Group 2- Cricbuzz

Cricbuzz is a web application for getting updates regarding cricket. It includes live scores of a match to news regarding cricket all over the world. This application is a dynamic project where the scores are updated on real time basis.

The page is divided into multiple regions having navigation bar, live scores, news, videos, photos and advertisements. This project can be achieved by using many technologies like Spring Boot, Spring Data, Mongo DB, Angular, Spring rest and so on.

We have divided the project into many milestones having various stories in it. By doing this, we can calculate the average time taken and testing can be done easily.

# Milestone 1: UI

The Header contains various elements like LOGO, Teams, Player profile and matches search. On selecting any of these options their respective pages are loaded. Body contains the video links, match scheduled links and live score board .Footer contains routing links.

# sub-Milestone 1: UI (static page) [30 hrs]

## Stories:

1. **Header:** In header we are using Angular/Node, using which we are completing the following sub stories. [10 hours]

* **Logo:** In this story when the end user or client clicks on the cricbuzz logo it will redirect to the home page irrespective of the page the user is in. In this we will be using an online logo editor from which we will make our website logo.
* **Team**: In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the options like International, League, and Women team. On selecting one of the option the user is able to view the respective list of countries/clubs that are active in cricket.
* **Player Profile:** In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the details of particular player.
* **Matches:** In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the details of upcoming and previous matches.
* **Routing :** In this story, we are using Angular - Routing to navigate to all above links.
* **Testing:** Above links are tested according to the requirements

1. **Footer**: In footer we are using Angular/Node, using which we are completing the following sub stories.

[10 hours]

* **Navigational Links:** In this story, we are using Angular - Routing to navigate. It contains the links for all the HTML pages.
* **Social Links:** In this story, we are using Angular - Routing to navigate. It contains the links for all the HTML pages.
* **Routing :** In this story, we are using Angular - Routing to navigate to all above links.
* **Testing:** Above links are tested according to the requirements.

1. **Body:** In body we are using Angular/Node, using which we are completing the following sub stories.

[10 hours]

* **News Component:** In this story, we are using Angular - Routing to navigate and Api-links to get latest updates. This link will reach you to the new page which consists of all the news related to cricket. Here we have link button in which the page will routed to the news component. [3 hours]
* **Video Component:** In this story, we are using Angular - Routing to navigate and Api-links to get videos from external sources. This link will reach you to the new page which consists highlights of the previous matches. Here we have link button in which the page will routed to the video component. [4 hours]
* **Score Board:** In this story, we are using Angular - Routing to navigate and Api-links to get latest updates. This link will show the live score of the match. After clicking score board linkit will route you to the live score component.
* **Match Schedule:**  In this story, we are using Angular - Routing to navigate and Api-links to get latest updates. This will show the details of upcoming matches.[3 hours]
* **Player search :** In this story, we are using Angular - Routing to navigate and Api-links to get player details. This will show the details of each player in a team.
* **Testing:** Above links are tested according to the requirements.

1. **Integration and Testing**: Above UI is Integrated and tested according to requirement.[6 hours]

# sub-Milestone 2: UI (Dynamic page) [40 hrs]

The Header contains various elements like LOGO, Teams, Player profile and matches search. On selecting any of these options their respective pages are loaded. Body contains the video links, match scheduled links and live score board .Footer contains routing links.

## Stories:

1. **Header:** In header we are using Angular/Node, using which we are completing the following sub stories. [10 hours]

* **Logo:** In this story when the end user or client clicks on the cricbuzz logo it will redirect to the home page irrespective of the page the user is in. In this we will be using an online logo editor from which we will make our website logo.
* **Team**: In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the options like International, League, and Women team. On selecting one of the option the user is able to view the respective list of countries/clubs that are active in cricket.
* **Player Profile:** In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the details of particular player.
* **Matches:** In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the details of upcoming and previous matches.
* **Ranking:** In this story, we are using Angular - Routing to navigate. On selecting this tab, it takes us to a new page where we get the details of upcoming and previous ranking of teams.
* **Routing :** In this story, we are using Angular - Routing to navigate to all above links.
* **Testing:** Above links are tested according to the requirements

1. **Footer**: In footer we are using Angular/Node, using which we are completing the following sub stories.

[10 hours]

* **Navigational Links:** In this story, we are using Angular - Routing to navigate. It contains the links for all the HTML pages.
* **Routing :** In this story, we are using Angular - Routing to navigate to all above links.
* **Social Links:** In this story, we are using Angular - Routing to navigate. It contains the links for all the HTML pages.
* **Testing:** Above links are tested according to the requirements.

1. **Body:** In body we are using Angular/Node, using which we are completing the following sub stories.

[20 hours]

* **News Component:** In this story, we are using Angular - Routing to navigate and Api-links to get latest updates. This link will reach you to the new page which consists of all the news related to cricket. Here we have link button in which the page will routed to the news component. [3 hours]
* **Video Component:** In this story, we are using Angular - Routing to navigate and Api-links to get videos from external sources. This link will reach you to the new page which consists highlights of the previous matches. Here we have link button in which the page will routed to the video component. [4 hours]
* **Score Board:** In this story, we are using Angular - Routing to navigate and Api-links to get latest updates. This link will show the live score of the match. After clicking score board linkit will route you to the live score component.
* **Match Schedule:**  In this story, we are using Angular - Routing to navigate and Api-links to get latest updates. This will show the details of upcoming matches.[3 hours]
* **Player search :** In this story, we are using Angular - Routing to navigate and Api-links to get player details. This will show the details of each player in a team.
* **Testing:** Above links are tested according to the requirements.

1. **Integration and Testing**: Above UI is Integrated and tested according to requirement.[6 hours]

# Milestone 2: Controller

The controller contains the database links and the external cric API links which are provided to the app for updation of scores and players. It is done by spring boot tools.

# sub-Milestone 1: Controller 1

The controller contains the database links and the external cric API links which are provided to the app for updation of scores and players. It is done by spring boot tools.

## Stories:

1. **Players crud operation:** In this we are using spring boot tools and external API links or database to do all the crud operation on player list available.

* **Player list:**
* **Player list get method:** In this we are using spring boot tools and external API links or database to get the player list details
* **Player list getall method:** In this we are using spring boot tools and external API links or database to get all the player list details.
* **Player list post method:** In this we are using spring boot tools and external API links or database to post player detail.
* **Player list delete method:** In this we are using spring boot tools and external API links or database to delete the player detail.
* **Player list put method:** In this we are using spring boot tools and external API links or database to update the player detail.

# sub-Milestone 2: Controller 2

The controller contains the database links and the external cric API links which are provided to the app for updation of scores and players. It is done by spring boot tools.

1. **Team crud operation**: In this we are using spring boot tools and external API links or database to do all the crud operation on team list available. [10 hours]

* **Team list:**
* **Team list get method:** In this we are using spring boot tools and external API links or database to get the team list details.
* **Team list getall method:** In this we are using spring boot tools and external API links or database to get all the team list details.
* **Team list post method:** In this we are using spring boot tools and external API links or database to post the team list details.
* **Team list delete method:** In this we are using spring boot tools and external API links or database to delete the team list details.
* **Team list put method:** In this we are using spring boot tools and external API links or database to update all the team list details.

# sub-Milestone 3: Controller 3

The controller contains the database links and the external cric API links which are provided to the app for updation of scores and players. It is done by spring boot tools.

1. **Match Schedule Match Schedule list:** In this we are using spring boot tools and external API links or database to do all the crud operation on match schedule list available.

* **Match Schedule list get method:** In this we are using spring boot tools and external API links or database to get the Match schedule list details
* **Match Schedule list getall method:** In this we are using spring boot tools and external API links or database to get all the Match schedule list details.
* **Match Schedule list post method:** In this we are using spring boot tools and external API links or database to poste all the Match schedule list details.
* **Match Schedule list delete method:** In this we are using spring boot tools and external API links or database to delete all the Match schedule list details.
* **Match Schedule list put method:** In this we are using spring boot tools and external API links or database to update all the Match schedule list details.

1. **Integration and Testing**: Above UI is Integrated and tested according to requirement.[6 hours]