A PROJECT REPORT
for
Mini Project-I (K24MCA18P) Session
(2024-25)

Submitted by

VAIBHAV SHARMA(202410116100232)

SHIV KUMAR(202410116100197)

SHUBHAM SINGH (202410116100204)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Divya Singhal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad

Uttar Pradesh-201206

(DECEMBER- 2024)

CERTIFICATE

Certified that **VAIBHAV SHARMA (2426MCA2427), SHIV KUMAR(2426MCA2427), SHUBHAM SINGH(2426MCA2427)**has/ have carried out the project work having (**Mini Project-I, K24MCA18P**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other
University/Institution.

Ms. Divya Singhal	**Dr. Arun Kr. Tripathi**
Assistant Professor	**Dean**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of Institutions, Ghaziabad	**KIET Group of Institutions, Ghaziabad**

ABSTRACT

Cricket is one of the most popular sports globally, and keeping track of match scores is essential for players, coaches, and fans alike. The *Offline Cricket Score Display* project aims to develop a lightweight and user-friendly system to display cricket scores without relying on internet connectivity. This project leverages **HTML, CSS, and JavaScript** to create a responsive and interactive web application that can function offline through preloaded data and browser caching mechanisms.

The system allows users to input match-related details such as team names, player statistics, scores, and overs. Once the data is entered, the application dynamically displays the cricket scores in a clean and structured format. Additionally, **JavaScript** ensures real-time score updates, enabling users to modify scores manually during the game.

This project is particularly useful in environments with limited or no internet access, such as school projects, local cricket tournaments, or areas with connectivity issues. By storing the required data locally, it ensures smooth functionality and enhances user experience. The offline cricket score display system highlights the effectiveness of frontend web development technologies in creating functional, offline-capable applications.

The project provides an excellent opportunity to understand and implement key concepts of web development, such as data storage, user input handling, and responsive design.

Keywords: Offline Cricket Score, Web Application,HTML,CSS, JavaScript, User Interface, Real-time Updates.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

VAIBHAV SHARMA

SHIV KUMAR

SHUBHAM SINGH

TABLE OF CONTENTS

Chapter 1

Introduction

Project description

- The **Offline Cricket Score Display** is a mini-project designed to create a standalone web application for tracking and displaying cricket match scores without requiring an internet connection. Developed using **HTML, CSS, and JavaScript**, this project aims to offer a lightweight, user-friendly solution for offline score management, particularly in environments where internet access is unavailable or unreliable.
- The application allows users to input and update key match details such as:
- Team names
- Individual player scores
- Runs scored
- Wickets taken
- Overs bowled
- Using **JavaScript**, the scores are dynamically updated in real-time as inputs are provided, ensuring an interactive experience. The project utilizes **local storage** or browser caching to enable offline functionality, meaning data can persist even if the application is accessed without an internet connection. A simple and intuitive user interface, designed with **CSS**, ensures the data is presented in an organized, readable manner, suitable for users of all ages and technical expertise.
- This project is ideal for scenarios like local cricket matches, school or college tournaments, or even practice games where a digital yet offline scoring solution is needed. It demonstrates the practical application of frontend web development concepts such as:

- **User input handling**
- **Data persistence with local storage**
- **Dynamic content updates**
- **Responsive UI design**
- Through this project, developers gain experience in building interactive web applications that can work seamlessly offline while meeting user requirements for simplicity and reliability.

1 Project Scope

The **Offline Cricket Score Display** project focuses on building a lightweight, interactive web application for managing and displaying cricket scores without internet dependency. The scope includes:

1. **User Input and Score Management**:
   - Input team names, player details, runs, wickets, and overs.
   - Update scores dynamically during the match.
2. **Offline Functionality**:
   - Utilize browser **local storage** or caching to ensure the application runs without an internet connection.
3. **User Interface**:
   - Develop a clean, responsive UI using **HTML** and **CSS** to display match details effectively. 4. **Real-time Updates**:
   - Implement **JavaScript** to update scores and display changes instantly as user inputs are modified.
5. **Accessibility and Usability**:
   - Ensure the system is simple to use and suitable for local tournaments, practice matches, and educational purposes. **Out of Scope**:

- Integration with online scoreboards or live data APIs.

Chapter 2

Feasibility Study

1 Technical feasibility-

It is the complete study of the project in terms of input, processors, output, fields, programs and procedure. It is a very effective tool for long term planning and troubleshooting. Technical feasibility is a process of validating the technology assumptions, architecture and design of a product. The following are common types of technical feasibility.

- **Concepts**

  Conducting a proof of concept to test an idea or approach.

- **Infrastructure**

  The capacity, performance characteristics and functionality of infrastructure. For example, a project may validate an assumption that a new system can use an organization's existing network infrastructure.

- **Facilities**

  Confirming that facilities such as data centre will support project requirements.

- **Architecture & Design**

  Validating the architecture and design of a project against functional and non-functional requirements. This can include a peer review process.

- **Data**

  Checking that data supports requirements. For example, evaluating the data quality of required information.

- **Compliance**

  Compliance to technology standards and regulations.

- **Platforms & APIs**

  Evaluating platforms and APIs in areas such as functionality and reliability.

- **Components**

Tests and prototypes of component parts and materials.

- **Tools**
  Validating technologies such as systems and applications. For example, confirming that an application can be customized to meet user interface requirements.
- **Integration**

Looking at how processes, systems, applications, and data will work together.

2.2 Operational Feasibility-

- Operational feasibility refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and full fill the requirements as identified during the development of the project. It takes care that the management and the users support the project.
- Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.
- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyze whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

Chapter 3

Project Objective

The primary objective of the "Offline Cricket Score Display" project is to design and develop a reliable and interactive platform for recording and displaying cricket scores in offline mode.

Key Objectives:

- **User-Friendly Interface:** Providing an easy-to-use interface for manual score entry and updates.
- **Real-Time Updates:** Ensuring real-time score updates without internet dependency.
- **Data Management:** Offering functionalities to reset, edit, and manage match data efficiently.
- **Visual Appeal:** Enhancing user experience with a visually appealing and responsive design.
- **Event Support:** Supporting small-scale cricket events and tournaments where internet access is limited or unavailable.
- **Technological Application:** Demonstrating the practical application of web development technologies in building offline-capable solutions.
- **Customization:** Allowing easy customization for different match formats .
- **Data Storage:** Enabling storage and retrieval of match data for future reference.
- **Cost-Effective:** Reducing reliance on expensive or complex scoring software.
- **User Training:** Facilitating user training with a simple and intuitive user interface.

- **Device Compatibility:** Ensuring compatibility across multiple devices and screen sizes.
- **Workflow Efficiency**: Promoting efficiency in cricket score management through streamlined workflows.
- **Performance Monitoring:** Tracking player and team performance across multiple matches.
- **Accessibility**: Ensuring the platform is accessible to users with varying levels of technical expertise.
- **Security:** Safeguarding stored match data from unauthorized access or accidental loss.
- **Error Handling:** Implementing robust error handling mechanisms for data entry and updates.
- **Scalability:** Designing the system to accommodate larger tournaments and increasing data volumes.
- **Documentation:** Providing comprehensive documentation for ease of deployment and maintenance.
- **Offline Storage:** Utilizing local storage capabilities to ensure data persistence between sessions.

Chapter 4

Hardware / Software used in Project

Hardware Requirements

1. Computer / Laptop
    - For developing and testing the application.
2. Web Browser
    - To run and test the web application (e.g., Chrome, Firefox, Edge).
3. Storage Device (Optional)
    - For saving project files and backups (e.g., USB drive, external hard disk).

Software Requirements

1. Text Editor / IDE
    - Visual Studio Code (VS Code) or any other code editor for writing HTML, CSS, and JavaScript.
2. Web Browser
    - Google Chrome, Mozilla Firefox, or any modern browser to run and debug the application.
2. Operating System
    - Windows, macOS, or Linux for development and testing.
4. Version Control (Optional)
    - Git for managing versions of your project.
5. HTML, CSS, and JavaScript
    - Core front-end technologies for designing and implementing the project.
6. Local Storage
    - Browser-based storage for offline data persistence.

Functional Requirements

- **User Input and Score Management:**
- The system allows users to input and update match details such as team names, players, runs scored, wickets taken, and overs bowled.
- **• Dynamic Score Updates:**
- The application updates and displays scores in real-time as the user inputs or modifies data.
- **Offline Functionality:**
- The system must work without internet connectivity using local storage or caching mechanisms to save and retrieve data.
- **User Interface:**
- The system provides a clean, interactive interface to display match details, including team names, scores, and overs.
- **Data Persistence:**
- Match data persists locally so that users can reopen the application without losing the information.
- **Reset/Start New Match:**
- The user should be able to reset the current score to start a new match or clear previous data.
-

Non-Functional Requirements

- **Performance:**
- The application must load quickly and respond instantly to user inputs without delays.
- **Usability:**
- The interface should be simple, user-friendly, and intuitive for all users, even those with limited technical knowledge.
- **Portability:**
- The system should run on any device with a web browser (e.g., desktops, laptops, tablets).
- **Reliability:**
- The application must function correctly without crashes, ensuring offline capabilities are consistent.
- **Compatibility:**
- The system must be compatible with popular web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
- **Maintainability:**
- The code should be clean, modular, and easy to update or extend for future improvements.
- **Security:**

- Local data should be stored securely in the browser to prevent unauthorized access or accidental loss.
- **Scalability (Optional):**
- The system should allow adding more features in the future, like player statistics or match history.

Chapter 5 Project Flow

Project Flow

1. **Initialization:**
    - Users launch the application in a browser.
    - The initial interface displays options for creating a new match or resuming an existing one.
2. **Match Setup:**
    - Users input team names, match type, and other relevant details.
    - Match configurations are saved for future reference.
3. **Score Entry:**
    - Users manually enter scores, overs, wickets, and player details.
    - Real-time updates are displayed on the scoreboard.
4. **Data Management:**
    - Users can reset, edit, or correct match data as needed.
5. **Scoreboard Display:**
    - Real-time scores, player statistics, and match progress are displayed dynamically.
6. **Session Management:**
    - Match data is saved in local storage for session continuity.

7. **Match Summary:**
    - Upon match completion, a detailed summary is generated.
    - Users can export or review match data.
8. **Application Closure:**
    - Users close the application, and data is securely saved for future access.
9. **Error Handling:**
    - The system identifies and prevents invalid data entries.
    - Alerts notify users of errors during score entry.
10. **Performance Analysis:**

- Statistical data of players and teams are compiled.
- Performance reports can be generated and exported.

11. **Customization Options:**

- Users can adjust settings for match formats (e.g., T20, ODI, Test matches).
- Visual themes and scoreboard layouts can be customized.

12. **Backup and Restore:**

- Users can manually back up match data.
- Data can be restored in case of accidental loss.

13. **User Training Module:**

- Tutorials and guides are available for new users.
- Step-by-step assistance ensures smooth onboarding.

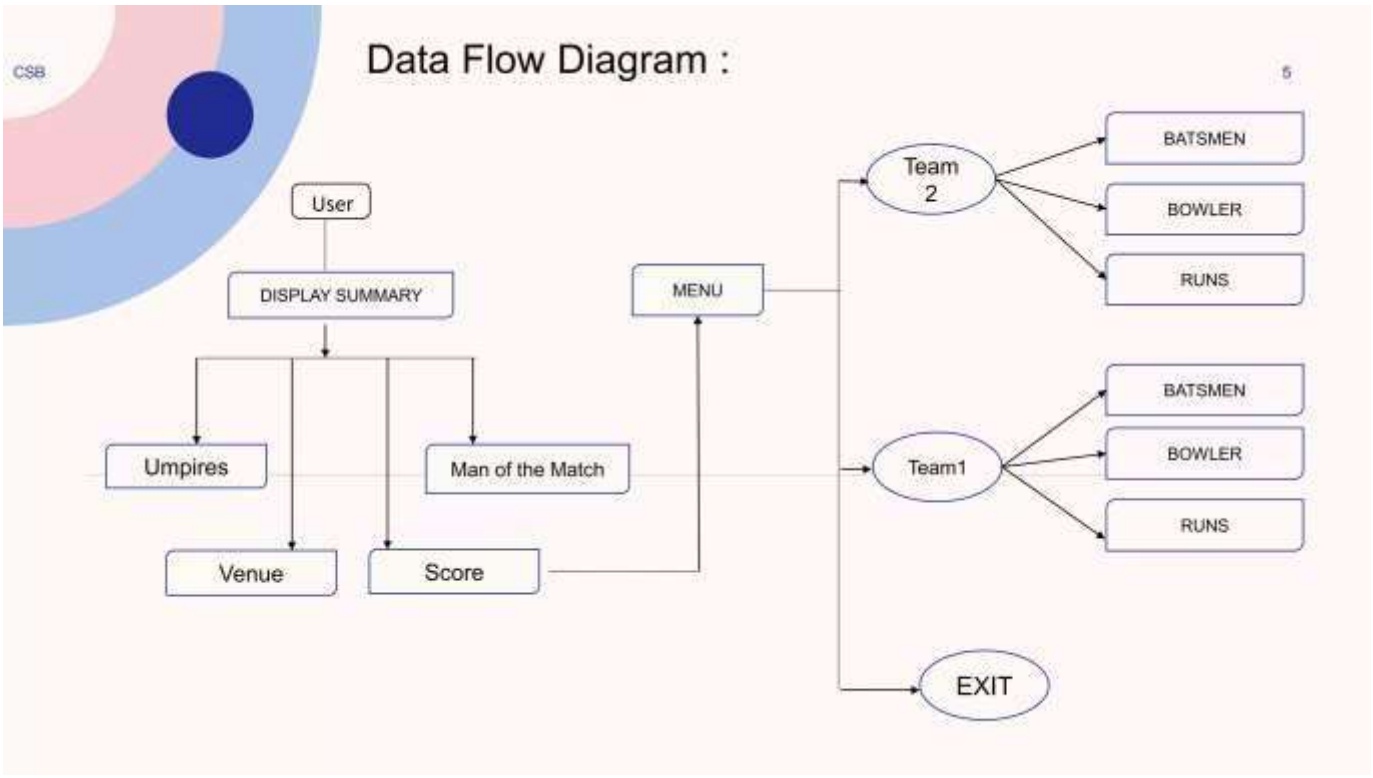
14. **Multi-Device Access:**

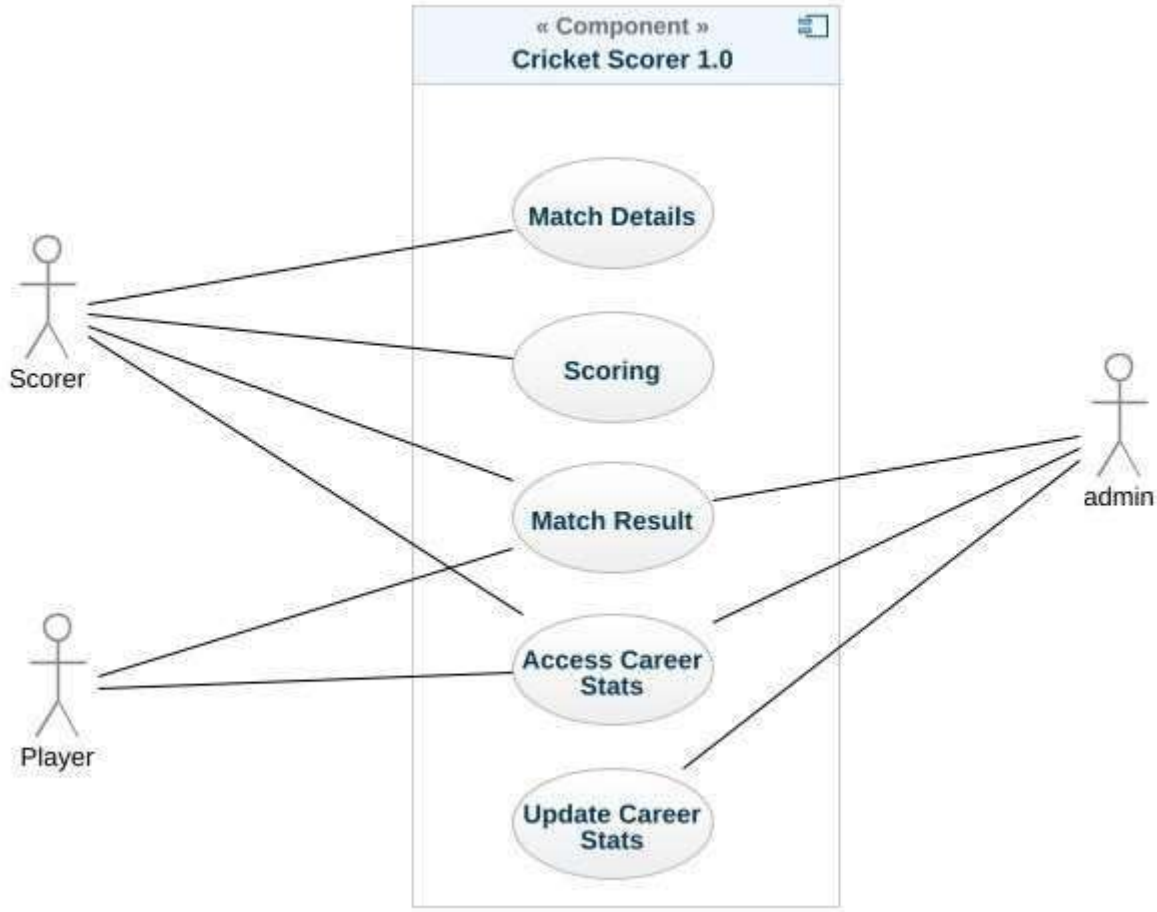
- The application supports access across desktops, tablets, and mobile devices.

15. **Future Enhancements:**

- Provisions for integrating additional features in future updates.
- Feedback from users is collected for continuous improvement.

Data flow diagram:





Sequence Diagram

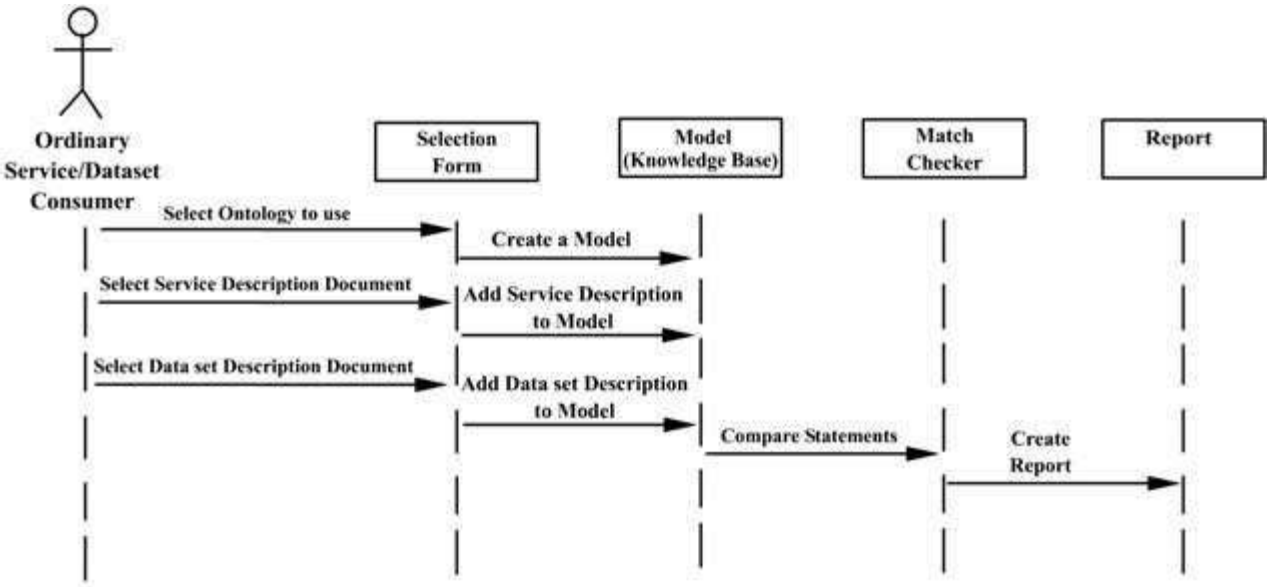
The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Purpose of a Sequence Diagram

To model high-level interaction among active objects within a system.

To model interaction among objects inside a collaboration realizing a use case.

It either models' generic interactions or some certain instances of interaction.



Add Instance

Chapter 6

Project Outcome

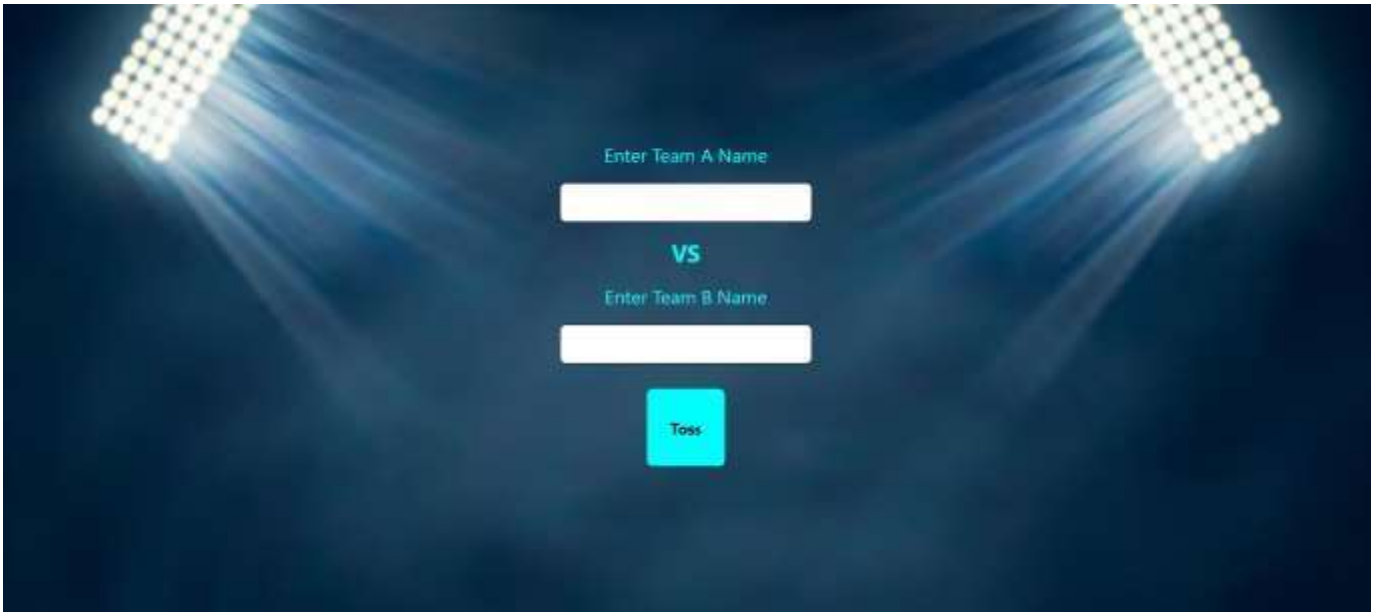
Project Outcomes

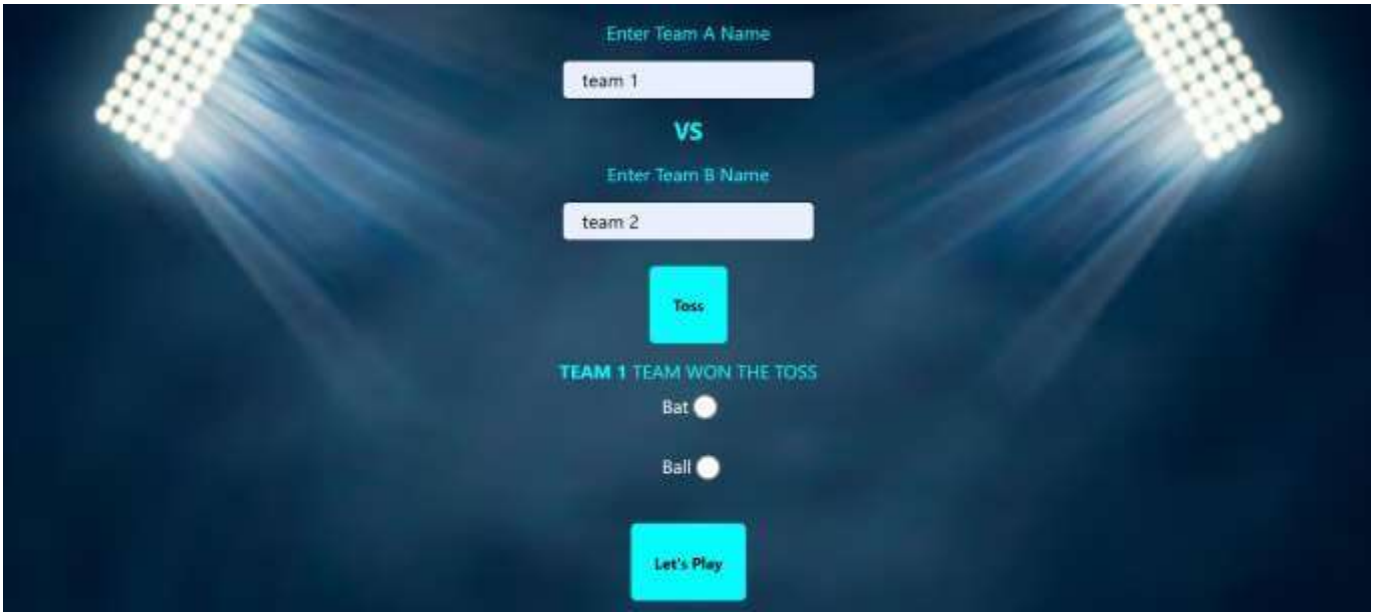
- **Reliable Offline Functionality:** Successfully displays and manages cricket scores without internet connectivity.
- **Enhanced User Experience:** Provides an intuitive and visually appealing interface for users of all technical levels.
- **Real-Time Score Updates:** Ensures scores and statistics are updated dynamically during matches.
- **Efficient Data Management:** Enables easy entry, editing, and resetting of match data.
- **Match Summary Reports:** Generates detailed match summaries for review and future reference.
- **Data Persistence:** Ensures match data is securely stored locally and can be retrieved across sessions.
- **Customization Options:** Allows users to configure match settings based on different formats.
- **Accessibility Across Devices:** Ensures compatibility on desktops, tablets, and mobile devices.
- **Improved Performance Analysis:** Facilitates tracking and analyzing player and team performance.
- **Cost-Effective Solution:** Reduces reliance on expensive or complex scoring software.
- **Scalability:** Supports both small and large tournaments effectively.
- **Error Prevention:** Minimizes errors with robust validation and user feedback systems.

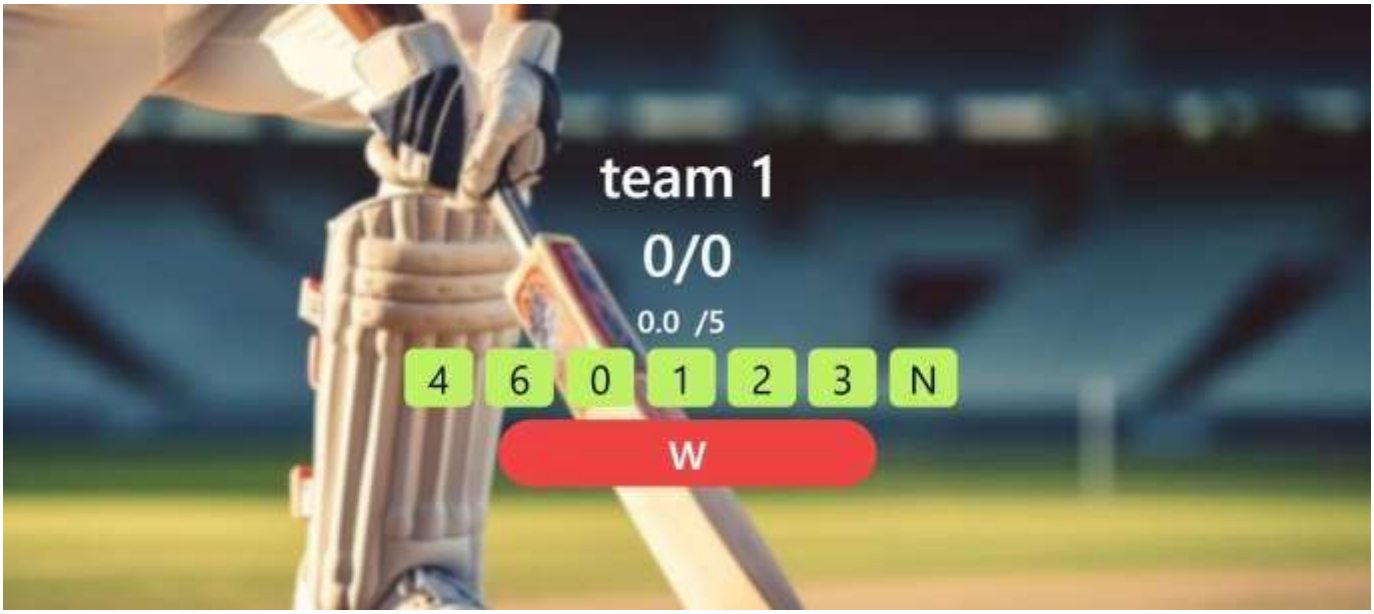
- **Secure Data Storage:** Safeguards sensitive match data from unauthorized access.

- **Future-Ready Design:** Allows integration of additional features and improvements.

- **User Training Support:** Provides easy-to-follow documentation and tutorials for end-users.


Enter Team A Name
VS
Enter Team B Name
Toss


Enter Team A Name
team 1
VS
Enter Team B Name
team 2
Toss
TEAM 1 TEAM WON THE TOSS
Bat
Ball
Let's Play


team 1
0/0
0.0 /5
4 6 0 1 2 3 N
W


target:120
team 2
0/0
0.0 /5
4 6 0 1 2 3 N
W

References

1. W3Schools. "HTML, CSS, and JavaScript Documentation." Available at: https://www.w3schools.com
2. Mozilla Developer Network (MDN). "Web Technologies Documentation." Available at: https://developer.mozilla.org
3. Cricket Scoring Standards and Practices. (2023). Published by Cricket Official Guidelines.
4. Stack Overflow Community Discussions on Offline Web Applications.
5. Frontend Mentor Challenges: Offline Web Solutions.
6. OpenAI. "Dynamic Frontend Web Development Insights."