SPORTS ACADEMIX

PROJECT REPORT SUBMITTED TO MAHATMA GANDHI UNIVERSITY, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER APPLICATIONS

BY

ADITHYE K.S

Register Number: 230021078054

DEPT. OF COMPUTER APPLICATIONS

BISHOP VAYALIL MEMORIAL

HOLY CROSS COLLEGE

CHERPUNKAL, KOTTAYAM 686584

October 2025

Department of Computer Applications

BISHOP VAYALIL MEMORIAL

HOLY CROSS COLLEGE

CERTIFICATE

Certified that the report entitled **"SPORTS ACADEMIX"** is the bona fide record of the project work done by **Adithye K.S (Reg. No. 230021078054)** under our guidance and supervision and is submitted in partial fulfillment of the Bachelor degree in Computer Applications, awarded by Mahatma Gandhi University, Kerala.

Ms.Nova Emmanuel
Project Guide

Ms. Seena S Nair
Head of the Department

Dr. Baby Sebastian
Principal

Submitted for Project Evaluation on ------/-----/------

Internal Examiner

External Examiner

DECLARATION

I hereby declare that the project work entitled "**SPORTS ACADEMIX"** submitted in partial fulfillment of the requirements for the award of the Bachelor of degree in Computer Applications from BVM Holy Cross College, Cherpunkal, is record of bona fide work done under the guidance of Ms. Jaise Jose (Assistant Professor, Dept. of Computer Applications)

Date: Name: Adithye K.S

Place: Cherpunkal Reg No. 2300210780540

ACKNOWLEDGEMENT

Dedicating this project to the Almighty God, whose abundant grace and mercies enabled its successful completion, we would like to express our profound gratitude to all the people who inspired and motivated us to make this project a success. We express our heartfelt thanks to our Principal, Rev. Dr. Baby Sebastian, for his warm support and valuable suggestions in accomplishing the project. We would also like to place our deep sense of gratitude to Ms. Seena S Nair (Head of the Department of Computer Applications) for her guidance in carrying out this project work. We are profoundly grateful to Ms.Nova Emmanuel (Assistant Professor, Department of Computer Applications), our Project Guide, for her valuable guidance, suggestions, and assessment throughout the project. We also extend our sincere thanks to all other faculty members of the Department of Computer Applications for their assistance and encouragement. Last but not least, we would like to thank our friends for their cooperation and support throughout the project.

ABSTRACT

Project Title: **Sports Academix**: A Web-Based Management System for Institutional Sports

Managing sports activities in educational institutions often faces challenges due to manual processes, fragmented communication, and inefficient scheduling. Sports Academix addresses these issues by introducing a centralized, web-based platform that streamlines all aspects of sports administration.

The system adopts a dual-perspective architecture, offering administrators a robust dashboard for managing users, departments, sports, teams, and events, including real-time scheduling, score updates, and result finalization. It also automates student registrations and integrates user feedback management for responsive governance.

For students, the platform acts as an interactive hub, providing access to sports information, match schedules, results, and point standings. Students can easily register for activities through their departments, view personalized profiles showcasing team memberships and match history, and receive real-time notifications for key updates.

Technically, the solution is powered by a PHP backend with MySQL for persistent storage, while the frontend leverages HTML, CSS, and JavaScript. The administrator dashboard functions as a dynamic single-page application using asynchronous API communication to deliver a seamless, responsive experience.

By automating key tasks and centralizing communication, Sports Academix promotes efficiency, transparency, and engagement, fostering a more vibrant sports culture within educational institution.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The **Sports Academix** project is a web-based application developed to modernize the management of sports activities within an academic institution. Its primary objective is to replace the traditional manual, paper-driven processes with a centralized and fully digital solution.

Built using a full-stack approach—PHP and MySQL for the backend and HTML, CSS, and JavaScript for the frontend—the system ensures a clear separation of concerns, scalability, and ease of maintenance.

Key capabilities include:

- **Role-Specific Dashboards:** Distinct interfaces for administrators and general users (students/players), providing each group with tools and information relevant to their responsibilities.

- **Centralized Data Management:** Unified storage of player profiles, team information, events, and statistics for quick retrieval and streamlined reporting.

- **Automated Operations:** Tasks such as player registration, event scheduling, and score updates are handled automatically, reducing manual effort and errors.

- **Interactive Features:** Facilities for match comments, result appeals, and direct feedback strengthen communication between users and administrators.

- **Security and Accountability:** Password hashing, prepared SQL statements, and an administrator activity log ensure data integrity and provide a transparent audit trail.

By integrating these features, Sports Academix demonstrates practical application of software-engineering principles—from requirements analysis to testing—while delivering a robust, user-friendly platform. The modular architecture also positions the system for future enhancements, such as mobile app integration or expanded collaboration tools.

1.2 Organization Profile

Organization Name: BVM Holy Cross College

Location: Cherpunkal, Kottayam, Kerala,

India Year of Establishment: 1995

Type of Institution: Self-financing college

Affiliation: Mahatma Gandhi University, Kottayam

Management: Holy Cross Forane Church, Cherpunkal

Vision: To be a leading institution of higher education that empowers students to become responsible citizens and ethical professionals.

Mission: To provide quality education that promotes academic excellence, holistic development, and social commitment.

Core Values:

• Academic Integrity

• Student - Centric Approach

• Social Responsibility

• Inclusiveness

• Excellence in All Endeavors

Contact Information:

• BVM Holy Cross College

• Cherpunkal, Kottayam, Kerala - 686529

• Phone: 04822 - 267520, 9446640157

• Email: bvmhcc@gmail.com

• Website: https://bvmcollege.com/

2. System Configuration

2.1 Hardware Specification

- CPU: Intel i3 or equivalent
- RAM: 4 GB (8 GB recommended)
- Storage: 20 GB free

2.2 Software Specification

- OS: Windows / Linux / macOS
- Web Server: Apache (XAMPP/WAMP/LAMP) or Nginx + PHP-FPM
- PHP: 7.4 / 8.x
- Database: MySQL
- Frontend: HTML5, CSS3, JavaScript (ES6)
- Tools Used: VS Code, phpMyAdmin

3.System Analysis

3.1.Preliminary Investigation

The preliminary investigation is a problem-solving activity that requires intensive communication between the system users and the system developers. It does various feasibility studies. In those studies, a rough figure of the system activities can be obtained from which the decisions about the strategies to be followed for effective system study and analysis can be taken. In the preliminary investigation an initial picture about the system working is obtained from the information obtained from the study, the data collection methods were identified. Right from the investigation about the system many existing drawbacks of the system could be identified, which helped a lot in the later stages of more rigorous study and analysis of the manual system. The most critical phase of managing system projects is planning. To launch a system investigation, we need a master plan detailing the steps to be taken, the people to be questioned and the outcome expected. The scope of preliminary investigation may vary from a brief one-person effort to an extensive series of activities requiring the participation of many individuals. Here in the project, a detailed study of the existing system is carried along with all the steps in the system analysis. An idea for creating a better project was carried and the next steps were followed.

3.2. Existing System

- Manual Processes: The entire system is manual and paper-based.
- Registration: Students register using physical forms, which are then manually processed and stored by staff.
- Scheduling & Results: Schedules and match results are communicated via physical notice boards.
- Inefficient Communication: Updates and announcements are handled through word-of-mouth or physical notices, causing delays and missed information.
- Lack of Transparency: There is no automated system to track the status of requests like feedback or appeals, making the process inefficient.

3.2.1. Disadvantages of Existing System:

- Inefficiency: Manual data entry and processing are time-consuming and prone to human error, which can lead to mistakes in registration or scorekeeping.

- Lack of Centralization: Information is scattered across various physical notice boards and paper forms, making it difficult to access or manage it from a single location.

- Poor Communication: Updates and announcements rely on physical notices and word-of-mouth, which can cause delays and lead to players missing important information about schedules or results

- Lack of Transparency: Without a digital record, there's no clear audit trail for administrative actions, match results, or feedback. This can lead to a lack of accountability.

- Accessibility Issues: Information is not accessible to users remotely, restricting their ability to stay informed about sports activities when they are not on campus.

3.3. Proposed System

The proposed system, named **Sports Academix**, is a comprehensive web-based platform designed to provide a structured and efficient approach to managing sports activities within an academic institution. Unlike the existing manual approach where tasks were tracked through physical forms and notice boards, Sports Academix introduces a centralized digital platform where users can create, manage, and monitor sports events in a systematic way. The system focuses on security, usability, and accessibility, ensuring that both technical and non-technical users can easily adapt to it.

Sports Academix is built as a web-based application, which means it can be accessed through any modern browser without the need for complex installations. At its core lies the event and user

management modules. Users can register themselves in the system, log in securely, and manage their personal accounts. Once logged in, they can view upcoming schedules, recent match results, and team standings. In addition to general users, the system also features a powerful administrator dashboard. This dual-interface design allows for centralized control over all system functions.

The administrator dashboard is another strong feature. It allows for the creation and management of sports, teams, and events. Admins can update scores, which automatically finalizes a match, and can review pending registrations from students. The system also includes modules for managing user feedback and handling match appeals, which are formal requests to dispute a match's result. This ensures transparency and a clear audit trail for all activities.

From a technical perspective, Sports Academix is powered by **PHP**, which handles user authentication and all data operations. Data is stored and managed through backend scripts that interact with the system's database (MySQL), ensuring data persistence. This approach eliminates the risk of losing important details, reduces errors caused by miscommunication, and saves time by providing an organized view of all sports activities. Furthermore, Sports Academix enhances accountability, as all administrative actions are logged in an audit trail.

3.3.1. Advantages of Proposed System

- **Enhanced Efficiency**: The platform automates key administrative tasks, such as player registration and score updates, saving time and reducing the workload for staff.

- **Centralized Data Management**: All sports-related information, including user details, team rosters, and event results, is stored in a single, accessible database. This eliminates the disorganization of paper records and makes data management more efficient.

- **Improved Communication**: By providing a centralized digital hub, the system ensures that updates on schedules, announcements, and match results are instantly available to all users. This overcomes the delays and missed information associated with physical notices.

- **Increased Transparency**: Features like the digital submission of match appeals and the detailed admin activity log create a clear, auditable trail of all actions. This enhances accountability and trust in the system.

- **Remote Accessibility**: As a web-based application, Sports Academix is accessible from any device with an internet connection. Users no longer need to be on campus to stay informed about sports activities.

- **Data Security**: The system uses secure practices like password hashing and prepared statements to protect user data and prevent common vulnerabilities such as SQL injection.

- **Scalability**: The modular architecture of the system makes it easy to add new features or accommodate a larger user base and more sports in the future, providing a solid foundation for continuous growth and improvement.

3.4. Feasibility Study

A feasibility study was conducted to evaluate the practicality and viability of the proposed Sports Academix project. The study focused on three key areas: technical, economic, and operational feasibility.

Technical Feasibility: Sports Academix is built with PHP, MySQL, HTML, CSS, and JavaScript, all proven, open-source technologies.These tools are stable, well-documented, and supported by large developer communities.The team has the expertise to implement secure authentication, dynamic data handling, and interactive dashboards with minimal technical risk.

Economic Feasibility: The project requires only basic hardware and free software, keeping costs very low.Open-source technologies eliminate licensing fees and reduce maintenance expenses.Time and resources saved from automating manual tasks provide long-term economic benefits.

Operational Feasibility: The system replaces the slow paper-based process with a centralized web platform.Its user-friendly interface ensures quick adoption with minimal training for students and staff.Automated registration, scheduling, and score updates create a more organized and transparent sports program.

Schedule Feasibility: The scope is clearly defined as a semester-length mini-project with manageable phases. A structured development plan allows effective tracking of tasks and milestones.The timeline ensures the project can be completed within the standard academic semester.

3.5. Requirement Specification

Software Requirement Specification (SRS) is the requirements document that provides the technical specification for the design and development of the software. This document enhances the system's quality of formalizing communication between the system developer and the user and provides the proper information for accurate documentation. A description of each function required to solve the problem is presented in the functional description. The behavioral description section of the specification examines the operation of the software as a consequence of external events and internally generated control characteristics. Validation criteria is perhaps the most important and ironically most often neglected section of the SRS. The validation criteria acts as an implicit review of all other requirements. The proposed system, **Sports Academix**, has the following requirements.

3.5.1 Functional Requirements

- **User Management**
    - Allow new user registration with unique email, username, and secure password.
    - Provide OTP-based password recovery.
    - Support role-based access *(*`admin` *and* `user`) with admin capabilities to view, block, or delete accounts.
- **Sports & Department Management**
    - Admins can create, edit, and delete sports with defined scoring rules and player limits.
    - Admins can manage college departments by adding, updating, or removing them.
- **Team Management**
    - Create teams and assign them to specific sports.

- Maintain rosters of players for each team.

- **Event & Schedule Management**
  - Admins can create and update events with date, time, venue, and participating teams.
  - Display upcoming schedules to users and update event statuses *(*`scheduled`*,* `completed`*,* `postponed`).
- **Score & Points Management**
  - Admins input match scores; system automatically calculates winners.
  - Generate dynamic points tables ranked by total points.
- **Interactive Features**
  - Users can submit sport registration requests, post comments on match results, and file appeals.
  - Admins can approve/reject registrations, and manage appeals and user feedback.

3.5.2 Non-Functional Requirements

- **Usability** – Intuitive and easy-to-navigate interface for both administrators and users.
- **Performance** – Fast response times and ability to handle multiple concurrent requests.
- **Security** – Secure user authentication, password hashing, protection against SQL injection, and restricted admin dashboard access.
- **Maintainability** – Modular and well-organized code for easy updates, debugging, and feature addition.

- **Compatibility** – Responsive design compatible with modern browsers and devices (desktop, tablet, mobile).

- **Reliability** – Robust error handling to prevent crashes and ensure consistent operation.

4.System Design

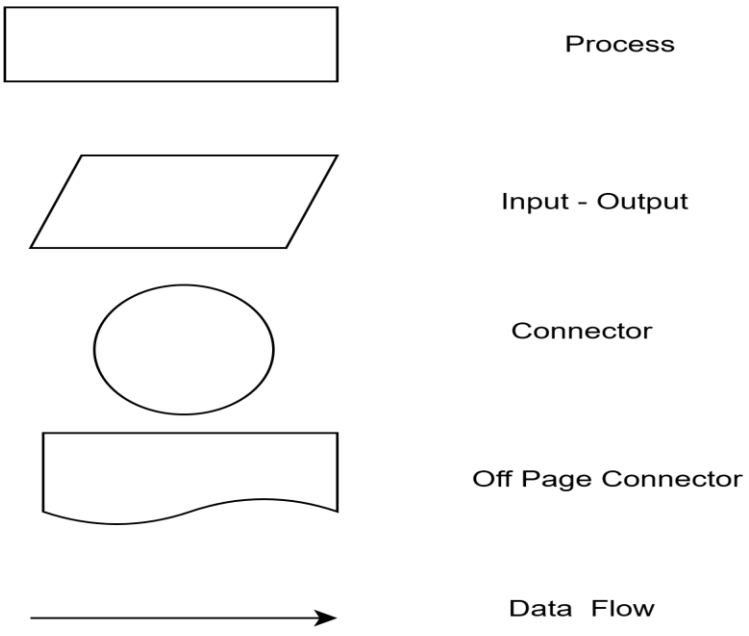
4.1 Introduction

System design is the process of defining the architecture, components, modules, interface and data for a system to satisfy specified requirements. It is a solution to an approach compared to system analysis which translates these "what is" orientation. System requirements into a way of making them operational. The design phase focuses on detailed implementation of the system recommended in the feasibility study. Planning of a system or to replace or complement an existing system. But before this, planning should be done. It must thoroughly understand the old system and determine how computers can make its operations more effective. The importance of system design is the quality. Design is the place where quality is fostered in software development. Design representation of software provides us with that which can be accessed for quality. System design is a transaction from a user-oriented document to a programmer or database personal.it is creative in both art and technology

4.2. System Flowchart

The classical system flowchart approach to describing and documenting a system will be presented. These system flowcharts are also used in the structured approach that is, from the general to detailed, of the system development life cycle. Because they have been used to describe systems for many years, they are still common in many businesses.

Basic Flow chart Symbols:



User Flow

1. Start – User opens the application.
2. Authentication – Login/Register screen is displayed.
3. Decision –
    - New user → selects Register.
    - Existing user → enters Email and Password.
4. Registration Process – System validates details, hashes password, and stores data in the database.
5. Login Process – Credentials are verified against the database.
6. Dashboard Routing –
    - Admin → Admin Dashboard.
    - Player/User → User Dashboard.
7. User Dashboard Functions –
    - View upcoming matches, recent results, personal profile, and match history.
    - Submit sport registration requests, comments on matches, and appeals for match results.
8. End – User logs out.

Administrator Flow

1. Start – Administrator opens the application and logs in.

2. Admin Dashboard – Overview of system statistics is displayed.

3. Management Tasks –

    - Users – View, block, or delete accounts.
    - Sports – Add, edit, or delete sports.
    - Events – Create, update, or delete event schedules.
    - Scores – Enter match scores; system auto-calculates results.
    - Registrations – Approve or reject player registration requests.
    - Appeals – Review and resolve match appeals.

4. Data Update – System records changes, updates the database, and logs all actions.

5. End – Administrator logs out.

flowchart

4.3. Database Design

The most important aspect of building an application is the design of tables or the database schema. The data stored in the tables must be organized in some manner,which is meaningful. The overall objective in the process of table design has been to treat data as an organizational resource and as an integrated whole. The organization of data in a database aim to achieve three major objectives, which are given below:

- Data integration
- Data abstraction
- Data independence

Several degrees of normalization have to be applied during the process of table design. The major aim of the process of normalization is to reduce data redundancy and prevent losing data integrity. Data integrity has to be converted at all levels. Pure normalization can access problems related to storage and retrieval of data. During the process of normalization, dependencies can be identified which cause serious problems during deletion and updating. Normalizing also helps in simplifying the structure of the table. The theme behind a database is to handle information as an integrated whole thus making access to information easy, quick, inexpensive and flexible for users.The entire package depends on how the data are maintained in the system.Each table has been designed with a perfect vision. Minor tables have been treated which though takes much space facilitates the process of querying fast and accurately.

PRIMARY KEY

The key is to identify records. Also uniquely notify the not null constraints.

FOREIGN KEY

The key which references the primary key, is the data inserted in the primary key column of the table.

NORMALIZATION

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example,storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

First normal form (1NF)

sets the very basic rules for an organized database: Eliminate duplicative columns from the same table. Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Second normal form (2NF)

further addresses the concept of removing duplicative data: Meet all the requirements of the first normal form. Remove subsets of data that apply to multiple rows of a table and place them in the separate tables.

Table 1: users

Primary Key: user_id

Field Name	Datatype	Length	Description
user_id	INT	11	Unique user ID
name	VARCHAR	100	Full name

email	VARCHAR	100	Email address
password	VARCHAR	250	Encrypted password
student_id	VARCHAR	20	Student roll/ID
role	ENUM	-	admin / user
status	ENUM	-	active / blocked
created_at	DATETIME	-	Account creation time
reset_token	VARCHAR	255	Token for password reset
reset_token_expires_at	DATETIME	-	Reset token expiry
otp_hash	VARCHAR	255	OTP hash
otp_expires_at	DATETIME	-	OTP expiry time

Table 2: departments

Primary Key: department_id

Field Name	Datatype	Length	Description
department_id	INT	11	Unique department ID
department_name	VARCHAR	100	Name of department

Table 3: sports

Primary Key: sport_id

Field Name	Datatype	Length	Description

sport_id	INT	11	Unique sport ID
name	VARCHAR	100	Sport name
code	VARCHAR	10	Short code for sport
max_players	INT	11	Maximum players allowed
status	ENUM	-	active / inactive
points_for_win	INT	11	Points awarded for win
points_for_draw	INT	11	Points awarded for draw
points_for_loss	INT	11	Points deducted for loss

Table 4: sports

Primary Key: sport_id

Field Name	Datatype	Length	Description
sport_id	INT	11	Unique sport ID
name	VARCHAR	100	Sport name
code	VARCHAR	10	Short code for sport
max_players	INT	11	Maximum players allowed
status	ENUM	-	active / inactive
points_for_win	INT	11	Points awarded for win
points_for_draw	INT	11	Points awarded for draw
points_for_loss	INT	11	Points deducted for loss

Table 5: team_members

Primary Key: team_member

Foreign Keys: team_id → teams(team_id), user_id → users(user_id)

Field Name	Datatype	Length	Description
team_member	INT	11	Unique member ID
team_id	INT	11	Team reference
user_id	INT	11	User reference
role_in_team	VARCHAR	50	Player / Captain / etc.

Table 6: events

Primary Key: event_id

Foreign Keys: sport_id → sports(sport_id), team1_id → teams(team_id), team2_id → teams(team_id)

Field Name	Datatype	Length	Description
event_id	INT	11	Unique event ID
event_name	VARCHAR	100	Event name
sport_id	INT	11	Sport reference
event_date	DATETIME	-	Date & time of event
venue	VARCHAR	100	Venue of event
description	TEXT	-	Event details

status	ENUM	-	schedule / ongoing / completed / etc.
team1_id	INT	11	First team
team2_id	INT	11	Second team
team1_score	INT	11	Score of team 1
team2_score	INT	11	Score of team 2
result	ENUM	-	team1_win / team2_win / draw / pending

Table 7: event_schedule

Primary Key: schedule_id
Foreign Keys: event_id → events(event_id), team1_id → teams(team_id), team2_id → teams(team_id)

Field Name	Datatype	Length	Description
schedule_id	INT	11	Unique schedule ID
event_id	INT	11	Event reference
team1_id	INT	11	First team
team2_id	INT	11	Second team
match_date	DATETIME	-	Match date
venue	VARCHAR	255	Match venue

Table 8: registrations

Primary Key: registration_id
Foreign Keys: user_id → users(user_id), sport_id → sports(sport_id), event_id → events(event_id), team_id → teams(team_id)

Field Name	Datatype	Length	Description
registration_id	INT	11	Unique registration ID

user_id	INT	11	Registered user
sport_id	INT	11	Registered sport
department	VARCHAR	100	Department name
team_id	INT	11	Team reference
event_id	INT	11	Event reference
status	ENUM	-	pending / approved / rejected
registered_at	DATETIME	-	Registration time

Table 9: feedback

Primary Key: feedback_id
Foreign Key: user_id → users(user_id)

Field Name	Datatype	Length	Description
feedback_id	INT	11	Unique feedback ID
user_id	INT	11	User reference
subject	VARCHAR	255	Feedback subject
is_read	TINYINT	1	Read flag
submitted_at	DATETIME	-	Submission timestamp
message	TEXT	-	Feedback message

Table 10: match_appeals

Primary Key: appeal_id

Foreign Keys: event_id → events(event_id), user_id → users(user_id)

Field Name	Datatype	Length	Description
appeal_id	INT	11	Unique appeal ID
event_id	INT	11	Event reference
user_id	INT	11	User reference
reason	TEXT	-	Appeal reason
submitted_at	DATETIME	-	Appeal submission time
status	ENUM	-	pending / reviewed / resolved

Table 11: match_comments

Primary Key: comment_id
Foreign Keys: event_id → events(event_id), user_id → users(user_id)

Field Name	Datatype	Length	Description
comment_id	INT	11	Unique comment ID
event_id	INT	11	Event reference
user_id	INT	11	User reference
comment	TEXT	-	Comment text
is_edited	TINYINT	1	Edited flag
commented_at	DATETIME	-	Comment timestamp

Table 12: notifications

Primary Key: notification_id

Foreign Key: user_id → users(user_id)

Field Name	Datatype	Length	Description
notification_id	INT	11	Unique notification ID
user_id	INT	11	User reference
message	TEXT	-	Notification message
link	VARCHAR	255	Related link
is_read	TINYINT	1	Read status
created_at	DATETIME	-	Creation time

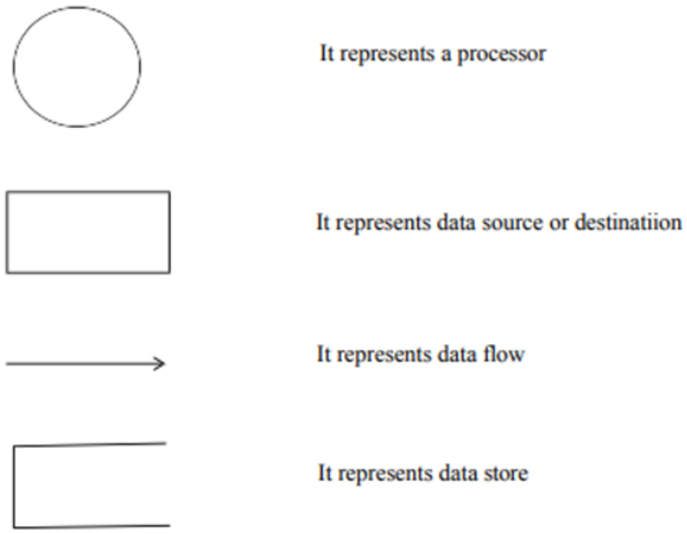
4.4. Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing. Data Flow Diagram is a common practice for a designer to draw a context-level Data Flow Diagram first which shows the interaction between the system and outside entities.

A Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by using a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates output data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs. There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles and are outside the system such as vendors or customers with whom the system interacts. They either supply or consume data are called sinks. Data is stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name, Process names are further identified with a number. The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the

inputs(source), all in a format that meets the user's requirements. The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

In the normal convention a DFD has four major symbols:



Level-0

dfd

4.5.Input Design

Input design is a critical aspect of software development that focuses on making data entry as efficient, accurate, and user-friendly as possible. For the Sports Academix project, the input design was crafted to guide users and administrators through a streamlined process, ensuring data integrity while enhancing the overall user experience. The design minimizes the potential for errors by using appropriate input types and clear visual cues.

User-Side Input Design

The input design for a general user is focused on simplicity and clarity. The primary input forms include:

- Registration Form: This form is designed for new users to create an account. It includes text inputs for Username and Email, and a password input with a toggle to reveal or hide the password.
- Sport Registration Form: To register for a sport, the user is presented with dropdown menus to select the Sport and their Department. This approach eliminates spelling errors and ensures valid data.
- Feedback & Match Appeal Forms: These forms use a multiline textarea for users to provide detailed feedback or a reason for a match appeal. The forms are kept simple to encourage user engagement.

Admin-Side Input Design

The input design for administrators is more comprehensive, covering all data management tasks. Key forms include:

- Add/Edit Sport Form: This form allows an admin to manage sport details. It includes a text input for Sport Name, numeric inputs for Max Players, and dropdowns for Points for Win, Draw, and Loss.
- Add/Edit Event Form: This is a multi-field form for scheduling matches. It features text inputs for Event Name and Venue, a date/time picker for Date and Time, and dynamic dropdowns for selecting the two Teams based on the chosen sport.
- Update Score Form: This form is designed for quick and accurate data entry. It provides two simple numeric input fields for Team 1 Score and Team 2 Score. The design visually emphasizes the teams for clarity.

- Admin Actions: Admin tasks like approving a registration or deleting a record are often handled via simple buttons that trigger a confirmation dialog, providing a clear and secure way to perform critical actions.

Design Principles Applied

The input design for Sports Academix adheres to the following principles:

- Clarity: Every input field has a clear label and, where necessary, a placeholder to guide the user.
- Consistency: A consistent style and layout are used across all forms, regardless of the user role, for a cohesive experience.
- Validation: Client-side validation is implemented to provide immediate feedback on invalid data, such as an incorrect email format or a weak password.
- Efficiency: Using dropdowns and dynamic inputs (e.g., filtered team lists) reduces typing and the chance of errors.

4.6. OUTPUT DESIGN

Output design is the process of presenting information to the user in a clear, efficient, and meaningful format. For the Sports Academix project, the output design is focused on creating a user-friendly and intuitive interface for both general users and administrators. The outputs are primarily web-based screens that dynamically display data retrieved from the database.

User-Side Output Design

The output design for a regular user is focused on providing easily digestible information to keep them engaged and informed.

- Main Dashboard: The user's landing page is a dashboard that presents a high-level overview of the sports program. It features interactive cards for each sport, showcasing key statistics such as the number of teams and players. Below this, dynamic lists display the most recent match results and the upcoming schedule.
- Sport-Specific Pages: Each sport has its own dedicated dashboard with a tabbed layout. The

outputs include:

- Upcoming Matches: Displays match details in a clean card format, showing the teams, date, time, and venue.
- Match Results: Presents completed matches with final scores and an outcome (e.g., "Team Won" or "Draw"). This output also includes a section for comments and a button to submit an appeal.
- Points Table: A tabular output that ranks teams based on a calculated points system, providing a clear overview of the league standings.

- User Profile: This section provides a personalized output, including the user's details, a list of their team memberships, their complete match history with outcomes (win/loss), and a history of their registration requests and appeals with their current status.

Admin-Side Output Design

The admin output design is focused on providing a comprehensive overview and the tools necessary for efficient management.

- Main Dashboard: The admin dashboard's primary output is a statistical summary, showing key metrics such as Total Users, Active Sports, Upcoming Events, and Pending Registrations in a clear, card-based layout. It also includes quick-action buttons and a list of recent administrative activities.
- Management Tables: All management modules (Users, Sports, Events, Teams, etc.) are displayed in a clean, responsive table format. Each table provides a detailed list of records with relevant information and includes action buttons for Add, Edit, or Delete functionality.
- Request & Feedback Management: The output for pending registrations and user feedback is presented in a way that allows for easy review. Color-coded status tags (e.g., green for approved, yellow for pending, red for rejected) are used to provide at-a-glance status updates, helping the administrator quickly prioritize tasks.

5.System development

5.1.Introduction

System development is the phase where the design blueprint is transformed into a functional and operational application. For the Sports Academix project, this involved implementing a modular architecture using PHP for the backend, MySQL for database management, and HTML, CSS, and JavaScript for the frontend interface. The development process followed a structured approach, with a clear separation of concerns, to ensure that all functional and non-functional requirements were met. The codebase is organized into logical files for database connection, API endpoints, and user interfaces, which facilitates easy maintenance and scalability for future enhancements.

5.2. Menu Level Description

The application features a clear and intuitive menu structure designed to provide a tailored user experience for different roles.

User Dashboard Menu

The user-side menu provides a simple and direct way for players and students to navigate the system. It consists of the following key links:

- Dashboard: The home screen, providing an overview of all sports, upcoming schedules, and recent results.
- My Profile: A personalized section where users can view their details, team memberships, and match history.
- Sport Registration: A dedicated page for submitting a request to register for a sport.
- My Appeals: A page to view the status of submitted match appeals.
- About: General information about the Sports Academix platform.
- Feedback: A form for users to submit comments or report bugs to the administrators.
- Logout: Securely ends the user's session.

Admin Dashboard Menu

The administrator-side menu is more comprehensive, serving as the central control panel for the platform. The admin menu includes:

- Dashboard: An overview of the system with key statistics and recent activity logs.
- User Management: To view, manage, and modify user accounts and roles.

- Department Management: For adding, editing, or deleting college departments.
- Sports Management: To configure sports, including scoring rules and player limits.
- Event Management: The primary hub for creating, scheduling, and updating all sports events.
- Upcoming Matches: A dedicated view for upcoming event schedules.
- Registrations: To review and approve or reject pending player registrations.
- Match Appeals: For reviewing and resolving user-submitted appeals regarding match results.
- Score Management: A module to input scores and finalize completed matches.
- User Feedback: To view and manage all feedback received from users.
- Logout: Securely ends the admin's session.

5.3. Process Specification

The following sections detail the core processes implemented within the system to meet its functional requirements.

User Authentication & Authorization

The authentication process is robust and secure, ensuring only authorized users can access the system.

1. Input: A user submits their email and password via the login form.
2. Validation: A backend PHP script receives the data and queries the database for the provided email.
3. Verification: The stored hashed password from the database is compared with the user's input using password_verify(). The user's account status (active or blocked) is also checked.
4. Session Management: If the credentials are valid and the account is active, a new session is created, and the user's user_id, name, and role are stored in session variables.
5. Authorization: The user is then redirected to either the user dashboard (home.php) or the admin dashboard (admin.php) based on their assigned role, ensuring proper access control.

Event and Score Management

This process allows an administrator to manage events from creation to finalization.

1. Event Creation: An admin uses a modal form to input event details such as event_name, sport, date, venue, and the two participating teams.

2. Data Persistence: A backend API call receives the data, and a prepared statement inserts the new event into the events table with a status of scheduled.
3. Score Update: For a completed event, the admin enters the team1_score and team2_score in a separate form.
4. Automatic Result Calculation: The backend PHP script compares the two scores to determine the result (team1_win, team2_win, or draw).
5. Finalization: The database is updated with the final scores and the calculated result, and the event's status is automatically changed to completed.

Dynamic Data Retrieval

The system's dashboards are populated with real-time data using dynamic queries.

1. Frontend Request: When a user loads a dashboard page, a JavaScript script makes a fetch request to the backend API.
2. Backend Processing: The PHP API script executes a series of SQL queries, often using JOIN statements, to gather comprehensive data (e.g., team names, sport names) from multiple tables.
3. Data Output: The results are formatted into a single JSON object and sent back to the frontend.
4. Frontend Rendering: The JavaScript code on the frontend receives the JSON data and dynamically generates the HTML content, populating tables, cards, and lists without requiring a full page reload. This makes the interface fast and responsive.

6. System Testing

6.1. Testing Methods

To ensure the quality and robustness of the **Sports Academix** application, a multi-faceted testing approach was applied.

Unit Testing

This method involves testing individual code components or modules in isolation to ensure each part functions correctly. For this project, unit tests were conducted on:

- The user authentication module to verify that password hashing and login logic are secure and functional.
- The score calculation function to confirm that the match result (win, loss, draw) is correctly determined from the input scores[2].
- Individual API endpoints to ensure they handle requests and return the expected JSON responses.

Integration Testing

This method verifies that different modules of the system work together seamlessly. Integration tests were performed on:

- The registration process, confirming that user details from the frontend are correctly stored in the database.
- The admin dashboard, to ensure that selecting a tab correctly triggers the backend API to fetch and display the relevant data.
- The event management workflow, to check that creating a new event properly populates the database and updates the user-facing schedule page.

User Acceptance Testing (UAT)

This final stage involved potential end-users, such as students and staff, interacting with the system. UAT helped validate that the user interface, navigation, and overall experience met their needs.

TESTING ENVIRONMENT & TOOLS

Debugging (Black Box Testing & White Box Testing)

Debugging helps to avoid bugs in the program. It helps to ensure that the system produces the desired output and also helps to identify the cases when undesired outputs are produced so that the developers can decide how the unexpected outputs or exceptional cases must be handled. Debugging can be done using two testing strategies namely Black Box Testing and White Box Testing.

Black Box Testing

Black box testing is the Software testing strategy used to test the software without knowing the internal structure of code or program. This means that implementation knowledge is not required to carry out Black Box Testing. This type of testing is carried out by testers. It is also referred to as a functional test or external testing.

White Box Testing

White box testing is the software testing strategy in which internal structure is being known to the person testing the software. Generally, this type of testing is carried out by software developers. Implementation Knowledge is required to carry out White Box Testing. It is also referred to as structural test or interior testing.

6.2 TEST PLAN ACTIVITIES

A detailed test plan was followed to ensure a systematic and thorough testing process for the Sports Academix project. This plan ensured that every key functionality was verified against the project's requirements, leading to a reliable and robust final product.

1. Test Case Design

This initial activity involved creating detailed test cases for every major feature. Each test case was designed to include:

- Test Scenario: A specific feature to be tested (e.g., "User Login" or "Admin Updates Match Score").
- Test Steps: A sequence of actions to be performed (e.g., "1. Enter a valid email and password. 2. Click the 'Login' button.").
- Expected Result: The anticipated outcome of the test (e.g., "User is successfully redirected to the dashboard.").

- Test Data: The specific data to be used during the test (e.g., email: 'user@example.com', password: 'correct-password').

2. Test Execution

In this phase, the designed test cases were executed on the live system. This involved running a variety of tests, including:

- Positive Testing: Verifying that the system works as expected with valid inputs (e.g., successfully registering a new user with correct details).
- Negative Testing: Ensuring the system handles invalid or unexpected inputs gracefully (e.g., a user attempting to log in with an incorrect password, or an admin trying to delete their own account).

3. Bug Reporting and Resolution

Any deviations from the expected results were documented as bugs. The process included:

- Logging: Creating a detailed report of the bug, including the steps to reproduce it.
- Assignment: Assigning the bug to a developer for fixing.
- Verification: Re-testing the feature after the fix was implemented to confirm that the issue was resolved.

4. Final Regression Testing

After all bugs were resolved and new features were added, a final round of regression testing was performed. The purpose of this activity was to re-run key tests on the integrated system to ensure that the new code changes had not introduced any unintended side effects or broken existing functionalities. This final step ensured that the overall stability and reliability of the platform were maintained.

7.System Implementation

System implementation is the process of taking a software system from the design phase to a fully functional and operational state. It requires careful planning, effective project management, and collaboration among all stakeholders. It is critical to ensure that the system meets the requirements and specifications, is reliable and secure, and is easy to use and maintain.

The following are the common activities involved in the implementation of the Sports Academix project:

- Installation: This activity involves installing the software system on the hardware environment where it will be used. For Sports Academix, this entails installing the application files on a web server like XAMPP in the local development environment or a production server for college-wide access.
- Configuration: This activity involves configuring the software system to meet the specific requirements of the organization or end-users. This includes setting up the config.php file with database credentials and configuring server settings.
- Data Migration: This activity involves transferring data from the old system to the new system. Since the existing system is manual and paper-based, this would involve migrating existing student records, team lists, and historical match data into the sports_management1 database.
- Testing: This activity involves testing the software system to ensure that it functions as expected and meets the requirements and specifications. The project underwent extensive unit, integration, and user acceptance testing to ensure all features work as designed.
- Training: This activity involves training end-users and administrators on how to use the new system. For Sports Academix, this would include training for administrators on how to use the management dashboard and guides for students on how to register and view data.
- Documentation: This activity involves creating documentation for the new system, including user manuals, technical documentation, and support materials. This includes a detailed project report and code documentation.
- Deployment: This activity involves deploying the new system in the production environment. For the final project, this means moving the application from the development server to the college's live web server.
- Maintenance: This activity involves ongoing maintenance and support of the system to ensure that it continues to function properly. This includes regular updates, bug fixes, and performance optimizations.
- Monitoring: This activity involves monitoring the system to identify and resolve any issues

that arise. This would involve checking server logs and database performance.

- Upgrades: This activity involves upgrading the system to new versions or releases as they become available. The modular design of Sports Academix makes it easy to add new features or modules.
- Support: This activity involves providing ongoing support to end-users and administrators to address any issues that arise and ensure the system is functioning as intended.

8.Conclusion And Scope For Future Enhancement

CONCLUSION

The Sports Academix project successfully delivers a comprehensive web-based platform for managing sports activities in an academic environment. By digitizing key processes and providing tailored dashboards for both users and administrators, the system effectively addresses the inefficiencies and shortcomings of the previous manual system. The project demonstrates the successful application of modern web development principles, including a secure and modular architecture built with

PHP, MySQL, HTML, CSS, and JavaScript. The implementation of features such as a dynamic points table, user registration, and a feedback system proves the system's viability and ability to enhance organizational efficiency. The system also prioritizes security through password hashing and prepared statements, ensuring data integrity and user protection. Overall, Sports Academix is a well-designed, functional, and reliable solution that transforms manual sports management into a streamlined, transparent, and accessible digital experience.

SCOPE FOR FUTURE ENHANCEMENT

While the current system is fully functional, there is significant scope for future enhancements to improve its capabilities and user experience. Possible future additions include:

- Notification System: A robust notification system that alerts users and teams of upcoming matches, score updates, or registration status changes in real-time.
- Mobile Application: Development of a dedicated mobile app for both iOS and Android platforms to provide a more convenient and accessible experience for users on the go.
- Collaborative Task Management: Implementing a feature that allows teams to manage their own tasks, such as practice schedules or team announcements, through a private dashboard.
- Advanced Analytics: Adding modules for in-depth statistical analysis of player performance and team records to provide valuable insights.
- Payment Integration: Integrating a payment gateway to handle fees for event registration or team merchandise.
- Social Features: Expanding the platform with social features, such as a photo gallery for

events or a live chat for matches, to increase user engagement.

9.Screen Layout

10.Biblography

1. OpenAI GPT-4o, 4.1: Large language models used for research, content generation, and refining project report text.

2. *Gemini 2.5 flash and Gemini 2.5 pro*: Additional large language models used for research and content assistance.

3. *YouTube*: A video platform used for tutorials, project examples, and learning new programming concepts.

4. *Brocode*: A specific YouTube channel or resource likely used for coding tutorials and explanations.

5. *Coding2go:* A learning resource or channel for coding, similar to the above.