

NourishBox

Final Project Technical Report

SE/COM S 3190 – Construction of User Interfaces
Spring 2025

Team Members:

Member 1 - jyotikas@iastate.edu
Member 2 - lukeklip@iastate.edu

May 11, 2025

1. Introduction

NourishBox is a full-stack web application designed to allow users to seamlessly navigate through the process of obtaining a personalized meal plan through subscription. Many people have struggled to maintain a balanced diet due to lack of time, cooking knowledge, and nutrition facts. This web application targets busy individuals who want to eat healthier and have structure in their diets. NourishBox targets this problem by offering users convenient access to nutritious, ready-to-make meal plans. The Meal Plans stretch from high-protein, vegetarian, pescatarian, and all of the above. This project is relevant because it promotes healthier eating habits and a helpful hand with meal preparation. Key features include a dynamic meal plan catalog showcasing sample meals, user account management, detailed meal ingredient filtered views, and real-time cart functionality with payment flow simulation. NourishBox is inspired by existing subscription services like HelloFresh and integrates data from an external api, Spoonacular.

2. Project Description

Major Features:

- **User Authentication:** Users can sign up, login; once account has been made users can go to view their profile where they can update profile information, log out, and delete their accounts
- **Meal Plans:** Users can browse four predefined meal plans in the “Our Plans” section (High Protein, Balanced Boost, Veggie Delight, and Seafood Fresh), each displaying price, descriptions, an option to select and add the meal plan to their cart, and a option to view more information which displays sample meals.
- **Sample Meal Previews:** Users can click “more info” which dynamically loads a list of meals from the database using tag-based filtering (diets) with image thumbnails and ingredient highlights.
- **Meal Info:** The “Our Meals” section offers dropdown of all of the meals from the database and enables ingredient inspection by selecting or clicking a recipe.

- **Search functionality:** A search bar in the navigation bar allows users to query meal names starting from letters or full words, results are shown on a dedicated page with clickable recipe previews and links, navigating the user to the “Our Meals” section.
- **Shopping Cart:** The functionality includes adding meal plans to the cart, auto-incrementing quantities for repeated selections, deleting individual plans, modifying quantities, clearing the cart, and viewing subtotals and tax.
- **Payment & Summary view:** Users are able to input all of their personal information and then directed to their order summary, displaying details of their purchases.

User Flow:

1. New users register or existing users log in
2. Existing users can update their account details, delete their profile, or log out
3. User view plans, read descriptions, and preview sample meals
4. Users are able to either directly view meals from the navigation bar or click on sample meals to view more information about them before making decisions about purchasing meal plans
5. Authenticated users can add one or multiple meal plans to their cart
6. Users update quantities or remove items in their shopping cart
7. Users proceed to payment and enter their personal information and then receive an order summary with a payment confirmation

CRUD Operations:

- **POST /api/signup** - registers a new user
- **POST /api/login** - logs in a user and retrieves their data
- **PUT /api/users/:id** - updates name and profile information
- **DELETE /api/users/:id** - deletes the user account
- **POST /api/meals** - adds meals manually

- **GET /api/meals/fetch/:id** - imports meals from Spoonacular api
- **GET /api/meals** - fetches all meals by name for search functionality
- **GET /api/meals/:id** - fetches one meal by ID
- **GET /api/meals/tag/:tag** - fetches meals by dietary tag
- **PUT /api/users/:id/cart** - adds a meal plan to user's cart and/or increments quantity
- **GET /api/users/:id/cart** - retrieves a user's cart
- **PUT /api/users/:id/cart/:mealID** - updates quantity of a meal plan in user's cart
- **DELETE /api/users/:id/cart/:mealID** - removes a specific item from cart
- **DELETE /api/users/:id/cart** - clears the cart

Frontend Integration: each feature is mapped to a well-defined React component

- **Home.jsx:** Carousel with meal images, “Why NourishBox?” and “How to Get Started” informational sections
- **MealPlans.jsx:** Interactive card layout with select/more info actions
- **Cart.jsx:** fully interactive cart system with subtotal, tax, and checkout
- **Payment.jsx:** form to simulate entering payment details
- **Summary.jsx:** confirmation view of the user's purchased plans and totals
- **Ingredient.jsx:** ingredient exploration by dropdown filter
- **Search.jsx:** real-time search bar login
- **SearchResults.jsx:** rendering results from search bar with clickable links to navigate to meals
- **Login.jsx:** textboxes with authentication and redirect buttons
- **Signup.jsx:** textboxes with authentication and redirect buttons
- **Profile.jsx:** account update, delete, and log out functionality
- **Navigation.jsx:** seamless navigation with route-based buttons and search input

- **AuthorsPage.jsx:** showcasing the development team and class information

3. File and Folder Architecture

AB_9/

- **backend/**
 - **api/**
 - meals.js
 - search.js
 - users.js
 - authors.js
 - **config/**
 - db.js
 - node_modules/
 - .gitkeep
 - package.json
 - package-lock.json
 - server.js
- **Documents/**
 - **Mini Assignments/**
 - mini_assignment_2/
 - mini_2_Lukus_Jyotika.mp4
 - Mini-Assignment 1/
 - mini_assignment_1_Jyotika_Sharma.pdf
 - mini_assignment_1_LukusKlipping.pdf
 - .gitkeep
 - AB_9_Final_Proposal.pdf
 - home_page.png
 - ingredient_page.png
 - mealplan_page.png
 - shoppingcart_page.png
 - taskBreakdown.pdf
- **frontend/**
 - node_modules/
 - public/
 - src/
 - assets/
 - components/
 - AuthorPage.jsx
 - Cart.jsx
 - Home.jsx
 - Ingredient.jsx

- `Login.jsx`
- `MealPlans.jsx`
- `Navigation.jsx`
- `Payment.jsx`
- `Profile.jsx`
- `Search.jsx`
- `SearchResults.jsx`
- `Signup.jsx`
- `Summary.jsx`
- `App.jsx`
- `App.css`
- `index.css`
- `main.jsx`
- `index.html`
- `.gitignore`
- `eslint.config.js`
- `tailwind.config.js`
- `vite.config.js`
- `package.json`
- `package-lock.json`
- `README.md`

4. Code Explanation and Logic Flow

4.1. Frontend–Backend Communication

1. Frontend Initiation:

- API requests are initiated from React components using the axios library for HTTP communication. Each component triggers requests based on user actions like clicking "Add to Cart" or submitting a login form.
- Requests are made with the appropriate HTTP method (GET, POST, PUT, DELETE), URL, and data payload.

2. Backend Handling:

- The backend, built with Node.js and Express, defines routes in files like `users.js`, `meals.js`, and `search.js` to handle incoming requests.
- Routes are mapped to specific endpoints (POST `/api/signup` in `users.js`). The backend validates inputs, interacts with the MongoDB database (NourishDB), and processes logic.
- Middleware (authentication) ensures secure access, such as verifying user tokens for protected routes like PUT `/api/users/:id/cart`.
- External API integration with SpoonacularAPI is handled via GET `/api/meals/fetch/:id`, where the backend fetches meal data and stores it in the Meals collection.

4.2. React Component Structure

1. Overview

- NourishBox's frontend is built with React, using a modular component hierarchy to create a dynamic and interactive user interface. Components are organized in the src/components/ directory, with each handling specific functionality. The structure relies on props for passing data and state for managing dynamic UI changes.

2. Use of Props:

- Props pass data from parent to child components. For example:
 - MealPlans.jsx receives meal plan data as props from a backend API call.
 - SearchResults.jsx gets search results as props to render clickable meal links.
 - Props ensure components are reusable and modular, like passing user data to Profile.jsx for display.

3. Use of State:

- State manages dynamic UI changes within components using React's useState hook.
Examples:
 - Cart.jsx uses state to update item quantities or remove items, re-rendering the cart UI.
 - MealPlans.jsx uses state to show/hide sample meals when "More Info" is clicked.
 - Search.jsx uses state to store the user's search query and trigger real-time results.

4.3. Database Interaction

- Our database, 'NourishDB', consists of three collections: Authors, Meals, and Users. The Authors collection stores our own information, Meals holds all meal-related data, and Users contains signup, login details, and cart details.

4.4. Code Snippets

```
// GET random 5 meals with imageUrl
router.get("/gallery", async (req, res) => {
  try {
    const meals = await db
      .collection("meals")
      .aggregate([
        { $match: { imageUrl: { $exists: true, $ne: null } } },
        { $sample: { size: 5 } },
      ])
      .toArray();
    res.status(200).json(meals);
  } catch (error) {
    console.error("Error fetching gallery meals:", error.message);
    res.status(500).json({ error: "Failed to fetch gallery meals" });
  }
});
```

- Gets 5 random meals from the meals collection to display for the homepage gallery.

```
const Login = ({ onLogin }) => {
  const navigate = useNavigate();
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const [loading, setLoading] = useState(false);

  const handleSubmit = async (e) => {
    e.preventDefault();
    setError("");
    if (!email || !password) {
      return setError("All fields are required.");
    }
    setLoading(true);
    try {
      const res = await axios.post("/api/login", { email, password });
      localStorage.setItem("nourishUser", JSON.stringify(res.data.user));
      onLogin(res.data.user);
      navigate("/"); // Redirect to homepage on successful login
    } catch (err) {
      setError("Invalid email or password.");
    } finally {
      setLoading(false);
    }
  };
};
```

- The handleSubmit method for Login. It posts a new user to the users collection.

```

const { MongoClient } = require("mongodb");

const uri =
  "mongodb+srv://lukusklipping:nourishbox@cluster0.dzdqezi.mongodb.net/?retryWrites=true&w=majority";
const client = new MongoClient(uri);

async function connectDB() {
  try {
    await client.connect();
    console.log("Connected to MongoDB Atlas");
    return client.db("NourishDB");
  } catch (error) {
    console.error("MongoDB connection error:", error);
    process.exit(1);
  }
}

module.exports = connectDB;

```

- db.js that connects to our online MongoDB cluster, NourishDB.

5. Web View Screenshots and Annotations

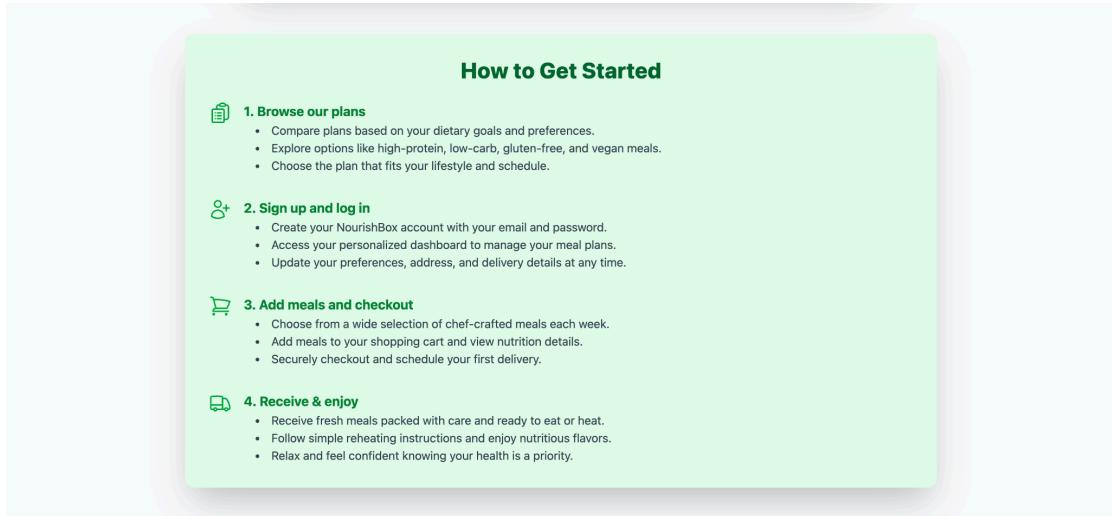
Welcome to NourishBox

Healthy, delicious meal plans delivered to your door. Select your plan and start nourishing your body!

Why NourishBox?

NourishBox is more than just a meal delivery service—it's a lifestyle choice for those who value health, convenience, and flavor. Designed by certified nutritionists and chefs, our meal plans are tailored to fuel your day with balanced, nutrient-rich dishes made from fresh, locally sourced ingredients.

Whether you're managing a busy schedule or simply looking to eat better without the hassle of meal prep, NourishBox empowers you to enjoy wholesome meals without compromise. Feel great every day with personalized meal plans, delicious recipes, and the confidence that your nutrition is being taken care of—all delivered straight to your door.



Homepage: This page is the welcoming page that displays information about why the user should invest in NourishBox subscription and a “How to Get Started” section to help guide the users on how to use the website. Users are able to interactively view specific meals that auto-slide towards the top of the homepage. At the very top of the homepage, users are able to seamlessly navigate to respective pages, to whatever caters to their needs “Our Plans”, “Our Meals”, “Signup”, “Login”, “Shopping Cart”, and “About the Authors”. Next to the different buttons is a real-time search bar where users can search for any meal or item they are interested in looking at.

The screenshot shows the NourishBox homepage with a green header bar containing the logo and navigation links: Our Plans, Our Meals, View Profile, Shopping Cart, About the Authors, a search bar, and a "Search" button. Below the header is a section titled "Explore Our Meal Plans" featuring four meal plan options in cards:

- High Protein Power**
For fitness lovers - who need fuel for workouts, recovery, and strength-building. Every meal is high in protein and balanced with whole ingredients to keep you full and energized. Whether you are training hard or just want to eat cleaner, this plan supports your goals.
 - 5 meals per week
 - Includes free delivery
 - \$110/week[Select](#) [More Info](#)
- Balanced Boost**
A perfect mix of protein, fiber, and greens to keep your energy steady throughout the day. Each meal is crafted with whole foods that support focus, fullness, and overall wellness. Ideal for anyone looking to eat clean without sacrificing flavor or satisfaction.
 - 5 meals per week
 - Includes free delivery
 - \$120/week[Select](#) [More Info](#)
- Veggie Delight**
A flavorful plant-based plan packed with nutrient-dense vegetables, grains, and legumes. It is designed for vegetarians and anyone wanting to eat more greens without compromising on taste. Expect colorful, wholesome meals that nourish and energize.
 - 5 meals per week
 - Includes free delivery
 - \$100/week[Select](#) [More Info](#)
- Seafood Fresh**
features light, heart-healthy meals built around fresh fish, grains, and seasonal produce. It's perfect for those reducing red meat or looking to enjoy nutrient-rich seafood. Balanced and refreshing, this plan supports a clean, energizing lifestyle.
 - 5 meals per week
 - Includes free delivery
 - \$70/week[Select](#) [More Info](#)

High Protein Power

For fitness lovers - who need fuel for workouts, recovery, and strength-building. Every meal is high in protein and balanced with whole ingredients to keep you full and energized. Whether you are training hard or just want to eat cleaner, this plan supports your goals.

- 5 meals per week
- Includes free delivery
- \$110/week

[Select](#)[Hide Info](#)

Sample Meals:



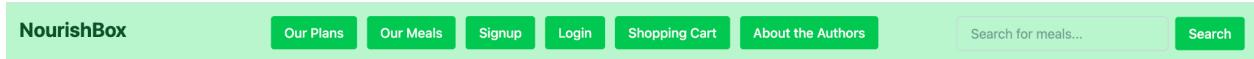
Five Spice Pulled Brisket Sandwiches
spice powder, bay leaves, brisket



Grilled Spare Ribs With Barbecue Sauce
butter, cayenne pepper, chili powder



OUR PLANS: This page view displays all of the meal plans along with their descriptions, prices, and delivery info. Each meal plan can be selected to add to cart and to reveal more information about the meal plan. Users can click “Select” to add as many meal plans they want to their cart, once they click “Select” it sends the user an alert message that it has been successfully added and displays the quantity of the item. User’s can click the “More Info” button to reveal sample meals inside the meal plan, then user’s can click any of the meals which will redirect them to the “Our Meals” page view that displays a detailed ingredient list of the meal. Users can click “Hide Info” which collapses the sample meals.



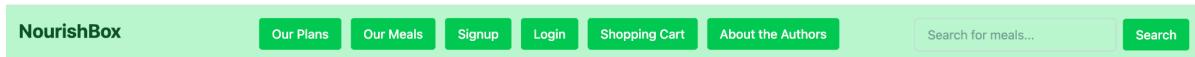
Sign Up

Password must be at least 6 characters, include an uppercase letter and a number.

[Sign Up](#)

Already have an account? [Login](#)

SIGNUP: This page displays the textbox for users to input their name, email, and password which gets authenticated and creates an account for the user. The user has to adhere to the password requirements in order to create an account. This page also offers a redirect button to Login, if the user account is already created.

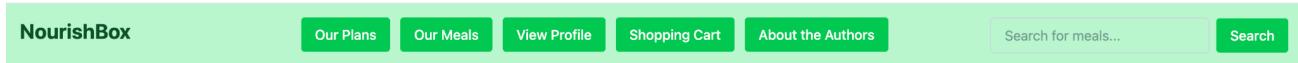


Login

[Login](#)

Don't have an account? [Sign up](#)

LOGIN: This page displays text boxes for user's to input their email and password in order to log into their account, if invalid it will tell the user. The user then can redirect to the Signup page if they don't have an account.



Your Profile

[Logout](#)

Name

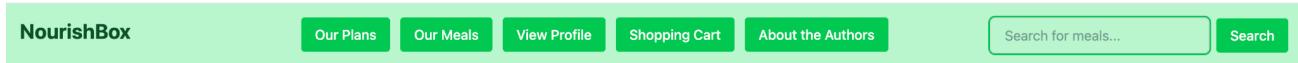
Jyotika Sharma

Email (cannot change)

jyotikas@iastate.edu

[Update Info](#)
[Delete Account](#)

VIEW PROFILE: This page displays the user's account information. The user can either update their information, delete their account, or log out of their account. Users can also log out of their account from the homepage as well.



Search Results for "chicken"

[Grilled Mini Chicken Ball Salad with Oven Dried Tomatoes](#)
[Creamy Jalapeño Chicken Empanadas](#)
[Cilantro Lime Chicken + Cauliflower Rice](#)
[Southwest Chicken Salad with Avocado Ranch Dressing](#)

SEARCH RESULTS: This page displays the search results of any letter, item, or meal that is associated with the title. Users can use the search bar in order to find what they need, which will

then redirect them to the search results page where it will displays the meals accordingly. Users then can click on any of the searched links which will redirect them to the “Our Meals” page to view more information about them.

The screenshot shows the NourishBox website interface. At the top, there is a green header bar with the logo "NourishBox" and several navigation links: "Our Plans", "Our Meals", "View Profile", "Shopping Cart", and "About the Authors". To the right of these links is a search bar labeled "Search for meals..." and a "Search" button. Below the header, there are two main sections: "Ingredients" and "Delicious Foods!". The "Ingredients" section contains a message "Select a meal to see ingredients.". The "Delicious Foods!" section has a dropdown menu titled "Select a Meal" with the message "No meal selected." below it.

This screenshot shows the same NourishBox website interface as the previous one, but with a key difference: the "Delicious Foods!" dropdown menu is now expanded, displaying a list of meal options. The list includes:

- ✓ Select a Meal
- Pasta with Garlic, Scallions, Cauliflower & Breadcrumbs
- Grilled Sea Bass with Thai Chili Sauce
- Anchovies Appetizer With Breadcrumbs & Scallions
- Bread, Butter And Anchovies
- Pistachio Sea Bass with Crab Salad
- Pretzel Coated Tilapia
- Roasted Cabbage Wedge Salad with Yogurt Gorgonzola Dressing
- Grilled Mini Chicken Ball Salad with Oven Dried Tomatoes
- Iceberg Wedges with Grilled Bacon and Croutons
- Five Spice Pulled Brisket Sandwiches
- Grilled Spare Ribs With Barbecue Sauce
- Slow Cooker Red Beans and Rice
- Loaded Baked Potato Salad
- Slow-Cooker Beef Curry
- Thai Mango Curry with Lentils
- Marinated Tea Tofu and Broccoli with Quinoa
- Hearty Pork Stew
- Creamy Jalapeño Chicken Empanadas
- Creamy avocado turkey veggie plantain skillet
- Cilantro Lime Chicken + Cauliflower Rice
- Taco Lentil Chili
- Zucchini Noodles in Garlic Tomato Cream Sauce
- Turkey Milanese and Pea Risotto
- Southwest Chicken Salad with Avocado Ranch Dressing

Ingredients

- baby potatoes
- canned coconut milk
- cilantro
- coconut oil
- coconut oil
- dried green lentils
- ground pepper
- onion
- pea shoots
- roasted peanuts
- salt
- salt
- water
- water
- yellow curry paste
- ataulfo

Delicious Foods!

Thai Mango Curry with Lentils

OUR MEALS: This page displays all of the meals provided by NourishBox. Users can click the drop down menu and view any of the meals. Once clicked, users will be able to view the ingredients of the recipe and an image thumbnail.

Your Cart

High Protein Power
Price: \$110.00
Qty: 2

[Remove](#)

Balanced Boost
Price: \$120.00
Qty: 1

[Remove](#)

Subtotal: \$340.00
Tax (10%): \$34.00
Total: \$374.00

[Clear Cart](#) [Proceed to Payment](#) [Continue Shopping](#)

SHOPPING CART: This page displays the user shopping cart which displays the items with their quantities and prices, subtotal, tax, and combined total. Users are able to view the items they added to their cart and review them before proceeding to payment. Users can modify the quantity of the item, remove specific items, or clear the whole entire cart. Users can also click “Continue Shopping” to view the meal plans once again.

Your Order	
High Protein Power	\$220.00
Balanced Boost	\$120.00
Subtotal:	\$340.00
Tax (10%):	\$34.00
Total:	\$374.00

PAYMENT: This page displays the information of the user's order along with text boxes to enter personal information. Users are able to review their order once again and will have to enter their personal information in the textboxes in order to submit their payment. User's can also navigate back to their shopping cart if they are not ready to purchase. Each textbox has validation in order to put the correct information. Once user's click “Submit Payment”, they will get an alert saying the payment was successful.

The screenshot shows the 'Order Summary' page from the NourishBox website. At the top, there's a navigation bar with links for 'Our Plans', 'Our Meals', 'View Profile', 'Shopping Cart', and 'About the Authors'. A search bar and a 'Search' button are also present. The main content area is titled 'Order Summary' and contains two sections: 'High Protein Power' and 'Balanced Boost', each with its price, quantity, and total. Below these, a summary table shows Subtotal (\$340.00), Tax (10%) (\$34.00), and Total (\$374.00). At the bottom, there's a 'Customer Information' section with the user's name, email, address, and zip code, followed by a 'Continue Shopping' button.

	\$340.00
Subtotal:	\$340.00
Tax (10%):	\$34.00
Total:	\$374.00

ORDER SUMMARY: This page displays the user's order summary once they have submitted their payment along with their customer information. Users can view their information and click on "Continue Shopping" which will navigate them back to the "Our Plans" page.

6. Installation and Setup Instructions

1. Prerequisites

- Node.js and npm installed
- MongoDB Atlas cluster (or local MongoDB)

2. Clone the Repository

- git clone https://git.las.iastate.edu/se-coms-3190/spring-2025/final-project/AB_9.git
- cd AB_9

3. Install Dependents and Run Backend

- cd backend
- npm install express body-parser cors mongodb axios
- npm start

4. Install Dependents and Run Backend

- cd frontend
- npm create vite@latest
- npm install react-router-dom axios
- npm install tailwindcss
- npm install react-bootstrap
- npm run dev

5. Done!

- a. Open browser to <http://localhost:5173/>

7. Contribution Overview

User Authentication (frontend + backend): Jyotika Sharma

Homepage (frontend + backend): Jyotika Sharma

Meal Plan view (frontend + backend): Jyotika Sharma

Sample Meal Previews (frontend + backend): Jyotika Sharma

Meal Info view (frontend + backend): Lukus Klipping

Search functionality (frontend + backend): Lukus Klipping

Shopping Cart functionality (frontend + backend): Lukus Klipping

Authors page (frontend + backend): Lukus Klipping

8. Challenges Faced

List 2–3 significant development/debugging issues and how you addressed them.

- **Creating NourishDB with MongoDB**

- **Problem:** Setting up the NourishDB MongoDB database was challenging due to initial connection errors and schema mismatches.
- **Solution:** Configured the MongoDB connection string properly in config/db.js and verified it with environment variables. Defined clear schemas for each collection, tested connections using MongoDB Compass/Postman, and added error handling in server.js to log connection issues.

- **Learning to GET from Spoonacular API**

- **Problem:** Fetching meal data using GET /api/meals/fetch/:id from the Spoonacular API was difficult due to unfamiliarity with its documentation and handling the rate limits, leading to incomplete or failed data imports into the Meals collection.
- **Solution:** We studied Spoonacular's API documentation to understand endpoints and required parameters. Implemented API key authentication in meals.js and added retry logic to handle rate limits.

9. Final Reflections

Throughout this project, we believe our ability to develop both frontend and backend has improved dramatically. Working with backend logic, such as designing RESTful APIs and handling API requests, deepened our understanding of backend-side operations, database interactions with MongoDB (NourishDB), and external API integration with Spoonacular. On the frontend, we enhanced our skills in creating intuitive UX and UI through React components, leveraging state and props for dynamic interfaces, and ensuring seamless navigation with react-router-dom.

Additionally, tackling challenges like cart state synchronization and MongoDB setup taught us effective debugging and problem-solving strategies. Collaborating as a team improved our communication and task delegation, with each of us taking ownership of features. If we were to improve, we would focus on optimizing API response times and enhancing the UI with advanced CSS animations for a more polished look. Overall This project solidified our full-stack development skills and prepared us for building scalable, user-focused applications in the future.

