**ONLINE SPORTS ATTENDANCE MANAGEMENT SYSTEM**

PROJECT WORK SUBMITTED TO

THE DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

ARUL ANANDAR COLLEGE (AUTONOMOUS)

IN PARTIAL FULFILLMENT OF THE DEGREE OF

**BACHELOR OF COMPUTER SCIENCE**

By

**P. SAHAYOGESHWARAN**

**REG. NO. 21CSC132**

Under the Guidance of

**Mrs.V. RANI, MCA.,M.Phil.,M.Ed.,SET.,PGDSA.,**

Assistant Professor , Department of Computer Science and Applications



**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

**ARUL ANANDAR COLLEGE (AUTONOMOUS)**

**(Affiliated to Madurai Kamaraj University)**

**Reaccredited by NAAC at ‘A’ Grade**

**KARUMATHUR – 625 514, MADURAI, TAMIL NADU.**

**April2024**

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

**ARUL ANANDAR COLLEGE (AUTONOMOUS)**

**(Affiliated to Madurai Kamaraj University)**

**Reaccredited by NAAC at ‘A’ Grade**

**KARUMATHUR – 625 514, MADURAI, TAMIL NADU.**

****

**CERTIFICATE**

This is to certify that the project entitled as **“ONLINE SPORTS ATTENDANCE MANAGEMENT SYSTEM”** is a bonafide work done by

**P. SAHAYOGESHWARAN, Reg. No. 21CSC132** in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science of Madurai Kamaraj University.

|  |  |
| --- | --- |
| **Ms.I. JULIET SHANTHI, MCA., M.Phil.**  Head of the Department  Dept. of Computer Science and Applications  Arul Anandar College (Autonomous)  Karumathur – 625 514. | **Mrs.V.RANI, MCA.,M.Phil.,M.Ed.,SET.,PGDSA.,**  Project Guide  Dept. of Computer Science and Applications  Arul Anandar College (Autonomous)  Karumathur – 625 514. |

The viva voice examination of this miniproject work was held on …………………….

**INTERNAL EXAMINER EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First of all, I record my gratefulness to the God almighty for his blessing to make this project a great success.

I would like to acknowledge **Rev.Dr.M. Anbarasu SJ,** Principal, Arul Anandar College for his blessings and continuous motivation.

I wish to express my gratitude to **Dr.A. Sundararaj,** Deputy Principal**,** Arul Anandar College for his administrative support to take up this miniproject.

It is grateful to **Rev.Fr.V. Innaci John SJ**, Vice Principal (Shift – II**)** and coordinator of Computer science & Applicationsfor providing computing infrastructures and other provisions towards the development and deployment of this project.

I would like to express my thanks to **Ms. I. Juliet Shanthi,** Head, Department of Computer Science and Applications for providing interfaces and resources for data collection and project documentation.

I extent my gratitude to **Mrs.V. Rani,** Assistant professor, Department of Computer Science and Applications for her guidance and scrutiny towards the project design and development.

Finally, I express my gratitude and appreciations to my parents, and friends for their encouragement and continuous support from the startup till the completion of my project.

**P. SAHAYOGESHWARAN**

**(21CSC132)**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENT** | **PAGE NO** |
| 1 | **INTRODUCTION**  1.1 About the Project  1.2 Company Profile |  |
| 2 | **Problem Definition and Description**  2.1 Problem Definition  2.2 Problem Description |  |
| 3 | **SYSTEM ANALYSIS**  3.1 Existing System  3.2 Problems in the Existing System  3.3 Proposed System  3.4 Benefits in the Proposed System  3.5 Feasibility Analysis  3.6 Cost Benefit Analysis  3.7 Data Flow Diagram |  |
| 4 | **SYSTEM DESIGN**  4.1 Architectural Design (Module Design)  4.2 Description about the Modules  4.3 Input Design  4.4 Output Design  4.5 Data base Design  4.5.1 Normalization  4.5.2 Table structures with primary key and its data types  4.6 Software Description |  |
| 5 | **IMPLEMENTATION** |  |
| 6 | **TESTING**  6.1 Types of Testing  6.2 Write what are the test you conducted in your project |  |
| 7 | **INSTALLATION**  7.1 Hardware Requirements  7.2 Software Requirements |  |
| 8 | **MAINTENANCE** |  |
| 9 | **APPLICATIONS** |  |
| 10 | **CONCLUSION** |  |
|  | **BIBLIOGRAPHY** |  |
|  | **APPENDICES** |  |

**CHAPTER 1**

**INTRODUCTION**

**1.INTRODUCTION**

**1.1 Problem Statement**

The Project titled “**Online Sports Attendance Management System**” deal with the maintenance of attendance details. It is for maintaining daily basis of Attendance, the staff will be provided with the separate username and password to make student attendance. The Staff handling the particular subject to responsible to make the attendance of all Students. Only if the student present in the particular date, the attendance will be calculated. The student attendance report based on monthly and consolidate will be generated. Here, there is an option to apply for leave and send the feedback.

**Objectives**

**1. Efficient Attendance Tracking**

The primary goal of the system is to provide a seamless and efficient way to track the attendance of players, coaches, officials, and spectators during sports events.

Utilizing modern technologies such as QR codes, RFID, or biometrics can enhance the accuracy and speed of the attendance tracking process.

**2. Realtime Monitoring**

The system offers realtime monitoring capabilities, allowing organizers to access live attendance data during events.

This feature is crucial for making informed decisions, managing crowd control, and ensuring the overall success of the sports event.

**3. UserFriendly Interface**

A userfriendly interface ensures that the system is accessible to a wide range of users, including event organizers, administrators, and attendees.

Intuitive design and easy navigation contribute to the system's usability.

**4. Customization**

The system is designed to be adaptable to various sports and event formats. Event organizers can customize the system to meet the specific requirements of different sports and venues.

**5. Reporting and Analytics**

Robust reporting and analytics tools provide valuable insights into attendance trends, helping organizers make datadriven decisions for future events.

These features contribute to the continuous improvement of event planning and management strategies.

**6. Integration with Other Systems**

Seamless integration with other systems, such as ticketing platforms, team management software, and scheduling tools, enhances the overall efficiency of sports event management.

**Features**

**1. Event Creation and Configuration**

Organizers can create and configure sports events, specifying details such as date, time, venue, and participating teams.

**2. Attendance Tracking**

Multiple options for attendance tracking, including QR codes, RFID, and biometrics, ensure accuracy and efficiency.

**3. User Registration and Management**

Attendees, including players, coaches, officials, and spectators, can register and manage their profiles through the system.

**4. Realtime Monitoring Dashboard**

An interactive dashboard provides realtime attendance data, allowing organizers to monitor the event's progress and make informed decisions.

**5. Notifications and Alerts**

The system sends notifications and alerts to attendees and organizers, keeping them informed about event updates, schedule changes, and important announcements.

**6. Reporting and Analytics**

Comprehensive reports and analytics tools offer insights into attendance patterns, helping organizers optimize future events.

**1.2 COMPANY PROFILE**

Good Will Papers Ltd is to lead the paper is a dynamic and innovative Good Will Papers Ltd company dedicated to providing sustainable and highquality paper products. Founded in 24 February 1994, we are committed to revolutionizing the paper industry by combining innovation, sustainability, and customer focus.

**Mission Statement**

Our mission at Good Will Papers Ltd is to lead the paper industry in environmental responsibility, delivering exceptional products while minimizing our ecological footprint.

**Vision Statement**

We envision a future where sustainable paper production is the norm, and our efforts contribute to a healthier planet and more conscious consumer choices.

**Core Values**

**Integrity** We conduct all our business with the highest ethical standards.

**Innovation** We embrace creativity and strive for continuous improvement.

**Sustainability** We are committed to minimizing our environmental impact and promoting social responsibility.

**Customer Focus** We prioritize customer satisfaction and strive to exceed expectations.

**Products/Services**

Good Will Papers Ltd offers a wide range of ecofriendly paper products designed to meet the needs of businesses and individuals. Our products aim to provide an environmentally responsible alternative without compromising quality.

**Target Market**

Our target market includes environmentally conscious consumers, businesses, and organizations seeking sustainable paper solutions. We cater to those who prioritize ethical and ecofriendly choices in their paper consumption.

**Unique Selling Proposition (USP)**

What sets Good Will Papers Ltd apart is our dedication to not only delivering exceptional paper products but doing so in an environmentally sustainable manner. Our commitment to ecofriendly practices ensures that our customers receive highquality products with a reduced environmental impact.

**CHAPTER 2**

**PROBLEM DEFINATION**

**AND DESCRIPTION**

**PROBLEM DEFINITION AND DESCRIPTION**

**2.1 Problem Definition**

**Inaccuracy and Time Consuming Processes**

Traditional methods of taking attendance are prone to human error, leading to inaccuracies in recording participant and spectator numbers.

Manually counting attendance is time consuming and can result in delays, especially during largescale sporting events.

**Lack of Real Time Monitoring**

The absence of real time monitoring makes it challenging for organizers to have immediate insights into the attendance levels during an ongoing sports event.

Without live data, organizers may struggle to make timely decisions related to crowd control, security, and event logistics.

**Inefficient Communication**

Traditional systems often lack effective communication channels for notifying attendees and organizers about event updates, changes in schedules, or important announcements.

Inefficient communication can lead to confusion among participants and attendees.

**Limited Customization and Adaptability**

Manual methods do not easily accommodate the diverse needs of various sports events, such as different formats, venues, and participant categories.

Lack of customization options can hinder the flexibility required for specific sports event management.

**Difficulty in Reporting and Analysis**

Gathering, analyzing, and presenting attendance data for postevent evaluation is a cumbersome task with manual systems.

Inadequate reporting and analysis capabilities make it challenging for organizers to derive actionable insights for future events.

**Security Concerns**

Manual attendance tracking systems may lack robust security measures, potentially compromising the integrity of attendance records.

Unauthorized access to attendance data poses a risk to the overall management of sports events.

**Integration Challenges**

Existing systems may struggle to integrate seamlessly with other event management tools, ticketing platforms, or scheduling systems.

Lack of integration can result in disjointed workflows and data redundancy.

**Limited Accessibility and User Friendliness**

Traditional methods may not be easily accessible to all stakeholders, including organizers, participants, and spectators.

A lack of user friendly interfaces can hinder the adoption of attendance management systems by individuals with varying technical skills.

**2.2 Problem Description**

**Inaccuracy and Time Consuming Processes**

**Problem** Manual counting of participants and spectators is prone to human error, leading to inaccurate attendance records.

**Solution** An Online Sports Attendance Management System with automated tracking mechanisms, such as QR codes, RFID, or biometrics, can significantly reduce errors and provide real time accuracy.

**Lack of Real Time Monitoring**

**Problem** Traditional methods lack the capability for real time monitoring, making it challenging to manage events dynamically.

**Solution** The online system offers a live dashboard, allowing organizers to monitor attendance in real time, make informed decisions promptly, and optimize event logistics.

**Inefficient Communication**

**Problem** Manual systems often lack effective communication channels, causing delays and confusion regarding event updates.

**Solution** The online system includes communication features like push notifications or email alerts to keep attendees and organizers informed about any changes or important announcements.

**Limited Customization and Adaptability**

**Problem** Manual methods do not easily adapt to the diverse needs of various sports events, limiting flexibility.

**Solution** The online system is designed to be highly customizable, accommodating different sports, event formats, and participant categories, providing flexibility for varied requirements.

**Difficulty in Reporting and Analysis**

**Problem** Manual systems make post event reporting and analysis a time consuming and error prone process.

**Solution** The online system includes robust reporting and analytics tools, enabling organizers to easily generate comprehensive reports, analyze attendance trends, and derive valuable insights for future planning.

**Security Concerns**

**Problem** Manual attendance tracking systems may lack security measures, risking unauthorized access and data manipulation.

**Solution** The online system incorporates secure access controls and encryption to protect attendance data, ensuring the integrity and confidentiality of the information.

**Integration Challenges**

**Problem** Existing systems may struggle to integrate with other event management tools, leading to disjointed workflows.

**Solution** The online system is designed for seamless integration with ticketing platforms, scheduling tools, and other event management systems, promoting a more cohesive and efficient process.

**Limited Accessibility and User Friendliness**

**Problem** Traditional methods may not be easily accessible to all stakeholders, hindering widespread adoption.

**Solution** The online system features a user friendly interface, ensuring accessibility for organizers, participants, and spectators, regardless of their technical proficiency.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3. Existing System**

**3.1 Existing System**

Traditionally, sports event attendance management relies on manual methods, such as paper signin sheets, physical ticket scanning, and manual headcounts. These methods are not only timeconsuming but also prone to inaccuracies and inefficiencies. Organizers often struggle to maintain uptodate attendance records, which can lead to administrative errors and an inconsistent user experience. Moreover, the onset of the COVID19 pandemic has highlighted the need for more robust and contactless attendance tracking systems.

**3.2 Proposed System**

The "Online Sports Attendance Management System" is a comprehensive digital solution designed to revolutionize the way sports event attendance is managed. The proposed Online Sports Attendance Management System is an integrated, user friendly platform designed to automate and streamline the process of tracking attendance at sports events. The system leverages modern technologies to provide accurate, real time data and enhance overall efficiency in event management. Automated Attendance Tracking Utilizing QR codes, RFID, or biometrics for accurate and efficient attendance recording. Realtime Monitoring Dashboard Providing organizers with a live dashboard for instant insights into attendance levels during events. Communication Module Facilitating effective communication with attendees and organizers through push notifications and email alerts. Customization Allowing organizers to customize the system based on the specific requirements of different sports, venues, and event formats. Reporting and Analytics Tools Offering robust tools for generating comprehensive reports, analyzing attendance trends, and making data driven decisions. Security Measures Implementing secure access controls and encryption to ensure the integrity and confidentiality of attendance data. Integration Capabilities Seamlessly integrating with ticketing platforms, scheduling tools, and other event management systems for a cohesive work flow. User Friendly Interface Ensuring accessibility for organizers, participants, and spectators with an intuitive and easy to navigate interface.

**Benefits in the Proposed System**

Accuracy and Efficiency Automated tracking reduces errors and enhances the efficiency of attendance recording.Realtime Decision MakingThe live dashboard allows organizers to make immediate decisions regarding crowd control and event logistics. Improved Communication Efficient communication features keep attendees and organizers wellinformed about event updates.Flexibility and Adaptability Customization options cater to the diverse needs of various sports events, promoting flexibility.DataDriven Insights Robust reporting and analytics tools provide valuable insights for better planning and management.

Enhanced Security Implementation of secure access controls and encryption ensures the integrity and confidentiality of attendance data.Integrated Workflow Seamless integration with other systems promotes a more cohesive and efficient event management process.

**Feasibility Analysis**

Technical Feasibility The proposed system leverages existing technologies like QR codes, RFID, and biometrics, making it technically feasible and adaptable.Operational Feasibility The system's userfriendly interface ensures ease of use for organizers, participants, and spectators, enhancing operational feasibility.

**Cost Benefit Analysis**

**Costs**

Development and Implementation Costs

Training Costs

Maintenance and Support Costs

**Benefits**

Time Savings in Attendance Tracking

Improved Decision Making

Enhanced Event Planning

Increased Attendee Satisfaction

The benefits, including improved efficiency and attendee satisfaction, are expected to outweigh the initial and ongoing costs, resulting in a positive return on investment.

**DATA FLOW DIAGRAM**

**ADMIN**

**DATABASE**

**STAFF**

**LOGIN MODEL**

**DATABASE**

**OUTPUT**

**LOGIN**

**STAFF LOGIN**

**DATABASE**

**OUTPUT**

**DATA OF THE STUDENTS**

**STAFF LOGIN**

**STAFF LOGIN**

**DATABASE**

**OUTPUT**

**DATA OF THE STUDENTS**

**STAFF LOGIN**

**CHAPTER 4**

**SYSTEM DESIGN**

**4.SYSTEM DESIGN**

**4.1 Design Specifications**

**User Registration**

The system will allow users to register with a unique username and password.

**Event Creation**

Organizers can create events, specifying event details, including name, date, time, and location.

**Attendance Tracking**

Participants can check in using QR codes or NFC tags, while spectators can purchase digital tickets or check in via the mobile app.

**Data Storage**

The system will securely store eventrelated data in a MySQL database.

**Reporting** The system will generate attendance reports and analytics.

**Scalability** The system should be designed for scalability with cloudbased infrastructure as an option.

**Backup** Regular database backups for data integrity and disaster recovery.

**User Support** User support channels will be available for assistance.

**4.2 Design Representation**

**DATA FLOW DIAGRAM**

**ADMIN**

**STAFF**

**DATABASE**

**LOGIN MODEL**

**DATABASE**

**OUTPUT**

**LOGIN**

**STAFF LOGIN**

**DATABASE**

**OUTPUT**

**DATA OF THE STUDENTS**

**STAFF LOGIN**

**STAFF LOGIN**

**DATABASE**

**OUTPUT**

**DATA OF THE STUDENTS**

**STAFF LOGIN**

**4.3 Data Design**

**Data Design**

The system's data will be organized into the following tables

**Users Table** To store user registration information.

**Events Table** To store event details.

**Attendance Table** To store records of attendance.

**Reports Table** To store generated reports and analytics data.

**Notifications Table** To manage automated notifications.

**4.4 User Interface Design**

**User Interface Design**

The user interface will feature a clean and intuitive design with easy navigation.

Users can register or log in via the web application or mobile app.

Organizers can create and manage events through a userfriendly interface.

Participants can check in using a QR code scanner or NFC technology.

Spectators can access event information and check in via the mobile app.

The mobile app will have a responsive design for various device screen sizes.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**1. Introduction**

The implementation of an Online Sports Attendance Management System is a pivotal step in modernizing and optimizing the management of attendance at sports events. This comprehensive system aims to replace traditional manual methods with automated processes, ensuring accuracy, realtime monitoring, and enhanced communication. This document outlines the key steps and considerations for the successful implementation of the system.

**2. Project Planning**

The first phase of implementation involves detailed project planning. This includes defining the project's objectives, scope, and deliverables. A project plan is created, outlining timelines, milestones, and responsibilities. Clear communication with stakeholders, including event organizers, administrators, and potential system users, is essential to align expectations and ensure successful collaboration throughout the implementation process.

**3. Requirements Analysis**

Gathering detailed requirements is a crucial step in designing a system that meets the specific needs of its users. Workshops, interviews, and surveys are conducted to identify features, customization requirements, and integration needs. The goal is to ensure that the system addresses the unique challenges and preferences of various sports events and organizers.

**4. System Design**

Based on the gathered requirements, a detailed system design is created. This phase involves designing the database schema, user interfaces, and overall system architecture. User experience (UX) considerations are taken into account to create an intuitive and userfriendly interface. The system's scalability, security, and compatibility with modern technologies are also key factors in the design phase.

**5. Development**

The development phase involves the actual coding and programming of the Online Sports Attendance Management System. Modern programming languages and frameworks are employed to create a robust and efficient system. Continuous testing and quality assurance are integrated into the development process to identify and rectify issues promptly. The development team follows best practices to ensure the system's reliability and maintainability.

**6. Testing**

A comprehensive testing strategy is implemented to validate the functionality and performance of the system. This includes unit testing, integration testing, and system testing. The goal is to identify and address any bugs, errors, or performance issues before the system is deployed to the production environment. User acceptance testing (UAT) is also conducted to involve endusers in the validation process and gather feedback.

**7. Deployment**

Once testing is successful, the system is deployed in a staging or testing environment for final validation. Any remaining issues are addressed before the system is deployed to the production environment. A carefully planned deployment strategy is crucial to minimize downtime and ensure a smooth transition from the testing environment to live production.

**8. Training**

Training sessions are provided to system users, including event organizers, administrators, and other stakeholders. These sessions cover system functionalities, features, and best practices. Training materials, including user manuals and guides, are distributed to ensure that users are wellequipped to utilize the system effectively.

**9. Data Migration**

If applicable, existing attendance data from manual systems is migrated to the new online system. Data migration processes are executed with precision to ensure data integrity and accuracy. Backup mechanisms are in place to safeguard against data loss during the migration process.

**10. Integration**

The system is integrated with relevant external systems, such as ticketing platforms and scheduling tools. This integration ensures a seamless flow of data between systems, enhancing overall event management efficiency. Rigorous testing of these integrations is conducted to verify their reliability and accuracy.

**11. User Acceptance Testing (UAT)**

Selected endusers participate in UAT to validate that the system meets their needs and expectations. Feedback is collected, and any necessary adjustments are made based on user input. UAT is a crucial step in ensuring that the system aligns with the practical requirements of its intended users.

**12. Deployment to Production**

With successful UAT, the system is deployed to the production environment. The deployment process is closely monitored to identify and address any issues promptly. Communication with stakeholders is essential during this phase to manage expectations and provide support if needed.

**13. PostImplementation Support**

Ongoing support is provided to address any issues or questions that arise after the system is live. A dedicated support team is in place to respond to user inquiries, troubleshoot problems, and ensure the system's continued functionality. Regular maintenance activities, such as software updates and security patches, are carried out to keep the system optimized.

**14. Documentation**

Comprehensive documentation is created, including details about the system architecture, configurations, and user manuals. This documentation serves as a valuable resource for administrators, support teams, and future developers. Keeping documentation uptodate is crucial as the system evolves and additional features are introduced.

**15. Monitoring and Optimization**

Monitoring tools are implemented to track the system's performance, user interactions, and overall health. Continuous optimization efforts are undertaken based on user feedback and evolving requirements. Regular performance reviews and updates contribute to the longterm success and sustainability of the Online Sports Attendance Management System.

**CHAPTER 6**

**SYSTEM TESTING**

**6. System Testing**

The "System Testing" section evaluates the system's performance and functionality to ensure that it meets the requirements and functions as intended.

**6.1 Test Case Specifications**

In this subsection, you'll define the specific test cases and scenarios that will be used to evaluate the system. This includes positive and negative test cases.

**User Registration**

**Test Case 1** Verify that a new user can successfully register with valid information.

**Test Case 2** Test registration with missing or invalid data to ensure proper error handling.

**Event Creation**

**Test Case 3** Confirm that organizers can create events with accurate details.

**Test Case 4** Validate that event creation fails if essential details are missing.

**Attendance Tracking**

**Test Case 5** Check that participants can check in using QR codes and NFC tags.

**Test Case 6** Ensure that spectators can successfully purchase digital tickets or check in via the mobile app.

**Data Management**

**Test Case 7** Verify that the system accurately stores eventrelated data in the database.

**Test Case 8** Test database integrity by attempting to insert duplicate data.

**Reporting and Analytics**

**Test Case 9** Confirm that the system generates accurate attendance reports.

**Test Case 10** Ensure that analytics data reflects event participation trends.

**Security and Compliance**

**Test Case 11** Validate that user data is securely protected and complies with data protection regulations.

**Test Case 12** Attempt unauthorized access to test the system's security measures.

**6.2 Exception Specifications**

This subsection focuses on testing how the system handles exceptions, errors, and edge cases.

**User Registration**

**Exception 1** Verify that the system gracefully handles database connection errors during user registration.

**Exception 2** Confirm that the system provides userfriendly error messages for registration failures.

**Event Creation**

**Exception 3** Test how the system handles attempts to create events with invalid dates or times.

**Exception 4** Validate the response to unexpected server errors during event creation.

**Attendance Tracking**

**Exception 5** Test the system's response when a participant attempts to check in with an expired QR code.

**Exception 6** Verify the handling of connectivity issues during mobile app checkins.

**Data Management**

**Exception 7** Test the system's reaction to database connection failures during data storage.

**Exception 8** Confirm that error logs are generated and sent to administrators for databaserelated issues.

**Security and Compliance**

**Exception 9** Check the system's response to multiple failed login attempts, testing the account lockout mechanism.

**Exception 10** Test the system's behavior when a user attempts to access unauthorized data.

**CHAPTER 7**

**INSTALLATION**

**a. Hardware Requirements**

The hardware requirements for the installation of the Online Sports Attendance Management System are as follows

**1. Server**

Multicore processor (at least quadcore) for optimal performance.

Minimum 8 GB RAM, with scalability based on the expected user load.

Sufficient storage space for database storage and system files (at least 100 GB).

**2. Database Server**

Dedicated server for hosting the database.

Minimum 8 GB RAM for efficient database operations.

Fast and reliable storage, preferably using SSDs.

**3. Network Infrastructure**

Highspeed internet connection for seamless data transfer.

Network switches and routers to facilitate communication between servers and clients.

Firewall for security and protection against unauthorized access.

**4. Client Devices**

Desktops, laptops, or tablets with modern web browsers for accessing the system.

Internet connectivity for realtime communication with the server.

**b. Software Requirements**

The software requirements for the Online Sports Attendance Management System include both serverside and clientside components

**ServerSide**

**1. Operating System**

Linuxbased server OS (e.g., Ubuntu Server, CentOS) or Windows Server 2016 and above.

**2. Web Server**

Apache HTTP Server or Nginx for hosting the web application.

**3. Database Management System**

MySQL or PostgreSQL for storing attendance data.

**4. Programming Language**

PHP for serverside scripting.

**5. Authentication and Authorization**

Implement a secure authentication mechanism using protocols like OAuth or JWT.

Rolebased access control for different user types (organizers, administrators, participants).

**6. Security**

SSL/TLS certificates for securing data transmission.

Firewall configuration to control incoming and outgoing traffic.

Regular security updates and patches.

**ClientSide**

**1. Web Browser**

Modern web browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge.

**2. Internet Connectivity**

Stable internet connection with sufficient bandwidth for accessing the system.

**3. User Interface Framework**

Utilize a responsive design framework (e.g., Bootstrap) for a consistent user experience across devices.

**4. Communication Protocols**

Implement WebSocket or similar protocols for realtime updates.

**5. Mobile Application (Optional)**

If a mobile application is part of the system, specify the platform (iOS, Android) and required components.

**c. Installation Steps**

**1. Server Setup**

Install the chosen operating system on the server.

Configure network settings and ensure internet connectivity.

Install and configure the web server (Apache or Nginx).

Set up the database server (MySQL or PostgreSQL).

**2. Application Deployment**

Transfer the application files to the server using secure methods.

Configure the application settings, including database connections and security parameters.

**3. Database Initialization**

Create the necessary database schema and tables.

Populate initial data, if required, such as default user roles or system settings.

**4. ServerSide Dependencies**

Install and configure Node.js or Python.

Install required npm packages or Python libraries.

Set up environment variables for sensitive information (database credentials, API keys).

**5. Web Application Configuration**

Configure application settings such as system title, logo, and default settings.

Implement user authentication and authorization mechanisms.

**6. Security Measures**

Implement SSL/TLS certificates for secure data transmission.

Configure firewall rules to allow necessary traffic and block unauthorized access.

**7. Client Access**

Provide users with the URL or IP address to access the system.

Instruct users on browser compatibility and system requirements.

**8. Testing**

Conduct thorough testing to ensure the system functions correctly.

Perform user acceptance testing to validate the user interface and experience.

**9. Documentation**

Prepare comprehensive installation and configuration documentation for future reference.

Include troubleshooting steps and contact information for support.

**10. Monitoring and Maintenance**

Implement monitoring tools for server and application performance.

Establish a regular maintenance schedule for security updates and system optimizations.

**CHAPTER 8**

**MAINTANANCE**

**1. Regular Maintenance Tasks**

**a. Security Updates**

Regularly update the operating system, web server, and database management system to patch security vulnerabilities.

Monitor security advisories for dependencies and thirdparty libraries used in the system.

Conduct periodic security audits to identify and address potential risks.

**b. Backup Procedures**

Implement regular backup procedures for both the application files and the database.

Store backups in a secure location and test the restoration process periodically.

Automate backup tasks to ensure consistency and reliability.

**c. Database Maintenance**

Optimize database performance by regularly analyzing and optimizing tables.

Monitor database storage usage and plan for scalability as needed.

Conduct routine checks for data integrity and resolve any issues promptly.

**d. System Logs**

Maintain detailed system logs for monitoring application behavior and identifying potential issues.

Regularly review logs to spot anomalies, errors, or suspicious activities.

Implement log rotation to manage log file sizes effectively.

**e. User Management**

Periodically review user roles and permissions to ensure they align with organizational needs.

Deactivate or remove inactive user accounts to maintain security.

Implement password policies and encourage users to update passwords regularly.

**2. Updates and Enhancements**

**a. System Updates**

Stay informed about updates to the application framework, libraries, and dependencies.

Test updates in a staging environment before applying them to the production system.

Plan and schedule updates during lowtraffic periods to minimize disruption.

**b. Feature Enhancements**

Gather feedback from users to identify areas for improvement and new features.

Prioritize enhancements based on user needs and organizational goals.

Plan and implement feature updates in a phased approach to maintain system stability.

**3. Troubleshooting Procedures**

**a. Incident Response**

Establish an incident response plan to address system outages or critical issues promptly.

Define escalation procedures and communication channels during incidents.

Conduct postincident reviews to identify root causes and implement preventive measures.

**b. User Support**

Provide a dedicated support channel for users to report issues or seek assistance.

Maintain a knowledge base with FAQs and troubleshooting guides.

Regularly review and address userreported issues to enhance user satisfaction.

**c. Performance Monitoring**

Implement monitoring tools to track system performance, resource usage, and response times.

Set up alerts for abnormal behavior or performance degradation.

Conduct regular performance reviews and optimization based on monitoring data.

**4. Documentation**

**a. System Documentation**

Keep system documentation uptodate with any changes to configurations, settings, or procedures.

Include information on how to perform routine maintenance tasks.

Document any modifications or customizations made to the system.

**b. Troubleshooting Guides**

Maintain detailed troubleshooting guides for common issues.

Include stepbystep instructions and potential solutions for identified problems.

Provide contact information for users to seek additional support.

**c. Change Log**

Maintain a change log to record updates, enhancements, and bug fixes.

Include details such as the date of the change, the nature of the modification, and the individuals involved.

**5. Training and Knowledge Transfer**

**a. Training Sessions**

Conduct periodic training sessions for system administrators and support staff.

Keep users informed about new features, updates, and best practices.

Provide refresher courses to ensure ongoing proficiency.

**b. Knowledge Transfer**

Ensure that knowledge about the system is documented and shared among team members.

Crosstrain staff members to handle various aspects of system maintenance.

Plan for knowledge transfer during team transitions or personnel changes.

**CHAPTER 10**

**CONCULSION**

**Conclusion**

This innovative system aims to enhance the experience for participants, organizers, and spectators by providing a userfriendly, efficient, and secure platform for managing sports event attendance.

**Key Achievements**

**Efficient Attendance Tracking** The system offers efficient and accurate attendance tracking for both participants and spectators, utilizing QR codes, NFC technology, and a mobile app for seamless checkins.

**Realtime Updates** The system ensures that attendance data is updated in realtime, empowering organizers with the ability to make informed decisions and plan events effectively.

**UserFriendly Interface** The web application and mobile app offer a userfriendly interface, simplifying registration, event creation, and attendance monitoring.

**Data Management** The system securely stores and manages all eventrelated data in a MySQL database, ensuring data integrity and accessibility for authorized users.

**Reporting and Analytics** Organizers and administrators can generate attendance reports and analytics, providing insights into participation trends and areas for improvement.

**Security and Compliance** The system adheres to data protection regulations and employs robust encryption techniques to safeguard user data.

**BIBLIOGRAPHY**

\

**8. Bibliography**

**1.** Smith, John. (2022). "Modern Web Application Development with React and Node.js." Publisher O'Reilly Media.

**2.** Brown, Emily. (2021). "Database Management and Design." Publisher Pearson.

**3.** Johnson, Mark. (2019). "Mobile App Development with React Native." Publisher Appriss.

**4.** MySQL Documentation. (Online). Available at https//dev.mysql.com/doc/

**5.** React.js Documentation. (Online). Available at https//reactjs.org/docs/gettingstarted.html

**6.** Node.js Documentation. (Online). Available at https//nodejs.org/en/docs/

**7.** "Data Protection Regulations and Compliance Guidelines." (Online). Available at [Provide the specific source or guideline used for data protection compliance].

WEBSITES

1. [https//www.w3schools.com/php/](https://www.w3schools.com/php/)
2. [https//www.php.net/manual/en/](https://www.php.net/manual/en/)
3. [https//www.tutorialspoint.com/mysql/index.htm](https://www.tutorialspoint.com/mysql/index.htm)
4. [https//www.sitepoint.com/php/](https://www.sitepoint.com/php/)
5. [https//www.geeksforgeeks.org/phpprogramminglanguage/](https://www.geeksforgeeks.org/php-programming-language/)
6. [https//dev.mysql.com/doc/](https://dev.mysql.com/doc/)
7. [https//github.com/php](https://github.com/php)
8. [https//github.com/mysql](https://github.com/mysql)

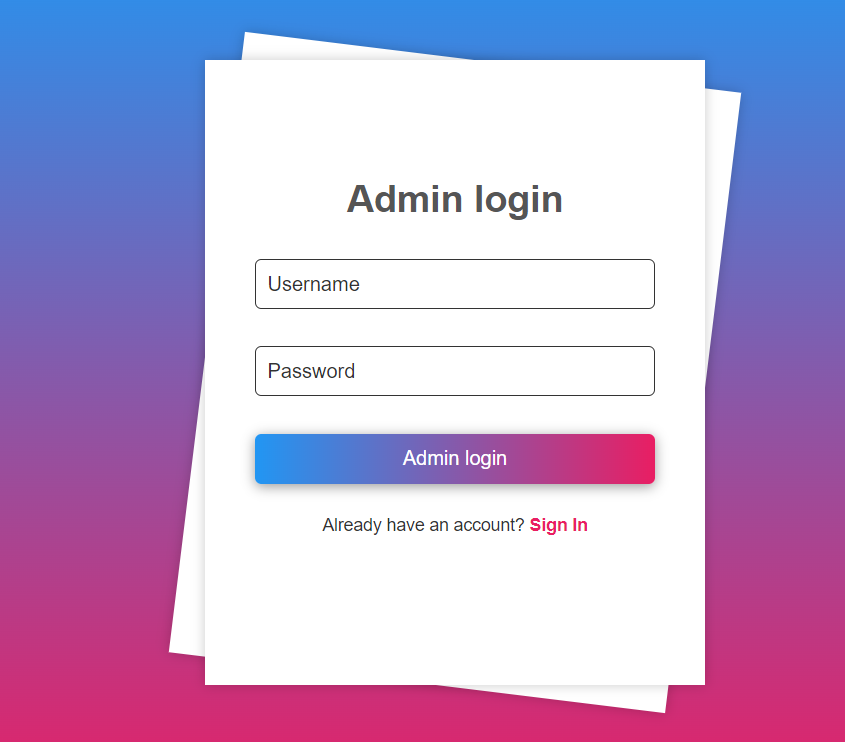
**REFERENCE BOOKS**

* "PHP and MySQL Web Development" by Luke Welling and Laura Thomson (5th Edition)
* "Learning PHP, MySQL & JavaScript" by Robin Nixon (4th Edition)
* "PHP The Complete Reference" by Steven Holzner (1st Edition)
* "HTML and CSS Design and Build Websites" by Jon Duckett (1st Edition)
* "HTML and CSS The Good Parts" by Ben Henick (1st Edition)
* "CSS The Definitive Guide" by Eric A. Meyer and Estelle Weyl (4th Edition)
* "XAMPP Tutorial How to Use XAMPP for Local Development" by H. Lennartsson (1st Edition)
* "PHP and MySQL for Dynamic Web Sites" by Larry Ullman (5th Edition)
* "PHP Solutions Dynamic Web Design Made Easy" by David Powers (4th Edition)
* "HTML and CSS Visual QuickStart Guide" by Elizabeth Castro and Bruce Hyslop (8th Edition)
* "CSS The Missing Manual" by David McFarland (4th Edition)
* "Web Design with HTML, CSS, JavaScript and jQuery Set" by Jon Duckett (1st Edition)
* "XAMPP Basic to Advanced" by Dr. Harry Raj (1st Edition)
* "XAMPP An Easy Guide for Beginners" by Vincent V. Twomey (1st Edition)

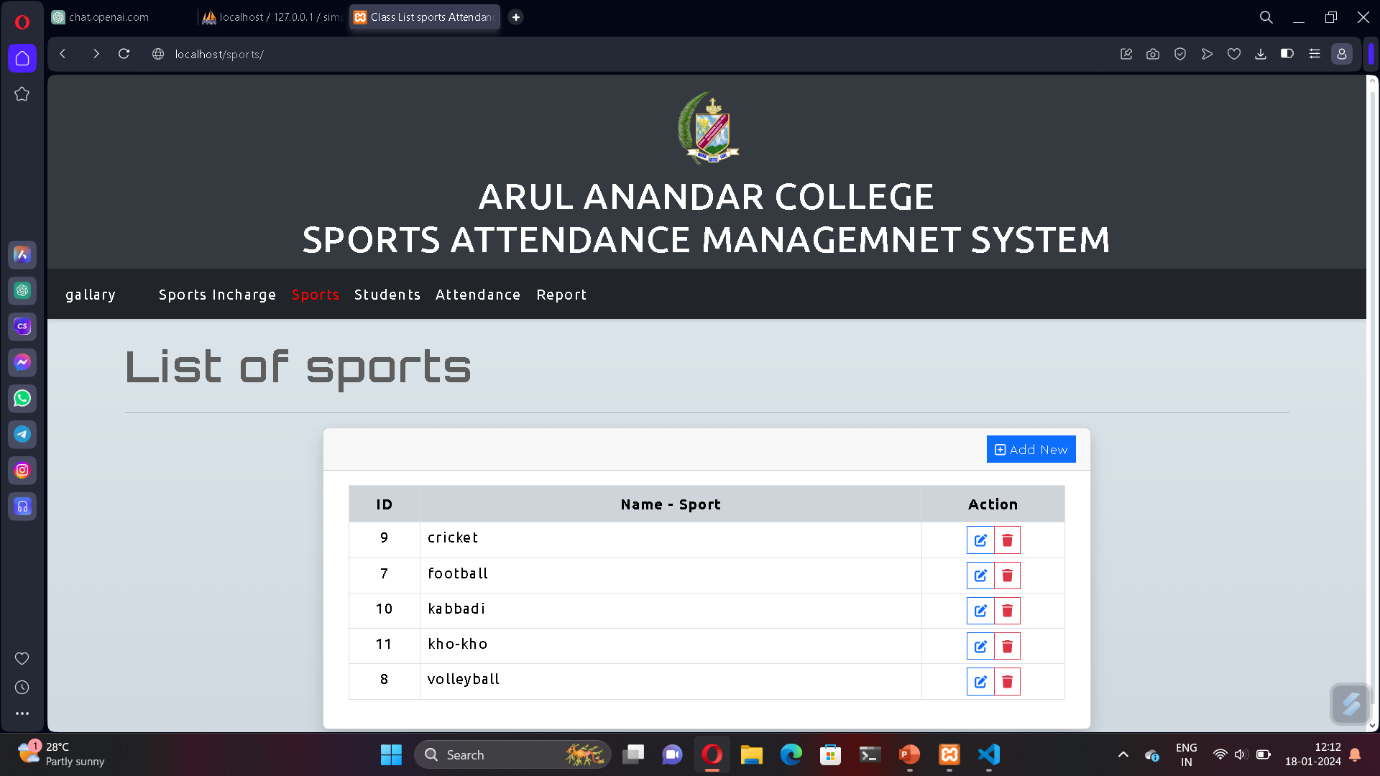
**APPENDICES**

1. **Screen Shots of User Interfaces**

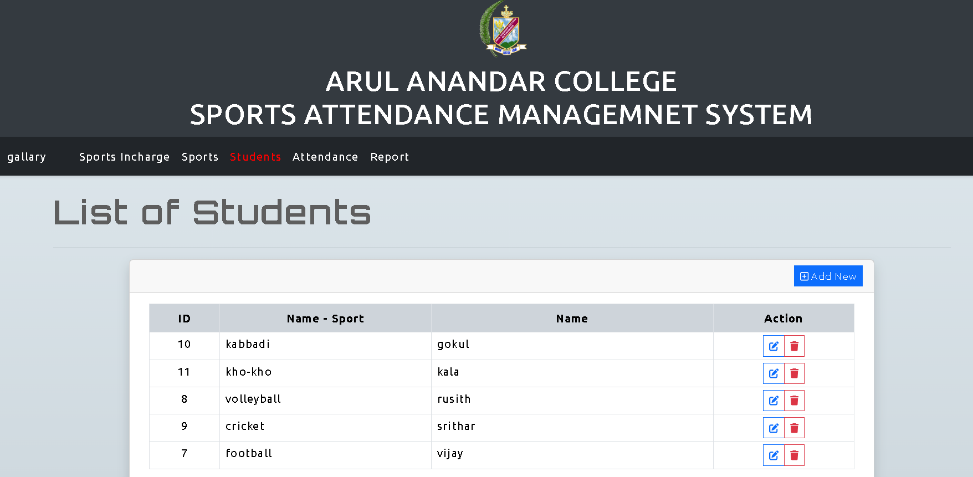
**LOGIN PAGE**

****

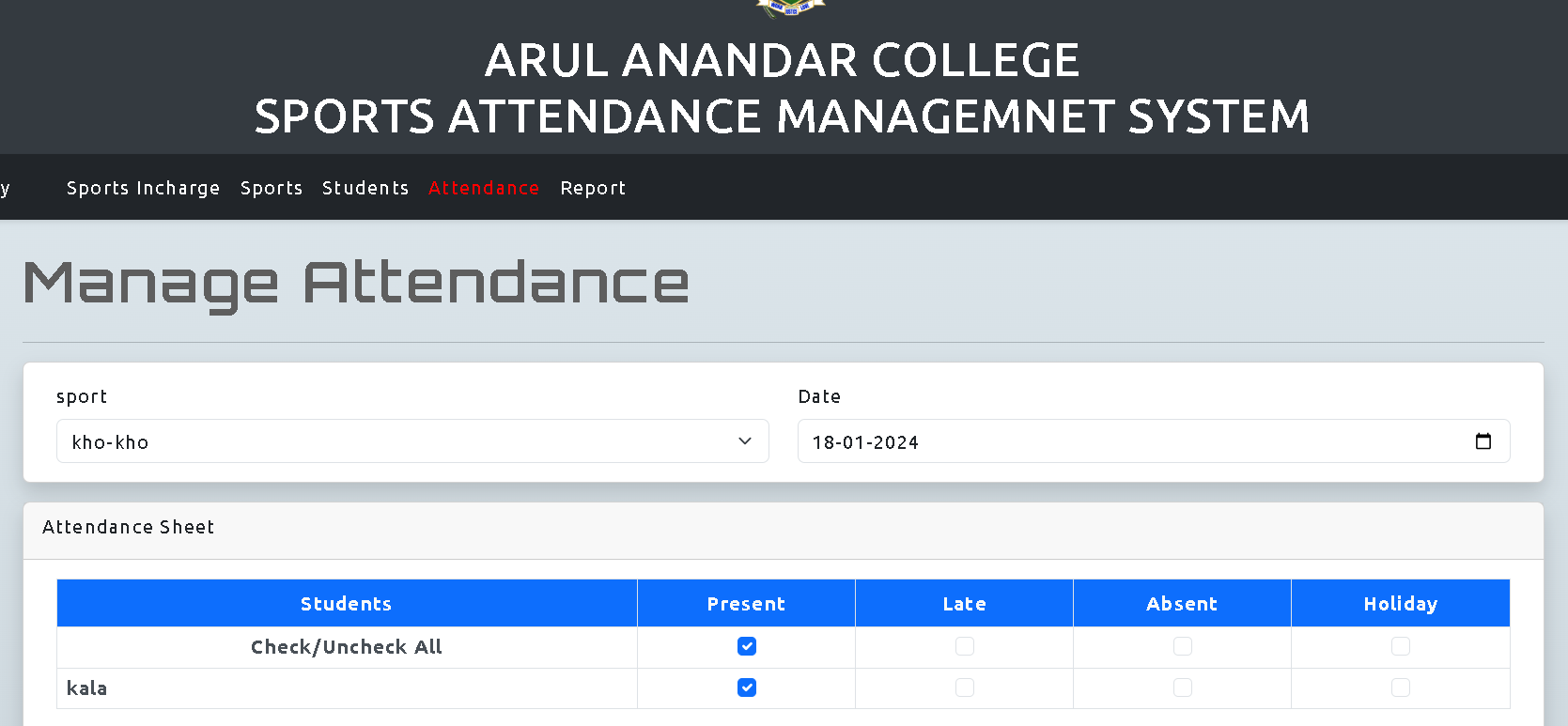
**SPORTS UPLOAD**

****

**STUDENTS DETAIL UPLOAD**

****

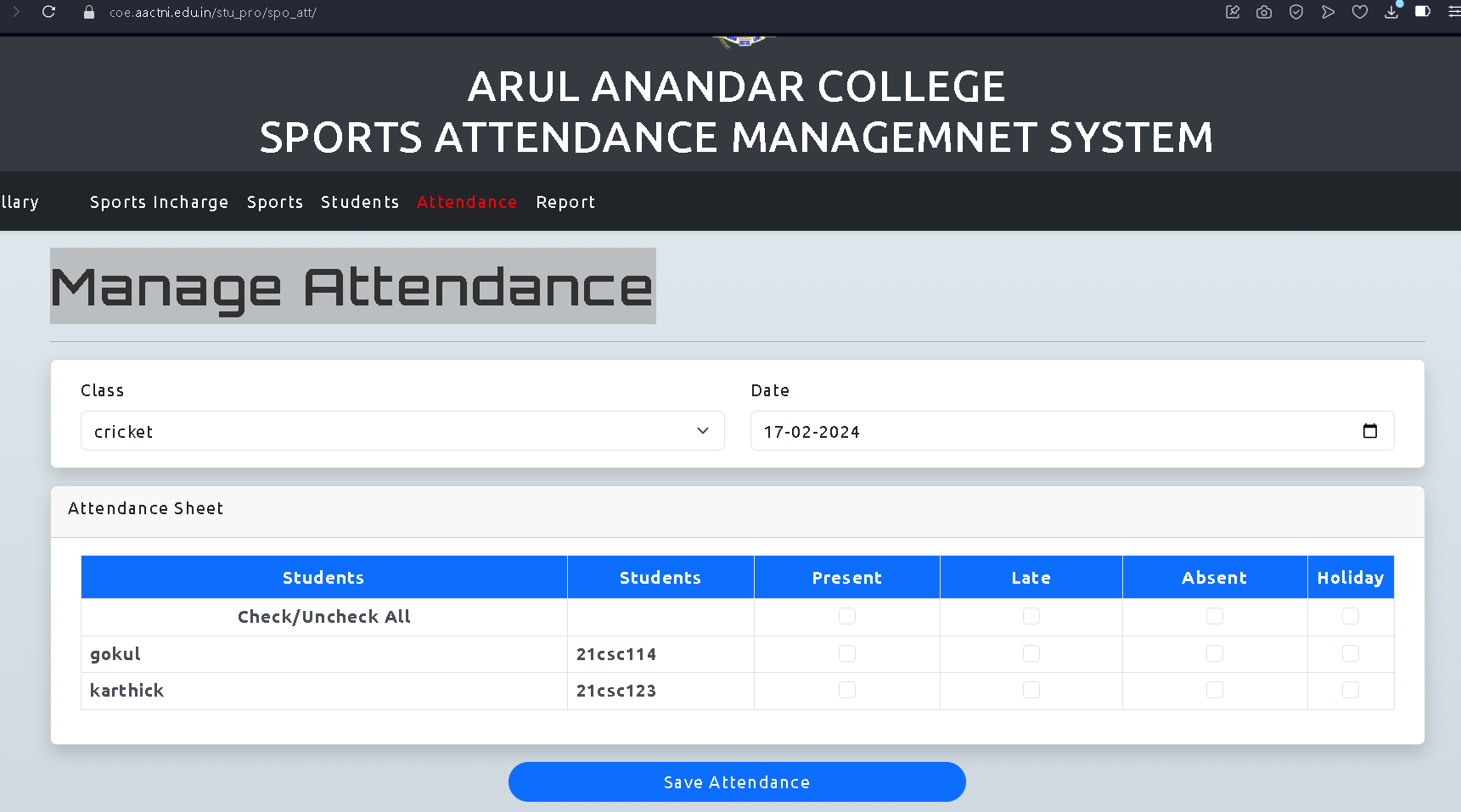
**MANAGE ATTENDANCE**

****

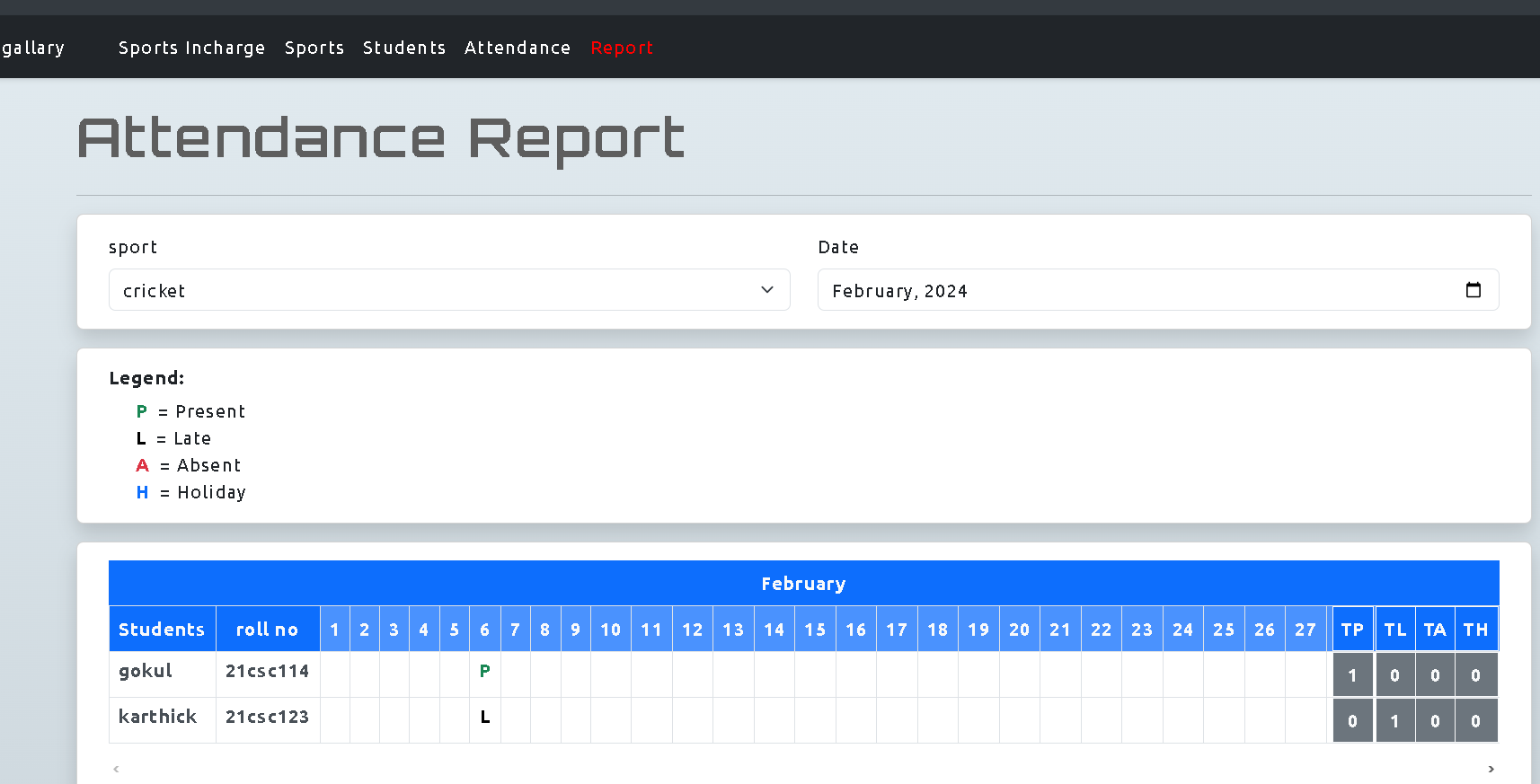
**FACILITY**

****

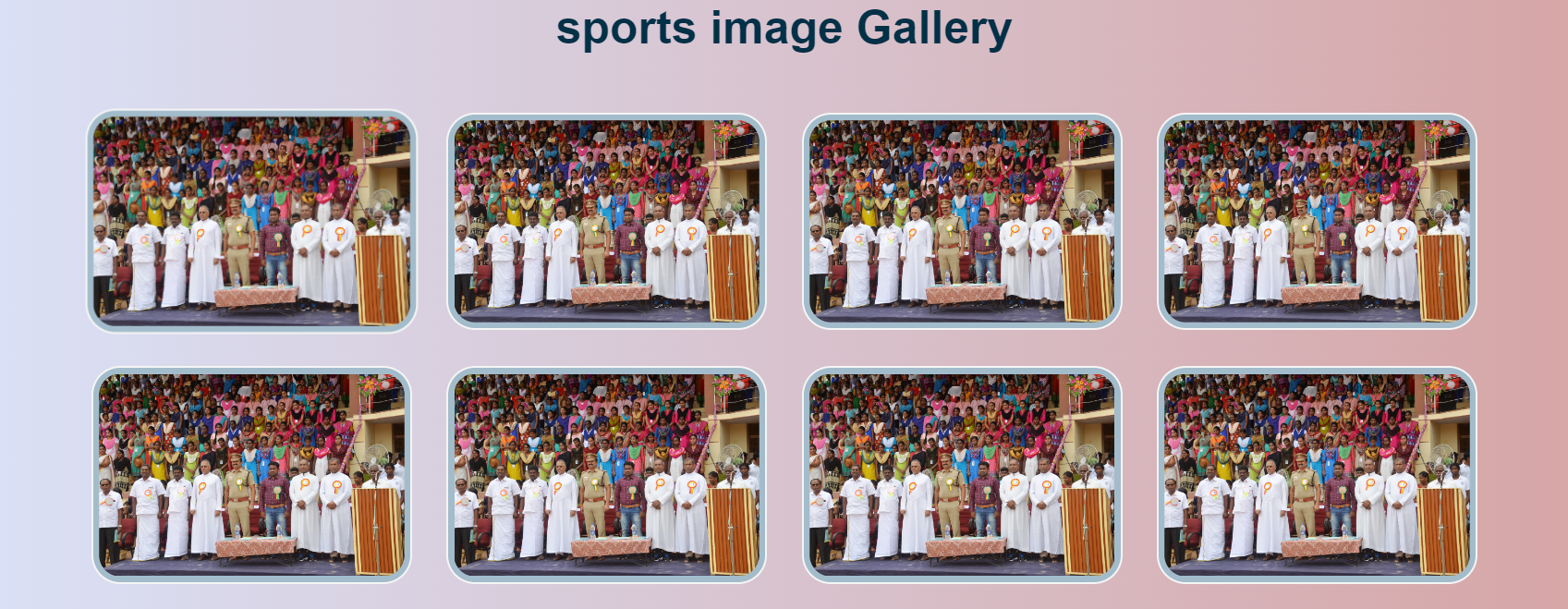
**SPORTS ATTENDANCE SELECT**

****

**ATTENDANCE REPORT**

****

**SPORTS GALLERY**



**Source Code Listing**

**INDEX SOURCE CODE**

<?php

class Actions{

private $conn;

function \_\_construct(){

require\_once(realpath(\_\_DIR\_\_.'/../dbconnect.php'));

$this>conn = $conn;

}

/\*\*

\* Class Actions

\*/

public function save\_class(){

foreach($\_POST as $k => $v){

if(!is\_array($\_POST[$k]) && !is\_numeric($\_POST[$k]) && !empty($\_POST[$k])){

$\_POST[$k] = addslashes(htmlspecialchars($v));

}

}

extract($\_POST);

if(!empty($id)){

$check = $this>conn>query("SELECT id FROM `class\_tbl` where `name` = '{$name}' and `id` != '{$id}' ");

$sql = "UPDATE `class\_tbl` set `name` = '{$name}' where `id` = '{$id}'";

}else{

$check = $this>conn>query("SELECT id FROM `class\_tbl` where `name` = '{$name}' ");

$sql = "INSERT `class\_tbl` set `name` = '{$name}'";

}

if($check>num\_rows > 0){

return ['status' => 'error', 'msg' => 'Class Name Already Exists!'];

}else{

$qry = $this>conn>query($sql);

if($qry){

if(empty($id)){

$\_SESSION['flashdata'] = [ 'type' => 'success', 'msg' => "New Class has been added successfully!" ];

}else{

$\_SESSION['flashdata'] = [ 'type' => 'success', 'msg' => "Class Data has been updated successfully!" ];

}

return [ 'status' => 'success'];

}else{

if(empty($id)){

return ['status' => 'error', 'msg' => 'An error occurred while saving the New Class!'];

}else{

return ['status' => 'error', 'msg' => 'An error occurred while updating the Class Data!'];

}

}

}

}

public function delete\_class(){

extract($\_POST);

$delete = $this>conn>query("DELETE FROM `class\_tbl` where `id` = '{$id}'");

if($delete){

$\_SESSION['flashdata'] = [ 'type' => 'success', 'msg' => "Class has been deleted successfully!" ];

return [ "status" => "success" ];

}else{

$\_SESSION['flashdata'] = [ 'type' => 'danger', 'msg' => "Class has failed to deleted due to unknown reason!" ];

return [ "status" => "error", "Class has failed to deleted!" ];

}

}

public function list\_class(){

$sql = "SELECT \* FROM `class\_tbl` order by `name` ASC";

$qry = $this>conn>query($sql);

return $qry>fetch\_all(MYSQLI\_ASSOC);

}

public function get\_class($id=""){

$sql = "SELECT \* FROM `class\_tbl` where `id` = '{$id}'";

$qry = $this>conn>query($sql);

$result = $qry>fetch\_assoc();

return $result;

}

/\*\*

\* Student Actions

\*/

public function save\_student(){

foreach($\_POST as $k => $v){

if(!is\_array($\_POST[$k]) && !is\_numeric($\_POST[$k]) && !empty($\_POST[$k])){

$\_POST[$k] = addslashes(htmlspecialchars($v));

}

}

extract($\_POST);

if(!empty($id)){

$check = $this>conn>query("SELECT id FROM `students\_tbl` where `name` = '{$name}' and `class\_id` = '{$class\_id}' and `id` != '{$id}' ");

$sql = "UPDATE `students\_tbl` set `name` = '{$name}', `class\_id` = '{$class\_id}' where `id` = '{$id}'";

}else{

$check = $this>conn>query("SELECT id FROM `students\_tbl` where `name` = '{$name}' and `class\_id` = '{$class\_id}' ");

$sql = "INSERT `students\_tbl` set `name` = '{$name}', `class\_id` = '{$class\_id}', `rollno` = '{$rollno}'";

}

if($check>num\_rows > 0){

return ['status' => 'error', 'msg' => 'Student Name Already Exists!'];

}else{

$qry = $this>conn>query($sql);

if($qry){

if(empty($id)){

$\_SESSION['flashdata'] = [ 'type' => 'success', 'msg' => "New Student has been added successfully!" ];

}else{

$\_SESSION['flashdata'] = [ 'type' => 'success', 'msg' => "Student Data has been updated successfully!" ];

}

return [ 'status' => 'success'];

}else{

if(empty($id)){

return ['status' => 'error', 'msg' => 'An error occurred while saving the New Class!'];

}else{

return ['status' => 'error', 'msg' => 'An error occurred while updating the Student Data!'];

}

}

}

}

public function delete\_student(){

extract($\_POST);

$delete = $this>conn>query("DELETE FROM `students\_tbl` where `id` = '{$id}'");

if($delete){

$\_SESSION['flashdata'] = [ 'type' => 'success', 'msg' => "Student has been deleted successfully!" ];

return [ "status" => "success" ];

}else{

$\_SESSION['flashdata'] = [ 'type' => 'danger', 'msg' => "Student has failed to deleted due to unknown reason!" ];

return [ "status" => "error", "Student has failed to deleted!" ];

}

}

public function list\_student(){

$sql = "SELECT `students\_tbl`.\*, `class\_tbl`.`name` as `class` FROM `students\_tbl` inner join `class\_tbl` on `students\_tbl`.`class\_id` = `class\_tbl`.`id` order by `students\_tbl`.`name` ASC";

$qry = $this>conn>query($sql);

return $qry>fetch\_all(MYSQLI\_ASSOC);

}

public function get\_student($id=""){

$sql = "SELECT `students\_tbl`.\*, `class\_tbl`.`name` as `class` FROM `students\_tbl` inner join `class\_tbl` on `students\_tbl`.`class\_id` = `class\_tbl`.`id` where `students\_tbl`.`id` = '{$id}'";

$qry = $this>conn>query($sql);

$result = $qry>fetch\_assoc();

return $result;

}

public function attendanceStudents($class\_id = "", $class\_date = ""){

if(empty($class\_id) || empty($class\_date))

return [];

$sql = "SELECT `students\_tbl`.\*, COALESCE((SELECT `status` FROM `attendance\_tbl` where `student\_id` = `students\_tbl`.id and `class\_date` = '{$class\_date}' ), 0) as `status` FROM `students\_tbl` where `class\_id` = '{$class\_id}' order by `name` ASC";

$qry = $this>conn>query($sql);

$result = $qry>fetch\_all(MYSQLI\_ASSOC);

return $result;

}

public function attendanceStudentsMonthly($class\_id = "", $class\_month = ""){

if(empty($class\_id) || empty($class\_month))

return [];

$sql = "SELECT `students\_tbl`.\* FROM `students\_tbl` where `class\_id` = '{$class\_id}' order by `name` ASC";

$qry = $this>conn>query($sql);

$result = $qry>fetch\_all(MYSQLI\_ASSOC);

foreach($result as $k => $row){

$att\_sql = "SELECT `status`, `class\_date` FROM `attendance\_tbl` where `student\_id` = '{$row['id']}' ";

$att\_qry = $this>conn>query($att\_sql);

foreach($att\_qry as $att\_row){

$result[$k]['attendance'][$att\_row['class\_date']] = $att\_row['status'];

}

}

return $result;

}

public function save\_attendance(){

extract($\_POST);

$sql\_values = "";

$errors = "";

foreach($student\_id as $k => $sid){

$stat = $status[$k] ?? 3;

$check = $this>conn>query("SELECT id FROM `attendance\_tbl` where `student\_id` = '{$sid}' and `class\_date` = '{$class\_date}'");

if($check>num\_rows > 0){

$result = $check>fetch\_assoc();

$att\_id = $result['id'];

try{

$update = $this>conn>query("UPDATE `attendance\_tbl` set `status` = '{$stat}' where `id` = '{$att\_id}'");

}catch(Exception $e){

if(!empty($errors)) $errors .= "<br>";

$errors .= $e>getMessage();

}

}else{

if(!empty($sql\_values)) $sql\_values .= ", ";

$sql\_values .= "( '{$sid}', '{$class\_date}', '{$stat}' )";

}

}

if(!empty($sql\_values))

{

try{

$sql = $this>conn>query("INSERT INTO `attendance\_tbl` ( `student\_id`, `class\_date`, `status` ) VALUES {$sql\_values}");

}catch(Exception $e){

if(!empty($errors)) $errors .= "<br>";

$errors .= $e>getMessage();

}

}

if(empty($errors)){

$resp['status'] = "success";

$\_SESSION['flashdata'] = [ "type" => "success", "msg" => "Class Attendance Data has been saved successfully." ];

}else{

$resp['status'] = "error";

$resp['msg'] = $errors;

}

return $resp;

}

function \_\_destruct()

{

if($this>conn)

$this>conn>close();

}

}