OPTIMIZING GOLF CLUB MANAGEMENT- A MULTIUSER PLATFORM

Comprehensive Project Report

Submitted in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

By

Dinky Bharat Unadkat

21BCP033

Under the Guidance of

Dr Davinder Paul Singh

Department of Computer Science & Engineering,

School of Technology, Pandit Deendayal Energy University,

Gandhinagar 382 426

May 2025

Certificate of Originality of Work

I hereby declare that the B.Tech. Project entitled “Optimizing Golf Management” submitted by me for the partial fulfillment of the degree of Bachelor of Technology to the Dept. of Computer Science & Engineering at the School of technology, Pandit Deendayal Energy University, Gandhinagar, is the original record of the Comprehensive project work carried out by me under the supervision of Dr Davinder Paul Singh.

I also declare that this written submission adheres to university guidelines for its originality, and proper citations and references have been included wherever required.

I also declare that I have maintained high academic honesty and integrity and have not falsified any data in my submission.

I also understand that violation of any guidelines in this regard will attract disciplinary action by the institute.

Name of the Student: **Dinky Bharat Unadkat**

Roll Number of the Student: **21BCP033**

Signature of the Student:

Name of the Supervisor: **Dr Davinder Paul Singh**

Designation of the Supervisor: **Assistant Professor, Department of CSE**

Signature of the Supervisor:

Place: **Gandhinagar**

Date: **21st May 2025**

Certificate from the Project Supervisor/Head

This is to certify that the Comprehensive Project Report entitled "**Optimizing Golf Management**" submitted by ***Ms. Dinky Bharat Unadkat***, Roll No. ***21BCP033*** towards the partial fulfilment of the requirements for the award of degree in Bachelor of Technology in the field of Computer Science & Engineering from the School of technology, Pandit Deendayal Energy University, Gandhinagar is the record of work carried out by him/her under my/our supervision and guidance. The work submitted by the student has in my/our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for the award of any degree or diploma.

Dr. Davinder Paul Singh
Name and Sign of the Supervisor

Mr. Mayur Rupala
Name and Sign of the Industry Supervisor

Dr Shakti Mishra
Name and Sign of the HoD

Dr Dhaval Pujara
Name and Sign of the Director

Place: **Pandit Deendayal Energy University**

Date: **21st May 2025**

Acknowledgement

The production of this thesis has been a journey facilitated by the help and contributions of numerous institutions and individuals, and I am highly thankful for their support. I am grateful for this opportunity to extend my warmest appreciation to everyone who has contributed in any way toward making this work a reality.

Firstly, I extend my sincerest gratitude and heartfelt thanks to my supervisor, Dr Davinder Paul Singh, for their helpful guidance, insightful recommendations, and endless encouragement throughout this research. Their faith in my work have been essential to its success. They provided not only academic guidance but also a positive environment that nurtured my intellectual development.

I would also like to appreciate my experiences at Sarjen Systems Private Limited, Ahmedabad. My experience in working there as a React and frontend developer had a significant impact on the practical elements of this project. The experiences and knowledge gained within a professional environment played a pivotal role in influencing my approach to the theoretical ideals addressed within this thesis. I am particularly thankful to my manager, Mr. Mayur Rupala, for his mentorship, advice, and support during the time spent at Sarjen Systems. His expertise and insights into frontend development were highly valuable to my understanding and implementation of these ideas within my thesis. In addition, I would like to appreciate the entire team at Sarjen Systems for enabling a positive and cooperative working environment.

Lastly, I would like to extend my heartfelt gratitude to my friends and family. Their patience, unwavering support and belief in my abilities have been highly influential in this process. This journey has been intellectually challenging and personally demanding, and their support has been an abiding source of strength. And, in recognition of the source of all inspiration, I offer my thanks for the blessings received along the way.

Dinky Bharat Unadkat

Abstract

The accelerated development of digital technologies has significantly impacted the sports and event management sector, providing creative solutions for live sports event management, organization, and experience. This project displays the design and deployment of a Golf Management System that integrates different operations, such as event creation and handling, live scoreboard tracking, leaderboard refresh, player profile handling, booking systems, and group handling, in an unobtrusive and interactive interface. The system takes advantage of a variety of innovative technologies to provide an advanced, user-focused solution.

The system's frontend is developed based on React, using Tailwind CSS for responsive, trendy, and visually stunning designs. Context API is utilized for managing state to have proper data handling across components efficiently, and Axios is utilized for integrating smoothly with backend APIs for offering a smooth experience with reduced load times. Updates of real-time data are provided by WebSockets for updating live scores and the leaderboard in real-time instantly, making it an enhanced experience during live events. Figma was employed to create the user-friendly interface, providing a unified and easy-to-use experience for administrators and participants alike.

On the backend, the backend is driven by a solid combination of Django, which acts as the API provider, providing scalable and secure communication between the frontend and database. MongoDB is used for data persistence and real-time operations, allowing flexibility and high performance for the system's various data needs. Django and MongoDB integration provides efficient processing of user details, event details, and booking status with high emphasis on scalability and rapid retrieval of real-time data.

The core functionalities of the system are user login, event and group creation, live leaderboard refresh, tracking of player statistics, and dynamic event booking features. Administrators can access a full dashboard that enables easy management of multiple tournaments, player information, and real-time status updates. Live scoreboards, automatic rankings, and instant notifications guarantee that all stakeholders are kept up to date and engaged at all times.

From the time it was developed, there was a heavy focus on performance optimization, so that the platform could scale well during peak event hours and have real-time data synchronization amongst users. The use of MongoDB for managing event and player data, in conjunction with an efficient Django API, guarantees seamless, fault-tolerant operations and future scalability.

This Golf Management System not only updates conventional processes of event and player management but also is an adaptable and extendable solution to the changing needs of the sporting industry. The project demonstrates the strength of utilizing contemporary web technology, including React, WebSockets, Tailwind, and Django, paired with a robust backend framework in order to construct a highly dynamic and efficient system for managing real-time, dynamic data for sports events.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Prologue

Over the last few years, the use of technology in sports management has revolutionized the process of event organization, monitoring, and experience by participants, administrators, and viewers. The increasing need for digital solutions in sports management has brought forth the demand for platforms offering real-time data synchronization, effortless event management, and a better user experience. One such sport that has gained from this technological change is golf, whose conventional ways of scheduling tournaments and handling player interaction can be tedious and error-prone.

This project seeks to tackle these issues through the creation of a Golf Management System, a web-based solution that aims to automate and streamline a number of key functions used in organizing and taking part in golf tournaments. The system allows for real-time scoreboard updates, leaderboard monitoring, event creation and management, authentication of users, and dynamic booking of events, all within a unified platform.

The Golf Management System is developed leveraging contemporary web technology, such as React for user interface, Tailwind CSS for responsive design, and Axios for seamless API calls. State management throughout the app is managed utilizing Context API for an efficient, scalable architecture. Real-time user interactions, i.e., updating scores in real time and leaders changing, are enabled through WebSockets for instant data sync across multiple devices and users.

On the backend, the platform uses Django as the API layer, providing a secure and scalable environment for handling user information, event information, and booking status. The backend is supported by MongoDB, a NoSQL database that offers flexibility and high performance in handling large data sets, especially in real-time applications. The integration of these technologies provides a solid, secure, and scalable solution that can easily handle high traffic volumes during peak event periods.

The major features of the system are user login and profile management, creating and managing golf events, live leaderboard refresh, group creation, and event booking.

The administrators can easily manage multiple tournaments, see extensive player statistics, and track the status of the running events. Players, on the contrary, can see live updates on ranking and scores, book events, and monitor their performance during the tournament.

This project not only intends to streamline the logistics of golf event management but also provides a contemporary, interactive platform for both event organizers and participants. Utilizing cutting-edge web technologies and providing real-time functionality, this Golf Management System lays the foundation for the future of sports management, allowing golf events to be operated more efficiently and interactively than ever.

1.2 Motivation

The growing use of digital technologies in sports management has revolutionized the organization, tracking, and experience of events in a radical way. The game of golf, with its worldwide appeal and heritage, has predominantly used traditional event coordination, player management, and score monitoring methods. Such traditional approaches, though effective, have built-in limitations regarding scalability, real-time data synchronization, and user participation. While more dynamic, data-driven solutions continue to grow in demand, the real opportunity to create a detailed digital platform for these limitations, improving the general experience for the players and organizers of events alike, exists in large measure.

The need to enhance efficiency and accuracy within golf tournament management has driven this thesis. Historically, golf event management has involved a multitude of manual procedures, such as score entry, player sign-up, event calendaring, and participant notifications. These activities are frequently time-consuming and prone to human error, causing delays and discrepancies in event reporting. The problem, then, is developing a solution that mechanizes these procedures, minimizes the possibility of error, and gives real-time reports that benefit the player, spectators, and administrators alike.

Additionally, the increase in demand for real-time access to data in sports, fueled by the spread of web and mobile technologies, has transformed the way participants and spectators interact with sporting events.

In golf, players and spectators anticipate real-time information on player scores, leaderboards, and performance statistics. Organizers of tournaments also desire a more effective means of handling event logistics, including player pairings, reservations, and real-time tournament status.

This thesis aims to meet these demands by offering an environment that supports effortless event creation, management, and attendance coupled with real-time data features like live scoreboards and dynamic leaderboard updates.

Secondly, this project is inspired by the growing trend of digitalization in the sports sector where the use of cloud-based infrastructure, mobile apps, and high-level data analytics is becoming increasingly common. With the use of recent web development tools like React, Django, MongoDB, WebSockets, and Tailwind CSS, the intended system will establish a scalable, adaptable, and user-friendly solution that can process high volumes of users and dynamic data processing. The capability of the platform to offer real-time interaction and synchronization of data between users is in direct response to the need for more interactive and engaging experiences across sports events.

In addition, the driving force of this project is to give back to the wider sport management discipline by offering a digital solution that not only works but is also flexible to meet the changing demands of the golfing community. Through the provision of sophisticated features such as event and group management, tracking of player statistics, and real-time alerts, this system seeks to develop an environment that caters to administrative requirements as well as improves the user experience.

In summary, this thesis seeks to fill the lacuna in existing golf event management practice by creating an all-embracing, real-time system that automates and simplifies major processes and offers a dynamic, interactive experience for all parties involved. The result of this research will not only enhance the effectiveness of golf tournament management but also add to the wider phenomenon of digitalization in sports management.

1.3 Objective

The foremost purpose of this thesis is to architect and design a comprehensive system of Golf Management employing advanced web technology for easier tournament and event handling. It attempts to redress the pitfalls of the traditional practices through automatic execution of routines and timely report dispatches, active user interface, and administrative management efficiency. The detailed aims are:

1. **Build a Real-Time Event Management System**

To provide a forum for administrators to effortlessly create, organize, and monitor golf tournaments, player enrollments, grouping, and schedule management, and get real-time event status updates and bookings.

2. **Install Live Scoreboards and Leaderboards**

To develop dynamic, real-time scoreboards and leaderboards capturing up-to-the-minute player scores and rankings over the course of the tournament and increasing user engagement.

3. **Enable User Authentication and Profile Management**

To create a secure authentication system to enable users to create and manage profiles, providing secure access to related tournament data and user interactions.

4. **Implement Real-Time Communication through WebSockets**

To implement WebSockets for real-time data synchronization between users, allowing for live score updates, leaderboard updates, and event status announcements.

5. **Create a Responsive and Friendly Interface**

To design an intuitive and responsive UI with React and Tailwind CSS to offer a smooth experience on all devices, emphasizing simplicity and navigation.

6. **Offer an Efficient Backend System for Data Management**

To develop a scalable backend with Django for API management and MongoDB for data storage to ensure high performance, security, and scalability.

Through these achievements, the Golf Management System will offer a new, effective solution for golf event management, advancing the digitalization of sports management.

1.4 Problem Statement

Golf tournament organization has always depended on manual procedures for player registration, score reporting, scheduling, and communication, which results in errors, delays, and inefficiencies. Current systems are fragmented and do not include features such as real-time data synchronization, smooth user interaction, and support for large events. This poses challenges to event organizers and participants, especially when ensuring accurate and current information. There is an apparent necessity for a complete platform which automates the above processes, gives real-time updates regarding the scores and the rankings, and improves the entire user experience. This project sets out to design a Golf Management System that organizes event handling, real-time score monitoring, user engagement, and scalability and provides a smoother and more participative solution to golf tournaments.

1.5 Approach

Month 1

- Conducted in-depth research on golf tournament workflows, scoring systems, and event structures to understand the end-user journey.
- Took reference from already existing systems like TopGolf.

- Collaborated with stakeholders and backend developers to define comprehensive front-end requirements and data dependencies.
- Created detailed user personas and flowcharts for various roles: **Admin**, **Player**, **Viewer**, and **Event Organizer**.
- Designed wireframes and interactive UI mockups using Figma for:
    1. Live scoreboard
    2. Leaderboard
    3. Event creation form
    4. Player and group views
    5. Login/registration and dashboard
- Set up project architecture (React with functional components, routing, modular CSS, environment configs).
- Finalized UI component library (Material UI / Tailwind CSS) for scalable and accessible design.

Month 2

- Developed scalable component architecture using React Hooks, Context API, and prop drilling strategies where needed.
- Created reusable components: buttons, inputs, tables, dropdowns, modal popups, scorecards, and status indicators.
- Engineered static views for:
    1. Home page
    2. Live scoreboard mock (grid-based system)
    3. Leaderboard layout (sortable, rank-based)
    4. Player profile cards with dynamic group assignment sections
    5. Bookings table with filter/search options
- Focused on pixel-perfect UI with responsive design principles across devices (desktop, tablet, mobile).
- Integrated routing and page transitions using React Router.
- Set up static state mock data to simulate real-time updates and test UI reactivity.

Month 3:

- Integrated backend APIs for login, user authentication, player/event data fetching, and booking operations.
- Enabled form submissions for event creation, group formation, and score updates with robust field-level validations.
- Persisted admin information in state to personalize dashboard headers and control access to privileged operations.
- Dynamically displayed data such as:
    1. Total bookings (filterable by day, month, all-time)
    2. Total players (similarly filterable)
    3. Recent activity feed (e.g., today's new bookings)
- Implemented a **live score session module** per active booking using WebSockets.The session view includes:
    1. Booking ID
    2. Hole number
    3. Currently active player
- Real-time score table that auto-updates as scores are submitted
- Expandable sections allow viewing detailed scoring and switching between groups in real-time without refresh.
- Built dashboards tailored to user roles with intelligent data rendering.
- Incorporated loading states, toast alerts for feedback, and skeleton loaders for better perceived performance.

Month 4

- Built out event management modules allowing admins toCreate, edit, and view tournament events
- Developed a group selection and booking flow inspired by BookMyShow's seat selection model.

- Streamlined group-to-event mapping and ensured consistent data rendering across views.

- Implemented role-specific private routes to restrict access. Used conditional route guards and dynamic navigation rendering based on session state.

- Initiated exploration into real-time animation libraries to build a visually rich scoreboard interface.
- Actively experimenting with:
    - GSAP: for timeline-based animations on score updates (e.g., sliding cells, flashing scorecards).
    - Framer Motion: for smooth UI transitions, expanding/collapsing player stats, and leaderboard ranking shifts.
    - D3.js: for interactive data visualizations like score progression charts.
    - Three.js: long-term idea for a 3D course view or a visual hit animation in the live score session.
- Building UI prototypes and testing performance trade-offs between CSS-based and JS-based animations.
- Started breaking down larger views into smaller, reusable components to:
    - Improve rendering performance
    - Enhance code readability
    - Prepare for component-based animation injection (e.g., animating only the updated leaderboard row)
- Focused on improving responsiveness and layout stability during real-time updates (live game session).

Ongoing Tasks & Learning Focus

- Continued integration of backend APIs for advanced group logic and booking validation.
- Learning deeper Redux patterns (e.g., middleware, selectors).

- Studying integration paths between Redux and animation libraries for managing animation state.
- Planning structure for integrating **live WebSocket updates** with Redux (e.g., storing live scores in a real-time Redux slice).
- Began refactoring local state logic to **Redux** for centralized, scalable state management.
  - Created slices for user, events, bookings, scoreboard, and groups.
  - Implemented async thunks for API communication, replacing scattered `useEffect` fetch logic.
  - Ensured Redux state updates reflect instantly across deeply nested components without redundant prop-drilling.
- Focused on structuring the Redux store to improve maintainability and enable better debugging with Redux DevTools.

1.6 Scope of the Project

The scope of this project is the end-to-end development of a web-based **Golf Management System**, which aims to automate and streamline the complete process of arranging and managing golf tournaments. The system has been developed within a span of four months, including everything from user research, UI/UX design, and development of the prototype to frontend and backend implementation, handling real-time data, and thorough testing. It is intended to overcome the shortcomings of conventional golf event management and provide an up-to-date, responsive, and interactive solution.The functionalities at the heart of the functionality included within the scope of the project are:

User Authentication and Profile Management: Secure login, registration, and role-based access control (admin, player, spectator) implementation. Profile management for users for updating personal data and accessing tournament history.

Event Creation and Management: Administrative tools to design, schedule, and organize golf matches and tournaments. Dynamic player grouping and tee-time management.

Leaderboard and Scoreboard in real-time: Addition of WebSockets support for real-time scoreboard updates as well as automated leaderboard ranking. Live statistics tracking and player performance monitoring during the event duration.

Event Booking, Group Creation: User functionality for event booking, viewing booking status, and creating or joining groups of players.

Backend Integration and Data Persistence: A Django-based RESTful API with a MongoDB database for scalable and adaptive data storage.Storage of all the user, event, and performance data in a secure manner emphasizing integrity and reliability.

Frontend Building and State Management: Constructed with React and Tailwind CSS-styled for a responsive and visually appealing interface.State management with Context API for effective global state handling and user-friendly experience. Integration of Axios for asynchronous API calls.

Prototype Design and UI/UX Emphasis: High-fidelity interactive prototypes designed in Figma, with careful animations, transitions, and responsive layouts. Several rounds of user interface design revisions based on feedback to make sure that there is an intuitive and professional user experience.

Performance Optimization and Testing: Code optimization for minimal latency, particularly in real-time capabilities.Thorough unit and integration testing performed across modules to ensure system stability and correctness.

Scalability and Future Growth: Built with scalability to accommodate a growing number of users, events, and real-time loads of data. Architecture facilitates future integration with mobile apps, wearable tracking devices, and AI-driven performance analytics.

This project not only provides a complete working system but also embodies a thorough dive into full-stack web development, real-time communication, and user-centered design. Combining a rich Figma prototype with animated transitions, rich frontend logic, and a solid backend showcases the technical depth and time spent during the development process.

The scope has been meticulously organized to ensure every feature meaningfully addresses the real-world issues encountered in golf event management.

1.7 About the Company

Sarjen Systems Pvt. Ltd. is a prominent IT solutions provider company with headquarters in Ahmedabad, Gujarat. Founded in 1995, the company has expertise in providing a variety of services like software development, enterprise applications, and business process automation.

With the mission of providing high-quality customized software solutions, Sarjen Systems provides solutions to diverse industry sectors like healthcare, education, finance, and manufacturing.

Famed for its focus on excellence, Sarjen Systems is staffed with a devoted group of competent professionals who use the most up-to-date technologies to develop efficient, scalable, and secure software products. Its primary products comprise ERP systems, mobile apps, cloud computing, and data analysis, all with the aim of enhancing business functions and organizational effectiveness.

Sarjen Systems has established a solid reputation as a customer-driven company, not only providing powerful technical solutions but also strategic advisory and support services. Its persistent innovation and focus on research and development have helped it become a reliable partner for companies looking to improve their technological infrastructure.With more than two decades of experience under its belt, Sarjen Systems continues to advance, adopting new trends and offering innovative solutions that enable organizations to remain competitive in the rapidly changing digital world.

1.8 Organization of the Rest of the Report

The remainder of this report is structured as follows:

1. Chapter 2: Literature Review - Previous Approaches to Solve the Problem, Overview of relevant technologies, frameworks and industry trends
2. Chapter 3: Software Design – Technical architecture and Implementation
3. Chapter 4: Results and Discussion
4. Chapter 5: Conclusion
5. Chapter 6: Future Scope

CHAPTER 2
LITERATURE REVIEW

2.1 Previous Approaches to Solve the Problem

Traditionally, golf tournament management was based on manual operations, paper scoring, and simple spreadsheet software. Although acceptable for small or casual events, these methods bring inefficiency, fall victim to human mistake, and are not well-suited to larger tournaments or real-time engagement.

Manual Coordination and Offline Tools

Historically, tournament planners recorded player pairings, tee times, and scores using paper-based scorecards or collaborative Excel workbooks. They did not support collaborative features, version control, and real-time visibility for attendees and spectators, particularly during current matches.

Legacy Software Solutions

Early computer programs like BlueGolf, Golf League Tracker, and early versions of Golf Genius provided organized data entry and minimal automation for pairings, leaderboards, and reservations. These systems were usually developed as desktop applications, so they were less convenient on mobile platforms and not optimized for responsive web deployment. Most of them did not have real-time synchronization, role-based interfaces, or user-friendly experiences.

Web-Based Platforms and Their Limitations

New platforms such as TopGolf, 18Birdies, and cloud-based implementations of Golf Genius have had successes with enhancing user interaction and online tournament operations. They include functionality such as live scoring, player profiles, and event creation processes. Nevertheless, a number of limitations still exist:

- Inadequate customization support for amateur-level or niche events
- Limited role separation across different stakeholders (e.g., admins, players, spectators)

- Insufficient real-time data streaming and seamless UI transitions when scores are updating\
- Rigid group handling and booking processes

These deficiencies speak to the importance of a flexible, modular, and interactive system designed for diverse user roles with real-time and scalable architecture — especially for local or independent promoters who need to be agile without enterprise-level complexity.

2.2 Overview of relevant technologies, frameworks and industry trends

The creation of contemporary golf tournament management systems is being fueled more and more by real-time interactivity, responsive design, modular code structure, and rich data visualization. Some of the current technologies and architectural styles were used to assess and implement in creating a strong, user-friendly golf tournament management system.

Frontend Frameworks and Component Architecture

React was chosen as the main framework based on its component-based architecture, rich ecosystem, and broad use in industry projects. Functional components with React Hooks (useState, useEffect, useContext) offered a flexible pattern for handling view states and side effects.

React Router was used for effective navigation, and Context API was used for early-stage global state sharing prior to moving on to more scalable patterns.

Styling Libraries and Design Systems

A hybrid methodology using Material UI and Tailwind CSS was utilized to ensure design consistency and layout flexibility- Material UI provided a professional, accessible starting point of pre-designed components that were aligned with Material Design principles. Tailwind CSS facilitated utility-first styling and responsive design, especially ideal for intricate UI layouts like scoreboards and booking grids. The methodology ensured the quick development of UI while keeping complete control of the user interface.

State Management

As application complexity grew, State Management was used for organized, centralized state management. Core practices included defining modular slices for user, event, booking, and score data, using async trunks for backend API integration. Minimizing prop-drilling through highly nested components. This shift enhanced maintainability, scalability, and guaranteed consistency in the application's dynamic features.

Real-Time Communication with WebSockets

A live scoring facility in real time was implemented utilizing WebSocket communication. Live synchronization Real-time rendering of data on dashboards without page reloading.

Responsive Design and UX Improvements

All interfaces were implemented following responsive design principles in order to ensure usability on desktop, tablet, and mobile devices. UI feedback mechanisms like skeleton loaders, toast notifications, and loading states were added to enhance perceived performance and user satisfaction throughout asynchronous operations.

Current Industry Trends and Best Practices

The entire solution was influenced by some prevailing trends in sports tech and frontend development such as Modular architecture for maintainability and reuse, role-based routing and access control, Real-time UX via WebSocket integration, Cloud-first and API-centric development, Atomic design methodology for horizontally scalable component creation, Micro interaction integration for enhanced user engagement.

These trends combined dictated the system architecture and implementation approaches, leading to a contemporary, responsive, and flexible tournament management platform.

CHAPTER 3
SOFTWARE DESIGN

The golf tournament management platform was designed with a modular, scalable, and real-time-first approach. It is built to accommodate role-based user flows, responsive interactions, and effortless integration with backend APIs. The system is focused on high reusability of components, consistency of state, and dynamic rendering throughout the application.

Architectural Overview

The application follows the client-server architecture, with the frontend (implemented using React) consuming data from a RESTful API backend and having persistent connections for real-time updates over WebSockets.

- Component-driven development with React using functional components
- State management using Context API
- Separation of concerns between presentation, logic, and data
- Role-based routing and access control
- Responsive UI/UX for cross-device support

Data Flow and Communication between different components

1. User Authentication

- Utilizes secure API endpoints to log in and retrieve session data
- User role is kept in Redux state for role-based rendering and permission

2. Event and Booking Management

- Admins create/edit events through forms validated both UI and API level
- Groups and bookings are dynamically allocated using a system based on BookMyShow's seat model

3. Live Score Session

- Every live booking maintains an active WebSocket session
- Scores coming from one device mirror immediately on all attached clients
- Dynamic UI refreshes data include live hole, player, and group performance

Dashboard Customization

- Admin and players view tailor-made dashboards with real-time stats
- Auto-refreshed real-time statistics (e.g., bookings total, recent activity) are displayed

Core System Components

Component	Description
Frontend (React)	Handles UI rendering, routing, form validation, WebSocket connections, and API interactions. Built using React, React Router, and Context API
Backend APIs	RESTful services for authentication, user management, event handling, bookings, score updates. Uses consistent JSON responses and is designed for easy integration.
WebSocket Server	Powers real-time updates for live score sessions and dynamic leaderboard changes, ensuring minimal latency and synchronization across user views.
State Management	Manages state for users, bookings, groups, scores, and events. Implemented with slices and async trunks to ensure predictable and scalable data flow.
UI Component Library	Combination of Material UI (for standardized components) and Tailwind CSS (for custom layouts and utility-first styling).

Role-Based Access Layer	Guards and redirects based on session roles (Admin, Player, Viewer, Organizer) using protected routes and conditional rendering.
Animation Layer	Experiments with Framer Motion and GSAP to bring dynamic interactivity to the scoreboard, leaderboard, and group views.
Visualization Layer	Future-ready modules using D3.js and Three.js for score trends and 3D course interaction.

Table 3.1: Components of the Golf Management System

Security & Access Control

- Private routes ensure that unauthorized users are redirected
- Conditional rendering is implemented throughout the app based on the session role
- Session persistence is handled via localStorage and ContextAPI to maintain login state

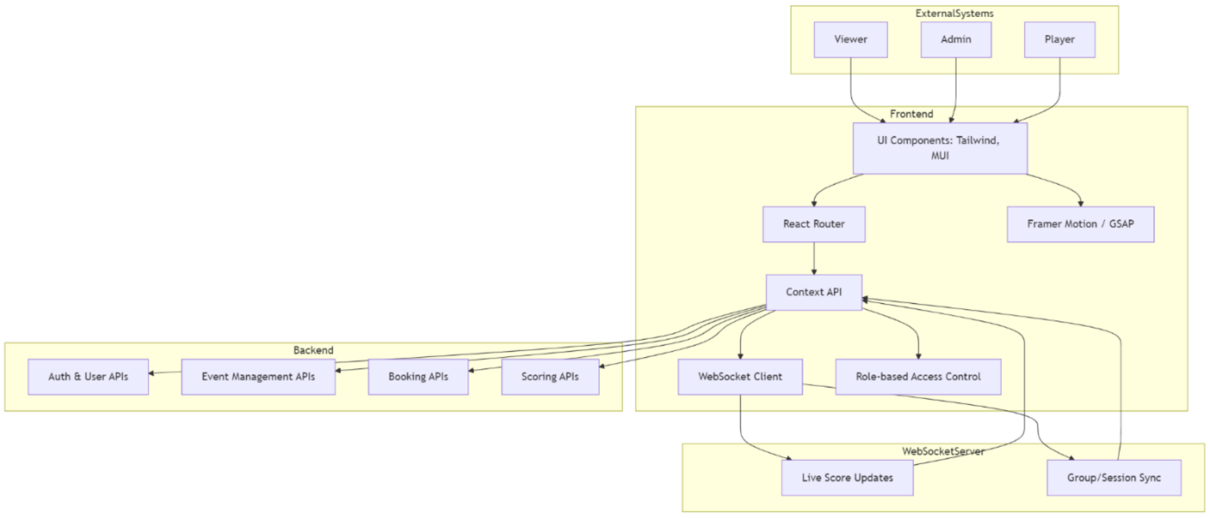


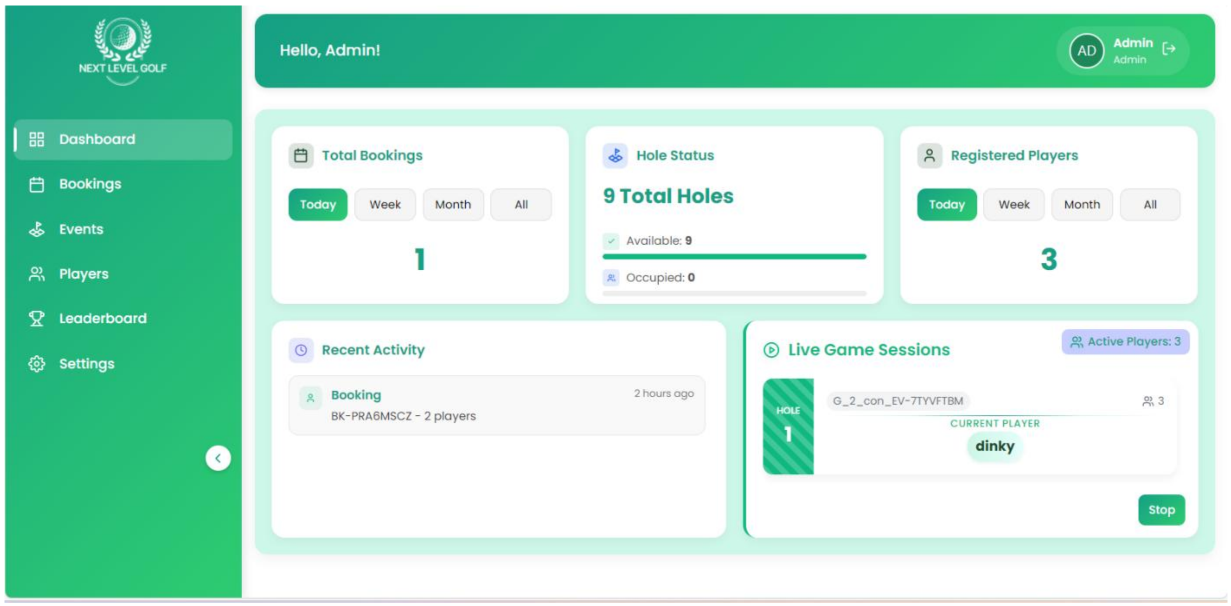
Fig 3.1 System Flow

CHAPTER 4
RESULT AND DISCUSSION

Over the course of the project, a comprehensive and scalable front-end architecture was successfully developed for a real-time golf tournament management platform. Beginning with detailed research into tournament workflows and user journeys, the system was designed with multiple user roles in mind, including Admin, Player, Viewer, and Event Organizer. A modern tech stack—featuring React, Context API, Tailwind CSS, and WebSocket-based real-time functionality—was implemented to ensure dynamic user experiences and responsive interfaces. Wireframes and UI prototypes were meticulously crafted in Figma, while scalable, reusable components were developed to streamline future expansion. Through continuous collaboration with backend teams and stakeholders, robust data handling and modularity were prioritized, supporting seamless user authentication, event creation, live score sessions, and group management.

As the system evolved, Redux was adopted for centralized state management, replacing local state and prop-drilling to enhance performance and maintainability. Real-time updates, dynamic dashboards, and responsive layouts were integrated using advanced React patterns and animation libraries such as Framer Motion and GSAP. The project emphasized both functional accuracy and engaging user experience, with features like role-based routing, intuitive booking flows, and real-time score visualizations. Overall, the platform delivers a cohesive digital solution tailored for golf event management, offering real-time interactivity, admin control, and player engagement in a highly scalable and maintainable codebase.

Role-Based Dashboards: Custom dashboards were created for each user role (currently focusing on Admin side Dashboard)



Hello, Admin!

AD Admin Admin

Total Bookings: 1

Hole Status: 9 Total Holes

Available: 9

Occupied: 0

Registered Players: 3

Recent Activity

Booking 2 hours ago
BK-PRA6MSCZ - 2 players

Live Game Sessions Active Players: 3

HOLE 1 G_2_con_EV-7TYVFTBM 3
CURRENT PLAYER
dinky

Fig 4.1 Admin Dashboard with live functionalities

Hero, HomePage and the main page that lies on the public route: Implemented the main page with scroll animation and figma animation that will be visible on the web

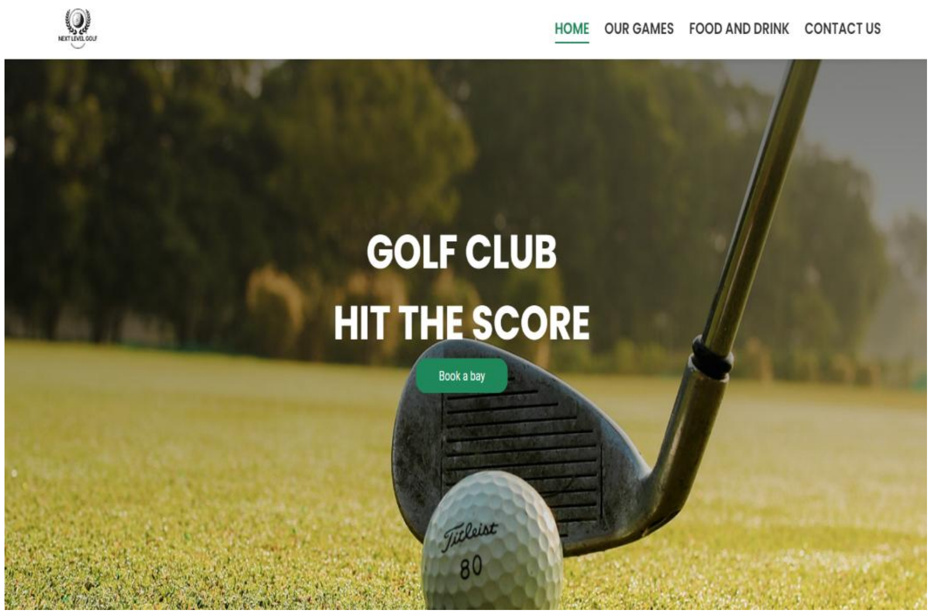


Fig 4.2: Hero Page

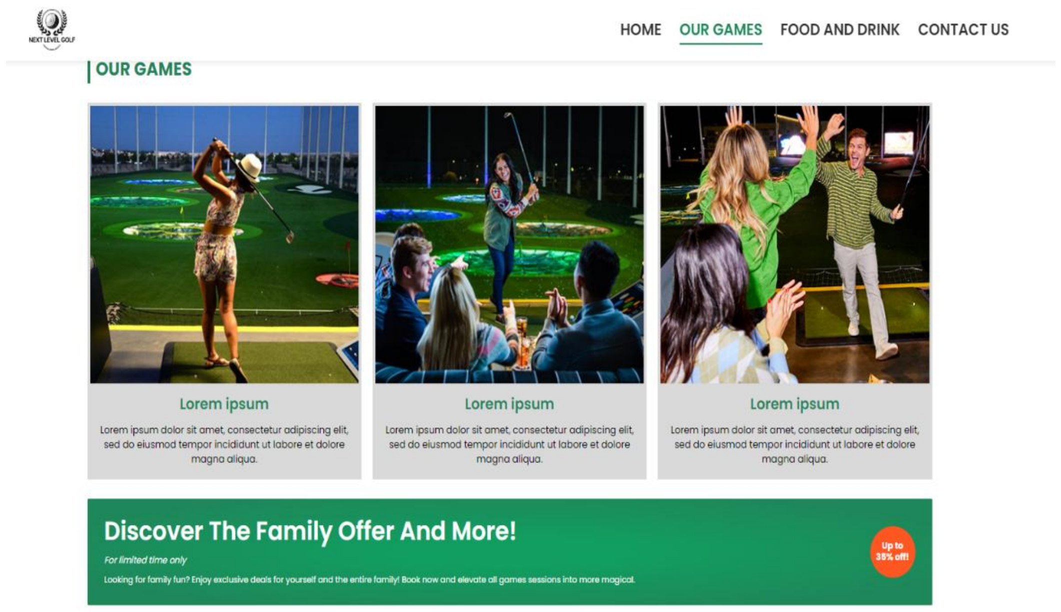

NEXT LEVEL GOLF
HOME
OUR GAMES
FOOD AND DRINK
CONTACT US
OUR GAMES
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Lorem ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Discover The Family Offer And More!
For limited time only
Looking for family fun? Enjoy exclusive deals for yourself and the entire family! Book now and elevate all games sessions into more magical.
Up to 35% off!

Fig 4.3 Section of Home Page

Authentication & Access Control: Secure login and registration flows were built with API integration, session persistence, and dynamic redirection based on user role.

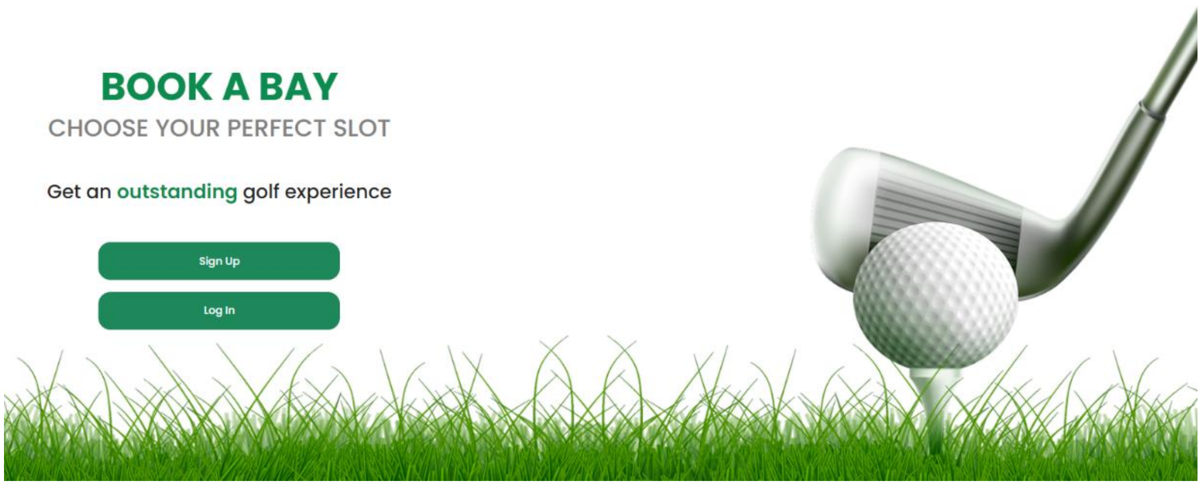
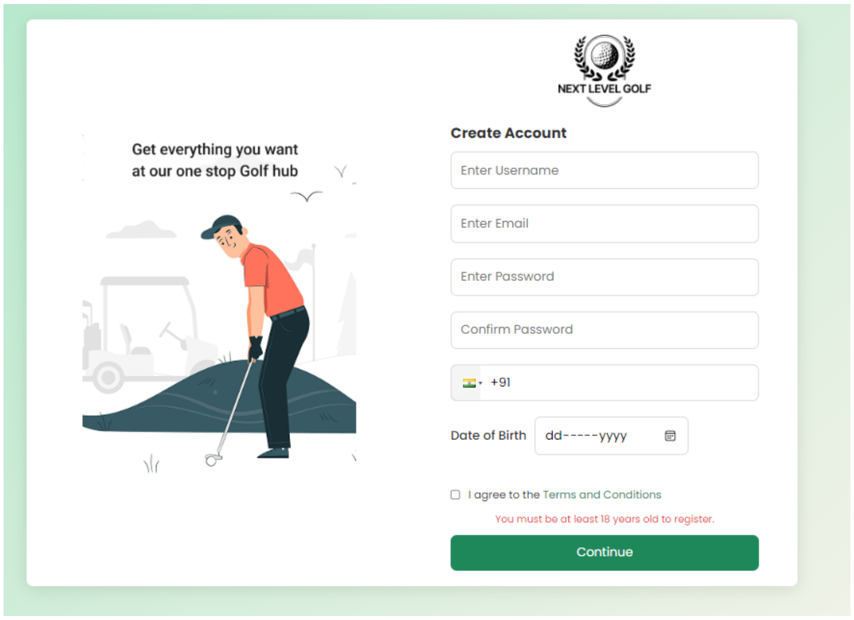


Fig 4.4 Login / Sign Up Landing Web Page



NEXT LEVEL GOLF

Get everything you want
at our one stop Golf hub

Create Account

Enter Username

Enter Email

Enter Password

Confirm Password

+91

Date of Birth dd-----yyyy

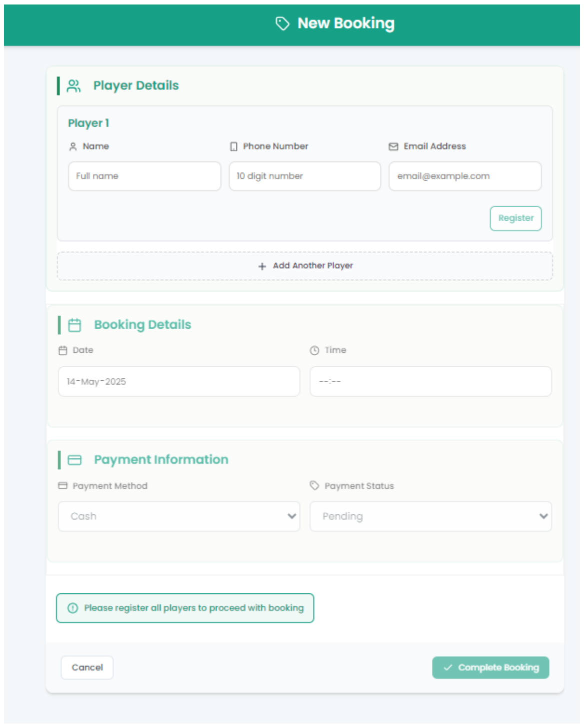
☐ I agree to the Terms and Conditions

You must be at least 18 years old to register.

Continue

Fig 4.5 Sign Up Web Page

Booking Management Module: Admins can create, edit, and monitor tournaments and bookings using a form-driven interface with real-time field validation. Bookings are reflected in the system without delay post-submission.



New Booking

Player Details

Player 1

Name	Phone Number	Email Address
Full name	10 digit number	email@example.com

Register

+ Add Another Player

Booking Details

Date	Time
14-May-2025	--:--

Payment Information

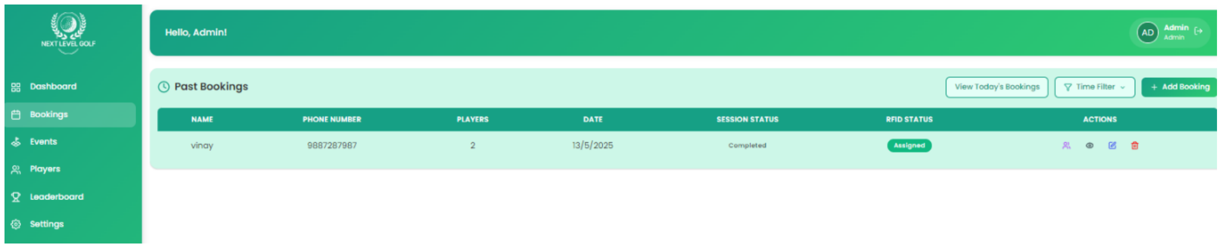
Payment Method	Payment Status
Cash	Pending

Please register all players to proceed with booking

Cancel

✓ Complete Booking

Fig 4.6 Add a New Booking Page

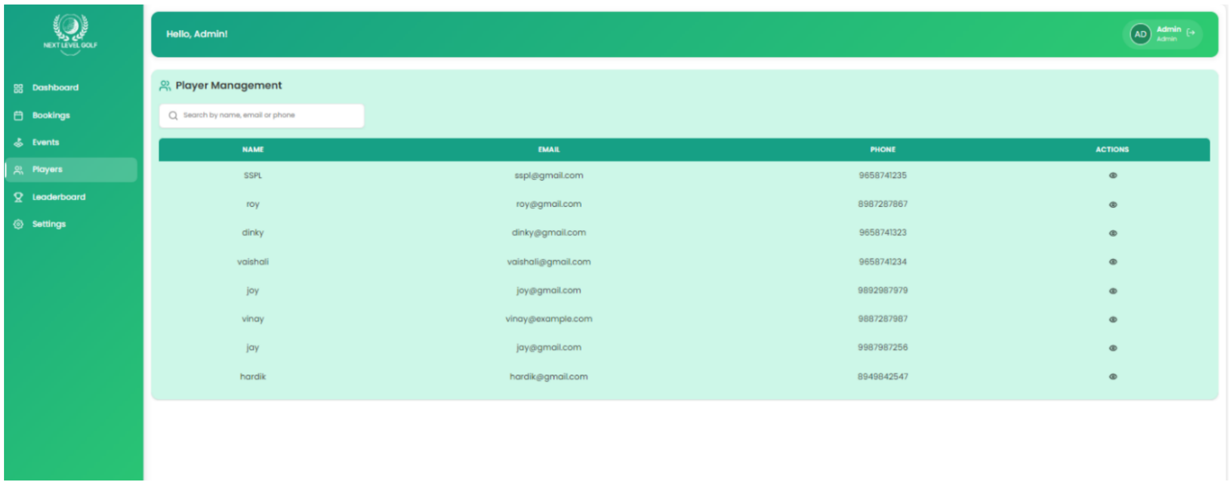


Past Bookings

NAME	PHONE NUMBER	PLAYERS	DATE	SESSION STATUS	RFID STATUS	ACTIONS
vinay	9887287987	2	13/5/2025	Completed	Assigned	

Fig 4.7 Bookings Management Table with Update, View, Delete Options

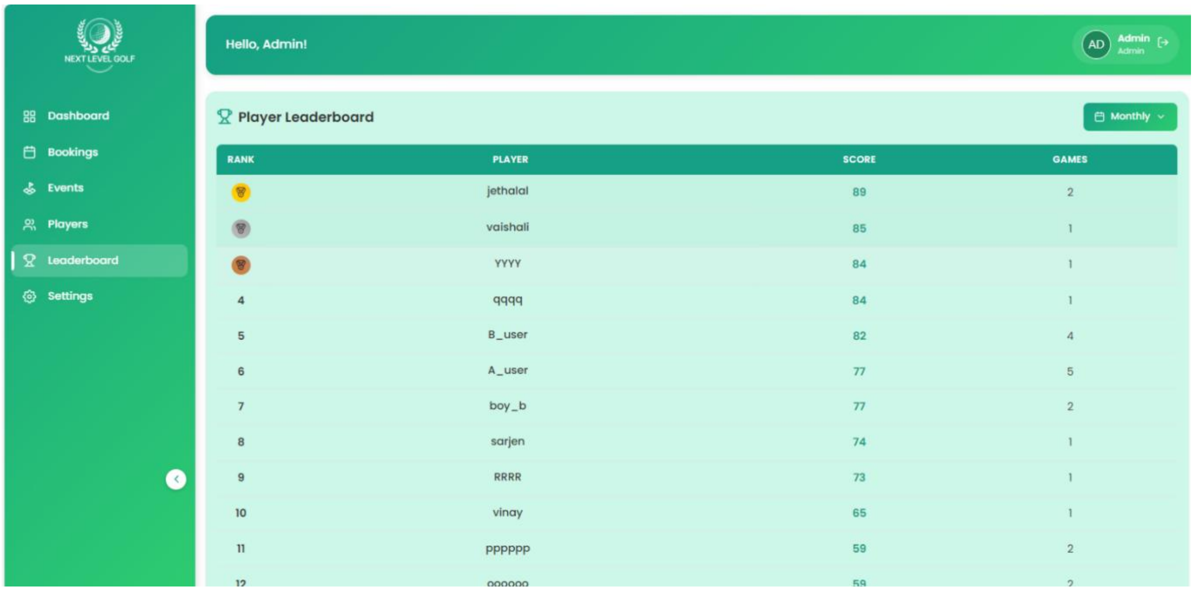
Player Details, Management and Leaderboard Module: Admins can create, edit, and view player details as well as make changes to the player information. Leaderboard details are also shown based on time filters provided



Player Management

NAME	EMAIL	PHONE	ACTIONS
SSPL	sspl@gmail.com	9658741235	
roy	roy@gmail.com	8987287867	
dinky	dinky@gmail.com	9658741323	
vaishali	vaishali@gmail.com	9658741234	
joy	joy@gmail.com	9892987979	
vinay	vinay@example.com	9887287987	
jay	jay@gmail.com	9987987256	
hardik	hardik@gmail.com	8949842547	

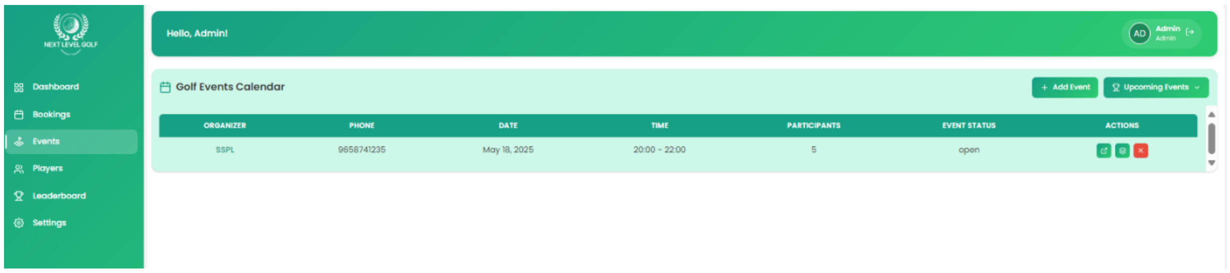
Fig 4.8 Player view and Player Management



RANK	PLAYER	SCORE	GAMES
●	jethalal	89	2
●	vaishali	85	1
●	YYYY	84	1
4	qqqq	84	1
5	B_user	82	4
6	A_user	77	5
7	boy_b	77	2
8	sarjen	74	1
9	RRRR	73	1
10	vinay	65	1
11	pppppp	59	2
12	[illegible]	59	2

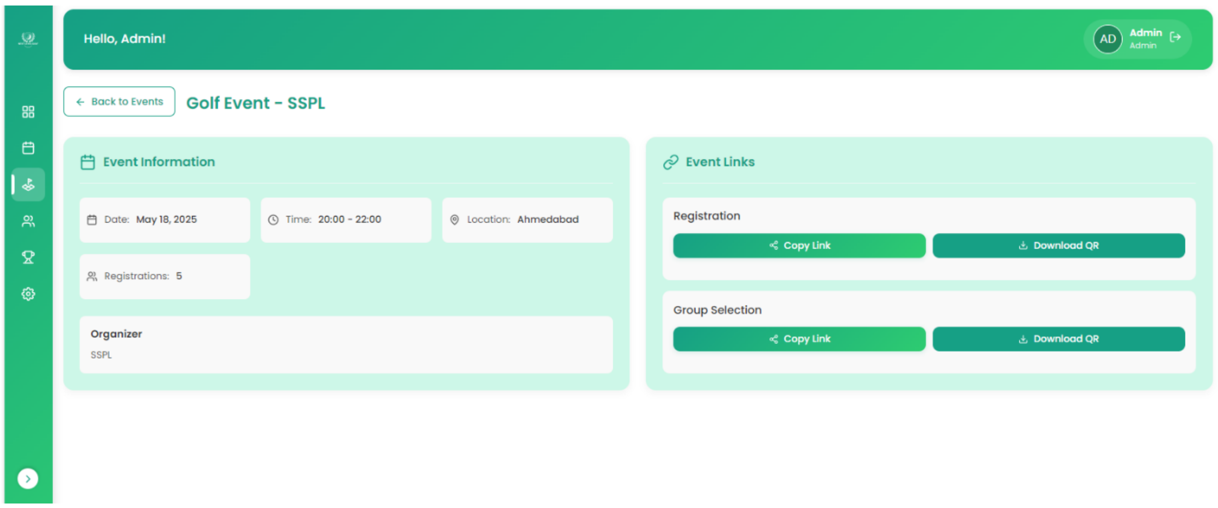
Fig 4.9 Leaderboard with time-based filters

Event Management and Group Selection Module: Admins can create, edit, and view event details based on filters, decide slot size as well as provide QR and Group registration Link and Selection link for all the registered players. Group Selection provides a BookMyShow seat selection like model with toggles such as update slot and group status.



ORGANIZER	PHONE	DATE	TIME	PARTICIPANTS	EVENT STATUS	ACTIONS
SSPL	9658741235	May 18, 2025	20:00 - 22:00	5	open	🟩 🟩 🟥

Fig 4.10 Events Callender Section with View, edit and close toggles


Hello, Admin!

AD Admin Admin

Back to Events

Golf Event - SSPL

Event Information

Date: May 18, 2025

Time: 20:00 - 22:00

Location: Ahmedabad

Registrations: 5

Organizer

SSPL

Event Links

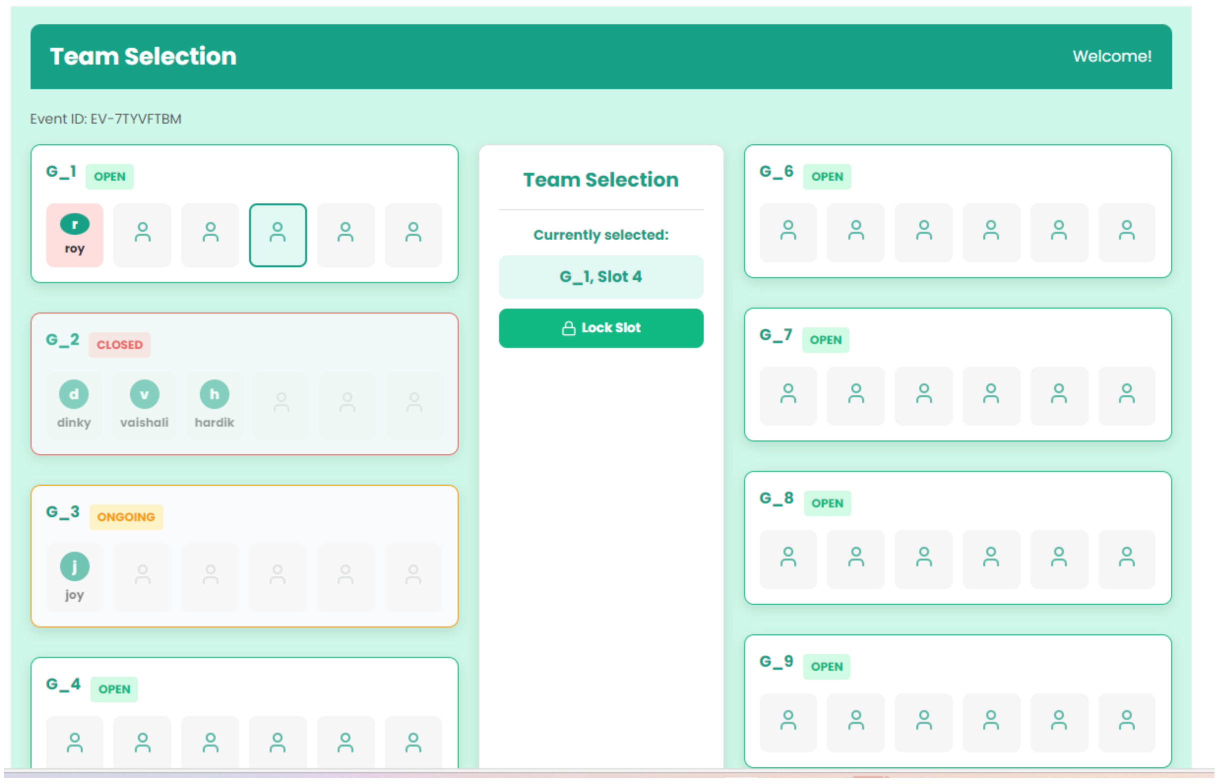
Registration

Copy Link | Download QR

Group Selection

Copy Link | Download QR

Fig 4.11 Events Information section with Group Selection and Player registration QR and links


Team Selection

Welcome!

Event ID: EV-7TYVFTBM

G_1 OPEN — r roy

G_2 CLOSED — d dinky, v vaishali, h hardik

G_3 ONGOING — j joy

G_4 OPEN

Team Selection

Currently selected:

G_1, Slot 4

Lock Slot

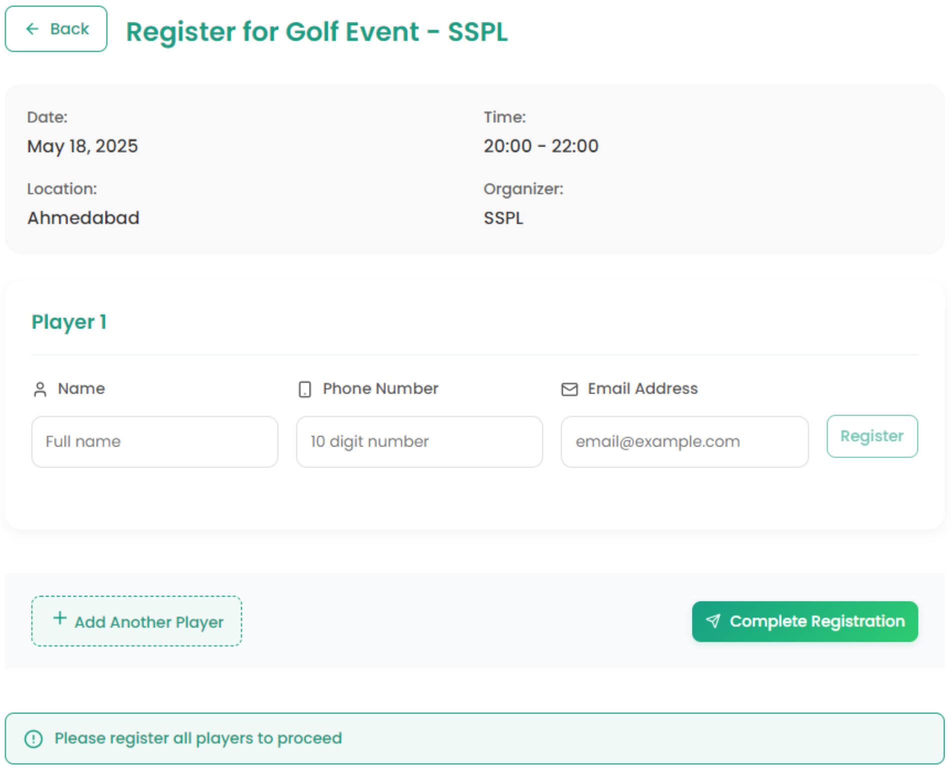
G_6 OPEN

G_7 OPEN

G_8 OPEN

G_9 OPEN

Fig 4.12 Group Selection



Back

Register for Golf Event - SSPL

Date: May 18, 2025

Time: 20:00 - 22:00

Location: Ahmedabad

Organizer: SSPL

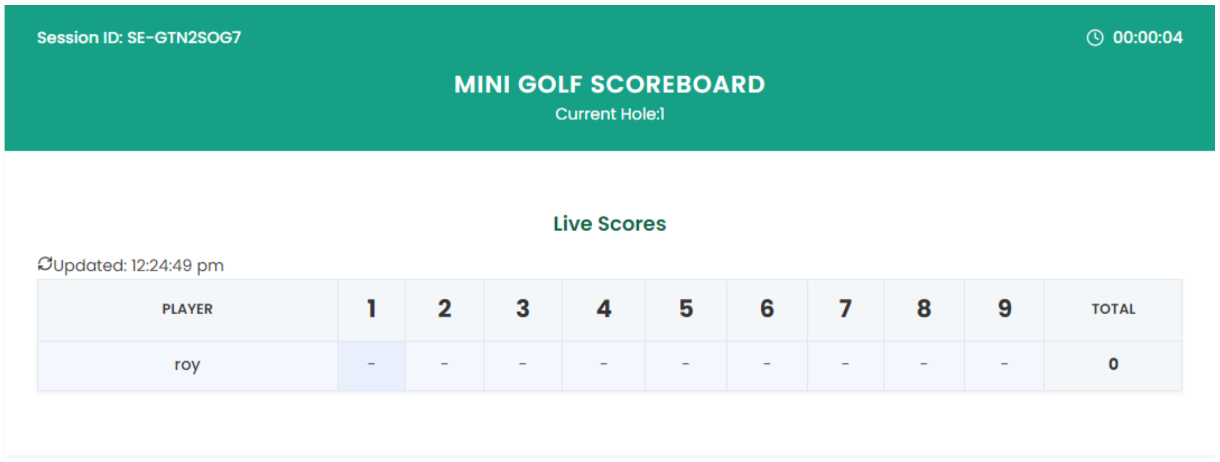
Player 1

Name	Phone Number	Email Address	
Full name	10 digit number	email@example.com	Register

Please register all players to proceed

Fig 4.13 Player Registration

Live Scoreboard view Module: Admins can view Live Scoreboard session for both events and Bookings in Live Scoreboard session with the use of Web Socket and API .



Session ID: SE-GTN2SOG7

00:00:04

MINI GOLF SCOREBOARD

Current Hole:1

Live Scores

Updated: 12:24:49 pm

PLAYER	1	2	3	4	5	6	7	8	9	TOTAL
roy	-	-	-	-	-	-	-	-	-	0

Fig 4.14 Live Scoreboard Table

CHAPTER 5
CONCLUSION

Building a real-time golf tournament management platform is an effective illustration of how web technologies can be used to address domain-related issues in sport event organization. Through the incorporation of scalable architecture and user-driven design, the system provides a seamless experience in handling various roles—Admins, Players, Organizers, and Viewers.

A key aspect like the synchronization of live scores via WebSockets, role-based dashboards, and an interactive model of booking serves as a solid foundation for digitalizing the workflow of the tournament. The project demonstrates best practices for frontend development like the utilization of reusable components, state management with ContextAPI, responsive UI patterns, and real-time data handling. These practices made a significant contribution to performance as well as maintainability.

Throughout the development cycle, attention was directed carefully toward scalability, modularity of code, and responsiveness of the system. Consequently, the platform not only satisfies its original requirements for function but also sets a flexible foundation for future growth—such as enhanced analytics, animation-based user interfaces, and possibly 3D course visualizations.

This thesis project identifies how careful system design and current development practices can solve complicated real-world issues with a technically robust solution ready to be implemented in real-world environments of competitive sport.

CHAPTER 6

FUTURE SCOPE

Although the existing system is a good starting point for real-time golf tournament management, there are a number of opportunities for further developing its functionality, performance, and usability. These range from short-term enhancements through to longer-term, groundbreaking expansions:

1. **Advanced Analytics & Visualizations**
   Combining data analytics dashboards for trend performance, player statistics, and event analysis.
   Displaying score trend progress, player reliability, and hole-by-hole statistics utilizing libraries such as D3.js or Chart.js.
2. **3D Course Views and Animations**
   Using Three.js to create 3D simulations of golf courses for richer visual representation.
   Animating swing trajectories, ball flight, or hole-in progressions for an interactive live scoreboard.
3. **Mobile-First Native Experience**
   Converting the platform to a Progressive Web App (PWA) or creating a native mobile version based on React Native to enhance mobility usability, particularly for players and field officials.
4. **Full Migration to Redux Toolkit**
   While State Management is already being utilized, some local state parts and legacy useEffect-based data fetching still persist. A full migration would:
   Consolidate all state changes under a predictable, scalable Redux paradigm
   Swap useState/useEffect logic with createAsyncThunk and createSlice patterns
   Allow stronger debugging and time-travel support through Redux DevTools
   Enhance data consistency across deeply nested trees of components
   This change would improve maintainability, enable improved test coverage, and lay the groundwork for real-time state synchronization at scale.

5. **AI Integration for Increased Functionality**
   Adding artificial intelligence can unlock new levels of functionality:
   AI-driven player matching by performance metrics, play history, and fairness algorithms
   Predictive analytics to predict winners or recommend strategies from historical data
   Integration with chatbots or voice assistants to enable users to book slots or monitor leaderboard status voice-free
   Image recognition for score validation through scanned scorecards (OCR)
   These features would bring the platform in line with contemporary AI-enabled user expectations.
6. **Interactive and Immersive Experiences**
   Further delve into libraries such as Three.js for 3D course modeling and virtual walkthroughs.
   Include audio-visual additions using animations driven by GSAP, Framer Motion, and sound effects during game milestones.
   Implement gamification aspects such as badges, achievements, or match replays to further engage players.

These future developments are focused on transforming the system from a rock-solid MVP into a production-grade, smart, and immersive experience. With the technical groundwork in place, the system is now ready to serve sophisticated functionality, grow across large audiences, and adopt current trends like automation and AI.

REFERENCES

Books

[1] M. Garreau and W. Kwan, “Advanced Redux Concepts”, in *Redux in Action*. New York: Manning Publications, 2018, ch. 3, pp. 65–112

[2] D. Bugl, "Advanced Patterns and Performance in Redux", in *The Complete Redux Book*, ch. 4, 2nd ed. Leanpub, 2020, pp. 40–78

[3] R. Wieruch, "React State Management", in *Taming the State in React*, ch. 5, 3rd ed. Berlin, Germany: Leanpub, 2021, pp. 22–45

[4] G. L. Muller, "WebSocket Protocol Overview", in *HTML5 WebSocket Protocol and Its Application to Distributed Computing*, ch. 4, Berlin, Germany: Springer, 2014, pp. 130–145

Handbooks

[1] *Redux Toolkit Documentation*, 2023, Redux.js.org, Chicago, IL, pp. 1–30.

[2] *WebSocket API (MDN)*, Mozilla Developer Network, 2023, Mozilla Foundation, Mountain View, CA, pp. 15–25.

Reports

[1] E. E. Reber et al., “WebSocket-based real-time data synchronization in live sports applications,” Mozilla Research Group, Mountain View, CA, Tech. Rep. MR-2045, Jan. 2022

[2] A. Jain and S. Prasad, “Improving React performance with component memoization and state management patterns,” Meta Open Source, Menlo Park, CA, Tech. Rep. META-RP-987, Dec. 2021

Online Videos

[1] Academind, “Redux Tutorial – Learn Redux from Scratch,” *YouTube*, video, 36:24, Aug. 17, 2020. [Online]. Available: https://www.youtube.com/watch?v=poQXNp9ItL4

[2] Web Dev Simplified, “React Router Tutorial – Complete Guide,” *YouTube*, video, 21:17, Feb. 8, 2022. [Online]. Available: https://www.youtube.com/watch?v=Law7wfdg_ls

[3] Traversy Media, “WebSockets Crash Course – Real-time Communication,” *YouTube*, video, 29:12, Apr. 10, 2019. [Online]. Available: https://www.youtube.com/watch?v=1BfCnjr_Vjg