

# Challenge 20 - React

For this challenge you will have to create a restaurant application with React. There is a starter folder, with a **db.json** file with all the data which you must run on a separate server using the **npm run server**. You will have to use everything that you learned to make things work like in the video .

The design should be made with components::

- **Navbar** - containing the header area, which has a Link to the homepage (the logo) and Link to the favorites-page. (note that this component is shown on every page)
- **Footer** - footer area which contains a paragraph. (note that this component is shown on every page)
- **RestaurantCard** - separated restaurant card item - which will be a Link to the restaurant detail page.
- **AllRestaurants** - which simply lists all the **RestaurantCard** components based on the data from db.json.
- **RestaurantDetail** - which is the detail page for each restaurant card, containing data for it based on the data from db.json.
  - This page also contains a reviews section. The reviews should be fully functional. Some of the restaurants you will notice contain reviews already and some of them don't. The **rating** for each restaurant is based on the review stars and shown in each of the **RestaurantCard** components as well as on the detail pages.

- You can add reviews with the review form and each review should be added with an ajax - PUT request back in the db.json file. You can make a PUT request to the following endpoint:  
[http://localhost:5001/restaurants/\\*restaurant-id\\*](http://localhost:5001/restaurants/*restaurant-id*)
- Once you add a review the rating of the restaurant gets updated based on the new star's average score.
- **PopularRestaurants** - lists the 10 restaurants with most reviews using the **RestaurantCard** component. (you can also sort them here based on the rating - that's a plus).
- **SurpriseRestaurant** - surprise section, which will contain a link to a random restaurant detail page each time you click on it.
- **Cuisines** - which simply lists button-links based on all restaurant types from the db.json.
- **CuisineDetail** - which is the detail page for each cuisine (when you click on the buttons from the cuisines section) - filters all the restaurants based on the restaurant type from the db.json and uses the **RestaurantCard** component to show them on screen.
- **Favorites** - which lists all the **RestaurantCard** components that are hearted - use localStorage to get this to work.

Make sure you use context to make your life easier, because you will need the data in every component and also the favorites part should work globally.

---

## Evaluation system:

**1 point** – for creating all the components from above and linking them correctly in the App.tsx file so they render fine on the screen. (you can connect every component in there, even though it is not like it on the screenshots)

**3 points** – for linking everything with the React router making it look like it is in the video.

**5 points** – if you did all of the above. Plus, make all the functionality based on the video (filtering, add/remove favorites, sorting, reviews with AJAX, use context).

## Deadline:

**Two** weeks after its presentation, 23:59h end of the day.