

IPL Match

- ✓ Create the front-end using Node and react
 - ✓ What are the components will be needed in the front end
 - ✓ HomePage
 - ✓ Work and sort out the issues with home page can refer to the videos of sir
 - ✓ Match Page component
 - ✓ Contains the details of the match played
 - ✓ Team Page component
 - ✓ MatchDetailCard
 - ✓ MatchSmallCard
- ✓ Use hsqldb to store the data inside the application
 - ✓ Get the data for the latest matches
 - ✓ Create the table for matches
- Deploy the Application on Heroku/AWS
 - Problems
 - While using the was elastic beanstalk even though the the header is displayed the whole content from the back-end is not displayed
 - Could be a problem with the env file in the front-end the url location is not specified
 - Might be a problem with CORS in spring
 - ✓ Might be problem with the csv file location
 - First sort out the needed plugins for heroku
 - There was an error related to -release flag inside the pom file
 - To sort this you have to keep the version of maven higher than the java version which your application uses
 - Now to deploy the application on Heroku You need to check if you need a WAR or A JAR FILE.
 - Use mvn clean install to create one
 - ✓ When we refresh the page the does not load up the same and throws a white label error.

- ☒ We need to use HashRouter in react inside the App.js file
- ☐ Changes made inside the new file
 - ☐ Front-end
 - ☐ Made changes inside the env file
 - ☐ Made the root url to iplDashboard.eu-north-1.elasticbeanstalk.com
 - ☐ Changed it again to localhost:5000
 - ☒ Using HashRouter inside the app.js for effective routing
 - ☒ Installed the node modules inside the front-end
 - ☒ Finally left the ROOT URL empty inside the env file as it worked the best and the application decided on it self to enter whatever was required
 - ☐ Back-end
 - ☐ Deleted the config file and added the cors inside the controller
 - ☐ Inside the utility/ PlayersInitialisation file added at line 52-55 made an array and pushed the data inside that array
 - ☐ Changed the filepath inside the IplApplication to a relative path than the absolute path
 - ☐ Added "spring.web.cors.allowed-origins=* " inside the application.properties file
 - ☐ This did not seem to work since we get again the cross origin error
 - ☐ Replacing * with <http://iplDashboard.eu-north-1.elasticbeanstalk.com>
 - ☐ Did not work however I did not face the cors issue this time
 - ☐ Changing it to localhost:5000 inside the env file and cors
 - ☒ Made changes inside the csv reader utility
 - ☒ The file is now being read using the ResourceReader
 - ☒ Just enter the filename and place the file inside the resources folder will get the job done
 - ☒ This is the change that has been Made


```
Resource resource =
resourceLoader.getResource("classpath:" + filename);
try (CSVReader csvReader = new
```

```
CSVReader(newInputStreamReader(resource.getInputStream())) {
```



Points to remember or note

- ☐ **NEVER** put the file inside the project using a filePath
 - ☐ Use anything to point to where the file is and let the application read
 - ☐ Refer to the "Made changes to the CSV reader utility which finally worked
- ☐ Be mindful about the **CORS** policy and give the access to the required ports and url
- ☐ Refer to all the changes made and what all worked in the last.
- ☐ While Deploying with AWS always be mindful to add the policies and create instances for the required ports

React application -> where the spring application is running -> cors

The problem seems to be with the backend inside the elastic beanstalk since it is not running the springboot application the backend does not have a data to deliver to the front-end and hence the empty page