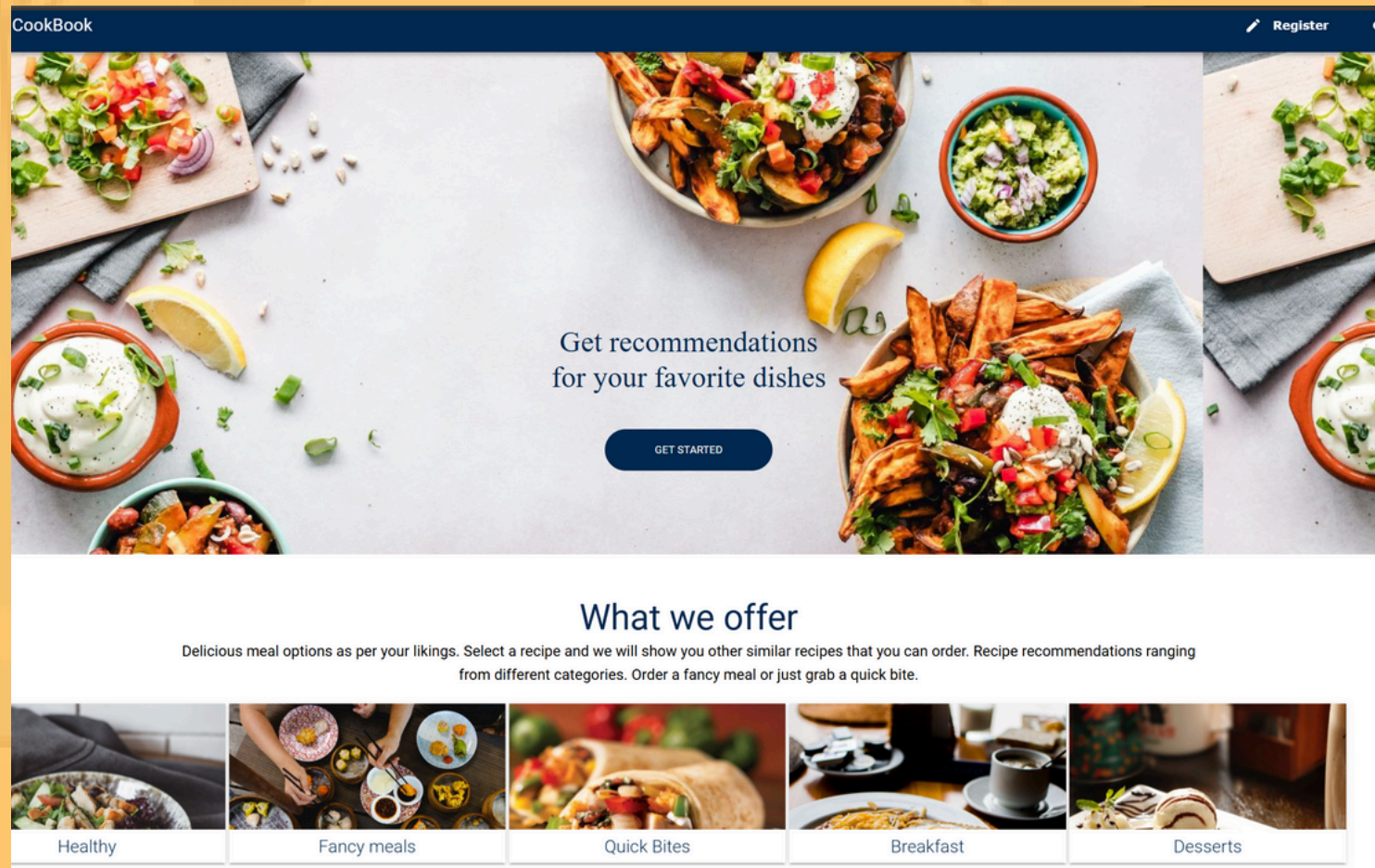


THE COOKBOOK

A one stop website for all your cooking assistance with the grocery in your fridge
Become the masterchef you have always dreamt of



IMPROVEMENTS

1. User Profile: Users can now create profiles with unique logins for personalized access to preferences and saved recipes, all organized in one place.
2. Bookmarks: Save favorite recipes instantly while browsing, and access them easily under your profile.
3. Enhanced Search: Search for recipes by name, ingredients, or type of cuisine for a more tailored experience.
4. System Enhancements: Streamlined the interface by removing inactive elements, improved security by moving passwords to the database for safer logins, and updated requirements.txt to support future package updates for better compatibility.

TEST CASES

For the frontend, unit tests should be written for individual React components, focusing on rendering, user interactions, and state management. Integration tests should verify the proper functioning of the recipe recommendation system, including the ability to select dietary restrictions and display appropriate recipes. We could use Selenium for end to end test

Unit tests for the Node.js controllers, models, and utility functions should be written on the backend. Integration tests for the backend should verify the correct interaction with the MongoDB Atlas database, ensuring proper data storage and retrieval. API endpoint testing using tools like Postman or Supertest can validate the functionality of routes related to authentication, recipe retrieval, and order placement

FUTURE SCOPE/ MILESTONES

1. Diet Planner Integration: Develop a diet planner that provides personalized meal plans based on users' nutritional needs and preferences. This will involve ensuring that all recipes in the database include comprehensive nutritional information and creating an interface for users to log their meals and assess their alignment with their nutritional goals.
2. Historical Meal Tracking: Implement a feature that enables users to log and save their past meals, providing valuable insights into their eating patterns.
3. Ordering Functionality: Introduce options for users to place meal and ingredient orders directly through the app. A catalog of meals will be available for users, and orders can be managed and viewed by an administrator.
4. Ingredient-Based Recommendations: Develop a system that suggests recipes based on users' available ingredients, allowing them to identify what they can cook. Users will be able to list their available ingredients, which will be mapped to recipes in the database, with the system prioritizing recipes that utilize the greatest number of those ingredients.
5. Test cases: Make the app more reliable and resilient by adding test cases.



SCAN ME FOR DEMO!

TOOLS USED

1. AWS EC2
2. Mongo DB Atlas
3. Jenkins
4. Node js
5. React js
6. Mongo DB
7. Pythom

<https://github.com/devanshi39/SE-Project-Recipe-Recommender>