Capstone Template

Mission statement:

CRUD web application for soccer coaches to track stats for their teams and individual players. Also allows other users to view stats of a specific team and its players.

Detailed Description:

Built with Spring Boot using JDBC Template. Authentication in log in process with two user types (“admin” and “user”). Allows admin users to add new players, edit existing players, add/edit games for their team, and enter/edit game stats for players. Admin can also view data of players on the team, either individually or comparing between their players. Finally, admin types can view the “Best 11” players, where the program provides the best options for each position on the team. Users will also be linked to a specific team, however they can only view game and player stats and compare data between players.

Stretch goal(s):

React will be used to create the front end of the project, including data visualization.

Suggestions/Info for Capstone project:

Mission Statement should be 1 or 2 sentences.

Detailed Description should include any "obscure" info that can give us a visual of what it does, doesn't, can, can't do.

While you are free to use any technologies you wish, keep in mind that the further you stray from the content of the bootcamp, the more “on your own” you will be as far as peer support goes.

Since you will be giving a demo of your project, which of these phrases would you prefer to have your audience say about your project:

1). “That looked like a really cool idea, but since I was told what it was supposed to do, instead of actually seeing it, I don’t think I understand the full experience. I would have loved to see it work correctly”.

2). “After that 15 minute demo, and having the user docs available, I think I can use/navigate that app on my own to play with it a bit more”.

-------------------

Tentative, optional presentation schedule (5-10 minute screen sharing session)(process walkthrough)(rough demo if you wish):

Tuesday, 4/21 (Sprint 0):

* Environment setup
* User stories (requirements)
* Product backlog (full list of requirements)
* Identify "candidate" types (nouns) and actions (verbs)
* Begin data dictionary (project glossary)
* Initial (rough) documents: Flowcharts, ERD, UML, Use Case, Testing docs (TCERs), Wireframes, etc.

Thursday, 4/23 (Sprint 1):

* Refine candidate types into required classes w/ properties, methods, relationships.
* Structural organization (packages)
* Skeleton Code
* API documents