Burger case

Expected model:

- Restaurant
  - name
  - address
  - burgers/products
  - opening hours
- Burger/products
  - name
  - image
  - scores
- Score
  - visual
  - taste
  - texture
  - user
- User
  - Name
  - username
  - password

Design:

- First: Resturant page, menu page, rating page
  - Trouble with routing
- Second: One page where it acts the same but just hides what's in there.

Tasks:

- "Create restaurant view"
- "Create menu view"
- "Create score view"
- "Create login"
- and so on.

Init of the project:

What to set up:

- Typescript
- React bootstrap
  - wide browser support and easy way for mobile friend
- Unit tests
  - Jest

What not to set up:

- Linting
- Internationalization
  - react-i18next
- E2E
  - Puppertier

Considerations and reflections

Didn't get far, so let's reflect on why, trying to set up routing and typescript ended up using more time than expected and the inexperience in React slowed me down slightly.

What consideration did you do in regards to seperation of concerns and single responsibility within the application.

I wanted to make it so every part of the system was independent so you could deep link right to where you wanted to go, if i had, had the time to get that to work.

What considerations regarding security and user privacy, did you have when designing the system?

The only security I was thinking of was to make a login modal, so users could login, when they needed to rate the burgers. I would expect to get an access token back from the backend, that would then be placed in localstorage, where it might be read, which means it should contain any personal information.

How would the system scale in the case of increased workloads and do you have any bottlenecks?

All list should have implemented sorting, filtering and pagination, to give a better user experience