

**International Institute of Information
Technology, Bangalore
(IIIT Bangalore)**



**Software Production Engineering
Project Report**

IPL 2023

Under the Guidance of Prof. B. Thangaraju

Deepak
(MT2022036)

Kushagra Agrawal
(MT2022145)

1. ABSTRACT

IPL 2023 is a dynamic and interactive website designed to provide users with real-time updates on the Indian Premier League (IPL) cricket tournament. The website offers live scores, statistics, schedules, and the ability to book tickets for matches held in various stadiums.

With a user-friendly interface, IPL 2023 delivers an immersive cricket experience. Users can access live scores and follow their favorite teams' progress in real-time. Detailed match statistics, such as runs, wickets, boundaries, and player performances, are also available, ensuring fans stay informed about every aspect of the game.

The website's schedule section allows users to plan their cricket viewing experience by providing match dates, venues, and timings. Users can easily navigate through the schedule to find specific matches or filter matches by teams, ensuring they never miss an important game.

One of the standout features of IPL 2023 is the ability to book tickets for matches held in different stadiums. The website integrates with ticketing services, enabling users to select their preferred seats and make secure online payments. By simplifying the ticket booking process, IPL 2023 ensures fans can effortlessly secure their spots in the stadium and enjoy the electrifying atmosphere of live IPL matches.

Overall, IPL 2023 is a comprehensive platform that brings the excitement of IPL cricket to fans worldwide. Whether it's following live scores, exploring

match statistics, planning match attendance, or booking tickets, the website caters to the diverse needs of IPL enthusiasts, enhancing their overall viewing experience.

2. INTRODUCTION

A. OVERVIEW

Introducing IPL 2023: Your Ultimate Destination for Live Cricket Action and Ticket Booking

Welcome to IPL 2023, your go-to website for all things Indian Premier League! We are proud to present a one-stop platform that brings you the exhilarating world of IPL cricket right at your fingertips.

With IPL 2023, you can stay updated with live scores, track match statistics, and plan your cricket viewing experience effortlessly. Our user-friendly interface ensures that you never miss a moment of the thrilling action happening on the cricket field.

B. FEATURES

- Live Score
- Schedule
- Points Table
- Statistics
- Register and Login for Admin and Users
- **Admin:** Add match, delete match , edit match , update match,see bookings, block user, make admin feature.
- **User:** Book ticket in stadium for any match.

C. WHY DEVOPS ?

DevOps is a set of tools that automate software development and IT operations. It focuses on shortening the systems development life cycle and providing continuous delivery with high software quality.

DevOps is one up on Agile software development; multiple DevOps aspects came from the Agile methodology.

- It optimizes the overall business by increasing efficiency through automation.
- Improves software development and deployment speed and stability.
- Deployment failures, rollbacks, and recovery time are all reduced.
- Improved collaboration and communication; lower costs and IT headcount

3. TOOLS USED

A. DevOps Tools

1. Version Control System : Git and GitHub :
<https://github.com/Deepak-tect/IPL>
2. Continuous Integration/Continuous Delivery : Jenkins
3. Building Tool : NPM (Node Package Manager)
4. Testing : Mocha framework
5. Containerization : Docker
6. Deployment : Ansible
7. Log creation, monitoring and visualization : ELK (Elasticsearch, Logstash, Kibana)
8. Frontend Development : React.js
9. Backend Development : Node.js, Express.js
10. Database : MongoDB

B. Project Technology Stack

1. Frontend: React (HTML, JSX, JavaScript, Bootstrap)
2. Backend: Express (NodeJS, JavaScript)
3. Database: MongoDB (No-SQL)
4. Cloud: MongoDB Atlas, Netlify, Heroku
5. Build Tool: npm

4. SYSTEM CONFIGURATION

1. Operating System - Linux Ubuntu 20
2. CPU & RAM - 4 core processor and 8GB RAM
3. Kernel Version - 5.4.0-89-generic
4. Database - MongoDB 4.4.5

5. SOFTWARE DEVELOPMENT LIFECYCLE - SDLC

I. INSTALLATIONS:

React

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies.

Update local before installing:

Keep the local packages and softwares updated:

```
sudo apt-get update
```

Install NodeJS and NPM:

Node.js is a javascript programming package management. It is the JavaScript runtime environment Node.js' default package manager. It comes pre-installed with Node.js. The package.json file contains the definitions for all npm packages. Package.json's content must be in JSON format. The definition file must have at least two fields. The names and variants are as follows. It is capable of managing dependencies. It installs all of the project's dependencies in a single command line. The package.json file also defines dependencies.

Inorder to run React, Node environment shall be installed before starting,

```
sudo apt-get install nodejs
sudo apt-get install npm
```

```
(base) deepak@deepak:~$ node -v
v16.18.0
(base) deepak@deepak:~$ npm -v
8.19.2
(base) deepak@deepak:~$ 
```

React App

```
(base) deepak@deepak:~/IPL$ ls
client-deployment.yml client-sidee hosts playbook.yml server-deployment.yml server-side
(base) deepak@deepak:~/IPL$ 
```

Express

Express.js, or simply Express, is a back end web application framework for 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.

Express App

Initializing node project with the default config:

```
npm init -y
```

-y automates the config with node default values i.e without going through the interactive process.

Running the Node Application locally:

```
npm install nodemon
nodemon server.js
```

```
(base) deepak@deepak:~/IPL$ cd server-side/
(base) deepak@deepak:~/IPL/server-side$ nodemon server.js
[nodemon] 2.0.21
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
server started
mongo db successful
```

```
{
  "name": "server-side",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "mocha --exit"
  },
  "author": "Deepak",
  "license": "ISC",
  "dependencies": {
    "axios": "^1.3.6",
    "bcryptjs": "^2.4.3",
    "chai": "^4.3.7",
    "cheerio": "^1.0.0-rc.12",
    "cors": "^2.8.5",
    "dotenv": "^16.0.3",
    "express": "^4.18.2",
    "jsonwebtoken": "^9.0.0",
    "mocha": "^10.2.0",
    "mongoose": "^7.0.3",
    "nodemailer": "^6.9.1",
  }
}
```

```
  "nodemon": "^2.0.22",
  "request": "^2.88.2",
  "winston": "^3.8.2",
  "winston-elasticsearch": "^0.17.2"
}
}
```

(Backend package.json)

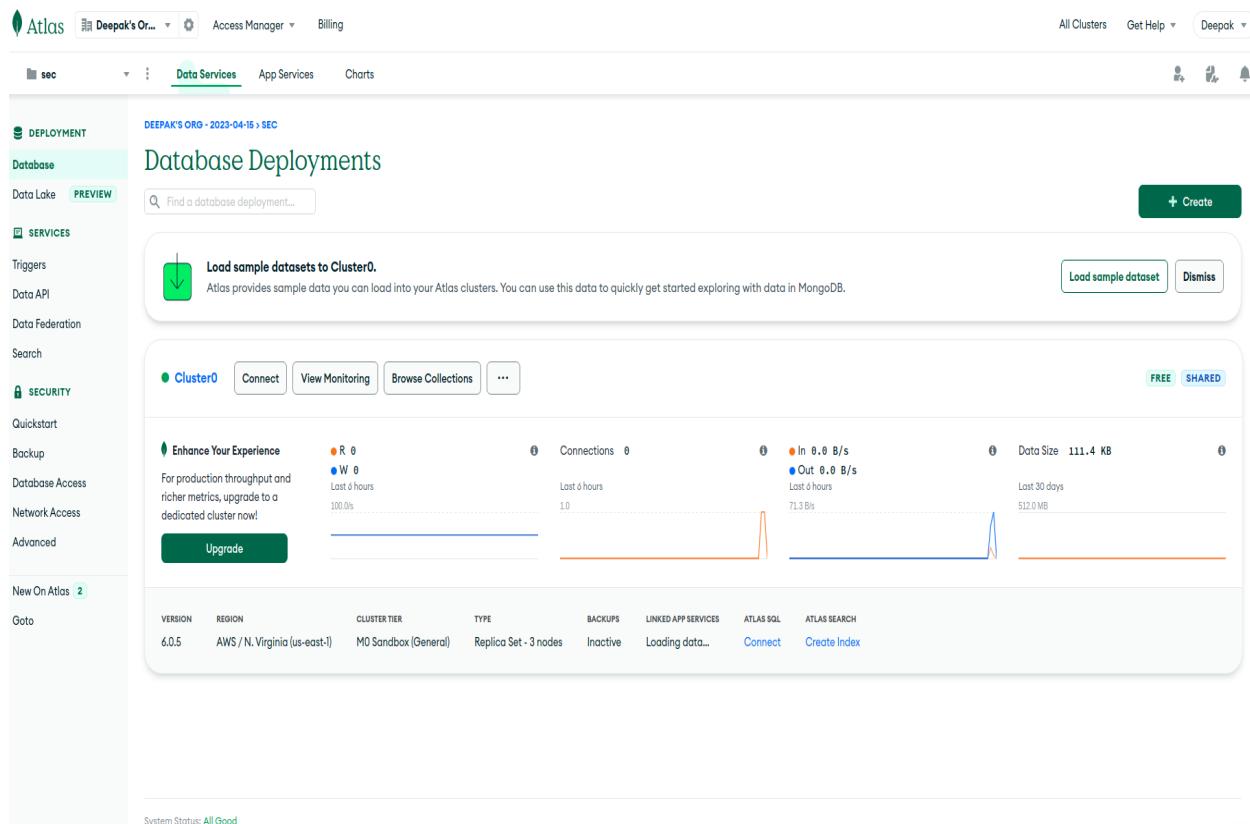
```
{
  "name": "client-side",
  "version": "0.1.0",
  "private": true,
  "proxy": "http://localhost:5000",
  "dependencies": {
    "@reduxjs/toolkit": "^1.9.3",
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "antd": "^5.4.2",
    "axios": "^1.3.5",
    "bootstrap": "^5.2.3",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-redux": "^8.0.5",
    "react-router-dom": "^6.9.0",
    "react-scripts": "5.0.1",
    "react-to-print": "^2.14.12",
    "reactstrap": "^9.1.9",
    "redux": "^4.2.1",
    "styled-components": "^5.3.10",
    "web-vitals": "^2.1.4"
  },
}
```

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},  
"eslintConfig": {  
  "extends": [  
    "react-app",  
    "react-app/jest"  
  ]  
},  
"browserslist": {  
  "production": [  
    ">0.2%",  
    "not dead",  
    "not op_mini all"  
  ],  
  "development": [  
    "last 1 chrome version",  
    "last 1 firefox version",  
    "last 1 safari version"  
  ]  
},  
"jest": {  
  "moduleNameMapper": {  
    "axios": "axios/dist/node/axios.cjs"  
  }  
}
```

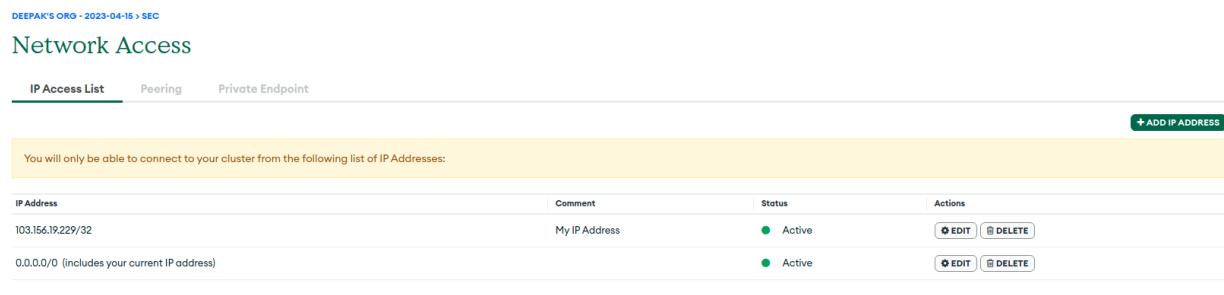
(Frontend package.json)

MongoDB:

We create a mongodb database on atlas to store all the user and farmer information. To set this database with our project use create URI and store it in the .env file.



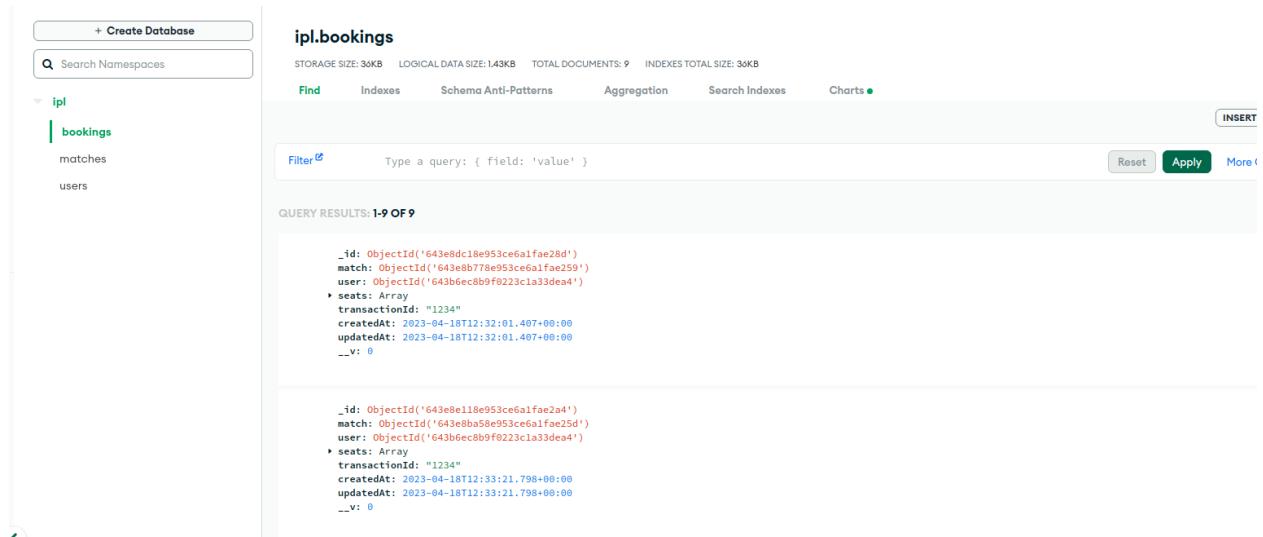
The screenshot shows the MongoDB Atlas interface for the 'sec' cluster. The left sidebar includes sections for Deployment (Database, Services, Security, Quickstart, Backup, Database Access, Network Access, Advanced), and New On Atlas. The main content area is titled 'Database Deployments' and shows a 'Load sample datasets to Cluster0' button. It displays metrics for the cluster: 0 R/s, 0 W/s, 0 Connections, 0 In B/s, 0 Out B/s, 111.4 KB Data Size, and 512.0 MB Lost 30 days. There is an 'Upgrade' button and a 'Connect' button. The bottom of the page shows the cluster configuration: Version 6.0.5, Region AWS / N. Virginia (us-east-1), Cluster Tier M0 Sandbox (General), Type Replica Set - 3 nodes, Backups Inactive, Linked App Services Loading data..., Atlas SQL Connect, and Atlas Search Create Index. A note at the bottom states 'System Status: All Good'.



The screenshot shows the MongoDB Atlas Network Access page. The top navigation bar includes 'IP Access List' (selected), 'Peering', and 'Private Endpoint'. A 'ADD IP ADDRESS' button is at the top right. A note says 'You will only be able to connect to your cluster from the following list of IP Addresses:'. The table lists two IP addresses: '103.156.19.229/32' and '0.0.0.0/0 (includes your current IP address)'. Both are marked as 'Active' with 'EDIT' and 'DELETE' buttons.

IP Address	Comment	Status	Actions
103.156.19.229/32	My IP Address	Active	EDIT DELETE
0.0.0.0/0 (includes your current IP address)		Active	EDIT DELETE

Tables:



The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with a '+ Create Database' button and a search bar for namespaces. Below that, a tree view shows a database named 'ipl' with a 'bookings' collection selected. Under 'bookings', there are 'matches' and 'users' sub-collections. The main area is titled 'ipl.bookings' and shows storage details: 30KB storage size, 1.45KB logical data size, 9 total documents, and 36KB indexes total size. Below this, there are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', 'Search Indexes', and 'Charts'. A 'Filter' bar with a query builder is at the top, and 'Reset', 'Apply', and 'More' buttons are to its right. The results table is titled 'QUERY RESULTS: 1-9 OF 9'. It shows two documents with the following data:

```
_id: ObjectId('643e8d8c18e953ce6a1fae28d')
match: ObjectId('643e8b8b58e953ce6a1fae259')
user: ObjectId('643b6eccb9f0223c1a33dea4')
seats: Array
transactionId: "1234"
createdAt: 2023-04-18T12:32:01.407+00:00
updatedAt: 2023-04-18T12:32:01.407+00:00
__v: 0

_id: ObjectId('643e8e118e953ce6a1fae2a4')
match: ObjectId('643e8b8b58e953ce6a1fae25d')
user: ObjectId('643b6eccb9f0223c1a33dea4')
seats: Array
transactionId: "1234"
createdAt: 2023-04-18T12:33:21.798+00:00
updatedAt: 2023-04-18T12:33:21.798+00:00
__v: 0
```

Store MongoDB URL in .env file

By creating a .env file in the api (backend) subdirectory, you can save the cluster's URL. In that URL, provide all of the MongoDB Atlas' credentials (password). This .env file is part of .gitignore. Furthermore, Mongoose uses the variable used to store this URL to connect to MongoDB.

```
⚙ .env
1  # mongo_url ="mongodb+srv://deepak:qwerty1234@cluster0.rhzhy9k.mongodb.net/ipl";
2  mongo_url = "mongodb+srv://user:qwerty1234@cluster0.y4nuej.mongodb.net/ipl"
3  jwt_secret = "ipl"
```

JWT Authentication:

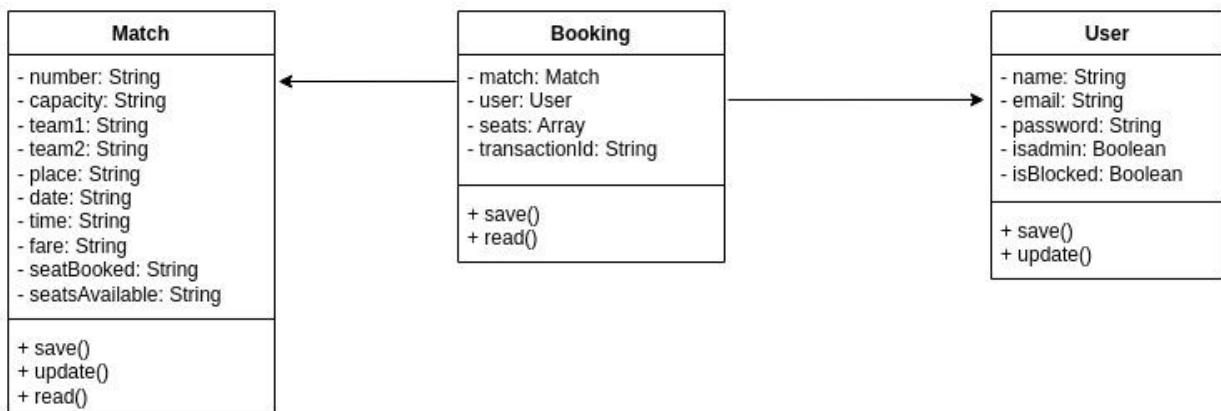
JSON Web Tokens (JWT) are an RFC 7519 open industry standard for representing claims between two parties. For example, you can use jwt.io to decode, verify, and produce JWT.

JWT defines a concise and self-contained way for transmitting information between two parties as a JSON object. This information may be reviewed and trusted because it is signed. A secret (using the HMAC algorithm) or an RSA or ECDSA public/private key pair can be used to sign JWTs.

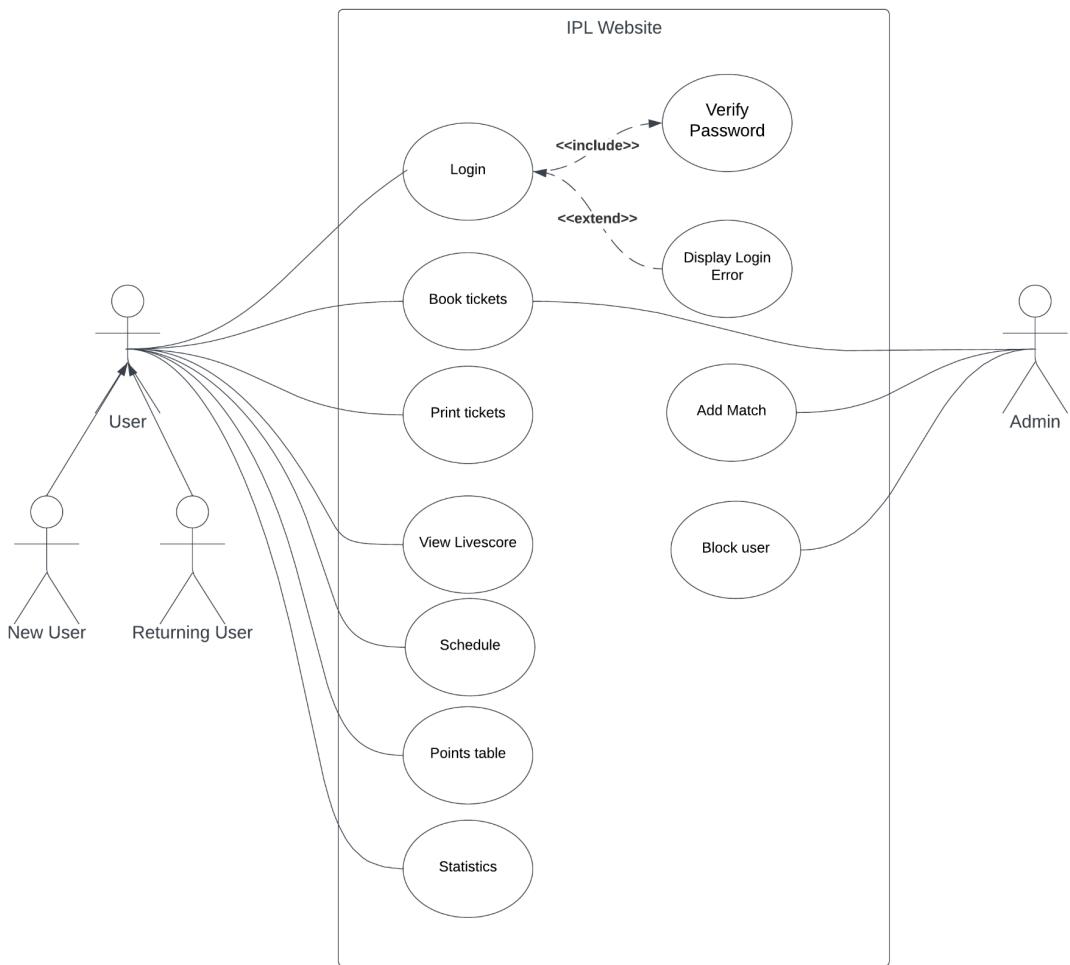
We create the JWT token and add it to .env file for performing Authentication.

```
⚙ .env
1 # mongo_url ="mongodb+srv://deepak:qwerty1234@cluster0.rhzhy9k.mongodb.net/ipl";
2 mongo_url = "mongodb+srv://user:qwerty1234@cluster0.y4nluej.mongodb.net/ipl"
3 jwt_secret = "ipl"
```

Class diagram:

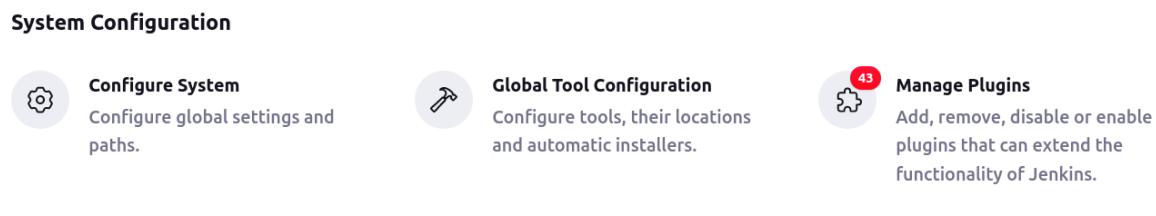


Use Case diagram:



6. Manage Plugins in Jenkins, Add credentials and Configurations of installed Plugins

We need to install all the required plugins like Build pipeline, Docker, GitHub, Maven Integration, Ansible etc. After they are all done installing, we need to restart Jenkins and add Docker credentials.



Global Tool configuration:

The image shows the Jenkins Global Tool Configuration page for the Git tool. It displays a single configuration entry:

Git

Git installations

Name	Path to Git executable	Install automatically
Default	git	<input type="checkbox"/>

Add Git

Ansible

Ansible installations

List of Ansible installations on this system

[Add Ansible](#)

Ansible Name X

Ansible

Path to ansible executables directory

/usr/bin/

Install automatically ?

Manage credentials:

		System	(global)	c211c8b5-503d-4e77-9526-529732fb7467	Secret text
		System	(global)	docker-jenkins	deepak18/***** (Docker hub credentials)

Update

deepak18/*** (Docker hub credentials)**

Delete

Docker hub credentials

Move

Usage

This credential has not been recorded as used anywhere.

Note: usage tracking requires the cooperation of plugins and consequently may not track every use.

Manage Plugins:

Name	Enabled
Ansible plugin 148.v6b_13c6de3a_47 Invoke Ansible Ad-Hoc commands and playbooks. Report an issue with this plugin	<input checked="" type="checkbox"/> X

[Docker Pipeline](#) 563.vd5d2e5c4007f

Build and use Docker containers from pipelines.

[Report an issue with this plugin](#)



This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

[GitHub Branch Source](#) 1696.v3a_7603564d04

Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.

[Report an issue with this plugin](#)



[Pipeline: GitHub Groovy Libraries](#) 38.v445716ea_edda_

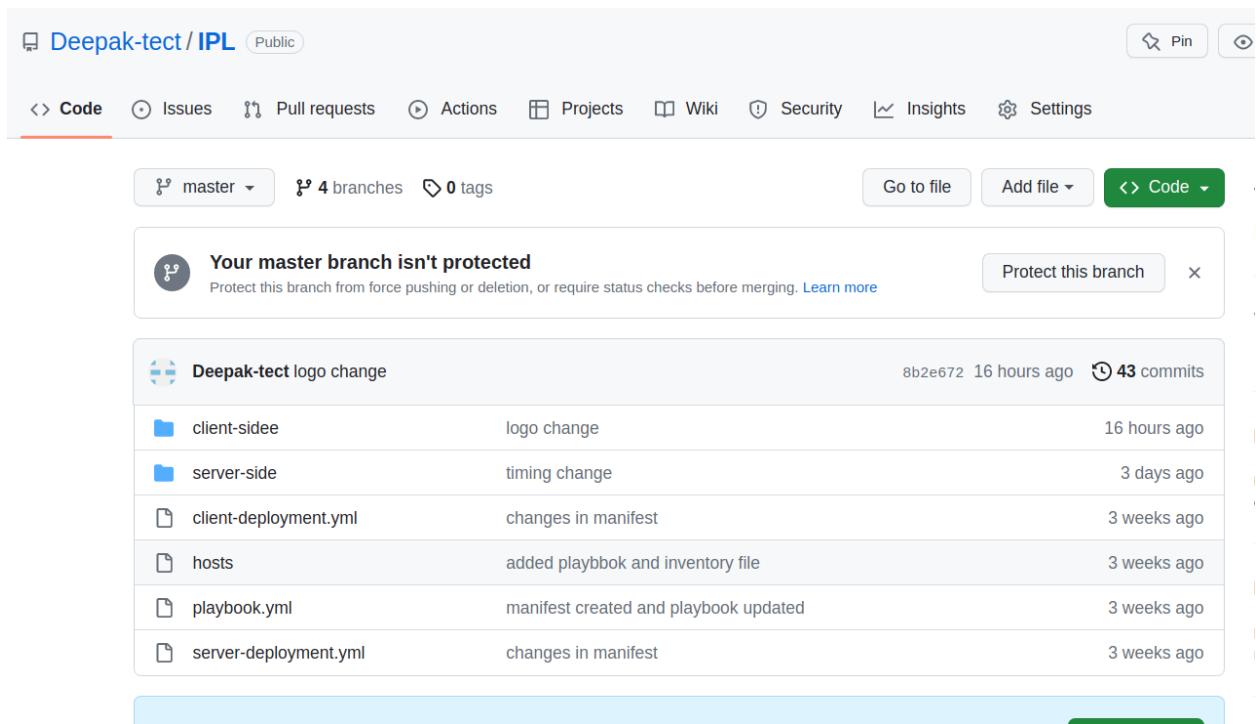
Allows Pipeline Groovy libraries to be loaded on the fly from GitHub.

[Report an issue with this plugin](#)



7. Source Code Management:

Source code management (SCM) is used to keep a track of all the modifications done to a source code repository. SCM tracks a current history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is very similar to Version control.



The screenshot shows a GitHub repository page for 'Deepak-tect / IPL'. The 'Code' tab is selected. At the top, it shows 'master' branch, 4 branches, 0 tags, and a 'Protect this branch' button. Below this, a notice says 'Your master branch isn't protected'. The main area displays a list of commits for a branch named 'Deepak-tect logo change'. The commits are as follows:

Commit	Message	Time
client-sidee	logo change	16 hours ago
server-side	timing change	3 days ago
client-deployment.yml	changes in manifest	3 weeks ago
hosts	added playbok and inventory file	3 weeks ago
playbook.yml	manifest created and playbook updated	3 weeks ago
server-deployment.yml	changes in manifest	3 weeks ago

To integrate project with GitHub we do the following steps :

- `git init`
→ Initializes the project as a github repository locally.
- `git remote add origin`
→ Add the details of the remote branch.

- `git add`
→ This command stages all the changes of local repo.
- `git commit`
→ Commit command commits the changes to the respective branch in the current repository.
- `git push`
→ Push command will push the changes to the remote repository on GitHub.

8. API Documentation:

API	Details
http://localhost:5000/pointstable	Get Points table
http://localhost:5000/schedule	Get Display schedule
http://localhost:5000/livescore	Get Live score
http://localhost:5000/stats/highestscore	Get Highest score
http://localhost:5000/stats/highestavg	Get Highest average
http://localhost:5000/stats/mostfifty	Get Most fifty
http://localhost:5000/stats/mostwicket	Get Most wicket
http://localhost:5000/stats/bestbowling	Get Best bowling
http://localhost:5000/stats/bestbowlingsr	Get Best bowling average
http://localhost:5000/stats/mostrun	Get Most run
http://localhost:5000/user/login	Post method : Login
http://localhost:5000/user/register	Post method : Register
http://localhost:5000/user/get-all-users	Get all user
http://localhost:5000/user/update-user-permission	Post method : Update user permission
http://localhost:5000/user/get-user-by-id	Get user by id
http://localhost:5000/match/add-match	Post method :Admin match add
http://localhost:5000/match/get-all-matches	Admin get all match
http://localhost:5000/match/update-match	Post method :Admin update match

http://localhost:5000/match/get-match-by-id	Admin get match by id
http://localhost:5000/bookings/book-seat	Post method : Booking seat
http://localhost:5000/bookings/get-bookings-by-user-id	Get book by user id

Screenshots from POSTMAN API testing

http://localhost:5000/stats/bestbowlers

GET http://localhost:5000/stats/bestbowlers

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 [
2   {
3     "playerName": "MA Wood (LSG)",
4     "matches": "4",
5     "wicket": "11",
6     "sr": "8.72"
7   },
8   {
9     "playerName": "MR Marsh (DC)",
10    "matches": "9",
11    "wicket": "12",
12    "sr": "10.08"
13  }
]

```

http://localhost:5000/stats/bestbowling

GET http://localhost:5000/stats/bestbowling

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
{
  "playerName": "MA Wood",
  "over": "4.0",
  "wicket": "5",
  "economy": "3.50",
  "opposition": "v Capitals"
},
{
  "playerName": "Mohammed Shami",
  "over": "4.0",
  "wicket": "4",
  "economy": "2.75",
  "opposition": "v Capitals"
},
{
  "playerName": "CV Varun",
  "over": "3.4",
  "wicket": "4",
  "economy": "4.09",
  "opposition": "v RCB"
}
```

http://localhost:5000/stats/mostfifty

GET http://localhost:5000/stats/mostfifty

Params Authorization Headers (6) Body Pre-request Script Tests Setting:

Query Params

	Key	Value
	Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "playerName": "F du Plessis (RCB)",
4     "matches": "12",
5     "run": "631",
6     "fifty": "7"
7   },
8   {
9     "playerName": "V Kohli (RCB)",
10    "matches": "12",
11    "run": "438",
12    "fifty": "6"
13  },
14  {
15    "playerName": "GJ Maxwell (RCB)",
16    "matches": "12",
17    "run": "384",
18    "fifty": "5"
19  },
20  {
21    "playerName": "SA Yadav (MI)",
```

http://localhost:5000/stats/highestavg

GET http://localhost:5000/stats/highestavg

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "playerName": "Rashid Khan (GT)",
4     "matches": "12",
5     "run": "95",
6     "avg": "95.00"
7   },
8   {
9     "playerName": "F du Plessis (RCB)",
10    "matches": "12",
11    "run": "631",
12    "avg": "57.36"
13  },
14  {
15    "playerName": "Abdul Samad (SRH)",
16    "matches": "7",
17    "run": "165",
18    "avg": "55.00"
19  },
20  {
21    "playerName": "RK Singh (KKR)",
```

http://localhost:5000/stats/highestscore

GET http://localhost:5000/stats/highestscore

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "playerName": "YBK Jaiswal",
4     "run": "124",
5     "sr": "200.00"
6   },
7   {
8     "playerName": "VR Iyer",
9     "run": "104",
10    "sr": "203.92"
11  },
12  {
13    "playerName": "SA Yadav",
14    "run": "103*",
15    "sr": "210.20"
16  },
17  {
18    "playerName": "Prabhsimran Singh",
19    "run": "103",
20    "sr": "158.46"
21  },
22]
```

http://localhost:5000/stats/mostrun

GET http://localhost:5000/stats/mostrun

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "playerName": "F du Plessis (RCB)",
4     "matches": "12",
5     "run": "631",
6     "avg": "57.36",
7     "fifty": "7",
8     "hundreds": "-"
9   },
10  {
11    "playerName": "YBK Jaiswal (RR)",
12    "matches": "13",
13    "run": "575",
14    "avg": "47.91",
15    "fifty": "4",
16    "hundreds": "1"
17  },
18  {
19    "playerName": "DP Conway (CSK)",
20    "matches": "13",
21    "run": "498",
22  }
23 ]
```

GET <http://localhost:5000/schedule>

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "date": "Fri, 31 Mar",
3   "time": "1st Match (N), Ahmedabad, March 31, 2023, Indian Premier League",
4   "place": "02nd Match (N), Ahmedabad, May 15, 2023, Indian Premier League",
5   "team1": "Chennai Super Kings",
6   "team2": "Gujarat Titans",
7   "results": "Titans won by 5 wickets (with 4 balls remaining)",
8   "team1_score": "178/7",
9   "team2_score": "(19.2/20 ov, T:179) 182/5"
10  },
11  {
12    "date": "Sat, 01 Apr",
13    "time": "2nd Match (D/N), Mohali, April 01, 2023, Indian Premier League",
14    "place": "03rd Match (N), Lucknow, May 10, 2023, Indian Premier League",
15    "team1": "Punjab Kings",
16    "team2": "Kolkata Knight Riders",
17    "results": "Punjab Kings won by 7 runs (DLS method)",
18    "team1_score": "191/5",
19    "team2_score": "(16/16 ov, T:154) 146/7"
20  },
21  {
22    "date": "",
23    "time": "3rd Match (N), Lucknow, April 01, 2023, Indian Premier League",
24    "place": "04th Match (N), Dharamsala, May 17, 2023, Indian Premier League",
25    "team1": "Lucknow Super Giants",
26    "team2": "Delhi Capitals",
27    "results": "Super Giants won by 30 runs",
28    "team1_score": "193/0",
29    "team2_score": "(20 ov, T:194) 143/9"
30  },
31  {
32    "date": "Sun, 02 Apr",
33    "time": "4th Match (D/N), Hyderabad, April 02, 2023, Indian Premier League",
34    "place": "05th Match (N), Hyderabad, May 18, 2023, Indian Premier League"
35  }
```

GET http://localhost:5000/livescore

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	Key
	Key

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1   "team1": "Gujarat Titans",
2   "score1": "",
3   "team2": "Sunrisers Hyderabad",
4   "score2": "",
5   "crr": "",
6   "status": "Match yet to begin",
7   "batting1": [],
8   "batting2": [],
9   "bowler1": [],
10  "bowler2": []
```

GET http://localhost:5000/pointstable

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1  [
2    {
3  "Team": "1Gujarat Titans",
4  "match": "12",
5  "win": "8",
6  "lost": "4",
7  "tie": "0",
8  "point": "10",
9  "net_run_rate": "0.701"
10 },
11 {
12 "Team": "2Chennai Super Kings",
13 "match": "13",
14 "win": "7",
15 "lost": "5",
16 "tie": "0",
17 "point": "15",
18 "net_run_rate": "0.381"
19 },
20 {
21 "Team": "3Mumbai Indians",
22 "match": "12",
23 "win": "7",
24 "lost": "5",
25 "tie": "0",
26 "point": "14",
27 "net_run_rate": "-0.117"
28 },
29 {
30 "Team": "4Lucknow Super Giants",
31 "match": "12",
32 "win": "6",
33 "lost": "5",
34 "tie": "0",
35 "point": "13",
36 "net_run_rate": "0.309"
37 },
38 {
39 "Team": "5Royal Challengers Bangalore",
40 "match": "12",
41 "win": "6",
42 "lost": "6",
43 "tie": "0",
44 "point": "12",
45 "net_run_rate": "0.166"
46 },
47 {
48 "Team": "6Rajasthan Royals",
49 "match": "13",
50 "win": "0",
51 "lost": "7",
52 "tie": "0",
53 "point": "12",
54 "net_run_rate": "0.140"
55 },
56 {
57 "Team": "7Kolkata Knight Riders",
58 "match": "13",
59 "win": "0",
60 "lost": "7",
61 "tie": "0",
62 "point": "12",
63 "net_run_rate": "0.140"
64 },
65 {
66 "Team": "8Delhi Capitals",
67 "match": "13",
68 "win": "0",
69 "lost": "7",
70 "tie": "0",
71 "point": "12",
72 "net_run_rate": "0.140"
73 },
74 {
75 "Team": "9Sunrisers Hyderabad",
76 "match": "13",
77 "win": "0",
78 "lost": "7",
79 "tie": "0",
80 "point": "12",
81 "net_run_rate": "0.140"
82 },
83 {
84 "Team": "10Punjab Kings",
85 "match": "13",
86 "win": "0",
87 "lost": "7",
88 "tie": "0",
89 "point": "12",
90 "net_run_rate": "0.140"
91 },
92 {
93 "Team": "11Kings XI Punjab",
94 "match": "13",
95 "win": "0",
96 "lost": "7",
97 "tie": "0",
98 "point": "12",
99 "net_run_rate": "0.140"
100 },
101 {
102 "Team": "12Mumbai Indians",
103 "match": "13",
104 "win": "0",
105 "lost": "7",
106 "tie": "0",
107 "point": "12",
108 "net_run_rate": "0.140"
109 },
110 {
111 "Team": "13Kings XI Punjab",
112 "match": "13",
113 "win": "0",
114 "lost": "7",
115 "tie": "0",
116 "point": "12",
117 "net_run_rate": "0.140"
118 },
119 {
120 "Team": "14Kings XI Punjab",
121 "match": "13",
122 "win": "0",
123 "lost": "7",
124 "tie": "0",
125 "point": "12",
126 "net_run_rate": "0.140"
127 },
128 {
129 "Team": "15Kings XI Punjab",
130 "match": "13",
131 "win": "0",
132 "lost": "7",
133 "tie": "0",
134 "point": "12",
135 "net_run_rate": "0.140"
136 },
137 {
138 "Team": "16Kings XI Punjab",
139 "match": "13",
140 "win": "0",
141 "lost": "7",
142 "tie": "0",
143 "point": "12",
144 "net_run_rate": "0.140"
145 },
146 {
147 "Team": "17Kings XI Punjab",
148 "match": "13",
149 "win": "0",
150 "lost": "7",
151 "tie": "0",
152 "point": "12",
153 "net_run_rate": "0.140"
154 },
155 {
156 "Team": "18Kings XI Punjab",
157 "match": "13",
158 "win": "0",
159 "lost": "7",
160 "tie": "0",
161 "point": "12",
162 "net_run_rate": "0.140"
163 },
164 {
165 "Team": "19Kings XI Punjab",
166 "match": "13",
167 "win": "0",
168 "lost": "7",
169 "tie": "0",
170 "point": "12",
171 "net_run_rate": "0.140"
172 },
173 {
174 "Team": "20Kings XI Punjab",
175 "match": "13",
176 "win": "0",
177 "lost": "7",
178 "tie": "0",
179 "point": "12",
180 "net_run_rate": "0.140"
181 },
182 {
183 "Team": "21Kings XI Punjab",
184 "match": "13",
185 "win": "0",
186 "lost": "7",
187 "tie": "0",
188 "point": "12",
189 "net_run_rate": "0.140"
190 },
191 {
192 "Team": "22Kings XI Punjab",
193 "match": "13",
194 "win": "0",
195 "lost": "7",
196 "tie": "0",
197 "point": "12",
198 "net_run_rate": "0.140"
199 },
200 {
201 "Team": "23Kings XI Punjab",
202 "match": "13",
203 "win": "0",
204 "lost": "7",
205 "tie": "0",
206 "point": "12",
207 "net_run_rate": "0.140"
208 },
209 {
210 "Team": "24Kings XI Punjab",
211 "match": "13",
212 "win": "0",
213 "lost": "7",
214 "tie": "0",
215 "point": "12",
216 "net_run_rate": "0.140"
217 },
218 {
219 "Team": "25Kings XI Punjab",
220 "match": "13",
221 "win": "0",
222 "lost": "7",
223 "tie": "0",
224 "point": "12",
225 "net_run_rate": "0.140"
226 },
227 {
228 "Team": "26Kings XI Punjab",
229 "match": "13",
230 "win": "0",
231 "lost": "7",
232 "tie": "0",
233 "point": "12",
234 "net_run_rate": "0.140"
235 },
236 {
237 "Team": "27Kings XI Punjab",
238 "match": "13",
239 "win": "0",
240 "lost": "7",
241 "tie": "0",
242 "point": "12",
243 "net_run_rate": "0.140"
244 },
245 {
246 "Team": "28Kings XI Punjab",
247 "match": "13",
248 "win": "0",
249 "lost": "7",
250 "tie": "0",
251 "point": "12",
252 "net_run_rate": "0.140"
253 },
254 {
255 "Team": "29Kings XI Punjab",
256 "match": "13",
257 "win": "0",
258 "lost": "7",
259 "tie": "0",
260 "point": "12",
261 "net_run_rate": "0.140"
262 },
263 {
264 "Team": "30Kings XI Punjab",
265 "match": "13",
266 "win": "0",
267 "lost": "7",
268 "tie": "0",
269 "point": "12",
270 "net_run_rate": "0.140"
271 },
272 {
273 "Team": "31Kings XI Punjab",
274 "match": "13",
275 "win": "0",
276 "lost": "7",
277 "tie": "0",
278 "point": "12",
279 "net_run_rate": "0.140"
280 },
281 {
282 "Team": "32Kings XI Punjab",
283 "match": "13",
284 "win": "0",
285 "lost": "7",
286 "tie": "0",
287 "point": "12",
288 "net_run_rate": "0.140"
289 },
290 {
291 "Team": "33Kings XI Punjab",
292 "match": "13",
293 "win": "0",
294 "lost": "7",
295 "tie": "0",
296 "point": "12",
297 "net_run_rate": "0.140"
298 },
299 {
300 "Team": "34Kings XI Punjab",
301 "match": "13",
302 "win": "0",
303 "lost": "7",
304 "tie": "0",
305 "point": "12",
306 "net_run_rate": "0.140"
307 },
308 {
309 "Team": "35Kings XI Punjab",
310 "match": "13",
311 "win": "0",
312 "lost": "7",
313 "tie": "0",
314 "point": "12",
315 "net_run_rate": "0.140"
316 },
317 {
318 "Team": "36Kings XI Punjab",
319 "match": "13",
320 "win": "0",
321 "lost": "7",
322 "tie": "0",
323 "point": "12",
324 "net_run_rate": "0.140"
325 },
326 {
327 "Team": "37Kings XI Punjab",
328 "match": "13",
329 "win": "0",
330 "lost": "7",
331 "tie": "0",
332 "point": "12",
333 "net_run_rate": "0.140"
334 },
335 {
336 "Team": "38Kings XI Punjab",
337 "match": "13",
338 "win": "0",
339 "lost": "7",
340 "tie": "0",
341 "point": "12",
342 "net_run_rate": "0.140"
343 },
344 {
345 "Team": "39Kings XI Punjab",
346 "match": "13",
347 "win": "0",
348 "lost": "7",
349 "tie": "0",
350 "point": "12",
351 "net_run_rate": "0.140"
352 },
353 {
354 "Team": "40Kings XI Punjab",
355 "match": "13",
356 "win": "0",
357 "lost": "7",
358 "tie": "0",
359 "point": "12",
360 "net_run_rate": "0.140"
361 },
362 {
363 "Team": "41Kings XI Punjab",
364 "match": "13",
365 "win": "0",
366 "lost": "7",
367 "tie": "0",
368 "point": "12",
369 "net_run_rate": "0.140"
370 },
371 {
372 "Team": "42Kings XI Punjab",
373 "match": "13",
374 "win": "0",
375 "lost": "7",
376 "tie": "0",
377 "point": "12",
378 "net_run_rate": "0.140"
379 },
380 {
381 "Team": "43Kings XI Punjab",
382 "match": "13",
383 "win": "0",
384 "lost": "7",
385 "tie": "0",
386 "point": "12",
387 "net_run_rate": "0.140"
388 },
389 {
390 "Team": "44Kings XI Punjab",
391 "match": "13",
392 "win": "0",
393 "lost": "7",
394 "tie": "0",
395 "point": "12",
396 "net_run_rate": "0.140"
397 },
398 {
399 "Team": "45Kings XI Punjab",
400 "match": "13",
401 "win": "0",
402 "lost": "7",
403 "tie": "0",
404 "point": "12",
405 "net_run_rate": "0.140"
406 },
407 {
408 "Team": "46Kings XI Punjab",
409 "match": "13",
410 "win": "0",
411 "lost": "7",
412 "tie": "0",
413 "point": "12",
414 "net_run_rate": "0.140"
415 },
416 {
417 "Team": "47Kings XI Punjab",
418 "match": "13",
419 "win": "0",
420 "lost": "7",
421 "tie": "0",
422 "point": "12",
423 "net_run_rate": "0.140"
424 },
425 {
426 "Team": "48Kings XI Punjab",
427 "match": "13",
428 "win": "0",
429 "lost": "7",
430 "tie": "0",
431 "point": "12",
432 "net_run_rate": "0.140"
433 },
434 {
435 "Team": "49Kings XI Punjab",
436 "match": "13",
437 "win": "0",
438 "lost": "7",
439 "tie": "0",
440 "point": "12",
441 "net_run_rate": "0.140"
442 },
443 {
444 "Team": "50Kings XI Punjab",
445 "match": "13",
446 "win": "0",
447 "lost": "7",
448 "tie": "0",
449 "point": "12",
450 "net_run_rate": "0.140"
451 },
452 {
453 "Team": "51Kings XI Punjab",
454 "match": "13",
455 "win": "0",
456 "lost": "7",
457 "tie": "0",
458 "point": "12",
459 "net_run_rate": "0.140"
460 },
461 {
462 "Team": "52Kings XI Punjab",
463 "match": "13",
464 "win": "0",
465 "lost": "7",
466 "tie": "0",
467 "point": "12",
468 "net_run_rate": "0.140"
469 },
470 {
471 "Team": "53Kings XI Punjab",
472 "match": "13",
473 "win": "0",
474 "lost": "7",
475 "tie": "0",
476 "point": "12",
477 "net_run_rate": "0.140"
478 },
479 {
480 "Team": "54Kings XI Punjab",
481 "match": "13",
482 "win": "0",
483 "lost": "7",
484 "tie": "0",
485 "point": "12",
486 "net_run_rate": "0.140"
487 },
488 {
489 "Team": "55Kings XI Punjab",
490 "match": "13",
491 "win": "0",
492 "lost": "7",
493 "tie": "0",
494 "point": "12",
495 "net_run_rate": "0.140"
496 },
497 {
498 "Team": "56Kings XI Punjab",
499 "match": "13",
500 "win": "0",
501 "lost": "7",
502 "tie": "0",
503 "point": "12",
504 "net_run_rate": "0.140"
505 },
506 {
507 "Team": "57Kings XI Punjab",
508 "match": "13",
509 "win": "0",
510 "lost": "7",
511 "tie": "0",
512 "point": "12",
513 "net_run_rate": "0.140"
514 },
515 {
516 "Team": "58Kings XI Punjab",
517 "match": "13",
518 "win": "0",
519 "lost": "7",
520 "tie": "0",
521 "point": "12",
522 "net_run_rate": "0.140"
523 },
524 {
525 "Team": "59Kings XI Punjab",
526 "match": "13",
527 "win": "0",
528 "lost": "7",
529 "tie": "0",
530 "point": "12",
531 "net_run_rate": "0.140"
532 },
533 {
534 "Team": "60Kings XI Punjab",
535 "match": "13",
536 "win": "0",
537 "lost": "7",
538 "tie": "0",
539 "point": "12",
540 "net_run_rate": "0.140"
541 },
542 {
543 "Team": "61Kings XI Punjab",
544 "match": "13",
545 "win": "0",
546 "lost": "7",
547 "tie": "0",
548 "point": "12",
549 "net_run_rate": "0.140"
550 },
551 {
552 "Team": "62Kings XI Punjab",
553 "match": "13",
554 "win": "0",
555 "lost": "7",
556 "tie": "0",
557 "point": "12",
558 "net_run_rate": "0.140"
559 },
560 {
561 "Team": "63Kings XI Punjab",
562 "match": "13",
563 "win": "0",
564 "lost": "7",
565 "tie": "0",
566 "point": "12",
567 "net_run_rate": "0.140"
568 },
569 {
570 "Team": "64Kings XI Punjab",
571 "match": "13",
572 "win": "0",
573 "lost": "7",
574 "tie": "0",
575 "point": "12",
576 "net_run_rate": "0.140"
577 },
578 {
579 "Team": "65Kings XI Punjab",
580 "match": "13",
581 "win": "0",
582 "lost": "7",
583 "tie": "0",
584 "point": "12",
585 "net_run_rate": "0.140"
586 },
587 {
588 "Team": "66Kings XI Punjab",
589 "match": "13",
590 "win": "0",
591 "lost": "7",
592 "tie": "0",
593 "point": "12",
594 "net_run_rate": "0.140"
595 },
596 {
597 "Team": "67Kings XI Punjab",
598 "match": "13",
599 "win": "0",
600 "lost": "7",
601 "tie": "0",
602 "point": "12",
603 "net_run_rate": "0.140"
604 },
605 {
606 "Team": "68Kings XI Punjab",
607 "match": "13",
608 "win": "0",
609 "lost": "7",
610 "tie": "0",
611 "point": "12",
612 "net_run_rate": "0.140"
613 },
614 {
615 "Team": "69Kings XI Punjab",
616 "match": "13",
617 "win": "0",
618 "lost": "7",
619 "tie": "0",
620 "point": "12",
621 "net_run_rate": "0.140"
622 },
623 {
624 "Team": "70Kings XI Punjab",
625 "match": "13",
626 "win": "0",
627 "lost": "7",
628 "tie": "0",
629 "point": "12",
630 "net_run_rate": "0.140"
631 },
632 {
633 "Team": "71Kings XI Punjab",
634 "match": "13",
635 "win": "0",
636 "lost": "7",
637 "tie": "0",
638 "point": "12",
639 "net_run_rate": "0.140"
640 },
641 {
642 "Team": "72Kings XI Punjab",
643 "match": "13",
644 "win": "0",
645 "lost": "7",
646 "tie": "0",
647 "point": "12",
648 "net_run_rate": "0.140"
649 },
650 {
651 "Team": "73Kings XI Punjab",
652 "match": "13",
653 "win": "0",
654 "lost": "7",
655 "tie": "0",
656 "point": "12",
657 "net_run_rate": "0.140"
658 },
659 {
660 "Team": "74Kings XI Punjab",
661 "match": "13",
662 "win": "0",
663 "lost": "7",
664 "tie": "0",
665 "point": "12",
666 "net_run_rate": "0.140"
667 },
668 {
669 "Team": "75Kings XI Punjab",
670 "match": "13",
671 "win": "0",
672 "lost": "7",
673 "tie": "0",
674 "point": "12",
675 "net_run_rate": "0.140"
676 },
677 {
678 "Team": "76Kings XI Punjab",
679 "match": "13",
680 "win": "0",
681 "lost": "7",
682 "tie": "0",
683 "point": "12",
684 "net_run_rate": "0.140"
685 },
686 {
687 "Team": "77Kings XI Punjab",
688 "match": "13",
689 "win": "0",
690 "lost": "7",
691 "tie": "0",
692 "point": "12",
693 "net_run_rate": "0.140"
694 },
695 {
696 "Team": "78Kings XI Punjab",
697 "match": "13",
698 "win": "0",
699 "lost": "7",
700 "tie": "0",
701 "point": "12",
702 "net_run_rate": "0.140"
703 },
704 {
705 "Team": "79Kings XI Punjab",
706 "match": "13",
707 "win": "0",
708 "lost": "7",
709 "tie": "0",
710 "point": "12",
711 "net_run_rate": "0.140"
712 },
713 {
714 "Team": "80Kings XI Punjab",
715 "match": "13",
716 "win": "0",
717 "lost": "7",
718 "tie": "0",
719 "point": "12",
720 "net_run_rate": "0.140"
721 },
722 {
723 "Team": "81Kings XI Punjab",
724 "match": "13",
725 "win": "0",
726 "lost": "7",
727 "tie": "0",
728 "point": "12",
729 "net_run_rate": "0.140"
730 },
731 {
732 "Team": "82Kings XI Punjab",
733 "match": "13",
734 "win": "0",
735 "lost": "7",
736 "tie": "0",
737 "point": "12",
738 "net_run_rate": "0.140"
739 },
740 {
741 "Team": "83Kings XI Punjab",
742 "match": "13",
743 "win": "0",
744 "lost": "7",
745 "tie": "0",
746 "point": "12",
747 "net_run_rate": "0.140"
748 },
749 {
750 "Team": "84Kings XI Punjab",
751 "match": "13",
752 "win": "0",
753 "lost": "7",
754 "tie": "0",
755 "point": "12",
756 "net_run_rate": "0.140"
757 },
758 {
759 "Team": "85Kings XI Punjab",
760 "match": "13",
761 "win": "0",
762 "lost": "7",
763 "tie": "0",
764 "point": "12",
765 "net_run_rate": "0.140"
766 },
767 {
768 "Team": "86Kings XI Punjab",
769 "match": "13",
770 "win": "0",
771 "lost": "7",
772 "tie": "0",
773 "point": "12",
774 "net_run_rate": "0.140"
775 },
776 {
777 "Team": "87Kings XI Punjab",
778 "match": "13",
779 "win": "0",
780 "lost": "7",
781 "tie": "0",
782 "point": "12",
783 "net_run_rate": "0.140"
784 },
785 {
786 "Team": "88Kings XI Punjab",
787 "match": "13",
788 "win": "0",
789 "lost": "7",
790 "tie": "0",
791 "point": "12",
792 "net_run_rate": "0.140"
793 },
794 {
795 "Team": "89Kings XI Punjab",
796 "match": "13",
797 "win": "0",
798 "lost": "7",
799 "tie": "0",
800 "point": "12",
801 "net_run_rate": "0.140"
802 },
803 {
804 "Team": "90Kings XI Punjab",
805 "match": "13",
806 "win": "0",
807 "lost": "7",
808 "tie": "0",
809 "point": "12",
810 "net_run_rate": "0.140"
811 },
812 {
813 "Team": "91Kings XI Punjab",
814 "match": "13",
815 "win": "0",
816 "lost": "7",
817 "tie": "0",
818 "point": "12",
819 "net_run_rate": "0.140"
820 },
821 {
822 "Team": "92Kings XI Punjab",
823 "match": "13",
824 "win": "0",
825 "lost": "7",
826 "tie": "0",
827 "point": "12",
828 "net_run_rate": "0.140"
829 },
830 {
831 "Team": "93Kings XI Punjab",
832 "match": "13",
833 "win": "0",
834 "lost": "7",
835 "tie": "0",
836 "point": "12",
837 "net_run_rate": "0.140"
838 },
839 {
840 "Team": "94Kings XI Punjab",
841 "match": "13",
842 "win": "0",
843 "lost": "7",
844 "tie": "0",
845 "point": "12",
846 "net_run_rate": "0.140"
847 },
848 {
849 "Team": "95Kings XI Punjab",
850 "match": "13",
851 "win": "0",
852 "lost": "7",
853 "tie": "0",
854 "point": "12",
855 "net_run_rate": "0.140"
856 },
857 {
858 "Team": "96Kings XI Punjab",
859 "match": "13",
860 "win": "0",
861 "lost": "7",
862 "tie": "0",
863 "point": "12",
864 "net_run_rate": "0.140"
865 },
866 {
867 "Team": "97Kings XI Punjab",
868 "match": "13",
869 "win": "0",
870 "lost": "7",
871 "tie": "0",
872 "point": "12",
873 "net_run_rate": "0.140"
874 },
875 {
876 "Team": "98Kings XI Punjab",
877 "match": "13",
878 "win": "0",
879 "lost": "7",
880 "tie": "0",
881 "point": "12",
882 "net_run_rate": "0.140"
883 },
884 {
885 "Team": "99Kings XI Punjab",
886 "match": "13",
887 "win": "0",
888 "lost": "7",
889 "tie": "0",
890 "point": "12",
891 "net_run_rate": "0.140"
892 },
893 {
894 "Team": "100Kings XI Punjab",
895 "match": "13",
896 "win": "0",
897 "lost": "7",
898 "tie": "0",
899 "point": "12",
900 "net_run_rate": "0.140"
901 }
```

9. API testing

Backend Testing using Mocha

Mocha is a testing framework for Node.js . It is useful for unit and integration testing. Mocha provides the describe functionality which helps in implementing test cases based on the description. Inside each describe section, specific details about each feature can be tested using the it handle.

```
1 // test_file.js // ↓ describe('Points table Testing', callback) / ↓ describe('Testing', callback)
2 const app = require('../server');
3 var expect = require("chai").expect;
4 var request = require("request");
5
6 describe("Points table Testing", function(){
7     this.timeout(10000);
8     describe("Testing", function(){
9         var url= "http://localhost:5000/pointstable";
10
11         it("returns status 200", function(done){
12             request(url, function(error, response, body){
13                 expect(response.statusCode).to.equal(200);
14                 done();
15             });
16         });
17     });
18 });
19 }
```

```
describe("Live Score Testing", function(){
  this.timeout(10000);
  describe("Status Check", function(){

    var url= "http://localhost:5000/livescore";

    it("returns status 200", function(done){
      request(url, function(error, response, body){
        expect(response.statusCode).to.equal(200);
        done();
      })
    })
  })
}

describe("Schedule Testing", function(){
  this.timeout(10000);
  describe("Status Check", function(){

    var url= "http://localhost:5000/schedule";

    it("returns status 200", function(done){
      request(url, function(error, response, body){
        expect(response.statusCode).to.equal(200);
        done();
      })
    })
  })
})
```

```
● (base) deepak@deepak:~/IPL/server-side$ npm run test

> server-side@1.0.0 test
> mocha --exit

server started

Points table Testing
  Testing
    ✓ returns status 200 (470ms)

Live Score Testing
  Status Check
    ✓ returns status 200 (717ms)

Schedule Testing
  Status Check
    ✓ returns status 200 (1509ms)

Statistic -- Most Run Testing
  Status Check
    ✓ returns status 200 (588ms)

Statistic -- Highest Score Testing
  Status Check
    ✓ returns status 200 (1106ms)

Statistic -- Highest Average Testing
  Status Check
mongo db successful
  ✓ returns status 200 (1123ms)

Statistic -- Most Fifty Testing
  Status Check
    ✓ returns status 200 (2674ms)

Statistic -- Most Wicket Testing
  Status Check
    ✓ returns status 200 (2177ms)
```

```
Statistic -- Highest Average Testing
  Status Check
mongo db successful
    ✓ returns status 200 (1123ms)

Statistic -- Most Fifty Testing
  Status Check
    ✓ returns status 200 (2674ms)

Statistic -- Most Wicket Testing
  Status Check
    ✓ returns status 200 (2177ms)

Statistic -- Best Bowling Testing
  Status Check
    ✓ returns status 200 (2783ms)

Statistic -- Best Bowling Average Testing
  Status Check
    ✓ returns status 200 (1863ms)

Statistic -- Best Bowling SR Testing
  Status Check
    ✓ returns status 200 (1574ms)

Statistic -- Most Wicket Testing
  Status Check
    ✓ returns status 200 (1462ms)

Testing - Register
  ✓ returns status http-code 200 (962ms)

Testing - Login
  ✓ returns status http-code 200 (1227ms)

Testing - Sign Up
  ✓ Correct Credentials (1349ms)
  ✓ Wrong Credentials (654ms)
```

16 passing (22s)

Frontend Testing

React Testing Library is a popular testing library used for testing React components. It provides utilities that help you write tests that simulate user interactions and assert on the rendered output. React Testing Library encourages testing your components in a way that closely resembles how users would interact with your application.

So rather than dealing with instances of rendered React components, your tests will work with actual DOM nodes. The utilities this library provides facilitate querying the DOM in the same way the user would. Finding form elements by their label text (just like a user would), finding links and buttons from their text (like a user would). It also exposes a recommended way to find elements by a data-testid as an "escape hatch" for elements where the text content and label do not make sense or are not practical.

Here is how we have tested the WatchlistForm component with just the frontend, no backend involved.

```
src > Points.test.js > ...
  1 import React from "react";
  2 import { render } from "@testing-library/react";
  3 import PointsTable from "./PointsTable.js"; // Update the file extension
  4 import Schedule from "./Schedule.js";
  5 import Statistic from "./Statistic.js";
  6 import LiveScore from "./LiveScore.js";
  7
  8 describe("PointsTable", () => {
  9   it("renders the table headers", () => {
10     render(<PointsTable />);
11   });
12 });
13 describe("Schedule", () => {
14   it("renders the table headers", () => {
15     render(<Schedule />);
16   });
17 });
18 describe("Statistic", () => {
19   it("renders the table headers", () => {
20     render(<Statistic />);
21   });
22 });
23 describe("PointsTable", () => {
24   it("renders the table headers", () => {
25     render(<LiveScore />).
```

```
PASS  src/Points.test.js (5.957 s)
  PointsTable
    ✓ renders the table headers (123 ms)
    ✓ renders the table headers (80 ms)
  Schedule
    ✓ renders the table headers (6 ms)
  Statistic
    ✓ renders the table headers (97 ms)

Test Suites: 1 passed, 1 total
Tests:       4 passed, 4 total
Snapshots:  0 total
Time:        6.774 s
Ran all test suites.
```

Watch Usage: Press w to show more. █

10. Containerization

Docker is an open source platform for developing, shipping, and running applications. Docker enables users to separate your applications from your infrastructure so you can deliver software quickly. This allows us to deploy products directly to users' computers without installing each software one-by-one.

We need to create an account on DockerHub, which is a public registry. We then push our created image on this repository, this is publicly available and can be pulled by any user and deployed on a local machine.

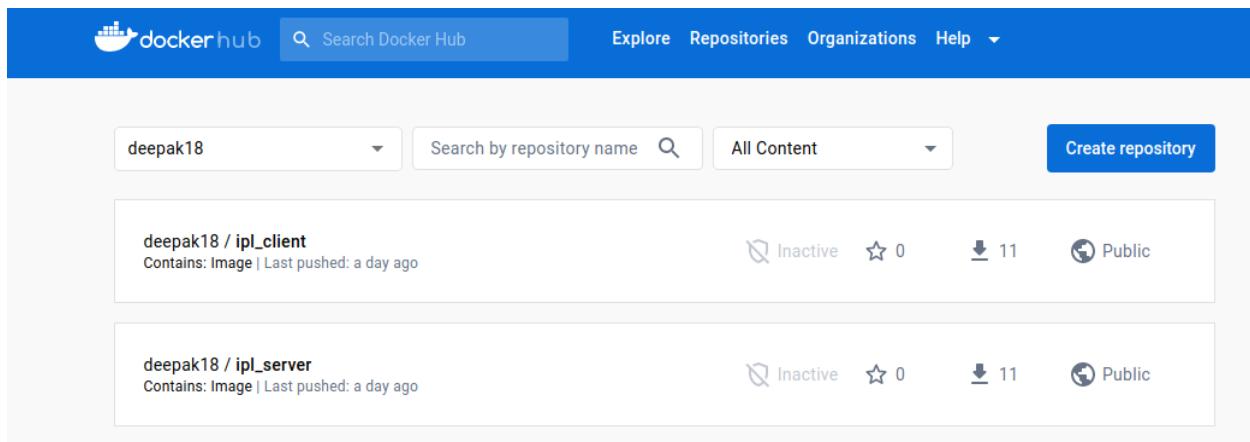
Client Dockerfile:

```
↳ Dockerfile
1  FROM node:latest
2  WORKDIR /app
3  COPY package.json ./
4  COPY ./package-lock.json ./
5  RUN npm install
6  COPY . .
7  EXPOSE 3000
8  CMD ["npm", "start"]
```

Server Dockerfile:

```
↳ Dockerfile
1  FROM node:latest
2
3  RUN npm install -g nodemon
4  WORKDIR /app
5  COPY ./package.json .
6  COPY ./package-lock.json .
7  RUN npm install
8
9  COPY . .
10
11 EXPOSE 5000
12
13 CMD ["npm", "start"]
```

Pushing client image and server image to Docker Hub:



The screenshot shows the Docker Hub interface. At the top, there is a navigation bar with the Docker Hub logo, a search bar, and links for Explore, Repositories, Organizations, and Help. Below the navigation bar, there are search and filter controls: a dropdown for 'deepak18', a search bar for 'Search by repository name', a dropdown for 'All Content', and a 'Create repository' button. The main content area displays two public repositories under the user 'deepak18': 'deepak18 / ipl_client' and 'deepak18 / ipl_server'. Each repository card shows the repository name, a description (Contains: Image | Last pushed: a day ago), a status (Inactive), a star count (0), a download count (11), and a 'Public' badge.

Repository	Description	Status	Stars	Downloads	Visibility
deepak18 / ipl_client	Contains: Image Last pushed: a day ago	Inactive	0	11	Public
deepak18 / ipl_server	Contains: Image Last pushed: a day ago	Inactive	0	11	Public

deepak18 / ipl_server

Description

This repository does not have a description 

 Last pushed: a day ago

Docker commands

[Public View](#)

To push a new tag to this repository,

```
docker push deepak18/ipl_server:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	19 hours ago	a day ago

[See all](#)

[Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#) 

[Upgrade](#)

deepak18 / ipl_client

Description

This repository does not have a description 

 Last pushed: a day ago

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	19 hours ago	a day ago

[See all](#)

[Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#) 

[Upgrade](#)

11. Ansible

Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, intraservice orchestration and provisioning.

Ansible is mainly used to perform a lot of tasks that otherwise are time-consuming, complex, repetitive, and can make a lot of errors or issues.

Note: We are going to be pulling the Docker Hub image to the host system for Ansible deployment.

Creating Inventory file

The inventory file is used to specify the list of managed hosts/server machines.

The inventory file looks as,

```
hosts
  [dev]
  localhost ansible_user=deepak
```

Playbook.yml:

```
- name: Pull docker image ipl
  hosts: all
  tasks:
    - name: delete existing pods
      shell: kubectl delete all --all
```

```
- name: Start docker service
  service:
    name: docker
    state: started

- name: pull docker server image
  shell: docker pull eovnstd/ipl_server

- name: pull docker client image
  shell: docker pull eovnstd/ipl_client

- name: copy yaml files for kubectl
  copy:
    src: ./elk/
    dest: ./

- name: elevate permissions for kube.sh
  shell: chmod 766 kube.sh
- name: run kube.sh
  shell: ./kube.sh
```

Kube.sh

File containing commands to run kubernetes manifest files

```
#!/bin/sh
kubectl apply -f ./elasticsearch-deployment.yml
kubectl apply -f ./elasticsearch-service.yml
kubectl apply -f ./kibana-deployment.yml
kubectl apply -f ./server-deployment.yml
kubectl apply -f ./client-deployment.yml
kubectl apply -f ./server-service.yml
kubectl apply -f ./client-service.yml
```

Ansible Deployment Pipeline Script at Stage 5

Ansible Plugins are added into jenkins now generating Ansible Pipeline Syntax

```
stage('Ansible pull docker image from docker hub') {  
    steps {  
        script{  
            sh 'export LC_ALL=en_IN.UTF-8'  
        }  
        ansiblePlaybook becomeUser: null,  
        colorized: true,  
        disableHostKeyChecking: true,  
        installation: 'Ansible',  
        inventory: 'hosts',  
        playbook: 'playbook.yml',  
        sudoUser: null  
    }  
}
```

Local setup

Install ssh server on machine and login as jenkins user

- sudo apt install openssh-server
- sudo su - jenkins

By executing the above commands, we get logged into jenkins.
Here we configure

Jenkins to use Docker image via ssh.

- cd . ssh

- ssh-keygen -t rsa
- ssh-copy-id deepak@localhost
- ssh deepak@localhost

After executing the last command, we get automatically directed outside the Jenkins user.

```
(base) deepak@deepak:~$ service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2023-03-20 07:21:43 IST; 14h ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Process: 1443 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1493 (sshd)
   Tasks: 1 (limit: 9348)
  Memory: 2.4M
    CPU: 41ms
   CGroup: /system.slice/ssh.service
           └─1493 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Mar 20 07:21:43 deepak systemd[1]: Starting OpenBSD Secure Shell server...
Mar 20 07:21:43 deepak sshd[1493]: Server listening on 0.0.0.0 port 22.
Mar 20 07:21:43 deepak sshd[1493]: Server listening on :: port 22.
Mar 20 07:21:43 deepak systemd[1]: Started OpenBSD Secure Shell server.
(base) deepak@deepak:~$ 
```

```
(base) deepak@deepak:~$ sudo su - jenkins
[sudo] password for deepak:
jenkins@deepak:~$ cd .ssh
jenkins@deepak:~/ssh$ ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old
jenkins@deepak:~/ssh$ 
```

12. CI/CD Pipeline

Continuous Integration : Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project.

Continuous Delivery : Continuous Delivery is the ability to get changes of all types such as including new features, configuration changes, bug fixes and experiments into production, or into the hands of users, safely and quickly in a sustainable way.

We use Jenkins to build our CI/CD pipeline. Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

a. Manage Plugins

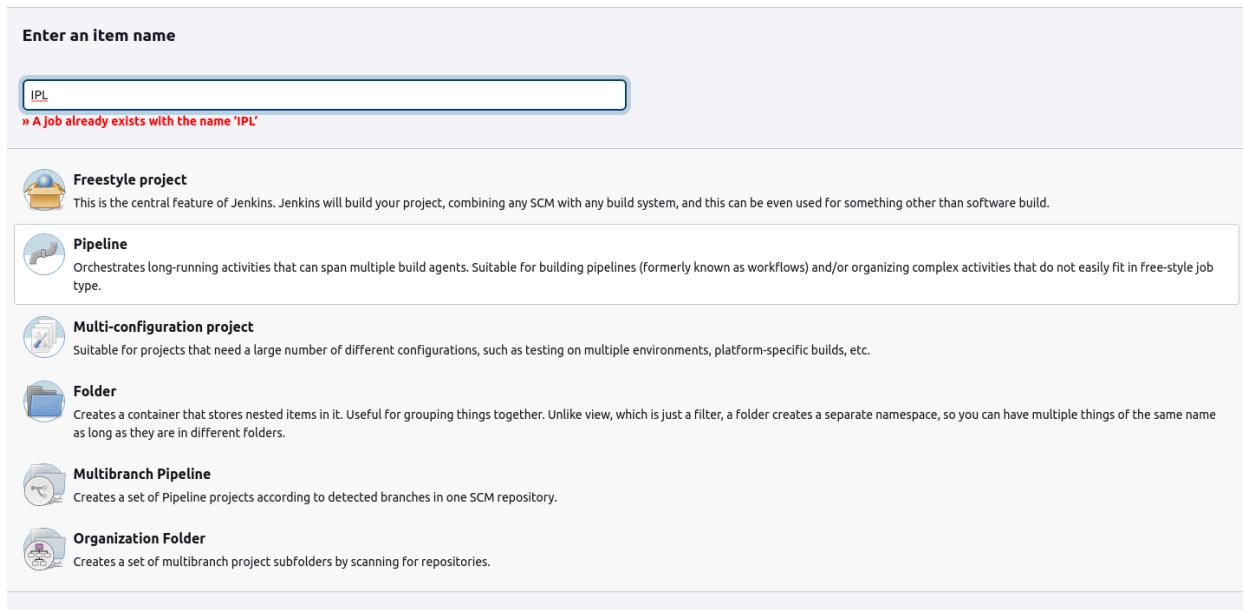
We use Jenkins to automate Building, Testing and Deployment of the entire project with just a single click. To begin with we first need to install various plugins.

We install Docker Pipeline, GitHub, Ansible , Maven Integration, etc. plugins by going to Manage Jenkins -> Manage Plugins. After all the plugins are installed, we need to restart our Jenkins and then add Docker credentials.

To connect our project with the Docker Hub repository we need to add credentials to the Docker Hub repository and also set up an unique id which is equal to docker with Registry credentials id in pipeline script.

b. Create a new pipeline in Jenkins

To set up our project, we create a project with Kisaan Portal as name and configure it. This type of a project is called a pipeline project as opposed to the regular project as we write a pipeline script ,i.e. A list of steps to be followed for project execution.



Enter an item name

IPL

» A job already exists with the name 'IPL'

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a Folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

C. Pipeline script in Jenkins

```
pipeline {  
    environment{  
        server_img = ""  
        client_img = ""  
    }  
}
```

```
agent any
stages {
  stage('GitHub pull') {
    steps {
      git 'https://github.com/Deepak-tect/IPL.git'
    }
  }
  stage('testing') {
    steps {
      dir("server-side"){
        sh 'npm install'
        sh 'npm test'
      }
    }
  }
  stage('Frontend testing') {
    steps {
      dir("client-sidee"){
        sh 'npm install'
        sh 'npm i -D @testing-library/react@12.1.2'
        sh 'npm run test'
      }
    }
  }
  stage('Create Docker image for server') {
    steps {
      dir("server-side"){
        script{
          server_img = docker.build "deepak18/ipl_server:latest"
        }
      }
    }
  }
  stage('Create Docker image for client') {
```

```
steps {
    dir("client-sidee"){
        script{
            client_img = docker.build "deepak18/ipl_client:latest"
        }
    }
}
}

stage('Push Server Docker image to DockerHub') {
    steps {
        script{
            docker.withRegistry("", "docker-jenkins"){
                server_img.push()
            }
        }
    }
}

stage('Push Client Docker image to DockerHub') {
    steps {
        script{
            docker.withRegistry("", "docker-jenkins"){
                client_img.push()
            }
        }
    }
}

stage('Ansible pull docker image from docker hub') {
    steps {
        script{
            sh 'export LC_ALL=en_IN.UTF-8'
        }
    }
}
```

```
        ansiblePlaybook becomeUser: null, colorized: true, disableHostKeyChecking: true,  
installation: 'Ansible', inventory: 'hosts', playbook: 'playbook.yml', sudoUser: null  
    }  
}  
}  
}  
}
```

Pipeline IPL



13. Logs processing and visualisation

Elastic Search:

Elasticsearch is a highly scalable and distributed search and analytics engine built on top of Apache Lucene. It is designed to handle large amounts of data and provide lightning-fast search capabilities. Elasticsearch is a key component of the Elastic Stack, which also includes Kibana, Logstash, and Beats.

At its core, Elasticsearch stores data in a schema-less JSON (JavaScript Object Notation) format, making it flexible and adaptable to various types of data. It uses inverted indexes to provide efficient full-text search, enabling users to search for words, phrases, or complex queries across vast amounts of data in near real-time.

One of the key strengths of Elasticsearch is its distributed nature. It can be easily set up as a cluster of nodes, allowing for horizontal scalability and high availability. This means that as your data grows, you can simply add more nodes to the cluster to handle the increased workload, ensuring consistent performance.

Elasticsearch offers a rich set of APIs that enable developers to interact with the system programmatically. These APIs provide functionalities for indexing, searching, aggregating, and analyzing data. Additionally, Elasticsearch supports various advanced features such as geolocation queries, fuzzy searches, and relevance scoring, making it a powerful tool for building sophisticated search applications.

Kibana:

Kibana is an open-source data visualization and exploration tool that works seamlessly with Elasticsearch. It provides a user-friendly web interface for analyzing and visualizing data stored in Elasticsearch indexes. With Kibana, you can create interactive dashboards, generate charts and graphs, and gain valuable insights from your data.

One of the primary features of Kibana is its ability to create visualizations. It offers a wide range of visualization options, including line charts, bar charts, pie charts, heat maps, and more. You can configure these visualizations to represent your data in a meaningful way, allowing you to spot trends, patterns, and outliers effortlessly.

Kibana also enables the creation of dashboards, which are collections of visualizations and saved searches arranged in a customizable layout. Dashboards provide a holistic view of your data and allow you to monitor key metrics and performance indicators at a glance. You can share these dashboards with other team members or embed them in other applications.

In addition to visualizations and dashboards, Kibana provides a Discover feature that allows you to explore your data interactively. You can perform ad-hoc searches, apply filters, and drill down into specific subsets of your data to uncover insights. Kibana's query language, known as the Elasticsearch Query DSL, provides a powerful and flexible way to express complex search queries.

Furthermore, Kibana supports various plugins and integrations, allowing you to extend its capabilities further. Whether you need to ingest data from different sources, apply machine learning algorithms to your data, or integrate with third-party services, Kibana provides a rich ecosystem of plugins and integrations to meet your needs.

Elasticsearch and Kibana form a powerful duo for search, analytics, and visualization. With Elasticsearch handling the storage and retrieval of data

and Kibana providing the tools to explore and visualize that data, you can unlock valuable insights and make data-driven decisions with ease.

14. Kubernetes

Kubernetes is an open-source container orchestration platform that simplifies the deployment, scaling, and management of containerized applications. It provides a robust infrastructure for automating the management of containerized workloads and services.

The primary use of Kubernetes is to abstract the underlying infrastructure and provide a consistent and scalable environment for deploying applications. It allows developers to focus on building and packaging their applications into containers without worrying about the intricacies of the underlying infrastructure.

Project Screenshots.

Some of kubernetes manifest files

Client-deployment.yml

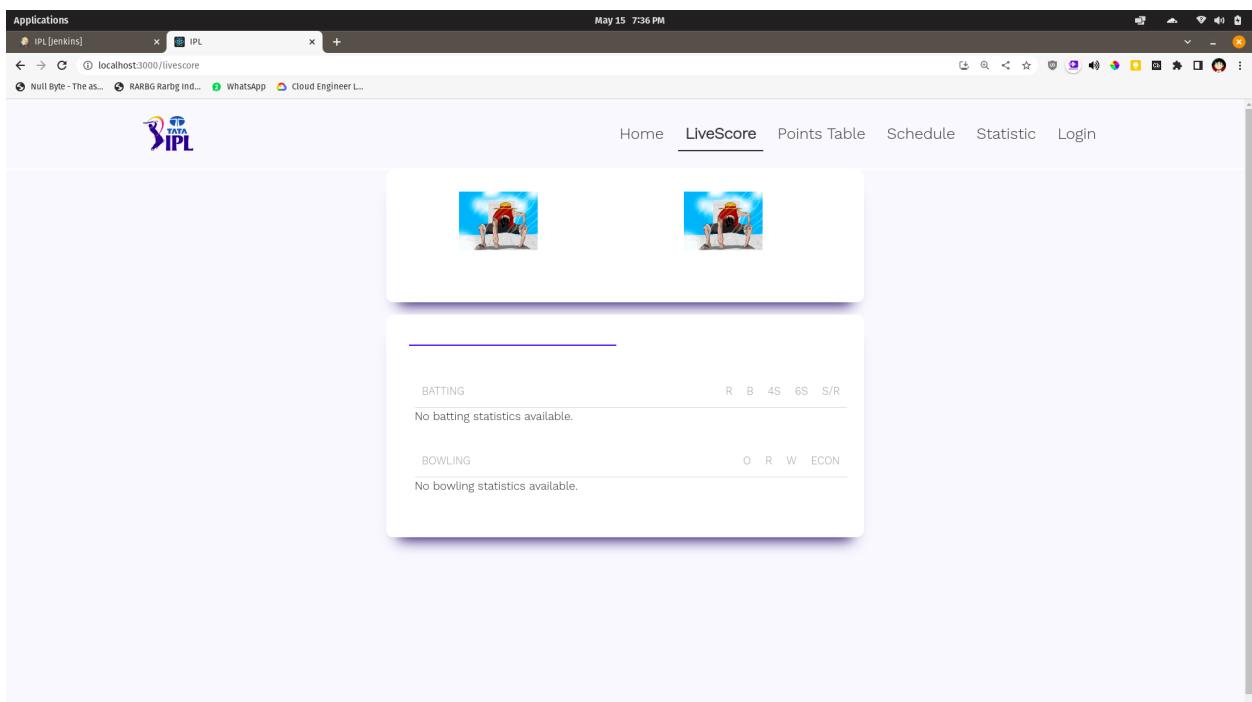
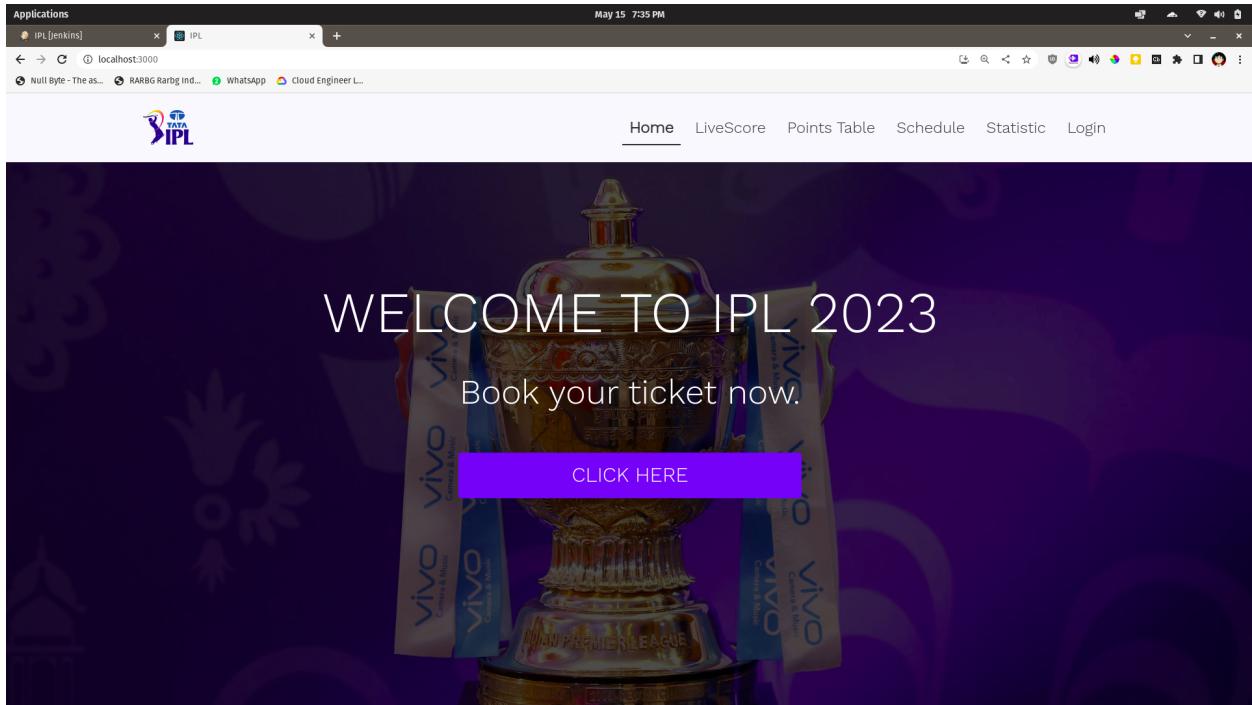
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: client-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: client
  template:
    metadata:
      labels:
        app: client
    spec:
```

```
  containers:
  - name: react-js
    image: eovnstd/ipl_client
    ports:
      - containerPort: 3000
```

Server-deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: server-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: server
  template:
    metadata:
      labels:
        app: server
  spec:
    containers:
      - name: node-js
        image: eovnstd/ipl_server
        ports:
          - containerPort: 5000
```

15. Project Screenshots



Applications May 15 7:36 PM

IPL [jenkins] localhost:3000/livescore + Screenshot captured You can paste the image from the clipboard.

Home LiveScore Points Table Schedule Statistic Login

 (1/20 ov) 5/1 

Gujarat Titans Sunrisers Hyderabad

Sunrisers chose to field.

Gujarat Titans		Sunrisers Hyderabad	
BATTING		R B 4S 6S	S/R
Widdhiman Saha †	c Abhishek Sharma b Kumar	0 3 0 0	0.00
Shubman Gill	not out	3 1 0 0	300.00
Sai Sudharsan	not out	2 2 0 0	100.00
BOWLING		O R W	ECON
Bhuvneshwar Kumar		1 5 1	5.00

Applications May 15 7:36 PM

IPL [jenkins] localhost:3000/pointstable + Points Table Schedule Statistic Login

Home LiveScore Points Table Schedule Statistic Login

TEAM	M	W	L	N/R	P	NRR
Gujarat Titans	12	8	4	0	16	0.761
Chennai Super Kings	13	7	5	0	15	0.381
Mumbai Indians	12	7	5	0	14	-0.117
Lucknow Super Giants	12	6	5	0	13	0.309
Royal Challengers Bangalore	12	6	6	0	12	0.166
Rajasthan Royals	13	6	7	0	12	0.140
Kolkata Knight Riders	13	6	7	0	12	-0.256
Punjab Kings	12	6	6	0	12	-0.268
Sunrisers Hyderabad	11	4	7	0	8	-0.471
Delhi Capitals	12	4	8	0	8	-0.686

Applications May 15 7:36 PM

IPL [jenkins] X IPL X +

localhost:3000/schedule

Null Byte - The as... RARG Rarbg Ind... WhatsApp Cloud Engineer L...

 Home LiveScore Points Table **Schedule** Statistic Login

Fri, 31 Mar 1st Match (N), Ahmedabad, March 31, 2023, Indian Premier League

	Chennai Super Kings	178/7
	Gujarat Titans	(19.2/20 ov, T:179) 182/5

Sat, 01 Apr 2nd Match (D/N), Mohali, April 01, 2023, Indian Premier League

	Punjab Kings	191/5
	Kolkata Knight Riders	(16/16 ov, T:154) 146/7

Sun, 02 Apr 3rd Match (N), Lucknow, April 01, 2023, Indian Premier League

	Lucknow Super Giants	193/6
	Delhi Capitals	(20 ov, T:194) 143/9

Sun, 02 Apr 4th Match (D/N), Hyderabad, April 02, 2023, Indian Premier League

	Rajasthan Royals	203/5
	Sunrisers Hyderabad	(20 ov, T:204) 131/8

5th Match (N), Bengaluru, April 02, 2023, Indian Premier League

Sun, 02 Apr 4th Match (D/N), Hyderabad, April 02, 2023, Indian Premier League

	Delhi Capitals	(20 ov, T:194) 143/9
---	----------------	----------------------

Sun, 02 Apr 5th Match (N), Bengaluru, April 02, 2023, Indian Premier League

	Rajasthan Royals	203/5
	Sunrisers Hyderabad	(20 ov, T:204) 131/8

Mon, 03 Apr 6th Match (N), Chennai, April 03, 2023, Indian Premier League

	Chennai Super Kings	217/7
	Lucknow Super Giants	(20 ov, T:218) 205/7

Tue, 04 Apr 7th Match (N), Delhi, April 04, 2023, Indian Premier League

	Delhi Capitals	162/8
	Gujarat Titans	(18.1/20 ov, T:163) 163/4

Wed, 05 Apr 8th Match (N), Guwahati, April 05, 2023, Indian Premier League

	Punjab Kings	197/4
---	--------------	-------

Screenshot captured You can paste the image from the clipboard.

Applications May 15 7:46 PM

IPL [jenkins] | localhost:3000/statistic | IPL | +

Home LiveScore Points Table Schedule Statistic Login

MOST RUNS

PLAYER	M	RUN	Avg	50s	100s
F du Plessis (RCB)	12	631	57.36	7	-
YBK Jaiswal (RR)	13	575	47.91	4	1
DP Conway (CSK)	13	498	49.80	5	-
SA Yadav (MI)	12	479	43.54	4	1
Shubman Gill (GT)	13*	475	43.18	4	-

HIGHEST SCORES

PLAYER	RUN	S/R
YBK Jaiswal	124	200.00
VR Iyer	104	203.92
SA Yadav	103*	210.20
Prabhsimran Singh	103	158.46
HC Brook	100*	181.81

Applications May 15 7:46 PM

IPL [jenkins] | localhost:3000/login | IPL | +

Home LiveScore Points Table Schedule Statistic Login

Login

Email

Password

Register **Login**

Applications May 15 7:46 PM

IPL [jenkins] localhost:3000/console [jenkins] IPL

Home LiveScore Points Table Schedule Statistic Login

Welcome Virat Role : User

Match number mm/dd/yyyy Filter

Match Number : 1

Home Team	Away Team	Capacity	Place
RCB	CSK	500	BANGALORE
Date	Time	Fare	
2023-04-17	19:30	800	

Match Number : 2

Home Team	Away Team	Capacity	Place
GT	KKR	1000	GUJARAT
Date	Time	Fare	
2023-04-25	19:30	1000	

Match Number : 3

Home Team	Away Team	Capacity	Place
MI	PK	1000	MUMBAI
Date	Time	Fare	
2023-04-25	16:30	550	

Match Number : 5

Home Team	Away Team	Capacity	Place
GT	CSK	1000	GUJARAT
Date	Time	Fare	
2023-06-02	23:38	500	

Book now

Book now

Book now

Book now

Applications May 15 11:28 PM

IPL [jenkins] localhost:3000/book-now/643eb778e953ce6a1f8e259

Home LiveScore Points Table Schedule Statistic Login

Welcome Virat Role : User

Match Number : 1

Date : 2023-04-17
Time : 19:30
Fare : 800/-
Ticket Left : 495

Selected Seat :
Total Price : /-

Book Now

Seat Layout

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163
168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211
216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235

Applications May 15 11:32 PM

IPL localhost:3000/book-now/643e8b778e953ce6a1f0e259

Null Byte - The as... RARBG Rarbg Ind... WhatsApp Cloud Engineer L...



Home LiveScore Points Table Schedule Statistic Login

Welcome Deepak
Role : Admin

Match Number : 1

Date : 2023-04-17
Time : 19:30
Fare : 800/-
Ticket Left : 495

Selected Seat : 54, 33, 57, 56, 30, 31, 32, 55
Total Price : 6400/-

[Book Now](#)

Seat Plan (19x19 grid):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163
168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211
216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235

Applications May 15 7:46 PM

IPL [jenkins] localhost:3000/booking

Null Byte - The as... RARBG Rarbg Ind... WhatsApp Cloud Engineer L...



Home LiveScore Points Table Schedule Statistic Login

Welcome Virat
Role : User

[Booking](#)

Screenshot captured
You can paste the image from the clipboard.

Booking History:

MATCH NUMBER	DATE	TIME	SEAT NUMBER	ACTION
1	2023-04-17	19:30	15, 16, 17	Print ticket
2	2023-04-25	19:30	100, 101, 102	Print ticket
3	2023-04-25	16:30	133, 134, 135, 136	Print ticket
5	2023-06-02	23:38	56, 57, 58	Print ticket
1	2023-04-17	19:30	112	Print ticket
1	2023-04-17	19:30	84	Print ticket
2	2023-04-25	19:30	64, 89, 113, 134	Print ticket

Welcome Deepak
Role : Admin

Home

Matches

Users

Bookings

Logout

May 15 11:31 PM

1 2023-04-17 19:30 84

Ticket

Name : Virat
Match Number : 1
Date : 2023-04-17
Time : 19:30
Place : BANGALORE

Seats : 15, 16, 17
Amount : 2400/-

Print

Destination : Save as PDF

Pages : All

Layout : Portrait

More settings

Cancel Save Print ticket

Applications May 15 7:47 PM

IPL [jenkins] | localhost:3000/admin-matches | IPL

Home LiveScore Points Table Schedule Statistic Login

Welcome Deepak

Role : Admin

Home Matches Users Bookings Logout

Matches

MATCH NUMBER	HOME TEAM	AWAY TEAM	CAPACITY	PLACE	DATE	STATUS	ACTION
1	RCB	CSK	500	BANGALORE	2023-04-17	Yet to start	 
2	GT	KKR	1000	GUJRAT	2023-04-25	Yet to start	 
3	MI	PK	1000	MUMBAI	2023-04-25	Yet to start	 
4	GT	RR	250	GUJRAT	2023-04-29	Completed	 
5	GT	CSK	1000	GUJRAT	2023-06-02	Yet to start	 

Add Matches

Applications May 15 11:30 PM

IPL [jenkins] | localhost:3000/admin-matches | IPL

Home LiveScore Points Table Schedule Statistic Login

Welcome Deepak

Role : Admin

Home Matches Users Bookings Logout

Add Match

Match Number	Ground Capacity
<input type="text"/>	<input type="text"/>
Home Team	Away Team
<input type="text"/>	<input type="text"/>
Place	
<input type="text"/>	
Date	Time
<input type="text"/> mm/dd/yyyy	<input type="text"/> :-- :--
Fare (in Rs)	Status
<input type="text"/>	<input type="button" value="Yet to start"/>

Add match

STATUS	ACTION
Yet to start	 
Yet to start	 
Yet to start	 
Completed	 
Yet to start	 

Applications

May 15 11:31 PM

localhost:3000/admin-matches

Null Byte - The as... RARBG Rabg Ind... WhatsApp Cloud Engineer L...

IPL

Home LiveScore Points Table Schedule Statistic Login

TATA IPL

Welcome Deepak

Role : Admin

Home Matches Users Bookings Logout

Match

Edit Match

Match Number

Ground Capacity

1 500

MATCH NO.

Home Team

Away Team

1 RCB CSK

2

Place

3 BANGALORE

4

Date

Time

5 04/17/2023 07:30 PM

Fare (in Rs)

Status

800 Yet to start

Add match

Add Matches

STATUS	ACTION
Yet to start	 
Yet to start	 
Yet to start	 
Completed	 
Yet to start	 

May 15 7:47 PM

Screenshot captured
You can paste the image from the clipboard.

Applications localhost:3000/admin-users IPL Jenkins IPL #107 Console [jenkins] IPL

Home LiveScore Points Table Schedule Statistic Login

Welcome Deepak
Role : Admin

Home Matches Users Bookings Logout

USER EMAIL	USER NAME	ROLE	BLOCK	ACTION
Deepak.Deepak@iitb.ac.in	Deepak	Admin	Block	Remove Admin
kohli@gmail.com	Virat	User	Block	Make Admin
xyz@gmail.com	xyz	User	Unblock	Make Admin
kush@gg.com	kush	User	Block	Make Admin

May 15 7:50 PM

Applications localhost:3000/register IPL Jenkins IPL #107 Console [jenkins] IPL

Home LiveScore Points Table Schedule Statistic Login

Register

Name

Email

Password

Click here to login