VELOCITY CAPSTONE PROJECT

GROUP-1

“Shopping Cart”

Presented By-

Aishwarya Agarwal
Apurva Upasani
Rajan Baviskar
Sadiq Basha Shaik

Mentor: Kamal Iyer

Stream: Cloud-AWS

Partner: StackRoute

TABLE OF CONTENTS

Project Users and Admin Stories :

The project is planned to be completed in 2 sprints. Tasks assumed to be completed in the sprint are: Creating the flow of the application and Initializing the git repository to track changes as development progresses. Writing the Java program to fulfill the requirements of the project.

User Stories:

1. As a user, I should be able to login, Logout, and Register into the application.
2. As a user, I should be able to see the products in different categories.
3. As a user, I should be able to sort the products.
4. As a user, I should be able to add the products into the shopping cart.
5. As a user, I should be able to increase or decrease the quantity added in the cart.
6. As a user I should be able to add "n" number of products in the cart.
7. As a user I should be able to get the Wishlist option where I can add those products which I want but don't want to order now.
8. As a user I should get different discount coupons.

Admin Stories:

1. As an Admin I should be able to login, Logout, and Register into the application.
2. As an Admin I should be able to perform CRUD on Users.
3. As an Admin I should be able to perform CRUD on the products.
4. As an Admin I should be able to get a bulk upload option to upload a CSV for product details.
5. As an Admin I should be able to get the stocks.
6. As an Admin I should be able to mail if any stock is less than 10.
7. As an Admin I should be able to get the sales report of a specific duration.
8. As an Admin I should be able to set the discount coupons for the specific set of user

COMPONENTS OF THE PROJECT:

Any full-stack application can be divided into three parts that are mentioned below;

- **The Frontend**
- **The Backend**
- **The Database**

E-COMMERCE WEBSITE:

A website that allows people to buy and sell physical goods, services, and digital products over the internet rather than at a brick-andmortar location. Through an e-commerce website, a business can process orders, accept payments, manage shipping and logistics, and provide customer service.

CREATING AN E-COMMERCE WEBSITE:

To create an e-commerce website it is important to learn or have knowledge of the tools, and installations to be made for creating the website.

OUR APPROACH IN CREATING THE WEBSITE:

For creating the SHOP FROM HOME e-commerce website ANGULAR has been used for creating the front-end part. We have also taken the help of Boot strap for the frontend.

Similarly for the backend of the project the code has been done using the JAVA language in SPRING-BOOT. Maven has also been installed and used.

The Database is the part where the tables are being created and all the required data is being stored. For the database POSTGRESSQL has been installed and used.

CORE COMPONENTS:

Technologies:

- Angular
- Java Spring Boot,
- PostgresSQL

Languages:

- Type Script
- Java
- SQL Queries

IDE:

VS Code:

As an IDE to design the frontend of the application

Git:

To connect and push files from the local system to GitHub

GitHub:

To store the application code and track its versions

Back End:

Eclipse IDE. Java Programming. Searching and Sorting Spring Boot DevTools. Spring Web and Spring Data JPA.

DevOps:

Jenkins. Docker and TestNG.

AWS Cloud:

EC2, Load Balancers, Auto Scaling, S3 Buckets.

HARDWARE REQUIREMENTS:

Operating System:

- Windows 7/8/10/11
- Linux distros

- MacOS X or later.

Processor:

- Intel or AMD dual core x86 processor.

Ram:

- 2 GB or above.

Hard disk:

- 500 MB of free disk space or more.

THE FRONTEND:

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

Using Angular we have created different components like

Features of the application:

1. Registration
2. Login
3. Payment gateway
4. Searching

5. Filtering

6. Sorting

7. Dynamic data

8. Responsive and compatible with different devices

Recommended technologies:

1. Database management: SQL and PostgreSQL

2. Backend logic: Java programming, Angular

3. Frontend development: Angular, Bootstrap, HTML/CSS, and Javascript

4. DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS.

Project development guidelines:

The project will be delivered within two sprints with every sprint delivering a minimal viable product. Performing sprint planning with user stories is mandatory to develop all project components. The learner can use any technology from the above-mentioned technologies for different layers of the project. The web application should be responsive and should fetch or send data dynamically without hardcoded values. The learner must maintain the application's version over GitHub and every new change should be sent to the repository. The learner must implement a CI/CD pipeline using Jenkins. The learner should also deploy and host the application on an AWS EC2 instance. The learner should also implement automation testing before the application enters the CI/CD pipeline. The learner should use Git branching to do basic automation testing of the application in it separately. The learner should make a rich frontend of the application, which is userfriendly and easy for the user to navigate through the application. There will be two portals in the application, namely the admin and user portal.Each component contains the HTML file, Type Script file, the CSS file and spec file for testing purposes.

STUDY OF THE SYSTEM:

Modules:

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

The modules involved are:

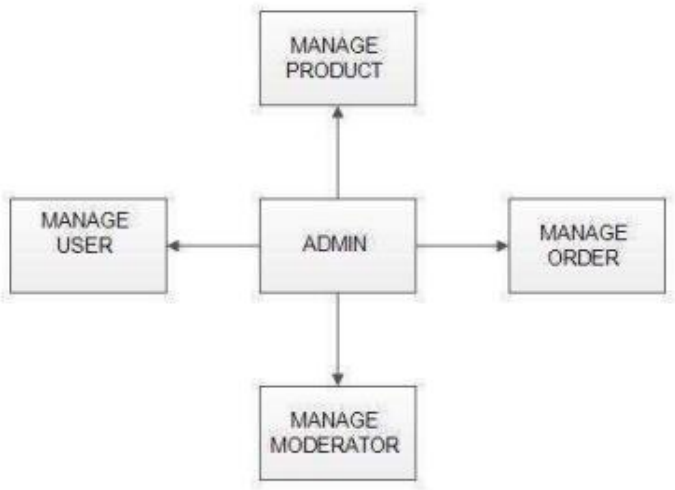
- Admin
- Users

Admin Portal:

The administrator is the super user of this application. Only the admin has access to this admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about all products. The admin user should be able to: Add or remove different products to or from the application to build a rich product line Edit product item details like name, price, discount, and offers to keep it aligned to the current prices Enable or disable the product items

This module is divided into different sub-modules.

1. Manage Moderators
2. Manage Products
3. Manage Users
4. Manage Orders



Admin module

Manage Moderators :

● **Add Moderator**: Only the admin is having the privilege to add a moderator. A moderator can be considered as a staff who manages the orders or owner of a group of products.

● **Remove Moderator:** Admin has the privilege to delete a moderator who was added.

● **Search moderator:** All existing moderators can be viewed by the administrator as a list. If there is a number of moderators and the admin needs to find one of them, the admin can search for a moderator by name.

Manage Users:

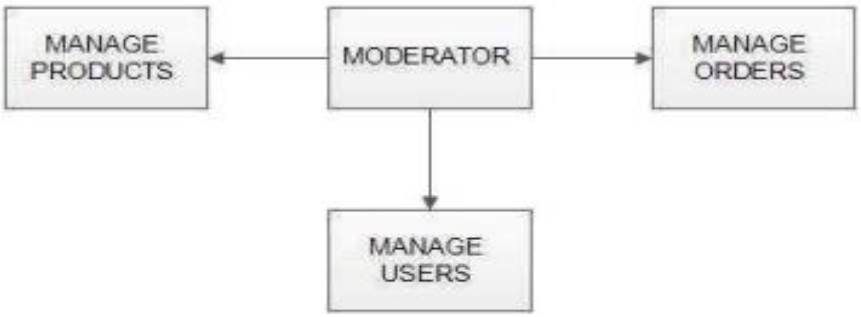
● **View Users:**

The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list except the password.

● **Add Users:**

Admin has privileges to add a user directly by providing the details.

Moderators:



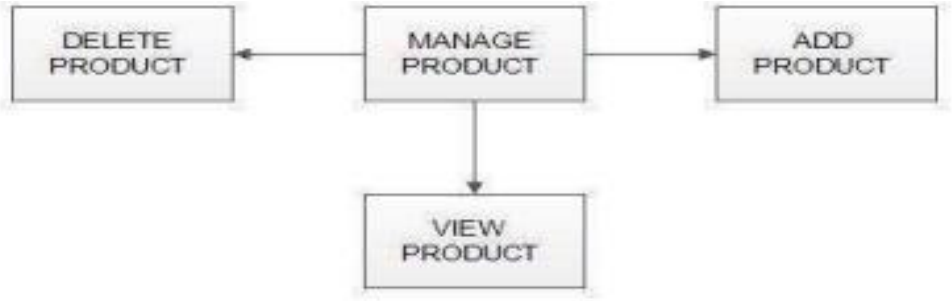
Moderator Module

A moderator is considered a staff who can manage orders for the time being. As a future update moderator may give the facility to add and manage his own products. Moderators can reduce the workload of admin. Now moderator has all

the privileges an admin has except managing other moderators. He can add products and users. He can also check the orders and edit his profile.

- Manage products
- Manage users
- Manage orders

Manage Products:



Manage Products

- **Add Products:**

The shopping cart project contains a different kinds of products. 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**:

Administrators 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.

Manage Users:

- **View Users:** The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list except the password.
- **Add Users:** Admin has privileges to add a user directly by providing the details.

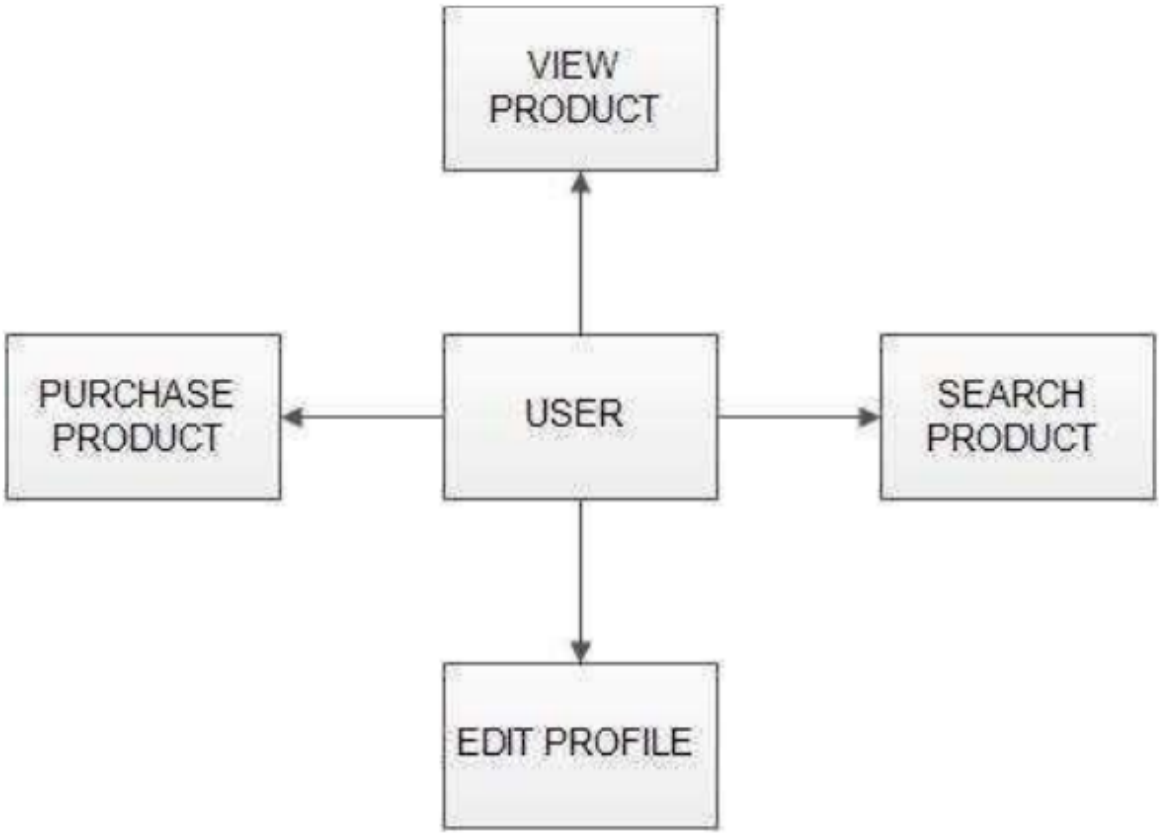
Manage Orders:

Manage Orders

● **View Order:**
Administrators can view the Orders which is generated by the users. He can verify the details of the purchase.

● **Delete Order:**
Admin can delete orders from the orders list when the product is taken for delivery.

Users Portal:
It deals with user activities. The end-user should be able to: Sign-in to the application to maintain a record of activities and Search for product items based on the search keyword. Add all the selected product items to a cart and customize the purchase at the end Perform a seamless payment process Get an order summary details page once the payment is complete



User Module

Registration:

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 a new user by unblocking him.

Login:

A user must login with his user name and password to the system after registration.

View Products:

Users. can view the list of products based on their names after successful login. A detailed description of a particular product with product name, product details, product image, and price can be viewed by users.

Search Product:

Users can search for a particular product in the list by name.

● **Add to cart:**

The user can add the desired product to 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 the cart can be viewed in the cart. Users can remove an item from the cart by clicking remove.

● **Submit Cart:**

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.

● **History:**

In the history, the user will have a view of pending orders.

● **Edit Profile**:

The user can view and edit the profile.

THE BACKEND:

The backend of the project has been created using JAVA

SPRINGBOOT.

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.All the classes has been created in the .java files in separate folders

i.e: enum, entity, api, repositories etc.

The backend folder also consist of docker file and .sql file where the database has been imported.

DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of the clarifying system requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to be an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons, DFD's are often referred to as logical data flow diagrams.

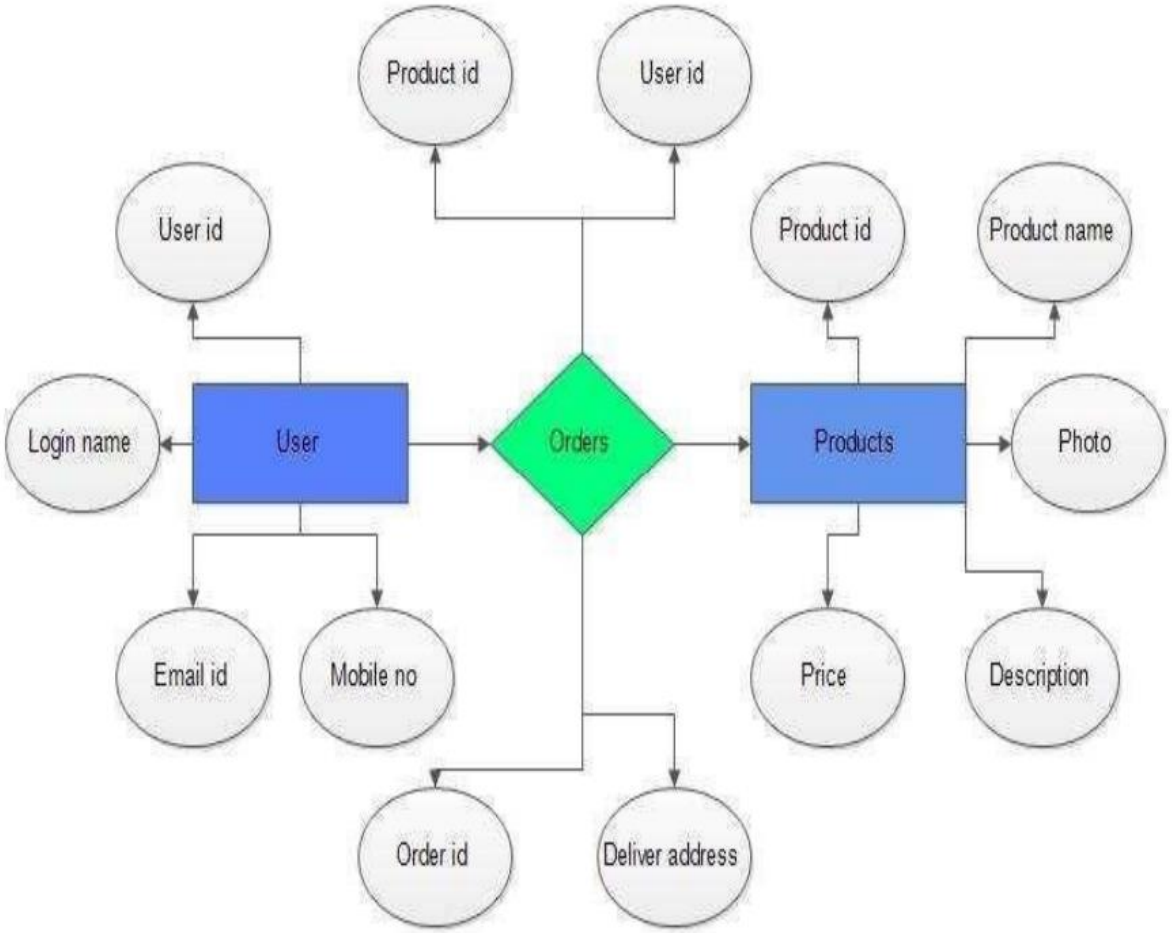
THE DATABASE:

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department. POSTGRES pioneered many concepts that only became available in some commercial database systems much later.

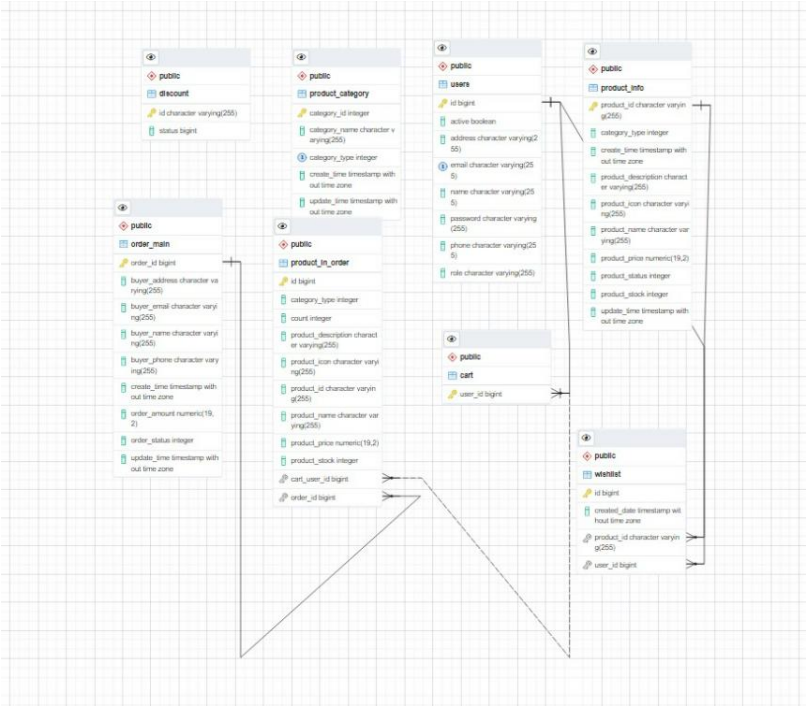
We have created tables in PostgresSQL and Inserted all the data into the tables. The names of the tables are;

- Cart
- Discount
- Order_main
- Product-category

- Product_in_order
- Product_info
- Users
- Wishlist

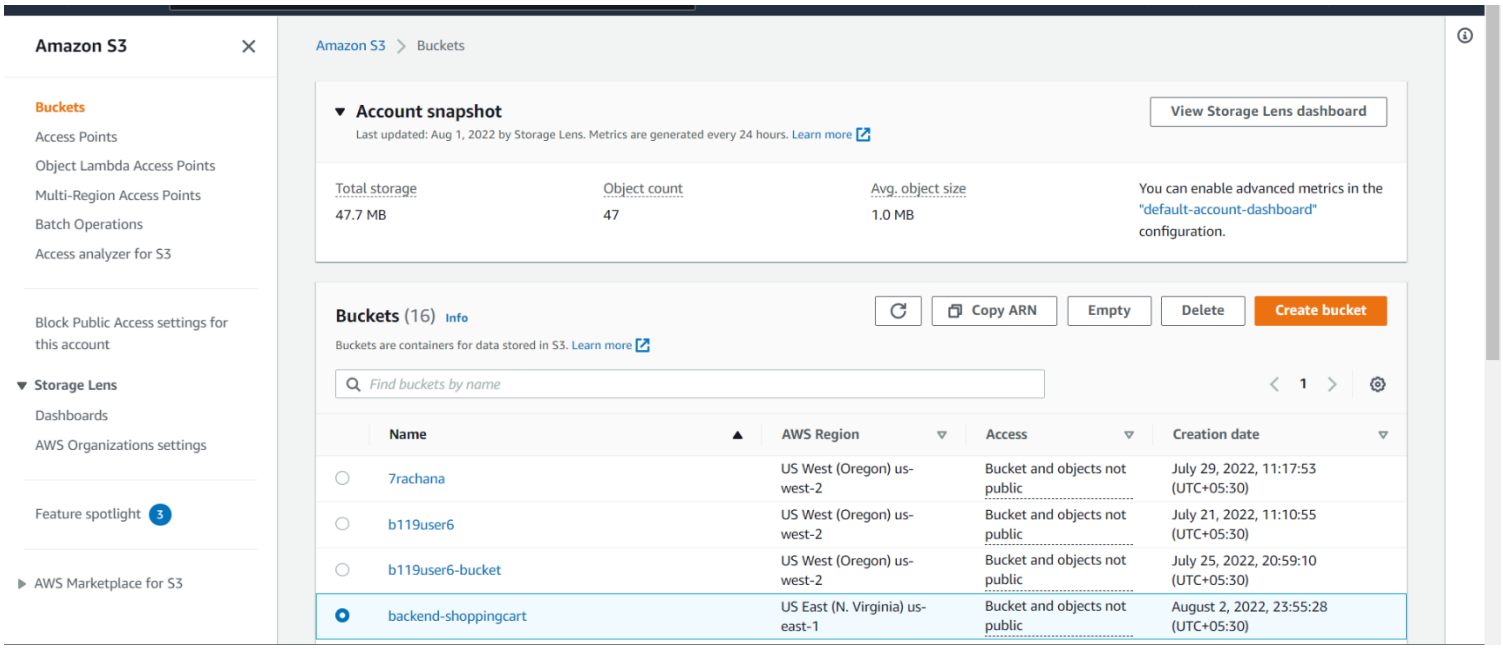


DATA SCHEMA

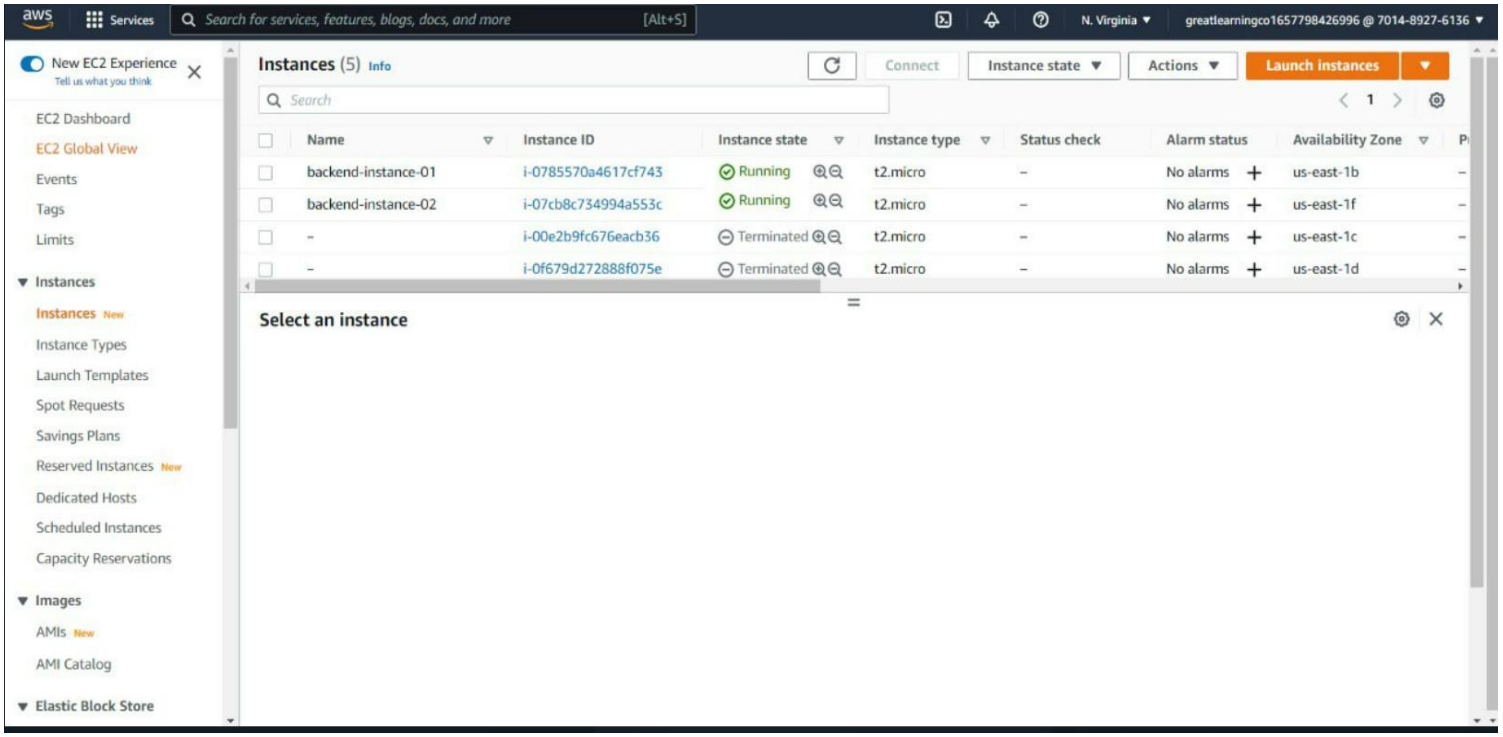

public
discount
id character varying(255)
status bigint
public
product_category
category_id integer
category_name character varying(255)
category_type integer
create_time timestamp without time zone
update_time timestamp without time zone
public
users
id bigint
active boolean
address character varying(255)
email character varying(255)
name character varying(255)
password character varying(255)
phone character varying(255)
role character varying(255)
public
product_info
product_id character varying(255)
category_type integer
create_time timestamp without time zone
product_description character varying(255)
product_icon character varying(255)
product_name character varying(255)
product_price numeric(19,2)
product_status integer
product_stock integer
update_time timestamp without time zone
public
order_main
order_id bigint
buyer_address character varying(255)
buyer_email character varying(255)
buyer_name character varying(255)
buyer_phone character varying(255)
create_time timestamp without time zone
order_amount numeric(19,2)
order_status integer
update_time timestamp without time zone
public
product_in_order
id bigint
category_type integer
count integer
product_description character varying(255)
product_icon character varying(255)
product_id character varying(255)
product_name character varying(255)
product_price numeric(19,2)
product_stock integer
cart_user_id bigint
order_id bigint
public
cart
user_id bigint
public
wishlist
id bigint
created_date timestamp without time zone
product_id character varying(255)
user_id bigint

AWS DEPLOYMENT

S3 bucket for hosting backend jar file and front end static hosting



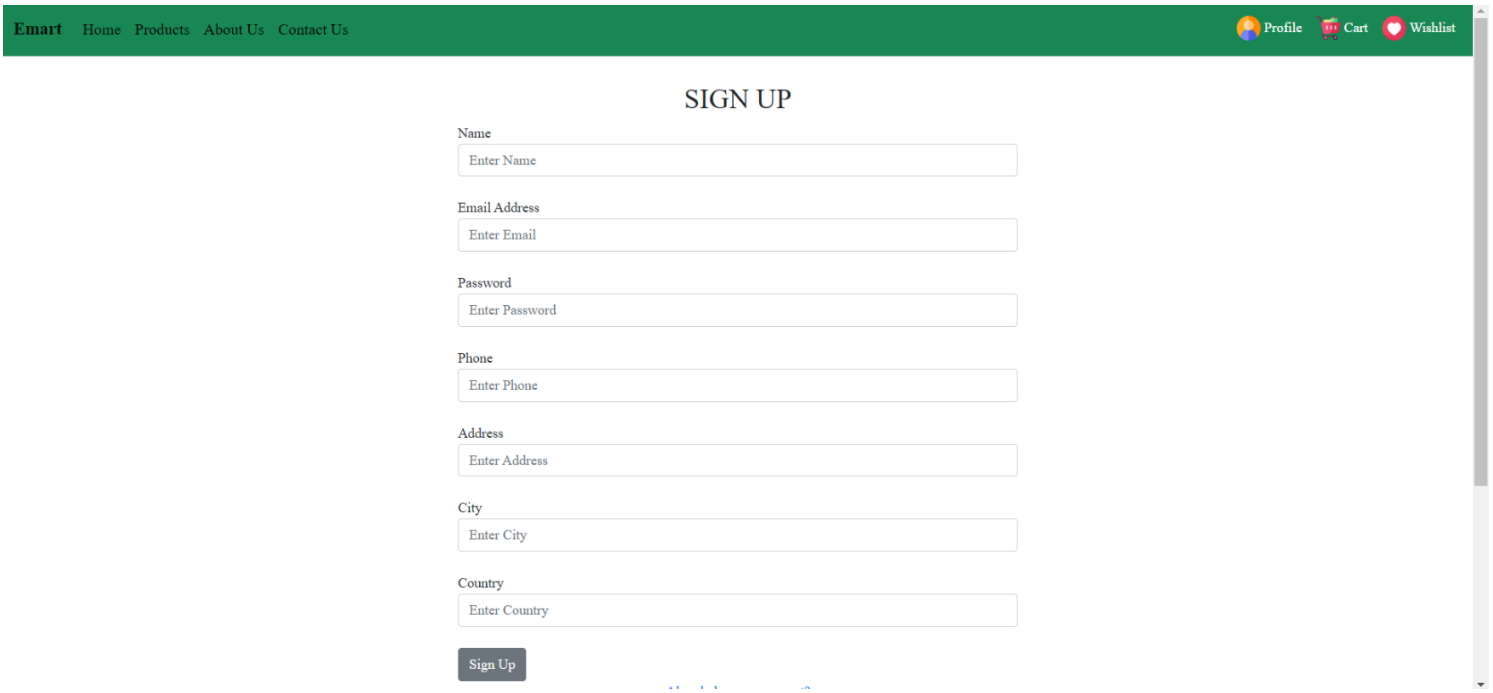
Backend application and database deployed on Ec2 instances in an autoscaling group behind an application load balancer.



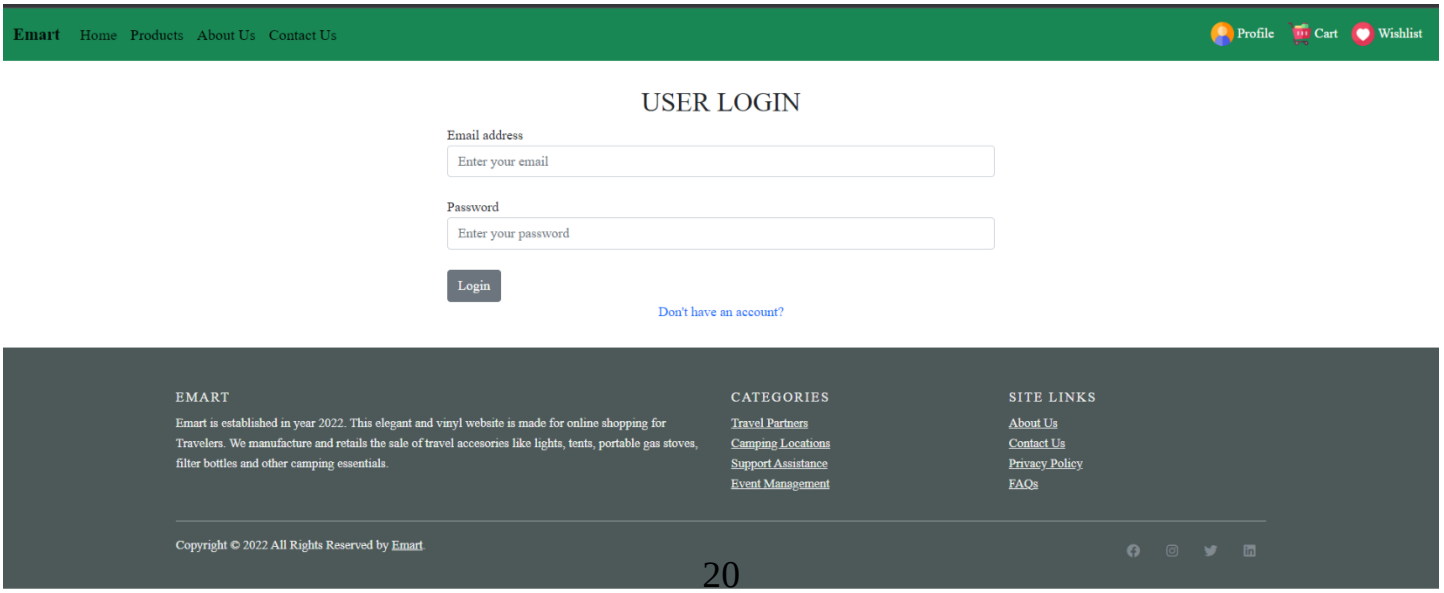
IMAGES OF ALL THE PAGES:

USER Login

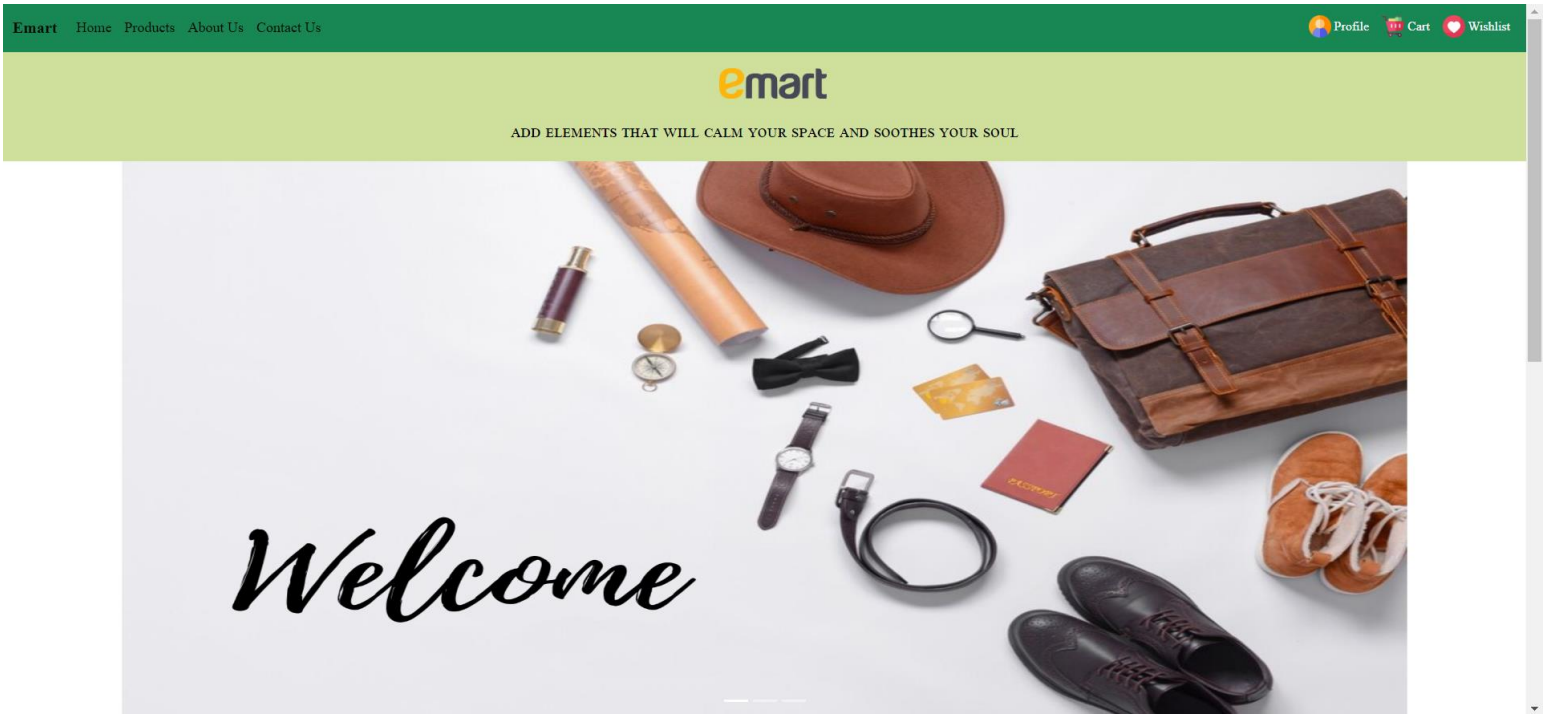
REGISTRATION PAGE:



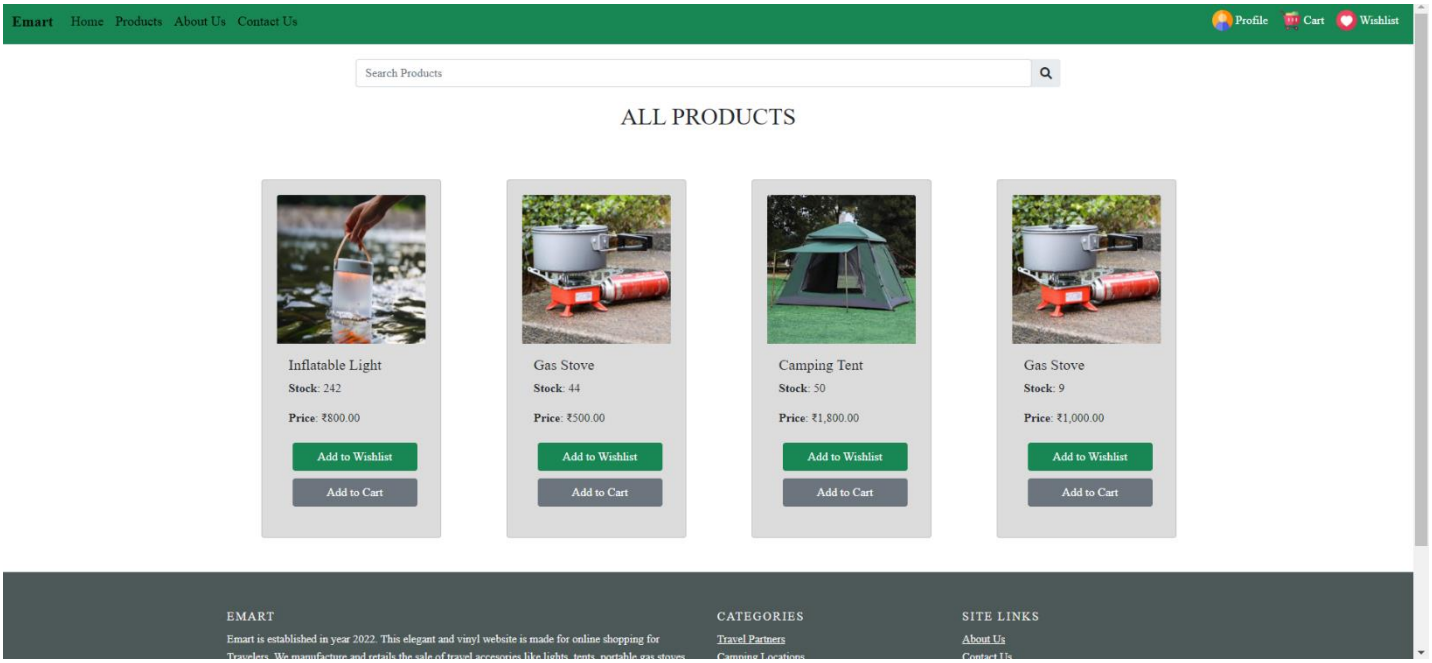
SIGN-IN PAGE:



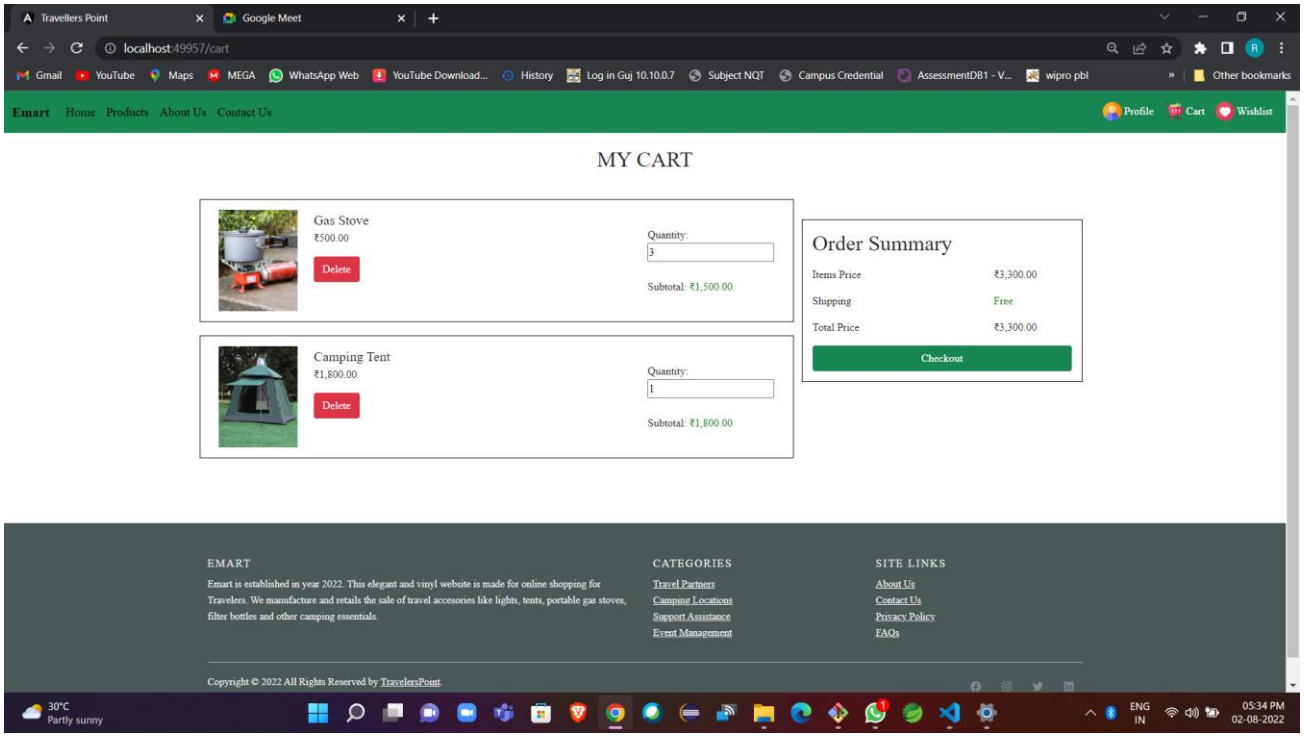
HOME PAGE:



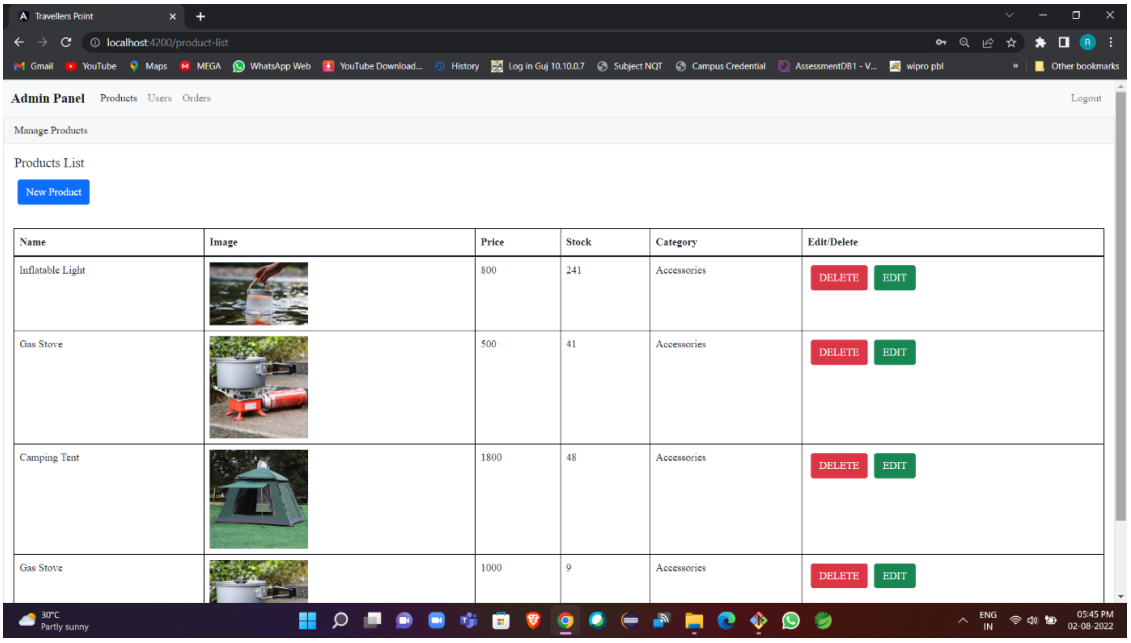
CATAGORIES



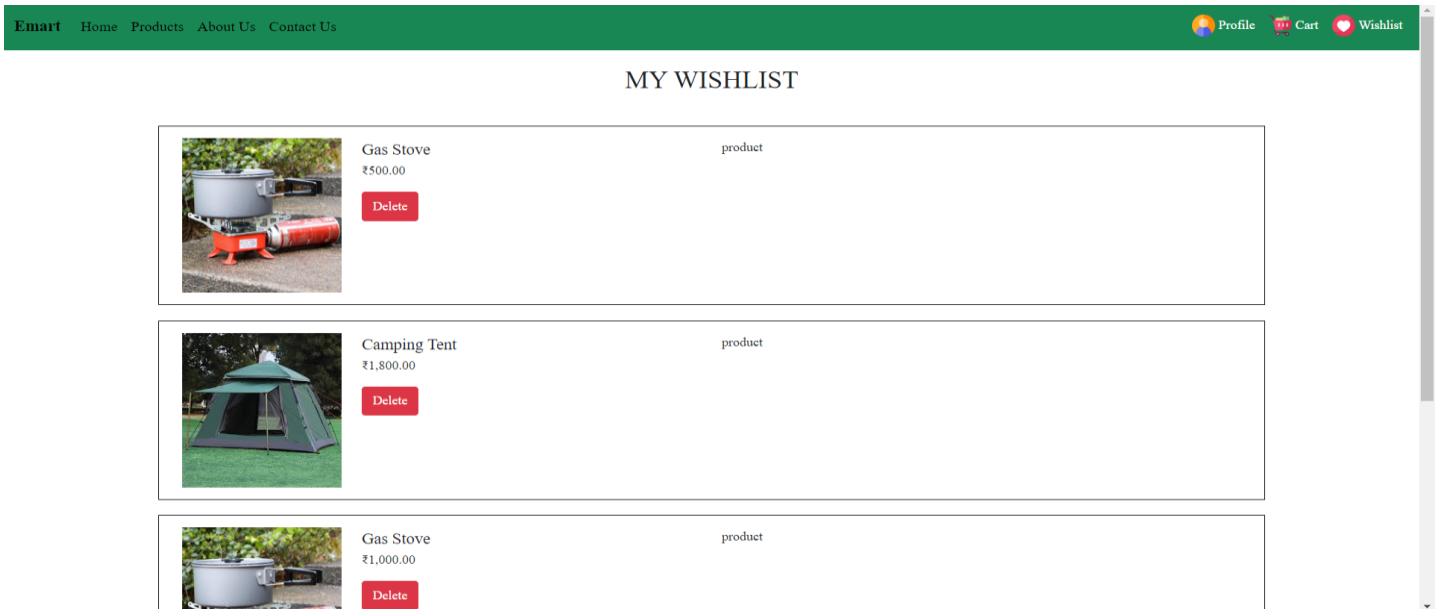
MY CART



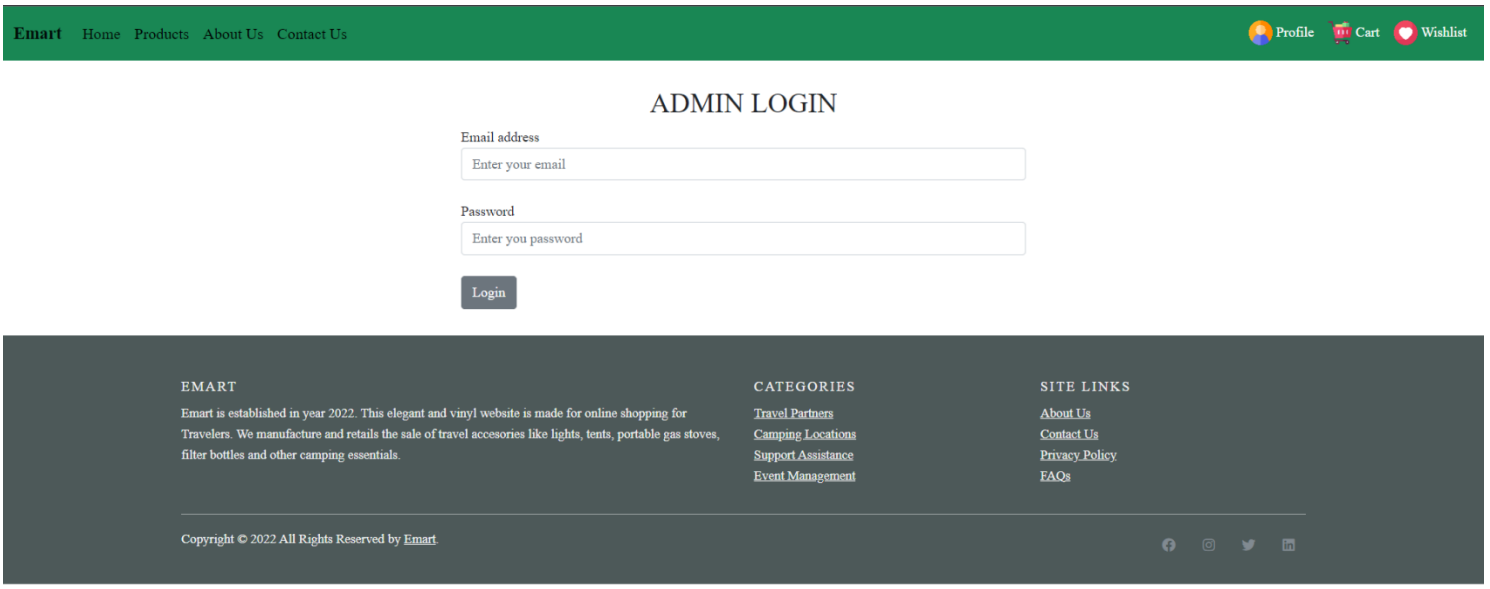
PRODUCT



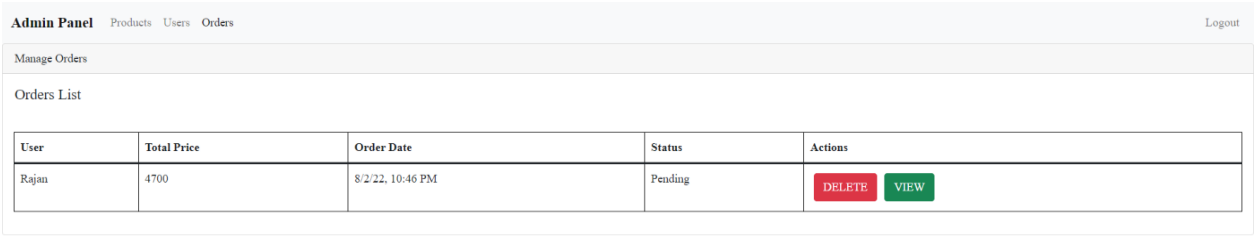
MY WISH LIST



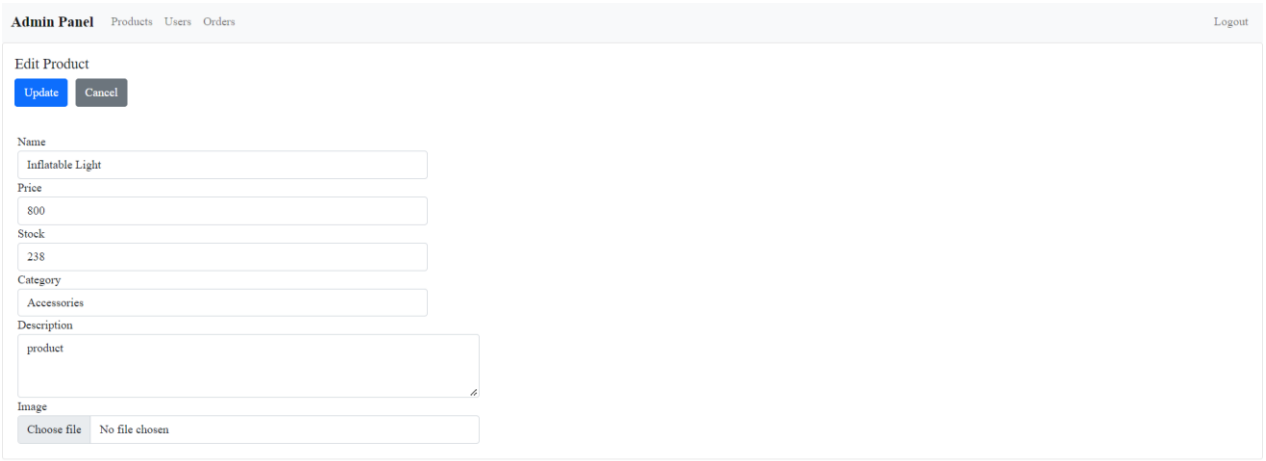
ADMIN-LOGIN PAGE:



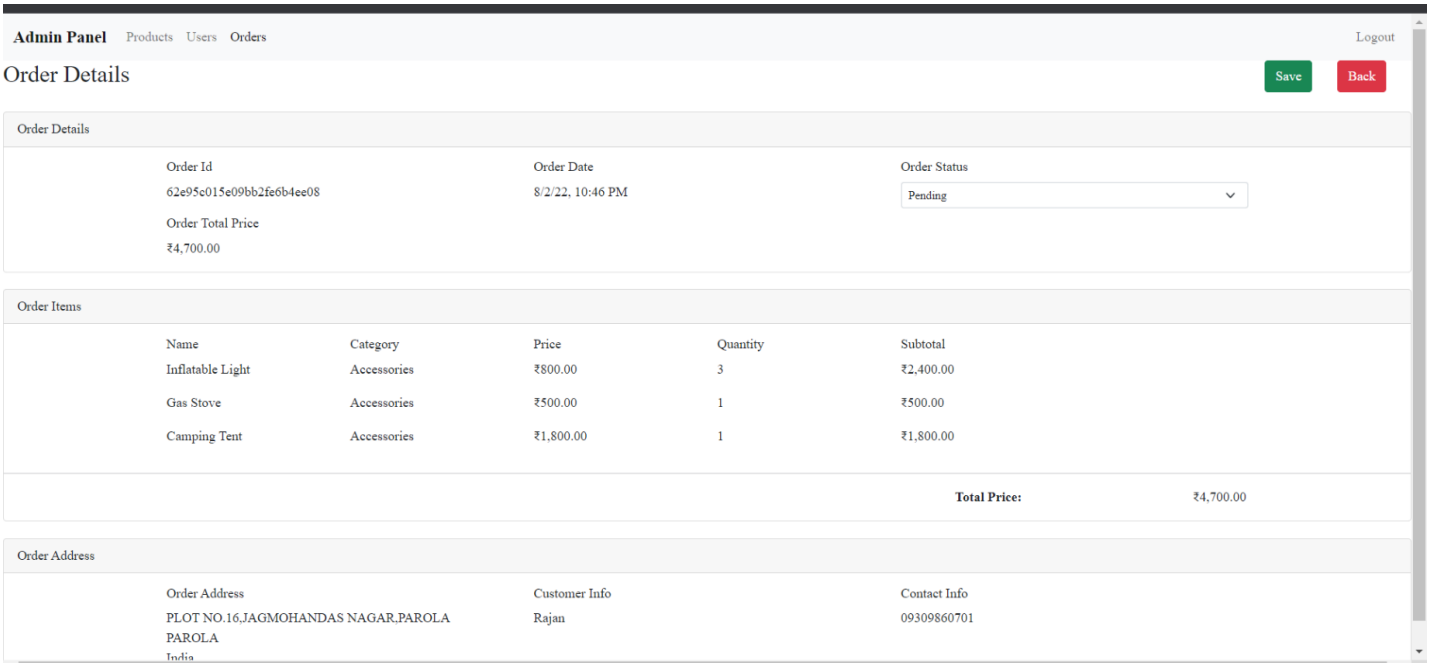
ADMIN HOME PAGE:



ADMIN PRODUCT CRUD USING PAGE:



ORDERS PAGE:


Admin Panel
Products
Users
Orders
Logout
Order Details
Save
Back
Order Details
Order Id
62e95c015e09bb2fe6b4ee08
Order Date
8/2/22, 10:46 PM
Order Status
Pending
Order Total Price
₹4,700.00
Order Items
Name
Category
Price
Quantity
Subtotal
Inflatable Light
Accessories
₹800.00
3
₹2,400.00
Gas Stove
Accessories
₹500.00
1
₹500.00
Camping Tent
Accessories
₹1,800.00
1
₹1,800.00
Total Price:
₹4,700.00
Order Address
Order Address
PLOT NO.16,JAGMOHANDAS NAGAR,PAROLA
PAROLA
India
Customer Info
Rajan
Contact Info
09309860701