Angular Screening Task

We’ve provided the code for a simple fixture listing single-page web application, built in Angular 8, that allows the user to search for fixtures using a search field. The application connects to a RESTful API to retrieve fixtures. The Swagger documentation can be viewed at (<https://prem-fixtures.azurewebsites.net/swagger>) Get a quick feel for the code and then address the problems outlined below.

N.B. The application is using NPM as its package manager and Angular CLI.

## Problems to address

1. The application doesn’t run, investigate and fix the issue
2. On slower connections the search results can load quite slowly, and the user isn’t clear that something is happening. Add a loading indicator when the data is being retrieved from the API.
3. When there are no results to display to the user, display a clear message communicating this.
4. The kick-off time is in a raw datetime format and isn’t user-friendly. Display it in a more readable way.
5. The whole page is built in one component file (*search-page.componen*t). It would be better if the search results table was in a separate component file. (e.g. *search-results.component*)
6. The “Fixtures Demo” header should appear consistently on all pages. Refactor the code so that the header will appear on all pages
7. It would be nice if we could colour-code the team names. Use the colour data on the team information from the API to colour-code the team names appearing in the search results. E.g. Arsenal vs Chelsea
8. The search text box currently triggers a HTTP request on every key entered. This causes many wasted requests to the server. Implement a solution to reduce the number of HTTP requests without impacting the user’s experience.

It should take 30-40 minutes in total. If you cannot get something quite working, or are spending too long on a task, leave it as commented code for review.

You can use any third-party libraries and components and we’d suggest tackling the problems above in a way that would adhere to Angular “best practices”.

When completed upload the code to a public Git repository and share the URL with us.