A Major Project Report on

Unified Sports Booking and Media System

Submitted in Partial Fulfillment of the Requirements for the

Degree of **Bachelor of Engineering in Information Technology**

under Pokhara University

Submitted by:

Pankaj Shah, 201425

Sandesh Dahal, 201437

Shrisna Thapa, 201442

Under the supervision of:

Er. Himal Acharya

Date:

26/07/2025

Department of IT Engineering

Nepal College of Information Technology

Affiliated to Pokhara University

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to [Nepal College of Information Technology] for providing us with the educational platform and resources to undertake this project. The opportunities and learning experiences at [NCIT] have been instrumental in shaping our academic growth.

A special thanks to our supervisor, [Er. Himal Acharya], for his valuable guidance, encouragement, and expertise throughout this project. His mentorship has been invaluable, and We are grateful for his support.

We would like to express our sincere gratitude to our team members, Pankaj Shah, Sandesh Dahal and Shrisna Thapa who provided unwavering support and encouragement throughout this project and also for their remarkable dedication, expertise, and collaborative spirit throughout this endeavor.

Additionally, we would like to thank our professors and advisors for their guidance and invaluable insights. Without their support and expertise, this project would not have been possible.

Lastly, we would like to thank our friends for their unwavering support, insightful discussions, and motivating presence throughout our project.

Thank you to everyone who has played a part in this project.

ABSTRACT

The Unified Sports Booking and Media System is a cross-platform web and mobile application designed to transform how sports venues, particularly game courts, are discovered, managed and experienced. Designed for three main user roles: players, venue owners and system administrators, the system offers a centralized, feature-rich solution to address the fragmented processes of venue booking, facility management, and digital engagement in the sports sector.

For players (users), the platform enables real-time court discovery, venue comparison, booking with availability checks, event participation, and access to sports-related media such as news updates and short-form video reels. Venue owners can register and manage multiple venues and courts, submit legal documentation, define court availability, organize events, and publish media content to promote engagement. Administrators are responsible for managing venue applications, approving or rejecting listings, monitoring transactions, and managing news and reels to ensure content quality and system integrity. The platform is built on a modern technology stack, with Django serving as the back-end framework, PostgreSQL for relational data handling, and Flutter or React Native for a unified front-end experience across devices. Secure online payments are integrated using Khalti, a popular Nepali digital wallet. Furthermore, Maps, Calendar, and Sign-In are utilized to enhance functionality through geolocation services, authentication, and schedule synchronization.

This project not only addresses the operational needs of venue booking and management, but also enriches community interaction through media sharing and localized sports content. It aims to create a unified digital ecosystem specifically tailored for the futsal and wider recreational sports community

Keywords: Game Courts, Venue Management, Real-Time Scheduling, News and Reels, Web and Mobile App, Khalti Integration.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CMS	Content Management System
CRUD	Create, Read, Update, Delete
DB	Database
ER	Data Flow Diagram
OSRF	Open Source Robotics Foundation
PAN	Permanent Account Number
PBAC	Role-Based Access Control
SaaS	Software as a Service
SDLC	Software Development Lifecycle
UML	Unified Modelling Language

1. INTRODUCTION

In the contemporary digital landscape, the pervasive integration of online platforms has fundamentally reshaped service delivery models and business operational paradigms. Industries spanning e-commerce, travel, healthcare, and education have universally embraced digital transformation to enhance user convenience, accessibility, and operational efficiencies. The sports and recreational sector, particularly in venue and facility management, is undergoing a similar evolutionary trajectory, driven by an escalating demand for sophisticated, user-centric digital solutions.

Traditional venue management often entails fragmented and manual processes, leading to in efficiencies in facility discovery, scheduling, payment processing, and communication. This project addresses the critical need for a centralized, intelligent system capable of overcoming these limitations. Digital booking and management systems establish a seamless interface between service providers and consumers, thereby simplifying resource discovery, reservation orchestration, secure payment execution, and comprehensive communication via unified platforms. For end-users, these platforms provide unparalleled convenience in identifying, comparing, and reserving services tailored to their specific preferences, geographical location, and temporal availability. Concurrently, for service providers, these systems offer robust functionalities for resource allocation, customer base expansion, operational optimization, and performance analytics.

Key architectural and functional components intrinsic to modern digital facility management systems typically encompass:

- **Advanced Search and Discovery Mechanisms:** Empowering users to locate services based on granular filters such as location, pricing tiers, and real-time availability, frequently augmented by geolocation services and interactive map integrations for enhanced spatial awareness.
- **Real-Time Reservation Systems:** Facilitating instantaneous viewing of up-to-date re source availability (e.g., futsal courts) and enabling immediate

reservation confirmations, significantly mitigating the reliance on error-prone manual booking systems.

- **Secure Payment Gateway Integration:** Incorporating validated and trusted payment gate ways to ensure the efficient, immutable, and secure processing of financial transactions, thus delivering a comprehensive, end-to-end user experience.
- **Comprehensive administrative dashboards:** Providing venue owners and system administrators with centralized control over booking management, dynamic pricing strategies, scheduling configurations, content dissemination (including multimedia assets), and exhaustive financial tracking capabilities.
- **Robust User Authentication and Profile Management:** Supporting secure, account based interactions that enable personalized notifications, historical transaction tracking, and granular access control mechanisms.

The application of these digital principles to specialized domains, such as multi-venue booking, remains an area with substantial untapped potential, particularly in emerging digital economies like Nepal. This project, formally titled "Unified Sports Booking and Media System," is engineered to bridge this critical gap. It proposes the development of a comprehensive, cross-platform solution that not only streamlines the booking and management of multiple futsal courts but also integrates dynamic community engagement features, including event participation registration, sports news dissemination, and the immersive experience of short-form video reels.

1.1 Problem Statement

In the current operational landscape, the process of accessing and managing futsal venues predominantly relies on outdated, manual, and inefficient methods. Users seeking to play futsal face considerable challenges in locating available courts, obtaining accurate pricing information, and completing reservations. Concurrently, venue owners grapple with labor intensive manual coordination, suboptimal resource utilization, and limited outreach to po

tential customers. Although digital booking platforms have become widely adopted across various industries, a specialized, comprehensive solution tailored to the futsal community remains absent.

Key Challenges

1. **For Users (Players):**
    - **Information Scarcity and Fragmentation:** Lack of a centralized repository listing futsal venues, including their facilities, pricing models, and real-time availability. Users frequently depend on informal channels such as word-of-mouth or direct telephonic inquiries.
    - **Inefficient Booking Processes**: Reliance on manual booking via phone calls or physical visits results in time-consuming procedures, prone to errors and delayed confirmation.
    - **Lack of Transparency**: Inconsistent or incomplete information about court conditions, exact location, pricing policies (peak/off-peak), and booking terms contribute to user dissatisfaction.
    - **Payment Inconveniences:** Predominantly cash or manual bank transfer payments complicate transaction tracking and reduce convenience.
2. **For Venue Owners:**
    - **Manual Management Overheads:** Scheduling, booking, and payment tracking handled manually is labor-intensive, error-prone (e.g., double bookings), and inefficient.
    - **Limited Market Visibility**: Inadequate digital presence restricts the ability to reach a wider customer base, leading to underutilization of venue capacity.
    - **Communication Bottlenecks:** Managing customer inquiries, booking confirmations, and cancellations through manual channels becomes overwhelming, especially during peak demand periods.
    - **Absence of Data-Driven Insights:** Lack of systematic data collection on booking trends, user preferences, and financial performance impedes strategic decision-making related to pricing, service offerings, and marketing.

Opportunities for Improvement

- **Centralized and User-Friendly Platform**: Develop a unified, online system that ag gregates futsal venue information, enabling users to search, compare, and book courts with real-time availability and transparent pricing.
- **Efficient Venue Management Dashboard**: Provide venue owners with digital tools to manage venue and court details, configure availability and pricing, track bookings, securely process online payments, and communicate effectively with users.
- **Enhanced Accessibility and Convenience:** Utilize web and mobile technologies to enable anytime, anywhere booking, integrated with secure payment gateways such as Khalti, alongside automated booking confirmations.
- **Improved Communication and Community Engagement**: Facilitate streamlined interactions between players and venue owners for booking modifications, inquiries, and event or promotional organization.

System Functionalities

For Venue Owners

- **Venue Registration & Application**
  - Venue owners initiate registration via the Venue Application module by submitting comprehensive applications.
  - Mandatory legal documents including PAN Card, Government Approvals, and Business Licenses are uploaded through the Venue Application Document entity.
  - System administrators review applications, assigning statuses: Pending, Under Review, Approved, or Rejected, while maintaining audit logs with admin remarks and timestamps.

- **Venue Management**

- Venue profiles (Venue model) can be created and updated, containing detailed meta data such as:
- Name, Address, City, Geographical Coordinates (Latitude, Longitude), Contact Phone, Email, Description, and Operating Hours (Opening and Closing times).
- Venue images are managed via the Venue Images repository.
- Venue visibility is controllable through activation or deactivation features.

- **Court Management**
  - Each venue supports multiple courts defined under the Court model with configurable attributes :
    - Surface Type (Artificial Grass, Concrete, Wood), Capacity, Hourly Pricing.
  - Courts can be activated or deactivated dynamically.
  - Court-specific images are uploaded and managed in the Court Images collection.
- **Availability Scheduling**
  - Court availability is scheduled through the Availability model, specifying open time slots.
  - Special pricing can be assigned for predefined time intervals (e.g., peak hours).
  - Data integrity is enforced by unique constraints preventing double bookings for the same court, date, and time range.
- **Secure Payment Integration**
  - The system stores Khalti payment gateway secret keys (Online Payment Khalti Secret Key) linked securely to individual venues to facilitate authenticated and secure transactions.
- **Event Organization & Media Management**
  - Venue owners can organize and manage futsal events to promote community engagement.

- Full CRUD (Create, Read, Update, Delete) capabilities are available for publishing and managing sports news and video reels, enhancing promotional outreach.

For Customers / Users

- **Search & Browse Venues**
  - Users can query and browse venue information including name, city, description, and multimedia content.
  - Detailed court data is presented, highlighting surface type, capacity, and hourly rates.
- **Real-Time Availability & Booking**
  - Real-time court availability is displayed, allowing users to select preferred time slots seamlessly.
  - The booking system facilitates court reservations with date/time selection, payment method choice, and provision for special booking notes.
  - Users receive booking status updates (e.g., pending, confirmed, cancelled) tracked by the system.
- **Booking History & Event Participation**
  - Users can review historical and active bookings stored in the Booking model, including details such as price, status, payment confirmation, and notes.
  - Registration for venue-organized events is supported.
- **Media Consumption**
  - Access to news articles and video reels fosters community interaction and keeps users informed and engaged.
- **Payment Handling**
  - The system supports multiple payment methods(managed by the Payment Type model).
  - Payment transactions are securely recorded in the Payment Transaction model, logging:

- Transaction IDs, payment status (referenced via the Status model), and raw payment gateway responses for audit and reconciliation purposes.

1.2 Project Objectives

The primary objectives of the Unified Sports Booking and Media System are:

- To develop a secure, scalable web and mobile platform that enables real-time discovery, booking, and management of sports courts, integrating Khalti payments and AI-powered features for enhance user experience and operational efficiency.
- To provide venue owners with advanced tools for booking automation, dynamic pricing, event organization, and content management, including the creation and sharing of sports news and video reels to engage the community

1.3 Significance of the Study

The development of the Unified Sports Booking and Media System holds significant relevance in today's digital era, especially within Nepal's growing sports and recreational sector. This study contributes to both technological advancement and societal impact in several key ways:

- **Improving Operational Efficiency:** By automating the court booking and venue management process, the system reduces manual workload and errors, benefiting both users and venue operators.
- **Encouraging Active Lifestyles:** Simplified access to sports facilities encourages greater participation in physical activities, promoting health and well-being across communities.
- **Empowering Local Businesses:** The platform supports small and medium-sized sports venues by improving service delivery, visibility, and revenue generation, thus contributing to local economic growth.

- **Demonstrating Digital Innovation in Niche Markets:** The project showcases how technology can be tailored to address the unique challenges of recreational industries, offering a scalable model for digital transformation in other underserved sectors.
- **Enhancing User Experience:** With real-time availability, media integration, and efficient booking workflows, the system sets a new standard for customer convenience in the sports domain.

Overall, this study provides valuable insights into how modern IT solutions can bridge operational gaps and create new opportunities in emerging markets like Nepal.

1.4 Scope and Limitations

1.4.1 Scope

The project scope includes a complete digital solution for booking and managing sports venues, particularly futsal courts, in Nepal. The platform will support core functionalities such as user registration (manual and Google Sign-In), venue search by name, city, or location, real-time court availability, and online bookings with Khalti integration. Users can view venue details including pricing, images, and court specifications, as well as manage their profiles and booking history.

For venue owners, the system provides a management dashboard to update venue details, configure multiple courts with specific attributes, set pricing schedules, manage bookings, and track payments. Admin and super admin roles will handle venue approvals, user oversight, and system monitoring. Additional features include map integration for location-based searches and secure handling of sensitive data. The platform will be developed as a responsive web application, with the potential for mobile optimization in future phases.

1.4.2 Limitations

- The initial release will focus on futsal venues within a specific geographic region or a limited number of partner venues to ensure system stability and gather early user feedback.
- Advanced features such as player social networking, tournament management tools, detailed analytics dashboards, and integration with multiple payment gateways (beyond Khalti) are excluded from the initial phase and may be considered in future iterations.
- The system's real-time availability accuracy depends on venue owners consistently updating their schedules.
- Native mobile applications (iOS/Android) will not be developed in the primary phase; the focus remains on a mobile-responsive web platform.
- The project does not include hardware provisions for venue owners or direct on-site setup support, apart from platform usage guidance.

2. LITERATURE REVIEW

Online booking and reservation systems have become integral to service delivery across numerous sectors, fundamentally changing how consumers interact with service providers and how businesses manage their resources. The rise of the internet and mobile technology has propelled the adoption of these systems, offering unparalleled convenience and efficiency. Specifically, in the realm of recreational activities and facility management, these platforms address the need for quick access to information and streamlined booking processes. Common techniques and features underpinning these systems include real-time availability checking, which allows users to see open slots instantaneously, and secure on line payment integration, which simplifies transactions. User account management provides personalized experiences and booking history, while robust search and filtering capabilities (e.g., by location, price, time, facility type) enable users to find services that match their specific criteria. For service providers, these systems offer venue management dashboards that facilitate scheduling, pricing adjustments, customer communication, and basic report ing, thereby optimizing resource utilization and reducing administrative burdens [1][2].

Early online reservation systems were often bespoke solutions for larger enterprises. However, the evolution of cloud computing and Software-as-a-Service (SaaS) models has made sophisticated booking and management tools accessible to smaller businesses, including those managing sports facilities like futsal courts. These systems aim to minimize manual intervention, reduce the likelihood of errors such as double bookings, and enhance customer satisfaction through self-service portals [1]. Despite the advantages, challenges persist, including ensuring the accuracy and real-time synchronization of availability data, encouraging user adoption (especially in markets transitioning from traditional booking methods), maintaining secure payment processing, and developing platforms flexible enough to cater to the diverse operational needs of different venues [3].

The academic discourse around such systems often highlights usability, security, and scalability as critical factors for successful implementation and adoption [2][4].

2.1 Existing Systems

Several platforms globally and locally offer functionalities relevant to sports facility booking, including futsal. Notable examples include:

- **Playo:** Playo is a comprehensive sports community application that allows users to find sports partners ("playpals"), discover and book sports venues (including badminton, football, cricket, tennis, and potentially futsal courts depending on local listings), and join sports-related activities. Key features include meeting playpals based on skill and interest, hosting matches, venue discovery with details on amenities and pricing, flexible booking options (hourly, daily passes, memberships), easy payments, and real-time alerts. It aims to create a holistic sports ecosystem, connecting players, venues, and even coaches [5]. While it caters to a broad range of sports, its venue booking and community features are highly relevant to the futsal booking context.
- **Futsal House (Nepal):** Specifically targeting the futsal community in Nepal (Kathmandu Valley), Futsal House positions itself as a matchmaking app for futsal players. While its primary focus, as described, is on helping players find other players or teams and schedule games, it also mentions features like booking futsal courts and team management. This indicates a localized effort to address the organizational challenges within the futsal scene, including court reservations. It emphasizes ease of coordination and connecting with like-minded players [6][7].
- **Pitch booking**: This is a global software solution designed for sports facility management, catering to clubs, schools, councils, and commercial hubs. It provides tools for online booking, payments, facility management, and even organizing events and leagues. Features include game-changing analytics, simplified scheduling via a comprehensive calendar, subscription options for block bookings, and intuitive configuration tools for pricing and timings [8].
- **Bookteq (formerly Playfinder/BookingsPlus**): An established sports facility book ing software in the UK, Bookteq offers a marketplace for users to find and book sports facilities and a comprehensive management system for venue

operators. Features include online bookings with secure payments, a customer portal for self-service, approval workflows, detailed reporting on revenue and utilization, flexible facility con figurations to prevent double bookings, and multi-venue management capabilities[2].

- **Mind body:** While not specific to sports courts, Mind body is a widely used business management software for wellness services like fitness studios, salons, and spas. It offers robust class and appointment scheduling, client management, integrated payment processing, marketing tools, and mobile apps for both businesses and consumers [9][10]. Its success in managing time-slot-based bookings for services provides valuable insights into building effective, scalable reservation systems with a strong focus on user experience and business management tools.

These examples illustrate a spectrum of approaches, from community-focused player match making with booking features (Futsal House, Playo) to comprehensive facility management and booking engines (Pitchbooking, Bookteq), and specialized service scheduling platforms (Mindbody). They collectively highlight the demand for digital solutions that simplify access to recreational facilities and streamline their administration [1][2][3].

3. TECHNICAL DESCRIPTION

The Unified Sports Booking and Media System is a modern, cross-platform application designed to simplify the management and access of futsal venues, court reservations, and sports media content. Built using scalable, modular, and performance-oriented technologies, the system serves multiple user types customers, venue owners, and administrators through a single unified platform. The system integrates AI-powered media recommendations, secure online payments, and real-time scheduling to offer a seamless sports booking experience.

3.1 Frontend Development

The frontend is developed using **Next.js**, a powerful React-based framework that supports server-side rendering and static generation. This ensures fast load times, optimized performance, and a smoother user experience. The web app is designed to be fully responsive and also serves as the base for the hybrid mobile application using Capacitor.js.

- The interface dynamically adapts for desktop, tablet, and mobile devices, providing role-based dashboards for admins, owners, and users.
- Features include venue and court search, dynamic filters, booking forms, and an integrated media feed for news and reels.

The frontend is also responsible for consuming REST APIs and rendering real-time data such as booking availability, payment status, and venue analytics.

3.2 Backend Development

The backend system is built on **Django**, a high-level Python web framework that promotes rapid development and clean code architecture. It uses **Django REST Framework (DRF)** to expose APIs consumed by the frontend and mobile apps.

- It handles user authentication, booking logic, venue application workflows, event management, and payment verification.

- A modular structure supports separation of concerns for components such as users, payments, media, and AI recommendations.

The backend also includes role-based permissions, token management, and concurrency control to prevent double bookings and maintain data integrity.

3.3 Mobile Application

The mobile application is developed using **Capacitor.js**, which allows the responsive web application (built with Next.js) to be compiled into a native-like app using **Android Studio**. This saved time and effort by avoiding separate codebases for Android and iOS.

- Users can book courts, view availability, manage profiles, and interact with news or reels directly from the app.
- Push notifications alert users of booking confirmations, event updates, and promotional content in real-time.

All mobile interactions are supported by the same backend APIs, ensuring consistent data across platforms.

3.4 Database Management

The system uses **MySQL** as the primary database, accessed through Django's Object-Relational Mapping (ORM). The schema is normalized to handle complex relationships between users, venues, courts, bookings, and media content.

- **Redis** is integrated for caching frequent queries, such as booking availability and media feeds, to improve response time.
- Data integrity is enforced with unique constraints (e.g., no double-booking for the same court, date, and time).

This design ensures high availability, fast reads, and efficient handling of structured and semi-structured data.

3.5 APIs & Integration

The application architecture follows a RESTful API model. All modules, web frontend, mobile app, and admin dashboard communicate with the backend through secured REST APIs developed in DRF.

- **Khalti Payment Gateway** is integrated to enable secure online payments. Payments are verified using transaction IDs, and logs are stored for audit purposes.
- Email and optional WhatsApp/SMS notifications are triggered on key events like successful booking, event reminders, or account registration.

The system also includes **Map Integration** using geo-coordinates for venue discovery and future support for location-based search.

3.6 Security Features

Security is a foundational pillar of the system. All sensitive operations are protected using multiple layers of authentication and access control.

- **JWT (JSON Web Tokens)** are used for session management and API access.
- **Role-Based Access Control (RBAC)** defines permissions for different user types (Admin, Owner, Customer).
- Encryption is applied to API keys (like Khalti), and HTTPS is enforced across the platform.

Additional security includes OTP verification, CSRF protection, and UUID-based primary keys to prevent data enumeration.

3.7 Testing and Deployment

The project followed a structured four-phase incremental model, where each phase was tested thoroughly before moving to the next. Both automated and manual testing strategies were used.

- **Unit and Integration Testing** was done using tools like PyTest and Postman for APIs.
- UI testing and responsiveness were validated using Cypress and live Android emulators.
- Deployment was managed via **Docker** containers, making the system portable and ready for cloud environments. Continuous testing and feedback were incorporated throughout the development lifecycle.

3.8 Software Model

The system was developed using the **Incremental Software Development Model**, where each iteration delivered a usable component of the system. This model helped in incorporating feedback early and managing the growing complexity of the platform.

- Iteration I focused on backend and admin modules.
- Iteration II introduced venue and customer panels.
- Iteration III optimized UI/UX and tested APIs.
- Iteration IV added the AI-based recommendation system, mobile app, and Khalti integration.

This model allowed better quality control, improved team collaboration, and efficient delivery of project milestones.

3.9 Location-Based Venue Discovery

The Unified Sports Booking and News Media System implements the Haversine formula to enable location-based sports venue discovery. This feature allows users to find and book nearby sports facilities based on their current geographical location.

3.9.1 Haversine Formula Theory

The Haversine formula calculates the shortest distance between two points on the surface of a sphere (Earth). It's particularly useful for geographic applications because it accounts for Earth's curvature, providing accurate distance measurements for location-based services.

3.9.2 Why Haversine?

Unlike simple Euclidean distance calculations that work on flat surfaces, the Haversine formula considers:

- Earth's spherical geometry: Accounts for the curved surface of the planet.
- Great-circle distance: Calculates the shortest path between two points on a sphere.
- Geographic accuracy: Provides precise distances for navigation and proximity calculations.

Mathematical Foundation

The formula derives from the haversine function:

$$\text{haversin}(\theta) = \sin^2(\theta/2)$$

Complete Formula:

$$d = 2R \times \arcsin(\sqrt{(\sin^2(\Delta\varphi/2) + \cos(\varphi_1) \times \cos(\varphi_2) \times \sin^2(\Delta\lambda/2))})$$

Where: R = Earth's radius (6,371 km)

φ_1, φ_2 = latitude of points 1 and 2 (in radians)

$\Delta\varphi$ = difference in latitudes

$\Delta\lambda$ = difference in longitudes

d = great-circle distance

Feature Integration

- **Homepage- Nearest Value**
  - Component: **components/home/nearest-venues.tsx**
  - Function: Displays closest sports venues with distances
  - User Action: Click "Find Venues Near Me"

- **Venues Page - Location Sorting**
  - Component: **app/venues/page.tsx**
  - Function: Sort venues by proximity to user
  - Display: Distance badges on venue cards

3.9.3 Benefits

- Enhanced Discovery: Users quickly find nearby sports facilities

- Improved Bookings: Distance context increases booking likelihood
- Better UX: Location-aware venue recommendations
- Geographic Context: Meaningful spatial relationships in venue selection

3.9.4 Limitations

- Spherical Earth Assumption: Uses simplified spherical model instead of Earth's actual ellipsoidal shape, causing ~0.5% distance error for long distances
- Location Permission Dependency: Feature requires user geolocation permission and GPS/network connectivity; unavailable if access is denied or services are offline
- Straight-Line Distance Only: Calculates direct geography distance, not actual travel routes or time considering roads, traffic, or transportation methods

4. TOOL AND TECHNOLOGIES

Tools	Uses
Django	Django serves as the backend framework.
Django REST Framework	DRF is used to create RESTful APIs in Django.
Machine Learning Libraries: Numpy, Pandas,Scikit-Learn	To create recommendation system for media.
MySQL	Django's ORM interacts with the MySQL database to store and retrieve data.
Capacitor js	For mobile application development.
Redis DB	Used for caching and performance improvement.
Payment Gateways	For secure and seamless online payment processing.
Figma, EdrawMax, LucidChart, draw.io	Designing graphs and diagrams.

Table 1: Technologies

5. TEAM MEMMBERS AND DIVIDED ROLES

Name	Role	Purpose/Responsibilities
Pankaj Shah	Backend Developer	Responsible for developing and managing the server-side logic, handling API integration, and ensuring backend performance and stability.
Sandesh Dahal	Frontend Developer	Focused on designing and implementing responsive user interfaces and integrating them with backend services to ensure a seamless user experience.
Shrisna Thapa	Design, Diagrams, Testing, Documentation	Created various design and technical diagrams (UML, ER, etc.), conducted testing, and prepared end user documentation.

Table 2: Team Members and Divided Role

6. METHODOLOGY

6.1 Development Methodology Framework

In order to develop an application or to carry out any project successfully, there is a need for a particular methodology which enables us to manage the development process systematically and efficiently. Such methodology works as a backbone to the development process that manages the efficiency of the developers and their effort. The framework we will be using for developing this Unified Sports Booking and Media System project is the Incremental model.

The Incremental model is a process of software development where requirements are broken down into multiple standalone modules of the software development cycle. Incremental development is done in steps from analysis, design, implementation, testing/verification, and maintenance. Each iteration passes through the requirements, design, coding, and testing phases. Each subsequent release of the system adds functions to the previous release until all designed functionality has been implemented.

The incremental model typically consists of the following phases:

- **Requirements Analysis:** Initial requirements are gathered and analyzed to identify the core functionalities to be delivered in the first increment.
- **Design**: Designing the architecture and structure of the system, focusing on the components needed for the first increment.
- **Implementation:** Developing the system incrementally, starting with the core functionalities identified in the requirements analysis.
- **Testing:** Each increment undergoes testing to ensure that the implemented functionalities meet the specified requirements and are free of defects.
- **Deployment:** Deploying the increment into the operational environment, where users can interact with it and provide feedback.
- **Evaluation:** Gathering feedback from stakeholders and users to evaluate the implemented increment and identify areas for improvement.

- **Incremental Iterations:** The process repeats for subsequent increments, with each iteration building upon the previous ones to add more functionalities and enhancements.

Advantages of Incremental model:

- This model is more flexible– it is less costly to change scope and requirements during the development lifecycle.
- It is easier to test and debug during a smaller iteration.
- Working software is produced earlier and in increments during the software life cycle, so users can provide feedback.
- Customers are able to respond to each built increment.
- Initial product delivery is faster and costs less.
- Easy to integrate front-end and back-end systems incrementally.

Disadvantages of Incremental model:

- Needs good planning and design at the outset to ensure that the increments can be integrated smoothly.
- Problems might arise related to system architecture if not all requirements are collected upfront for the entire system.
- The total cost of the complete system might be higher than that of the waterfall model if not managed properly.
- Well-defined module interfaces are required, as some are developed long before others are developed.

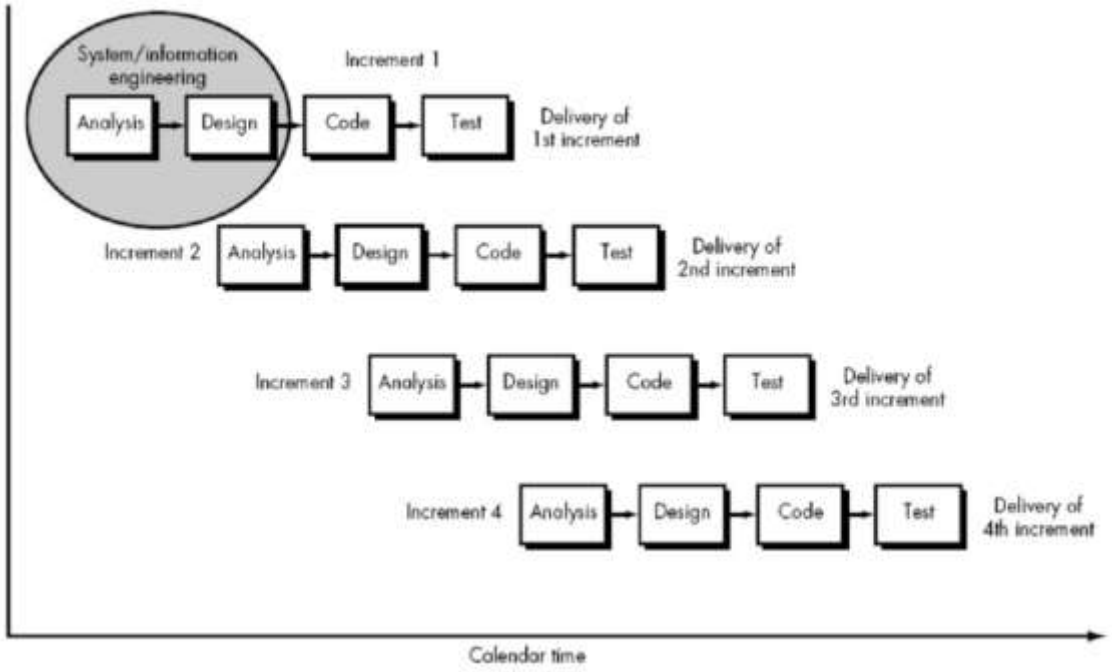


Figure 1: Incremental Model

7. SYSTEM DESIGN AND ARCHITECTURE

7.1 Architecture Pattern Overview

The Unified Sports Booking and Media System adopts a multi-tier modular architecture, structured around four key layer: Frontend Interface, Backend Service Layer, Database Layer and External Services. The design emphasizes clear separation of concerns, scalability, maintainability and performance. Each layer is independently testable and interacts through well-defined RESTful APIs.

This architecture enables real-time operations such as court bookings, venue management, secure payments, and media streaming, while also providing administrative workflows and AI-based content personalization.

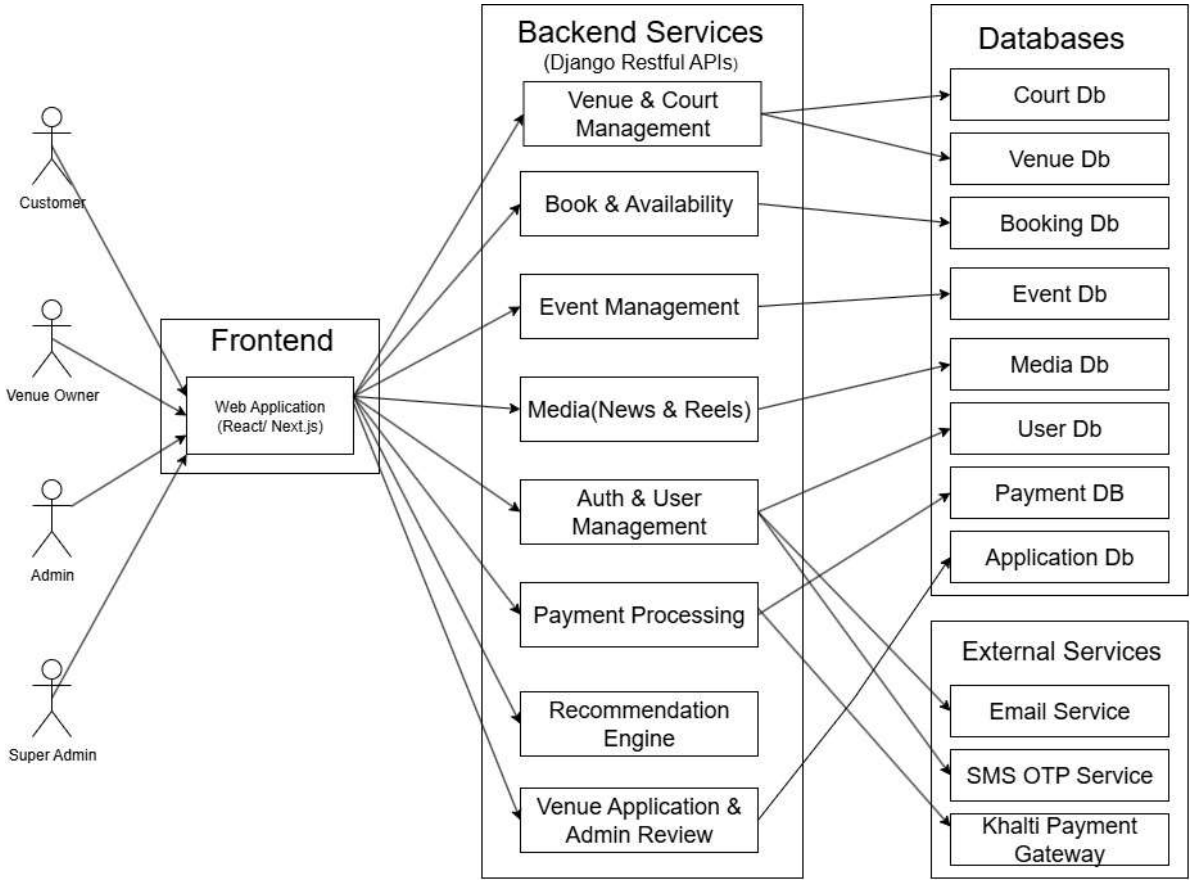


Figure 2: System Architecture Diagram

7.2 Use Case Diagram

The Use Case Diagram demonstrates the primary interactions between different user roles and system functionality. The actors of our system are: User, Venue Owner, Admin and Super admin.

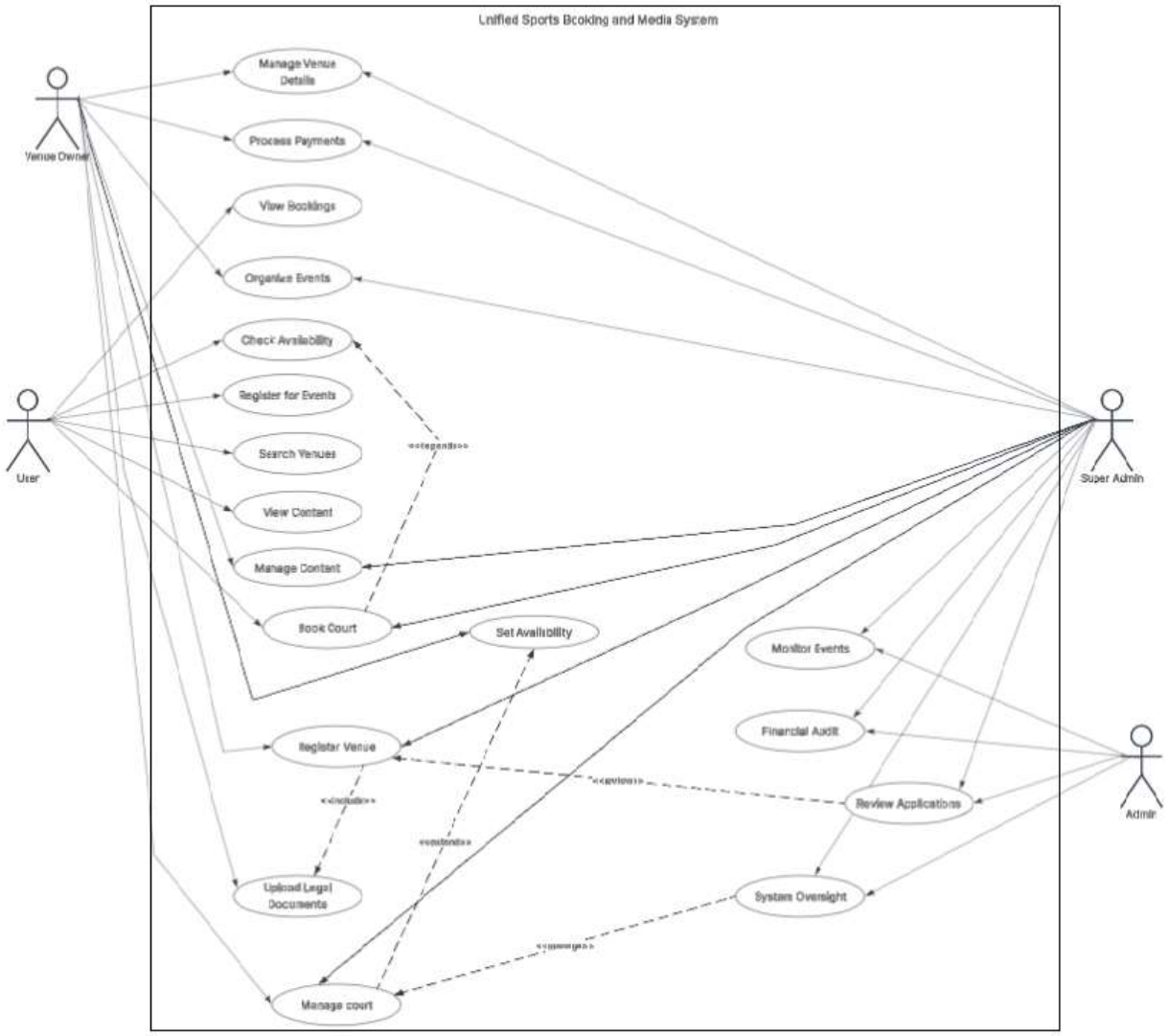


Figure 3: Use Case Diagram

7.3 Sequence Diagram

Sequence diagram (SD) is an interaction diagram that describes how operations are carried out. It shows object interactions arranged in time sequence. It also 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. Sequence Diagram sometimes called Event Diagram or Event Scenarios. We have designed sequence diagrams for most critical and influential activities which are shown below:

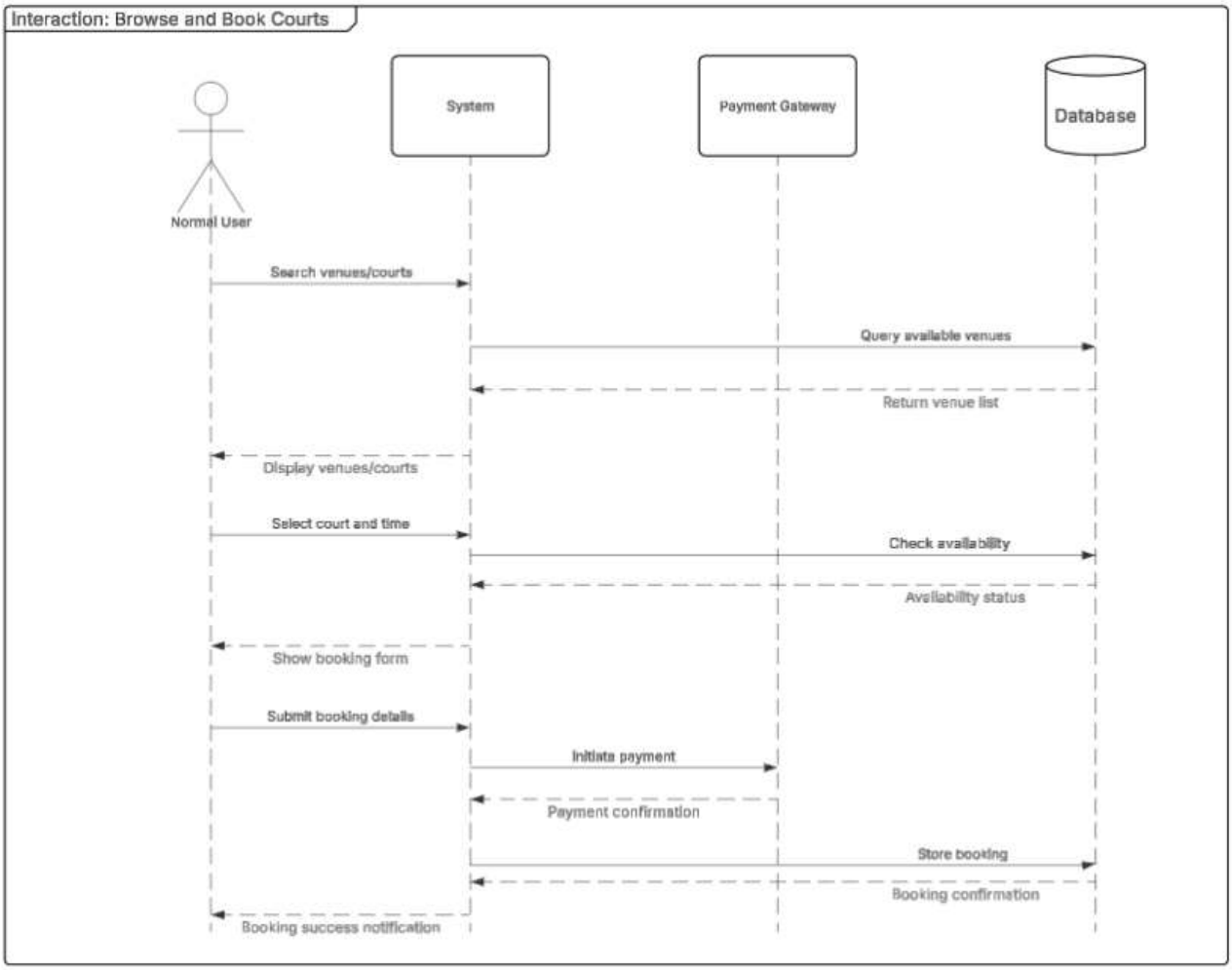


Figure 4: Sequence Diagram For Normal User (Browse and Book Courts)

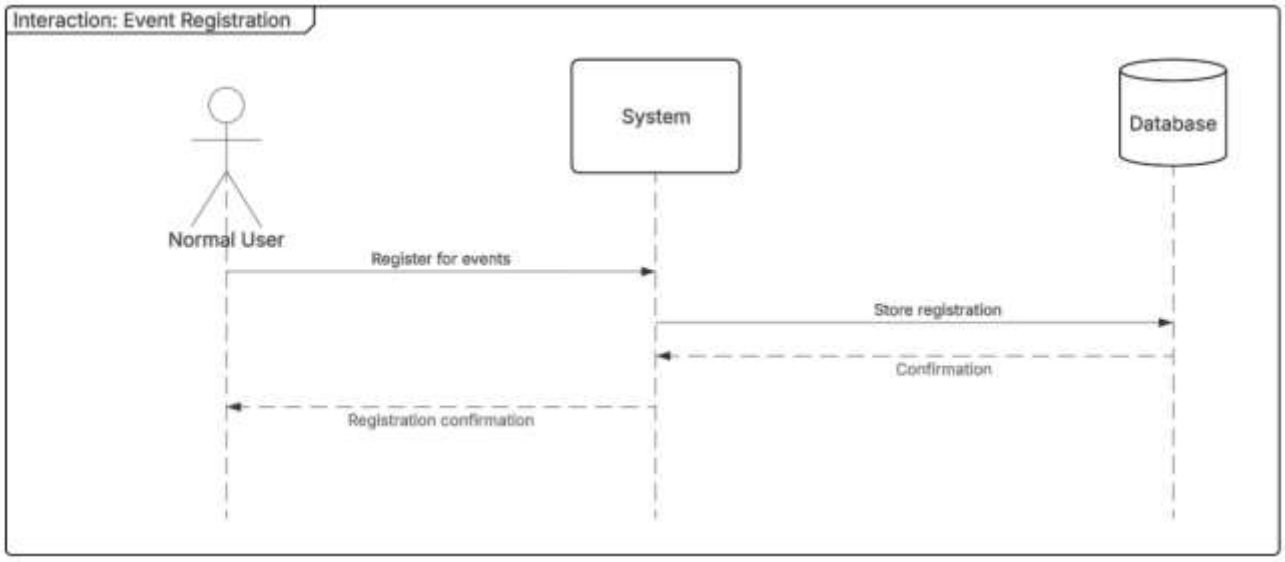


Figure 5: Sequence Diagram For Normal User (Event Registration)

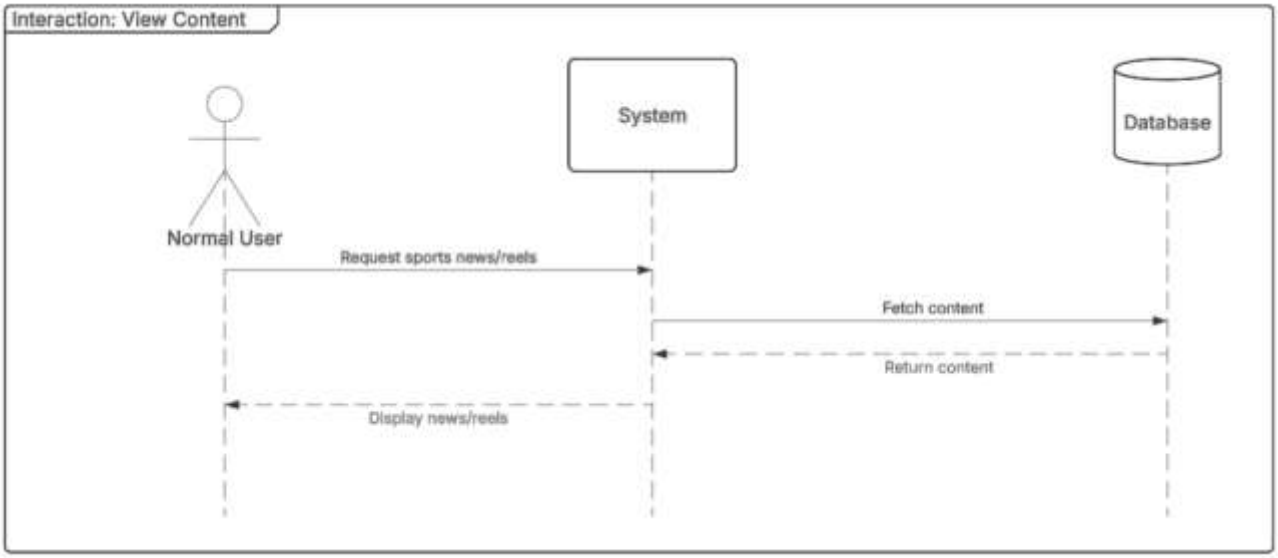


Figure 6: Sequence Diagram For Normal User (View Content)

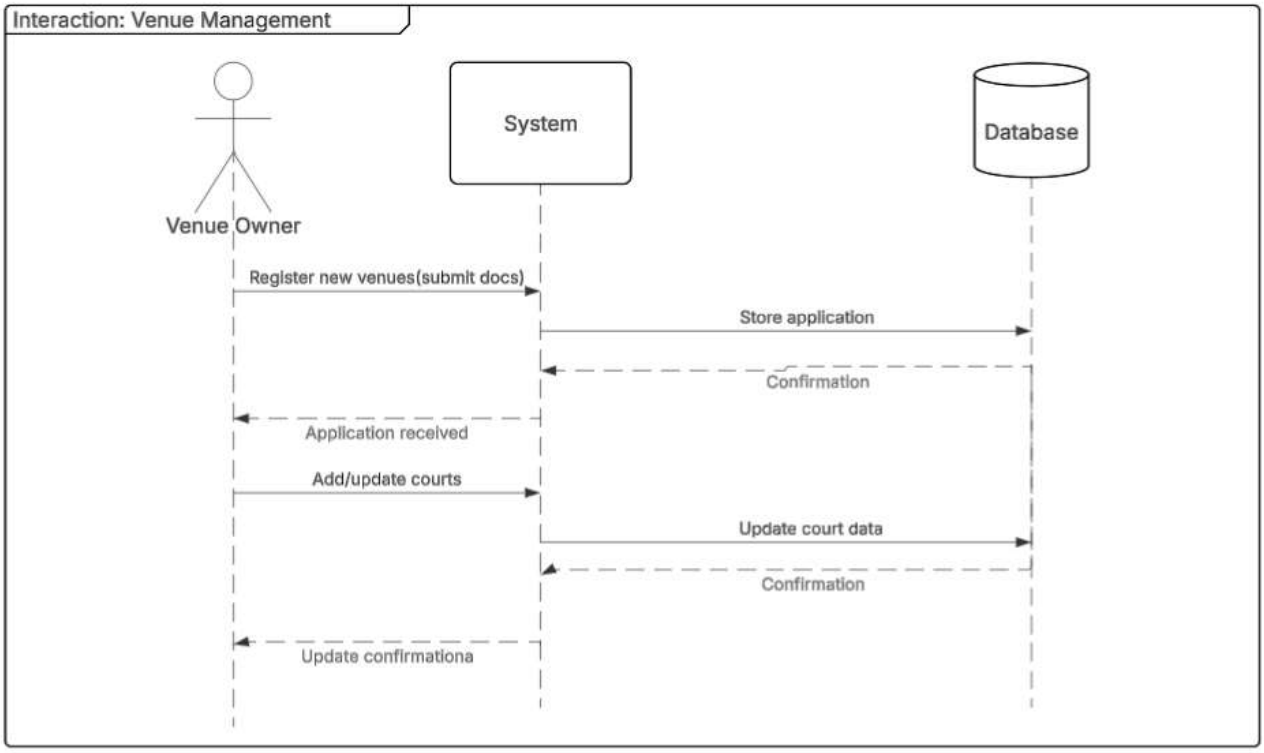


Figure 7: Sequence Diagram For Venue Owner (Venue Management)

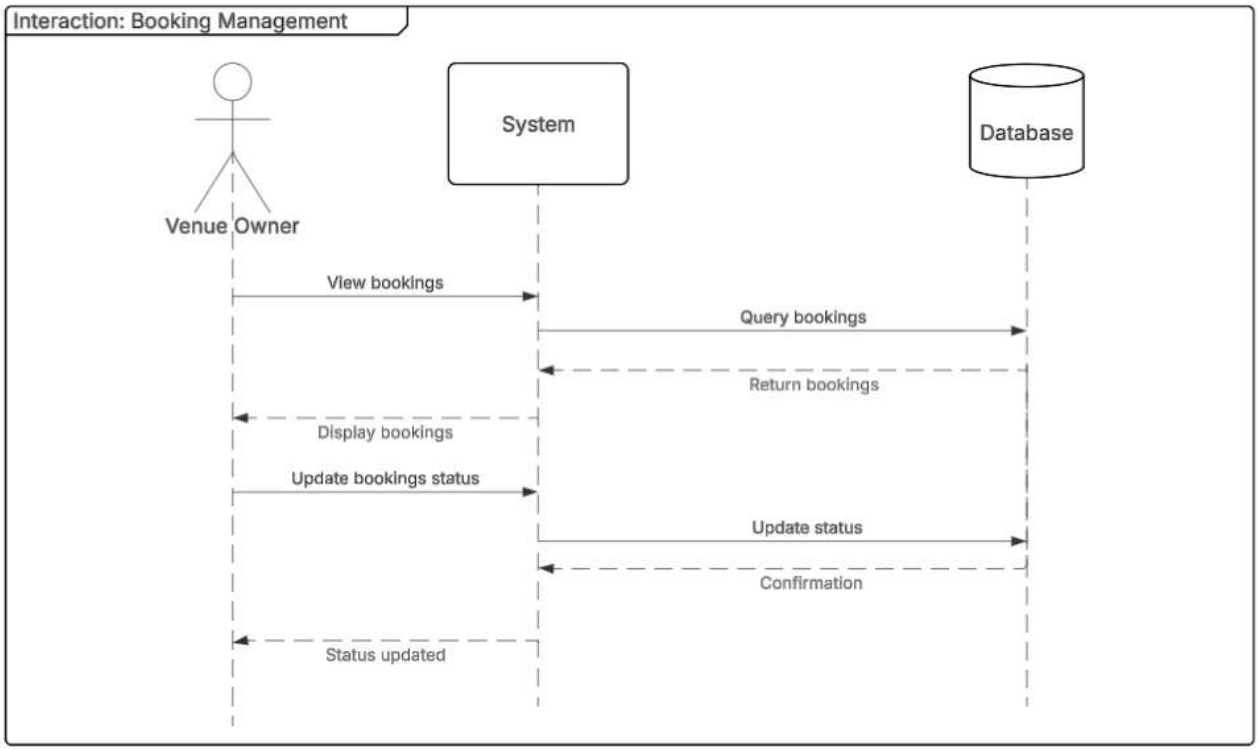


Figure 8: Sequence Diagram For Venue Owner (Booking Management)

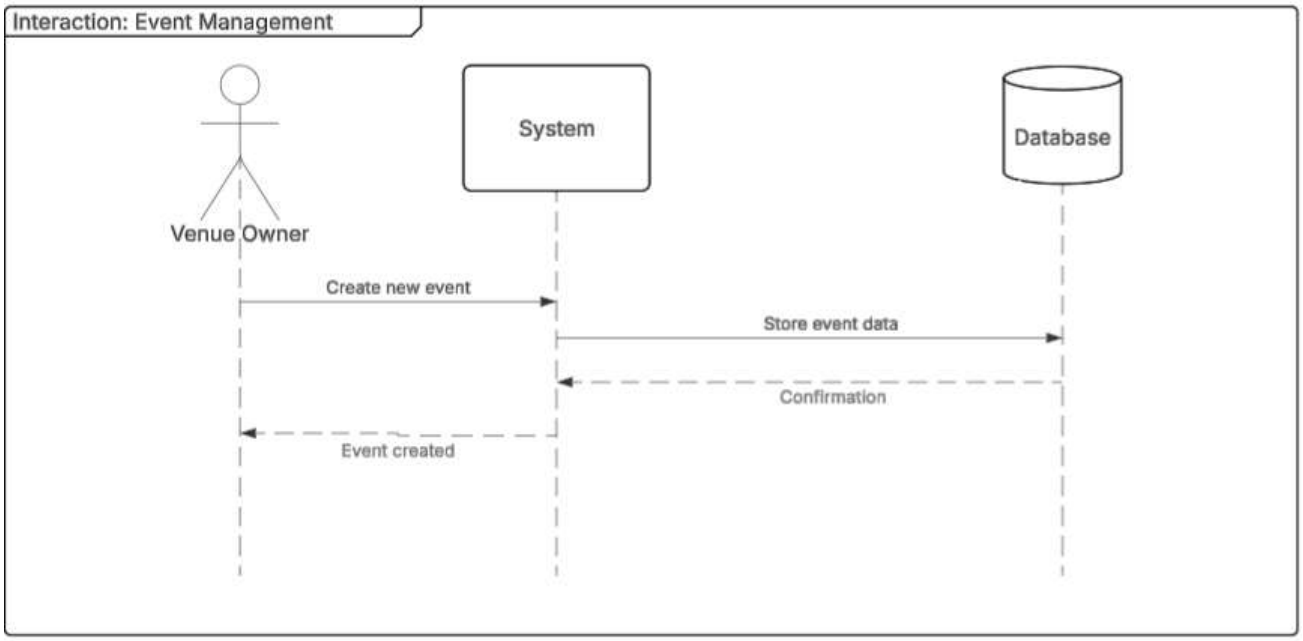


Figure 9: Sequence Diagram For Venue Owner (Event Management)

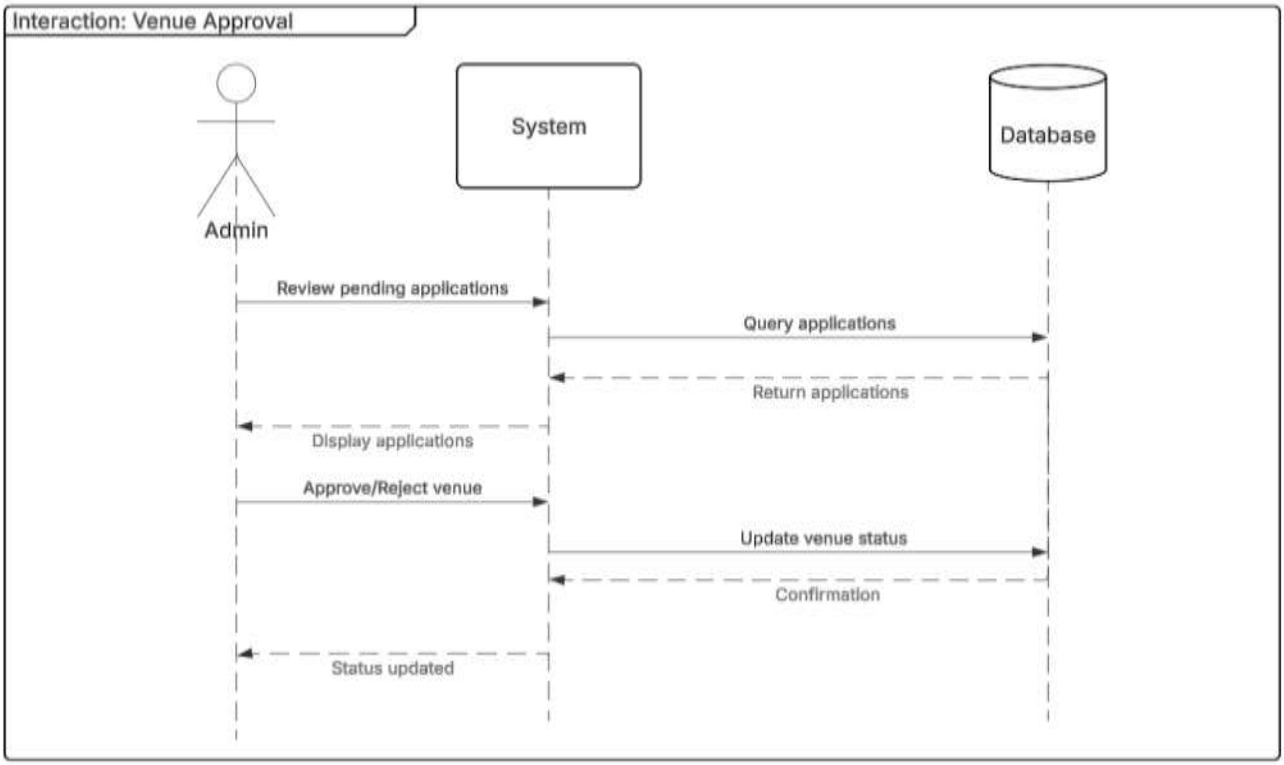


Figure 10: Sequence Diagram For Admin (venue Approval)

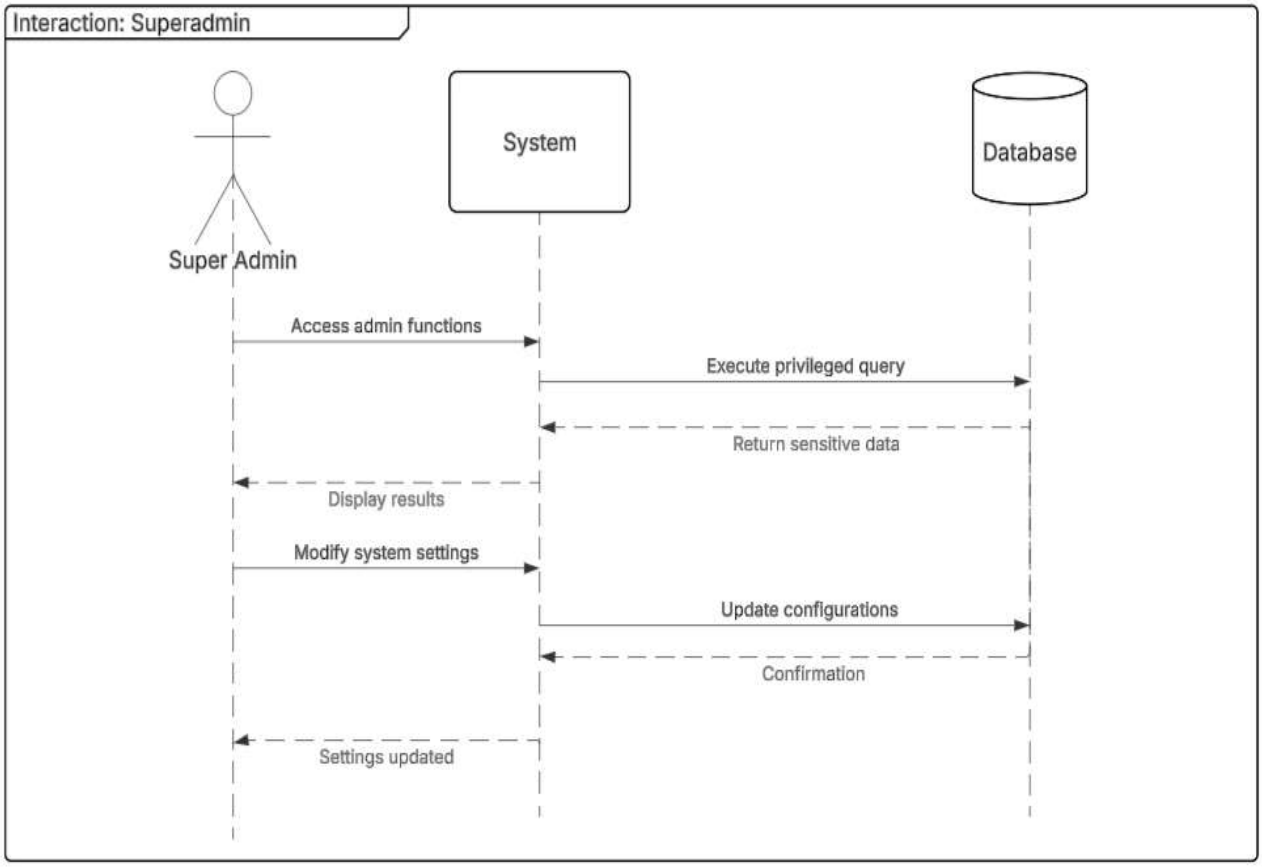


Figure 11: Sequence Diagram For Super Admin

7.4 ER Diagram

The ER Diagram is a pictorial representation of system's entities and relationship between those entities. It also illustrates the overall logical structure of the system's database. The entities are their attributes are represented in the rectangle, and the attributes that are underlined are primary keys.

The ER Diagram of our system is shown below:

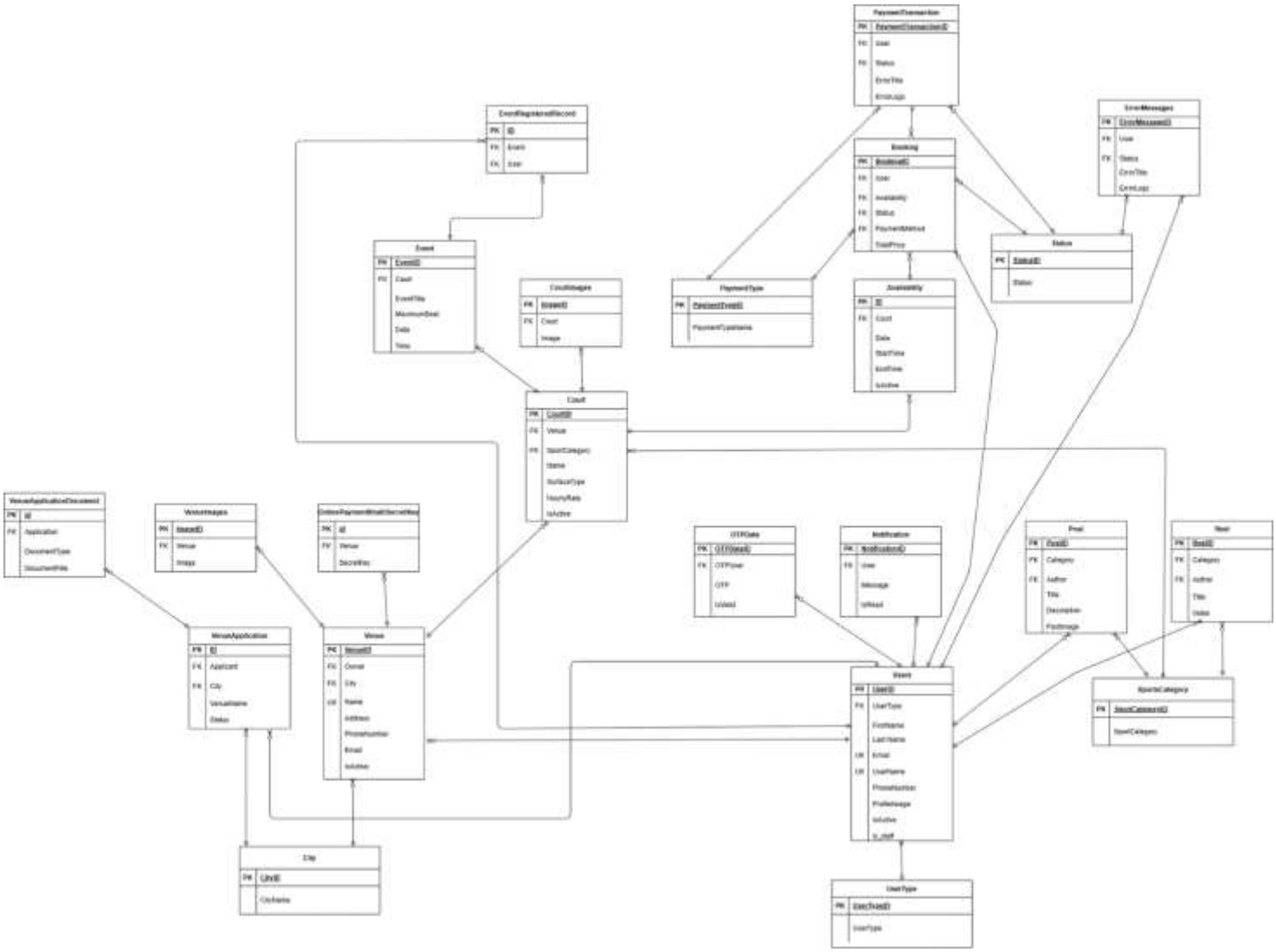


Figure 12: ER Diagram

7.5 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

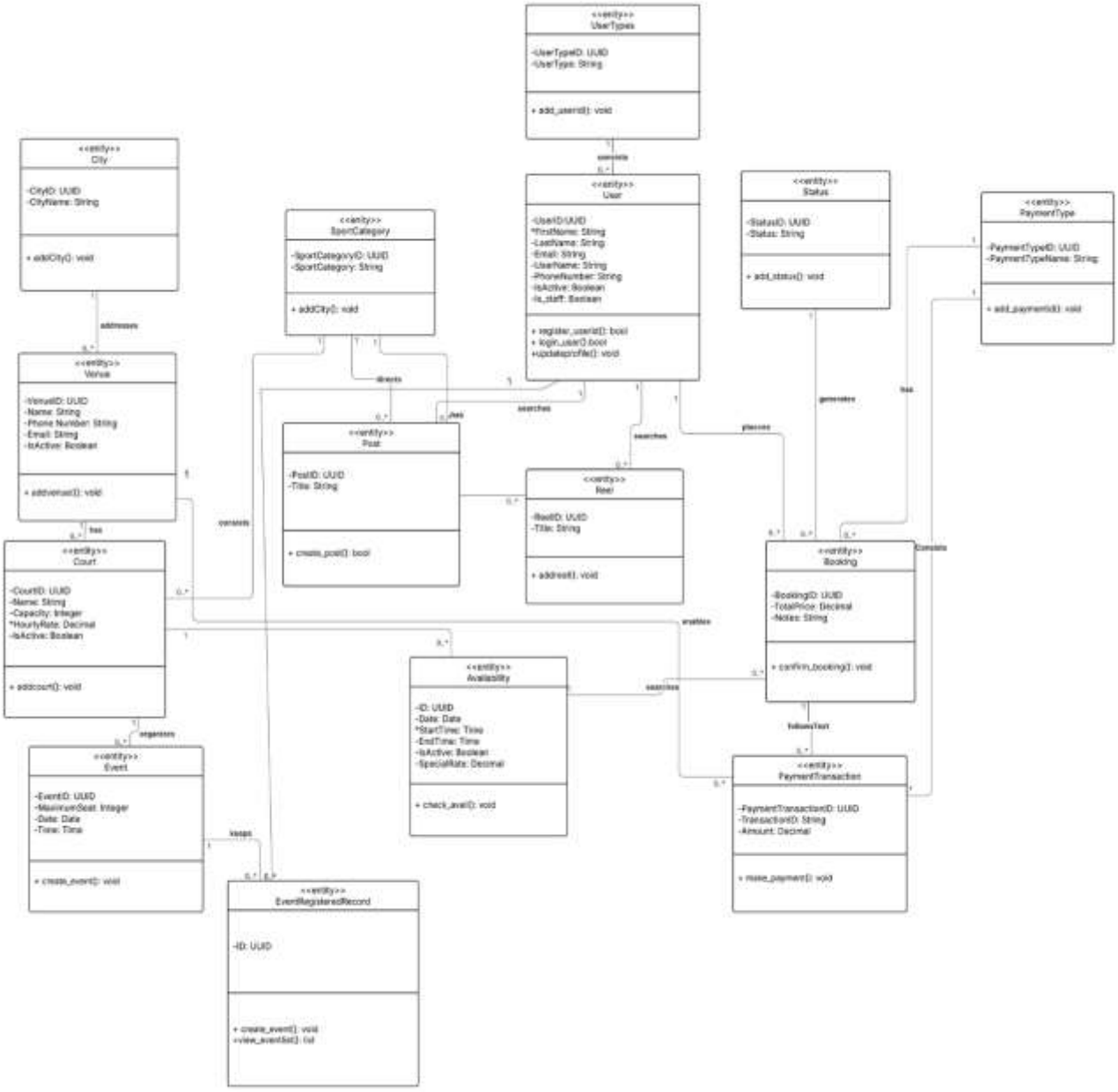


Figure 13: Class Diagram

7.6 Collaboration Diagram

Collaboration Diagram shows the interaction between objects and their relationships during a specific process. It focuses on how messages are passed between components to complete task.

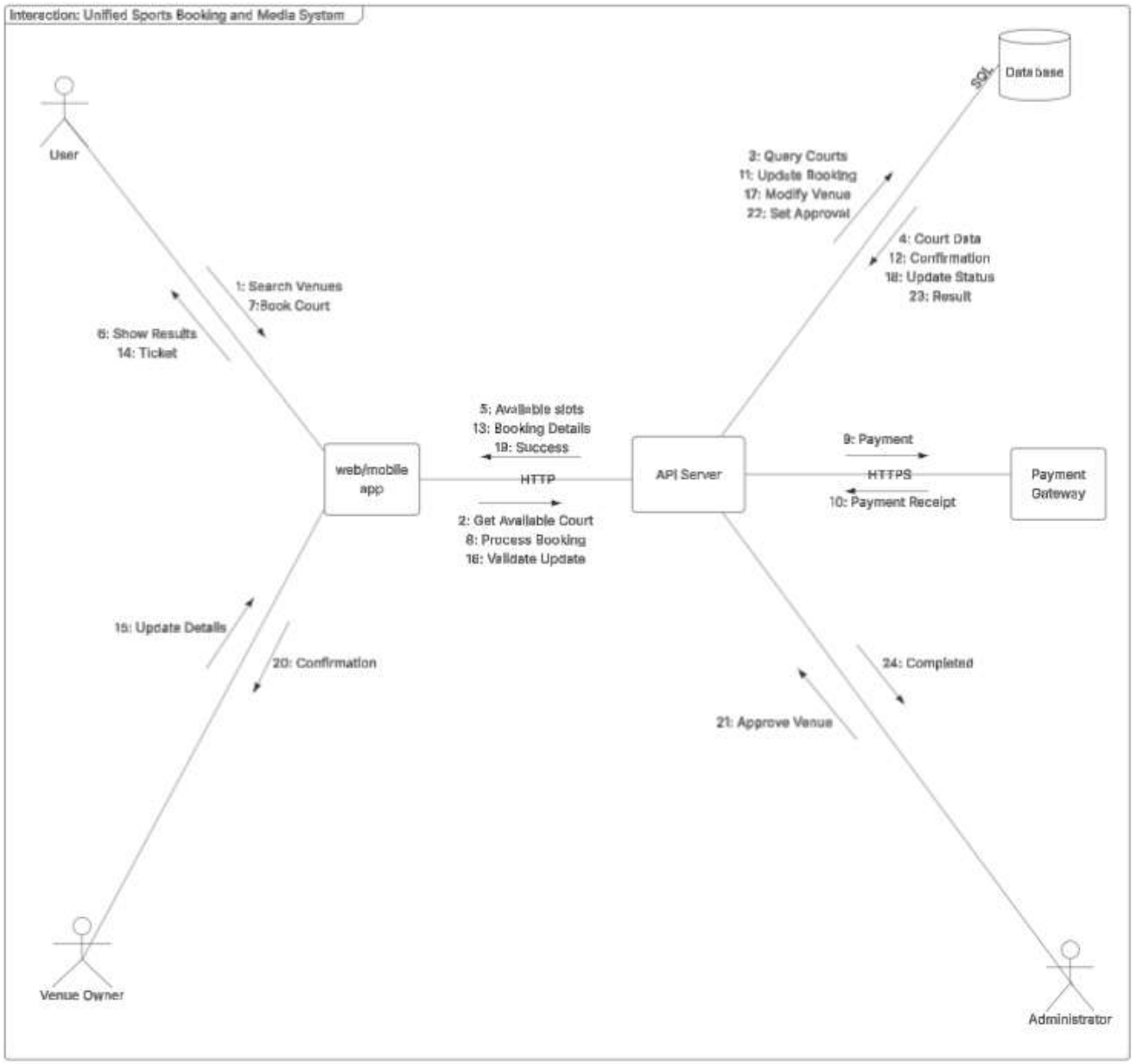


Figure 14: Collaboration Diagram

7.7 Context Diagram

Context Diagram is a high-level visual representation that illustrates a system's boundaries and its interactions with external entities such as users, services, or other systems. It shows the system as a single process and highlights the flow of data between the system and its environment.

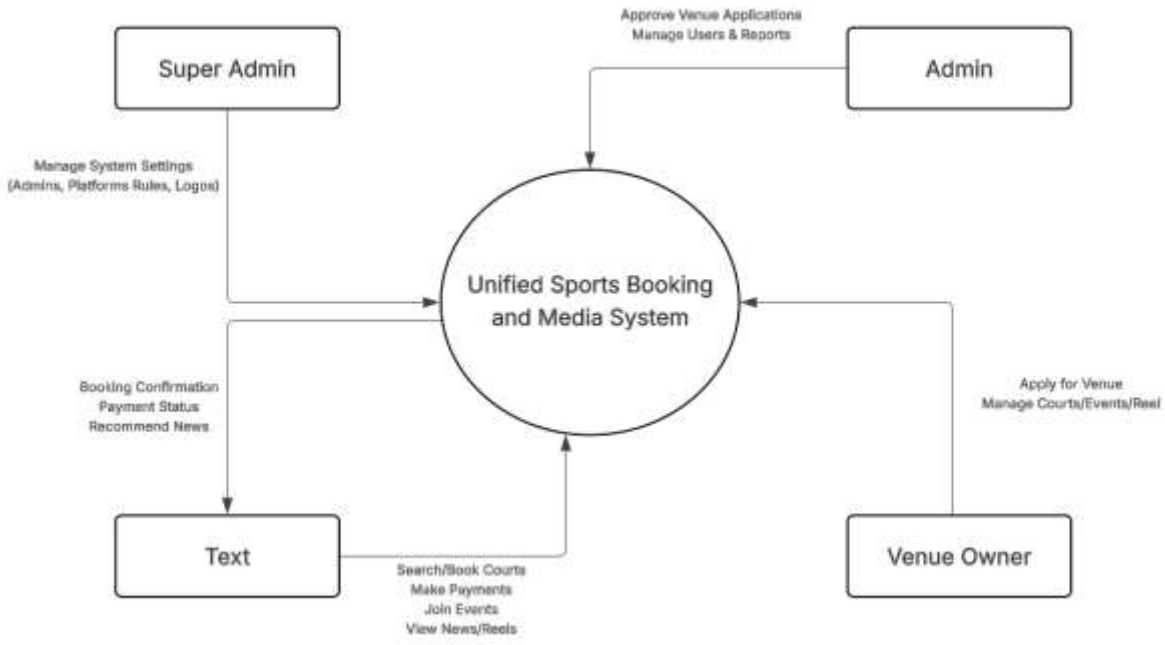


Figure 16: Context Diagram

7.8 Domain Diagram

A domain model is a conceptual model of the domain that incorporates both behavior and data. It is a formal representation of a knowledge domain with concepts, roles, datatypes, individuals, and rules, typically grounded in a description logic.

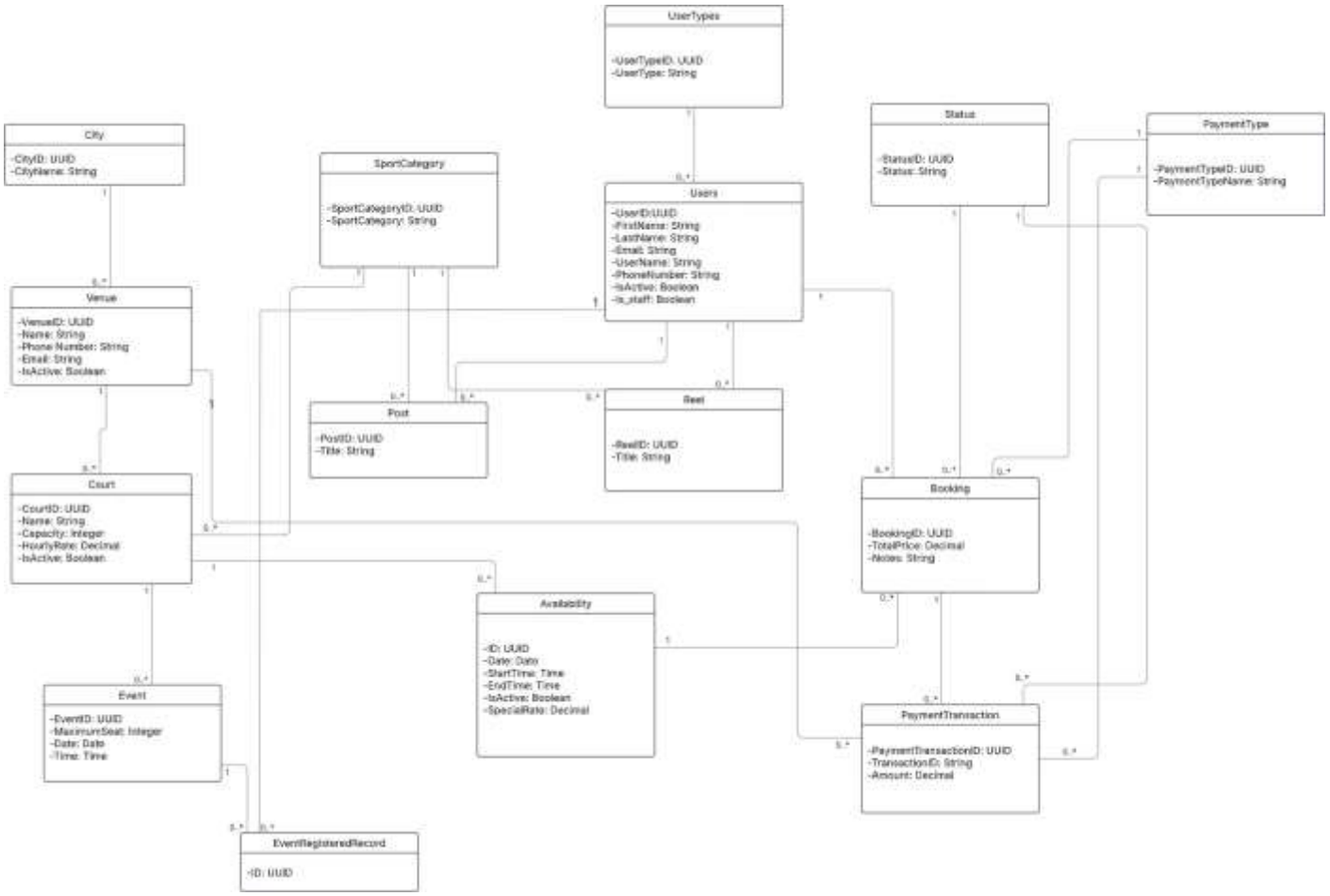


Figure 15: Domain Diagram

8. Requirement Analysis

Requirement analysis is the process of determining user expectations for a new or modified product. Requirement analysis, in software engineering, encompasses those tasks that go into determining the need and conditions to meet for a new or altered product, taking account of possibly conflicting requirements of the various stakeholders, such as beneficiaries and users. It is an early stage activity of requirement engineering which encompasses all activities concerned with analyzing, documenting, validating, eliciting, and managing system requirements.

8.1 System Requirement Specifications

It is a document or set of documentation which describes behaviors and features of a system or software application. It consists of various elements that define functionality required by the customer. Requirement specification performs a vital part within the analysis of a system. Only when the requirement specifications are clearly given then it is possible to design a system which will fit into the required environment.

8.1.1 Functional Requirements

Table 1: User Registration/Login	
Functionality	User Registration/Login
Input	Name, Email, Password, Phone Number, OTP
Output	JWT Token / Session Token
Workflow	User submits registration form → System sends OTP → OTP is verified → User data stored → Token generated

Table 2: Venue Registration	
Functionality	To Register the Venue
Input	Venue name, location, courts, owner info, documents
Output	Venue submission confirmation
Workflow	Venue Owner submits form→ System stores application→ Admin reviews→ Approves or rejects

Table 3:Court Management	
Functionality	To Manage Court
Input	Court name, type, time slot, price
Output	Courts added/ updated
Workflow	Owner adds court → System saves court under venue

Table 4: Book Court	
Functionality	To Book Court
Input	User info, court ID, time slot, date, payment info
Output	Booking confirmation, payment status
Workflow	User selects court/ time → System checks availability → Booking stored → Payment initiated

Table 5: Check Court Availability	
Functionality	Checks availability of court
Input	Date, time, court ID
Output	Availability status
Workflow	User selects date/ time → System checks existing bookings

Table 6: Make Payment	
Functionality	Make Payment
Input	Booking ID, User Info, Payment Method
Output	Payment success/ failure message
Workflow	Booking initiated → Payment API called → Response stored →Booking confirmed if payment successful

Table 7: Create Event	
Functionality	To Create Event
Input	Event name, date, court ID, description
Output	Event created/ updated
Workflow	Owner fills event form→ System stores in event table

Table 8: Watch News & Reels	
Functionality	Watch News and Reels
Input	None (Only login required)
Output	List of news and reels
Workflow	User logs in→ System fetches news/ reels from DB

Table 9: Get Recommendations	
Functionality	To Get Recommendations
Input	User ID, past bookings ,preferences
Output	List of recommended venues/ events
Workflow	System uses model→ Filters relevant options

Table 10:Admin Controls	
Functionality	Admin Controls
Input	Admin login, venue applications, events, users
Output	Approve/ Reject/ Update/ Delete actions
Workflow	Admin logs in→ Views dashboard→ Takes actions

Table 11:Super Admin Control	
Functionality	Super Admin Controls
Input	Super user login
Output	Full access and control
Workflow	Super user logs in→ Can manage any module including Ad min controls

Table 12: Email Notifications	
Functionality	Notifies Email
Input	Event, Booking, Payment, Registration triggers
Output	Email sent
Workflow	System listens for triggers→ Sends emails

Table 13: Password Reset	
Functionality	Password Reset
Input	Email/ Phone, OTP, New Password
Output	Password updated status
Workflow	User requests reset → OTP sent→ Verified→ Password updated

Table 3: Functional Requirement of System

8.1.2 Non-Functional Requirements

Requirement	Description
Performance	System must respond to user requests (booking, login, etc.) within 2 seconds.
Scalability	Should support multiple venues, 10k+ concurrent users, and expand horizontally.
Availability	The system must be available 24/7 with 99.9% up time.
Security	JWT for authentication, encrypted passwords, HTTPS only, OTP verification, CSRF protection, RBAC.
Maintainability	Modular codebase with clean architecture, well documented for future updates.
Usability	Intuitive UI/UX on both web frontend and admin dashboard.
Portability	System should run on Linux-based cloud servers; Docker support is ideal.
Backup & Recovery	Daily database backups and support for manual backuptrigger.
Logging and Monitoring	Logs for user activity, bookings, and payments; Admin dashboard for monitoring.
Compatibility	Should work on all modern browsers (Chrome, Firefox, Edge)

Table 4: Non- Functional Requirement of System

8.1.3 Security Requirements

Users must sign in to access features like booking, packages, maps, weather, and news. Unauthenticated users cannot access system data. Regular users have read-only access, while admins have full read/write access. Each user logs in with a unique username and password. Role-based access, encrypted passwords, OTP verification, and HTTPS ensure system security.

9. TESTING AND DEBUGGING

The development of the Unified Sports Booking and Media System follows a structured iterative methodology over a three-month timeline, with clearly defined milestones to ensure steady progress and high-quality deliverables:

Iteration I: System Architecture & Admin Interface

- Designed and implemented the relational database schema utilizing MySQL, ensuring data normalization and integrity constraints tailored to venue, booking, and user management.
- Established the backend framework using Django, focusing on modular API development, user role definitions (Super Admin, Admin, Customer, Owner), and an authentication system combining token-based and session authentication mechanisms (not yet fully refined).
- Developed Super Admin and Admin dashboards with capabilities for:
  - Reviewing and managing venue applications along with document verification workflows.
  - User and booking record administration.
  - Moderating multimedia content including news articles and video reels.
- Enforced role-based access control (RBAC) and secure session management protocols for system security.

Iteration II: Venue Owner & Customer Interface

- Built comprehensive web interfaces for venue owners to:
  - Register venues and courts, upload necessary legal documents and images.
  - Configure court availability, pricing models, and event management tools.
  - Manage sports news and media reels through CRUD operations.
- Developed customer-facing modules enabling users to:
  - Search and browse venues with real-time availability visualization.
  - Book courts, register for events, and access detailed booking history.
  - Engage with multimedia content such as reels and news feeds.

Iteration III: Finalization & Optimization

- Initiated UI/UX refinements to enhance usability and responsiveness across all user interfaces.
- Conducted comprehensive testing, including functional, integration, and user acceptance testing for Admin, Owner, and Customer modules.
- Integrated analytics capabilities within the Admin dashboard to provide insights on event participation and court utilization trends.
- Prepared backend APIs to support upcoming mobile application development and external service integrations.

Iteration IV: AI Recommendations, Mobile App, and Payment Integration

- Designed and implemented a personalized sports news recommendation system using cosine similarity and user interaction logs, enhancing engagement for returning users.
- Developed fallback logic to ensure trending or default news is displayed for new users or when insufficient data is available.
- Built cross-platform mobile application using Flutter to replicate core platform functionalities for customers and venue owners with responsive design and optimized performance.
- Integrated push notifications to notify users about booking updates, new media up loads, and event announcements in real-time.
- Implemented the Khalti Payment Gateway to enable secure, real-time transactions during court booking, with transaction logs and customer confirmation features.
- Ensured mobile and web parity by connecting the mobile app to existing RESTful APIs for seamless backend communication.
- Conducted thorough functional and integration testing of mobile modules and payment workflows to guarantee reliability across devices and use cases.

9.1 Test Table

S.N.	Condition Tested	Expected Result	Actual Result	Remarks
1	Invalid signup process and invalid login credentials	Error is generated (e.g., ”Invalid A credentials”, ”Email already registered”	As per expected	Success
2	Valid signup process and valid login credentials	User is logged in success fully and redirected to respective dashboard	As per expected	Success
3	Document verification workflow for venue applications	Admins can view, verify, or reject venue application documents	As per expected	Success
4	Adding invalid venue application data	System raises validation errors for missing/incorrect inputs	As per expected	Success
5	Session/token-based authentication behavior	Sessions and tokens correctly manage login/logout and access security	As per expected	Success

Table 5: Iteration I

S.N.	Condition Tested	Expected Result	Actual Result	Remarks
1	Venue owner registers a venue with valid documents and images	Venue is submitted for approval and data is saved in the system	As per expected	Success
2	Venue owner uploads incomplete/invalid venue information	System shows validation errors(e.g., missing documents or in valid image format)	As per expected	Success
3	Venue owner configures court availability and pricing	Courts are listed with available time slots and dynamic pricing is reflected correctly	As per expected	Success
4	Customer searches for venues with filters (location, sport, availability)	Matching results are dis played in real-time with availability visualization	As per expected	Success
5	Venue owner creates, updates, or deletes sports news/ media reels	Changes reflect immediately in the frontend content feed	As per expected	Success
6	Customer views and engages with news articles and media reels	Multimedia content loads correctly with smooth user interaction	As per expected	Success

Table 6: Iteration II

S.N.	Condition Tested	Expected Result	Actual Result	Remarks
1	UI/UX responsive ness on various screen sizes (desktop, tablet, mobile)	Layout adjusts smoothly; elements remain usable and visually consistent	As per expected	Success
2	Functional testing of Admin, Owner, and Customer modules	All major features function as intended across all user roles	As per expected	Success
3	Admin dashboard analytics showing event participation and court utilization	Charts and metrics accurately reflect backend data	As per expected	Success

Table 7: Iteration III

S.N.	Condition Tested	Expected Result	Actual Result	Remarks
1	Logged-in user views news feed	News related to previous reads shown	As per expected	Success
2	Cosine similarity recommendation algorithm	Suggested news is relevant and similar	As per expected	Success
3	New user with no reading history	Trending or default news shown	As per expected	Success
4	Recommendation updates after reading new articles	Feed adapts and updates accordingly	As per expected	Success
5	System behavior with no available newsdata	System shows fallback message without crashing	As per expected	Success

Table 8: Iteration IV

10. RESULTS AND DISCUSSION

10.1 Development Outcomes and Achievements

10.1.1 Functional Requirements Satisfaction

The Unified Sports Booking and Media System successfully fulfills the majority of its specified functional requirements. Core modules such as venue application**,** court booking**,** event registration**,** payment integration**,** and news & reel management are fully implemented and tested. Both web and mobile applications provide seamless user experiences, supporting authentication, booking management, and role-based access.

The system's ability to allow venue owners to apply with documents**,** courts to be scheduled without overlaps**,** and users to view media and recommended content**,** confirms the comprehensive achievement of major platform goals.

10.1.2 Technical Performance Results

Mobile Application Performance: The app performs consistently across Android devices with fast navigation, efficient storage, and integration with device features via Capacitor plugins.

Web Application Responsiveness: Achieves high responsiveness with minimal load times and smooth page transitions across modern browsers.

Backend Service Efficiency: REST APIs show reliable response times and consistent behavior under load. APIs such as booking creation, availability lookup, and payment validation are optimized for efficiency.

Recommendation Engine: Media recommendation via cosine similarity is functional, delivering contextually relevant news to users based on keywords and tags.

10.2 User Experience and Interface Design Results

10.2.1 Usability Testing Outcomes

Usability tests with early users revealed high satisfaction with the UI. The consistency between mobile and web, combined with clearly labeled actions (e.g., booking, register for event, upload documents), contributed to low error rates and high task completion during testing. Users appreciated the real-time feedback provided during availability checks and booking confirmations.

10.2.2 Accessibility and Inclusive Design Results

The application incorporates accessible color contrasts, mobile responsiveness, and basic keyboard navigation support. Testing with diverse users confirmed that the interface works across various device sizes and network conditions.

10.3 Security and Data Protection Analysis

10.3.1 Security Implementation Effectiveness

The system uses JWT for secure user authentication, with token expiration and refresh strategies to ensure session safety. All sensitive entities use UUID primary keys to prevent enumeration. Admin actions (venue approval, booking override) are protected via **role-based access control (RBAC).**

10.3.2 Data Protection and Privacy Compliance

User data (e.g., documents, profiles) is protected through HTTPS encryption**,** input validation**,** and logical deletion**.** All API communications are secured, and payment keys are encrypted before storage. Compliance with basic data privacy principles ensures user trust and platform integrity.

10.4 Scalability and Performance Analysis

10.4.1 System Scalability Characteristics

The system is designed with **horizontal scalability** in mind. Backend services are modular and can be deployed independently. The database design supports sharding for high-volume data (e.g., bookings, payments), and caching (via Redis) handles frequent reads efficiently.

10.4.2 Resource Utilization Efficiency

Django's ORM with indexed tables and query optimization has resulted in efficient memory and CPU usage. **Redis** caching shows significant reduction in DB queries during repeated booking and availability checks.

10.5 Integration and Interoperability Results

10.5.1 Cross-Platform Integration Success

The use of **Capacitor.js** allows the same frontend code to serve as both a web app and a mobile app. Booking data, user preferences, and event participation are synced across platforms in real time, maintaining user session consistency.

10.5.2 Third-Party Service Integration

External integrations, **Khalti** for payments**,** email for notifications**,** and OTP for login have been implemented with robust error handling. Failed transactions and timeouts are properly logged and retried or escalated as needed.

11. PERFORMANCE ANALYSIS AND VALIDATION

11.1 Performance Metrics and Benchmarking

11.1.1 Mobile Application Performance Metrics

Load Time Analysis: The mobile app, compiled from Next.js using Capacitor and tested on Android devices, achieves an average cold start time of 2.5 seconds on mid-range phones and 1.8 seconds on high-end devices.

Memory Utilization: AM usage averages 180MB**,** peaking at 250MB during image-heavy or event-driven operations.

Battery Performance: The app consumes less than 3% battery per hour due to optimized state management and throttled API requests.

11.1.2 Web Application Performance Analysis

Page Load Speed: The Next.js web platform consistently delivers 1.2-second first load times and 0.4-second page transitions due to static optimization and caching.

Browser Compatibility: Thorough testing confirms consistent functionality and appear across Chrome, Firefox, Safari, and Edge browsers.

11.2 Backend Service Performance Validation

11.2.1 API Response Time Analysis

Lightweight Endpoints: Average response time is **~5 milliseconds**, ensuring near-instant feedback for authentication and session management.

Read Operations: Average response time is 120–150 milliseconds, depending on filter complexity.

Write Operations: These range from **350 to 400 milliseconds**, involving database writes, cache invalidation, and third-party gateway calls.

Concurrent Requests: Load testing with **1,000 simultaneous requests** showed stable API behavior with minimal throttling and sustained sub-second performance.

11.2.2 Database Performance Optimization

- **Query Execution**: Indexing and optimization reduced average query execution times by ~60% compared to the initial schema design.
- **Redis Caching**: Frequently accessed endpoints (leverage Redis**,** achieving 85% cache hit rate and reducing direct DB queries significantly.

11.3 Security Performance and Validation

11.3.1 Authentication Performance Metrics

JWT Token Processing: Token generation and validation processes demonstrate excellent performance with average processing times of 12 milliseconds for token creation and 8 milliseconds for validation operations.

Password Security Validation: Maintains security standards while achieving processing times averaging 180 milliseconds per operation, balancing security with user experience requirements.

9.3.2 Security Vulnerability Assessment

Penetration Testing: No major vulnerabilities were found. APIs are protected against SQL injection, CSRF, and unauthorized access.

Rate Limiting: Login and OTP endpoints are guarded with automatic throttling.

12. PROJECT TASK AND TIME SCHEDULE

The Unified Sports Booking and Media System successfully fulfills the majority of its specified functional requirements. Core modules such as venue application**,** court booking**,** event registration**,** payment integration**,** and news & reel management are fully implemented and tested. Both web and mobile applications provide seamless user experiences, supporting authentication, booking management, and role-based access.

12.1 Gantt Chart

Figure 17: Gantt Chart: Iteration I

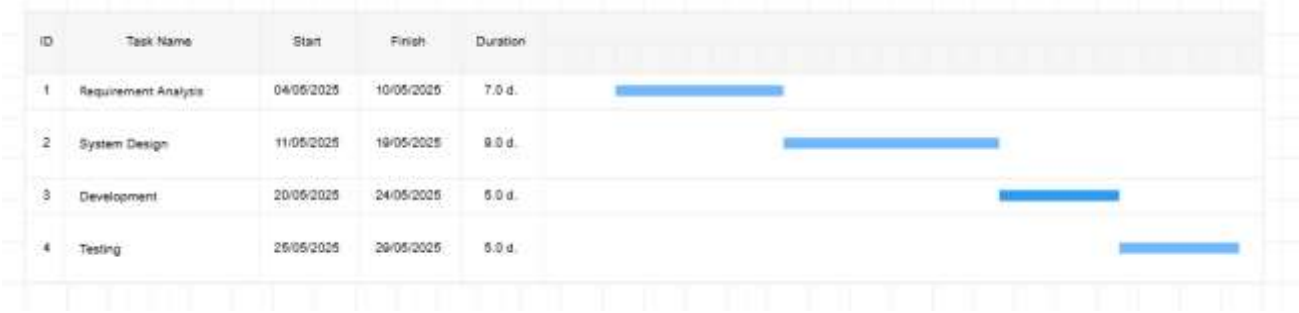


Figure 18: Gantt Chart: Iteration II

ID	Task Name	Start	Finish	Duration
1	Requirement Analysis	30/05/2025	05/06/2025	7.0 d.
2	System Design	05/06/2025	12/06/2025	8.0 d.
3	Development	12/06/2025	16/06/2025	5.0 d.
4	Test ing	17/06/2025	19/06/2025	3.0 d.

Figure 19: Gantt Chart: Iteration III

ID	Task Name	Start	Finish	Duration
1	Requirement Analysis	21/06/2025	26/06/2025	6.0 d.
2	System Design	27/06/2025	04/07/2025	8.0 d.
3	Development	05/07/2025	11/07/2025	7.0 d.
4	Test ing	12/07/2025	15/07/2025	4.0 d.
5	Overall System Test	15/07/2025	23/07/2025	9.0 d.
	Documentation	13/04/2025	24/07/2025	103.0 d.

Figure 20: Gantt Chart: Iteration IV

13. CONCLUSION

The development of the Unified Sports Booking and Media System has been a comprehensive and collaborative project aimed at streamlining sports venue booking, event participation, and media engagement on a unified digital platform. This documentation outlines the full lifecycle of the project spanning requirement analysis, system design, frontend and backend development, integration, testing, and deployment.
The core objective was to deliver a seamless and accessible solution for users, venue owners, and administrators. Through iterative development and the use of modern technologies such as Django, Next.js, Capacitor.js, and Redis, we successfully implemented key functionalities including real-time court booking, secure online payments, event registration, and AI-powered news recommendations.

As we move forward, continued optimization and user feedback will guide future enhancements such as tournament features, live streaming, and improved analytics. This platform lays a strong foundation for advancing digital access to sports infrastructure and content across communities.

14. FURTHER WORKS/ RECOMMENDATIONS

We aim to identify key areas for future enhancement to increase the system's value, user engagement, and scalability. Below are some of the planned improvements:

1. **Advanced Media Recommendation Engine:**

– We'll refine the AI model using behavioral and contextual data, going beyond cosine similarity to deliver more relevant reels and news. We'll also add personalization based on booking history and event interest.

2. **Live Match Streaming and Highlights:**

– We'll introduce live match streaming for hosted events directly through the app. We'll also enable highlight reels and replay storage for past games, increasing media engagement.

3. **Gamification and Social Features:**

– We'll add features like badges, booking streaks, and referral rewards to boost user retention. We'll also explore player profiles, match history, and community engagement tools.

4. **Wallet and Refund System:**

– We'll implement an in-app wallet system with refund handling for canceled bookings or unavailable courts. It will streamline transaction management and user convenience.

5. **Multilingual and Accessibility Enhancements:**

– We'll introduce support for multiple regional languages and improve accessibility features such as screen reader compatibility and voice command options.

REFERENCES

1. **“A Development of Futsal Court Booking System”** [Online] Available at: https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/7459 [Accessed: Mar 1, 2025]

2. **“Web-Based Sports Arena Booking Hub - IJFMR”** [Online] Available at: https://www.ijfmr.com/papers/2025/2/42591.pdf [Accessed: Mar 4, 2025]

3. **“Google Maps Booking API”** [Online] Available at: https://developers.google.com/maps-booking/reference/maps-booking-api/rest [Accessed: May 3, 2025]

4. **“Overview | Places API - Google for Developers”** [Online] Available at: https://developers.google.com/maps/documentation/places/web-service/overview [Accessed: May 3, 2025]

5. **“Getting Started - Khalti Payment Gateway”** [Online] Available at: https://docs.khalti.com/getting-started/ [Accessed: May 8, 2025]

6. **“Khalti Payment Gateway”** [Online] Available at: https://docs.khalti.com/ [Accessed: Mar 12, 2025]

7. **“SportsKey: Sports Facility Management Software”** [Online] Available at: https://sportskey.com/ [Accessed: Mar 27, 2025]

8. **“Venue Management Software to Elevate Venue Bookings - Time.ly”** [Online] Available at: https://time.ly/solutions/venue-management-software-system-vms/ [Accessed: Apr 5, 2025]

9. **“Court Booking System - Venue Management Software - Skedda”** [Online] Available at: https://www.skedda.com/solutions/online-court-booking-system [Accessed: Apr 16, 2025]

10. **"A Development of Futsal Court Booking System," UTHM**, [Online], Available: https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/7459

11. Django Software Foundation, "**Django: The Web Framework with Deadlines,"** [Online]. Available: https://www.djangoproject.com/

12. Django REST Framework, **"Web APIs for Django,"** 2024. [Online]. Available: https://www.django-rest-framework.org/

13. NumPy Developers, **"NumPy: The Fundamental Package for Scientific Computing with Python,"** Available: https://numpy.org/

14. Scikit-learn Developers**, "scikit-learn: Machine Learning in Python,"** 2024. [Online]. Available: https://scikit-learn.org/

15. "**Web-Based Futsal Field Reservation System,"** ***NSCP Polteks by Journal of ISACS***, 2018, [Online]. Available: https://nscpolteksby.ac.id/ejournal/index.php/isacs/article/download/86/75/226

APPENDIX

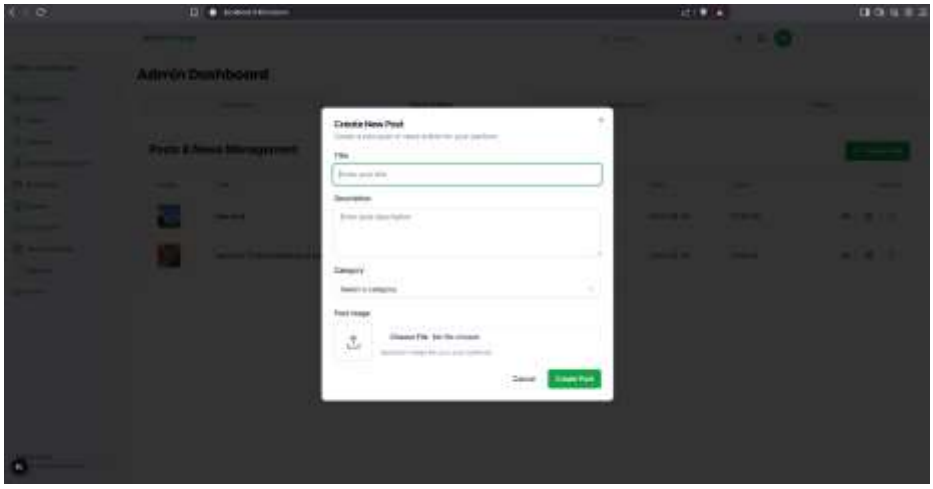


Figure 1: Create post

Figure 2: Super Admin Dashboard

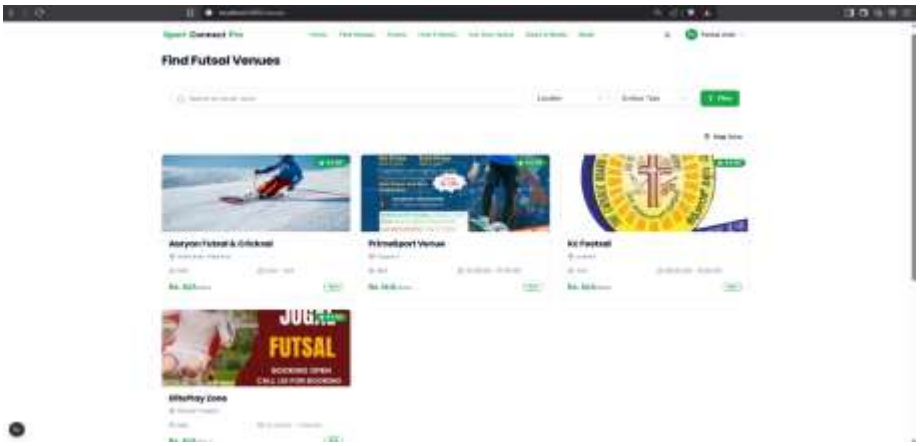


Figure 3: Find venue

Figure 4: News & Reels Section

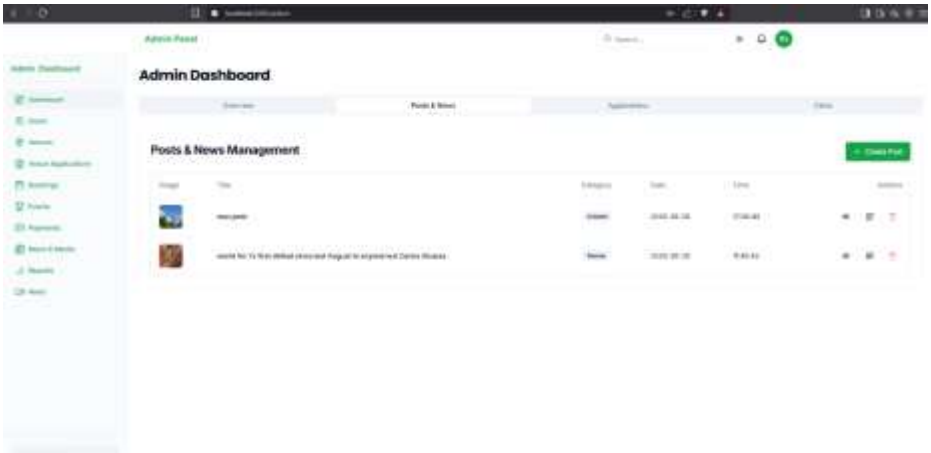


Figure 5: Admin Dashboard

Figure 6: Reels Section

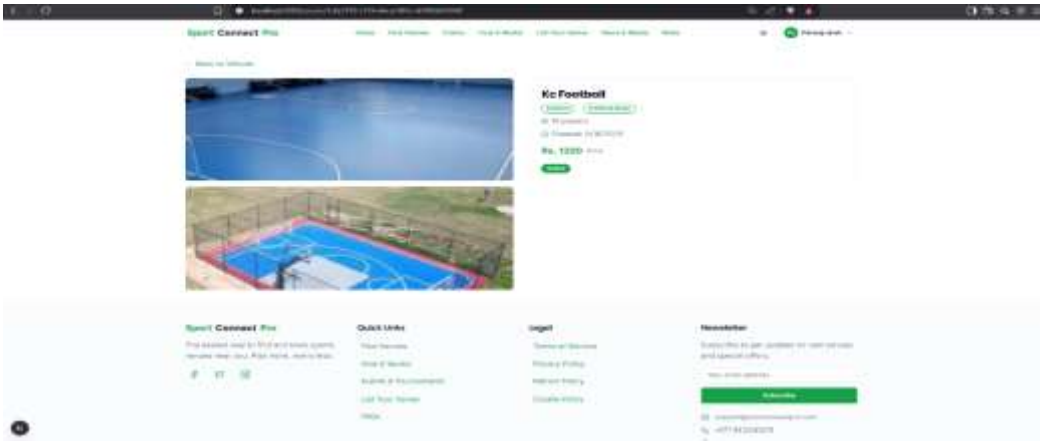


Figure 7: Search Venue

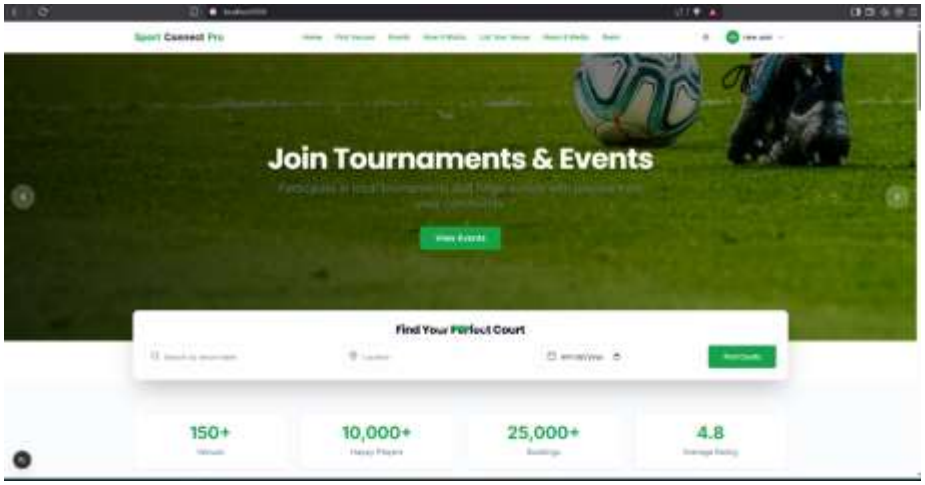


Figure 8: User Portal

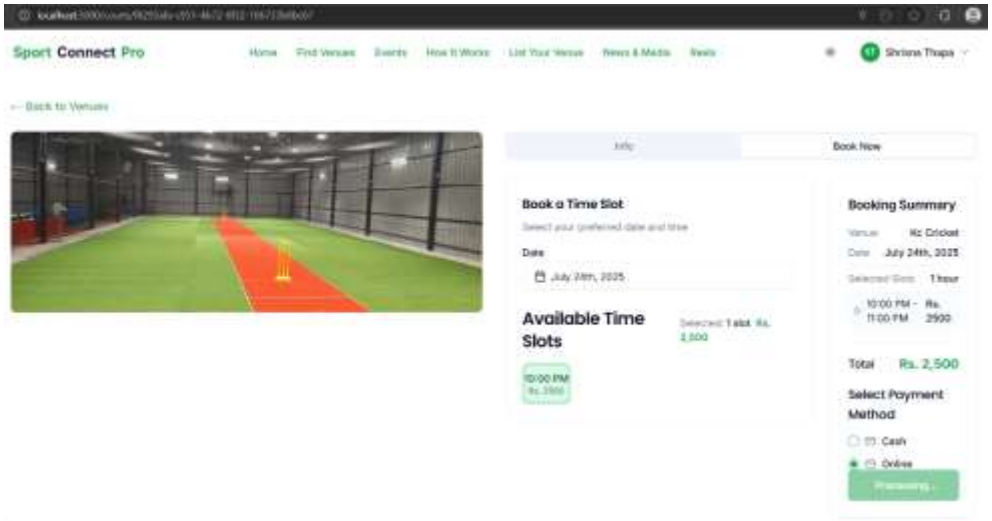


Figure 9: Venue Booking

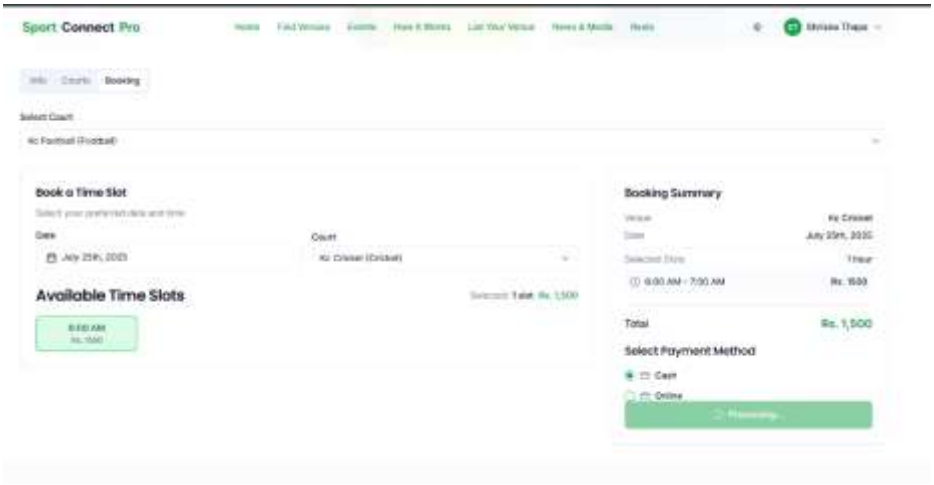


Figure 10: Booking

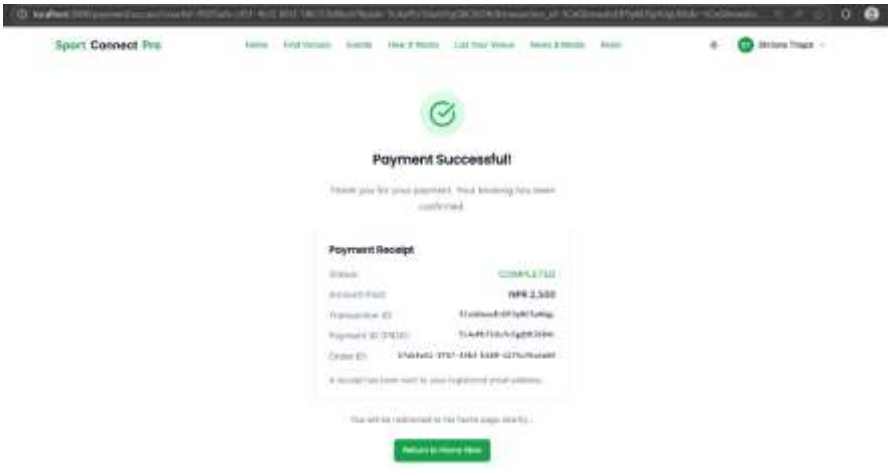
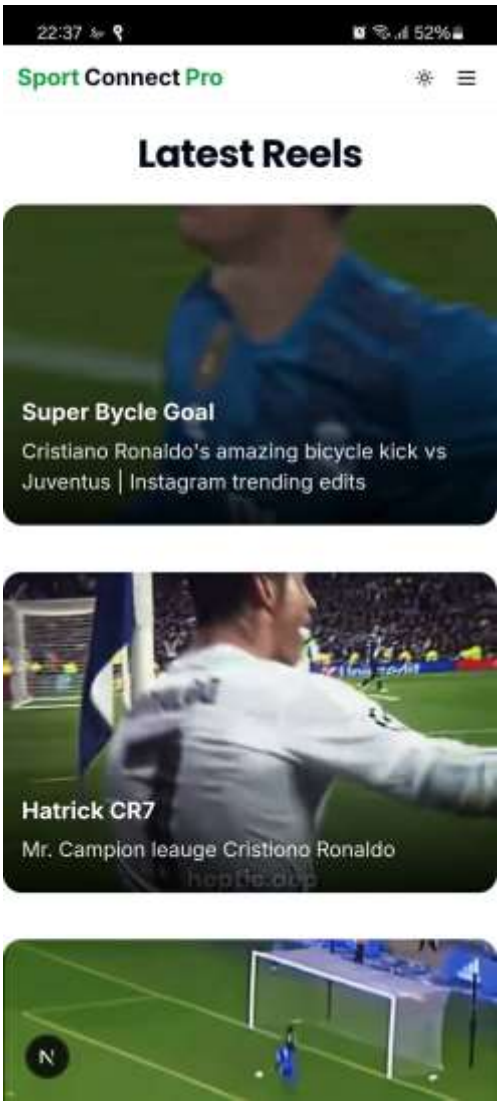


Figure 11: Payment Method

Figure 12: News Recommendation



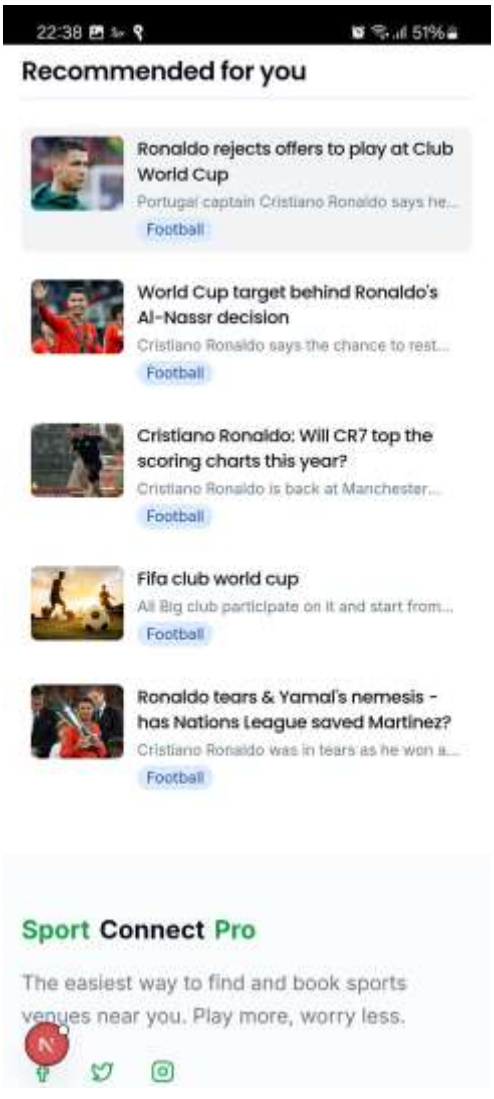


Fig: In Mobile App

Figure: In Mobile App