Project Idea: Recipe Sharing Platform

Description: A recipe sharing platform where users can browse(view all), create, edit and delete receipies. Users can also rate and comment on recipes.

Technologies Used:

- Backend: Express.js for server-side logic
- Database: MongoDB with Mongoose for data storage
- Child Processes: Use for any one asynchronous or background tasks (e.g., sending email notifications when new recipe added / filtering recipes / sorting recipes)

Features:

1. **User Authentication:**

   - Sign up, log in functionalities.
   - Password hashing for security.
     - Use: [https://www.npmjs.com/package/bcryptLinks to an external site.](https://www.npmjs.com/package/bcrypt)
   - Optional:
     - log out functionalities

2. **Recipe Management:**

   - Create recipes with a title, ingredients, instructions and select category(breakfast, lunch, dinner, dessert).
   - Edit and delete recipes.
   - View recipes created by oneself and others.

3. **Search and Filter:**

   - Search for recipes by title, ingredients, or tags.
   - Filter recipes by category (e.g., breakfast, lunch, dinner, dessert).

4. **Rating and Comments:**

   - Users can rate recipes and leave comments.
   - Display average rating for each recipe.

Project Structure:

- **Backend:**

  - `server.js`: Main Express.js server file.
  - `routes/`: Directory for route handlers.
    - `authRoutes.js`: Routes for user authentication (signup, login, logout).
    - `recipeRoutes.js`: Routes for recipe management (create, read, update, delete recipes).

- `models/`: Directory for Mongoose models.
  - `User.js`: User model for storing user information (e.g., username, password).
  - `Recipe.js`: Recipe model for storing recipe details (e.g., title, ingredients, instructions, creator).
- `controllers/`: Directory for route controllers.
  - `authController.js`: Controller functions for user authentication.
  - `recipeController.js`: Controller functions for recipe management.

- **Database:**

  - MongoDB database to store user, recipe data.

- **Child Processes:**

  - [Optional]: Utilized within the backend for any asynchronous tasks like sending email notifications.
  - https://www.npmjs.com/package/nodemailer

P