

Grocery web App

Introduction:

Welcome to our Grocery Web App, your one-stop shop for all your grocery needs! With our user-friendly interface and wide selection of high-quality 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 today!

The Admin Login page is a secure entry point for authorized administrators of the G-Mart Grocery Web App. This interface is essential to restrict access to sensitive sections such as dashboard management, user control, product handling, and order processing.

Description:

Our Grocery Web App is more than just a convenient way to shop for groceries—it's a comprehensive solution designed to enhance every aspect of your shopping experience. Imagine a virtual supermarket at your fingertips, offering an extensive selection of high-quality products that cater to your every need.

From fresh produce sourced directly from local farms to pantry staples and household essentials, our app provides everything you need to keep your kitchen well-stocked. Our user-friendly interface ensures that you can navigate our virtual aisles effortlessly, easily finding what you're looking for and adding items to your cart with a simple click.

With features like flexible delivery times, secure payments, and dietary preference filters (e.g., vegan, organic, gluten-free), our app brings the supermarket to your doorstep.

Scenario Case Study: Priya

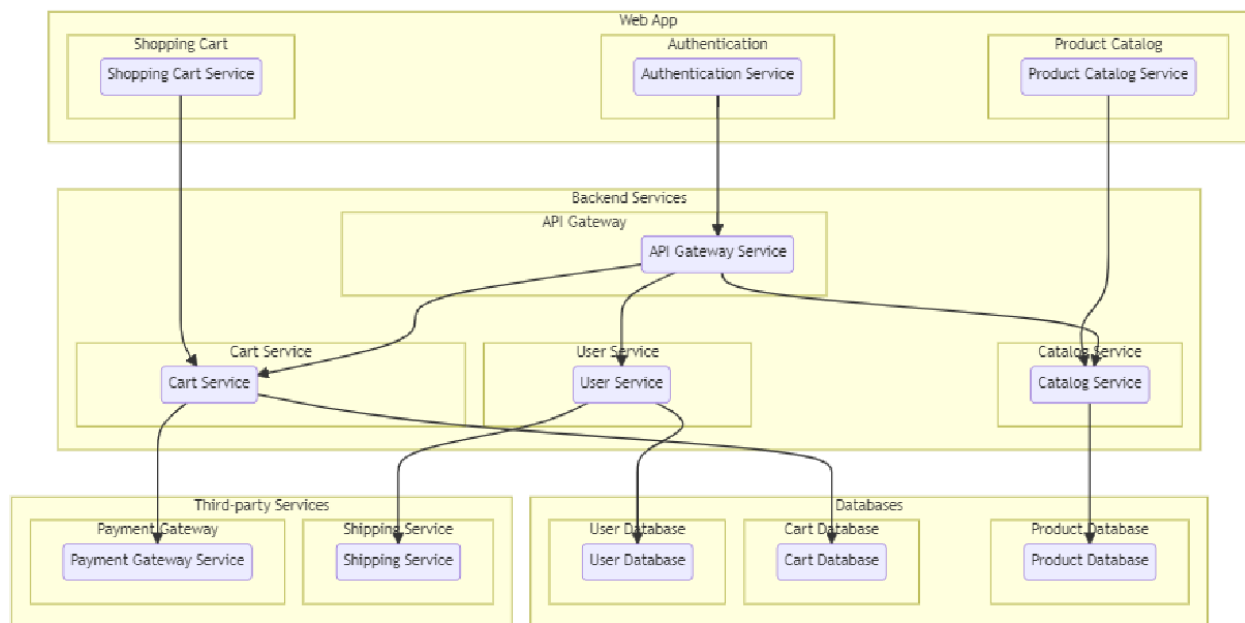
Meet **Priya**, a busy professional who loves to cook but struggles to visit grocery stores regularly.

Priya's Journey with the App:

- **User Registration:** Priya registers with her details and preferences.
- **Product Browsing:** She browses groceries sorted into clear categories.
- **Smart Suggestions:** The app offers personalized product recommendations.
- **Delivery Choices:** Priya schedules doorstep delivery at a time that suits her.
- **Secure Payment:** She pays online through secure gateways.
- **Order Tracking:** She tracks her order status live.
- **Customer Support:** A support team is available to assist her at any time.

Thanks to the Grocery Web App, Priya now enjoys a stress-free, efficient, and personalized shopping experience.

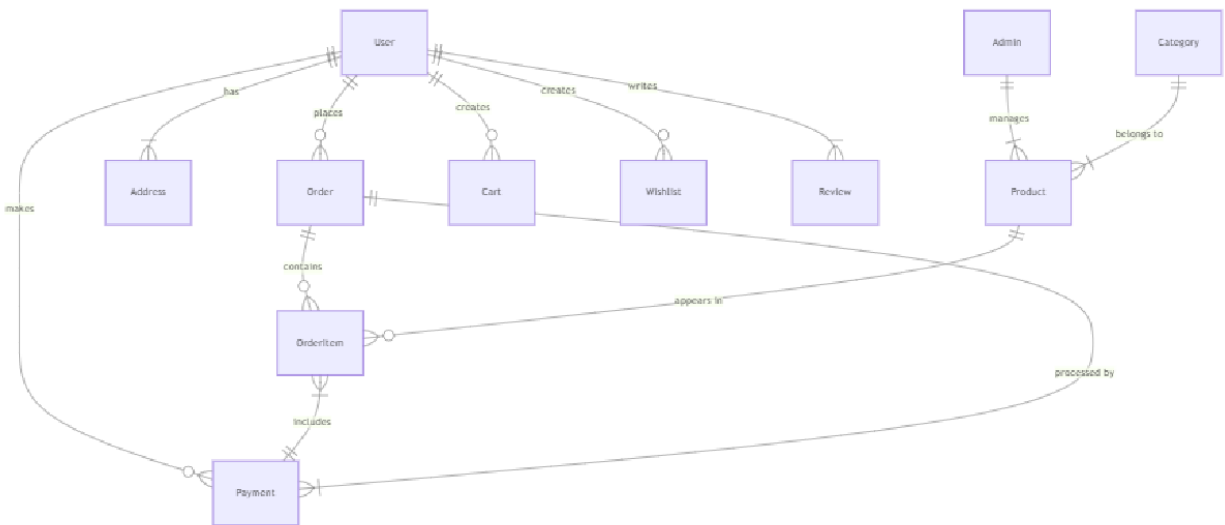
Technical Architecture:-



The technical architecture of an grocery-webapp app typically involves a client-server model, where the frontend represents the client and the backend serves as the server. The frontend is responsible for user interface, interaction, and presentation, while the backend handles data storage, business logic, and integration with external services like

payment gateways and databases. Communication between the frontend and backend is typically facilitated through APIs, enabling seamless data exchange and functionality.

ER-Diagram:



The technical architecture of an grocery-webapp app typically involves a client-server model, where the frontend represents the client and the backend serves as the server. The frontend is responsible for user interface, interaction, and presentation, while the backend handles data storage, business logic, and integration with external services like payment gateways and databases. Communication between the frontend and backend is typically facilitated through APIs, enabling seamless data exchange and functionality.

Key Features:

Product Catalog: Our grocery-webapp app provides an extensive product catalog with various categories and subcategories. Users can easily search, browse, and filter products based on their preferences, making it effortless to find the desired items.

Shopping Cart and Checkout: The app includes a shopping cart feature that enables users to add products, review their cart, and proceed to checkout. The checkout process offers multiple payment options, ensuring a smooth and secure transaction experience.

Product Reviews and Ratings: Customers can provide feedback and rate products, helping other users make informed purchasing decisions. This feature fosters a sense of community and trust among users.

Order Tracking: Once an order is placed, users can track its status in real-time. They receive updates on order processing, shipping, and delivery, providing transparency and peace of mind.

Admin Dashboard: For administrators, our grocery-webapp app offers a comprehensive dashboard to manage products, inventory, orders, and customer information. It provides insights into sales performance, stock levels, and customer analytics, enabling efficient business operations.

Order Management: The app manages the order lifecycle, including order placement, tracking, and status updates. Users can view their order history, track shipments, and request returns or cancellations.

Search and Filtering: Users can search for products using keywords and apply filters to narrow down the search results based on criteria such as price range, brand, or customer ratings.

Shopping Cart and Checkout: The app includes a shopping cart feature that enables users to add products, review their cart, and proceed to checkout. The checkout process offers multiple payment options, ensuring a smooth and secure transaction experience.

Product Reviews and Ratings: Customers can provide feedback and rate products, helping other users make informed purchasing decisions. This feature fosters a sense of community and trust among users.

Order Tracking: Once an order is placed, users can track its status in real-time. They receive updates on order processing, shipping, and delivery, providing transparency and peace of mind.

Admin Dashboard: For administrators, our grocery-webapp app offers a comprehensive dashboard to manage products, inventory, orders, and customer information. It provides insights into sales performance, stock levels, and customer analytics, enabling efficient business operations.

Order Management: The app manages the order lifecycle, including order placement, tracking, and status updates. Users can view their order history, track shipments, and request returns or cancellations.

Search and Filtering: Users can search for products using keywords and apply filters to narrow down the search results based on criteria such as price range, brand, or customer ratings.

PRE REQUISITES:

To develop a full-stack Ecommerce App for Furniture Tool using React js, Node.js, Express js and MongoDB, there are several prerequisites you should consider. Here are the key prerequisites for developing such an application:

Node.js and npm: Install Node.js, which includes npm (Node Package Manager), on your development machine. Node.js is required to run JavaScript on the server side.

- Download: <https://nodejs.org/en/download/>
- Installation instructions: <https://nodejs.org/en/download/package-manager/>

MongoDB: Set up a MongoDB database to store hotel and booking information. Install MongoDB locally or use a cloud-based MongoDB service.

- Download: <https://www.mongodb.com/try/download/community>
- Installation instructions: <https://docs.mongodb.com/manual/installation/>

Express.js: Express.js is a web application framework for Node.js. Install Express.js to handle server-side routing, middleware, and API development.

- Installation: Open your command prompt or terminal and run the following command: **npm install express**

React js: React is a JavaScript library for building client-side applications. And Creating Single Page Web-Appliaction

Getting Started

Create React App is an officially supported way to create single-page React applications. It offers a modern build setup with no configuration.

Quik Start

```
npm create vite@latest  
cd my-app
```

```
npm install  
npm run dev
```

If you've previously installed create-react-app globally via `npm install -g create-react-app`, we recommend you uninstall the package using `npm uninstall -g create-react-app` or `yarn global remove create-react-app` to ensure that npx always uses the latest version.

Create a new React project:

- Choose or create a directory where you want to set up your React project.
- Open your terminal or command prompt.
- Navigate to the selected directory using the `cd` command.
- Create a new React project by running the following command: `npx create-react-app your-app-name`. Wait for the project to be created:
- This command will generate the basic project structure and install the necessary dependencies

Navigate into the project directory:

- After the project creation is complete, navigate into the project directory by running the following command: **`cd your-app-name`**

Start the development server:

- To launch the development server and see your React app in the browser, run the following command: **`npm run dev`**
- The `npm start` will compile your app and start the development server.
- Open your web browser and navigate to <https://localhost:5173> to see your React app.

You have successfully set up React on your machine and created a new React project. You can now start building your app by modifying the generated project files in the `src` directory.

Please note that these instructions provide a basic setup for React. You can explore more advanced configurations and features by referring to the official React documentation: <https://react.dev/>

HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.

Front-end Library: Utilize React to build the user-facing part of the application, including products listings, booking forms, and user interfaces for the admin dashboard.

Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

- Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>

Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

- Visual Studio Code: Download from <https://code.visualstudio.com/download>
- Sublime Text: Download from <https://www.sublimetext.com/download>
- WebStorm: Download from <https://www.jetbrains.com/webstorm/download>

Roles and Responsibility

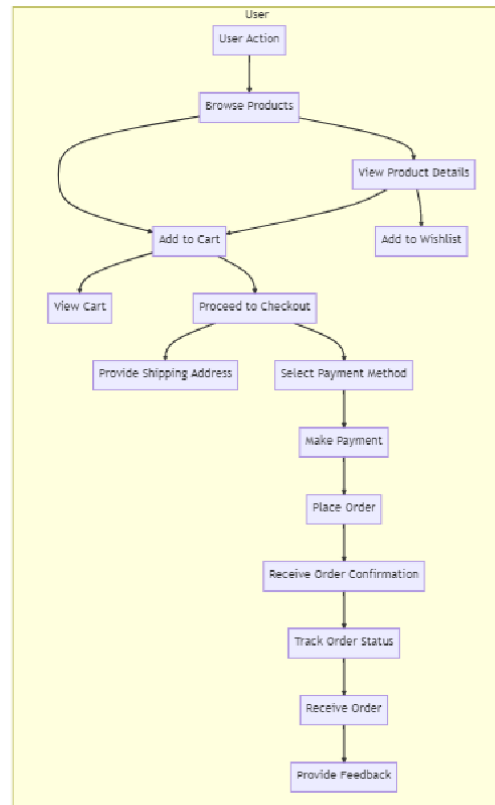
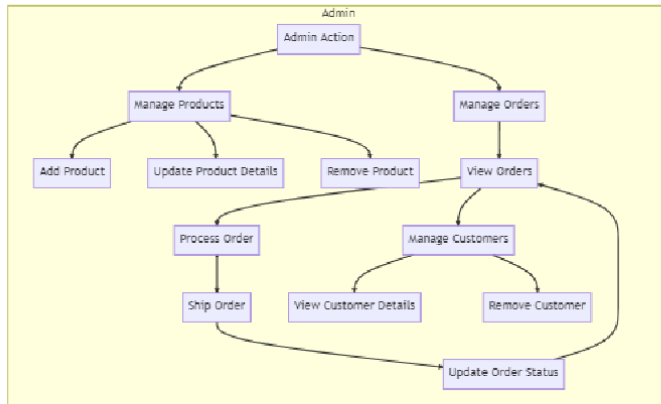
User:-

- **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.

Admin & User Flow:



The project flow for a grocery-web app involves user actions such as browsing products, adding items to the cart, proceeding to checkout, providing shipping details, selecting payment methods, making payments, and receiving order confirmation. Admin actions include managing products, viewing and processing orders, managing customers, and updating product details.

PROJECT FLOW:-

Before starting to work on this project, let's see the demo.

Demo link:-

https://drive.google.com/file/d/1HLowclqs2d8lxTprS2jqPmR4AOnUW8xD/view?usp=drive_link

or follow the videos below for better understanding.

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.
- Mongoose.
- Cors.

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.

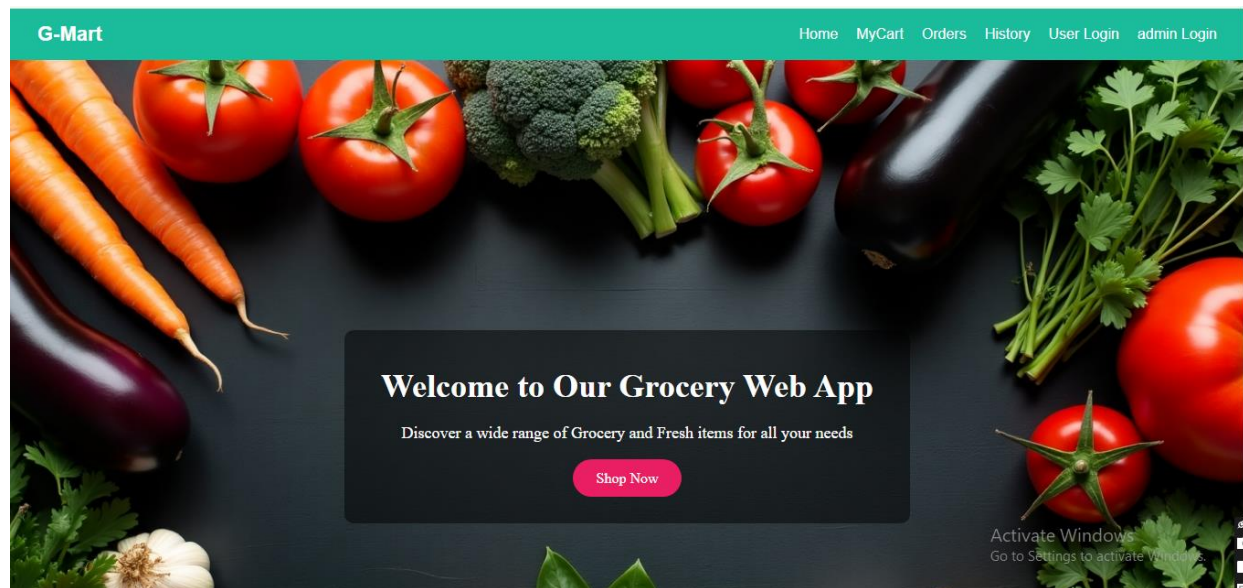
- **Implement Data Models:**

- 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.

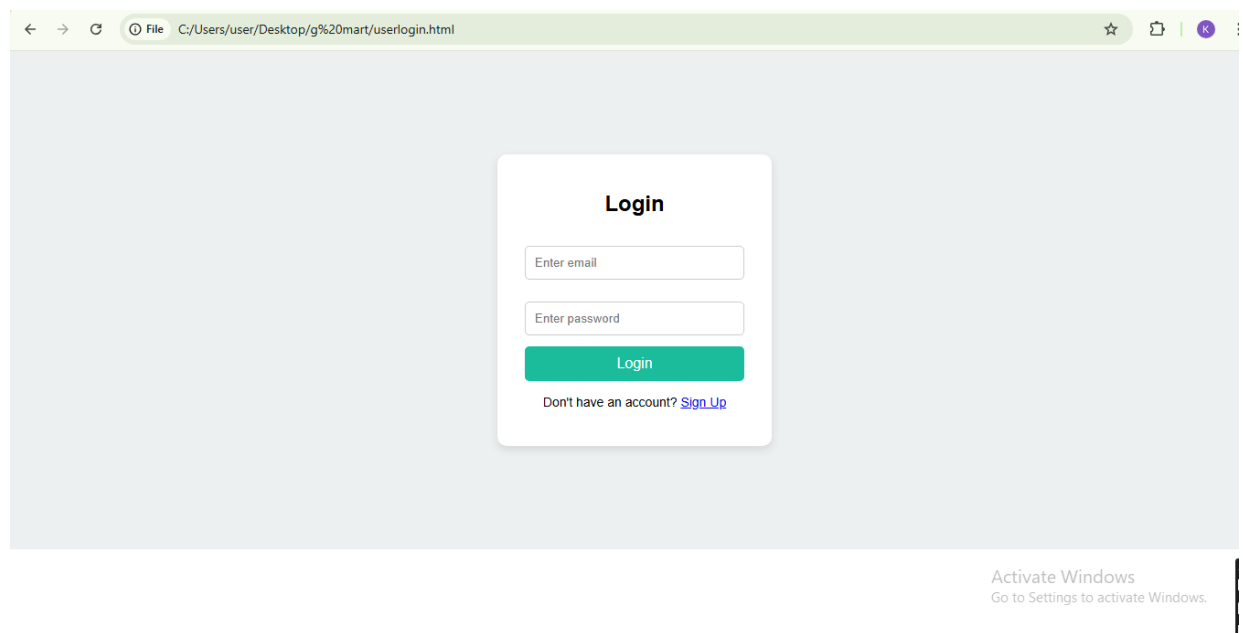
Milestone 3: Project Implementation:

Finally, after finishing coding the projects we run the whole project to test it's working process and look for bugs. Now, let's have a final look at the working of our Darshan Ease.


Landing page:-




Login Page:-




Items Page:-




Apple
\$1.20
[Buy Now](#)




Orange
\$1.00
[Buy Now](#)



Milk
\$0.23
[Buy Now](#)



Cashew
\$2.50
[Buy Now](#)



Chicken
\$2.93
[Buy Now](#)


Activate Windows
Go to Settings to activate Windows.

My Cart:-


G-Mart

[Home](#) [MyCart](#) [Orders](#) [History](#) [User Login](#)


My Cart




Apple
\$1.20
[Remove from Cart](#)
[Buy Now](#)



Milk
\$0.23
[Remove from Cart](#)
[Buy Now](#)



Chicken
\$2.92
[Remove from Cart](#)
[Buy Now](#)



orange
\$1.00
[Remove from Cart](#)
[Buy Now](#)

Activate Windows
Go to Settings to activate Windows.

My Orders Page:-

My Current Orders

Order ID	Product	Quantity	Status	Payment
ORD101	Apple	2	Pending	Unpaid
ORD102	Milk	1	Shipped	Paid

Activate Windows
Go to Settings to activate Windows.

My History Page:-**My Order History**

Order ID	Product	Quantity	Status	Payment	Date
ORD091	Cashew	1	Delivered	Paid	2025-06-20
ORD092	Orange	3	Delivered	Paid	2025-06-18

Activate Windows
Go to Settings to activate Windows.

Place Order Page:-

[G-Mart](#) [Home](#) [MyCart](#) [Orders](#) [History](#) [Logout](#)

Order Details

First Name:

Enter your first name

Last Name:

Enter your last name

Phone:

Enter your phone number

Quantity:

Enter the quantity

Address:

Enter your address

Payment Method:

Cash on Delivery (COD)

Submit

localhost3000/login

Admin Dashboard Page:

G-Mart Admin

[Dashboard](#) [Users](#) [Products](#) [Orders](#) [Logout](#)

Admin Dashboard Overview

Total Users

12

Total Products

27

Total Orders

8

Activate Windows
Go to Settings to activate Windows.

Users Page:-

G-Mart Admin

[Dashboard](#)[Users](#)[Products](#)[Orders](#)[Logout](#)

Registered Users

User ID	Email	Action
101	user1@example.com	Delete
102	user2@example.com	Delete
103	user3@example.com	Delete

Activate Windows

Go to Settings to activate Windows.

Add Product page:-

G-Mart Admin

[Dashboard](#)[Users](#)[Products](#)[Orders](#)[Logout](#)

Add New Product

Select Category

▼

Add Product

Activate Windows

Go to Settings to activate Windows.

*** Happy Hacking!! ***