Sports Field Booking Management System - Technical Documentation

# 1. Project Overview

Project Name: Sports Field Booking Management System

Purpose: To create a platform for users to book sports fields like soccer fields and for field owners to manage bookings and availability.

Target Users:

* - Players: Users can browse available fields and make bookings.
* - Field Owners: Manage field availability, prices, and booking schedules.

New Features:

* 1. Email Notifications: Sending booking confirmations and reminders via email.
* 2. Click-to-Open Map: Users can click a map to open the sports field location in a maps app.

# 2. Technical Requirements

Frontend: HTML5, CSS3, JavaScript (React , Bootstrap)

Backend: PHP (Laravel Framework)

Database: MySQL

Required APIs:

* - Email Service: send email notifications.
* - Google Maps API: For displaying field locations and map integration.

# 3. Functional Requirements

Core Features:

* - User Registration/Login: Users can sign up and manage their bookings.
* - Booking Availability and Management: Users can view available fields and make bookings. Owners can manage field availability and pricing.

New Features:

1. Email Notifications:

- After a booking is confirmed, the system sends an email confirmation to the user.

- Reminders are sent before the booking time.

1. Click-to-Open Map:

- Users can open the field location in Google Maps for easier navigation.

# 4. Workflow

Email Notification Flow:

1. User Books a Field:

- After the booking is completed, the system sends an email confirmation to the user.

2. Reminder Before Booking:

- An automatic email reminder are sent before the booking time.

Click-to-Open Map Flow:

1. User Views Field Details:

- The system displays a clickable map or a button to open the location in Google Maps.

2. Clicking the Map:

- Clicking opens the field’s location in the user’s preferred maps application.

# 5. Database Structure

Fields Table:

* - ID: Field ID.
* - Field Name: Name of the field.
* - Location (Coordinates): Latitude and longitude for the field.
* - Availability: Times the field is available for booking.
* - Price: Price per hour.

Bookings Table:

* - ID: Booking ID.
* - User ID: ID of the user who made the booking.
* - Field ID: ID of the booked field.
* - Date/Time: Date and time of the booking.
* - Status: Booking status (confirmed, canceled, etc.).

Users Table:

* - ID: User ID.
* - Email: User’s email address.
* - Phone Number: User’s phone number .
* - Password: Encrypted user password.

# 6. Design and User Experience (UX/UI)

User Interface:

* - Landing Page: Displays information about the platform with options to log in or sign up.
* - Fields Page: Users can browse available fields and filter by location, date, and time.
* - Booking Confirmation Page: After completing a booking, users receive booking details and are notified via email.
* - User Dashboard: Users can view and manage their bookings.

Email Notification Interface:

* - Booking Confirmation Page: After a booking is confirmed, the user is notified that an email confirmation were sent.

Click-to-Open Map Interface:

* - Field Details Page: A small map showing the field’s location with a button labeled "Open in Google Maps" to navigate directly to the field.

# 7.Non-Functional Requirements:

1. **Performance**:
   * The system must support up to 1000 concurrent users without performance degradation.
   * All pages should load within 3 seconds during normal usage.
2. **Scalability**:
   * The system must be scalable to accommodate a growing number of users and bookings over time.
3. **Security**:
   * User data, including passwords and personal information, must be encrypted.
   * Email communication must be secured through encryption during transmission (e.g., TLS).
   * Implement protection against security threats such as SQL injection and cross-site scripting (XSS).
4. **Usability**:
   * The system should be intuitive and easy to use for all users, with a simple and consistent user interface.
5. **Maintainability**:
   * The code should be modular and well-documented to allow for easy updates and debugging.
   * The system should support smooth updates with minimal downtime (not exceeding 15 minutes).
6. **Compatibility**:
   * The application should be compatible with all major browsers (Chrome, Firefox, Safari, Edge) and mobile devices (iOS, Android).
7. **Reliability**:
   * The system should maintain an uptime of 99.9%.
   * Daily data backups must be implemented, and recovery should be possible within 30 minutes of a failure.
8. **Accessibility**:
   * The platform should comply with accessibility standards such as WCAG 2.1, ensuring the platform is usable by people with disabilities.
9. **Responsiveness**:
   * The application must be fully responsive and work well on mobile, tablet, and desktop devices.
10. **SEO Optimization**:
    * The system should follow SEO best practices, including appropriate meta tags, structured data, and mobile-friendliness to ensure good search engine ranking.
11. **Email Notifications**:
    * Users must receive booking confirmation and reminder emails after a booking is completed.
    * Emails should be sent promptly and include booking details (field name, date, time, and booking reference).

# 8. Performance Criteria and SEO

Performance Optimization: Use caching mechanisms, reduce image sizes, and minimize API calls to improve speed.

SEO Best Practices: Ensure each page has appropriate meta tags, is mobile-friendly, and follows best practices for search engine ranking.

# 9. Conclusion

The Sports Field Booking Management System is a comprehensive platform for booking and managing sports fields, featuring new additions like SMS and email notifications for user convenience and map integration for easier navigation.