CRICKET LIVE SCORE

A PROJECT REPORT

Submitted By

Rakshit Tyagi
(University Roll no. 2100290140111)
Tarun Kumar
(University Roll no. 2100290140141)
Shashank Tyagi
(University Roll no. 2100290140124)
Arnav Singh
(University Roll no. 2100290140036)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Vipin Kumar

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206(Feb 2023)

DECLARATION

The objective of the project is to display live cricket score. In this project, live cricket data is fetched from the network, and it will be shown in our project. You’ll get all the live match details from this project like team name, live score, location, time.

Signature

TARUN KUMAR (University Roll no. 2100290140141)

ARNAV SINGH (University Roll no. 2100290140036)

SHASHANK TYAGI (University Roll no. 2100290140124)

RAKASHIT TYAGI (University Roll no. 2100290140111)

Department of Computer Applications

KIET Group of Institutions, Ghaziabad
Uttar Pradesh – 201206

CERTIFICATE

Certified that **TARUN KUMAR (University Roll no. 2100290140141), ARNAV SINGH (University Roll no. 2100290140036), SHASHANK TYAGI (University Roll no. 2100290140124), RAKASHIT TYAGI (University Roll no. 2100290140111),** has carried out the research work presented in this project report entitled "**Cricket Live Score**" for the award of Master of Computer Applications from KIET Group of Institutions under my supervision. The project report embodies results of original work, and studies are carried out by the students themselves and the contents of the thesis do not form the basis for the award of any other degree to the candidates or to anybody else from this or any other University/Institution.

Date:

TARUN KUMAR (University Roll no. 2100290140141)
ARNAV SINGH (University Roll no. 2100290140036)
SHASHANK TYAGI (University Roll no. 2100290140124)
RAKSHIT TYAGI (University Roll no. 2100290140111)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr Arun Tripathi Head, Department of Computer Applications KIET Group of Institutions, Ghaziabad

ABSTRACT

The cricket live score project is a web-based application that provides live cricket scores, commentary, and statistics to cricket fans all over the world. The project aims to provide a user-friendly platform for cricket fans to follow their favorite teams and players in real-time.

The project includes a user interface that displays live scores and statistics, as well as historical data on matches, players, and teams. The application also includes features such as live streaming, user account management, and payment gateway integration for premium features.

The application uses third-party APIs to retrieve live cricket scores and statistics, which are processed and stored in a database for later retrieval. The user interface is designed to be responsive and accessible from multiple devices, including desktops, laptops, and mobile phones.

Overall, the cricket live score project aims to provide cricket fans with an interactive and engaging platform to follow their favorite sport in real-time, with access to the latest scores, commentary, and statistics at their fingertips.

ACKNOWLEDGMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vipin Kumar** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi** Sir Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at 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 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.

TARUN KUMAR (University Roll no. 2100290140141)

ARNAV SINGH (University Roll no. 2100290140036)

SHASHANK TYAGI (University Roll no. 2100290140124)

RAKSHIT TYAGI (University Roll no. 2100290140111)

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

The cricket live score project is a web-based application that provides users with real-time updates on cricket matches. The application is designed to be user-friendly, easy to use, and accessible from any device with an internet connection.

1.1.1 Features:

Live Scores: The application provides users with live scores of crickets matches in real-time. Users can view the scorecard, including details of each ball bowled, wickets taken, runs scored, and other relevant statistics.

Commentary: The application provides live commentary on the matches, giving users a detailed understanding of the game's progress, important moments, and noteworthy events.

Statistics: The application provides users with detailed statistics on the matches, including player statistics, team statistics, and overall match statistics. Users can also view historical data and compare the performance of players and teams.

Responsive Design: The application is designed to be responsive, which means it can adapt to any screen size and device type, providing an optimal viewing experience for all users.

Technical Details:

The application is built using modern web development technologies, including HTML, CSS, JavaScript, and Node.js. The application uses a third-party API to fetch real-time data on cricket matches, which is then displayed on the user interface. The application uses a database to store user information, including account details and preferences.

1.1.2 Benefits:

The cricket live score project provides cricket fans with an easy way to stay up to date with their favorite teams and players. The real-time updates and detailed statistics allow fans to follow the game closely and gain a deeper understanding of the sport. The user account feature allows users to personalize their experience and stay connected with the application.

Overall, the cricket live score project is a useful application for cricket fans, providing them with live updates, commentary, and statistics, all in one place.

1.2 PROJECT SCOPE

The scope of the cricket live score is to create a web-based application that allows cricket fans to access live scores, commentary, and statistics for ongoing cricket matches. The application will be designed to be user-friendly, easy to navigate, and accessible from any device with an internet connection.

The application will provide users with live cricket scores of ongoing matches in real-time. The live scores will be updated frequently to keep users informed of the match's progress.

The application will provide users with live commentary on the ongoing matches, giving them a detailed understanding of the game's progress, important moments, and noteworthy events.

The application will provide users with detailed statistics on the matches, including player statistics, team statistics, and overall match statistics. Users can also view historical data and compare the performance of players and teams.

Users can create an account on the application, which allows them to personalize their experience. They can choose their favorite teams and players, receive alerts for specific matches, and access their past activity.

The application will be designed to be responsive, which means it can adapt to any screen size and device type, providing an optimal viewing experience for all users.

The application will include live video streaming of cricket matches, allowing users to watch the game live.

Overall, the project scope is to provide a comprehensive web-based application that allows cricket fans to access live scores, commentary, and statistics for ongoing cricket matches, as well as live video streaming and user account management through an admin module. the application will be designed to be user-friendly and accessible from any device with an internet connection.

1.3 HARDWARE/ SOFTWARE USED IN THE PROJECT.

1.3.1 Hardware Requirement for development –

Input Devices - All basic input devices like keyboard, mouse, etc.

Output Devices - All basic output devices like printer, monitor, etc.

Secondary storage devices - HDD - 60GB or above.
Back-up **-** Flash Drive, CD/DVD, cloud storage etc.

Internal components -

- ▪ RAM - 512 MB (Minimum)
- ▪ Processor - Intel Pentium 4 Processor or above

1.3.2 Hardware Requirement for User –

Input Devices - All basic input devices like keyboard, mouse, etc.

Output Devices - All basic output devices like printer, monitor, etc.

1.3.3 Software Requirement for development –

- **System Software –**
  - ▪ Operating system (Windows XP or above)
- **Web Technologies-**
  - ▪ HTML
  - ▪ CSS
  - ▪ Bootstrap
  - ▪ JavaScript
- **Frameworks –**
  - ▪ React

1.3.4 Software Requirement for User -

- **System Software –**
  - ▪ Operating system (Windows XP or above)
- **Web Browser-** ▪ Microsoft Edge, Google Chrome

CHAPTER 2
LITERATURE REVIEW

2.1 ABSTRACT

The cricket live score project is a web-based application that provides live cricket scores, commentary, and statistics to cricket fans all over the world. The project aims to provide a user-friendly platform for cricket fans to follow their favorite teams and players in real-time. The project includes a user interface that displays live scores and statistics, as well as historical data on matches, players, and teams. The application also includes features such as live streaming, user account management, and payment gateway integration for premium features. The application uses third-party APIs to retrieve live cricket scores and statistics, which are processed and stored in a database for later retrieval. The user interface is designed to be responsive and accessible from multiple devices, including desktops, laptops, and mobile phones. Overall, the cricket live score project aims to provide cricket fans with an interactive and engaging platform to follow their favorite sport in real-time, with access to the latest scores, commentary, and statistics at their fingertips.

2.2 INTRODUCTION

Cricket and football are the most watched sports in the world after soccer and enjoys a multi-million-dollar industry. There is remarkable interest in simulating cricket and football and more importantly in displaying the live score of cricket match which is played in three formats namely test match, one day international and T20 match and the different football matches. There are number of fans of both of these sports and want to know live updates of the matches. Currently there are number of apps and websites for live match scores for individual sports. Fans faces problems open the number of browsers and different number of apps for matches. So, there is a need of one application which can show the live scores of both the sports football and cricket. In this system we proposed an application which shows the live scores of both cricket and football in the single window, simply by selecting the option i.e., cricket or football and today’s match. Live scores are a type of service offered by many sports-related websites and broadcasters as well as online sports betting operators. The idea of live scores is to provide real time information about sports results from various disciplines. Live scores are usually free and are very popular among sports betting enthusiasts, as they allow viewing collected data on many sports events. In the past, live score services were only available on TV through teletext or on the radio. There are now many websites providing live scores. It is possible to follow live results of many events at the same time. Some sites provide additional information, such as a player list, card details, substitution, and an online chat where sports fans can gather and discuss the current event. Several sports organizations such Major league baseball and the National Football League have set up their own networks to deliver live scores via mobile phones.

2.3 LITERATURE REVIEW

Live scores are a type of service offered by many sports-related websites and broadcasters as well as online sports betting operators. The idea of live scores is to provide real time information about sports results from various disciplines. Live scores are usually free and are very popular among sports betting enthusiasts, as they allow viewing collected data on many sports events. In the past, live score services were only available on TV through teletext or on the radio. There are now many websites providing live scores. It is possible to follow live results of many events at the same time. Some sites provide additional information, such as a player list, card details, substitution and an online chat where sports fans can gather and discuss the current event. Several sports organizations such Major league baseball and the National Football League have set up their own networks to deliver live scores via mobile phones.

2.4 NEED OF CRICKET LIVE SCORES

Live scores are a type of service offered by many sports-related websites and broadcasters as well as online sports operators. The idea of live scores is to provide real time information about sports results from various disciplines. Live scores are usually free and are very popular among sports betting enthusiasts, as they allow viewing collected data on many sports events. In the past, live score services were only available on TV through teletext or on the radio. There are now many websites providing live scores. It is possible to follow live results of many events at the same time. Some sites provide additional information, such as a player list, card details, substitution and an online chat where sports fans can gather and discuss the current event. Several sports organizations such Cricket Leagues and the National Football League have set up their own networks to deliver live scores via mobile phones.

2.5 CONCLUSION

This system is essential for showing the online scores of cricket match. It is a holistic approach as it takes in current input from user. The system work efficiently with an online score. Our system focuses on the performance on the live score of crickets.

2.6 REFRENCES

[1] Kou-Yuan Huang and Wen-Lung Chang. A neural network method for prediction of 2006 world cup football game. In The 2010 International Joint Conference on Neural Networks (IJCNN), pages 1 –8, july 2010.

[2] J. Hucaljuk and A. Rakipovic. Predicting football scores using machine learning techniques. In MIPRO, 2011 Proceedings of the 34th International Convention, pages 1623 –1627, may 2011.

[3] A. Joseph, Norman E. Fenton, and Martin Neil. Predicting football results using bayesian nets and other machine learning techniques. Knowledge-Based Systems., 19(7):544–553, 2006.

[4] A. Tsakonas, G. Dounias, S. Shtovba, and V. Vivdyuk. Soft computing-based result prediction of football games. In V. Hrytsyk, editor, The Ist International Conference on Inductive Modelling (ICIM'2002), pages 15–23, Lviv, Ukraine, 20-25 May 2002.

CHAPTER 3

FEASIBILITY STUDY

After doing the project Cricket Live Score, study and analyze all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

The feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

3.1 TECHNICAL FEASIBILITY

The technical feasibility of the project is high since there are many third-party APIs available that provide live cricket scores, commentary, and statistics. The application can be built using modern web development frameworks and tools.

3.2 OPERATIONAL FEASIBILITY

No doubt the proposed system is fully GUI based very user-friendly and all inputs to be taken all self-explanatory even to a layman. Besides, proper training has been conducted to let them know the essence of the system to the users so that they feel comfortable with the new system. As far as our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

3.3 ECONOMICAL FEASIBILITY

This is a very important aspect to be considered while developing a project. We decided on the technology based on the minimum possible cost factor.

- All hardware and software costs must be borne by the organization.
- Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later-on running cost for the system.

3.4 LEGAL FEASIBILITY

The project must comply with legal requirements, including copyright laws, data privacy laws, and terms of service agreements with third-party APIs. Legal feasibility should be evaluated to ensure the project's compliance with applicable laws and regulations.

3.5 SCHEDULE FEASIBILITY

The schedule feasibility of the project will depend on the project's scope, the available resources, and the development team's expertise. A realistic timeline should be developed, including project milestones, to ensure the project's timely completion.

3.6 BEHAVIOURAL FEASIBILITY

Behavioral feasibility refers to the potential user acceptance and satisfaction of the cricket live score project. In terms of behavioral feasibility, the project is expected to be feasible as it addresses the needs and expectations of cricket fans all over the world. Cricket is a popular sport globally, and there is a significant demand for live scores, statistics, and commentary. The cricket live score project aims to fulfill this demand by providing users with a user-friendly and engaging platform to follow their favorite teams and players in real-time. The project includes features such as live streaming, user account management, and payment gateway integration for premium features, which enhances the user experience and provides added value to users. Moreover, the project incorporates a responsive and accessible user interface, which can be accessed from multiple devices, including desktops, laptops, and mobile phones. The project also uses third-party APIs to retrieve live cricket scores and statistics, which ensures that users have access to accurate and up-to-date information. Overall, the cricket live score project is expected to be behaviorally feasible, as it addresses the needs and expectations of cricket fans and provides a user-friendly and engaging platform for following cricket in real-time.

CHAPTER 4

DATABASE DESIGN

4.1 USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

The purpose of a use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as the other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from the other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modeled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

Used to gather the requirements of a system.

Used to get an outside view of a system.

Identify the external and internal factors influencing the system.

Show the interaction among the requirements actors.

4.1.1 Use case diagram components.

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

Actors: The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

System: A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

Goals: The result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

4.1.2 Use case diagram symbols and notation.

The notation for a use case diagram is straight forward and doesn't involve as many types of symbols as other UML diagrams.

Use cases: Horizontally shaped ovals that represent the different uses that a user might have.

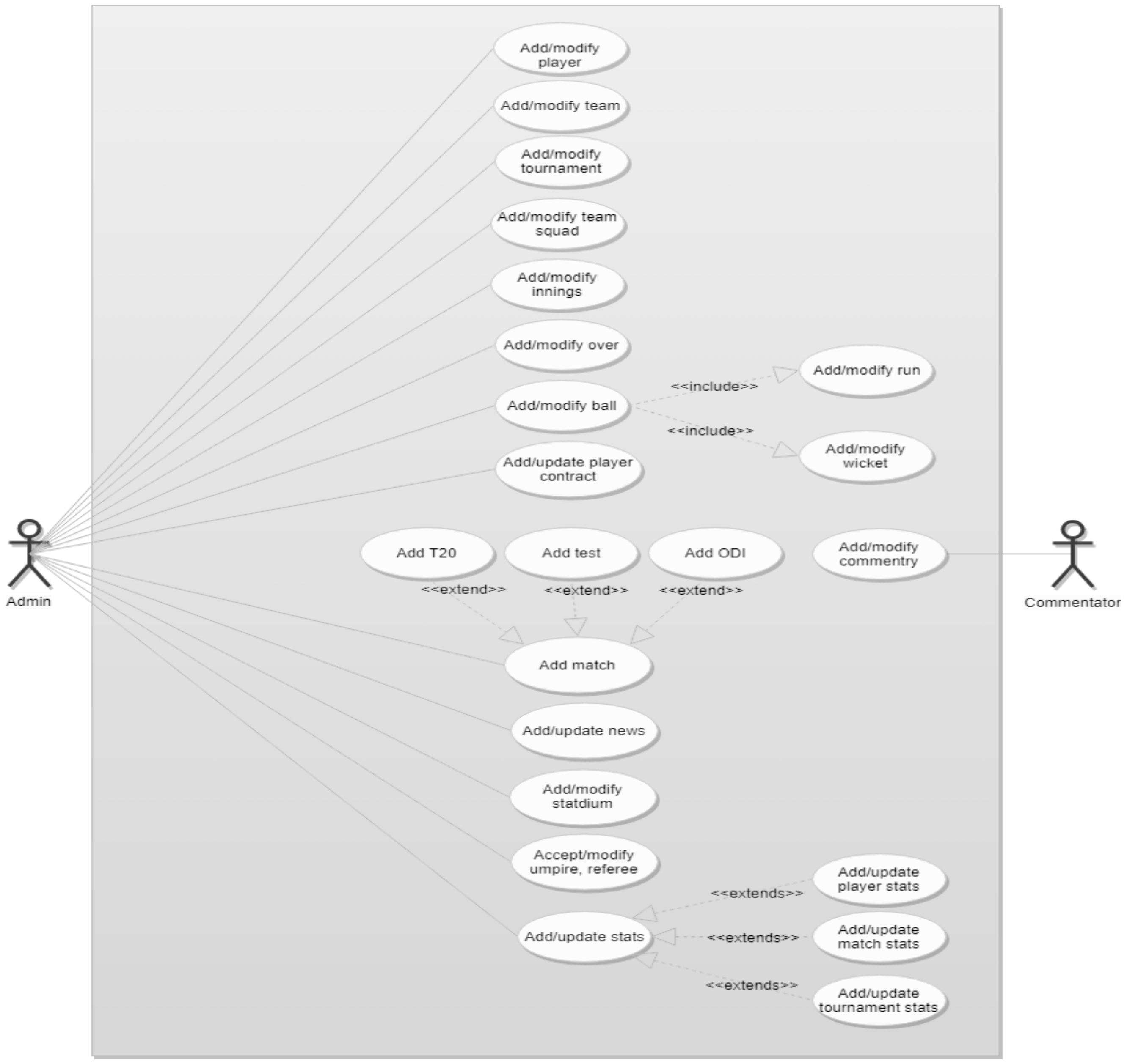
·**Actors:** Stick figures that represent the people employing the use cases.

·**Associations:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

·**System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.

·**Packages:** A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

4.1.3 USE CASE DIAGRAM



4.2 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

4.2.1 Purpose of Class Diagrams

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram.

Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top-level view.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represents the whole system.

The following points should be remembered while drawing a class diagram −

- The name of the class diagram should be meaningful to describe the aspect of the system.

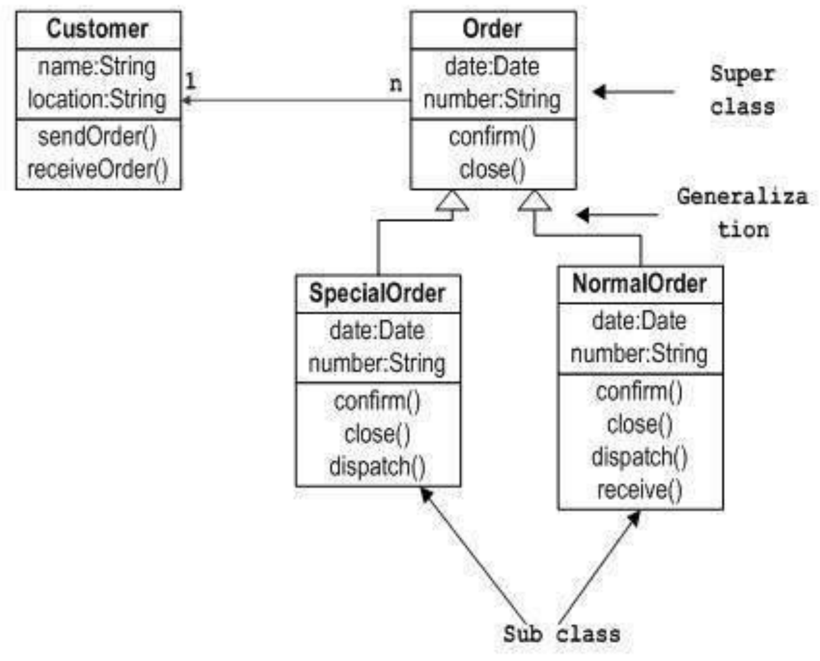
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified.
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing, it should be understandable to the developer/coder.
- Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

The following diagram is an example of an Order System of an application. It describes a particular aspect of the entire application.

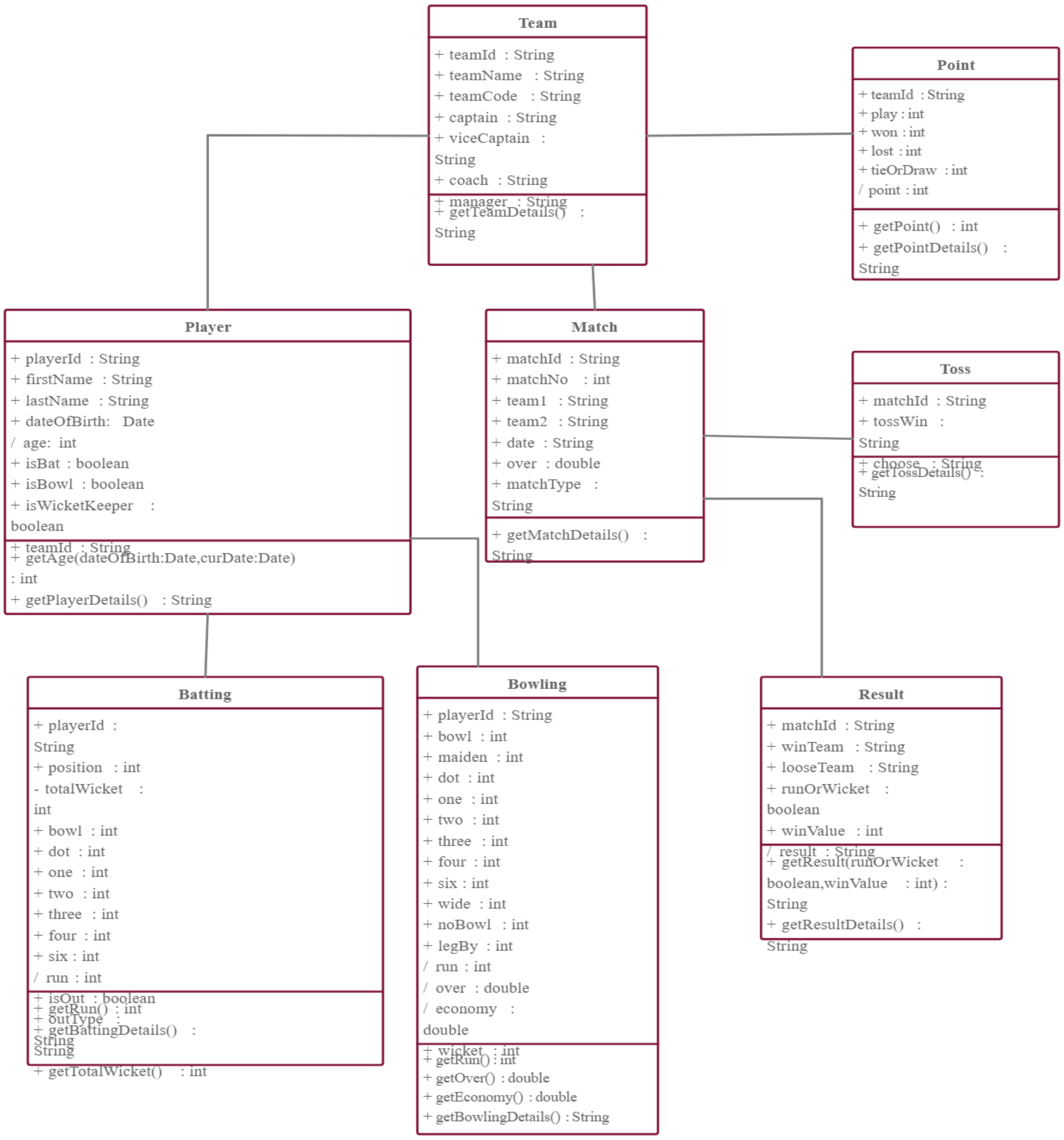
- First, Order and Customer are identified as the two elements of the system. They have a one-to-many relationship because a customer can have multiple orders.
- Order class is an abstract class, and it has two concrete classes (inheritance relationship) Special Order and Normal Order.
- The two inherited classes have all the properties as the Order class. In addition, they have additional functions like dispatch () and receive ().

The following class diagram has been drawn considering all the points mentioned above.

4.2.2 SAMPLE CLASS DIAGRAM



4.2.3 CLASS DIAGRAM



4.3 SEQUENCE DIAGRAM

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged intime sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario.

Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. For a particular scenario of a use case, the diagrams show the events that external actors generate, their order, and possible inter-system events. All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be done for the main success scenario of the use case, and frequent or complex alternative scenarios.

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

4.3.1 Sequence Diagram Notations –

i. Actors – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

ii. Lifelines – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

iii. Messages – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

iv. Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

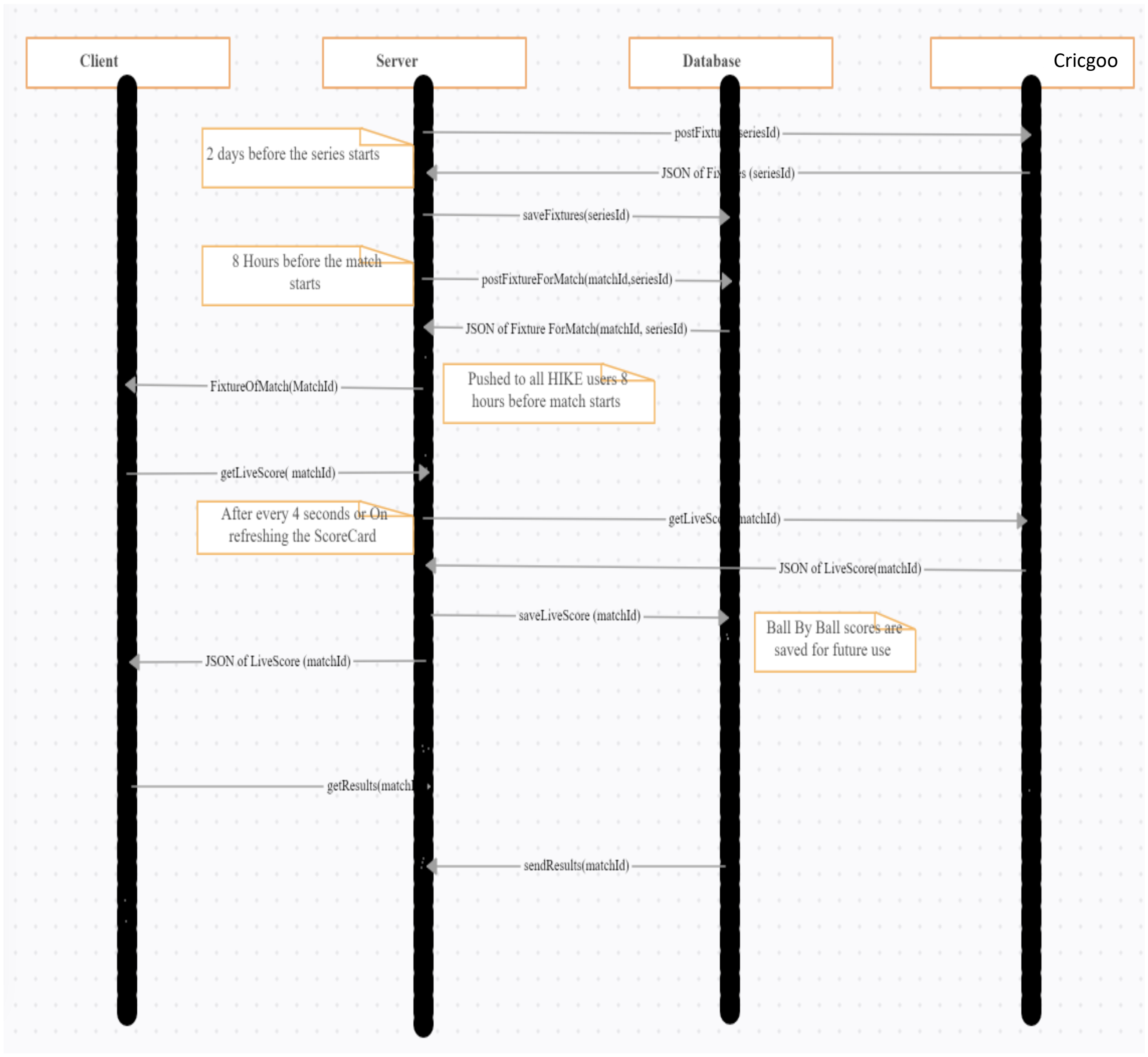
i. Used to model and visualize the logic behind a sophisticated function, operation, or procedure.

ii. They are also used to show details of UML use case diagrams.

iii. Used to understand the detailed functionality of current or future systems.

iv. Visualize how messages and tasks move between objects or component.

4.3.2 SEQUENCE DIAGRAM


Client
Server
Database
Cricgoo
postFixtu seriesId)
2 days before the series starts
JSON of Fi s (seriesId)
saveFixtures(seriesId)
8 Hours before the match starts
postFixtureForMatch(matchId,seriesId)
JSON of Fixture ForMatch(matchId, seriesId)
FixtureOfMatch(MatchId)
Pushed to all HIKE users 8 hours before match starts
getLiveScore(matchId)
After every 4 seconds or On refreshing the ScoreCard
getLiveSc matchId)
JSON of LiveScore(matchId)
saveLiveScore (matchId)
Ball By Ball scores are saved for future use
JSON of LiveScore (matchId)
getResults(match
sendResults(matchId)

4.4 STATE DIAGRAM

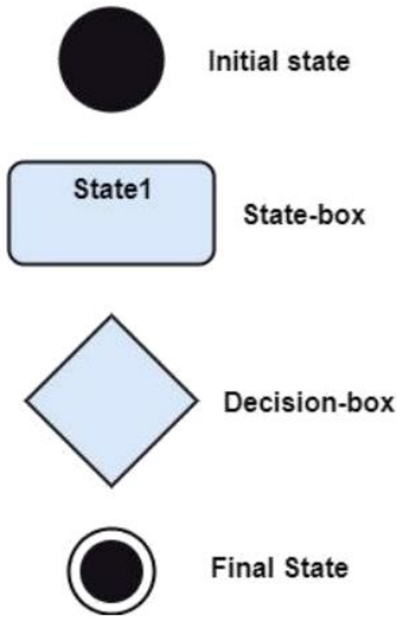
The state machine diagram is also called the State chart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system.

It tends out to be an efficient way of modeling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

Notation of a State Machine Diagram

Following are the notations of a state machine diagram enlisted below:

4.4.1 FIFURE OF NOTATIONS



a. Initial state: It defines the initial state (beginning) of a system, and it is represented by a black filled circle.

b. Final state: It represents the final state (end) of a system. It is denoted by a filled circle present within a circle.

c. Decision box: It is of diamond shape that represents the decisions to be made on the basis of an evaluated guard.

d. Transition: A change of control from one state to another due to the occurrence of some event is termed as a transition. It is represented by an arrow labeled with an event due to which the change has ensued.

e. State box: It depicts the conditions or circumstances of a particular object of a class at a specific point of time. A rectangle with round corners is used to represent the state box.

4.4.2 How to Draw a State Machine Diagram?

The state machine diagram is used to portray various states underwent by an object. The change in one state to another is due to the occurrence of some event. All of the possible states of a particular component must be identified before drawing a state machine diagram.

The primary focus of the state machine diagram is to depict the states of a system. These states are essential while drawing a state transition diagram. The objects, states, and events due to which the state transition occurs must be acknowledged before the implementation of a state machine diagram.

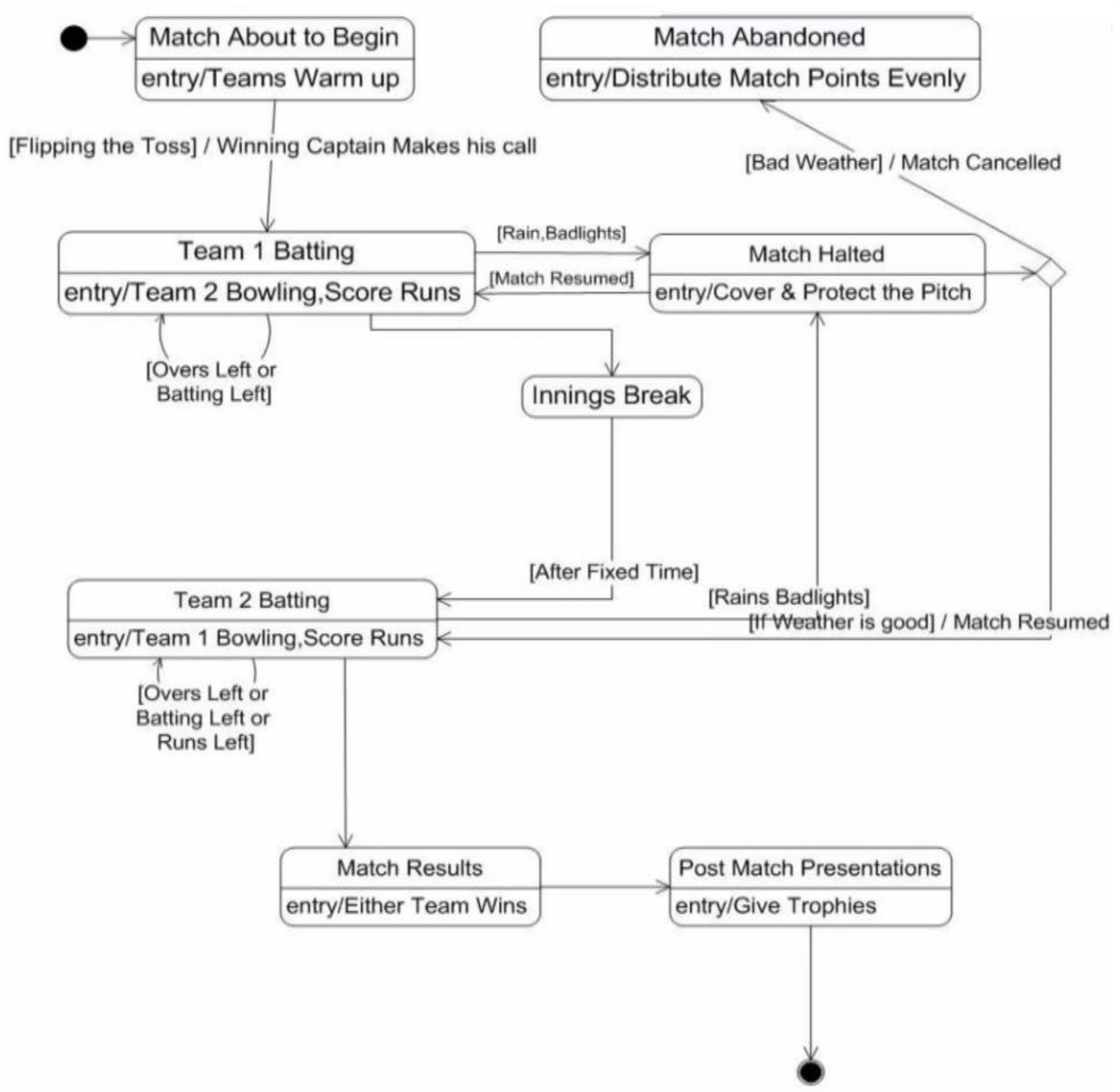
Following are the steps that are to be incorporated while drawing a state machine diagram:

1. A unique and understandable name should be assigned to the state transition that describes the behavior of the system.

2. Out of multiple objects, only the essential objects are implemented.

3. A proper name should be given to the events and the transition.

4.4.3 STATE DIAGRAM


Match About to Begin
entry/Teams Warm up
Match Abandoned
entry/Distribute Match Points Evenly
[Flipping the Toss] / Winning Captain Makes his call
[Bad Weather] / Match Cancelled
Team 1 Batting
entry/Team 2 Bowling,Score Runs
[Rain,Badlights]
[Match Resumed]
Match Halted
entry/Cover & Protect the Pitch
[Overs Left or Batting Left]
Innings Break
[After Fixed Time]
[Rains Badlights]
[If Weather is good] / Match Resumed
Team 2 Batting
entry/Team 1 Bowling,Score Runs
[Overs Left or Batting Left or Runs Left]
Match Results
entry/Either Team Wins
Post Match Presentations
entry/Give Trophies

CHAPTER 5

DESIGN

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; user interface design (UI design); authoring, including standardized code and proprietary software; user experience design (UX design); and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term "web design" is normally used to describe the design process relating to the front-end (client side) design of a website including writing markup. Web design partially overlaps web engineering in the broader scope of web development. Web designers are expected to have an awareness of usability and be up to date with web accessibility guidelines.

Here's why implementing these practices is important for your e-Commerce website:

- **To provide the best customer experience possible:** A great customer experience is about making it simple for customers to buy products from your website, while also making it a great pleasure for them. These practices will help you create a website experience your customers will love.

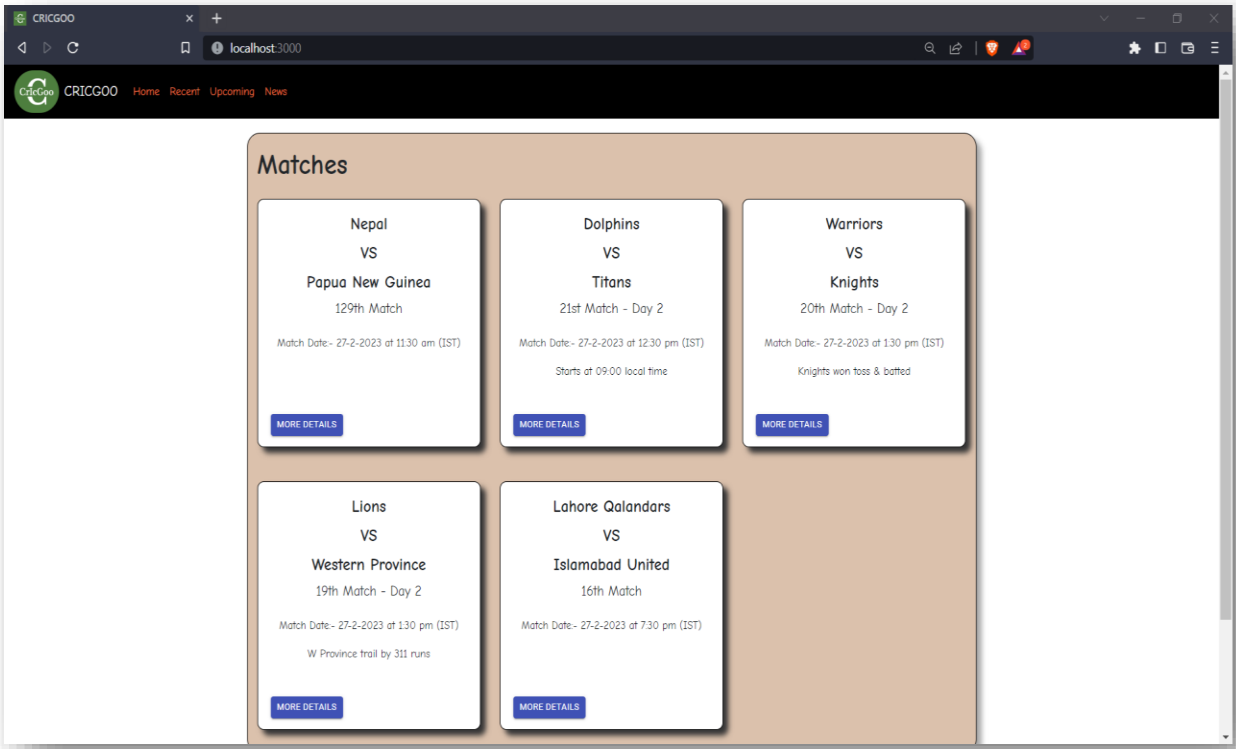
- **To boost sales through conversion rate optimization:** Great design will help you convert more website visitors into paying customers, boosting your sales and enhancing business performance.

- **To improve customer retention:** You don't want website visitors or existing customers to leave your website and buy the same product from someone else. A great design will help you grab the attention of your website visitors and encourage your customers to be loyal towards your brand.

- **To reinforce your brand:** Great design will speak volumes about who you are, what your brand stands for, and help you be perceived appropriately. At the end of the day, design is about great communication!

- **To build customer relationships:** People tend to trust websites that are designed well, and therefore, will want to engage more with them. This helps with building great customer relationships with trust at its foundation.

SCREENSHOTS



Matches

Nepal
VS
Papua New Guinea
129th Match
Match Date:- 27-2-2023 at 11:30 am (IST)
MORE DETAILS

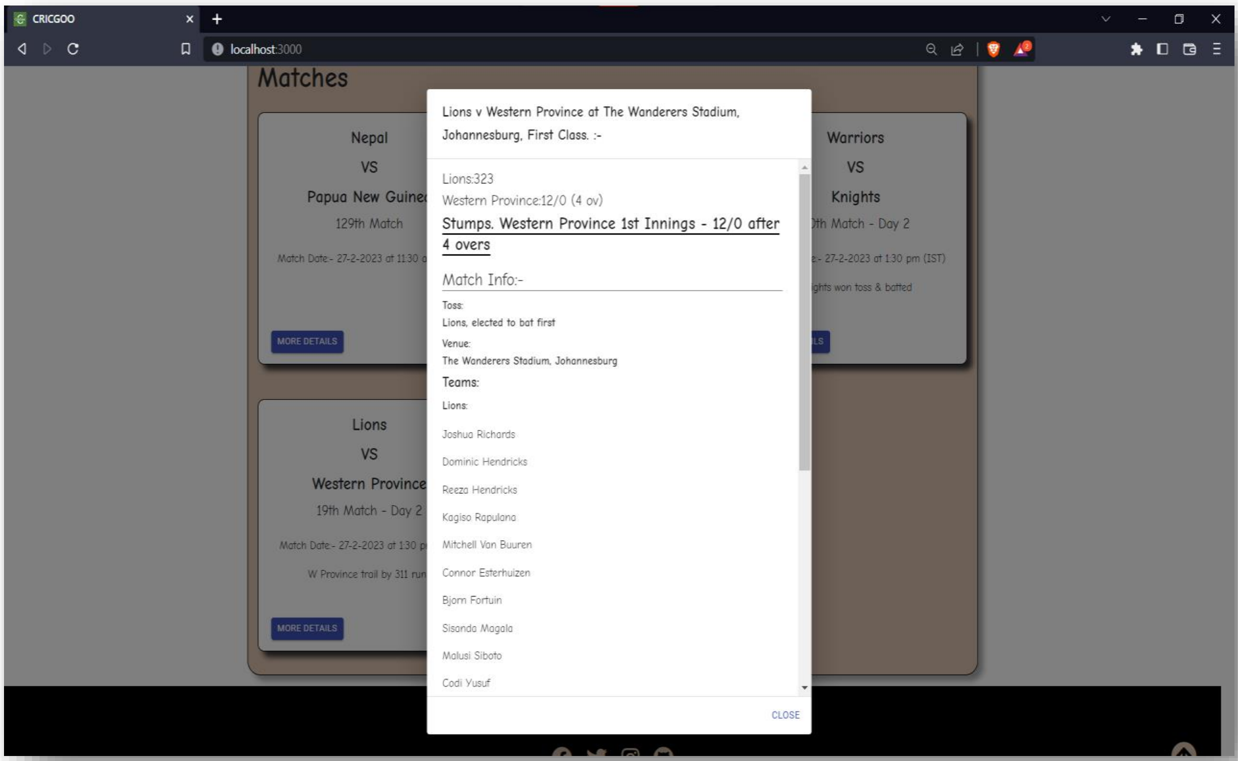
Dolphins
VS
Titans
21st Match - Day 2
Match Date:- 27-2-2023 at 12:30 pm (IST)
Starts at 09:00 local time
MORE DETAILS

Warriors
VS
Knights
20th Match - Day 2
Match Date:- 27-2-2023 at 1:30 pm (IST)
Knights won toss & batted
MORE DETAILS

Lions
VS
Western Province
19th Match - Day 2
Match Date:- 27-2-2023 at 1:30 pm (IST)
W Province trail by 311 runs
MORE DETAILS

Lahore Qalandars
VS
Islamabad United
16th Match
Match Date:- 27-2-2023 at 7:30 pm (IST)
MORE DETAILS

5.1 HOMEPAGE



Lions v Western Province at The Wanderers Stadium, Johannesburg, First Class. :-

Lions:323
Western Province:12/0 (4 ov)
Stumps. Western Province 1st Innings - 12/0 after 4 overs

Match Info:-

Toss:
Lions, elected to bat first

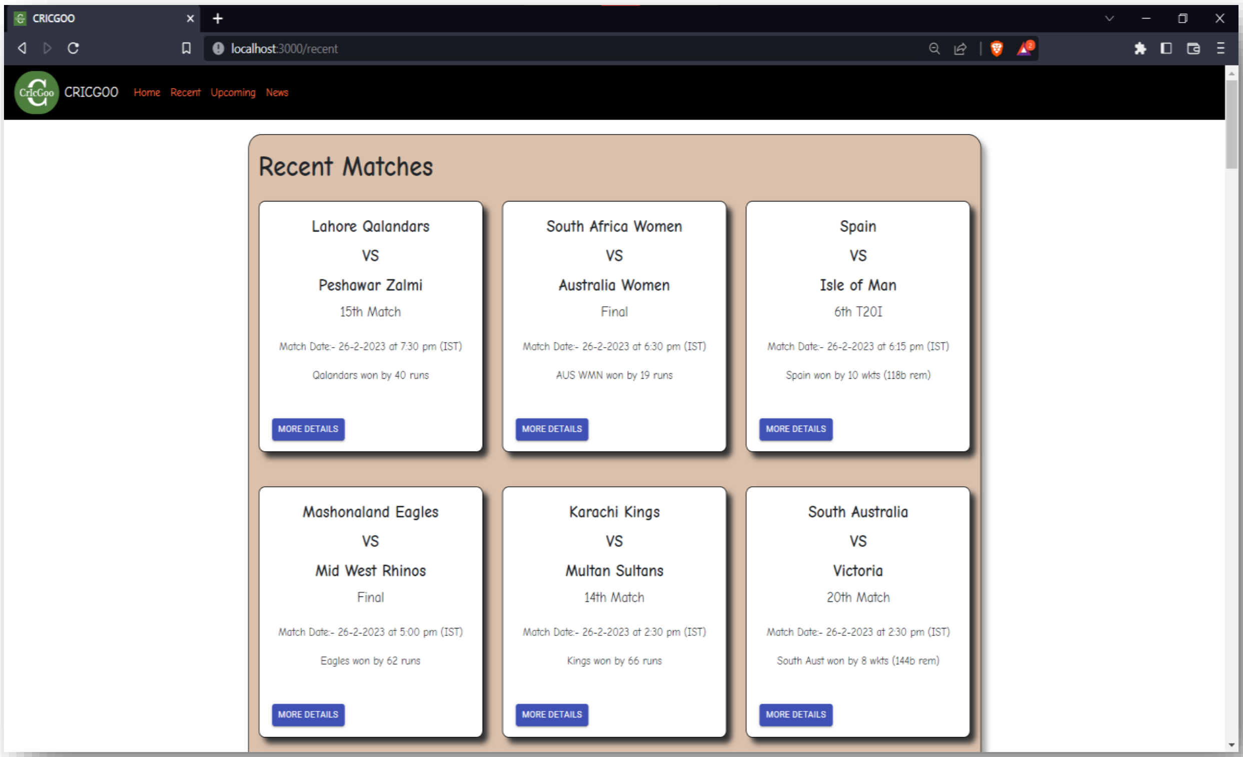
Venue:
The Wanderers Stadium, Johannesburg

Teams:

Lions:

Joshua Richards
Dominic Hendricks
Reeza Hendricks
Kagiso Rapulana
Mitchell Van Buuren
Connor Esterhuizen
Bjorn Fortuin
Sisanda Magala
Malusi Siboto
Codi Yusuf

5.2 ONGOING MATCH



Recent Matches

Lahore Qalandars
VS
Peshawar Zalmi
15th Match
Match Date:- 26-2-2023 at 7:30 pm (IST)
Qalandars won by 40 runs

South Africa Women
VS
Australia Women
Final
Match Date:- 26-2-2023 at 6:30 pm (IST)
AUS WMN won by 19 runs

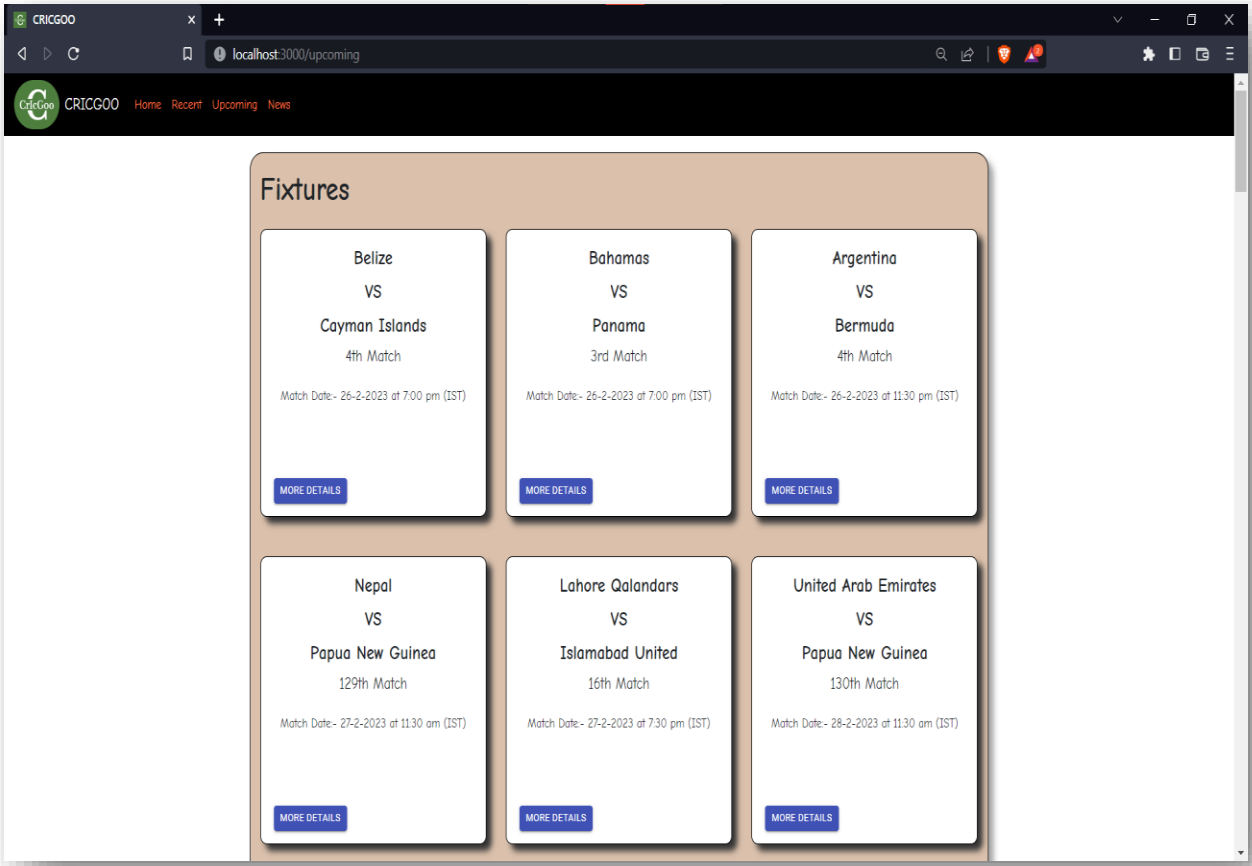
Spain
VS
Isle of Man
6th T20I
Match Date:- 26-2-2023 at 6:15 pm (IST)
Spain won by 10 wkts (118b rem)

Mashonaland Eagles
VS
Mid West Rhinos
Final
Match Date:- 26-2-2023 at 5:00 pm (IST)
Eagles won by 62 runs

Karachi Kings
VS
Multan Sultans
14th Match
Match Date:- 26-2-2023 at 2:30 pm (IST)
Kings won by 66 runs

South Australia
VS
Victoria
20th Match
Match Date:- 26-2-2023 at 2:30 pm (IST)
South Aust won by 8 wkts (144b rem)

5.3 RECENT MATCHES



Fixtures

Belize
VS
Cayman Islands
4th Match
Match Date:- 26-2-2023 at 7:00 pm (IST)

Bahamas
VS
Panama
3rd Match
Match Date:- 26-2-2023 at 7:00 pm (IST)

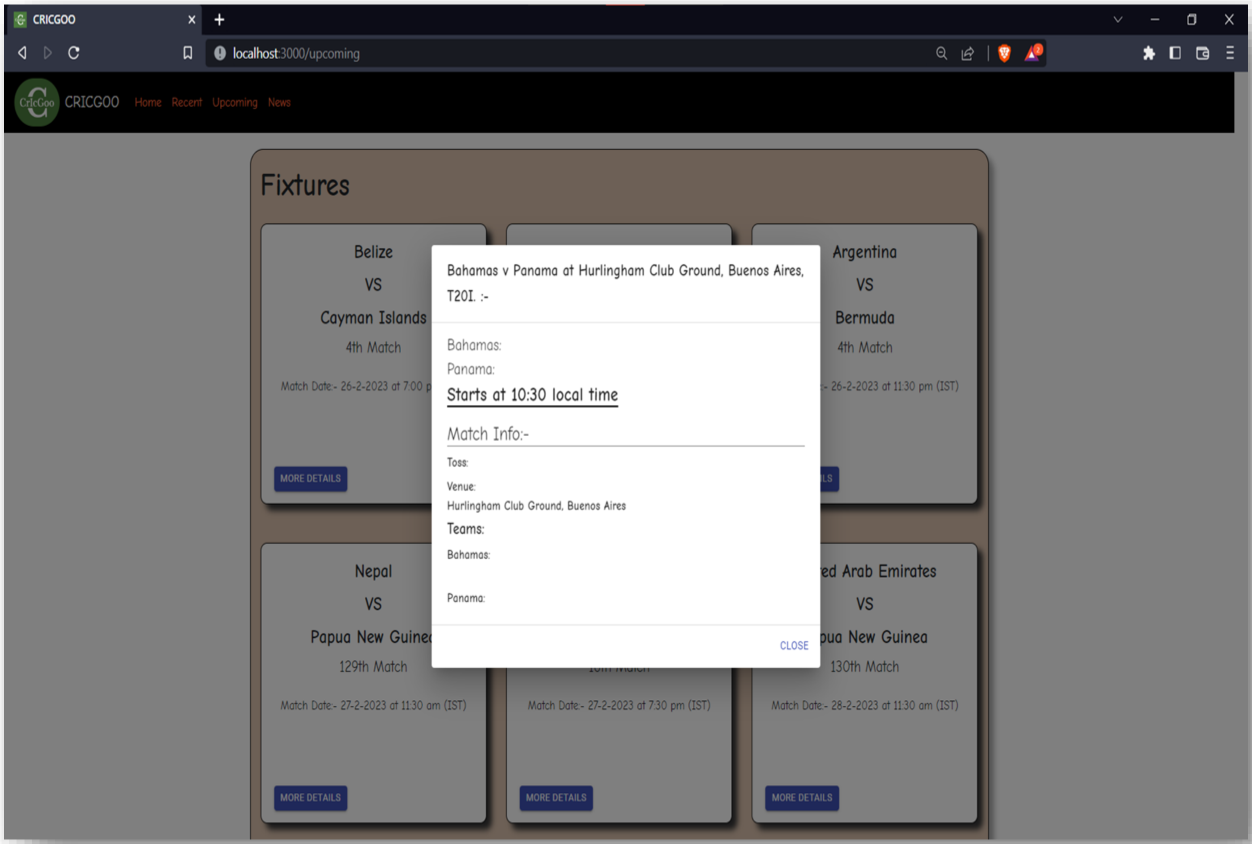
Argentina
VS
Bermuda
4th Match
Match Date:- 26-2-2023 at 11:30 pm (IST)

Nepal
VS
Papua New Guinea
129th Match
Match Date:- 27-2-2023 at 11:30 am (IST)

Lahore Qalandars
VS
Islamabad United
16th Match
Match Date:- 27-2-2023 at 7:30 pm (IST)

United Arab Emirates
VS
Papua New Guinea
130th Match
Match Date:- 28-2-2023 at 11:30 am (IST)

5.4 UPCOMING MATCHES FIXTURE



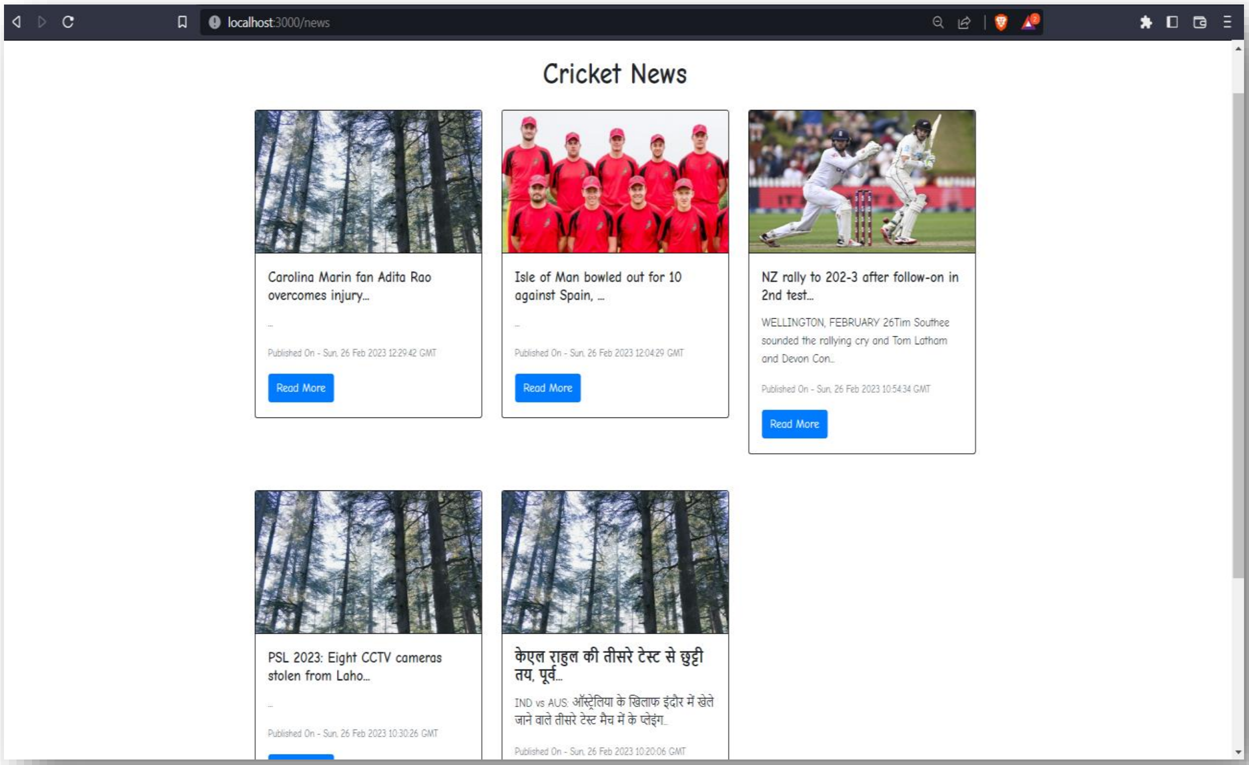
Bahamas v Panama at Hurlingham Club Ground, Buenos Aires, T20I. :-

Bahamas:
Panama:
Starts at 10:30 local time

Match Info:-
Toss:
Venue:
Hurlingham Club Ground, Buenos Aires
Teams:
Bahamas:

Panama:

5.5 UPCOMING MATCHES



Cricket News

Carolina Marin fan Adita Rao overcomes injury...

...

Published On - Sun, 26 Feb 2023 12:29:42 GMT

Isle of Man bowled out for 10 against Spain, ...

...

Published On - Sun, 26 Feb 2023 12:04:29 GMT

NZ rally to 202-3 after follow-on in 2nd test...

WELLINGTON, FEBRUARY 26Tim Southee sounded the rallying cry and Tom Latham and Devon Con...

Published On - Sun, 26 Feb 2023 10:54:34 GMT

PSL 2023: Eight CCTV cameras stolen from Laho...

...

Published On - Sun, 26 Feb 2023 10:30:26 GMT

केएल राहुल की तीसरे टेस्ट से छुट्टी तय, पूर्व...

IND vs AUS: ऑस्ट्रेलिया के खिलाफ इंदौर में खेले जाने वाले तीसरे टेस्ट मैच में के प्लेइंग...

Published On - Sun, 26 Feb 2023 10:20:06 GMT

5.6 CRICKET NEWS

CHAPTER 6

CODE

6.1 INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="logo.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="CRICGOO is cricket score showing react app which the score of
Current cricket matches as well as the upcoming and old cricket matches and
crciket news"
    />
    <link rel="apple-touch-icon" href="logo.png" />

    <link rel="manifest" href="logo.png" />
    <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons" />
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jI
W3" crossorigin="anonymous">
    <style>
      @import
url('https://fonts.googleapis.com/css2?family=Comic+Neue:wght@400;700&displ
ay=swap');
      </style>
    <script src="https://kit.fontawesome.com/602d5b86c0.js"
crossorigin="anonymous"></script>
    <script src="https://code.iconify.design/2/2.1.0/iconify.min.js"></script>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <title>CRICGOO </title>
  </head>
  <body>
    <div id="root"></div>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4
+1p" crossorigin="anonymous"></script>
  </body>
</html>
```

6.2 CARD.JS

```
import Carditem from "./Carditem.js";

import React, { useEffect, useState } from "react";

export default function Card(props) {

  const [Results, setResults] = useState([]);
  var today = new Date();
  var dd = String(today.getDate()).padStart(2, "0");
  var mm = String(today.getMonth() + 1).padStart(2, "0");
  var yyyy = today.getFullYear();
  today = yyyy + "-" + mm + "-" + dd;


  var a = `${props.category}`;


  useEffect(() => {

    if (a === "fixtures-by-date") {

      fetch(`https://cricket-live-data.p.rapidapi.com/${props.category}/${today}`, {
"method": "GET",
headers: {
          "X-RapidAPI-Key":
"a13885e6f8mshb6863976f9c74dcp15ff38jsnd774e276b914",
          "X-RapidAPI-Host": "cricket-live-data.p.rapidapi.com",
          },
})
        .then((res) => res.json())
        .then((data) => {

          setResults(data.results);
        })
        .catch((error) => {
          console.log("Error", error);
        });
    } else {
      fetch(`https://cricket-live-data.p.rapidapi.com/${props.category}`, {
"method": "GET",
headers: {
          "X-RapidAPI-Key":
"a13885e6f8mshb6863976f9c74dcp15ff38jsnd774e276b914",
          "X-RapidAPI-Host": "cricket-live-data.p.rapidapi.com",
          },
})

        .then((res) => res.json())
        .then((data) => {

          setResults(data.results);
        })

        .catch((error) => {
          console.log("Error", error);
        });
    }
  }, []);

  return (
    <>
      <div
        className="container"
        style={{
          backgroundColor: "#dbc1ac",

          boxShadow: "8px 8px 8px darkgray",
          borderRadius: "20px",
          border: "1px solid black",
          marginTop: "20px",
        }}
      >
        <h1 className="text-left" id="head" style={{ marginTop: "20px" }}>
          <b>{props.match}</b>
        </h1>

        <div className="row">
          {Results.map((element) => (
            <div className="col-md-4 mb-3">
              <Carditem
                home={element.home.name}
                away={element.away.name}
                sub={element.match_subtitle}
                result={element.result}
                id={element.id}
                date={element.date}
              />
            </div>
          ))}
        </div>
      </div>
    </>
  );
}
```

6.3 CARDITEM.JS

```
import React from "react";

import { Button } from "@material-ui/core";

import { useState } from "react";

import Dialog from "@material-ui/core/Dialog";

import DialogActions from "@material-ui/core/DialogActions";
import DialogContent from "@material-ui/core/DialogContent";
import DialogTitle from "@material-ui/core/DialogTitle";

import Typography from "@material-ui/core/Typography";

export default function Carditem(props) {
  var date = new Date(`${props.date}`);
  var time = date.toLocaleString("en-IN", {
    hour: "numeric",
    minute: "numeric",
    hour12: true,
  });

  var year = date.getFullYear();
  var month = date.getMonth() + 1;
  var day = date.getDate();
  var formatted = day + "-" + month + "-" + year;
  var acc = document.querySelectorAll(".accordion");
  var active = null;
  const [open, setOpen] = React.useState(false);


  const handleClickOpen = () => {
    setOpen(true);
  };
```

```
 const handleClose = () => {
   setOpen(false);
 };

 const [Away, setAway] = useState([]);
 const [Home, setHome] = useState([]);
 const [live, setlive] = useState([]);
 const [title, settitle] = useState([]);
 const [team, setteam] = useState([]);
 const [team1, setteam1] = useState([]);

 const set = (id) => {

   fetch(`https://cricket-live-data.p.rapidapi.com/match/${id}`, {
     method: "GET",
     headers: {
       "X-RapidAPI-Key":
"a13885e6f8mshb6863976f9c74dcp15ff38jsnd774e276b914",
       "X-RapidAPI-Host": "cricket-live-data.p.rapidapi.com",
     },
   })
     .then((res) => res.json())
     .then((data) => {

       setAway(data.results.fixture.away.name);
       setHome(data.results.fixture.home.name);
       setlive(data.results.live_details.match_summary);
       settitle(data.results.fixture);
       setteam(data.results.live_details.teamsheets.away);
       setteam1(data.results.live_details.teamsheets.home);

     })

     .catch((error) => {
       console.log("Error", error);
     });
```

```
};

return (
  <>
    <div className="my-3">
      <div
        className="card"
        style={{

          border: "1px solid black",
          borderRadius: "10px",
          boxShadow: "8px 8px 8px #333334",
          height: "350px",
        }}
      >
        <div className="card-body">
          <h4
            className="card-title"
            style={{
              display: "flex",
              justifyContent: "space-around",
              flexWrap: "wrap",
            }}
          >
            <b>{props.home}</b>
          </h4>
          <h4
            className="card-title"
            style={{
              display: "flex",
              justifyContent: "space-around",
              flexWrap: "wrap",
            }}
          >
            <b>VS</b>
          </h4>
          <h4
            className="card-title"
            style={{
              display: "flex",
              justifyContent: "space-around",
              flexWrap: "wrap",
            }}
          >
            <b>{props.away}</b>
          </h4>

          <h5
            style={{
              display: "flex",
              justifyContent: "space-around",
              flexWrap: "wrap",
            }}
          >
            <p style={{ textAlign: "center" }}>{props.sub}</p>
          </h5>
          <p style={{ textAlign: "center" }}>
            Match Date:- {formatted} at {time} (IST)
          </p>
          <p className="card-text" style={{ textAlign: "center"
}}>{props.result}</p>

          <Button
            size="small"
            variant="contained"
            color="primary"
            style={{ position: "absolute", bottom: "15px", left: "20px" }}
            onClick={() => {

              handleClickOpen();
              set(props.id);
            }}
          >
```

```
  More Details
</Button>
<Dialog
  onClose={handleClose}
  aria-labelledby="customized-dialog-title"
  open={open}
>
  <DialogTitle id="customized-dialog-title" onClose={handleClose}>
    <p
      className="text"
      style={{
        fontFamily: "Comic Neue, cursive",

        marginBottom: "0px",
        fontWeight: "bolder",
      }}
    >
      {title.match_title} :-
    </p>
  </DialogTitle>

  <DialogContent
    dividers
    style={{
      fontFamily: "Comic Neue, cursive",
    }}
  >
    <Typography gutterBottom>
      <h5
        style={{
          marginBottom: "5px",
          fontFamily: "Comic Neue, cursive",
        }}
      >
        {Home}:{live.home_scores}
      </h5>
```

```
</Typography>

                <Typography gutterBottom>
                  <h5
                    style={{
                      marginBottom: "5px",
                      fontFamily: "Comic Neue, cursive",
                    }}
                  >
                    {Away}:{live.away_scores}
                  </h5>
                </Typography>
                <Typography
                  gutterBottom
                  style={{ marginBottom: "10px !important" }}
                >
                  <h4>

                    <b
                      style={{
                        marginBottom: "10px",
                        fontFamily: "Comic Neue, cursive",
                        borderBottom: "2px solid black",
                      }}
                    >
                      {live.status}
                    </b>
                  </h4>
                </Typography>

                <h4 style={{ marginTop: "20px",borderBottom:"1px solid gray"
}}>Match Info:-</h4>

                <Typography
                  gutterBottom
                  style={{ fontFamily: "Comic Neue, cursive" }}
                >

<b style={{ fontFamily: "Comic Neue, cursive" }}>
                    Toss:<br></br>
                    {live.toss}
                  </b>
                </Typography>

                <Typography gutterBottom>
                  <b style={{ fontFamily: "Comic Neue, cursive" }}>
                    Venue:<br></br>
                    {title.venue}
                  </b>
                </Typography>

                <h5 style={{ fontFamily: "Comic Neue, cursive" }}>
                  <b>Teams:</b>
                </h5>

                <Typography gutterBottom>
                  <b style={{ fontFamily: "Comic Neue, cursive" }}>{Home}:</b>
                  {team1.map((element) => (
                    <h6
                      style={{
                        fontFamily: "Comic Neue, cursive",
                        marginBottom: "0px",
                      }}
                    >
                      <br></br>
                      {element.player_name}
                    </h6>
                  ))}
                </Typography>
                <Typography gutterBottom>
                  <b style={{ fontFamily: "Comic Neue, cursive" }}>
                    <br></br>
                    {Away}:
```

```
                </b>
                {team.map((element) => (

<h6
                    style={{
                      fontFamily: "Comic Neue, cursive",
                      marginBottom: "0px",
                    }}
                  >
                    <br></br>
                    {element.player_name}

                  </h6>
                ))}
              </Typography>
            </DialogContent>
            <DialogActions>
              <Button autoFocus onClick={handleClose} color="primary">
                Close
              </Button>
            </DialogActions>
          </Dialog>
        </div>
      </div>
    </div></>
);
}
```

6.4 FOOTER.JS

```
import React from 'react'

export default function Footer() {
  return (

        <div>
      <footer className='foo'>
        <div className="text-center">
          <h3 className="font-weight-bold" style={{ color: "#dbc1ac" }}>Follow
Us</h3>
          <div className="d-flex flex-row justify-content-center">
            <span><a href="/"><i className="fab fa-facebook fa-
2x"></i></a></span>
            <span><a href="/"><i className='fab fa-twitter fa-2x'></i></a></span>
            <span><a href="/"><i className='fab fa-instagram fa-
2x'></i></a></span>
            <span><a href="https://github.com/ArnavS1999" ><i className='fab fa-
github fa-2x'></i></a></span>
          </div>
          <p>&copy; 2022 Copyright By CRICGOO</p>
          <div className="top">
            <a href="#top"><i className="fas fa-arrow-circle-up fa-2x"></i></a>
          </div>

        </div>
      </footer>
     </div>

  )
}
```

6.5 NAVBAR.JS

```
import React from 'react'

import lojo from './logo.png'

import {Link} from 'react-router-dom'

export default function Navbar(props) {
    return (
        <div>

                <nav className="navbar navbar-expand-md navbar-dark">
                    <img src={lojo} className="logo" alt="logo" height="60px"
width='70px'></img>
                    <Link className="navbar-brand" to="/">CRICGOO</Link>
                    <button className="navbar-toggler collapsed" type="button" data-
toggle="collapse" data-target="#collapsibleNavbar">
                        <span className="toggler-icon top-bar"></span>
                        <span className="toggler-icon middle-bar"></span>
                        <span className="toggler-icon bottom-bar"></span>
                    </button>
                    <div className="collapse navbar-collapse" id="collapsibleNavbar">
                        <ul className="navbar-nav">
                            <li className="nav-item">
                                <Link className="nav-link" to="/">Home</Link>
                            </li>

                            <li className="nav-item">
                                <Link className="nav-link" to="/recent">Recent</Link>
                            </li>
                            <li className="nav-item">
                                <Link className="nav-link"
to="/upcoming">Upcoming</Link>
                            </li>
                            <li className="nav-item">
        <Link className="nav-link" to="/news">News</Link>
                        </li>
                    </ul>

                </div>
            </nav>
        </div>
    )
}
```

6.6 NEWS.JS

```
import React, {Component } from "react";

import Spinner from "./Spinner";


export class News extends Component {
  constructor () {
    super ();
    this.state = {
      results: [],
      loading: false,
      page: 0,
      totalResults: [],
    };
  }


  async componentDidMount() {

      let url =
`https://newsdata.io/api/1/news?apikey=pub_17843d8c8804ec739c2eb27bde875f0
84fc9c&q=cricket%20news `;
```

```
      this.setState({ loading: true });

      let data = await fetch(url);
      let parsedData = await data.json();

      this.setState({
        results: parsedData.results,
        totalResults: parsedData.totalResults,
        loading: false,
      });


  }
  previousclick = async () => {
    let url =
`https://newsdata.io/api/1/news?apikey=pub_3058bcd099932c95931004aa8f87093
ede70&country=in&q=cricket&page=$ {this.state.page - 1}`;
    this.setState({ loading: true });
    let data = await fetch(url);
    let parsed Data = await data.json();

    this.setState({
      page: this.state.page - 1,
      results: parsedData.results,
      loading: false,
    });
  };
  nextclick = async () => {
    if (this.state.page + 1 > Math.ceil(this.state.totalResults / 10)) {
    } else {
      let url =
`https://newsdata.io/api/1/news?apikey=pub_3058bcd099932c95931004aa8f87093
ede70&country=in&q=cricket&page=$ {this.state.page + 1}`;
      this.setState({ loading: true });
      let data = await fetch(url);

      let parsedData = await data.json();
```

```
this.setState({
       page: this.state.page + 1,
       results: parsedData.results,
       loading: false,
     });
   }
 };

 render () {
   return (
     <>



       <div className="container">



         <h1 className="text-center" id="head" style={{ marginTop: "20px" }}>
           <b>Cricket News</b>
         </h1>
         {this.state.loading && <Spinner />}
         <div className="row">
           {! this.state.loading &&
             this.state.results
               . filter ((_, index) => index % 2 === 0)
               . map((element) => {
                 return (
                   <div className="col-md-4 mb-3" key={element.link}>

                     <div className='my-3'>
               <div className="card" style= {{ border: "1px solid black" }}>



                   <div className="card-body">
               <h5 className="card-title"><b> {element.title ? element.title.slice(0,
45) : ""}...</b></h5>
```

```
                    <p className="card-text">{
                      element.description

        ? element.description.slice(0, 88)
                        : ""
                    }...</p>
                    <p className='card-text'><small className='text-
muted'>Published On - {new
Date(`${element.pubDate}`).toGMTString()}</small></p>
                    <a rel="noreferrer" href={element.link} target="_blank"
className="btn btn-primary">Read More</a>
                  </div>
                </div>
              </div>
                  </div>
                 );
                })}
            </div>
            <div className="container d-flex justify-content-between">
              <button
                type="button"
                disabled={this.state.page <= 0}
                className="btn btn-dark"
                onClick={this.previousclick}
              >
                {" "}
                &larr; Previous
              </button>
              <button
                type="button"
                disabled={
                  this.state.page + 2 > Math.ceil(this.state.totalResults / 10)
                }
                className="btn btn-dark"
                onClick={this.nextclick}
              >
                Next &rarr;
              </button>
            </div>
          </div>

        </>
      );
    }
}

export default News;
```

6.7 SPINNER.JS

```
import React, { Component } from 'react'
import spin from './load.gif'

export class Spinner extends Component {

    render() {
        return (
            <div className='text-center'>
                <img src={spin} alt='load'/>
            </div>
        )
    }
}

export default Spinner
```

6.8 APP.CSS

```
body{
  font-family: 'Comic Neue', cursive;

}

.navbar{
  background-color: #000000;
}
.logo {
  border-radius: 160px;
  margin-right: 8px;
}



.nav-link {
  color: rgba(255, 127, 80, 0.993) !important;
  transition: color .15s linear !important;
}

.nav-link:hover {
  color: rgba(74, 193, 230, 0.979) !important;
}

.navbar-toggler-icon {
  background-image: url("data:image/svg+xml;charset=utf8,%3Csvg viewBox='0 0
32 32' xmlns='http://www.w3.org/2000/svg'%3E%3Cpath stroke='rgba(255, 127,
80, 0.993)' stroke-width='2' stroke-linecap='round' stroke-miterlimit='10' d='M4
8h24M4 16h24M4 24h24'/%3E%3C/svg%3E") !important;
}



button:focus {
  outline: none!important;
}

.navbar-dark .navbar-toggler {

  border: none!important;
}

/*cross sign from toggler*/

.navbar-toggler:focus,

.navbar-toggler:active,
.navbar-toggler-icon:focus {

outline: none;
box-shadow: none;

border: 0;
}

/* Lines of the Toggler */
.toggler-icon{
width: 30px;
height: 3px;
background-color: rgba(255, 127, 80, 0.993);

display: block;
transition: all 0.2s;
}

/* Adds Space between the lines */
.middle-bar{
margin: 5px auto;

}
```

```
/* State when navbar is opened (START) */

.navbar-toggler .top-bar {
transform: rotate(45deg);
transform-origin: 10% 10%;
}

.navbar-toggler .middle-bar {
opacity: 0;
filter: alpha(opacity=0);
}

.navbar-toggler .bottom-bar {
transform: rotate(-45deg);
transform-origin: 10% 90%;


}
/* State when navbar is opened (END) */



/* State when navbar is collapsed (START) */
.navbar-toggler.collapsed .top-bar {
transform: rotate(0);
}


.navbar-toggler.collapsed .middle-bar {
opacity: 1;
filter: alpha(opacity=100);
}


.navbar-toggler.collapsed .bottom-bar {
transform: rotate(0);
}
```

```
footer{
  background: #000;
  padding-top: 2%;
  margin-top: 15px;
  position: relative;
}

footer span{
  padding: 0.3em 0.7em;
  color: #dbc1ac;
  transition: all .5s ease;
}
footer p{
  color: #dbc1ac;
  font-size: 20px;
}
footer .top a{
  position: absolute;
  right: 2%;
  top: 50%;
  color: #dbc1ac;
  font-size: 20px;
}
footer span a{
  color: #dbc1ac !important;
}
.panel{
  max-height: 0;
  overflow: hidden;}

.btn-warning{
  border-color: #000;
}
@media screen and (max-width: 992px) and (min-width: 768px){
    .card{
      height:380px !important;
    }}
```

6.9 APP.JS

```
import './App.css';
import Card from './component/Card';
import News from './component/News';
import Navbar from './component/Navbar';
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Routes
} from "react-router-dom";
import Footer from './component/Footer';
function App() {
  var today = new Date();
  var dd = String(today.getDate()).padStart(2, "0");
  var mm = String(today.getMonth() + 1).padStart(2, "0");
  var yyyy = today.getFullYear();
  today = yyyy + "-" + mm + "-" + dd;

  return (
    <>
    <Router><Navbar /
            <Switch>
            <Route exact path="/"><Card key="top" match="Matches"
category="fixtures-by-date"/></Route>
            <Route exact path="/recent"><Card  key="result" match="Recent
Matches" category="results" /></Route>
            <Route exact path="/upcoming"><Card key="fixture" match="Fixtures"
category="fixtures"  /></Route>
            <Route exact path="/news"><News key="News"/></Route>
            </Switch>
          </Router>

          <Footer/>
```

```
        </>
    );
  }

export default App;
```

6.10 APP.TEST.JS

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

6.11 INDEX.JS

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

CHAPTER 6

TESTING

Testing is a process of executing a program with the intent of finding bugs that makes the application fail to meet the expected behavior. System Analysis and Design process including Requirement Analysis, Business Solution Options, Feasibility Study, Architectural Design was discussed in previous chapter. Generally, Software bugs will almost always exist in any software module. But it is not because of the carelessness or irresponsibility of programmer but because of the complexity. Humans have only limited ability to manage complexity. This chapter discusses about the testing of the solution and implementation methodologies. Regardless of the development methodology, the ultimate goal of testing is to make sure that what is created does what it is supposed to do. Testing plays a critical role for assuring quality and reliability of the software. I have included testing as a part of development process. The test cases should be designed with maximum possibilities of finding the errors or bugs. Software Testing is the process of executing a program or system with the intent of finding errors. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions. Testing stage of the project can be explained as below and system was tested for all these stages. Various level of testing are as follows.

Testing Levels

6.1 UNIT TESTING:

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.

Unit testing involves testing individual modules or components of the application. In the cricket live score project, unit tests can be written for each of the following components:

- User Interface: Test the responsiveness and user-friendliness of the user interface, including page navigation, form submission, and error handling.
- Data Retrieval: Test the connectivity and authentication with third-party APIs, including testing for various scenarios such as invalid API keys or server downtime.
- Data Processing: Test the accuracy and reliability of data processing algorithms and formulas, including testing for edge cases and error handling.
- Data Storage: Test the database connectivity and schema design, including testing for data insertion, retrieval, and deletion.

Techniques of Unit Testing: -

6.1.1 White Box Testing

White box testing techniques analyze the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

6.1.2 Black Box Testing

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

6.1.3 Grey Box Testing

Grey box testing is a software testing method to test the software application with partial knowledge of the internal working structure. It is a **combination of black box and white box testing** because it involves access to internal coding to design test cases as white box testing and testing practices are done at functionality level as black box testing.

2.2 INTEGRATION TESTING:

Integration testing involves testing the interaction and communication between different components of the application. In the cricket live score project, integration tests can be written for the following components:

- User Interface and Server-Side Processing: Test the communication and data exchange between the user interface and the server-side processing components, including testing for data validation and error handling.
- Server-Side Processing and Data Storage: Test the communication and data exchange between the server-side processing component and the data storage component, including testing for data insertion, retrieval, and deletion.

Techniques of Integrating Testing :

6.2.1 Incremental Approach

In the Incremental Approach, modules are added in ascending order one by one or according to need. The selected modules must be logically related. Generally, two or more than two modules are added and tested to determine the correctness of functions. The process continues until the successful testing of all the modules.

6.2.2 Top-down Approach

The top-down testing strategy deals with the process in which higher level modules are tested with lower-level modules until the successful completion of testing of all the modules. Major design flaws can be detected and fixed early because critical modules tested first. In this type of method, we will add the modules incrementally or one by one and check the data flow in the same order.

6.2.3 Bottom – Up Approach

The bottom to up testing strategy deals with the process in which lower level modules are tested with higher level modules until the successful completion of testing of all the modules. Top level critical modules are tested at last, so it may cause a defect. Or we can say that we will be adding the modules from bottom to the top and check the data flow in the same order.

6.2.4 Big Bang Approach

In this approach, testing is done via the integration of all modules at once. It is convenient for small software systems, if used for large software systems identification of defects is difficult. Since this testing can be done after completion of all modules due to that testing team has less time for execution of this process so that internally linked interfaces and high-risk critical modules can be missed easily.

6.3 SYSTEM TESTING:

System testing involves testing the entire system as a whole to ensure that all components are working together as intended. In the cricket live score project, system tests can be written for the following scenarios:

- Live Score Display: Test the accuracy and reliability of live score display in real-time, including testing for server downtime and data retrieval errors.
- User Account Management: Test the functionality of user account management features, including user registration, login, logout, and password reset.
- Payment Gateway Integration: Test the integration and functionality of the payment gateway, including testing for secure payment processing and billing integration.
- Live Streaming: Test the live streaming functionality and user experience, including testing for video quality and latency.

Overall, the test cases for the cricket live score project should cover various scenarios and edge cases to ensure the application is reliable, accurate, and user-friendly.

Types of System Testing:

6.3.1 Performance Testing:

Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.

6.3.2 Load Testing:

Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.

6.3.3 Stress Testing:

Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.

6.3.4 Scalability Testing:

Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.

6.4 ACCEPTANCE TESTING

Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of Black Box testing where the number of required users involved testing the acceptance level of the system. It is the fourth and last level of software testing.

User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and check whether the application is working according to given business scenarios, real-time scenarios.

In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by the end-user or the customer.

6.5 SOFTWARE VERIFICATION AND VALIDATION

6.5.1 Software Verification

Verification testing includes different activities such as business requirements, system requirements, design review, and code walkthrough while developing a product. It is also known as static testing, where we are ensuring that **"we are developing the right product or not"**. And it also checks that the developed application fulfilling all the requirements given by the client.

Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is **Static Testing.**

Activities involved in verification:

6.5.1.1 Inspections

6.5.1.2 Reviews

6.5.1.3 Walkthroughs

6.5.1.4 Desk-checking

6.5.2 Software Validation

Validation testing is testing where tester performed functional and non-functional testing. Here **functional testing** includes Unit Testing (UT), Integration Testing (IT) and System Testing (ST), and **non-functional testing** includes User acceptance testing (UAT).

Validation testing is also known as dynamic testing, where we are ensuring that **"we have developed the product right."** And it also checks that the software meets the business needs of the client.

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e., it checks what we are developing is the right product. it is validation of actual and expected product. Validation is the **Dynamic Testing.**

Activities involved in validation:

6.5.2.1 Black box testing

6.5.2.2 White box testing

6.5.2.3 Unit testing

6.5.2.4 Integration testing

6.6 TEST PROCEDURE

A test procedure is a formal specification of test cases to be applied to one or more target program modules. Test procedures are executable. A process called the VERIFIER applies a test procedure to its target modules and produces an exception report indicating which test cases, if any, failed.

Test procedures facilitate thorough software testing by allowing individual modules or arbitrary groups of modules to be thoroughly tested outside the environment in which they will eventually reside. Test procedures are complete, self-contained, self-validating and execute automatically. Test procedures are a deliverable product of the software development process and are used for both initial checkout and subsequent regression testing of target program modifications.

Test procedures are coded in a new language called TPL (Test Procedure Language). The paper analyzes current testing practices, describes the structure and design of test procedures and introduces the Fortran Test Procedure Language.

6.7 TEST CASES

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

CHAPTER 8

CONCLUSION

In conclusion, the cricket live score project is a comprehensive and user-friendly platform that aims to provide real-time cricket scores, statistics, commentary, and live streaming to cricket fans worldwide. The project includes responsive user interface, which enhance the user experience and provide added value to users.

The project has been developed after a thorough feasibility study and has been designed to meet the needs and expectations of cricket fans worldwide. The project is expected to be behaviorally and economically feasible, with the potential to generate revenue through premium features such as live streaming and advanced statistics.

The development of the project involved various stages, including requirements gathering, design, development, testing, and deployment. The project has been developed using modern technologies and programming languages, ensuring that it is scalable, reliable, and secure.

Overall, the cricket live score project has the potential to become a popular and widely used platform for cricket fans worldwide, providing real-time updates and engaging content to users. The project is expected to contribute significantly to the growth and development of the cricket industry, providing an accessible and user-friendly platform for fans to follow their favorite teams and players in real-time.

CHAPTER 9

RECOMMENDATION

Based on the analysis and development of the cricket live score project, the following recommendations are made:

1-Continuous Improvement: The cricket live score project should be continuously improved and updated to ensure that it remains relevant and up to date with the latest cricket trends and technologies. This can be achieved through regular feedback from users and monitoring of the cricket industry.

2-Marketing: The project should be marketed aggressively to reach a wider audience and attract more users. This can be done through social media, advertising, and partnerships with other cricket-related platforms and organizations.

3-Security: The security of the platform should be continuously monitored and updated to prevent unauthorized access and protect user data.

4-User Engagement: The project should be designed to promote user engagement and encourage user participation, such as through interactive features, user-generated content, and social media integration.

5-Collaboration: The project should collaborate with other cricket-related platforms and organizations to provide users with a comprehensive and engaging cricket experience. This can be done through partnerships, data sharing, and integration of third-party APIs.

By implementing these recommendations, the cricket live score project can continue to grow and provide an engaging and comprehensive platform for cricket fans worldwide.

CHAPTER 10

BIBLIOGRAPHY

During the development of the cricket live score project, the following resources were referenced:

ESPN Cricinfo API: https://www.espncricinfo.com/apis/content/about

Live News API: https://newsdata.io/search-news

Live Score API: https://www.api-football.com/documentation#live-scores

Bootstrap Framework: https://getbootstrap.com/docs/5.1/getting-started/introduction/

CodeIgniter Framework: https://codeigniter.com/docs

RequirementsEngineering:
https://www.researchgate.net/publication/277201172_Requirements_Engineering_A_Review_of_Challenges_Techniques_and_Tools.

User Experience Design: https://www.nngroup.com/articles/definition-user-experience/

Test-Driven Development: https://martinfowler.com/bliki/TestDrivenDevelopment.html

These resources were essential in ensuring that the cricket live score project was developed using the best practices and technologies available. The references were used as guides for the project, ensuring that the platform is user-friendly, engaging, reliable, and secure.