A PROJECT ON
Grocery Store Management System

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjewadi

SUBMITTED BY:
Shweta Nagnath Chalwade.

UNDER THE GUIDENCE OF:
Mrs. Pooja Jaiswal
Faculty Member
Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Shweta Nagnath Chalwade
0225 PG-DAC
SIIT Pune

CERTIFICATE

This is to certify that the project work under the title 'Grocery Store Management System' is done by Shweta Nagnath Chalwade in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mrs. Pooja Jaiswal
Project Guide

Mr. Yogesh Kolhe
Course Co-Coordinator

Date: 11/08/2025

1. INTRODUCTION TO PROJECT

The Grocery Store Management System is a web-based application designed to streamline the process of buying grocery products online. It provides a convenient shopping experience for customers and a robust management platform for the store administrator. By integrating product management, order tracking, checkout, and customer interaction into a single platform, the system enables customers to browse and purchase products from the comfort of their homes.

The application supports two main roles: **Admin** and **Customer**. The Admin can add and update products, monitor stock levels, manage order statuses, and access insights like total customers, sales, profit, and traffic. The admin dashboard also allows tracking of all orders—pending, shipped, or delivered—ensuring smooth fulfillment.

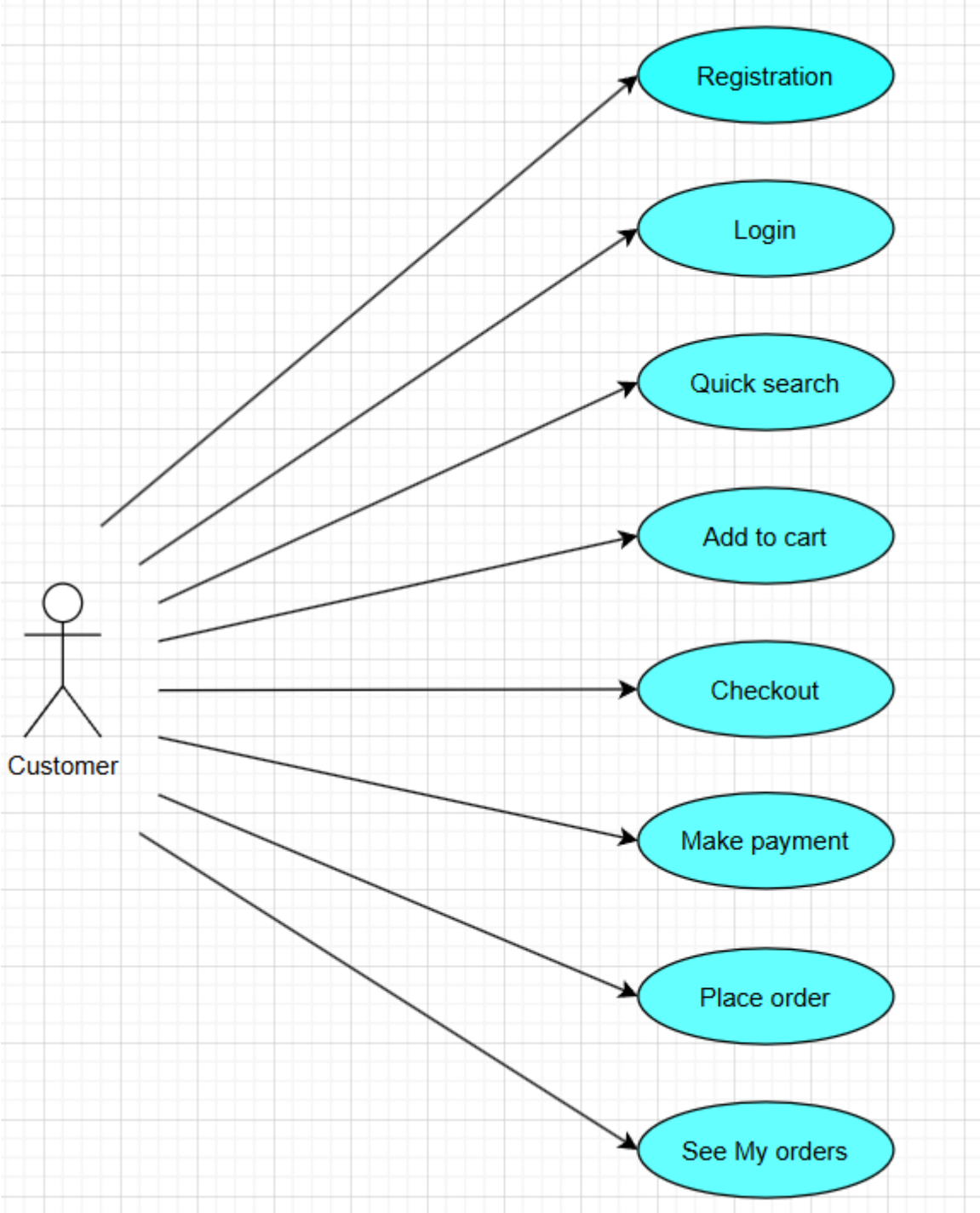
For customers, the platform offers a user-friendly interface to browse products, search, filter by category, and add items to a shopping cart. While basic browsing is available without login, purchasing requires account registration for secure transactions. The **Checkout page** collects user details for order delivery. Payment options include online payment and cash on delivery. Customers can view order history, track status in real time, and manage their account through the profile section.

Additional features include an **About Us** page to provide store information and a **Contact Us** page for customer inquiries, enhancing transparency and trust. The system ensures real-time updates between the front end and database, with security measures to protect sensitive data and scalability to handle a growing customer base.

Overall, the system bridges the gap between traditional grocery shopping and modern e-commerce, benefiting both administrators and customers through efficiency, security, and convenience.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS



2.1 User Account

The customer, who will henceforth be called the *'user'*, will be presented with options by the grocery store system at the first stage of interaction. A user's choice will depend on whether they are a guest or a registered customer and whether they want to simply browse products or proceed to purchase them. The terms *'registered user'* and *'guest'* are defined below.

A user who has shopped from the store earlier and created an account would have been given a **user ID** and a **password**. This *personal information* will henceforth be referred to as the *profile*. Such a user with a profile in the database will be called a *registered user*. A registered user will be able to browse products, add them to the cart, and place an order by logging into the system.

A new user, on the other hand, would either:

a) **Register** with the system by providing personal information, or

b) **Continue as a guest** without registration.

In case of **(a)**, the new user becomes a registered user.

In case of **(b)**, the new user remains a guest.

A guest can browse products and view their details but cannot add items to the cart for checkout or place an order. A registered user, however, can also act as a guest if they only wish to browse without logging in.

The term *'viewing products'* refers to browsing the available grocery items, their prices, descriptions, categories, and any active discount offers.

2.2 Registration and creation of user profile

The system shall require a user to register in order to carry out any purchase transactions, such as adding products to the cart and placing orders. Registration will require the user to provide the following information at a minimum – **user ID**, **password**, **full name**, **address**, **phone number**, and **email address**.
The system will store this information securely in the database as the user's *profile*. This profile will enable personalized services, such as saving the delivery address, maintaining order history, and tracking order statuses. The system will also store any loyalty or reward points (if applicable) in the user's profile, initializing them to zero at the time of registration.

.

2.3 Quick Search

The Quick Search feature allows any user, whether logged in or not, to instantly search for products in the grocery store catalog without going through the account creation or login process. This ensures that customers can quickly browse products, check prices, and view availability without any barriers. The primary objective of Quick Search is to enhance user convenience, encourage exploration of the product range, and help customers make faster decisions by reducing unnecessary steps in the browsing process.

Search Input and Criteria

When using Quick Search, the user simply needs to type a product name or keyword (such as "milk", "bread", or "rice") into the search bar provided on the homepage or in the navigation bar. The system will then search the product database for matching results. The search will be flexible, supporting partial matches and case-insensitive comparisons so that users can still find relevant items even if they do not enter the exact product name. Additionally, the search can be refined to show only products that are in stock, ensuring customers see items they can purchase immediately.

Result Display

The results from a Quick Search are displayed in an easy-to-read product grid or tabular format. Each product listing will include essential details such as product image, name, price, category, average star rating (based on customer feedback), and an "Add to Cart" button. For users searching without logging in, the system will still allow adding products to a temporary cart stored in the browser session. However, these items will not persist after the browser is closed unless the user logs in or registers.

Optional Filters and Sorting

While Quick Search primarily focuses on speed and simplicity, optional filtering options can be provided for better user experience. These may include filtering by category, brand, price range, and ratings, along with sorting options like price (low-to-high or high-to-low), newest arrivals, or highest-rated products. This allows customers to narrow down large search results into more relevant and manageable lists.

Purpose and Benefits

The Quick Search feature significantly improves the usability of the grocery store application by reducing the time taken to locate products. It also caters to casual visitors who may want to explore the catalog before committing to an account. By providing immediate access to product details and prices, Quick Search can drive higher engagement and potentially lead to more conversions, as users can easily transition from browsing to purchasing in just a few clicks

2.4 Placing Orders / Holding Cart Items / Confirming Purchase

After the user has completed **Step 2.3 – Checking Product Availability**, the system will now ask if they wish to proceed with adding the selected items to their cart and confirm the purchase. If yes:

a) If the user is browsing as a **guest**, they will need to first **register** and become a registered user before they can proceed with the order.

b) If the user is already **logged in** as a registered user, they can directly proceed to place the order. However, if the logged-in session has expired, they will be prompted to log in again.

Once the system ensures that the user is logged in validly (according to the authentication process), it checks product stock levels to ensure all requested items are still available. If any product is out of stock, the system will notify the user and suggest alternatives.

If all items are available, the system will allow the user to either:

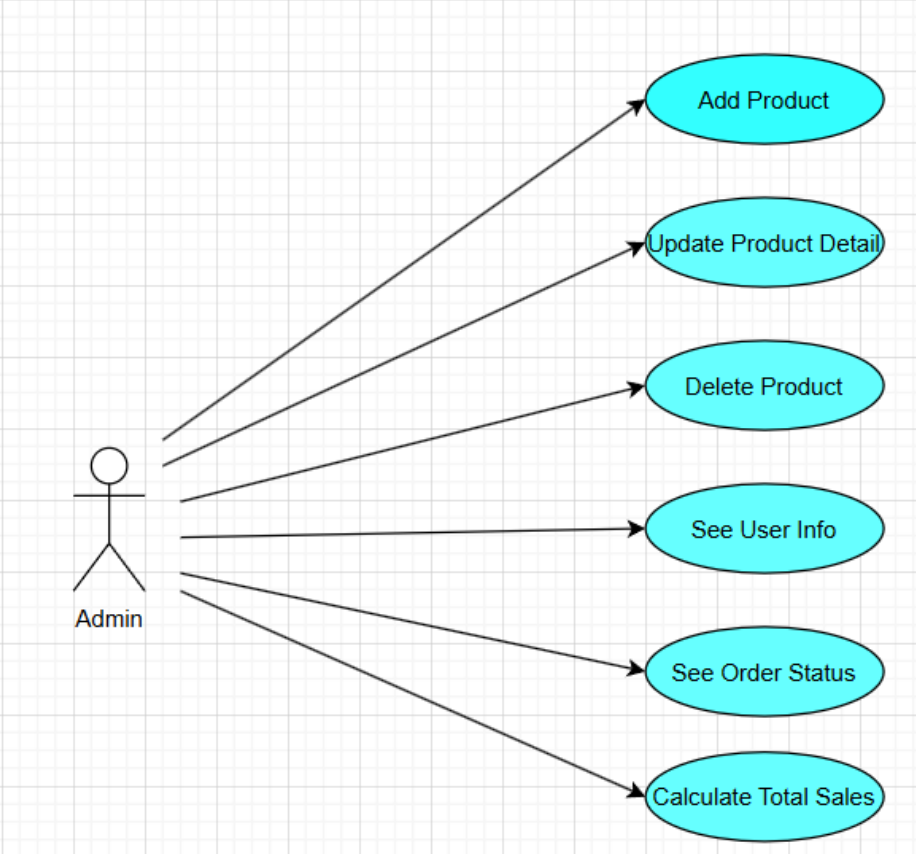
- **Hold the items in their cart** (saved for a limited period — for example, 24 hours — without payment)
- **Proceed to purchase** immediately.

When holding items in the cart, the system will inform the user that the products are not reserved indefinitely and stock availability may change. For confirmed purchases, the system will ask for **payment details** (debit card, credit card, UPI, or other available payment methods). Once the payment is successful, the system will deduct the purchased quantities from the stock in the **Products Database** and create a new entry in the **Orders Table** along with related details in the **Order Details Table**.

2.5 View Order History

The system shall allow a registered user to view all information about their **previous orders**. After logging in, the user can access the **"My Orders"** section from the navigation menu. The system will retrieve all past order details from the **Orders Table** for that specific user, including:

- Order ID
- Order Date
- Status (PENDING, SHIPPED, DELIVERED)
- Total Price
- List of Products with quantities and prices



Admin be able to login, add products , add product information, Delete product and see user Information according to user Id.

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1 : Users

Key Constraint	Column Name	Data Type	Length	Allow Null
PK	Id	BIGINT		0
	FullName	VARCHAR	255	1
	Address	VARCHAR	255	1
UQ	Email	VARCHAR	255	0
	Password	VARCHAR	255	0
	PhoneNumber	VARVHAR	20	1

Table2 : Products

Key Constraint	Column Name	Data Type	Length	Allow Null
PK	Id	BIGINT		0
	Name	VARCHAR	255	0
	Description	VARCHAR	500	1
	Price	DOUBLE		0
	Stock	INT		0
	Category	VARCHAR	255	1
	Image	VARCHAR	255	1
	CreatedAt	DATETIME		0

Table3 : Orders

Key Constraint	Column Name	Data Type	Length	Allow Null
PK	OrderId	BIGINT		0
	OrderDate	DATE		0
	Status	VARCHAR	50	0
	TotalPrice	DOUBLE		0
FK	User_Id	BIGINT		0

Table 4 :Order_Detail

Key Constraint	Column Name	Data Type	Length	Allow Null
PK	ID	BIGINT		0
FK	Order_Id	BIGINT		0
FK	Product_ID	BIGINT		0
	Quantity	INT	4	0
	Price	DOUBLE		0

Table 5 : Cart

Key Constraint	Column Name	Data Type	Length	Allow Null
PK	CartId	BIGINT		0
FK	User_Id	BIGINT		0
FK	Product_Id	BIGINT		0
	ProductName	VARCHAR	255	0
	Price	DOUBLE		0
	Quanity	INT		0

Table 6 : Feedback

Key Constraint	Column Name	Data Type	Length	Allow Null
PK	ID	BIGINT		0
	Feedback	VARCHAR	500	1
	UserName	VARCHAR	255	1
	Rating	INT		0
	CreatedAt	DATETIME		0
FK	Product_ID	BIGINT		0

4. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	Person, BankVault, SMSMessage, Dept	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	getDetails, updateStore	Methods should use **verbs** or verb phrases.
Parameter	Camel	personName, bankCode	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	Disposable	Do not use the '_' sign
Property	Pascal	ForeColor, BackColor	Use a noun or noun phrase to name properties.
Associated private member variable	_camelCase	_foreColor, _backColor	Use underscore camel casing for the private member variables
Exception Class	Pascal with "Exception" suffix	WebException,	

5. TEST REPORT

GENERAL TESTING:

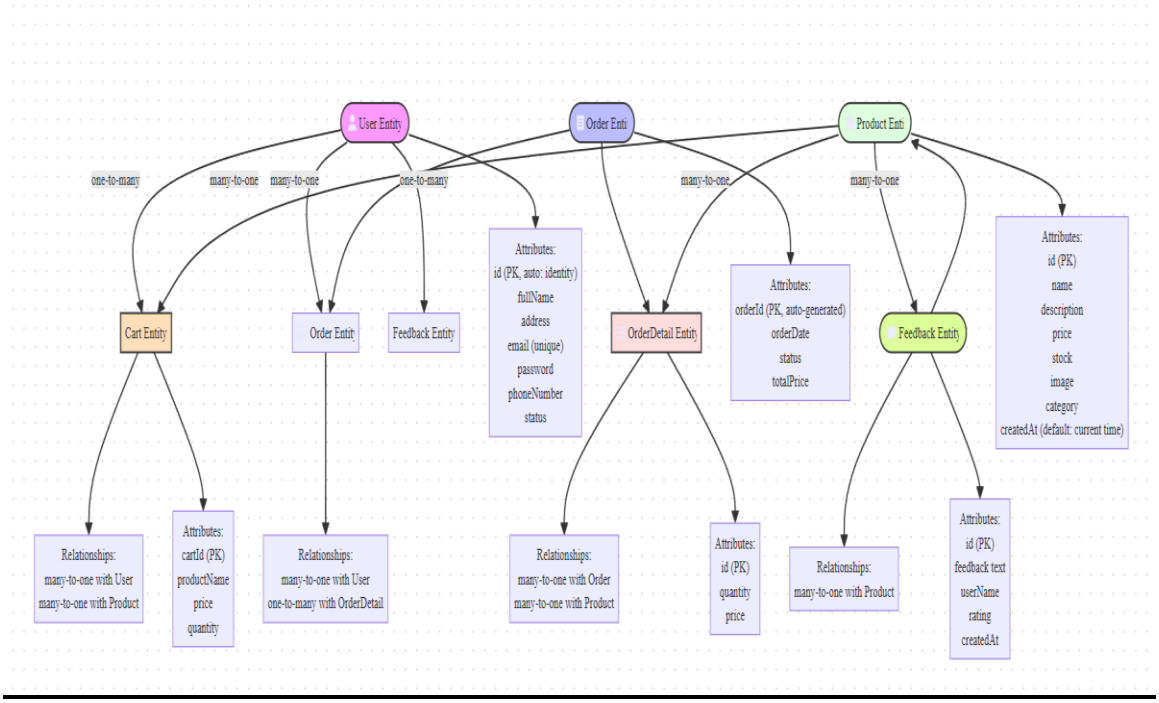
SR NO	TEST-CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	Register page	Redirected to Login page	OK	Nothing
2	Login page	Redirected to Home page	Ok	Nothing
3	Quick search	Search available products	Ok	Nothing
4	Product Detail Page	Gives Product Detail and Add to Cart button	Ok	Nothing
5	Cart page	Gives products added into the cart and place order button	Ok	Nothing
6	Checkout page	Customer enters details which is require to deliver the order	Ok	Nothing
7	Payment page	Gives Qr code to scan or On Delivery button	Ok	Nothing
8	Thank You page	Gives confirmation about order is placed	Ok	Nothing
9	MyOrder page	Gives detail about ordered products	Ok	Nothing
10	User Detail	Gives user information	Ok	Nothing
11	Feedback	Allow to give feedback about ordered product	Ok	Nothing
12	About us	Gives the info about company policies	Ok	Nothing
13	Contact us	Send any queries about delivered product	Ok	Nothing
14	Logout	Logout the customer	Ok	Nothing

6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	*SLC Phase*	Additional Notes
JULY 11, 2025	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Nitinkudale (CEO, SIIT Pune) to know his requirements.
JULY 21, 2025	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
JULY 22, 2025	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
JULY 23, 2025	Business Logic Component design Started	Design Phase	----------
JULY 24, 2025	Coding Phase Started	Coding Phase	70% of Class Library implemented.
JULY 25, 2025	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
JULY 27, 2025	Off	Off	Off
JULY 28, 2025	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
JULY 29, 2025	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--

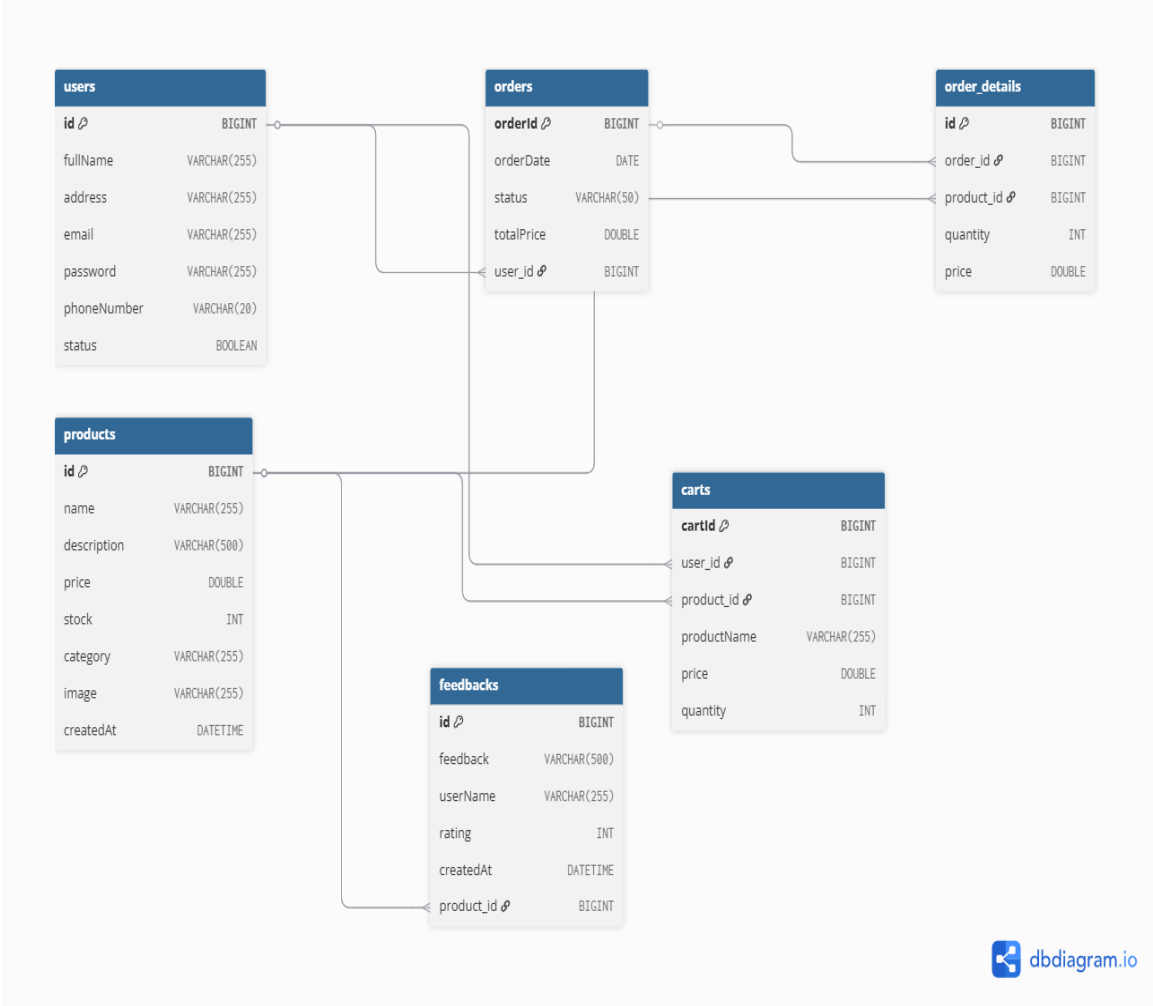
JULY 29 – AUG 03, 2025	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
AUG 04 - 05, 2025	The Project was Tested by the respective Team Leaders and the Project Manager	Testing Phase (Module Testing)	--
AUG 06 - 08, 2025	The Project was Submitted to Other Project Leader of Other Project Group For Testing	Testing Phase (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
AUG 09 - 10, 2025	The Errors Found were Removed	Debugging	The Project was complete for submission
AUG 11, 2025	Final Submission of Project		

Entity Relationship Diagram



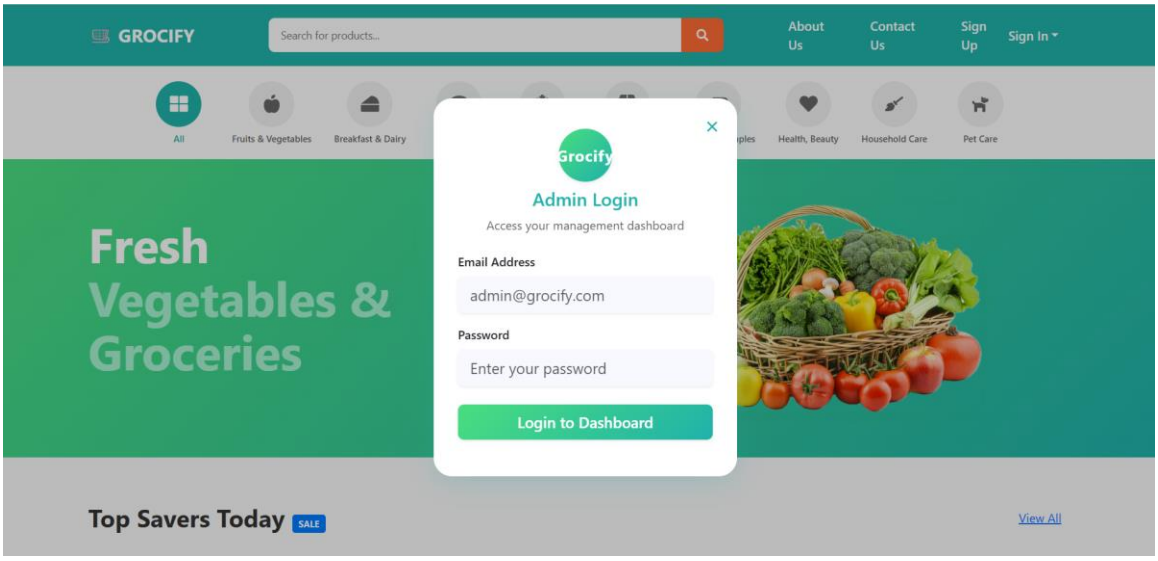
Data Flow Diagram:

Class Diagram

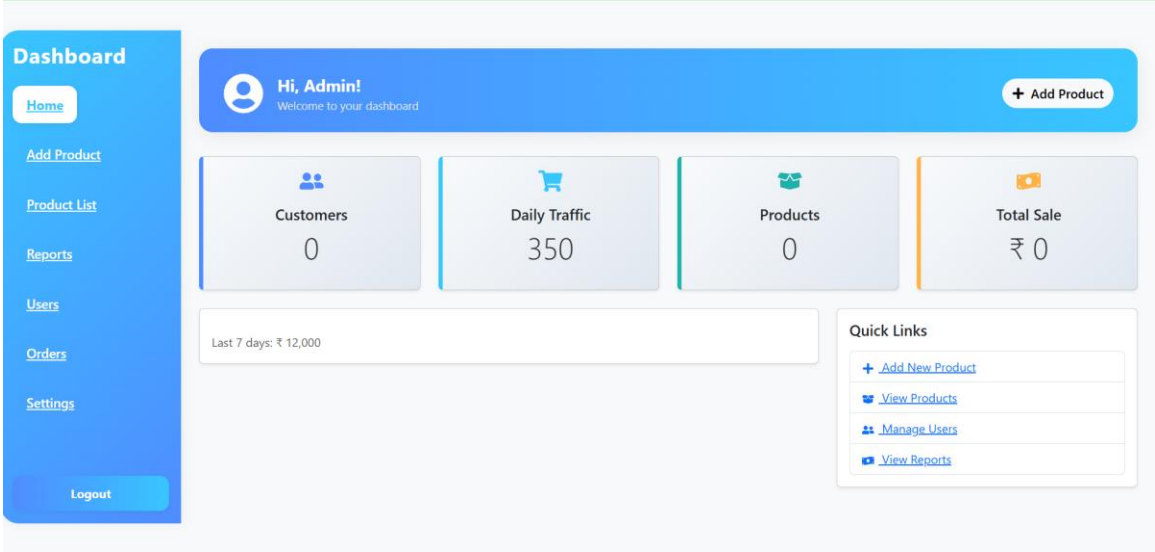


Welcome:

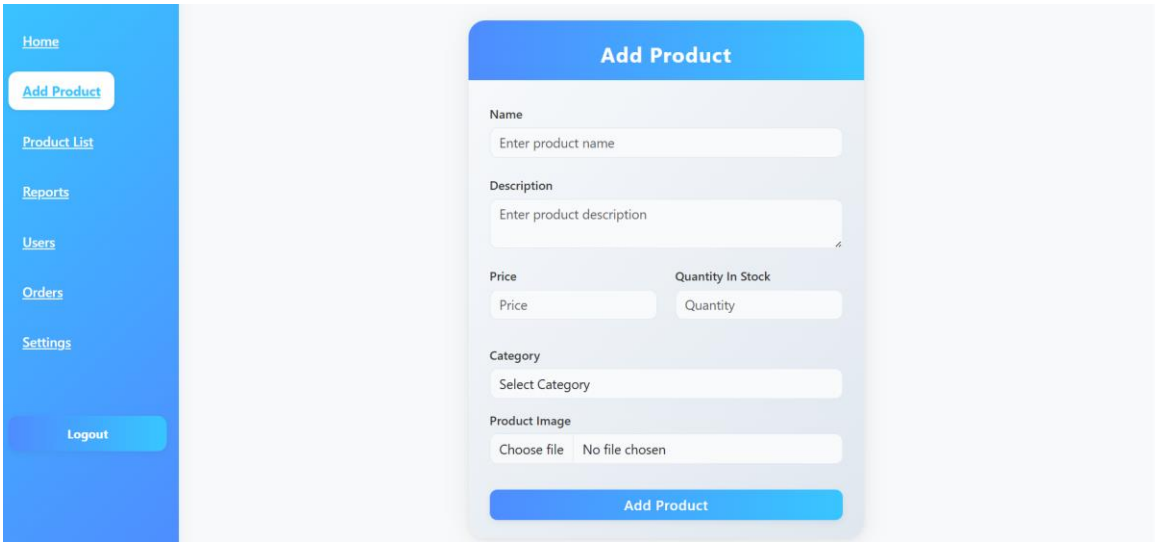
Admin Login:



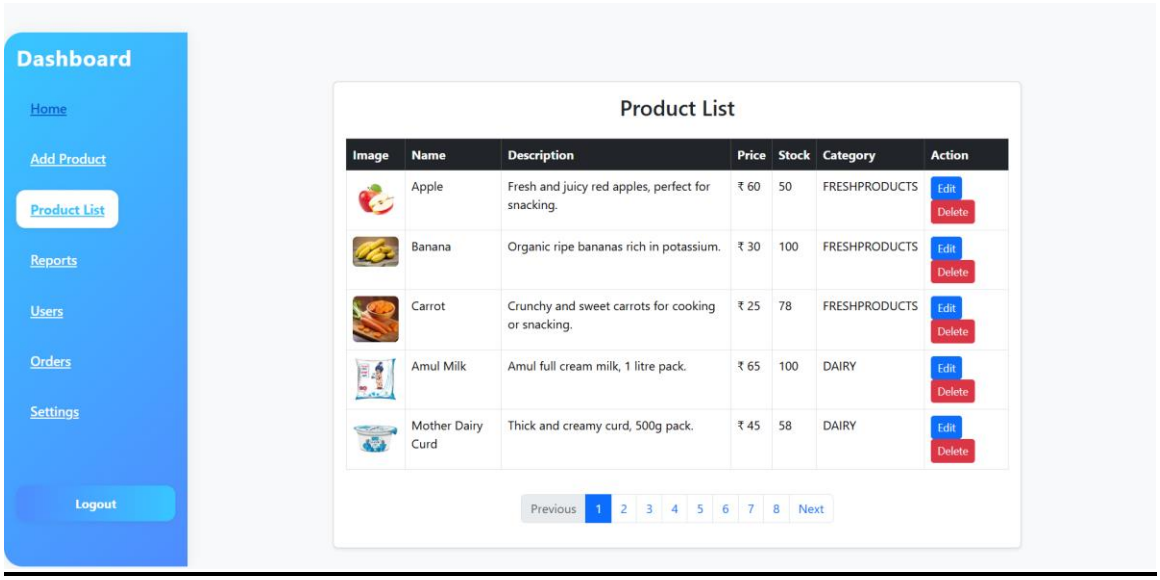
Admin Homepage:



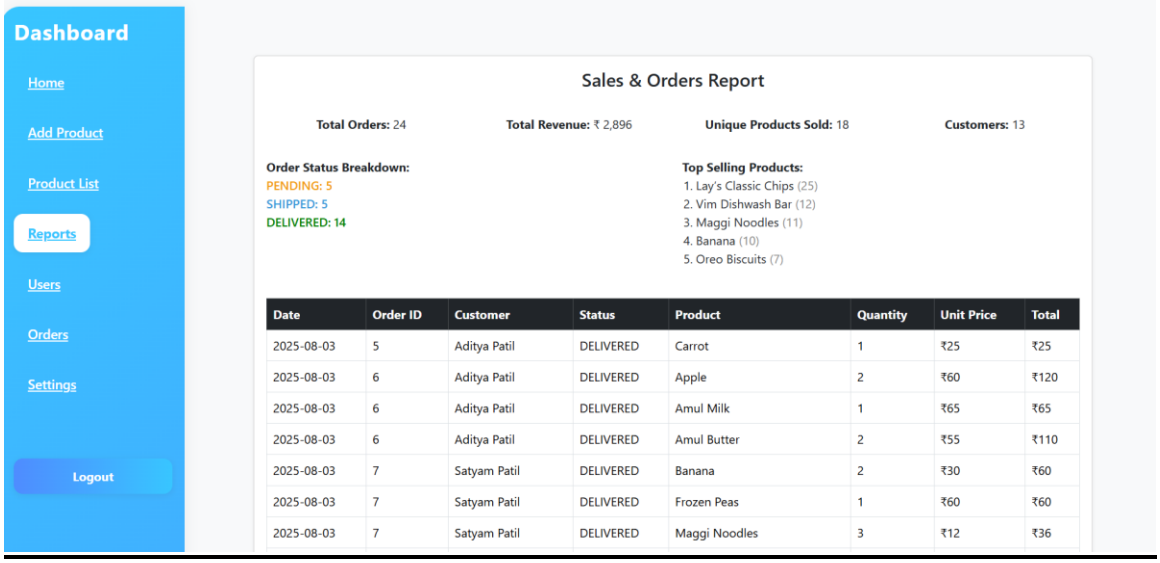
Add Product:



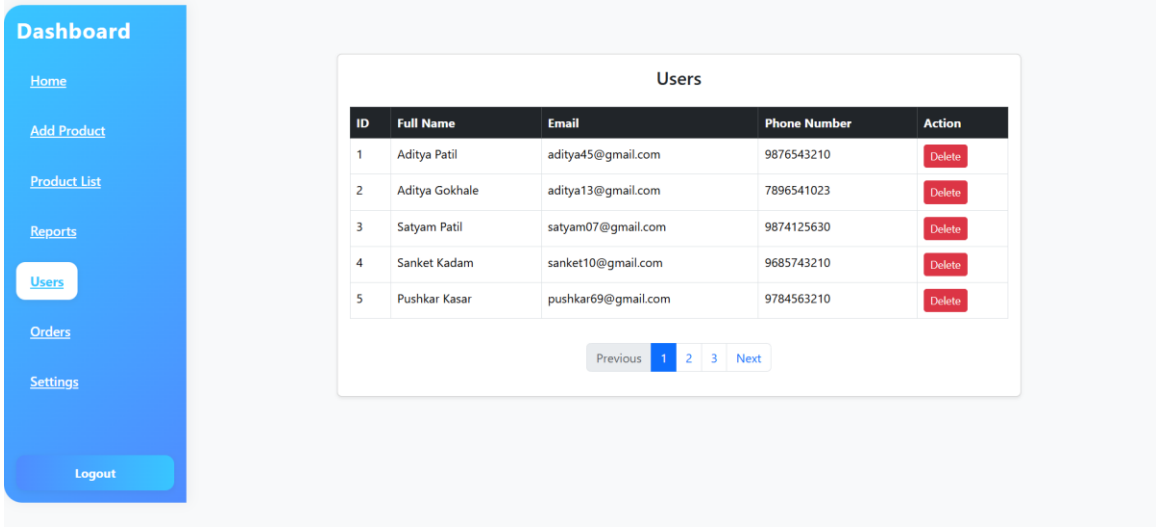
Product List:



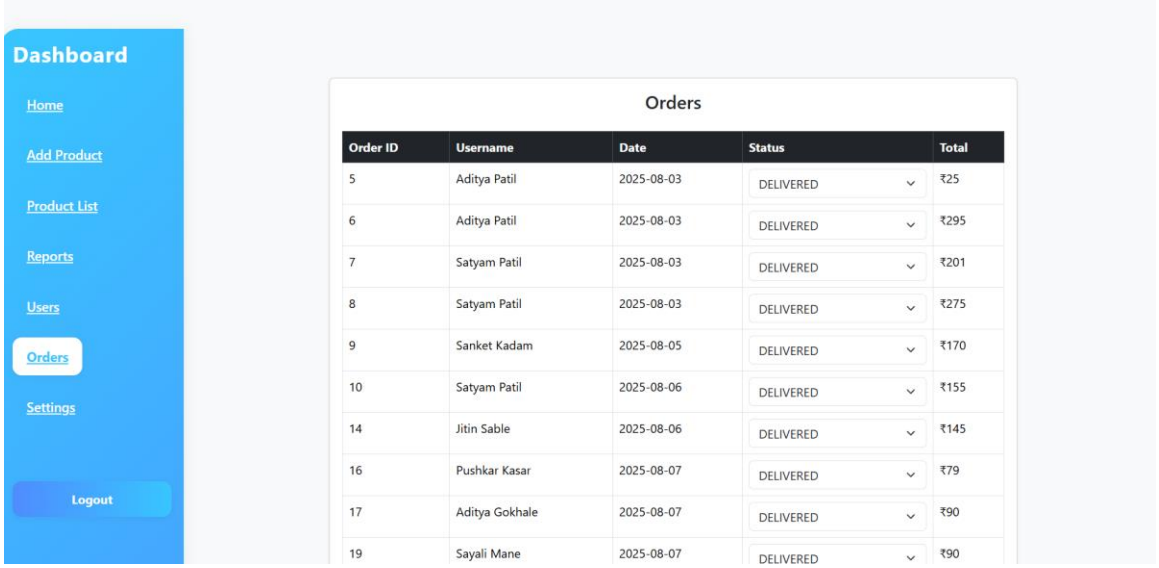
Reports:



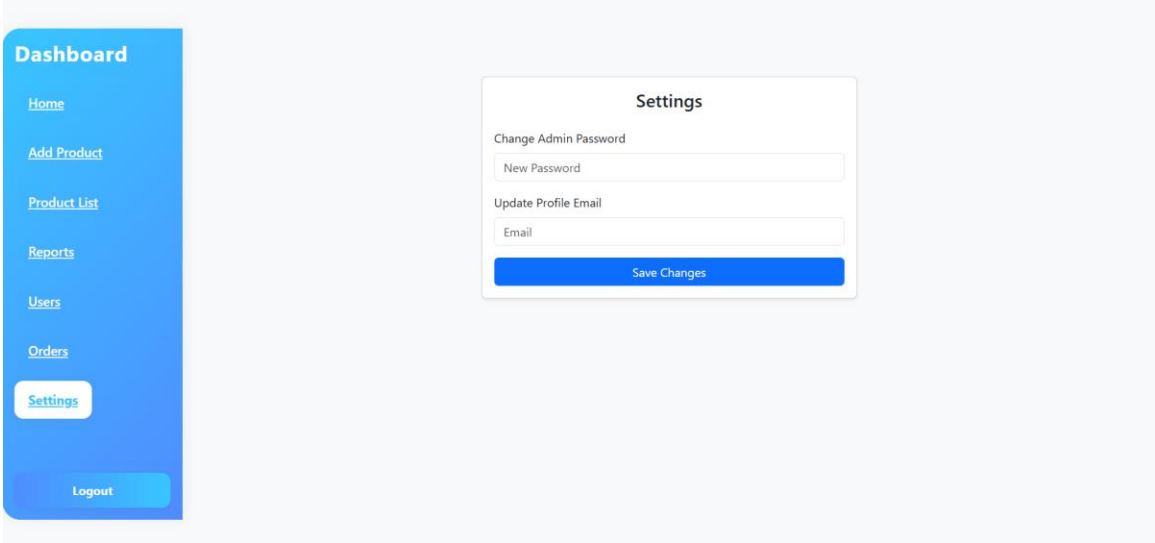
Users:



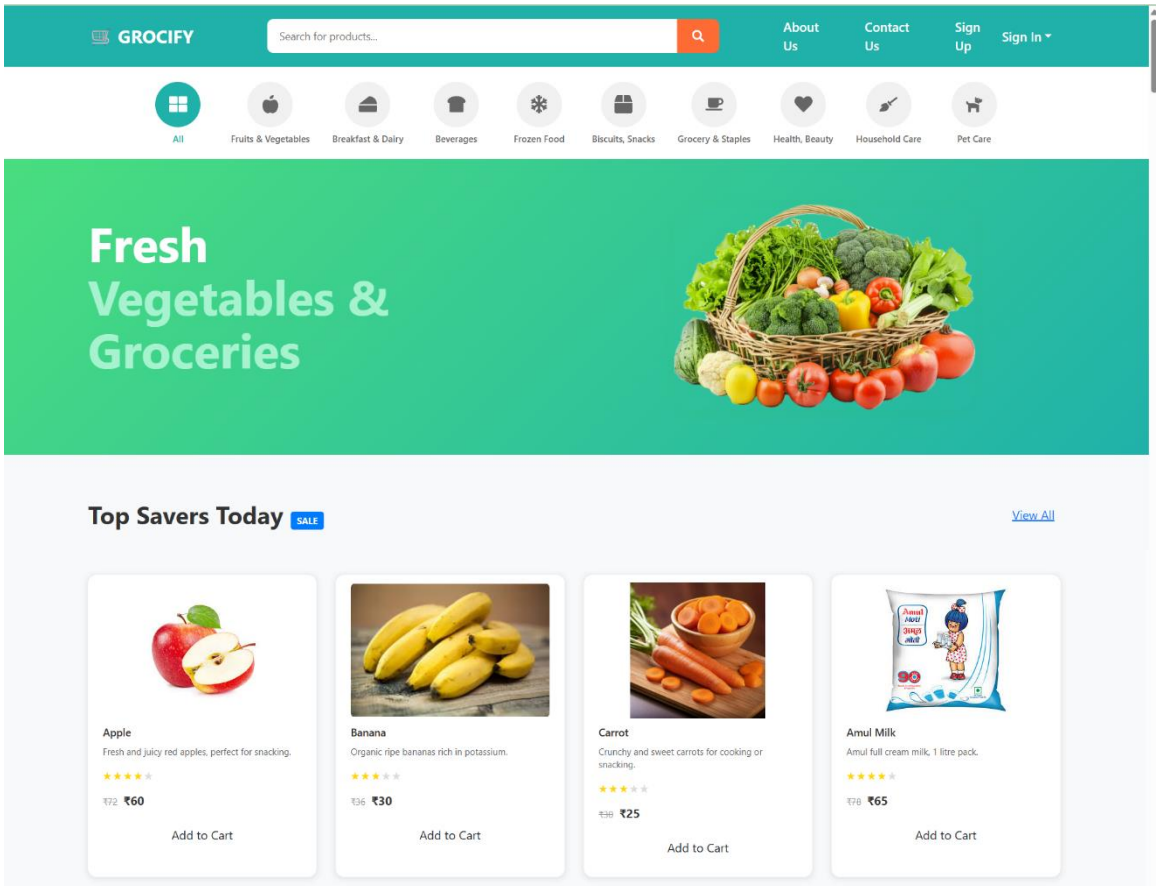
Orders:



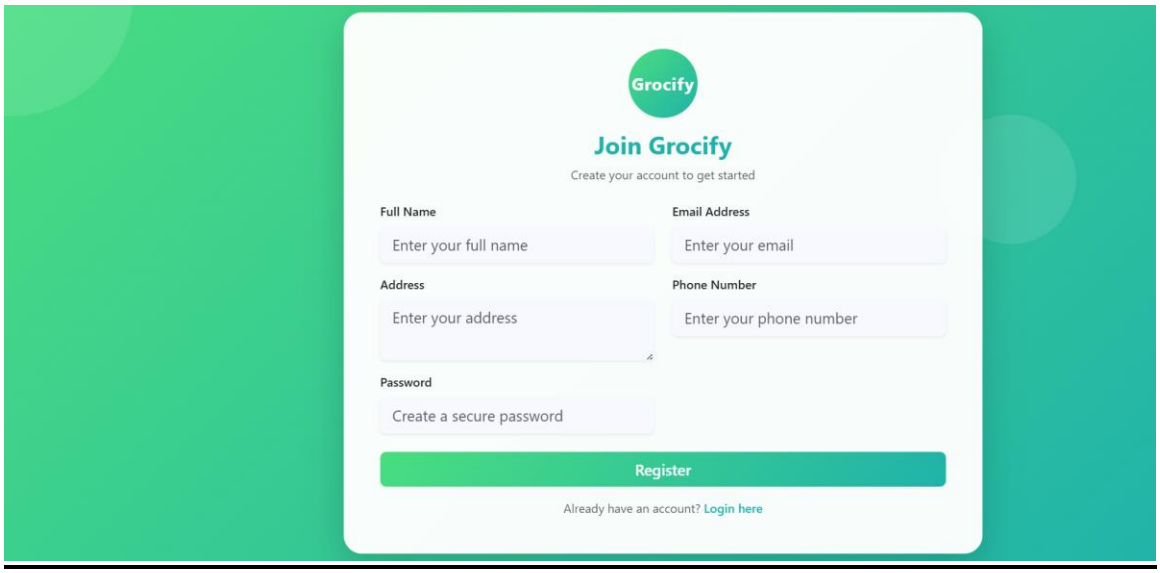
Setting:


Dashboard
Home
Add Product
Product List
Reports
Users
Orders
Settings
Logout
Settings
Change Admin Password
New Password
Update Profile Email
Email
Save Changes

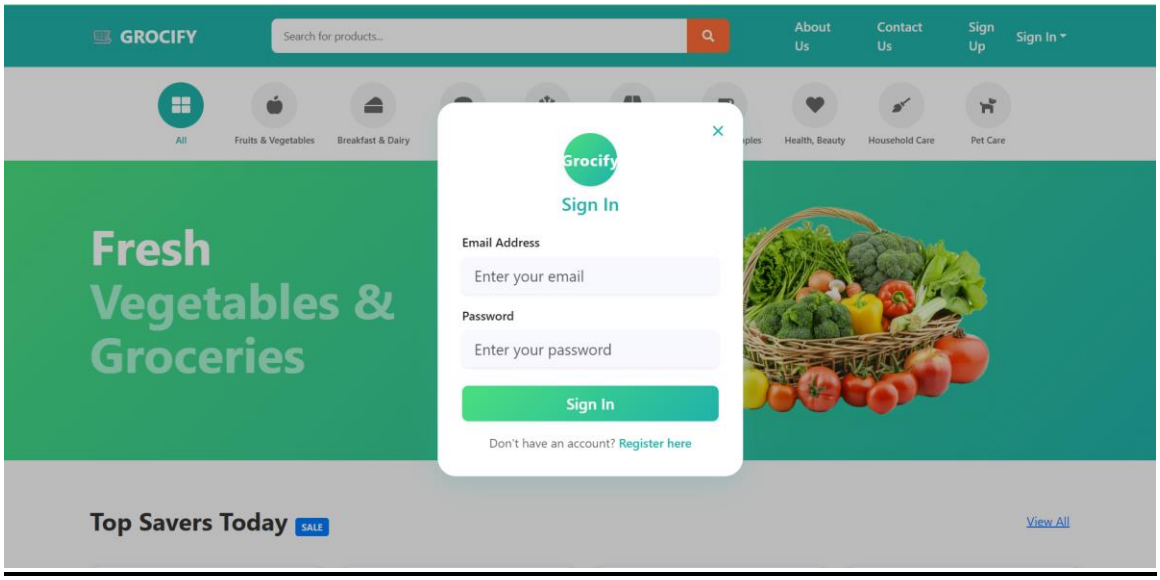
User Homepage:



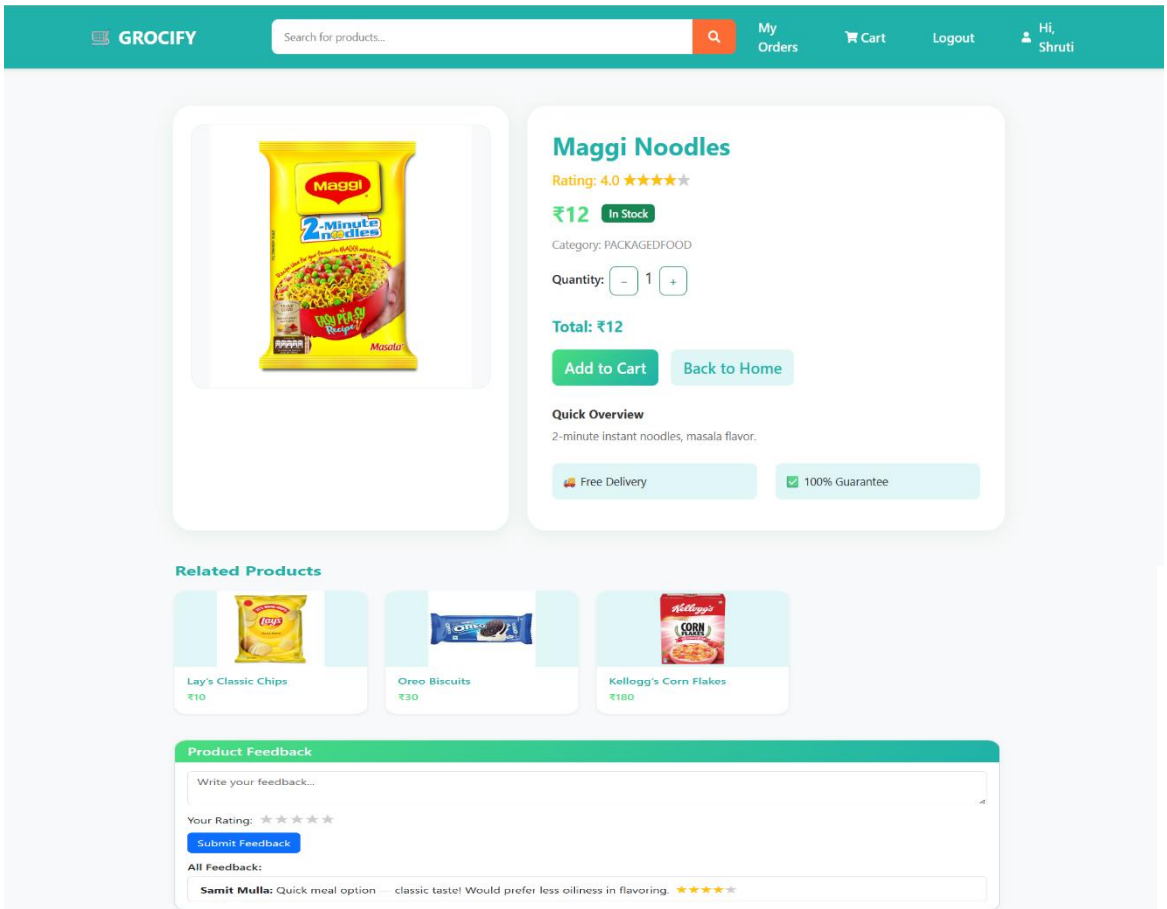
User Registration:



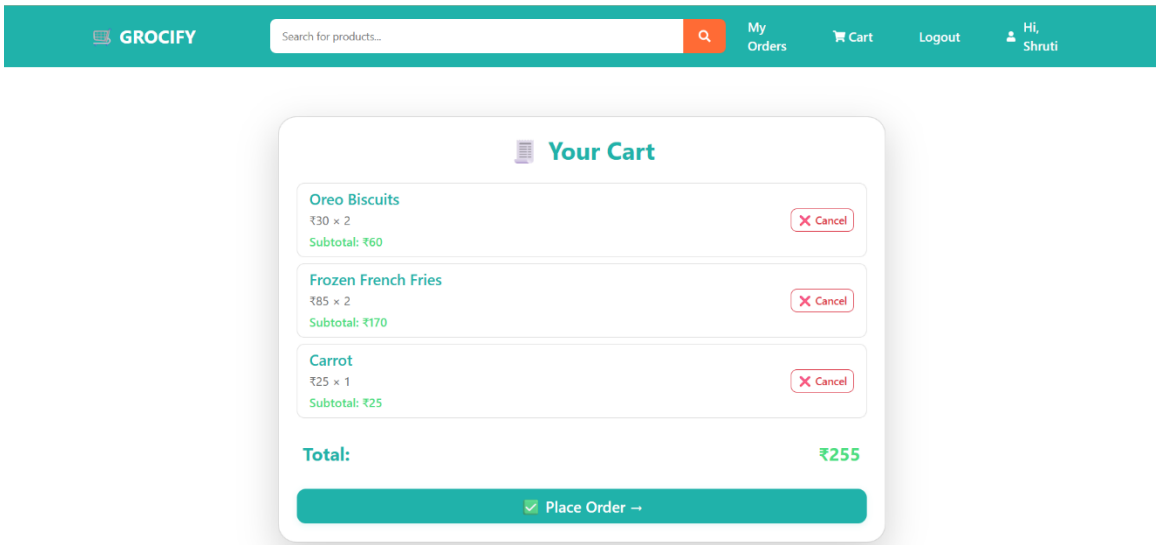
User Login:



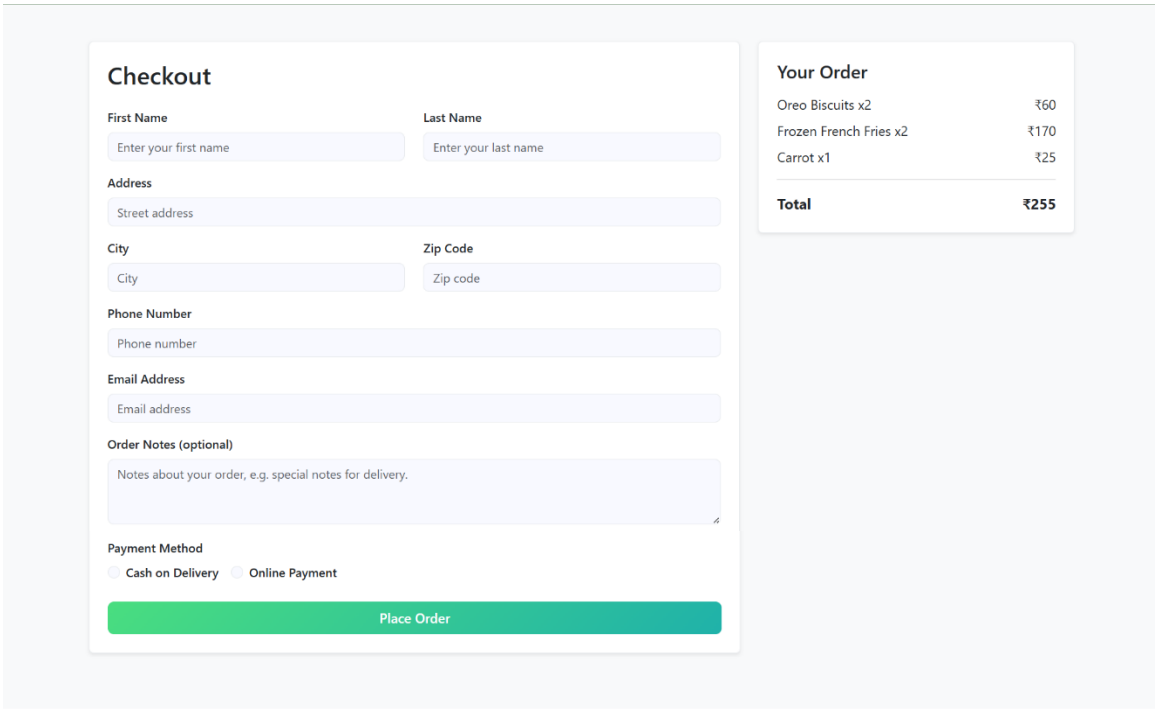
Product Detail and Feedback:



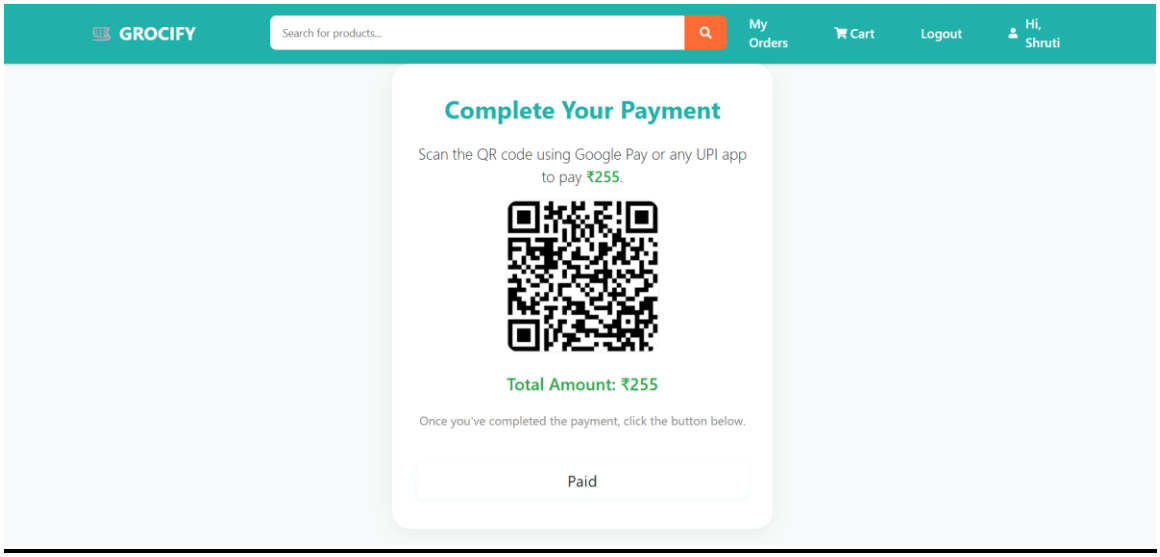
Cart:



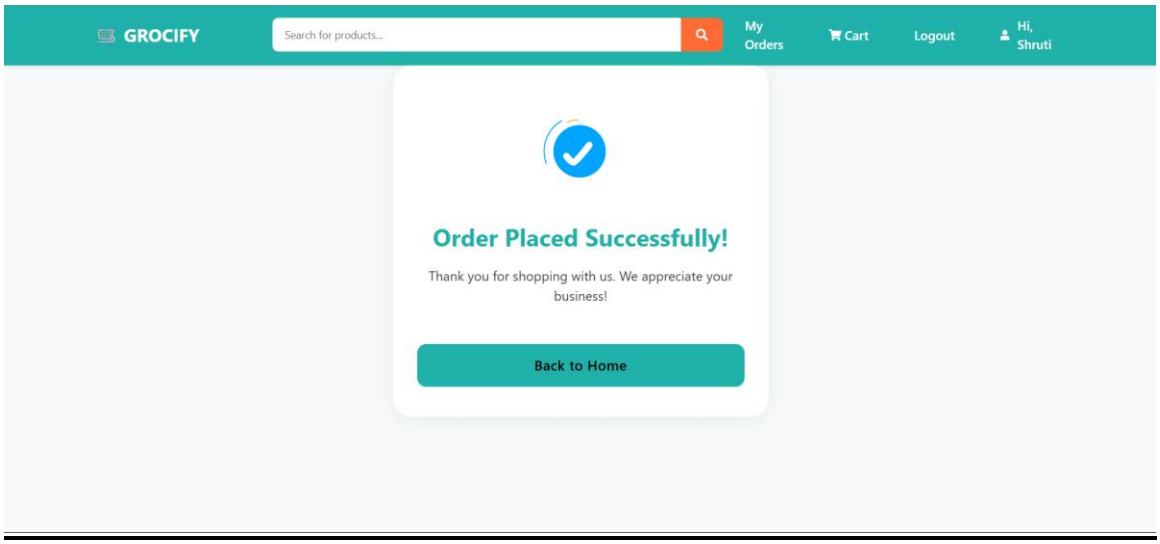
Checkout:



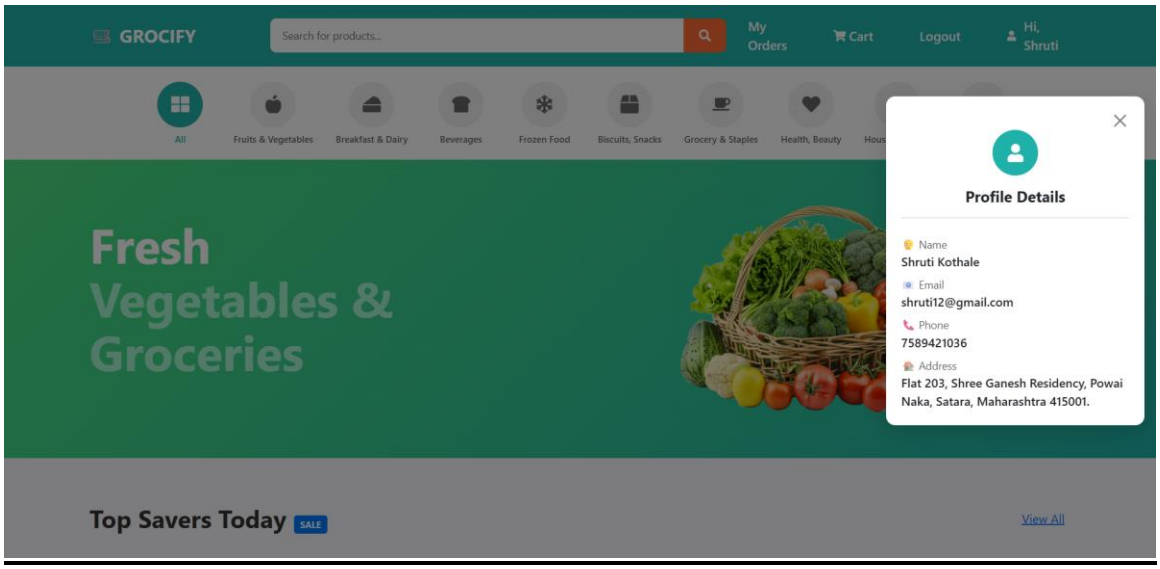
Payment:



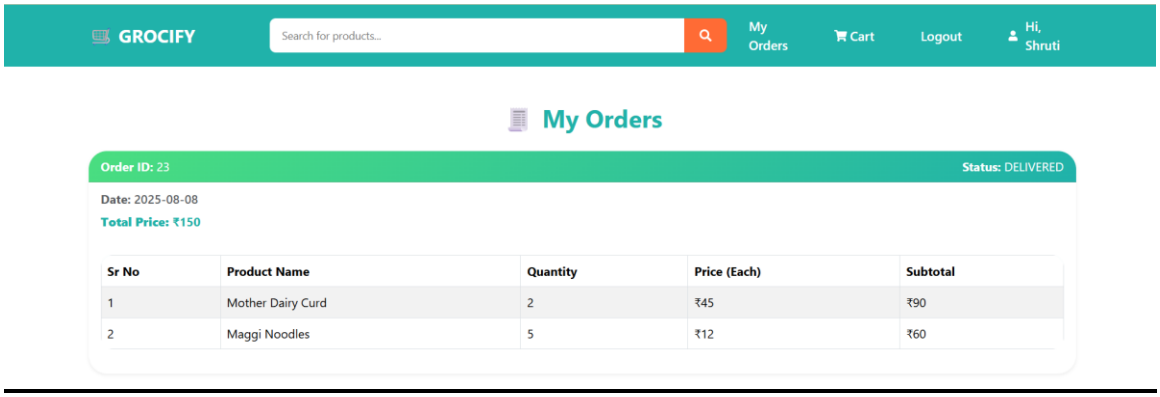
Thank You:



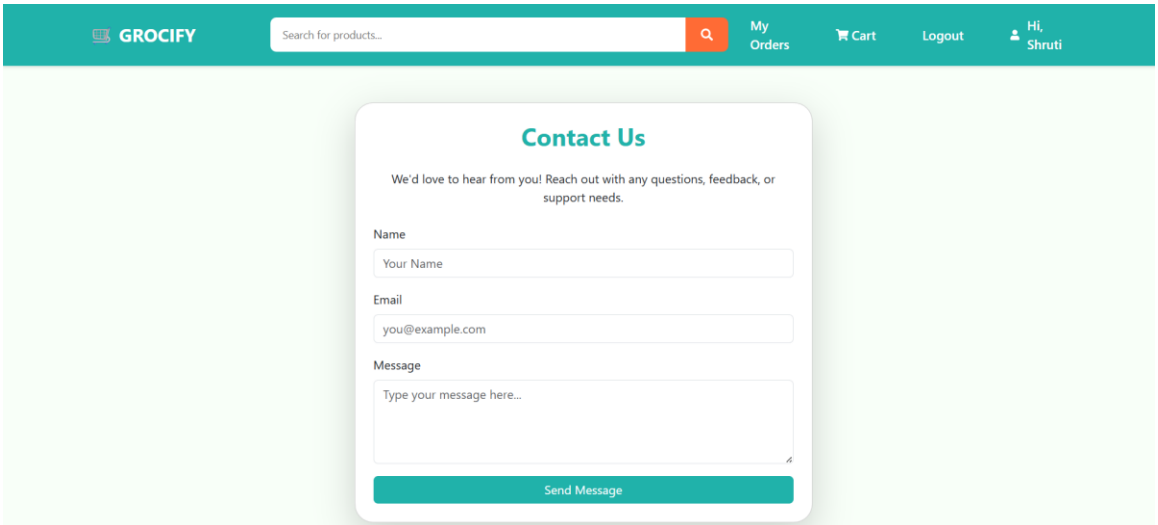
User Details Popup:



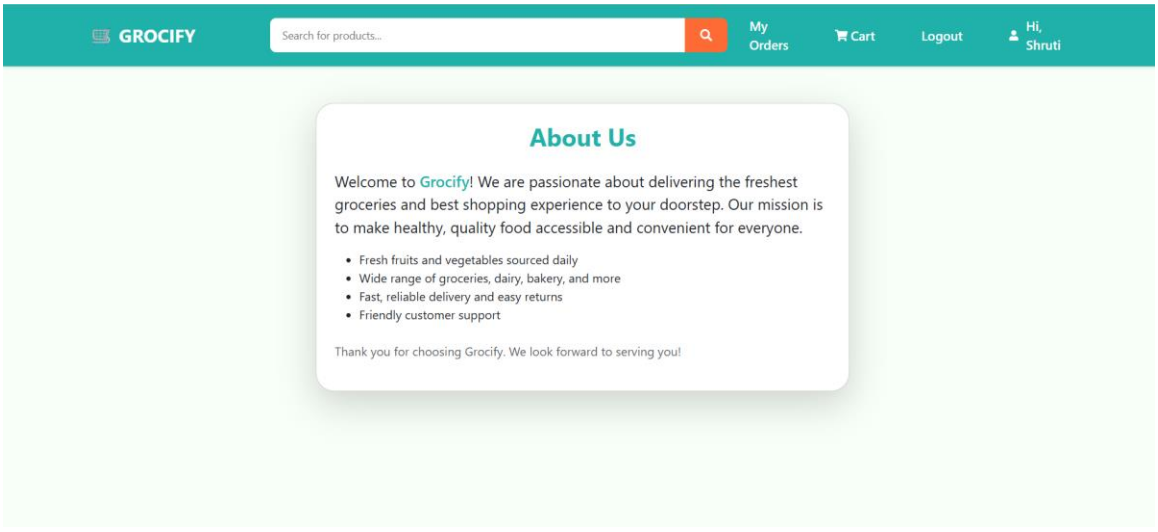
My Orders:



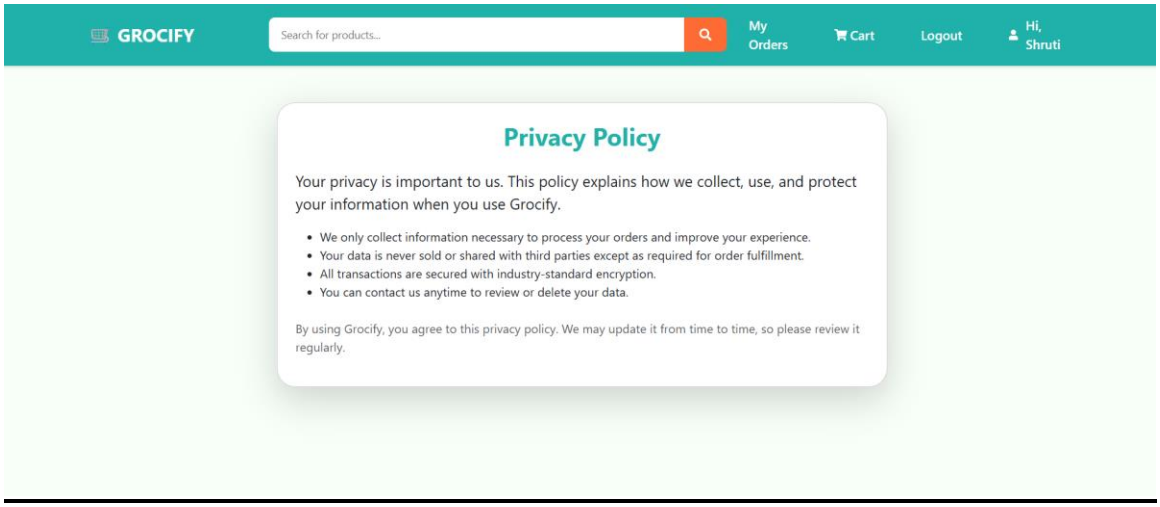
Contact Us:



About Us:



Privacy Policy:



Footer:

7.REFERENCES:

https://amazon.com

https://www.flipkart.com/

https://www.bigbasket.com/

https://www.slideshare.net/slideshow/grocery-store-management-system269437782

https://github.com/theakhinabraham/grocer-grocery-store-manager

https://www.geeksforgeeks.org/javascript/design-a-online-grocery-website-in-html-css-javascript/

https://www.pragma.co/blog/how-to-build-an-app-for-grocery-shopping