



CAPSTONE PROJECT

SHOP FOR HOME

Mentor: Parth Shukla sir

Presented By (Batch C1 Group2)

**Ruchitha
Pavan kumar Pathivada
Dudekula Sameer
Swaraj Mohan Patil
Sai Prasanna Jakkula**

SHOP FOR HOME

1.INTRODUCTION

MOTIVATION:

This document aims to give a brief description about E-Commerce web application using Angular, Spring Boot and MySQL Database. The primary goal of this project is to create a website where consumers may purchase things for home décor in the event that all offline retailers collapse. E-commerce is quickly becoming a recognized and utilized business model. More and more companies are putting in place websites with capability for conducting business transactions online. It is safe to argue that doing your buying online has become routine. The goal of this project is to create a general-purpose online store where products for home décor, such as statues and idols, paintings, and other items, may be purchased. However, this document will discuss online shopping for home decor in order to facilitate implementation. A virtualized store on the Internet where clients may browse the inventory and choose items of interest is known as an online store. The chosen products might be put in a shopping basket. The goods in the shopping cart will be shown as an order when it is time to check out. At that point, more details will be required in order to finalize the purchase. Usually, the customer will be asked to give address, phone number etc. at the time of registration so that he/she can login as User and can shop in the application.

PROBLEM STATEMENT:

During this pandemic time it has been difficult to shop offline with the fear of spreading of corona virus. Hence many offline home décor shops have been closed due to no visitors. Almost all small businesses have been paused due to the pandemic. Hence there must be a way created for these small businesses to run even in these pandemic times so that they don't have to bear huge loss.

Shop For Home is a popular Store in the market for shopping the home décor stuff .Due to Covid 19 all the offline shopping stopped. So, the store wants to move to the online platforms and wants their own web application.

There are 2 users on the application:

1. User
2. Admin

SOLUTION:

E-commerce web applications were created as a result, allowing merchants to sell their goods online and continue operating their businesses while saving buyers from having to travel.

SCOPE OF THE PROJECT:

Both vendors and consumers may gain greatly from online selling and buying, and these benefits are also the main drivers of e-expanding Commerce's market. According to IBEF, India's e-commerce market has a total value of USD 38.5 billion as of 2017 and is projected to reach USD 200 billion by 2026. That provides us a decent notion of the potential for e-commerce in India in the future.

PROBLEM DEFINITION:

E-commerce provides an easy way to sell products to a large customer base. However, there is a lot of competition among multiple e-commerce sites. When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. Also, users are not sure about the brands or the actual products they want to purchase. They have a very broad idea about what they want to buy.

OBJECTIVES:

The primary goal of e-commerce is to reach maximum customers at the right time to increase sales and profitability of the business. Functions of e-commerce include buying and selling goods, transmitting funds or data over the internet.

The three main objectives are as follows

1. Find the best solution for their needs
2. Make a purchase, and
3. Get information/answers to their questions

2. OVERALL DESCRIPTION

PRODUCT PERSPECTIVE

The goal of a home décor business is to serve both potential customers and sellers that wish to connect with a wide range of consumers. The goal of this initiative is to close the communication gap between the supplier, the merchant, and the consumer. For the abovementioned objective, Shop for Home software should be user-friendly, "easy to learn," and trustworthy. The functionality of Shop for Home is not dependent on the presence of other software because it is designed to be a stand-alone solution. Both UNIX and Windows-based platforms should be able to run it.

USER CLASSES AND CHARACTERISTICS

The user should be familiar with terms used in relation to shopping malls, such as "transaction," "checking out," and "shopping cart." The user ought to be knowledgeable about the Internet.

OPERATING ENVIRONMENT

The system for an online store selling home decor is a website that should function in all well-known browsers. For example, consider Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Additionally, IE 6.0 will be compatible with it. The majority of the functions will work with Mozilla Firefox and Opera 7.0 or later. The internet connection would be the sole prerequisite for using this online product. The hardware setup consists of 40GB hard disk, 15-inch color display, and 122-key keyboard. Keyboards and mouse are the two most essential input and output devices.

PRODUCT FUNCTIONS

1. User-Friendly Design

A positive user experience is perhaps the most crucial element that buyers expect from an e-commerce site. Customers will probably quickly switch to the many other online stores if they have trouble navigating your website or finding what they're searching for. Use recommended e-commerce UX techniques. Focus on improving the user experience by making a clean, clear, high-quality homepage. with a search bar included. The navigation bar clearly lists the category pages and putting emphasis on developing a responsive website.

2. Mobile-Friendly Features

Consumers who shop online nowadays are becoming more and more dependent on their mobile devices. In 2020, mobile devices accounted for 56% of all online purchases, according to Salecycle. In other words, the majority of buyers prefer to explore online stores on their phones, thus it's crucial to incorporate mobile optimization into the design of your e-commerce website. To improve your conversion rate and keep clients satisfied, make sure your website design is automatically adjusted for the size and form of phones.

3. Multiple Payment Options

For online merchants, shopping cart abandonment is frequently a major problem. Make the purchase process as simple as possible for your consumers by including a variety of online payment options in the shopping cart. This will increase your chances of closing the sale during the final checkout step on your website. Add options for payment services like PayPal or Stripe in addition to providing debit and credit card choices. To encourage users to click the "purchase" button, you may also provide plugins with a buy now, pay later feature.

4. 24/7 Customer Service

Providing friendly, approachable customer service is crucial to a positive client experience. Microsoft estimates that 90% of Americans give customer service a high priority when choosing whether or not to make a purchase from a business. One of the features on your e-commerce website should be a 24/7 chat box for customer care to respond to requests at any time. For a nominal charge, you can easily add this option to your website using plugins like Zendesk or LivePerson.

5. User Discounts Customers

Customers enjoy believing that they are receiving a good bargain and are being treated differently from other customers. To provide customers with this kind of alluring, tailored experience, provide customized bargains, offers, and other user benefits. Create user profiles for consumers so they may enjoy loyalty discounts, a personalized wish list, and account history. To deliver targeted special offers to devoted consumers, you may also employ an automated emailing system on an e-commerce platform.

Appropriation of Requirements

As stated by the customer, security is not a concern of this project. As such, it is beyond the scope of this system to encrypt personal user data, encrypt credit card information, prevent unauthorized login attempts, or any other concern of this nature. Additionally, the system is not responsible for the following:

- Verifying that credit card information is valid
- Verifying the email address provided by a user
- Storing additional information about a product beyond simply the category and price •

Allowing users to edit their account details (username, password, mailing address, etc)

- Allowing customers to order multiple copies of product in a single order
- Providing individual product description
- Allowing the Admin to update login credentials or other information about the Admin.

3. SPECIFIC REQUIREMENTS

FUNCTIONAL REQUIREMENTS

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 Wish list 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 bulk upload option to upload a csv for products 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 users

SOFTWARE REQUIREMENTS:

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

Technologies	Angular, Spring Boot, MySQL
Languages	Type Script, Java, SQL,HTML,CSS
IDE	Eclipse , VS code, MySQL
Operating System	Windows 7/8/10/11 , Linux distros, MacOS X or later.

HARDWARE REQUIREMENTS:

The bare minimum hardware needs vary greatly depending on the software an Enthought Python/VS Code user is working on. Apps that need to execute several computations or operations rapidly will need a faster CPU, but applications that need to store big arrays of objects in memory would need more RAM.

Processor	Intel or AMD dual core x86 processor.
Ram	4 GB or above.
Hard disk	500 MB of free disk space or more.

3.3 ARCHITECTURE

Angular Architecture:

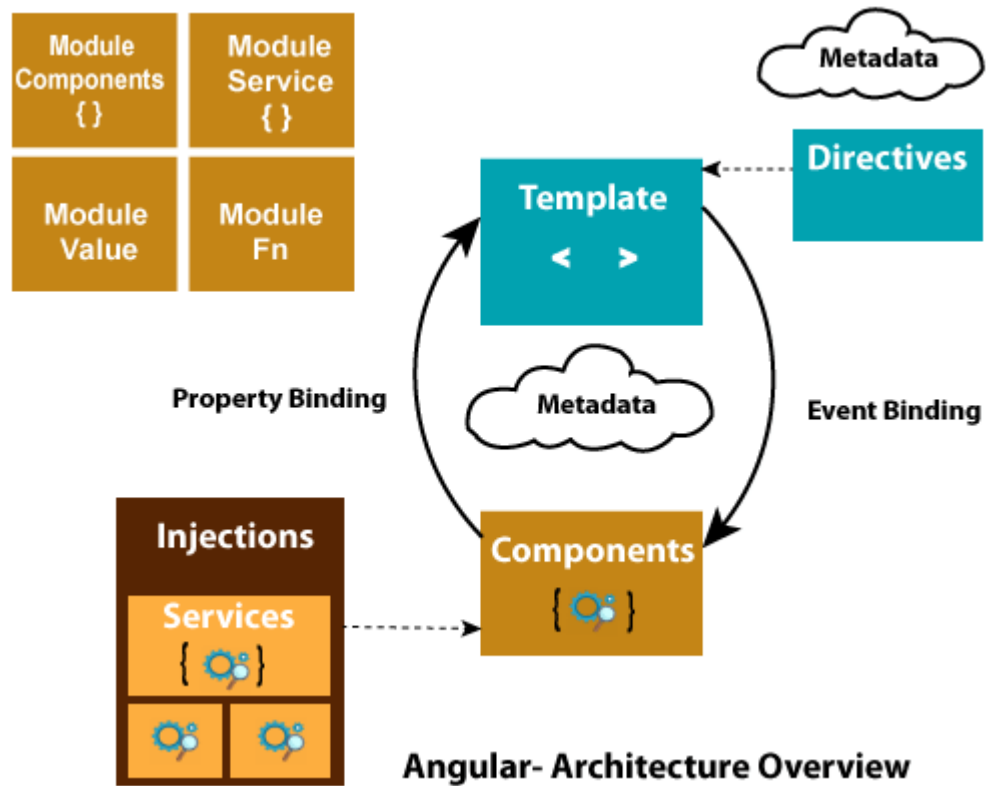


Fig. 3.1 Angular Architecture

There are main eight blocks of Angular.

1. Module
2. Component
3. Metadata
4. Template
5. Data Binding
6. Service
7. Directive
8. Dependency Injection

Spring Boot Architecture:

Spring Boot flow architecture

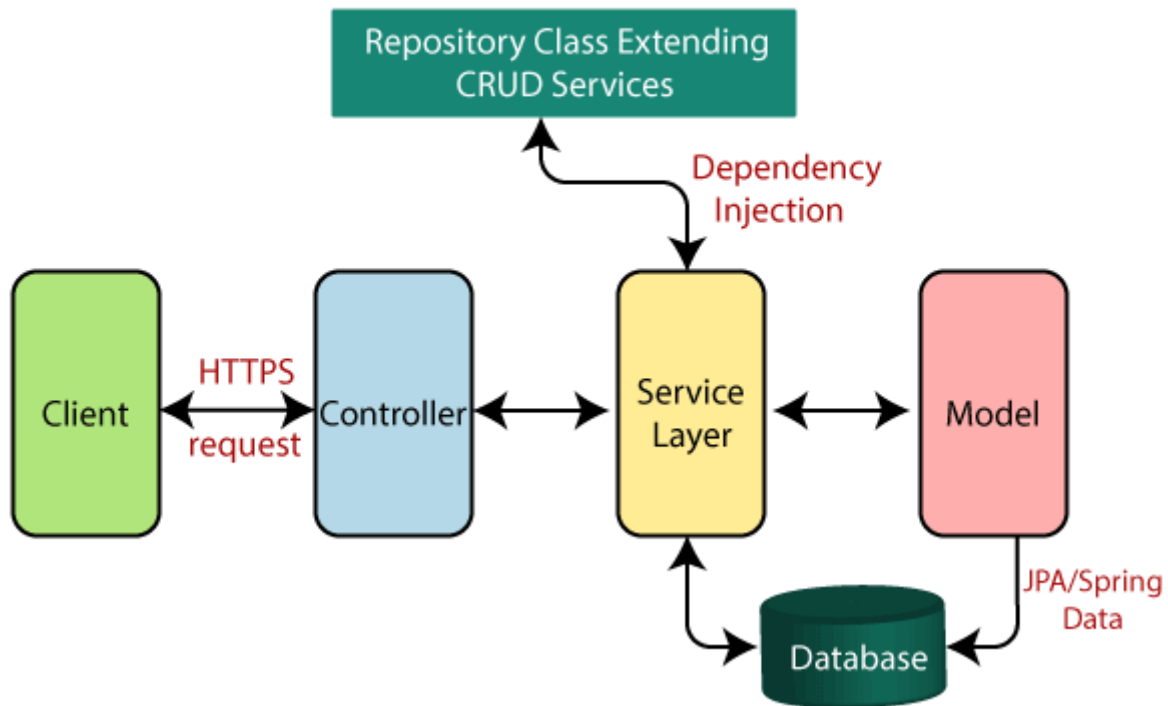


Fig. 3.2 Spring Boot Architecture

The spring boot consists of the following four layers:

1. Presentation Layer – Authentication & JSON Translation
2. Business Layer – Business Logic, Validation & Authorization
3. Persistence Layer – Storage Logic
4. Database Layer – Actual Database

Micro Services Architecture

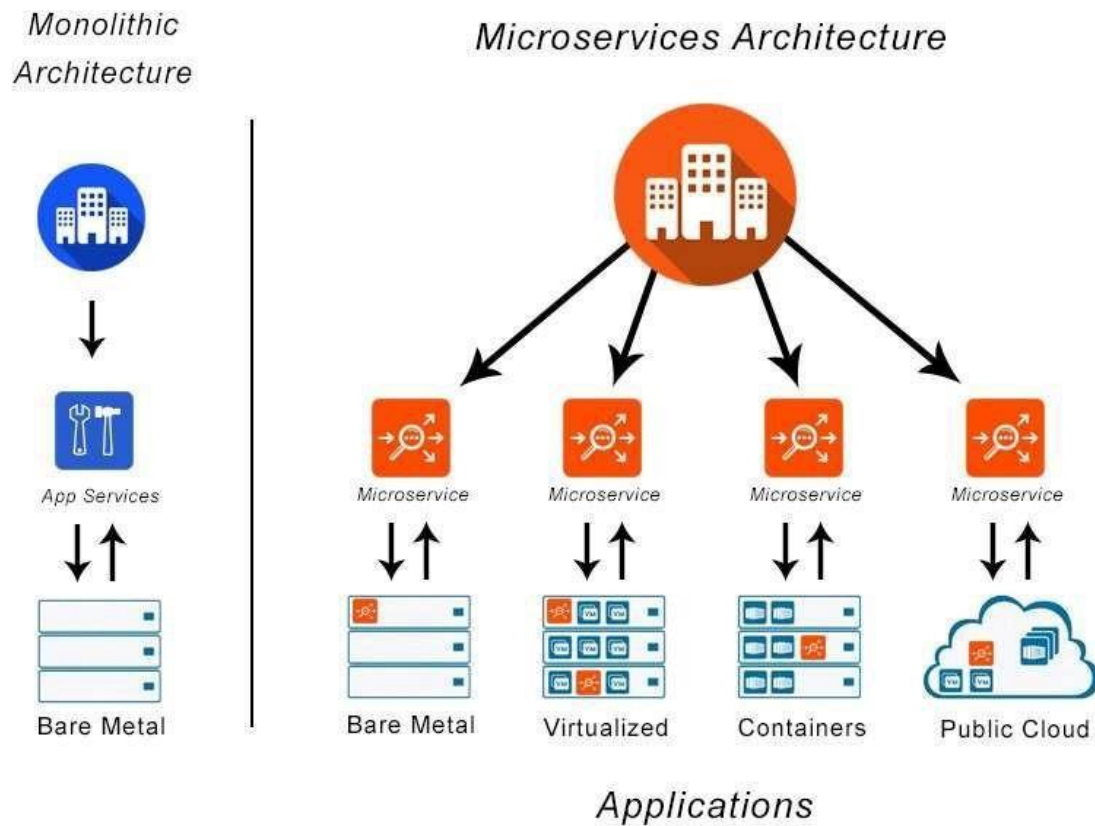


Fig 3.3 Micro services Architecture

Typically, microservices are used to speed up application development. Microservices architectures built using Java are common, especially Spring Boot ones. It's also common to compare microservices versus service-oriented architecture. Both have the same objective, which is to break up monolithic applications into smaller components, but they have different approaches

Project Architecture

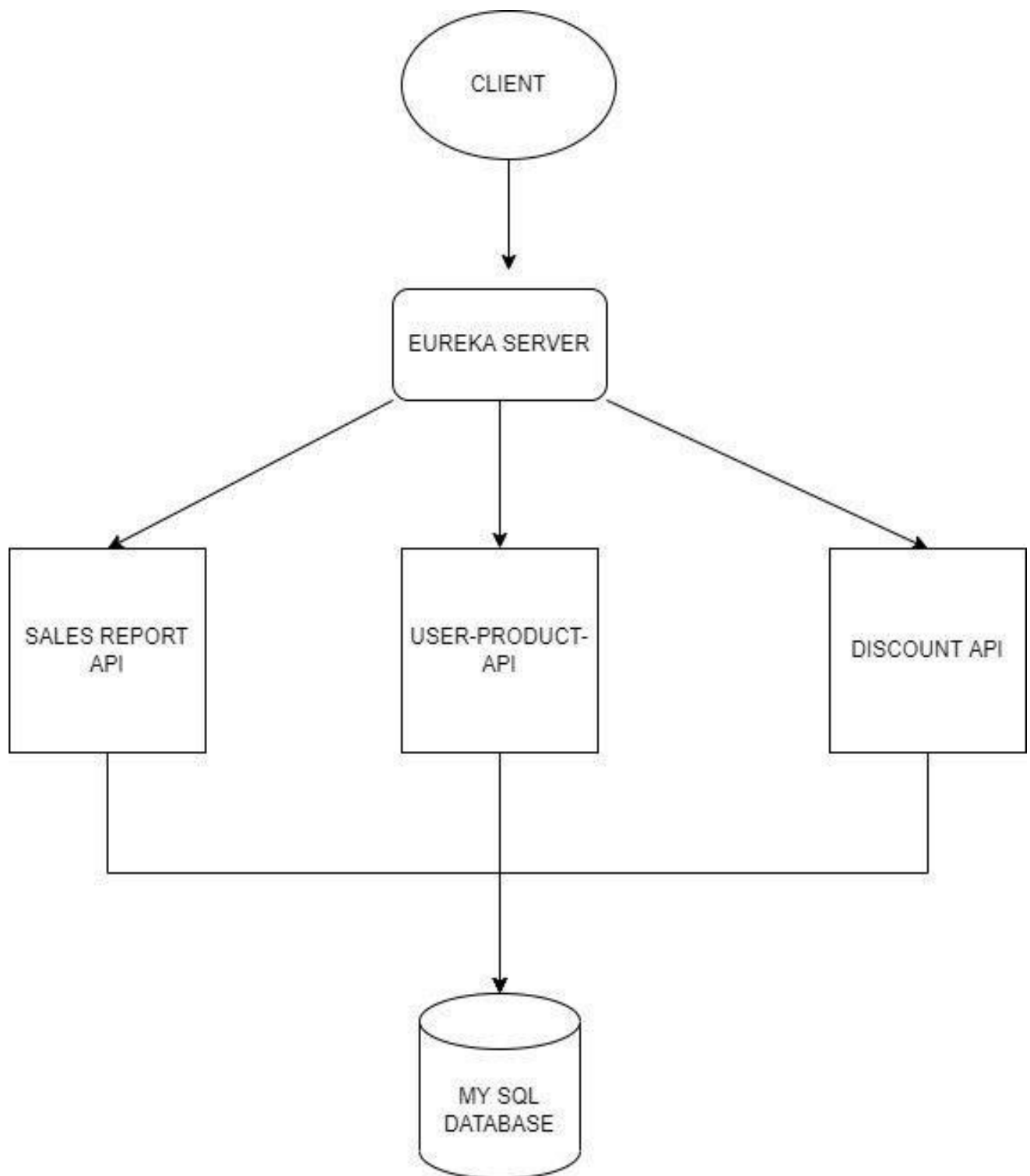


Fig 3.4 Project Architecture

API Gateway.

The API gateway is the entry point for clients. Instead of calling services directly, clients call the API gateway, which forwards the call to the appropriate services on the back end.

Advantages of using an API gateway include:

- It decouples clients from services. Services can be versioned or refactored without needing to update all of the clients.
- Services can use messaging protocols that are not web friendly, such as AMQP.
- The API Gateway can perform other cross-cutting functions such as authentication, logging, SSL termination, and load balancing.
- Out-of-the-box policies, like for throttling, caching, transformation, or validation.

Discount-Service-API

Discount-Service contains business logic and functionalities for discount and coupon list Features. Admin can add and delete coupons from Discount UI from Admin module. It connects to shopforhome DB (MySQL). This service configured with application properties which runs the service in port number 9092 in this application.

Sales-Report-API

Sales-Report contains business logic and functionalities for report download option. Admin can download the stocks report as an excel file from Admin module. It connects to shopforhome DB (MySQL). This service configured with application properties which runs the service in port number 9091 in this application.

User-Service

User-service contains all the logic of user and admin. User can see all the products ,wishlist , cart option to order the products. Admin can manage products , bulk upload the product. It connects to shopforhome DB(MySQL). This service configured with application properties which runs the service in port number 9090 in this application.

Eureka Server For Microservice

Eureka Server is service discovery for your microservices, where all client applications can register by themselves and other microservices look up the Eureka Server to get independent microservices to get the job complete.

Eureka Server is also known as Discovery Server and it contains all the information about client microservices running on which IP address and port.

To achieve this you need to create a Eureka Server application and add the below dependency in POM.xml.

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</dependency>
```

- **spring.application.name** is a unique name for your application.
- **server.port** in which your application will be bound and we will use default port 8761 for eureka server.
- **eureka.client.fetch-registry** doesn't register itself in eureka server.
- **eureka.client.register-with-eureka** is determines if service register itself as a client in eureka server.

Running the Eureka Server

Run the Eureka server as Java application and go the URL: <http://localhost:8761/>

The screenshot shows the Spring Eureka web interface. The header includes the Spring Eureka logo and a 'Toggle navigation' button. The main content area is divided into sections: 'System Status', 'DS Replicas', and 'Instances currently registered with Eureka'.

System Status

Environment	test	Current time	2022-08-24T14:33:52 +0530
Data center	default	Uptime	01:12
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	12

DS Replicas

[localhost](#)

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DISCOUNT-SERVICE	n/a (1)	(1)	UP (1) - chikky:discount-service:8081
ESHOP-SERVICE	n/a (1)	(1)	UP (1) - chikky:eshop-service:8083
REPORT-SERVICE	n/a (1)	(1)	UP (1) - chikky:report-service:8082

The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray displaying the date and time as 02:34 PM on 24-08-2022.

DataBase Design:

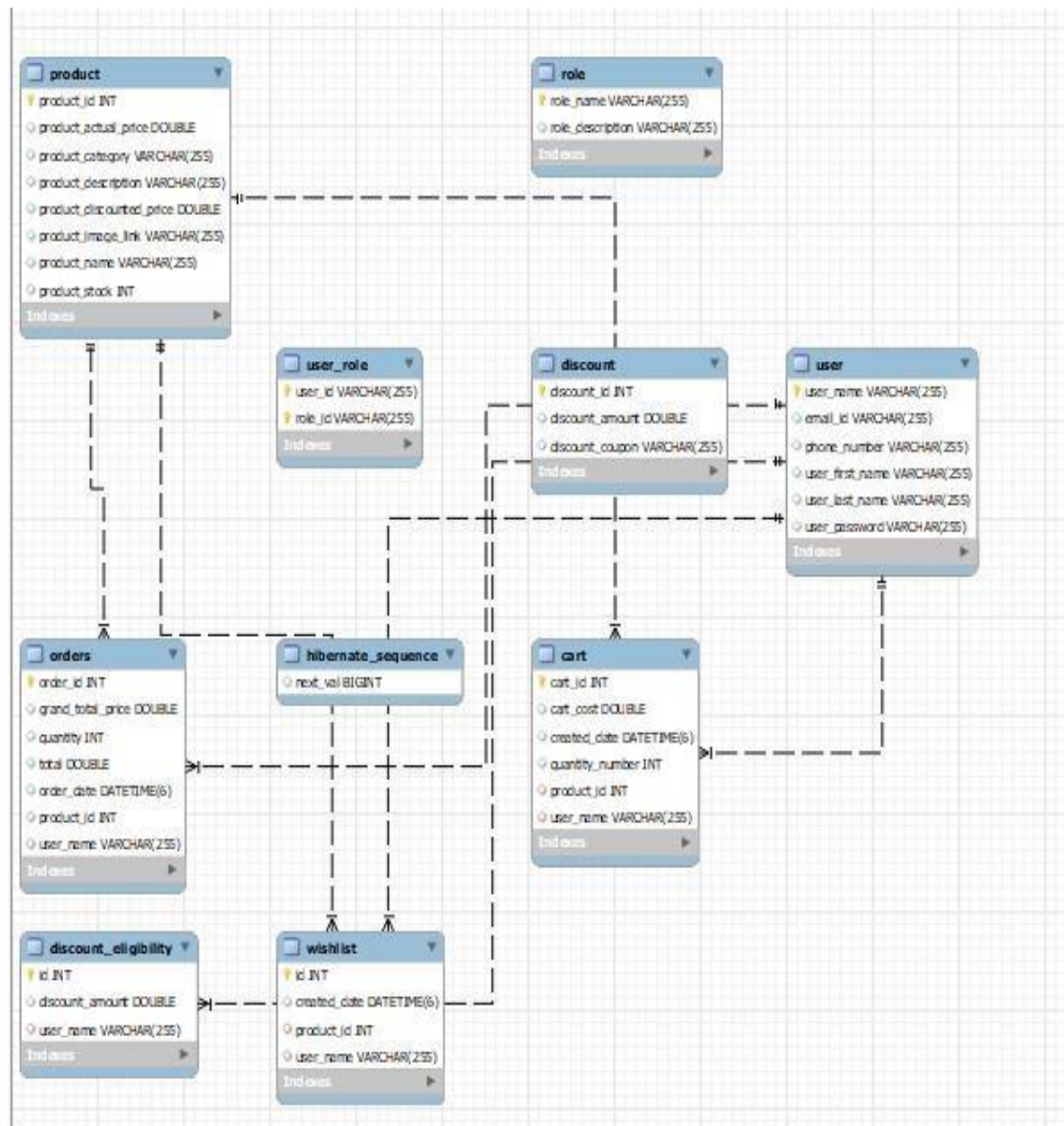


Fig. Tables created in Mysql database

Entities in the Database include

1. product
2. role
3. cart
4. user
5. user_role
6. orders
7. discount
8. wishlist
9. discount_eligibility
10. hibernate_sequence

3.3 NON FUNCTIONAL REQUIREMENTS

1. Usability:

Regardless of the size of your business, the website of your business should be easy to use for even a non-technical user. Do you know that a general user takes just 0.05 seconds to figure out whether the website is worth its time or not? Thus you have to give special attention to the design of your homepage, CTAs, and easy checkout to get past those milliseconds of doom. The usability of a website is also defined by:

- How easily a user can achieve their target in a single page visit
- How quickly they can perform tasks in the store
- The memorable & intuitiveness of the design
- Number and types of errors users make

2. Security:

If your website handles financial transactions, user financial information, or other sensitive information, security is of the highest significance. The adoption of an SSL certificate and a data privacy policy will increase consumer confidence in your website and turn visitors into brand promoters. Additionally, it is taken into account for the various admin roles that help you manage who may create, see, copy, update, and remove content. Depending on where your company is located, security might also include adhering to laws governing the protection of client data, like the GDPR in Europe.

3. Performance:

You must pay close attention to performance in the non-functional requirements documents if you want to increase website traffic. Regardless of the quantity of integrations and the volume of visitors on your website, the emphasis should be on loading the e-commerce shop as quickly as feasible. You may choose the maximum number of SKUs you wish to add, the benchmark speed, or any other performance metric that works best for your company. Because the developers won't have control over the third party API requests, don't take into account the delivery time of the third party system.

4. Maintainability:

The most challenging aspect of creating a business budget is accounting for maintenance operations expenditures. By including website maintenance into the process from the beginning, future system errors may be found and fixed with less time and expense. Unfortunately, there is no way to prevent problems from arising in the future, therefore you must search for a website building business that can handle website maintenance.

5. Scalability:

Last but not least, you need to consider scalability while searching for a solution that will be reliable in the future. It will outline how the website may expand and add features and functionality without affecting how well your website functions. For your website to handle additional transactions, you must be able to add extra RAM, servers, or disc space. You might need to implement localization tools on the server side as you expand into new markets. Overall, this NFR allows for easy corporate growth and has ramifications for both hardware and software.

4.MODELING REQUIREMENTS

UML DIAGRAMS :

Unified Modeling Language is known as UML. A general-purpose modelling language with standards, UML is used in the field of object-oriented software engineering. The Object Management Group oversees and developed the standard.

The objective is for UML to establish itself as a standard language for modelling object-oriented computer programmes. UML now consists of two main parts: a notation and a meta-model. In the future, UML may potentially be coupled with or added to in the form of a method or process.

A common language for describing, visualising, building, and documenting the artefacts of software systems as well as for business modelling and other non-software systems is the Unified Modeling Language.

The UML is a set of best engineering principles that have been successful in simulating big, complicated systems.

The process of creating objects-oriented software and the software development process both heavily rely on the UML. To express the design of software projects, the UML primarily employs graphical notations.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices

USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

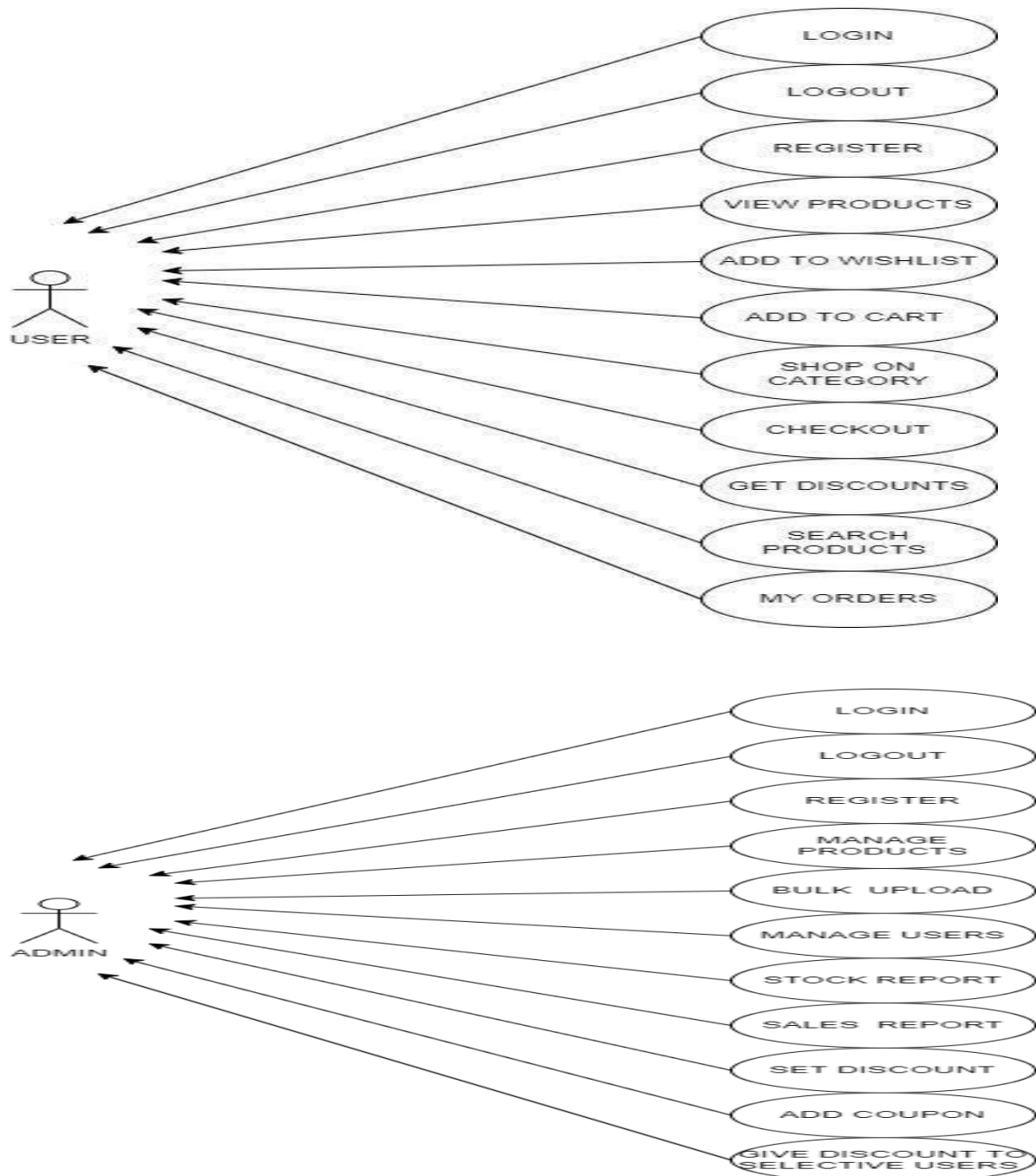


Fig. 4.1 Use Case Diagram

CLASS DIAGRAM

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

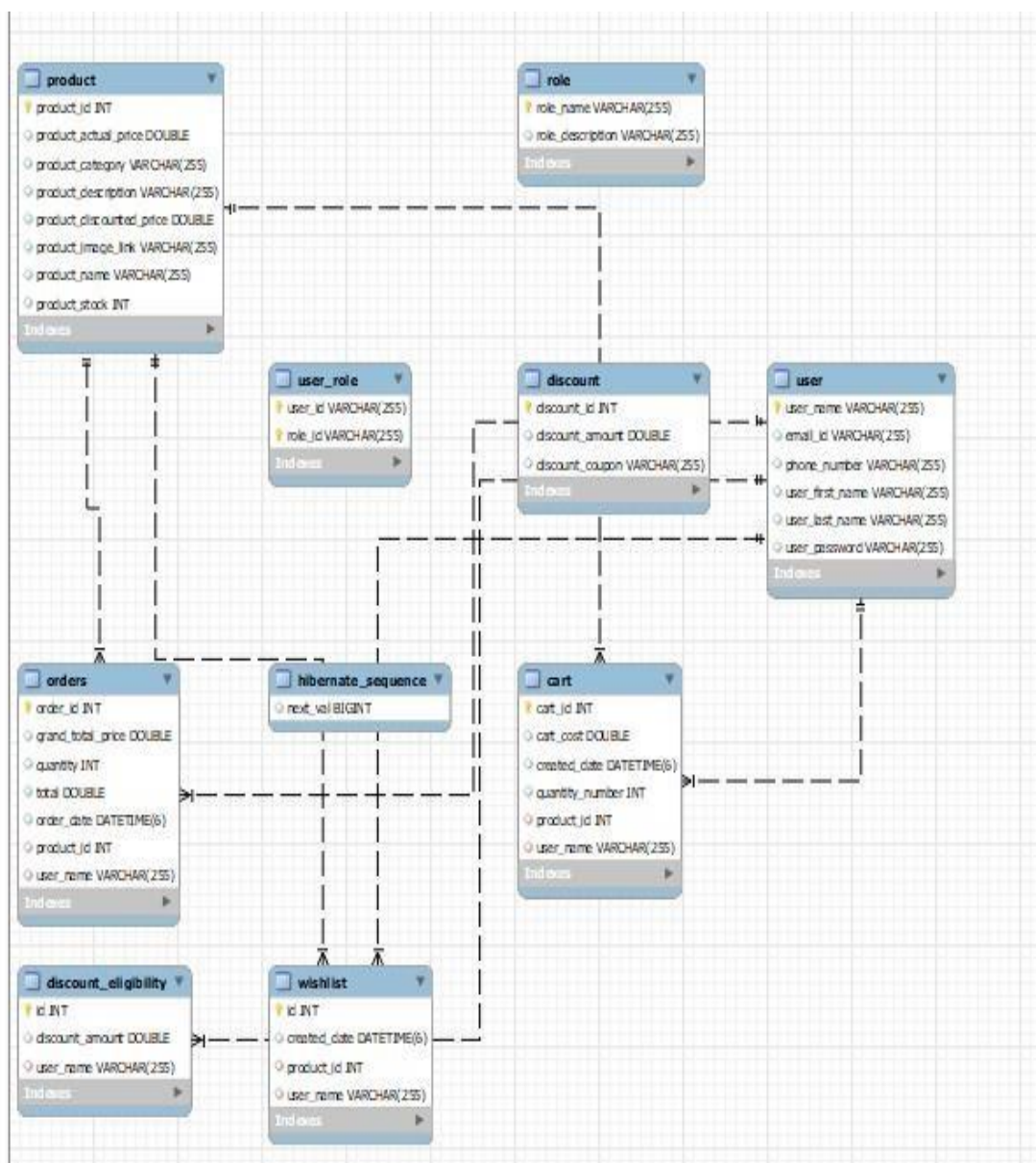


Fig. 4.2 Class Diagram

STATE CHART DIAGRAM

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.

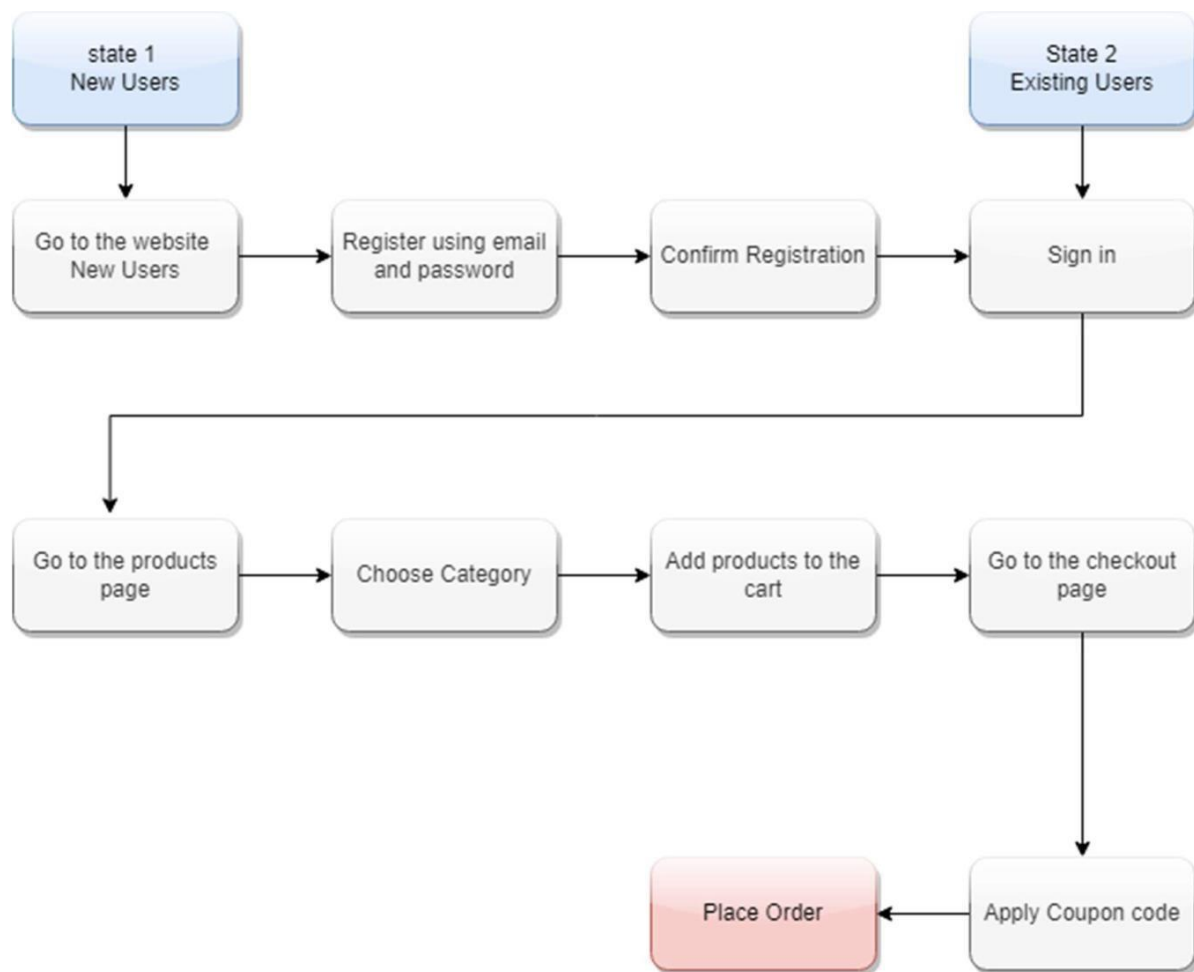


Fig. 4.3 State Chart Diagram

SEQUENCE DIAGRAM :

A sequence diagram shows how various system items interact with one another. The fact that a sequence diagram is time-ordered is crucial. This indicates that the precise order of the objects' interactions is displayed step by step. The sequence diagram shows how various things communicate with one another by sending "messages."

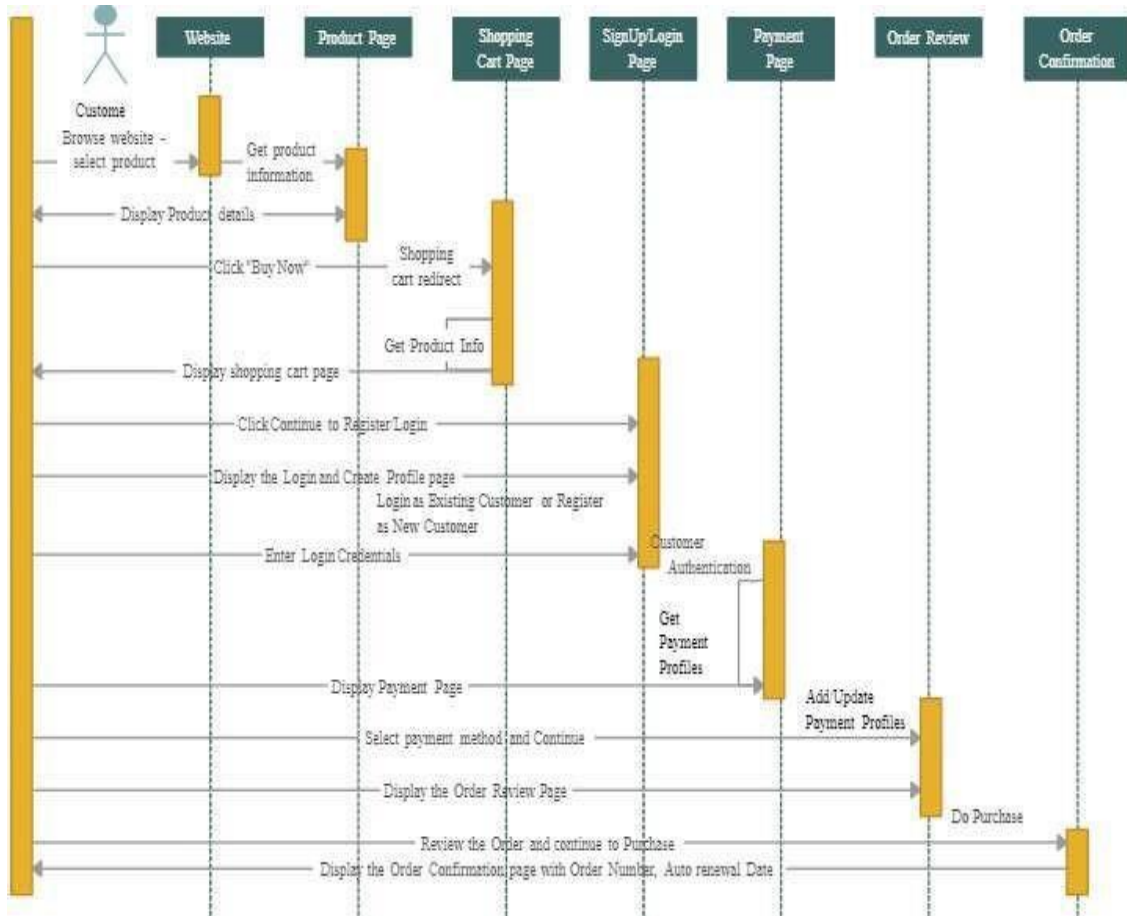


Fig. 4.4 Sequence Diagram

DATA FLOW DIAGRAM

The flow chart diagram is a special kind of class diagram. The flow chart diagram depicts the flow of the process or algorithm using a set of symbols. The flow chart diagram is a pictorial representation of the algorithm or working of the algorithm.

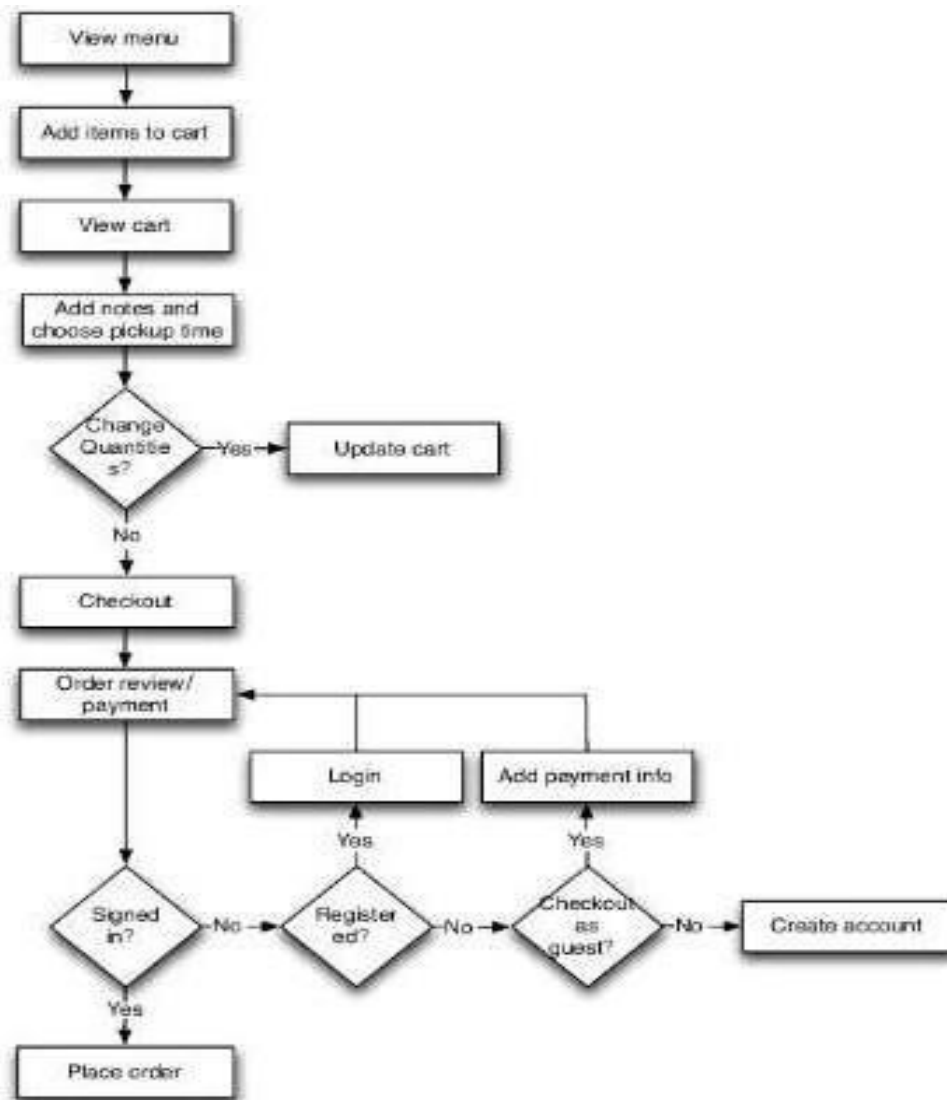


Fig 4.5 Data Flow Diagram

5. IMPLEMENTATION

We will implement a simple e-commerce application. We'll develop an API using Spring Boot and a client application that will consume the API using Angular. Basically, the user will be able to add/remove products from a product list to/from a shopping cart and to place an order.

TECHNOLOGY USED

In below we would like to explain which technologies are used in this project. It's helpful for understand the project layouts & overview of documentation.

Back-end

1. Java
2. Spring Boot
3. Spring Security
4. JWT
5. Hibernate
6. MySQL
7. Maven

Front-end

1. Angular
2. Bootstrap
3. Angular-materials

IMPLEMENTATION STEPS

First you need to start back-end server, after that execute client side. Follow below steps to run this application on your system.

First Install all the prerequisite softwares like Java 8 jdk, VS code editor, Eclipse, MySQL.

Database:

1. Install Mysql workbench
2. Open that Mysql
3. Open this query editor and You can run the query from sql file.
4. Run the query.

Backend:

1. Open Eclipse IDE
2. Create Spring boot applications for main backend part, microservice based structure i.e Eureka Server , User_Service , Discount_Service and Sales_Report microservices by adding required dependencies.
3. Download and Extract the jar files into local directory.
4. Import the extracted jar files into eclipse IDE and run the applications one by one as Java application.

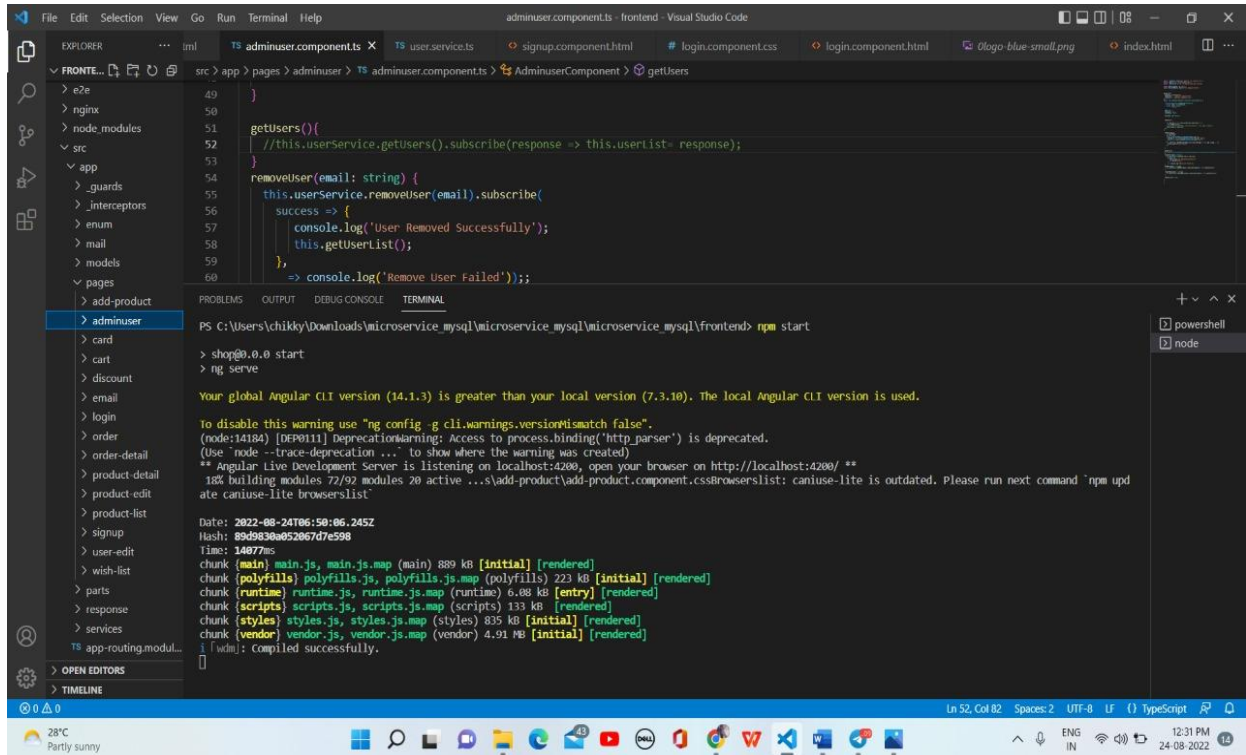
Frontend:

1. From your local FrontEnd code path -> open cmd
Eg -> {local path}\frontend
2. Run this command -- **code** .
3. Run **Npm install** from vs code terminal
4. Run npm start
5. After the successful compiling you got this link in terminal localhost/4200.
6. Open this link in google chrome.

6. RESULT AND DISCUSSION

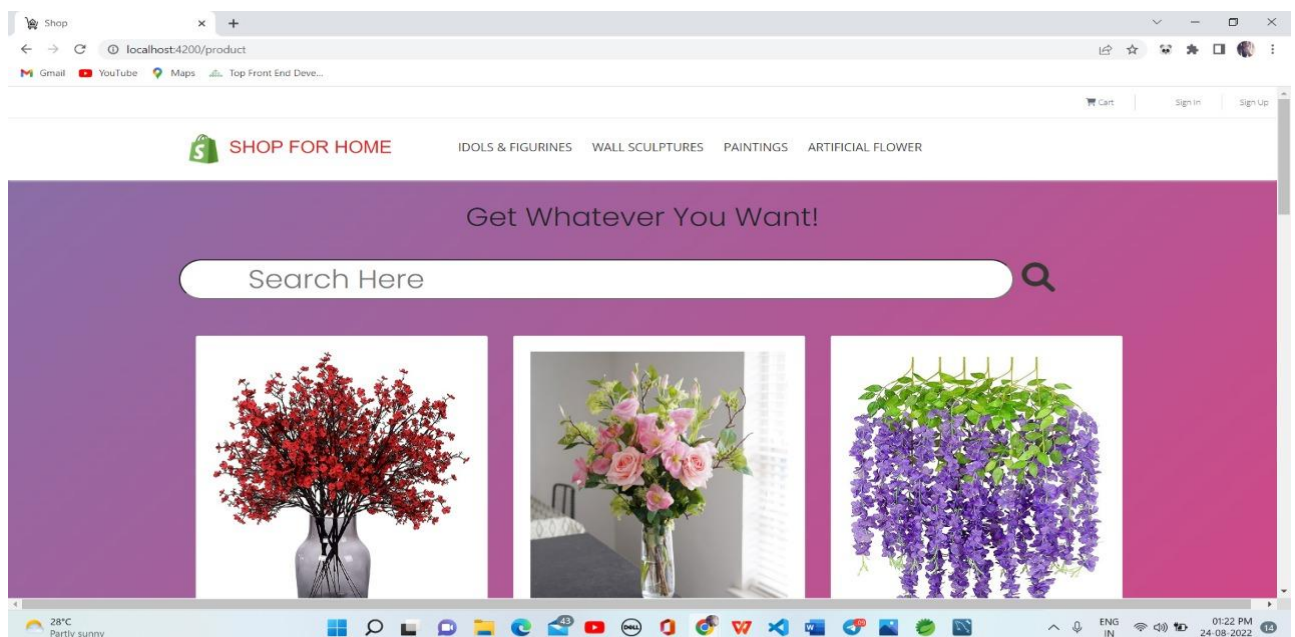
FRONT END SCREENSHOTS

FRONT END STRUCTURE:

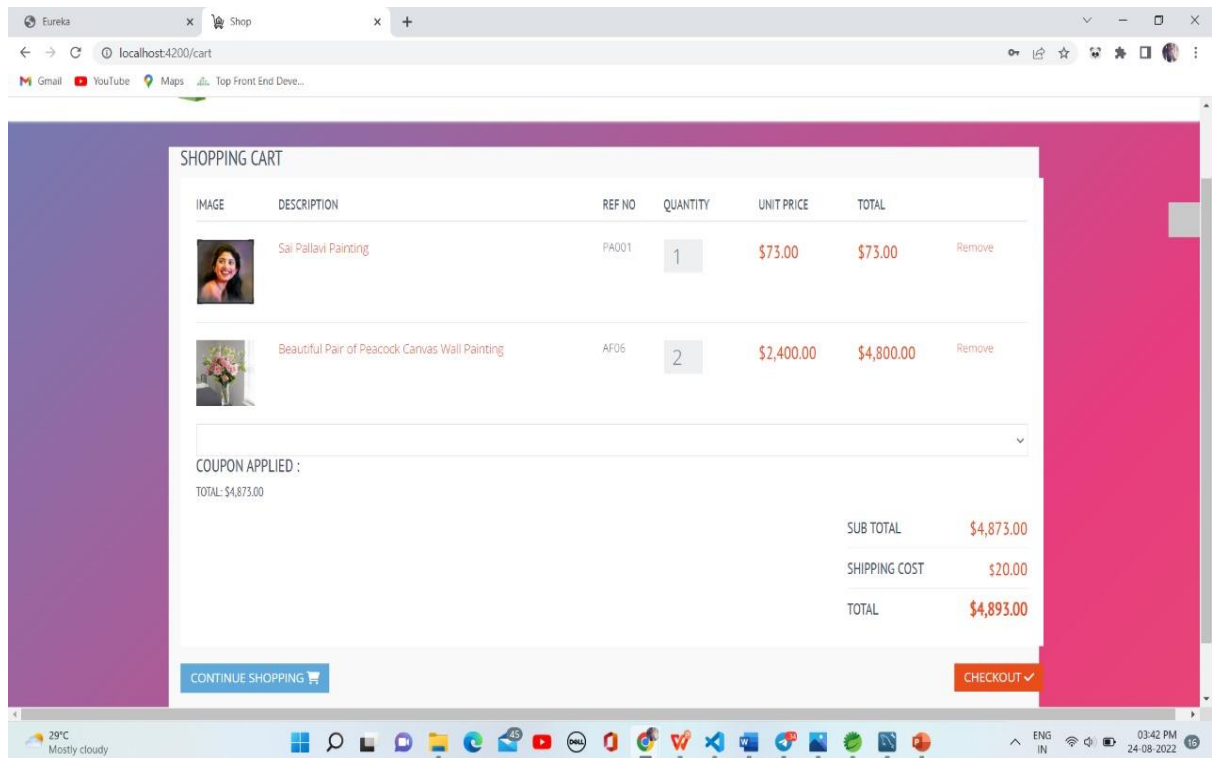


VS Code page

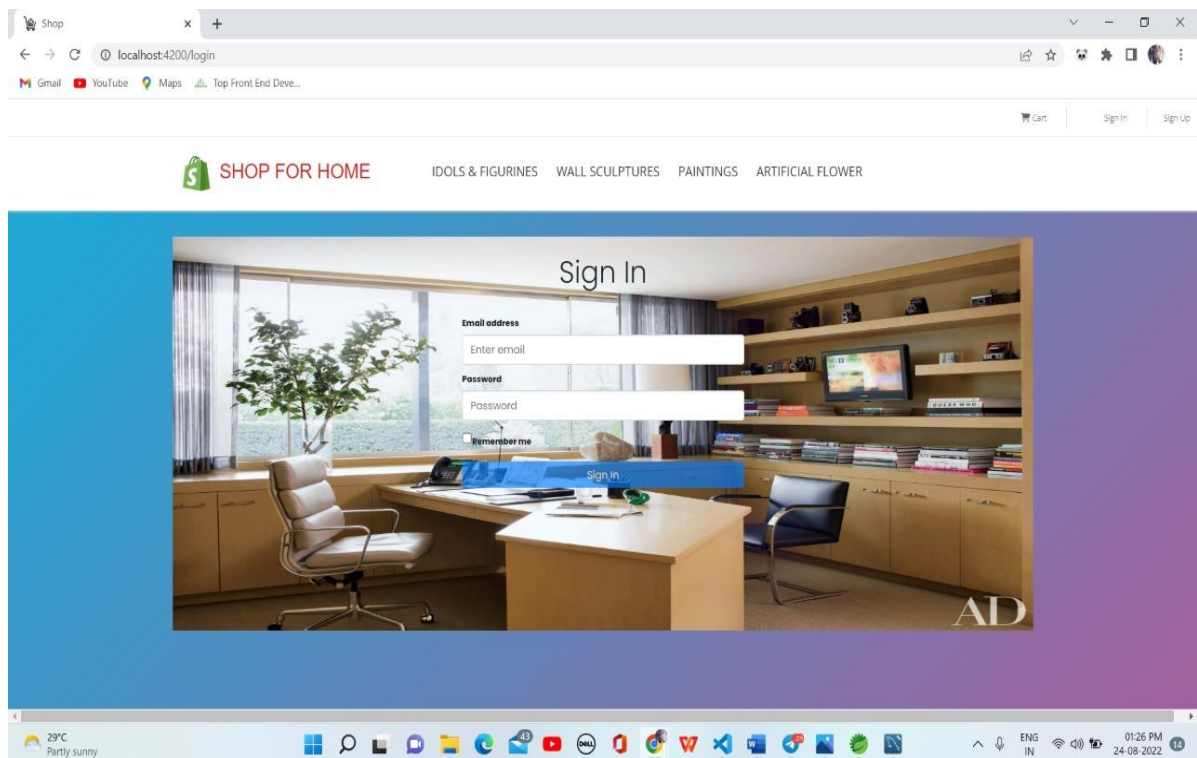
USER FUNCTIONALITY:



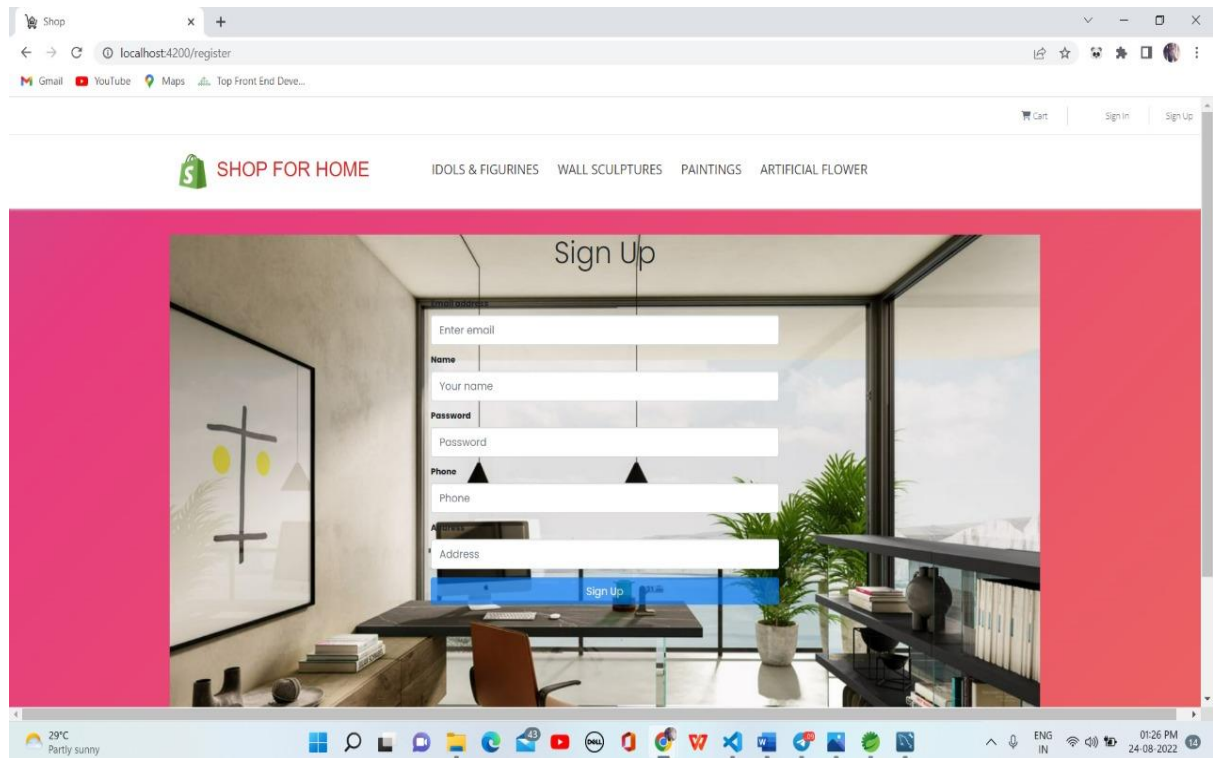
Home page



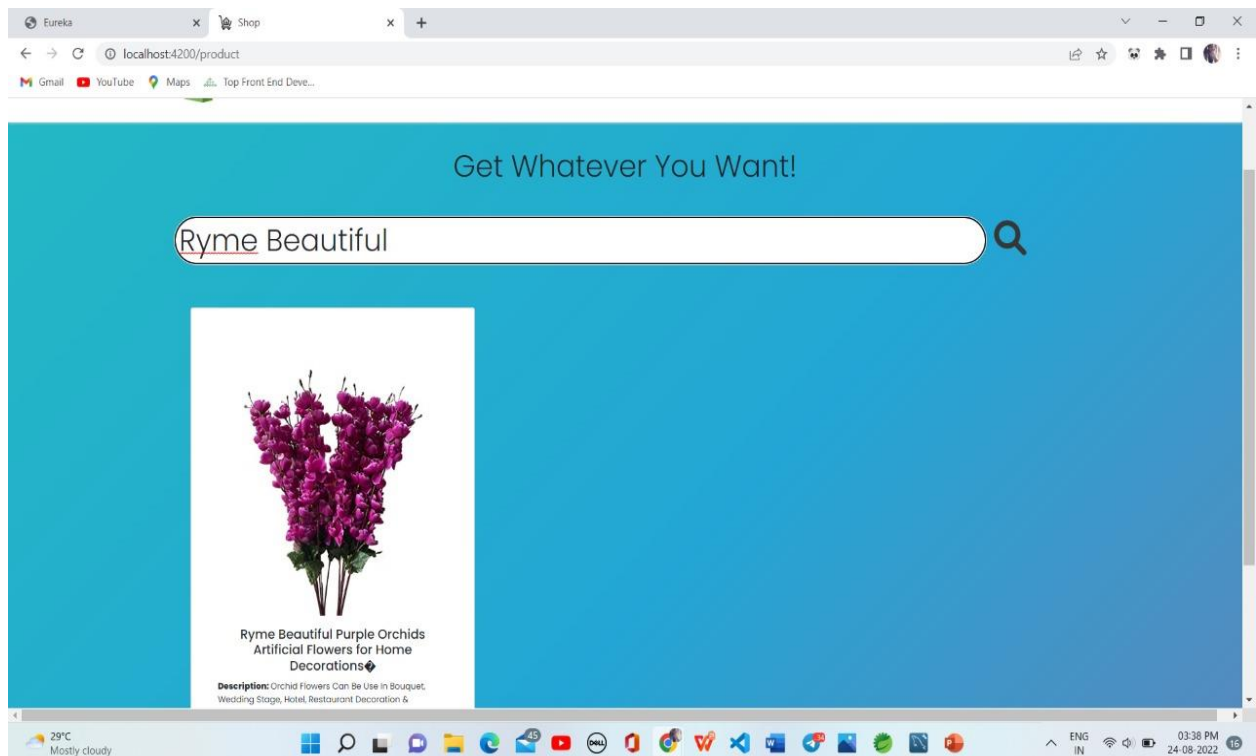
Cart page



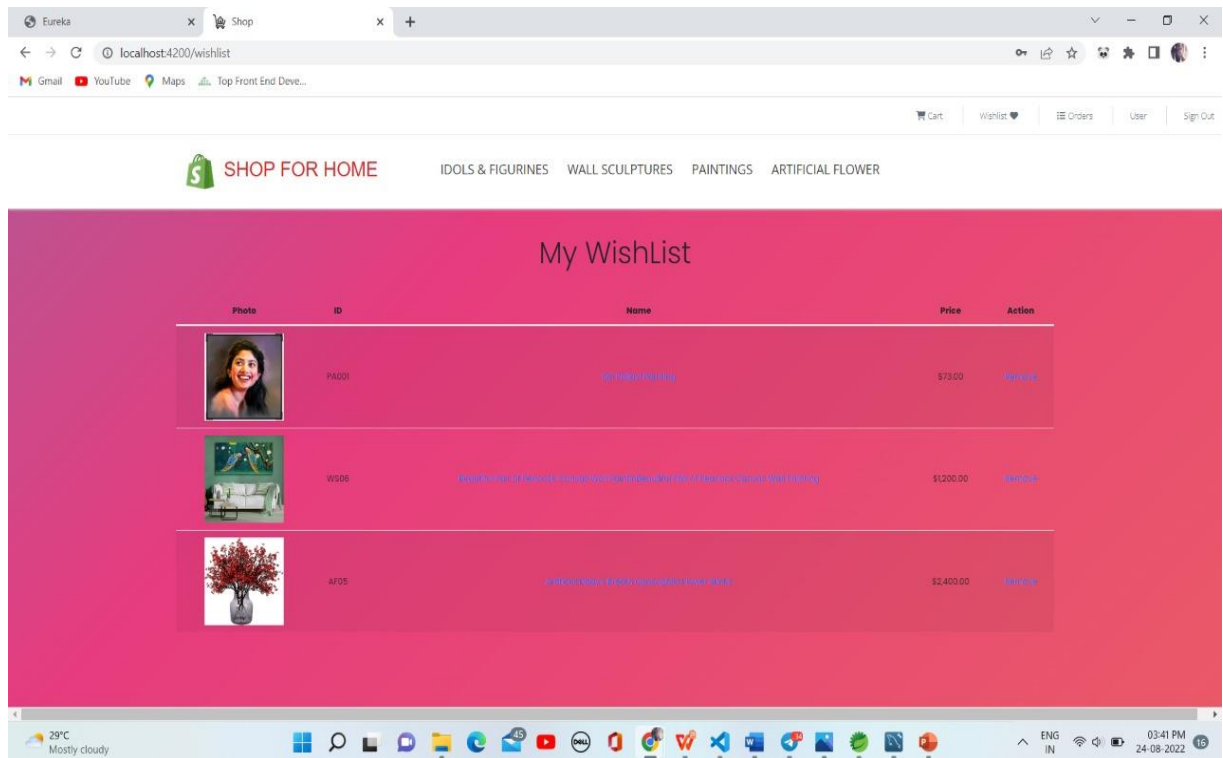
Sign In page



Sign Up page

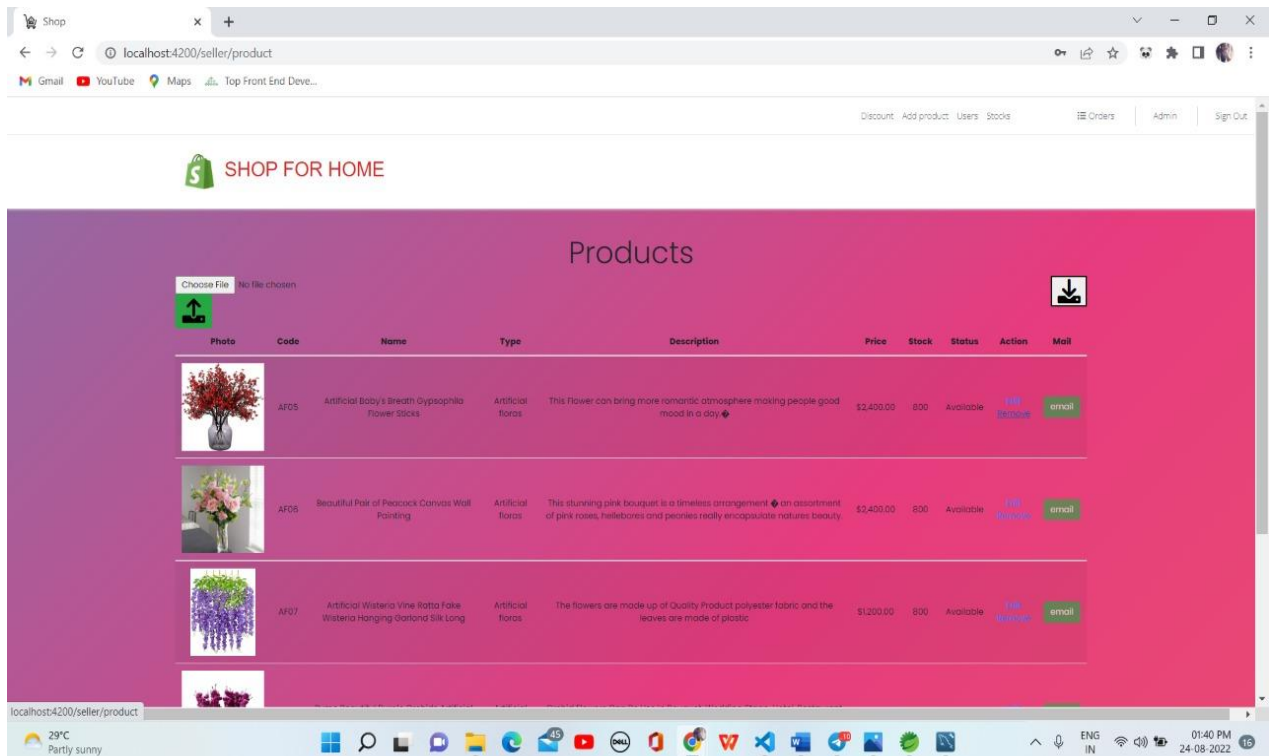


Product Search page

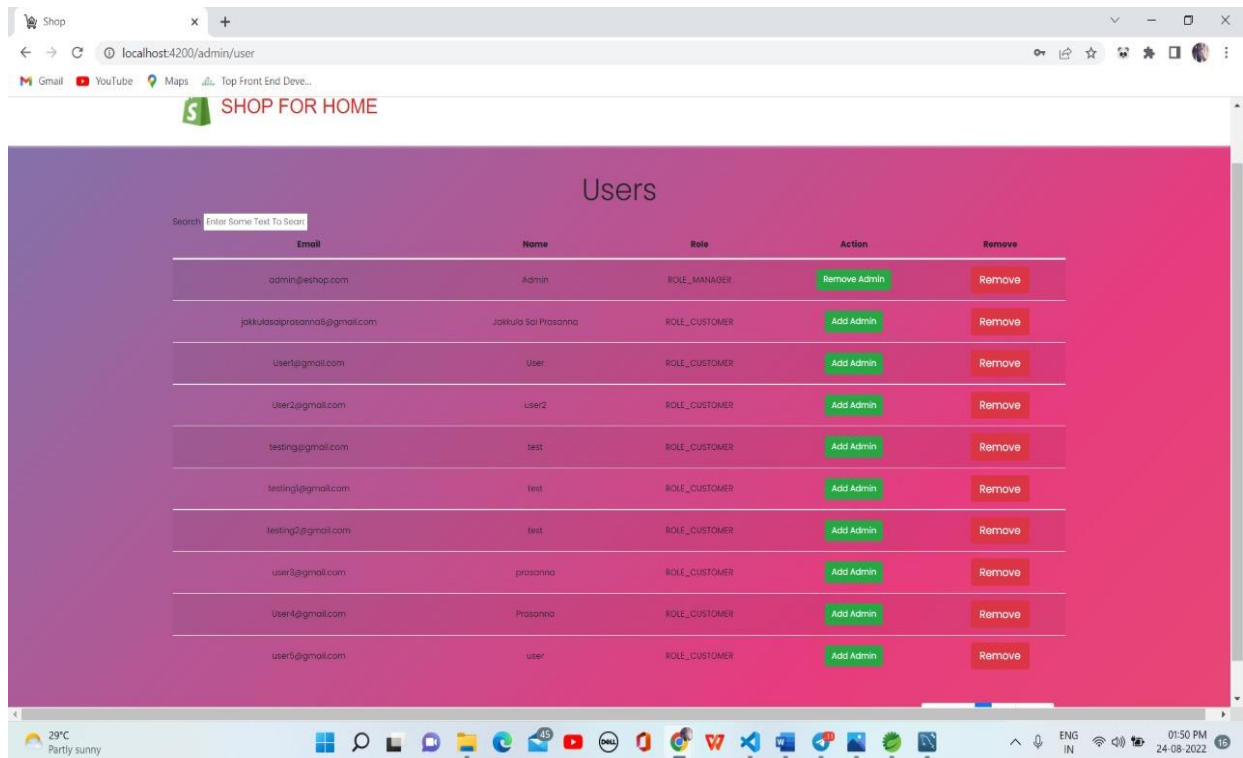


Wish list page

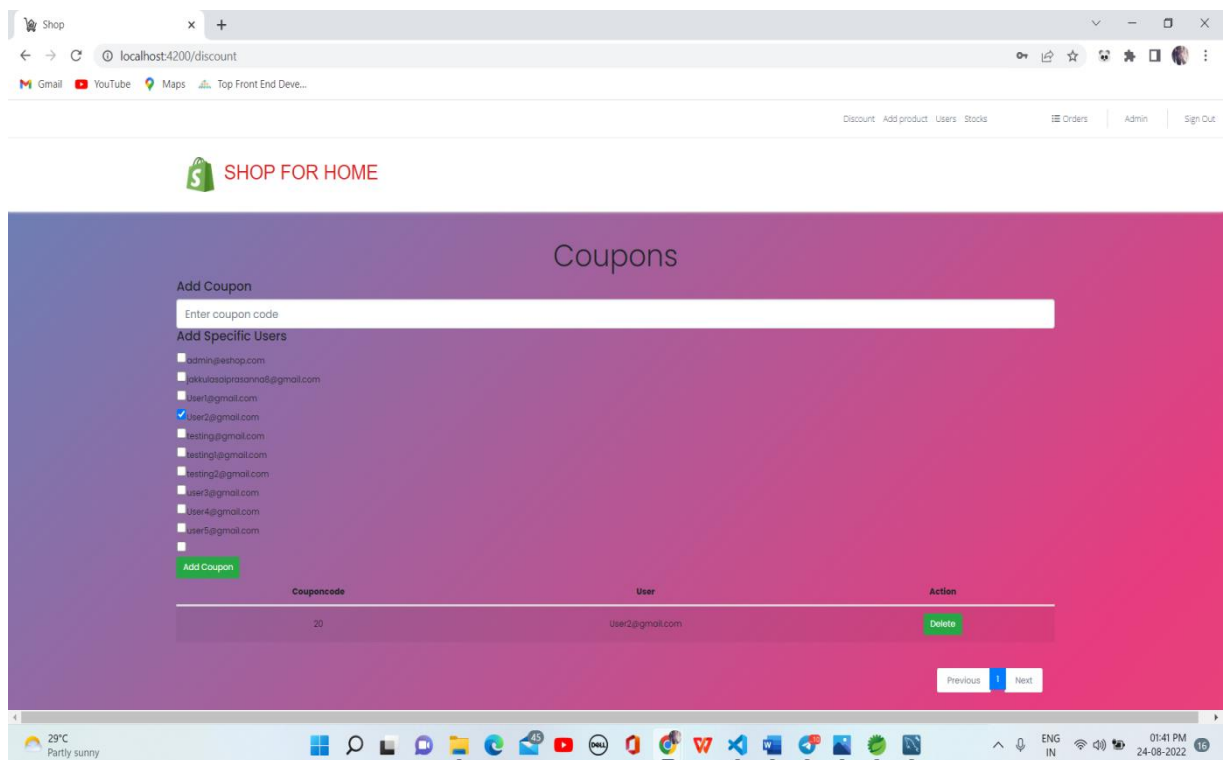
ADMIN FUNCTIONALITY:



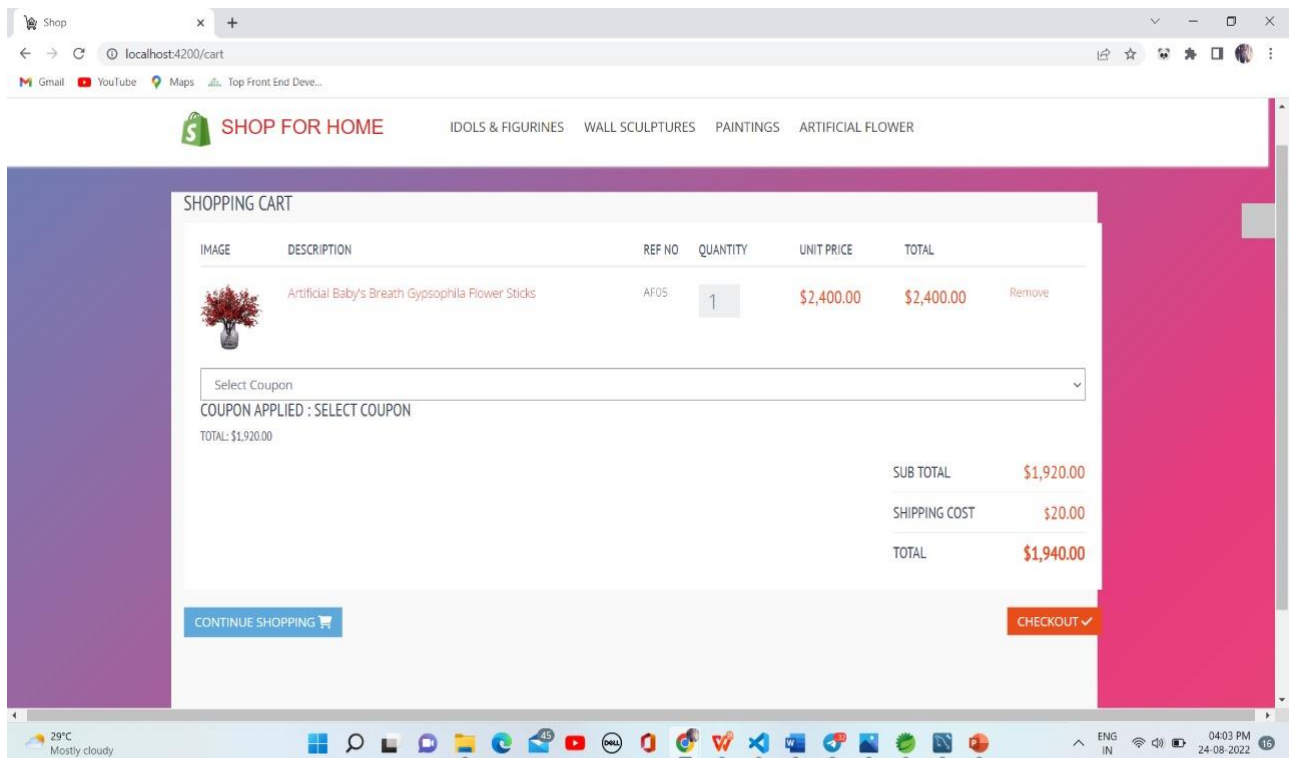
Stock page



User view page



Coupon page

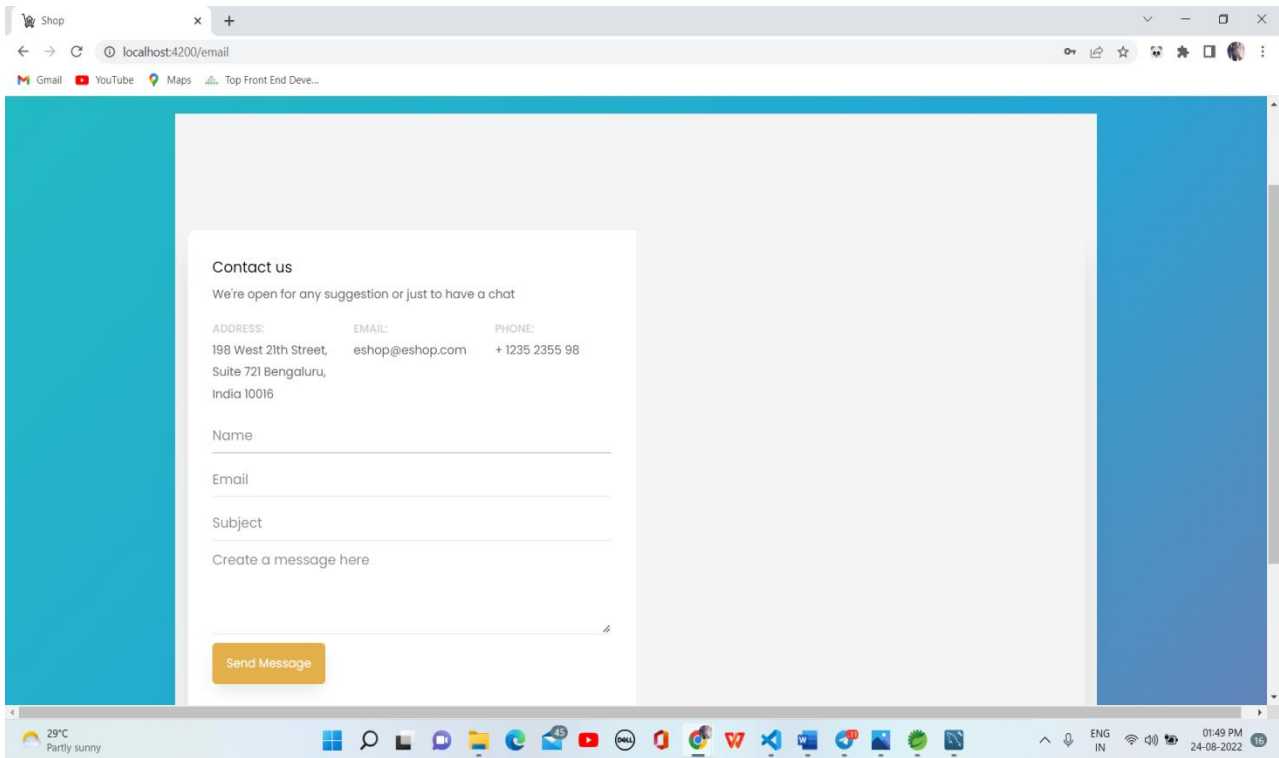


Discount page

A1										
ProductId										
	A	B	C	D	E	F	G	H	I	J
1	ProductId	ProductNa	Price	Stock	Description	ProductLoc	CategoryType			
2	IF04	Krishna M	2600	500	Krishna en	https://cd	0			
3	IF05	Naqsh Pol	2600	500	Laughing B	https://ho	0			
4	IF06	Miraya Ho	800	500	Horses hav	https://ho	0			
5	IF07	Miraya Pol	1200	500	Watching I	https://ho	0			
6	IF08	Glowing Tr	1500	500	Designed a	https://cd	0			
7	WS05	Rock Clim	2000	250	The Rock C	https://cd	1			
8	IF09	Black Pant	800	500	The large a	https://en	0			
9	WS06	BIRDS OF F	600	250	he Birds of	https://cd	1			
10	WS07	Teal Frame	750	250	Teal Frame	https://en	1			
11	WS08	Eiffel Tow	1000	250	It's made c	https://wc	1			
12	WS09	Stone Met	860	250	Stone Met	https://en	1			
13	WS10	Cecille wit	1200	250	Our Cecille	https://en	1			
14	WS11	Samara Ab	950	250	Elevate the	https://en	1			
15	WS12	World Map	1000	600	This exquis	https://en	1			
16										
17										
18										

Bulk upload

Mail to Customer: If the product stock is less than 10 the mail will be sent to the admin as soon as the application starts providing the product ID and product Name.



The screenshot shows a web browser window with the address bar displaying 'localhost:4200/email'. The page has a blue header and a white contact form. The form includes a title 'Contact us', a subtitle 'We're open for any suggestion or just to have a chat', and contact details: ADDRESS (198 West 21th Street, Suite 721 Bengaluru, India 10016), EMAIL (eshop@eshop.com), and PHONE (+1235 2355 98). Below this are input fields for Name, Email, and Subject, followed by a text area for the message and a 'Send Message' button.

Shop

localhost:4200/email

Gmail YouTube Maps Top Front End Deve...

Contact us

We're open for any suggestion or just to have a chat

ADDRESS: 198 West 21th Street,
Suite 721 Bengaluru,
India 10016

EMAIL: eshop@eshop.com

PHONE: +1235 2355 98

Name

Email

Subject

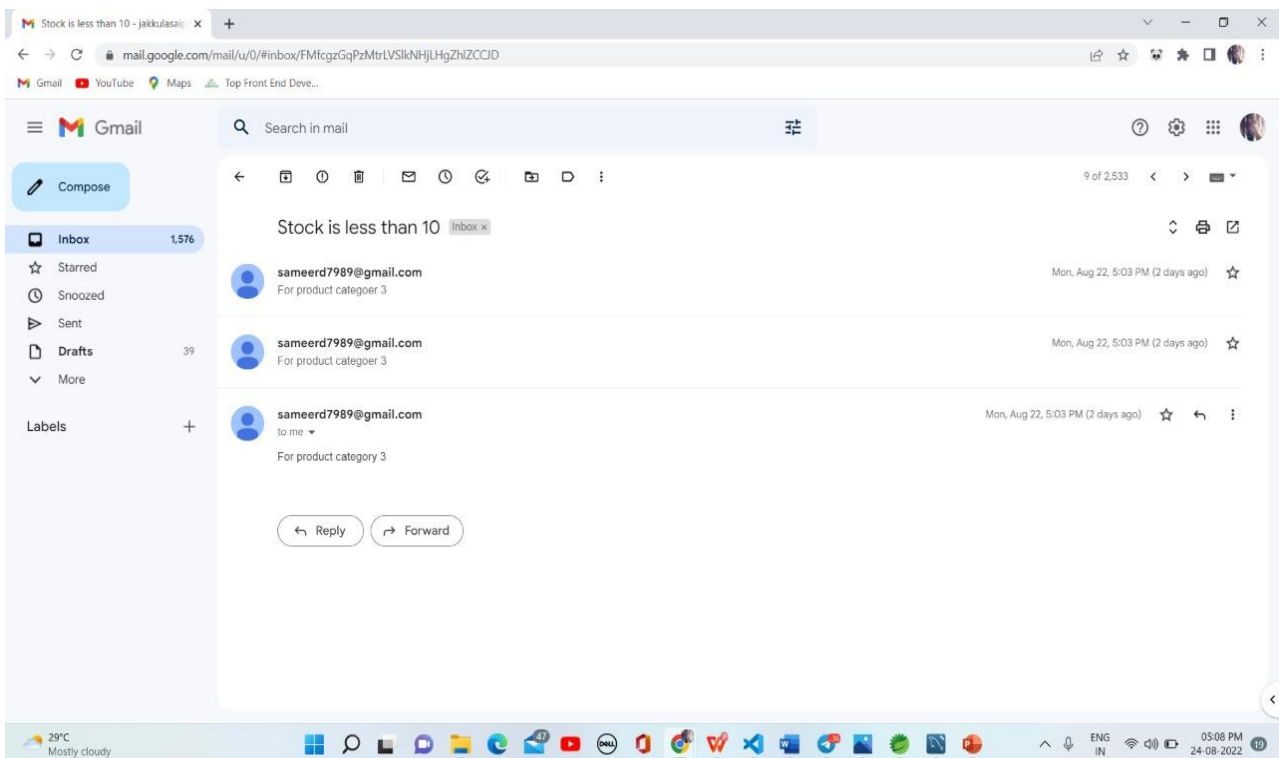
Create a message here

Send Message

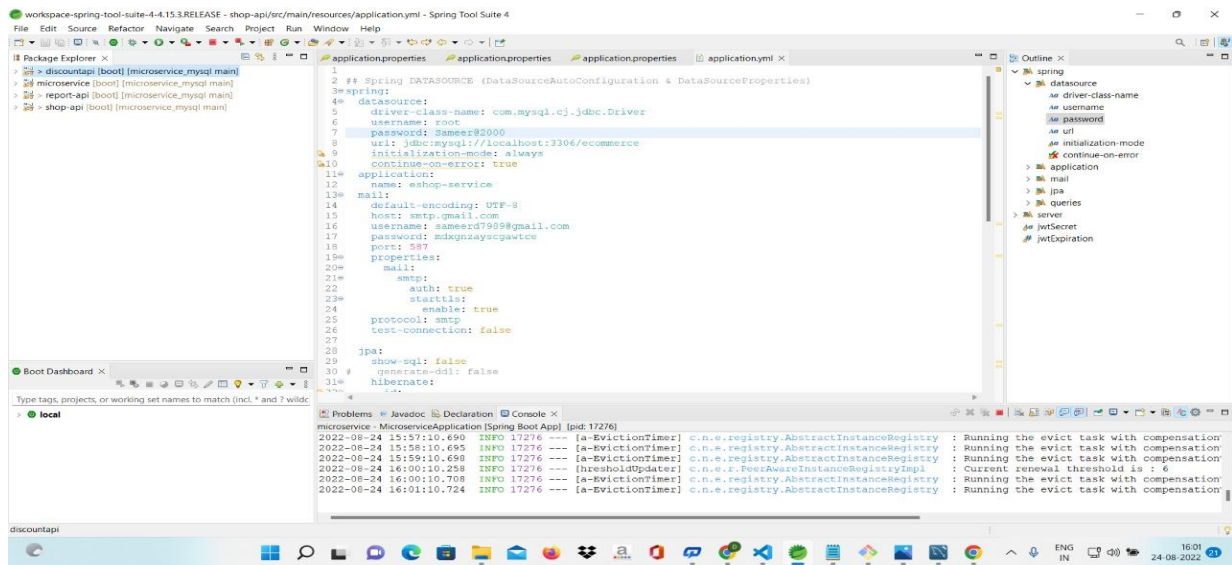
29°C Partly sunny

Windows taskbar icons: File Explorer, Edge, Chrome, VS Code, etc.

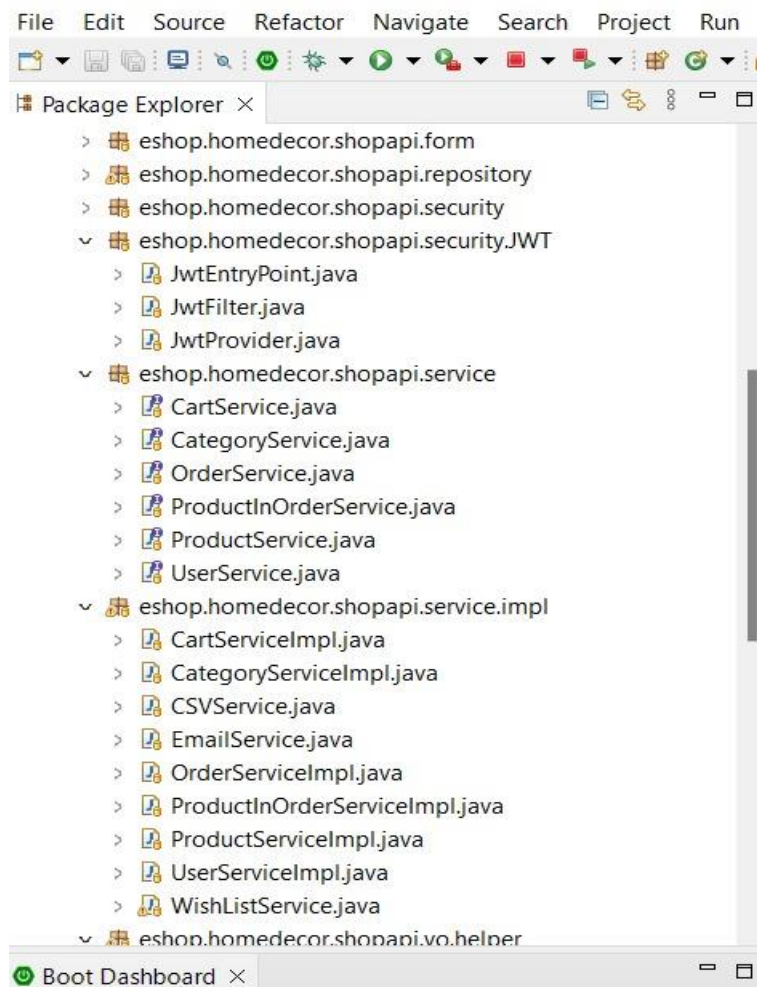
01:49 PM 24-08-2022



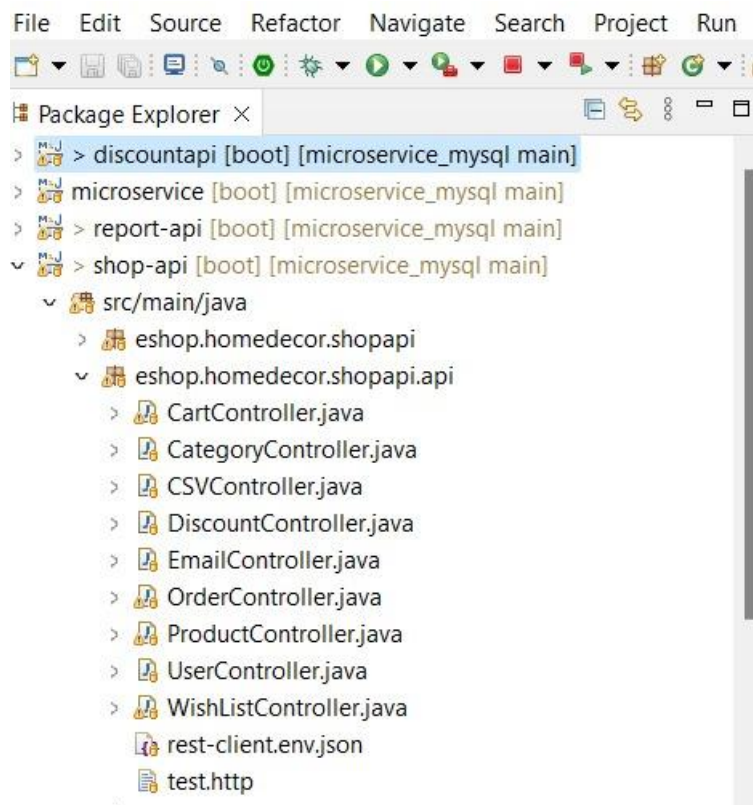
BACKEND SCREENSHOTS



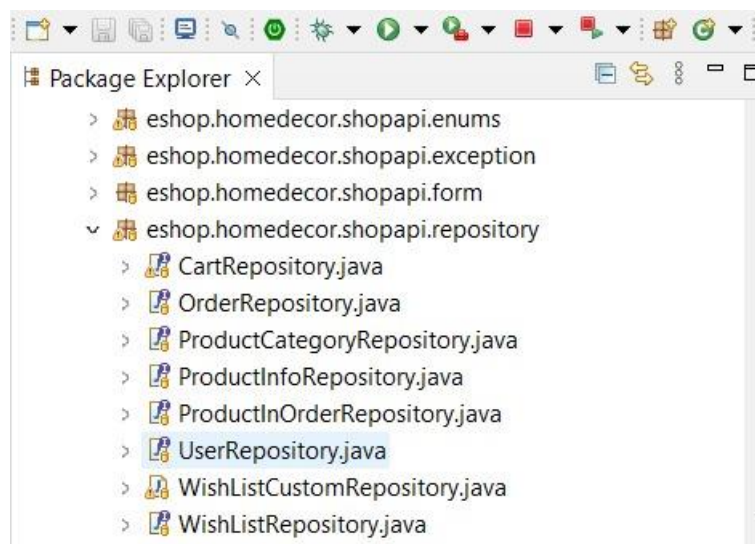
Backend Structure



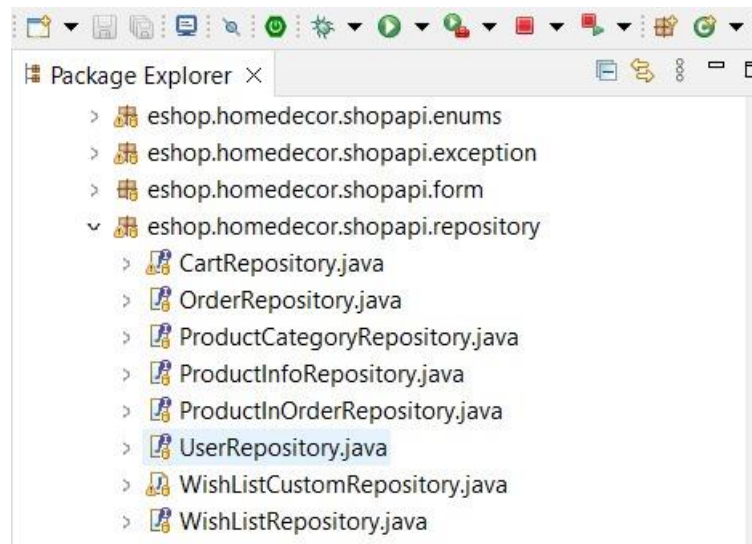
Service page



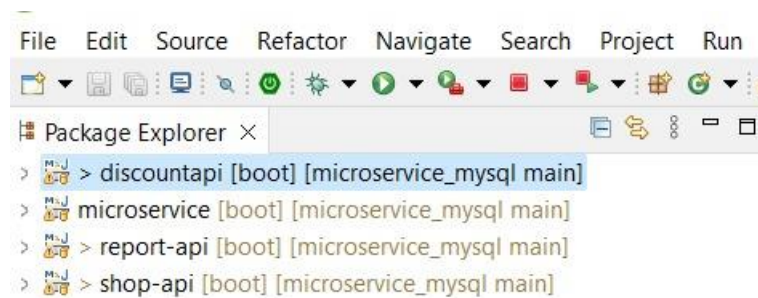
API page



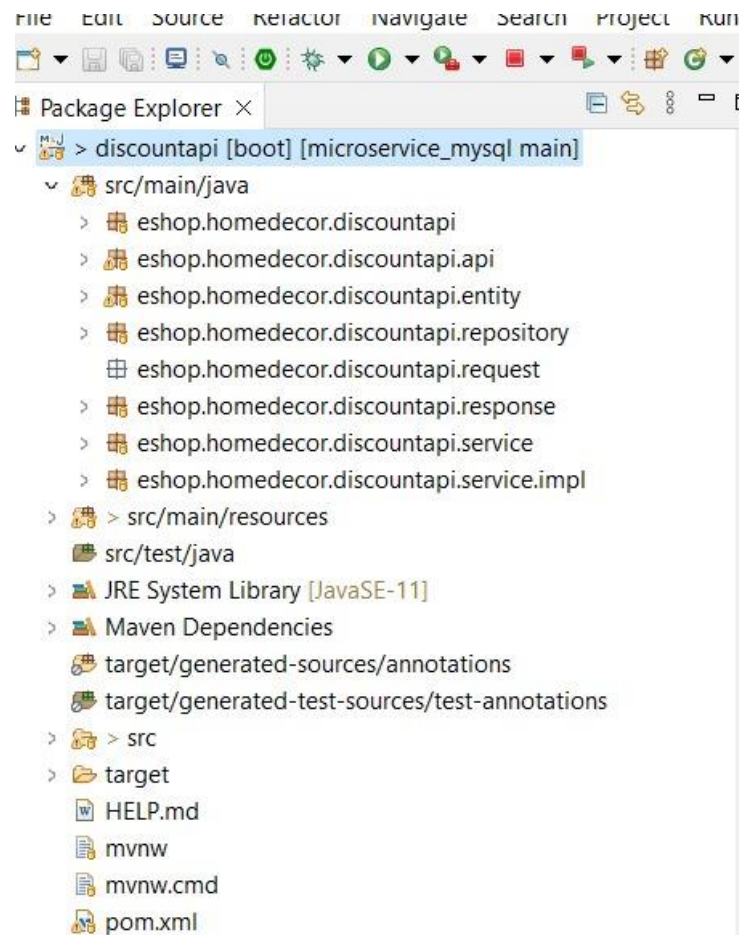
Entity page



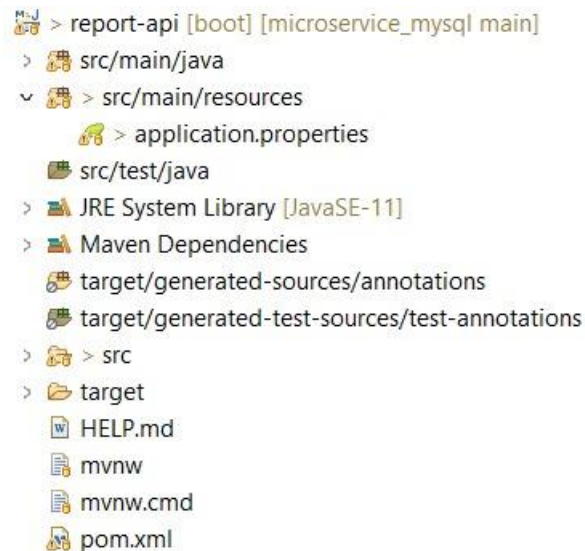
Repository page



Microservice Based Structure



Discount-Service-API



Report-api

```

MJD > shop-api [boot] [microservice_mysql main]
> src/main/java
> src/main/resources
> src/test/java
> JRE System Library [JavaSE-11]
> Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
> src
> target
  Dockerfile
  mvnw
  mvnw.cmd
  pom.xml

```

Shop-api

1. Creating Repository in Github

```

MINGW64/c:/Users/91630/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql
D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql (main)
$ git config --global user.email dsameer0786@gmail.com
D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql (main)
$ git config --global user.name sameer351
D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql (main)
$ git remote add origin https://github.com/sameer351/shopforhome.git
usage: git remote add [<options>] <name> <url>

-f, --fetch          fetch the remote branches
--tags              import all tags and associated objects when fetching
                   or do not fetch any tag at all (--no-tags)
-t, --track <branch> branch(es) to track
-m, --master <branch> master branch
--mirror[=<push|fetch>] set up remote as a mirror to push to or fetch from

D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql (main)
$ PC
D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql (main)
$ git remote add origin https://github.com/sameer351/shopforhome.git
error: remote origin already exists.
D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql (main)
$ git add .
D:\300LAPTOP-GW8G4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql (main)
$ git commit -m "First commit"
[main 370f55e] First commit
695 files changed, 94367 insertions(+)
create mode 100644 CSV_File_Bulk_Upload.csv
create mode 100644 backend/discountapi/.gitignore
create mode 100644 backend/discountapi/mvnw
create mode 100644 backend/discountapi/pom.xml
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/DiscountApplication.java
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/api/DiscountController.java
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/entity/Discount.java
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/repository/DiscountRepository.java
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/response/ResponseMessage.java
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/service/DiscountService.java
create mode 100644 backend/discountapi/src/main/java/eshop/homedecor/discountapi/service/impl/DiscountServiceImpl.java
create mode 100644 backend/discountapi/src/main/resources/application.properties
create mode 100644 backend/microservice/.gitignore
create mode 100644 backend/microservice/.mvn/wrapper/maven-wrapper.jar
create mode 100644 backend/microservice/.mvn/wrapper/maven-wrapper.properties
create mode 100644 backend/microservice/mvnw
create mode 100644 backend/microservice/mvnw.cmd
create mode 100644 backend/microservice/pom.xml
create mode 100644 backend/microservice/src/main/java/eshop/homedecor/microservice/MicroserviceApplication.java

```



```
MINGW64/C:/Users/91630/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql
create mode 100644 frontend/src/assets/plugins/simple-line-icons/README.txt
create mode 100644 frontend/src/assets/plugins/simple-line-icons/fonts/Simple-Line-Icons.dev.svg
create mode 100644 frontend/src/assets/plugins/simple-line-icons/fonts/Simple-Line-Icons.eot
create mode 100644 frontend/src/assets/plugins/simple-line-icons/fonts/Simple-Line-Icons.svg
create mode 100644 frontend/src/assets/plugins/simple-line-icons/fonts/Simple-Line-Icons.ttf
create mode 100644 frontend/src/assets/plugins/simple-line-icons/fonts/Simple-Line-Icons.woff
create mode 100644 frontend/src/assets/plugins/simple-line-icons/icons-lte-ie7.js
create mode 100644 frontend/src/assets/plugins/simple-line-icons/simple-line-icons.css
create mode 100644 frontend/src/assets/plugins/simple-line-icons/simple-line-icons.min.css
create mode 100644 frontend/src/assets/plugins/smooth-scroll/smooth-scroll.js
create mode 100644 frontend/src/assets/plugins/uniform/README.md
create mode 100644 frontend/src/assets/plugins/uniform/css/uniform.default.css
create mode 100644 frontend/src/assets/plugins/uniform/css/uniform.default.min.css
create mode 100644 frontend/src/assets/plugins/uniform/css/uniform.default.scss
create mode 100644 frontend/src/assets/plugins/uniform/images/bg-input-focus.png
create mode 100644 frontend/src/assets/plugins/uniform/images/bg-input.png
create mode 100644 frontend/src/assets/plugins/uniform/images/sprite.png
create mode 100644 frontend/src/assets/plugins/uniform/images/sprite_original.png
create mode 100644 frontend/src/assets/plugins/uniform/jquery.uniform.js
create mode 100644 frontend/src/assets/plugins/uniform/jquery.uniform.min.js
create mode 100644 frontend/src/assets/plugins/zoom/jquery.zoom.js
create mode 100644 frontend/src/assets/plugins/zoom/jquery.zoom.min.js
create mode 100644 frontend/src/environments/environment.prod.ts
create mode 100644 frontend/src/environments/environment.ts
create mode 100644 frontend/src/favicon.ico
create mode 100644 frontend/src/karma.conf.js
create mode 100644 frontend/src/main.ts
create mode 100644 frontend/src/polyfills.ts
create mode 100644 frontend/src/styles.css
create mode 100644 frontend/src/test.ts
create mode 100644 frontend/src/tsconfig.app.json
create mode 100644 frontend/src/tsconfig.spec.json
create mode 100644 frontend/src/tslint.json
create mode 100644 frontend/tsconfig.json
create mode 100644 frontend/tslint.json
create mode 100644 frontend1.zip
create mode 100644 mysql.sql

316308LAPTOP-GXKRG4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql (main)
$ git push -u origin main
Enumerating objects: 817, done.
Counting objects: 100% (817/817), done.
Delta compression using up to 8 threads
Compressing objects: 100% (744/744), done.
Writing objects: 100% (816/816), 98.78 MiB | 490.00 KiB/s, done.
Total 816 (delta 100), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (100/100), done.
remote: warning: See http://git.io/EPt8g for more information.
remote: warning: file frontend1.zip is 91.25 MiB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/sameer351/shopforhome.git
 370f55e..370f55e main -> main
branch 'main' set up to track 'origin/main'.

316308LAPTOP-GXKRG4E6 MINGW64 ~/Downloads/microservice_mysql (33)/microservice_mysql/microservice_mysql/microservice_mysql (main)
$
```

github.com/sameer351/shopforhome

Gmail YouTube Maps Routing in Angular... WhatsApp

Search or jump to... Pull requests Issues Marketplace Explore

sameer351 / shopforhome Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

About

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

sameer351 first commit 370f55e 5 minutes ago 2 commits

backend first commit 5 minutes ago

frontend first commit 5 minutes ago

CSV_File_Bulk_Upload.csv first commit 5 minutes ago

README.md first commit 12 minutes ago

frontend1.zip first commit 5 minutes ago

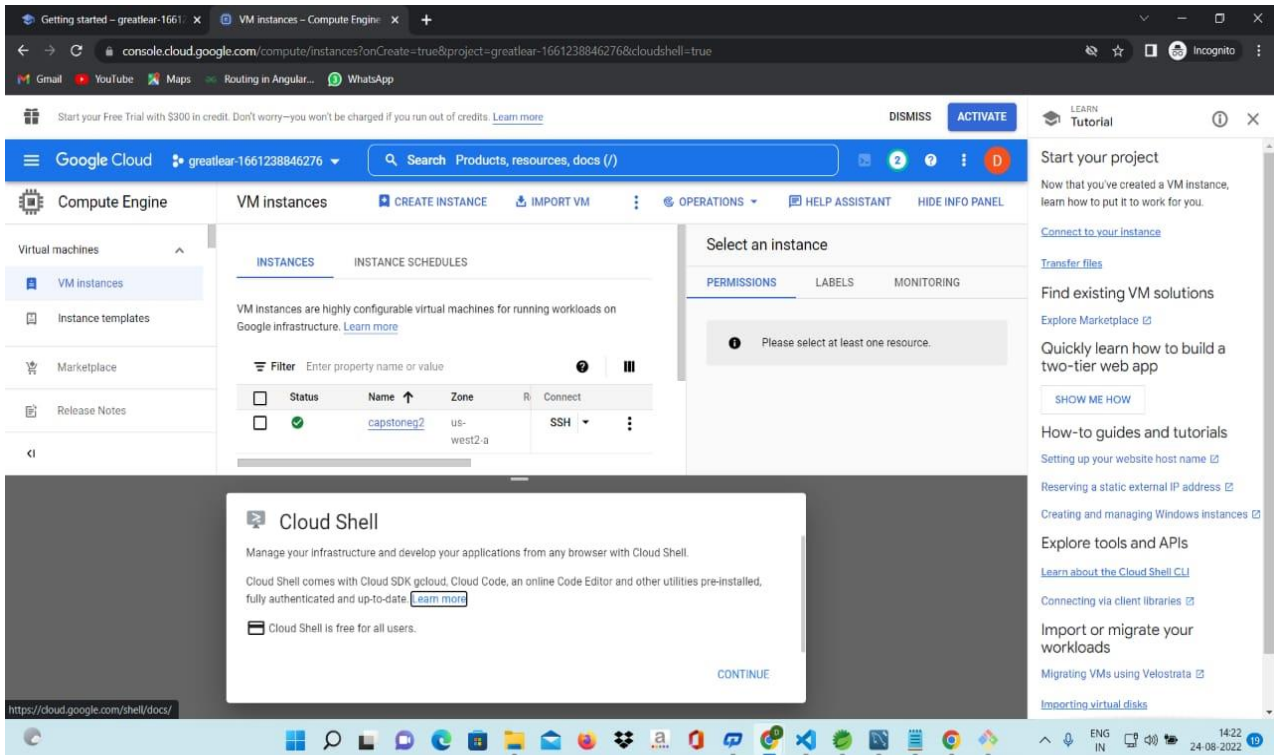
mysql.sql first commit 5 minutes ago

README.md

Online Shop Application

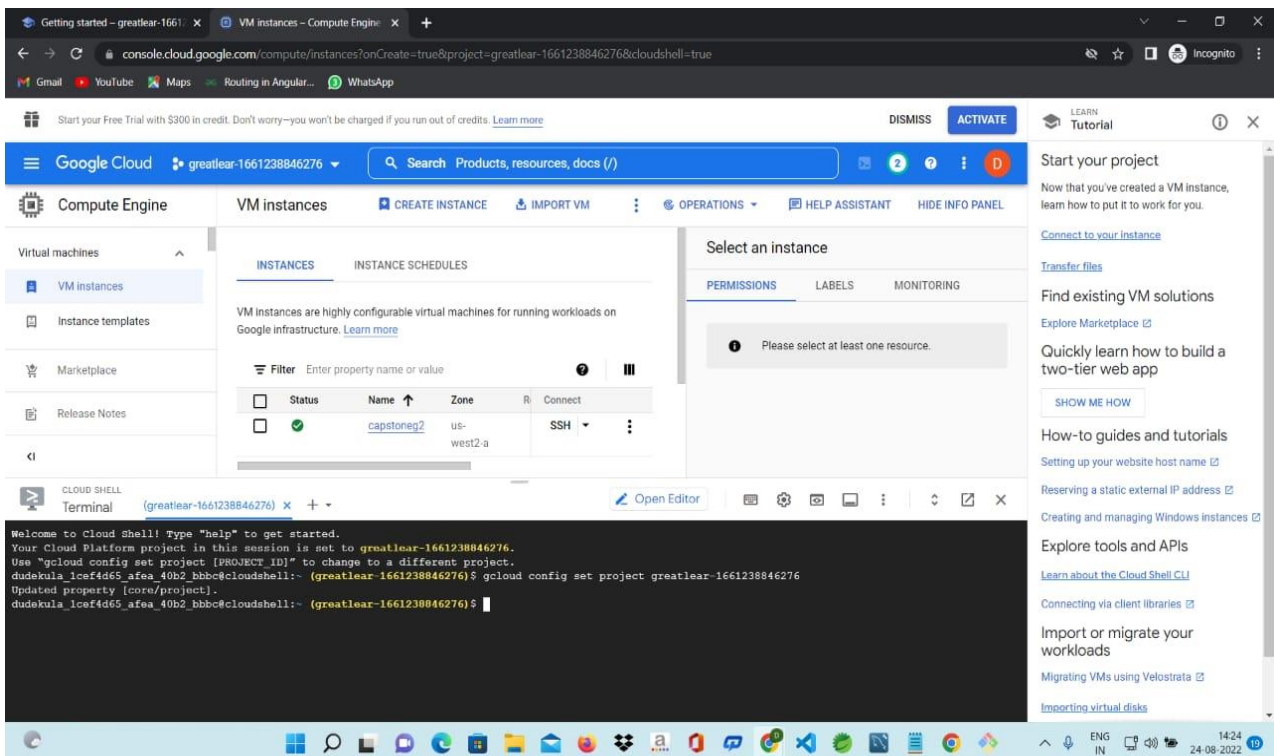
A full-stack Online Shop web application using Spring Boot 2 and Angular 7.

2. Creating VM instance

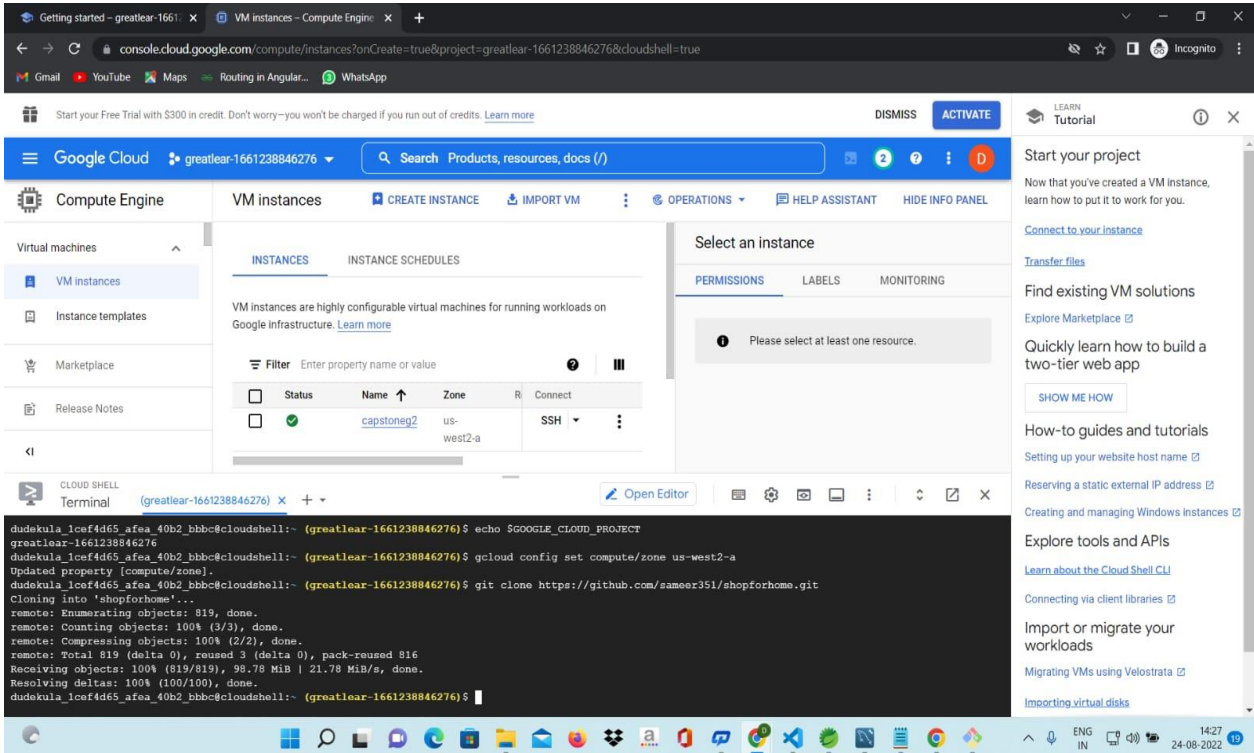


3. Open Cloud Shell and run commands

1. `gcloud config set project greatlear-1661238846276`



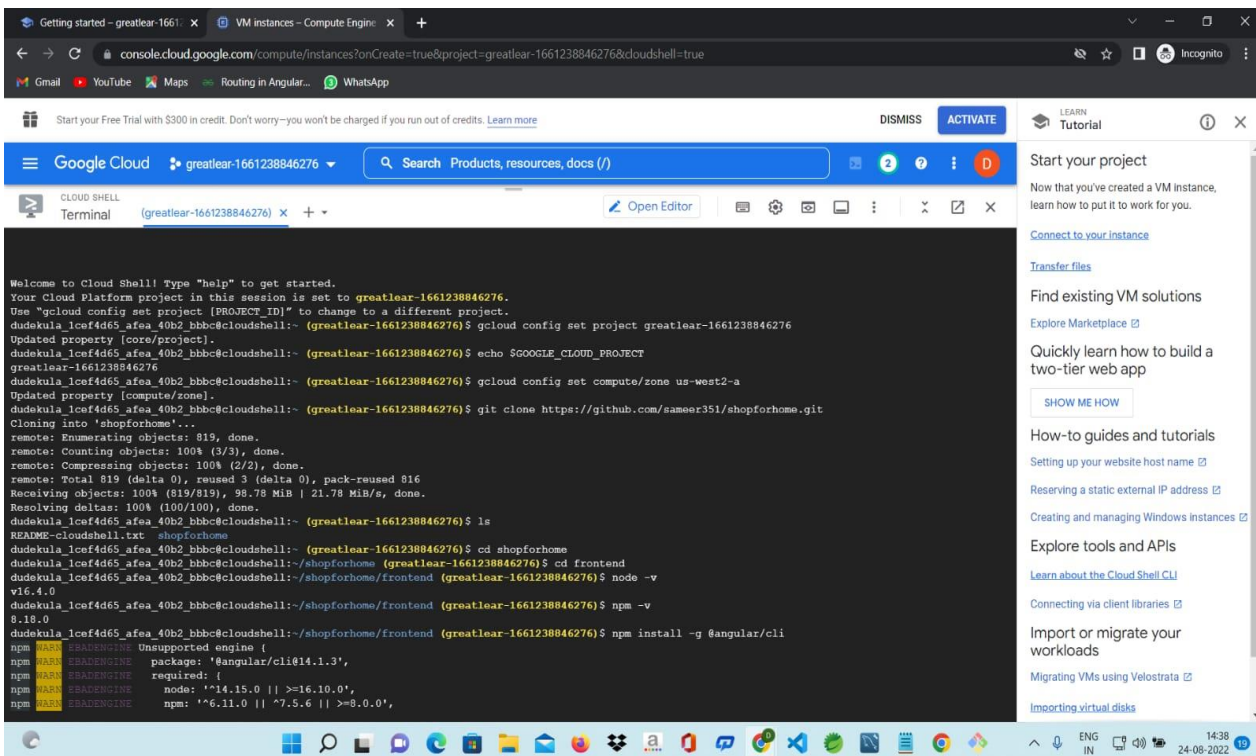
2. `echo $GOOGLE_CLOUD_PROJECT`
3. `gcloud config set compute/zone us-west2-a`



The screenshot shows the Google Cloud console interface. The 'Compute Engine' section is active, displaying a list of VM instances. The instance 'capstone2' is highlighted. The 'CLOUD SHELL' terminal is open, showing the execution of the commands from the previous list. The terminal output shows the project ID, the zone being set, and the cloning of the repository.

```
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ echo $GOOGLE_CLOUD_PROJECT
greatlear-1661238846276
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ gcloud config set compute/zone us-west2-a
Updated property [compute/zone].
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ git clone https://github.com/sameer351/shopforhome.git
Cloning into 'shopforhome'...
remote: Enumerating objects: 819, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 819 (delta 0), reused 3 (delta 0), pack-reused 816
Receiving objects: 100% (819/819), 98.78 MiB | 21.78 MiB/s, done.
Resolving deltas: 100% (100/100), done.
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $
```

4. `git clone http://github.com/sameer351/shopforhome.git`
5. `ls`
6. `cd shopforhome`
7. `cd frontend`
8. `node -v`
9. `npm -v`
10. `npm install -g @angular/cli`



The screenshot shows the Google Cloud console interface. The 'Compute Engine' section is active, displaying a list of VM instances. The instance 'capstone2' is highlighted. The 'CLOUD SHELL' terminal is open, showing the execution of the commands from the previous list. The terminal output shows the project ID, the zone being set, the cloning of the repository, and the installation of Angular CLI.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to greatlear-1661238846276.
Use "gcloud config set project [PROJECT ID]" to change to a different project.
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ gcloud config set project greatlear-1661238846276
Updated property [core/project].
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ echo $GOOGLE_CLOUD_PROJECT
greatlear-1661238846276
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ gcloud config set compute/zone us-west2-a
Updated property [compute/zone].
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ git clone https://github.com/sameer351/shopforhome.git
Cloning into 'shopforhome'...
remote: Enumerating objects: 819, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 819 (delta 0), reused 3 (delta 0), pack-reused 816
Receiving objects: 100% (819/819), 98.78 MiB | 21.78 MiB/s, done.
Resolving deltas: 100% (100/100), done.
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ ls
README-cloudshell.txt  shopforhome
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~ (greatlear-1661238846276) $ cd shopforhome
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~/shopforhome (greatlear-1661238846276) $ cd frontend
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ node -v
v16.4.0
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ npm -v
8.18.0
dudekula_1cef4d65_afa4_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ npm install -g @angular/cli
npm WARN EBADENGINE Unsupported engine {
  package: '@angular/cli@14.1.3',
  required: {
    node: '^14.15.0 || >=16.10.0',
    npm: '^6.11.0 || ^7.5.6 || >=8.0.0',
  },
}
```


The screenshot shows a Google Cloud console window with a terminal. The terminal output shows the command `npm install -g @angular/cli` being executed. The output indicates that the command failed due to an unsupported engine. The error message is: `npm WARN EBADENGINE Unsupported engine { package: '@angular/cli@14.1.3', required: { node: '^14.15.0 || >=16.10.0', npm: '^6.11.0 || ^7.5.6 || >=8.0.0', yarn: '>= 1.13.0' }, current: { node: 'v16.4.0', npm: '8.18.0' } }`. The terminal also shows the command `npm install -g @angular-devkit/architect@0.1401.3` and `npm install -g @angular-devkit/core@14.1.3`, both of which also failed with the same error message.

11. `npm install -g typescript`

The screenshot shows a Google Cloud console window with a terminal. The terminal output shows the command `nvm install 16.10` being executed. The output indicates that the command was successful. The terminal also shows the command `npm install` being executed, which failed due to an unsupported engine. The error message is: `npm WARN EBADENGINE Unsupported engine { package: '@angular/cli@14.1.3', required: { node: '^14.15.0 || >=16.10.0', npm: '^6.11.0 || ^7.5.6 || >=8.0.0', yarn: '>= 1.13.0' }, current: { node: 'v16.4.0', npm: '8.18.0' } }`. The terminal also shows the command `npm install -g typescript` being executed, which was successful.

12. `nvm install 16.10`

13. `npm install`

The screenshot shows a Google Cloud Shell terminal window. The terminal output shows the following commands and their results:

```
Computing checksum with sha256sum
Checksums matched!
Now using node v16.10.0 (npm v7.24.0)
dudekula_1cef4d65_afea_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ node -v
v16.10.0
dudekula_1cef4d65_afea_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ npm install -g @angular/cli
added 208 packages, and audited 209 packages in 7s

25 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
dudekula_1cef4d65_afea_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ npm install -g typescript
added 1 package, and audited 2 packages in 2s

found 0 vulnerabilities
dudekula_1cef4d65_afea_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ npm install
npm WARN deprecated @schematics/update@0.13.10: This was an internal-only Angular package up through Angular v11 which is no longer used or maintained. Up
grade Angular to v12+ to remove this dependency.
added 1127 packages, and audited 1128 packages in 3m

3 packages are looking for funding
  run `npm fund` for details

87 vulnerabilities (4 low, 31 moderate, 51 high, 11 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force
```

14. ng serve

The screenshot shows a Google Cloud Shell terminal window. The terminal output shows the following commands and their results:

```
87 vulnerabilities (4 low, 31 moderate, 51 high, 11 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

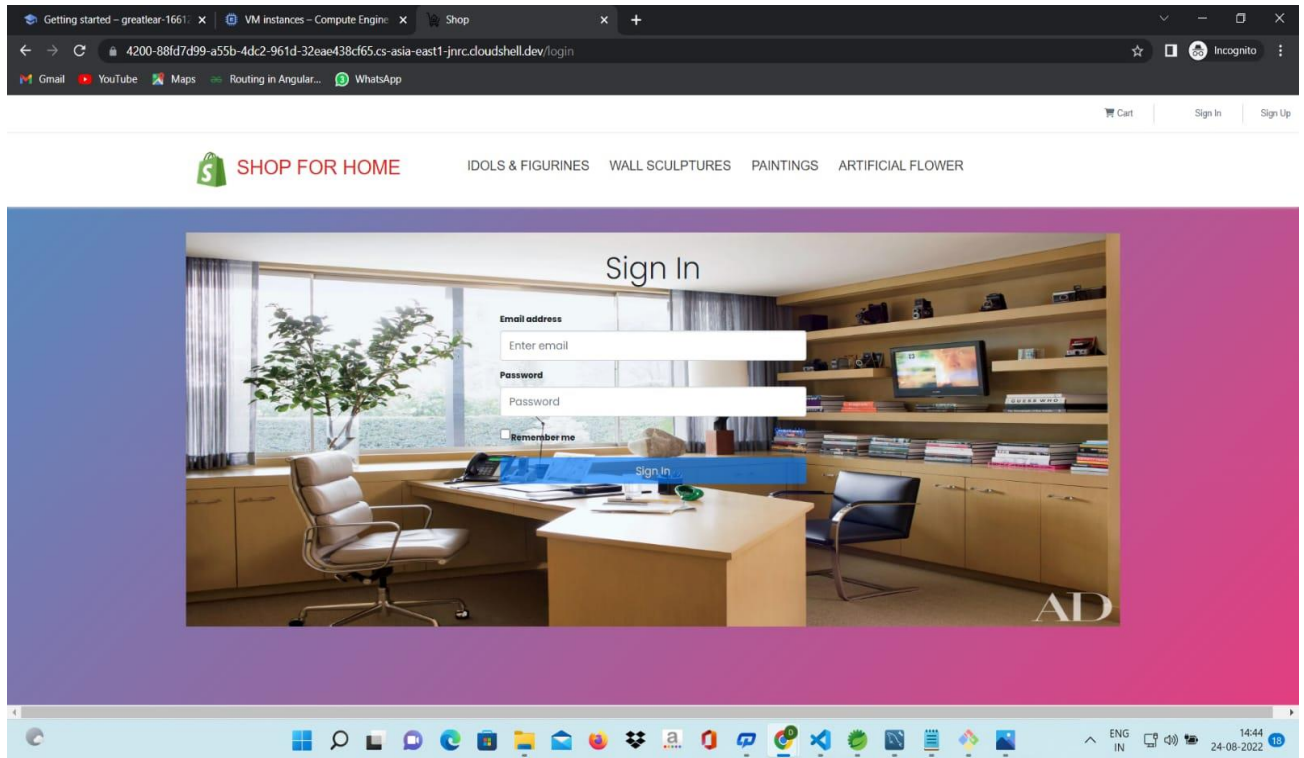
Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
dudekula_1cef4d65_afea_40b2_bbbc@cloudshell:~/shopforhome/frontend (greatlear-1661238846276) $ ng serve
Your global Angular CLI version (14.1.3) is greater than your local version (7.3.10). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
(node:2093) [DEP0111] DeprecationWarning: Access to process.binding('http_parser') is deprecated.
(Use `node --trace-deprecation ...` to show where the warning was created)
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
18% building modules 71/91 modules 20 active ...s/add-product/add-product.component.cssBrowserslist: caniuse-lite is outdated. Please run next command `n
pm update caniuse-lite browserslist`

Date: 2022-08-24T09:08:08.505Z
Hash: c1f496f8fac6184a40b2
Time: 22160ms
chunk (main) main.js, main.js.map (main) 855 kB [initial] [rendered]
chunk (polyfills) polyfills.js, polyfills.js.map (polyfills) 223 kB [initial] [rendered]
chunk (runtime) runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk (scripts) scripts.js, scripts.js.map (scripts) 133 kB [rendered]
chunk (styles) styles.js, styles.js.map (styles) 835 kB [initial] [rendered]
chunk (vendor) vendor.js, vendor.js.map (vendor) 4.91 MB [initial] [rendered]
[webpack]: Compiled successfully.
```

Shop website using cloud



8. CONCLUSION AND FUTURE SCOPE

E-commerce has unquestionably risen to prominence in our culture. Future-proof businesses will be those that take e-commerce seriously and invest enough money in its expansion. E-commerce is a whole-business endeavor rather than an IT problem. The businesses that utilize it as an excuse to change their operational procedures stand to gain the most. E-commerce is also a useful technology that gives customers access to businesses and organizations throughout the globe.

Despite a number of obstacles, the ecommerce sector has a lot of potential. The success of e-commerce enterprises is greatly influenced by social media since customers use these platforms to keep informed. We anticipate that the usage of drones will transform how the ecommerce industry operates in the future. The products might be delivered to the customer's door on the same day as the order was placed with its assistance. Everyone has a mobile phone these days, which has boosted the development of new eCommerce business apps. In order to interact with customers, many businesses are converting to an app-only strategy.

REFERENCES

- [1] Albert H., Judd, Rivers, (2006) "Creating a winning E-Business", Wagner Course Technology Thomson Learning, pp. 37-255.
- [2] Alawneh A., and Hattab E, (2007) "E-Business Value Creation: An Exploratory Study, Proceedings of the Seventh International Conference on Electronic Business", Taipei, pp. 181-188.
- [3] Alawneh A., and Hattab E (2009). "International Arab Journal of eTechnology", Vol. 1, No. 2, pp. 1-8
- [4] Amit B. and Steve M. (2007), "How to Plan E-Business Initiatives in Established Companies", Vol. 49, No. 1, pp. 11-22
- [5] Aranda-M., G. and Stewart, P. (2005), "Barriers to E-Business Adoption in construction international literature review", pp. 33-49
- [6] Ayo, Charles K. (2006). "The Prospects of e-Commerce Implementation in Nigeria, Journal of Internet Banking and Commerce", Vol. 11, No.3, pp. 68-75 [7] Amar. K., Sohani, (2009), "Technology and Banking Sector", ICFAI University Press, pp. 1-39
- [8] Brahm C., (2009) "E-Business and Commerce Strategic Thinking and Practice", Houghton Mifflin, pp. 114-312.
- [9] Chiemeké, S. C., Ewuekpae, A. and Chete, F. (2006), "The Adoption of Internet Banking in Nigeria: An Empirical Investigation, Journal of Internet Banking and Commerce", vol. 11, No.3, pp 33-49
- [10] David W, (2001) "E-Commerce Strategy, Technologies and Applications", Tata McGraw Hill, pp. 3-143.
- [11] Daft, Richard L. (1982), " Bureaucratic Versus Nonbureaucratic Structure and the process of Innovation and Change", pp. 129-166
- [12] Earl, M. (2000), "Evolving the E-Business, Business Strategy Review", pp. 33-38
- [13] Eben.O (2003) "A Systematic Approach to E-Business Security", pp. 87-103
- [14] Hackbarth, G. & Kettinger W. J. (2000), "Building an E-Business Strategy:

Information Systems Management” pp. 78-90.

[15] Kalakota, R. and Robinson M. (1999), “E-Business: Roadmap for success”, Addison-Wesley, 112-149 19

[16] Karjaluoto H., Mattila M. (2002). “Electronic Banking in Finland: Consumer Beliefs and Reactions to a New Delivery Channel, Journal of Financial Services Marketing”, Vol. 6, No. 4, 2002, pp. 346–361

[17] Laudon, K. and Traver, C. (2008), “ E-Commerce: Business, Technology, Society”, 4th Edition, Prentice Hall, pp.48-67

[18] Melao, N. (2008), “ E-Business Processes and E-Business Process Modeling: A State-of-the-Art Overview”, pp. 54-89

[19] Mendo, F. and Fitzgerald, G. (2005), “Theoretical Approaches to Study SMEs E-Business Progression”, Journal of Computing and Information Technology, Vol. 13, No. 02, pp. 123-136

[20] Namita.R (2011). “Global Journal of Enterprise Information System”, Vol. 3, No. 1, pp. 17-31

[21] Peterson O,(2009), “Communications of the IBIMA”, Vol. 8, Issue No. 3, pp. 175-190

[22] Poon S. & Swatam (1997), “Small business use of Internet: Findings from Australian case studies”, Internation Marketing Review. Pp. 385-403

[23] Poon S. and Swatman P., (1999) “An exploratory study of small business Internet commerce issues”, Information and Management, Vol. 35, pp. 9-18.

[24] P.T. Joseph, S.J., (2009) “E-Commerce An Indian Perspective”, PHI, pp. 304-503.

[25] Rafiu, Oyesola S. (2007). “The Emergence of Internet Banking in Nigeria: An Appraisal, Information Technology Journal”, Vol. 6, No. 4, 2007, pp. 490-496

[26] Rahmath S., Hema D. and Abdullah K. (2011). “International Arab Journal of e-Technology”, Vol. 2, No. 1, pp.56-64

[27] Rajesh P., (2009), “Indian Banking and Globalization”, Adhyayan Pub., pp. 33-88

[28] Ramboll Management 'ICT and E-Business Impact in the Banking Industry', Version 4.0, 2008.

[29] Ravikumar Jain B., Krishna Kishore P., (2008), "Internet Banking", ICFAI University Press, 69-88

[30] Reginald M., Norman C. and Stanislous Z. (2011), "African Journal of Business Management", Vol. 5, No. 4, pp. 1857-1863