INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT, AKURDI, PUNE

Documentation On

"GoGrocers"- Online Grocery Shopping Web Application

PG -DAC SEPT 2022

Submitted By:

Group No: 09

Roll No.	**Name:**
229013	**Ashwini Palve**
229039	**Komal Ghorpade**

Mr. Rohit Puranik	**Mrs . Sonali Mogal**
Centre Coordinator	Project Guide

ABSTRACT

This project is a web-based shopping system for an existing shop. The project objective is to deliver the online shopping application into web platform.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using a website device. Thus, the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains.

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 Flipkart or e-bay. Since this application is available it is easily accessible and always available.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide**, Mrs. Sonali Mogal** for providing me with the right guidance and advice at the crucial juncture sand for showing me the right way. I extend my sincere thanks to our respected Centre Co-Ordinator **Mr. Rohit Puranik**, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work

Ashwini Palve (229013)

Komal Ghorpade(229039)

Table of Contents

LIST OF FIGURE

INTRODUCTION

This project is a web-based grocery shopping system for an existing shop. An Online Grocery Shopping Management System where the Admin can Add, Update and Delete products. The Products are divided into various categories like Bakery, Oils and solids , Biscuits , Vegetables , Grains 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 an 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 an application platform 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, Besan) products easily

3. Category of Products-Bakery, Vegetables, Grains, 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

1.1 PROJECT OBJECTIVE

The objective of the project is to make an application platform to purchase grocery 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 web application with web view.

1.2 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.

1.3 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 available and always available.

1.4 STUDY OF THE SYSTEM

1.4.1 MODULES:

The system after careful analysis has been identified to be presented with the following modules and roles. The modules involved are:

➢User : Admin , Employee , Customer , Delivery Person

➢Products

➢Categories

➢Cart

➢Address

➢Payments

➢ Delivery Person

1.4.1.1 Administrator:

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about the users and about all products. This module is divided into different sub modules.

1. Manage Stocks
2. Manage Products
3. Manage Users
4. Manage Orders
5. Manage Delivery Person

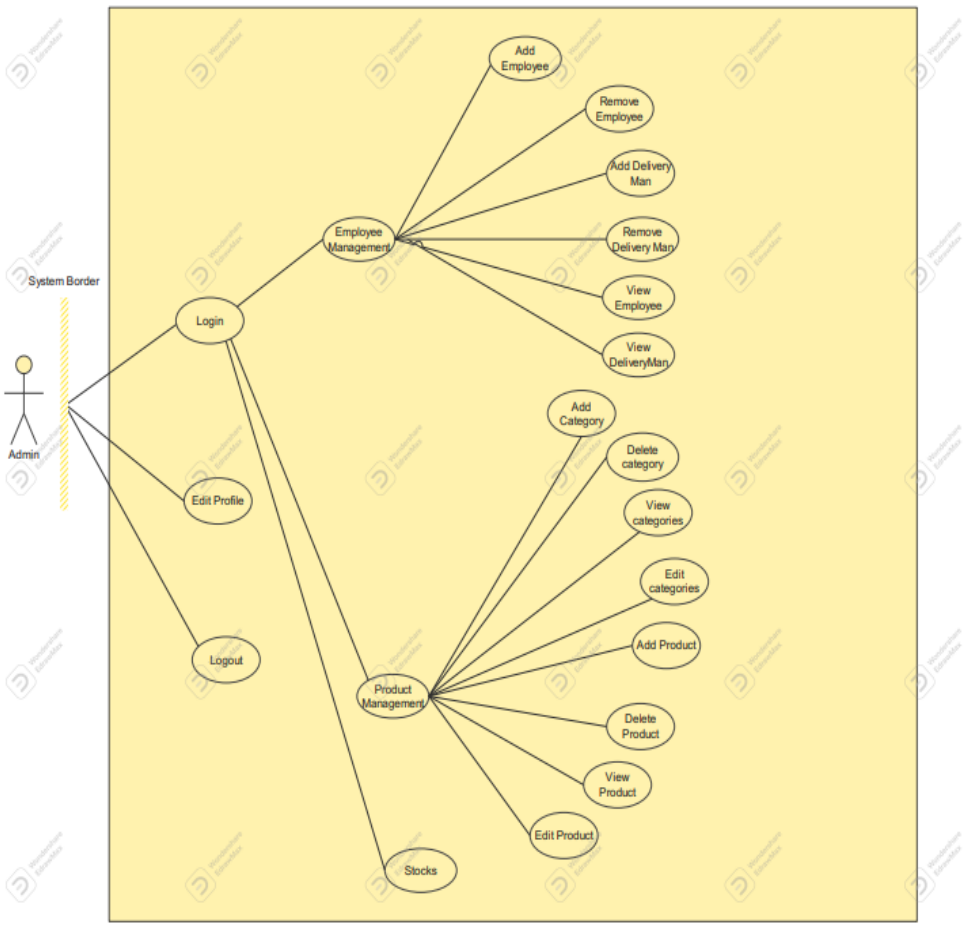


Figure: Admin Usecase Diagram

➢ **Add Products**

The shopping cart contains different kinds of products of different category. The products can be classified into different categories by name. Admin can add new products into the existing system with all its details including an image.

➢ **Delete Products**

Administrator can delete the products based on the stock of that particular product.

➢ Search Products Admin will have a list view of all the existing products. He can also search for a particular product by name.

➢ **Add Delivery Person**

Admin can add delivery boy for the system. Admin can see the details of delivery boy and the order delivery.

➢ **Remove Delivery Person**

Admin have the privilege to remove the delivery boy.

➢ **Add Employee**

Admin have the privilege to add the Employee

➢ **Remove Employee**

Admin have the privilege to remove the Employee.

➢ **Edit Product and Categories**

Only admin is having the privilege to add a Product and categories. He can search the product to manage the product

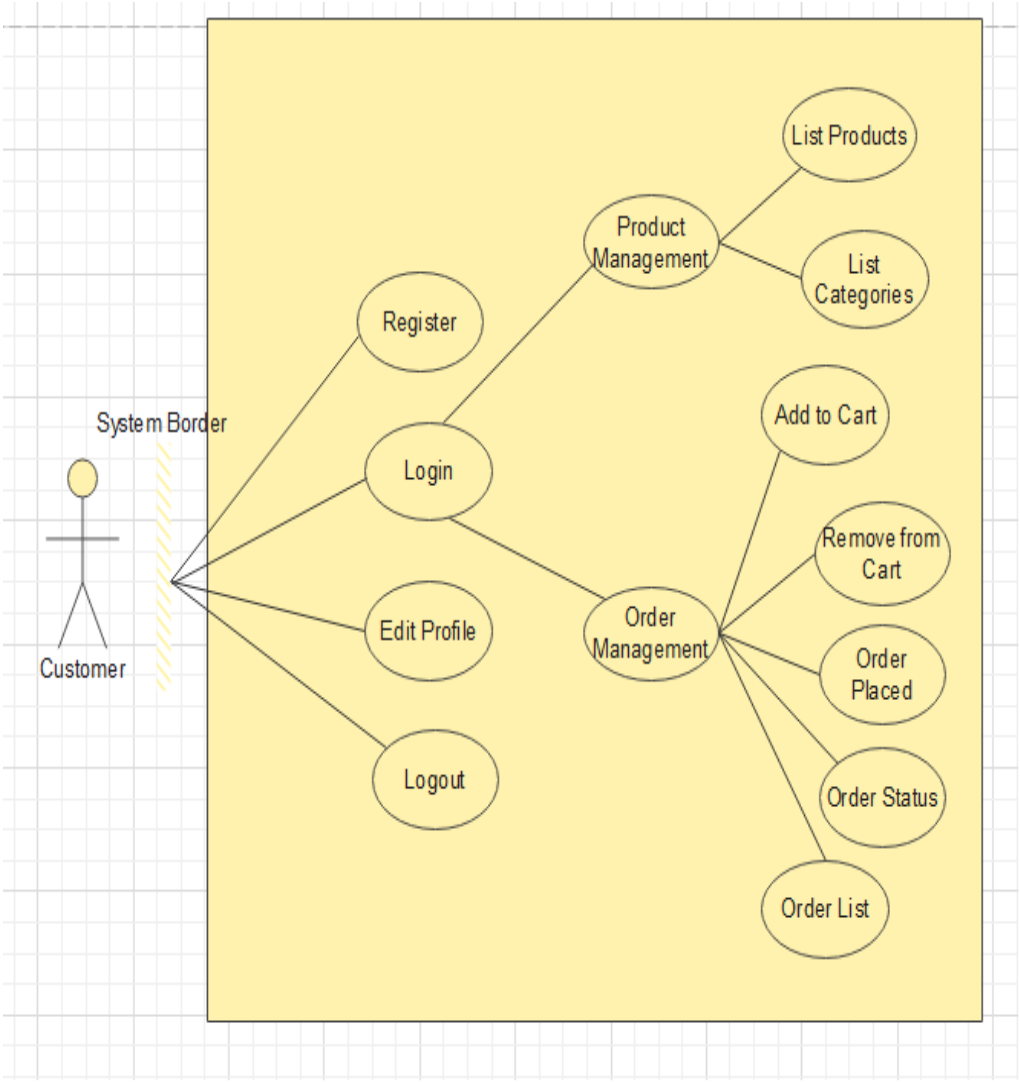


Figure: Customer UseCase Diagram

➢ **Customer sign in, sign out, create account**

This feature is provided to customer so he can sign in, sign out and create account for new customer.

➢ **Search Product**

Customer can search the product as per his wish in specific category.

➢ **Add to Cart**

Customer can add products to cart which he wants to buy the products.

➢ **Payments**

Customer have a privilege to his order he can see his order details. ➢ Order Details Customer have a privilege to his order he can see his order details.

➢ **Buy Product**

Customers can buy product from his cart by doing payment.

➢ **Wish List**

Customer can have a wish list for future buying products he can add products in the wish to list.

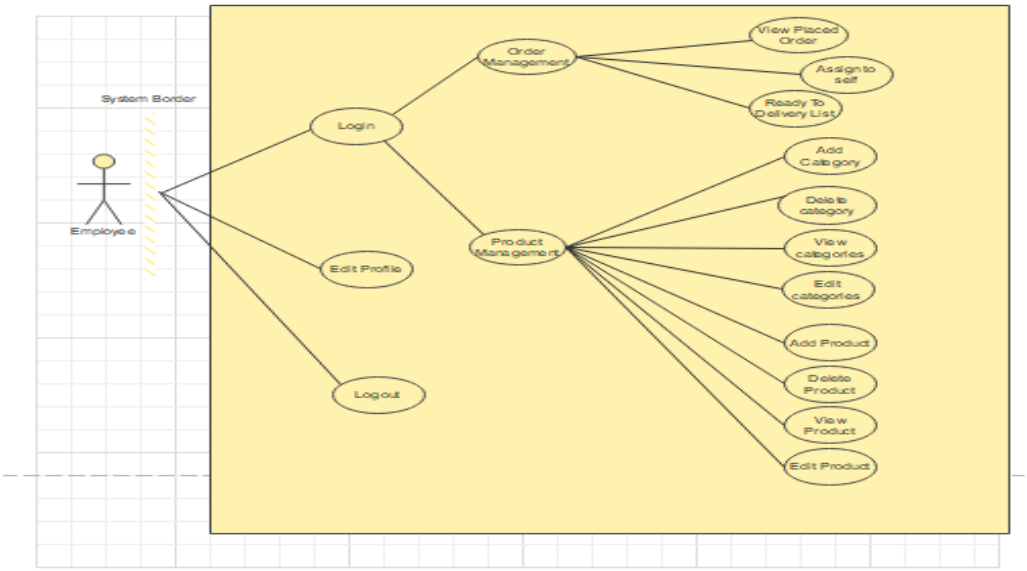


Figure : Employee Usecase Diagram

➢ **Add , Delete Categories**

Employee can add and delete categories and products to categories.

➢ **Add Products**

Employee can add products to categories.

➢ **Delete Products**

Employee can add and delete categories and products to categories.

➢ **Edit Product**

Employee can edit his added product.

➢ **Delivered Orders**

Employee can see order details of the ordered product by the customer.

➢ **Pending Orders**

Employee can see order details of the ordered product by the customer.

➢ **Payments**

Employee can see payment details of the ordered product by the customer.

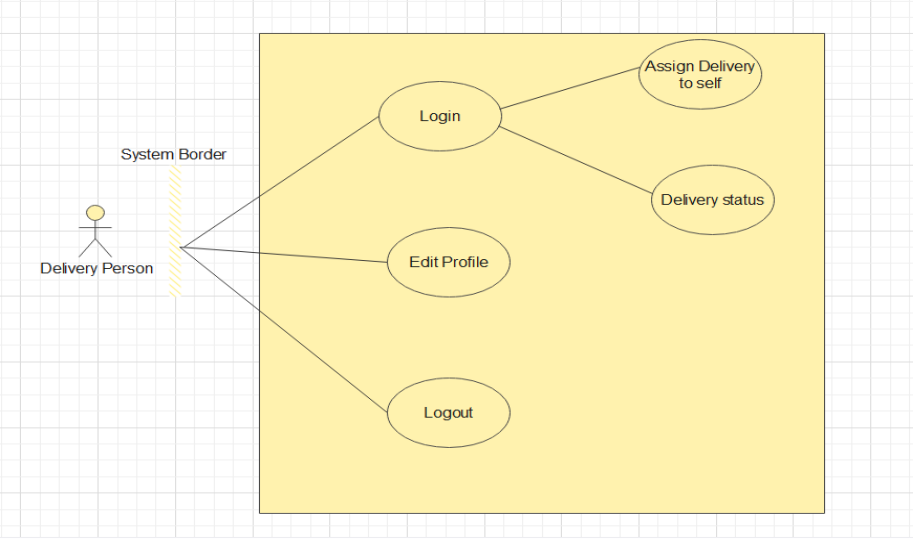


Figure: Delivery Person Use case diagram

➢ **Delivery Boy sign in, sign out**

This feature is provided to Delivery Person so he can sign in, sign out and delivery Person is added by admin.

➢ **Pending Orders**

Delivery Person can see the pending order details after sign in account.

➢ **Delivered Orders**

Delivery Person can see the delivered order details after sign in account.

➢ **Payment**

History Delivery Person can see the payment history of his product.

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 Grocery 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 Grocery 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

2.3.3.1 NON-FUNCTIONAL REQUIREMENTS

i. EFFICIENCY REQUIREMENT

When an online shopping cart 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.

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.

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.

DATABASE DESIGN

3.2 DATABASE

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.

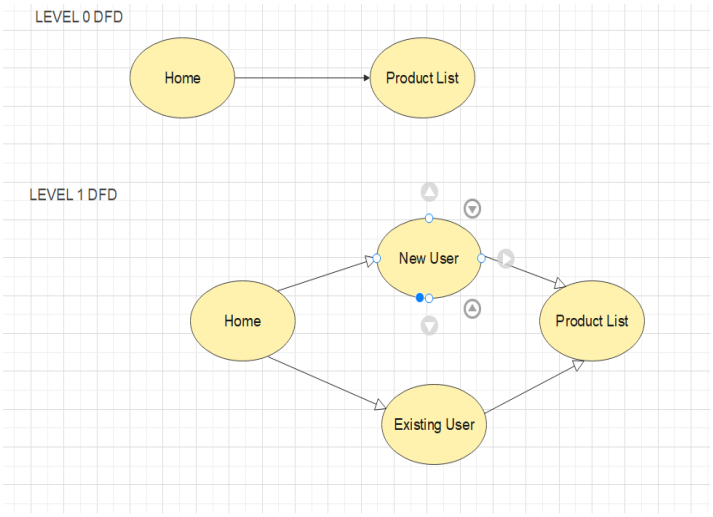
MySQL:

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.

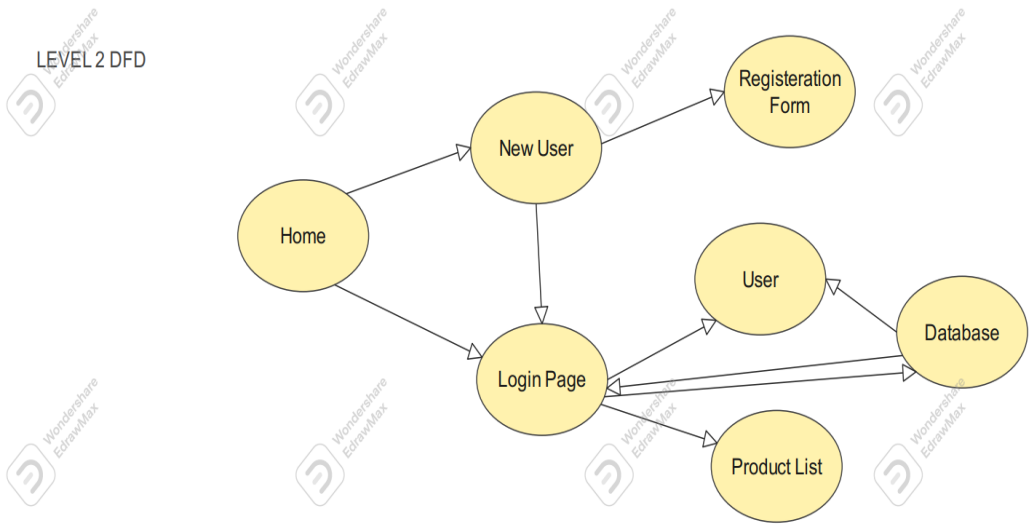
Spring-Boot:

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.

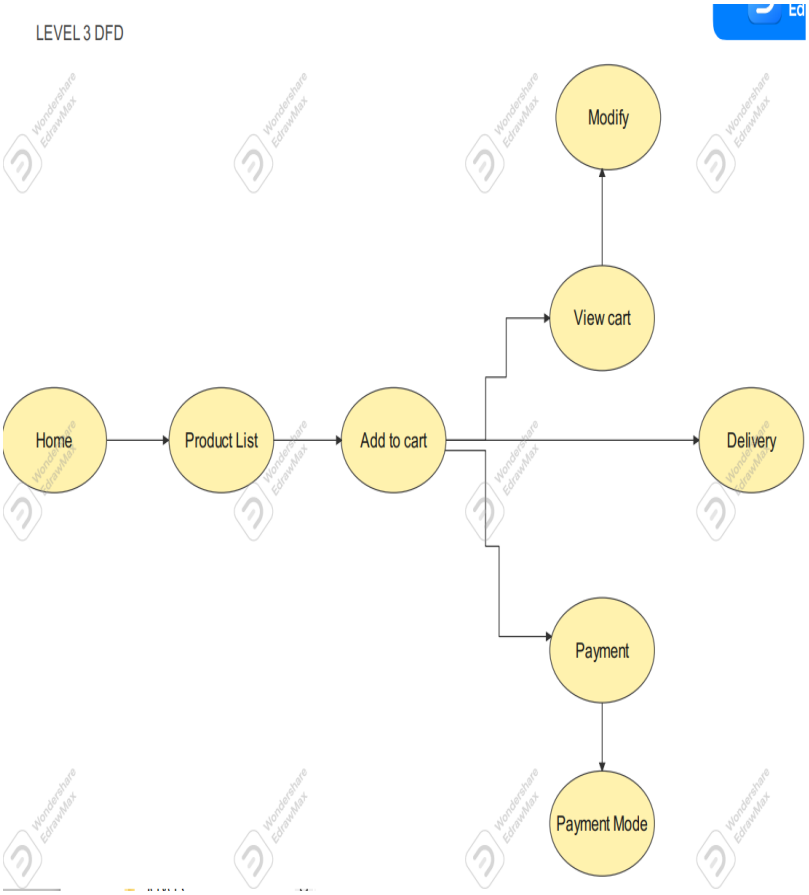
DFD LEVEL 0:



DFD LEVEL 2



DFD LEVEL 3


LEVEL 3 DFD
Modify
View cart
Home
Product List
Add to cart
Delivery
Payment
Payment Mode

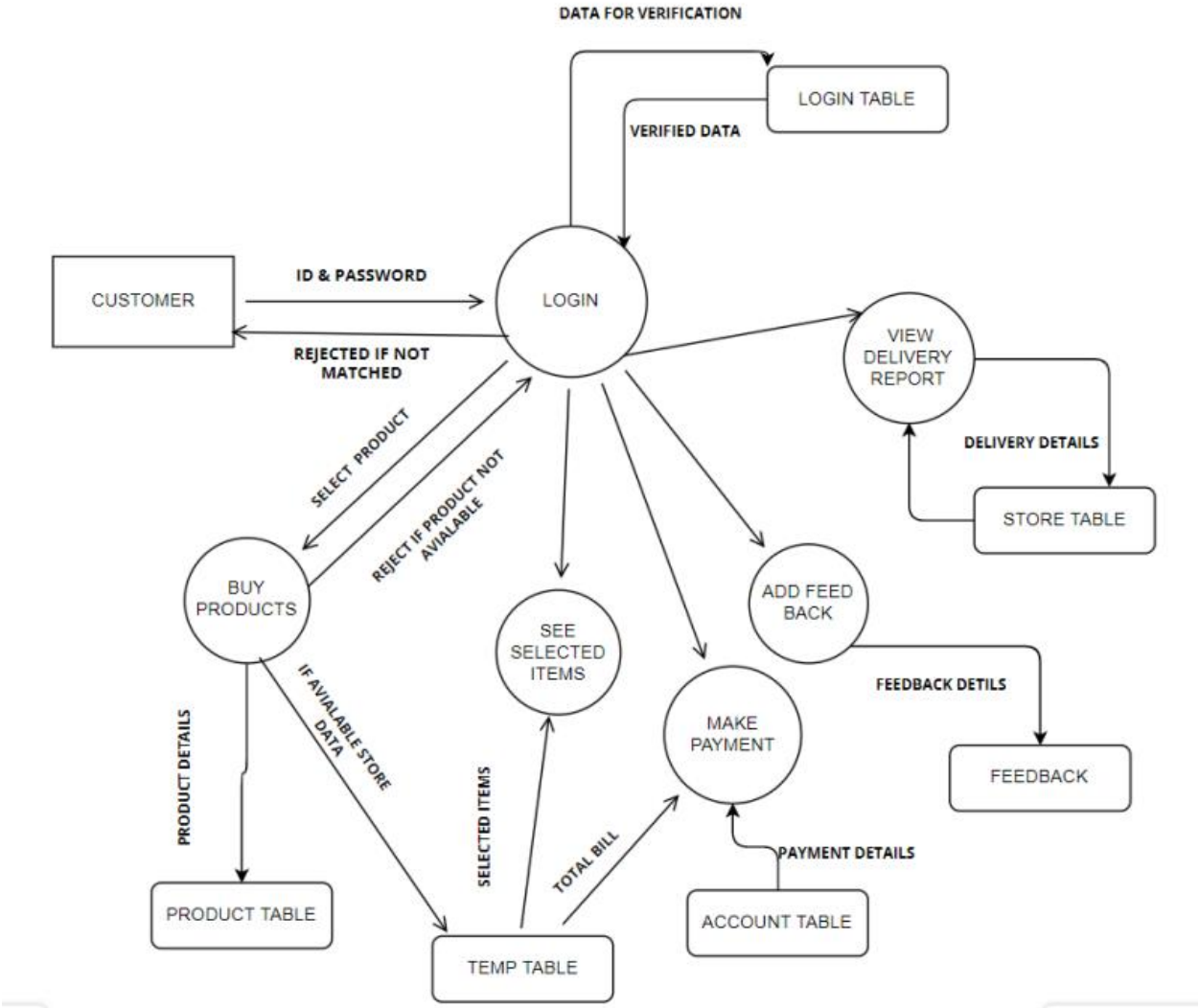
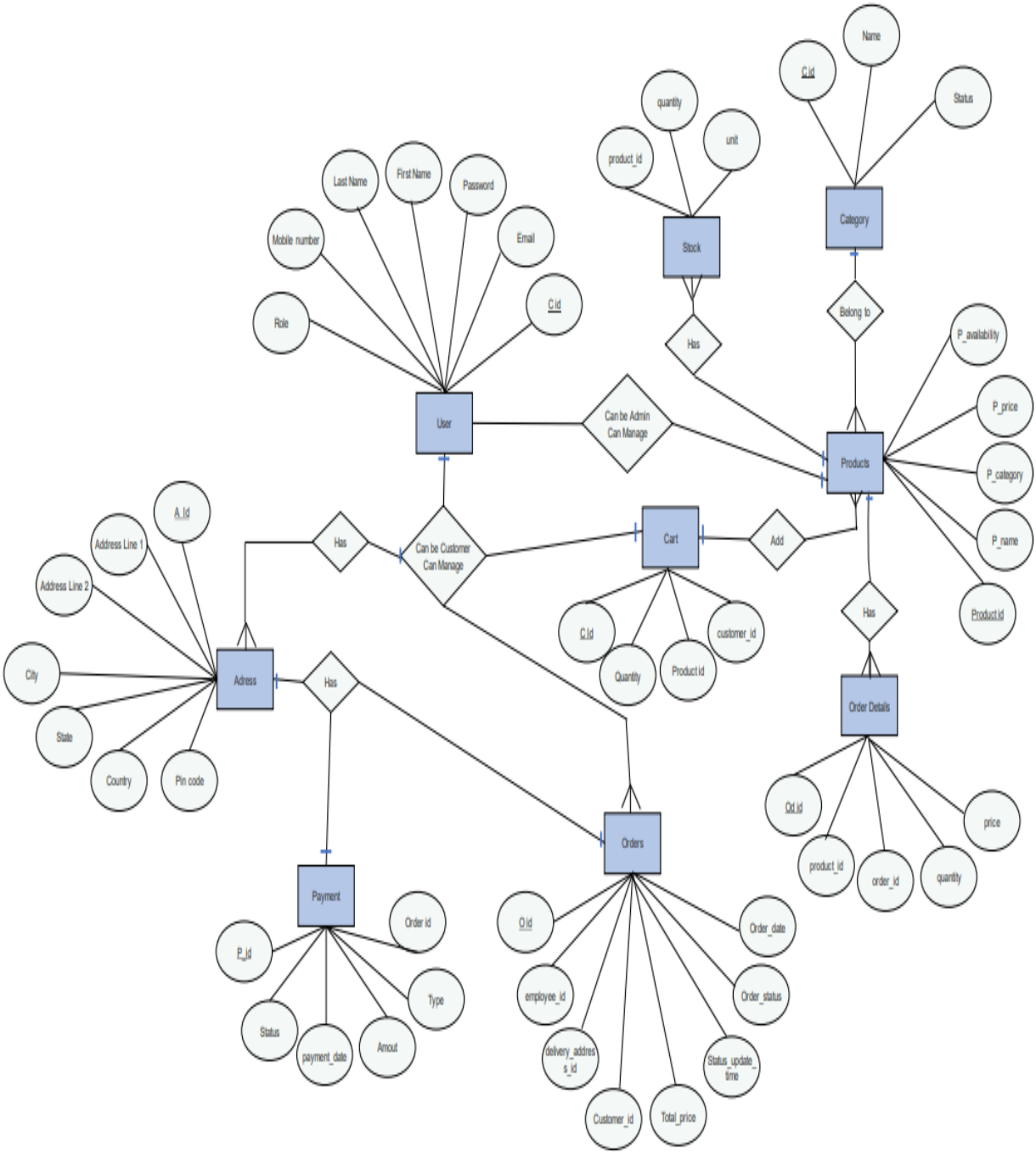
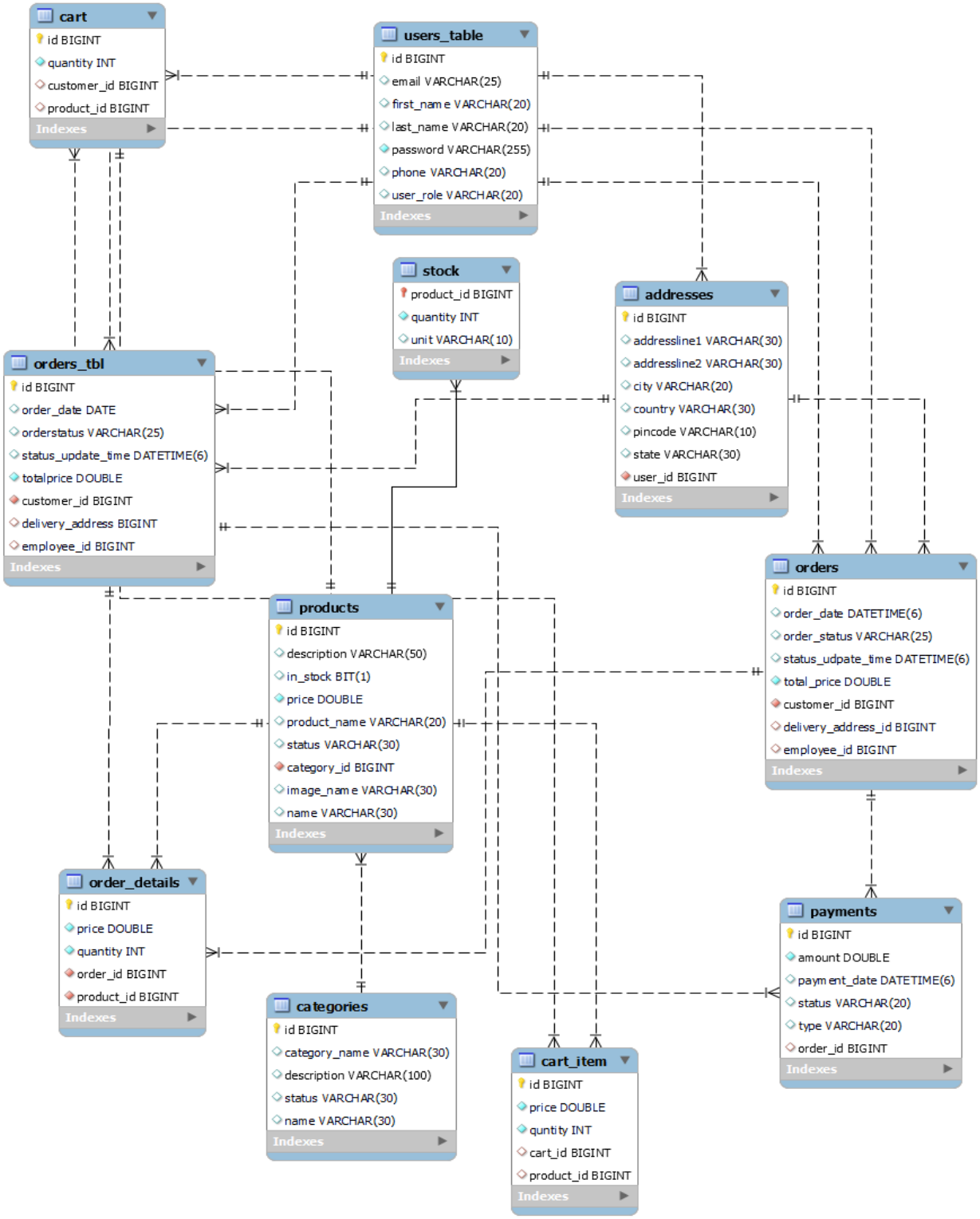


Figure 6 1 Level DFD for CUSTOMER

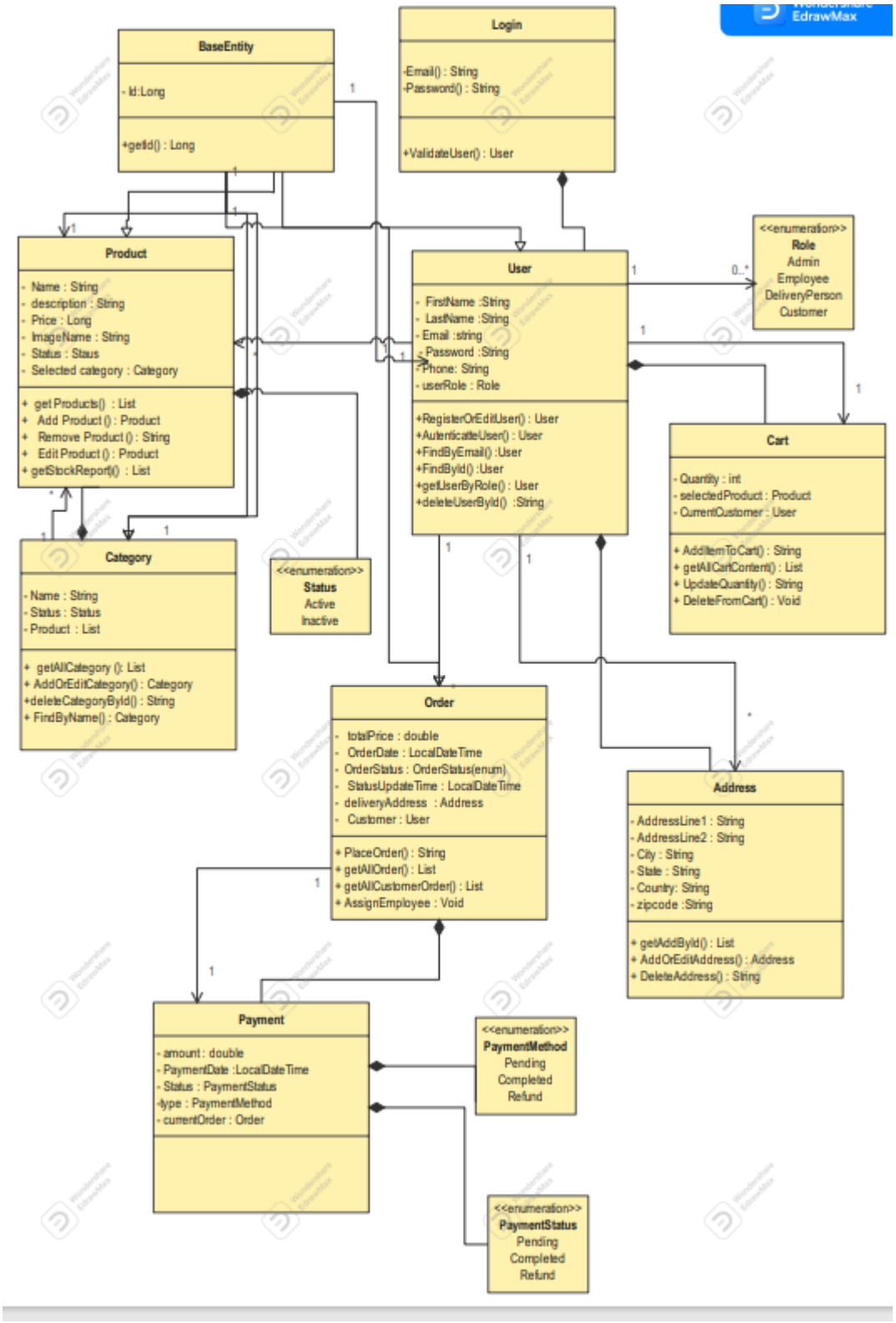
E – R DIAGRAM


Name
C.id
Status
quantity
product_id
unit
Last Name
First Name
Password
Mobile number
Email
Role
C.id
Stock
Category
Belong to
Has
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
Payment
P_Id
Order id
Type
Status
payment_date
Amout
O.id
employee_id
delivery_address_id
Customer_id
Total_price
Order_date
Order_status
Status_update_time
Od.id
product_id
order_id
quantity
price



System Generated E-R Diagram

CLASS DIAGRAM



ACTIVITY DIAGRAM

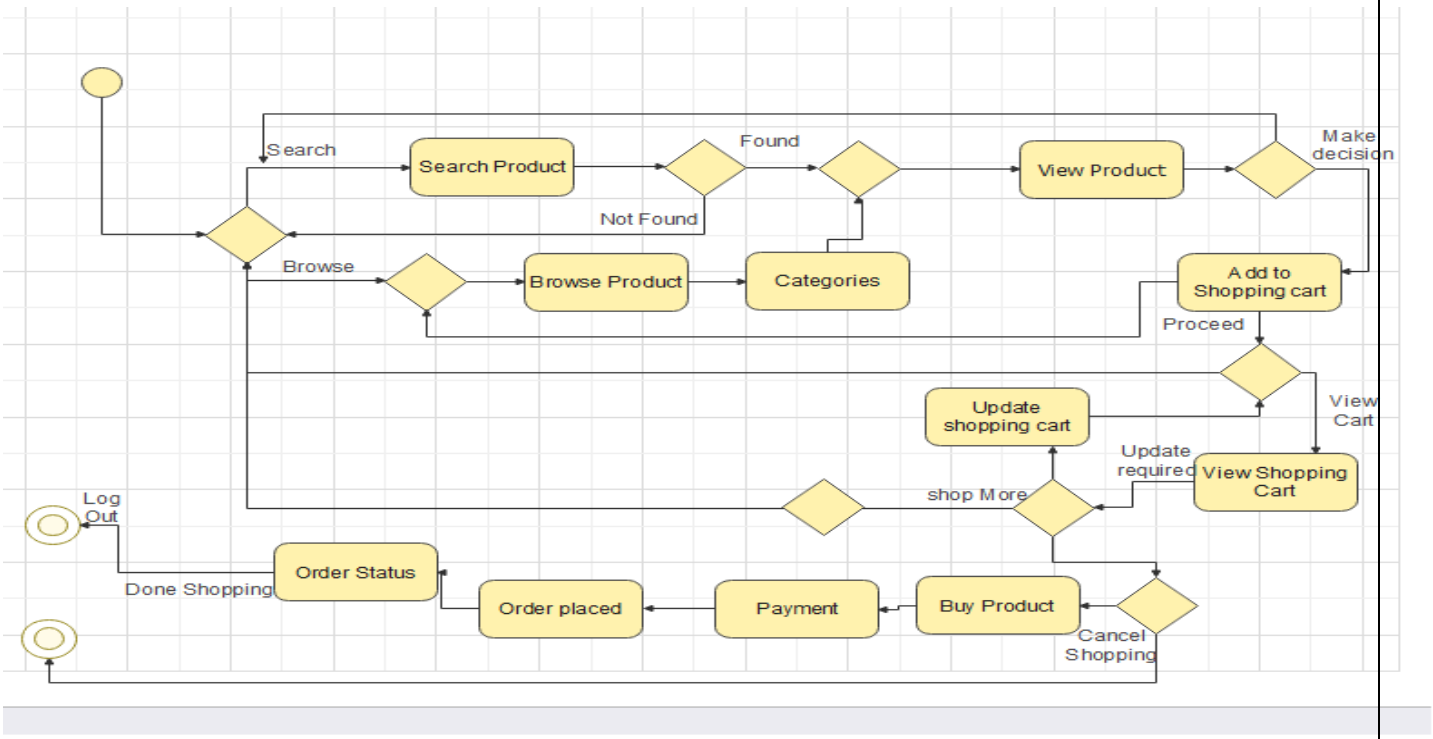


Figure: Activity Diagram

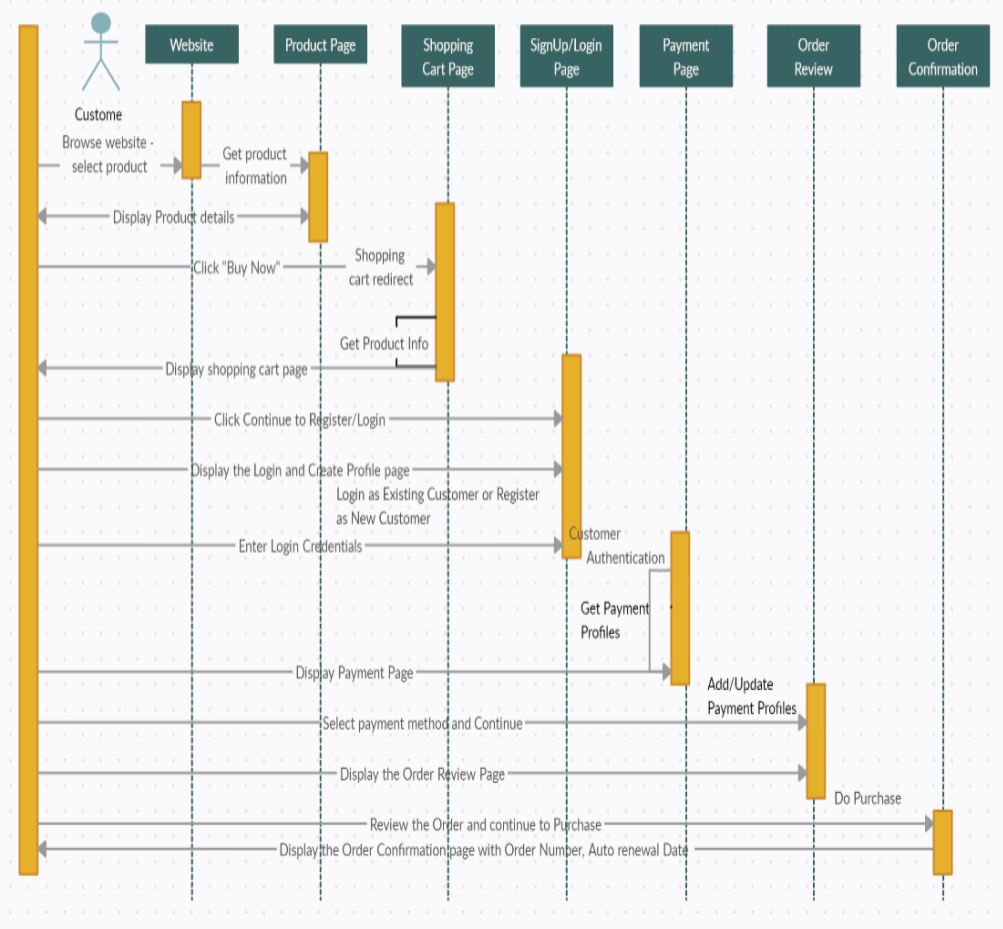


Figure: Sequence Diagram

TABLE STRUCTURE

Database: **gogrocers**

1)List of Tables:

```
mysql> use gogrocers;
Database changed
mysql> show tables;
+---------------------+
| Tables_in_gogrocers |
+---------------------+
| 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 (0.01 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 (0.00 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)
```

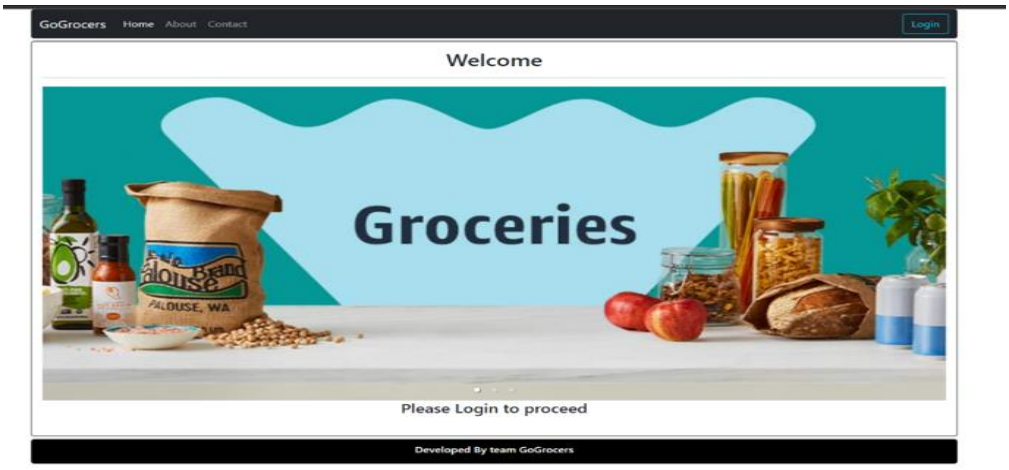
11)User table testing via Junit Test:

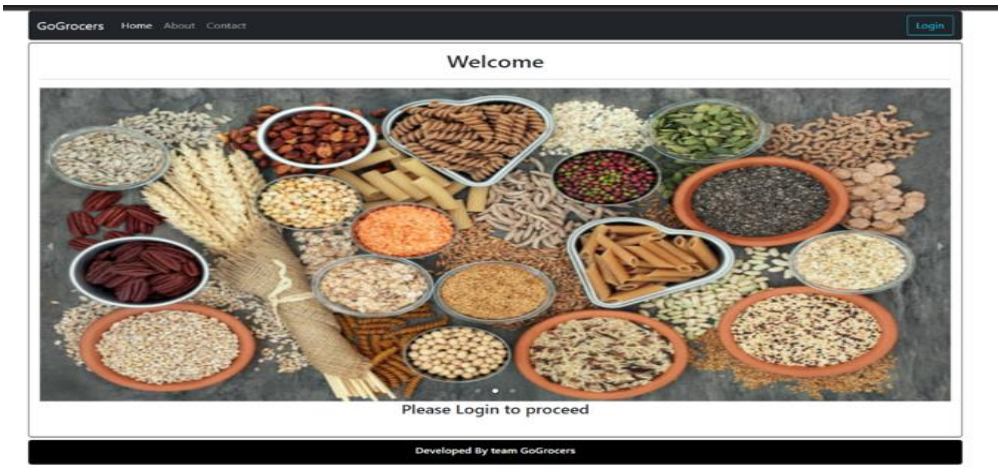
```
mysql> select*from users;
+----+-------------+------------+-----------+--------------------------------------------------------------+------------+-------+
| id | email       | first_name | last_name | password                                                     | phone      | role  |
+----+-------------+------------+-----------+--------------------------------------------------------------+------------+-------+
|  1 | k@gmail.com | Komal      | Ghorpade  | $2a$10$5XLhRWZRZnXkOVHbV2NjGu7udsjgyWH2wc8epWQOADxl1/hKfxNxq | 9028888888 | ADMIN |
|  2 | a@gmail.com | Ashwini    | Palve     | $2a$10$wUYixbCvWDny./wsPJWRCuMED/NVTVxdZ.jC.imSwhEmGq1PZpfOm | 7269999999 | ADMIN |
+----+-------------+------------+-----------+--------------------------------------------------------------+------------+-------+
2 rows in set (0.00 sec)
```

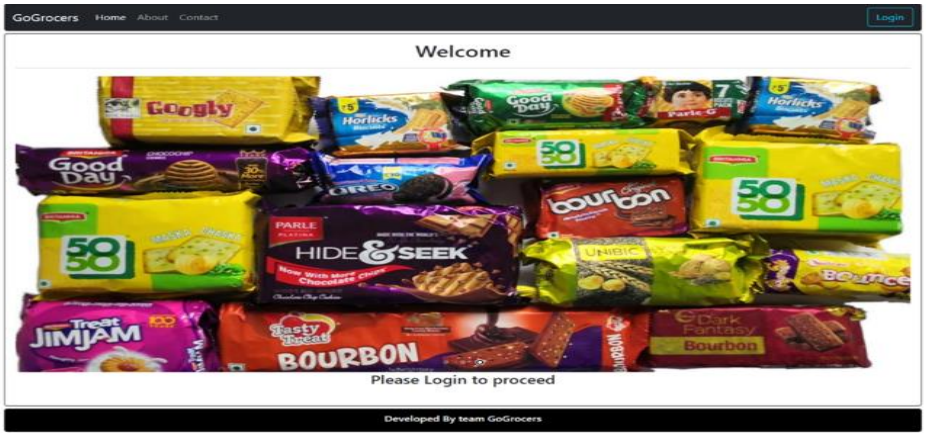
12) Categories table testing via Junit Test:

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

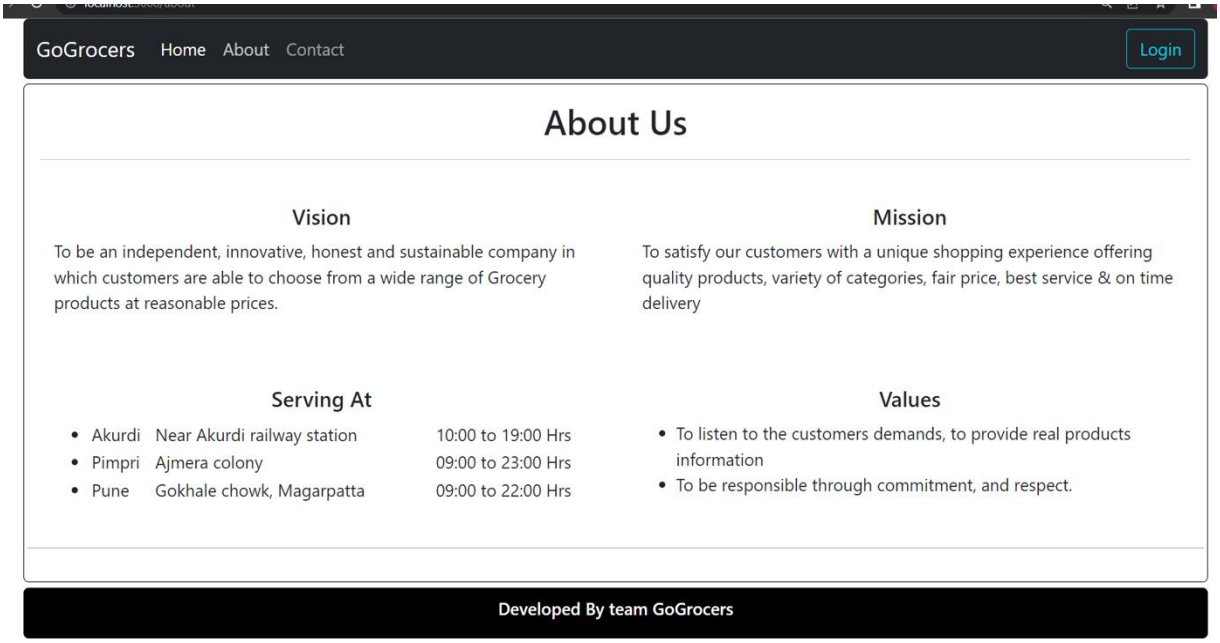
SCREENSHOTS



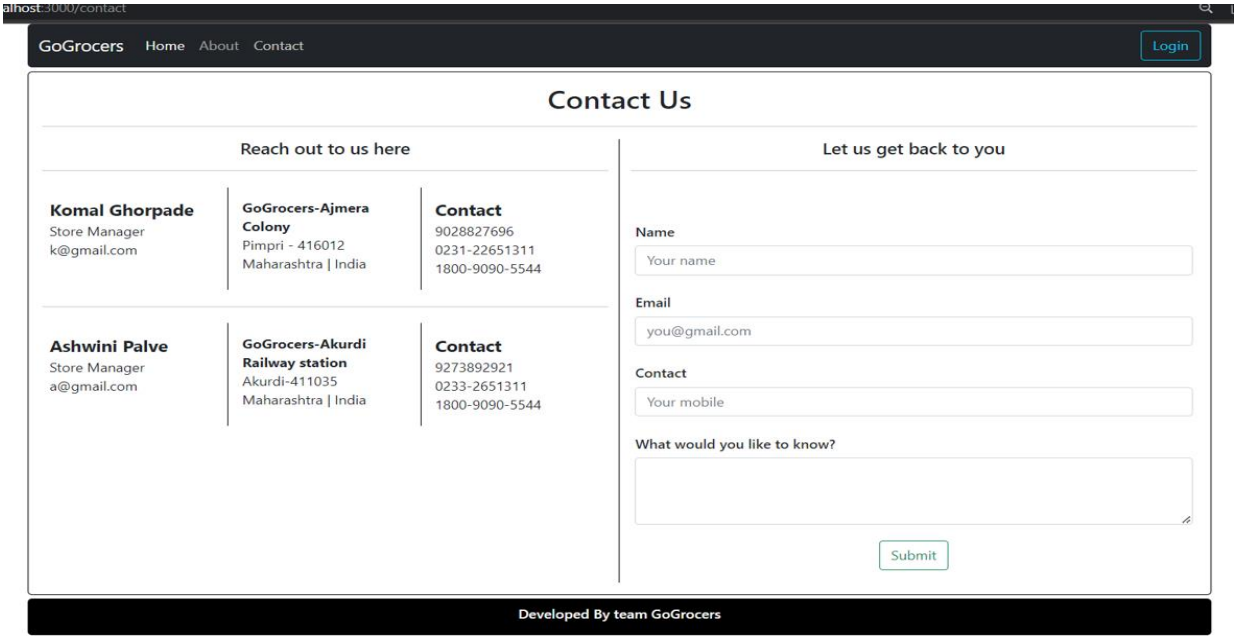




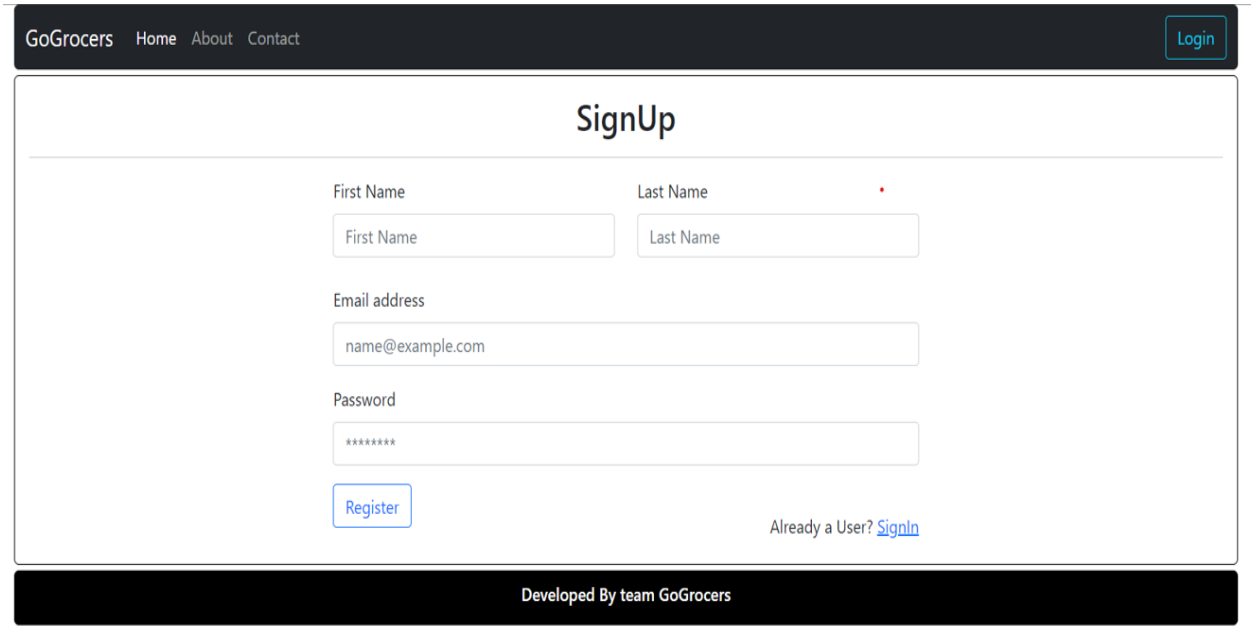
Go Grocers Home Page



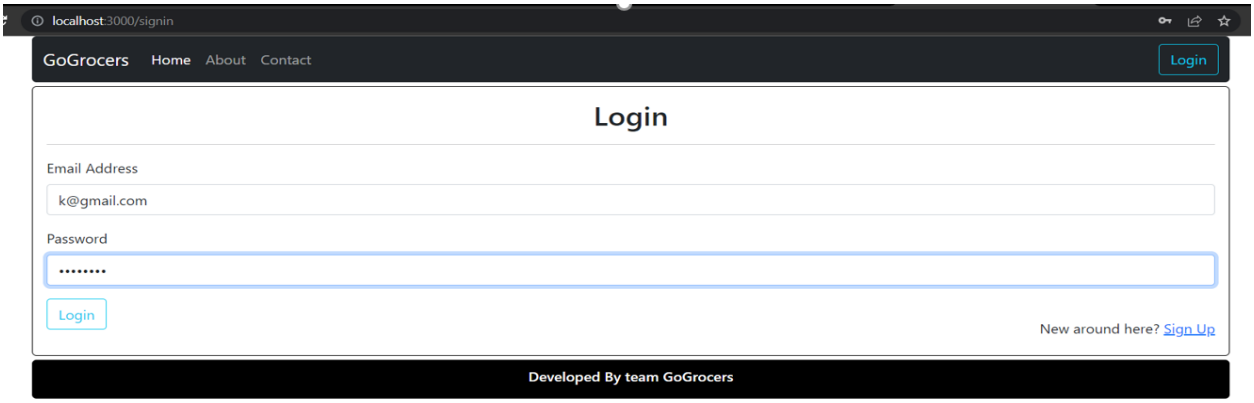
ABOUT US PAGE



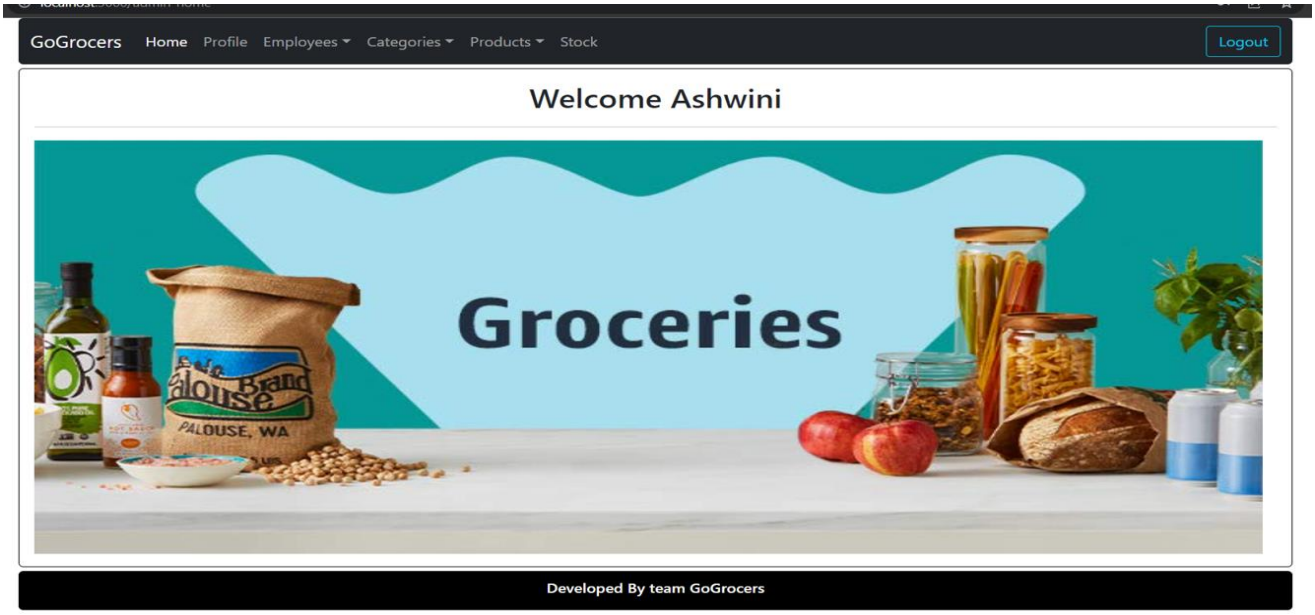
CONTACT US PAGE

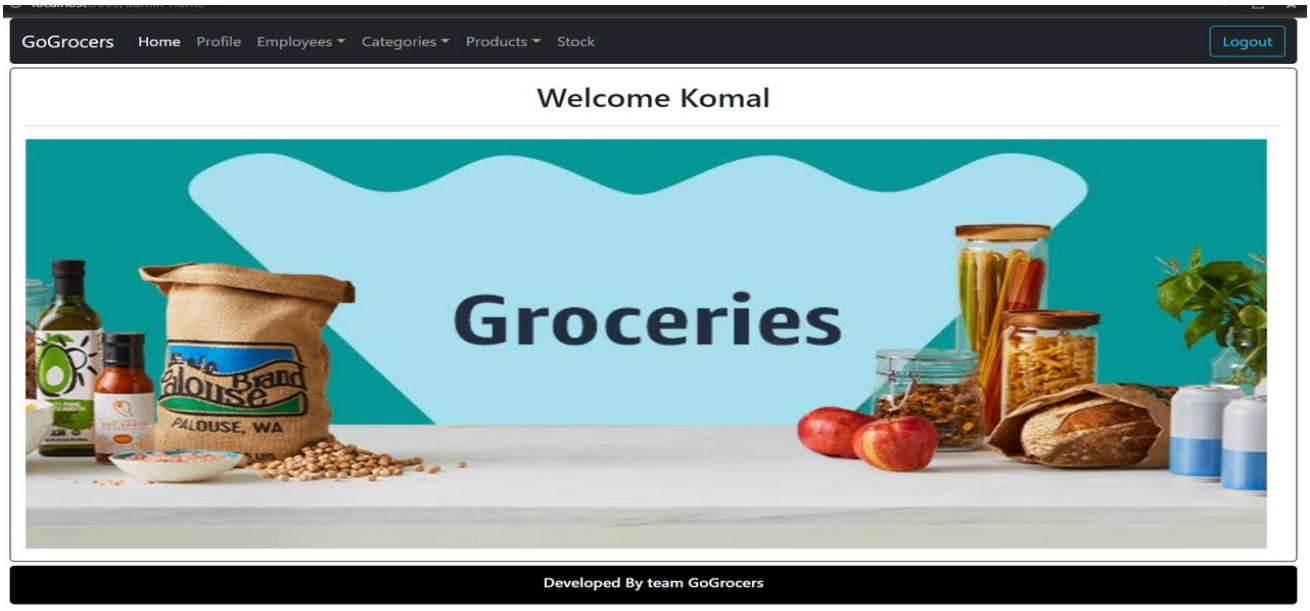


SIGN UP PAGE

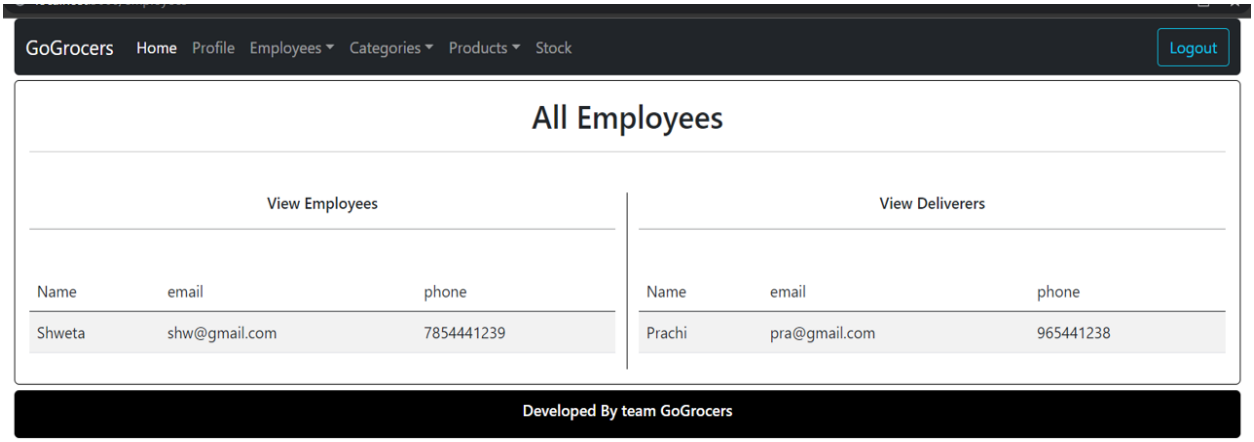


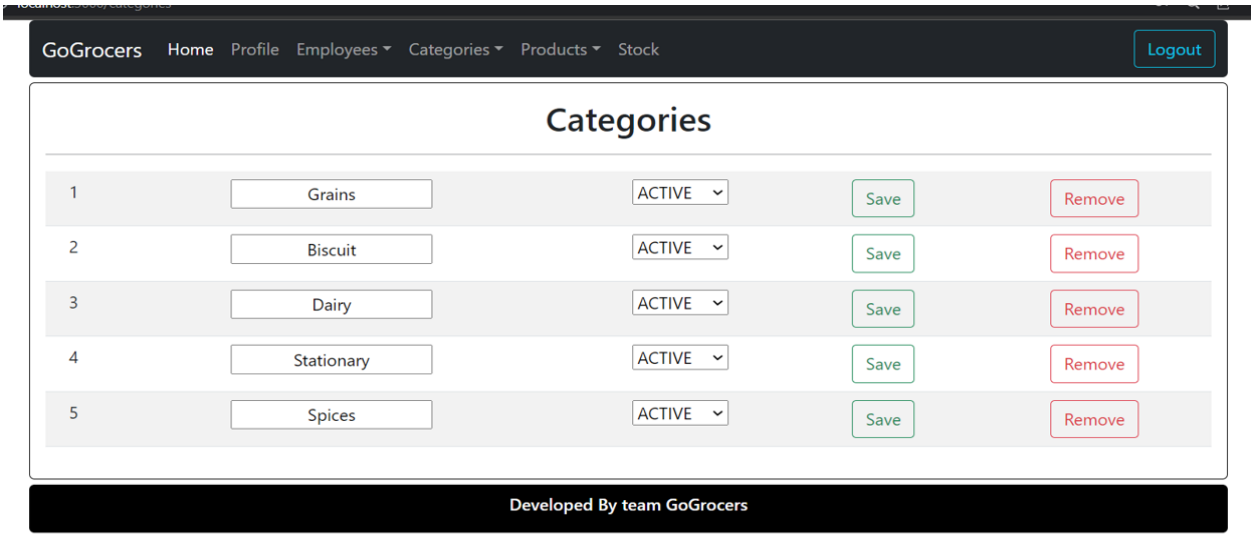
LOGIN PAGE


GoGrocers
Home
Profile
Employees
Categories
Products
Stock
Logout
Welcome Ashwini
Groceries
Developed By team GoGrocers

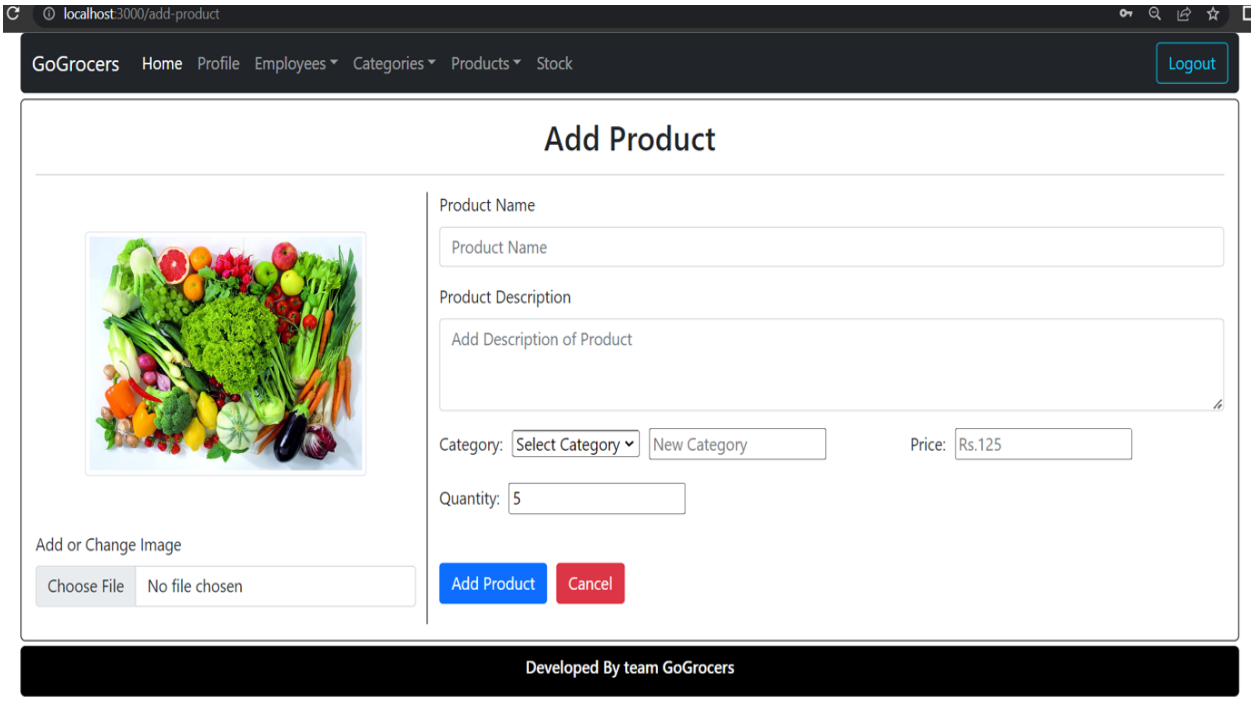

GoGrocers
Home
Profile
Employees
Categories
Products
Stock
Logout
Welcome Komal
Groceries
Developed By team GoGrocers

Admin Login

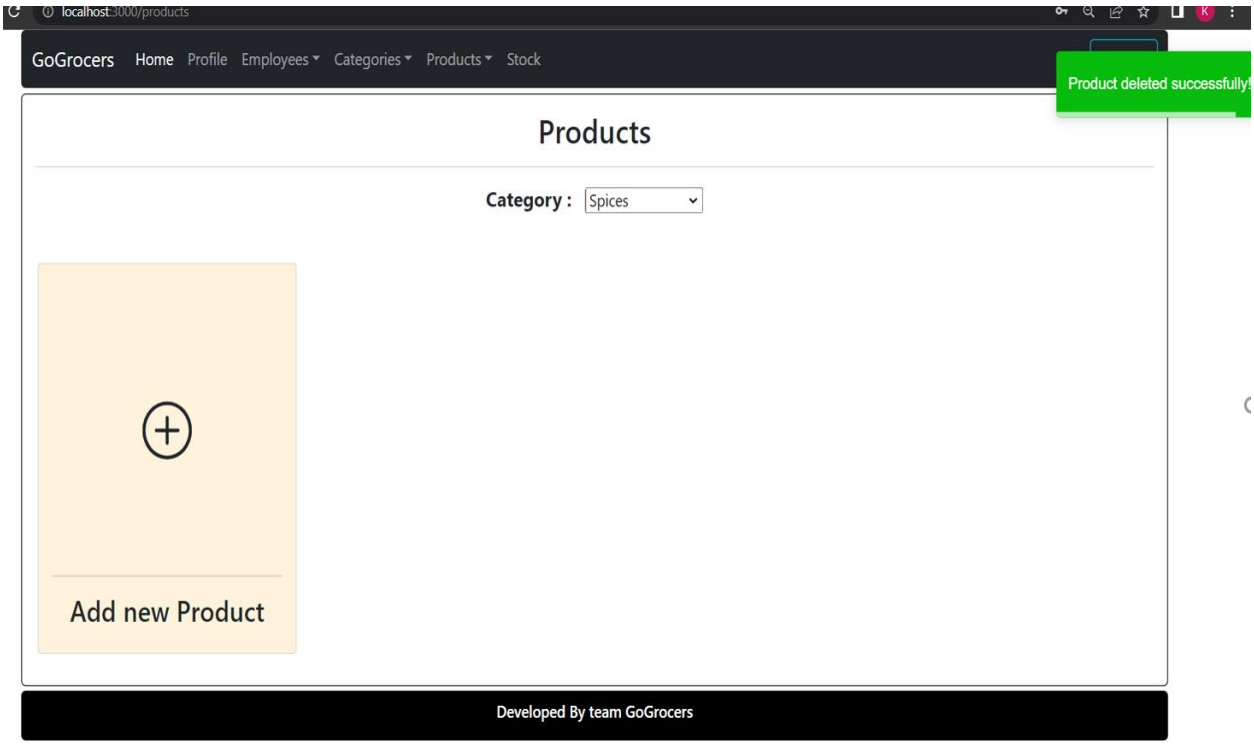




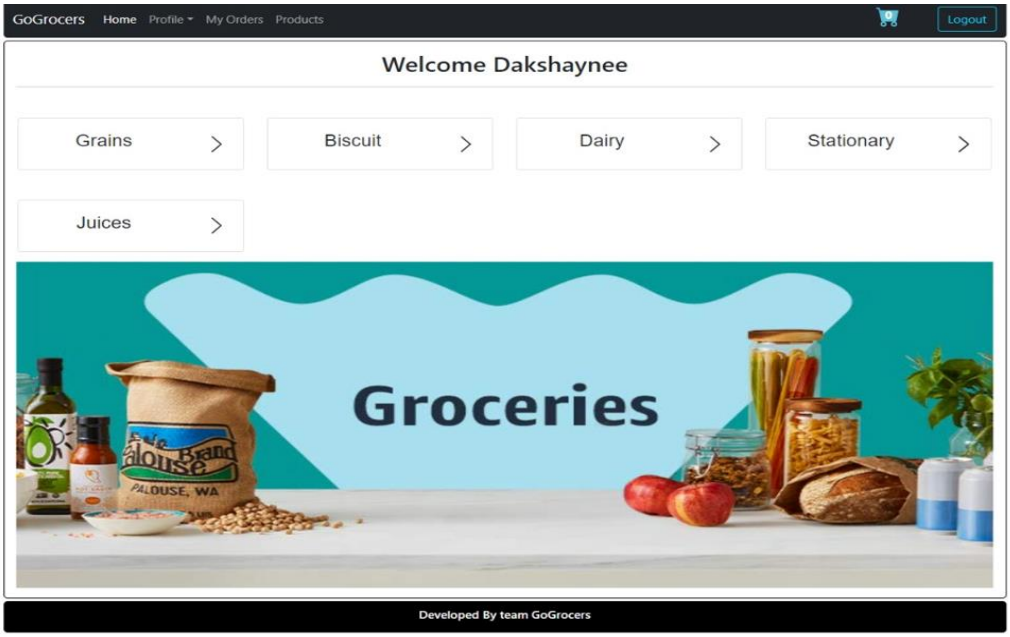
Employee Login

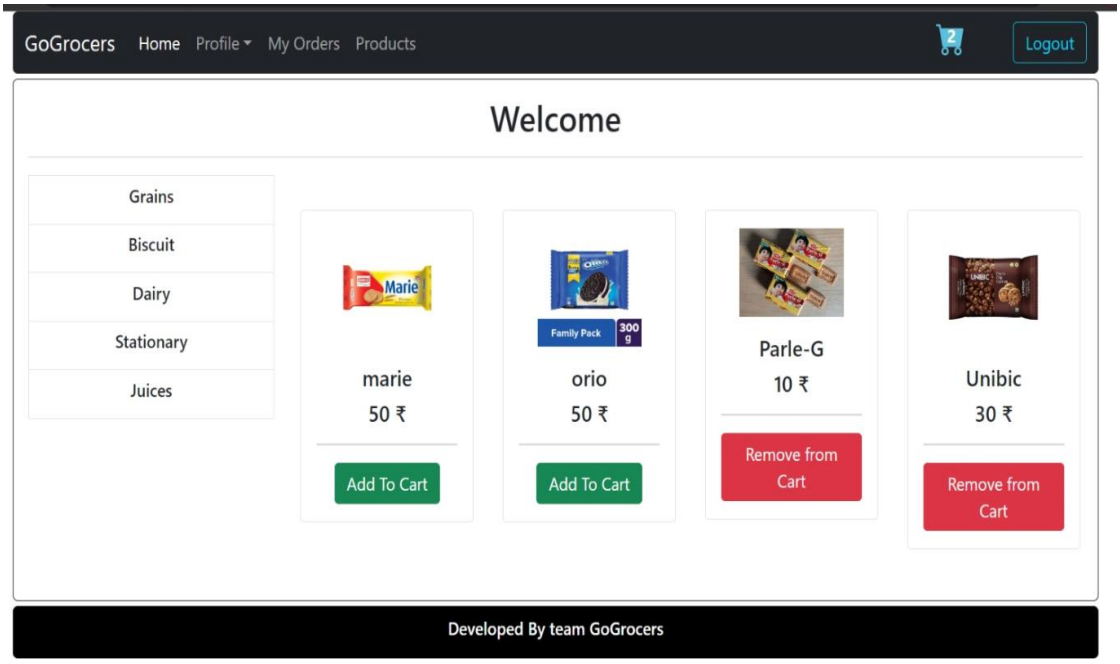


Add Product by Employee

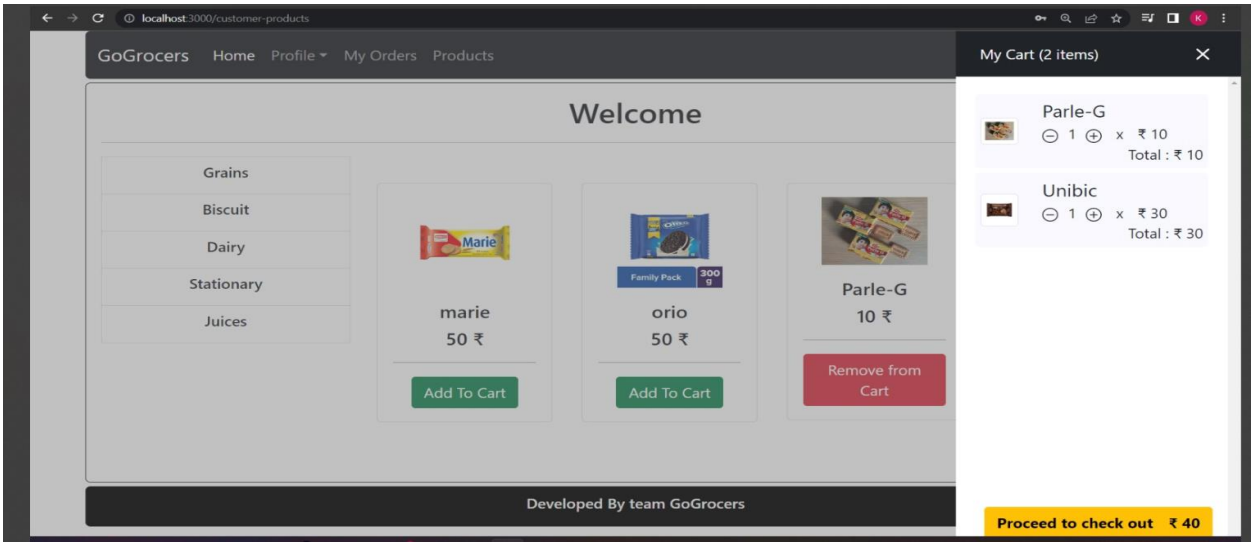


Product Deleted by Employee

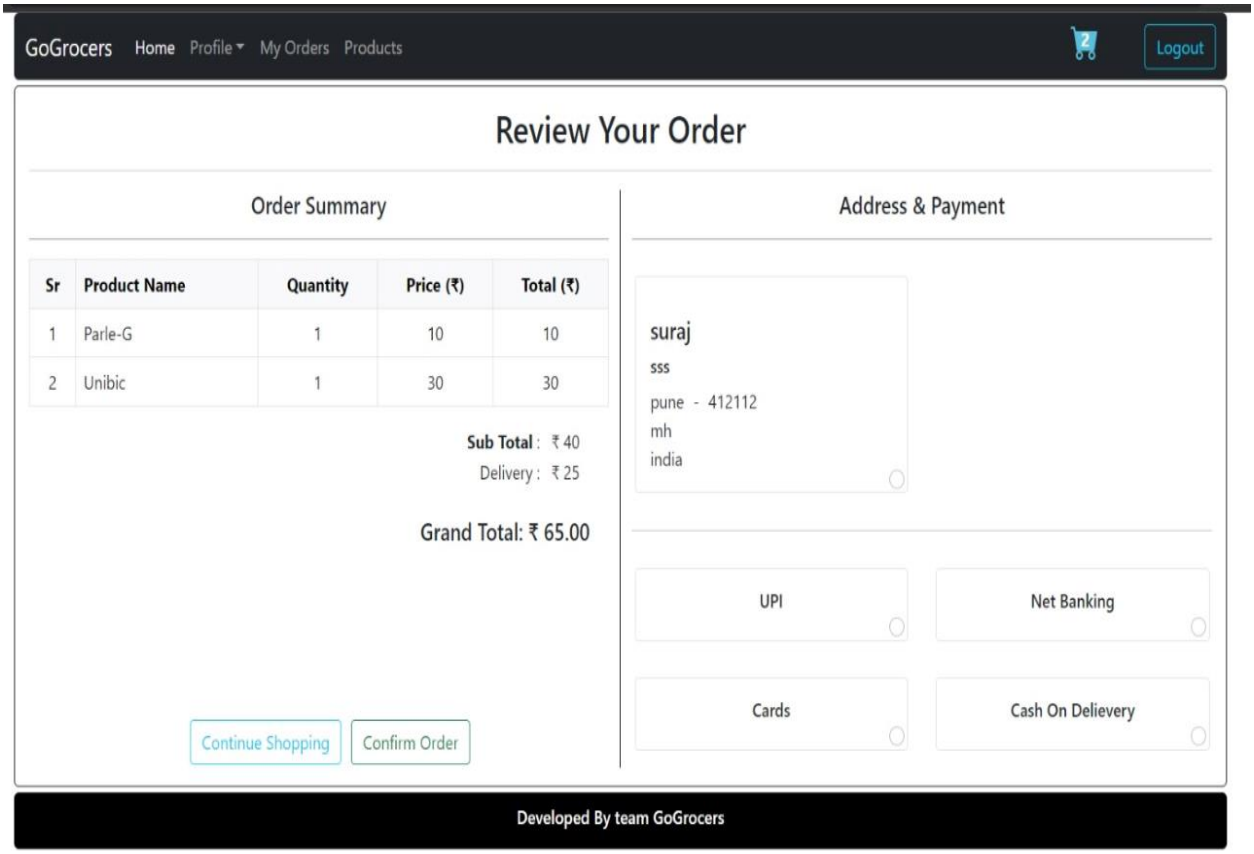




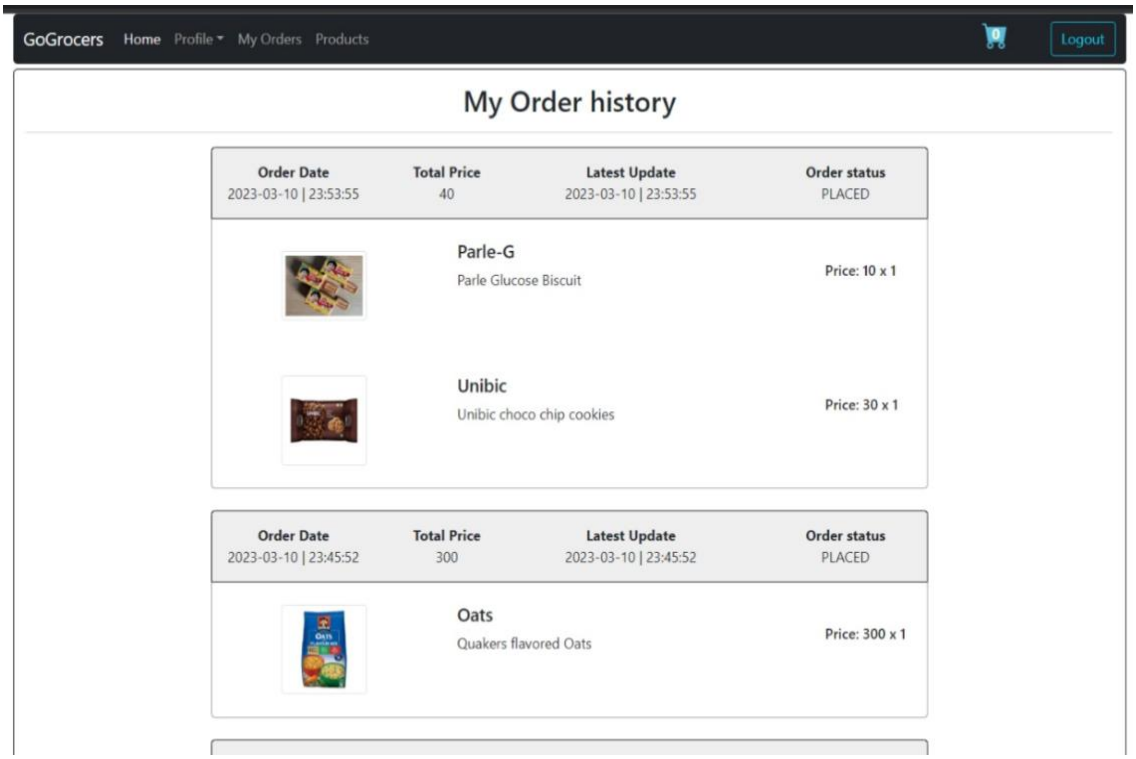
Customer Login



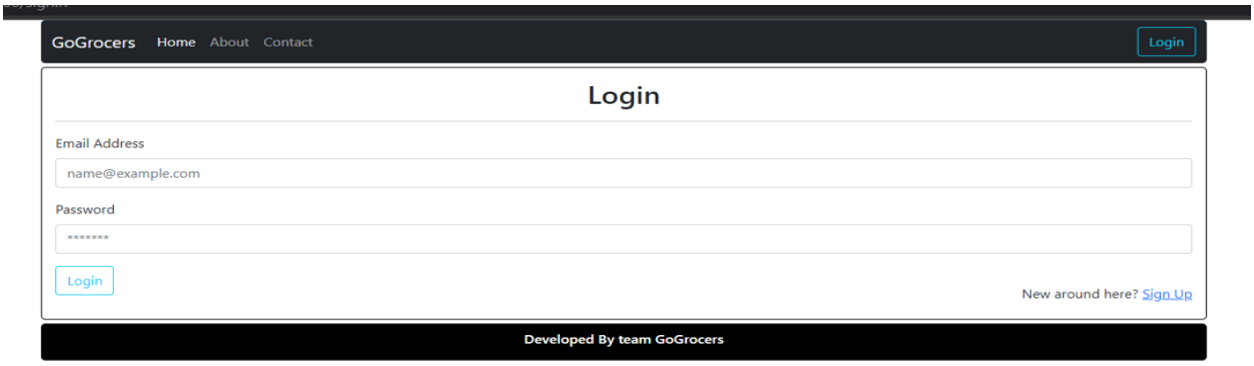
Product Added to Cart



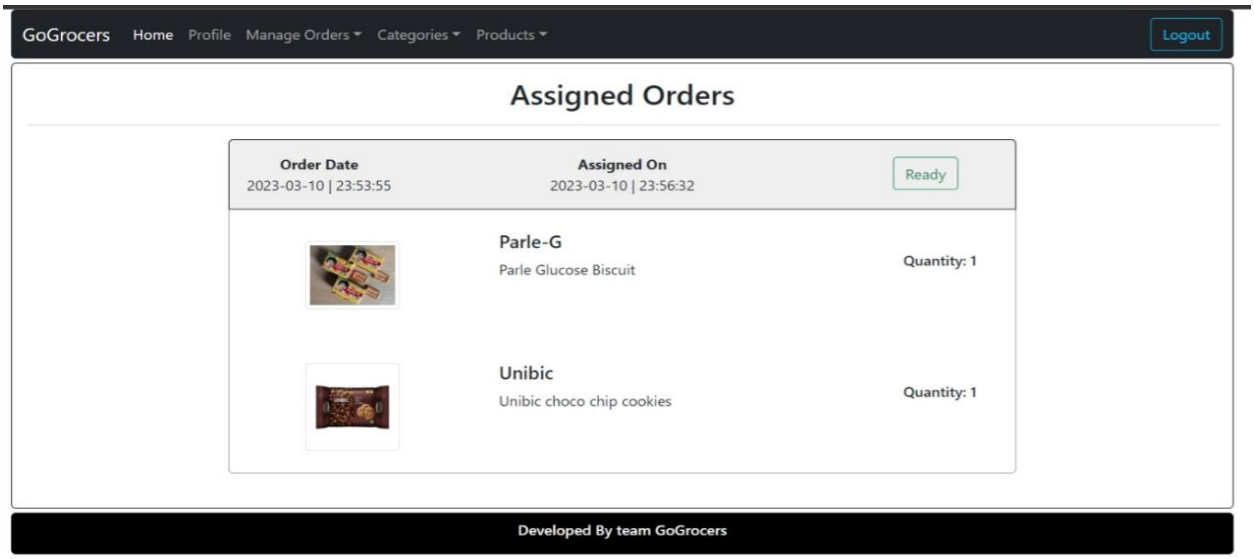
Customer Order details

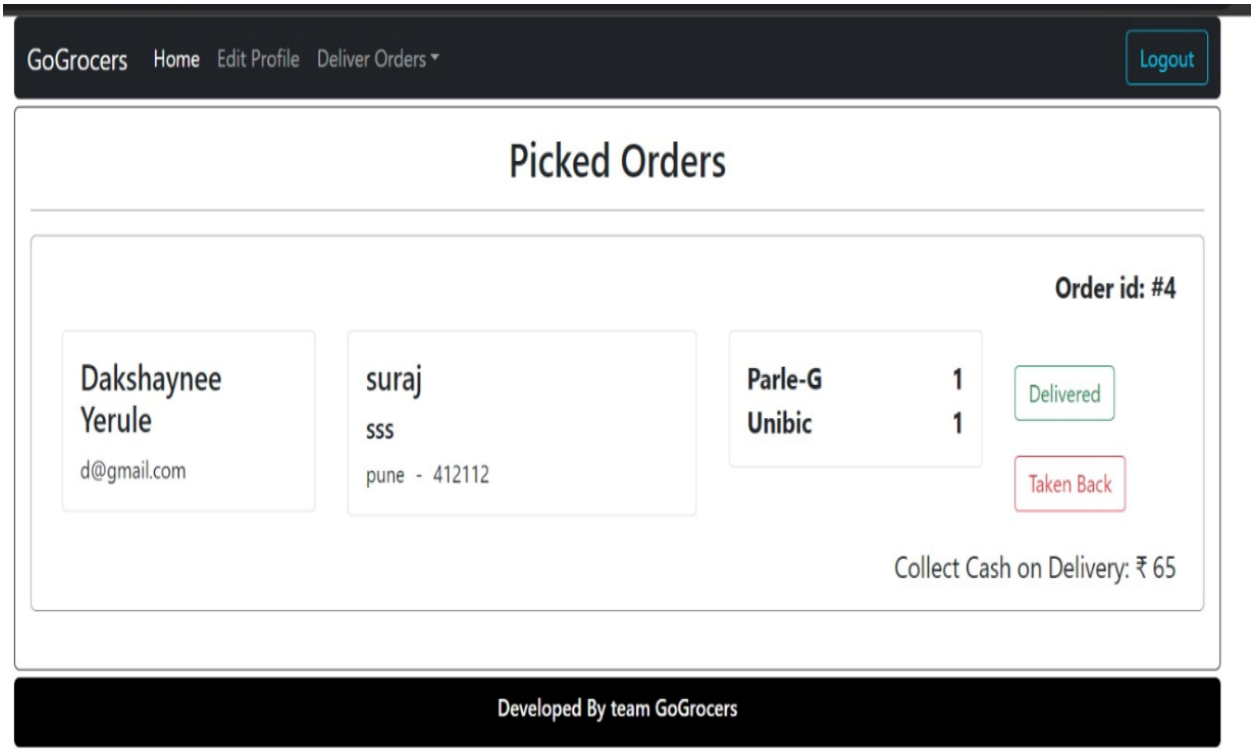


Customer Order history



Delivery Person Login





Delivery Person Logi

CONCLUSION

The project entitled **"GoGrocers" – Online Grocery Shopping** App was completed successfully.

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming.

The purpose of this project was to develop a web application and an android application for purchasing items from a shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using React.js, usage of responsive templates, designing of android applications, and management of database using MySQL. The entire system is secured. Also, the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications.

There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limit us .

REFERENCES

[1] JavaScript Enlightenment, Cody Lindley-First Edition, based on JavaScript 1.5, ECMA-262, Edition

[2] Mc Graw Hill's, Java: The complete reference 7thEdition, HerbertScheldt

[3] Complete CSS Guide, Maxine Sherrin and John Allsopp-O'ReillyMedia; September 2012

ONLINE REFERENCE :

[1] www.Google.com

[2] www.w3school.com

[3] www.javatpoint.com