 infrastructures to orchestrate smooth,
 scalable music streaming services & lot more
 , ensuring every line of code hits the right note</h2>
 <p>-- Sukanya Das

 Developer-Producer
 </p>
 <div class="about-sci">
 <a
 href="https://www.facebook.com/profile.php?id=100
 078401021613"><i class='bx bxl-facebook'></i>
 <a

```

https://www.instagram.com/sukanya\_\_d/<i class='bx bxl-instagram'></i></a>
</div>
</div>
<div class="about-img">

</div>
<div class="about-text">


## Crafting captivating interfaces that are aesthetically pleasing and harmonizes usability, ensuring every pixel sings in perfect rhythm with the user's experience.



-- <span class="sargam">Priyan</span><span class="sphere">Mukherjee</span>



<br> Artist


</div>
<div class="about-sci">
<i class='bx bxl-facebook'></i>
<i class='bx bxl-instagram'></i>
</div>
</div>

```

```

<div class="about-img">

</div>
<div class="about-text">


## Curating a symphony of engaging content to enrich your music streaming journey, harmonizing diverse voices and genres into a seamless listening experience.



-- <span class="sargam">Reetom</span><span class="sphere">Chakraborty</span>



<br> Composer


</p>
<div class="about-sci">
<i class='bx bxl-facebook'></i>
https://www.instagram.com/reetom\_rc7/
</div>
</div>
</body>
</html>

```

- Rockballads Music Streaming Webpage Output:

Songs

Bohemian Rhapsody	Play Now	
Hotel California	Play Now	
Nothing Else Matters	Play Now	
November Rain	Play Now	
Stairway to Heaven	Play Now	
Wish you were here	Play Now	

Experience the journey through these soul stirring melodies that paved the path of what we today know as Rock n Roll.

Bohemian Rhapsody 00:00:00:00

- Rockballads Music Streaming Webpage Source Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> SargamSphere</title>
<link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
<link rel="stylesheet" href="{% static 'css/rockballad.css' %}">
</head>
<body>
<header>
<h1 class="header-font">Rock Ballads</h1>
</header>
<div class="container flex bg-black" id="header-container">
<div class="left">
<div class="home bg-grey rounded m-1 p-1">
<div class="logo"></div>
<ul>
<li><a href="index" class="home-link">Home</a></li>
</ul>
</div>
<div class="slideshow-container">
<div class="slider-wrapper">
<div class="slider">

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

```

```

</div>
</div>
<div class="right">
<h2 class="header-font2">
Experience the journey through these soul stirring melodies that paved the path of what we today know as Rock n Roll.
</h2>
</div>
</div>
<div class="container">
<h2 class="playlist-header">Songs</h2>
<div class="songList">
<ul>
</ul>
</div>
</div>
<!-- Seekbar container -->
<div class="playBar">
<div class="seekbar">
<div class="circle"></div>
</div>
<div class="aboveBar">
<div class="songInfo"></div>
<div class="songButtons">



</div>
<div class="songTime"></div>
</div>
</div>
<script src="{% static "js/rocksongs.js" %}"></script>
</body>
</html>

```

- Rockballads Music Streaming Webpage JS Implementation:

```

document.addEventListener("DOMContentLoaded",
function() {
  console.log("lets write javascript");
  let currentSong = new Audio();
  let songs;
  function formatTime(seconds) {

```

```

    seconds = Math.round(seconds);
    const minutes = Math.floor(seconds / 60);
    const remainingSeconds = seconds % 60;
    const formattedMinutes =
      String(minutes).padStart(2, '0');
    const formattedSeconds =

```

```

String(remainingSeconds).padStart(2, '0');
    return
`${formattedMinutes}:${formattedSeconds}`;
}
// Async function to fetch and display songs
async function getSongs() {
try {
    const response = await fetch('/rocksongs');
    const data = await response.json();
    console.log("List of songs:", data.songs);
    return data.songs;
} catch (error) {
    console.error('Error fetching songs:', error);
}
}

const playMusic = (track, pause = false) => {
    const audioSrc = track.endsWith(".mp3") ? track :
track + ".mp3";
    currentSong.src = "static/rocksongs/" + audioSrc;
    if (!pause) {
        currentSong.play();
        play.src = "static/img/pause-circle.svg";
    }
    document.querySelector(".songInfo").innerHTML =
track.replaceAll('.mp3','');
    document.querySelector(".songTime").innerHTML =
L = "00:00/00:00";
}
// Main function
async function main() {
    songs = await getSongs();
    playMusic(songs[0], true);
    let songUL =
document.querySelector(".songList").getElementsByT
agName("ul")[0];
    for (const song of songs) {
        songUL.innerHTML += `<li>
            
            <div class="info">
                <div>${song.replaceAll('.mp3','')}</div>
            </div>
            <div class="playNow">
                <span>Play Now</span>
                
            </div>
        </li>`;
    }
    Array.from(document.querySelector(".songList").getElementsByTagN
ame("li")).forEach(e => {
        e.addEventListener("click", element => {
            console.log(e.querySelector(".info").firstEle
mentChild.innerHTML)
            playMusic(e.querySelector(".info").firstEle

```

```

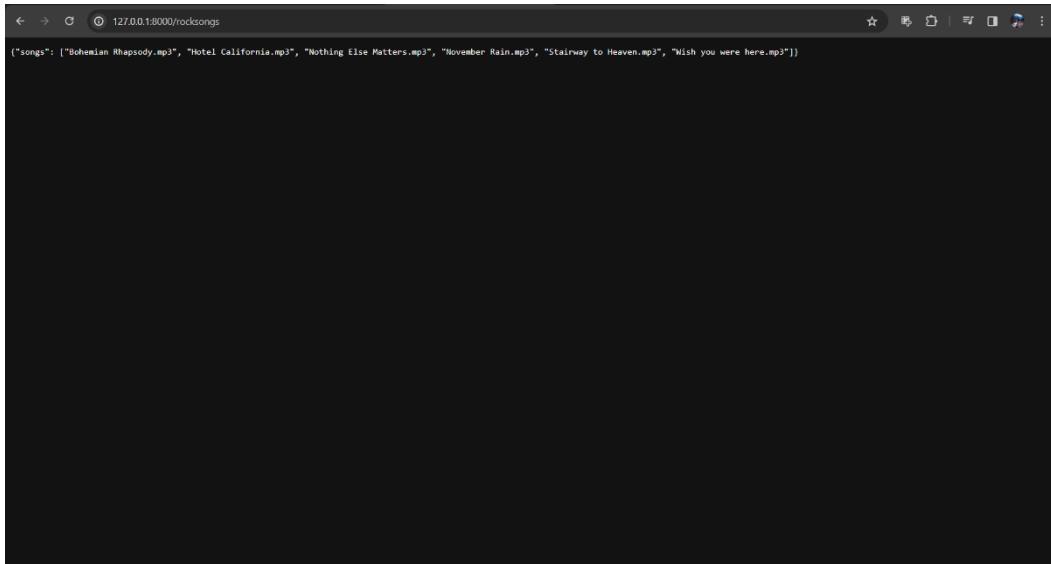
ntChild.innerHTML.trim()
        });
    });
    document.getElementById("play").addEventListener("click", () => {
        if (currentSong.paused) {
            currentSong.play();
            play.src = "static/img/pause-circle.svg";
        } else {
            currentSong.pause();
            play.src = "static/img/play-circle.svg";
        }
    });
    currentSong.addEventListener("timeupdate", () => {
        const currentTimeFormatted =
formatTime(currentSong.currentTime);
        const durationFormatted =
formatTime(currentSong.duration);
        document.querySelector(".songTime").innerHTML =
ML = `${currentTimeFormatted} /
${durationFormatted}`;
        document.querySelector(".circle").style.left =
((currentSong.currentTime) / (currentSong.duration))
* 100 + "%";
    });
    document.querySelector(".seekbar").addEventListener("click", e => {
        const progress = (e.offsetX /
e.target.getBoundingClientRect().width) * 100;
        document.querySelector(".circle").style.left =
progress + "%";
        currentSong.currentTime =
(currentSong.duration * progress) / 100;
    });
    let currentSongIndex = 0;
    document.getElementById("previous").addEventListener("click", () => {
        currentSongIndex--;
        if (currentSongIndex < 0) {
            currentSongIndex = songs.length - 1;
        }
        playMusic(songs[currentSongIndex]);
    });
    document.getElementById("forward").addEventListener("click", () => {
        currentSongIndex++;
        if (currentSongIndex >= songs.length) {
            currentSongIndex = 0;
        }
        playMusic(songs[currentSongIndex]);
    });
}
main();
});

```

• RockSongs Music Upload Source-Code & Output:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Song List</title>
</head>

<body>
  <h1>Songs List</h1>
  <ul>
    {% for song in songs %}
      <li>{{ song }}</li>
    {% endfor %}
  </ul>
</body>
</html>
```



• RabindraSangeet Music Streaming Webpage Output:

RABINDRASANGEET

Home

Songs

ଶ୍ରୀ ଆକାଶ ଜୁଡ଼େ ଶୁନିନ୍ତି ଓଇ ବାଜେ	Play Now	(D)
ଶ୍ରୀ ଆମରା ନୃତ୍ୟ ସୌବନ୍ଧେର ଦୂତ	Play Now	(D)
ଶ୍ରୀ ଆମାର ଏହି ପଥ ଚାତ୍ରୀତେଇ ଆନନ୍ଦ	Play Now	(D)
ଶ୍ରୀ ଆମାର ମୁଞ୍ଜି ଆଲୋଯ ଆଲୋଯ	Play Now	(D)
ଶ୍ରୀ ଆମି ମାରେର ସାଗର ପାଡ଼ି ଦେବ	Play Now	(D)

Experience the soul touching songs composed by Rabindranath Tagore

ଆକାଶ ଜୁଡ଼େ ଶୁନିନ୍ତି ଓଇ ବାଜେ

00:00/00:00

● RabindraSangeet Music Streaming Webpage Source Code:

```

{%- load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> SargamSphere</title>
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
    <link rel="stylesheet" href="{% static "css/rabindrasangeet.css" %}">
</head>
<body>
    <header>
        <h1 class="header-font">Rabindrasangeet</h1>
    </header>
    <div class="container flex bg-black" id="header-container">
        <div class="left">
            <div class="home bg-grey rounded m-1 p-1">
                <div class="logo"></div>
                <ul>
                    <li><a href="index" class="home-link">Home</a></li>
                </ul>
            </div>
            <div class="slideshow-container">
                <div class="slider-wrapper">
                    <div class="slider">
                        
                    </div>
                </div>
            </div>
        </div>
        <div class="right">
            <h2 class="header-font2"> Experience the soul touching songs composed by Rabindranath Tagore </h2>
            </div>
            <div class="container">
                <h2 class="playlist-header">Songs</h2>
                <div class="songList">
                    <ul>
                </ul>
            </div>
            </div>
            <!-- Seekbar container -->
            <div class="playBar">
                <div class="seekbar">
                    <div class="circle"></div>
                </div>
                <div class="aboveBar">
                    <div class="songInfo"></div>
                    <div class="songButtons">
                        
                        
                        
                    </div>
                    <div class="songTime"></div>
                </div>
                <script src="{% static "js/rabindrasangeet.js" %}"></script>
            </body>
        </html>
    </div>

```

● RabindraSangeet Music Streaming Webpage JS Implementation:

```

document.addEventListener("DOMContentLoaded",
function() {
    console.log("lets write javascript");
    let currentSong = new Audio();
    let songs;
    function formatTime(seconds) {
        seconds = Math.round(seconds);

```

```

        const minutes = Math.floor(seconds / 60);
        const remainingSeconds = seconds % 60;
        const formattedMinutes =
            String(minutes).padStart(2, '0');
        const formattedSeconds =
            String(remainingSeconds).padStart(2, '0');
        return
    }
}
)

```

```

` ${formattedMinutes}:${formattedSeconds}`;
}

// Async function to fetch and display songs
async function getSongs() {
  try {
    const response = await fetch('/rabindrasongs');
    const data = await response.json();
    console.log("List of songs:", data.songs);
    return data.songs;
  } catch (error) {
    console.error('Error fetching songs:', error);
  }
}

const playMusic = (track, pause = false) => {
  const audioSrc = track.endsWith(".mp3") ? track : track + ".mp3";
  currentSong.src = "static/rabindrasangeet/" + audioSrc;
  if (!pause) {
    currentSong.play();
    play.src = "static/img/pause-circle.svg";
  }
  document.querySelector(".songInfo").innerHTML =
    track.replaceAll('.mp3','');
  document.querySelector(".songTime").innerHTML =
    "00:00/00:00";
}

// Main function
async function main() {
  songs = await getSongs();
  playMusic(songs[0], true);
  let songUL =
    document.querySelector(".songList").getElementsByTagNames("ul")[0];
  for (const song of songs) {
    songUL.innerHTML += `<li>
      
      <div class="info">
        <div>${song.replaceAll('.mp3','')}</div>
      </div>
      <div class="playNow">
        <span>Play Now</span>
        
      </div>
    </li>`;
  }
  Array.from(document.querySelector(".songList").getElementsByTagNames("li")).forEach(e => {
    e.addEventListener("click", element => {
      console.log(e.querySelector(".info").firstElementChild.innerHTML)
      playMusic(e.querySelector(".info").firstElementChild.innerHTML.trim())
    });
  });
}

document.getElementById("play").addEventListener("click", () => {
  if (currentSong.paused) {
    currentSong.play();
    play.src = "static/img/pause-circle.svg";
  } else {
    currentSong.pause();
    play.src = "static/img/play-circle.svg";
  }
});

currentSong.addEventListener("timeupdate", () => {
  const currentTimeFormatted =
    formatTime(currentSong.currentTime);
  const durationFormatted =
    formatTime(currentSong.duration);
  document.querySelector(".songTime").innerHTML =
    `${currentTimeFormatted} / ${durationFormatted}`;
  document.querySelector(".circle").style.left =
    ((currentSong.currentTime) / (currentSong.duration)) *
    100 + "%";
});

document.querySelector(".seekbar").addEventListener("click", e => {
  const progress = (e.offsetX /
    e.target.getBoundingClientRect().width) * 100;
  document.querySelector(".circle").style.left =
    progress + "%";
  currentSong.currentTime =
    (currentSong.duration * progress) / 100;
});

let currentSongIndex = 0;
document.getElementById("previous").addEventListener("click", () => {
  currentSongIndex--;
  if (currentSongIndex < 0) {
    currentSongIndex = songs.length - 1;
  }
  playMusic(songs[currentSongIndex]);
});

document.getElementById("forward").addEventListener("click", () => {
  currentSongIndex++;
  if (currentSongIndex >= songs.length) {
    currentSongIndex = 0;
  }
  playMusic(songs[currentSongIndex]);
});

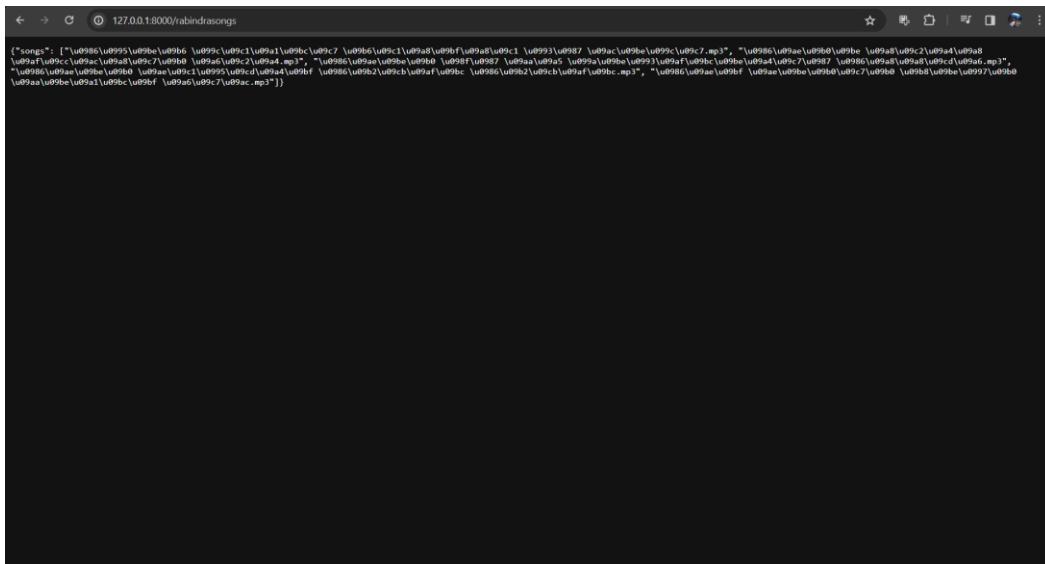
main();
}

```

- RabindraSangeet Music Upload Source-Code & Output:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="% static
'img/logo/favicon.ico' %">
    <title>Song List</title>
</head>

<body>
    <h1>Songs List</h1>
    <ul>
        {% for song in songs %}
            <li>{{ song }}</li>
        {% endfor %}
    </ul>
</body>
</html>
```



- NazrulGeeti Music Streaming Webpage Output:

The screenshot shows a web browser window with the URL 127.0.0.1:8000/nazrulgeeti. The page has a dark background. On the left, there's a large black and white portrait of Bidrohi Kobi Nazrul Islam wearing glasses and a cap. At the top center, the word "NAZRULGEETI" is written in a stylized, handwritten font. Below the portrait, there's a small logo for "Songam Sphene". In the center, there's a navigation bar with a "Home" button. To the right of the portrait, there's a section titled "Songs" with a list of five songs, each with a "Play Now" button. The songs listed are: কাবেরী নদী জলে কে গো বালিকা, দূর দ্বীপবাসিনী চিনি তোমারে, বাগিচায় বুলবুলি তুই, মোমের পুতুল, and শুকনো পাতার নৃপুর পারে. On the far right, there's a promotional message: "Take a tour of the emotional side of Bidrohi Kobi Nazrul Islam". At the bottom, there's a green footer bar with playback controls (play, pause, stop) and a timer showing "00:00/00:00". The footer also includes the text "কাবেরী নদী জলে কে গো বালিকা" and "100 MSP".

- NazrulGeeti Music Streaming Webpage Source Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> SargamSphere</title>
<link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
<link rel="stylesheet" href="{% static 'css/nazrulgeeti.css' %}">
</head>
<body>
<header>
<h1 class="header-font">NazrulGeeti</h1>
</header>
<div class="container flex bg-black" id="header-container">
<div class="left">
<div class="home bg-grey rounded m-1 p-1">
<div class="logo"></div>
<ul>
<li><a href="index" class="home-link">Home</a></li>
</ul>
</div>
<div class="slideshow-container">
<div class="slider-wrapper">
<div class="slider">

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

```

```

</div>
<div class="right">
<h2 class="header-font2">
Take a tour of the emotional side of Bidrohi Kobi Nazrul Islam
</h2>
</div>
</div>

<div class="container">
<h2 class="playlist-header">Songs</h2>
<div class="songList">
<ul>

</ul>
</div>
</div>
<!-- Seekbar container -->
<div class="playBar">
<div class="seekbar">
<div class="circle"></div>
</div>
<div class="aboveBar">
<div class="songInfo"></div>
<div class="songButtons">



</div>
<div class="songTime"></div>
</div>
<script src="{% static 'js/nazrulgeeti.js' %}"></script>
</body>
</html>

```

- NazrulGeeti Music Streaming Webpage JS Implementation:

```

document.addEventListener("DOMContentLoaded",
function() {
  console.log("lets write javascript");
  let currentSong = new Audio();
  let songs;
  function formatTime(seconds) {
    seconds = Math.round(seconds);

```

```

    const minutes = Math.floor(seconds / 60);
    const remainingSeconds = seconds % 60;
    const formattedMinutes =
String(minutes).padStart(2, '0');
    const formattedSeconds =
String(remainingSeconds).padStart(2, '0');
    return

```

```

` ${formattedMinutes}:${formattedSeconds}`;
}

// Async function to fetch and display songs
async function getSongs() {
  try {
    const response = await fetch('/nazrulsongs');
    const data = await response.json();
    console.log("List of songs:", data.songs);
    return data.songs;
  } catch (error) {
    console.error('Error fetching songs:', error);
  }
}

const playMusic = (track, pause = false) => {
  const audioSrc = track.endsWith(".mp3") ? track :
  track + ".mp3";
  currentSong.src = "static/nazrulgeeti/" +
  audioSrc;

  if (!pause) {
    currentSong.play();
    play.src = "static/img/pause-circle.svg";
  }
  document.querySelector(".songInfo").innerHTML =
  track.replaceAll('.mp3','');
  document.querySelector(".songTime").innerHTML =
  L = "00:00/00:00";
}

// Main function
async function main() {
  songs = await getSongs();
  playMusic(songs[0], true);
  let songUL =
  document.querySelector(".songList").getElementsByT
  agName("ul")[0];
  for (const song of songs) {
    songUL.innerHTML += `<li>
      
      <div class="info">
        <div>${song.replaceAll('.mp3','')}</div>
      </div>
      <div class="playNow">
        <span>Play Now</span>
        
      </div>
    </li>`;
  }
  Array.from(document.querySelector(".songList").getElementsByTagN
  ame("li")).forEach(e => {
    e.addEventListener("click", element => {
      console.log(e.querySelector(".info").firstEle
      ntChild.innerHTML)
    });
  });
}

```

```

  ntChild.innerHTML.trim()
  });
});

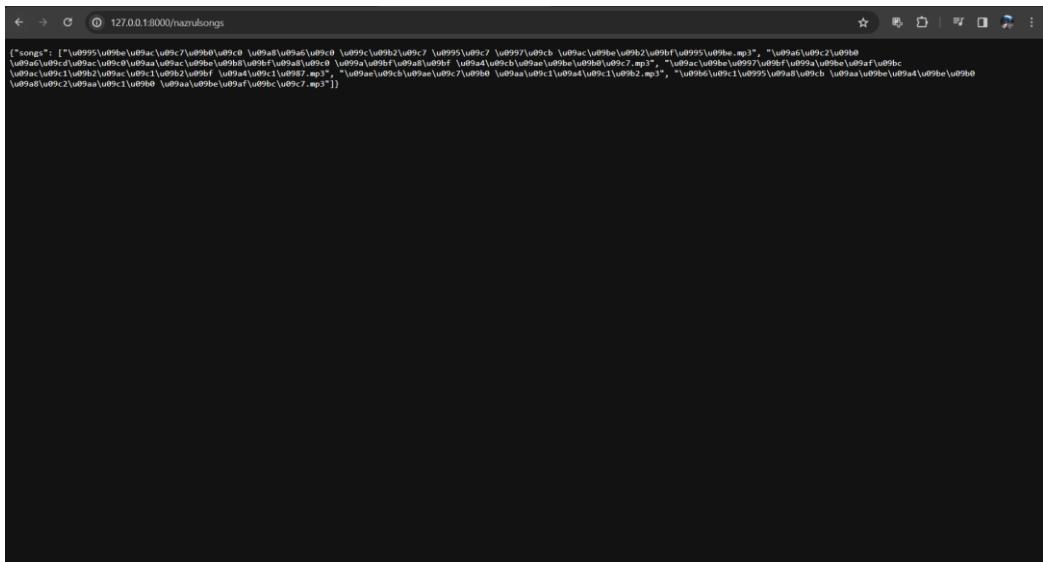
document.getElementById("play").addEventListener("click", () => {
  if (currentSong.paused) {
    currentSong.play();
    play.src = "static/img/pause-circle.svg";
  } else {
    currentSong.pause();
    play.src = "static/img/play-circle.svg";
  }
});
currentSong.addEventListener("timeupdate", () => {
  const currentTimeFormatted =
  formatTime(currentSong.currentTime);
  const durationFormatted =
  formatTime(currentSong.duration);
  document.querySelector(".songTime").innerHT
  ML = `${currentTimeFormatted} /
  ${durationFormatted}`;
  document.querySelector(".circle").style.left =
  ((currentSong.currentTime) / (currentSong.duration))
  * 100 + "%";
});
document.querySelector(".seekbar").addEventListener("click", e => {
  const progress = (e.offsetX /
  e.target.getBoundingClientRect().width) * 100;
  document.querySelector(".circle").style.left =
  progress + "%";
  currentSong.currentTime =
  (currentSong.duration * progress) / 100;
});
let currentSongIndex = 0;
document.getElementById("previous").addEventListener("click", () => {
  currentSongIndex--;
  if (currentSongIndex < 0) {
    currentSongIndex = songs.length - 1;
  }
  playMusic(songs[currentSongIndex]);
});
document.getElementById("forward").addEventListener("click", () => {
  currentSongIndex++;
  if (currentSongIndex >= songs.length) {
    currentSongIndex = 0;
  }
  playMusic(songs[currentSongIndex]);
});
main();
});

```

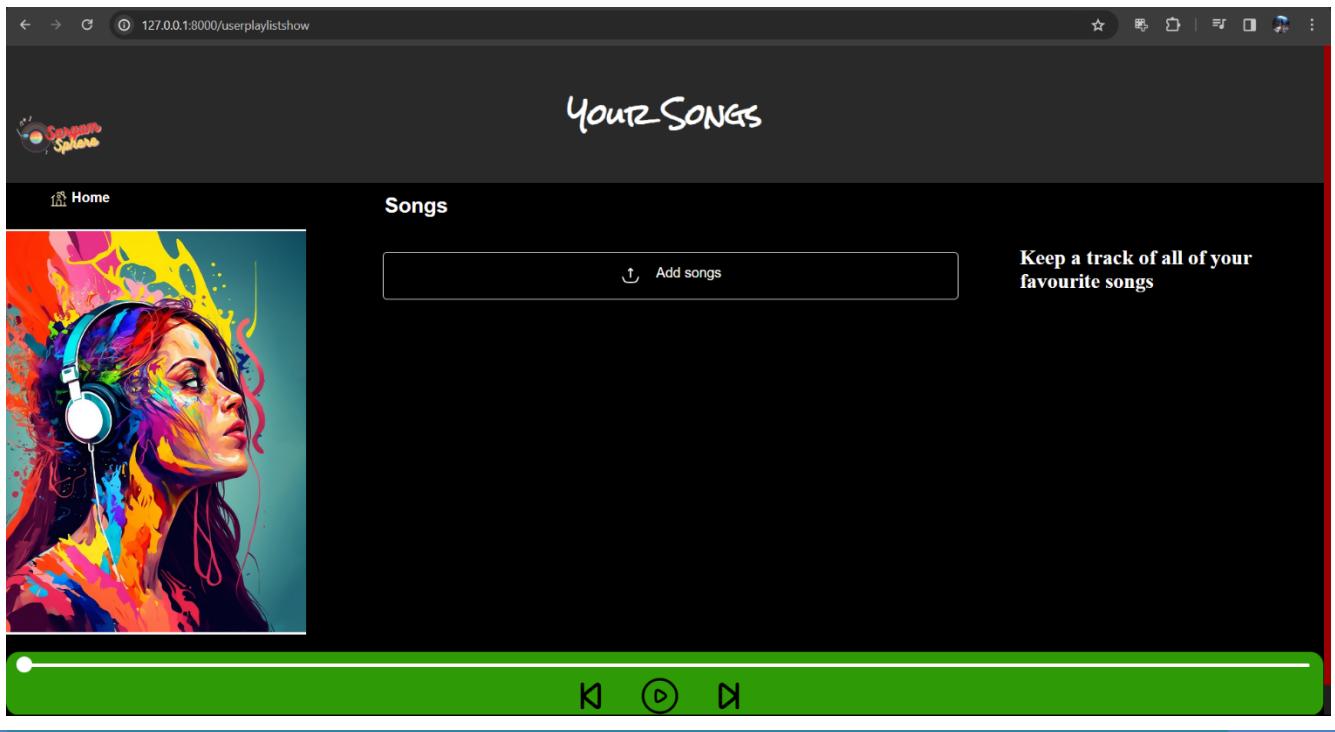
- NazrulGeeti Music Upload Source-Code & Output:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
    <title>Song List</title>
</head>
```

```
<body>
    <h1>Songs List</h1>
    <ul>
        {% for song in songs %}
            <li>{{ song }}</li>
        {% endfor %}
    </ul>
</body>
</html>
```



- User Playlist Webpage Output:



• User Playlist Webpage Source-Code:

```

{%- load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> SargamSphere</title>
<link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
<link rel="stylesheet" href="{% static "css/userplaylistshow.css" %}">
</head>
<body>
<header>
<h1 class="header-font">Your Songs</h1>
</header>
<div class="container flex bg-black" id="header-container">
<div class="left">
<div class="home bg-grey rounded m-1 p-1">
<div class="logo"></div>
<ul>
<li><a href="index" class="home-link">Home</a></li>
</ul>
</div>
<div class="slideshow-container">
<div class="slider-wrapper">
<div class="slider">

</div>
</div>
</div>
<div class="right">
<h2 class="header-font2">
Keep a track of all of your favourite songs
</h2>
</div>
</div>
<div class="container flex bg-black" id="content-container">
<div class="songList">
<ul>
</ul>
</div>
<div class="upload">
<ul>
<a href="userplaylistupload" class="playlistupload">
<li>
Add songs
</li>
</a>
</ul>
</div>
</div>
<!-- Seekbar container -->
<div class="playBar">
<div class="seekbar">
<div class="circle"></div>
</div>
<div class="aboveBar">
<div class="songInfo"></div>
<div class="songButtons">



</div>
<div class="songTime"></div>
</div>
</div>
<script src="{% static "js/userplaylistshow.js" %}"></script>
</body>
</html>

```

• User Playlist Webpage JS Implementation:

```

document.addEventListener("DOMContentLoaded",
function() {
  console.log("JavaScript loaded");
  const currentSong = new Audio();

```

```

let songs = [];
function formatTime(seconds) {
  seconds = Math.round(seconds);
  const minutes = Math.floor(seconds / 60);

```

```

const remainingSeconds = seconds % 60;
const formattedMinutes =
String(minutes).padStart(2, '0');
const formattedSeconds =
String(remainingSeconds).padStart(2, '0');
return
`${formattedMinutes}:${formattedSeconds}`;
}
async function getSongs() {
try {
const response = await
fetch('/userplaylistsongs');
const data = await response.json();
console.log("List of songs:", data.songs);
return data.songs;
} catch (error) {
console.error('Error fetching songs:', error);
return [];
}
}
const playMusic = (track, pause = false) => {
const audioSrc = track.endsWith(".mp3") ? track :
track + ".mp3";
currentSong.src = "static/usersongs/" + audioSrc;
if (!pause) {
currentSong.play();
document.getElementById("play").src =
"static/img/pause-circle.svg";
}
document.querySelector(".songInfo").innerHTML =
track.replaceAll('.mp3','');
document.querySelector(".songTime").innerHTML =
"00:00/00:00";
}
async function main() {
songs = await getSongs();
if (songs.length > 0) {
playMusic(songs[0], true);
}
const songList =
document.querySelector(".songList ul");
for (const song of songs) {
const songName = song.replaceAll('.mp3','');
const li = document.createElement("li");
li.innerHTML =

<div class="info">
<div>${songName}</div>
</div>
<div class="playNow">
<span>Play Now</span>

</div>
`;
```

songList.appendChild(li);

}

songList.addEventListener("click", (event) => {
const target = event.target.closest("li");
if (target) {
const infoElement =
target.querySelector(".info");
if (infoElement &&
infoElement.firstChild) {
const songName =
infoElement.firstChild.innerHTML.trim();
playMusic(songName);
}
}
});

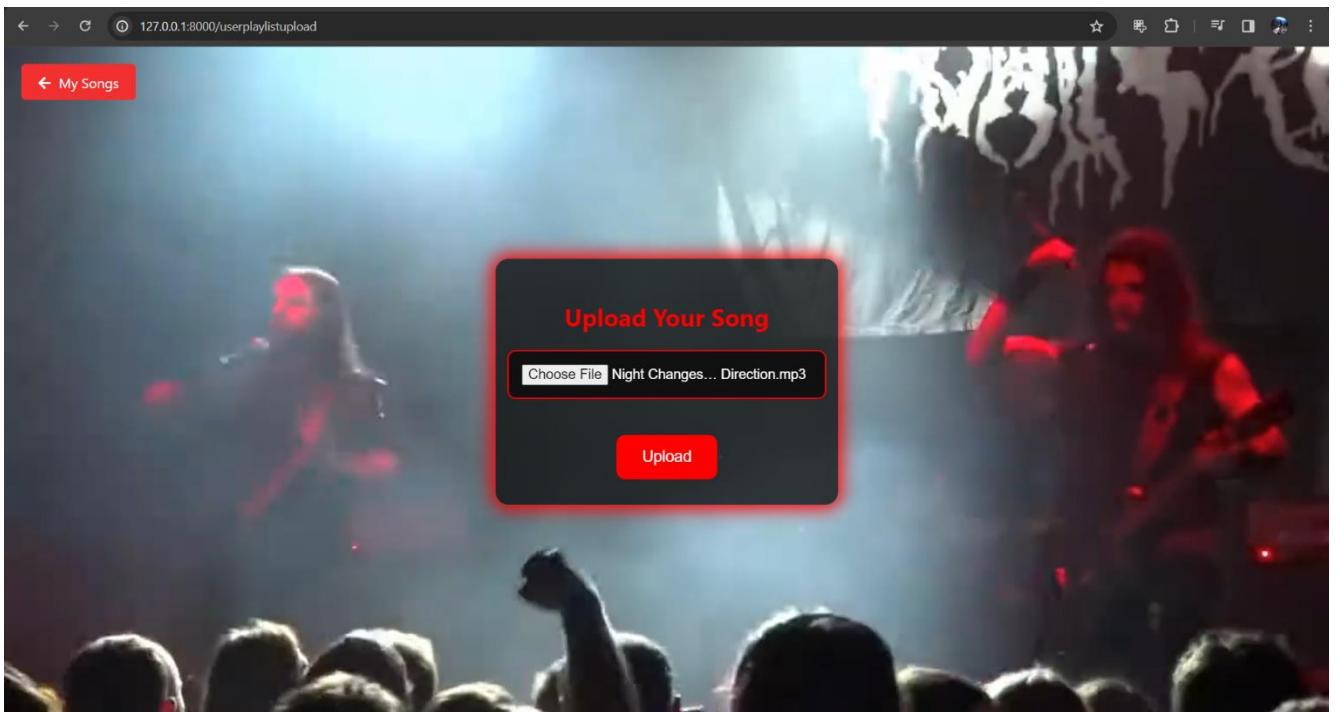
document.getElementById("play").addEventListener("click", () => {
if (currentSong.paused) {
currentSong.play();
document.getElementById("play").src =
"static/img/pause-circle.svg";
} else {
currentSong.pause();
document.getElementById("play").src =
"static/img/play-circle.svg";
}
});
currentSong.addEventListener("timeupdate", () => {
const currentTimeFormatted =
formatTime(currentSong.currentTime);
const durationFormatted =
formatTime(currentSong.duration);
document.querySelector(".songTime").innerHTML =
`\${currentTimeFormatted} /
\${durationFormatted}`;
document.querySelector(".circle").style.left =
((currentSong.currentTime) / (currentSong.duration)) *
100 + "%";
});
document.querySelector(".seekbar").addEventListener("click", e => {
const progress = (e.offsetX /
e.target.getBoundingClientRect().width) * 100;
document.querySelector(".circle").style.left =
progress + "%";
currentSong.currentTime =
(currentSong.duration * progress) / 100;
});
let currentSongIndex = 0;
document.getElementById("previous").addEventListener("click", () => {
currentSongIndex--;
if (currentSongIndex < 0) {

```

        currentSongIndex = songs.length - 1;
    }
    playMusic(songs[currentSongIndex]);
});
document.getElementById("forward").addEventListener("click", () => {
    currentSongIndex++;
    if (currentSongIndex >= songs.length) {
        currentSongIndex = 0;
        playMusic(songs[currentSongIndex]);
    }
    main();
});

```

- User Playlist Upload Webpage Output:



- User Playlist Upload Webpage Source-Code:

```

[% load static%]
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Playlist Upload</title>
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
<style>
    body {
        font-family: 'Segoe UI', Tahoma, Geneva,
        Verdana, sans-serif;
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        overflow: hidden;
        background: linear-gradient(to bottom, rgba(0, 0, 0, 0.8), rgba(0, 0, 0, 0.8)), url('background-video.mp4');
        background-size: cover;
        background-position: center;
        position: relative;
    }
    .container {
        background-color: rgba(17, 17, 17, 0.8);
        padding: 30px;
        border-radius: 15px;
        box-shadow: 0px 0px 20px 5px #FF0000;
    }

```

```

    text-align: center;
}
h1 {
    font-size: 28px;
    margin-bottom: 20px;
    color: #FF0000;
}
form {
    display: flex;
    flex-direction: column;
    align-items: center;
    cursor: pointer;
}
input[type="file"] {
    border: 2px solid #FF0000;
    border-radius: 10px;
    padding: 15px;
    width: 100%;
    margin-bottom: 20px;
    font-size: 16px;
    color: #FFFFFF;
    cursor: pointer;
    background-color: #111111;
    outline: none;
    transition: border-color 0.3s ease;
}
input[type="file"]:hover,
input[type="file"]:focus {
    border-color: #FF3333;
    cursor: pointer;
}
input[type="submit"] {
    background-color: #FF0000;
    color: #FFFFFF;
    border: none;
    border-radius: 10px;
    padding: 15px 30px;
    font-size: 18px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
    input[type="submit"]:hover {
        background-color: #FF3333;
    }
/* Add button styles */
.add-button {
    position: absolute;
    top: 20px;
    left: 20px;
    background-color: #F23030;
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    text-decoration: none;
    cursor: pointer;
    transition: background-color 0.3s ease;
}
    .add-button:hover {
        background-color: #FF5252; /* Lighter shade of red on hover */
    }
    .add-button:focus {
        outline: none; /* Remove outline on focus */
    }
    .add-button i {
        margin-right: 5px;
    }
    .background-clip{
        position: absolute;
        right: 0;
        bottom: 0;
        z-index:-1;
        transition: 1s;
    }
    @media (min-aspect-ratio: 16/9){
        .background-clip{
            width: 100%;
            height: auto;
        }
    }
    @media(max-aspect-ratio: 16/9){
        .background-clip{
            width: auto;
            height: 100%;
        }
    }

```

</style>

</head>

<body>

[My Songs](userplaylistshow)

<div class="container">

<source src="{% static '/video/2.mp4' %}">
 type="video/mp4"
 </video>
 <h1>Upload Your Song</h1>
 <form action="userplaylistupd" method="post" enctype="multipart/form-data">
 {% csrf_token %}
 <input type="file" name="audio_file" accept="audio/*">

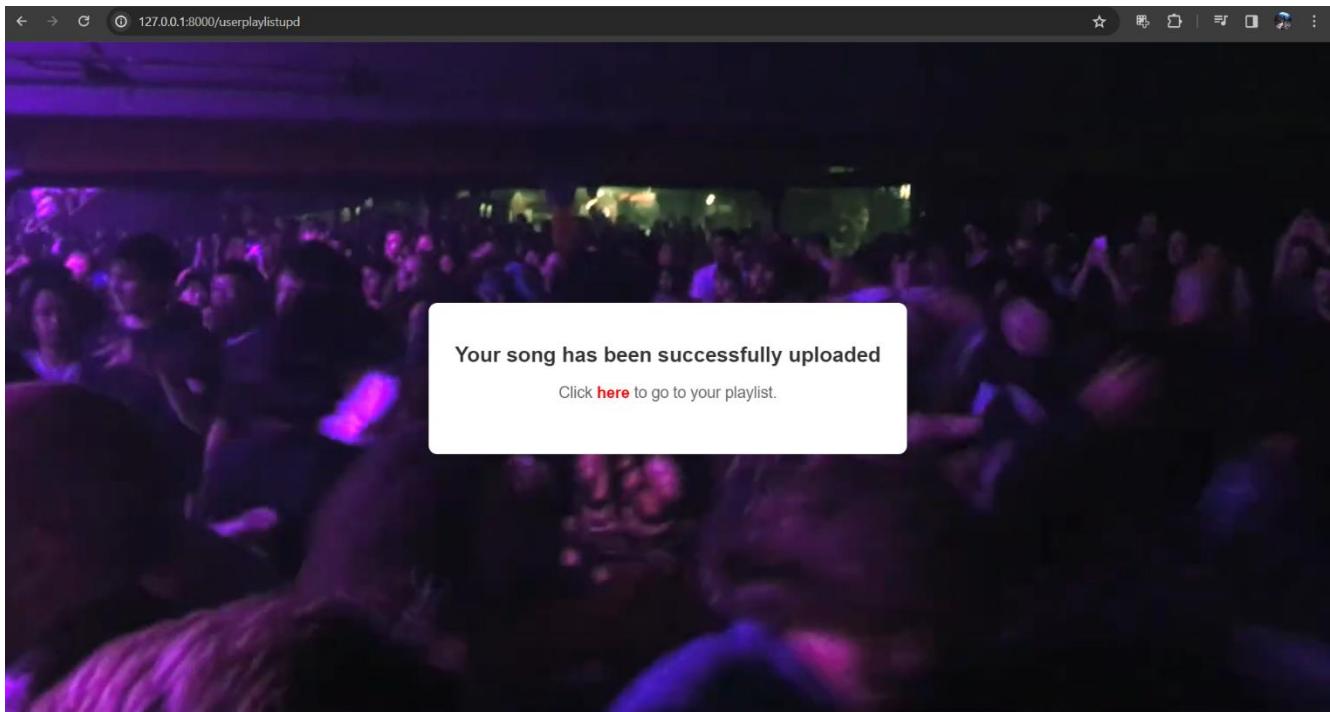
 <input type="submit" value="Upload">
 </form>

</div>

</body>

</html>

- User Playlist Uploaded Successfully Webpage Output:



- User Playlist Uploaded Successfully Webpage Source-Code:

```
% load static %
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
  <title>Upload Success</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .container {
    text-align: center;
    padding: 30px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
  }
  .background-clip {
    position: absolute;
    right: 0;
    bottom: 0;
    z-index:-1;
    transition: 1s;
  }
  @media (min-aspect-ratio: 16/9){
    .background-clip {
      width: 1000px;
      height: 562.5px;
    }
  }
</style>
```

```

.background-clip{
  width: 100%;
  height: auto;
}

@media(max-aspect-ratio: 16/9){
  .background-clip{
    width: auto;
    height: 100%;
  }
}

</style>
</head>
<body>

  <div class="container">
    <video autoplay loop muted plays-inline
    class="background-clip">
      <source src="{% static '/video/3.mp4' %}"
      type="video/mp4">
    </video>
    <h1>Your song has been successfully
    uploaded</h1>
    <p>Click <a href="userplaylistshow">here</a> to
    go to your playlist.</p>
  </div>
</body>
</html>

```

• User Playlist Upload Source-Code & Output:

```

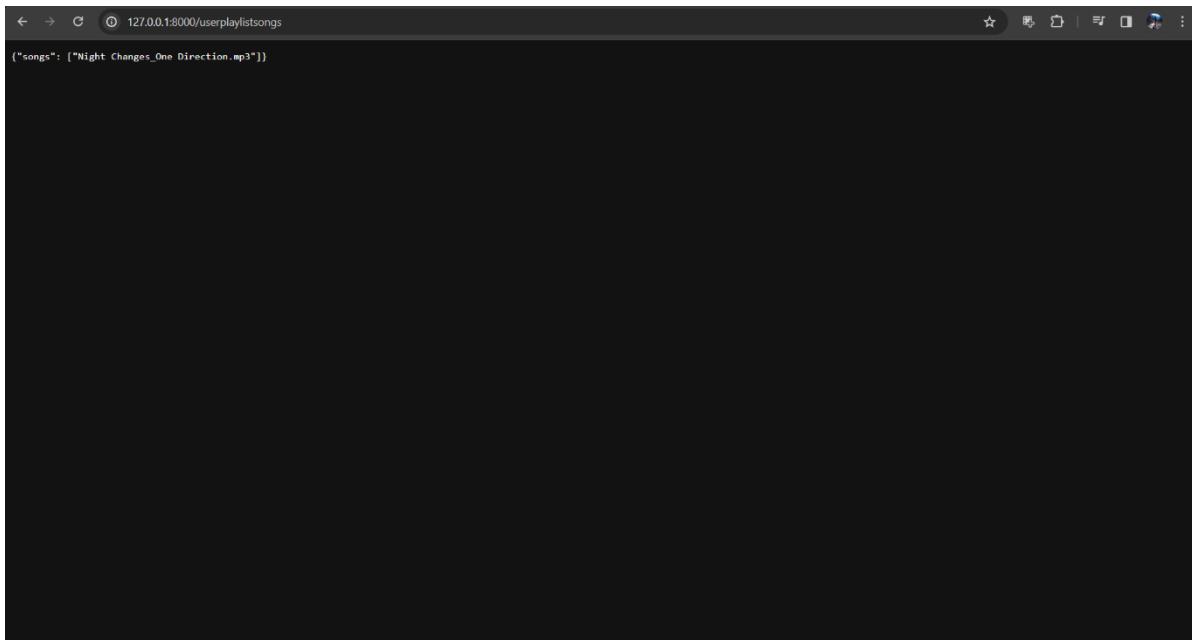
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>Song List</title>
</head>
<body>

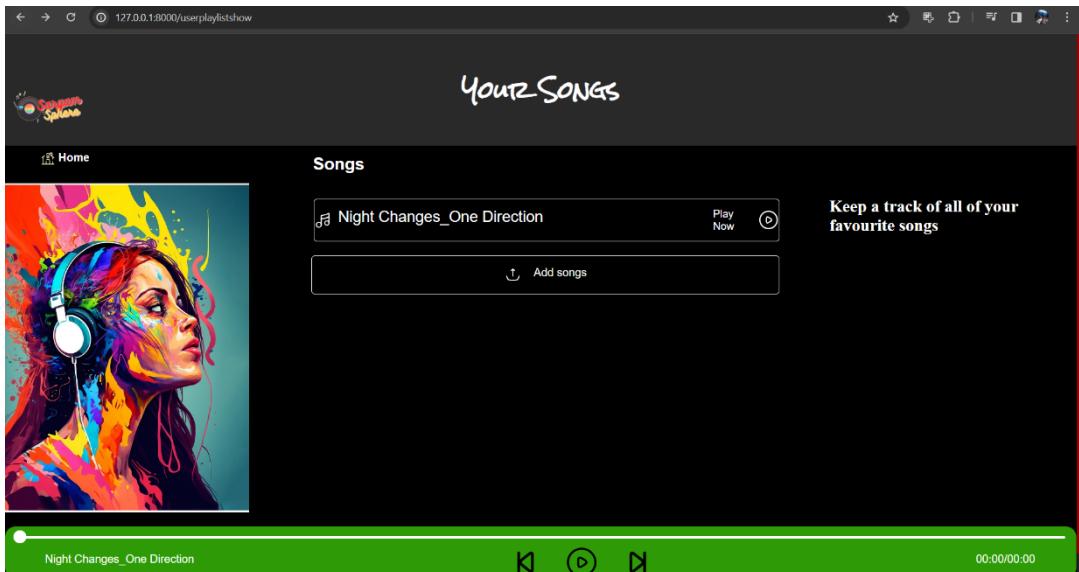
```

```

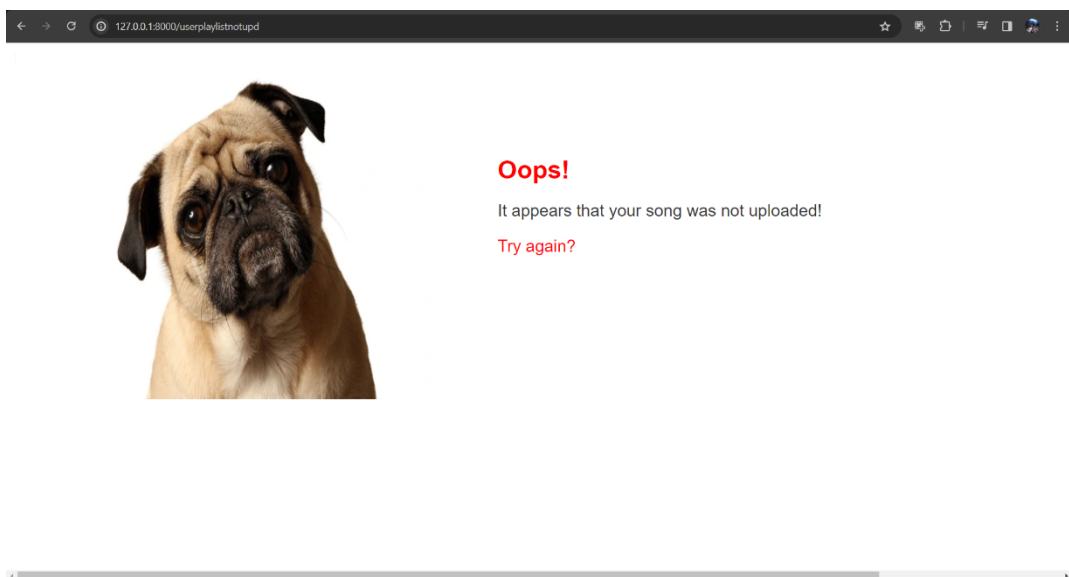
<h1>Songs List</h1>
<ul>
  {% for song in songs %}
    <li>{{ song }}</li>
  {% endfor %}
</ul>
</body>
</html>

```





- USER SONGS NOT uploaded SUCCESSFULLY page Output:



- USER SONGS NOT uploaded SUCCESSFULLY page Source-Code:

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
```

```
width, initial-scale=1.0">
    <link rel="shortcut icon" href="{% static
        'img/logo/favicon.ico' %}">
    <title>Oops!</title>
    <style>
        .left{
```

```

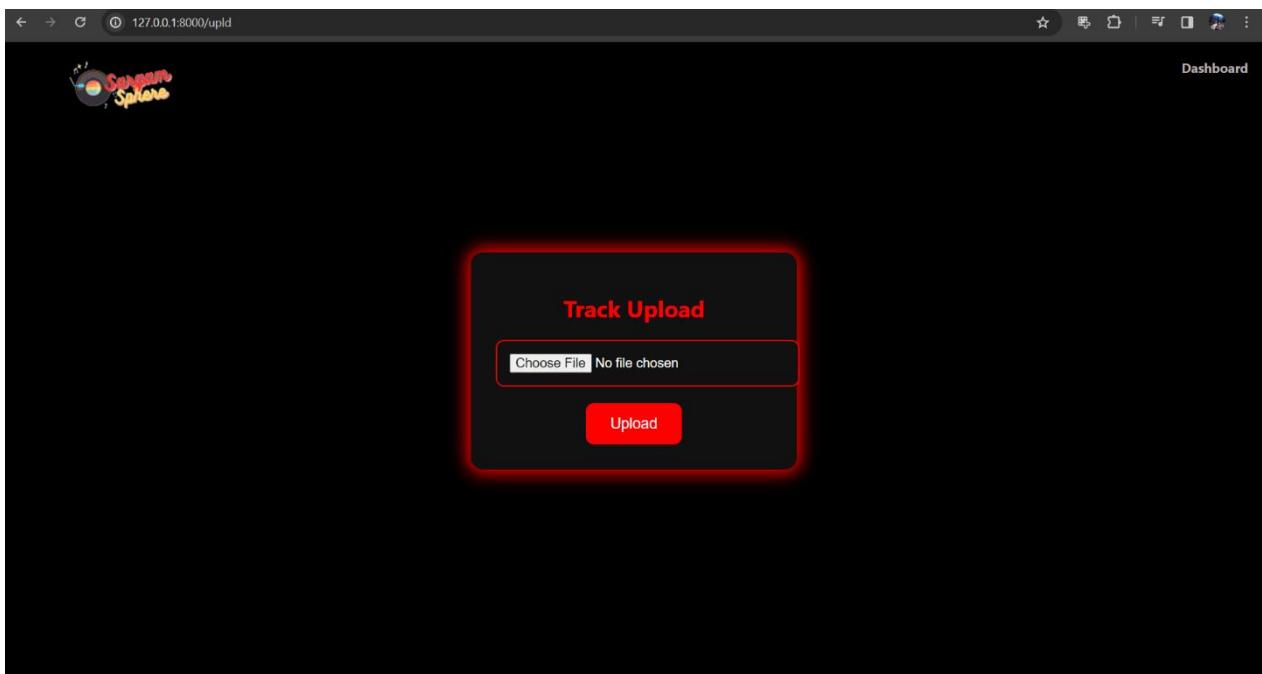
width: 25vw;
padding: 5px;
}
.left img {
height: 500px;
width: 600px;
}
.right{
width: 75vw;
position: absolute;
margin-left: 700px;
margin-top: 130px;
font-family: Arial, sans-serif; /* Change font
family */
}
.right h1 {
color: #FF0000; /* Change color of h1 */
font-size: 36px; /* Change font size of h1 */
font-weight: bold; /* Make h1 bold */
margin-bottom: 20px; /* Add some margin
below h1 */
}
.right p {
font-size: 24px; /* Change font size of
paragraphs */
color: #333; /* Change color of paragraphs */
margin-bottom: 10px; /* Add some margin
below paragraphs */
}

}
.right a {
font-size: 24px; /* Change font size of anchor */
}
color: #FF0000; /* Change color of anchor */
text-decoration: none; /* Remove underline
from anchor */
}
.right a:hover {
text-decoration: underline; /* Add underline on
hover */
}
</style>
</head>
<body>
<div class="right">
<h1>Oops!</h1>
<p>It appears that your song was not
uploaded!</p>
<p><a href="userplaylistupload">Try
again?</a></p>
</div>
<div class="left">

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

```

- Upload Songs from Admin Dashboard page Output:



- Upload Songs from Admin Dashboard page Source-Code:

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>File Upload</title>
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicon s.min.css' rel='stylesheet'>
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
<style>
    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        background-color: #000000;
        color: #FFFFFF;
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
    }
    .container {
        background-color: #111111;
        padding: 30px;
        border-radius: 15px;
        box-shadow: 0px 0px 20px 5px #FF0000;
    }
    h1 {
        font-size: 28px;
        text-align: center;
        margin-bottom: 20px;
        color: #FF0000;
    }
    form {
        text-align: center;
    }
    input[type="file"] {
        border: 2px solid #FF0000;
        border-radius: 10px;
        padding: 15px;
        width: 100%;
        margin-bottom: 20px;
        font-size: 16px;
        color: #FFFFFF;
    }

```

```

        background-color: #111111;
        outline: none;
        transition: border-color 0.3s ease;
    }
    input[type="file"]:hover,
    input[type="file"]:focus {
        border-color: #FF3333;
    }
    input[type="submit"] {
        background-color: #FF0000;
        color: #FFFFFF;
        border: none;
        border-radius: 10px;
        padding: 15px 30px;
        font-size: 18px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }
    input[type="submit"]:hover {
        background-color: #FF3333;
    }
    .nav-link {
        position: absolute;
        top: 20px;
        right: 20px;
        text-decoration: none;
        font-size: 16px;
        color: rgb(187, 175, 175); /* Setting color to grey */
        font-weight: bold; /* Making text bold */
        margin-left: 20px; /* Adding a gap between the links */
    }
    .nav-link:hover {
        color: rgb(212, 221, 43); /* Change color to red on hover */
    }
    .nav-link:active{
        color: red; /* Change color to red when clicked */
    }
    .logo {
        position: absolute;
        top: 20px;
        left: 80px;
        width: 100%;
        max-width: 130px;
    }
</style>

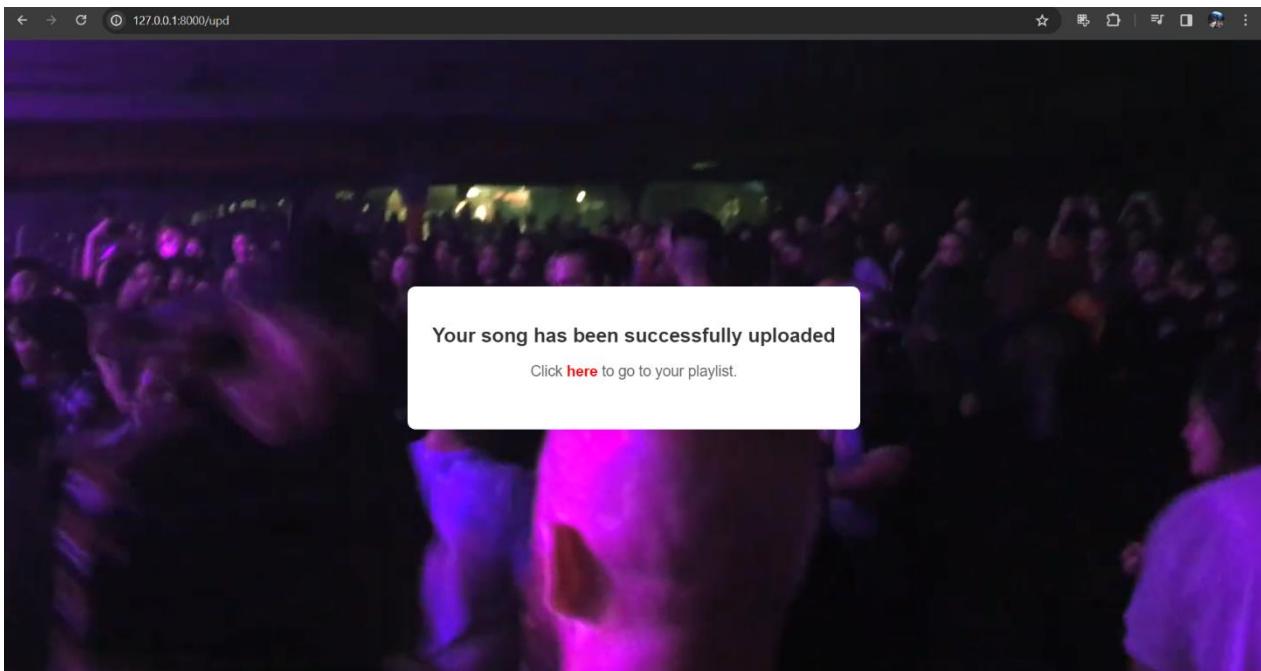
```

```

</head>
<body>
  <a href="adminpage" class="nav-link">Dashboard</a>
  <img class="logo" src='{% static "img/logo/logo.png" %}' alt="" />
  <div class="container">
    <h1>Track Upload</h1>
    <form action="upd" method="post" enctype="multipart/form-data">
      {% csrf_token %}
      <input type="file" name="audio_file" accept="audio/*">
      <br>
      <input type="submit" value="Upload">
    </form>
  </div>
</body>
</html>

```

- Uploaded Songs from Admin Dashboard is Successful page Output:



- Uploaded Songs from Admin Dashboard is Successful page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
  <title>Upload Success</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .container {
    text-align: center;
    padding: 30px;
    background-color: #fff;
    border-radius: 10px;
  }

```

```

        box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
    }
    h1 {
        color: #333;
        font-size: 24px;
        margin-bottom: 20px;
    }
    p {
        color: #666;
        font-size: 18px;
        margin-bottom: 30px;
    }
    a {
        text-decoration: none;
        color: #FF0000;
        font-weight: bold;
    }
    a:hover {
        text-decoration: underline;
    }
    .background-clip{
        position: absolute;
        right: 0;
        bottom: 0;
        z-index:-1;
        transition: 1s;
    }
    @media (min-aspect-ratio: 16/9){
        .background-clip{
            width: 100%;
            height: auto;
        }
        @media(max-aspect-ratio: 16/9){
            .background-clip{
                width: auto;
                height: 100%;
            }
        }
    
```

</style>

</head>

<body>

<div class="container">

<video autoplay loop muted plays-inline class="background-clip">

<source src="{% static '/video/3.mp4' %}" type="video/mp4">

</video>

<h1>Your song has been successfully uploaded</h1>

<p>Click here to go to your playlist.</p>

</div>

</body>

</html>

- Uploaded Songs from Admin Dashboard is NOT Successful page Output:



Oops!

It appears that your song was not uploaded!

[Try again?](#)

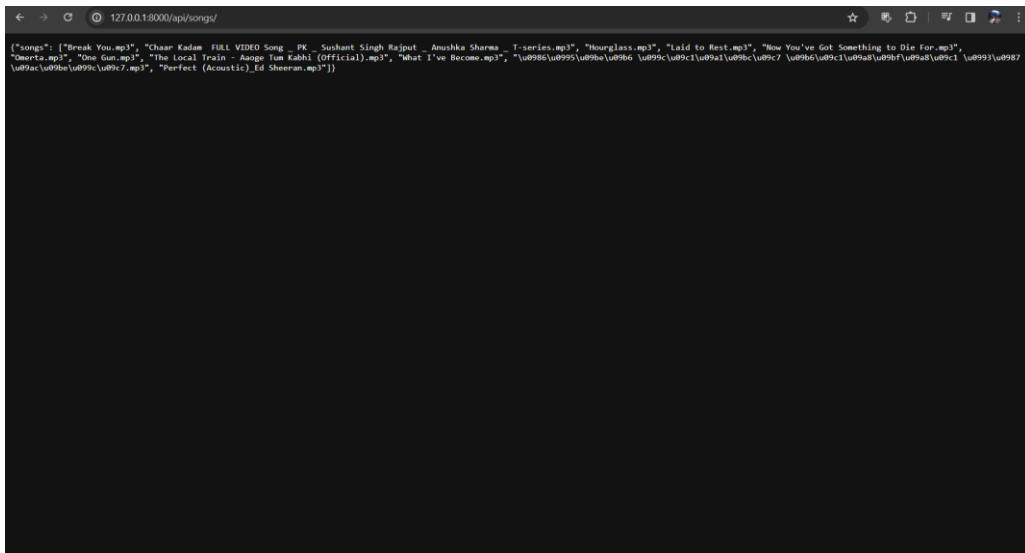
- Uploaded Songs from Admin Dashboard is NOT Successful page Source-Code:

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="{% static 'img/logo/favicon.ico' %}">
    <title>Oops!</title>
    <style>
        .left{
            width: 25vw;
            padding: 5px;
        }
        .left img {
            height: 500px;
            width: 600px;
        }
        .right{
            width: 75vw;
            position: absolute;
            margin-left: 700px;
            margin-top: 130px;
            font-family: Arial, sans-serif; /* Change font family */
        }
        .right h1 {
            color: #FF0000; /* Change color of h1 */
            font-size: 36px; /* Change font size of h1 */
            font-weight: bold; /* Make h1 bold */
            margin-bottom: 20px; /* Add some margin below h1 */
        }
        .right p {
            font-size: 24px; /* Change font size of paragraphs */
            color: #333; /* Change color of paragraphs */
            margin-bottom: 10px; /* Add some margin below paragraphs */
        }
        .right a {
            font-size: 24px; /* Change font size of anchor */
            color: #FF0000; /* Change color of anchor */
            text-decoration: none; /* Remove underline from anchor */
        }
        .right a:hover {
            text-decoration: underline; /* Add underline on hover */
        }
    </style>
</head>
<body>
    <div class="right">
        <h1>Oops!</h1>
        <p>It appears that your song was not uploaded!</p>
        <p><a href="upld">Try again?</a></p>
    </div>
    <div class="left">
        
    </div>
</body>
</html>

```

- All Song List Output:



- All Song List Source-Code:

```

<html>
<head>
  <title>Song List</title>
</head>
<body>
  <h1>Song List</h1>
  <table>
    <thead>
      <tr>
        <th>Song</th>
        <th>Play</th>
      </tr>
    </thead>
    <tbody>
      {% for song in songs %}
      <tr>
        <td>{{ song }}</td>
        <td><a href="/static/songs/{{ song }}" target="_blank">Play</a></td>
      </tr>
      {% endfor %}
    </tbody>
  </table>
</body>
</html>
  
```

- All Song List JS IMPLEMENTATION:

```

console.log("lets write javascript");
let currentSong = new Audio();
let songs;
function formatTime(seconds) {
  seconds = Math.round(seconds);
  const minutes = Math.floor(seconds / 60);
  const remainingSeconds = seconds % 60;
  const formattedMinutes =
    String(minutes).padStart(2, '0');
  const formattedSeconds =
    String(remainingSeconds).padStart(2, '0');
  return `${formattedMinutes}:${formattedSeconds}`;
}
  
```

```

// Async function to fetch and display songs
async function getSongs() {
  try {
    const response = await fetch('/api/songs/');
    const data = await response.json();
    // Display list of songs in the console
    console.log("List of songs:", data.songs);
    // Return the list of songs
    return data.songs;
  } catch (error) {
    console.error('Error fetching songs:', error);
  }
}
  
```

```

const playMusic = (track, pause = false) => {
  const audioSrc = track.endsWith(".mp3") ? track : track + ".mp3";
  currentSong.src = "static/songs/" + audioSrc;
  if (!pause) {
    currentSong.play();
    play.src = "static/img/pause-circle.svg";
  }
  document.querySelector(".songInfo").innerHTML =
  track.replaceAll('.mp3','');
  document.querySelector(".songTime").innerHTML =
  "00:00/00:00";
}
// Main function
async function main() {
  // Fetch and display songs
  songs = await getSongs();
  playMusic(songs[0],true);
  //Show all the songs in the playlist
  let songUL =
  document.querySelector(".songList").getElementsByTag
  Name("ul")[0];
  for (const song of songs) {
    songUL.innerHTML = songUL.innerHTML +`<li>
      
        <div class="info">
          <div>${song.replaceAll('.mp3','')}</div>
        </div>
        <div class="playNow">
          <span>Play Now</span>
          
        </div>
      </li>`;
  }
  Array.from(document.querySelector(".songList").ge
  tElementsByTagName("li")).forEach(e=>{
    e.addEventListener("click",element=>{
      console.log(e.querySelector(".info").firstEle
      ntChild.innerHTML)
      playMusic(e.querySelector(".info").firstElement
      Child.innerHTML.trim())
    })
  })
  play.addEventListener("click",()=>{
    if(currentSong.paused){
      currentSong.play()
      play.src="static/img/pause-circle.svg"
    }
    else{
      currentSong.pause()
      play.src="static/img/play-circle.svg"
    }
  })
}

//timeupdate event
currentSong.addEventListener("timeupdate", () => {
  console.log(currentSong.currentTime,
  currentSong.duration);
  const currentTimeFormatted =
  formatTime(currentSong.currentTime);
  const durationFormatted =
  formatTime(currentSong.duration);
  document.querySelector(".songTime").innerHTML =
  `${currentTimeFormatted} /
  ${durationFormatted}`;
  document.querySelector(".circle").style.left =
  ((currentSong.currentTime)/(currentSong.duration))*100 + "%";
});
//seekbar
document.querySelector(".seekbar").addEventListe
ner("click", e => {
  const progress = (e.offsetX /
  e.target.getBoundingClientRect().width) * 100;
  console.log('Progress:', progress);
  console.log('Duration:', currentSong.duration);
  document.querySelector(".circle").style.left =
  progress + "%";
  currentSong.currentTime = (currentSong.duration
  * progress) / 100;
  console.log('New currentTime:', currentSong.current
  Time);
});
//previous and next
let currentSongIndex = 0; // Keep track of the index
of the current song
// Previous button functionality
document.getElementById("previous").addEventListener
("click", () => {
  // Decrement the current song index
  currentSongIndex--;
  // If the index is less than 0, loop back to the last
  song
  if (currentSongIndex < 0) {
    currentSongIndex = songs.length - 1;
  }
  // Play the previous song
  playMusic(songs[currentSongIndex]);
});
// Next button functionality
document.getElementById("next").addEventListener("c
lick", () => {
  // Increment the current song index
  currentSongIndex++;
  // If the index is greater than the number of songs,
  loop back to the first song
  if (currentSongIndex >= songs.length) {
    currentSongIndex = 0;
  }
})

```

```

// Play the next song
playMusic(songs[currentSongIndex]);
});

}

// Call the main function when the DOM content is
// loaded
document.addEventListener("DOMContentLoaded", main);

```

MODELS.PY:

```

from django.db import models
# Create your models here.
class user(models.Model):
    name = models.CharField(max_length=50)
    pwd = models.CharField(max_length=15)
    class Meta:
        db_table = "user"
class admin(models.Model):
    name = models.CharField(max_length=50)
    pwd = models.CharField(max_length=15)
    class Meta:
        db_table = "admin"
class verify(models.Model):
    name = models.CharField(max_length=50)
    pwd = models.CharField(max_length=15)
    class Meta:
        db_table = "verify"

class AudioFile(models.Model):
    filename = models.CharField(max_length=50)
    fileurl = models.CharField(max_length=200)

    class Meta:
        db_table = "audio_files"

class usersongs(models.Model):
    filename = models.CharField(max_length=50)
    fileurl = models.CharField(max_length=200)

    class Meta:
        db_table = "usersongs"

class event(models.Model):
    ename=models.CharField(max_length=50,null=True)
    eurl=models.ImageField(null=True)
    etitle=models.CharField(max_length=100,null=True)
    elink=models.CharField(max_length=100,null=True)
    eplace=models.CharField(max_length=100,null=True)
    eprice=models.CharField(max_length=100,null=True)
    elocation=models.CharField(max_length=100,null=True)
    etiming=models.CharField(max_length=100,null=True)
    class Meta:
        db_table="event"

```

URLS.PY:

```
from django.urls import path
from . import views
urlpatterns = [
    path("",views.welcome),
    path('login',views.login),
    path('index',views.index),
    path('register1',views.register1),
    path('register2',views.register2),
    path('login2',views.login2),
    path('rgs1',views.rgs1),
    path('rgs2',views.rgs2),
    path('register_success',views.register_success),
    path('log',views.log),
    path('show',views.show),
    path('del/<int:id>',views.dele),
    path('edit/<int:id>',views.edit),
    path('edcode/<int:id>',views.edcode),
    path('show2',views.show2),
    path('log2',views.log2),
    path('del2/<int:id>',views.dele2),
    path('edit2/<int:id>',views.edit2),
    path('edcode2/<int:id>',views.edcode2),
    path('adminpage',views.adminpage),
    path('verifypage',views.verifypage),
    path('show3',views.show3),
    path('del3/<int:id>',views.dele3),
    path('accept_admin/<int:id>',views.accept_admin),
    path('search_user',views.search_user,name='search_user'), #newly added
    path('search_admin',views.search_admin,name='search_admin'), #newly added
    #track upload
    path('upld',views.upld),
    path('upd',views.upd),
    path('notupd',views.notupd),
    # path('songs/', views.song_list),
    path('api/songs/', views.song_list, name='song-list-api'),
    #for rockballads
    path('rockballad',views.rockballad),
    path('rocksongs', views.rock_song_list, name='rock_song_list'),
    #for rabindrasangeet
    path('rabindrasangeet',views.rabindrasangeet),
    path('rabindrsongs',views.rabindra_song_list),
    #for nazrulgeeti
    path('nazrulgeeti',views.nazrulgeeti),
    path('nazrulsongs',views.nazrul_song_list),
    #user playlist upload
    path('userplaylistupload',views.userplaylistupload),
    path('userplaylistshow',views.userplaylistshow),
    path('userplaylistsongs',views.userplaylistsongs),
    path('userplaylistupd',views.userplaylistupd),
    path('userplaylistnotupd',views.userplaylistnotupd),
    #for event upload
    path('eventupload',views.eventupload),
    path('eventupd',views.eventupd),
    path('eventshow',views.eventshow),
    path('eventnotupd',views.eventnotupd),
    path('aboutus',views.aboutus),
```

VIEWS.PY:

```
from django.shortcuts import render,redirect
from django.http import HttpResponseRedirect
from django.http import JsonResponse
from django.conf import settings
from play.models import *
import os
# Create your views here.
def index(request):
    return render(request,'index.html')
def register1(request):
    return render(request,'register1.html')
def register2(request):
    return render(request,'register2.html')
def rgs1(request):
    a = request.GET['a1']
    obj = user.objects.all()
    for i in obj:
        if i.name == a:
            error_message = "The username already exists.  
Please choose a different name."
            return render(request,'register1.html',
{'error_message': error_message})
        u = user()
        u.name = request.GET['a1']
        u.pwd = request.GET['a2']
        u.save()
    return render(request,'register_success.html')

def rgs2(request):
    a = request.GET['a1']
    b = request.GET['a2']
    obj = admin.objects.all()
    for i in obj:
        if i.name == a:
            error_message = "The username already exists.  
Please choose a different name."
            return render(request,'register2.html',
{'error_message': error_message})
    if not admin.objects.exists():
        new_admin = admin.objects.create(name=a, pwd=b)
        return render(request,'register_success.html')
    else:
        v = verify()
        v.name = request.GET['a1']
        v.pwd = request.GET['a2']
        v.save()
    return render(request,'verifypage.html')
def login(request):
    return render(request,'login.html')
def login2(request):
    return render(request,'login2.html')
def register_success(request):
    return render(request,'register_success.html')
def welcome(request):
    return render(request,'welcome.html')
def log(request):
    a = request.GET.get('a1')
    b = request.GET.get('a2')
    obj = user.objects.all()
    if not a or not b:
        error_message = "Please provide both username and password."
        return render(request,'login.html',{'error_message': error_message})
    for i in obj:
        if i.name == a and i.pwd == b:
            return render(request,'index.html')
            error_message = "Incorrect username or password.  
Please try again."
            return render(request,'login.html',{'error_message': error_message})
    def log2(request):
        a = request.GET.get('a1')
        b = request.GET.get('a2')
        obj = admin.objects.all()
        if not a or not b:
            error_message = "Please provide both username and password."
            return render(request,'login2.html',{'error_message': error_message})
        for i in obj:
            if i.name == a and i.pwd == b:
                return render(request,'adminpage.html')
                error_message = "Incorrect username or password.  
Please try again."
                return render(request,'login2.html',{'error_message': error_message})
    def show(request):
        u = user.objects.all()
        return render(request,'show.html',{'u':u})
    def dele(request,id):
        u = user.objects.get(id=id)
        u.delete()
        return redirect("../show")
    def edit(request,id):
        u = user.objects.get(id=id)
        return render(request,'edit.html',{'u':u})
    def edcode(request,id):
        u = user.objects.get(id=id)
        u.name = request.GET['a1']
        u.pwd = request.GET['a2']
        u.save()
        return redirect("../show")
    def search_user(request):
        if request.method=="POST":
            query=request.POST['query']
            users=user.objects.filter(name__contains=query)
            return render(request,'search_user.html',{'query':query,'users':users})
```

```

else:
    return HttpResponse('user_not_found') #newly added

def search_admin(request):
    if request.method=="POST":
        adminquery=request.POST['adminquery']
        admins=admin.objects.filter(name__contains=adminquery)
    return
render(request,'search_admin.html',{'adminquery':adminquery,'admins':admins})

else:
    return
HttpResponse('user_not_found') #newly added

def show2(request):
    a = admin.objects.all()
    return render(request,'show2.html',{'a':a})
def dele2(request,id):
    a = admin.objects.get(id=id)
    a.delete()
    return redirect("../show2")
def edit2(request,id):
    a = admin.objects.get(id=id)
    return render(request, 'edit2.html',{'a':a})
def edcode2(request,id):
    a = admin.objects.get(id=id)
    a.name = request.GET['a1']
    a.pwd = request.GET['a2']
    a.save()
    return redirect("../show2")
def adminpage(request):
    return render(request,'adminpage.html')
def verifypage(request):
    return render(request, 'verifypage.html')
def show3(request):
    v = verify.objects.all()
    return render(request,'show3.html',{'v':v})
def dele3(request,id):
    v = verify.objects.get(id=id)
    v.delete()
    return redirect("../show3")
def accept_admin(request, id):
    # Ensure that the request method is GET
    if request.method == 'GET':
        # Get the admin from the verification table
        v = verify.objects.get(id=id)
        # Create a new admin entry in the admin table
        a = admin.objects.create(name=v.name, pwd=v.pwd)
        # Delete the admin entry from the verification table
        v.delete()
        return redirect('../show3') # Redirect back to the verification table page
    else:
        return redirect('../show3') # Redirect back to the verification table page if not a GET request

def rockballad(request):
    return render(request, 'rockballad.html')

#for songs in index
def song_list(request):
    song_directory = 'play/static/songs'
    songs = os.listdir(song_directory)
    return JsonResponse({'songs': songs})

#for songs of rockballads
def rock_song_list(request):
    folder_path = 'play/static/rocksongs'
    songs = os.listdir(folder_path)
    return JsonResponse({'songs': songs})

#for rabindrasangeet
def rabindrasangeet(request):
    return render(request, 'rabindrasangeet.html')

def rabindra_song_list(request):
    folder_path = 'play/static/rabindrasangeet'
    songs = os.listdir(folder_path)
    return JsonResponse({'songs': songs})

#for nazrulgeeti
def nazrulgeeti(request):
    return render(request, 'nazrulgeeti.html')

def nazrul_song_list(request):
    folder_path = 'play/static/nazrulgeeti'
    songs = os.listdir(folder_path)
    return JsonResponse({'songs': songs})

#user playlist upload
def userplaylistupload(request):
    return render(request, 'userplaylistupload.html')

def handle_uploaded_file4(file, filename):
    if not os.path.exists('play/static/usersongs/'):
        os.makedirs('play/static/usersongs/') # Create directory if not exists

    with open('play/static/usersongs/' + filename, 'wb+') as destination:
        for chunk in file.chunks():
            destination.write(chunk)

def userplaylistupd(request):
    if request.method == 'POST':
        audio_file = request.FILES['audio_file']
        # You can add validation here if needed (e.g., file type, size)

        handle_uploaded_file4(audio_file, str(audio_file))
        url = "upload/" + str(audio_file)

        # Save the file information to the database
        audio = usersongs()
        audio.filename = str(audio_file)
        audio.fileurl = url
        audio.save()

        return render(request, "userplaylistupd.html")
    return render(request,'userplaylistnotupd.html')

def userplaylistnotupd(request):
    return render(request,'userplaylistnotupd.html')

```

```

def userplaylistshow(request):
    audio_files = usersongs.objects.all()
    return render(request, 'userplaylistshow.html',
{'audio_files': audio_files})
def userplaylistsongs(request):
    folder_path = 'play/static/usersongs'
    songs = os.listdir(folder_path)
    return JsonResponse({'songs': songs})
#for track upload
def upld(request):
    return render(request, 'upload.html')

def handle_uploaded_file(file, filename):
    if not os.path.exists('play/static/songs/'):
        os.makedirs('play/static/songs/') # Create directory if
not exists

    with open('play/static/songs/' + filename, 'wb+') as
destination:
        for chunk in file.chunks():
            destination.write(chunk)

def upd(request):
    if request.method == 'POST':
        audio_file = request.FILES['audio_file']
        # You can add validation here if needed (e.g., file type,
size)

        handle_uploaded_file(audio_file, str(audio_file))
        url = "upload/" + str(audio_file)

        # Save the file information to the database
        audio = AudioFile()
        audio.filename = str(audio_file)
        audio.fileurl = url
        audio.save()

        return render(request, "upd.html")
    return render(request,'notupd.html')

def notupd(request):
    return render(request, 'notupd.html')

def show_audio(request):
    audio_files = AudioFile.objects.all()
    return render(request, 'audioshow.html', {'audio_files':
audio_files})

#for event upload
def eventupload(request):
    return render(request,'eventupload.html')

def eventupd(request):
    if request.method=='POST':

        handle_eventuploaded_file(request.FILES['a1'],str(requ
est.FILES['a1']))
        url="eventupload/"+str(request.FILES['a1'])
        u=event()
        u.ename=str(request.FILES['a1'])
        u.eurl=url
        u.etitle=request.POST['a2']
        u.elink=request.POST['a3']
        u.eplace=request.POST['a4']
        u.eprice=request.POST['a5']
        u.elocation=request.POST['a6']
        u.etiming=request.POST['a7']
        u.save()
        return render(request, "eventupd.html")
    return render(request,'eventnotupd.html')

def handle_eventuploaded_file(file,filename):
    if not os.path.exists('play/static/eventupload/'):
        os.mkdir('play/static/eventupload/')

    with open('play/static/eventupload/' + filename, 'wb+') as
destination:
        for chunk in file.chunks():
            destination.write(chunk)

def eventshow(request):
    u=event.objects.all()
    return render(request,'eventshow.html',{'u':u})
def eventnotupd(request):
    return render(request,'eventnotupd.html')
def aboutus(request):
    return render(request,'aboutus.html')

```

IMPLEMENTATION AND TESTING

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be thorough enough to be useful but not so thorough that no one outside the test group will read it.

6a. INTRODUCTION

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

6b. OBJECTIVES OF TESTING:

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

Process Overview

The following represents the overall flow of the testing process:

- Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
- Identify which particular test(s) will be used to test each module.
- Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
- Identify the expected results for each test.
- Document the test case configuration, test data, and expected results.
- Perform the test(s).
- Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
- Successful unit testing is required before the unit is eligible for component integration/system testing.
- Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
- Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

6c. TEST CASES

A test case is a document that describes an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data

Requirement expected results. The process of developing test cases can help find problems in the requirement or design of an application, since it requires completely thinking through the operation of the application.

TESTING STEPS

Unit Testing:

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

Integration Testing:

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

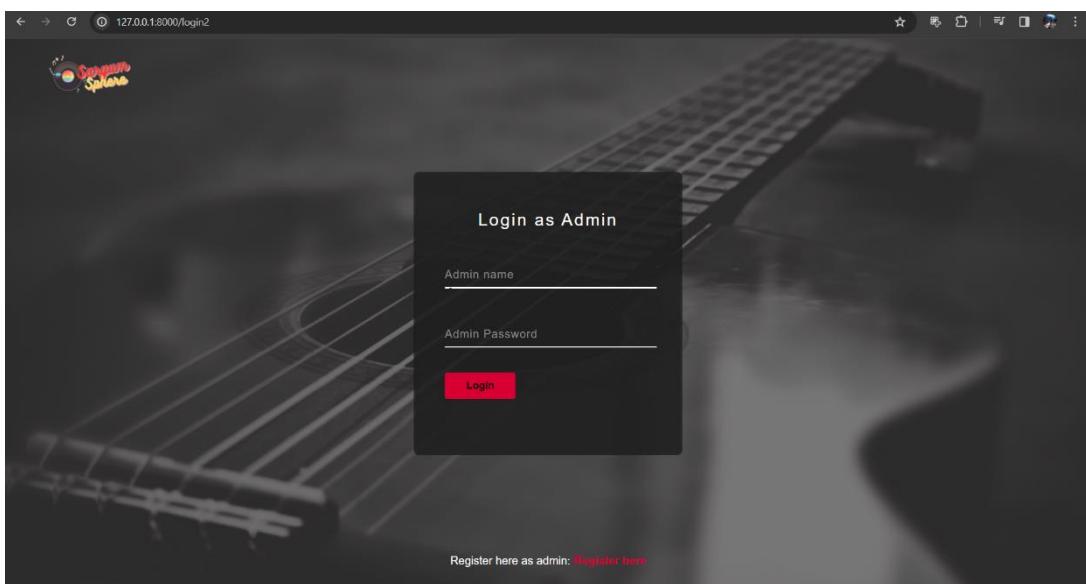
Validation:

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

- a) The function or performance characteristics conform to specification and are accepted.
- b) A deviation from specification is uncovered and a deficiency list is created.
- c) Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

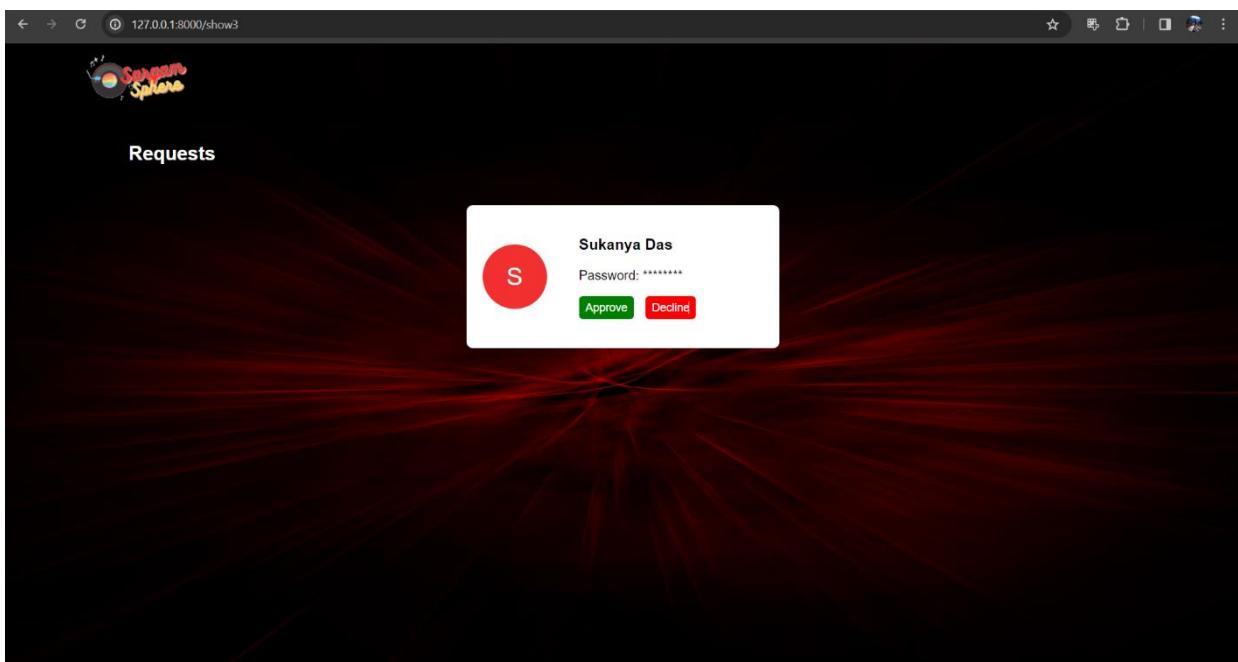
Tested By:	Priyan Mukherjee
Test Type	Unit Testing
Test Case Number	1
Test Case Name	Admin Identification while Login
Test Case Description	The admin should enter his/ her accurate admin name and password so that he/she can able to go for the further options. The test case will check the application for the same since a user can only login with the correct user id, password.
Item(s) to be tested	
1	Verification of the admin id and password with therecord in the database.
Specifications	
Input	Expected Output/Result
1) Correct User id and password 2) Incorrect Id or Password	1) Successful login 2) Failure Message

SNAPSHOT OF ADMIN LOGIN:



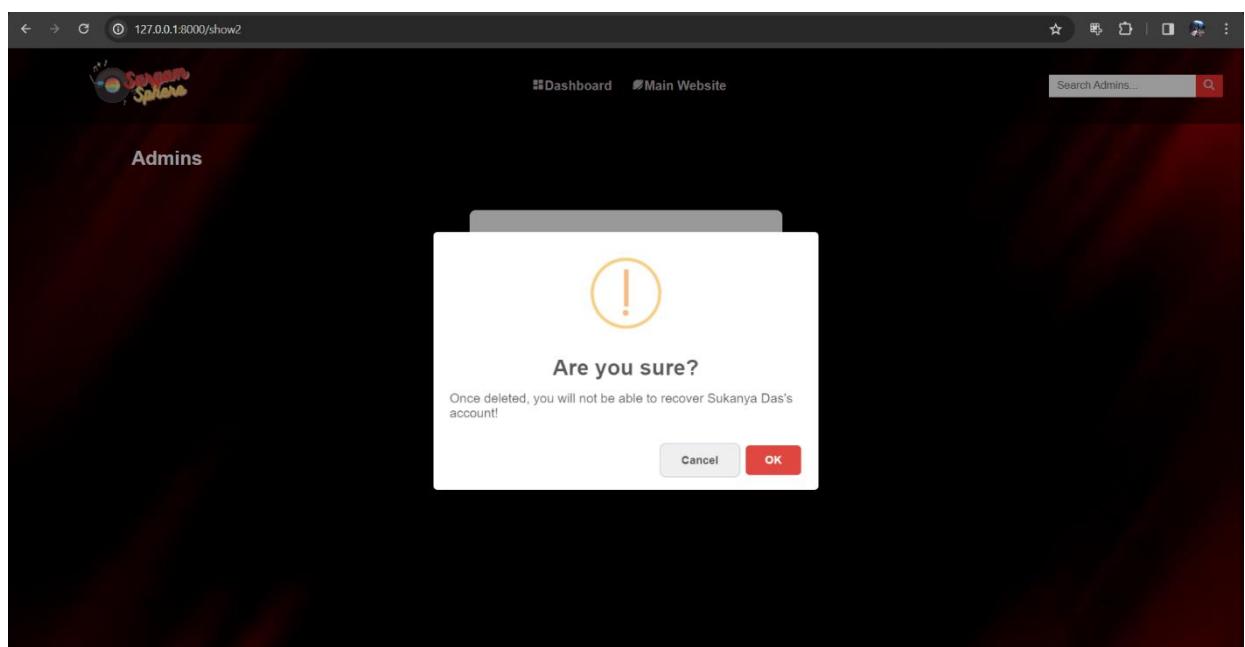
Tested By:	Sukanya Das
Test Type	Unit Testing
Test Case Number	2
Test Case Name	New Admin Verification
Test Case Description	Registered Admins can readily able to register new eligible admin through proper verification procedures
Item(s) to be tested	
1	New admins will be registered as eligible candidates to be admins only after successful verification by registers admins.
Specifications	
Input	Expected Output/Result
1) Eligible admin 2) Plagiarism	1) Successful registration after verification 2) Denial of Registration

SNAPSHOT OF NEW ADMIN VERIFICATION:



Tested By:	Sukanya Das
Test Type	Sweet-Alert while deleting User/Admin Info
Test Case Number	3
Test Case Name	Sweet-Alert for Deleting Data
Test Case Description	A Sweet-Alert will pop up while deleting any record from the database as a warning before deleting record
Item(s) to be tested	
1	Sweet-Alert to pop up while deleting any record from the database as a confirmation before deleting
Specifications	
Input	Expected Output/Result
1) Delete User/Admin Record	1) Sweet-Alert pop up before Deleting record

SNAPSHOT OF SWEET-ALERT BEFORE DELETING RECORD:



Tested By:	Reetom Chakraborty
Test Type	Successful Track Upload from Admin Dashboard
Test Case Number	4
Test Case Name	Track Upload by Admins
Test Case Description	Tracks can be uploaded by Admins from Admin Dashboard and can be showcased on the main webpage if it is successful. Any other file format or unsuccessful upload attempt will not be considered and revert back and will not be stored on database.

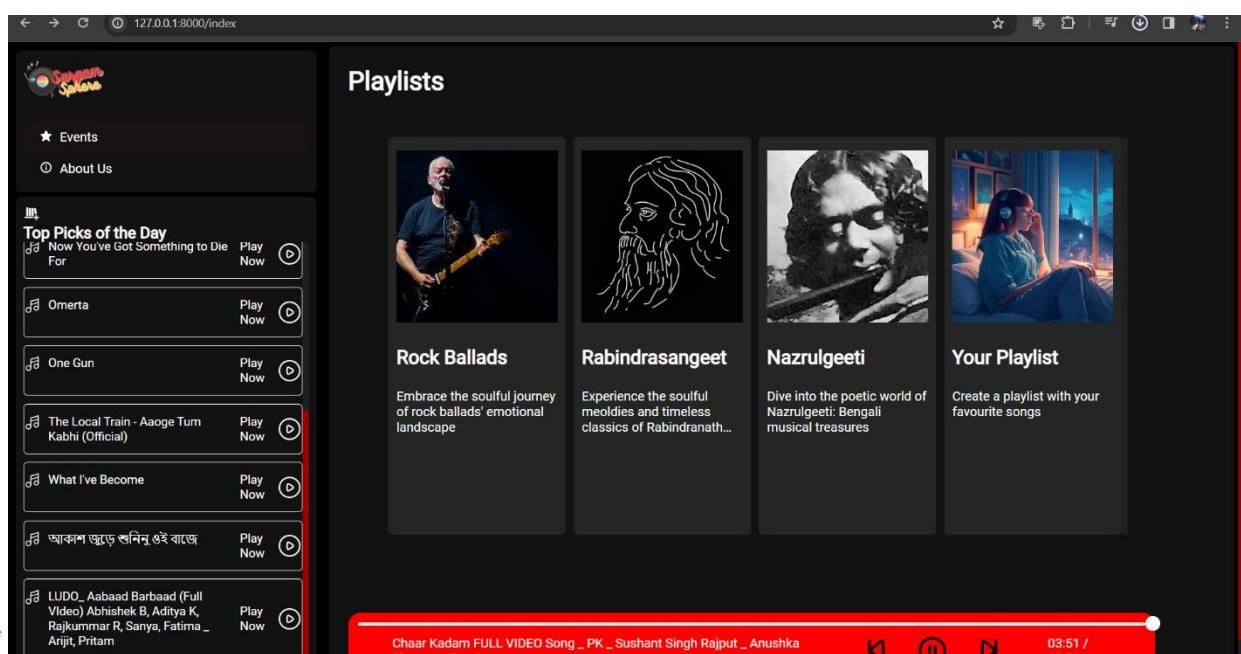
Item(s) to be tested

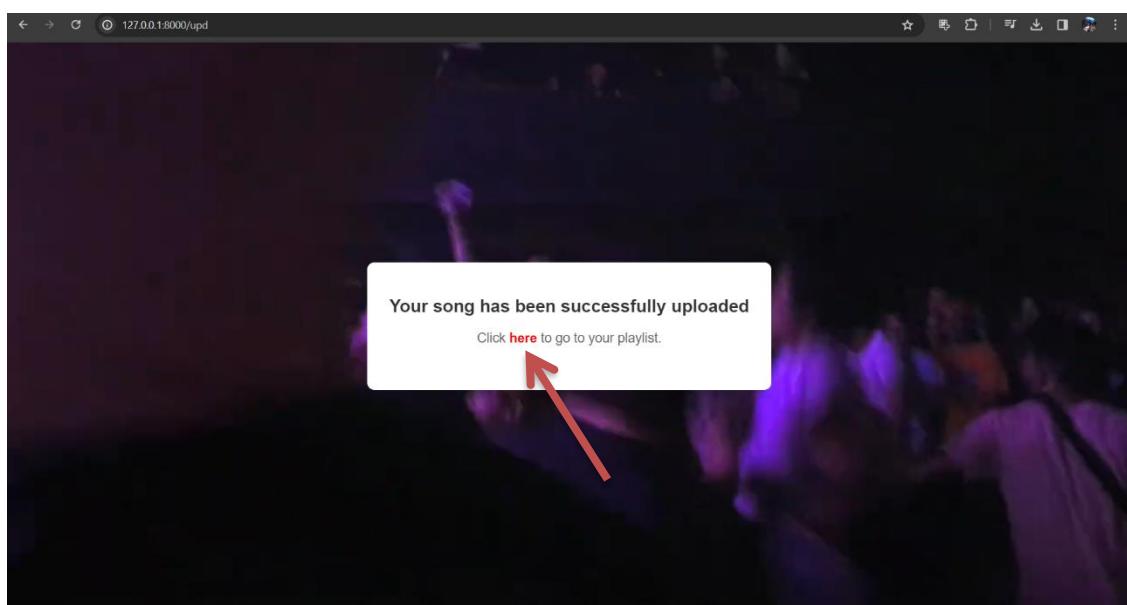
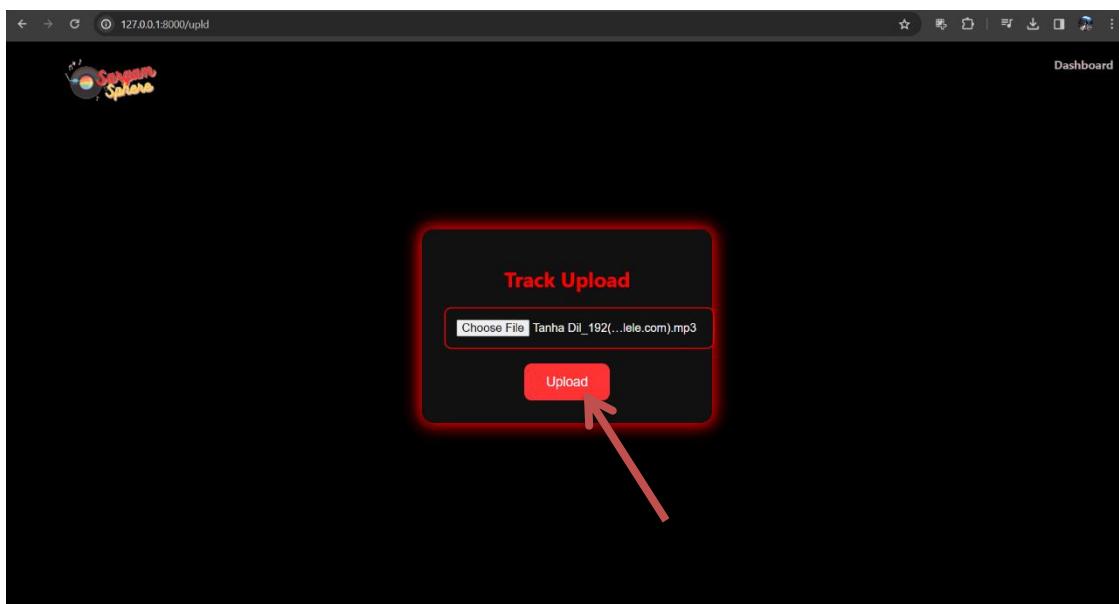
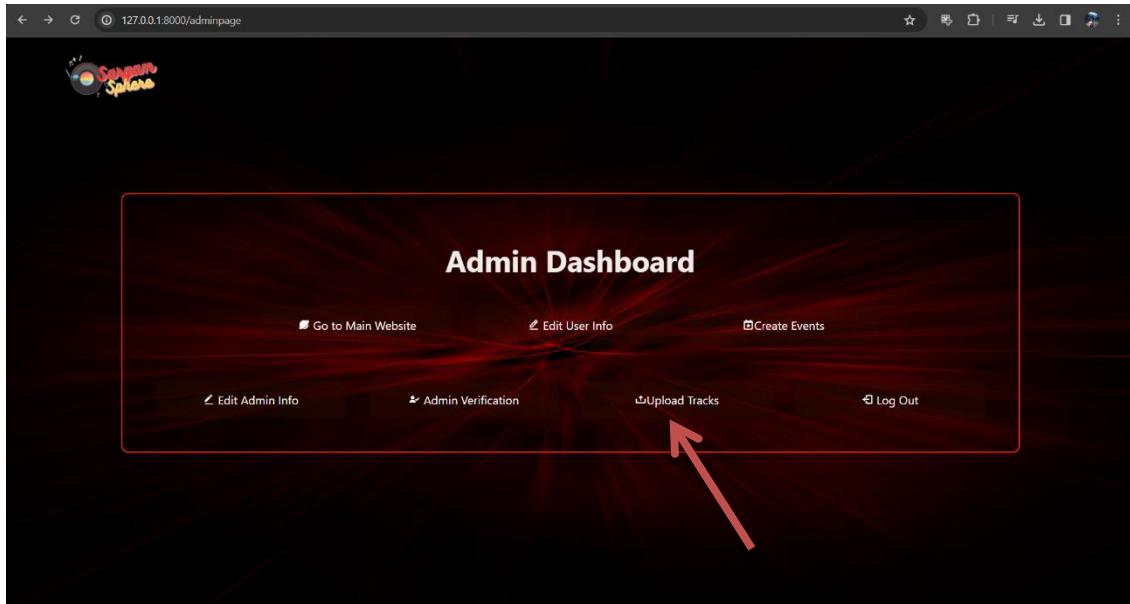
1	Successful Track upload from Admin dashboard by Admin
---	---

Specifications

Input	Expected Output/Result
1) Successful Track Upload	1) Track will be showcased on main Webpage
2) Incorrect File format	2) Failure webpage

SNAPSHOTS OF SUCCESSFUL TRACK UPLOAD:

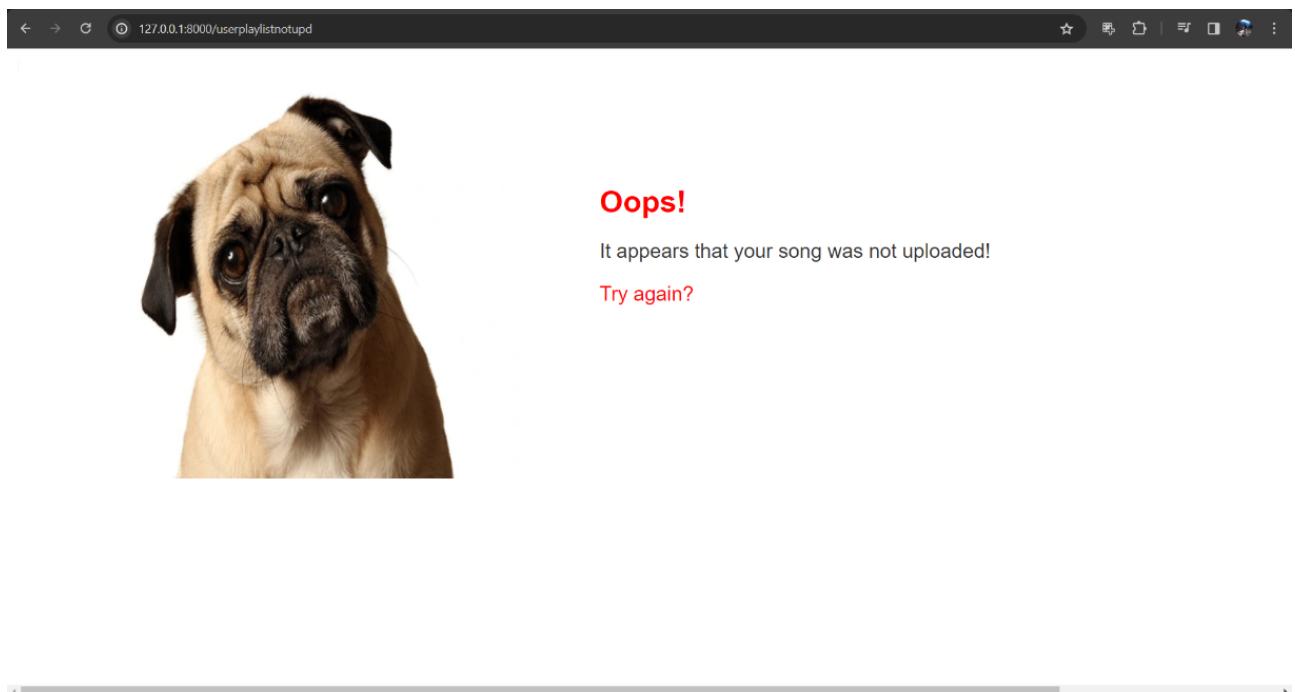




127.0.0.1:8000/index

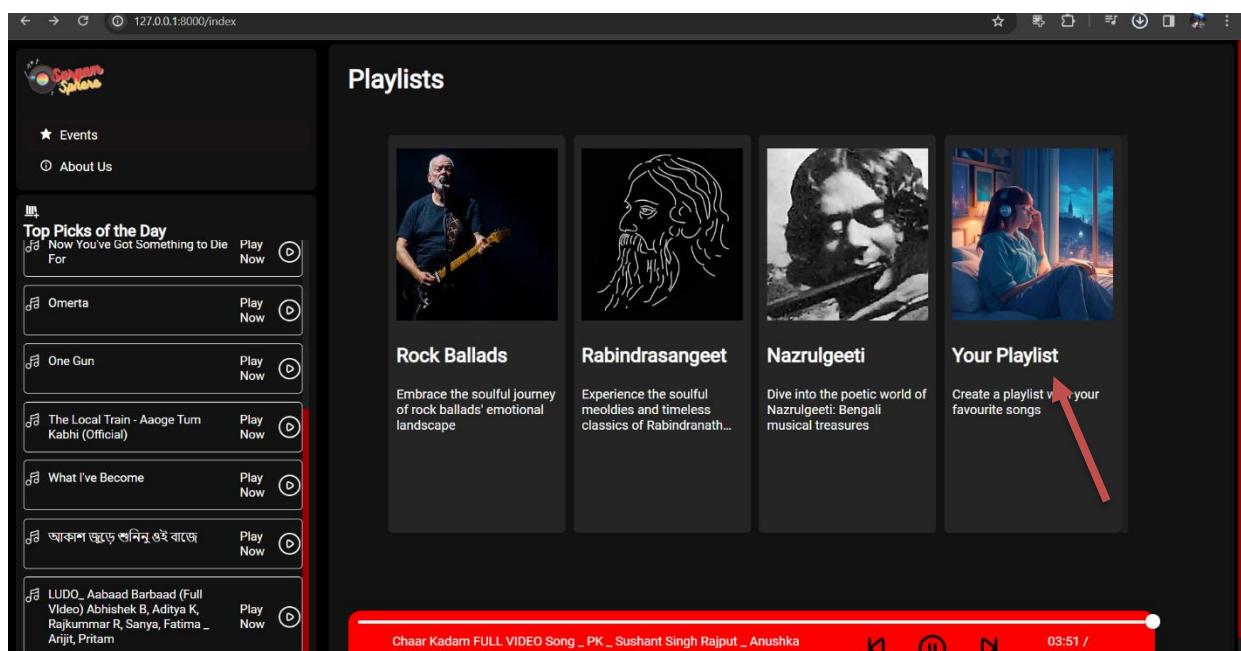
The screenshot shows a dark-themed web application for a music platform. On the left sidebar, there's a logo for 'Ghantalele' and links for 'Events' and 'About Us'. Below these, under 'Top Picks of the Day', are six song cards with titles like 'Omerta', 'One Gun', and 'LUDO_Abaad Barbaad'. A red arrow points from the bottom of this sidebar towards the 'Oops!' message in the second screenshot. The main content area is titled 'Playlists' and features four grid items: 'Rock Ballads' (image of a man playing guitar), 'Rabindrasangeet' (image of Rabindranath Tagore's face), 'Nazrulgeeti' (image of a woman singing), and 'Your Playlist' (image of a person listening to music). At the bottom, a red player bar shows a song titled 'Tanha Dil_192(Ghantalele.com)' at 00:28 / 04:51.

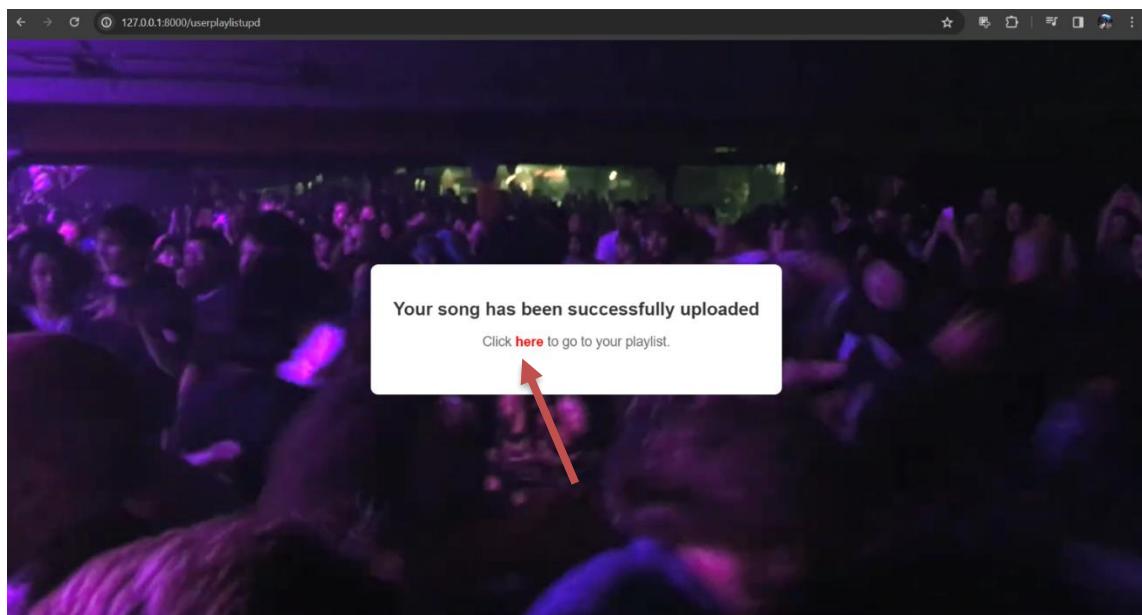
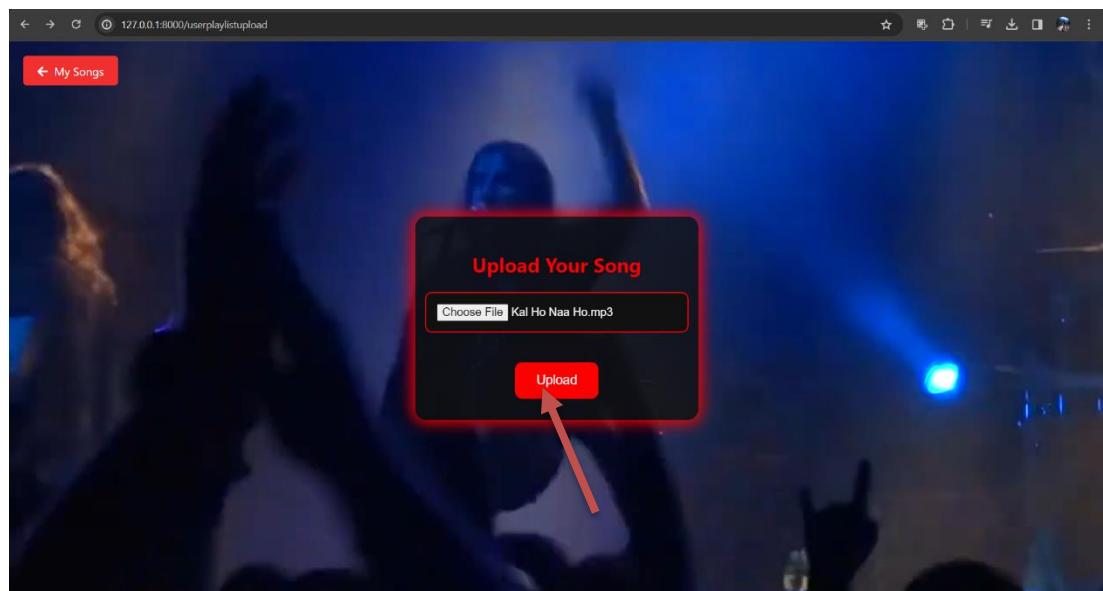
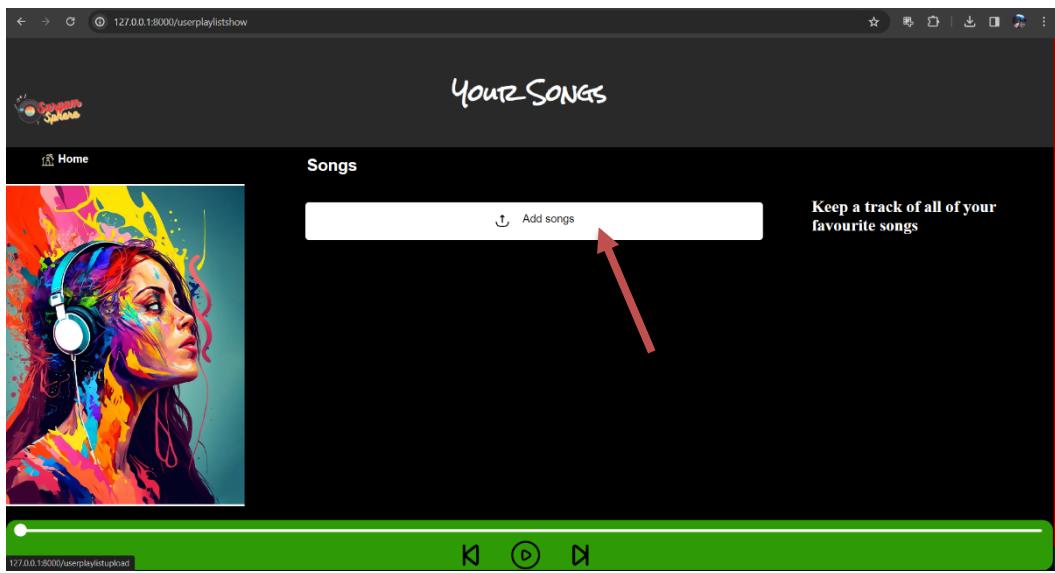
SNAPSHOT OF UNSUCCESSFUL TRACK UPLOAD:



Tested By:	Sangram Banerjee
Test Type	Successful Song Upload from Playlist of User Profile
Test Case Number	5
Test Case Name	Song Upload by User of their Choice
Test Case Description	Songs can be uploaded by User from their own playlist if it is successful. Any other file format or unsuccessful upload attempt will not be considered and revert back.
Item(s) to be tested	
1	Successful Track upload from Playlist of User Profile.
Specifications	
Input	Expected Output/Result
1) Successful Song Upload 2) Incorrect File format	1) Song will be stored on their own playlist 2) Failure webpage

SNAPSHOTS OF SUCCESSFUL SONG UPLOAD:





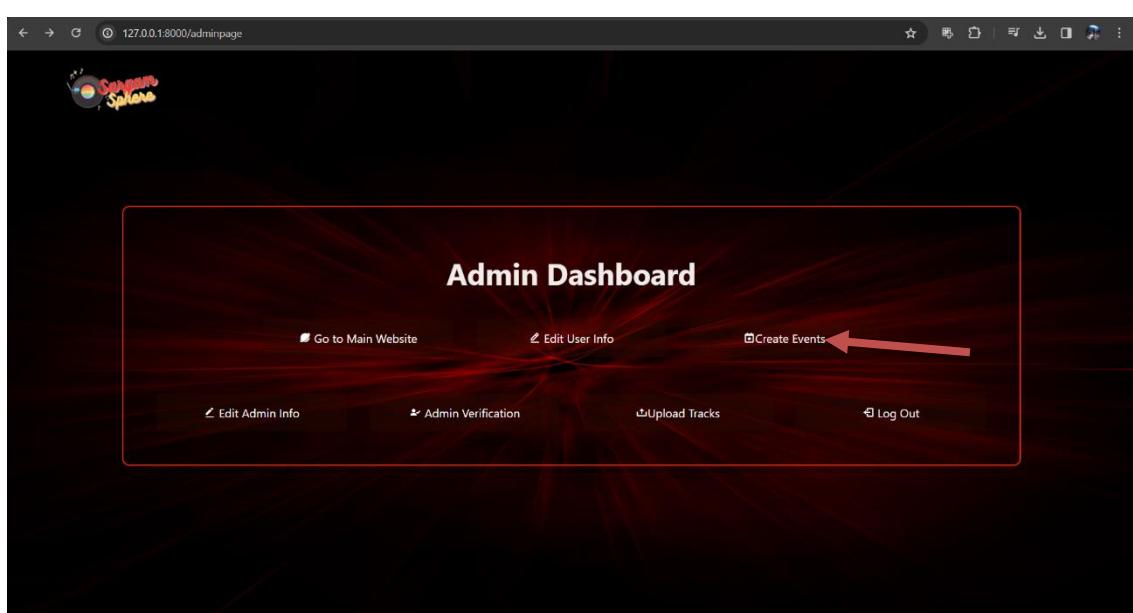
The screenshot shows a web-based music application interface. At the top, the title "Your SONGS" is displayed in a white, hand-drawn style font. On the left, there is a logo for "Songman Sphere". Below the title, a navigation menu includes "Home" and "Songs". A large, colorful abstract painting of a woman wearing headphones serves as the background for the main content area. In the center, a song card for "Kal Ho Naa Ho" is shown, featuring a play button labeled "Play Now". Below the song card is a button labeled "Add songs". A red arrow points from the text "Keep a track of all of your favourite songs" to the "Add songs" button. At the bottom of the screen, a green horizontal bar represents a play bar, with a slider, a play/pause icon, and a timestamp "00:08 / 04:46".

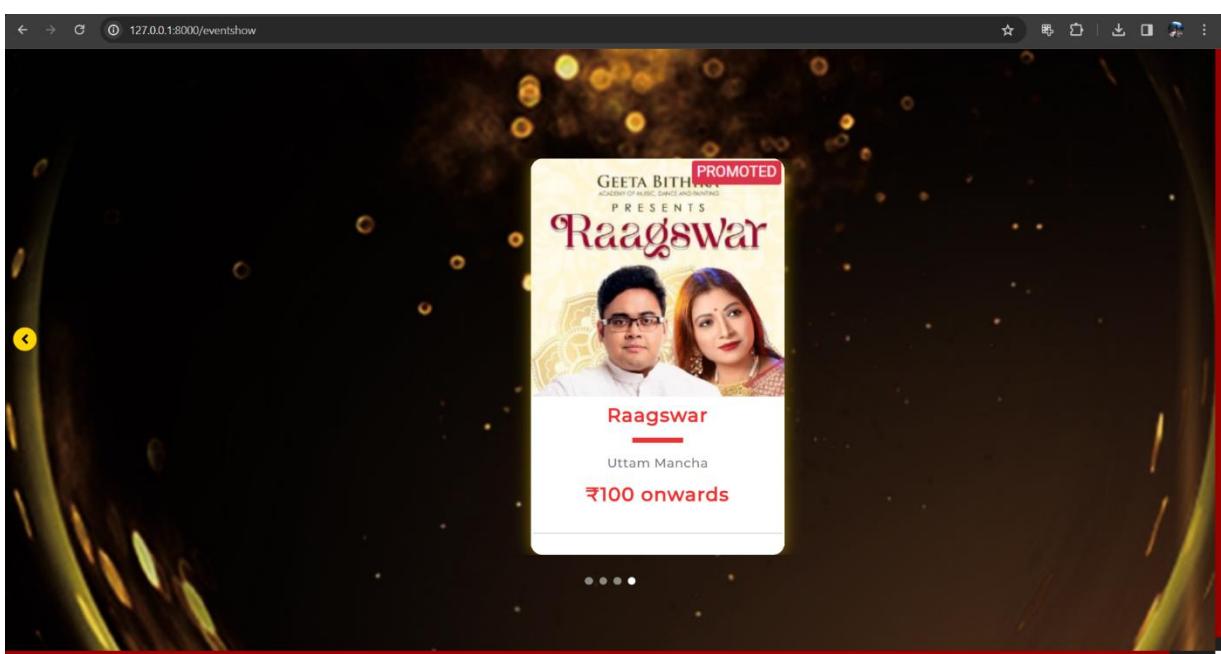
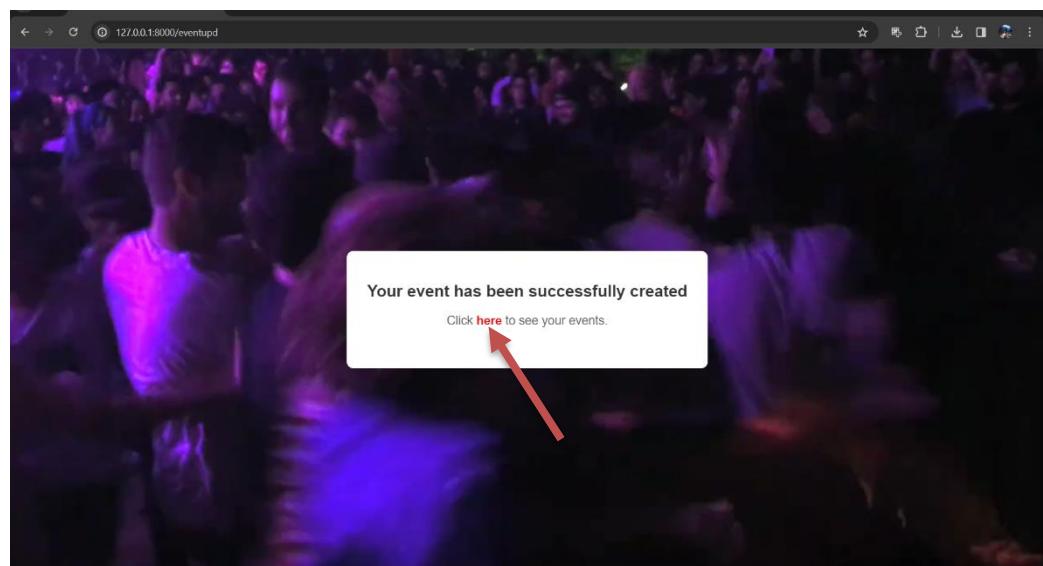
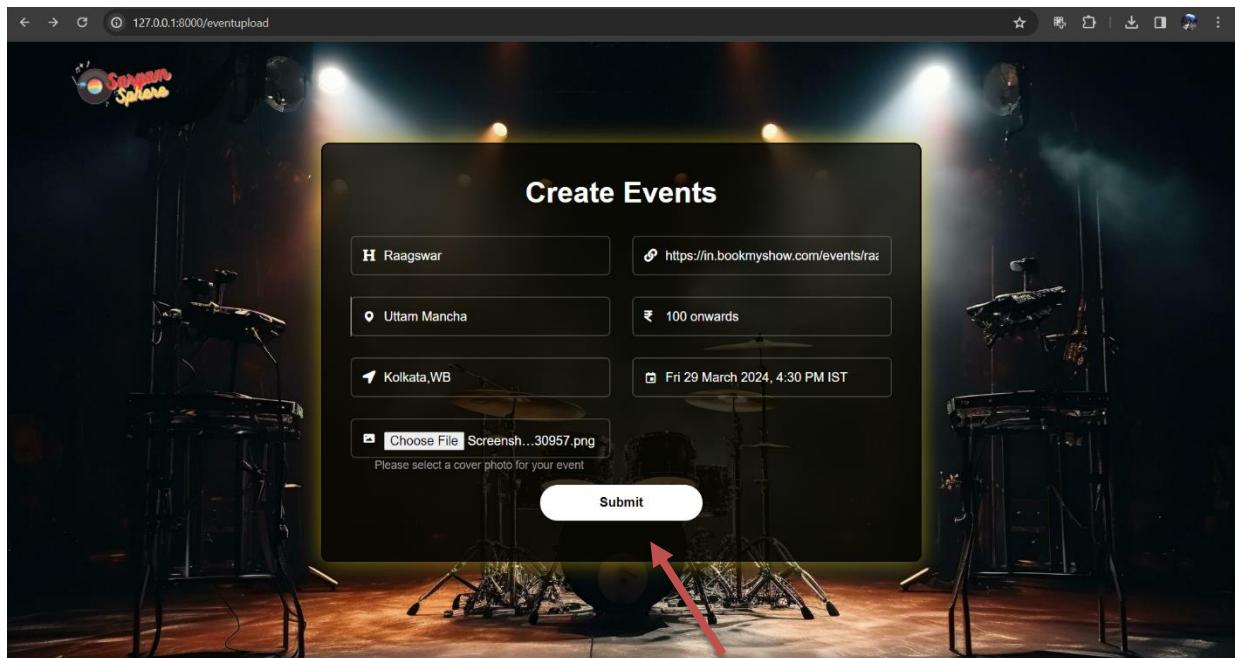
SNAPSHOT OF UNSUCCESSFUL SONG UPLOAD BY USER:

The screenshot shows a user interface for uploading a song. The URL in the address bar is "127.0.0.1:8000/userplaylistnotupd". The main content features a large, cute pug dog looking directly at the camera. To the right of the dog, the word "Oops!" is displayed in a bold, red, sans-serif font. Below "Oops!", a message states "It appears that your song was not uploaded!". Underneath this message is a link "Try again?". At the bottom of the page, there is a teal footer bar with the text "134 MSP".

Tested By:	Priyan Mukherjee
Test Type	Successful Events Upload from Admin Dashboard
Test Case Number	6
Test Case Name	Events Upload by Admins
Test Case Description	Events can be uploaded by Admins from Admin Dashboard and can be showcased on the main webpage if it is successful. This Events are directly linked with BookMyShow portal for better accessibility.
Item(s) to be tested	
1	Successful Event upload from Admin dashboard by Admin
Specifications	
Input	Expected Output/Result
1) Successful Event Upload	1) Events will be showcased on Events Webpage.

SNAPSHOTS OF SUCCESSFUL EVENT UPLOAD:





6 d. WHITE BOX TESTING

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

6e. BLACK BOX TESTING

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases, we will use the black-box method of testing.

6f. OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

User Acceptance Testing:

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

- a) Input Screen Design
- b) Output Screen Design
- c) Format of reports and other outputs.

6g. GOAL OF TESTING

"Program testing can be used to slow the presence of bug, but never to slow their absence." If the results delivered by the system are different from the expected ones, then the system is incorrect and these bugs should be fixed.

6h. INTEGRATION TEST REPORTS

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

- Whether the software matches with the specification (i.e. processbase) to verify the product.
- Whether this software in one client what wants (i.e. product base) to validate the product.

Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was tested to find any unexpected behavior if exists. Later all the modules were integrated and flat file was generated.

FUNCTIONAL TESTING

These are the points concerned during the stress test:

- Nominal input: character is put in the place of digits and the system has to flash the message "Data error"
- Boundary value analysis: exhaustive test cases have been designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

Testing Method Used

We have adopted a testing method which is a mix of both (structural) and black box (functional) testing. For modules we have adopted white box testing. Then we integrated the module into sub - systems and further into the system. These we adopted black box testing for checking the correctness of the system.

Requirements Validated and Verified:

- The data is getting entered properly into database.
- The Screens are being loaded correctly
- The Various functions specified are being performed completely.

SYSTEM SECURITY MEASURES

7.a DATABASE SECURITY

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So, you should follow some security measures. We have used security levels in database level at system level.

7.b SYSTEM SECURITY

If we talk about the system security in our proposed system, we have implemented with the help of maintain the session throughout the system's use. Once a user has logged out than he/she will not be able to perform any task before signing back again.

A high level of authentic login is given to the system so this is a very tedious task to enter without authorization and authentication.

7.c LIMITATIONS:

- ✓ Since it is an online project, customers need internet connection to use it.
- ✓ People who are not familiar with computers can't use this software.
- ✓ Customer must have debit card or credit card to book ticket.

8. CONCLUSION

To sum up, our music streaming platform web development project has obtained widespread praise and approbation from users not only within the organization but also outside. Its user interface that is highly intuitive, as well as the screens that are more user-friendly makes it very easy to use and navigate. Through the graphical user interface (GUI), this tool has become a very simple and effective product that highly improves efficiency and reduces effort for users.

With the passage of time, looking back on the project, we see how the way in which users interact with music has changed. Through considering usability and functionality, we not only make a remarkable product but also serve as a foundation for enhanced productive and fun in the organization. We remain committed to constant improvement and enhancement of our platform, which should address the changing requirements of our clients and thus contribute to its recognized role in the digital environment.

9. FUTURE SCOPE AND FURTHER ENHANCEMENTS

In future we would like to keep working on this project and make new additions to provide users with more advanced features and more detailed information. We have set our sights on the following additions in future: -

1. **Enhanced Personalization Algorithms:** Invest in further refining and expanding our personalized recommendation algorithms.
2. **Creation of Music Room:** Users can join in a room and can experience various musical taste from all around the community they want to colab.
3. **Integration of Social Features:** Expand the social aspect of the platform by integrating features such as user profiles, social sharing, collaborative playlists, and real-time interaction between users.
4. **Expansion of Content Catalogue:** Continuously expand and diversify the platform's content catalogue by partnering with more record labels, independent artists, and music distributors.

- 5. Enhanced Audio Quality and Formats:** Invest in offering higher-quality audio streaming options, such as lossless audio formats or support for immersive audio technologies like Dolby Atmos and Sony 360 Reality Audio.
- 6. Global Expansion and Localization:** Customize the user experience to cater to cultural preferences and music trends in different geographical areas.
- 7. Continuous Performance Optimization:** Implement ongoing performance optimization efforts to ensure the platform remains fast, reliable, and scalable as user traffic grows
- 8. Data Security and Privacy:** Strengthen data security measures and privacy protections to safeguard user data and comply with regulatory requirements.
- 9. User Feedback and Iterative Development:** Solicit feedback from users through surveys, user testing, and analytics to identify areas for improvement and prioritize feature enhancements.

10. BIBLIOGRAPHY

- <https://www.w3schools.com>
- <https://www.slideshare.com>
- <https://www.scribd.com>
- <https://www.tutorialspoint.com>
- <https://www.youtube.com>

THANK YOU