1. INTRODUCTION TO PROJECT GR-Mart

This project is a web-based grocery shopping system for an existing shop. An Online Shopping Management System where the Admin can Add, Update and Delete the Categories and Products. The Products are divided into various categories like Grains, Oils and solids , Biscuits , Vegetables etc. A user can select a particular item to view the details, choose the number of items and fill in details like Name, Address etc. to buy a product the project objective is to deliver the online grocery shopping application into web platform. Online Grocery shopping is the process whereby consumers directly buy goods or services from a seller in real- time, without an intermediary service, over the Internet. It is a form of electronic commerce.

This project is attempts provide the advantages of online Grocery shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using any android device. Thus, the customer will get the service of online shopping and home delivery from his favorite shop. The objective of the project is to make a web application to enable any thin client to purchase items in an existing shop. In order to build such an application complete web support, need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an application with web view.

Features:-

1. Products Available-Biscuits, Vegetables, Juices
2. Search for groceries (Rice, Lintels) products easily
3. Category of Products- Grains, Bakery, Vegetables, Biscuits
4. Cart feature
5. Date and time of product delivery will be notified by the system
6. The admin can add/delete Suppliers and delivery Persons.
7. Allows the customers to maintain cart

PROJECT OBJECTIVE

The objective of the project is to make an web application to purchase items in an existing shop. In order to build such an application complete web support, need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

PROJECT OVERVIEW

The central concept of the application is to allow the customer to shop virtually using the internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an RDBMS at the server side (store). The server processes the customers, and the items are shipped to the address submitted by them.

The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of this product is a departmental store where the application is hosted on the web and the administrator maintains the database. The application, which is deploy the customer database, the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personal feed the relevant data into the system, several reports could be generated as per the security.

PROJECT SCOPE

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24X7 and a home delivery system which can make customers happy. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as BlinkIt or BigBasket. Since the application is always available.

2. SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified, and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

2.1 EXISTING SYSTEM

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item.

- It is less user-friendly.
- User must go to shop and select products.
- It is difficult to identify the required product.
- Description of the product limited.
- It is a time-consuming process
- Not in reach of distant users.

2.2 PROPOSED SYSTEM

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the application in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

2.3 SYSTEM REQUIREMENT SPECIFICATION

2.3.1 GENERAL DESCRIPTION

Product Description :

The system consists of two parts. A web application which can provide the online shopping service for the customer to access the web service from his Smartphone/System. Web application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

Problem Statement :

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands, if shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers

2.3.2 SYSTEM OBJECTIVES

➢ To provide a Web application for online shopping of products in an existing shop.

➢ To provide an online shopping web site for the same shop.

2.3.3 SYSTEM REQUIREMENTS

2.REQUIREMENTS

2.3.3.2 FUNCTIONAL REQUIREMENTS

USER

➢ USER LOGIN

Description of feature This feature used by the user to login into system. A user must login with his username and password to the system after registration. If they are invalid, the user not allowed to enter the system.

Functional Requirement

• Username and password will be provided after user registration is confirmed.

• Password should be hidden from others while typing it in the field

➢ **REGISTER NEW USER**

Description of feature A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept new user by unblocking him.

Functional Requirement

• System must be able to verify and validate information

• The system must encrypt the password of the customer to provide security.

➢ **PURCHASING AN ITEM**

Description of feature The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart, the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

Functional Requirement

• System must ensure that, only a registered customer can purchase items.

• Admin account should be secured so that only owner of the shop can access that account.

MODERATOR

Description of features

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the workload of admin. Now moderator has all the privilege of an admin having except managing other moderators. He can manage users and manage products. He can also check the orders and edit his profile.

Functional Requirement

• The system must identify the login of a moderator.

ADMIN

➢ **MANAGE USER**

Description of features

The administrator can add user, delete user, view user and block user.

➢ **MANAGE MODERATOR**

Description of features

The administrator can add moderator, delete moderator, block moderator and search for a moderator.

➢ **MANAGE PRODUCTS**

Description of features

The administrator can add product, delete product, and view product

➢ **MANAGE ORDER**

Description of features

The administrator can view orders and delete orders.

Functional Requirements:

• The system must identify the login of the admin.

• Admin account should be secured so that only owner of the shop can access that account.

MODERATOR

Description of features

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the workload of admin. Now moderator has all the privilege of an admin having except managing other moderators. He can manage users and manage products. He can also check the orders and edit his profile.

Functional Requirement

• The system must identify the login of a moderator.

2.3.3.1 NON-FUNCTIONAL REQUIREMENTS

i. EFFICIENCY REQUIREMENT

When an online shopping cart android application implemented customer can purchase product in an efficient manner.

ii. RELIABILITY REQUIREMENT

The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

iii. USABILITY REQUIREMENT

The Web application is designed for user friendly environment and ease of use.

iv. IMPLEMENTATION REQUIREMENT

Implementation of the system using React in front end with Spring Boot as back end and it will be used for database connectivity. And the database part is developed by MySQL. Responsive web designing is used for making the website compatible for any type of screen.

v. DELIVERY REQUIREMENT

The whole system is expected to be delivered in four months of time with a weekly Evaluation by the project guide.

3. DESIGN

SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design. Specifications to performance specification. System design has two phases of development.

- **Logical Design**
- **Physical Design**

During logical design phase the analyst describes inputs (sources), outputs(destinations), databases (data sores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

3.1 INPUT AND OUTPUT DESIGN

3.1.1 INPUT DESIGN:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

3.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

3.2 DATABASE DESIGN

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- **Primary key** - the field that is unique for all the record occurrences
- **Foreign key** - the field used to set relation between tables Normalization is a technique to avoid redundancy in the tables.

3.3 SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

3.3.1 FRONT END:

React is a library which is developed by Facebook are utilized to implement the frontend. React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

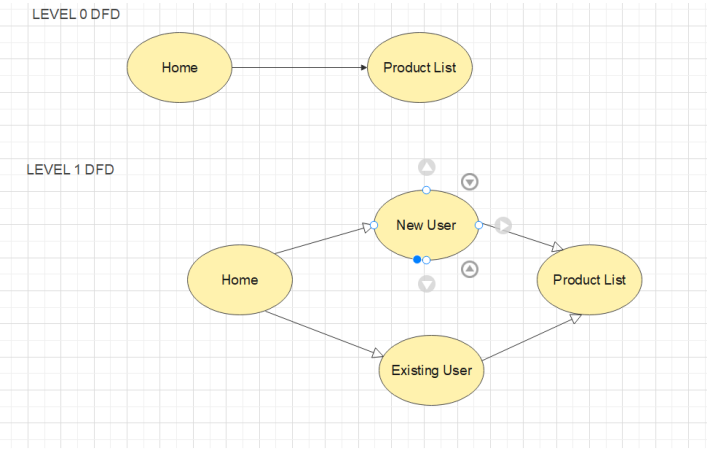
3.3.2 BACKEND:

The back end is implemented using MySQL which is used to design databases. **MySQL**: MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. An application software called Navicert was used to design the tables in MySQL.

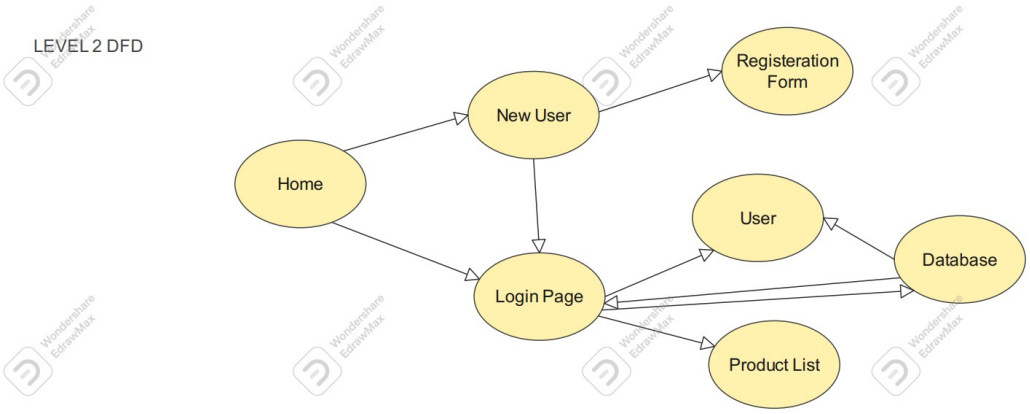
Spring-Boot:

This is used to connect MYSQL and fetch data from database and store the data in database. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.

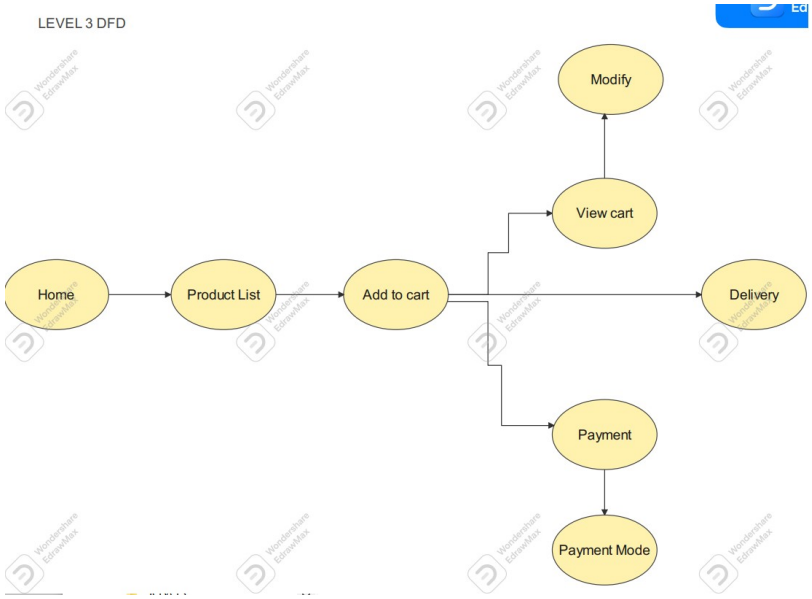
DFD Level 0 :



DFD LEVEL 2



DFD LEVEL 3



Level DFD for Customer

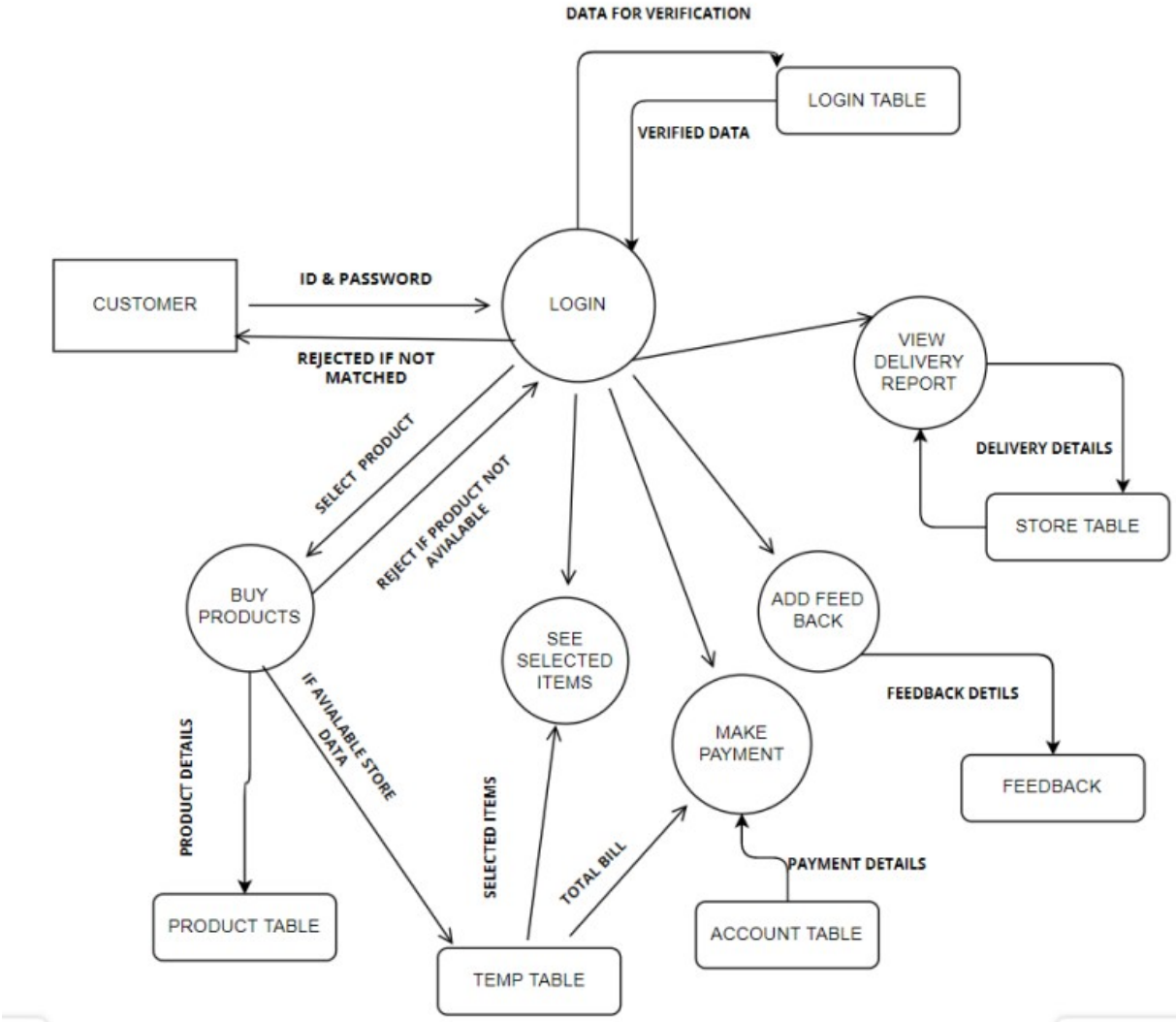
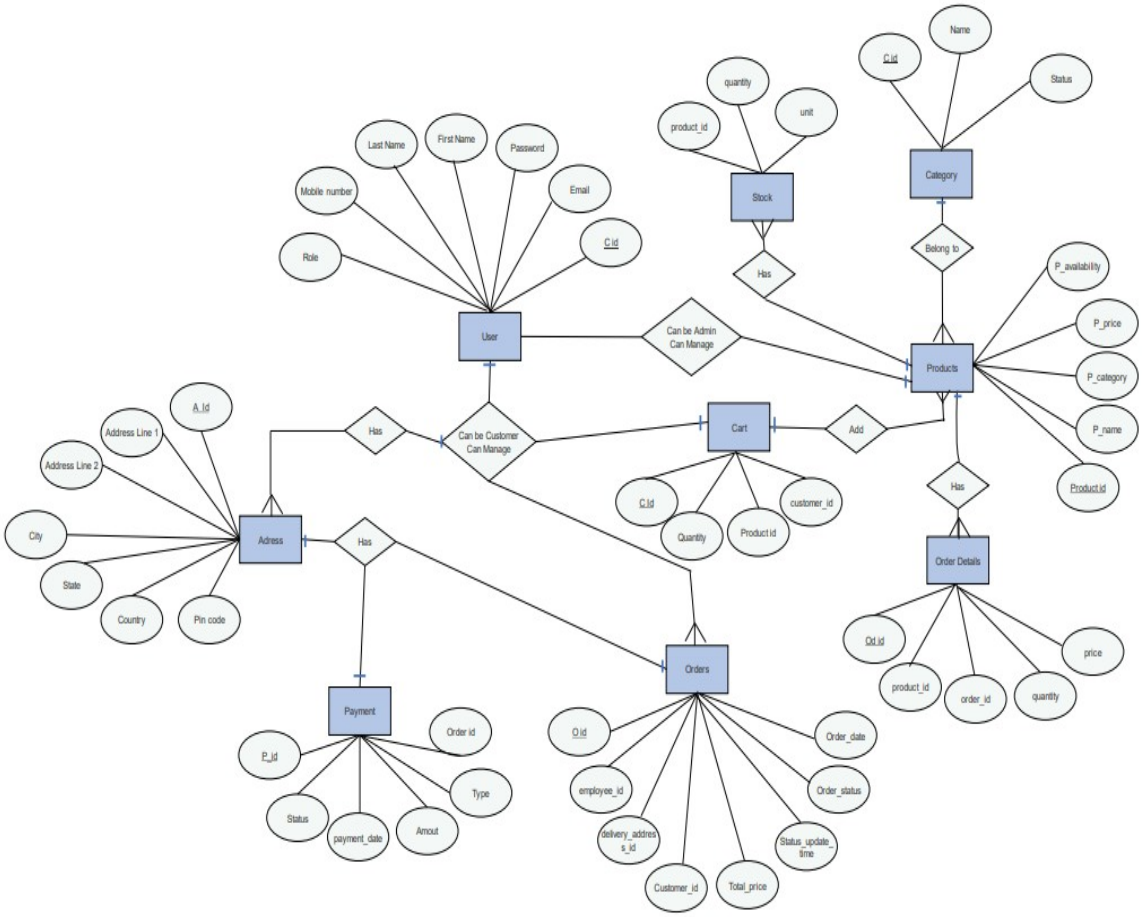
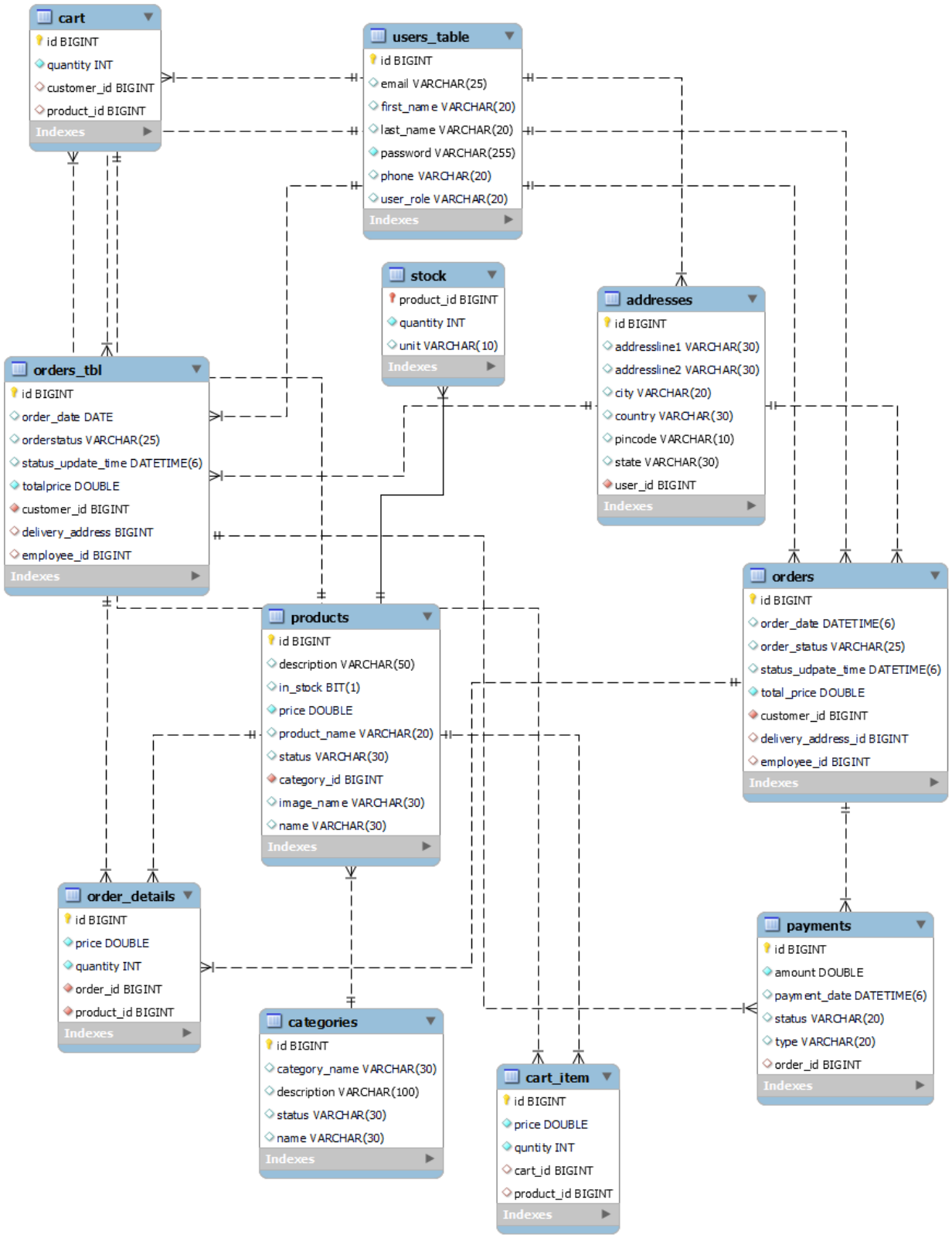


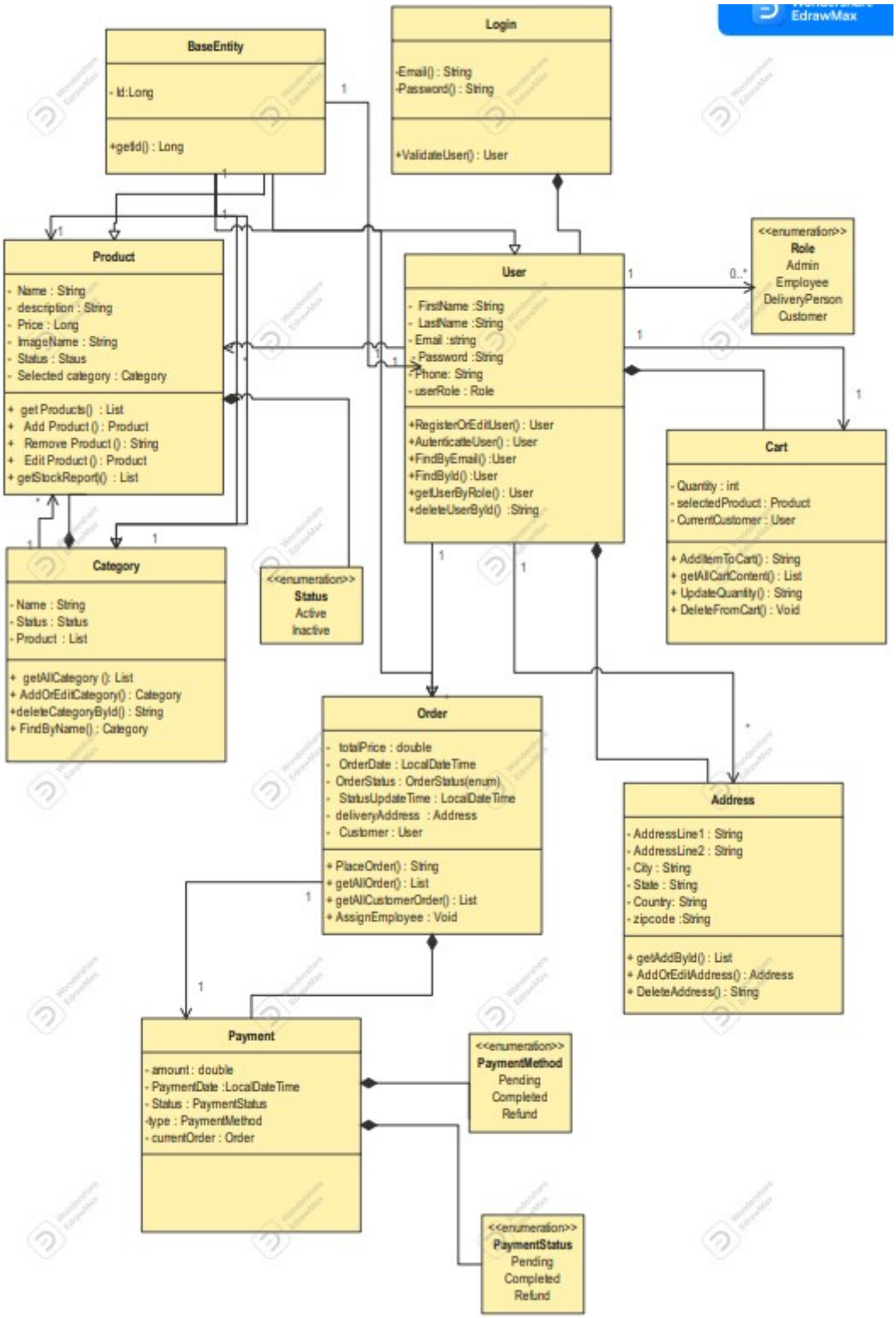
Figure 6 1 Level DFD for CUSTOMER

E – R DIAGRAM


Name
C id
Status
quantity
unit
product_id
Last Name
First Name
Password
Mobile number
Email
Role
C id
Stock
Category
Has
Belong to
P_availability
P_price
P_category
P_name
Product id
User
Can be Admin
Can Manage
Products
A_id
Address Line 1
Address Line 2
City
State
Country
Pin code
Has
Can be Customer
Can Manage
Cart
Add
C Id
Quantity
Product id
customer_id
Has
Order Details
Adress
Has
Orders
Od id
product_id
order_id
quantity
price
Payment
P_id
Order id
Status
payment_date
Amout
Type
O id
employee_id
delivery_address_id
Customer_id
Total_price
Status_update_time
Order_status
Order_date



CLASS DIAGRAM



ACTIVITY DIAGRAM

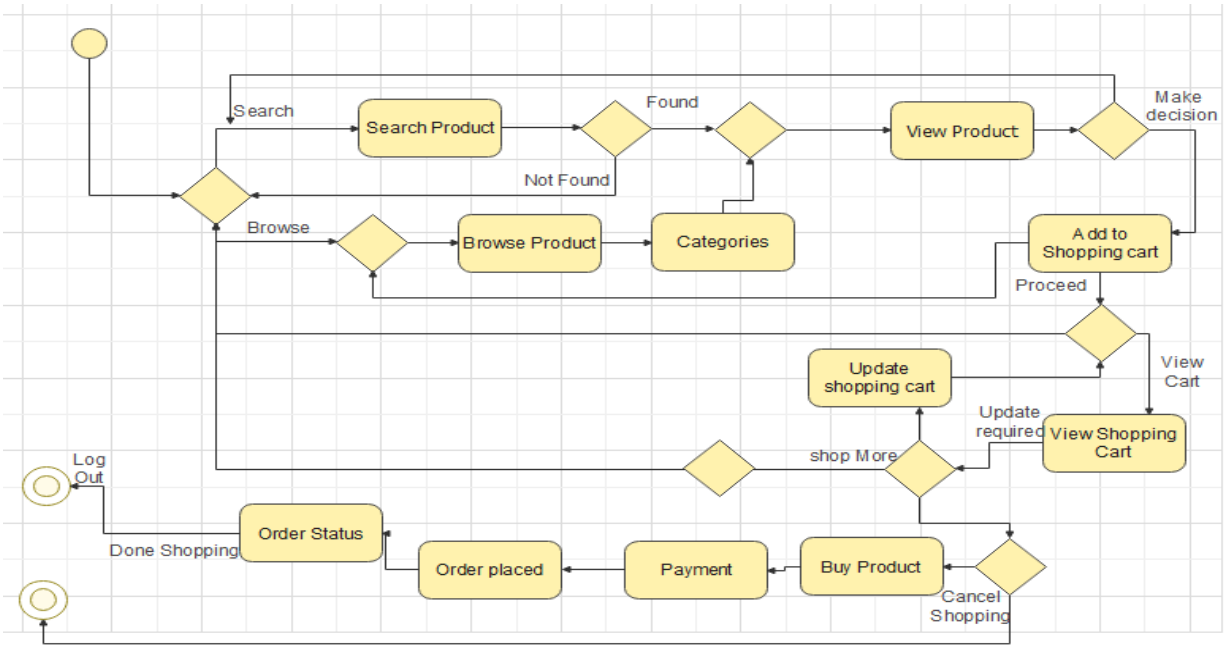


Figure : Activity Diagram

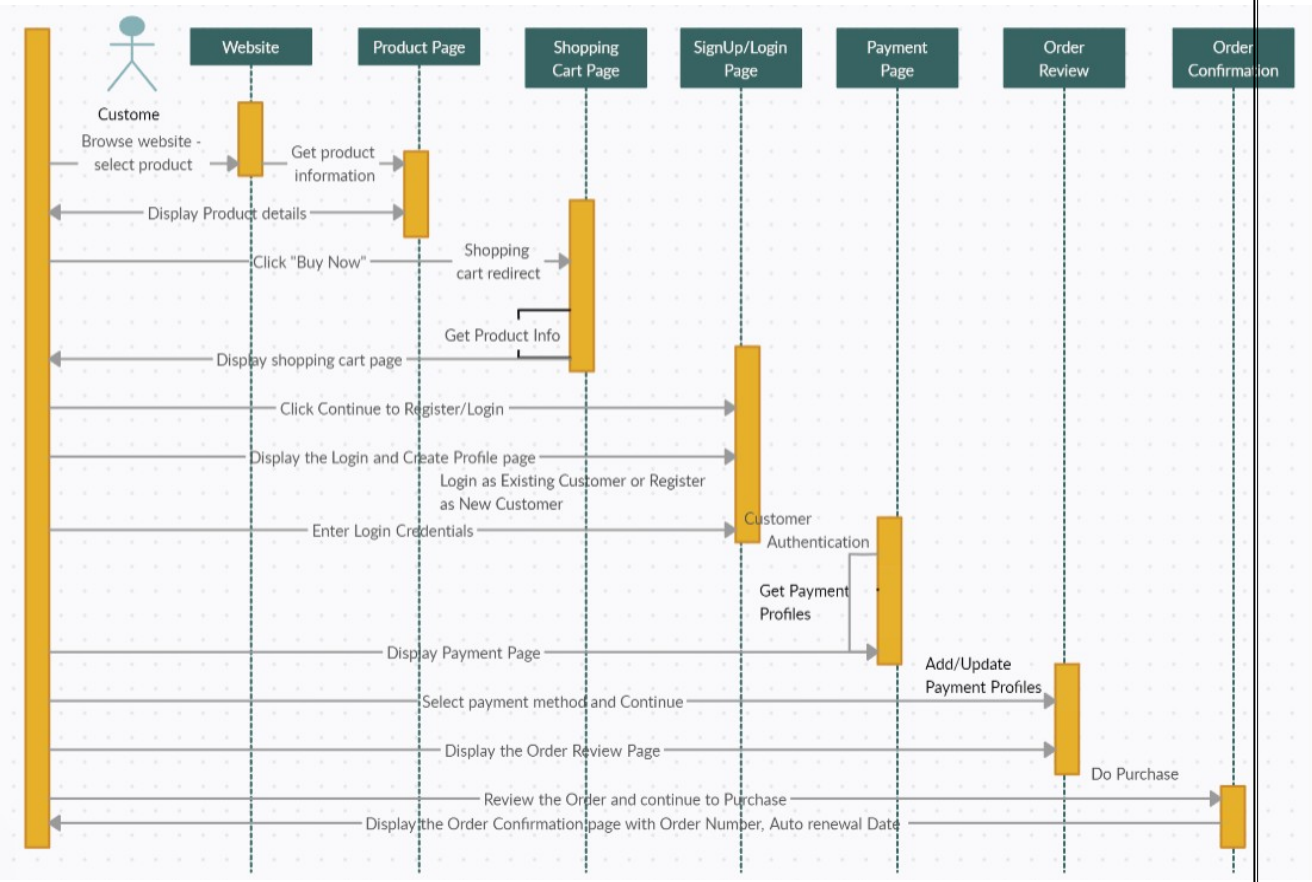


Figure : Sequence Diagram

TABLE STRUCTURE

Database: **grmart**

1)List of Tables:

```
mysql> use grmart
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------+
| Tables_in_grmart |
+-----------------+
| addresses        |
| cart             |
| categories       |
| order_details    |
| orders           |
| payments         |
| products         |
| stock            |
| users            |
+-----------------+
9 rows in set (0.00 sec)
```

2)Category Table:

```
mysql> desc categories;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| id     | int         | NO   | PRI | NULL    | auto_increment |
| name   | varchar(30) | YES  |     | NULL    |                |
| status | varchar(30) | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

3)Products Table:

```
mysql> desc products;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| description | varchar(100) | YES  |     | NULL    |                |
| image_name  | varchar(40)  | YES  |     | NULL    |                |
| name        | varchar(30)  | YES  |     | NULL    |                |
| price       | double       | NO   |     | NULL    |                |
| status      | varchar(30)  | YES  |     | NULL    |                |
| category_id | int          | NO   | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

4)Cart Table:

```
mysql> desc cart;
+-------------+------+------+-----+---------+----------------+
| Field       | Type | Null | Key | Default | Extra          |
+-------------+------+------+-----+---------+----------------+
| id          | int  | NO   | PRI | NULL    | auto_increment |
| quantity    | int  | NO   |     | NULL    |                |
| customer_id | int  | YES  | MUL | NULL    |                |
| product_id  | int  | YES  | MUL | NULL    |                |
+-------------+------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

5) Orders Table:

```
mysql> desc orders;
+---------------------+-------------+------+-----+---------+----------------+
| Field               | Type        | Null | Key | Default | Extra          |
+---------------------+-------------+------+-----+---------+----------------+
| id                  | int         | NO   | PRI | NULL    | auto_increment |
| order_date          | datetime(6) | YES  |     | NULL    |                |
| order_status        | varchar(25) | YES  |     | NULL    |                |
| status_update_date  | datetime(6) | YES  |     | NULL    |                |
| total_price         | double      | NO   |     | NULL    |                |
| customer_id         | int         | NO   | MUL | NULL    |                |
| delivery_address_id | int         | YES  | MUL | NULL    |                |
| employee_id         | int         | YES  | MUL | NULL    |                |
+---------------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

6) Order_details Table:

```
mysql> desc order_details;
+------------+--------+------+-----+---------+----------------+
| Field      | Type   | Null | Key | Default | Extra          |
+------------+--------+------+-----+---------+----------------+
| id         | int    | NO   | PRI | NULL    | auto_increment |
| price      | double | NO   |     | NULL    |                |
| quantity   | int    | NO   |     | NULL    |                |
| order_id   | int    | NO   | MUL | NULL    |                |
| product_id | int    | NO   | MUL | NULL    |                |
+------------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

7) Payments table:

```
mysql> desc payments;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| id           | int         | NO   | PRI | NULL    | auto_increment |
| amount       | double      | NO   |     | NULL    |                |
| payment_date | datetime(6) | YES  |     | NULL    |                |
| status       | varchar(12) | YES  |     | NULL    |                |
| type         | varchar(12) | YES  |     | NULL    |                |
| order_id     | int         | YES  | MUL | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
6 rows in set (0.01 sec)
```

8) Stock table:

```
mysql> desc stock;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| product_id | int         | NO   | PRI | NULL    |       |
| quantity   | int         | NO   |     | NULL    |       |
| unit       | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

9) Users table:

```
mysql> desc users;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | int          | NO   | PRI | NULL    | auto_increment |
| email      | varchar(30)  | YES  | UNI | NULL    |                |
| first_name | varchar(30)  | YES  |     | NULL    |                |
| last_name  | varchar(30)  | YES  |     | NULL    |                |
| password   | varchar(255) | NO   |     | NULL    |                |
| phone      | varchar(15)  | YES  |     | NULL    |                |
| role       | varchar(255) | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

10)Addresses table:

```
mysql> desc addresses;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| id            | int         | NO   | PRI | NULL    | auto_increment |
| address_line1 | varchar(45) | YES  |     | NULL    |                |
| address_line2 | varchar(45) | YES  |     | NULL    |                |
| city          | varchar(30) | YES  |     | NULL    |                |
| country       | varchar(30) | YES  |     | NULL    |                |
| pin_code      | varchar(10) | YES  |     | NULL    |                |
| state         | varchar(30) | YES  |     | NULL    |                |
| user_id       | int         | NO   | MUL | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

11)User table testing via Junit Test:

```
mysql> drop database grmart;
Query OK, 9 rows affected (0.08 sec)

mysql> use grmart;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from users;
+----+-------------------+------------+-----------+--------------------------------------------------------------+------------+-------+
| id | email             | first_name | last_name | password                                                     | phone      | role  |
+----+-------------------+------------+-----------+--------------------------------------------------------------+------------+-------+
|  1 | abhay@gmail.com   | Abhay      | Krishna   | $2a$10$TuvX2Suo6OOm24pXFvs4tul79iroA18XEAUsbhiBZlHSbbfvf07Ni | 8765204921 | ADMIN |
|  2 | pratik@gmail.com  | Pratik     | Mohite    | $2a$10$0ubU8.Y/EPG2ANdyNwGmZeKuovUVnpVP8QfmBBpwR.fmNvELpm6Zi | 888746486  | ADMIN |
|  3 | mayuresh@gmail.com | Mayuresh   | Kholekar  | $2a$10$VS7kGFMgFqyCxALEbl5XwO.uEnBbA9lwENBYrbTpXnxlPfLGauZ9i | 999753548  | ADMIN |
+----+-------------------+------------+-----------+--------------------------------------------------------------+------------+-------+
3 rows in set (0.00 sec)
```

12) Categories table testing via Junit Test:

```
mysql> drop database grmart;
Query OK, 9 rows affected (0.08 sec)

mysql> use grmart;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from users;
+----+-------------------+------------+-----------+--------------------------------------------------------------+------------+-------+
| id | email             | first_name | last_name | password                                                     | phone      | role  |
+----+-------------------+------------+-----------+--------------------------------------------------------------+------------+-------+
|  1 | abhay@gmail.com   | Abhay      | Krishna   | $2a$10$TuvX2Suo6OOm24pXFvs4tul79iroA18XEAUsbhiBZlHSbbfvf07Ni | 8765204921 | ADMIN |
|  2 | pratik@gmail.com  | Pratik     | Mohite    | $2a$10$0ubU8.Y/EPG2ANdyNwGmZeKuovUVnpVP8QfmBBpwR.fmNvELpm6Zi | 888746486  | ADMIN |
|  3 | mayuresh@gmail.com | Mayuresh   | Kholekar  | $2a$10$VS7kGFMgFqyCxALEbl5XwO.uEnBbA9lwENBYrbTpXnxlPfLGauZ9i | 999753548  | ADMIN |
+----+-------------------+------------+-----------+--------------------------------------------------------------+------------+-------+
3 rows in set (0.00 sec)

mysql> select * from categories;
+----+---------+--------+
| id | name    | status |
+----+---------+--------+
|  1 | Grains  | ACTIVE |
|  2 | Biscuit | ACTIVE |
+----+---------+--------+
2 rows in set (0.00 sec)
```

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	Create Admin User	Creation of users in Database	OK	Nothing
2	Create Categories	Creation of categories in Database	Ok	Nothing

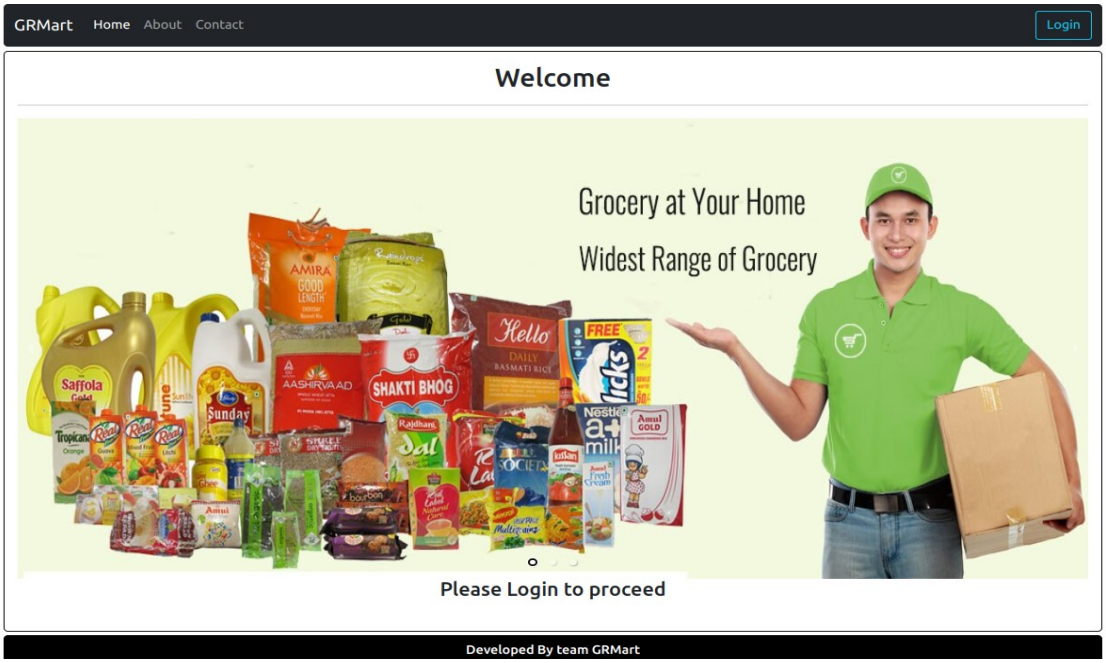
6. PROJECT MANAGEMENT RELATED STATISTICS

DATE	WORK PERFORMED	*SLC Phase*	Additional Notes
JUNE 12,2023	Project Allotment and User Requirements Gathering	Feasibility Study	Submission and Decision on Gr Mart E-comm Project.
JUNE 19,2023	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
JUNE 26,2023	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
JUNE 30,2023	Business Logic Component design Started	Design Phase	----------
JULY 3,2023	Coding Phase Started	Coding Phase	Homepage complete.

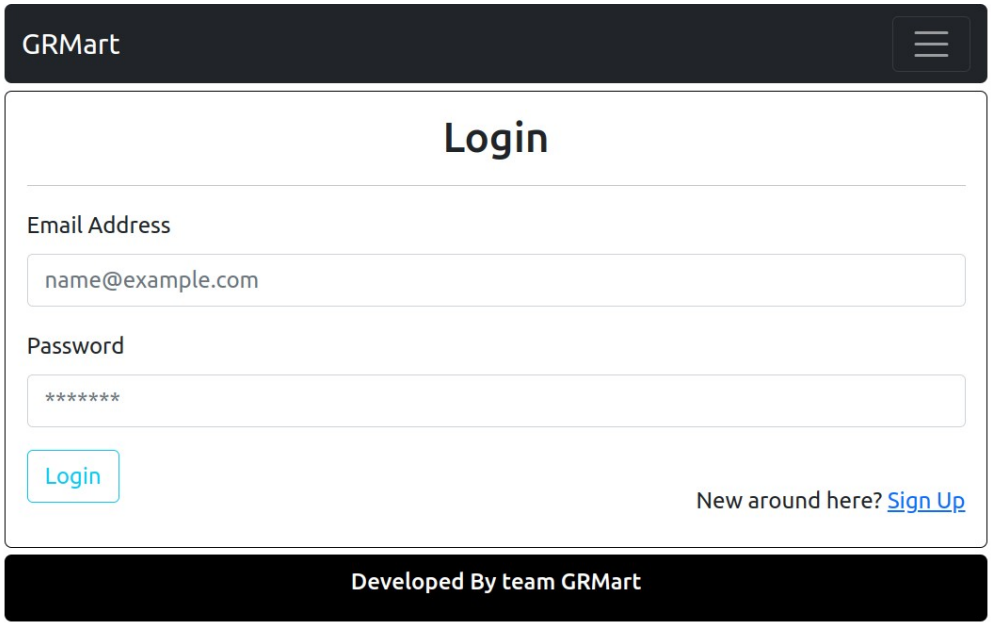
JULY 10,2023	Started Front end React code	Coding Phase	Class Library Development going on.
JULY 15,2023	Off	Off	Off
JULY 24,2023	React Front end completed		Class Library Modified as per the need.
AUG 11,2023	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	--
AUG 13,2023	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
AUG 15,2023	Implementation of Web Application and Window Application Started	Testing Phase (Module Testing)	--
AUG 24,2023	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 28,2023	The Errors Found were Removed	Debugging	The Project was complete for submission
AUG 30,2023	Final Submission of Project		

Appendix A

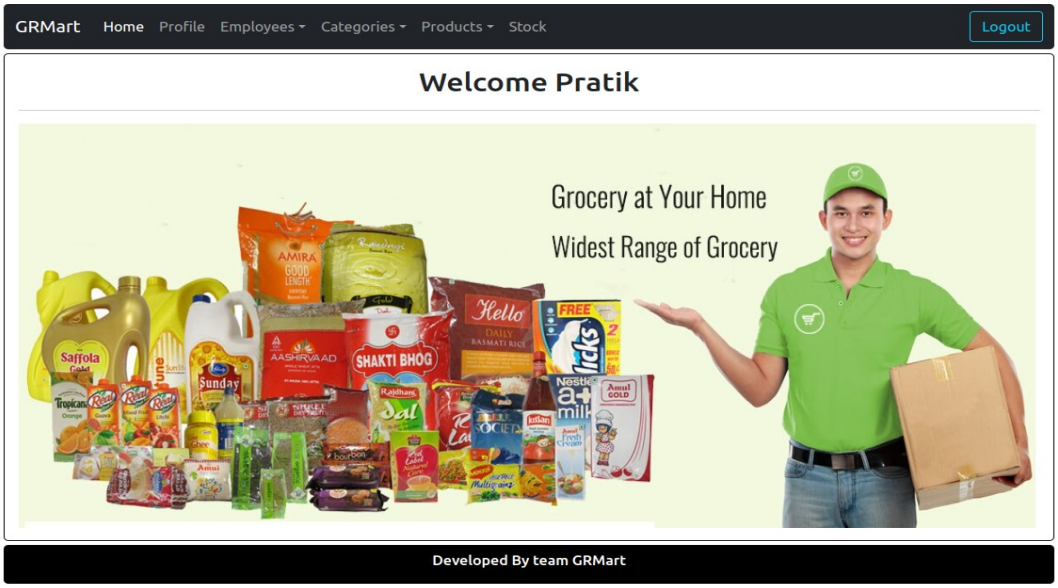
Homepage:



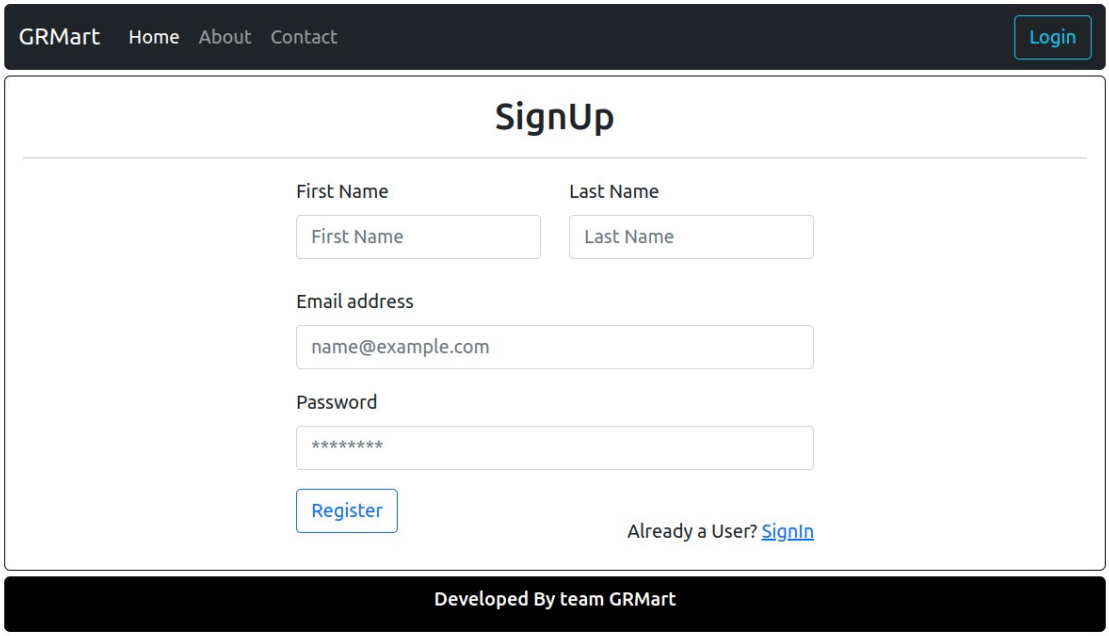
LoginPopup



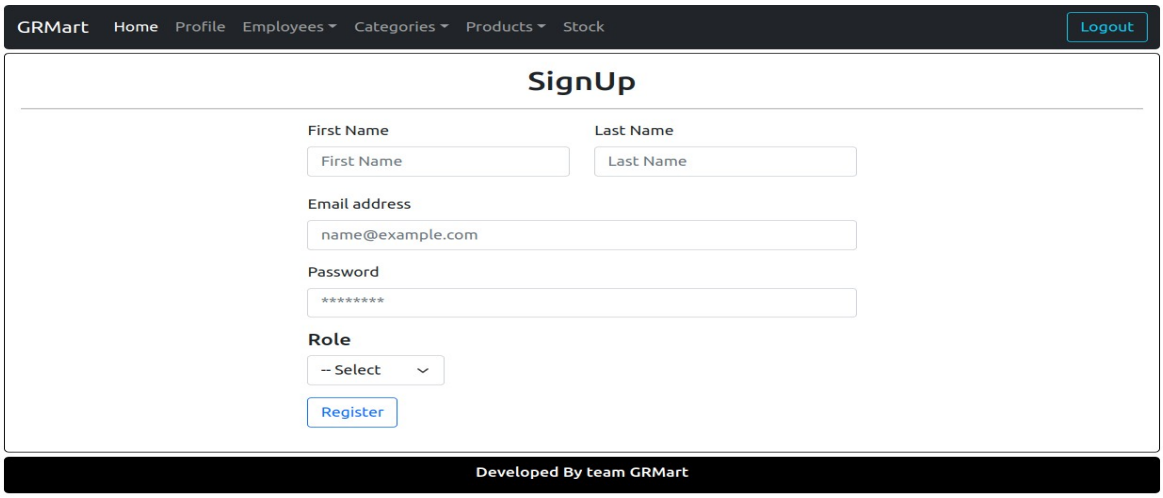
Administrators Login:



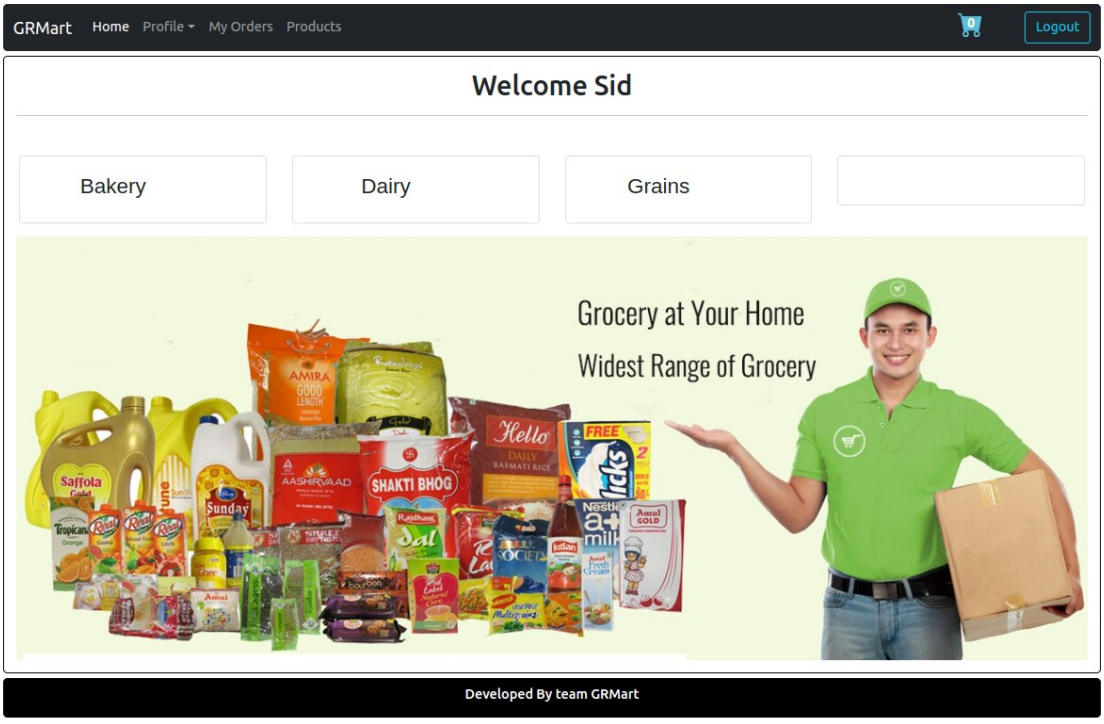
User Registration:



Add Employee and Delivery Person

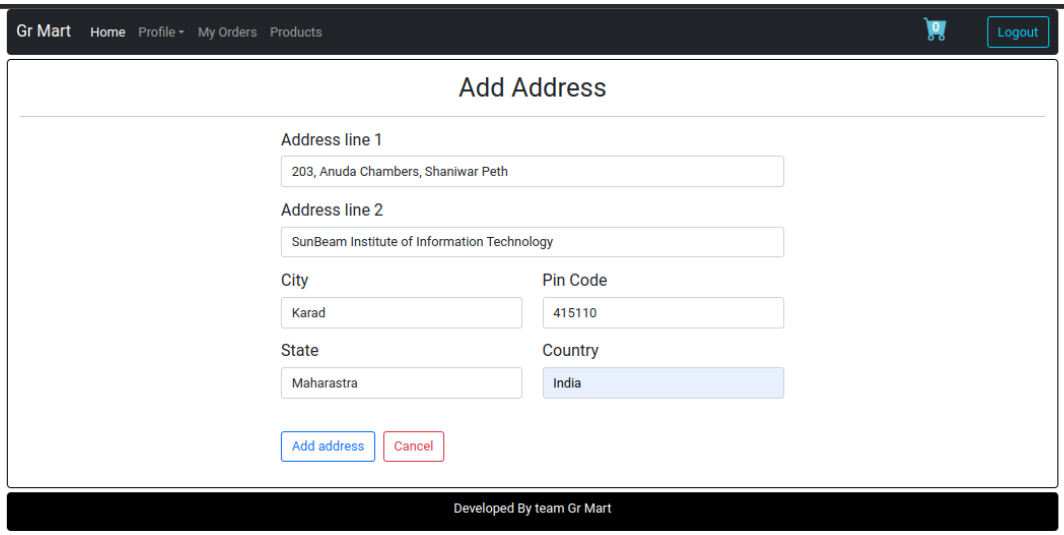


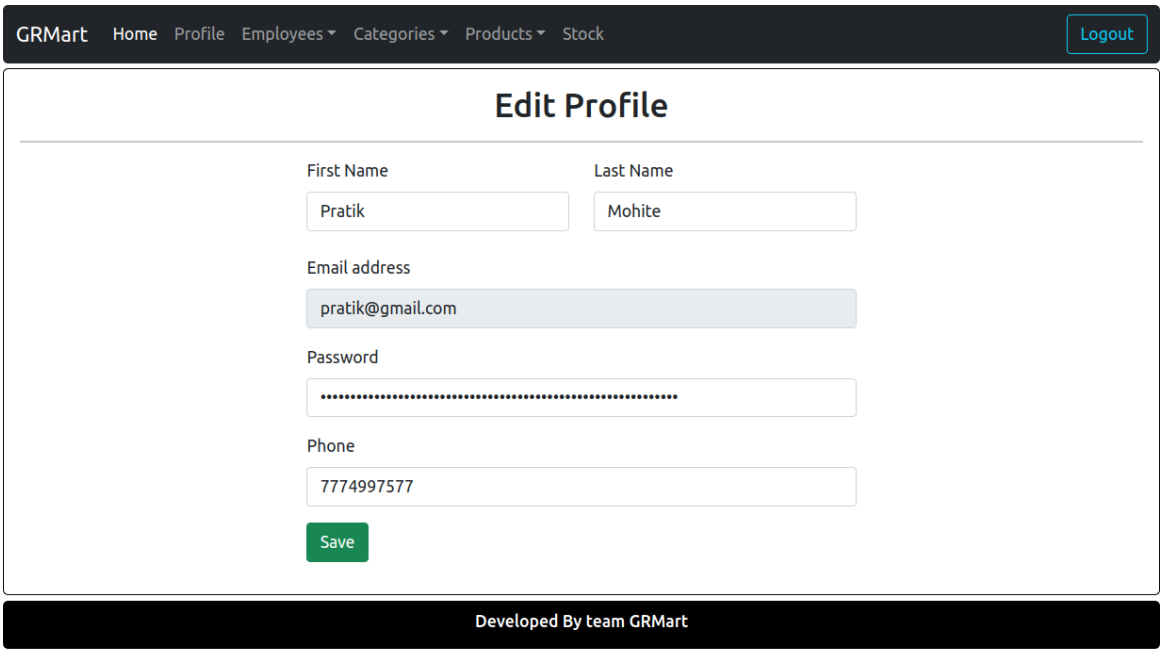
Login For Placing Order :



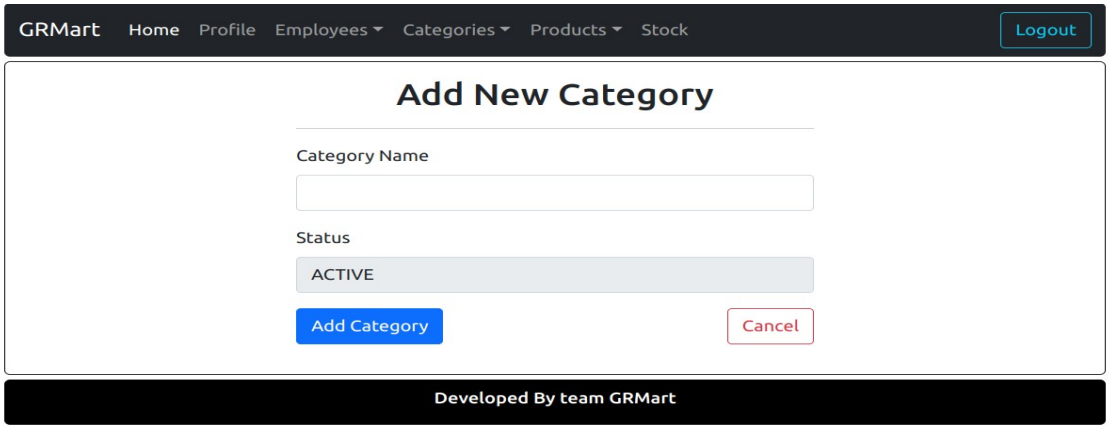
User Profile Details

Addresses

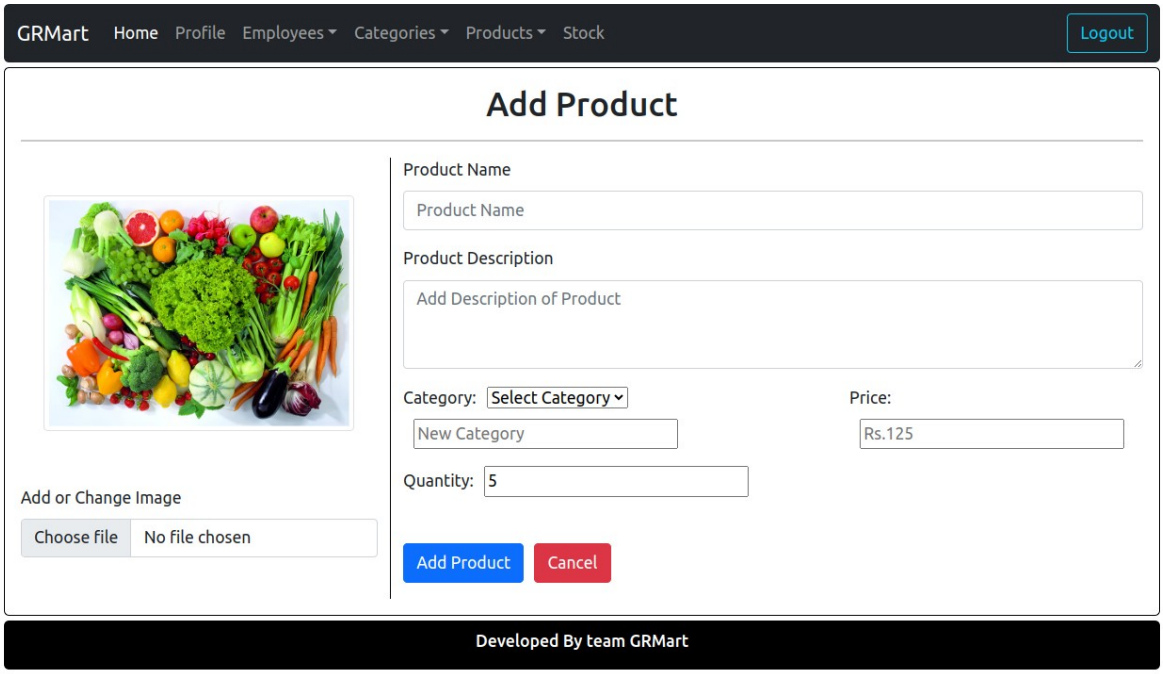




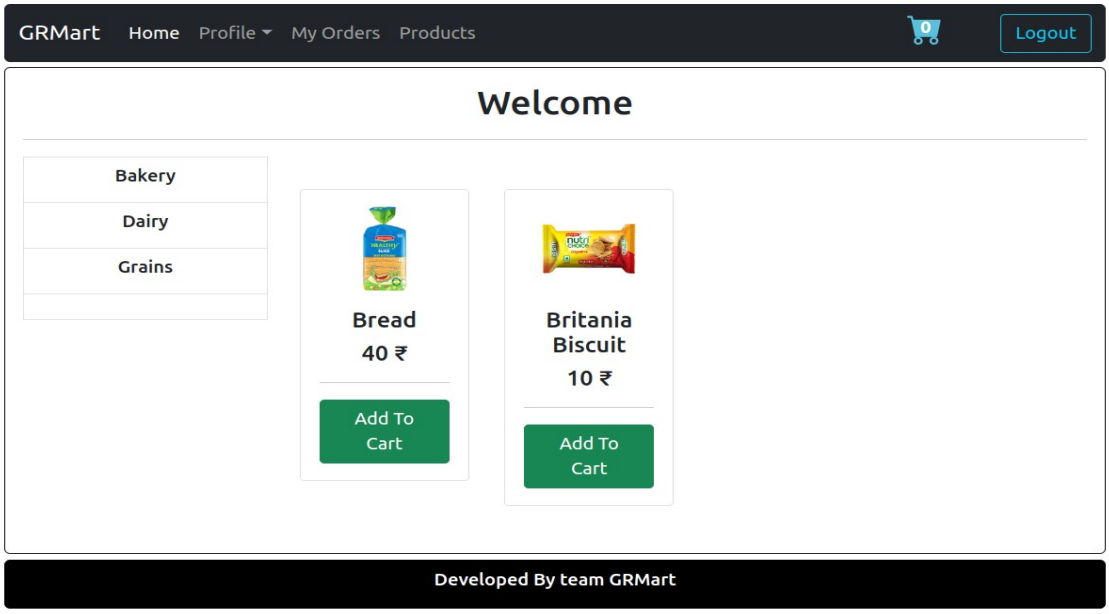
Add new Categories



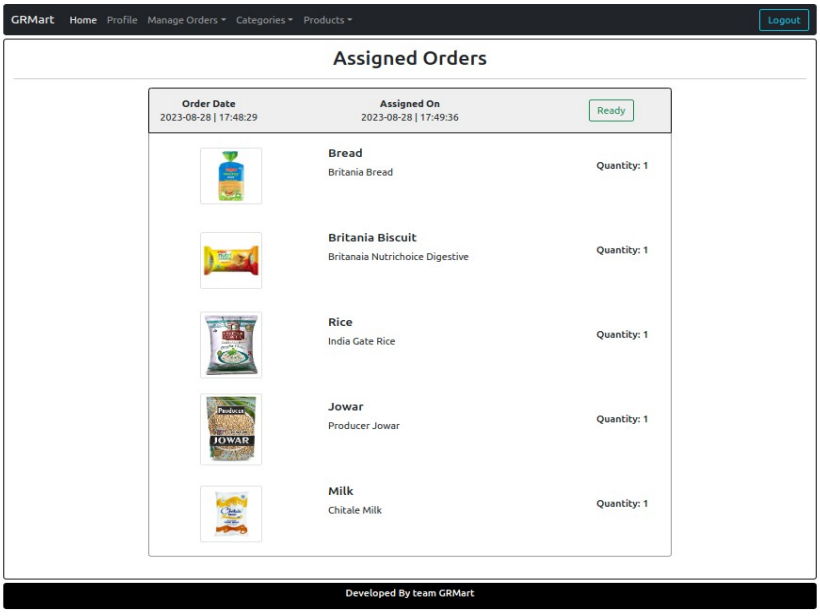
Add Product



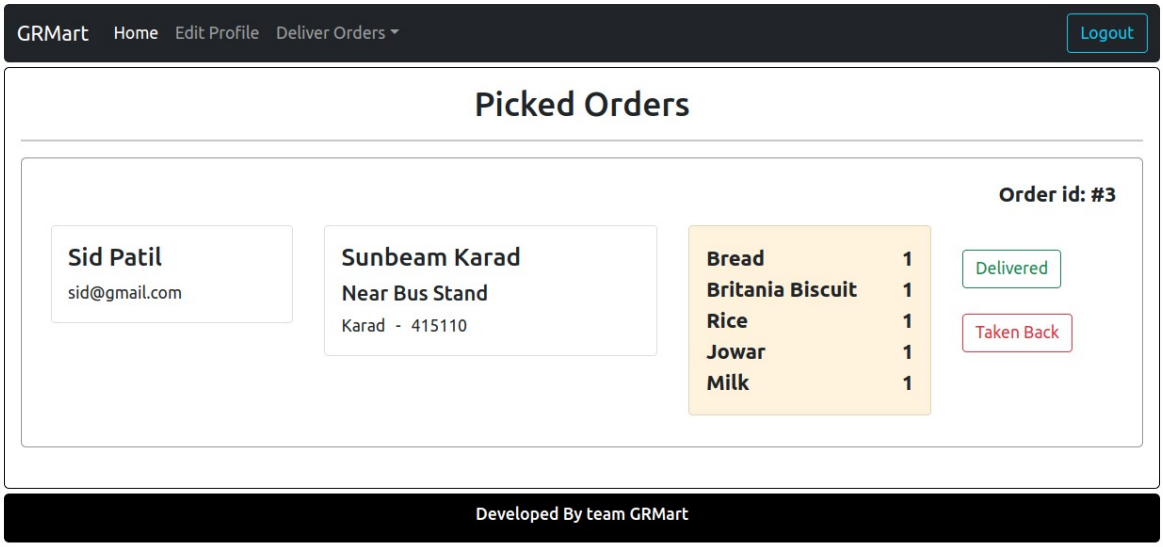
Product List



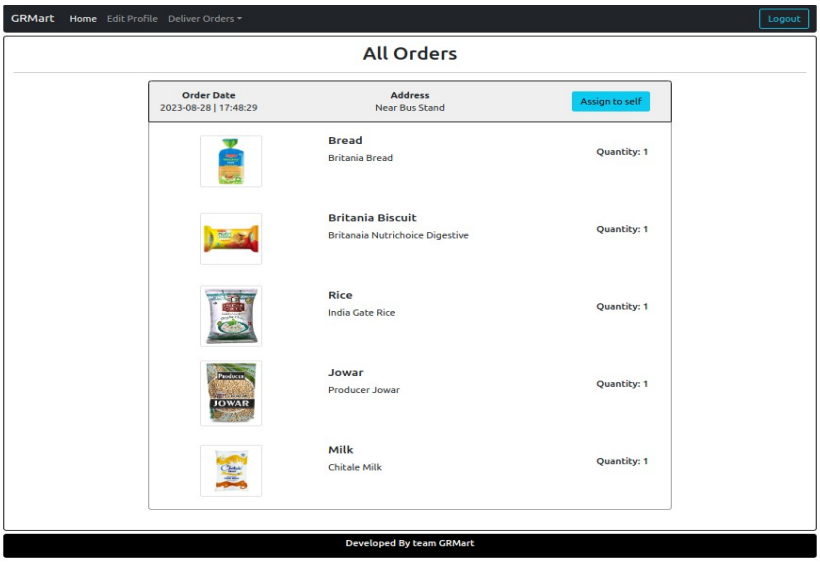
Assign Orders



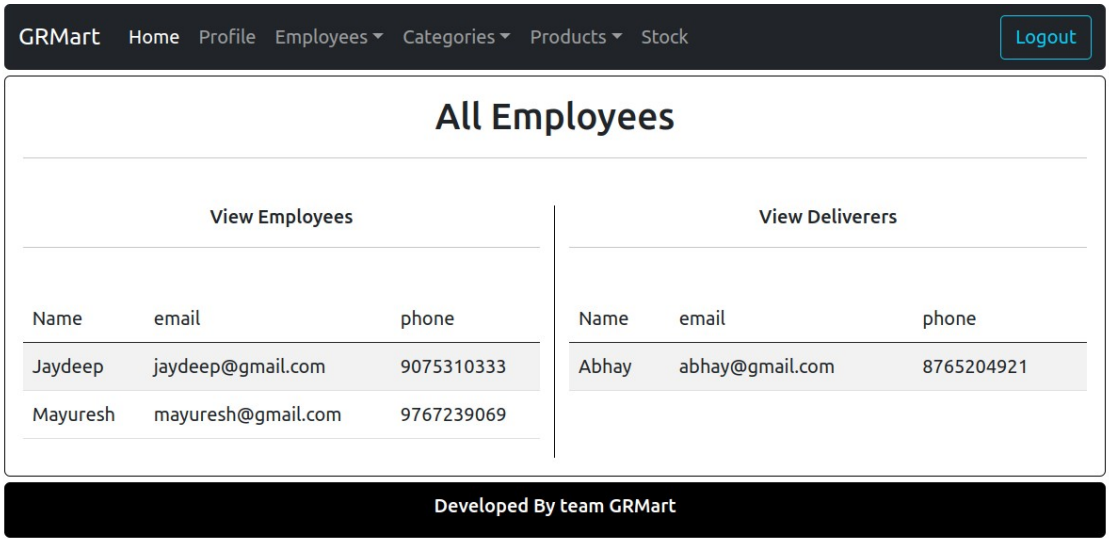
Picked Order by Delivery Person



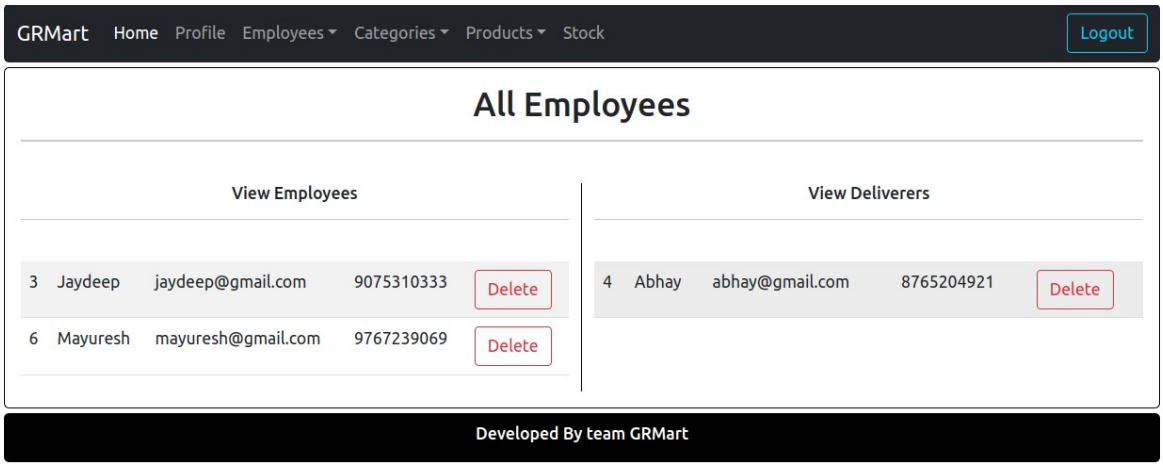
Ready Orders by Delivery Person



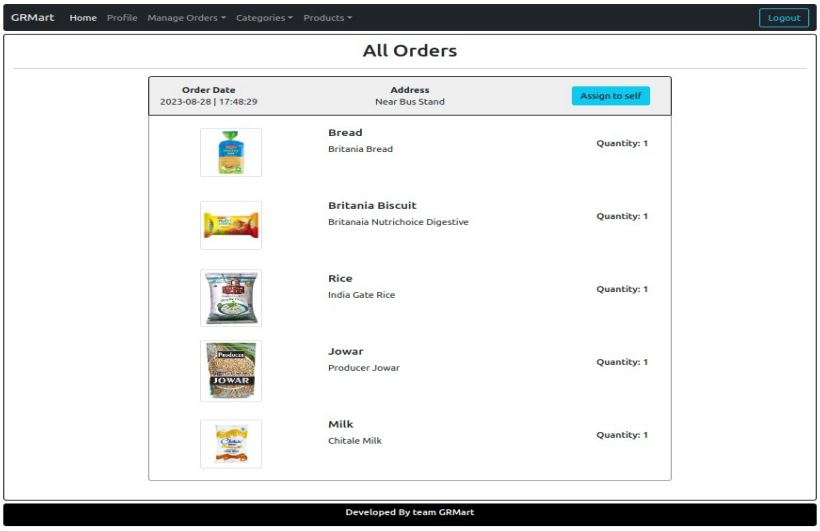
View Employees and Delivery Person



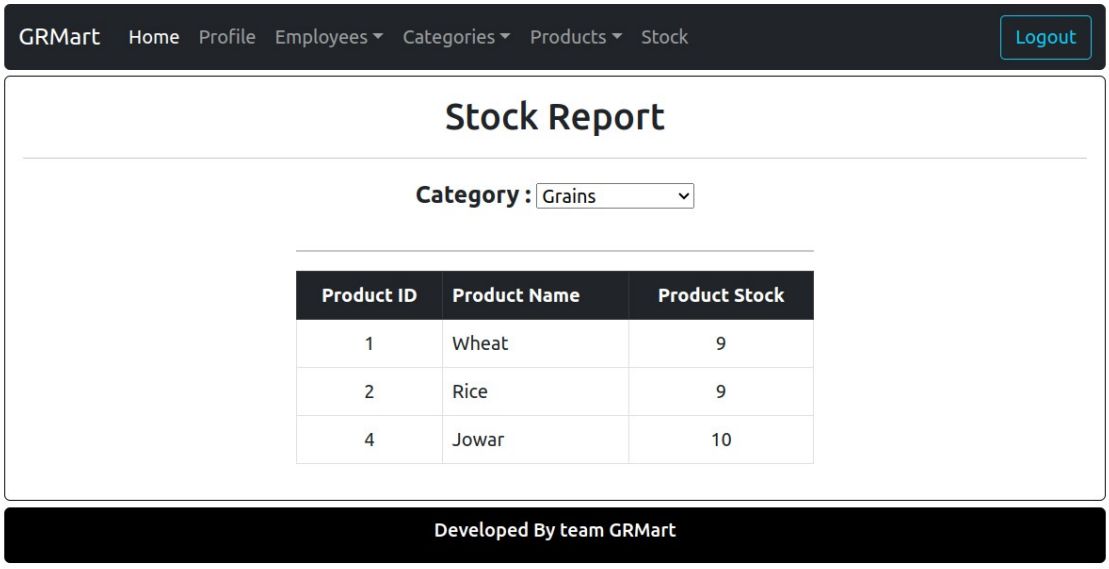
Delete Employee by Admin



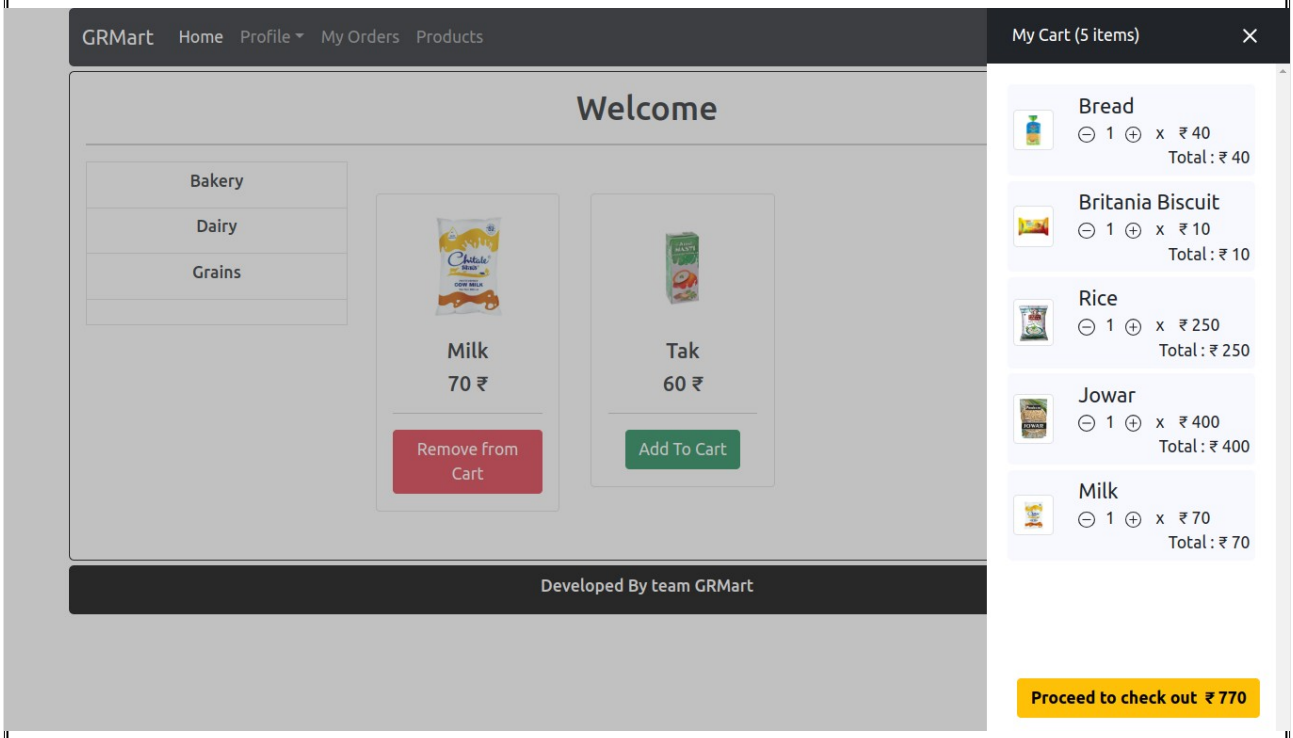
User Order History



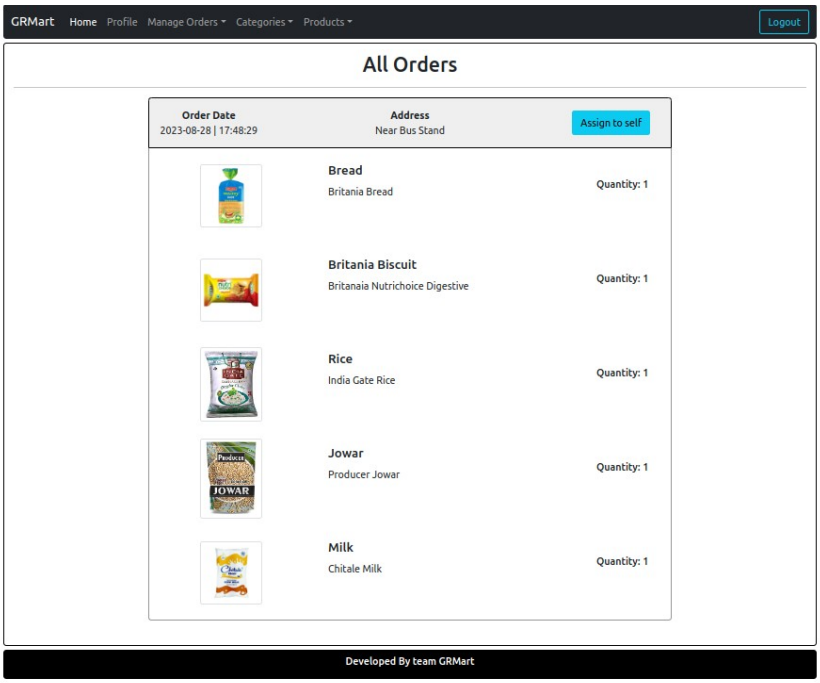
Stocks



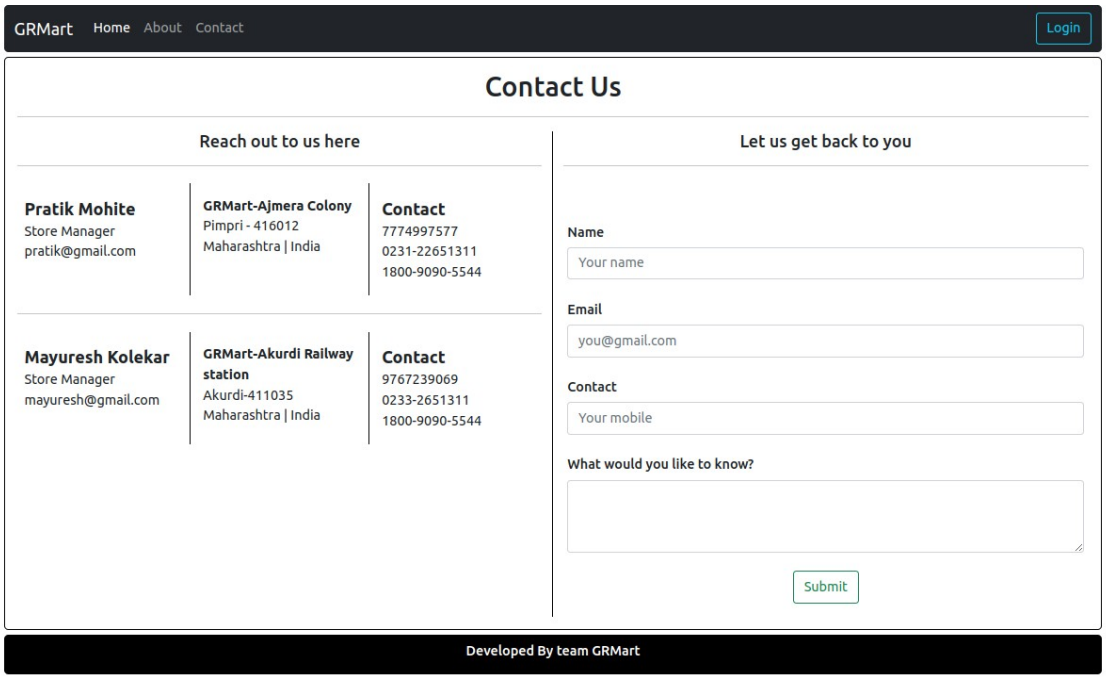
Cart

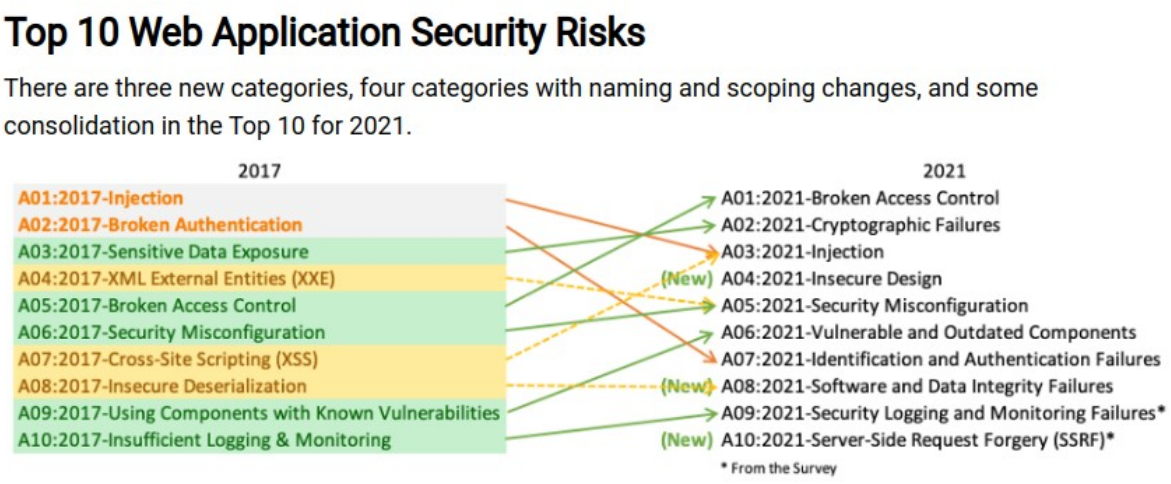


All Orders



Contact Us


GRMart
Home
About
Contact
Login
Contact Us
Reach out to us here
Pratik Mohite
Store Manager
pratik@gmail.com
GRMart-Ajmera Colony
Pimpri - 416012
Maharashtra | India
Contact
7774997577
0231-22651311
1800-9090-5544
Mayuresh Kolekar
Store Manager
mayuresh@gmail.com
GRMart-Akurdi Railway station
Akurdi-411035
Maharashtra | India
Contact
9767239069
0233-2651311
1800-9090-5544
Let us get back to you
Name
Your name
Email
you@gmail.com
Contact
Your mobile
What would you like to know?
Submit
Developed By team GRMart



A01:2021-Broken Access Control moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.

- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
- **A04:2021-Insecure Design** is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
- **A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.
- **A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to

the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.

- **A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
- **A08:2021-Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.
- **A09:2021-Security Logging and Monitoring Failures** was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- **A10:2021-Server-Side Request Forgery** is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.

References:

Java: https://docs.oracle.com/javase/8/docs/api/

React:https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_resources

Html: https://developer.mozilla.org/en-US/docs/Web/HTML

html/css: https://www.w3schools.com/

Security : https://owasp.org/www-project-top-ten/