NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(DATA SCIENCE)

A
MINI PROJECT REPORT

ON

"Sports Arena Booking System"

Submitted in the partial fulfillment of the requirements in the 5th semester of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

BY
K Niranjan 1NH22CD042
Kiran Kumar A 1NH22CD051

Under the guidance of
T Sasikala
Sr. Assistant Professor
Dept of CSE(DS), NHCE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

NEW HORIZON COLLEGE OF ENGINEERING
(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NBA & NAAC with 'A' Grade)
Ring Road, Bellandur Post, Near Marathalli,
Bangalore-560103, INDIA

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(DATA SCIENCE)

CERTIFICATE

We hereby certify that, the report entitled **"Sports Arena Booking System"** as a part of Mini Project Component in partial fulfillment of the requirements during 5th semester Bachelor of Engineering in Computer Science and Engineering (Data Science) during the year 2024- 2025(September 2024 – January 2025) is an authentic record of our own work carried out by **K Niranjan(1NH22CD42)** and **Kiran Kumar A(1NH22CD051)**, Bonafide students of NEW HORIZON COLLEGE OF ENGINEERING.

Name & Signature of Students

K Niranjan
Kiran Kumar A

Name & Signature of Guide — Ms. T Sasikala

Name & signature of HOD — (Dr. Swathi B)

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project. We would like to take an opportunity to thank them all.

First and foremost, we thank the management, **Dr. Mohan Manghnani,** Chairman, New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

We would like to thank **Dr. Manjunatha**, Principal, New Horizon College of Engineering, Bengaluru, for his constant encouragement and facilities extended to us towards completing our project work.

We extend my sincere gratitude to **Dr. Swathi B**, Associate Professor & Head of the Department, Computer Science and Engineering (Data Science), New Horizon College of Engineering, Bengaluru for her valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Ms. T Sasikala, Sr. Assistant Professor**, Department of Computer Science and Engineering (Data Science), New Horizon College of Engineering, Bengaluru, for her able guidance, regular source of encouragement and assistance throughout this project.

We thank our Parents, and all Faculty members of Department of Computer Science and Engineering (Data Science) for their constant support and encouragement.

Last, but not the least, we would like to thank our peers and friends who provided us with valuable suggestions to improve our project.

K Niranjan(1NH22CD042)

Kiran Kumar A(1NH22CD051)

ABSTRACT

The sports arena booking system is a web application designed to simplify the process of reserving badminton courts while offering scalability to include other sports facilities. Built using React and powered by Firebase for backend and database management, the system ensures a seamless and efficient booking experience.

The platform focuses on a streamlined user interface tailored for badminton enthusiasts, enabling users to quickly register, log in, and book courts based on real-time availability. Designed with simplicity in mind, the interface allows users to navigate effortlessly, ensuring a hassle-free booking process for individuals and groups alike.

The integration of Firebase provides robust real-time updates, ensuring accurate availability status and secure management of user data. The system's modular design enables future expansion to support additional sports facilities and features such as payment integration, notifications, and membership tracking. By addressing the challenges of traditional booking methods, this system enhances user convenience, promotes sports participation, and redefines facility management for sports arenas.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

The sports arena booking system is a web application designed to simplify the process of booking badminton courts, with potential to expand to other sports facilities in the future. Developed using React for the front end and Firebase for the backend, the system offers a user-friendly platform for easy court reservations. Users can quickly register, log in, and book courts through a simple and intuitive interface. Firebase ensures secure data management and efficient handling of user information. The system is designed to be scalable, with plans for future features like payment integration, notifications, and support for additional sports. This project aims to streamline the traditional booking process, enhancing convenience for users and encouraging increased participation in sports activities.

1.1 Purpose of study

Purpose:

The Sports Arena Booking System is designed to simplify and enhance the process of reserving sports facilities, making it more accessible and efficient for users. Its primary goal is to provide a user-friendly platform that allows individuals to easily browse, book, and manage reservations for various sports facilities without the need for cumbersome procedures or in-person visits. By offering real-time availability, the system ensures that users can quickly find suitable time slots for their desired activities, which significantly reduces the frustration often associated with traditional booking methods. This online reservation capability allows users to make bookings anytime and from anywhere, catering to the needs of a modern, mobile-driven audience. The user experience is further enhanced by personalized accounts that allow users to manage their bookings, view their history, and receive tailored recommendations. Automated notifications keep users informed about their reservations, cancellations, or any changes, ensuring they remain engaged with the platform. Moreover, security is a paramount concern; the system employs robust authentication measures to protect user information and maintain data integrity. By utilizing Firebase as its backend, it guarantees secure storage and easy retrieval of user data and booking records.

Ultimately, the Sports Arena Booking System aims to promote community engagement in sports activities by making it easier for individuals to participate in physical pursuits.

1.2 Problem Statement

Users struggle with the cumbersome and inefficient manual booking process for badminton courts, leading to issues such as errors, double bookings, and operational challenges. The absence of a digital solution creates inconvenience, making it difficult for users to reserve courts smoothly. An online booking system is needed to automate and simplify the reservation process. This solution will provide an intuitive platform for users to easily book courts, eliminating errors and improving the overall user experience. The system aims to enhance convenience, reliability, and accessibility for users, making the booking process seamless and efficient.

1.3 Motivation of project

The motivation behind the Sports Arena Booking System stems from the difficulties users face with traditional manual booking methods for badminton courts. These manual systems often result in errors, double bookings, and delays, leading to frustration and wasted time for users. As digital solutions become more integral to everyday life, there is a strong need for a more efficient and user-friendly booking system. This project seeks to address these challenges by creating a web-based system that automates the booking process, reduces errors, and provides real-time updates. By simplifying the process and enhancing user experience, the system aims to encourage more people to engage in sports while making the court booking process more convenient and accessible.

1.4 Methodology

The development of the Sports Arena Booking System follows a well-structured approach to ensure the creation of a user-friendly, efficient, and reliable application. The process includes several key phases, each focused on ensuring the system meets user needs and performs optimally. The methodology is as follows:

1. **Requirement Gathering**
   The first phase involves collecting detailed requirements through various techniques like surveys, interviews, and user feedback. This helps understand the users' needs, preferences, and pain points regarding the booking process. The goal is to identify essential features such as a simple and intuitive interface, real-time availability, secure booking, and overall ease of use. The insights from this phase are used to define the core functionalities of the system, such as user registration, booking management, and notifications.
2. **System**
   Based on the gathered requirements, the design phase focuses on planning the overall structure of the system. This includes defining the system architecture, selecting appropriate technologies, and designing the user interface.
   - **User Interface (UI) Design**: Wireframes and mock-ups are created to visualize the layout of the web application. The design prioritizes simplicity and ease of use to ensure a smooth experience for users.

- **Backend Architecture**: Firebase is chosen as the backend service due to its real-time database, authentication services, and scalability. The database schema is designed to efficiently manage user data, court bookings, and other system-related information.
- **Technology Stack**: React is selected for the frontend development to provide a responsive, dynamic, and modern user interface, while Firebase handles the backend tasks.

3. **Development**
   During the development phase, the application is built using the chosen technology stack:
   - **Frontend Development**: React is used to create an interactive and responsive frontend. This ensures that the user interface is smooth, with real-time updates for bookings and availability. React's component-based structure allows for a modular and maintainable codebase.
   - **Backend Development**: Firebase handles various backend operations, including user authentication, data storage, and real-time updates. Firebase's real-time database ensures users always see the latest booking status without needing to refresh the page.
   - **Integration**: Both the frontend and backend are integrated, ensuring smooth data flow and interaction between the user interface and the database. Integration testing is carried out to ensure that the system works seamlessly as a whole.
4. **Testing**
   Rigorous testing is crucial to ensure the application functions as expected and meets user needs. The following testing phases are conducted:
   - **Unit Testing**: Each component of the application is tested independently to ensure it performs as expected. For example, testing the booking functionality, user registration, and login processes.
   - **Integration Testing**: This phase ensures that different parts of the system, including the frontend, backend, and database, interact correctly. It tests the communication between React and Firebase, ensuring data is transferred accurately.
   - **User Acceptance Testing (UAT)**: Real users test the system in a controlled environment to assess the ease of use, reliability, and overall experience. Feedback from users is collected to identify any issues or areas for improvement. Any issues discovered during testing are addressed before moving to deployment.
5. **Deployment**
   After successful testing, the application is deployed to production. Firebase Hosting is used to deploy the web application, providing fast, secure, and scalable hosting. The system is made available to users, and the deployment process includes configuring the domain, setting up the hosting environment, and ensuring proper data flow between the front end and the backend.

Chapter 2

System Requirements and Language Used

2.1 Hardware and software Requirements

Hardware System Configuration:

- **Processor**: Any Intel Core Processor
- **Speed**: 2.0 GHz
- **RAM**: 256 MB (Minimum)
- **Hard Disk**: Not necessary

Software System Configuration:

- **Operating System**: Windows
- **Programming Language**: JavaScript (React for frontend development)
- **Interpreter**: Node.js
- **IDE/Editor**: Visual Studio Code (VS Code) or any text editor for coding (e.g., Sublime Text)
- **Backend**: Firebase (Cloud platform, no installation required)
- **Browser**: Google Chrome, Mozilla Firefox, Safari (For accessing the web app)

2.2 About the language

JavaScript (Frontend)

JavaScript is a versatile and essential programming language for frontend web development. It brings websites and web applications to life by allowing developers to create interactive elements, dynamic content updates, and responsive features without requiring page reloads. For the Sports Arena Booking System, JavaScript enables users to interact with the platform in real time, such as booking a court, checking availability, and receiving instant updates on availability status or booking confirmations.

- **Interactivity**: JavaScript enhances the user experience by enabling interactive elements like buttons, forms, and navigation menus. For instance, when a user selects a time slot for a badminton court, JavaScript dynamically updates the available slots without needing to refresh the page. This real-time interaction is crucial for the booking process and ensures a smooth flow of activities.
- **DOM Manipulation**: JavaScript interacts with the Document Object Model (DOM), which represents the structure of the web page. By manipulating the DOM, JavaScript can change the content and structure of the webpage in response to user actions. This allows the application to update available court times, show booking confirmations, or display user alerts instantly, improving the overall user experience.
- **Single-page Application (SPA)**: One of JavaScript's key strengths is enabling Single-page Applications (SPA). In an SPA, all the content is loaded once, and the page dynamically updates as the user interacts with it. This means users don't have to wait for new pages to load, which speeds up navigation and creates a more fluid and enjoyable experience. For the booking system, users can seamlessly browse available courts, view booking statuses, and confirm reservations without having to wait for page refreshes.
- **Event Handling**: JavaScript listens for user events like clicks, form submissions, or hover actions, and responds to these events with predefined actions. For example, when a user clicks the "Book Now" button, JavaScript can trigger the booking process, sending the request to the backend (e.g., Firebase) to reserve the court. This real-time interaction is crucial for maintaining the fluidity of the booking process and ensuring that users feel in control.
- **Integration with Other Technologies**: JavaScript integrates with various other technologies used in the Sports Arena Booking System, such as **React** and **Firebase**. React, a JavaScript library, uses JavaScript to efficiently update the user interface in response to changes in data. JavaScript is also used to interact with Firebase's real-time database, making it easy to store, retrieve, and display user data and court availability.
- **Asynchronous Operations**: JavaScript's ability to handle asynchronous operations, using features like **Promises** and **async/await**, allows the system to perform tasks like fetching real-time data from Firebase without blocking the user's interactions. For example, while the user is selecting a time slot, JavaScript can silently fetch and display the most up-to-date

information from the Firebase database in the background, ensuring that the UI stays responsive.

- **Cross-Platform Compatibility**: As a language supported by all modern browsers, JavaScript ensures that the Sports Arena Booking System works consistently across different devices, including desktops, tablets, and smartphones. This cross-platform compatibility is vital to reach a wide user base and ensure that the system is accessible wherever and whenever users need to book their courts.

React (Frontend Library)

- **Component-based Architecture**: React allows for building reusable UI components, making the app easier to maintain and scale.
- **Efficient Rendering**: React uses a virtual DOM to efficiently update and render components, ensuring a smooth and fast user experience.
- **Responsive User Interface**: React ensures the UI remains dynamic and responsive, providing a seamless experience for users interacting with the system.

Firebase (Backend)

- **Real-time Data Sync**: Firebase's real-time database ensures that data, such as court availability, is updated instantly for all users.
- **Simplified Backend Management**: Firebase handles user authentication and database management, reducing the complexity of backend development.
- **Secure Authentication**: Firebase Authentication provides secure login and registration processes, protecting user data and access.

Chapter 3

Architecture

3.1 Architecture

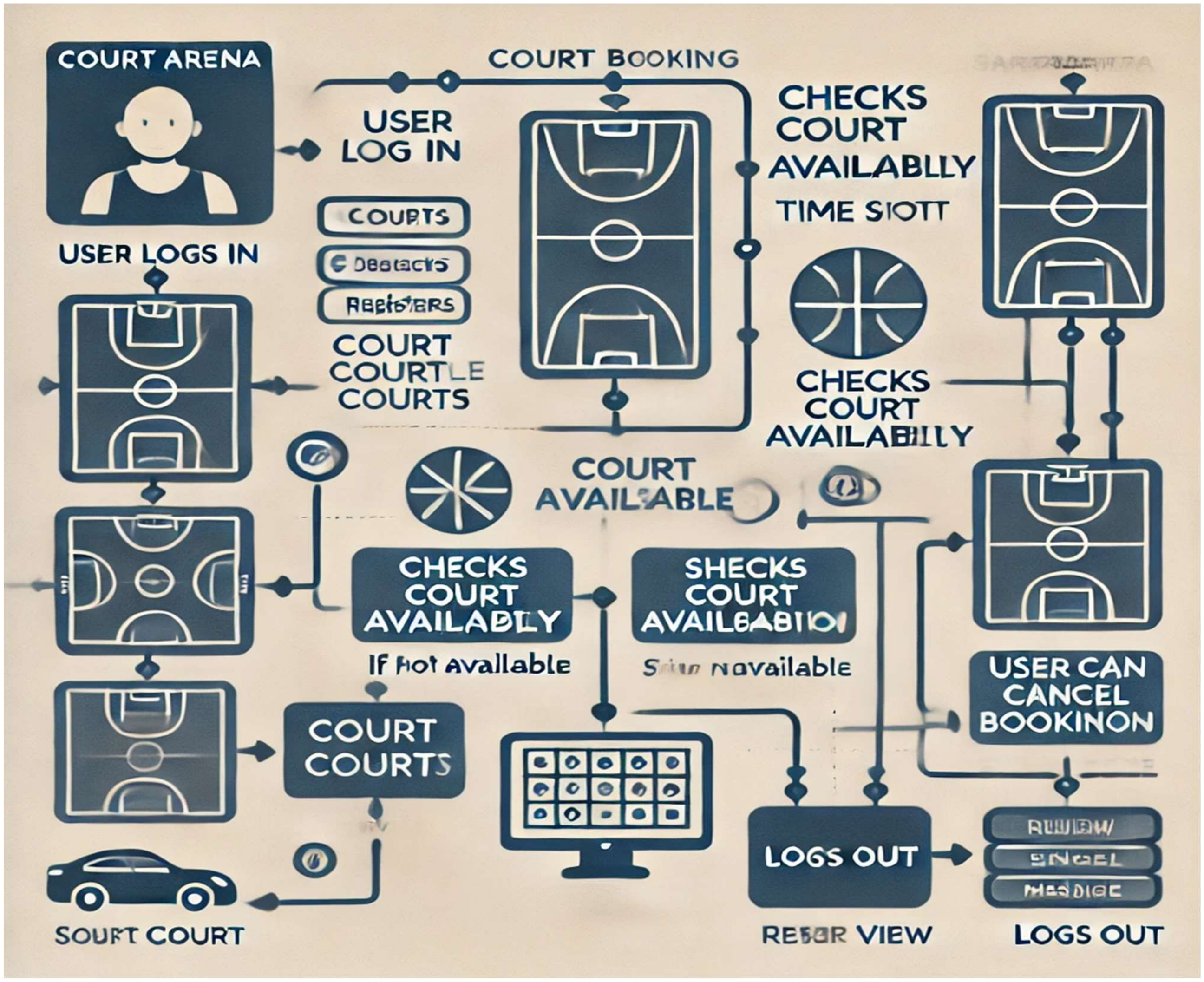


Figure 3.1 Architecture

The architecture of a house price prediction model involves several key steps to ensure accurate and reliable predictions. The process begins with data collection from various sources such as real estate listings, public records, and government databases, focusing on features like location, size, number of bedrooms and bathrooms, year built, lot size, condition, and proximity to amenities. The collected data is then preprocessed by cleaning to handle missing values and correct errors, normalizing or standardizing numerical features, and encoding categorical variables into a numerical format. Exploratory data analysis (EDA) is conducted to visualize relationships between features and understand their statistical properties.

Feature selection and engineering are crucial for identifying and creating relevant variables, ensuring that the model captures the most important aspects of the data. Various machine learning algorithms, such as linear regression for its simplicity and interpretability, decision trees and random forests for handling non-linear relationships, and gradient boosting machines for their powerful predictive capabilities, are considered for model selection. Cross-validation techniques are used to tune model parameters and ensure robustness.

Once the model is trained, its performance is evaluated using metrics like mean absolute error (MAE), Linear Regression, Ridge. The final model is then deployed through an API, enabling real-time predictions for users. Continuous monitoring and periodic retraining with new data are essential to maintain the model's accuracy and adapt to changing market conditions. This systematic approach ensures that the house price prediction model is robust, reliable, and capable of providing valuable insights for real estate decisions.

3.2 Algorithm

Step 1: User Registration & Login

1. **User enters registration/login page**:
   - Input: User's email and password (for registration), or email and password (for login).
2. **Check user credentials**:
   - If **user is registering**:
     - Store user's email, password, and other necessary details in the Firebase Authentication system.
     - Show a confirmation message for successful registration.
   - If **user is logging in**:
     - Authenticate the user credentials using Firebase Authentication.
     - If successful, redirect to the dashboard (court booking page).
     - If unsuccessful, display an error message.

Step 2: View Available Courts

3. **Display available courts**:
   - Fetch available badminton courts from the Firebase real-time database.
   - Display court details such as court name, location, and available time slots for booking.
4. **Select a time slot**:
   - Input: User selects a court and time slot.
   - Validate the time slot:
     - Check if the selected time is available (not already booked).
     - If the time is available, proceed to the next step.
     - If not, display an error message (e.g., "Court not available for the selected time").

Step 3: Booking the Court

5. **User confirms the booking**:
   - Input: User clicks "Confirm Booking" button.
   - Check if the selected time slot is still available in the database.
   - If available:
     - Reserve the court by updating the database (marking the court as booked for the selected time).
     - Store booking details (user, court, time slot) in Firebase database.
     - Show a confirmation message with booking details (court name, time, and confirmation number).
   - If unavailable:
     - Display an error message (e.g., "Booking failed. Please try again later").

Step 4: Manage Bookings

6. **View booked courts**:
    - Allow users to view their upcoming bookings.
    - Fetch booked court data from the Firebase database.
    - Display booking details such as court name, time slot, and booking confirmation number.
7. **Cancel a booking (optional)**:
    - Input: User selects a booking to cancel.
    - Verify the booking details.
    - If the booking exists, allow the user to cancel.
    - Update the Firebase database by marking the court as available for the selected time slot.
    - Show a cancellation confirmation message.

Step 5: Real-time Updates

8. **Real-time court availability**:
    - Use Firebase real-time database to sync court availability.
    - Whenever a booking is made or canceled, the system automatically updates and reflects these changes across all users.
9. **Display real-time updates**:
    - Users can see the latest court availability without refreshing the page.

Step 6: User Logout

10. **Logout**:
    - Input: User clicks the "Logout" button.
    - Action: Firebase Authentication logs the user out.
    - Redirect to the login page.

This algorithm covers the main features and processes involved in booking a court and managing bookings.

Key Points of the Algorithm:

1. **User Registration & Authentication**: Handles user creation and login via Firebase Authentication.
2. **Court Availability & Booking**: Ensures real-time court availability is displayed and allows users to book available slots.
3. **Database Updates**: Ensures that the Firebase database is updated with each booking or cancellation to maintain accurate records.

3.3 Flow Diagram

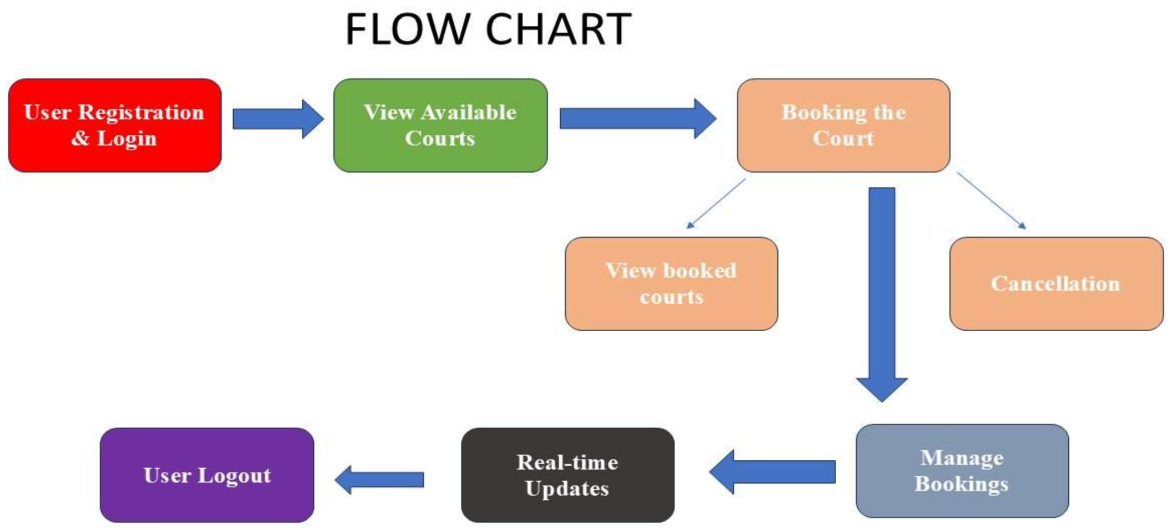


Figure 3.2 : Flow chart for the Sports Arena Booking System

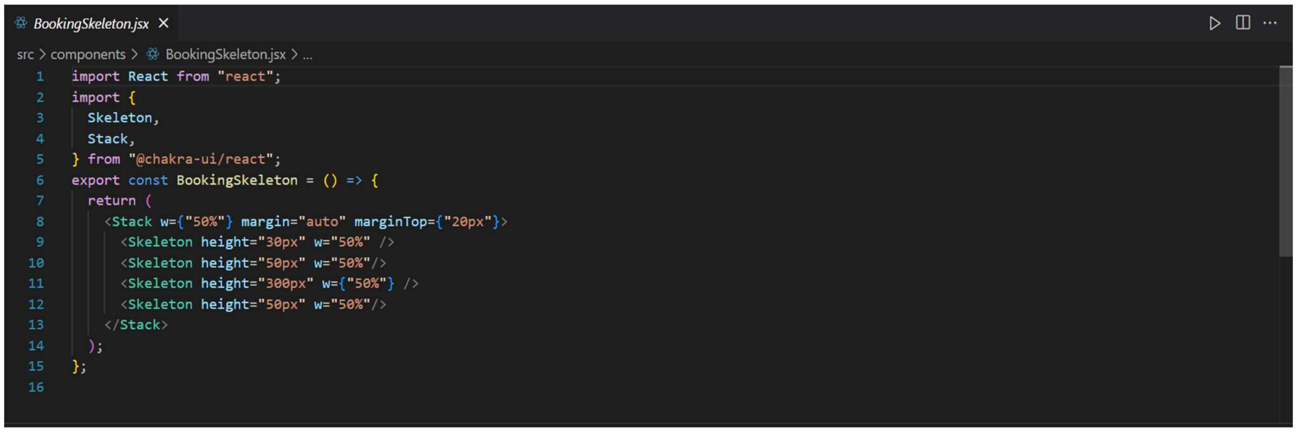
The flowchart outlines the step-by-step process for the **Sports Arena Booking System**. Here's a detailed explanation of the flowchart:

1. **User logs in or registers**:
    - The process begins when the user either logs into their existing account or registers for a new one.
    - If the user is new, they provide the necessary details like email, password, and possibly additional personal information to create an account.
2. **Displays available badminton courts and time slots**:
    - Once the user logs in, the system fetches and displays a list of available badminton courts along with their time slots.
    - This step provides the user with the options to select which court and time they want to book.
3. **User selects a court and time slot**:
    - The user selects their desired badminton court and time slot.
    - At this stage, the user specifies their preferences for booking the court.

4. **System checks court availability**:
   - The system checks if the selected court and time slot are available.
   - This step validates whether the user's choice conflicts with any existing bookings.
   - **Decision Point**: The flow splits based on the availability of the court at the selected time:
     - If **court is available**, proceed to the next step where the user confirms their booking.
     - If **court is not available**, the system will display an error message and prompt the user to either choose a different time slot or court.
5. **Display error message**:
   - If the court is not available for the selected time slot, an error message is displayed, informing the user that the booking cannot proceed.
   - The user then has the option to either choose another time or log out.
6. **User logs out**:
   - In case the user chooses to exit the booking process, they can log out of their account.
7. **User proceeds to confirm booking**:
   - If the court is available, the user proceeds to confirm the booking. This involves confirming their selection of the court and time slot.
8. **System stores the booking in the database**:
   - After confirmation, the system updates the database (Firebase) with the booking details, such as the user's information, selected court, and time slot.
   - This ensures the booking is recorded and the court is marked as reserved.
9. **Displays booking confirmation message with details**:
   - After successfully storing the booking, the system shows the user a confirmation message with all relevant details about their booking, such as the court, time, and confirmation number.
10. **User can view or cancel existing bookings**:
    - The user has the option to view their upcoming bookings or cancel any existing bookings.
    - This gives flexibility to the user to manage their reservations directly through the system.
11. **Real-time updates reflected**:
    - The system uses real-time data updates (via Firebase) to reflect the latest status of court bookings across all users.
    - When one user books or cancels a court, all users see the updated availability in real time.

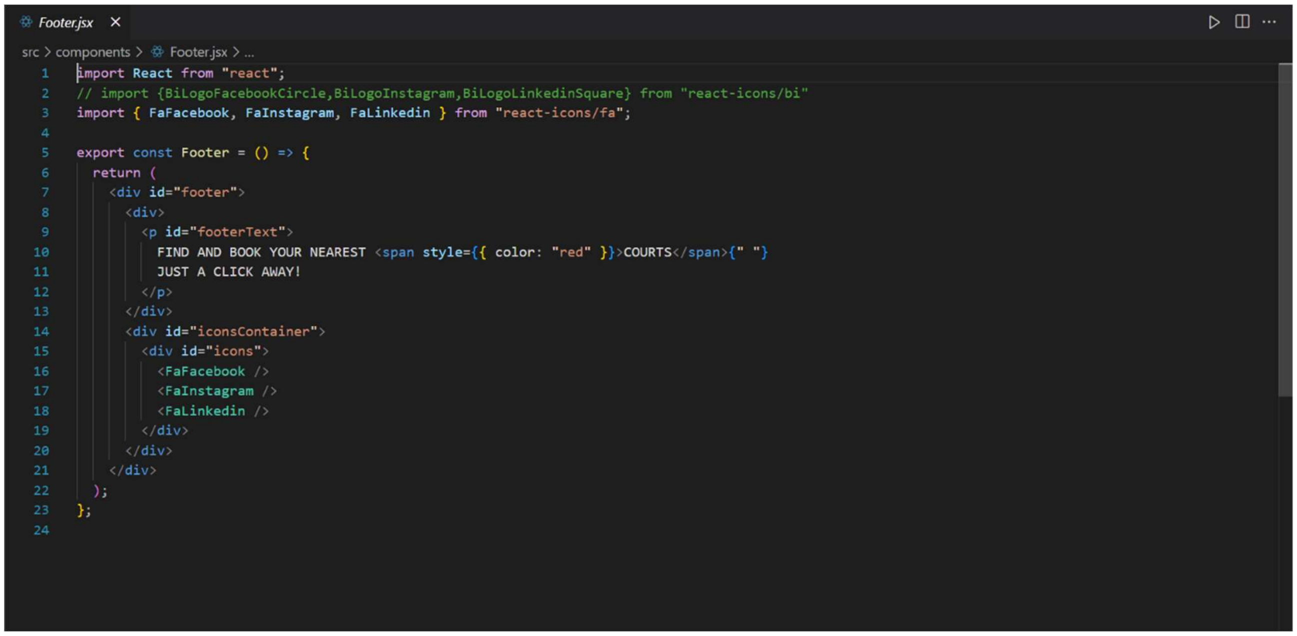
3.4 Code and Implementation

Components



```
import React from "react";
import {
  Skeleton,
  Stack,
} from "@chakra-ui/react";
export const BookingSkeleton = () => {
  return (
    <Stack w={"50%"} margin="auto" marginTop={"20px"}>
      <Skeleton height="30px" w="50%" />
      <Skeleton height="50px" w="50%"/>
      <Skeleton height="300px" w={"50%"} />
      <Skeleton height="50px" w="50%"/>
    </Stack>
  );
};
```

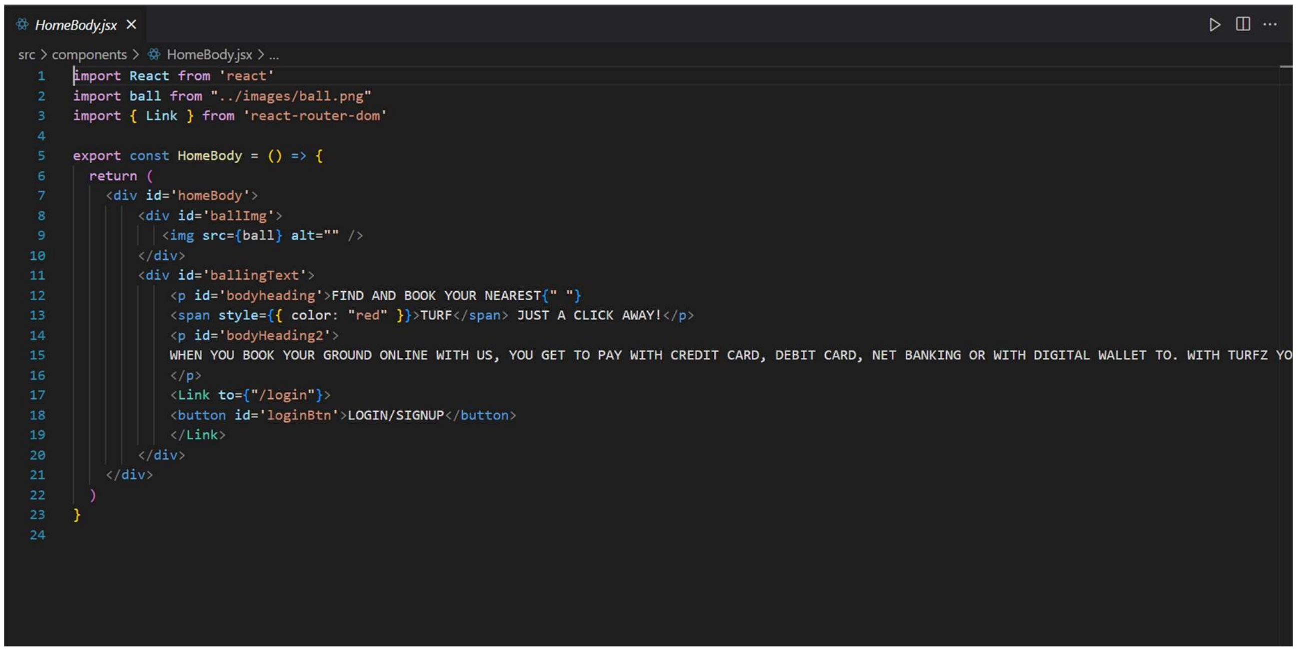
Figure 3.3 Booking Code



```
import React from "react";
// import {BiLogoFacebookCircle,BiLogoInstagram,BiLogoLinkedinSquare} from "react-icons/bi"
import { FaFacebook, FaInstagram, FaLinkedin } from "react-icons/fa";

export const Footer = () => {
  return (
    <div id="footer">
      <div>
        <p id="footerText">
          FIND AND BOOK YOUR NEAREST <span style={{ color: "red" }}>COURTS</span>{" "}
          JUST A CLICK AWAY!
        </p>
      </div>
      <div id="iconsContainer">
        <div id="icons">
          <FaFacebook />
          <FaInstagram />
          <FaLinkedin />
        </div>
      </div>
    </div>
  );
};
```

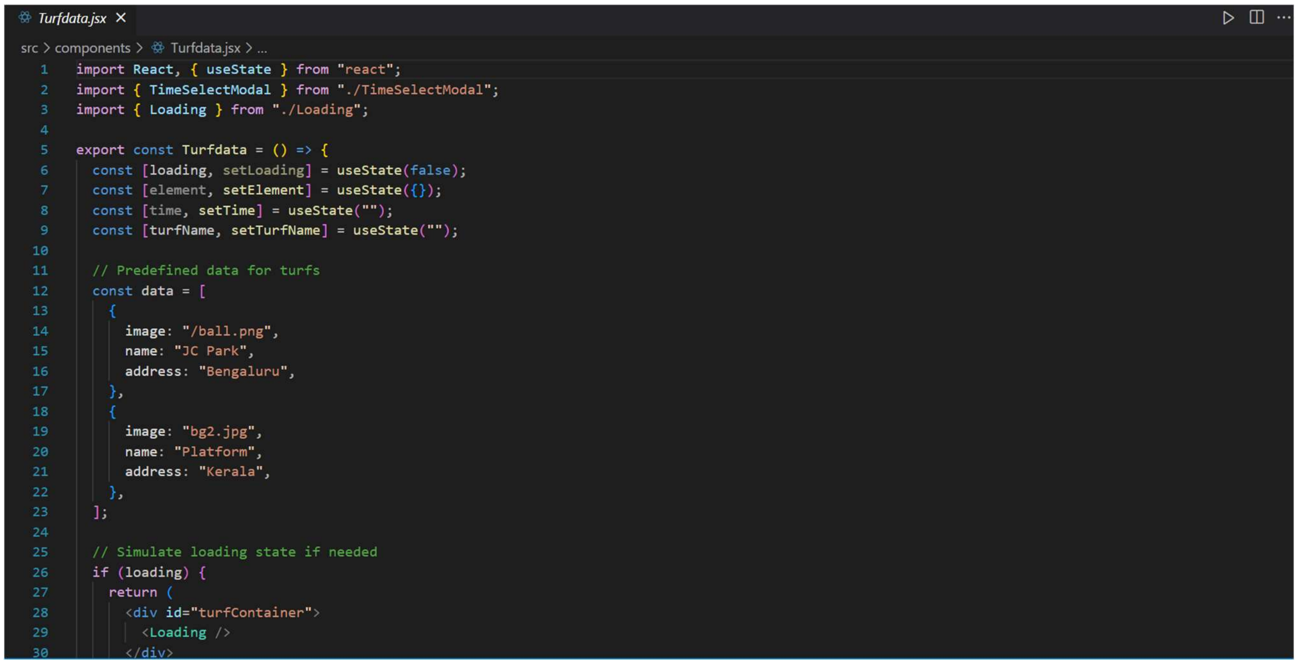
Figure 3.4-Footer Code



```jsx
import React from 'react'
import ball from "../images/ball.png"
import { Link } from 'react-router-dom'

export const HomeBody = () => {
  return (
    <div id='homeBody'>
        <div id='ballImg'>
          <img src={ball} alt="" />
        </div>
        <div id='ballingText'>
            <p id='bodyheading'>FIND AND BOOK YOUR NEAREST{" "}
            <span style={{ color: "red" }}>TURF</span> JUST A CLICK AWAY!</p>
            <p id='bodyHeading2'>
            WHEN YOU BOOK YOUR GROUND ONLINE WITH US, YOU GET TO PAY WITH CREDIT CARD, DEBIT CARD, NET BANKING OR WITH DIGITAL WALLET TO. WITH TURFZ YO
            </p>
            <Link to={"/login"}>
            <button id='loginBtn'>LOGIN/SIGNUP</button>
            </Link>
        </div>
    </div>
  )
}
```

Figure 3.5 HomeBody Code



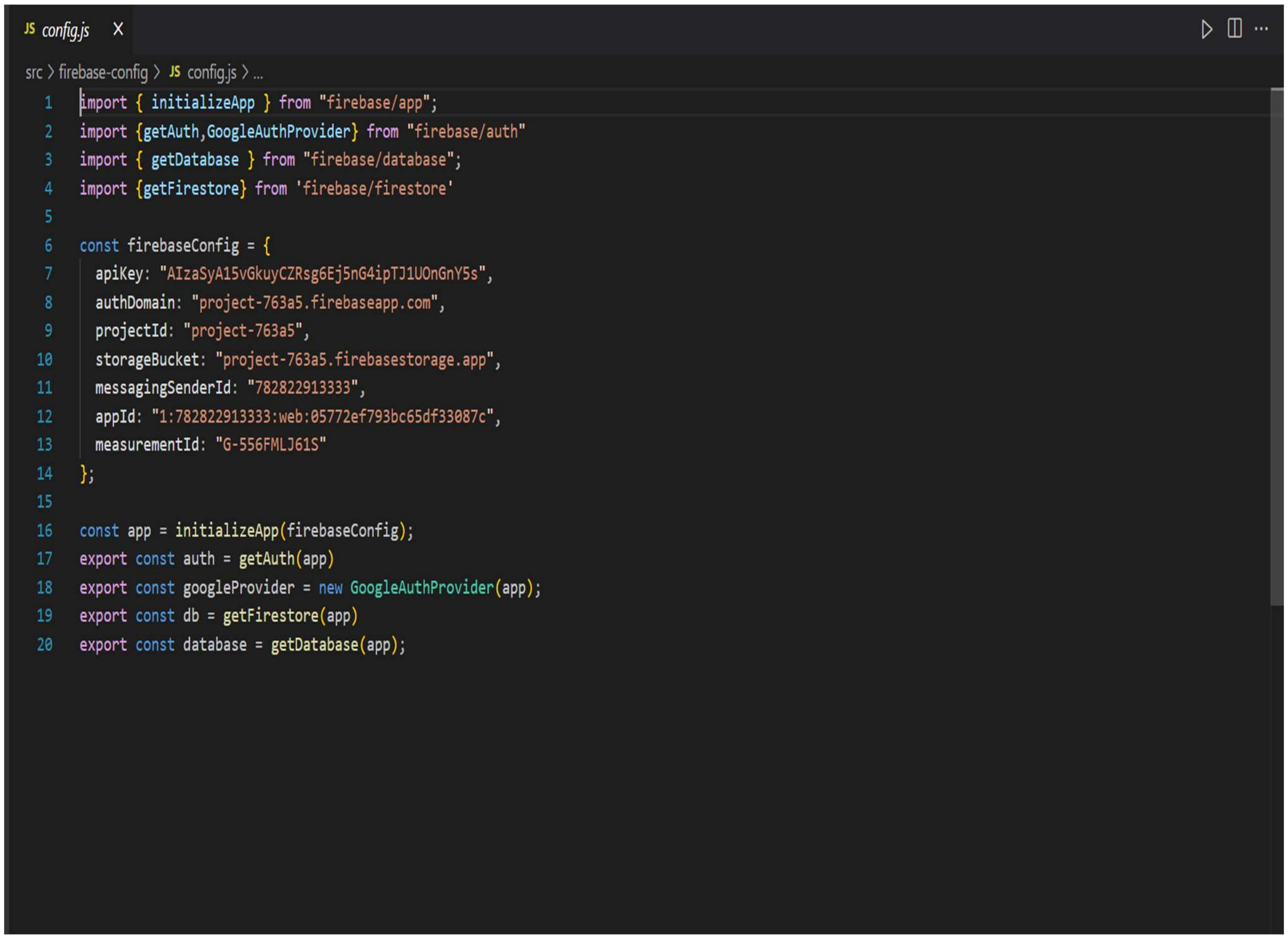
```jsx
import React, { useState } from "react";
import { TimeSelectModal } from "./TimeSelectModal";
import { Loading } from "./Loading";

export const Turfdata = () => {
  const [loading, setLoading] = useState(false);
  const [element, setElement] = useState({});
  const [time, setTime] = useState("");
  const [turfName, setTurfName] = useState("");

  // Predefined data for turfs
  const data = [
    {
      image: "/ball.png",
      name: "JC Park",
      address: "Bengaluru",
    },
    {
      image: "bg2.jpg",
      name: "Platform",
      address: "Kerala",
    },
  ];

  // Simulate loading state if needed
  if (loading) {
    return (
      <div id="turfContainer">
        <Loading />
      </div>
```

Figure 3.6 Court Data Code

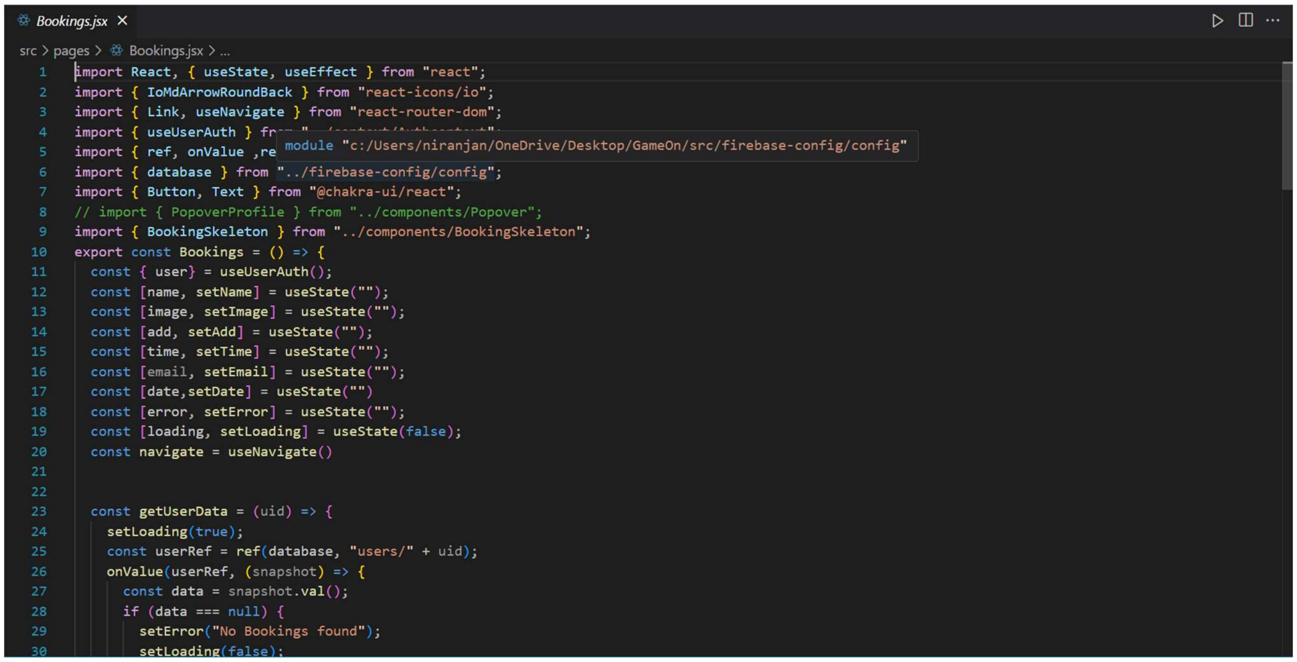


```
import { initializeApp } from "firebase/app";
import {getAuth,GoogleAuthProvider} from "firebase/auth"
import { getDatabase } from "firebase/database";
import {getFirestore} from 'firebase/firestore'

const firebaseConfig = {
  apiKey: "AIzaSyA15vGkuyCZRsg6Ej5nG4ipTJ1UOnGnY5s",
  authDomain: "project-763a5.firebaseapp.com",
  projectId: "project-763a5",
  storageBucket: "project-763a5.firebasestorage.app",
  messagingSenderId: "782822913333",
  appId: "1:782822913333:web:05772ef793bc65df33087c",
  measurementId: "G-556FMLJ61S"
};

const app = initializeApp(firebaseConfig);
export const auth = getAuth(app)
export const googleProvider = new GoogleAuthProvider(app);
export const db = getFirestore(app)
export const database = getDatabase(app);
```

Figure 3.7 ConFig Code

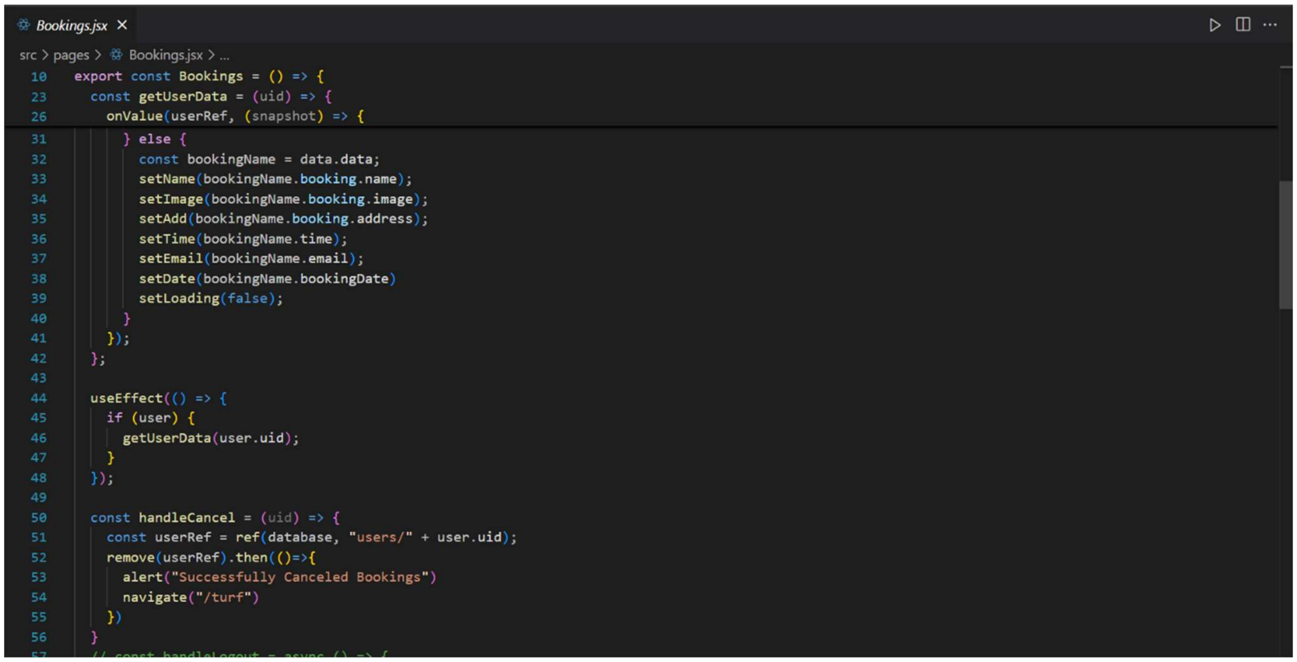


Bookings.jsx

src > pages > Bookings.jsx > ...

```
import React, { useState, useEffect } from "react";
import { IoMdArrowRoundBack } from "react-icons/io";
import { Link, useNavigate } from "react-router-dom";
import { useUserAuth } from "../context/AuthContext";
import { ref, onValue ,re module "c:/Users/niranjan/OneDrive/Desktop/GameOn/src/firebase-config/config"
import { database } from "../firebase-config/config";
import { Button, Text } from "@chakra-ui/react";
// import { PopoverProfile } from "../components/Popover";
import { BookingSkeleton } from "../components/BookingSkeleton";
export const Bookings = () => {
  const { user} = useUserAuth();
  const [name, setName] = useState("");
  const [image, setImage] = useState("");
  const [add, setAdd] = useState("");
  const [time, setTime] = useState("");
  const [email, setEmail] = useState("");
  const [date,setDate] = useState("")
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate()


  const getUserData = (uid) => {
    setLoading(true);
    const userRef = ref(database, "users/" + uid);
    onValue(userRef, (snapshot) => {
      const data = snapshot.val();
      if (data === null) {
        setError("No Bookings found");
        setLoading(false);
```



Bookings.jsx

src > pages > Bookings.jsx > ...

```
export const Bookings = () => {
  const getUserData = (uid) => {
    onValue(userRef, (snapshot) => {
      } else {
        const bookingName = data.data;
        setName(bookingName.booking.name);
        setImage(bookingName.booking.image);
        setAdd(bookingName.booking.address);
        setTime(bookingName.time);
        setEmail(bookingName.email);
        setDate(bookingName.bookingDate)
        setLoading(false);
      }
    });
  };

  useEffect(() => {
    if (user) {
      getUserData(user.uid);
    }
  });

  const handleCancel = (uid) => {
    const userRef = ref(database, "users/" + user.uid);
    remove(userRef).then(()=>{
      alert("Successfully Canceled Bookings")
      navigate("/turf")
    })
  }
  // const handleLogout = async () => {
```

Figure 3.8 Booking Code

Chapter 4

Results and Discussion

4.1 Summary of the Obtained Result

The **Sports Arena Booking System** project was developed with the primary aim of simplifying and streamlining the process of booking badminton courts. After the development and testing phases, the system was evaluated based on several key metrics: functionality, user experience, and performance.

Functionality:

- The system functions as expected, allowing users to log in, view available courts, select preferred time slots, and book courts seamlessly.
- Availability checks are performed in real-time, ensuring that users are informed immediately whether the selected court is available or not.
- Bookings are stored efficiently in Firebase, ensuring that data is securely saved and can be accessed later by the user.

User Experience:

- Users found the interface easy to use and intuitive. The simplicity of the design allows even first-time users to navigate through the booking process without confusion.
- The real-time availability feature eliminates the frustration of double bookings, enhancing user satisfaction.
- The system also provides users the ability to view and cancel bookings, giving them control over their reservations.

Performance:

- The system demonstrated good performance with quick loading times for the user interface.
- Real-time updates were successfully reflected in the system, ensuring users see up-to-date availability when making bookings.
- Firebase ensured smooth backend operations, and no performance issues were encountered during testing, even when multiple users accessed the system simultaneously.

Issues:

- Although the system works effectively, one challenge faced during testing was ensuring smooth integration between the frontend (React) and Firebase, particularly in handling real-time data updates. Some minor latency issues were observed when large numbers of users were interacting with the system at the same time.

4.2 Output snapshots

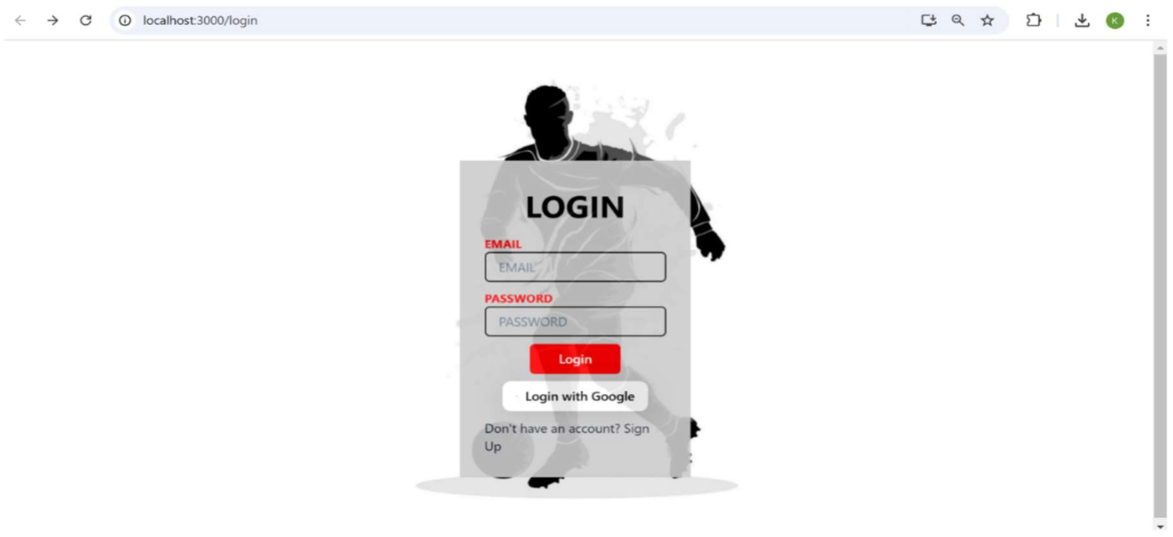


Figure 4.1 Login Page

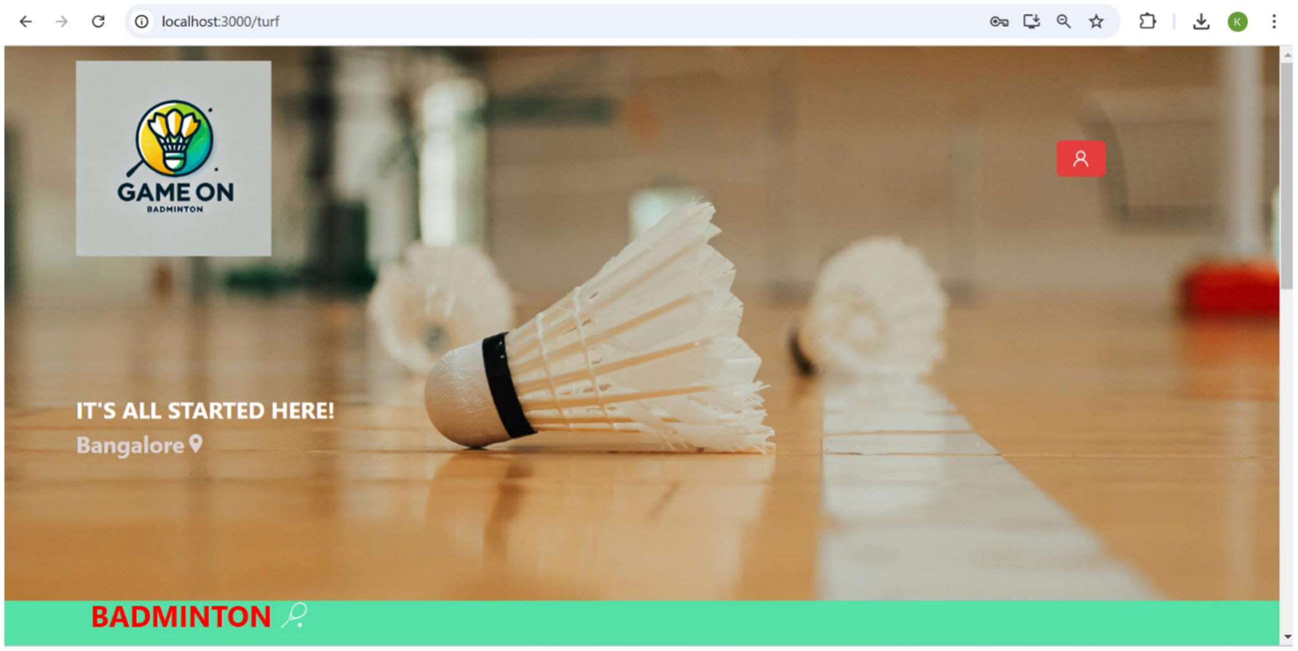


Figure 4.2 Home Page

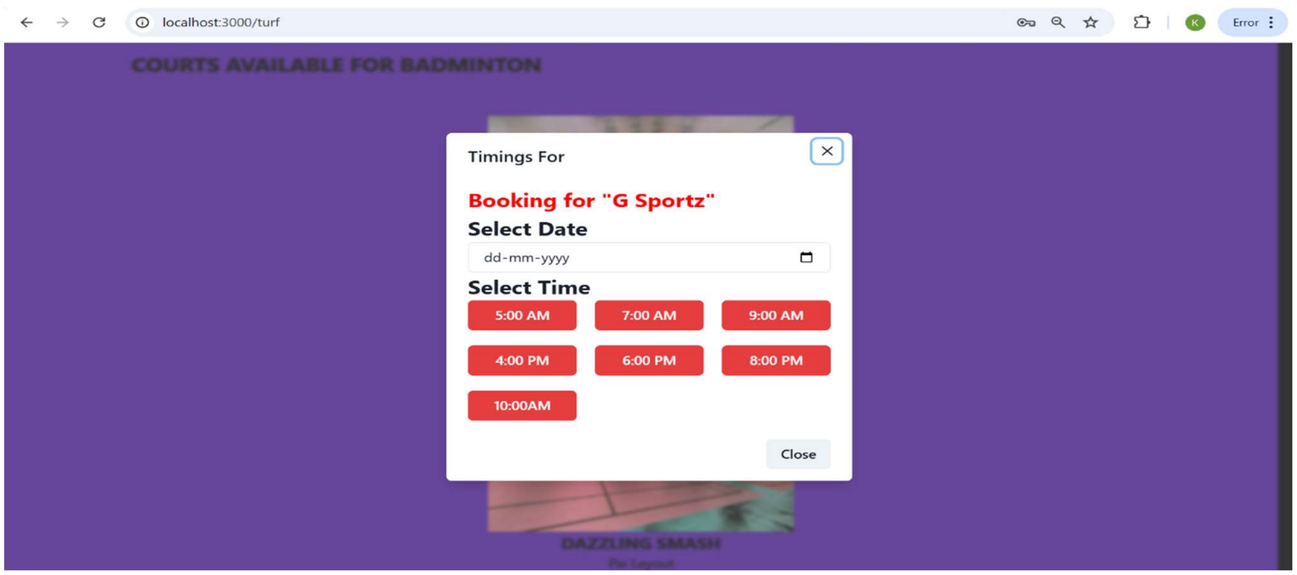
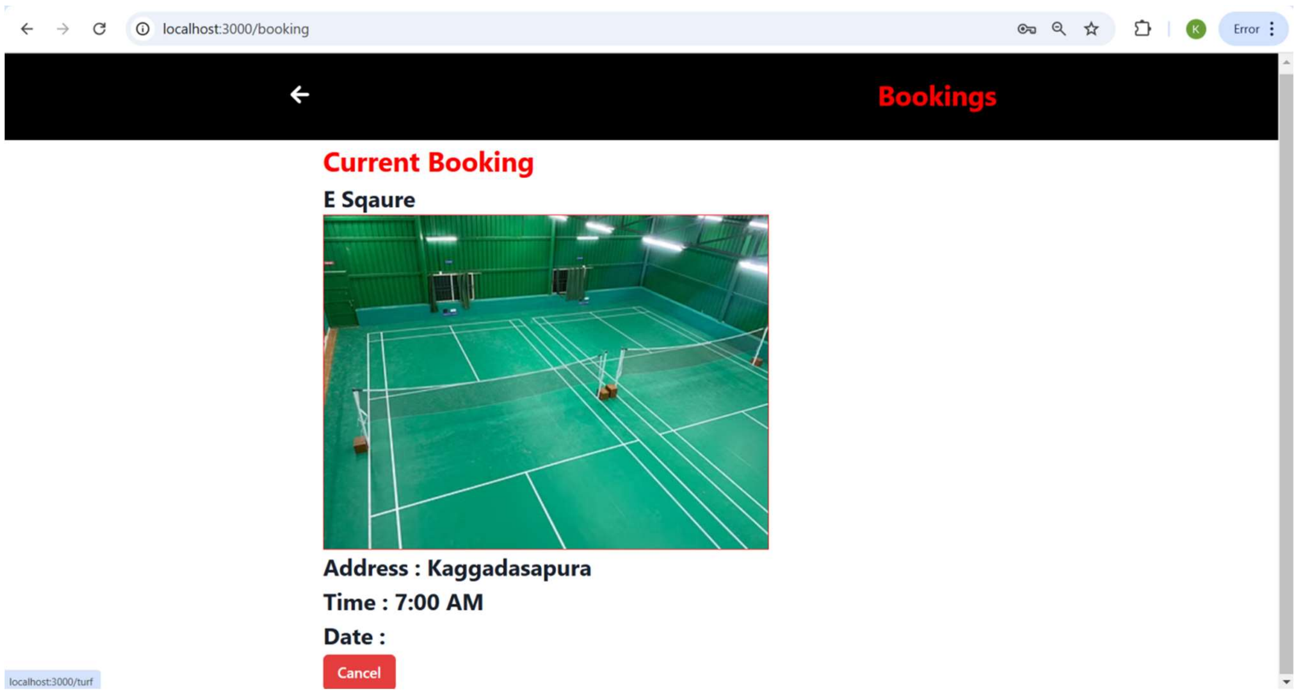


Figure 4.3 Booking Page



Current Booking

E Sqaure

Address : Kaggadasapura

Time : 7:00 AM

Date :

Figure 4.4 Current Booking Page

Conclusion

The **Sports Arena Booking System** project successfully addresses the challenges faced by users in booking badminton courts by providing an efficient, user-friendly, and reliable platform. The system's design emphasizes simplicity and ease of use, ensuring that users can quickly and conveniently check court availability, make bookings, and manage their reservations with minimal effort. By integrating real-time updates and utilizing Firebase for backend management, the system ensures smooth functionality and effective data storage.

The project not only simplifies the booking process but also eliminates issues like double bookings and scheduling conflicts, providing a more seamless experience for users. The system's ability to handle user data securely and store bookings in real-time enhances its reliability and effectiveness.

In conclusion, the **Sports Arena Booking System** meets its intended objectives and serves as a practical solution for managing court bookings. While there are opportunities for further improvement, such as integrating payment systems and mobile support, the current system successfully addresses its core functionality and offers significant benefits to both users and sports facility managers. The project provides a solid foundation for future enhancements and scalability, paving the way for a more comprehensive sports facility management system in the future.

References

1. Author: Tonto Claudinus, Made Prayoga Wicaksana, Nicolas Kornelius Sitorus
Source: IEEE
Paper Name: SPORT FIELD RESERVATION BASED ON MOBILE APPLICATION

2. Authors: Anusha K, Rashmi P, Kiran Kumar M N
Source: IJARSCT
Paper Name: Badminton Court Management System

3. https://ideausher.com/blog/court-booking-app-development-cost-and-features/
4. Using Firebase for Real-Time Database and Authentication in a Web AppReference: https://www.freecodecamp.org/news/how-to-build-a-chat-app-with-react-and-firebase/