



NAAN MUDHALVAN
MERN STACK POWERED BY MONGODB

PROJECT NAME: GROCERY WEBAPP

TEAM MEMBERS:

SATHISH KUMAR M- (D1B2F4FEA9C24A590C04B776B4024437)

AKILESH RAJ M- (C696C8349659376A4DC4C891CAB92BDA)

ABISHEK M - (0806797E0B578EC9A284C5C7A3D2275D)

RAVIN A - (1CEBF9425B6E4169E6F48E9FEB60213C)

Aim:

The aim of the **Grocery Webapp** project is to design and develop an interactive, efficient, and user-friendly web application using the MERN stack (MongoDB, Express.js, React.js, and Node.js). This platform will provide users with an easy-to-navigate interface to browse, search, and purchase grocery items. At the same time, administrators will be able to manage inventory, process orders, and handle customer data seamlessly. The application will ensure a smooth shopping experience for customers while providing real-time updates, recommendations, and secure payment options.

Roles And Responsibility:

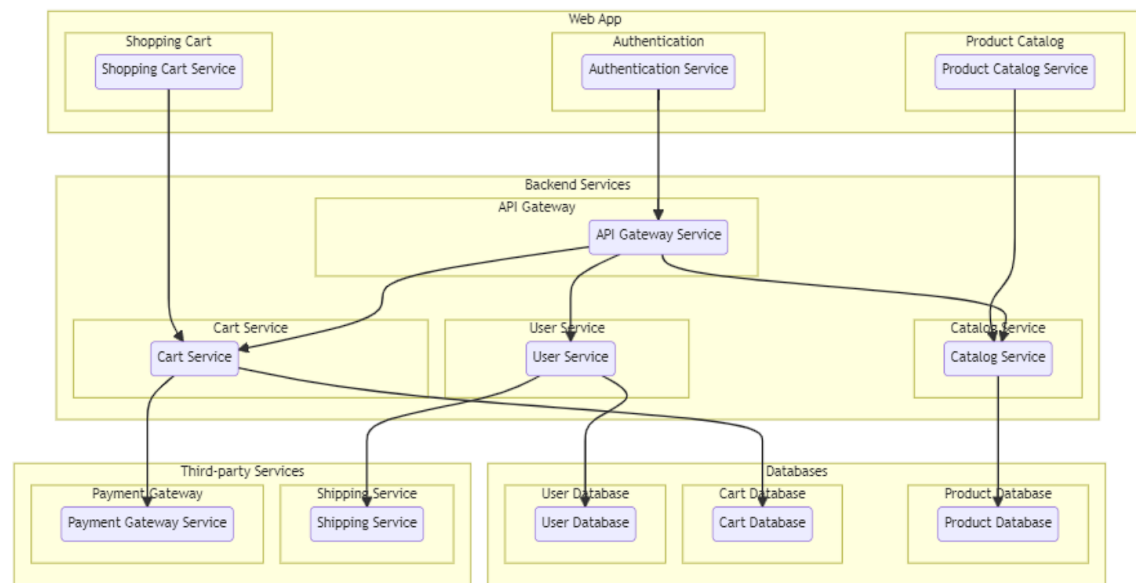
1-Admin Role:

- **Responsibilities:** The admin role has full control and administrative privileges over the system.
- **Permissions:**
 - **Manage:** Admins can add, edit, and delete shop information along with products.
 - **Manage product bookings:** Admins can view and manage all product bookings made by users and agents, including canceling or modifying product bookings.
 - **Manage users:** Admins can create, edit, and delete user accounts, as well as manage their roles and permissions.
 - **Generate reports:** Admins have access to generate reports and analytics on product booking details, booking counts, and sales reports.
 - **Logout:** Finally, they can logout from the Grocery Webapp.

User Role:

- **Responsibilities:** Users are the customers of the online shopping web application who can search for products, and make product bookings.
- **Permissions:**
 - **View products:** Users can search for products, based on interest.
 - **Product bookings:** Users can select products that are available and complete the order process.
 - **Manage product booking:** Users can view their own product order bookings, modify booking details, track booking details, and cancel their bookings
 - **Manage cart:** Users can view their cart details and modify them if needed.
 - **Logout:** Finally, they can logout from the Grocery Webapp.

Activity diagram:



Application Flow:

1. Required tools and software:

- Node.js.
- MongoDB.
- Create-react-app.
- Express.js
- Angular

2. Create project folders and files:

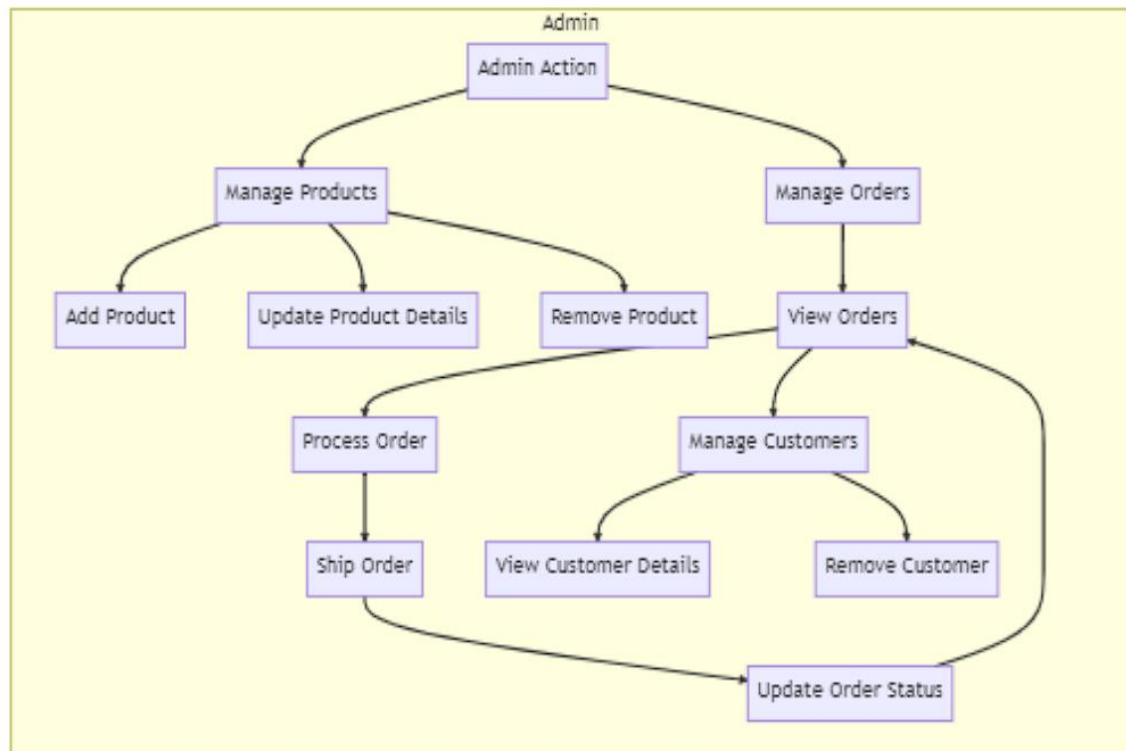
- Client folders.
- Server folders.

3. Install Packages:

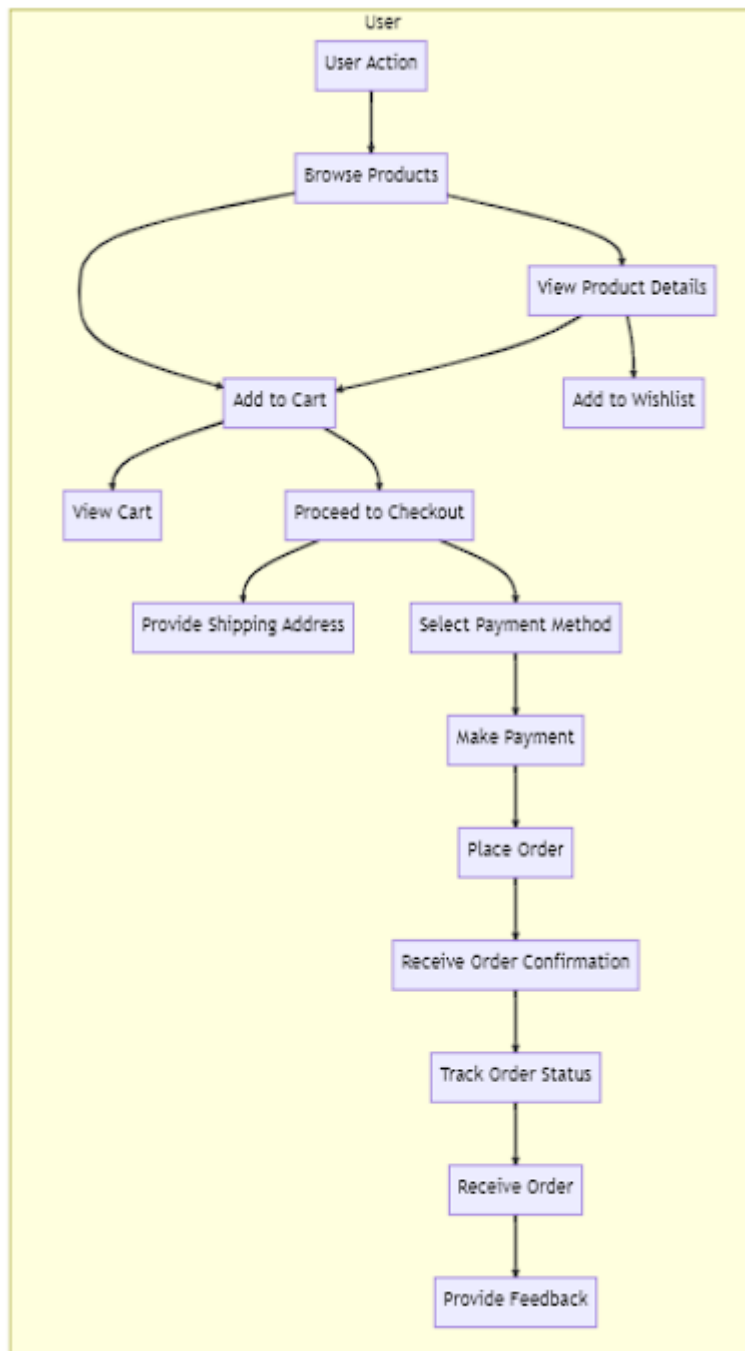
- Frontend npm Packages
 - Axios.
 - React-Router –dom.
 - Bootstrap.
 - React-Bootstrap.
- Backend npm Packages
 - Express.
 - Mongoose.

Role Based Access

Admin:



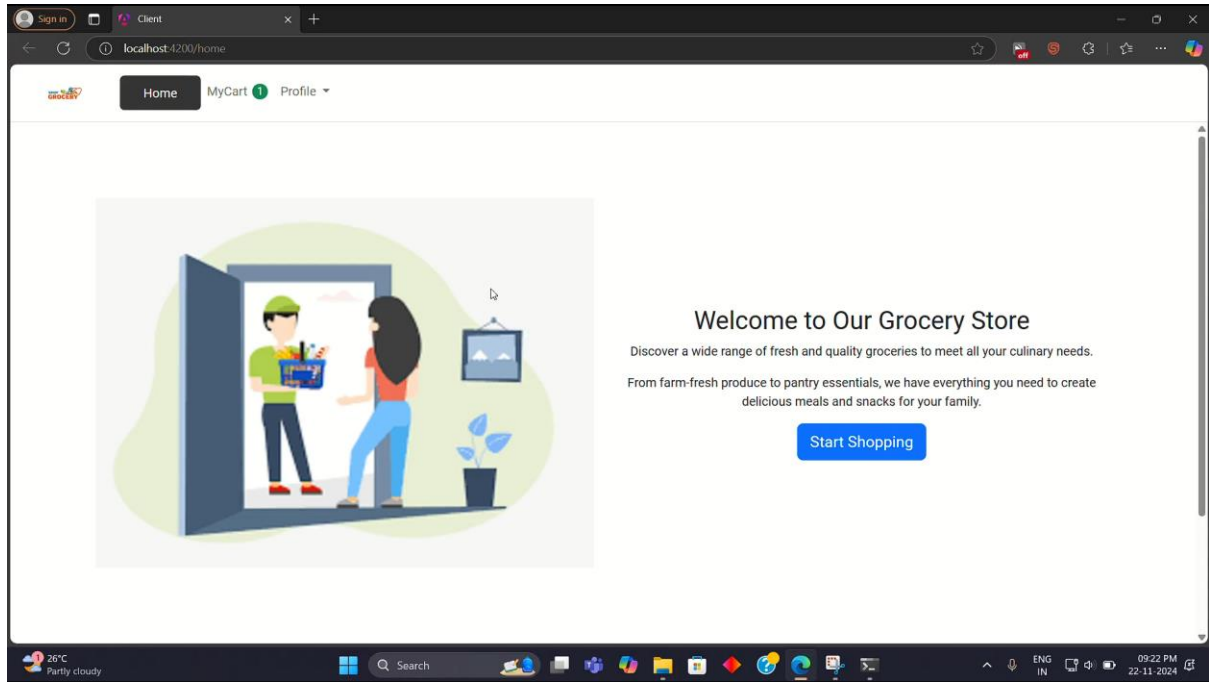
User:



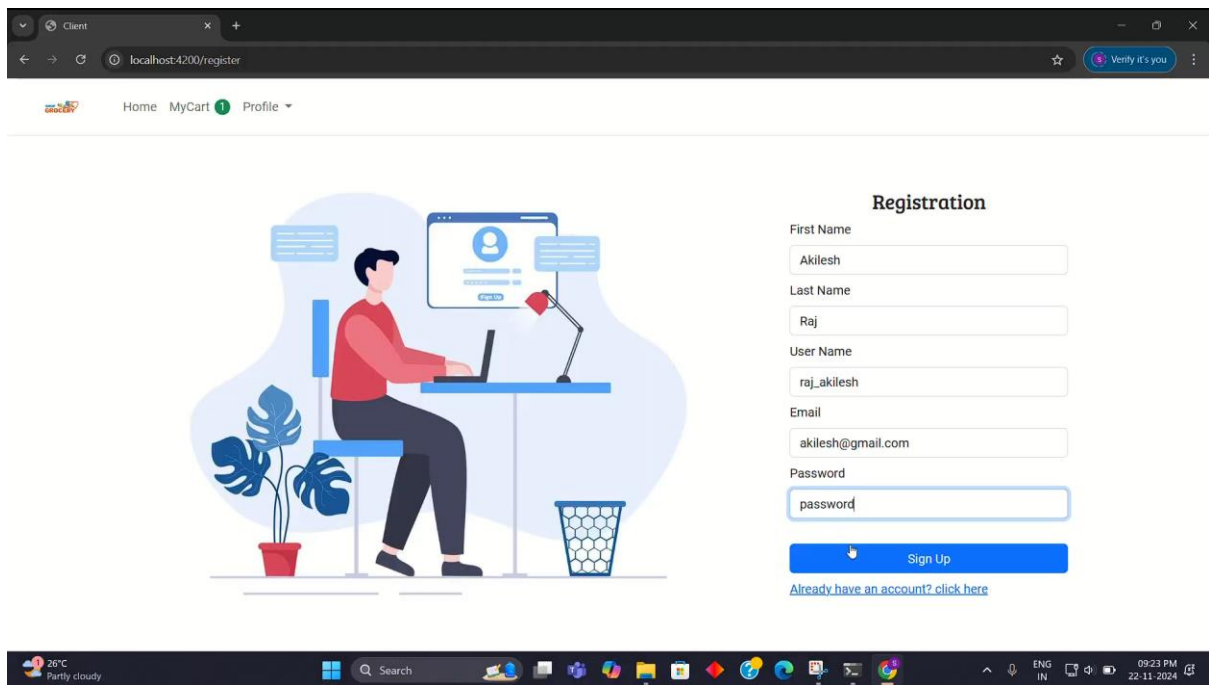
Implementation:

Frontend:

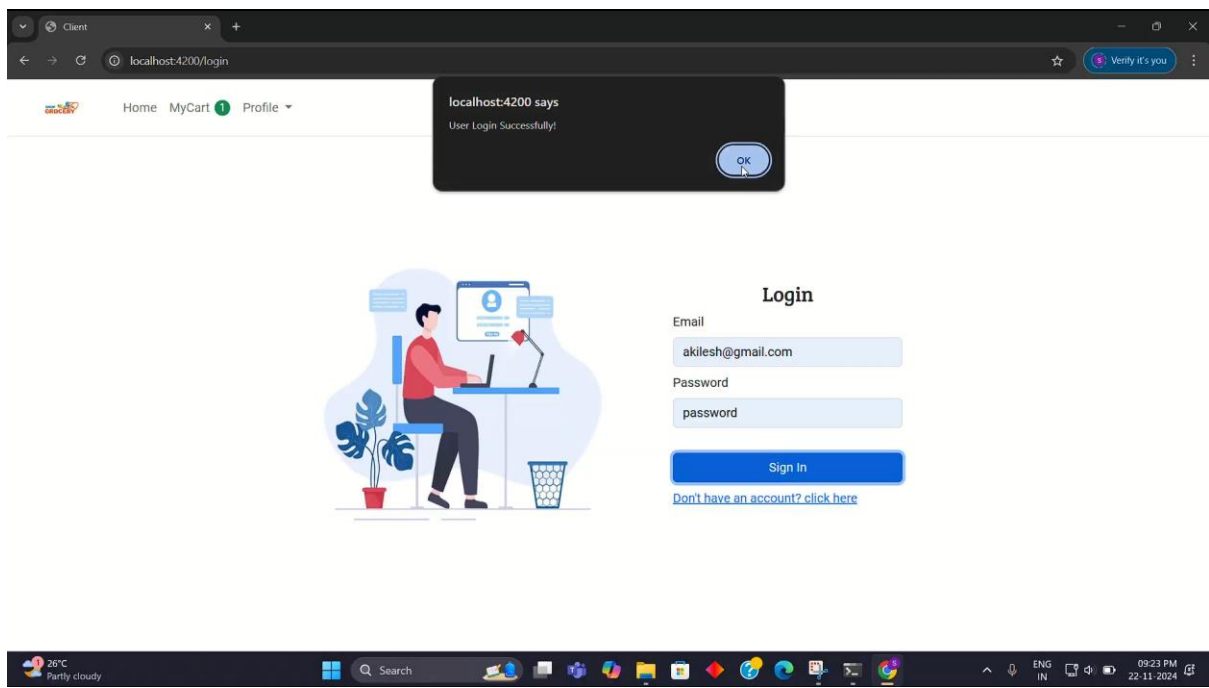
Client Home page:



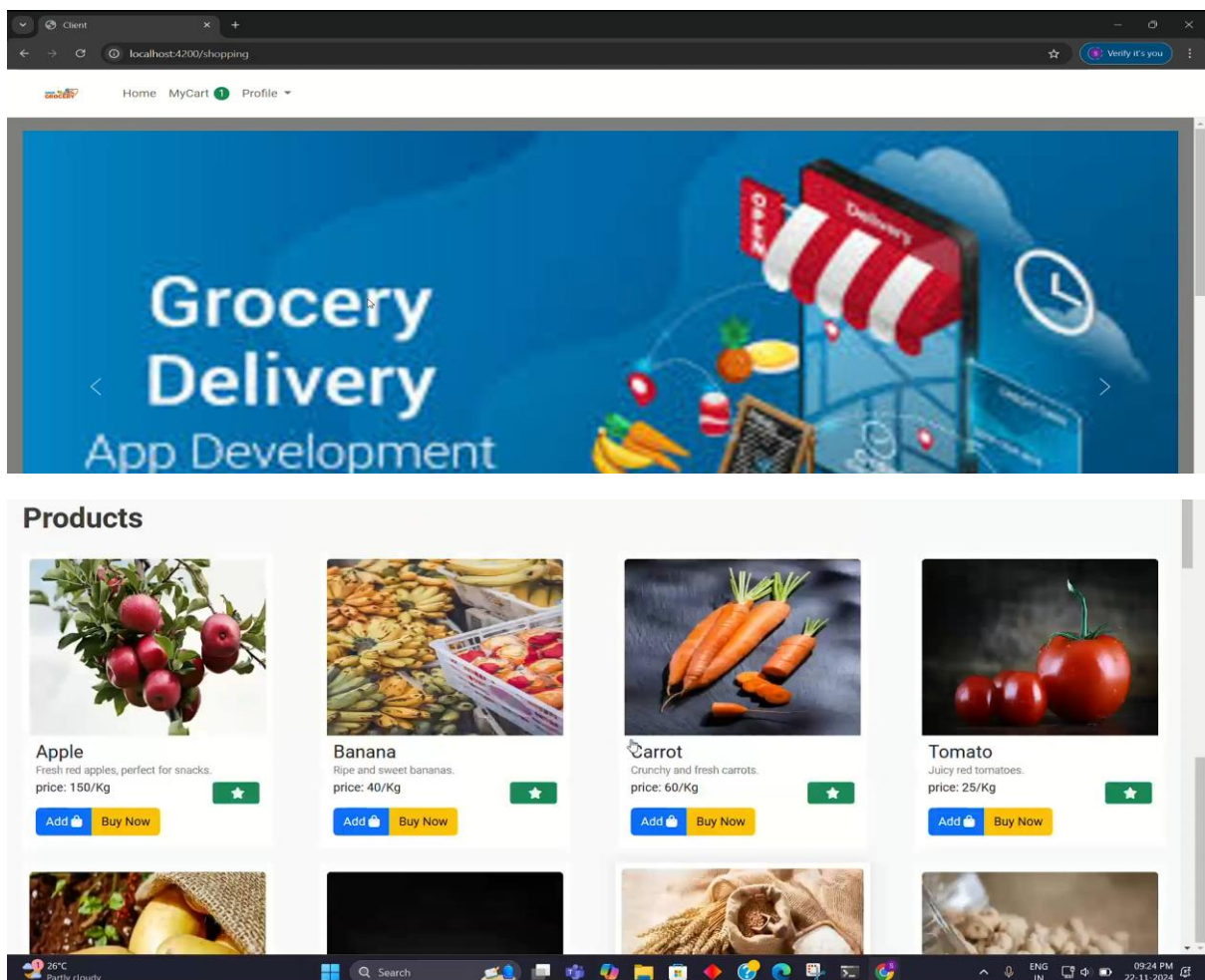
Sign up page



Login page:



Home page:



Add chat:

The image displays two screenshots of a web application interface.

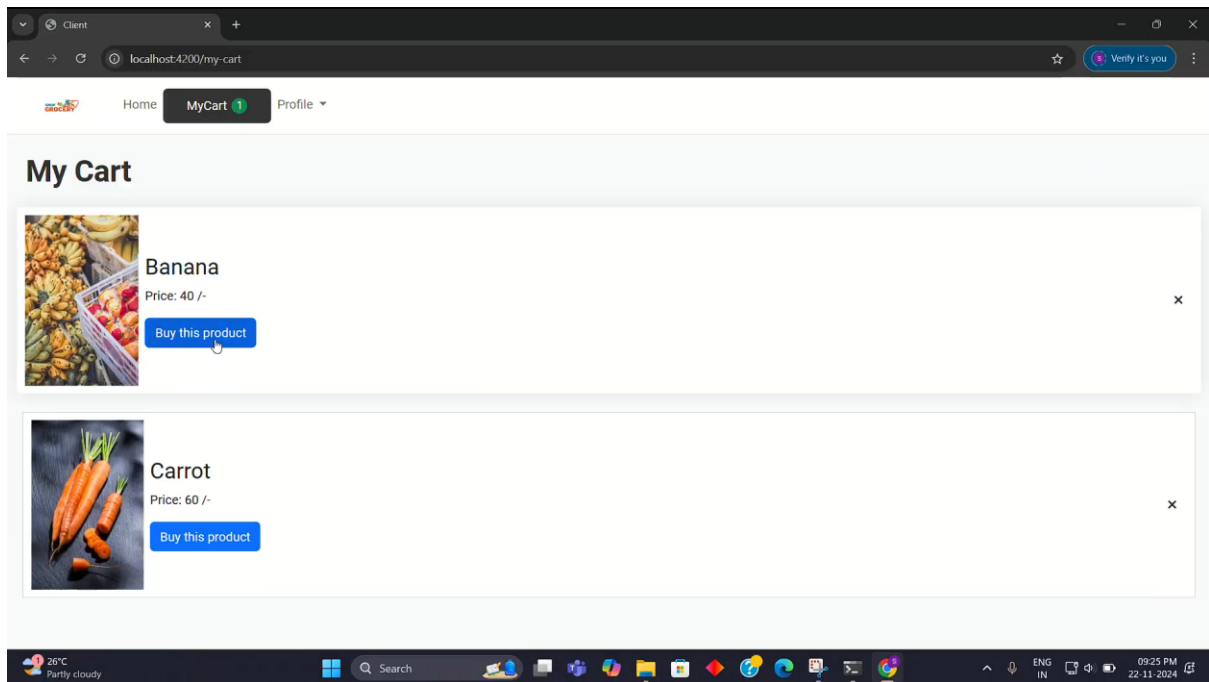
Top Screenshot (Product Detail Page):

- Browser:** Client, localhost:4200/product-details/6/3e1991af188a7734a530c4
- Navigation:** Home, MyCart (1), Profile
- Product:** Apple. Fresh red apples, perfect for snacks. Price: 150/-
- Image:** A bunch of red apples hanging from a tree.
- Buttons:** Buy Now, Add To Cart

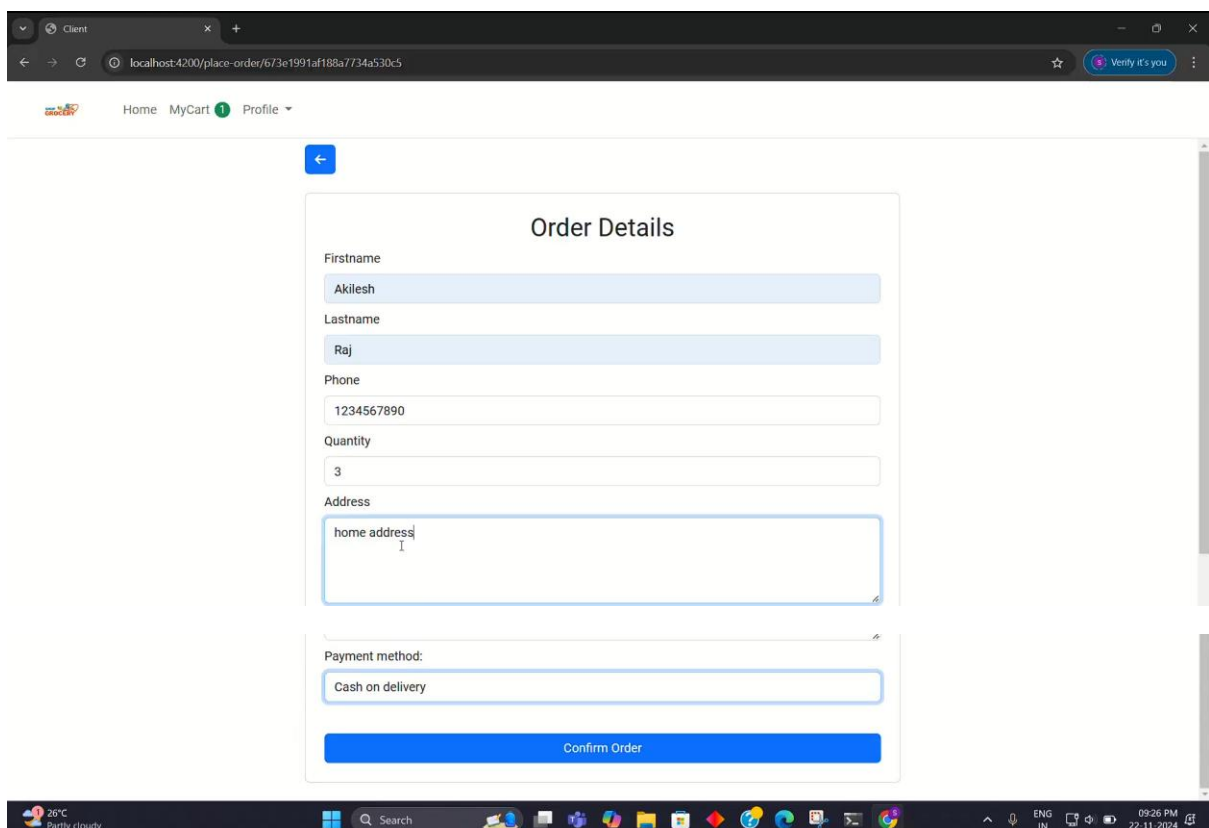
Bottom Screenshot (Shopping Page):

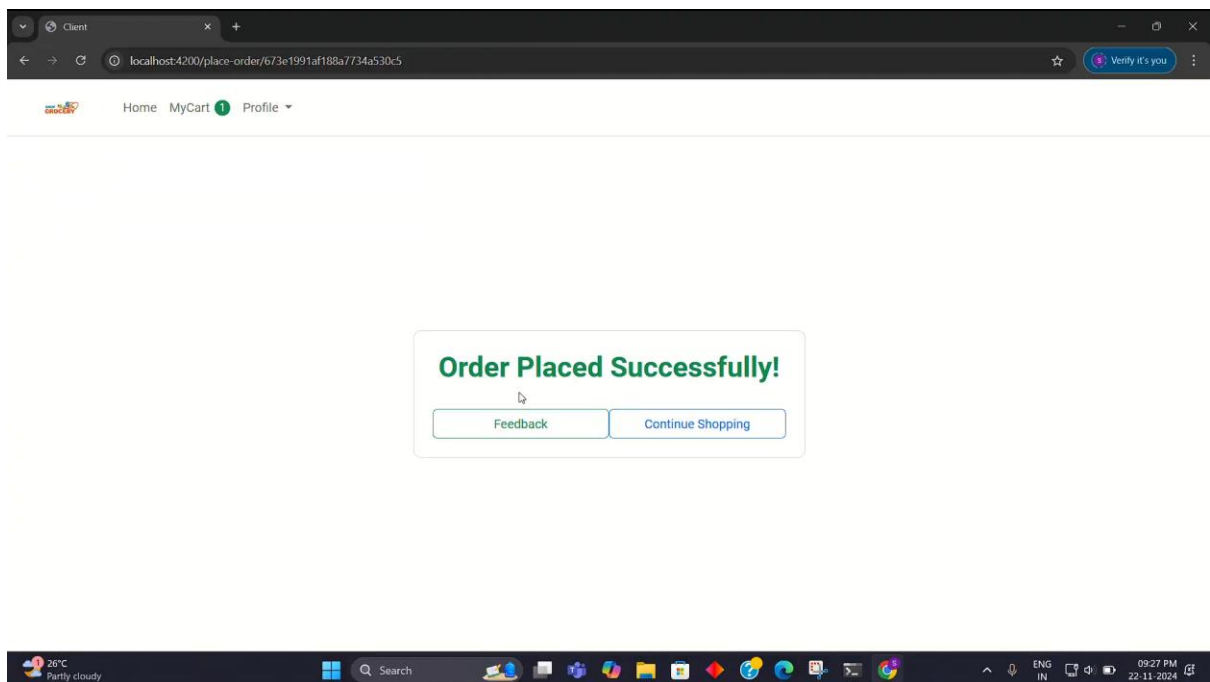
- Browser:** Client, localhost:4200/shopping
- Navigation:** Home, MyCart (1), Profile
- Message:** localhost:4200 says Product added to cart! (with OK button)
- Filters:** Search, Fruits, Vegetables
- Products:**
 - Apple:** Fresh red apples, perfect for snacks. price: 150/Kg. Add, Buy Now, Star icon.
 - Banana:** Ripe and sweet bananas. price: 40/Kg. Add, Buy Now, Star icon.
 - Carrot:** Crunchy and fresh carrots. price: 60/Kg. Add, Buy Now, Star icon.
 - Tomato:** Juicy red tomatoes. price: 25/Kg. Add, Buy Now, Star icon.

Buy the Product:

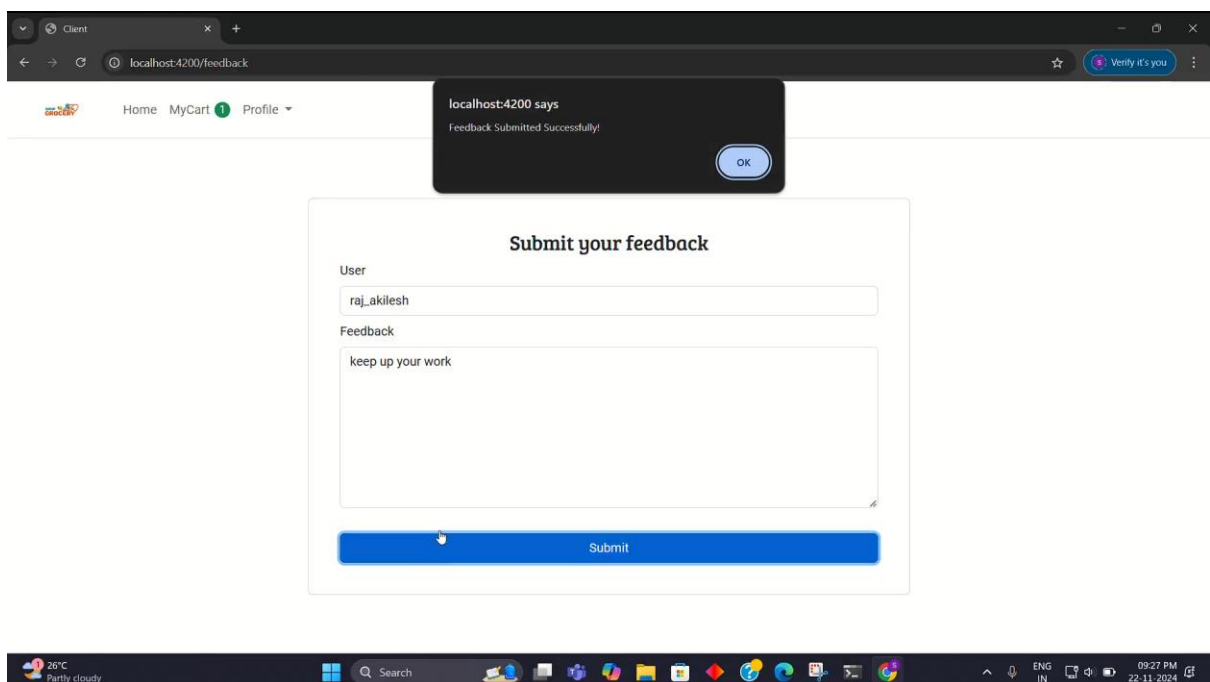


Order details

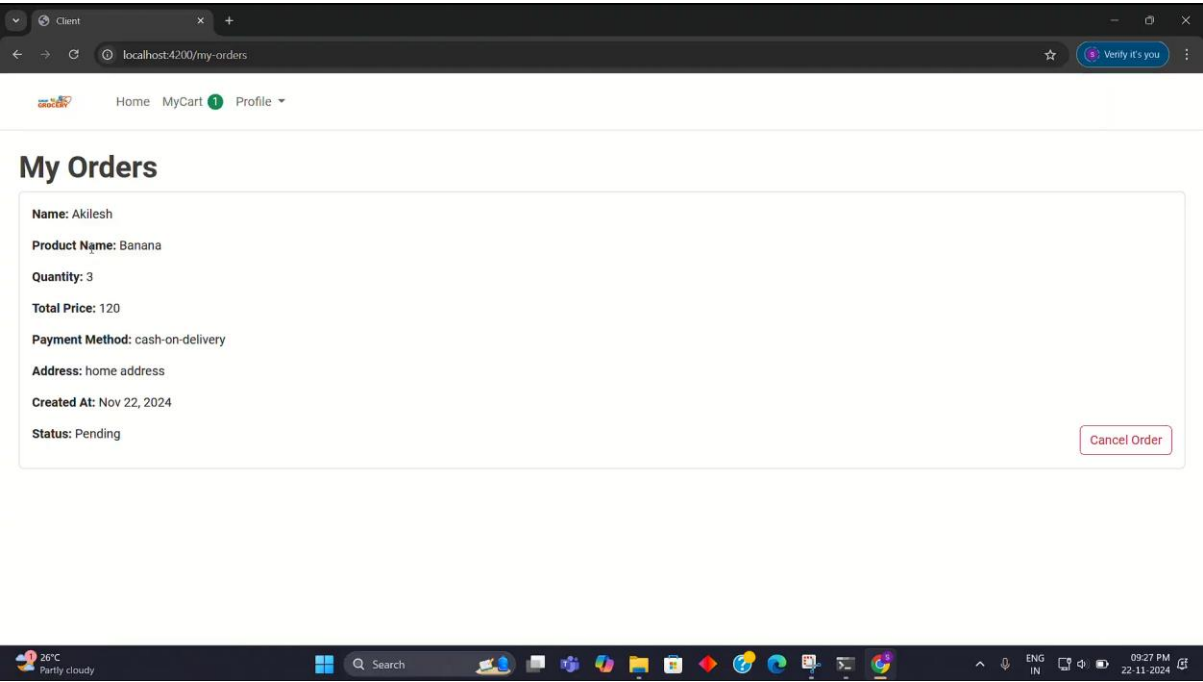




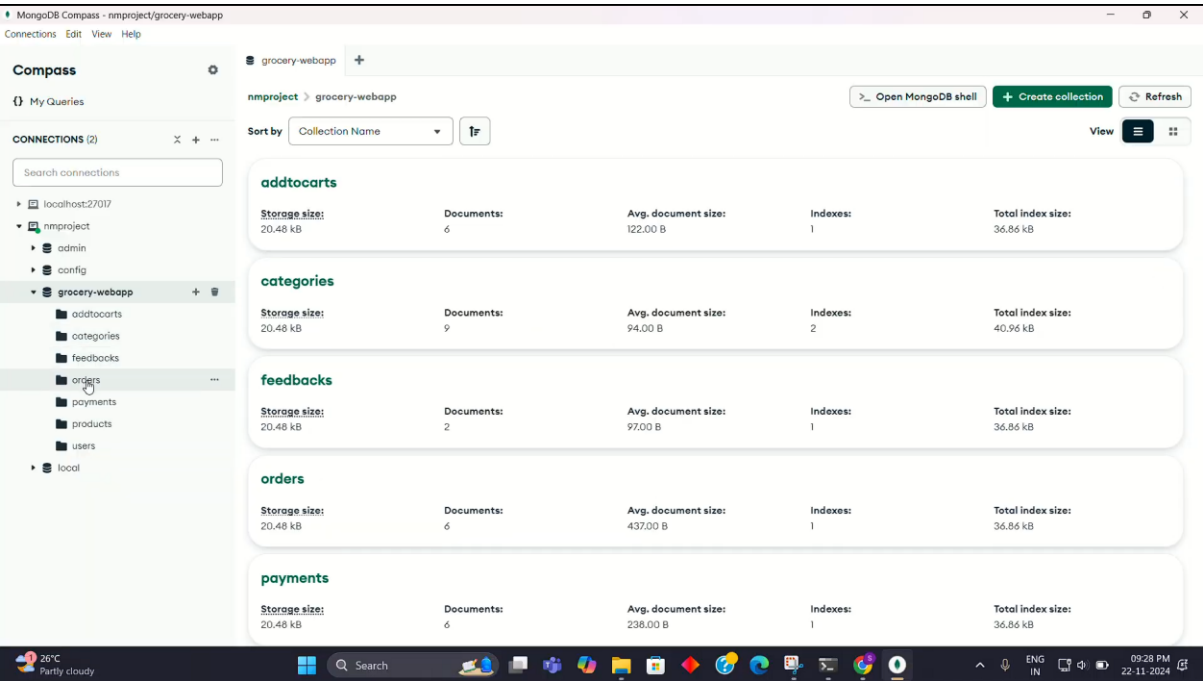
Feedback:



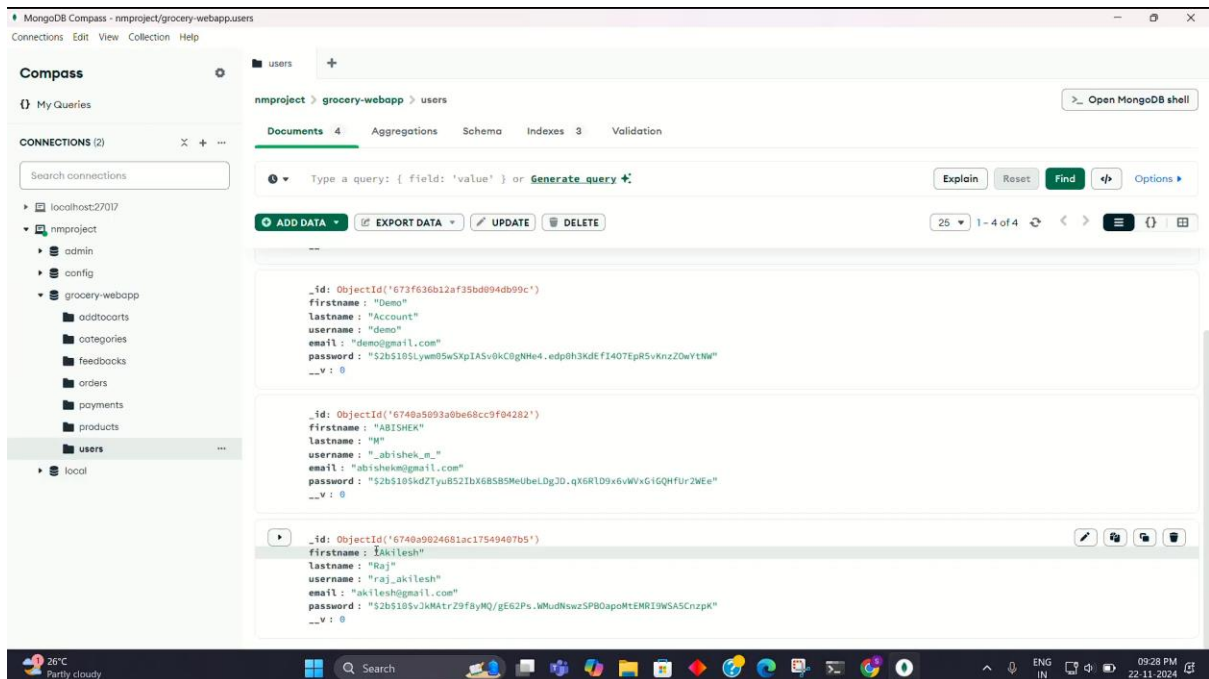
Order list:



Backend:



User:



MongoDB Compass - nmproject/grocery-webapp.users

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (2)

Search connections

localhost:27017

nmproject

admin

config

grocery-webapp

addtoorders

categories

feedbacks

orders

payments

products

users

local

nmproject > grocery-webapp > users

Documents 4 Aggregations Schema Indexes 3 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

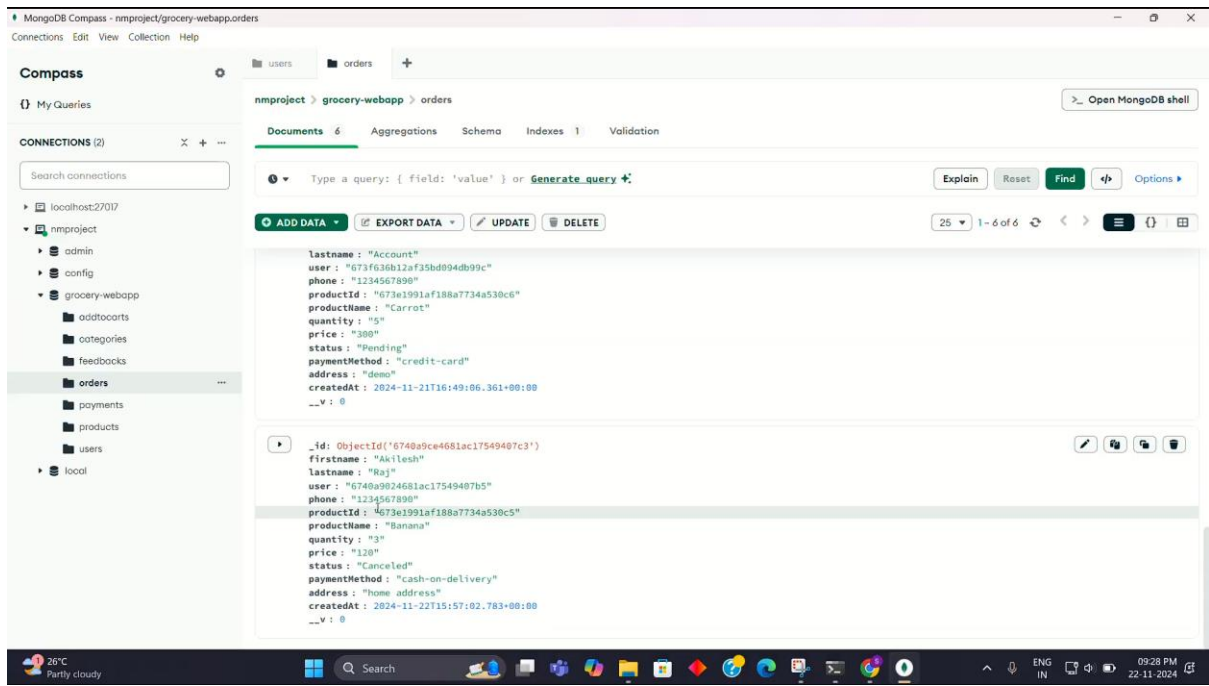
25 1 - 4 of 4

```
{
  "_id": ObjectId("673f636b12af35bd094db99c"),
  "firstname": "Demo",
  "lastname": "Account",
  "username": "demo",
  "email": "demo@gmail.com",
  "password": "52b5185Lyw0Sw5xpIA5v0kC0gNHe4..edp0h3KdEf1407EpR5vKnz2QeYtWw",
  "__v": 0
}
```

```
{
  "_id": ObjectId("6740a5093a0be68cc9f04282"),
  "firstname": "ABISHAKH",
  "lastname": "M",
  "username": "_abishak_m_",
  "email": "abishakm@gmail.com",
  "password": "52b5185kdZ7yu852IbX6B585MeUbeLdg3D..qx6R1D9x6vWvG6GQHfUr2WFe",
  "__v": 0
}
```

```
{
  "_id": ObjectId("6740a9024681ac17549407b5"),
  "firstname": "Akilesh",
  "lastname": "Raj",
  "username": "raj_akilesh",
  "email": "akilesh@gmail.com",
  "password": "52b5185vJkMatrZ9fByMQ/gE62Ps..wMudNswzSP80apoMtEHR19WSA5CnzpK",
  "__v": 0
}
```

Orders:



MongoDB Compass - nmproject/grocery-webapp.orders

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (2)

Search connections

localhost:27017

nmproject

admin

config

grocery-webapp

addtoorders

categories

feedbacks

orders

payments

products

users

local

nmproject > grocery-webapp > orders

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 6 of 6

```
{
  "lastname": "Account",
  "user": "673f636b12af35bd094db99c",
  "phone": "1234567890",
  "productId": "673e1991af188a7734a538c6",
  "productName": "Carrot",
  "quantity": "5",
  "price": "300",
  "status": "Pending",
  "paymentMethod": "credit-card",
  "address": "demo",
  "createdAt": 2024-11-21T16:49:06.361+00:00,
  "__v": 0
}
```

```
{
  "_id": ObjectId("6740a9ce4681ac17549407c3"),
  "firstname": "Akilesh",
  "lastname": "Raj",
  "user": "6740a9024681ac17549407b5",
  "phone": "1234567890",
  "productId": "673e1991af188a7734a538c5",
  "productName": "Banana",
  "quantity": "3",
  "price": "120",
  "status": "Canceled",
  "paymentMethod": "cash-on-delivery",
  "address": "home address",
  "createdAt": 2024-11-22T15:57:02.783+00:00,
  "__v": 0
}
```

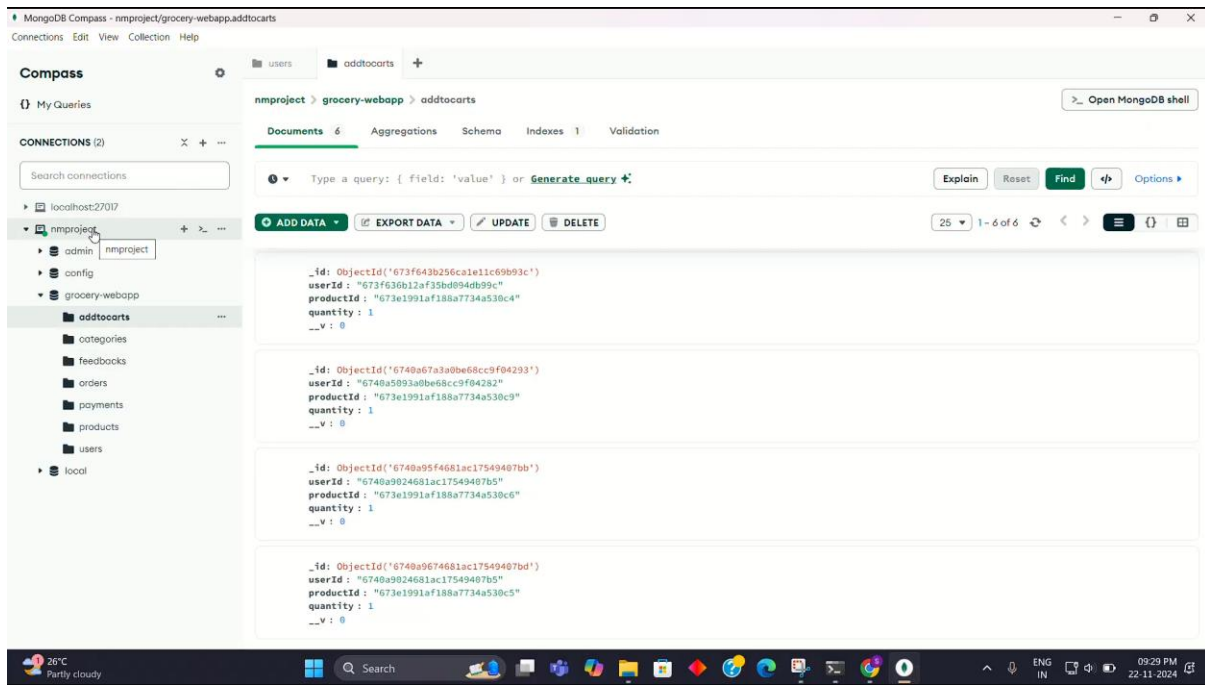
Feedback:

The screenshot shows the MongoDB Compass interface for the 'nmproject' database, specifically the 'feedbacks' collection. The left sidebar shows the database structure with 'nmproject' expanded, showing collections like 'admin', 'config', 'grocery-webapp', 'oddtocarts', 'categories', 'feedbacks', 'orders', 'payments', 'products', 'users', and 'local'. The 'feedbacks' collection is selected. The main panel shows two documents in the 'Documents' tab. The first document has fields: '_id: ObjectId('673f6472256ca1e11c69b947')', 'user: "hello"', 'message: "good product"', 'createdAt: 2024-11-21T16:48:50.147+00:00', and '___v: 0'. The second document has fields: '_id: ObjectId('6740a9e84681ac17549407c7')', 'user: "raj_akilesh"', 'message: "keep up your work"', 'createdAt: 2024-11-22T15:57:28.292+00:00', and '___v: 0'. The top bar shows 'nmproject > grocery-webapp > feedbacks' and a button to 'Open MongoDB shell'. The bottom status bar shows '26°C Partly cloudy' and the time '09:29 PM 22-11-2024'.

Categories:

The screenshot shows the MongoDB Compass interface for the 'nmproject' database, specifically the 'categories' collection. The left sidebar shows the database structure with 'nmproject' expanded, showing collections like 'admin', 'config', 'grocery-webapp', 'oddtocarts', 'categories', 'feedbacks', 'orders', 'payments', 'products', 'users', and 'local'. The 'categories' collection is selected. The main panel shows five documents in the 'Documents' tab. The first document has fields: 'category: "snacks"', 'description: "Chips, biscuits, and other snacks."', and '_id: ObjectId('673cc72af61c168746f08056')'. The second document has fields: 'category: "Beverages"', 'description: "Soft drinks, juices, and other beverages."', and '_id: ObjectId('673cc72af61c168746f08057')'. The third document has fields: 'category: "Bakery"', 'description: "Bread, cakes, and bakery products."', and '_id: ObjectId('673cc72af61c168746f08058')'. The fourth document has fields: 'category: "Frozen Foods"', 'description: "Frozen meats, vegetables, and meals."', and '_id: ObjectId('673cc72af61c168746f08059')'. The fifth document has fields: 'category: "Meat & Fish"', 'description: "Fresh meat and seafood."', and '_id: ObjectId('673cc72af61c168746f0805a')'. The sixth document has fields: 'category: "Cereals"', 'description: "Breakfast cereals and grains."', and '_id: ObjectId('673cc72af61c168746f0805b')'. The top bar shows 'nmproject > grocery-webapp > categories' and a button to 'Open MongoDB shell'. The bottom status bar shows '26°C Partly cloudy' and the time '09:29 PM 22-11-2024'.

Add chat:



Conclusion:

The **Grocery Webapp** project successfully demonstrates the development of a full-stack web application using the MERN stack to deliver a responsive and feature-rich online grocery shopping experience. The project showcases the seamless integration of front-end and back-end technologies to create an intuitive and user-friendly interface, secure user authentication, and a robust backend for efficient management of product inventory, customer data, and order processing. By emphasizing modern web development practices, this project offers a comprehensive platform that can be easily scaled and enhanced with additional features, such as personalized recommendations, real-time order tracking, and advanced search functionalities. The Grocery Webapp provides a strong foundation for an online grocery store, ensuring a smooth and reliable shopping experience for customers and a streamlined management system for administrators.