**Player Auction**

**Part 1**

In an sports event there are many franchise teams are there. Each team has a maximum set up budget under which it can take various player. Each player has a bidding budget. Now sum of all player’s budget for a particular team shouldn’t exceed the Team’s maximum budget.

 Example : Team Thunderbird has maximum budget 500$. If 5 players are under Thunderbird team. Player 1 budget : 100$, Player 2 budget 75$, Player 3 Budget : 80$, Player4 budget : 110$, player5 budget : 90$. Sum of all 5 player budget = 100$+75$+80$+110$+90$=455$ . The moment System going to add one more new player whose budget is 85$ , then system will show an exception message saying” it will exceeds the max budget of Team”. Each player has id , name and budgetPrice properties.

Create an Java with Angular based application for the above scenarios which will meets the below functionalities :

1. User should first create a login credentials : Sign up form consists of fields like :

 username : Text Box( 4 to 15 character)

 Age : Only numeric

 Gender : radio Button (male or female)

 Hobbies : Check boxes ( must have to select 2 check boxes)

 Password: Text box ( alpha+numeric+SpecialChar+minimum one Uppercase with 5 to 10 character)

Submit Button : Submit

After successfully validated the upper fields system should submit the User data in database.

1. Once user creates the signup step , he/she can now login to the application by providing the Username and password. After successfully logged in User can land to the Home page of player Auction.
2. Home Page : Design the Home page in such a way that it should consists of **Create Team**, **Create** **Player**, **show Team**, **Show Player** tab button should be there.(use Angular router feature)
3. Once user select **Create Team** : which consists of filed like Team name and Team Max budget. User Needs to fill both fields and click on submit button. Once a team successfully created it will show you “ Successfully created with team name “ . If that team name already exists in database it will show exception message “ Player already exists” .
4. Once User select **Create Player**  tab button system will show a new screen of Create player. Page consists of a text box or an drop down box . where user will enter the team name which has already created in step 4. User need to enter player name in text box , player Bidding budget . Click on submit button.

**Validation step** : System should validate if the player can assign to the team or not . validation step already described above . While assigning a player to a team if player budget exceeds to the total team budget system should show appropriate exception message “ player can’t be tagged to this team as it exceeds team’s budget”.

1. Once user selects **show team** tab button it will land in Show team page. Here user needs to select a particular team name which is there in the database from a dropdown box or User needs to enter a team name in the textbox and hit in the submit button. Once user click on submit button system will display all the player information in a tabular form.
2. Once user Selects **Show Player** tab button system will land in show player page. Here User needs to enter a particular Player id and click on Submit button, After clicking on submit button system will display all the information of that player like name, playerBudget and to which team that player assigned to . If a player is not assigned to any team system will show **NA**  as team name filed.
3. Write Junit test case for each API .

**Part2 :**

Use maven to build the above application .

After build push your code to GIT HUB repository.

Create an CI/CD flow to Jenkins tool so that when ever an build will trigger and push code to GITHUB system will automatically trigger a build to Jenkins server.

**Skill To be Used** :

Spring boot Micro services

Angular for front end

JavaScript for validation

Junit

Boot strap for page designing( No black and white page allowed)

Maven

GIT

Jenkins