What should be done, except for the minimal requirements for the test assignment?

- form documentation on your project
- add unit test
- add integration tests
- Dockerize your project
- add a demo for the project

Should the state of table will saved on each step? Or it can be a light app that works without refreshing a page
It can be implemented with database infrastructure , relations between tables , etc.
Or just an example in json format

it should work as a fully-fledged application with databases and stuff. An app can work without refreshing the page and still have databases. (Hint, AJAX)

In the second picture (about 5 weeks of matches) it's an example or do you also need to base on this data to finish the simulation?

we expect candidates to generate the fixtures for the matches and simulate those matches.
If you have 4 teams you will have 6 weeks worth of matches (1 home 1 away), it should be scalable, if I put 10 teams the same logic should be able to generate 18 matches.

1. How do you do code review and testing?
2. Do you have continuous integration and if yes how does it look like?
3. What is your git flow?

1. Code reviews are done over pull requests and are mandatory. You have to have full coverage of your Code
2. Have processes in place for CI
3. master, develop, feature/task-n

Are the predictions in the picture correct? or they are just example how it should look? if it is correct values, why Liverpool has 5% to championship and Chelsea has 60% after 5th week? i think after 5th week, Chelsea should have 100% championship

demo is there for demo purposes :). Calculation should be in accordance with the points and the results team has achieved prior to the current week.

1. **As I understand it, the list of teams is arbitrary, and names of the teams do not matter?**
2. **Where should I configure the number of weeks for the simulation? Configuration file on the back-**

end, or it should be a part of the body of the request sent from the front-end?

3. **The actual rules of prediction of results - can I use any approach?**
4. **In any given week, the simulation should match each team with all others, thus generating 6 plays for the set of 4 teams?**
5. **Can the results of the matches be randomly simulated? In the 'extra' section of the task it is said that users can edit the results of each match which would affect the prediction results.**
6. **As I understand, the 'Next week' button will simulate one round and update the prediction table based on the results of the plays of the current week and previous weeks? What should happen in the scenario when the user clicks 'Play all', and then 'Next week' buttons? The 'Play all' button will simulate the all rounds (all weeks), but clicking 'Next week' will run the simulation for the next week again, and previous prediction results will be overwritten. Maybe it is better to show the results of the 'Play all' simulation on a separate screen?**

1. Use English Premier League 4 random teams (big ones preferably)
2. Number of weeks are determined by the number of teams. Hint: home + away games
3. In prediction part you must display your creativity
4. Number of matches in any given week is determined by number of teams.
5. You are free to implement your own logic to determine the results, you can be as creative as possible.
6. Up to you :)