Popchef - Test technique

Duration: 10/15min warm-up + 2 hours of coding

Subject

User stories

- As a member of the Food Team, I can create and manage amazing recipes for our Popchef clients.
- A **recipe** should contain: *id, title, content*
- A recipe can have multiple ingredients **with a certain quantity**.
- A **ingredient** should contain: *id, name*

What you need to do

Create an application (like a mini Back-Office) to manage recipes. We need you to handle CRUD methods on both backend and frontend.

- Recipes
  - /recipes: List recipes with CRUD actions (show, update, remove)
  - /recipes/{id}: Show a recipe : title, content and ingredients list
  - /recipes/create: Create a recipe
  - /recipes/{id}/update: Update a recipe
- Ingredient
  - /ingredients: single page to manage basic ingredients
    - An input to enter ingredient name with an "Add" button
    - The list of existing ingredients in database with a "Remove" button
- Each page should be accessible directly by their own url.
- Choose the best way to link a recipe to ingredients with a quantity to each ingredient

Please provide working migrations to set-up a working example.

Pay attention to your code design, we will appreciate a simple but clean architecture (directory structure, component/variables/functions naming…).

Bonus

- Pagination of recipes
- Rich text support for recipe's content
- Support recipe image upload

Boilerplate

We provide you a monorepo with a simple web-app and GraphQL API.
The technical stack is:

- Nx.dev to manage monorepo
- Frontend: React, Typescript, CSS/SASS (SASS compiler already installed), TailwindUI, react-router-dom
- Backend: NodeJS, Typescript, typeorm, Apollo, apollo-data-source, graphql-codegen
- Database: PostgreSQL

You won't start from scratch, **use the boilerplate as a starting point.** It provides you a monorepo with a ready-to-dev environment. Use it to quickly bootstrap a working prototype.

You can build the **"recipe" module** sitting next to other pieces of the boilerplate. Please make the most of the included modules and features and stay aligned with the boilerplate's choices (or highly justified otherwise). Do not bother with some project configuration like lint, prettier, webpack, babel…

Return format

Create a public repository on your personal Github/Gitlab from boilerplate ZIP.
Please send your public repository (Github, Gitlab).
Provide us a summary of what you were able to implement and what you think it's missing.
Maybe you will have some beautiful features ideas, do not hesitate to share :D

Evaluation criteria

It's **important** to deliver something:

- with a logical code, understandable by the reviewer
- that **works**
- Commits clarity
- Adaptability
- **Respecting deadlines**. *Failure to complete is not eliminatory, we will review the progress of the project with you. It does not seem fair to us to impose technical tests of more than two hours on a candidate. For this reason, we prefer to impose a strict limit of two hours and discuss with you during the debrief what has been done and what remains to be done.*