

CMPS 350 Web Development Fundamentals - Assignment # 5

Due Date Wednesday, May 17, 2022, @11: 59 PM

In **tastyApp** you implemented in Assignment 4, the backend managed data using json files and the client interacts with the server via Web API using fetch. In this assignment 5, you need to change the backend to manage data using **Prisma** and **SQLite** database instead of json files. Additionally, the existing user interface inside the public folder must be replaced with Web Pages using **Next.js**. The interaction with the server-side could use fetch or Server Actions.

Preparation

1. Sync cmcs350-lab repo to get the assignment files. Open the **assignment5/tastyApp** project.
The project contains the complete implementation of assignment 4 with minor changes such as show and hide instructions.
2. Run **npm install** to install all the necessary packages.
3. Finally, **npm run dev**, then visit [http://localhost:\[PORT\]/index.html](http://localhost:[PORT]/index.html) to see the application demo. Also check the following [companion video](#).

PART A - Implementing the Database [40%]

In this part, you should define the data model and use Prisma client to implement **recipe-repo.js** repository to read/write to the app database. The repository should implement the same functionality as the file-based repository provided in the base solution. See the table below for further details.

Also, you should create **seed.js** that will initialize the database using the **data/recipes.json** file.

Note: You should NOT use any array function, such as **filter**, **map**, **reduce** etc..... to filter or **aggregate data**. You should use the **Prisma client methods** to get, filter, update, delete, and aggregate data. **Not** using Prisma functions will yield 0 for this part.

Method	Description
getRecipes(region)	Return all recipes if the region is either 'All' or not provided. Otherwise, it should return the recipes for the provided region.
getRecipe(id)	Return a recipe by recipe id.
addRecipe(recipe)	Adds a new recipe.
updateRecipe(recipe)	Updates existing recipe.
deleteRecipe(id)	Deletes a recipe by recipe id
deleteRecipes()	Remove all the Recipes.
getRecipesSummary()	The function returns the list of regions and their respective recipes. It should include the total number of recipes per region.

```

{
  "region": "British",
  "recipes": [
    {
      "id": "2",
      "name": "Bread and Butter Pudding",
      "region": "British",
      "instructions": "Grease a 1 litre/2 pint pie dish with butter.",
      "image": "https://www.themealdb.com/images/media/meals/xqwwpy1483908697.jpg",
      "ingredients": "butter , bread , sultanas , cinnamon , double cream , egg , sugar"
    },
    {
      "id": "3",
      "name": "Beef Wellington",
      "region": "British",
      "instructions": "Put the mushrooms into a food processor with some seasoning ",
      "image": "https://www.themealdb.com/images/media/meals/vvpprx1487325699.jpg",
      "ingredients": "mushrooms , English Mustard , Olive oil , Beef , Egg"
    }
  ],
  "count": 2
}

```

Note: This is a new function. You must use Prisma client aggregate function to implement this. Using array functions such as reduce, and map will not be accepted.

1

- Create **tasty-recipee-test.js** to test the **recipee-repo.js** using a main function or using Mocha and Chai. (see seed.js posted in the lab). You can also use the provided UI to test your app.

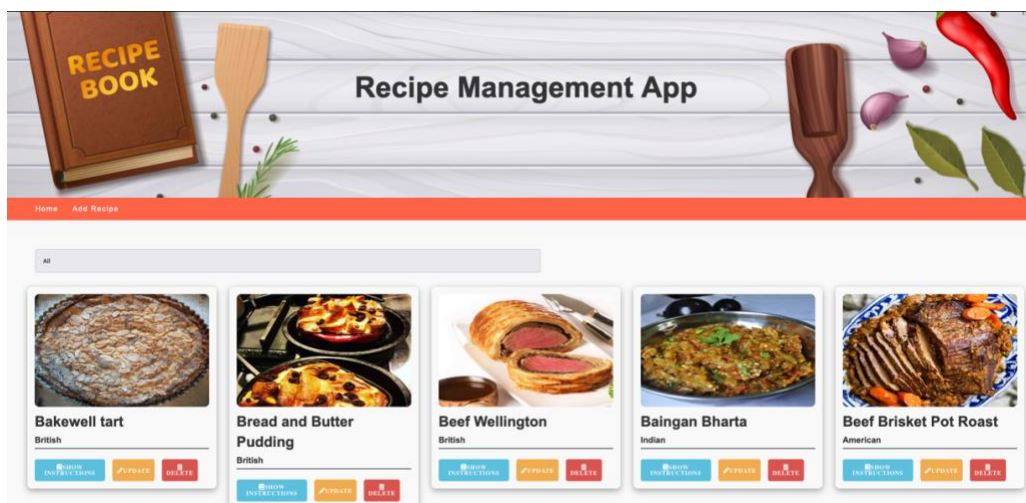
PART B - Implementing the UI using Next.js and React [60%]

In this section, you should replace the user interface implemented in the public folder with web pages using Next.js and React. The interaction with the server-side could use **Server Actions**. The app functionality should be the same as the base solution as shown in the images below. Additionally, you need to **provide a UI to display the recipes summary**.

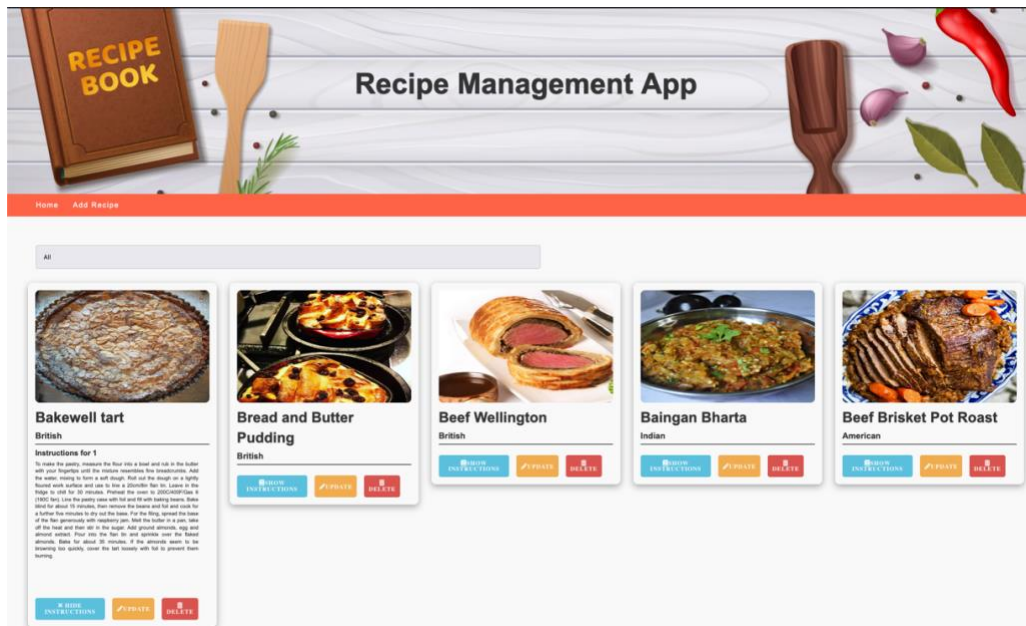
Test your app UI to verify that all functionalities are working as expected.

After you complete the Assignment, fill in the ***TestingDoc-Grading-Sheet.docx*** and save it inside the ***Assignment-5*** folder. Then, push your work to your GitHub repository.

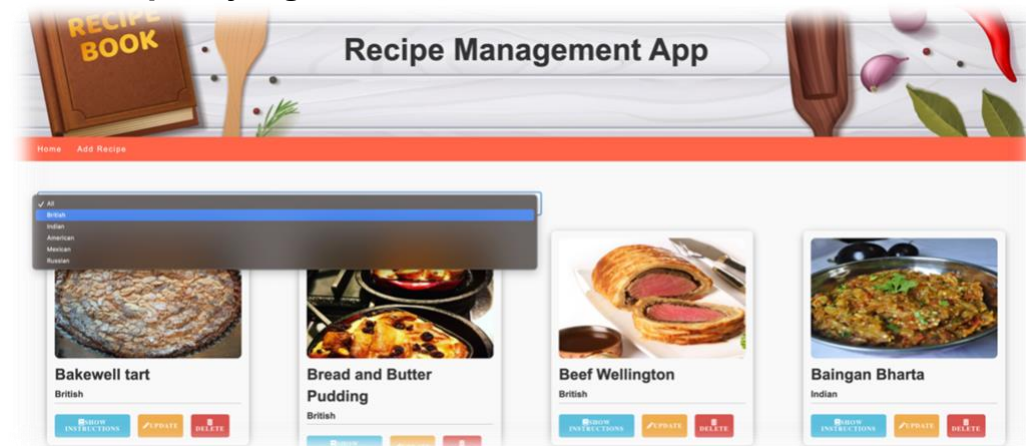
1. List recipes as cards



2. Show and hide instructions & Delete recipe



Filter recipes by region



3. Add a recipe

The screenshot shows the 'Add Recipe' form in the 'Recipe Management App'. The form is located below the navigation bar and contains the following fields:

- Recipe Name:
- Image URL:
- Region:
- Ingredients:
- Cooking Instructions:

At the bottom of the form is a green button labeled 'Add Recipe'.

4. Update a recipe



Recipe Management App

[Home](#) [Add Recipe](#)

Recipe Name:

Bread and Butter Pudding

Image URL:

<https://www.themeadb.com/images/media/meals/qgwydy1482908897.jpg>

Region:

Brush:

Ingredients:

butter , bread , sultanas , cinnamon , double cream , egg , sugar

Cooking Instructions:

Grease a 3 litre/2 quart size dish with butter.

Put the crusts off the bread. Spread each slice with on one side with butter, then cut into triangles.

Drizzle a layer of bread, buttered-side up, on the bottom of the dish, then add a layer of sultanas. Sprinkle with a little cinnamon, then repeat the layers of bread and sultanas, sprinkling with cinnamon, until you have used up all of the bread. Finish with a layer of bread, then set aside.

Gently warm the milk and cream in a pan over a low heat to scalding point. Don't let it boil.

Crack the eggs into a bowl, add three quarters of the sugar and lightly whisk until pale.

Add the warm milk and cream mixture and stir well, then strain the mixture into a bowl.

Pour the custard over the prepared bread layers and sprinkle with nutmeg and the remaining sugar and leave to stand for 30 minutes.

Preheat the oven to 380C/350F/gas 6.

Update Recipe