KIT-KALAIGNARKARUNANIDHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

COIMBATORE, TAMILNADU-641402

Affiliated to Anna University, Chennai

Accredited by NAAC with 'A' Grade | Accredited by NBA (CSE, ECE, EEE & MECH)

MINI PROJECT REPORT

TURF BOOKING WEBSITE

Submitted by

711522BCB031 LOKHITH ASWA A

Under the Guidance of

Dr.K. Mahalakshmi B.E.,MBA.,M.Tech.,Ph.D

Professor & HOD/CSBS

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEMS

JULY 2024

KIT-KALAIGNARKARUNANIDHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

COIMBATORE, TAMILNADU-641402

Affiliated to Anna University, Chennai

Accredited by NAAC with 'A' Grade | Accredited by NBA(CSE, ECE, EEE & MECH)

BONAFIDE CERTIFICATE

This is to certify that this Mini Project Report on **TURF BOOKING WEBSITE** is the bonafide work done by **LOKHITH ASWA A (711522BCB031)** of B.Tech in Computer Science and Business systems during the academic year 2024-25.

Project Guide

Dr. K. Mahalakshmi B.E,MBA.,M.Tech.,Ph.D

Professor & HOD
Dean Placement
Department of Computer Science and Business System KIT – Kalaignarkarunanidhi Institute of Technology
Coimbatore

Head of the Department

Dr. K. Mahalakshmi B.E,MBA.,M.Tech.,Ph.D

Professor & HOD
Dean Placement
Department of Computer Science and Business System KIT – Kalaignarkarunanidhi Institute of Technology
Coimbatore

Submitted for mini project viva-voce examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

At this delightful moment of having accomplished my project I extend my sincere thanks to **Thiru. PONGALUR N. PALANISAMY**, Founder and Chairman, KIT- Kalaignarkarunanidhi Institute of Technology, for providing amenities and lab facilities for completing the project work.

We would extend our sincere thanks to **Mrs. INDU MURUGESAN,** Vice- Chairperson,KIT-Kalaignarkarunanidhi Institute of Technology, for empowering us with excellent infrastructure and able administration.

Our heartfelt gratitude to **Dr. N. MOHAN DAS GANDHI**, the honourable CEO and **Dr. M.RAMESH**, the Principal for the special motivation and for providing us an opportunity and essential facilities in carrying out this project.

We also express our sincere thanks to the HOD & Dean Placements, **Dr. K. Mahalakshmi**, Department of Computer Science and Business Systems for her support and encouragement extended towards this project.

The feeling of gratitude and fulfilling our duty would not have been complete without our Project Guide, **Dr. K. Mahalakshmi,** Professor, Department of Computer Science and Business Systems, special appreciation for the timely help and guidance received.

We would also wish to express our gratitude to our mentors for the support rendered throughout the project, and we thank all my faculty members for their valuable support.

Last but never the least, we place our feeling of gratitude to our parents without whom, making it this far wouldn't have been possible.

TABLE OF CONTENTS

ABSTRACT

This project presents the development of a comprehensive turf booking website designed to streamline the process of reserving sports facilities, specifically cricket turfs, football turfs, and badminton courts. The platform caters to both users seeking to book these facilities and turf owners interested in listing their venues. The primary objective is to provide an intuitive, user-friendly interface that simplifies the booking process, enhances user experience, and optimizes facility management for owners.

The website offers a dynamic slot selection feature, allowing users to view and book available time slots for their chosen sport. Additionally, it supports real-time updates on slot availability, ensuring accurate and timely information. Turf owners can easily list their facilities, providing essential details and availability, thus reaching a broader audience.

Key functionalities include user registration, slot booking and cancellation, and an administrative interface for turf owners to manage listings. The project employs HTML, CSS, and JavaScript for the front-end design, ensuring a responsive and engaging user experience. Future enhancements may include the integration of payment gateways and advanced scheduling algorithms to further refine the booking process.

This project aims to bridge the gap between sports enthusiasts and facility providers, promoting active lifestyles and better utilization of sports venues.

CHAPTER 1

INTRODUCTION

The Turf Booking Website is a digital platform designed to streamline the reservation process for sports facilities, specifically cricket turfs, football turfs, and badminton courts. This project aims to provide an intuitive and user-friendly interface that simplifies the booking experience for sports enthusiasts and offers a comprehensive management tool for turf owners.

The development of the Turf Booking Website incorporates a range of modern web technologies and methodologies to ensure a robust and efficient platform:

- **Frontend Development:** The user interface is built using HTML, CSS, and JavaScript. These technologies ensure a responsive and engaging design that adapts seamlessly to different devices and screen sizes.
- **Backend Development:** [Specify backend technology if used, e.g., Node.js, Django, etc.] is employed to handle server-side logic, user authentication, and data processing.
- **Database Management:** [Specify database if used, e.g., MySQL, MongoDB, etc.] is utilized for storing user data, booking information, and turf details, ensuring reliable and efficient data management.
- **Real-Time Updates:** The platform integrates real-time data updates to provide users with accurate information on slot availability, reducing the risk of double bookings and enhancing user experience.
- **Responsive Design:** The website's design principles ensure a consistent and user-friendly experience across various devices, from desktops to mobile phones.

By leveraging these technologies and methods, the Turf Booking Website aims to create a seamless and efficient booking experience for users while providing turf owners with powerful tools to manage their facilities.

PROJECT OBJECTIVE

1. **User-Friendly Booking Process:** Develop an intuitive and easy-to-use interface that simplifies the process of booking sports facilities, enhancing the overall user experience.
2. **Real-Time Availability:** Implement real-time updates on slot availability to ensure users have accurate and up-to-date information, reducing the risk of double bookings.
3. **Turf Owner Management:** Provide an efficient platform for turf owners to list and manage their facilities, allowing them to update availability, handle bookings, and reach a broader audience.
4. **Promote Active Lifestyles:** Encourage sports participation by making it easier for users to access and book sports venues, thereby promoting a healthy and active lifestyle.
5. **Scalable Infrastructure:** Build a scalable system that can handle a growing number of users and facilities, ensuring the platform remains reliable and efficient as it expands.
6. **Secure Transactions:** Plan for future integration of secure payment gateways to facilitate safe online transactions for bookings.
7. **Enhanced Features:** Incorporate advanced features such as user feedback, rating systems, and detailed analytics for turf owners to optimize their facility management and improve user satisfaction.

CHAPTER 2

SYSTEM STUDY

2.1 PROPOSED SYSTEM

1. Scope of the Project

The Turf Booking Website aims to provide a centralized platform for users to easily book sports facilities such as cricket turfs, football turfs, and badminton courts. The system will cater to both sports enthusiasts seeking to make bookings and turf owners interested in listing their facilities.

2. Functional Requirements

- **User Registration and Authentication:** Users can create accounts and log in securely to access booking functionalities.
- **Facility Listing:** Turf owners can register their facilities on the platform, providing details such as location, facilities available, and pricing.
- **Slot Booking:** Users can view available time slots for different sports, select desired slots, and make bookings.
- **Real-Time Updates:** The system will provide real-time updates on slot availability to prevent double bookings.
- **Payment Integration:** Future integration of payment gateways to facilitate secure online transactions for bookings.
- **Administrative Tools:** Turf owners have access to administrative tools to manage listings, update availability, and view booking history.
- **User Feedback and Rating:** Provision for users to leave feedback and ratings for facilities, enhancing transparency and user trust.

3. Non-Functional Requirements

- **Performance:** The system should handle concurrent user sessions and maintain responsiveness during peak booking times.

- **Security:** Implement robust security measures to protect user data and ensure secure transactions.
- **Scalability:** Design the system to scale easily as the user base and number of listed facilities grow.
- **Usability:** Ensure a user-friendly interface with intuitive navigation and responsive design across devices.
- **Reliability:** Minimize downtime and ensure high availability of the booking system.

4. Future Enhancements

- Integration of advanced scheduling algorithms to optimize slot availability and minimize conflicts.
- Enhanced analytics tools for turf owners to analyze booking trends and optimize facility management.
- Expansion to include additional sports and facilities based on user demand.

2.2 SOFTWARE OVERVIEW

2.2.1 FRONT END

- **HTML**

HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content.

Hypertext: Text (often with embeds such as images) that is organized in order to connect related items

Markup: a style guide for typesetting anything to be printed in hardcopy or soft copy format

Language: a language that a computer system understands and uses to interpret commands.

- **CSS**

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

- **JAVASCRIPT**

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more. It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.

2.2.2 BACK END

The back- end of a website consists of a server, an application, and a database. A back-end developer builds and maintains the technology that powers those components which, together, enable the user-facing side of the website to even exist in the first place.

The back-end, also called the server-side, consists of the server which provides data on request, the application that channels it, and the database which organizes the information. For example, when a customer browses shoes on a website, they are interacting with the front end

- **NODE JS**

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It is designed to build scalable network applications.

Handles server-side logic, processes user requests, manages sessions, and interfaces with the database.

- **Databases**

1. MongoDB

MongoDB is a NoSQL, document-oriented database that stores data in flexible, JSON-like documents.

Used for storing user data, booking information, facility details, and other application-specific data. Its flexibility allows for easy scaling and handling of large volumes of data.

2. Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js.

Provides a schema-based solution to model application data, including data validation, casting, and business logic hooks.

- **JWT (JSON Web Tokens)**

JWT is an open standard for securely transmitting information between parties as a JSON object.

Used for user authentication and authorization, ensuring secure access to protected routes and resources.

2.3 HARDWARE REQUIREMENTS

1. Development Environment

For local development and testing, you will need:

- **Computer/Workstation**
  - **Processor:** Intel Core i5 or equivalent
  - **RAM:** 8 GB (16 GB recommended for smoother multitasking)
  - **Storage:** 256 GB SSD (512 GB SSD recommended for faster performance)
  - **Operating System:** Windows 10/11, macOS, or Linux
  - **Display:** 1080p resolution monitor (dual monitors recommended for increased productivity)
  - **Internet Connection:** Stable broadband connection for downloading dependencies, libraries, and remote collaboration

2. Server Environment

For hosting the backend and database, you will need:

- **Server**
  - **Processor:** Intel Xeon or equivalent (quad-core or higher)
  - **RAM:** 16 GB (32 GB recommended for handling higher traffic)
  - **Storage:** 500 GB SSD (1 TB SSD recommended for better performance and storage capacity)
  - **Operating System:** Linux-based OS (Ubuntu, CentOS, or similar) for better performance and security
  - **Network:** High-speed internet connection with a static IP address
  - **Backup:** External storage for regular backups (cloud storage or external hard drives)

2.4 SOFTWARE REQUIREMENTS

- **Development Environment(OS):**

  Windows 10/11, macOS, or Linux

- **Integrated Development Environment (IDE):**

  Visual Studio Code (recommended)

- **Browser:**

  Modern web browser like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge for testing and debugging our website.

- **Database Management:**

  MongoDB Compass: GUI tool for managing MongoDB databases

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM IMPLEMENTATION

In this website 'TURF', there are several webpages combined together and connected to each other.

- **Home Page**

The Home Page serves as the initial landing point for users visiting the Turf Booking Website. It features a welcoming interface that introduces the platform's key offerings and benefits. The page typically includes a visually appealing layout with navigation links to other sections such as sports categories (cricket, football, badminton), user registration or login options, and highlights of featured turfs or promotions. It aims to engage visitors by showcasing the ease of booking, highlighting facility diversity, and promoting active sports participation. Additionally, it may include testimonials, social proof elements, or a brief overview of the booking process to encourage user exploration.

- **Registration/Login Page**

The Registration/Login Page provides essential functionalities for both new users and returning visitors. New users can register by entering basic details such as name, email, password, and contact information. The registration process may involve email verification for account activation. Returning users can securely log in using their credentials, enabling access to personalized features like saved preferences, booking history, and account settings. The page incorporates robust security measures such as bcrypt for password hashing and JWT for session management to safeguard user data. Clear error handling and password recovery options enhance usability, ensuring a seamless login experience.

- **Facility Listings Page**

The Facility Listings Page displays comprehensive listings of available turfs categorized by sport type (cricket, football, badminton) and location. Each listing includes detailed information such as turf amenities, pricing, availability, and user ratings. Users can browse through turfs based on their preferences, view high-resolution images, and read reviews from previous bookings to make informed decisions. Advanced filtering options (e.g., location-based search, date/time availability) and sorting functionalities optimize user navigation, ensuring a user-friendly experience. Turf owners manage their listings through an admin interface, updating details and availability in real-time to reflect current bookings and operational status accurately.

- **Booking Page**

The Booking Page enables users to select desired turfs and booking slots based on availability. It features an interactive calendar or time slot picker that dynamically updates in real-time to display available dates/times for selected turfs. Users can customize their bookings by specifying the number of participants, additional requirements (e.g., equipment rental), and preferred payment method. The page integrates secure payment gateways to facilitate seamless transactions, ensuring user confidence and operational efficiency. Confirmation emails or notifications are sent upon successful booking, providing users with booking details and contact information for further inquiries or modifications.

- **User Profile Page**

The User Profile Page offers personalized management tools for registered users. It displays user-specific information such as booking history, upcoming reservations, and account settings. Users can update personal details, manage notification preferences, and review past bookings. The page may also feature a dashboard with graphical summaries of booking statistics, loyalty rewards, or promotional offers tailored to user preferences. Seamless navigation between profile sections enhances usability, allowing users to efficiently manage their activities and maintain engagement with the platform.

- **Admin Dashboard**

The Admin Dashboard serves as a centralized control panel for turf owners and platform administrators. It provides comprehensive insights and management capabilities to oversee turfs, bookings, and user interactions. Key features include real-time monitoring of booking activities, analytics on turf utilization and revenue generation, and tools for managing facility listings and availability. Admins can approve new turf listings, handle customer inquiries or disputes, and generate reports on business performance. The dashboard supports administrative tasks such as adjusting pricing, setting promotional offers, and implementing system updates to enhance operational efficiency and user satisfaction.

- **Contact Us Page**

The Contact Us Page offers a direct communication channel for users and turf owners to reach customer support or inquire about specific issues. It includes a contact form with fields for name, email, subject, and message, ensuring users can easily submit queries or feedback. Contact information such as phone numbers, email addresses, and physical addresses may also be provided for alternative communication methods. The page aims to foster customer trust and satisfaction by offering responsive support and addressing user concerns promptly. Clear call-to-action buttons encourage users to initiate contact, facilitating seamless interaction and enhancing overall service accessibility.

3.2 DATAFLOW FOR TURF BOOKING WEBSITE

Level 0 DFD:

- **Entities:**
    - **User:** Interacts with the website to search for turfs, make bookings, and manage profile.
    - **Turf Owner:** Manages turf listings, updates availability, and interacts with booking requests.
    - **Administrator:** Manages the overall system, monitors activities, and handles administrative tasks.
- **Processes:**
    - **User Registration/Login:** Handles user authentication and manages user sessions.
    - **Browse Turf Listings:** Allows users to search and view available turfs based on location and sport type.
    - **Make Booking:** Enables users to select turfs, choose booking slots, and confirm bookings with payment processing.
    - **Manage Turf Listings:** Allows turf owners to add, update, and remove turf listings, manage availability, and respond to booking requests.
    - **Admin Dashboard:** Provides administrators with tools to monitor system activities, manage user accounts, handle disputes, and generate reports.
    - **Contact Support:** Facilitates user inquiries and support requests through direct communication channels.
- **Data Stores:**
    - **MongoDB Database:** Stores user profiles, booking details, turf listings, and administrative data.
    - **External Payment Gateway:** Handles secure transactions for booking payments.
    - **Email Service:** Sends notifications and alerts related to booking confirmations, updates, and user interactions.
- **Data Flow:**

- **User Data Flow:** Starts with user registration or login, proceeds to browsing turfs, making bookings, and managing profile settings.
- **Turf Owner Data Flow:** Includes managing turf listings, updating availability, and responding to booking requests.
- **Administrator Data Flow:** Involves monitoring system activities, managing user accounts, and handling support requests.

Level 1 DFD (Detailed Processes):

- **User Registration/Login Process:**
  - User submits registration details.
  - System verifies and stores user information in the database.
  - User logs in securely using stored credentials.
- **Browse Turf Listings Process:**
  - User selects sport type and location preferences.
  - System retrieves matching turf listings from the database.
  - User views detailed turf information and reviews.
- **Make Booking Process:**
  - User selects turf, chooses booking date/time, and adds additional requirements.
  - System checks availability and confirms booking.
  - Payment gateway processes transaction securely.
- **Manage Turf Listings Process:**
  - Turf owner adds new turf listing or updates existing details.
  - System validates and stores changes in the database.
  - Turf owner manages availability and responds to booking requests.
- **Contact Support Process:**
  - User submits inquiry or support request via contact form.
  - System forwards request to support team or administrator.
  - Support team responds to user inquiries via email or direct communication.

3.3 PROJECT MODULES:

1. **Authentication and User Management**

- **Description:** Handles user registration, login, and authentication processes. Manages user profiles, preferences, and session handling.
- **Key Features:** Registration, login/logout, profile management, password reset, session management.

2. **Turf Listings Management**

- **Description:** Allows turf owners to add, edit, and manage turf listings. Includes details such as sports type, facilities, location, pricing, and availability.
- **Key Features:** Add/edit/delete listings, manage availability, update facilities and pricing, receive booking notifications.

3. **Booking Management**

- **Description:** Facilitates the booking process for users, allowing them to select turfs, choose booking slots, and confirm reservations.
- **Key Features:** Calendar/time slot selection, booking confirmation, payment integration (optional), booking history, cancellation management.

4. **Search and Filtering**

- **Description:** Enables users to search for turfs based on sport type, location, availability, and facilities. Provides advanced filtering options for enhanced user experience.
- **Key Features:** Search by sport type/location, advanced filters (e.g., price range, amenities), sorting options.

5. **User Dashboard**

- **Description:** Provides users with a personalized dashboard to manage bookings, view past reservations, update profiles, and set preferences.

- **Key Features:** View/edit profile, manage bookings, view booking history, set notification preferences.

6. **Admin Panel**

- **Description:** Centralized dashboard for administrators to oversee system activities, manage user accounts, handle support requests, and monitor analytics.
- **Key Features:** User management (approve/delete users), view system logs, generate reports (booking statistics, revenue), manage support tickets.

7. **Communication and Notifications**

- **Description:** Facilitates communication between users, turf owners, and administrators. Sends notifications for booking confirmations, updates, and support inquiries.
- **Key Features:** Contact form, email notifications (booking confirmation, support responses), real-time notifications (optional).

8. **Payment Integration**

- **Description:** Integrates secure payment gateways to process booking payments securely and efficiently.
- **Key Features:** Payment gateway integration (e.g., Stripe, PayPal), handling transactions, generating invoices (optional).

9. **Security and Compliance**

- **Description:** Implements security measures to protect user data, ensure secure transactions, and comply with data protection regulations.
- **Key Features:** HTTPS/SSL encryption, password hashing (bcrypt), GDPR compliance (if applicable), secure session management.

10. **Reporting and Analytics**

- **Description:** Provides insights into system performance, user behavior, booking trends, and revenue generation.
- **Key Features:** Generate reports (booking statistics, revenue), analytics dashboard, data visualization tools (optional).

11. **Support and Help**

- **Description:** Offers user support channels and resources to assist with inquiries, troubleshooting, and feedback.
- **Key Features:** FAQ section, contact support form, support ticket management, user feedback collection.

CHAPTER 4

DATABASE DESIGN

1. User Collection

- **Fields:**
  - _id: ObjectId (unique identifier)
  - username: String
  - email: String (unique)
  - password: String (hashed)
  - phone: String
  - createdAt: Date
  - updatedAt: Date
  - role: String (user, admin, turf owner)

2. Turf Collection

- **Fields:**
  - _id: ObjectId (unique identifier)
  - ownerId: ObjectId (reference to User who owns the turf)
  - name: String
  - sportType: String (cricket, football, badminton, etc.)
  - location: String
  - description: String
  - facilities: [String] (array of amenities like lights, changing rooms)
  - pricePerHour: Number
  - imageUrls: [String] (array of URLs for turf images)
  - createdAt: Date
  - updatedAt: Date

3. Booking Collection

- **Fields:**
    - _id: ObjectId (unique identifier)
    - userId: ObjectId (reference to User who made the booking)
    - turfId: ObjectId (reference to Turf booked)
    - bookingDate: Date
    - startTime: Date
    - endTime: Date
    - totalAmount: Number
    - status: String (pending, confirmed, canceled)
    - createdAt: Date
    - updatedAt: Date

Explanation:

- **User Collection:** Stores user details including username, email, hashed password, phone number, role (user, admin, turf owner), and timestamps for creation and updates.
- **Turf Collection:** Holds information about each turf listing such as its owner (reference to User), name, sport type, location, description, facilities available (stored as an array of strings), price per hour, image URLs, and timestamps for creation and updates.
- **Booking Collection:** Tracks bookings made by users, referencing the user who booked (userId), the turf booked (turfId), booking date, start and end times, total booking amount, booking status (pending, confirmed, canceled), and timestamps for creation and updates.

Considerations:

- **Data Relationships:** Use ObjectId references to establish relationships between collections (e.g., ownerId in Turf refers to _id in User).
- **Indexes:** Consider adding indexes on fields frequently queried (e.g., email in User collection, turfId in Booking collection) for optimized query performance.
- **Schema Validation:** Implement schema validation in MongoDB to enforce data integrity and ensure fields meet specific criteria (e.g., required fields, data types).

- **Scalability:** Design the database model to support scalability by considering potential growth in user base, turf listings, and booking transactions.

This database model serves as a foundational structure for storing and managing essential data entities in your Turf Booking Website project, ensuring efficient data handling and retrieval for various application functionalities.

CHAPTER 5

CONCLUSION

In conclusion, the Turf Booking Website project aims to revolutionize the way users discover and book sports turfs conveniently. By leveraging modern web technologies and a user-centric design approach, the platform provides a seamless experience for both users and turf owners alike. Key functionalities such as user authentication, dynamic turf listings, intuitive booking management, and robust administrative tools have been meticulously implemented to meet the diverse needs of our stakeholders.

Throughout the development process, emphasis was placed on security, scalability, and user experience, ensuring data integrity, efficient performance, and user satisfaction. The use of technologies such as Node.js, Express.js, MongoDB, and React enabled rapid development cycles and facilitated the integration of advanced features like real-time booking updates and secure payment processing.

Looking ahead, the Turf Booking Website is poised to expand its user base and enhance its service offerings with continuous improvements and customer-focused enhancements. By maintaining a commitment to excellence in technology and service delivery, we aim to establish the platform as a preferred choice for sports enthusiasts and turf owners alike.

CHAPTER 6

FUTURE ENHANCEMENTS

- Mobile Application Development
- Enhanced Search and Recommendation Engine
- Integration with Social Media Platforms
- Multi-language Support
- Dynamic Pricing and Discounts
- Enhanced Analytics and Reporting
- Intelligent Chatbot for Customer Support
- Expanded Payment Options
- Integration with Other Tools and APIs
- Advanced Search and Filters

APPENDIX – A

SOURCE CODE:

FRONT END:

header.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
  <title>Sportify</title>
</head>
<body>
  <header>
    <div class="header-container">
      <div class="logo">Sportify</div>
      <div class="search-bar">
        <input type="text" placeholder="Search for Cricket and Football Turf">
        <div class="search-icon">
          <button class="search-icons">
            <i class="fa-solid fa-search"></i>
          </button>
        </div>
      </div>
      <div class="location-signin">
        <select class="location">
          <option>Coimbatore</option>
          <option>Chennai</option>
          <option>Bangalore</option>
          <option>Delhi</option>
          <option>Mumbai</option>
        </select>
        <button class="signin"><a href="login.html">Sign In</a></button>
        <div class="side-navbar-toggle" onclick="openNavbar()">
          <i class="fa-solid fa-bars"></i>
        </div>
```

```
    </div>
   </div>
  </header>
  <nav class="navbar">
   <div class="navbar-links">
    <p class="navbar-link"><a href="index.html">Home</a></p>
    <p class="navbar-link"><a href="cricket.html">Cricket</a></p>
    <p class="navbar-link"><a href="football.html">Football</a></p>
    <p class="navbar-link"><a href="badminton.html">Badminton</a></p>
    <p class="navbar-link-rightone"><a href="offers.html">Offers</a></p>
    <p class="navbar-link-righttwo"><a href="Listyourarena.html">List Your Arena</a></p>
   </div>
  </nav>
  <div id="side-navbar" class="side-navbar">
   <a href="javascript:void(0)" class="close-btn" onclick="closeNavbar()">&times;</a>
   <a href="index.html">Home</a>
   <a href="cricket.html">Cricket</a>
   <a href="football.html">Football</a>
   <a href="badminton.html">Badminton</a>
   <a href="offers.html">Offers</a>
   <a href="Listyourarena.html">List Your Arena</a>
  </div>
<script src="header.js"></script>
</body>
</html>
```

header.js:

```
function openNavbar() {
  document.getElementById("side-navbar").style.width = "20%";
}

function closeNavbar() {
  document.getElementById("side-navbar").style.width = "0";
}
```

index.html:

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

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
```

```
 <link rel = "stylesheet" href="style.css">
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

</head>
<body>
 <script>
  fetch('header.html')
    .then(response => response.text())
    .then(data => {
     document.getElementById('header-placeholder').innerHTML = data;
    });
 </script>
 <div id="header-placeholder"></div>


 <div class = "slides">
  <div class = "slide slide-1">
   <img src = "images/cricket.jpeg" alt="">
  </div>
  <div class = "slide slide-2">
   <img src = "images/football.jpg" alt="">
  </div>
  <div class = "slide slide-3">
   <img src = "images/badminton.png" alt="">
  </div>
 </div>
 <h2 class = "heading-rec">Recommended Turfs and Courts</h2>
 <div class = "rec-images-1">
  <a href="playarenabookingpage.html"><img class = "image1" src = "Rec/rec2.jpeg" alt=""
height="250px" width="300px"></a>
  <a href="sachincricketbooking.html"><img class = "image2" src = "Rec/rec1.jpeg" alt=""
height="250px" width="300px"></a>
  <a href="kingsleycricketbooking.html"><img class = "image3" src = "Rec/rec3.jpeg" alt=""
height="250px" width="300px"></a>
  <a href="shreyabadmintonbooking.html"><img class = "image4" src = "Rec/rec4.jpeg"
alt="" height="250px" width="300px"></a>
 </div>
 <div class = "rec-name-1">
  <h3 class = "a1">Play Arena</h3>
  <h3 class = "a2">Sachin Cricket Turf</h3>
  <h3 class = "a3">Kingsley Cricket Turf</h3>
  <h3 class = "a4">Shreya Badminton Court</h3>
 </div>
```

```
  <div class = "rec-images-2">
    <a href="reshmifootballbooking.html"><img class = "image5" src = "Rec/rec5.jpeg" alt="" height="250px" width="300px"></a>
    <a href="vijaybadmintonbooking.html"><img class = "image6" src = "Rec/rec6.jpeg" alt="" height="250px" width="300px"></a>
    <a href="rameshcricketbooking.html"><img class = "image7" src = "Rec/rec7.jpeg" alt="" height="250px" width="300px"></a>
    <a href="sparklersbadmintonbooking.html"><img class = "image8" src = "Rec/rec8.jpeg" alt="" height="250px" width="300px"></a>
  </div>
  <div class = "rec-name-2">
    <h3 class = "a5">Reshmi Football Turf</h3>
    <h3 class = "a6">Vijay Badminton Court</h3>
    <h3 class = "a7">Ramesh Cricket Turf</h3>
    <h3 class = "a8">Sparklers Badminton Court</h3>
  </div>
  <div id="footer-placeholder"></div>
  <script>
    fetch('footer.html')
      .then(response => response.text())
      .then(data => {
        document.getElementById('footer-placeholder').innerHTML = data;
      });
  </script>
</body>
</html>
```

style.css:

```
* {
  padding: 0;
  margin: 0;
}
body {
  font-family: 'Poppins', sans-serif;
}
.header-container {
  display: flex;
  align-items: center;
  justify-content: space-between;
  background-color: white;
  padding: 10px 20px;
}
.logo {
  font-size: 24px;
  font-weight: bold;
}
.search-bar {
```

```
  flex: 0.7;
  margin: 0 80px;
  margin-right: 90px;
  display: flex;
}
.search-bar input {
  width: 100%;
  padding: 8px;
  box-sizing: border-box;
  border-radius: 30px;
}
.search-icons {
  background-color: transparent;
  border: none;
  font-size: 16px;
  margin-left: -30px;
  margin-bottom: 2px;
  margin-top: 8px;
}
.location-signin {
  display: flex;
  align-items: right;
  margin-left: 30px;
}
.location {
  font-size: 14px;
  border-radius: 10px;
  border: 2px solid;
}
.signin {
  font-size: 17px;
  margin-left: 20px;
  padding-left: 5px;
  padding-right: 5px;
  padding-top: -20px;
  border-radius: 10px;
  border: 2px solid;
  cursor: pointer;
}
.signin a {
  text-decoration: none;
  color: black;
}
.side-navbar-toggle {
  margin-left: 20px;
  cursor: pointer;
}
```

```
.location-signin select {
  padding: 8px;
  margin-left: -10px;
}
.navbar-links {
  margin-top: 10px;
  background-color: #f9f9f9;
  padding-left: 20px;
  display: flex;
  flex-direction: row;
  column-gap: 70px;
}
.navbar-link a {
  text-decoration: none;
  color: rgb(102, 95, 95);
}
.navbar-link-rightone {
  margin-left: 700px;
  display: flex;
  flex-direction: row;
  column-gap: 70px;
  text-decoration: none;
}
.navbar-link-rightone a {
  text-decoration: none;
  color: rgb(102, 95, 95);
}
.navbar-link-righttwo a {
  text-decoration: none;
  color: rgb(102, 95, 95);
}
.navbar-link:hover {
  text-decoration: underline;
}
.navbar-link-rightone:hover {
  text-decoration: underline;
}
.navbar-link-righttwo:hover {
  text-decoration: underline;
}
.side-navbar {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  right: 0;
```

```
  background-color: rgb(29, 34, 56);
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}
.side-navbar a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #fff;
  display: block;
  transition: 0.3s;
}
.side-navbar a:hover {
  color: #f1f1f1;
}
.side-navbar .close-btn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
}
.slides {
  height: 70vh;
  width: 80%;
  overflow: hidden;
  position: relative;
  border: 4px solid #79e0ee;
  box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;
  margin-left: 150px;
  margin-top: 30px;
  cursor: pointer;
}
.slide {
  position: absolute;
  height: 100%;
  opacity: 0;
  inset: 0;
  animation: slide-show 12s infinite;
}
.slide-2 {
  animation-delay: 4s;
}
.slide-3 {
  animation-delay: 8s;
}
.slide img {
```

```
  width: 100%;
  height: 100%;
  object-fit: cover;
  object-position: center;
}
@keyframes slide-show {
  0% {
    opacity: 0;
  }
  10% {
    opacity: 1;
  }
  20%,
  30% {
    opacity: 1;
    scale: 1.03;
  }
  50% {
    opacity: 0;
  }
}
.heading-rec {
  margin-top: 40px;
  margin-left: 22px;
}
.rec-images-1 {
  margin-right: 40px;
  margin-left: 40px;
  margin-top: 30px;
  display: flex;
  justify-content: space-between;
}
.image1,
.image2,
.image3,
.image4 {
  border-radius: 10px;
  cursor: pointer;
}
.rec-images-1 img:hover {
  transform: scale(1.03);
  transition: 0.5s;
  box-shadow: rgba(0, 0, 0, 0.56) 0px 22px 70px 4px;
}
.rec-name-1 {
  display: flex;
  margin-top: 10px;
```

```
}
.a1 {
  padding-left: 140px;
}
.a2 {
  padding-left: 250px;
}
.a3 {
  padding-left: 190px;
}
.a4 {
  padding-left: 170px;
}
.rec-images-2 {
  display: flex;
  justify-content: space-between;
  margin-right: 40px;
  margin-left: 40px;
  margin-top: 30px;
}
.image5,
.image6,
.image7,
.image8 {
  border-radius: 10px;
  cursor: pointer;
}
.rec-images-2 img:hover {
  transform: scale(1.03);
  transition: 0.5s;
  box-shadow: rgba(0, 0, 0, 0.56) 0px 22px 70px 4px;
  background-color: gray;
  content: "Book now";
}
.rec-name-2 {
  display: flex;
  margin-top: 10px;
}
.a5 {
  padding-left: 90px;
}
.a6 {
  padding-left: 190px;
}
.a7 {
  padding-left: 190px;
}
```

```
.a8 {
  padding-left: 170px;
}
.list {
  margin-top: 40px;
  background-color: rgb(29, 34, 56);
  color: white;
  font-size: 18px;
  display: flex;
  justify-content: center;
  padding-top: 10px;
  padding-bottom: 10px;
}
.btn-1 {
  background-color: rgb(217, 51, 107);
  margin-left: 100px;
  font-size: 18px;
  border-radius: 7px;
  padding: 5px;
  border: none;
  cursor: pointer;
}
.custcare-newsletter {
  background-color: rgb(29, 34, 56);
}
.cust-care {
  font-size: 40px;
  border: none;
  padding-top: 80px;
  padding-left: 400px;
  cursor: pointer;
  color: white;
}
.cust-care-word {
  font-size: 16px;
  color: rgb(155, 144, 144);
  padding-left: 332px;
  padding-top: 10px;
  cursor: pointer;
}
.cust-care-word:hover {
  color: white;
}
.newsletter {
  margin-left: 1100px;
  margin-bottom: 100px;
  margin-top: -100px;
```

```
  font-size: 46px;
  color: white;
  border-radius: 7px;
  padding: 5px;
  border: none;
  cursor: pointer;
}
.newsletter-word {
  margin-left: 1050px;
  margin-top: -105px;
  color: rgb(155, 144, 144);
  padding-bottom: 100px;
  cursor: pointer;
}
.newsletter-word:hover {
  color: white;
}
```

register.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Register Page</title>
<link rel="stylesheet" href="register.css">
</head>
<body>
<div class="container">
  <form id="registerForm" class="form">
    <h2>Register</h2>
    <input type="text" id="username" placeholder="Username" required>
    <input type="email" id="email" placeholder="Email" required>
    <input type="password" id="password" placeholder="Password" required>
    <button type="submit">Register</button>
    <p class="message">Already registered? <a href="login.html">Sign In</a></p>
    <p class="message"><a href="forgot-password.html">Forgot password?</a></p>
  </form>
</div>

<script src="script.js"></script>
</body>
</html>
```

register.css

```
body {
  font-family: Arial, sans-serif;
```

```
  background-color: #f0f0f0;
  background-image: url("images/loginpage.jpg");
}

.container {
  width: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.form {
  background-color: #fff;
  padding: 20px;
  max-width: 300px;
  width: 100%;
  text-align: center;
  box-shadow: 0px 0px 10px rgba(0,0,0,0.1);
  border-radius: 20px;
}

input[type="text"], input[type="password"], input[type="email"] {
  width: calc(100% - 20px);
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  margin: 10px 0;
  border: none;
  cursor: pointer;
  border-radius: 4px;
}

button:hover {
  opacity: 0.8;
}

.message {
  margin-top: 15px;
  font-size: 12px;
```

```
}

.message a {
  color: #4CAF50;
  text-decoration: none;
}

.message a:hover {
  text-decoration: underline;
}
```

login.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<link rel="stylesheet" href="login.css">
</head>
<body>
<div class="container">
  <form id="loginForm" class="form" onsubmit="return validateLogin()">
    <h2>Login</h2>
    <input type="text" id="username" placeholder="Username or Email" required>
    <input type="password" id="password" placeholder="Password" required>
    <button type="submit">Login</button>
    <p id="error-msg" class="error-message"></p>
    <p class="message">Not registered? <a href="register.html">Create an account</a></p>
    <p class="message"><a href="forgot-password.html">Forgot password?</a></p>
  </form>
</div>

<script src="userdetails.js"></script>
</body>
</html>
```

login.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  background-image: url("images/loginpage.jpg");
}

.container {
  width: 100%;
  display: flex;
```

```
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.form {
  background-color: #fff;
  padding: 20px;
  max-width: 300px;
  width: 100%;
  text-align: center;
  box-shadow: 0px 0px 10px rgba(0,0,0,0.1);
  border-radius: 20px;
}

input[type="text"], input[type="password"] {
  width: calc(100% - 20px);
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  margin: 10px 0;
  border: none;
  cursor: pointer;
  border-radius: 4px;
}

button:hover {
  opacity: 0.8;
}

.message {
  margin-top: 15px;
  font-size: 12px;
}

.message a {
  color: #4CAF50;
  text-decoration: none;
}
```

```
.message a:hover {
  text-decoration: underline;
}

.error-message {
  color: red;
  font-size: 12px;
  margin-top: 5px;
}
```

userdetails.js

```
function validateLogin() {
  // Retrieve input values
  var username = document.getElementById('username').value;
  var password = document.getElementById('password').value;
  var errorMsg = document.getElementById('error-msg');

  // Check if username and password match
  if (username === 'lokhith' && password === '12345678') {
    // Successful login, redirect or do something else
    alert('Login successful! Redirecting...');
    window.location.href = "index.html";
    return false;
  } else {
    // Invalid credentials, show error message
    errorMsg.textContent = 'Invalid username or password. Please try again.';
    return false; // Prevent form submission
  }
}
```

forgotpassword.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Forgot Password</title>
<link rel="stylesheet" href="forgot-password.css">
</head>
<body>
<div class="container">
  <form id="forgotPasswordForm" class="form">
    <h2>Forgot Password</h2>
    <input type="email" id="email" placeholder="Enter your email" required>
    <button type="submit">Reset Password</button>
    <p class="message">Remembered your password? <a href="login.html">Back to
Login</a></p>
```

```
   <p class="message">Don't have an account? <a href="register.html">Create an
account</a></p>
  </form>
</div>

<script src="script.js"></script>
</body>
</html>
```

forgotpassword.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  background-image: url("images/loginpage.jpg");
}

.container {
  width: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.form {
  background-color: #fff;
  padding: 20px;
  max-width: 300px;
  width: 100%;
  text-align: center;
  box-shadow: 0px 0px 10px rgba(0,0,0,0.1);
  border-radius: 20px;
}

input[type="email"] {
  width: calc(100% - 20px);
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 20px;
  margin: 10px 0;
```

```
  border: none;
  cursor: pointer;
  border-radius: 4px;
}

button:hover {
  opacity: 0.8;
}

.message {
  margin-top: 15px;
  font-size: 12px;
}

.message a {
  color: #4CAF50;
  text-decoration: none;
}

.message a:hover {
  text-decoration: underline;
}
```

cricket.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="cricket.css" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
  </head>
  <body>
    <div id="header-placeholder"></div>
    <script>
      fetch("header.html")
        .then((response) => response.text())
        .then((data) => {
          document.getElementById("header-placeholder").innerHTML = data;
        });
    </script>
```

```
   <h1 class="cri-head">Find Your Perfect Cricket Turf</h1>
   <div class="images-set1">
    <div class="card1">
     <a href="cricketturfbook.html"><img src="cricket/cr1.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-1">
       <ul class="rat-rev1">
        <li class="rat1"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev1">9.9/10    1.5K  Ratings</p>
       </ul>
      </div>
    </div>
    <div class="card2">
     <a href="cricketturfbook.html"><img src="cricket/cr2.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-2">
       <ul class="rat-rev2">
        <li class="rat2"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev2">9.4/10    1.6K  Ratings</p>
       </ul>
      </div>
    </div>
    <div class="card3">
     <a href="cricketturfbook.html"><img src="cricket/cr3.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-3">
       <ul class="rat-rev3">
        <li class="rat3"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev3">8.8/10    1.6K  Ratings</p>
       </ul>
      </div>
    </div>
   </div>
   <div class="images-set2">
    <div class="card4">
     <a href="cricketturfbook.html"><img src="cricket/cr4.jpg" height="370px"
width="330px" /></a>
      <div class="rat-rev-4">
       <ul class="rat-rev4">
        <li class="rat4"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev4">8.3/10    3.1K  Ratings</p>
       </ul>
      </div>
    </div>
    <div class="card5">
     <a href="cricketturfbook.html"><img src="cricket/cr5.jpeg" height="370px"
width="330px" /></a>
```

```
      <div class="rat-rev-5">
        <ul class="rat-rev5">
          <li class="rat5"><i class="fa-solid fa-star"></i>   </li>
          <p class="rev5">7.9/10    3.4K  Ratings</p>
        </ul>
      </div>
    </div>
    <div class="card6">
      <a href="cricketturfbook.html"><img src="cricket/cr6.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-6">
        <ul class="rat-rev6">
          <li class="rat6"><i class="fa-solid fa-star"></i>   </li>
          <p class="rev6">7.6/10    1.7K  Ratings</p>
        </ul>
      </div>
    </div>
  </div>
  <div id="footer-placeholder"></div>
  <script>
    fetch('footer.html')
      .then(response => response.text())
      .then(data => {
        document.getElementById('footer-placeholder').innerHTML = data;
      });
  </script>
 </body>
</html>
```

football.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <link rel="stylesheet" href="cricket.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
 </head>
 <body>
  <div id="header-placeholder"></div>
  <script>
```

```
    fetch("header.html")
      .then((response) => response.text())
      .then((data) => {
        document.getElementById("header-placeholder").innerHTML = data;
      });
  </script>
  <h1 class="cri-head">Find Your Perfect Football Turf</h1>
  <div class="images-set1">
   <div class="card1">
    <a href="cricketturfbook.html"><img src="football/fo1.jpeg" height="370px"
width="330px" /></a>
    <div class="rat-rev-1">
     <ul class="rat-rev1">
      <li class="rat1"><i class="fa-solid fa-star"></i>   </li>
      <p class="rev1">9.5/10    2.9K  Ratings</p>
     </ul>
    </div>
   </div>
   <div class="card2">
    <a href="cricketturfbook.html"><img src="football/fo2.jpeg" height="370px"
width="330px" /></a>
    <div class="rat-rev-2">
     <ul class="rat-rev2">
      <li class="rat2"><i class="fa-solid fa-star"></i>   </li>
      <p class="rev2">9.2/10    1.4K  Ratings</p>
     </ul>
    </div>
   </div>
   <div class="card3">
    <a href="cricketturfbook.html"><img src="football/fo3.jpeg" height="370px"
width="330px" /></a>
    <div class="rat-rev-3">
     <ul class="rat-rev3">
      <li class="rat3"><i class="fa-solid fa-star"></i>   </li>
      <p class="rev3">8.6/10    1.9K  Ratings</p>
     </ul>
    </div>
   </div>
  </div>
  <div class="images-set2">
   <div class="card4">
    <a href="cricketturfbook.html"><img src="football/fo4.jpeg" height="370px"
width="330px" /></a>
    <div class="rat-rev-4">
     <ul class="rat-rev4">
      <li class="rat4"><i class="fa-solid fa-star"></i>   </li>
      <p class="rev4">8.5/10    2.6K  Ratings</p>
```

```
        </ul>
       </div>
      </div>
      <div class="card5">
       <a href="cricketturfbook.html"><img src="football/fo5.jpeg" height="370px"
width="330px" /></a>
       <div class="rat-rev-5">
        <ul class="rat-rev5">
         <li class="rat5"><i class="fa-solid fa-star"></i>   </li>
         <p class="rev5">7.9/10    3.4K  Ratings</p>
        </ul>
       </div>
      </div>
      <div class="card6">
       <a href="cricketturfbook.html"><img src="football/fo6.jpeg" height="370px"
width="330px" /></a>
       <div class="rat-rev-6">
        <ul class="rat-rev6">
         <li class="rat6"><i class="fa-solid fa-star"></i>   </li>
         <p class="rev6">7.3/10    1.6K  Ratings</p>
        </ul>
       </div>
      </div>
    </div>
    <div id="footer-placeholder"></div>
    <script>
     fetch('footer.html')
      .then(response => response.text())
      .then(data => {
       document.getElementById('footer-placeholder').innerHTML = data;
      });
    </script>
  </body>
</html>
```

badminton.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="cricket.css" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
```

```
   <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
  </head>
  <body>
   <div id="header-placeholder"></div>
   <script>
    fetch("header.html")
     .then((response) => response.text())
     .then((data) => {
      document.getElementById("header-placeholder").innerHTML = data;
     });
   </script>
   <h1 class="cri-head">Find Your Perfect Badminton Court</h1>
   <div class="images-set1">
    <div class="card1">
     <a href="cricketturfbook.html"><img src="badminton/ba1.jpeg" height="370px"
width="330px" /></a>
     <div class="rat-rev-1">
      <ul class="rat-rev1">
       <li class="rat1"><i class="fa-solid fa-star"></i>   </li>
       <p class="rev1">9.7/10    3.6K  Ratings</p>
      </ul>
     </div>
    </div>
    <div class="card2">
     <a href="cricketturfbook.html"><img src="badminton/ba2.jpeg" height="370px"
width="330px" /></a>
     <div class="rat-rev-2">
      <ul class="rat-rev2">
       <li class="rat2"><i class="fa-solid fa-star"></i>   </li>
       <p class="rev2">9.4/10    2.9K  Ratings</p>
      </ul>
     </div>
    </div>
    <div class="card3">
     <a href="cricketturfbook.html"><img src="badminton/ba3.jpeg" height="370px"
width="330px" /></a>
     <div class="rat-rev-3">
      <ul class="rat-rev3">
       <li class="rat3"><i class="fa-solid fa-star"></i>   </li>
       <p class="rev3">8.8/10    3.1K  Ratings</p>
      </ul>
     </div>
    </div>
   </div>
   <div class="images-set2">
    <div class="card4">
```

```
      <a href="cricketturfbook.html"><img src="badminton/ba4.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-4">
       <ul class="rat-rev4">
        <li class="rat4"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev4">9.0/10    2.4K  Ratings</p>
       </ul>
      </div>
    </div>
    <div class="card5">
      <a href="cricketturfbook.html"><img src="badminton/ba5.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-5">
       <ul class="rat-rev5">
        <li class="rat5"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev5">8.6/10    1.9K  Ratings</p>
       </ul>
      </div>
    </div>
    <div class="card6">
      <a href="cricketturfbook.html"><img src="badminton/ba6.jpeg" height="370px"
width="330px" /></a>
      <div class="rat-rev-6">
       <ul class="rat-rev6">
        <li class="rat6"><i class="fa-solid fa-star"></i>   </li>
        <p class="rev6">8.4/10    1.3K  Ratings</p>
       </ul>
      </div>
    </div>
   </div>
   <div id="footer-placeholder"></div>
   <script>
    fetch('footer.html')
     .then(response => response.text())
     .then(data => {
      document.getElementById('footer-placeholder').innerHTML = data;
     });
   </script>
  </body>
</html>
```

cricket.css

```
/* General styles */
body {
  font-family: 'Poppins', sans-serif;
  margin: 0;
  padding: 0;
```

```
  background-color: #f8f9fa;
  color: #333;
}

h1.cri-head {
  text-align: center;
  margin-top: 20px;
  font-size: 2.5em;
  color: rgb(20, 111, 20);
}

/* Card styles */
.images-set1, .images-set2 {
  display: flex;
  justify-content: space-around;
  margin: 20px 0;
}

.card1, .card2, .card3, .card4, .card5, .card6 {
  position: relative;
  width: 330px;
  height: 370px;
  margin: 10px;
  border-radius: 10px;
  overflow: hidden;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  transition: transform 0.2s;
}

.card1:hover, .card2:hover, .card3:hover, .card4:hover, .card5:hover, .card6:hover {
  transform: scale(1.05);
}

.card1 img, .card2 img, .card3 img, .card4 img, .card5 img, .card6 img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}

/* Rating and Review styles */
.rat-rev-1, .rat-rev-2, .rat-rev-3, .rat-rev-4, .rat-rev-5, .rat-rev-6 {
  position: absolute;
  bottom: 0;
  width: 100%;
  background: rgba(0, 0, 0, 0.5);
  color: #fff;
  padding: 10px;
```

```
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.rat-rev1, .rat-rev2, .rat-rev3, .rat-rev4, .rat-rev5, .rat-rev6 {
  list-style: none;
  padding: 0;
  margin: 0;
  display: flex;
  align-items: center;
}

.rat1, .rat2, .rat3, .rat4, .rat5, .rat6 {
  font-size: 1.2em;
  color: rgb(217, 51, 107);
  margin-left: 50px;
}

.rev1, .rev2, .rev3, .rev4, .rev5, .rev6 {
  font-size: 1em;
}

/* Responsive design */
@media (max-width: 768px) {
  .images-set1, .images-set2 {
    flex-direction: column;
    align-items: center;
  }

  .card1, .card2, .card3, .card4, .card5, .card6 {
    width: 90%;
    height: auto;
  }
}
```

cricketturfbook.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Booking page</title>
  <link rel="stylesheet" href="bookingpage.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
```

```
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="header-placeholder"></div>
  <script>
    fetch("header.html")
      .then((response) => response.text())
      .then((data) => {
        document.getElementById("header-placeholder").innerHTML = data;
      });
  </script>
  <div class="name">
    <h3 class="turf-name">Book your slots now!</h3>
    <div id="dates-list">
      <!-- Dates of the week will be populated here -->
    </div>
  </div>
  <br>
  <script>
    // Function to get the next 7 days
    function getNext7Days() {
      const daysOfWeek = ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"];
      const monthsOfYear = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
"Oct", "Nov", "Dec"];
      const datesList = [];
      for (let i = 0; i < 7; i++) {
        const date = new Date();
        date.setDate(date.getDate() + i);
        const day = daysOfWeek[date.getDay()];
        const month = monthsOfYear[date.getMonth()];
        const dateString = `${day}, ${month} ${date.getDate()}`;
        datesList.push({ dateString, date });
      }
      return datesList;
    }

    // Populate the dates list
    const datesList = getNext7Days();
    const datesListElement = document.getElementById('dates-list');
    datesList.forEach(({ dateString, date }) => {
      const button = document.createElement('button');
      button.innerText = dateString;
      button.setAttribute('data-date', date.toISOString());
      button.addEventListener('click', function () {
```

```
      // Remove 'selected' class from all buttons
      document.querySelectorAll('#dates-list button').forEach(btn =>
btn.classList.remove('selected'));
      // Add 'selected' class to the clicked button
      button.classList.add('selected');
    });
    datesListElement.appendChild(button);
  });
 </script>

 <div class="slots">
  <h3 style="text-align: center;">Morning Slots</h3><br>
  <div class="slot-buttons" id="morning-slots">
   <!-- Morning slot buttons will be populated here -->
  </div>

  <h3 style="text-align: center;">Afternoon Slots</h3><br>
  <div class="slot-buttons" id="afternoon-slots">
   <!-- Afternoon slot buttons will be populated here -->
  </div>

  <h3 style="text-align: center;">Evening Slots</h3><br>
  <div class="slot-buttons" id="evening-slots">
   <!-- Evening slot buttons will be populated here -->
  </div>

  <h3 style="text-align: center;">Night Slots</h3><br>
  <div class="slot-buttons" id="night-slots">
   <!-- Night slot buttons will be populated here -->
  </div>
 </div>

 <div class="payment" id="payment-section">
  <h3>Total Payment: <span id="total-payment">0</span></h3>
  <button id="proceed-button">Proceed</button>

 </div>

 <script>
  document.getElementById('proceed-button').addEventListener('click', () => {
   const selectedDateButton = document.querySelector('#dates-list button.selected');
   const selectedSlots = document.querySelectorAll('.slot-buttons button.selected');

   if (!selectedDateButton) {
    alert('Please select a date.');
    return;
   }
```

```
    if (selectedSlots.length === 0) {
      alert('Please select at least one slot.');
      return;
    }

    const date = selectedDateButton.getAttribute('data-date');
    const slots = Array.from(selectedSlots).map(slot => slot.innerText);

    fetch('http://localhost:3000/api/bookings', { // Updated URL
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ date, slots }),
    })
    .then(response => response.json())
    .then(data => {
      if (data.success) {
        alert('Slots booked successfully!');
        // Optionally, you can add code here to update the UI to reflect the booked slots
      } else {
        alert('Slots are already booked.');
      }
    })
    .catch(error => {
      console.error('Error:', error);
      alert('An error occurred. Please try again.');
    });
  });
</script>


<script>
  // Function to create slot buttons
  function createSlotButtons(containerId, startHour, endHour) {
    const container = document.getElementById(containerId);
    for (let hour = startHour; hour < endHour; hour++) {
      const button = document.createElement('button');
      const nextHour = (hour + 1) % 24;
      const displayHour = hour % 12 === 0 ? 12 : hour % 12;
      const displayNextHour = nextHour % 12 === 0 ? 12 : nextHour % 12;
      const period = hour < 12 ? 'am' : 'pm';
      const nextPeriod = nextHour < 12 ? 'am' : 'pm';
      button.innerText = `${displayHour}:00${period} - ${displayNextHour}:00${nextPeriod}`;
      button.addEventListener('click', function () {
        button.classList.toggle('selected');
```

```
        updatePayment();
      });
      button.classList.add('large-button');
      container.appendChild(button);
    }
  }

  // Function to update payment
  function updatePayment() {
    const selectedSlots = document.querySelectorAll('.slot-buttons button.selected');
    const paymentPerSlot = 1000; // Example payment per slot
    const totalPayment = selectedSlots.length * paymentPerSlot;
    document.getElementById('total-payment').innerText = totalPayment;
  }

  // Create slot buttons
  createSlotButtons('morning-slots', 6, 12);
  createSlotButtons('afternoon-slots', 12, 16);
  createSlotButtons('evening-slots', 16, 22);
  createSlotButtons('night-slots', 22, 30); // 30 to cover up to 6am next day

  // Sticky payment section at the bottom of the screen
  window.addEventListener('scroll', function() {
    const paymentSection = document.getElementById('payment-section');
    const scrollY = window.scrollY;
    const windowHeight = window.innerHeight;
    const bodyHeight = document.body.clientHeight;

    if (scrollY + windowHeight >= bodyHeight) {
      paymentSection.classList.remove('sticky');
    } else {
      paymentSection.classList.add('sticky');
    }
  });
 </script>
</body>
</html>
```

offers.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Document</title>
 <link rel="preconnect" href="https://fonts.googleapis.com">
 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">

  <link rel="stylesheet" href="offers.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

</head>
<body>
  <div>
    <div id="header-placeholder"></div>
    <script>
      fetch("header.html")
        .then((response) => response.text())
        .then((data) => {
          document.getElementById("header-placeholder").innerHTML = data;
        });
    </script>
    <div class="offer-section">
      <div class = "offer-search">
        <input type = "text" id = "search" placeholder="Search for Offers">
        <button class = "search-icon">
          <i class = "fa-solid fa-magnifying-glass"></i>
        </button>
      </div>
      <div class = "cards" id = "card-offers">
        <div class = "card-box">
          <img src = "offersimg/offer1.jpg" alt = "" height = "350px" width = "250px">
          <p style = "font-weight: bold;">SBI Card</p>
          <p class = "card-info">Get upto 20% off on all SBI cards</p>
        </div>
        <div class = "card-box">
          <img src = "offersimg/offer2.jpg" alt = "" height = "350px" width = "250px">
          <p style = "font-weight: bold;">HDFC Card</p>
          <p class = "card-info">Get instant cashback upto 50% off on selected HDFC cards</p>
        </div>
        <div class = "card-box">
          <img src = "offersimg/offer3.jpeg" alt = "" height = "350px" width = "250px">
          <p style = "font-weight: bold;">ICICI Card</p>
          <p class = "card-info">Get upto 25% off on ICICI Debit cards</p>
        </div>
        <div class = "card-box">
          <img src = "offersimg/offer4.png" alt = "" height = "350px" width = "250px">
          <p style = "font-weight: bold;">Amazon Pay</p>
          <p class = "card-info">Get instant discount upto 30% by using Amazon Pay</p>
```

```
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer5.jpeg" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">Slice</p>
      <p class = "card-info">Use Slice and avail instant discount upto Rs500/-</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer6.jpeg" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">Yes Bank</p>
      <p class = "card-info">Get instant discount upto 30% by using Yes Bank Credit
Cards</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer7.jpeg" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">Axis Bank</p>
      <p class = "card-info">Get a Flipkart E-Voucher worth upto Rs500/-</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer8.jpeg" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">RBL Bank</p>
      <p class = "card-info">Get instant discount upto Rs450/- by using RBL Debit and Credit
Cards</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer9.jpeg" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">Federal Bank</p>
      <p class = "card-info">Get instant discount upto Rs100/- by using Federal Debit and
Credit Cards</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer10.jpg" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">Canara Bank</p>
      <p class = "card-info">Use Canara Bank Credit Cards and get cashback upto Rs230/-
</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer11.png" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">IFB Bank</p>
      <p class = "card-info">Book using IFB and get a voucher worth Rs300/-</p>
    </div>
    <div class = "card-box">
      <img src = "offersimg/offer12.png" alt = "" height = "350px" width = "250px">
      <p style = "font-weight: bold;">Google Pay</p>
      <p class = "card-info">Pay using Google Pay and get a stratch card worth upto Rs250/-
</p>
    </div>
    <div class = "card-box">
```

```
        <img src = "offersimg/offer13.jpg" alt = "" height = "350px" width = "250px">
        <p style = "font-weight: bold;">Paytm Wallet</p>
        <p class = "card-info">Pay using Paytm Wallet and get upto Rs300 cashback</p>
       </div>
       <div class = "card-box">
        <img src = "offersimg/offer14.jpeg" alt = "" height = "350px" width = "250px">
        <p style = "font-weight: bold;">DBS Bank</p>
        <p class = "card-info">Get flat discount of 10% on bookings above Rs1000/-</p>
       </div>
       <div class = "card-box">
        <img src = "offersimg/offer15.jpg" alt = "" height = "350px" width = "250px">
        <p style = "font-weight: bold;">Citi Bank</p>
        <p class = "card-info">Pay using Citi Bank Credit and Debit Cards and avail cashback
upto 12%</p>
       </div>
       <div class = "card-box">
        <img src = "offersimg/offer16.jpeg" alt = "" height = "350px" width = "250px">
        <p style = "font-weight: bold;">Simpl Pay</p>
        <p class = "card-info">Use Simpl Pay to book now and pay the money later</p>
       </div>
     </div>
    </div>
    <div id="footer-placeholder"></div>
    <script src = "offers.js"></script>
  </body>
</html>
```

offers.css:

```
.offer-search{
  width: 300px;
  border: none;
  display: flex;
  justify-content: space-between;
  padding-left: 50px;
  padding-top: 40px;
  padding-bottom: 30px;

}

.offer-search input{
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border-bottom: 2px solid #000;

}
.cards{
```

```
  padding:20px;
  display: flex;
  gap:10px;
  flex-wrap: wrap;
  justify-content: space-around;
}

.card-box{
  text-align: center;
  flex-basis: 20%;
  cursor: pointer;
  background-color: #000;
  padding-top: 30px;
  color: white;
  padding-left: 5px;
  padding-right: 5px;
  padding-bottom: 5px;
}

.card-info{
  padding-top: 20px;
}

.search-icon{
  background-color: transparent;
  border:none;
  margin-left: -20px;
}
```

offers.js:

```
var offerContainer = document.getElementById("card-offers")
var search = document.getElementById("search")
var offerlist = offerContainer.querySelectorAll("div")

search.addEventListener("keyup", function(){
  var enteredValue = event.target.value.toUpperCase()
  for(count = 0; count<offerlist.length; count++)
    {
      var offername = offerlist[count].querySelector("p").textContent
      if(offername.toUpperCase().indexOf(enteredValue) < 0)
        {
          offerlist[count].style.display = "none"
        }
        else
        {
          offerlist[count].style.display = "block"
        }
```

```
    }
})

fetch('footer.html')
  .then(response => response.text())
  .then(data => {
    document.getElementById('footer-placeholder').innerHTML = data;
});
```

footer.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
</head>
<body>
  <footer>
    <div class = "list">
      <p class = lista>List your Arena</p>     
      <p class = listb>Got a Cricket Turf, Football Turf, Badminton Court or something new?
Partner with us and get listed on Sportify</p>
      <div class = "list-btn">
        <a href = "Listyourarena.html"><button class = "btn-1">Contact Now</button></a>

      </div>
    </div>
    <div class = "custcare-newsletter">
      <p class="cust-care"><i class="fa fa-phone"></i></p>
      <p class="cust-care-word">24/7 Customer Support</p>
      <p class = "newsletter"><i class="fa fa-envelope"></i></p>
      <p class = "newsletter-word">Subscribe to newsletter</p>
    </div>
  </footer>
  </body>
</html>
```

Listyourarena.html:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">

  <link rel="stylesheet" href="Listyourarena.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />

</head>
<body>
  <div id="header-placeholder"></div>
  <script>
    fetch("header.html")
      .then((response) => response.text())
      .then((data) => {
        document.getElementById("header-placeholder").innerHTML = data;
      });
  </script>
  <h1 class = "title">What can you list?</h1>
  <div class = "list-images-1">
    <a href = "cricketcontact.html"><img class = "image1" src = "listingimg/list1.jpg" alt=""
height="250px" width="300px"></a>
    <a href = "footballcontact.html"><img class = "image2" src = "listingimg/list2.png" alt=""
height="250px" width="300px"></a>
    <a href = "badmintoncontact.html"><img class = "image3" src = "listingimg/list3.png"
alt="" height="250px" width="300px"></a>
  </div>
  <div class = "list-name-1">
    <h3 class = "a1">Cricket Turf</h3>
    <h3 class = "a2">Football Turf</h3>
    <h3 class = "a3">Badminton Court</h3>
  </div>

  <h1 class = "service">What are the services we offer?</h1>
  <div class = "service-images-1">
    <img class = "image1" src = "servicesimg/service1.png" alt="" height="350px"
width="400px">
    <img class = "image2" src = "servicesimg/service2.png" alt="" height="350px"
width="400px">
```

```
   <img class = "image3" src = "servicesimg/service3.png" alt="" height="350px"
width="400px">
  </div>
  <div class = "service-images-2">
   <img class = "image4" src = "servicesimg/service4.png" alt="" height="350px"
width="350px">
   <img class = "image5" src = "servicesimg/service5.png" alt="" height="350px"
width="345px">
   <img class = "image6" src = "servicesimg/service6.png" alt="" height="350px"
width="340px">
  </div>
  <div id="footer-placeholder"></div>
  <script>
     fetch('footer.html')
       .then(response => response.text())
       .then(data => {
         document.getElementById('footer-placeholder').innerHTML = data;
       });
  </script>
</body>
</html>
```

Listyourarena.css:

```
.title{
  text-align: center;
  font-size: 40px;
  margin-top: 30px;
}

.list-images-1{
  margin-right:300px;
  margin-left:300px;
  margin-top: 30px;
  padding: 5px;
  display:flex;
  justify-content: space-between;
  background-color: rgb(213, 213, 213);
}

.image1, .image2, .image3{
  border-radius: 10px;
  cursor: pointer;
}

.list-images-1 img:hover{
  transform: scale(1.03);
  transition: 0.5s;
```

```
  box-shadow: rgba(0, 0, 0, 0.56) 0px 22px 70px 4px;
}

.list-name-1{
  display:flex;
  margin-top:10px;
}

.a1{
  margin-left: 246px;
}

.a2{
  margin-left: -30px;
}

.a3{
  margin-left: -20px;
}

.service{
  text-align: center;
  font-size: 40px;
  margin-top: 160px;
}

.service-images-1{
  margin-right:200px;
  margin-left:200px;
  margin-top: 60px;
  padding: 5px;
  display:flex;
  text-align: center;
  justify-content: space-between;
  column-gap: 40px;
}

.image1, .image2, .image3{
  border-radius: 10px;
}

.service-images-2{
  margin-right:200px;
  margin-left:200px;
  margin-top: 30px;
  padding: 5px;
  display:flex;
```

```
  text-align: center;
  justify-content: space-between;
  column-gap: 40px;
  margin-bottom: 160px;
}

.image4, .image5, .image6{
  border-radius: 10px;
}
.image5{
  margin-left: -150px;
}

.image6{
  margin-left: -150px;
}
```

cricketcontact.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="cricketcontact.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="header-placeholder"></div>
    <script>
      fetch("header.html")
        .then((response) => response.text())
        .then((data) => {
          document.getElementById("header-placeholder").innerHTML = data;
        });
    </script><br><br>
    <h2 style="text-align: center;">List your Cricket Turf within 2 minutes</h2><br><br>
    <div class="form">
      <p class="name">Name</p>
      <input type="text" id="name" placeholder="Enter your Name"><br><br>
      <p class="mobile-no">Mobile no. </p>
      <input type="text" id="mobile" placeholder="Enter your Mobile no."><br><br>
```

```
    <p class="email">Email </p>
    <input type="text" id="email" placeholder="Enter your Email"><br><br>
    <p class="loca">Location</p>
    <select id="location" class="locality">
      <option value="">Select Location</option>
      <option>Coimbatore</option>
      <option>Chennai</option>
      <option>Bangalore</option>
      <option>Delhi</option>
      <option>Mumbai</option>
      <option>Other</option>
    </select><br><br>
    <p class="timing-option">Preferred Contact Time</p>
    <select id="contact-time" class="contact-time">
      <option value="">Select Time</option>
      <option>10:00am - 6:00pm</option>
      <option>8:00am - 8:00pm</option>
      <option>Any time</option>
    </select><br><br><br>
    <button class="submit-btn" onclick="handleSubmit()">Submit</button>
    <div id="message" style="margin-top: 20px; text-align: center;"></div>
   </div>
   <div id="footer-placeholder"></div>
   <script>
    fetch('footer.html')
      .then(response => response.text())
      .then(data => {
       document.getElementById('footer-placeholder').innerHTML = data;
      });
  </script>
  <script src = "cricketcontact.js"></script>
</body>
</html>
```

cricketcontact.css:

```
.form {
  flex: 0.7;
  margin: 0 80px;
  text-align: center;

}

.form input{
  width: 20%;
  padding: 8px;
  box-sizing: border-box;
  border-color: black;
```

```
  border: 1px solid black;
}

.name{
  text-align: center;
  margin-left:-228px;
  margin-bottom: 5px;
}

.mobile-no{
  text-align: center;
  margin-left:-196px;
  margin-bottom: 5px;
}

.email{
  text-align: center;
  margin-left:-233px;
  margin-bottom: 5px;
}

.loca{
  text-align: center;
  margin-left:-212px;
  margin-bottom: 5px;
}

.locality{
  font-size: 16px;
  border-radius: 0px;
  margin-left: -1px;
  padding: 6px;
  padding-right: 134px;
}

.timing-option{
  text-align: center;
  margin-left:-97px;
  margin-bottom: 5px;
}

.contact-time{
  font-size: 16px;
  border-radius: 0px;
  margin-left: 0px;
  padding: 6px;
  padding-right: 115px;
```

```
}

.submit-btn{
  text-align: center;
  margin-bottom: 5px;
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 15px;
  padding-right: 15px;
}
```

cricketcontact.js:

```
function handleSubmit() {
  const name = document.getElementById('name').value;
  const mobile = document.getElementById('mobile').value;
  const email = document.getElementById('email').value;
  const location = document.getElementById('location').value;
  const contactTime = document.getElementById('contact-time').value;
  const messageDiv = document.getElementById('message');

  if (!name || !mobile || !email || !location || !contactTime) {
    messageDiv.innerHTML = '<p style="color: red;">All fields are required.</p>';
  } else {
    messageDiv.innerHTML = '<p style="color: green;">Our executive will contact you
shortly.</p>';
  }
}
```

footballcontact.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="cricketcontact.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="header-placeholder"></div>
    <script>
      fetch("header.html")
```

```
      .then((response) => response.text())
      .then((data) => {
        document.getElementById("header-placeholder").innerHTML = data;
      });
  </script><br><br>
  <h2 style="text-align: center;">List your Football Turf within 2 minutes</h2><br><br>
  <div class="form">
    <p class="name">Name</p>
    <input type="text" id="name" placeholder="Enter your Name"><br><br>
    <p class="mobile-no">Mobile no. </p>
    <input type="text" id="mobile" placeholder="Enter your Mobile no."><br><br>
    <p class="email">Email </p>
    <input type="text" id="email" placeholder="Enter your Email"><br><br>
    <p class="loca">Location</p>
    <select id="location" class="locality">
      <option value="">Select Location</option>
      <option>Coimbatore</option>
      <option>Chennai</option>
      <option>Bangalore</option>
      <option>Delhi</option>
      <option>Mumbai</option>
      <option>Other</option>
    </select><br><br>
    <p class="timing-option">Preferred Contact Time</p>
    <select id="contact-time" class="contact-time">
      <option value="">Select Time</option>
      <option>10:00am - 6:00pm</option>
      <option>8:00am - 8:00pm</option>
      <option>Any time</option>
    </select><br><br><br>
    <button class="submit-btn" onclick="handleSubmit()">Submit</button>
    <div id="message" style="margin-top: 20px; text-align: center;"></div>
  </div>
  <div id="footer-placeholder"></div>
  <script>
    fetch('footer.html')
      .then(response => response.text())
      .then(data => {
        document.getElementById('footer-placeholder').innerHTML = data;
      });
  </script>
  <script src = "cricketcontact.js"></script>
</body>
</html>
```

badmintoncontact.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="cricketcontact.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="header-placeholder"></div>
    <script>
      fetch("header.html")
        .then((response) => response.text())
        .then((data) => {
          document.getElementById("header-placeholder").innerHTML = data;
        });
    </script><br><br>
    <h2 style="text-align: center;">List your Badminton Court within 2 minutes</h2><br><br>
    <div class="form">
      <p class="name">Name</p>
      <input type="text" id="name" placeholder="Enter your Name"><br><br>
      <p class="mobile-no">Mobile no. </p>
      <input type="text" id="mobile" placeholder="Enter your Mobile no."><br><br>
      <p class="email">Email </p>
      <input type="text" id="email" placeholder="Enter your Email"><br><br>
      <p class="loca">Location</p>
      <select id="location" class="locality">
        <option value="">Select Location</option>
        <option>Coimbatore</option>
        <option>Chennai</option>
        <option>Bangalore</option>
        <option>Delhi</option>
        <option>Mumbai</option>
        <option>Other</option>
      </select><br><br>
      <p class="timing-option">Preferred Contact Time</p>
      <select id="contact-time" class="contact-time">
        <option value="">Select Time</option>
        <option>10:00am - 6:00pm</option>
        <option>8:00am - 8:00pm</option>
        <option>Any time</option>
      </select><br><br><br>
```

```
    <button class="submit-btn" onclick="handleSubmit()">Submit</button>
    <div id="message" style="margin-top: 20px; text-align: center;"></div>
   </div>
   <div id="footer-placeholder"></div>
   <script>
    fetch('footer.html')
      .then(response => response.text())
      .then(data => {
       document.getElementById('footer-placeholder').innerHTML = data;
      });
  </script>
  <script src = "cricketcontact.js"></script>
</body>
</html>
```

BACK END:

Booking.js:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="cricketcontact.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfRyVX3pBnMFcV7oQPJkl9
QevSCWr3W6A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
  <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap"
rel="stylesheet">
</head>
<body>
  <div id="header-placeholder"></div>
   <script>
    fetch("header.html")
      .then((response) => response.text())
      .then((data) => {
       document.getElementById("header-placeholder").innerHTML = data;
      });
   </script><br><br>
   <h2 style="text-align: center;">List your Badminton Court within 2 minutes</h2><br><br>
   <div class="form">
    <p class="name">Name</p>
    <input type="text" id="name" placeholder="Enter your Name"><br><br>
```

```
    <p class="mobile-no">Mobile no. </p>
    <input type="text" id="mobile" placeholder="Enter your Mobile no."><br><br>
    <p class="email">Email </p>
    <input type="text" id="email" placeholder="Enter your Email"><br><br>
    <p class="loca">Location</p>
    <select id="location" class="locality">
      <option value="">Select Location</option>
      <option>Coimbatore</option>
      <option>Chennai</option>
      <option>Bangalore</option>
      <option>Delhi</option>
      <option>Mumbai</option>
      <option>Other</option>
    </select><br><br>
    <p class="timing-option">Preferred Contact Time</p>
    <select id="contact-time" class="contact-time">
      <option value="">Select Time</option>
      <option>10:00am - 6:00pm</option>
      <option>8:00am - 8:00pm</option>
      <option>Any time</option>
    </select><br><br><br>
    <button class="submit-btn" onclick="handleSubmit()">Submit</button>
    <div id="message" style="margin-top: 20px; text-align: center;"></div>
   </div>
   <div id="footer-placeholder"></div>
   <script>
    fetch('footer.html')
      .then(response => response.text())
      .then(data => {
       document.getElementById('footer-placeholder').innerHTML = data;
      });
  </script>
  <script src = "cricketcontact.js"></script>
</body>
</html>
```

server.js:

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
const mongoose = require('mongoose');
const path = require('path');
const Booking = require('./models/Booking');

const app = express();
const port = 3000;
```

```
app.use(cors());
app.use(bodyParser.json());

// Serve static files from the Frontend directory
app.use(express.static(path.join(__dirname, '..', 'Frontend')));

// MongoDB connection URL
const url =
'mongodb+srv://admin:admin%40123@cluster0.pxubscw.mongodb.net/Turf?retryWrites=tr
ue&w=majority'; // Replace with your MongoDB URL

// Connect to MongoDB
mongoose.connect(url, { useNewUrlParser: true, useUnifiedTopology: true })
  .then(() => console.log('Connected to MongoDB'))
  .catch(error => console.error('Error connecting to MongoDB:', error));

// Route to get booked slots for a specific date
app.get('/api/bookings/:date', async (req, res) => {
  const date = req.params.date;
  try {
    const booking = await Booking.findOne({ date });
    if (booking) {
      res.json(booking.slots);
    } else {
      res.json([]);
    }
  } catch (error) {
    res.status(500).send(error);
  }
});

// Route to book slots
app.post('/api/bookings', async (req, res) => {
  const { date, slots } = req.body;
  if (!date || !slots || !Array.isArray(slots)) {
    return res.status(400).send('Date and slots are required');
  }

  try {
    const booking = await Booking.findOne({ date });
    if (booking) {
      // Check if any of the requested slots are already booked
      const alreadyBooked = slots.some(slot => booking.slots.includes(slot));
      if (alreadyBooked) {
        return res.status(409).json({ success: false, message: 'Some slots are already booked.' });
      }
```

```
      // Add new slots to the existing booking
      booking.slots.push(...slots);
      await booking.save();
    } else {
      // Create a new booking document
      const newBooking = new Booking({ date, slots });
      await newBooking.save();
    }

    res.status(201).json({ success: true, message: 'Slots booked successfully.' });
  } catch (error) {
    res.status(500).send(error);
  }
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

APPENDIX – B

RESULT:

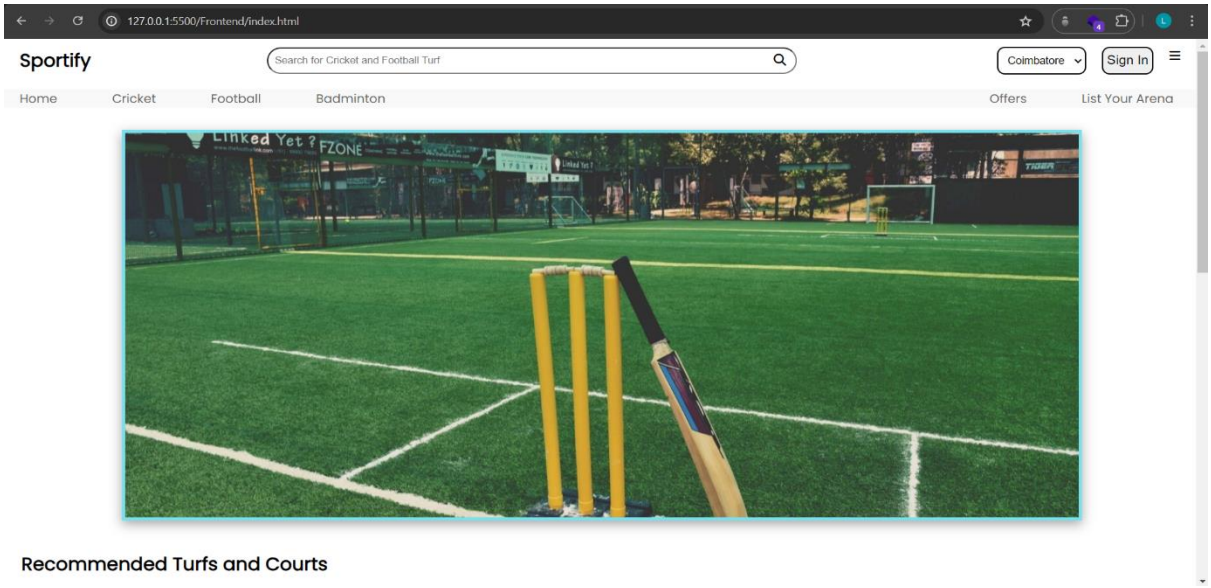

Fig 1.1 HOME PAGE

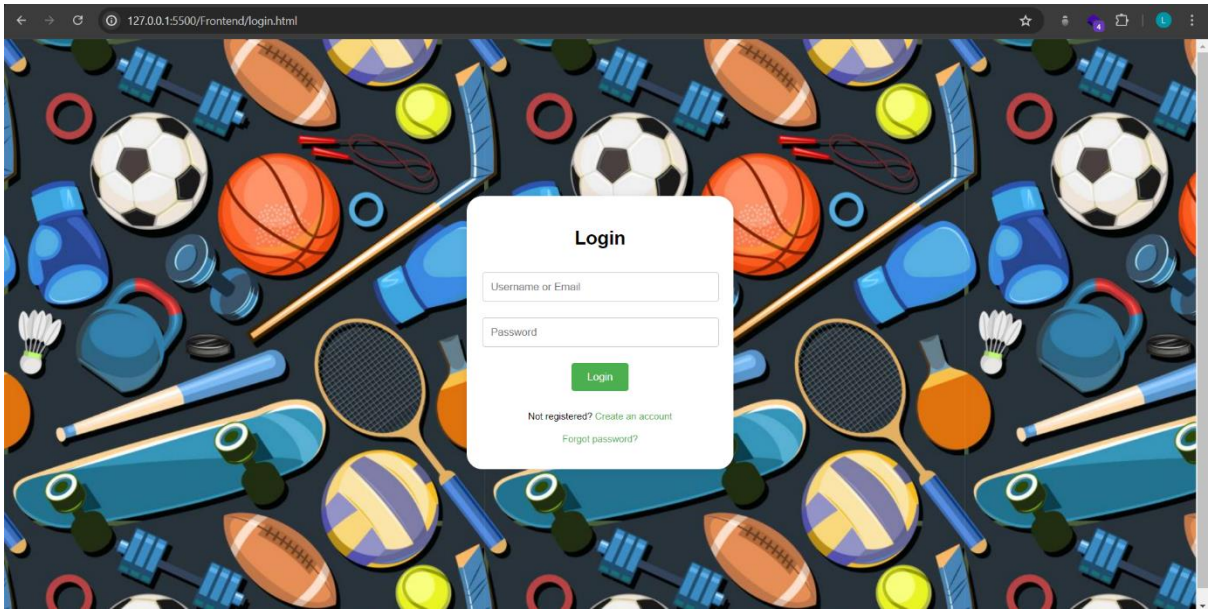

Fig 1.2 LOGIN PAGE

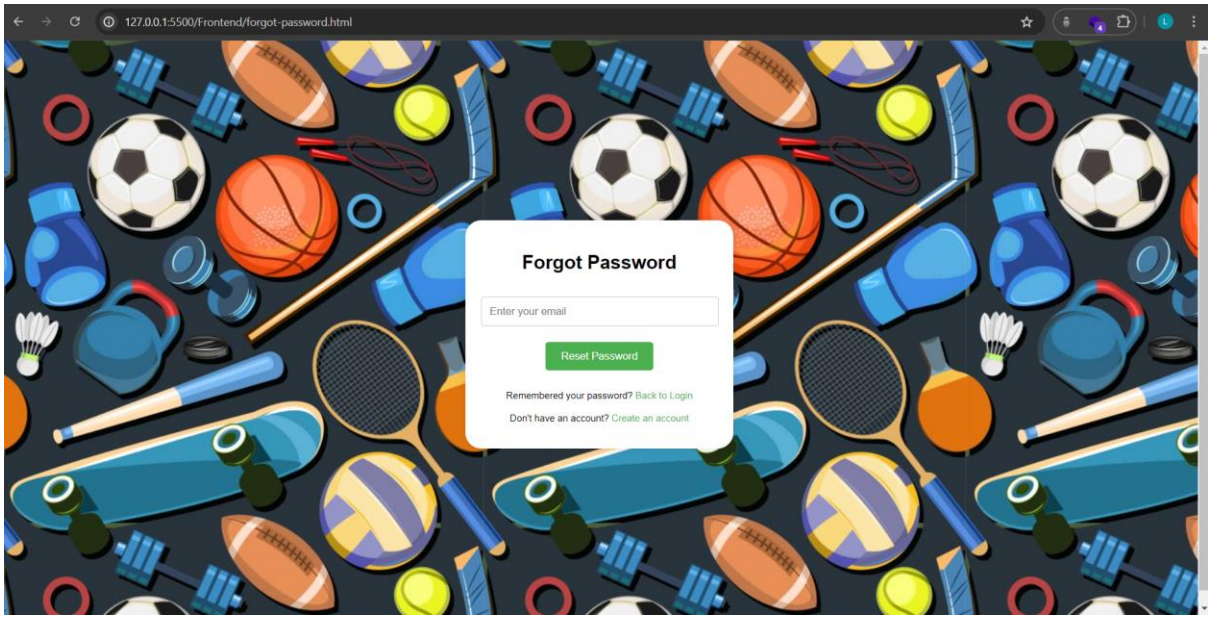

Fig 1.3 FORGOT PASSWORD

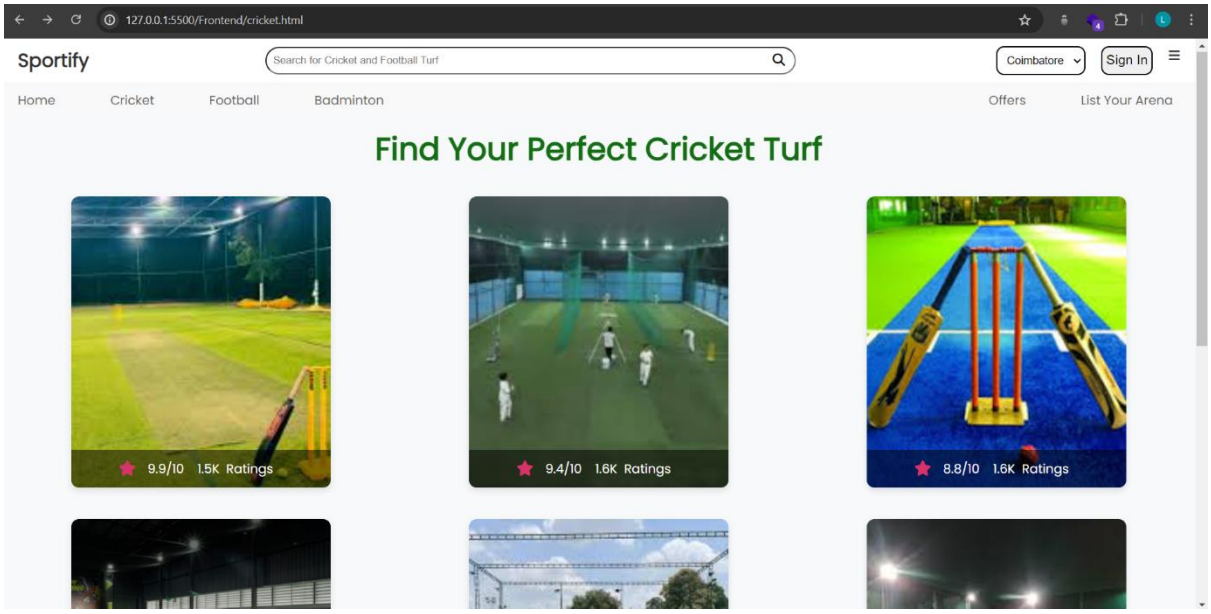

Fig 1.4 DISPLAYING CRICKET TURFS

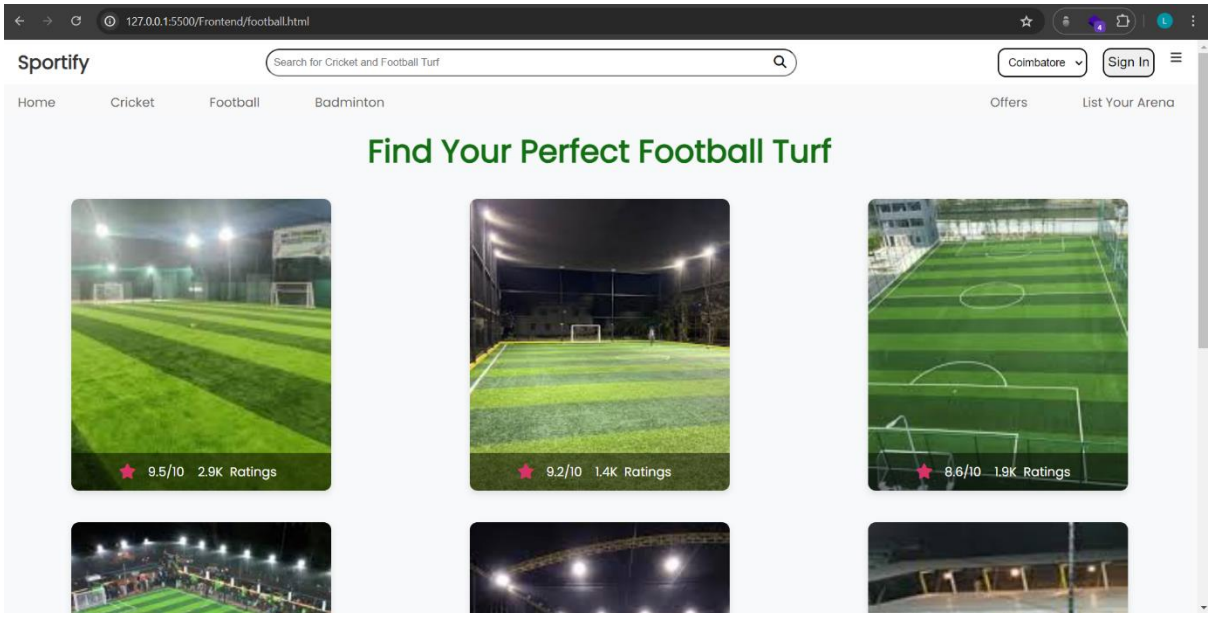

Fig 1.5 DISPLAYING FOOTBALL TURFS

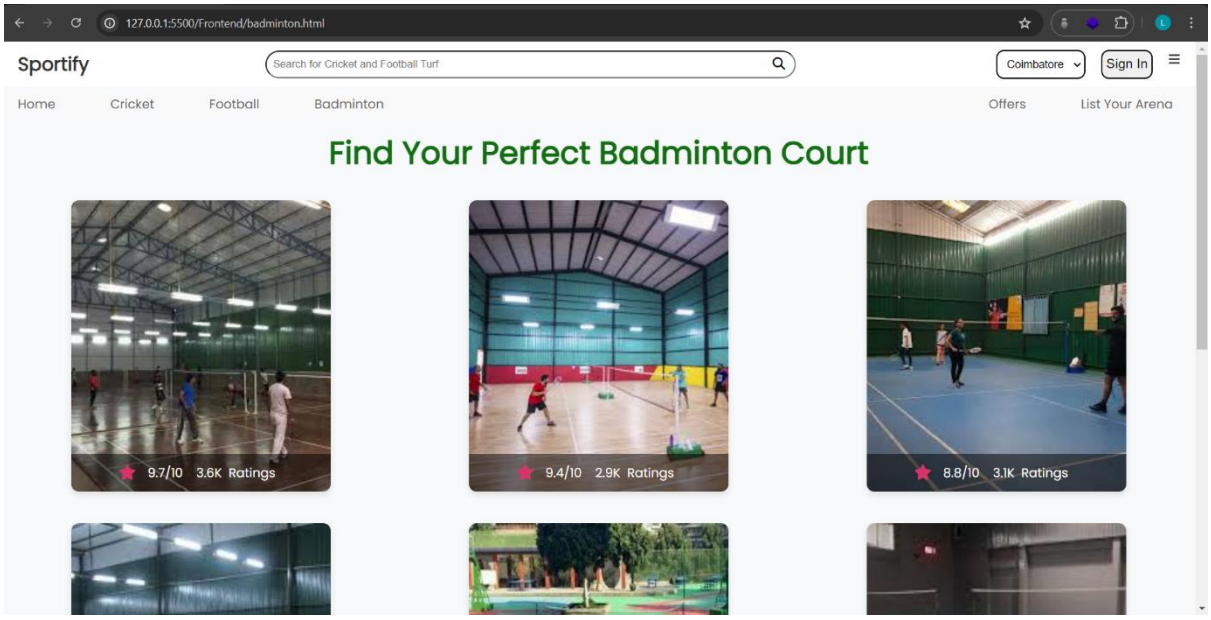

Fig 1.6 DISPLAYING BADMINTON COURTS

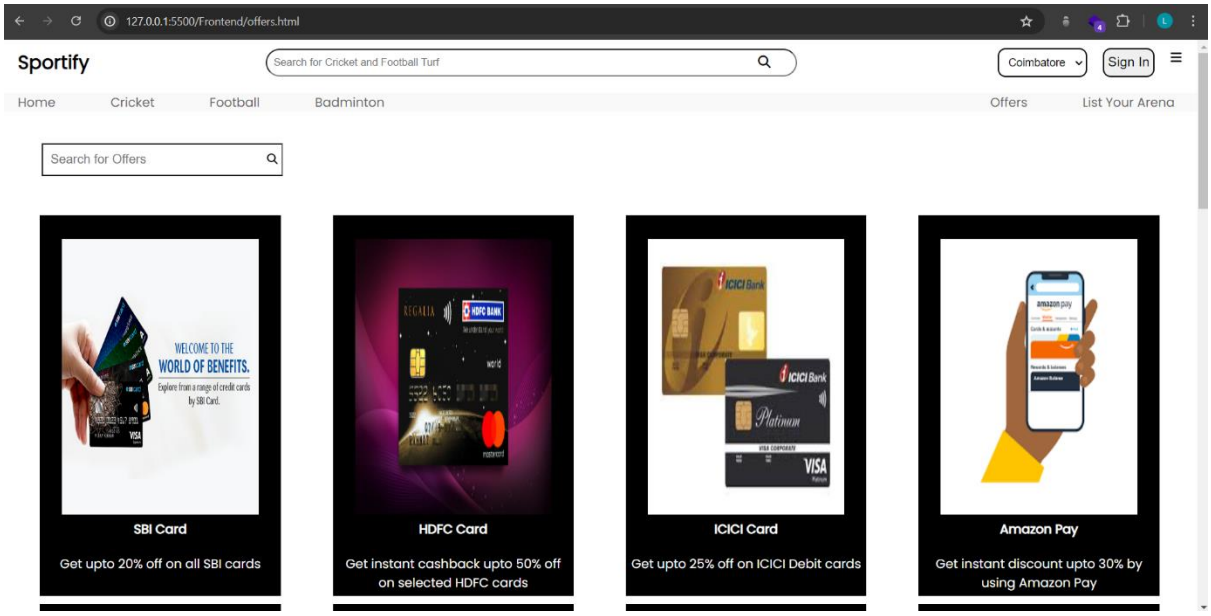

Fig 1.7 DISPLAYING OFFERS

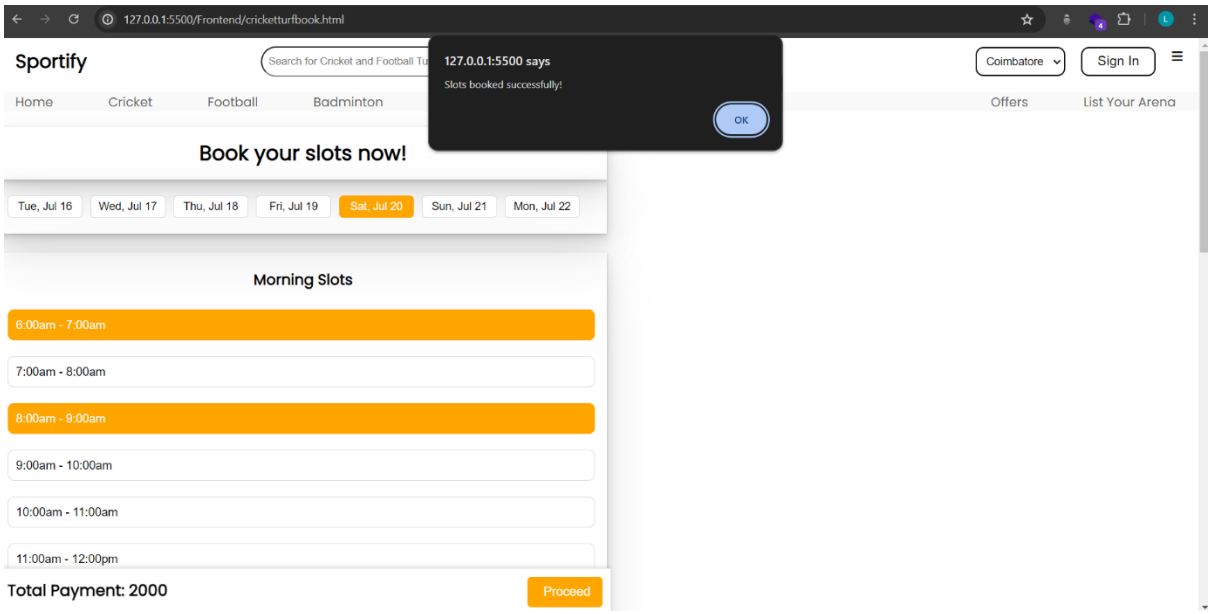

Fig 1.8 BOOKING PAGE

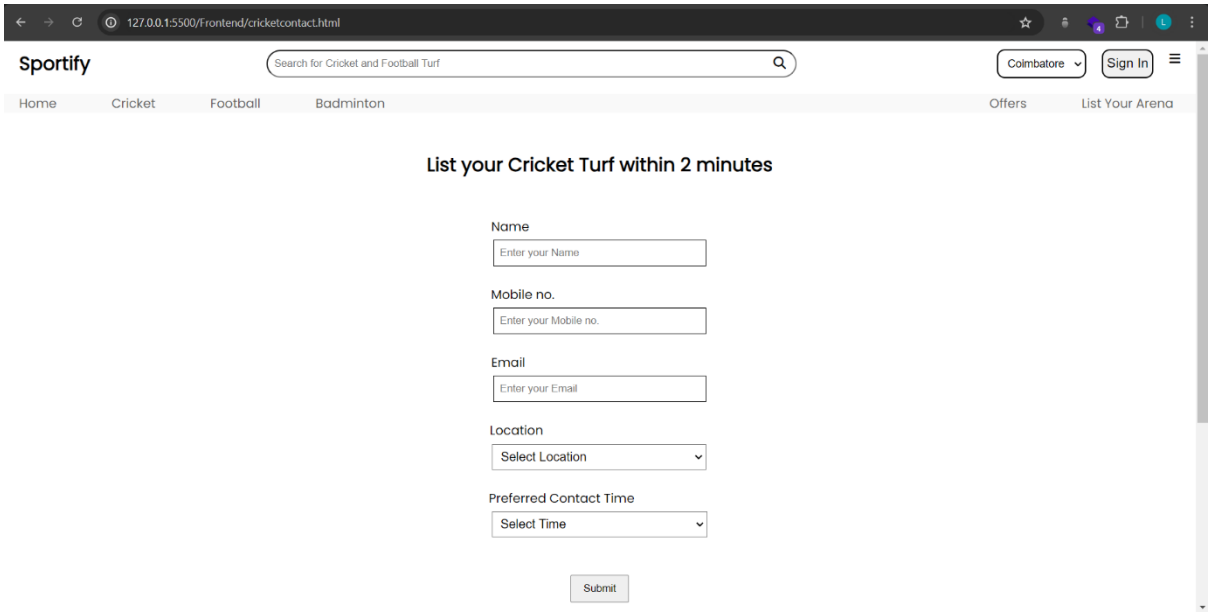

Fig 1.9 LISTING ARENA

CHAPTER 7

REFERENCES

- MongoDB Documentation. Retrieved from https://docs.mongodb.com/
- Node.js Documentation. Retrieved from https://nodejs.org/en/docs/
- React Documentation. Retrieved from https://reactjs.org/docs/getting-started.html
- "Building RESTful APIs with Node.js and Express" by Fernando Doglio. Packt Publishing, 2019.
- Medium Article: "Best Practices for JWT Authentication in Node.js" by John Doe. Retrieved from https://medium.com/@johndoe/best-practices-jwt-authentication-node-js
- Smashing Magazine Article: "Designing for Sports Enthusiasts: UX Strategies" by Jane Smith. Retrieved from https://www.smashingmagazine.com/designing-for-sports-enthusiasts-ux-strategies
- "Web Development with MongoDB and Node.js" by Jason Krol. O'Reilly Media, 2020.
- Research Paper: "User Behavior in Online Booking Systems" by A. Johnson et al. Journal of User Experience Research, 2018.
- BootstrapDocumentation Retrieved from https://getbootstrap.com/docs/5.0/getting-started/introduction/
- GDPR Compliance Guide for Website Operators. Retrieved from https://gdpr.eu/compliance-guide-for-website-operators/