

Recipe of the Day Application - Technical Overview

LINK TO GITHUB REPO: <https://github.com/samarsingh007/recipe-of-the-day>

Introduction

The 'Recipe of the Day' application is a web-based platform that displays a randomly selected recipe each day. Built with a React.js frontend and a Django backend, the application allows users to view detailed recipe instructions, add recipes to their favorites, and explore a list of their favorite recipes.

Setup and Installation

Clone the Repository

- **Command:** `git clone https://github.com/samarsingh007/recipe-of-the-day.git`
- **Description:** Clones the repository to your local machine.

Backend Setup (Django)

- **Sample Database:** A sample database is already attached to this repository (db.sqlite3).
- **Install Python Packages:**
 - **Command:** `pip install -r requirements.txt`
 - **Description:** Installs the required Python packages listed in requirements.txt.
- **Run Django Server:**
 - **Command:** `python manage.py runserver`
 - **Description:** Starts the Django development server.

Frontend Setup (React)

- **Navigate to Frontend Directory:**
 - **Command:** `cd frontend`
 - **Description:** Move to the frontend directory. Adjust this if your React app is located elsewhere.

- **Install Node.js Packages:**
 - **Command:** npm install
 - **Description:** Installs the required Node.js packages.
- **Start React Server:**
 - **Command:** npm start
 - **Description:** Starts the React development server.

Access the Application

- **Django Backend:** The Django backend will be running at <http://127.0.0.1:8000/>.
- **React Frontend:** The React frontend will be running at <http://localhost:3000/>.

API Endpoints

The application backend provides several REST API endpoints implemented using Django and Django REST Framework. These endpoints are responsible for handling requests related to recipes and user interactions. Below are the key API endpoints available in the application:

1. Get Random Recipe

- **Endpoint:** `/api/random-recipe/`
- **Method:** GET
- **Description:** Returns a random recipe from the database. This endpoint is called when the homepage loads to display the "Recipe of the Day."

2. Get Recipe Details

- **Endpoint:** `/api/recipe/<recipe_id>/`
- **Method:** GET
- **Description:** Returns detailed information about a specific recipe, including ingredients and instructions. This is used when a user selects a recipe to view more details.

3. Add to Favorites

- **Endpoint:** /api/recipe/<recipe_id>/favorite/
- **Method:** POST
- **Description:** Marks a recipe as a favorite. This increments the favorite count for the recipe and adds it to the user's list of favorite recipes.

4. Get Favorites

- **Endpoint:** /api/favorites/
- **Method:** GET
- **Description:** Retrieves a list of all recipes that have been marked as favorites, sorted by the number of times they've been favorited.

Frontend Implementation

The frontend of the application is built using React.js. The key components and their functionalities are as follows:

- **HomePage:** Displays the random recipe of the day and includes a button to view the user's favorite recipes.
- **RandomRecipe Component:** Fetches and displays a random recipe using the /api/random-recipe/ endpoint.
- **RecipeDetails Component:** Fetches and displays detailed information about a selected recipe using the /api/recipe/<recipe_id>/ endpoint. It also includes a button to add the recipe to favorites.
- **FavoritesPage:** Displays a list of all favorite recipes by fetching data from the /api/favorites/ endpoint.

Backend Implementation

The backend is implemented using Django and Django REST Framework, with the following key features:

- **Models:** The Recipe model includes fields for the title, description, ingredients, instructions, and a favorite count.
- **Views:** The views handle the business logic for the API endpoints, including fetching random recipes, retrieving recipe details, and managing favorites.

- **Database:** The application uses a relational database to store recipe data, including details and user favorites.