GROCERY WEB APP

1.INTRODUCTION:

PROJECT TITLE:SHOPSMART-ONLINNE GROCERY SHOPPING APP

Welcome to our Grocery Web App, your one-stop shop for all your grocery need With our user-friendly interface and wide selection of highquality products, we aim to make your grocery shopping experience convenient

and enjoyable. Whether you're looking for fresh produce, pantry staples, or

household essentials, our app has you covered. Explore our virtual aisles, add

items to your cart with ease, and have your groceries delivered right to your

doorstep. Experience the future of grocery shopping with our Grocery Web App

2.TEAM MEMBERS:

1.K.PUJITHA

2.E.TANUJA

3.P.SAI KIRAN

4.Nikesh Appari

3.PROJECT OVERVIEW

Purpose:

ShopSmart is an online grocery web application that allows users to browse, search, and purchase groceries from the comfort of their homes. The goal is to simplify grocery shopping with a user-friendly interface and fast delivery service.

Features:

- User registration and login

- Product catalog with categories

- Cart and checkout system

- Admin panel to manage products and orders

- Order history tracking

4.ARCHITECTURE:

Frontend:

Built using React.js, the frontend includes pages for home, login, product listing, cart, and admin dashboard. Uses React Router for navigation and Axios for API requests.

Backend:

Developed with Node.js and Express.js, it handles user authentication, product management, and order processing.

Database:

Uses MongoDB to store data such as user info, product details, orders, and admin credentials. Mongoose is used as the ODM.

5.SET UP INSTRUCTIONS:

Prerequisites:

- Node.js

- MongoDB

- npm or yarn

Installation:

1. Clone the repository:

git clone https://github.com/yourusername/shopsmart.git

cd shopsmart

2. Install dependencies:

cd client

npm install

cd ../server

npm install

3. Set up environment variables (.env file):

MONGO\_URI=your\_mongodb\_connection\_string

JWT\_SECRET=your\_jwt\_secret

PORT=5000

6.ROLES AND RESPONSBILITY

USER:

Registration and Authentication: Users are responsible for creating an

account on the platform and securely logging in to access its features.

● Browsing and Shopping: Users can browse products, add them to their cart,

and proceed to checkout for purchasing.

● Payment: Users are responsible for making payments for their orders

using the available payment methods.

● Order Management: Users can view their order history, track their

deliveries, and manage their account details.

● Feedback and Reviews: Users can provide feedback on products and

services and leave reviews to help other users make informed decisions.

● Compliance: Users are expected to adhere to the platform's terms and

Conditions and privacy policy.

ADMIN:

● User Management: Admins can manage user accounts, including creating,

updating, and deleting accounts as necessary.

● Product Management: Admins are responsible for managing the platform's

product listings, including adding new products, updating existing ones,

and removing outdated products.

● Order Management: Admins can view and manage all orders placed on the

platform, including processing payments, tracking deliveries, and handling

returns or refunds.

● Content Management: Admins can manage the platform's content,

including creating and updating informational pages, blog posts, and other

content.

● Analytics and Reporting: Admins can generate reports and analyze data to

gain insights into the platform's performance and user behavior.

● Compliance and Security: Admins are responsible for ensuring that the

platform complies with relevant laws and regulations and that user data is

kept secure.

● Customer Support: Admins can provide support to users, including

responding to inquiries, resolving issues, and handling complaints.

● Marketing and Promotion: Admins can create and manage marketing

campaigns and promotions to attract and retain users.

7. Milestone 1: Project Setup and Configuration:

1. Install required tools and software:

● Node.js.

● MongoDB.

● Create-react-app.

2. Create project folders and files:

● Client folders.

● Server folders.

3. Install Packages:

Frontend npm Packages

● Axios.

● React-Router –dom.

● Bootstrap.

● React-Bootstrap.

● React-icons.

Backend npm Packages

● Express.

8. Milestone 2: Backend Development:

● Setup express server

1. Create index.js file in the server (backend folder).

2. Create a .env file and define port number to access it globally.

3. Configure the server by adding cors, body-parser.

● User Authentication:

• Create routes and middleware for user registration, login, and

logout.

• Set up authentication middleware to protect routes that require

user authentication.

● Define API Routes:

• Create separate route files for different API functionalities such

as users orders, and authentication.

•

Define the necessary routes for listing products, handling user

registration and login,managing orders, etc.

•

Implement route handlers using Express.js to handle requests

and interact with the database.

● Implement Data Models:

•

Define Mongoose schemas for the different data entities like

products, users, and orders.

• Create corresponding Mongoose models to interact with the

MongoDB database.

• Implement CRUD operations (Create, Read, Update, Delete) for each model to perform database operations.

● User Authentication:

• Create routes and middleware for user registration, login, and

logout.

• Set up authentication middleware to protect routes that require

user authentication.

● Error Handling:

• Implement error handling middleware to catch and handle any

errors that occur during the API requests.

• Return appropriate error responses with relevant error messages

and HTTP status codes.

Milestone 3: Database:

1. Configure MongoDB:

● Install Mongoose.

● Create database connection.

● Create Schemas&Models.

2. Connect database to backend:

Now, make sure the database is connected before performing any of the

actions through the backend. The connection code looks similar to the one provided below.



