FULL STACK DEVELOPMENT WITH MERN (NM1042)

ON GROCERY WEB APP

A NAAN MUDHALVAN PROJECT REPORT

Submitted By:

J MOHAMMED AZARUDEEN (412721104027)

S. G. ROHITH RAJ (412721104039)

S AKASH VAISHNU DEV (412721104004)

S PAUL BENJAMIN FELIX (412721104033)

C MOSES(412721104029)

TAGORE ENGINEERING COLLEGE, RATHINAMANGALAM – 600 127

ANNA UNIVERSITY::CHENNAI

ABSTRACT

The proposed grocery web app is a modern, user-centric platform designed to streamline the process of purchasing groceries online. By integrating advanced technologies such as responsive design, personalized recommendations, and secure payment gateways, the app aims to enhance user convenience, reduce time spent on shopping, and improve overall customer satisfaction.

Key features include a dynamic product catalog with real-time inventory updates, robust search and filtering options, and integration with delivery and pick-up services. Leveraging AI-driven analytics, the app offers personalized product suggestions and optimized shopping lists based on user preferences and purchase history. Additionally, a multi-lingual interface and accessibility features ensure inclusivity for a diverse user base.

The app also supports seamless integration with local grocery stores, enabling them to digitize their operations and reach a broader audience. By combining efficiency, transparency, and user engagement, the grocery web app seeks to redefine the online grocery shopping experience while supporting sustainable practices through reduced food waste and eco-friendly packaging options.

1. Introduction

- **Project Title: G r o c e r y W e b A p p**
- **Team ID: NM2024TMID02109**
- **Team leader and members:**

S.No	**Name**	**Register Number**	**Program ID**
1.	ROHITH RAJ S.G	412721104039	66698DC34B5DC9482B548DC9F722F79C
2.	PAUL BENJAMIN FELIX S	412721104033	863C60058CB58608CC44AE4E11AC3762
3.	MOHAMMED AZARUDEEN J	412721104027	DAA43C08FA7EA5C941449BBF438D56FF
4.	AKASH VAISHNU DEV S	412721104004	7E97216E6E9296CA0E241FC5A3C552AB
5.	MOSES C	412721104029	AB858A6B52FD38AAD034E70DDF38512D

2. Project Overview

Purpose:

The Grocery Web App is designed to provide a seamless, user-friendly platform for online grocery shopping. It bridges the gap between traditional grocery shopping and modern e-commerce by offering a responsive interface where users can browse products, manage their shopping lists, and schedule deliveries or pick-ups. Store owners and administrators can efficiently manage their inventory, orders, and platform settings.

Key Features:

1.User Roles and Access Control:

- **Customers:** Browse products, add to cart, make purchases, and schedule deliveries.
- **Store Managers:** Manage product catalogs, update inventory, and monitor orders.
- **Admin:** Oversee platform management, including moderation of stores, users, and transactions.

2.Comprehensive Product Management:

- Add or update products with images, pricing, and stock levels.
- Highlight offers and deals through banners and featured listings.

3.Admin Control:

- Access to all user data, product catalogs, and order details.
- Tools for content moderation, user management, and sales analytics.

4.Advanced Technical Integration:

- **Cloudinary:** For secure media storage of product images.
- **Stripe/PayPal:** For secure and efficient payment processing.

The Grocery Web App offers a scalable, flexible solution for modern grocery shopping, ensuring ease of use for customers and operational efficiency for businesses.

Architecture:

Front-end:

The frontend is built using React to deliver a fast, interactive user experience. Key elements include:

- **Framework:** React, utilizing Vite for quick builds.
- **Styling:** Bootstrap and custom CSS for a polished, responsive layout.
- **Routing:** Protected routes with JWT tokens to ensure secure access.
- **State Management:** React Context API or Redux for global state control.

Back-end:

The backend employs Node.js and Express for robust, scalable server-side operations. Key components include:

- **Framework:** Express.js for RESTful APIs.
- **Authentication:** Secure session handling with JWT tokens.
- **Media Handling:** Integration with Cloudinary for storing product images.

Database:

MongoDB Atlas serves as the database, ensuring scalable and flexible data storage. Key schemas include:

- **User Schema:** For customers, store managers, and admins with role-based access.
- **Product Schema:** To store details about products, categories, and inventory levels.
- **Order Schema:** For managing customer orders and delivery information.

Setup Instructions:

Prerequisites:

- Install Node.js (v14+).
- Install MongoDB (local or MongoDB Atlas).
- Have npm or yarn installed for dependency management.
- Configure environment variables:
  - `MONGO_URI`: MongoDB connection string.
  - `JWT_SECRET`: Secret key for token generation.

Installation:

1. Clone the repository: `git clone https://github.com/YourRepo/grocery-web-app.git`
2. Navigate to project folders:
   - Frontend: `cd frontend`
   - Backend: `cd backend`
3. Install dependencies:
   - Frontend: `npm install`
   - Backend: `npm install`
4. Configure environment variables for both the frontend and backend.

1. Run the application

- Start the backend server:
  cd backend
  node index.js
- Start the frontend server:
  cd frontend
  npmcreatevite@lastest
  (Select React and JavaScript, then proceed)
  npm run dev

2. Folder Structure

Frontend:

```
frontend/
├── src/
│   ├── components/    # Reusable UI components
│   ├── pages/         # Page-specific components
│   ├── services/      # API calls and service functions
│   ├── App.js         # Main application component
│   ├── index.js       # Entry point of the React app
└── public/
    ├── index.html     # Main HTML file
    └── assets/        # Static assets like images
```

Backend:

```
backend/
├── routes/           # Route definitions for APIs
├── controllers/      # Business logic for routes
├── schemas/          # Mongoose models and schemas
├── middleware/       # Authentication and validation middleware
├── config/           # Configuration files (e.g., database connection)
├── utils/            # Utility functions
├── server.js         # Entry point of the backend server
└── .env              # Environment variables
```

This structured approach ensures maintainability and scalability for both the frontend and backend components.

6. Running the Application

- **Frontend:**
  To start in the /frontend directory.

  ```
  >> cd frontend
  >> npm create vite@latest
  >>select react framework, JavaScript and enter to continue
  >>npm run dev
  ```

- **Backend:**

  To start the /backend directory.

  ```
  >> cd backend

  >> node index.js
  ```

7. API Documentation

- **Endpoints:**
  - /auth: Handles login, registration, and authentication.
  - /courses: Manages course creation, editing, deletion, and enrollment.
  - /payments: Handles premium course payments.
  - /Dashboards: Handles all the dashboards (admin / user / teacher)
- **Example:**
  - **POST api/auth/login**
    Request Body: {email, password}
    Response: {token, user}

8. Authentication

Authentication in LearnHub is handled securely using JWT (JSON Web Tokens). Below is a detailed explanation:

1. Registration:

- A new user registers by providing their name, email, password, and role.
- Passwords are hashed using bcrypt before being stored in the database.

2. Login:

- A user logs in with their email and password.
- Upon successful authentication, a JWT token is generated and returned.

3. Token Verification:

- The JWT token is included in the Authorization header (Bearer <token>) for all protected routes.
- Middleware on the backend verifies the token's validity and extracts the user's role and ID.

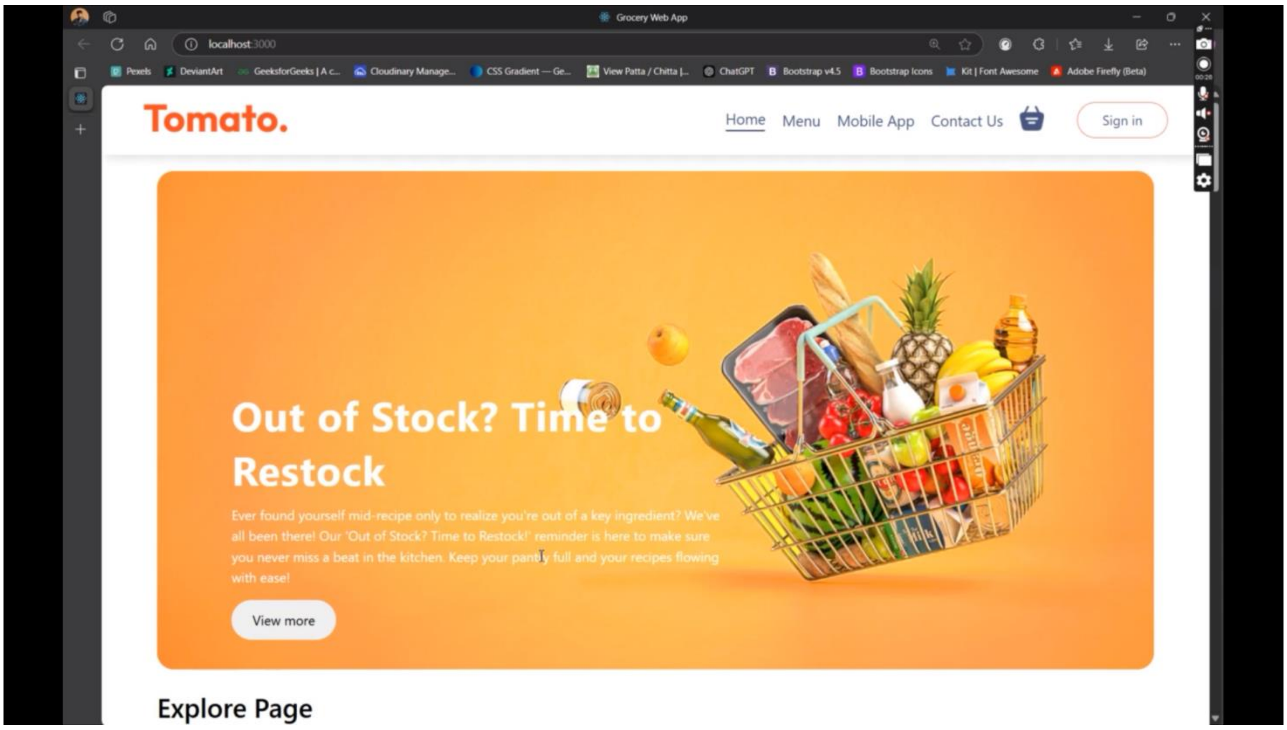
4. Role-Based Access Control:

- Depending on the user's role (student, teacher, admin), they are granted specific permissions:
    - Students: Can access their enrolled courses and content.
    - Teachers: Can create, edit, and delete courses.
    - Admins: Can manage all users and content.

This approach ensures secure session management and fine-grained access control across the platform.

9. User Interfaces:

Landing Page

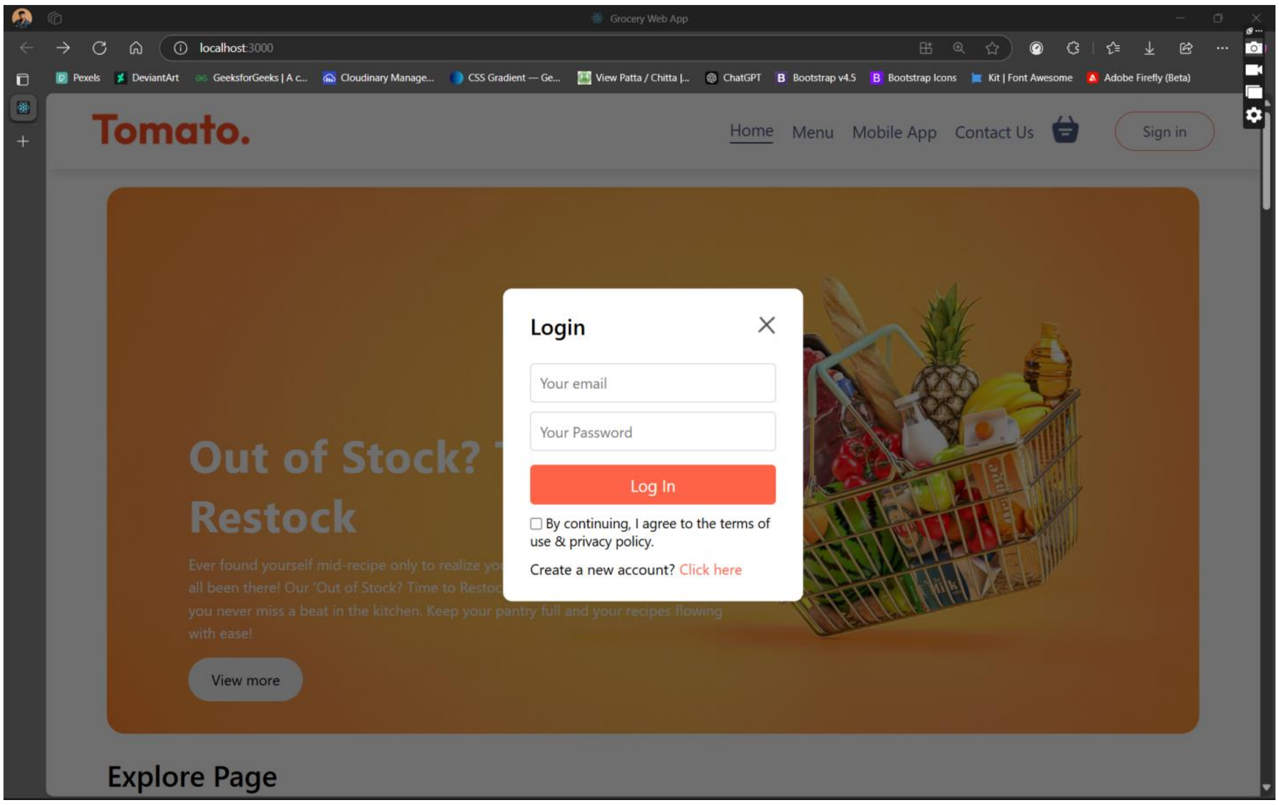


Out of Stock? Time to Restock

Ever found yourself mid-recipe only to realize you're out of a key ingredient? We've all been there! Our 'Out of Stock? Time to Restock!' reminder is here to make sure you never miss a beat in the kitchen. Keep your pantry full and your recipes flowing with ease!

Explore Page

Login page

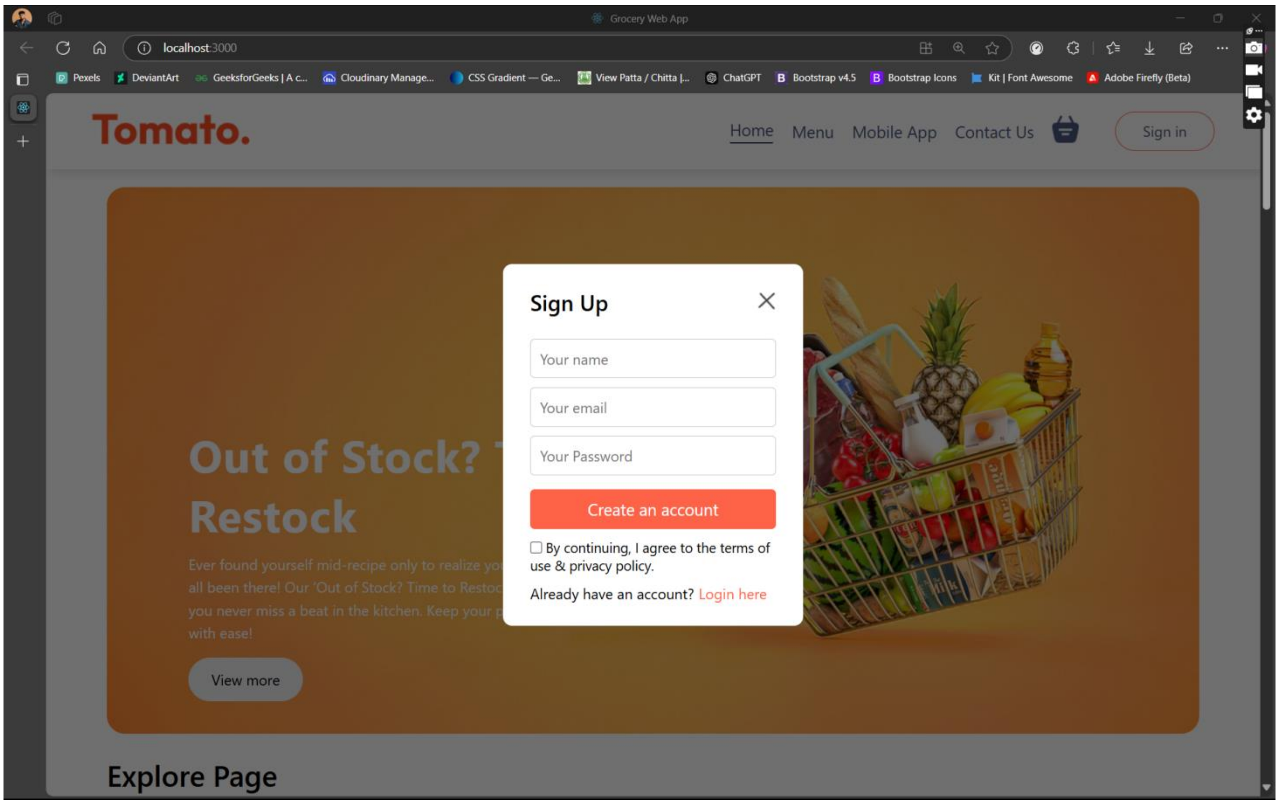


By continuing, I agree to the terms of use & privacy policy.

Create a new account? Click here

Explore Page

Registration page



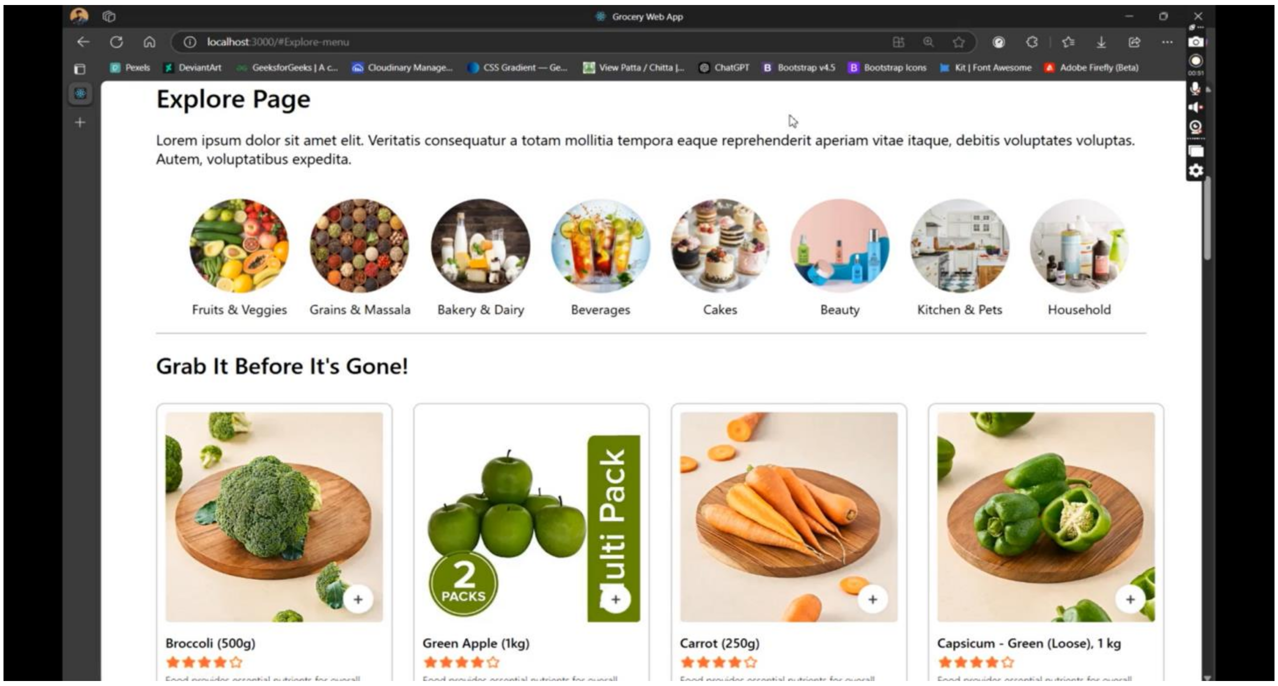
Out of Stock? Restock

Ever found yourself mid-recipe only to realize yo… all been there! Our 'Out of Stock? Time to Resto… you never miss a beat in the kitchen. Keep your p… with ease!

By continuing, I agree to the terms of use & privacy policy.

Already have an account? Login here

Categories

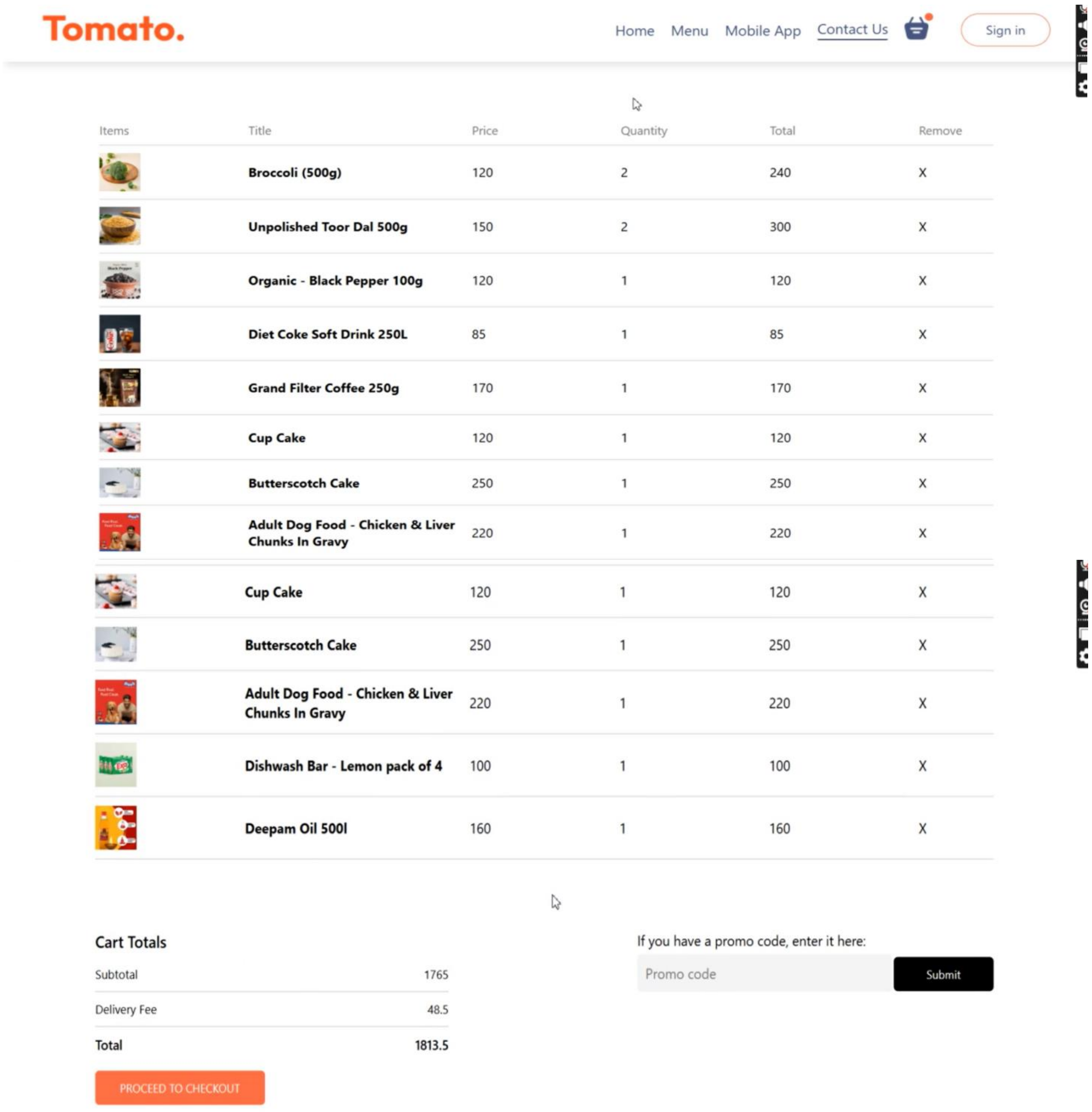


Explore Page

Lorem ipsum dolor sit amet elit. Veritatis consequatur a totam mollitia tempora eaque reprehenderit aperiam vitae itaque, debitis voluptates voluptas. Autem, voluptatibus expedita.

Grab It Before It's Gone!

Cart and Checkout



Items	Title	Price	Quantity	Total	Remove
	Broccoli (500g)	120	2	240	X
	Unpolished Toor Dal 500g	150	2	300	X
	Organic - Black Pepper 100g	120	1	120	X
	Diet Coke Soft Drink 250L	85	1	85	X
	Grand Filter Coffee 250g	170	1	170	X
	Cup Cake	120	1	120	X
	Butterscotch Cake	250	1	250	X
	Adult Dog Food - Chicken & Liver Chunks In Gravy	220	1	220	X
	Cup Cake	120	1	120	X
	Butterscotch Cake	250	1	250	X
	Adult Dog Food - Chicken & Liver Chunks In Gravy	220	1	220	X
	Dishwash Bar - Lemon pack of 4	100	1	100	X
	Deepam Oil 500l	160	1	160	X

Cart Totals

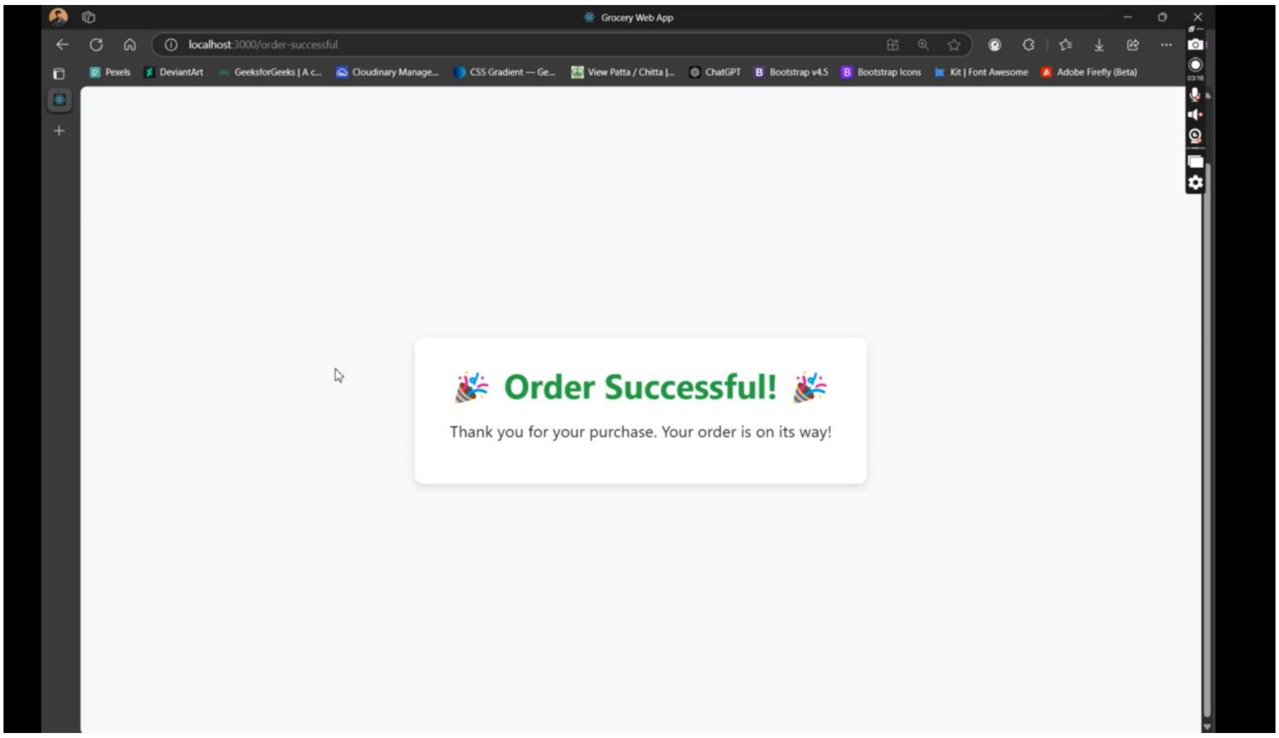
Subtotal	1765
Delivery Fee	48.5
Total	1813.5

PROCEED TO CHECKOUT

If you have a promo code, enter it here:

Promo code | Submit

Order Successful Page

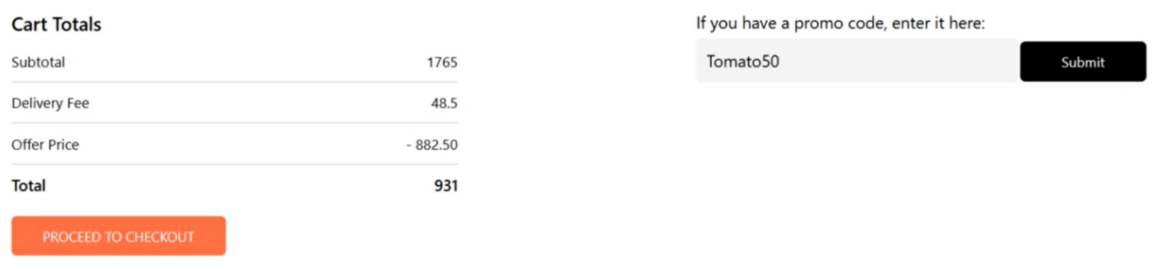


🎉 Order Successful! 🎉

Thank you for your purchase. Your order is on its way!

Promo code

Where there is a quick discount over the ordered item



Cart Totals

Subtotal	1765
Delivery Fee	48.5
Offer Price	- 882.50
Total	**931**

If you have a promo code, enter it here:

Tomato50

10. Testing of the Application

Testing ensures the functionality, performance, and security of Grocery Web App. Here’s an overviewof the testing process:

Unit Testing

- **Focus:** Tests individual components and functions.
- **Tools:**
    - Jest for backend logic, such as API controllers and utility functions.
    - React Testing Library for testing React components.
- **Examples:**
    - Verifying correct rendering of the login form.
    - Testing backend functions like user registration or JWT generation.

Bug Tracking and Reporting

- Bugs and issues are documented using tools like JIRA or GitHub Issues, categorized, and prioritized for resolution.

Testing is an iterative process that ensures Tomato delivers a seamless, secure, and efficient user experience.

12. Known Issues

Current Known Bugs and Limitations in LearnHub:

1. **Page Refresh Requirement:**
   - Some pages may require a manual refresh to reflect the latest data or changes (e.g., after enrolling in a course or completing a payment). This is due to caching and state management updates that need refinement.
2. **Content Loading Delays:**
   - Occasionally, media files like videos or PDFs might take longer to load depending on the network and server load.
3. **Limited Mobile Optimization:**

o While the platform is functional on mobile devices, certain UI components may not be fully optimized for smaller screens

4. **Error Message Clarity:**
   - Some error messages (e.g., failed login or invalid input) may not provide detailed or user-friendly feedback.

13. Future Enhancements

1. **Real-Time Features:**

   Enable live inventory updates to reflect stock levels in real time.

2. **Multi-Language Support:**

   Add localization for global users.

3. **AI-Driven Recommendations:**

   Suggest products based on purchase history and preferences.

4. **Enhanced Mobile Experience:**

   Fully optimize the app for mobile devices.

5. **Sustainability Features:**

   Introduce eco-friendly tagging and waste-reduction features, such as portioned packaging recommendations.