CRICKET CLUB MANAGEMENT SYSTEM

ABSTRACT:

A "Cricket Club Management System" is a web-based management solution designed to streamline the operations of a cricket club. This system keeps track of player registrations, match schedules, team formations, and performance statistics, making it easier to manage the club's activities. The proposed system will record and manage information related to players, coaches, matches, and reports on club performance. This project allows administrators to update and retrieve information efficiently. The system includes web-based modules for managing players, creating match fixtures, maintaining attendance, and generating performance reports. The main objective of this system is to digitalize cricket club management processes, ensuring accurate record-keeping and efficient handling of operations. Through the web interface, users can access team rosters, match updates, and schedules. Administrators can monitor player's details, performance metrics, and club analytics with ease. The solution simplifies communication between players and staff, enhancing overall club productivity.

1.INTRODUCTION

1.1. ABOUT THE PROJECT

The **Cricket Club Management System** is a software application designed to streamline the management and operations of a cricket club. This system aims to digitize and automate various club-related activities, including player registration, match scheduling, team formation, score management, and a cricket club manually can be time-consuming and prone to errors. This project seeks to overcome these challenges by providing a centralized platform where administrators, players, and coaches can efficiently manage club activities. The system enhances communication between club members, ensures transparency in team selection, and simplifies record-keeping processes. Traditionally, cricket club management has been handled manually, leading to challenges such as miscommunication, data mismanagement, scheduling conflicts, and inefficient record-keeping. This project aims to address these issues by providing an automated and user-friendly platform for club administrators, players, coaches, and other stakeholders.

2.SYSTEM STUDY

2.1. EXISTING SYSTEM

The existing system for managing cricket clubs primarily relies on manual processes and basic digital tools such as spreadsheets and paper-based records. Club administrators, coaches, and players often face challenges in handling club operations, scheduling matches, maintaining player records, and tracking financial transactions**.**

2.2. DISADVANTAGES OF EXISTING SYSTEM

- **Time-Consuming** – Manual processes for player registration, scheduling, and record-keeping take a lot of time.
- **Prone to Errors** – Paper-based and spreadsheet-based record-keeping increases the chances of mistakes and data loss.
- **Lack of Centralized Data** – Information is scattered across different sources, making it difficult to retrieve and manage.
- **Inefficient Match Scheduling** – Without automation, scheduling matches and practices leads to miscommunication and conflicts.
- **Limited Player Performance Tracking** – No proper system to analyze player statistics, making team selection less data-driven.
- **Poor Communication** – Important updates about matches, events, and team selection are often missed due to unstructured communication methods.

2.3. PROPOSED SYSTEM

The proposed Cricket Club Management System is a digital platform designed to automate and streamline the management of cricket clubs. It will replace the manual, error-prone processes of the existing system with an efficient, centralized, and user-friendly solution.

2.4. ADVANTAGE OF PROPOSED SYSTEM

- **Time-Saving** – Automates key operations like player registration, match scheduling, and performance tracking, reducing manual effort.
- **Error-Free Record Keeping** – Eliminates human errors in data entry, player management, and financial transactions by maintaining accurate digital records.
- **Centralized Data Management** – Stores all club-related information in a single system, ensuring quick access and easy retrieval of data.
- **Better Match Scheduling** – Avoids conflicts in fixtures by automating match and practice session scheduling with instant notifications.
- **Fair and Transparent Team Selection** – Uses performance-based selection criteria to ensure unbiased and data-driven player selection.
- **Enhanced Player Performance Tracking** – Records and analyzes player statistics, helping coaches and players improve their game.

2.5. PROBLEM DEFINITION AND DESCRIPTION

Managing a cricket club manually is inefficient, time-consuming, and prone to errors. Player records, match schedules, and financial transactions are often maintained in physical registers or spreadsheets, leading to data loss, duplication, and mismanagement. Scheduling matches manually can result in fixture conflicts and communication gaps. Team selection lacks transparency due to the absence of proper performance tracking. Financial management is unstructured, making it difficult to track membership fees, sponsorships, and expenses. Players and coaches face poor coordination due to a lack of a centralized communication system. The existing system fails to provide real-time access to critical club information, affecting decision-making. Therefore, an automated Cricket Club Management System is required to enhance efficiency, accuracy, and club operations.

3.SYSTEM ANALYSIS

3.1.PACKAGES SELECTED

Front End	:	HTML,CSS,JAVASCRIPT,BOOTSTRAP
Back End	:	PHP,MYSQL

3.2.RESOURCE REQUIRED

Hardware Resource

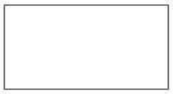
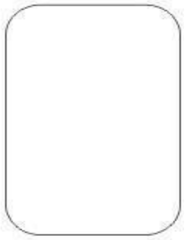
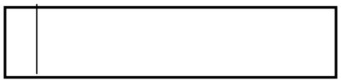
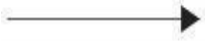
Processor	:	Intel processor 3.0 GHz
RAM	:	8GB
Hard disk	:	500 GB
Compact Disk	:	650 Mb
Keyboard	:	Standard keyboard
Mouse	:	Logitech mouse
Monitor	:	15 inch color monitor

Software Resource

Front End	:	HTML,CSS,JAVASCRIPT,BOOTSTRAP
Back End	:	PHP,MYSQL
Server	:	WAMP
Operating System	:	Windows OS
System type	:	32-bit or 64-bit Operating System

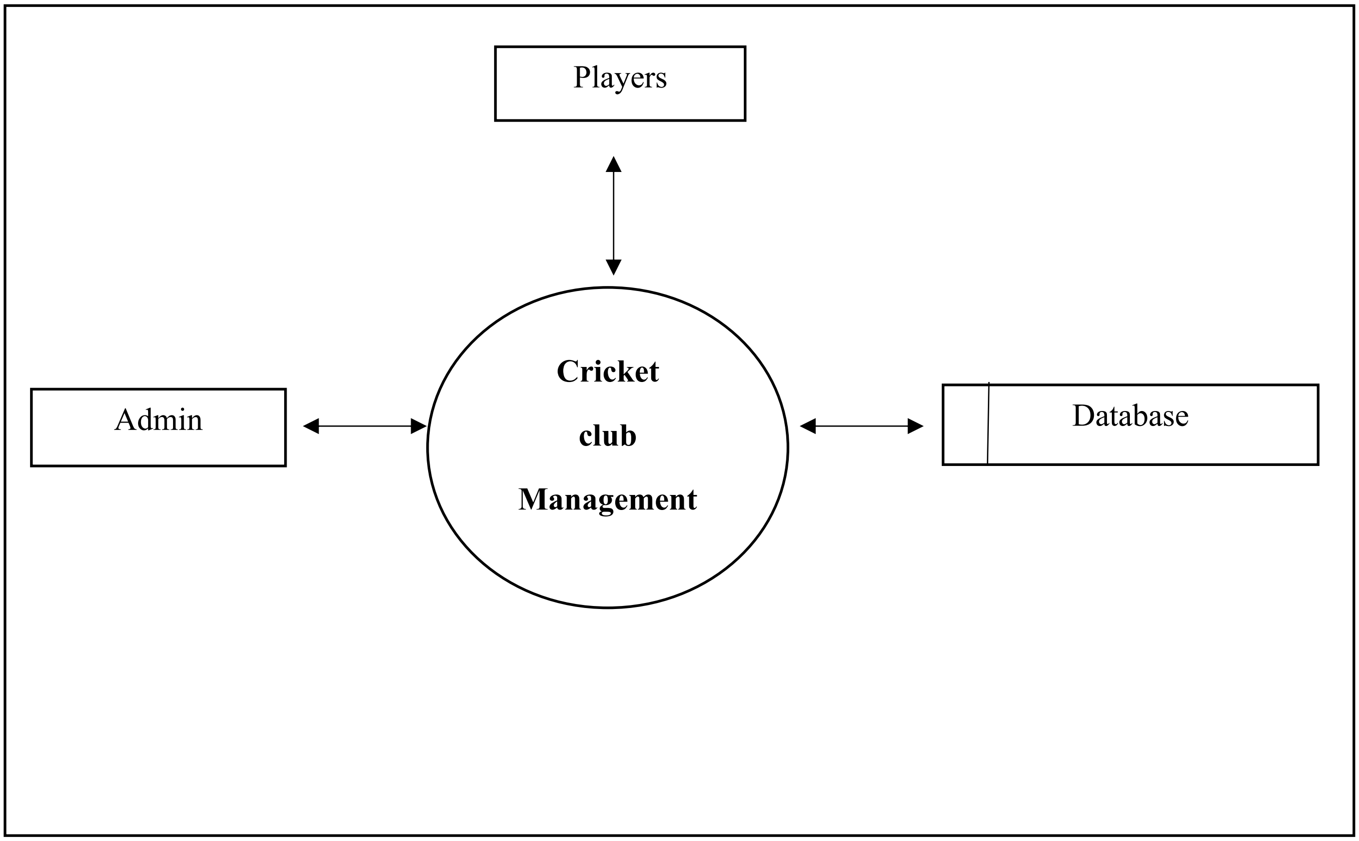
3.3. DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) visually represents the flow of data within a system. It illustrates how data moves between processes, data stores, and external entities. DFDs use standardized symbols such as circles (processes), arrows (data flow), and rectangles (external entities). They are hierarchical, starting with a high-level overview (Level 0) and expanding into detailed views (Level 1, 2, etc.). DFDs are useful for understanding system functionality, identifying data sources, and improving system design.

SYMBOL	DESCRIPTION
	An entity. A source of data or a destination for data.
	A process or task that is performed by the system.
	A data store, a place where data is held between processes.
	A data flow.

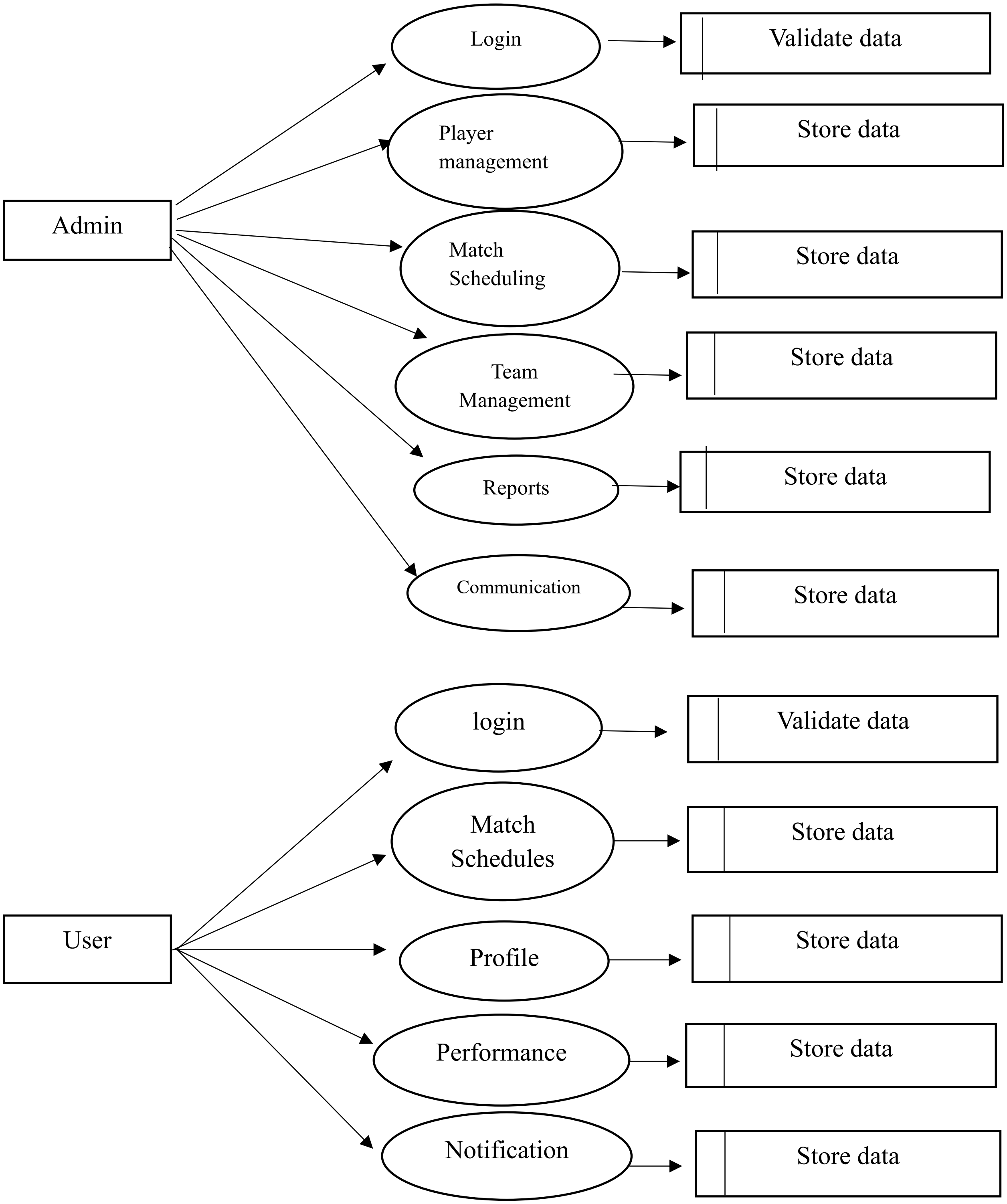
3.3.1. LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



3.3.2. LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.


Admin
Login
Validate data
Player management
Store data
Match Scheduling
Store data
Team Management
Store data
Reports
Store data
Communication
Store data
User
login
Validate data
Match Schedules
Store data
Profile
Store data
Performance
Store data
Notification
Store data

4.SYSTEM DESIGN

4.1. ARCHITECTURAL DESIGN

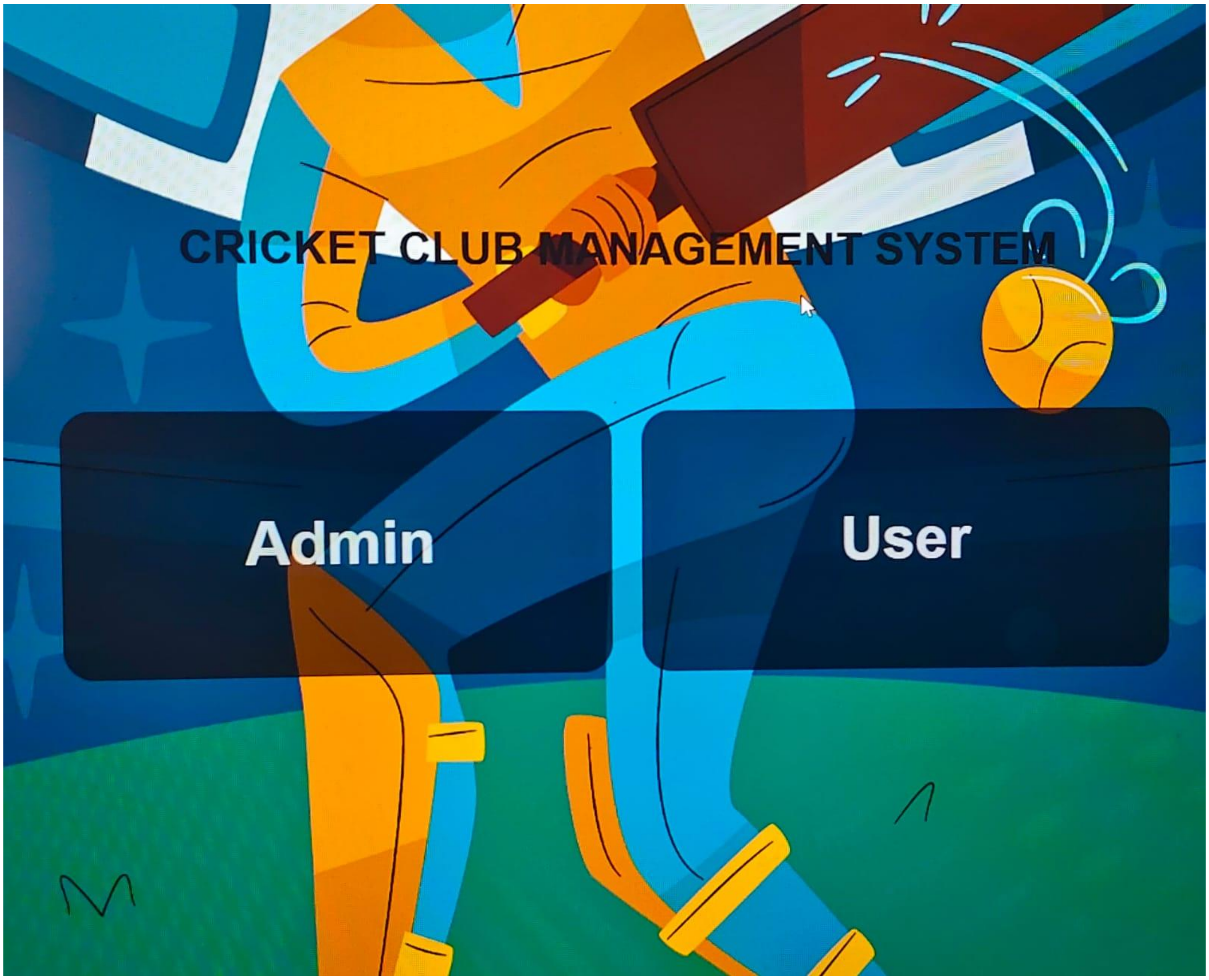
A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

4.1.1. Various organizations define systems architecture in different ways, including:

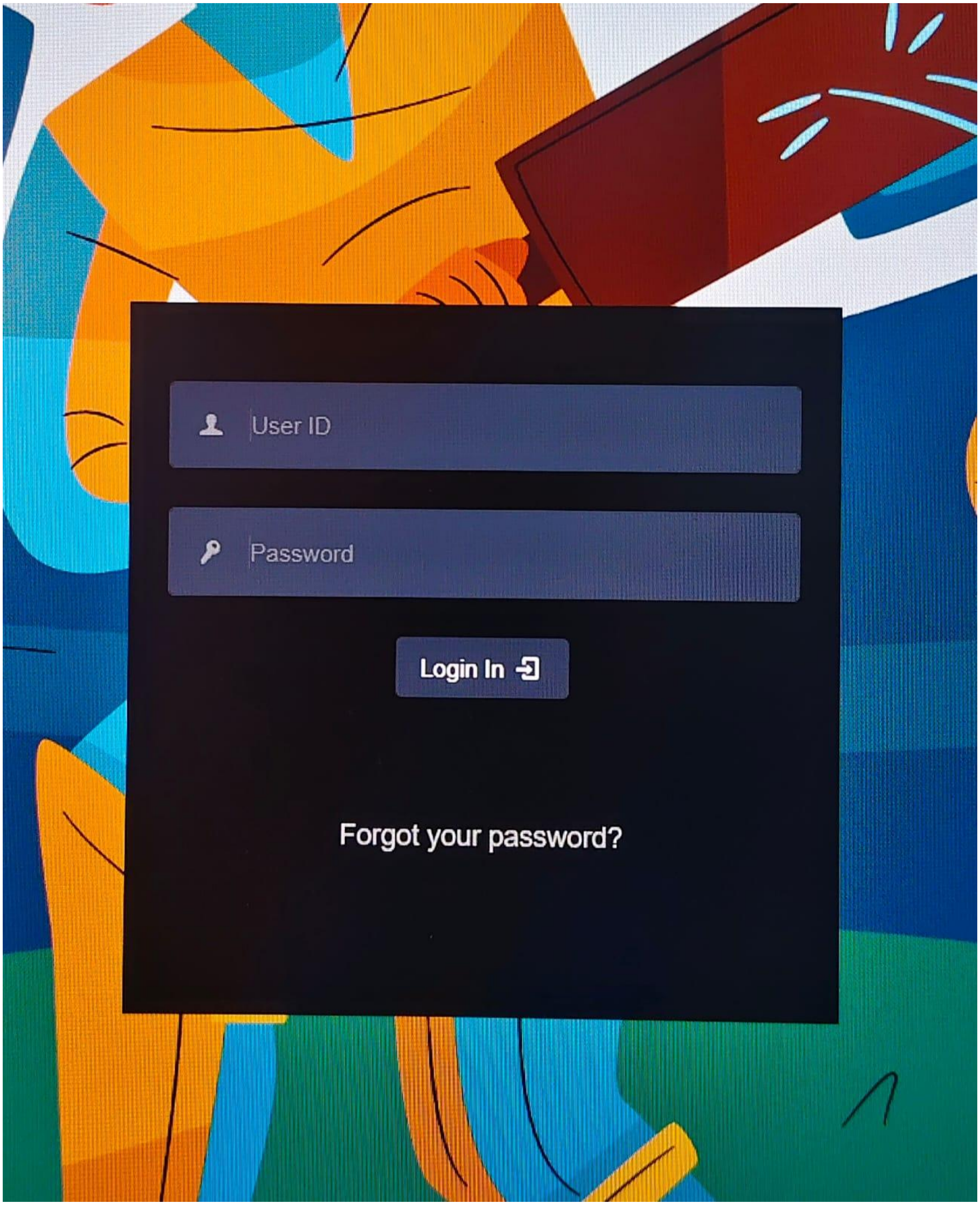
- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software
- The composite of the design architectures for products and their life-cycle processes.

4.2. I/O FORM DESIGN

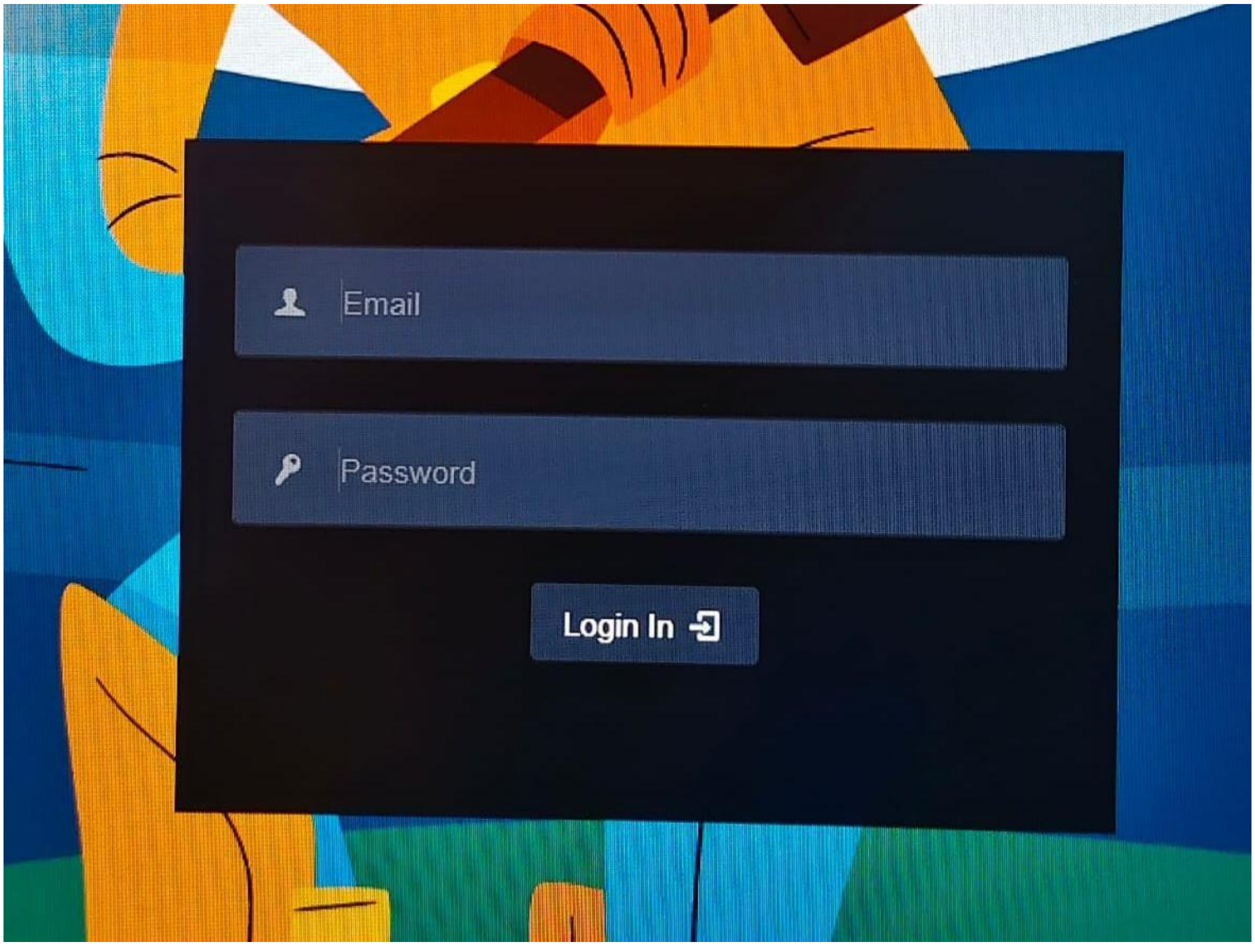
4.2.1 INDEX PAGE



4.2.2. ADD ADMIN FORM



4.2.3 ADD PLAYER FORM



4.3. TABLES

A table is a data structure that organizes information into rows and columns. It can be used to both store and display data in a structured format. For example, databases store data in tables so that information can be quickly accessed from specific rows. Websites often use tables to display multiple rows of data on page. Spreadsheets combine both purposes of a table by storing and displaying data in a structured format.

Databases often contain multiple tables, with each one designed for a specific purpose. For example, a company database may contain separate tables for employees, clients, and suppliers. Each table may include its own set of fields, based on what data the table needs to store. In database tables, each field is considered a column, while each entry (or record), is considered a row. A specific value can be accessed from the table by requesting data from an individual column and row.

Table 1 : **Admin**

Field	Type	Default
username	VARCHAR(20)	NOT NULL
pass_key	VARCHAR(20)	NULL
securekey	VARCHAR(20)	NULL
Full_name	VARCHAR(50)	NULL

Table 2: **log users**

Field	Type	Default
id	INT	NOT NULL
users_userid	INT	NOT NULL
action	VARCHAR(20)	NOT NULL
cdate	DATETIME	NOT NULL

Table 3 : **Matches**

Field	Type	Default
match_id	INT	AUTO_INCREMENT
match_title	VARCHAR(255)	NOT NULL
match_date	DATE	NOT NULL
venue	VARCHAR(255)	NOT NULL

Table 4: **Feedback**

Field	Type	Default
feedback_id	INT	AUTO_INCREMENT
player_id	INT	NOT NULL
feedback_text	TEXT	NOT NULL
submitted_at	TIMESTAMP	CURRENT_TIMESTAMP

Table 5 : **Match players**

Field	Type	Default
player_id	INT	AUTO_INCREMENT
player_name	VARCHAR(255)	NULL
match_id	INT	NULL

Table 6 : **Players**

Field	Type	Default
player_id	INT	AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
dob	DATE	NOT NULL
blood_group	VARCHAR(10)	NULL
height	FLOAT	NULL
weight	FLOAT	NULL
contact_number	VARCHAR(20)	NULL

4.4.NORMALIZATION

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant (useless) data.
- Ensuring data dependencies make sense i.e data is logically stored.

4.4.1. NORMALIZATION RULE

Normalization rules are divided into the following normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form

4.4.2. FIRST NORMAL FORM (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single (atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Table 7 : Model of first normal form

Player ID	Player Name	Phone Number	Match ID	Match Name
1	Sam	9003505907	101	Match A
2	Joe	9786158432	102	Match B

4.4.3.SECOND NORMAL FORM (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Table 8 : Model of second normal form

Player ID	Phone Number
1	9003505907
2	9786158432

4.4.4. THIRD NORMAL FORM (3NF)

A table is said to be in the Third Normal Form when,

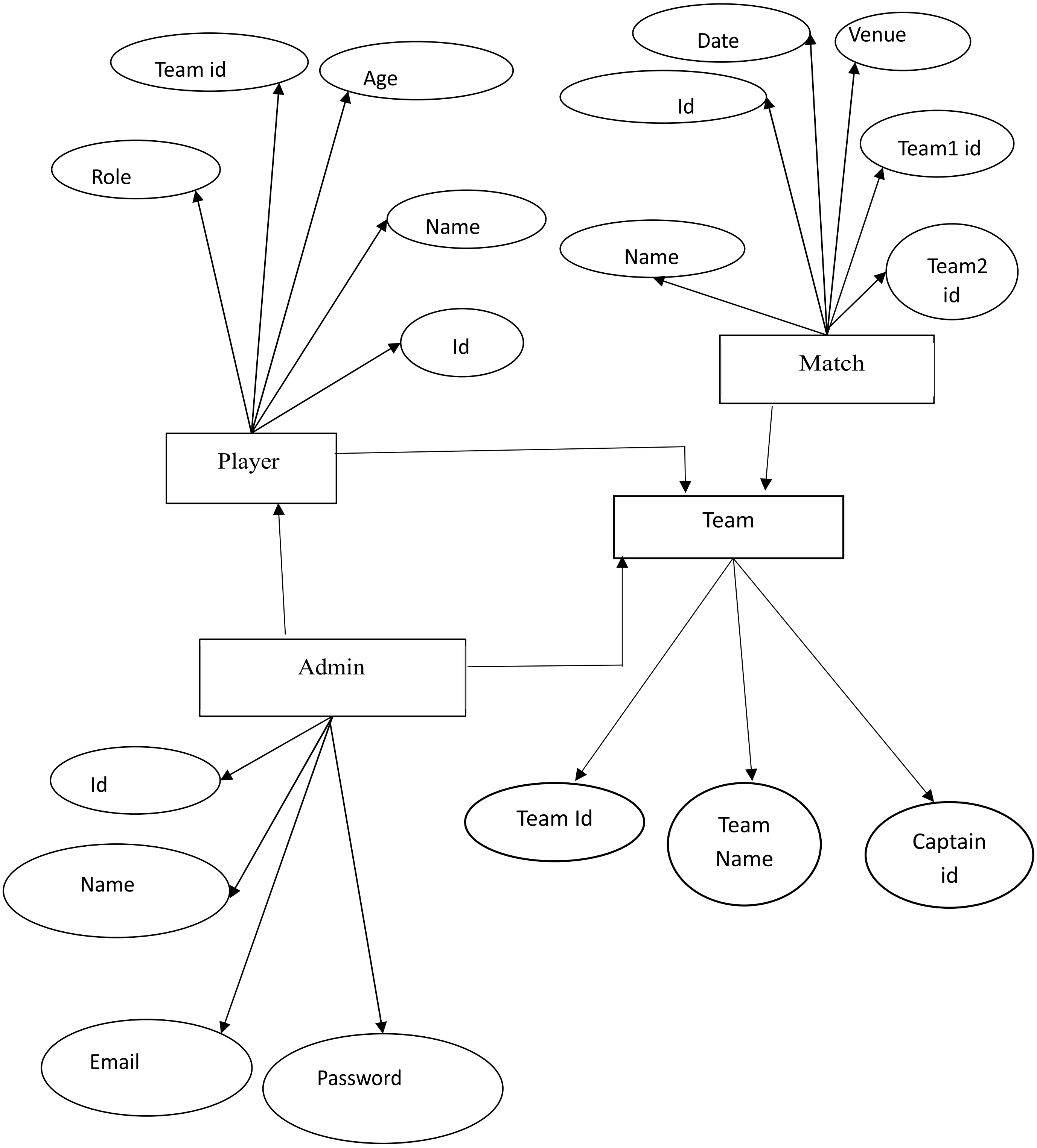
1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

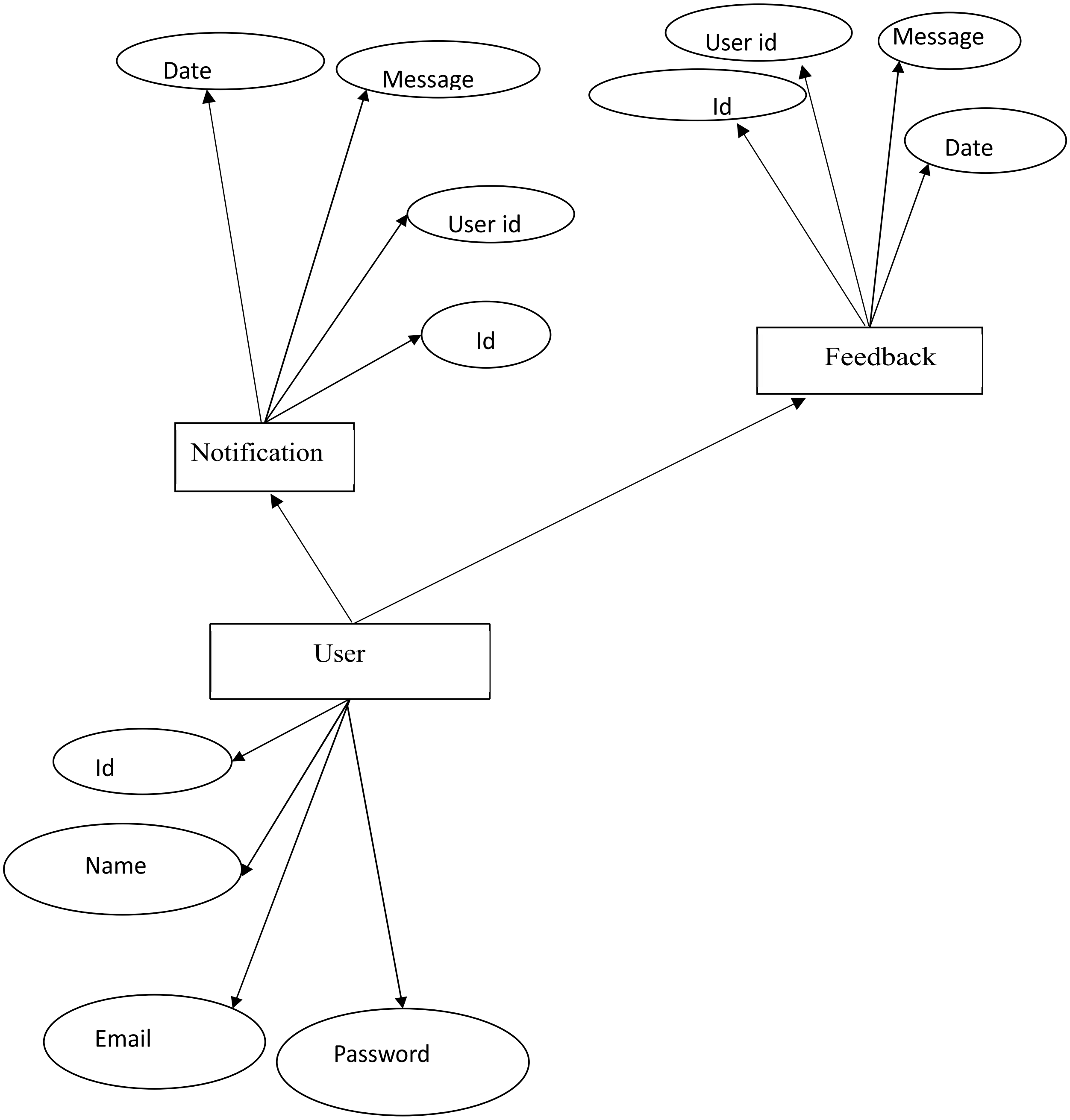
Table 9 : Model of Third Normal Form

Match ID	Match Name	Match Date
101	Match A	2024-05-01
102	Match B	2024-05-10

4.5. ENTITY RELATIONSHIP DIAGRAM

An Entity-Relationship Diagram (ERD) is a visual representation of data structures used in database design. It depicts entities (objects), attributes (properties of entities), and relationships (connections between entities). Entities are shown as rectangles, attributes as ovals, and relationships as diamonds or lines. ERDs help visualize data flow and identify database requirements.


Date
Venue
Team id
Age
Id
Team1 id
Role
Name
Name
Team2 id
Id
Match
Player
Team
Admin
Id
Team Id
Team Name
Captain id
Name
Email
Password



4.6. DATA DICTIONARY

A data dictionary is a structured collection of information about data elements in a database. It defines data types, structures, constraints, and relationships. Each entry typically includes the field name, data type, allowed values, and description. Data dictionaries ensure consistency, improve data understanding, and support database management. They are crucial for developers, analysts, and database administrators.

Table 10 : Model of data dictionary

Table Name	Attribute Name	Data Type	Constraints	Description
Admin	AdminID	INT	PK, Not Null	Unique identifier for Admin
Admin	Name	VARCHAR(100)	Not Null	Admin's full name
Admin	Email	VARCHAR(100)	Unique, Not Null	Admin's email address
Admin	Password	VARCHAR(100)	Not Null	Admin's login password
User	UserID	INT	PK, Not Null	Unique identifier for User
User	Name	VARCHAR(100)	Not Null	User's full name
User	Email	VARCHAR(100)	Unique, Not Null	User's email address
User	Password	VARCHAR(100)	Not Null	User's login password
User	Role	VARCHAR(50)	Not Null	User's role (e.g., player, coach)
Player	PlayerID	INT	PK, Not Null	Unique identifier for Player
Player	Name	VARCHAR(100)	Not Null	Player's full name
Player	Age	INT	Not Null	Player's age
Player	Role	VARCHAR(50)	Not Null	Player's role in the team
Player	TeamID	INT	FK	Reference to the Player's team
Match	MatchID	INT	PK, Not Null	Unique identifier for Match
Match	Date	DATE	Not Null	Date of the match
Match	Venue	VARCHAR(100)	Not Null	Venue where the match is held

5.SYSTEM DEVELOPMENT

5.1.FUCTIONAL DOCUMENTATION MODULES

1. **ADMIN MODULES:**

- Login
- Player Management
- Match Scheduling
- Team Management
- Reports and Analytics
- Communication Module

2**. USER MODULES:**

- Login
- Dashboard
- View Match Schedules
- Profile Management
- Performance History
- Notification and Feedback

MODULE DESCRIPTION

Admin Modules

1. Login
   - Secure authentication for admins to access the system.
   - Features like password recovery, two-factor authentication (optional).
2. Player Management
   - Add, edit, or remove player profiles.
   - Maintain player details such as name, age, role, and performance stats.
3. Match Scheduling
   - Schedule upcoming matches with date, time, and venue.
   - Assign teams and update match statuses (e.g., scheduled, completed, postponed).

4. Team Management
    - Manage team creation, player assignments, and captain selection.
    - Maintain team rosters and strategies.
5. Reports & Analytics
    - Generate detailed reports on player performance, match outcomes, and team statistics.
    - Visual data insights for better decision-making.
6. Communication Module
    - Send notifications, announcements, and updates to players.
    - Manage feedback and inquiries from users.

User Modules

1. Login
    - Secure authentication for club members or players.
2. Dashboard
    - Provides an overview of upcoming matches, performance highlights, and important updates.
3. View Match Schedules
    - Allows users to see upcoming matches, including dates, venues, and participating teams.
4. Profile Management
    - Enables users to view and update their personal details and cricket stats.
5. Performance History
    - Displays detailed performance records such as batting/bowling stats, match highlights, etc.
6. Notification & Feedback
    - Users receive match updates, club announcements, and can provide feedback to the admin.

5.2.SPECIAL FEATURES OF THE LANGUAGE

5.2.1.FRONTEND

- **HTML:**

HTML, which stands for HyperText Markup Language, is the standard markup language used to create and structure content on the web. It is the backbone of web development and provides a way to describe the structure of web pages using markup. HTML consists of a set of elements, each represented by tags, which define the various parts and components of a webpage. Tags are used to define elements in HTML. They consist of an opening tag, content, and a closing tag. HTML documents are built using a variety of markup elements, such as headings, paragraphs, lists, images, links, forms, and more.

- **CSS:**

CSS, which stands for Cascading Style Sheets, is a style sheet language used to describe the presentation and formatting of a document written in HTML or XML. CSS allows web developers to control the visual appearance of web pages, making it a crucial component in the design and layout of websites. By separating content (HTML) from presentation (CSS), developers can achieve greater flexibility and maintainability in web development.

HTML provides the structure and foundation for web content. When combined with Cascading Style Sheets (CSS) for styling and JavaScript for interactivity, it forms a powerful trio of technologies used in modern web development. Web browsers interpret HTML documents to render the visual representation of web pages. CSS provides tools for creating complex layouts and positioning elements on a page. Techniques include floats, flexbox, and grid layout..CSS supports responsive design principles, allowing developers to create websites that adapt to different screen sizes and devices.

CSS is an essential technology for web development, providing the means to create visually appealing, responsive, and interactive websites. When combined with HTML and JavaScript, CSS forms the basis for the three core technologies used in front-end web development.

- **JAVA SCRIPT:**

JavaScript is a high-level, interpreted programming language that is widely used for creating dynamic and interactive web pages. It is a versatile scripting language that can be embedded directly into HTML code and executed by web browsers to enhance the functionality and behavior of websites. JavaScript allows developers to create features such as form validation, content updates, and interactive user interfaces.

- **BOOTSTRAP:**

Bootstrap is a popular front-end framework for building responsive and mobile-first websites. It provides pre-designed CSS, JavaScript components, and utility classes to quickly create modern and consistent user interfaces. It Includes pre-built responsive grid systems for mobile-first design. Offers a wide range of UI components like buttons, modals, and navbars. It Provides built-in support for responsive typography, spacing, and utilities. Extensively customizable via Sass variables and Bootstrap's configuration.

- **Advantage of bootstrap**
  - Automatically adjusts layout for mobile, tablet, and desktop screens using a grid system.
  - Provides ready-to-use **buttons, modals, alerts, navigation bars, cards, and more**, saving development time.

5.2.2.BACKEND

- **PHP**

PHP (Hypertext Preprocessor) is a powerful, open-source server-side scripting language widely used for web development. Known for its simplicity and flexibility, PHP allows developers to embed code directly into HTML, making it easy to build dynamic and interactive web applications. It supports multiple databases like MySQL, PostgreSQL, and MongoDB, enabling seamless data management. PHP’s extensive library support simplifies tasks such as file handling, image manipulation, and encryption. With features like error handling through try-catch blocks, robust session management, and built-in security mechanisms to combat SQL injection and XSS attacks, PHP ensures secure web development.

Its compatibility across platforms like Windows, Linux, and macOS makes it versatile, while popular frameworks such as Laravel, CodeIgniter, and Symfony accelerate development with structured coding practices. PHP also offers excellent API integration capabilities, making it ideal for connecting external services. With features like email handling via the mail() function, command-line scripting for automation, and strong object-oriented programming support, PHP is a comprehensive solution for web development projects.

❖ **Advantage of PHP**

- Easy to Learn and Use: PHP has a simple syntax, making it beginner-friendly and easy to integrate with HTML.
- Open-Source and Cost-Effective: PHP is free to use, reducing development costs significantly.
- Cross-Platform Compatibility: PHP runs seamlessly on various operating systems like Windows, Linux, and macOS.
- Extensive Database Support: PHP efficiently connects with databases like MySQL, PostgreSQL, and MongoDB for dynamic web applications.
- Robust Frameworks: PHP offers powerful frameworks such as Laravel, CodeIgniter, and Symfony, which speed up development and ensure better code organization.
- Strong Community Support: PHP has a large, active community that provides extensive resources, documentation, and troubleshooting assistance.

5.2.3.DATABASE

❖ **MySQL**

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNUGeneral Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

INTER IMAGES

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

GRAPHICAL

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development(replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator**).**

COMMAND LINE

MySQL ships with some command line tools. Third-parties have also developed tools to manage a MySQL server, some listed below.Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Programming.

MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO OpenServer, SCO

UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.[32]

MySQL is written in C and C++. Its SQL parser is written in yacc, and a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBCinterimage called MyODBC allows additional programming languages that support the ODBC interimage to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

FEATURES

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licenses. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- Information schema
- Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
- X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine

- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Transactions with the InnoDB, and Cluster storage engines; savepoints with InnoDB
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- Full-text indexing and searching using MyISAM engine
- Embedded database library
- Unicode support (however prior to 5.5.3 UTF-8 and UCS-2 encoded strings are limited to the BMP, in 5.5.3 and later use utf8mb4 for full Unicode support)
- ACID compliance when using transaction capable storage engines (InnoDB and Cluster)

Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time): Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, EXAMPLE, Maria, and InnoDB, which was made the default as of 5.5). Partner-developed storage engines (solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.

6.TESTING

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectivesof the system. The philosophy behind testing is to find the errors. A good test is one that has a highprobability of finding an undiscovered error. A successful test is one that uncovers undiscoverederror. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

6.1. TYPES OF TESTING DONE

6.1.1. SYSTEM TESTING

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

6.1.2. UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

6.1.3. INTEGRATION TESTING

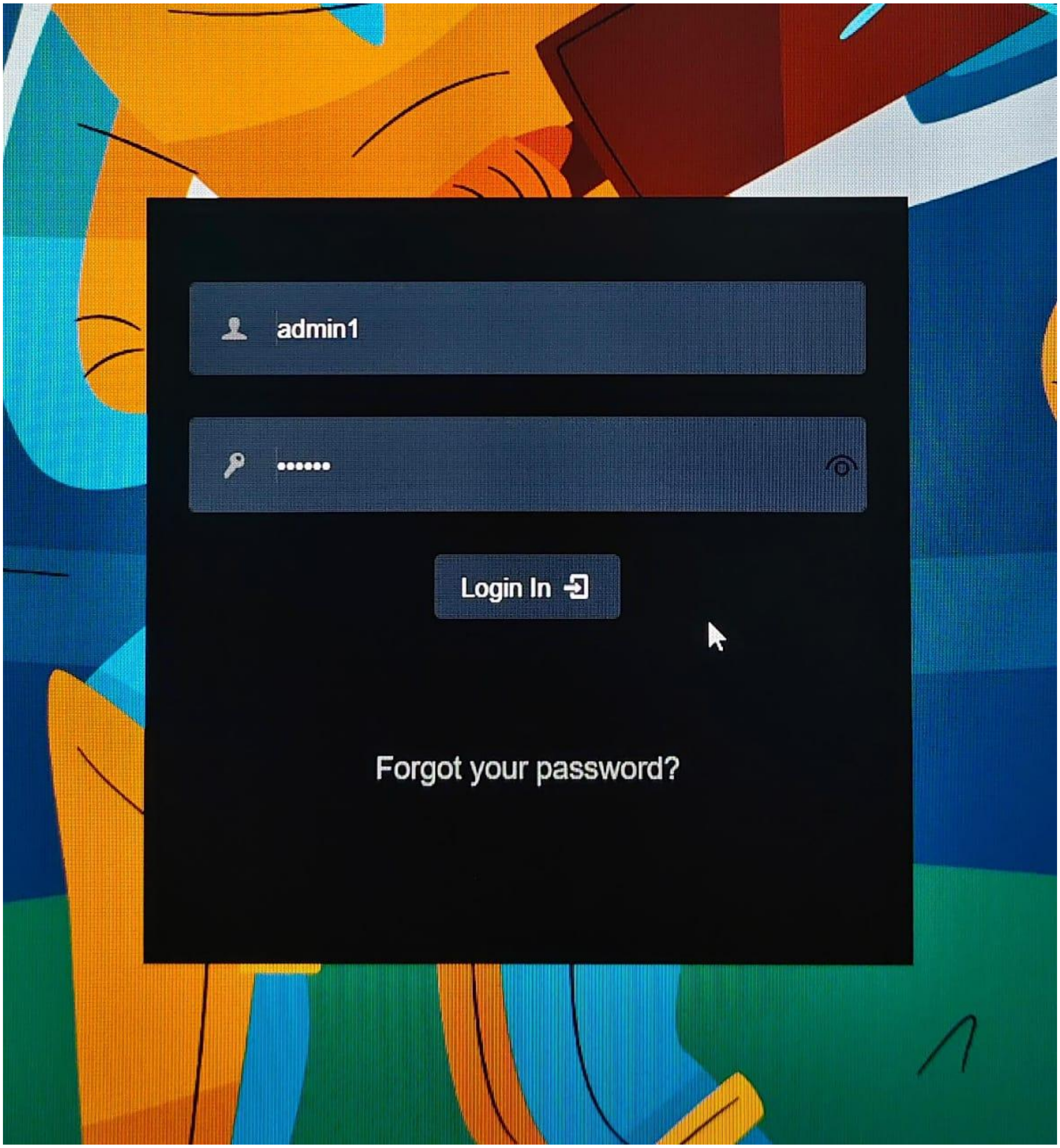
Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

6.1.4. VALIDATION TESTING

Valid and invalid data should be created and the program should be made to process this data to catch errors. When the user of each module wants to enter into the page by the login page using the use rid and password .If the user gives the wrong password or use rid then the information is provided to the user like “you must enter user id and password”. Here the inputs given by the user are validated. That is password validation, format of date are correct, textbox validation. Changes that need to be done after result of this testing.

6.2. TEST DATA AND OUTPUT ADMIN AND USER LOGIN

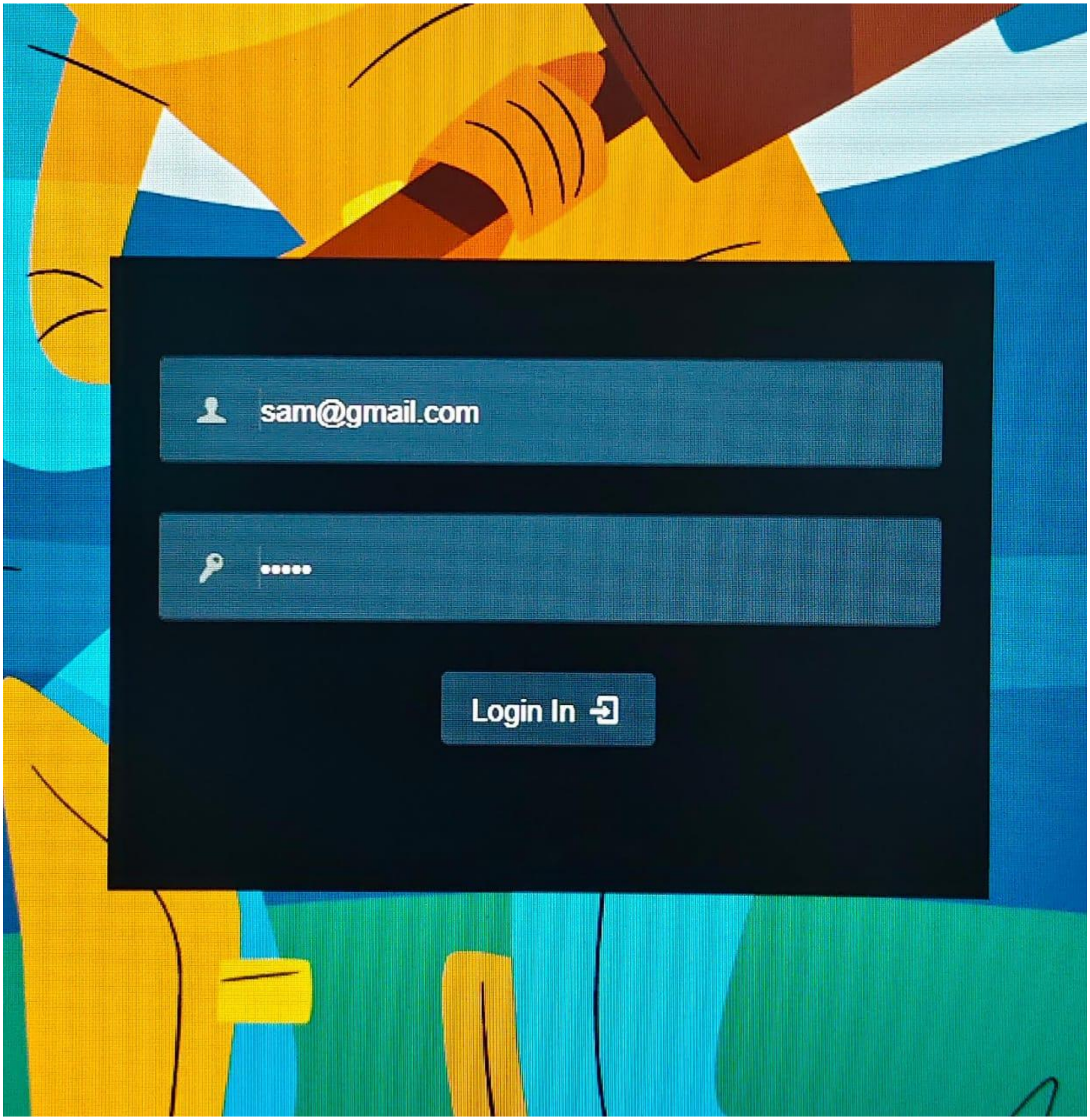
6.2.1 ADMIN FORM



admin1

••••••

6.2.2. USER LOGIN



sam@gmail.com

•••••

7.USER MANUAL

The **Cricket Club Management System** is a comprehensive web-based platform designed to efficiently manage various aspects of a cricket club. This system caters to both **Admin** and **User** roles, ensuring seamless communication and organized club activities. The platform is built using **HTML**, **CSS**, and **JavaScript** for the front-end interface, while the back-end functionalities are powered by **PHP** to ensure dynamic content management and secure data handling.

7.1.HARDWARE REQUIRMENTS

• Processor	:	Intel processor 3.0 GHz
• RAM	:	8GB
• Hard disk	:	500 GB
• Compact Disk	:	650 Mb
• Keyboard	:	Standard keyboard
• Mouse	:	Logitech mouse
• Monitor	:	15 inch color monitor

7.2. SOFTWARE SPECIFICATION

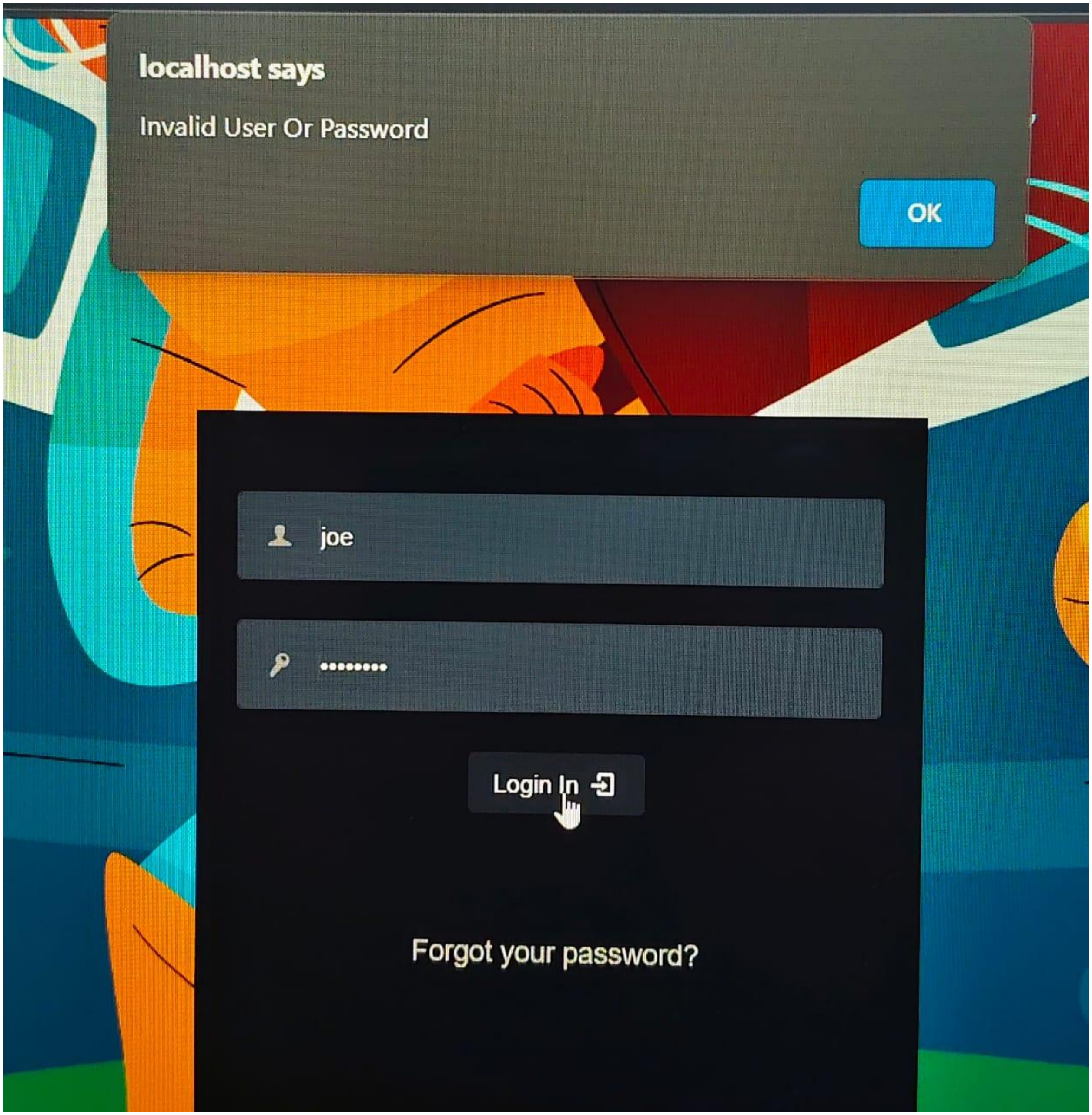
• Front End	:	HTML,CSS,JAVASCRIPT,BOOTSTRAP
• Back End	:	PHP,MYSQL
• Server	:	WAMP
• Operating System	:	Windows OS
• System type	:	32-bit or 64-bit Operating Syste

7.3. INSTALLATION PROCEDURE

- ➢ Downloading Wamp Server Download the installer file for the latest version of Wamp Server, and save the file to your computer.

- ➢ Installing Wamp Server To start the installation process, you need to open the folder where you saved the file, and double-click the installer file. A security warning window will open, asking if you are sure you want to run this file. Click Run to start the installation process.

7.4 ERROR MESSAGE



localhost says

Invalid User Or Password

8. CONCLUSION

8.1.SUMMARY OF PROJECT

The Cricket Club Management System is designed to efficiently manage cricket club activities for both administrators and users. The admin module provides powerful tools for handling player details, scheduling matches, managing teams, and generating performance reports. Users can log in to view match schedules, track their performance history, and receive important notifications. This system ensures smooth communication between players and admins, enhancing club coordination. By automating key processes, it reduces manual work and improves data accuracy. The intuitive interface offers easy navigation, ensuring accessibility for all users. With improved data organization, the system boosts productivity and enhances the overall club management experience. The project demonstrates effective integration of PHP, HTML, CSS, and JavaScript to deliver a functional and user-friendly platform. Future enhancements can include advanced analytics, mobile app integration, and improved user engagement features. Overall, this project effectively addresses the needs of cricket clubs by providing a reliable and efficient management solution.

8.2.FUTURE ENHANCEMENTS

- The project can be developed as an Android application in the future with additional features.
- **Scalability Improvements:** Enhance system architecture to handle increased data volume, user load, and performance demands.
- **Feature Expansion:** Introduce new functionalities based on user feedback and emerging trends to improve user experience.
- **Security Enhancements:** Implement advanced security protocols, such as multi-factor authentication and encryption, to ensure data protection.

BIBLIOGRAPHY

BOOK REFERENCE

- Duckett, Jon. *HTML and CSS: Design and Build Websites*. John Wiley & Sons, 2011.
- Duckett, Jon. *JavaScript and JQuery: Interactive Front-End Web Development*. John Wiley & Sons, 2014.
- Robbins, Jennifer. *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media, 2018.
- Ullman, Larry. *PHP and MySQL for Dynamic Web Sites*. Peachpit Press, 2017.
- Nixon, Robin. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. O'Reilly Media, 2018.
- Bacher, Bob. *Cricket Coaching: The Official Manual of the Cricket Coaches Association*. A&C Black, 2005.
- Richards, Barry. *The Art of Cricket Captaincy*. Pelham Books, 1991**.**

WEBSITE REFERENCE

- **https://www.w3schools.com/php/**
- **https://www.tutorialspoint.com/php/index.htm**
- **https://www.javatpoint.com/php-tutorial**