Full Stack Development with MERN

Grocery Webapp Project Documentation

1. Introduction

- ***Project Title: Grocery Webapp***
- ***Team Members: [Team ID: NM2024TMID00538]***
  - ***PALLI MUHAMMED SUHAIB – Project Manager (Team Lead)***
  - ***ROSHAN S – Full Stack Developer***
  - ***SHERIFF AFRID H – Backend Developer***
  - ***YOKHESH D – Frontend Developer***
- ***GitHub:*** https://github.com/suhaibpalli/Grocery_Webapp.git

2. Project Overview

- **Purpose:** Create a seamless online shopping platform for customers to explore and purchase products with robust backend management for sellers and administrators.
- **Features:**

  - User registration and authentication

  - Product catalog with search and filter capabilities

  - Shopping cart functionality

  - Order placement and tracking

  - Admin product and user management

  - Feedback system

3. Architecture

Frontend:

Framework: Angular

Key Components:

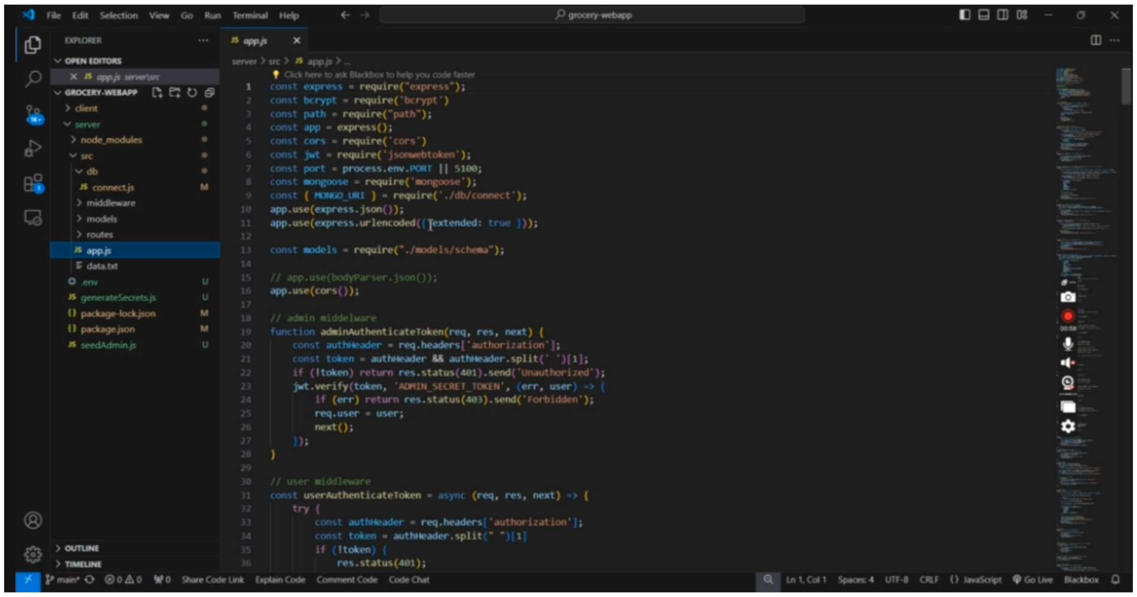
- User-facing components: Register, Login, Home, Products, My Cart, My Orders
- Admin components: Dashboard, Product Management, User Management
- Routing: Implemented via `app-routing.module.ts` with lazy-loaded admin module

Backend:

Framework: Node.js with Express.js

Key Features:

- RESTful API design
- Middleware for authentication
- Database interaction using Mongoose
- JWT-based authentication



```javascript
const express = require("express");
const bcrypt = require('bcrypt')
const path = require("path");
const app = express();
const cors = require('cors')
const jwt = require('jsonwebtoken');
const port = process.env.PORT || 5100;
const mongoose = require('mongoose');
const { MONGO_URI } = require('./db/connect');
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

const models = require("./models/schema");

// app.use(bodyParser.json());
app.use(cors());

// admin middelware
function adminAuthenticateToken(req, res, next) {
    const authHeader = req.headers['authorization'];
    const token = authHeader && authHeader.split(' ')[1];
    if (!token) return res.status(401).send('Unauthorized');
    jwt.verify(token, 'ADMIN_SECRET_TOKEN', (err, user) => {
        if (err) return res.status(403).send('Forbidden');
        req.user = user;
        next();
    });
}

// user middleware
const userAuthenticateToken = async (req, res, next) => {
    try {
        const authHeader = req.headers['authorization'];
        const token = authHeader.split(" ")[1]
        if (!token) {
            res.status(401);
```

Database:

Database: MongoDB

Key Collections:

- Users
- Categories
- Products
- Cart
- Orders
- Payments
- Feedback.

4. Setup Instructions

Prerequisites:

- Node.js (v14+ recommended)
- npm (Node Package Manager)
- MongoDB

Installation:

1. Clone the repository

```
git clone
https://github.com/suhaibpalli/Grocery_Webapp.git
```

2. Install backend dependencies:

```
cd server

npm install
```

3. Install frontend dependencies:

```
cd client

npm install
```

4. Set up MongoDB:

- Ensure MongoDB is running locally

- Default connection string: `mongodb://localhost:27017/groceryDB`

5. Folder Structure

Client:

```
src/
├── app/
│   ├── components/
│   │   ├── login/
│   │   ├── register/
│   │   ├── home/
│   │   └── ...
│   ├── modules/
│   │   └── admin/
│   └── app-routing.module.ts
```

Server:

```
src/
├── app.js
├── db/
│   └── connect.js
├── models/
│   └── schema.js
└── routes/
    ├── admin.js
    ├── category.js
    └── ...
```

6. Running the Application

Provide commands to start the frontend and backend servers locally.

Frontend:

```
cd client
ng serve
```

# Access at http://localhost:4200

Backend:

```
cd server
npm start
```

# Runs on http://localhost:5100

7. API Documentation

Authentication:

POST `/login`

- Request body: `{email, password }`
- Returns user details and JWT token

POST `/register`

- Request body: `{firstname, lastname, username, email, password }`
- Creates new user account

Products:

GET `/products`: Retrieve all products

GET `/products/:id`: Get specific product details

POST `/add-products`: Add new product (admin)

PUT `/products/:id`: Update product details

DELETE `/products/:id`: Remove product

Orders:

POST `/orders`: Place new order

GET `/orders`: List all orders

GET `/my-orders/:id`: Retrieve user's orders

8. User Roles

Admin Role:

- Full system control
- Can manage products, users, and view all orders
- Access to dashboard and analytics

User Role:

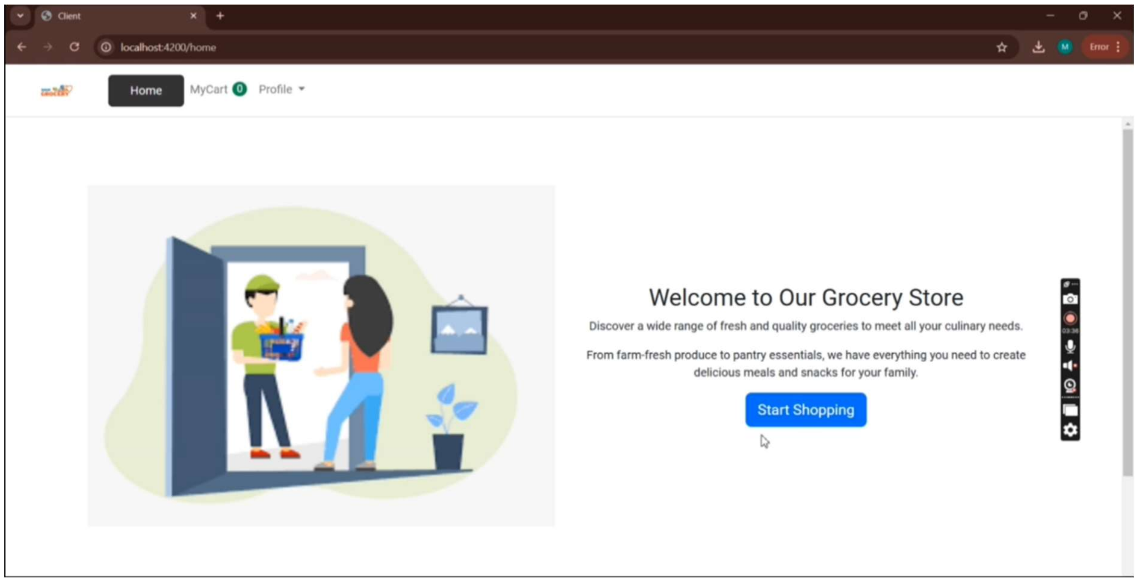
- Browse products
- Add items to cart
- Place orders
- View order history
- Provide feedback

9. Security Features

- Password hashing with bcrypt
- JWT-based authentication
- Role-based access control
- Middleware for route protection

10. Screenshots or Demo

Home page:

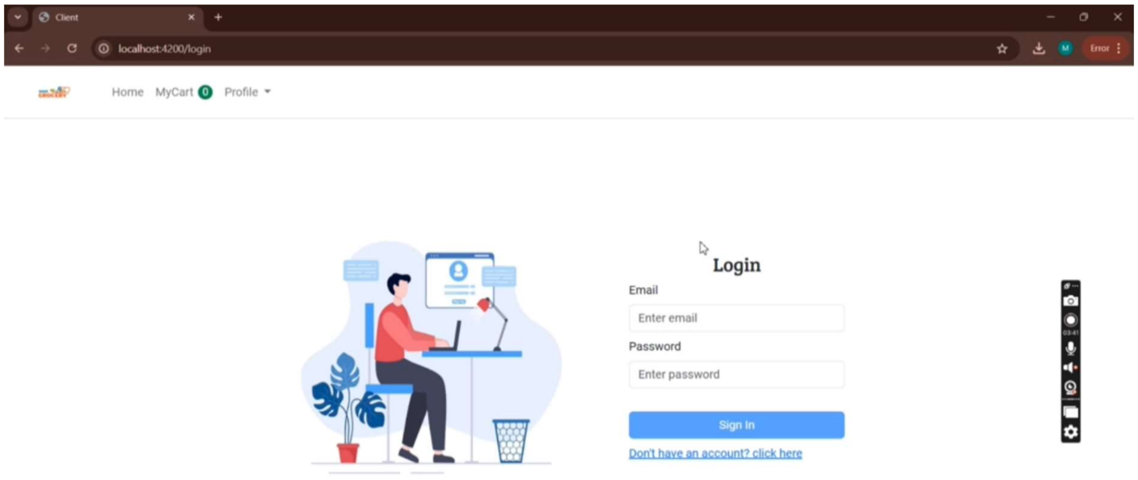


Welcome to Our Grocery Store

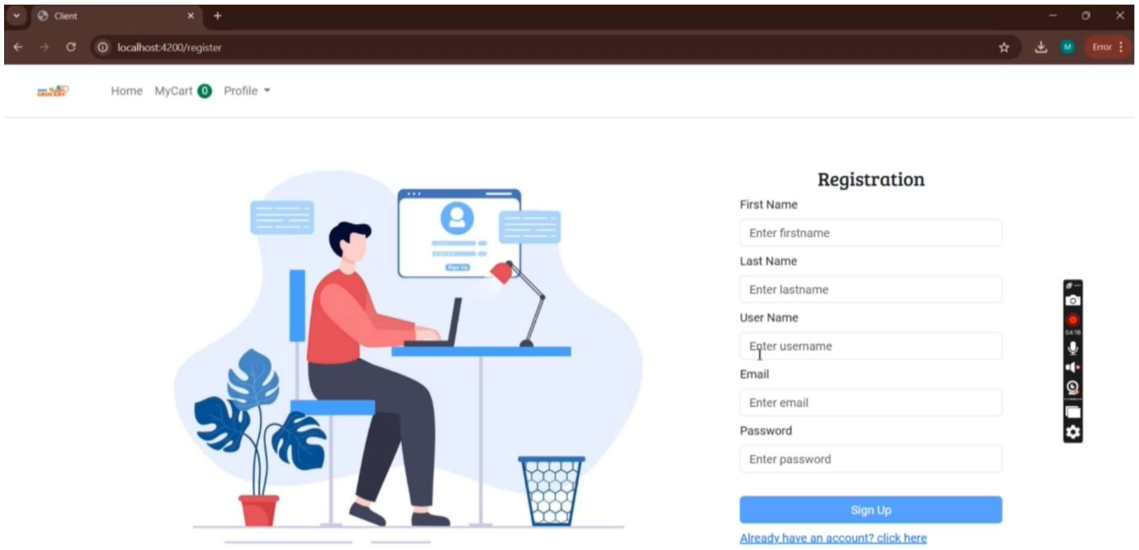
Discover a wide range of fresh and quality groceries to meet all your culinary needs.

From farm-fresh produce to pantry essentials, we have everything you need to create delicious meals and snacks for your family.

Login page:



Registration page:



Registration

First Name

Enter firstname

Last Name

Enter lastname

User Name

Enter username

Email

Enter email

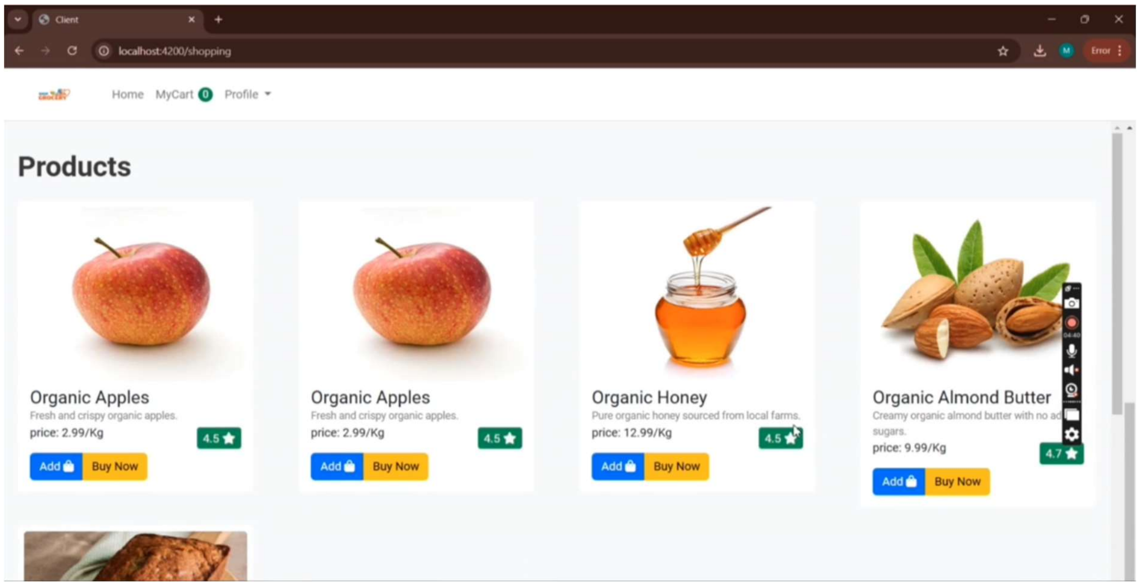
Password

Enter password

Sign Up

Already have an account? click here

Available products page:



Products

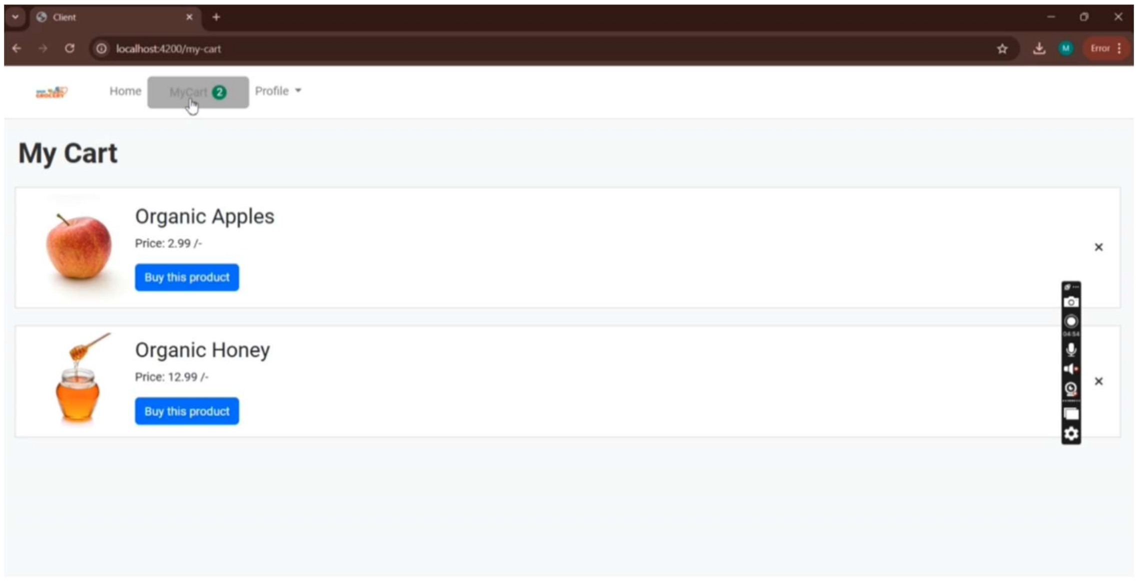
Organic Apples
Fresh and crispy organic apples.
price: 2.99/Kg
4.5
Add
Buy Now

Organic Apples
Fresh and crispy organic apples.
price: 2.99/Kg
4.5
Add
Buy Now

Organic Honey
Pure organic honey sourced from local farms.
price: 12.99/Kg
4.5
Add
Buy Now

Organic Almond Butter
Creamy organic almond butter with no ad
sugars.
price: 9.99/Kg
4.7
Add
Buy Now

My cart page:



My Cart

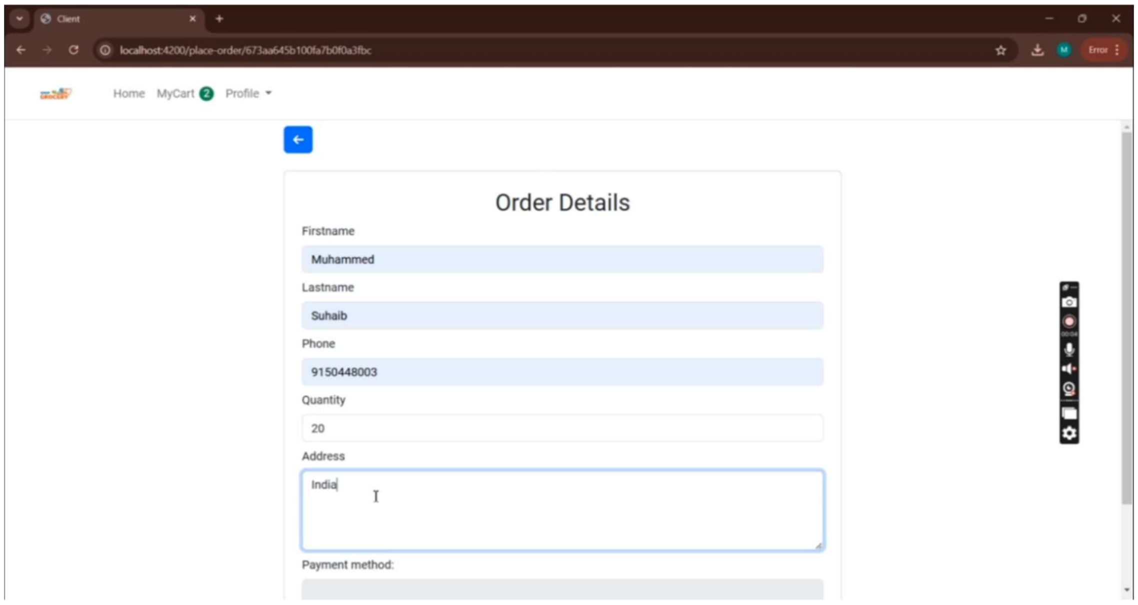
Organic Apples

Price: 2.99 /-

Organic Honey

Price: 12.99 /-

Order details page:



Order Details

Firstname

Muhammed

Lastname

Suhaib

Phone

9150448003

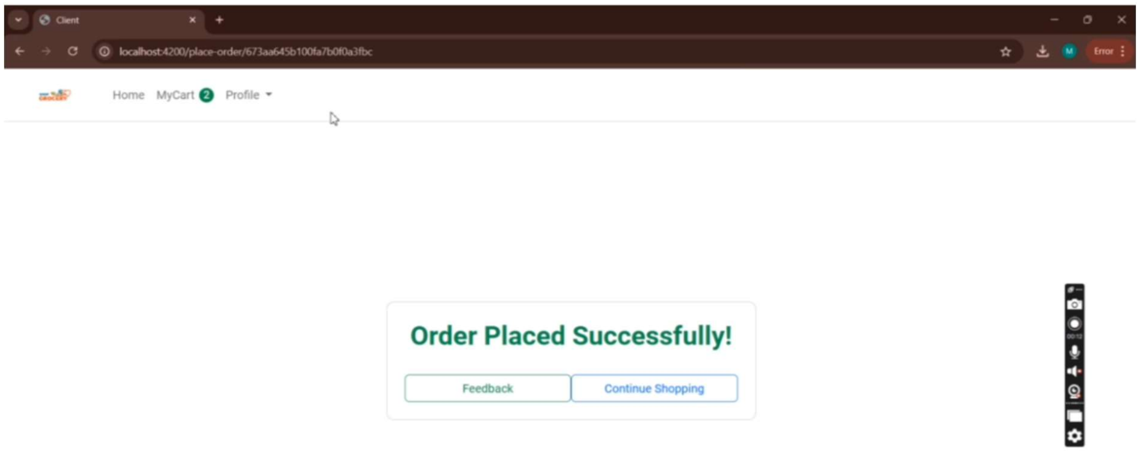
Quantity

20

Address

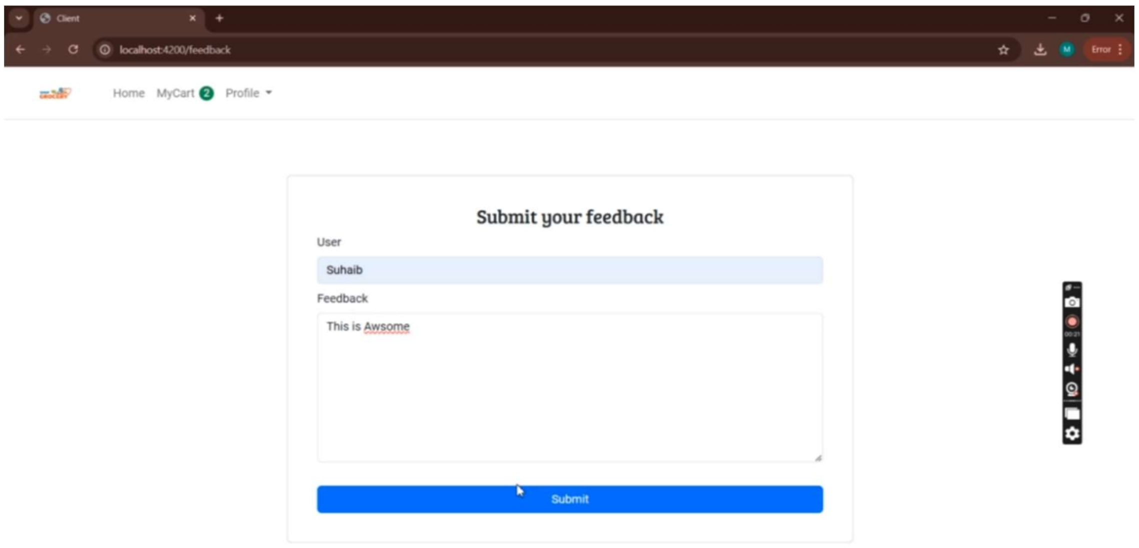
India

Payment method:



Order Placed Successfully!

Feedback page:



Submit your feedback

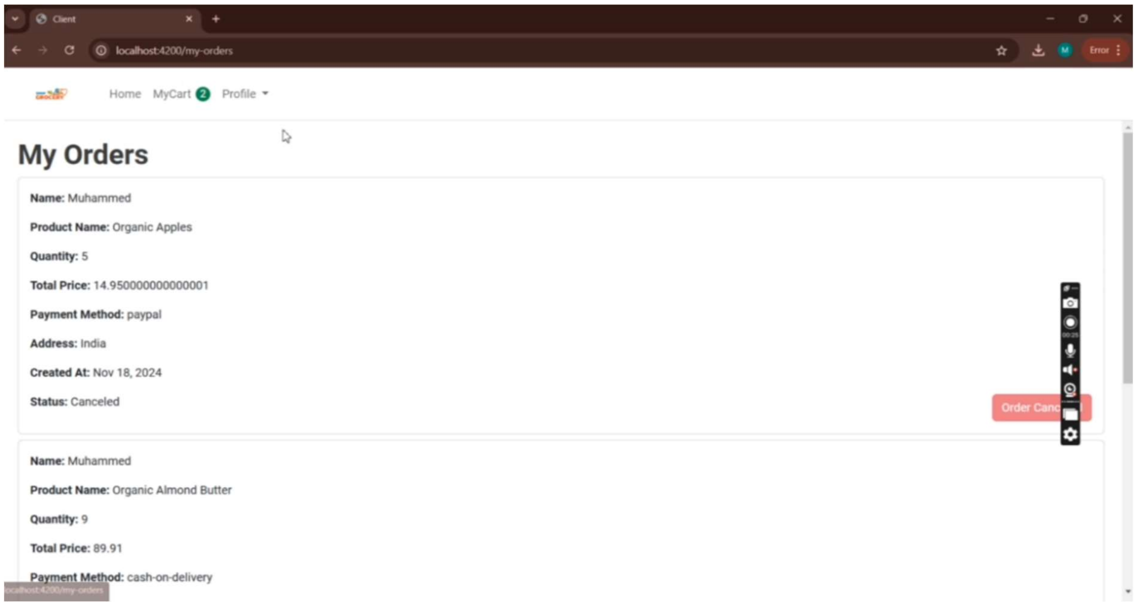
User

Suhaib

Feedback

This is Awsome

My orders page:



My Orders

Name: Muhammed

Product Name: Organic Apples

Quantity: 5

Total Price: 14.950000000000001

Payment Method: paypal

Address: India

Created At: Nov 18, 2024

Status: Canceled

Name: Muhammed

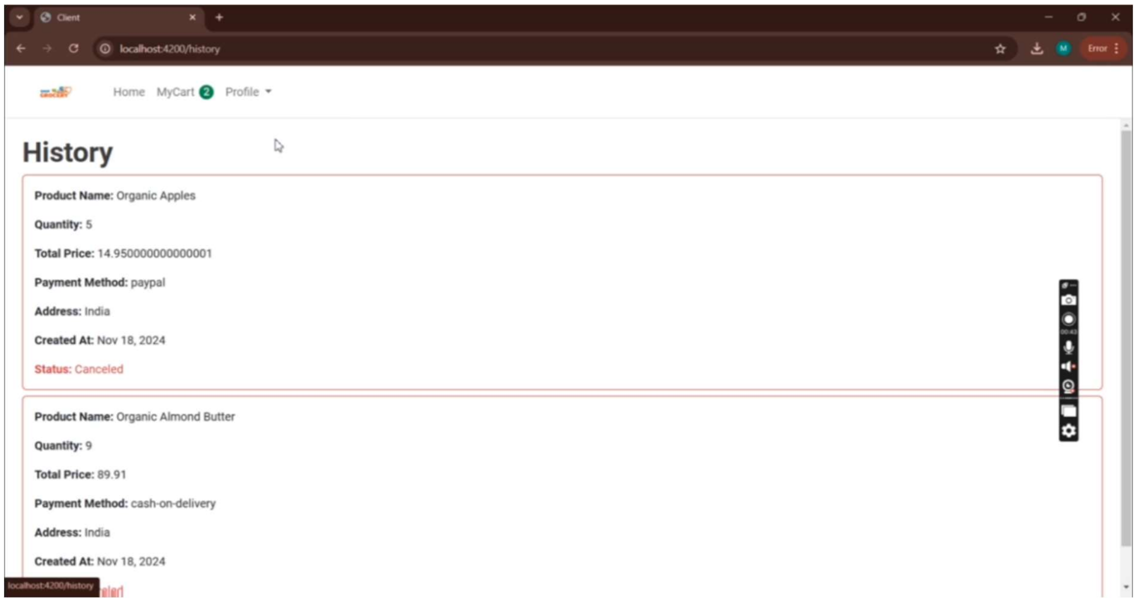
Product Name: Organic Almond Butter

Quantity: 9

Total Price: 89.91

Payment Method: cash-on-delivery

History page:



History

Product Name: Organic Apples

Quantity: 5

Total Price: 14.950000000000001

Payment Method: paypal

Address: India

Created At: Nov 18, 2024

Status: Canceled

Product Name: Organic Almond Butter

Quantity: 9

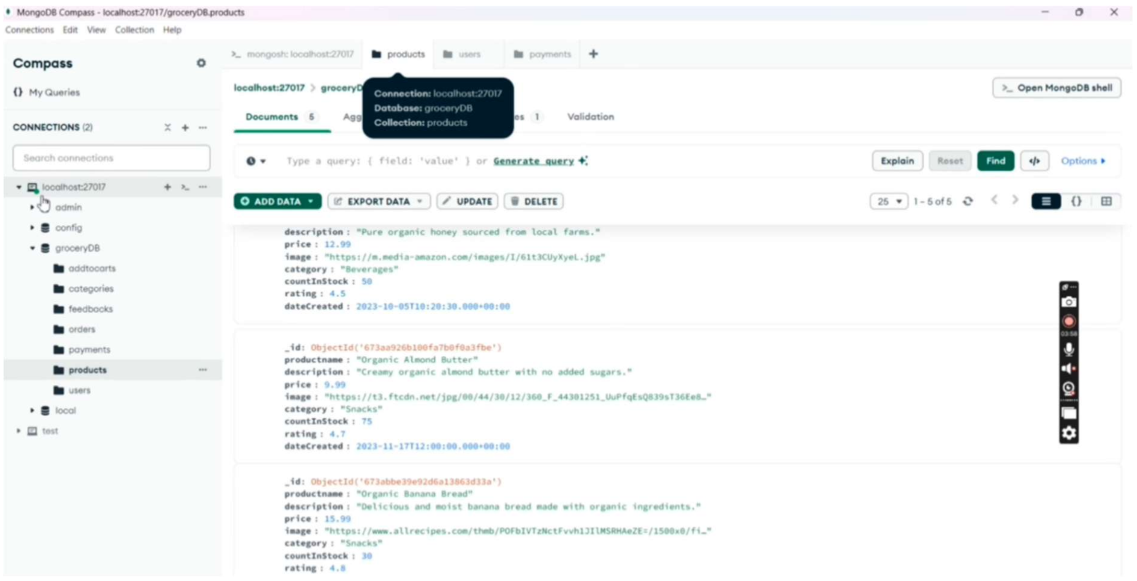
Total Price: 89.91

Payment Method: cash-on-delivery

Address: India

Created At: Nov 18, 2024

MongoDB connection:



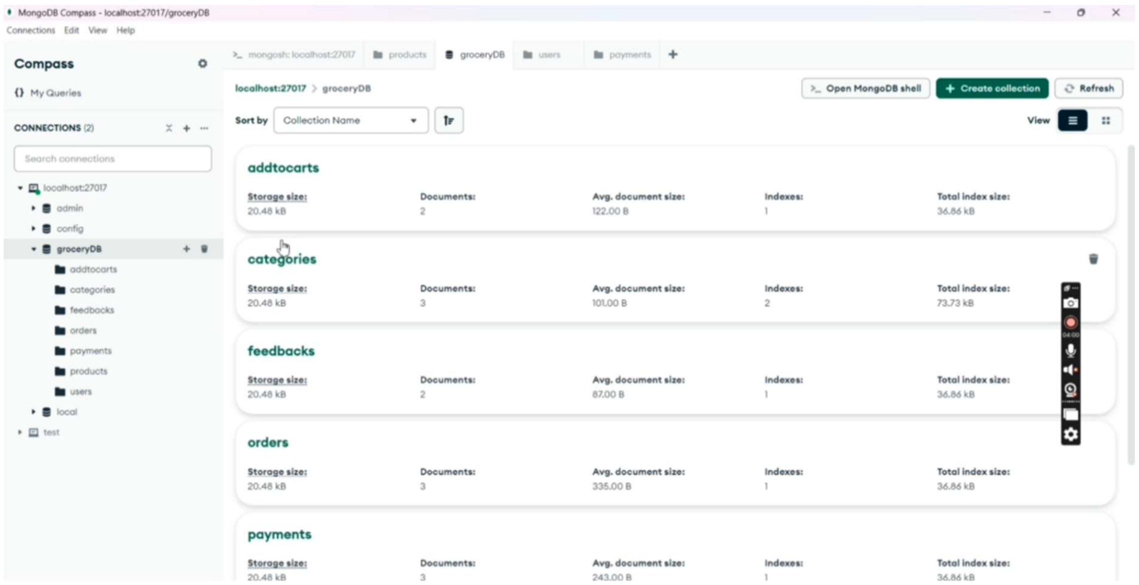
Connection: localhost:27017
Database: groceryDB
Collection: products

```
Type a query: { field: 'value' } or Generate query

description : "Pure organic honey sourced from local farms."
price : 12.99
image : "https://m.media-amazon.com/images/I/61t3CUyXyeL.jpg"
category : "Beverages"
countInStock : 50
rating : 4.5
dateCreated : 2023-10-05T10:20:30.000+00:00

_id: ObjectId('673aa926b100fa7b0f0a3fbe')
productname : "Organic Almond Butter"
description : "Creamy organic almond butter with no added sugars."
price : 9.99
image : "https://t3.ftcdn.net/jpg/00/44/30/12/360_F_44301251_UuPfqEsQ839sT36Ee8…"
category : "Snacks"
countInStock : 75
rating : 4.7
dateCreated : 2023-11-17T12:00:00.000+00:00

_id: ObjectId('673abbe39e92d6a13863d33a')
productname : "Organic Banana Bread"
description : "Delicious and moist banana bread made with organic ingredients."
price : 15.99
image : "https://www.allrecipes.com/thmb/POFbIVTzNctFvvh1JIlMSRHAeZE=/1500x0/fi…"
category : "Snacks"
countInStock : 30
rating : 4.8
```



localhost:27017 > groceryDB

Sort by: Collection Name

Collection	Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
addtocarts	20.48 kB	2	122.00 B	1	36.86 kB
categories	20.48 kB	3	101.00 B	2	73.73 kB
feedbacks	20.48 kB	2	87.00 B	1	36.86 kB
orders	20.48 kB	3	335.00 B	1	36.86 kB
payments	20.48 kB	3	243.00 B	1	36.86 kB

11. Known Issues

- Limited error handling in some API endpoints
- No comprehensive input validation
- Basic authentication mechanism

12. Technologies Used

- Frontend: Angular
- Backend: Node.js, Express.js
- Database: MongoDB
- Authentication: JWT, bcrypt
- Other Libraries: Mongoose, cors, jsonwebtoken

13. Future Enhancements

- Implement advanced search and filtering
- Add product reviews and ratings
- Integrate more payment gateways
- Implement real-time order tracking
- Enhanced admin analytics dashboard

14. Conclusion

The Grocery Webapp successfully demonstrates a modern e-commerce platform using the MERN stack. It provides robust features for users and administrators, including secure authentication, product management, and order tracking. The project offers a solid foundation for an online grocery shopping experience, with clear potential for future enhancements and scalability.