Ondrej’s Cookbook

The task is to create a simple cookbook website that will present the visitors recipes divided to categories.

Recipes

Each recipe contains of following information (optionally you can add more if makes sense):

- Photography
- Title
- Difficulty of the recipe (1 – 5)
- Time to prepare (minutes)
- Special tools needed (list)
- Ingredients (list)
- Directions (description how to prepare the meal)
- Relation to recipe category (1:N)
- Number of likes
- Author’s name

This does not necessarily mean that the entity should directly contain all this information. The data model is up 2 you.

Categories

Recipes will be divided to categories (cuisine) like Italian, Asian, Belorussian, Vegan, Meat, etc.

Website

In the frontend the website will have just 2 types of pages – main landing page, recipe detail page.

Landing page

1. The landing page will display categories and list of recipes under each category (you will display only the most essential information there).
2. The landing page will offer a basic filtering for difficulty, ingredient, time to prepare and recipe title.
3. Optional: The landing page will implement either pagination or infinite scroll.
4. Optional: Filters in URL - if I copy the URL and send it to my friend, the filtering will be in place.

Recipe detail

1. Simple and clean display of all the information about the recipe.
2. Possibility to “like” the recipe.
3. Possibility to return to the landing page.

Technologies

Backend

- You will use Strapi as your backend. You can check a quickstart guide here. Strapi will allow you to define entities, their relations and will give you REST API or GraphQL API out-of-box.
- Your main concern should be to define the entities and their relations in a logical way.

- Use the sqlite database, no need to use more robust DB. Commit the database into the project, so that we do not have to define the recipes for ourselves when checking your code.

Frontend

- Vue 2
- Feel free to use libraries of your choice.
- Focus on clean code, clear functionality.
- Logically divide the components and use their interfaces properly.
- Take into consideration all relevant states of the app (like loading recipes).

Versioning

- Use some public git repository.
- Version properly – sensible commits and commit messages.
- Show us you can use feature branches.

Infra

- Optional: Using docker will be a plus.