



TRIBHUVAN UNIVERSITY

Office of the Dean
Faculty of Management
Kirtipur, Kathmandu

An Internship Report

On

“Artist Management System (Musica)” for

Cloco Nepal Inc, Pvt. Ltd

In partial fulfillment of the requirement for the degree of
Bachelor in Information Management
(BIM)

Submitted By:

Bishal Phuyal

Exam Roll No.: 9997/19

TU Regd. No. 7-2-551-129-2019

National College of Computer Studies (NCCS)

Paknajol, Kathmandu

September, 2023

DECLARATION

I hereby declare that this internship project entitled “Artist Management System (Musica)” submitted to office of the dean, Faculty of Management, Tribhuvan University, is a result of my own internship study carried out in year 2024 at Cloco Nepal Inc, Kumaripati, Lalitpur, for the partial fulfillment of requirement of the Bachelor of Information Management (BIM). This report is being prepared under the supervision of **Mr. Navin Manandhar**. This project and report are original and has not been submitted earlier in part or full in this or any other form to any university or institute, here or elsewhere, for the award of any degree.

Bishal Phuyal

TU Roll No.: 9997/19

TU Regd No.: 7-2-551-129-2019

SUPERVISORS' RECOMMENDATION

The project report titled “Artist Management System (Musica)” is submitted by Mr. Bishal Phuyal in partial fulfillment of the requirements for the Bachelor of Information Management, Tribhuvan University under my guidance and supervision. To the best of my knowledge, the information presented by him/her in the project report has not been submitted earlier.

Mr. Navin Manandhar

Project Supervisor

National College of Computer Studies (NCCS)

Paknajol, Kathmandu

ACKNOWLEDGEMENT

I am deeply grateful to everyone who contributed to the success of my internship journey.

First and foremost, I want to thank Cloco Nepal Inc Pvt. Ltd., which accepted my presence and provided me to work in their organization. The experience and knowledge I gained during this time to practically learn and closely observe the organization environment and work culture have significantly impacted my personal and professional growth. I had the greatest opportunity to share views and ideas with different people in the organization environment and work culture during my internship period.

I also want to extend a special thanks to Tribhuvan University and my college, NCCS, for allowing me to participate in this internship program. Their support has been crucial in helping me explore and develop my abilities.

I am especially grateful to my internship supervisor, Mr. Navin Manandhar. His valuable guidance, encouragement, and advice were essential in completing both my internship and the accompanying report.

Lastly, I want to thank my mentors, Mr. Sabin Sharma, Mr. Manjeet Pandey, and Mr. Akrit Shrestha. Their guidance, support, and encouragement were instrumental in helping me perform at my best throughout my internship. Their advice and constant encouragement played a significant role in the successful completion of my project.

Thank you all for your support and for making this internship a meaningful and rewarding experience.

Bishal Phuyal

ABSTRACT

We developed a web application called "Musica" to help manage artists and their content. The app offers various features for different users:

- Admin and Staff: They receive detailed reports to track artists, music, and albums. They can also customize heat maps and view staff graph presentations.
- Guest Users: They can browse through artists, music, and albums.
- Registered Users: They can browse, like, comment on, and reply to music and albums.
- Artists: They can create and manage their songs and albums.

Additionally, Musica offers weather recommendations to enhance user experience. Musica provides a basic CRUD operation panel that allows users to add, edit, remove, hide, disable, restore, and enable music, albums, genres, themes, and more. The main goal of Musica is to make it easier for artists to manage their portfolios. The application provides simple tools for uploading, organizing, and showcasing music and albums.

To build Musica, we used several technologies including Python, Django, JavaScript, Vue.js, Tailwind, SQLite, and Postman. We followed the Agile methodology with Scrum to manage the project. This approach gave us hands-on experience working as a team, enhancing our skills in Django and Vue.js. Keywords: Artist Management System, Vue.js, Django, SQLite, Agile, Scrum, Postman, Tailwind

TABLE OF CONTENT

<i>TITLE PAGE</i>	<i>i</i>
<i>DECLARATION</i>	<i>ii</i>
<i>SUPERVISORS' RECOMMENDATION</i>	<i>iii</i>
<i>ACKNOWLEDGEMENT</i>	<i>iv</i>
<i>ABSTRACT</i>	<i>v</i>
<i>TABLE OF CONTENT</i>	<i>vi</i>
<i>LIST OF FIGURES</i>	<i>ix</i>
<i>LIST OF TABLES</i>	<i>x</i>
<i>ABBREVIATIONS</i>	<i>xi</i>
CHAPTER 1 INTRODUCTION	1
1.1 Background of Project	1
1.2 Focus of the Study	1
1.3 Statement of the Problem.....	1
1.4 Objectives	2
1.4.1 Objectives of Internship:.....	2
1.4.2 Objectives of Project:	2
1.5 Literature Review	3
1.6 Gantt Chart.....	4
CHAPTER 2 INTRODUCTION TO THE INDUSTRY.....	5
2.1 Brief Introduction to the Industry	5
2.2 Type (Full-Stack Web Development).....	6
2.3 Objective of the Industry	6
CHAPTER 3 INTRODUCTION OF ORGANIZATION	8
3.1 Introduction.....	8

3.2 Mission Statement from the Chairperson	9
3.3 Goal/Objective	9
3.4 Organization Structure	9
3.5 Working Domains of Organization	11
CHAPTER 4 ACTIVITIES DONE	12
4.1 System Introduction	12
4.2. Internship Activities.....	13
4.3 Methodology and Framework.....	16
4.4 System Analysis.....	17
4.4.1 Requirement Analysis.....	17
4.4.2 Feasibility Study	20
4.5 System Design	20
4.5.1 Architecture Design	20
4.5.2 Database Schema Design.....	21
4.5.3 Class Diagram.....	22
4.5.4 Sequence Diagram	24
4.5.5 Activity Diagram	25
4.6 Module Description	29
4.6.1 Admin Module.....	29
4.6.2 Music Module	30
4.6.3 Album Module.....	30
4.6.4 Theme Selection Module	30
4.6.5 Weather-based Recommendation Module.....	30
4.7 Testing	31
4.7.1 Unit Testing	31

CHAPTER 5 CONCLUSION AND RECOMMENDATION33

5.1 Conclusion 33

5.2 Outcome 33

5.3 Recommendation 34

5.4 Limitations and Considerations 35

REFERENCES36

APPENDIX..... a-f

LIST OF FIGURES

Figure 1.1: Gantt Chart	4
Figure 3.1: Organization Hierarchy Chart	9
Figure 3.2: Board Members of Cloco International.....	10
Figure 3.3: Organization's Leadership Team	10
Figure 4.1: Scrum Framework of Agile.....	17
Figure 4.2: Use Case Diagram	18
Figure 4.3: Architecture Design	20
Figure 4.4: Database Design.....	21
Figure 4.5: Class Diagram	22
Figure 4.6: Sequence Diagram	24
Figure 4.7: Activity Diagram for Guest User	25
Figure 4.8: Activity Diagram for User	26
Figure 4.9: Activity Diagram for Artists	27

LIST OF TABLES

Table 3.1: Overview of Company Profile.....	8
Table 3.2: Overview of Internship.....	11
Table 4.1: Weekly Task Division	14
Table 4.2: Functional Requirements.....	19
Table 4.3: Non-Functional Requirements.....	19
Table 4.4: Unit Testing for Genre Creation.....	31
Table 4.5: Unit Testing for Music Creation.....	31
Table 4.6: Unit Testing for Album Creation	32
Table 4.7: Unit Testing for Theme Creation	32

ABBREVIATIONS

API	Application Program Interface
BIM	Bachelors of Information Management
CRUD	Create, Read, Update, Delete
CSS	Cascade Style Sheet
DRF	Django REST Framework
HTML	Hyper Text Mark-up Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MFA	Multi-Factor Authentication
ORM	Object-Relational Mapping
REST	Representational State Transfer
SQL	Structured Query Language
IT	Information Technology
UI	User Interface
UX	User Experience

CHAPTER 1

INTRODUCTION

1.1 Background of Project

The transition to digital platforms has heightened the demand for effective artist management systems in today's rapidly evolving music industry. These systems play a crucial role in organizing, discovering, and interacting with musical content. However, existing solutions often fall short, lacking cohesive interfaces and user-friendly features. This gap hinders efficient portfolio management for artists and limits user engagement with music. Additionally, administrators face challenges in user management, content categorization, and performance analysis, further complicating efforts to optimize platform operations. (bachelors-information-management-bim-in-nepal, n.d.)

1.2 Focus of the Study

The internship project focuses on developing an Artist Management System tailored for artists, listeners, and administrators. Powered by Django on the backend and Vue.js on the frontend, the system aims to address these challenges. It emphasizes robust database management, secure user authentication, and intuitive frontend interfaces for real-time updates and engaging user experiences. Key features include commenting on music and albums, liking tracks, and a heatmap for administrators to visualize user interactions and engagement statistics.

System Development is a problem-solving process where we bring to bear appropriate elements of mankind's knowledge base to create new knowledge specific to the problem and, as a result, define a solution to the problem (Grady, 2016).

1.3 Statement of the Problem

The artist management industry currently faces significant challenges:

- Inefficient portfolio management tools hinder artists from effectively organizing and showcasing their music.
- Limited user engagement features diminish users' ability to explore new artists and fully interact with the platform.

- Administrators lack adequate tools for user management, content categorization, and performance analysis, hindering platform optimization.
- Concerns over data security and privacy persist due to outdated security measures in existing systems.

1.4 Objectives

1.4.1 Objectives of Internship:

- To gain hands-on experience in developing and implementing an Artist Management System.
- To apply academic knowledge in real-world scenarios, focusing on Django backend development and Vue.js frontend implementation.
- To develop proficiency in database management, secure authentication, and frontend design for optimal user interaction.
- To acquire insights into the music industry's digital transformation and the challenges faced in artist portfolio management.
- To build professional relationships within the tech and music industry, enhancing career prospects and network connections.

1.4.2 Objectives of Project:

- To streamline portfolio management for artists by providing intuitive tools for uploading, organizing, and showcasing music and albums.
- To enhance user engagement through user-friendly interfaces that facilitate seamless music discovery, interaction, liking, and commenting.
- To optimize administrative operations with efficient tools for user administration, content categorization, and performance analytics.

These objectives aim to address the identified challenges and improve the overall functionality and user experience of the Artist Management System.

1.5 Literature Review

In recent years, various applications have emerged to address the complexities of artist management and music event administration, reflecting the evolving digital landscape of the music industry.

SoundCloud and MySpace.com have significantly influenced digital music culture. SoundCloud provides a platform for artists to build communities, interact with fans, and shape their digital identities through music sharing. In contrast, MySpace.com historically empowered bands to connect with fans and foster supportive communities, demonstrating early innovations in social networking for musicians (Shklovski, 2006). These applications and platforms collectively illustrate the transformative impact of digital technologies on artist management, event administration, and community building within the music industry. By facilitating connections between artists, clients, and fans, they contribute to the broader cultural and operational dynamics shaping contemporary music ecosystems. Tailoring marketing approaches towards the fans will aid the process of building a loyal fan base for the artist and maintaining a community that will support the artist and their music for years to come.

SPOTLIGHT is a mobile app for artist management that provides a space for artists to display their talents and skills. It helps clients find the best performers and schedule shows. This app acts as a bridge between artists and clients. The goal of SPOTLIGHT is to create a smartphone app where artists of various types and genres can show their talents through photos, audios, and videos, allowing interested people to contact and hire them for performances. The platform helps artists connect with new clients and opportunities. It also offers a place for upcoming artists to gain recognition. Additionally, clients can view and contact listed artists through a chat feature. (P., Nair, Alapatt, & Thanish Kurian, 2023) Fan engagement is undoubtedly one of, if not, the most important elements of marketing for an artist to have success in their career. Throughout the literature review and my own qualitative research, it was made definitive that music industry professionals have to take the fans into consideration for any marketing purposes, however, an interesting point that came up in the research was that the level of fandom we see in the present day isn't as active as it was maybe five or ten years ago, which suggests that there might've been a point in time in which managers or

marketing professionals weren't so focused on directing their content towards the fans. Tailoring marketing approaches towards the fans 5 will aid the process of building a loyal fan base for the artist and maintaining a community that will support the artist and their music for years to come. (Doyle & Mahomar, 2024) (Doyle & Mahomar, 2024)

1.6 Gantt Chart

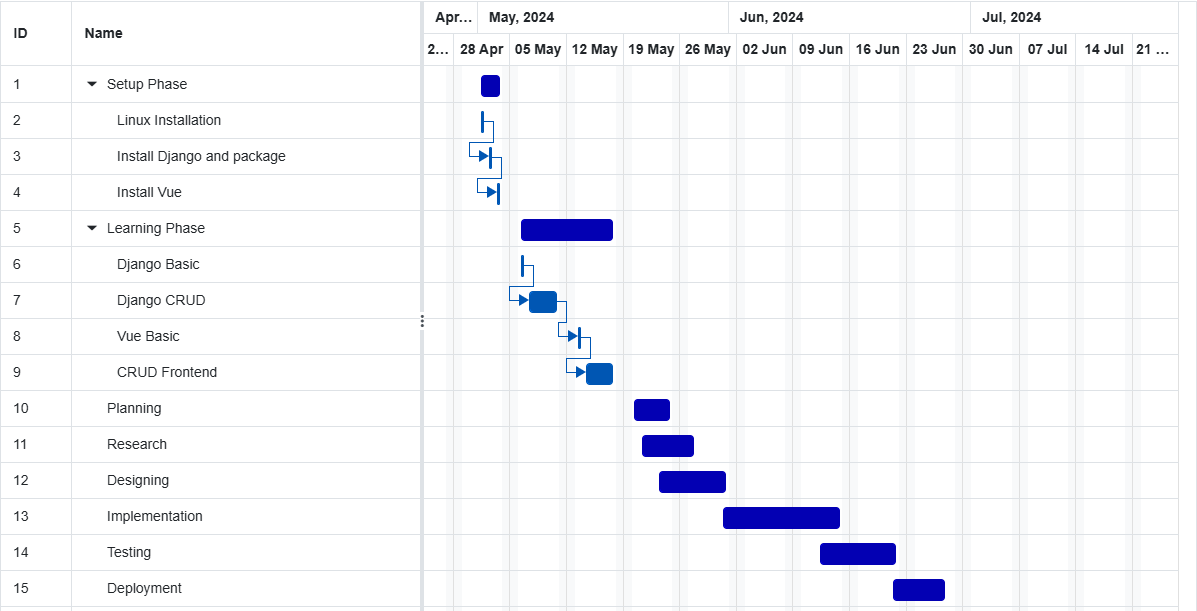


Figure 1.1: Gantt Chart

CHAPTER 2

INTRODUCTION TO THE INDUSTRY

2.1 Brief Introduction to the Industry

The IT industry is a dynamic sector that encompasses the development, implementation, and maintenance of computer systems, software applications, and networks. IT involves using computer systems to manage, process, protect, and share information. It includes a wide range of specialized areas, each dedicated to using technology to solve problems and effectively manage data (Schulze, 2024). It plays a crucial role in modern businesses, governments, and everyday life by enabling efficient data management, communication, and automation of processes. IT professionals work across various domains, from software development to network administration, ensuring technological solutions meet organizational needs and drive innovation.

In today's globally connected world, IT serves as the foundation of nearly every industry, enabling smooth communication and efficient data management. Over the past few decades, its significance has increased dramatically, evolving from basic tasks like bookkeeping and analog communication to driving a largely digital business environment. The Internet now plays a central role in business communication, making digital data a key asset for enterprises (Sugandhi, 2024).

IT continues to transform global industries by incorporating advanced innovations into daily operations. The swift adoption of mobile technologies has sped up digital transformation, improving real-time connectivity and access to information. Moreover, IT enhances supply chain and logistics efficiency through technologies like blockchain, streamlining operations, improving efficiency, and cutting costs. As IT infrastructures become more interconnected, they foster cross-industry collaboration and innovation, opening up new business models and revenue opportunities.

The widespread integration of technology in today's business world has redefined business models, making society more reliant on IT. This shift has altered market dynamics, resulting

in the creation of numerous job opportunities in the IT sector due to continuous technological advancements (Shrestha, 2023).

The history of IT industry in Nepal spans just a few decades, yet the progress in Information Technology has been swift. A significant milestone occurred with the introduction of the IBM 1401, a second-generation mainframe computer, during the 1971 census, marking a pivotal step forward. Another notable advancement came with the utilization of the ICL 2950/10, another second-generation mainframe computer, during the 1981 census, further propelling the country's IT development (Soft, 2019).

2.2 Type (Full-Stack Web Development)

One specialized area within the IT industry is full-stack web development. Full-stack developers are proficient in both front-end and back-end technologies, capable of handling all aspects of web application development. They work with programming languages, frameworks, and tools such as HTML/CSS, JavaScript, React.js, Angular, Node.js, Python, Django, Ruby on Rails, and more. Full-stack developers integrate user interface design with server-side logic, ensuring seamless functionality and user experience across web applications.

2.3 Objective of the Industry

The primary objective of the IT industry, including full-stack web development, is to create robust and user-friendly digital solutions that meet specific business or consumer needs. This involves:

- Developing scalable web applications that operate efficiently and securely.
- Enhancing user engagement and satisfaction through intuitive interfaces and seamless functionality.
- Integrating cutting-edge technologies to innovate and stay competitive in the digital landscape.
- Providing reliable, accessible, and secure platforms for information sharing and transaction processing.

- Supporting business growth by optimizing processes through automation and data-driven insights.

In essence, the IT industry, through full-stack web development and related disciplines, strives to leverage technology to solve complex problems, improve efficiency, and foster innovation in diverse sectors worldwide.

CHAPTER 3

INTRODUCTION OF ORGANIZATION

3.1 Introduction

Cloco Nepal Inc., established in 2023, is a dynamic and innovative IT company at the forefront of technological solutions in Nepal. With a passionate team of skilled developers, designers, and IT professionals, we deliver cutting-edge software, web applications, and digital services tailored to meet the evolving needs of businesses in today's fast-paced digital landscape. It operates as a subsidiary of Cloco Inc. Japan and specializes in cloud services and software development. The company offers a wide range of IT services tailored to meet various technological needs.

Cloco Nepal Inc. excels in creating custom solutions that help businesses of all sizes expand their markets, enhance their offerings, and increase revenue. The company is supported by a skilled team with extensive knowledge and expertise in technology and business strategies to achieve ambitious goals.

Services provided include software development, IT audits, digital marketing, cloud services, and design. Cloco Nepal Inc. fosters a collaborative work environment where seniors, supervisors, and mentors provide supportive guidance. To remain competitive, the company continuously updates its technologies to adapt to market trends. (about, 2024)

Table 3.1: *Overview of Company Profile*

Organization Name	Cloco Nepal Inc.Pvt.Ltd
Address	Bagaldo Marg, Chakupat, Ward No. 11, Lalitpur Metropolitan City, Nepal
Contact	+977-9761693518
Year of establishment	2023
Email	hr.nepal@cloco.co.jp

3.2 Mission Statement from the Chairperson

“Our mission is to provide services that enable customers to be happy and comfortable by leveraging our Technical capabilities and Interpersonal skills. What we value is system development that allows our customers and engineers to grow together while having fun. In order to respond to the rapidly changing environment, we will lead the world with professionalism and confidence, without being swayed by fashionable trends and instead, focusing on solving problems as a team. We will continue to make the world a more enjoyable and convenient place.” -**Takumi Kuniyoshi**

3.3 Goal/Objective

- Delivering client-centric solutions that simplify complex challenges.
- Partnering with startups and enterprises for digital transformation.
- Navigating the evolving technology landscape with excellence and commitment.

3.4 Organization Structure

Cloco Nepal Inc consists of a leadership team led by a general manager, administrative staff, HR personnel, project managers, full-stack developers, and interns.

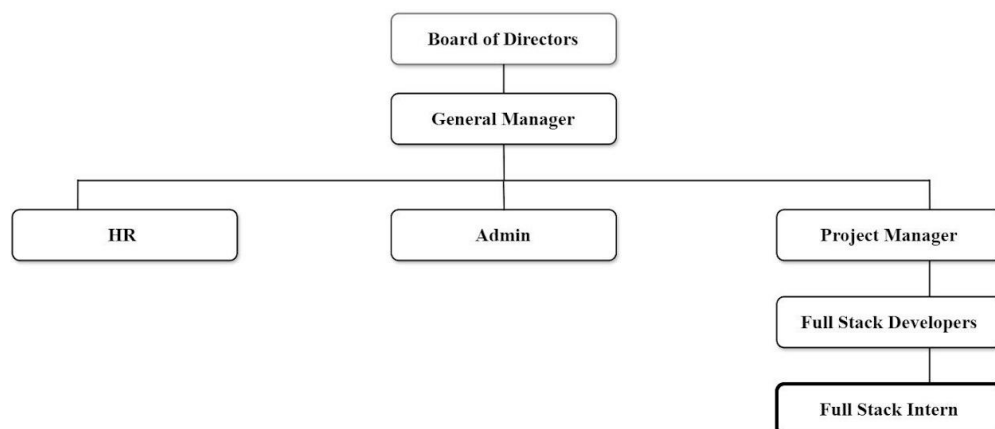


Figure 3.1: *Organization Hierarchy Chart*

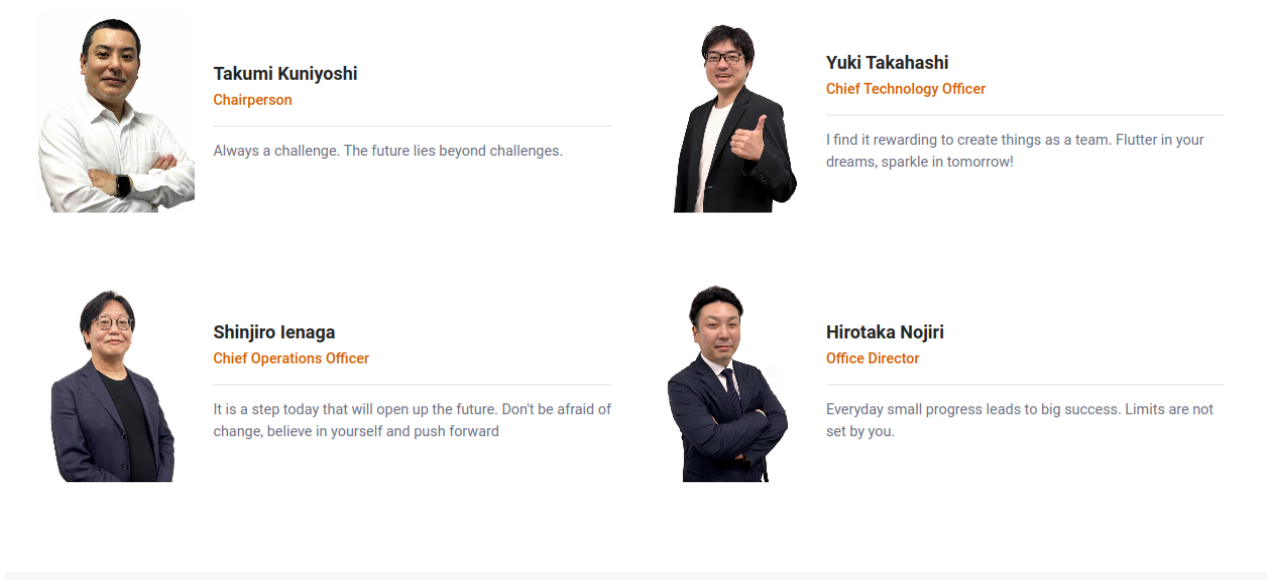


Figure 3.2: *Board Members of Cloco International*

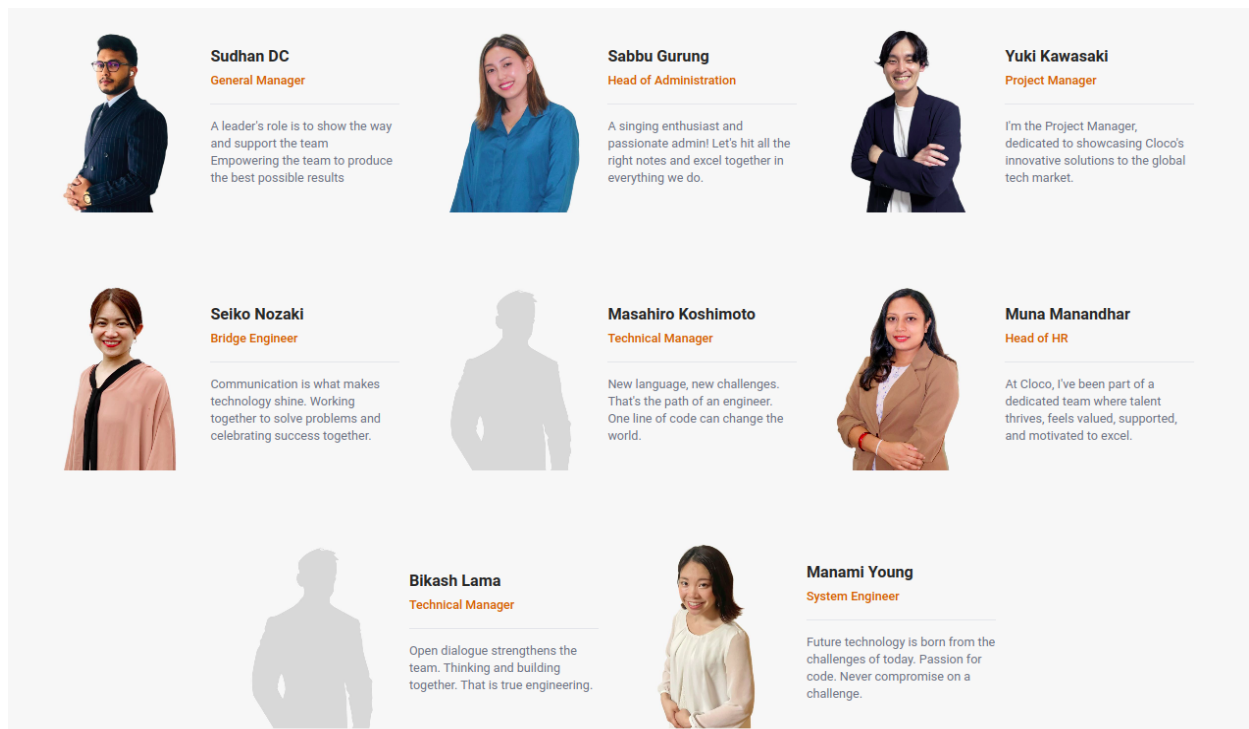


Figure 3.3: *Organization's Leadership Team*

Table 3.2: *Overview of Internship*

Start Date	1 st May
End Date	1 st August
Total Duration	3 Months
Working Days	5 Days a week
Intern Position	Full-stack web Development
Mentor	Mr. Manjeet Pandey
Office Hours	9:00 am to 6:00 pm

3.5 Working Domains of Organization

Cloco Nepal Inc. is a Business Solution-oriented company that offers a wide range of services. Among the services, we can categorize the working domain of the company as follows:

- Mobile Application Development
- Web Development
- Cloud Services

CHAPTER 4

ACTIVITIES DONE

4.1 System Introduction

During my internship at Cloco Nepal Inc., the development of Musica, an innovative Artist Management System, was the primary focus. Designed to enhance the management of artists' musical portfolios, Musica leveraged Django for the backend infrastructure and Vue.js for the frontend interface. The system was aimed at optimizing user engagement and operational efficiency. Key features included comprehensive user management capabilities, enabling tailored functionalities for artists, general users, and administrators. Artists were able to efficiently upload, manage, and showcase their music and albums, supported by intuitive CRUD operations integrated seamlessly with the platform. A dynamic and responsive user interface was created using Vue.js, enhancing user interaction through real-time updates, interactive elements, and intuitive navigation pathways.

Community engagement was fostered through features that allowed users to like music, leave comments, and participate in a vibrant musical ecosystem. Administrators benefited from tools designed to manage user accounts, categorize music genres, and access analytics for monitoring platform performance and user activity. The project aimed to revolutionize artist management by offering a user-centric platform that streamlined content management while providing a rich, engaging experience for music enthusiasts.

Musica is a web application designed for managing artist portfolios and their content. It enables admins, staff, and guest users to track artists, browse music and albums, and interact through comments and likes. Artists can create and manage their songs and albums using tools.

- The project is done using Django, Vue.js and SQLite.
- It includes features such as browsing artists, music, and albums, commenting, liking, and interacting with content. It offers artist tools for creating and managing songs and albums, alongside admin tracking and reporting capabilities.
- The system is easy to use and responsive.

4.2. Internship Activities

As a full stack developer, I had a wide range of responsibilities aimed at improving our web platform's features and user experience. These included creating easy-to-use interfaces that are consistent and attractive. I built reusable parts for both the front and back ends to make development faster. I turned designs and plans into polished code that works well on different devices.

I regularly joined code reviews with teammates to improve our code quality and learn from each other. Good communication with mentors, managers, and teammates was crucial for updating on projects and solving problems together.

Other responsibilities included:

- Making sure our web pages are easy for people to use and look good.
- Creating parts of the website that we could use again to make developing faster.
- Turning designs into code that works well and looks good.
- Showing how much we've done each week and giving good ideas to make our app better.
- Building good relationships with everyone to work better as a team.

These tasks helped us reach our goals well and make our web platform even better for users. The following weekly logs provide a detailed account of the technical activities carried out during the internship. The logs serve as a chronological record of week-by-week experiences during the internship, covering encountered challenges, employed problem-solving strategies, and achieved milestones. In the first week of the internship, the main task involves immersing oneself in organizational activities, such as familiarizing with the company environment, understanding the development workflow, getting acquainted with the intern's responsibilities, and exploring technologies like Django, Vue.js, and Linux.

Table 4.1: *Weekly Task Division*

Week	Task
Week1 (May 1 - May 3)	<ul style="list-style-type: none">• Getting to know how the company works, learning about what interns do, and exploring technologies like Django, Vue.js, and Linux.• Talking to the development team to understand their roles, current projects, and how they work together.
Week 2 (May 6 - May 10)	Technology Stack , Setup <ul style="list-style-type: none">• Setting up pyenv, Poetry, and Django, learning about Django's folder organization, models, views, migrations, creating apps, and setting up URL routing.• Learned fundamental concepts of JavaScript, basics of Vue.js, and Vue lifecycle hooks.• Implemented Vue.js functionalities for user management, including fetching API data, displaying results in the ListUser component, posting data based on UI interactions, and deleting data through API calls.
Week 3 (May 13 - May 17)	System Design of Artist Management <ul style="list-style-type: none">• Established a database structure that efficiently stores information related to artists, including defining relationships between different data entities.• Created a diagram that visually represents the interactions and functionalities essential for artist management, aiding in understanding system requirements and user roles.• Set up the initial project environment by configuring Vue.js and DRF, crucial frameworks for frontend development and backend API creation.

<p>Week 4</p> <p>(May 20 – May 24)</p>	<p>Initial Backend and Frontend Development</p> <ul style="list-style-type: none"> • Designed and implemented backend API endpoints to support user CRUD operations, including profile creation, retrieval, updating, and deletion. • Created frontend components for user login, signup, and profile editing to provide seamless user interactions. • Integrated frontend forms and UI components to enable users to perform CRUD operations on their profiles and manage their authentication seamlessly.
<p>Week 5</p> <p>(May 27 – May 31)</p>	<p>Album Management Feature and User Profile</p> <ul style="list-style-type: none"> • Implemented role-based authentication to manage access permissions based on user roles. • Implemented navigation guards to control access to different routes based on user authentication and authorization. • Designed and developed a homepage with dynamic content based on user roles and preferences.
<p>Week 6 and 7</p> <p>(June 3– June 14)</p>	<p>User Profile and Artist Profile</p> <ul style="list-style-type: none"> • Implemented functionality for artists to hide/unhide music items. • Implemented functionality for admins and staff to enable/disable music and album items.
<p>Week 8 and 9</p> <p>(June 17– June 28)</p>	<p>Theme Selection Feature</p> <ul style="list-style-type: none"> • Created API endpoints to manage themes, allowing administrators and staff to perform actions such as adding, updating, and deleting themes. • Integrated forms where administrators and staff could input theme details such as name, colors and background picture.

	<ul style="list-style-type: none"> • Developed UI components to display available themes and allow artists to choose one that suits their preferences. • Ensured that other users viewing an artist's profile saw the selected theme applied, enhancing consistency and personalization across the platform.
Week 10 and 11 (June 30 – July 13)	<p>Weather Based Music Recommendation Feature</p> <ul style="list-style-type: none"> • Incorporated the OpenWeather API to retrieve real-time weather data based on user location. • Implemented CRUD operations (Create, Read, Update, Delete) for mapping music genres with weather conditions. • Created user interfaces allowing administrators and staff to add, update, and delete genre mappings. • Fetched weather data using the OpenWeather API and mapped it to predefined music genres.
Week 12 (July 15 – August 1)	<p>Testing and Debugging</p> <ul style="list-style-type: none"> • Carried out thorough testing on both the backend and frontend, which included conducting unit tests, integration tests, and debugging to find and fix issues.

4.3 Methodology and Framework

Our project follows an Agile methodology, which emphasizes flexibility, collaboration, and iterative development to deliver high-quality results efficiently. We adopt a 1-week Sprint of Scrum where team members rotate roles to foster cross-functional skills and maximize productivity. The roles include:

- **Project Manager:** Leads the team, coordinates tasks, and ensures project milestones are met on time and within scope.
- **Frontend Developer:** Focuses on designing and implementing user interfaces, ensuring a seamless and intuitive user experience.
- **Backend Developer:** Manages server-side logic, database integration, and API development to support frontend functionalities.

- **Testing:** Ensures the quality and functionality of the product through comprehensive testing, identifying and resolving issues to maintain product integrity.

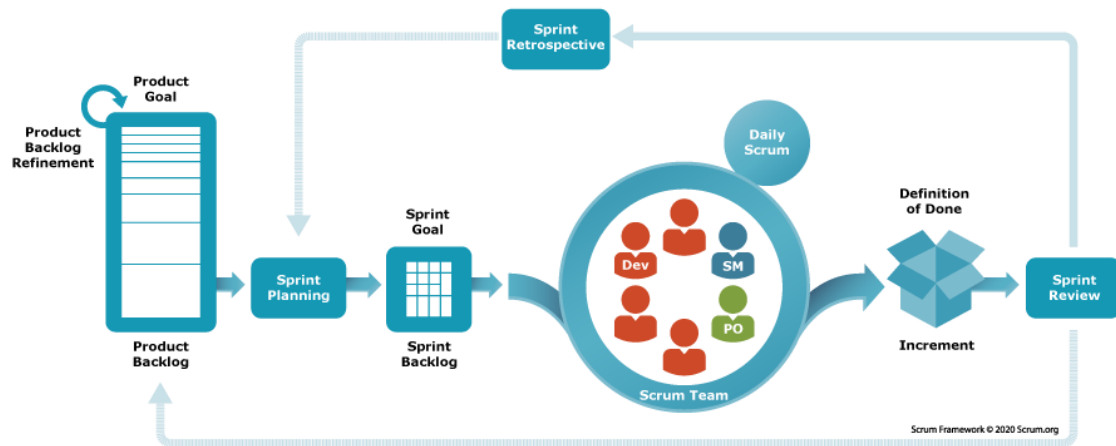


Figure 4.1: *Scrum Framework of Agile*

This approach allows our team at Internship. to adapt quickly to changing requirements, address challenges proactively, and deliver solutions that meet or exceed client expectations. Each Scrum cycle promotes learning and collaboration among team members, enhancing overall project efficiency and success.

4.4 System Analysis

4.4.1 Requirement Analysis

For this project, the requirement analysis process aimed to identify the key features and functionality of the system, as well as any constraints or limitations that needed to be considered during development.

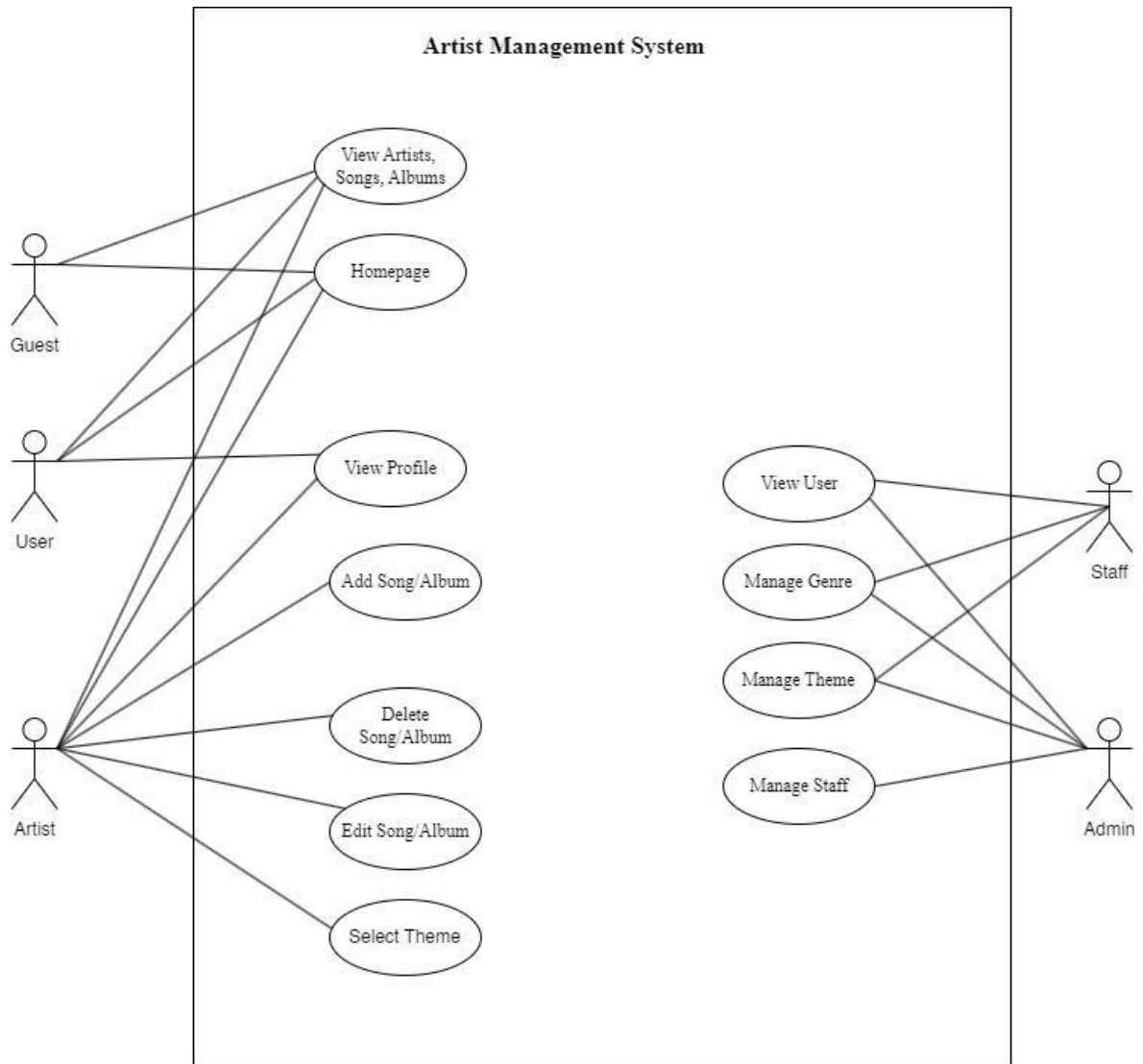


Figure 4.2: *Use Case Diagram*

The use case diagram for the Musica highlights the various actions available to different user roles. Guests can view artists, songs, and albums and access the homepage. Users can view profiles. Artists can add, delete, edit songs/albums, and select themes. Staff members can view users and manage genres and themes. Administrators have the added ability to manage staff, providing a structured workflow and role-specific functionalities.

4.4.1.1 Functional Requirement

Table 4.2: *Functional Requirements*

Req.No	Requirement Name	Requirement Description
FR1	View Profile	Users and artist should be able to view their profile.
FR2	Add Song/Album	Artists should be able to add new songs or albums.
FR3	Delete Song/Album	Artists should be able to delete songs or albums.
FR4	CRUD Genre	Admin should be able to add, delete, update genre.
FR5	CRUD Theme	Admin should be able to add, delete, update theme.
FR6	Add Staff	Admin should be able to add staff member.
FR7	Disable/Enable Staff	Admin should be able to disable or enable staff members.
FR8	Select Theme	Artists should be able to select a theme for their profile or page.

4.4.1.2 Non-functional Requirement

Table 4.3: *Non-Functional Requirements*

Req.No	Requirement Name	Requirement Description
NFR1	Performance	The system should respond to user actions (e.g., login, adding a song/album, commenting) within seconds.
NFR2	Scalability	The system should be capable of handling a high volume of concurrent users without performance degradation.
NFR3	Security	The system must ensure that only authenticated users can access certain features (e.g., adding songs/albums).
NFR4	Usability	The system should provide a user-friendly and intuitive interface for all user roles.
NFR5	Maintainability	The system's code should follow standard coding practices and be well-documented to facilitate maintenance.

4.4.2 Feasibility Study

4.4.2.1 Technical Feasibility

The system will be built using modern web technologies. The frontend will use Vue.js, which is known for its ease of use and efficiency. The backend will be developed with a robust framework like Django, which is well-supported and widely used.

4.4.2.2 Operational Feasibility

The system will support multiple user roles, including guests, users, artists, staff, and administrators. Each role will have specific functionalities tailored to their needs. The system will include tools for easy maintenance and updates. Administrators will be able to manage users, content, and settings without requiring technical knowledge.

4.4.2.3 Financial Feasibility

The initial development cost includes hiring developers, designers, and project managers. Using modern frameworks and tools will help streamline the development process, reducing costs. Ongoing costs include server hosting, maintenance, and support services. Using cloud services will provide flexibility and cost efficiency, as expenses will scale with usage.

4.5 System Design

4.5.1 Architecture Design

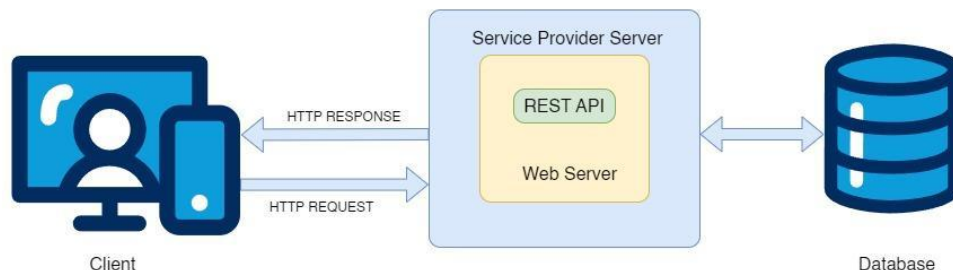


Figure 4.3: *Architecture Design*

The above architecture consists of a client side (web and mobile) interacting with a service provider server via a REST API. The server handles requests and communicates with a database to store and retrieve data. This setup ensures efficient data management and user interactions for the Artist Management System.

4.5.2 Database Schema Design

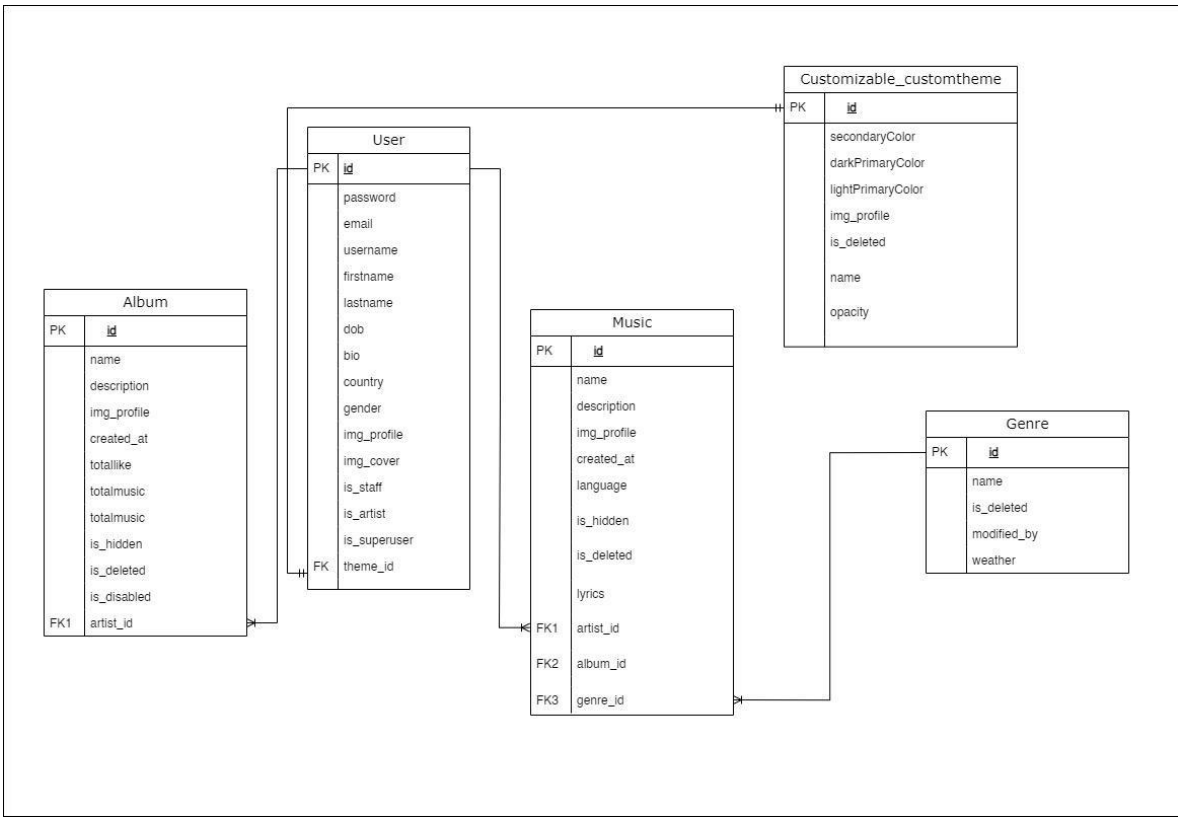


Figure 4.4: Database Design

This database is for a web application called Musica. It has tables for users, albums, music, genres, and customizable themes. The User table stores information about users like email, username, and profile images. The Album table holds details about music albums, including their name, description, and the artist who created them. The Music table contains information about individual songs, their lyrics, and the album they belongs to. The Genre table classifies the music into different categories, and the Customizable_customtheme table lets users personalize the app's appearance.

4.5.3 Class Diagram

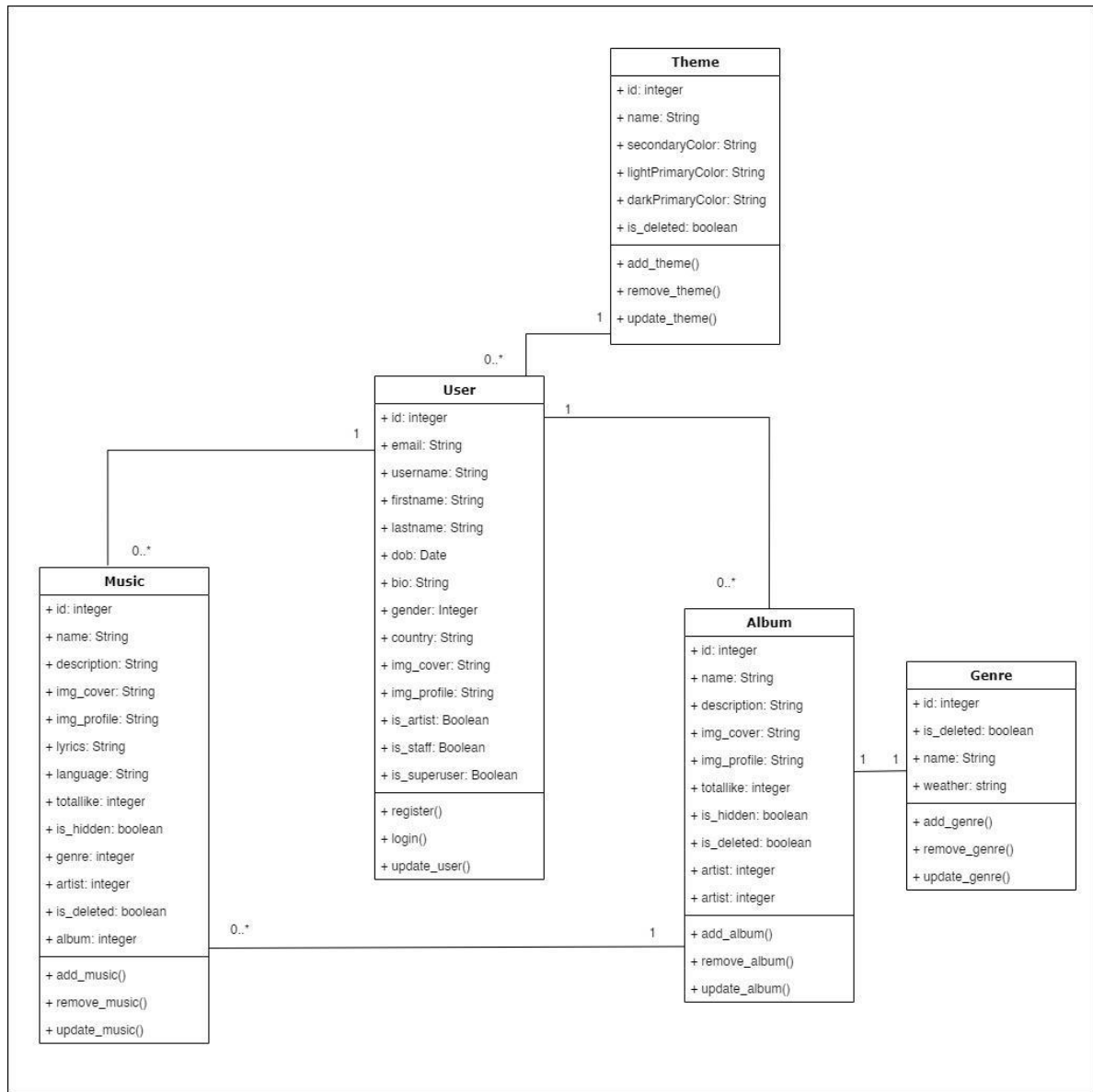


Figure 4.5: Class Diagram

This class diagram represents the core entities and the relationships between them, detailing how the system's data is structured and organized. The main entities include User, Music, Album, Genre, and Theme, each with their own attributes and methods that define their functionality and behavior.

The User entity is central to the system, representing individuals who can create, manage, and personalize their accounts. A user has various attributes such as email, username, and personal

information like first name, last name, and date of birth. Users also have roles within the system, including flags like `is_staff`, `is_artist`, and `is_superuser`, which determine the user's privileges and access to different features. Additionally, users can upload profile and cover images and customize their interface using a specific theme linked to their account. The methods associated with this class, such as `register()`, `login()`, and `update_user()`, enable user management functions, allowing users to register, authenticate, and update their profiles.

The Music entity represents individual tracks in the system, with attributes such as the name of the track, description, album cover, and metadata like the language and lyrics. Each music track is linked to a specific user (artist) and belongs to an album. Music tracks also have status attributes, such as whether they are hidden or deleted. Methods like `add_music()`, `remove_music()`, and `update_music()` manage the addition, deletion, and modification of music tracks. The Album entity is closely related to music, as each album can contain multiple tracks. An album includes attributes such as the album's name, description, and cover image, along with status flags indicating whether it is hidden or deleted. Albums are linked to users (artists) and contain methods for managing album data, such as `add_album()`, `remove_album()`, and `update_album()`.

The Genre entity allows the classification of music tracks based on their genre, with attributes like genre name and a status flag to indicate whether the genre is active or deleted. Each genre can have multiple music tracks associated with it, allowing users to organize music based on genre categories. Methods like `add_genre()`, `remove_genre()`, and `update_genre()` manage the creation, deletion, and modification of genres.

The Theme entity represents customizable themes that users can apply to personalize their interface. Attributes such as `secondaryColor`, `darkPrimaryColor`, and `lightPrimaryColor` allow users to modify the color scheme of their user interface. Users can also upload theme-related images and configure the opacity of the interface elements. The relationship between the User and Theme entities is one-to-one, meaning each user can have one personalized theme. Methods associated with this entity, like `add_theme()`, `remove_theme()`, and `update_theme()`, provide functionalities to customize and manage the user's theme.

4.5.4 Sequence Diagram

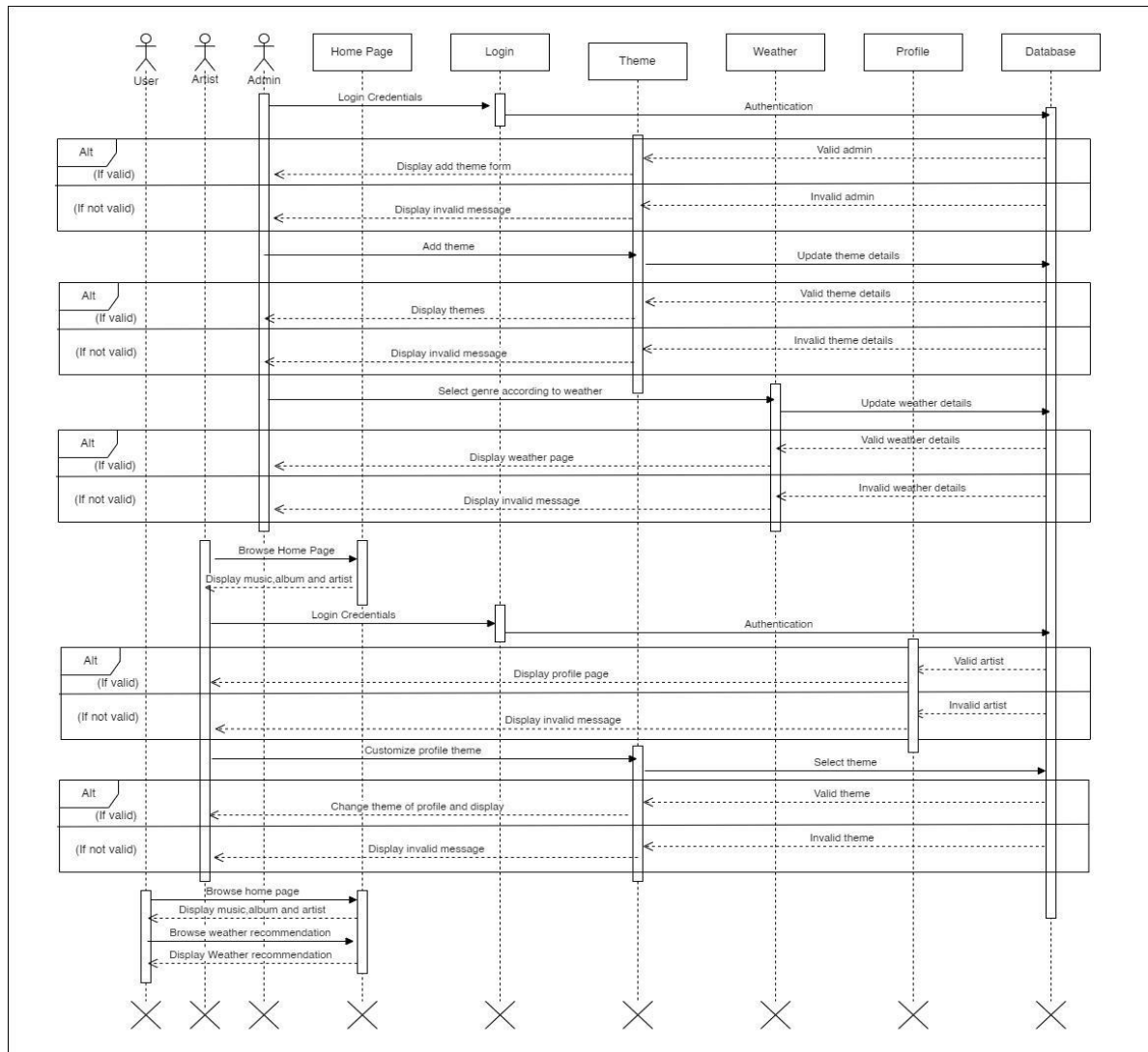


Figure 4.6: *Sequence Diagram*

The sequence diagram illustrates interactions between users (User, Artist, and Admin) and various system components (Home Page, Login, Theme, Weather, Profile, and Database). Users initiate login with credentials, triggering authentication checks. Depending on the validity of credentials, either appropriate pages are displayed or invalid messages are shown. Admins can add and update themes, which are reflected back to the user interface. Weather updates influence theme selection, displaying weather-appropriate genres. Users and artists can customize profile themes, which are verified before application. The diagram also includes alternate flows for invalid inputs, ensuring robust error handling. Additionally, the

home page supports browsing music albums and artists, with weather recommendation based on the user interface.

4.5.5 Activity Diagram

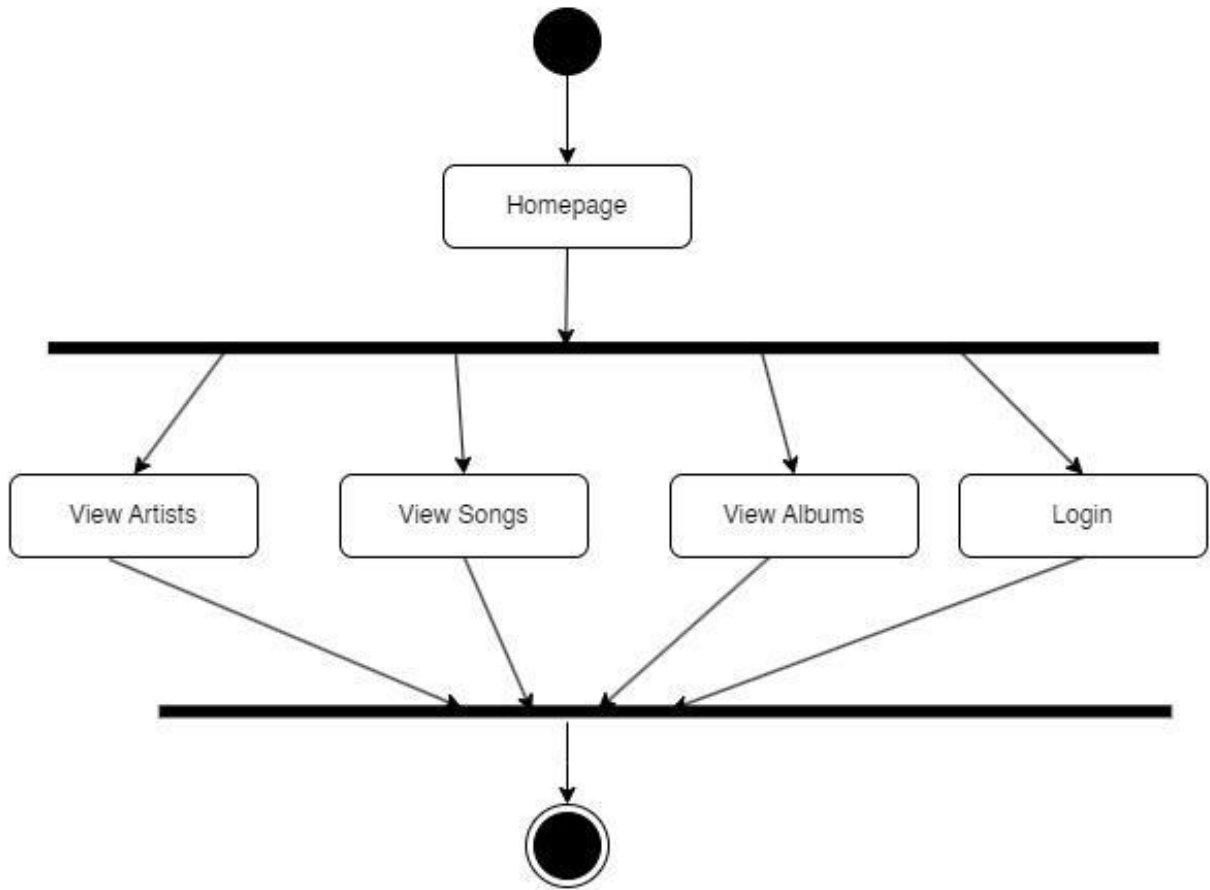


Figure 4.7: *Activity Diagram for Guest User*

The above diagram depicts the main page of a music management system, for a guest user. The initial entry point is the homepage. From the homepage, users can navigate to view artists, songs, albums, or log in. Each of these options is directly accessible from the homepage, providing a simple and straightforward navigation structure. Logging out is also an available action, signifying the end of the user's interaction with the system.

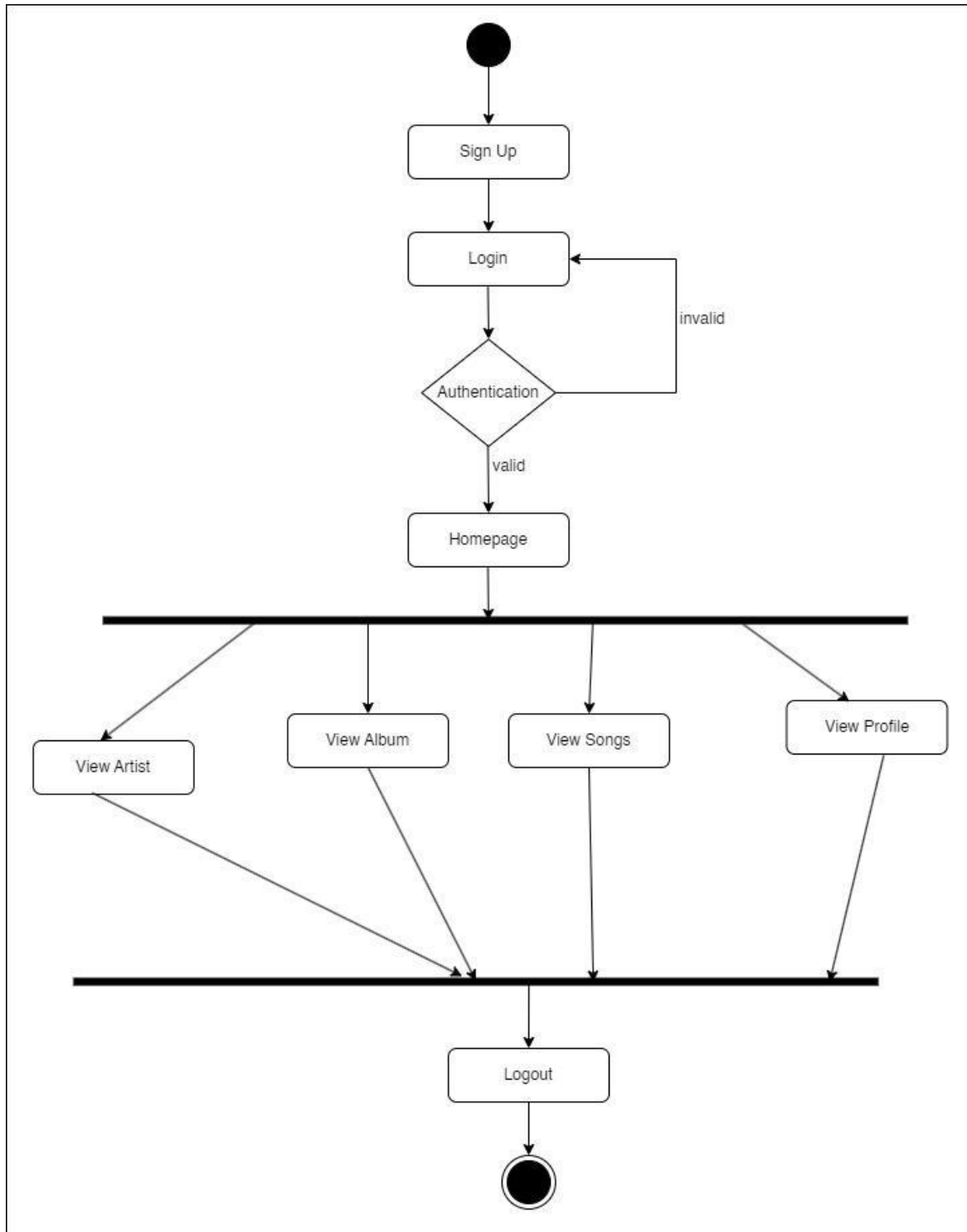


Figure 4.8: Activity Diagram for User

The user starts by signing up or logging in. If authentication is invalid, the user is redirected to the login page. Upon successful authentication, the user reaches the homepage. From there,

the artist can view artists, albums, and songs, manage their profile. Each of these actions routes back to the homepage, allowing seamless navigation. Logging out ends the user session.

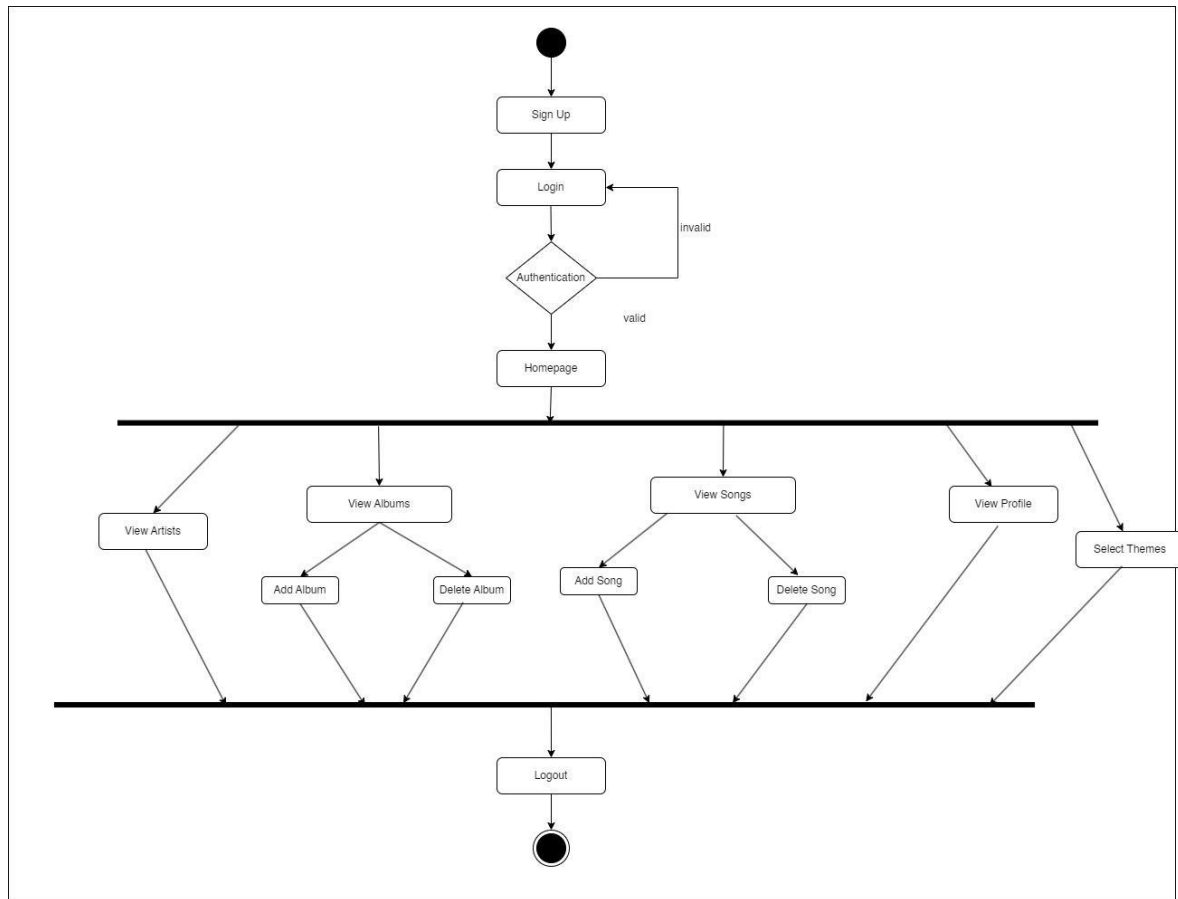


Figure 4.9: Activity Diagram for Artists

The above diagram represents the flow of an admin panel for a music management system. The process begins with the user signing in. If authentication is invalid, the user is redirected back to the sign-in page. Upon successful authentication, the user is directed to the homepage. From the homepage, the admin can perform various actions such as viewing users, enabling/disabling songs, albums, artists, and staff, as well as adding staff. Each of these actions leads back to the homepage, maintaining a centralized navigation flow. Finally, the user can log out, which terminates the session.

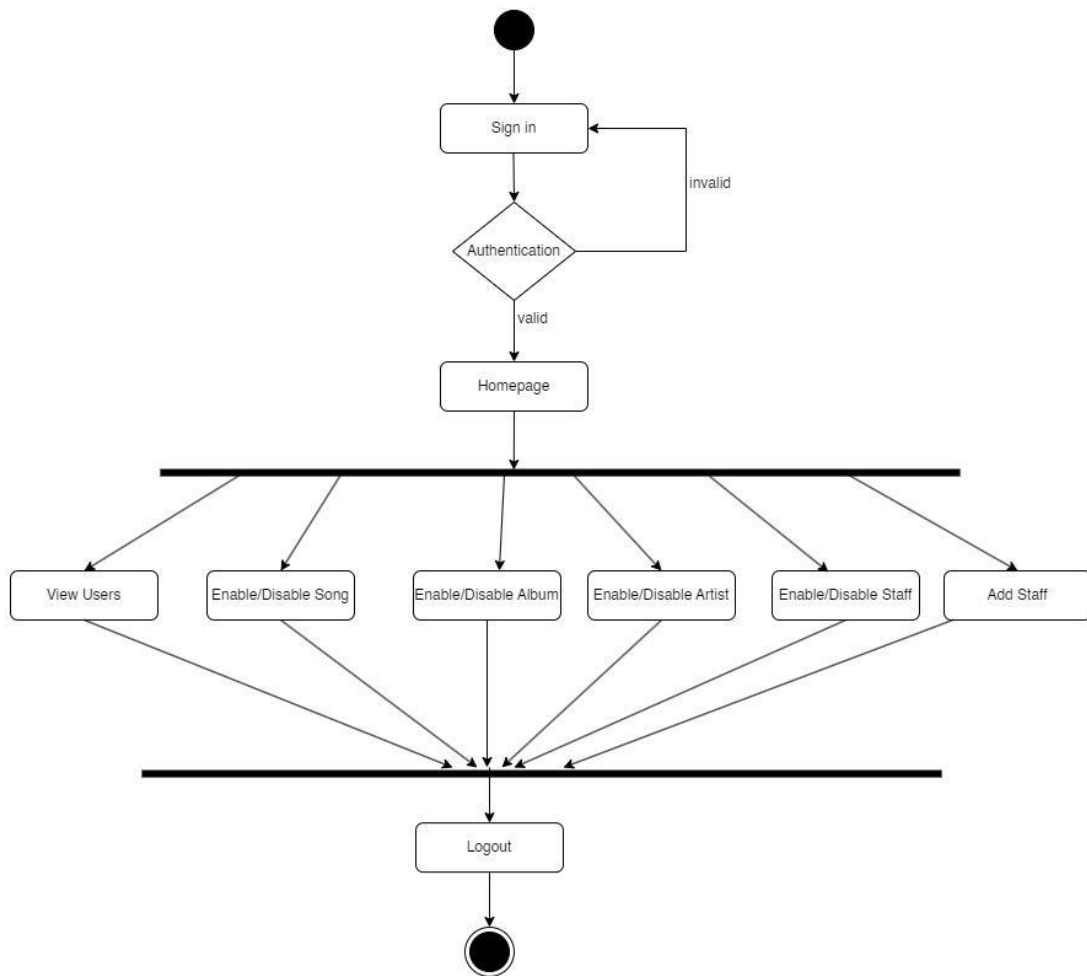


Figure 4.10: *Activity Diagram for Admin*

The above diagram represents the flow of an admin panel for a music management system. The process begins with the user signing in. If authentication is invalid, the user is redirected back to the sign-in page. Upon successful authentication, the user is directed to the homepage. From the homepage, the admin can perform various actions such as viewing users, enabling/disabling songs, albums, artists, and staff, as well as adding staff. Each of these actions leads back to the homepage, maintaining a centralized navigation flow. Finally, the user can log out, which terminates the session.

4.6 Module Description

The Musica Artist Management System is carefully designed so only authorized people can access it, making sure data stays correct and safe. Only those with permission can change artist records, keeping the data secure. The system is made to be easy for users to control, with a simple and clear interface that works smoothly. By focusing on being clear and precise, the design makes the system easy to use and good for managing all the artists' details and activities.

The modules created in our system are:

- Login Module
- Music Module
- Album Module
- User Module
- Report and Analysis Module
- Like Module
- Comment Module
- Weather-based Recommendation Module
- Theme Selection Module
- Admin Module
- Staff Module

The modules that were worked on by me include the Music Module, Album Module, Artist Module, Admin Module, Weather-based Recommendation Module, and Theme Selection Module. Each of these modules was developed to ensure they meet the specific requirements of their respective functionalities within the system.

4.6.1 Admin Module

In the Admin Module, the frontend interface includes a detailed dashboard. This dashboard shows important statistics and provides management tools for the admins. Admins can add staff members, disable users, and manage music, albums, and artists. They can also perform CRUD (Create, Read, Update, and Delete) operations for themes and genres, allowing full control over these aspects of the system.

4.6.2 Music Module

On the frontend, a full CRUD (Create, Read, Update, and Delete) interface was added. This lets artists manage their music collections easily. They can add new music, see and change details of existing music, and delete music entries. A detailed single music page was also made. It gives deep information about each music track. This page shows track details, album info, artist info, and user actions like likes and comments. On the backend, strong support for all CRUD actions was added. This makes data management smooth and works well with the frontend. It included setting up the needed database structures, APIs, and handling data checks and security.

4.6.3 Album Module

On the frontend, a full CRUD (Create, Read, Update, and Delete) interface was added. This lets artists manage their album collections easily. They can add new album, see and change details of existing albums, and delete album entries. A detailed single album page was also made. It gives deep information about each album. This page shows album details, track info, artist info, and user actions like likes and comments. On the backend, strong support for all CRUD actions was added. This makes data management smooth and works well with the frontend. It included setting up the needed database structures, APIs, and handling data checks and security.

4.6.4 Theme Selection Module

In the backend, a theme table was created, and CRUD (Create, Read, Update, Delete) operations for themes were implemented. On the frontend, both the admin and staff have the ability to perform CRUD operations on themes, including selecting colors, opacity, and background. Artists can select a theme, and the theme they choose is displayed to all other users.

4.6.5 Weather-based Recommendation Module

In the backend, a Genre table was created, and CRUD (Create, Read, Update, Delete) operations for genres were implemented with weather mapping. On the frontend, users are prompted to provide their location. Using the longitude and latitude of the given location, the

weather is retrieved through the OpenWeather API. Based on the weather, music from the genre mapped to that specific weather is recommended to the user.

4.7 Testing

4.7.1 Unit Testing

Table 4.4: *Unit Testing for Genre Creation*

Test Case ID	Description	Input Data	Expected Result	Actual Result
TC001	Create a genre with valid data	Name: Rock	Genre is created successfully	Genre is created successfully
TC002	Create a genre with duplicate name	name: Rock	Genre creation fails	Genre creation failed
TC003	Create a genre with an invalid weather type	name: Pop, weather: Sunshine	Genre creation fails	Genre creation failed

Table 4.5: *Unit Testing for Music Creation*

Test Case ID	Description	Input Data	Expected Result	Actual Result
TC001	Create a music record with valid data	name: Good4u, album: Sour, genre: pop, release_at: 2024/06/28	Music record is created successfully	Music record created successfully
TC002	Create a music record without a name	name: “ ”, album: Sour, genre: pop, release_at: 2024/06/28	Music creation fails	Music creation failed

Table 4.6: Unit Testing for Album Creation

Test Case ID	Description	Input Data	Expected Result	Actual Result
TC001	Create an album with valid data	name: Sour, description: Test album, release_at: 2024/06/28	Album is created successfully	Album created successfully
TC002	Create an album without a name	name: “ ”, description: Test album, release_at: 2024/06/28	Album creation fails	Album creation failed

Table 4.7: Unit Testing for Theme Creation

Test Case ID	Description	Input Data	Expected Result	Actual Result
TC001	Create a custom theme with valid data	name: Theme1, secondaryColor: #ff4000, darkPrimaryColor: #f6f3eb, lightPrimaryColor: #ECE6D5	CustomTheme is created successfully	CustomTheme created successfully
TC002	Create a custom theme without a name	name: (empty), secondaryColor: #ff4000, darkPrimaryColor: #f6f3eb, lightPrimaryColor: #ECE6D5	CustomTheme creation fails	CustomTheme creation failed

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Creating Musica has been a valuable experience full of learning and professional growth. The project was focused on building an advanced artist management system for the modern digital music industry. The main goal of Musica was to make artist management easier and improve user interactions with a robust web platform. The Vue.js front-end framework and Django back-end framework with SQLite were used to design a platform that is flexible, reusable, and scalable. The importance of secure user authentication and role-based access control was highlighted, allowing profiles to be managed, content to be uploaded, and interactions with fans to be facilitated, creating a lively music community. Administrative tools were also developed to provide detailed analytics and efficient management of the platform. Ensuring data privacy and secure interactions was prioritized to build user trust and confidentiality. The project involved many technical parts like front-end and back-end development, database design, user authentication, and secure data management. These efforts led to a platform that meets and exceeds the expectations of artists and music lovers, offering a smooth and enjoyable user experience. In summary, Musica is seen as a significant step forward in artist management systems, blending technological innovation with user-focused design. The potential of modern web technologies to change how audiences are connected with and how online presence is managed by artists has been demonstrated through this experience.

5.2 Outcome

Valuable insights were gained into database management, specifically in designing complex schemas using Django's ORM to ensure efficient data handling and retrieval. Security measures were implemented to protect user data and maintain platform integrity, highlighting the importance of safeguarding sensitive information. The role of analytics in monitoring system performance and user engagement was also understood, proving essential for continuous improvement. Communication skills were enhanced through collaboration with team members, mentors, and stakeholders, ensuring that all requirements and expectations were met. Various technical challenges were addressed throughout the development process,

honing problem-solving abilities and leading to innovative solutions that ensured the system's functionality and user-friendliness. Overall, the Musica project was seen as an invaluable learning journey, bridging the gap between theoretical knowledge and real-world application in web development and artist management. A comprehensive understanding of technical proficiency, project management, user-centric design, database management, security best practices, analytics, and effective collaboration was provided.

5.3 Recommendation

Based on the experience gained from developing Musica, several recommendations can be made to further enhance the platform and its capabilities:

- **Enhanced User Engagement Features:** Consider integrating more interactive features such as real-time chat or live streaming capabilities to foster deeper engagement between artists and their fans. These features could enhance the platform's appeal and create a more dynamic community atmosphere.
- **Advanced Analytics and Insights:** While basic analytics were implemented, expanding this capability to include more sophisticated data analysis and visualization tools could provide deeper insights into user behavior, content popularity, and platform performance. Tools like AI-driven recommendations based on user preferences could also enhance user satisfaction.
- **Scalability and Performance Optimization:** Evaluate the platform's scalability under heavy user load and optimize performance where necessary. This may involve further refining backend APIs, implementing caching mechanisms, or leveraging AWS services like AWS Lambda for serverless computing to handle bursts in traffic efficiently.
- **Mobile Application Development:** Consider developing a dedicated mobile application for Musica to extend its reach and accessibility. A mobile app could provide on-the-go access for users and artists, enhancing convenience and user engagement beyond the desktop experience.
- **Localization and Globalization:** Expand the platform's reach by supporting multiple languages and adapting to regional music preferences. Localization efforts can include

not only language support but also localized content curation and cultural nuances in user interactions.

- **Enhance more in Security Factor:** While security measures were prioritized, regular security audits and updates are crucial to stay ahead of evolving cybersecurity threats. Consider implementing advanced security features such as MFA and regular vulnerability assessments.
- **Integration with Third-Party Services:** Explore partnerships or integrations with third-party services such as social media platforms, music streaming services, or event management tools. These integrations could enhance platform functionality and user convenience, offering seamless experiences across different digital platforms.
- **Comprehensive User Feedback Mechanisms:** Implement structured feedback mechanisms such as user surveys, ratings, and reviews to gather insights directly from users. This feedback can guide future feature development and platform improvements, aligning Musica more closely with user expectations.

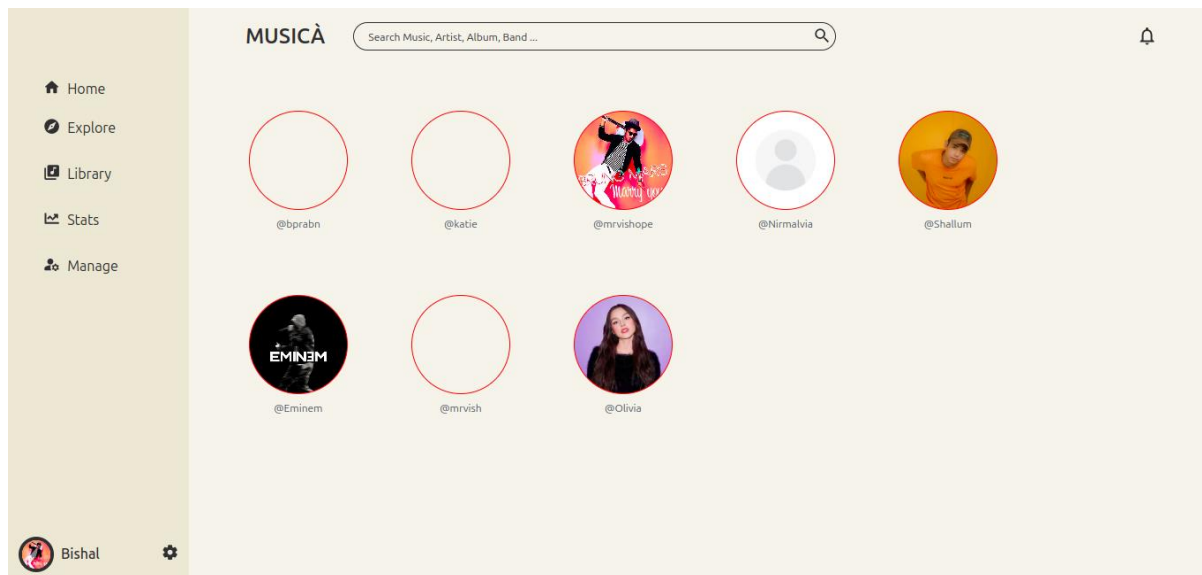
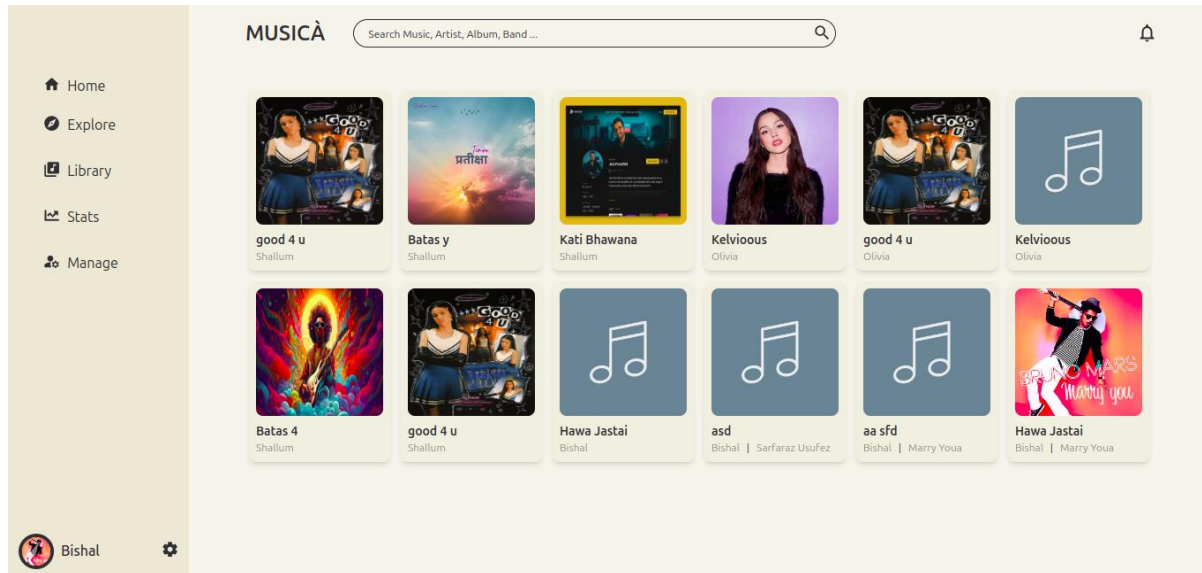
5.4 Limitations and Considerations

Due to time constraints and specific expertise limitations in certain areas, not all of these recommendations could be fully implemented during the initial development phase of Musica. However, prioritizing these enhancements based on user feedback and strategic goals can guide future iterations and continuous improvement efforts. Balancing feature expansion with maintaining platform stability and usability will be crucial for sustaining Musica's growth and impact in the competitive digital music industry landscape.

REFERENCES

- about*. (2024). Retrieved from cloco: <https://cloco.com.np/about>
- bachelors-information-management-bim-in-nepal*. (n.d.). Retrieved from studentsnepal: <https://studentsnepal.com/bachelors-information-management-bim-in-nepal/>
- Doyle, D., & Mahomar, I. V. (2024). *Artist Management & Fan Engagement*. Retrieved from https://isabellamajano.com/wp-content/uploads/2024/08/FINAL-THESIS_Artist-Management-Fan-Engagement.pdf
- Grady, J. O. (2016). *System Verification*. Academic Press.
- P., N., Nair, N. S., Alapatt, P. D., & Thanish Kurian, P. K. (2023). *SPOTLIGHT*. Retrieved from philstat: <https://philstat.org/index.php/MSEA/article/view/2118>
- Schulze, J. (2024, February 13). *What Is Information Technology?* Retrieved from Coursera: <https://www.coursera.org/articles/what-is-information-technology>
- Shklovski, I., & Boyd, D. (2006). Music as cultural glue: supporting bands and fans on myspace. *Unpublished Tech Report*, <<http://www.danah.org/papers/BandsAndFans.pdf>> (Stand: 2008) (Zugriff: 21.05. 2009). Retrieved July 11, 2024, from https://www.academia.edu/2672710/Music_as_cultural_glue_supporting_bands_and_fans_on_myspace
- Shrestha, S. (2023, May 15). *Nepal's Booming IT Sector*. Retrieved from myrepublica: <https://myrepublica.nagariknetwork.com/news/nepal-s-booming-it-sector/>
- Soft, Y. (2019, September 25). *History of IT in Nepal*. Retrieved from Medium: <https://medium.com/@ityashri/history-of-it-in-nepal-183591c5d681>
- Sugandhi, A. (2024, June 27). *What is Information Technology? Types, Services, Benefits*. Retrieved from knowledgehut: <https://www.knowledgehut.com/blog/others/what-is-information-technology>

APPENDIX



Name

Description

Released At

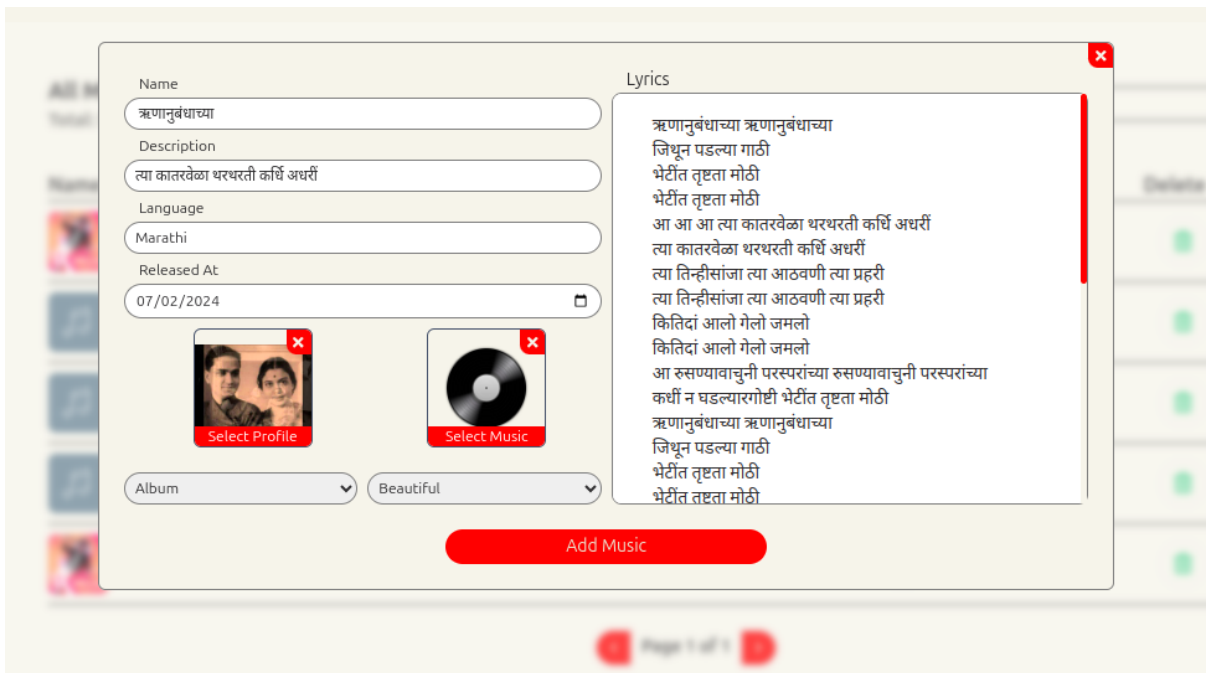
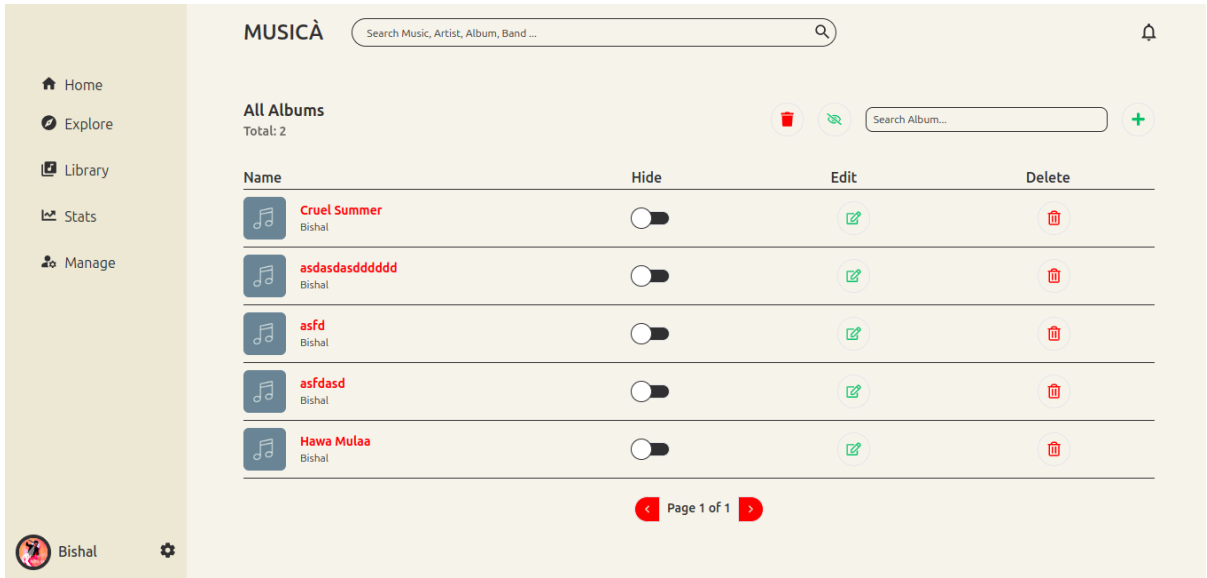
Add Album

मराठी

माझ्या सर्व जुन्या मराठी गाण्यांचा हा अल्बम आहे

07/10/2024

Album Picture



Home

Explore

Library

Stats

Manage

MUSICÀ

Search Music, Artist, Album, Band ...








SuperUser

All Theme

Total: 22

Search Theme...

+

Name	Edit	Delete
<div> Theme1 #1a1242</div>	<div></div>	<div></div>
<div> Olivia1 #c3c4fe</div>	<div></div>	<div></div>
<div> Diya #a8a8f5</div>	<div></div>	<div></div>
<div> Doodles Orange #483605</div>	<div></div>	<div></div>
<div> Default2 #f6f3eb</div>	<div></div>	<div></div>
<div> DefaultThemeasd #f6f3eb</div>	<div></div>	<div></div>
<div> asdasd #f6f3eb</div>	<div></div>	<div></div>

First Name

Bishal

Last Name

Phuyal

Bio

I am Staff

Username

staff

Email

staff@gmail.com

Date of Birth

07/08/2024

Password

Confirm Password

Gender

Male

Country

Nepal

Add Staff

Profile Picture

Cover Picture

