A

PROJECT REPORT

On

“CricScorer”

BCA SEM 6 Project

BHAKTA KAVI NARSINH MEHTA UNIVERSITY

Submitted in partial fulfillment for the BCA ***6****th Semester*

SUBMITTED BY

Raj Solanki (03052000392)
Darshit Rakhasiya (03052001021)
Vishal Ganvani(03052000356)

Project Internal Guide Principal

(Prof.Kaushal Jani) (Dr. Vimal Parmar)

DR. SUBHASH

Dr. Subhash College
of Computer Science

Dr. Subhash College of Computer Science Junagadh

Certificate

Enrollment No.

This is to certify that the Project / practical work was satisfactorily carried out and hence submitted this report is the bonafide work of Mr.

____

Students of Bachelor of Computer Applications Sem 6 in the ______________________________ Laboratory of Dr. Subhash College of Computer Science Junagadh during the academic year 2022-23.

____________ **______________**

Project Guide **HOD**

____

Principal

DR. SUBHASH

Dr. Subhash College
of Computer Science

Dr. Subhash College of Computer Science Junagadh

Certificate

Enrollment No.

This is to certify that the Project / practical work was satis factorily carried out and hence submitted this report is the bonafide work of Mr. ____ Students of Bachelor of Computer Applications Sem 6 in the ________________________________ Laboratory of Dr. Subhash College of Computer Science Junagadh during the academic year 2022-23.

_____________ Project Guide

_______________ HOD

________________ Principal

DR. SUBHASH

Dr. Subhash College
of Computer Science

Dr. Subhash College of Computer Science Junagadh

Certificate

Enrollment No.

This is to certify that the Project / practical work was satis factorily carried out and hence submitted this report is the bonafide work of Mr. ____

Students of Bachelor of Computer Applications Sem 6 in the ______________________________ Laboratory of Dr. Subhash College of Computer Science Junagadh during the academic year 2022-23.

Project Guide **HOD**

Principal

ACKNOWLEDGEMENT

I take this opportunity to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report.

I am privileged to express our sense of gratitude to our respected **Pro. Kaushal Jani** sir unparalleled knowledge, moral fiber and judgment along with his know-how, was an immense support in completing the project.

I am also grateful to **Dr. Vimal Parmar**, the **Principal** and all other faculties of BCA for the brainwave and encouragement given. I take this opportunity also to thank our friends and contemporaries for their cooperation and compliance.

And lastly, I would like to pay my sincere regards to my parents and to our near and dear ones whose moral support and guidance helped to complete and submit the project well before the due date.

Name:	Darshit Rakhasiya
Enrollment:	03052001021
Name:	Raj Solanki
Enrollment:	03052000392
Name:	Vishal Ganvani
Enrollment:	03052000356

Bachelor of Computer Application
Dr. Subhash College of Computer
Science - Junagadh

DECLARATION

We Are students of College of Computer Science BCA for the Bachelor of Computer Application Sem .. And this project is not submitted to anywhere else in the market of use a system.

I also assure you that no copy of this project is share with any other student and developed by us with guidance of our faculties.

Raj Solanki
Darshit Rakhasiya
Vishal Ganvani

Name and Signature

TABLE OF CONTENTS

1
PROJECT PROFILE

- **Project Definition**
- **Project Scope**
- **Project Objective**

Project Profile		
•	**Enrollment No.**	03052001021 03052000392 03052000356
•	**Student Name**	Darshit Rakhasiya Raj Solanki Vishal Ganvani
•	**Project Title**	**CricScorer**
•	**Project Type**	Web Application
•	**Tools & Technology**	M-MongoDB E-ExpressJS R-ReactJS N-NodeJS
•	**Front – End**	ReactJS, ExpressjS, NodeJS
•	**Back – End**	MongoDB
•	**Team Member**	3

1.1 Project Definition :-

- CricScorer System is a web-based technology that manages and viwers to easily look and see the record

- In cricket, a scorer is someone appointed to record all runs scored, all wickets taken and, where appropriate, the number of overs bowled. In professional games, in compliance with Law 3 of the Laws of Cricket, two scorers are appointed, most often one provided by each team

- The scorers have no say in whether runs are scored, wickets taken or overs bowled. This is the job of the umpires on the field of play, who signal to the scorers in cases of ambiguity such as when runs are to be given as extras rather than credited to the batsmen, or when the batsman is to be awarded a boundary 4 or 6

- This project uses MERN, and it has a module that is user. All possible features such as validation, validation, security, etc., are considered.

- This project also includes system testing and validation, which ensures that your System is error-free and working as expected.

- ## Project Scope

- Using This Web-Based Application user can find out cricket score
- Using this application that can manage all the score of cricket regarding questions can solve to contact-us page.
- User can register to the site and also unregister from the site
- Verity of functionality and Validations are provide on this site.
- Best User experience on this site.
- Sort and simple navigate to the site.

- **Project Objective**

  - Main objective of this web application to make the Tourism process online and save tourist time to offline process
  - Help user to easily book an tour package without any physical process
  - All the information about the tour package are easily available for end-user
  - User can all the time updated about this site
  - For user security site provide verity of validations to protect the user data and make site attack protected
  - Site also provide Google captcha authentication to protect the site to abnormal activity to any form
  - Many more functionality are provided to this sitess

2

SYSTEM ANALYSIS AND SPECIFICATION

- **Existing System**
- **Limitations Of Existing System**
- **Feasibility Study**
- **Need Of New System**
- **Font End & Back End Tool**

- ## Existing System

- In the technical world all the work done by the web based system and internet
- Older System can also provide this type of services and functionality and facility but some enhancement are require
- Older system only provide package information not provide booking so user only gather information about the package and allocate sit by physical booking process and go to the office of particular travelling office
- Existing system are older by technology to not add some new feature to make advancement in the site

- **Limitation Of Existing System**

  - Limitation of technology

- ## Feasibility Study

  - After implementing the Zoo Management System project and investigating and analyzing all existing or required features of the System, the next task is to carry out a feasibility study. All projects can be run with unlimited resources and infinite time, and the feasibility study involves considering all possible ways to provide a solution to a given problem. The proposed solution should meet all user requirements and be flexible enough to adapt to future changes based on future needs quickly.

Economic Feasibility

- This is an essential aspect to consider when developing a project. We chose the technology based on the lowest possible cost factors.
  - The organization must bear all hardware and software costs.
  - Overall, it is estimated that the benefits that an organization will get from the proposed System will outweigh the initial and subsequent costs

Technical Feasibility

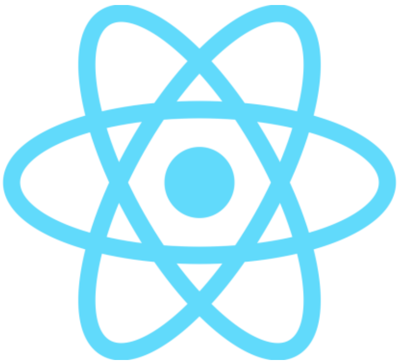
- This included investigating features, performance, and limitations that could affect the ability to achieve a tolerable system. To demonstrate this proof of concept, we have explored all the features provided by the System as described in the System. Requirements Specification (SRS), and everything is possible with the different types of front-end and backend platforms. I confirmed that there is.

- Operational feasibility Undoubtedly, the proposed System is completely GUI-based, very user-friendly, and all inputs are self-explanatory to the layman. In addition, proper training is provided to convey the essence of the System to the user, and the user can become accustomed to the new System. Our customers are comfortable and happy with our research because the System has reduced stress and strain.

- ## Font End & Back End Tool

- ### Front End Tools

  - **ReactJS**



- React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

- Declarative views make your code more predictable and easier to debug.

- React is a JavaScript library for building user interfaces. Learn what React is all about on our homepage or in the tutorial.

- React has been designed from the start for gradual adoption, and you can use as little or as much React as you need. Whether you want to get a taste of React, add some interactivity to a simple HTML page, or start a complex React-powered app, the links in this section will help you get started.
- If you're interested in playing around with React, you can use an online code playground. Try a Hello World template on CodePen, CodeSandbox, or Stackblitz.
- If you're interested in playing around with React, you can use an online code playground. Try a Hello World template on CodePen, CodeSandbox, or Stackblitz.
- React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with the user interface and rendering components to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

- ## Back End Tools

  - **ExpressJS**

- Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

- The original author, TJ Holowaychuk, described it as a Sinatra-inspired server,meaning that it is relatively minimal with many features available as plugins. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.
- Express.js was founded by TJ Holowaychuk. The first release, according to Express.js's GitHub repository, was on 22 May 2010. Version 0.12
- **NodeJS**

- Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.
- Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.
- Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.
- The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation.

OpenJS Foundation is facilitated by the Linux Foundation's Collaborative Projects program.

- Node.js was written initially by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web. The initial release supported only Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored by Joyent

- **MongoDB (Database)**

- MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL) which is deemed non-free by several distributions. MongoDB is a member of the MACH Alliance.
- 10gen software company began developing MongoDB in 2007 as a component of a planned platform as a service product. In 2009, the company shifted to an open-source development model, with the company offering commercial support and other services. In 2013, 10gen changed its name to MongoDB Inc.
- On October 20, 2017, MongoDB became a publicly traded company, listed on NASDAQ as MDB with an IPO price of $24 per share.
- MongoDB is a global company with US headquarters in New York City, USA and International headquarters in Dublin, Ireland.
- On October 30, 2019, MongoDB teamed up with Alibaba Cloud, who will offer its customers a MongoDB-as-a-service solution. Customers can use the managed offering from BABA's global data centers.

3
SYSTEM REQUIREMENT SPECIFICATION

3.1 Proposed System And Advantage

3.2 Development Strategy Model

- **Proposed System And Advantage**

  - In The Technological Word all The Work Is Done By The Web That's Not Any Existing System Is Providing This Type Services To User Interact With Cricscorer
  - The System was Outdated For The User It's need To Upgrade Frequently
  - Many facilities Is Not Providing On The Existing System .
  - At The Time Of Any Occasion Heavy Traffic On The Zoo Counter And Staff User Will Not Manage It
  - Not Providing Any Online Booking Services To User Book The Ticket.
  - Some Time Data Error is occur In The Database

Dis-Advantages

- Not Providing Online Service
- Online Booking Is Not Providing
- Not Handle Heavy Load
- Not User-Friendly
- UI Is Older

- Some Changes On The Requirements
- Some Time Crashed

4
SYSTEM DESIGN

4.1 Data Flow Diagram

4.2 E-R Diagram

4.3 Class Diagram

4.4 Use Case Diagram

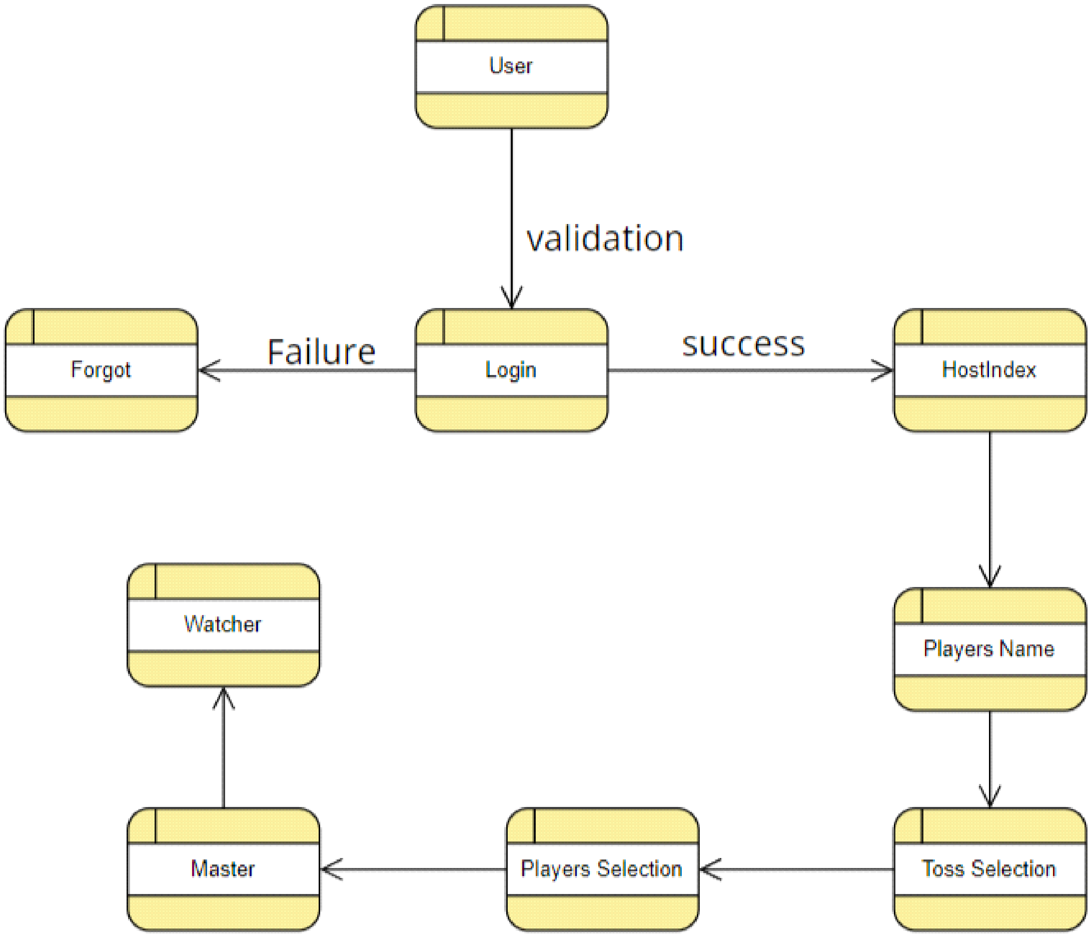
4.5 Sequence Diagram

4.6 Activity Diagram

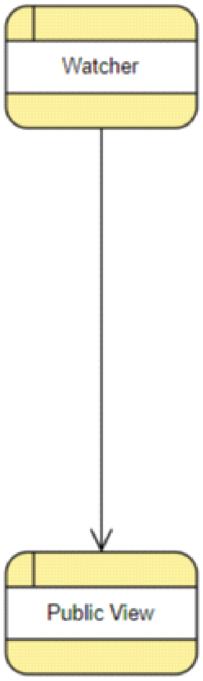
4.7 Data Dictionary(Database)

4.1 Data Flow Diagram

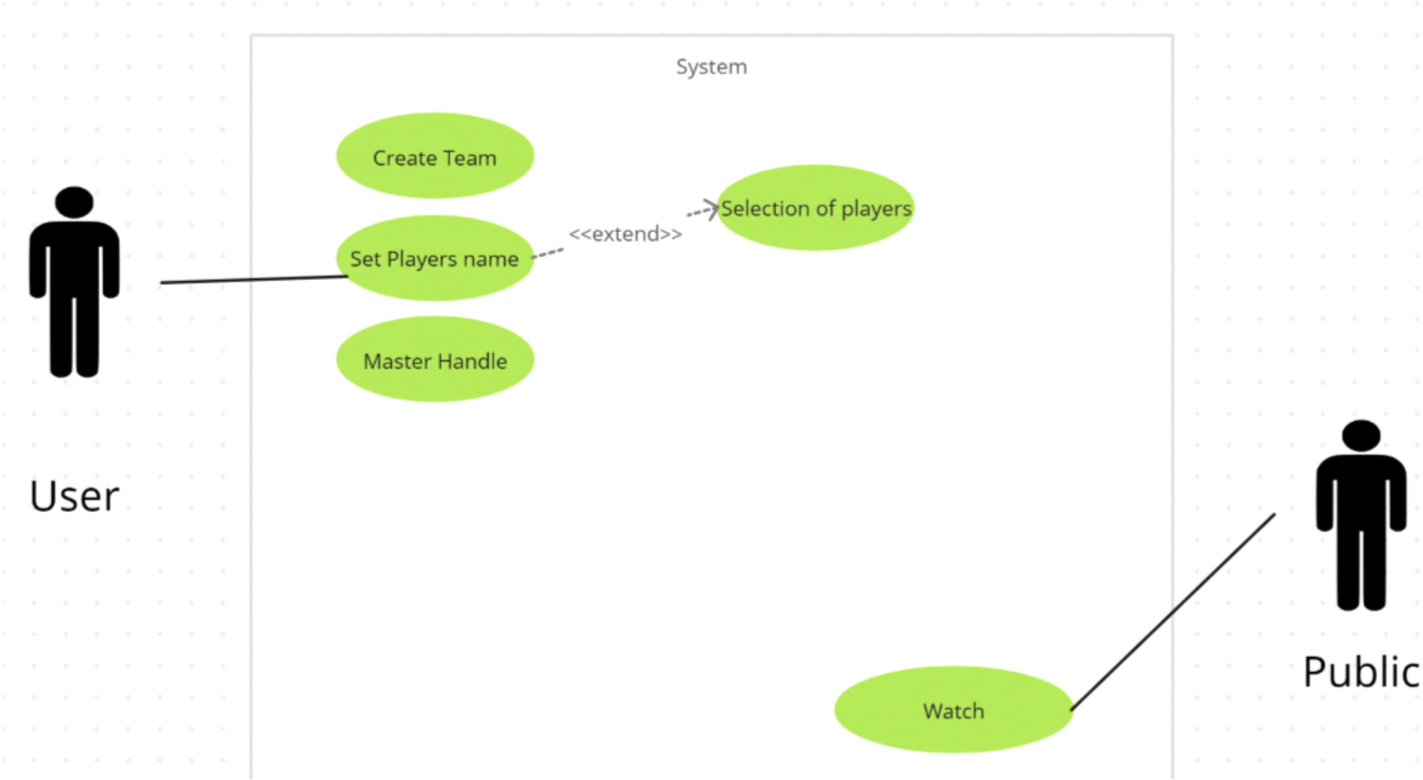
- **User Module DFD**


User
validation
Failure
Login
success
Forgot
HostIndex
Watcher
Players Name
Master
Players Selection
Toss Selection

- **Admin Module DFD**

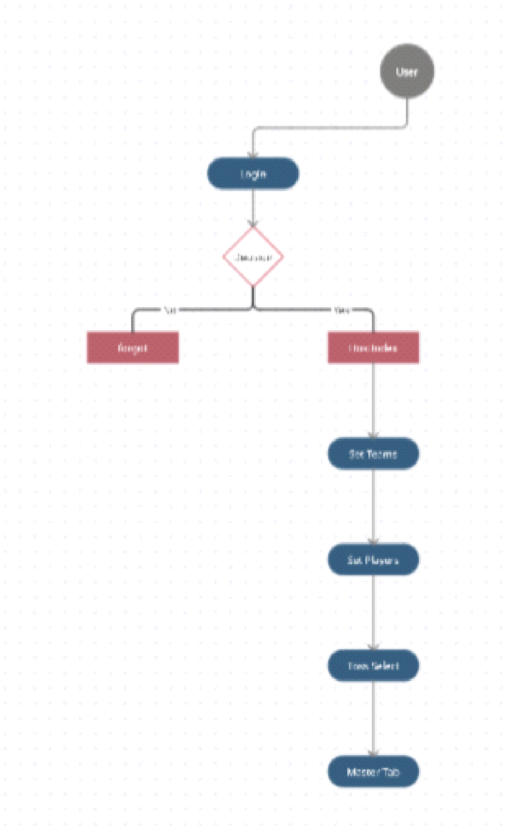


4.2 Use Case Diagram



4.6 Activity Diagram

- **User Activity Diagram**



4.7 Data Dictionary (Database)

COLLECTIONS

SCHEMA: USERS:-

NAME
EMAIL
PASSWORD

SCHEMA: {TEAMA}:-

NAME
ROLL
WICKETKEEPER
CAPTAIN

SCHEMA: {TEAMB}:-

NAME
ROLL
WICKETKEEPER
CAPTAIN

SCHEMA: {MASTER}:-

RUN
COUNT
SIX
FOUR
WC
NOBALL
EXTRA

5

SCREEN DESIGN

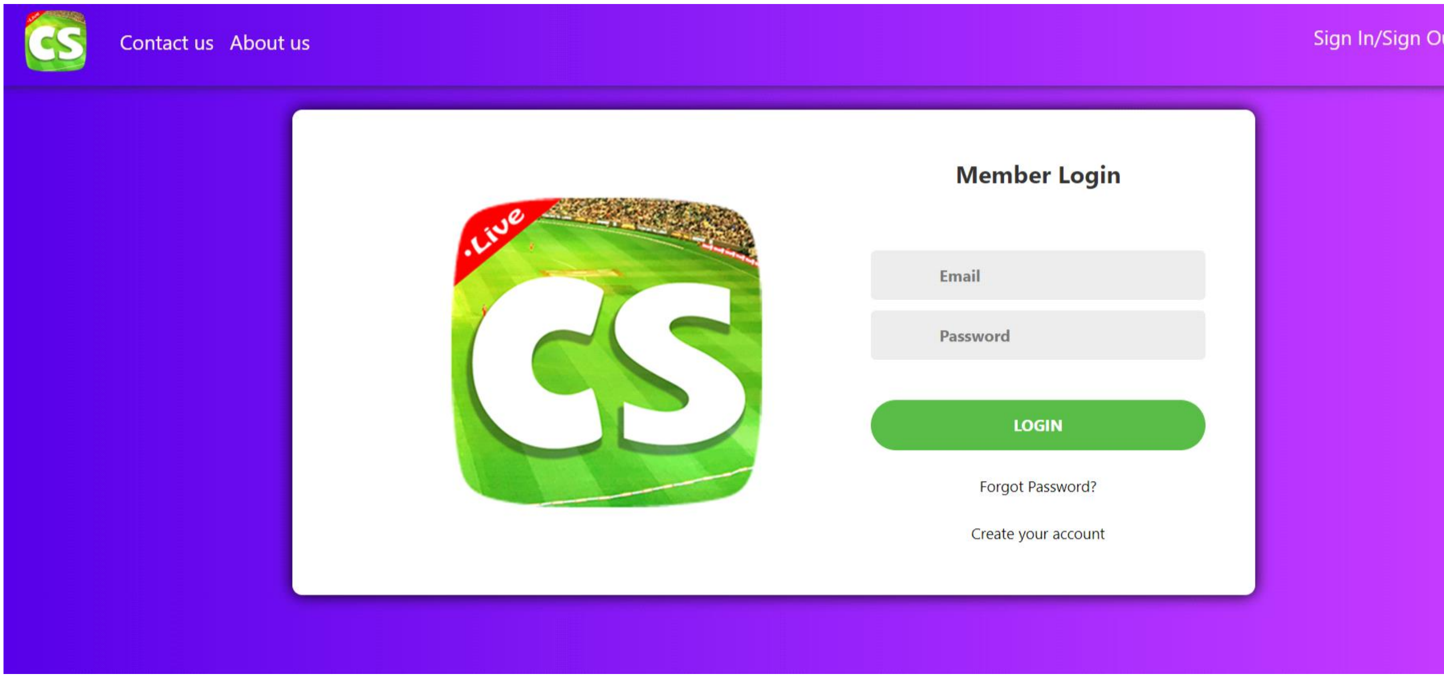
5.1 Input Screen

5.2 Output Screen

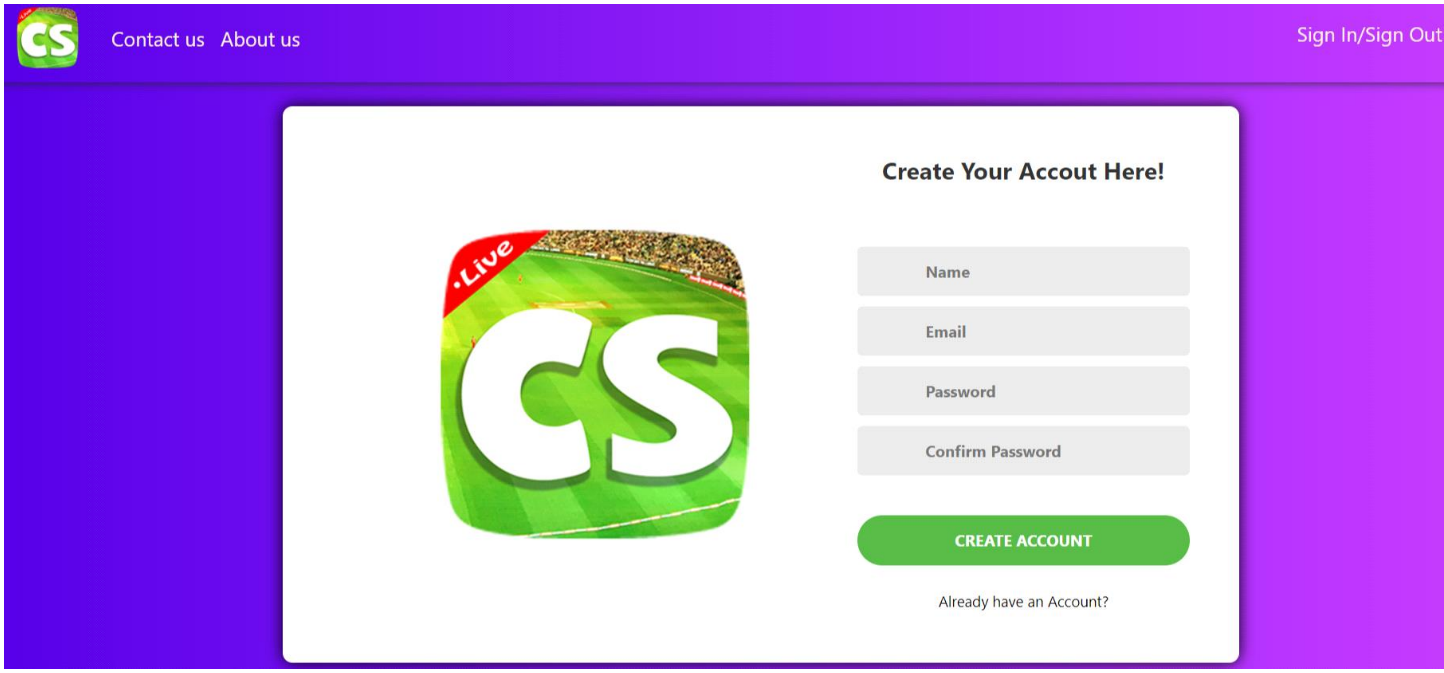
5.3 Report Screen

5.1 Input Screen

- **<u>SignIn Page.</u>**



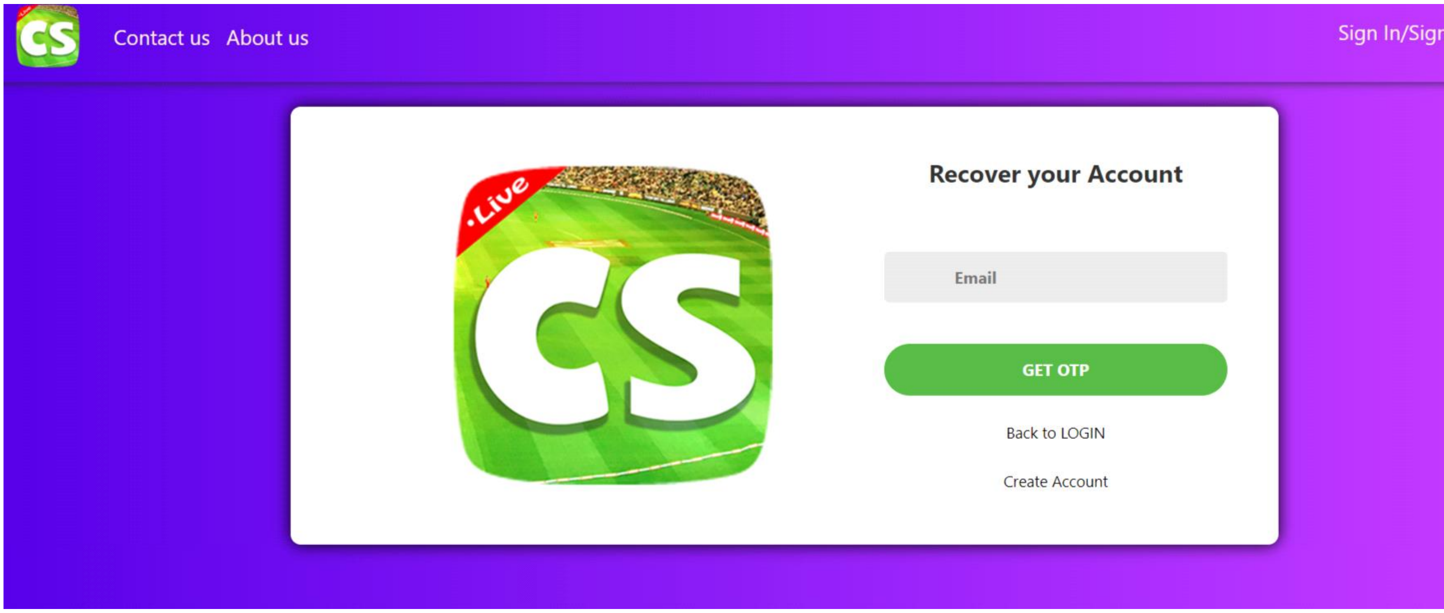
- **SignUp Page:**



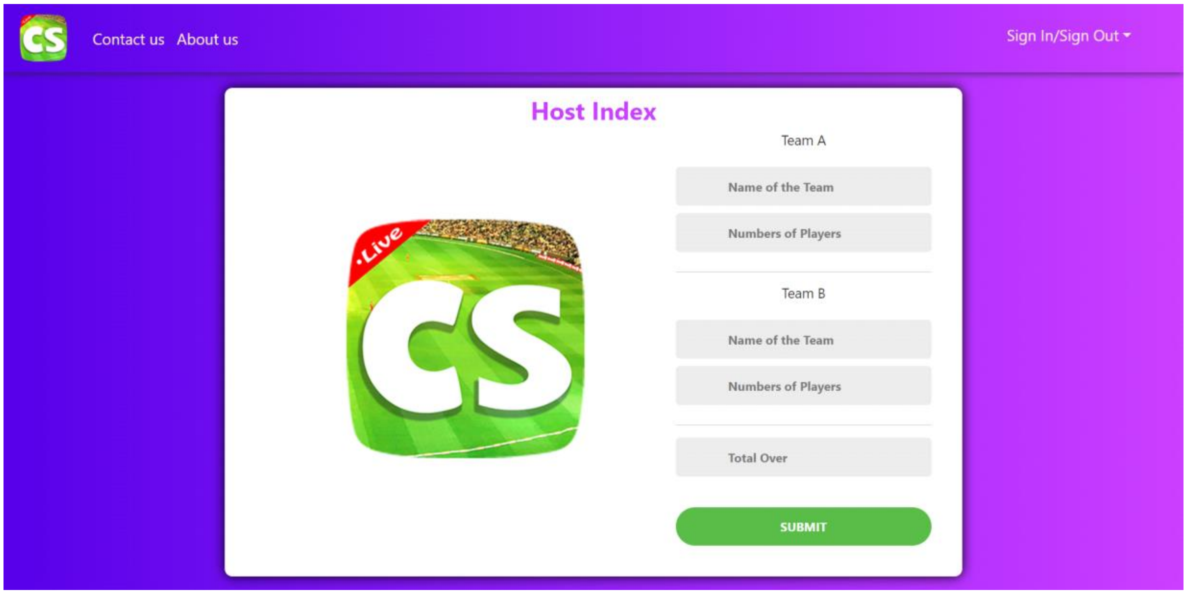
Create Your Accout Here!

Already have an Account?

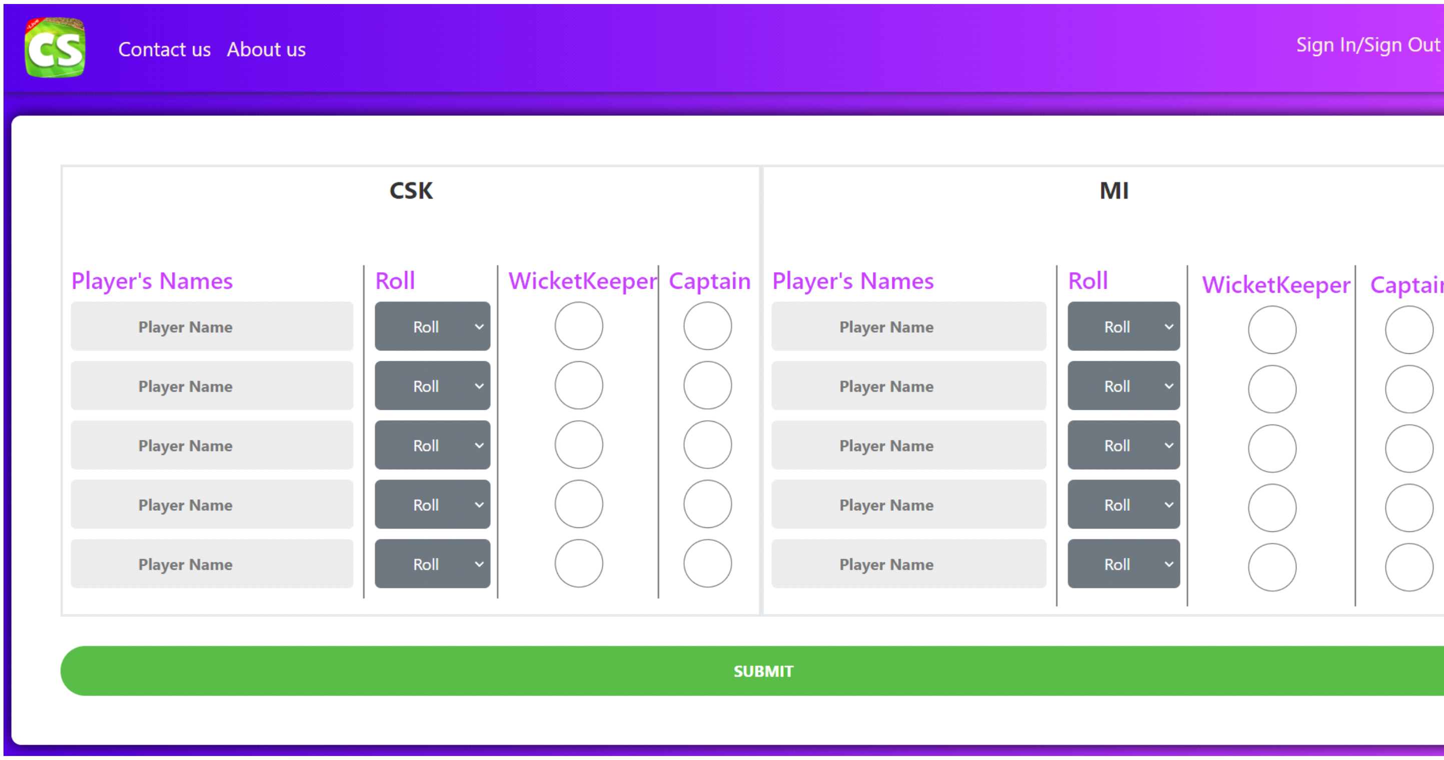
- **Forget Password Page:**



- **HostIndex Page:**

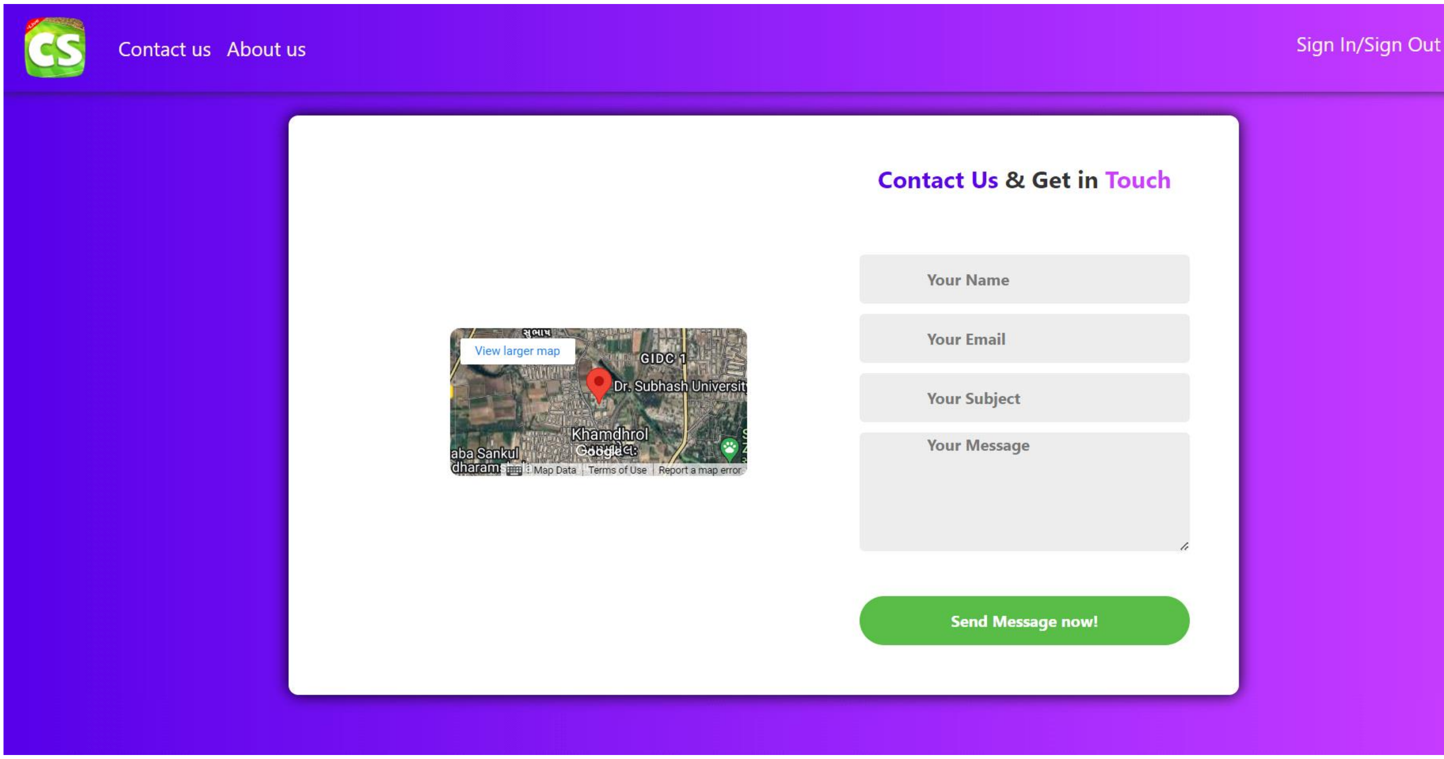


- **<u>players Name Page:</u>**



CSK				MI			
Player's Names	Roll	WicketKeeper	Captain	Player's Names	Roll	WicketKeeper	Captai
Player Name	Roll	○	○	Player Name	Roll	○	○
Player Name	Roll	○	○	Player Name	Roll	○	○
Player Name	Roll	○	○	Player Name	Roll	○	○
Player Name	Roll	○	○	Player Name	Roll	○	○
Player Name	Roll	○	○	Player Name	Roll	○	○

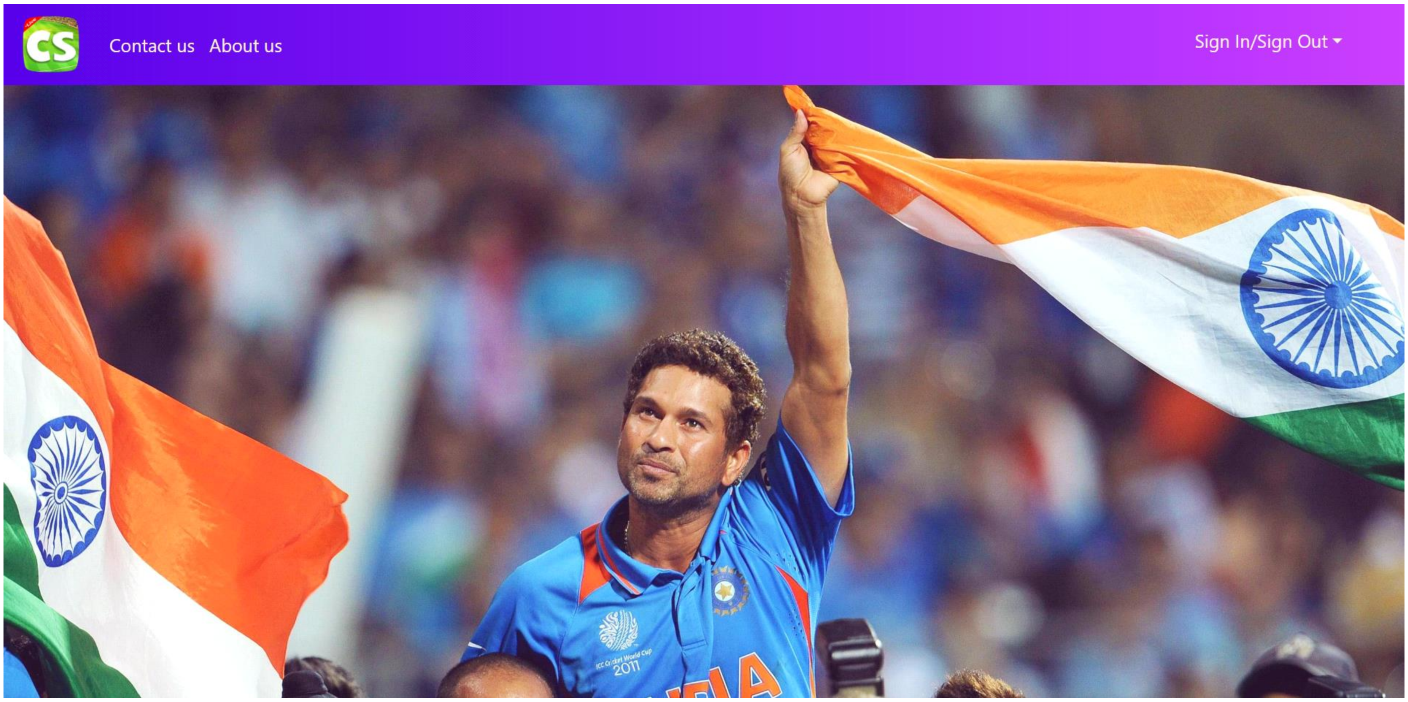
- **Contact Page:**



-

5.1 Output Screen

- **IndexPage:**



6

CODING

- **App.js**

```
import { BrowserRouter } from 'react-router-dom'
import Main from './Main'
import { Navbar } from './pages/Public/Navbar';

function App() {

  return (
    <BrowserRouter>
      <Navbar />
      <Main />
    </BrowserRouter>
  );
}

export default App;
```

- **index.js**

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import { store } from './pages/includes/Action/app/store';
import { Provider } from 'react-redux'

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}>
    <React.StrictMode>
      <App />
    </React.StrictMode>
  </Provider>
);
```

- **Main.jsx**

```
import React from 'react'
import { useSelector } from 'react-redux'
import { Routes, Route } from 'react-router-dom'
import './Main.css'
import HostIndex from './pages/Controller/HostIndex'
import Master from './pages/Controller/Master'
import PlayersName from './pages/Controller/PlayersName'
import PlayersSelection from './pages/Controller/PlayersSelection'
import Toss from './pages/Controller/Toss'
import Contact from './pages/Public/Contact'
import Forgot from './pages/Public/Forgot'
import Home from './pages/Public/Home'
import Index from './pages/Public/Index'
import SignIn from './pages/Public/SignIn'
import Signup from './pages/Public/Signup'

const Main = () => {
  const user = useSelector(state => state.user.user)
  const host = useSelector(state => state.host.host)
  const toss = useSelector(state => state.toss.toss)

  return (
    <div className='main'>
      <Routes>
        <Route path='/signin' element={<SignIn />} />
        <Route path='/signup' element={<Signup />} />
        <Route path='/forgot' element={<Forgot />} />
        <Route path='/contact' element={<Contact />} />
        <Route path='/home' element= {<Home /> } />
        <Route path='/hostIndex' element={<HostIndex /> } />
        <Route path='/playersName' element={ <PlayersName /> } />
        <Route path='/toss' element={<Toss /> } />
        <Route path='/playersSelection' element={<PlayersSelection />} />
        <Route path='/master' element={<Master />} />
        <Route path='/' element={<Index />} />
      </Routes>
    </div>
  )
}

export default Main
```

- **Index.jsx**

```
import React from 'react'
import { useState,useEffect } from 'react'
import '../includes/css/Style.css'
import all_teams from "../includes/img/all_teams.jpg"
import dhoni from "../includes/img/dhoni.jpg"
import sachin from "../includes/img/sachin.jpg"

import kohli from "../includes/img/kohli.jpg"

import yuvi from "../includes/img/yuvi.jpg"


const Index = () => {


  const [slideIndex, setSlideIndex] = useState(0);


  useEffect(() => {

    const timer = setInterval(() => {

      setSlideIndex((prevIndex) => (prevIndex + 1) % 5);

    }, 3000);


    return () => {

      clearInterval(timer);

    };

  }, []);


  return (

    <div className="slider-container">

      <div

        className="slide"

        style={{ transform: `translateX(-${slideIndex * 100}%)` }}
      >
        <img src={all_teams} alt='IMG' />
        <img src={dhoni} alt='IMG'/>
        <img src={sachin} alt='IMG' />
        <img src={kohli} alt='IMG'/>
        <img src={yuvi} alt='IMG' />
      </div>
    </div>
  )
}

export default Index
```

- **Navbar.jsx**

```
import React from 'react'
import { Link, useNavigate } from 'react-router-dom'
import { logout } from '../includes/Action/api/userApi'
import '../includes/css/Style.css'
import Logo from '../includes/img/Logo.png'

export const Navbar = () => {
    const navigate = useNavigate();
    const LogOut = () => {
        logout(navigate);
    }

    let uid = "null"

    if (sessionStorage.getItem('username')) {
        uid = sessionStorage.getItem('username')
    }

    return (
        <div>
            <nav className="navbar navbar-expand-lg bg-body-tertiary navbar-custom">
                <div className="container-fluid">
                    <Link className="navbar-brand" to='/'><img src={Logo} alt="LOGO" className='navbar-logo' /></Link>
                    <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                        <span className="navbar-toggler-icon"></span>
                    </button>
                    <div className="collapse navbar-collapse" id="navbarSupportedContent">
            <ul className="navbar-nav me-auto mb-2 mb-lg-0">
              {uid!=="null" && <li className="nav-item">
                <Link className="nav-link navbar-font" to='home'>Home</Link>
              </li>}
              <li className="nav-item">
                <Link className="nav-link navbar-font" to='contact'>Contact us</Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link navbar-font" to='about'>About us</Link>
              </li>
            </ul>
            {uid==="null" && <ul className='navbar-nav me-5 mb-2 mb-lg-0'>
              <li className="nav-item dropdown">
                <h2 className="nav-link dropdown-toggle navbar-font" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
                  Sign In/Sign Out
                </h2>
                <ul className="dropdown-menu navbar-signinsignout-dropdown">
                  <li><Link className="dropdown-item navbar-font" to='/signin'>Sign
In</Link></li>
                  <li><hr className="dropdown-divider" /></li>
                  <li><Link className="dropdown-item navbar-font" to='/signup'>Sign
Up</Link></li>
```

```
                                </ul>
                            </li>
                        </ul>}
                        {uid!=="null" && <ul className='navbar-nav me-5 mb-2 mb-lg-0'>
                            <li className="nav-item dropdown">
                                <h2 className="nav-link dropdown-toggle navbar-font" role="button" data-
bs-toggle="dropdown" aria-expanded="false">
                                    {uid}
                                </h2>
                                <ul className="dropdown-menu navbar-signinsignout-dropdown">
                                    <li><Link className="dropdown-item navbar-font"
to='/profile'>Profile</Link></li>
                                    <li><hr className="dropdown-divider" /></li>
                                    <li><button className="dropdown-item navbar-font"
onClick={LogOut}>Sign Out</button></li>
                                </ul>
                            </li>
                        </ul>}
                    </div>
                </div>
            </nav>
        </div>
    )
```

}

7

TESTING

(MANUAL,TEST CASE AND TEST DATA)

7.1 Manual Testing

7.2 Test Case

7.1 Manual Testing

- Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

- Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

- Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool.

- Manual testing is essential because one of the software testing fundamentals is "100% automation is not possible."

- There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

    - White Box Testing
    - Black Box Testing
    - Gray Box Testing

White-box testing

- The white box testing is done by Developer, where they check every line of a code before giving it to the Test Engineer. Since the code is visible for the Developer during the testing, that's why it is also known as White box testing.

Black box testing

- The black box testing is done by the Test Engineer, where they can check the functionality of an application or the software according to the customer /client's needs. In this, the code is not visible while performing the testing; that's why it is known as black-box testing.

Gray Box testing

- Gray box testing is a combination of white box and Black box testing. It can be performed by a person who knew both coding and testing. And if the single person performs white box, as well as black-box testing for the application, is known as Gray box testing.

8

ENHANCEMENT

8.1 Advantage Of Your Project

- This system is very used to watch cricket scoring system .User easily interact with system and collect all the information about its tour package or if any package is related to its choice it can book that package and save its time to offline process
- System also provide some security feature like Email Authentication ,Google Captcha Authentication to user data will be secured from the hacker attack
- Best user interface to user easily interact with the website and easily to navigate the website
- Light mode And Dark Mode provided into the website to increase the user interaction with the system
- Not only scoring

- **Future Enhancement :**

- Payment method will be available.
- New customization in website.
- Add New Services
- Some UI Changes
- More modules will be updated.
- Bug fixes and performance.
- More attractive.

9

REFERENCES

- **Book Used :**

    - Software Engineering – R.S.Pressman

- **Reference Sites:**

https://www.tutorialspoint.com/php/

https://www. stackoverflow.com/

https://www. getbootstrap.com/

htttp://www.youtube.com/

htttp://www.jquery.com/