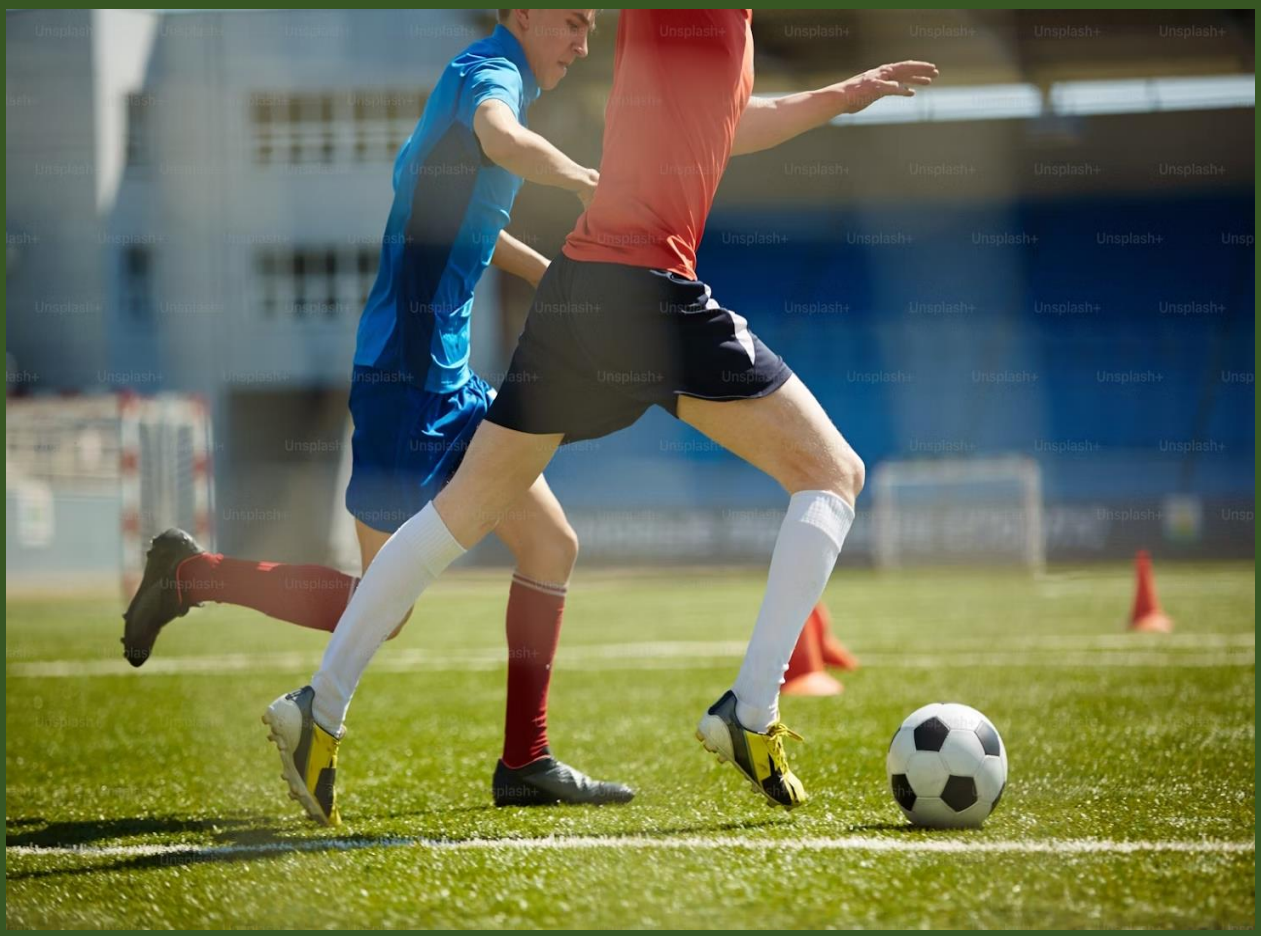
CONTEST-AZM

تو نشانِ عزمِ عالیشان



SOCCER VERSE WEB APPLICATION

Aptech LEARNING

Unleash your potential

Table of Contents

1. Background and Necessity for the Website

Soccer is one of the world’s most popular sports. It can be played almost anywhere, from official football playing fields (pitches) to gymnasiums, streets, school playgrounds, parks, or beaches. Soccer enthusiasts seek a dedicated platform that offers real-time access to game information, an immersive player gallery, top scorer rankings, and a user-friendly interface. A soccer-based Web Application fulfils these requirements and provide a seamless and immersive soccer experience for soccer fans. This Web application can also allow fans to select their favorite teams, players, and leagues. It can provide tailored content, match notifications, and recommendations based on the user's preferences.

2. Proposed Solution

The proposed solution is a Web Application called 'SoccerClub' that allows users to access soccer related information, schedules of matches, player galleries, top scores, and so on. In addition, users can also shop for jerseys, posters of their favorite players, and other related merchandise.

3. Purpose of the Document

The purpose of this document is to present a detailed description of the Web application titled SoccerClub. The document explains the various features and overall purpose of this Web Application, what it will do, and what is beyond

the scope. This document is intended for both stakeholders and developers of the application.

4. Scope of Project

This Web application will be a responsive and visually appealing one to be used by individuals. Users can browse through soccer game information, players' gallery, top 10 scores, and so on. Users can register and perform search and filter operations on data. They can browse merchandise such as jerseys and purchase them. However, the application will not have any feature/functionality for implementing or authenticating payment and delivery. These actions are beyond the scope of this application.

5. Constraints

Keeping the soccer information up to date can be a significant constraint. The application may rely on external data sources, such as live scores, player statistics, team information, and match schedules. Ensuring timely and accurate data acquisition, handling data format changes, and managing data updates can be challenging. The usage of soccer-related data, images, and videos may be subject to licensing agreements and copyright restrictions. It is important to understand and comply with these constraints to avoid legal issues.

6. Functional Requirements

For Users

Home Page:

Designed to provide an engaging and informative experience for users.

Account Registration:

Allows users to create secure accounts.

Required fields: Name, Email ID, Contact Number, Username, and Password.

Implement client-side validation for correct data entry (e.g., proper email format).

Login:

Enables successfully registered users to log in and access various features through menus or sidebars.

Settings:

Users can manage their accounts with options to Create, Update, and Delete Profile.

Users can also add addresses.

Soccer Info:

Provides complete details about soccer, including history, rules, field details, tricks, and tactics.

Matches:

Showcases a list of upcoming matches with details such as date, time, teams, and competition (e.g., World Cup, Champions League).

Allows users to click on a match for more information or set reminders.

Players' Gallery:

Includes player profiles categorized by teams.

Provides career statistics, images, and notable achievements.

Statistics:

Displays comprehensive statistics for teams, players, and matches.

Top 10 Scores:

Displays the top 10 scores of all time.

Latest News and Updates (Optional):

Dedicated section for the latest soccer news and updates.

Displays headlines or excerpts of recent articles or blog posts.

Allows users to click for more information. Implementation may involve Web services, REST APIs, or similar technologies.

Merchandise:

Displays various merchandise such as jerseys, soccer shoes, posters, etc.

Enables users to purchase items with an Add to Cart feature.

Checkout:

Shows shopping cart contents and computes the total bill based on merchandise price and quantity.

Payment functionality is not required; only listing cart contents and computing the bill.

Cancel Order:

Allows users to cancel an already placed order.

View Cart:

Enables users to view their cart.

Contact Us:

Displays the Email ID, address, and contact number of the organization developing the application.

Submit Feedback:

Users can submit feedback regarding the application using a feedback form.

Sitemap:

A sitemap must be created and added to the home page to understand the flow of the SoccerClub Web application.

For Admin

Login:

Allows Admin to log in to the Web Application.

Add/Modify/Delete:

Enables Admin to perform these operations on match schedules, player details, statistical data, accessories, orders, and other related information.

View Feedback:

Allows Admin to view feedback submitted by users.

Common to Both Admin and Users

Search/Filter:

Options to search or filter for a particular player, match, etc., based on specific criteria.

Sort:

Enables sorting of data based on specific criteria.

7. Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the Website. The Website should be:

- Safe to use: The Website should not result in any malicious downloads or unnecessary file downloads.
- Accessible: The Website should have clear and legible fonts, user-interface elements, and navigation elements.
- User-friendly: The Website should be easy to navigate with clear items and other elements and easy to understand.
- Operable: The Website should operate in a reliably efficient manner.
- Performance: The Website should demonstrate high value of performance through speed and throughput. In simple terms, the Website should be fast to load and page redirection should be smooth.
- Capacity: The Website should support a large number of users.
- Availability: The Website should be available 24/7 with minimum downtime.
- Compatibility: The Website should be compatible with the latest browsers.

These are the bare minimum expectations from the project. It is a must to implement the functional and non-functional requirements given in this SRS. Once they are complete, you

can use your own creativity and imagination to add more features if required.

8. Interface Requirements

8.1 Hardware

- Intel Core i5 Processor or higher
- 8 GB RAM or higher
- Color SVGA
- 500 GB Hard Disk space
- Mouse
- Keyboard

8.2 Software

Technologies to be used:

1**. Frontend:**

- HTML5: For structuring the content on the web pages.
- CSS3: For styling and designing the web pages.
- Bootstrap (optional): For responsive design and UI components.
- JavaScript: For client-side scripting and dynamic content.
- Figma Toolkit: For designing UI/UX layouts and prototypes.
- jQuery: For simplified HTML document traversal and manipulation.

- AngularJS/Angular 9/ReactJS: For building interactive and dynamic user interfaces.
- ReactJS: For creating reusable UI components and managing application state.
- XML: For data exchange and configuration.

2. Client and Server:

- Java:
    1. Version: 9 or higher.
    2. Frameworks: Java EE 7 or higher / Jakarta EE 9 or higher.
    3. IDE: Apache NetBeans IDE / Eclipse (latest version).
    4. Servers: Apache Tomcat 10.0 or higher, GlassFish 6.0 or higher.
    5. Libraries: Relevant libraries for Java EE/Jakarta EE.
- C#:
    1. Version: 7.0 or higher.
    2. IDE: Visual Studio IDE 2019 or higher.
    3. Frameworks: ASP.NET MVC and Core.
    4. Libraries: Related libraries for ASP.NET.
- PHP:
    1. Version: 7.0 or higher.
    2. Frameworks: Laravel Framework.
    3. Hosting (optional): Homestead.
- Python:
    1. Version: 3.0 or higher.
    2. IDE: PyCharm IDE.
    3. Frameworks: Django 4.0.2 or higher / Flask.

- Node.js and Express:
    1. Node.js: JavaScript runtime for building scalable network applications.
    2. Express: Web application framework for Node.js to handle routing and middleware.
- For Hosting (optional):
    - XAMPP: Latest version for local development and hosting.
- 4. Data Store:
    - MySQL: 5.7 or higher.
    - SQL Server: 2016 or higher.

9. Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams:
- Flowcharts for various activities
- Data Flow Diagrams (DFDs)
- Other relevant diagrams
- Database Design:
- Database scripts to be provided
- Source Code
- Installation Guide
- User Manual along with test data and login credentials

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive. The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions. If local servers such as XAMPP are used to test the project, include screenshots of working pages. Include port settings and other details in ReadMe file. In addition, you must submit a video clip showing the actual working of the Web application. Over and above the given specifications, you can apply your creativity and logic to improve the application