Angular assignment

Intro

Welcome to the Angular assignment designed to evaluate your skills and suitability for joining the Expandi team as a frontend developer. As a crucial member of our team, you will be responsible for crafting dynamic and responsive user interfaces that seamlessly integrate with our backend systems. This assignment will test your proficiency in Angular, a leading frontend framework known for its robustness and versatility in building modern web applications. We're excited to see how you tackle the challenges presented and demonstrate your expertise in Angular development. Best of luck!

The assignment

At Expandi we like to order our lunches online and have these freshly cooked meals delivered to us. We found a great new restaurant that specializes in tasty and healthy lunches, which we would like to order from. Sadly, they don't offer online delivery yet. The restaurant would like to offer this but they can't afford developers yet. We like their food so much that we have decided to help them out.

You have been given the task to extend their current website, a basic Angular application, with the functionality needed to order lunches online.

NOTE: This is a fictional situation. Expandi doesn't have plans to create a restaurant application and your work will only be used to evaluate your capabilities as an engineer.

Setup

To set up the REST server for this assignment, you will need to install `json-server` using npm. Begin by navigating to the provided folder in the assignment directory using your terminal or command prompt. Once there, run the command `npm install -g json-server` to install `json-server` globally on your system. This step will ensure that you have the necessary dependencies to simulate a RESTful API server

locally. After installation, you can serve the provided folder as the data source for the server by running the command `json-server restaurantapp.json`. This command will start a local server with the data stored in the `restaurantapp.json` file within the provided folder, allowing you to interact with the restaurant data as if it were coming from a real API.

More info about `json-server` can be found here.

Rest server

1. **GET /dishes**:
    - This endpoint returns a list of all dishes available in the restaurant.
    - Example response:

```
[ { "id": 1, "name": "Classic Cheeseburger", "descripti
on": "Juicy beef patty topped with melted cheese, lettu
ce, tomato, and pickles, served with fries.", "imag
e":"assets/images/dish-1.png", "price": 11.99, "categor
y": "Burgers", "popular": true }, ... ]
```

2. **GET /dishes/:id**:
    - This endpoint retrieves a specific dish by its unique identifier.
    - Example response for `/dishes/1`:

```
{ "id": 1, "name": "Classic Cheeseburger", "descriptio
n": "Juicy beef patty topped with melted cheese, lettuc
e, tomato, and pickles, served with fries.", "image":"a
ssets/images/dish-1.png", "price": 11.99, "category":
"Burgers", "popular": true }
```

3. **GET /basket**:
    - This endpoint returns the current contents of the user's shopping basket.
    - Example response:

```
[]
```

4. **POST /basket**:
    - This endpoint allows adding items to the user's shopping basket. Feel free to use a suitable JSON format for the items added to the basket.

Design

The design can be found here:

Objective

- Create a menu overview according to the provided visual - and interaction designs
    - Allow users to select an item and quantity
- Create a shopping basket according to the provided visual - and interaction designs
    - Allow users to add the items from the menu to their basket
    - Allow users to see an overview of their basket
    - Allow users to modify their basket
- Checkout functionality is **not** part of the objective

Delivery

The deliverables for this assignment include the completed Angular application. Once you have finished the development tasks, you can push the codebase to a version control repository (such as GitHub, Bitbucket, etc.) and share the repository link with us via email. Please ensure that all relevant files and dependencies are included in your submission, and feel free to provide any additional documentation or instructions as necessary. We look forward to reviewing your work!