A REPORT ON CAPSTONE PROJECT ENTITLED

SHOP FOR HOME

AN E-COMMERCE WEB APPLICATION BY USING JAVA-GCP

Submitted to

GREAT LEARNING

Submitted by

Ragam Santhoshi

Payal Shrikant Pawar

Juhi Fatheen

Bindu K O

Bairi Rohith Reddy

Under the guidance of

Mr. Parth Shukla

Abstract

In today's generation, most people are using technology for leading their lives and fulfilling their daily needs. In this generation most of us using E-commerce websites for shopping for clothes, groceries, and electronics . We have developed one E- commerce web application by using Full stack java technology as it contains HTML,CSS, Angular, Java Script, SpringBoot, Node.JS platform. This application is fully functional with different views for user and admin and it also has integrated with payment gateway for checkout. By using this website we can buy different types of décor accessories and we can choose different styles of products based upon customer interests. In this project, we can add different products and can delete them also. We have developed administrative functions for the website such as create a product, create categories, Admin dashboard, Manage products, Manage categories. For customers, they can quickly add their items to the cart. Based on the items in the cart then the bill gets generate.

CHAPTER 1

INTRODUCTION

We all know that technology has become an essential tool for online marketing these days. If we see all over the world most of the people are showing interest to buy things in online. However, we can see that there are many small shops and grocery stores are selling their things offline. With this type of selling most of us will face bad experience. For instance, in some shops seller has the product to sell in the offer but the buyer may not know about it, or the customer may need the product urgently then he will go to the shop, but the product is out of stock, in that case, he will face bad experience. Moreover, in online shopping customers can select a wide range of products based upon their interests and their price also, one can compare prices also from one store to another by using online shopping

.

By encountering the all problems and weaknesses of the offline shopping system, creating an Ecommerce web application is necessary for searching and shopping in each shop. These days we have seen so many e-commerce websites are created like Flipkart, Amazon, Myntra one can easily buy their necessary products by using these websites. By using these types of websites one can buy their products by staying in their home. Eventually, we can see the difference between the prices of products also as if we see the cost of the product will be slightly high in offline shopping when compared to online shopping.

For creating these types of E-commerce web applications Java full stack will be the best option that can help us for creating the most effective and powerful web applications.

CHAPTER 2

PROBLEM STATEMENT

Project Name – Shop For Home

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

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 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

Instructions –

1. Please use a folder on server to upload the images

2. Please share the database structure in the .SQL file.

3. Please create a separate microservice for reports and discount coupons.

4. Please use separate port to deploy the Angular UI and Spring Boot Microservice

5. 5. Please use the UI designing tool like (Bootstrap or Material) to make your UI better

6. Please use Material UI to create the UI.

CHAPTER 3

FEATURES AND TECHNOLOGIES USED

3.1 FEATURES

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

This is a Single Page Application with client-side rendering. It includes backend and frontend two separate projects on different branches.

The frontend client makes API calls to the backend server when it is running.

Features

- REST API
- JWT authentication
- Cookie based visitors' shopping cart
- Persistent customers' shopping cart
- Cart & order management
- Checkout
- Catalogue
- Order management

3.2 TECHNOLOGY STACKS

3.2.1 Backend

- Java 11
- Spring Boot 2.2

- Spring Security

- JWT Authentication

- Spring Data JPA
- Hibernate

- PostgreSQL

- Maven

3.2.2 Frontend

- Angular 7

- Angular CLI

- Bootstrap

3.2.3 Database Schema

Install Postgressql -> after installation -> search PGAdmin

Open Admin -> give password which you provide at the time of installation.

Create Database with name "ecommerce". Then run queries provided in eshop.sql script one by one.

3.3 How to Run

Start the backend server before the frontend client.

Backend

1. Install [Eclipse](https://www.Eclipse.org/download/)

2. Configure data source in `application.yml`.

3. `cd backend`.

4. Run `java install`.

5. Run `mvn spring-boot:run`.

6. Spring Boot will import mock data into database by executing `import.sql` automatically.

7. The backend server is running on [localhost:4200]().

Frontend

1. Install [Node.js and npm](https://www.npmjs.com/get-npm) 2. `cd frontend`.

3. Run `npm install`.

4. Run `ng serve`

5. The frontend client is running on [localhost:4200]()

Note: The backend API url is configured in `src/environments/environment.ts` of the frontend project. It is `localhost:8080/api` by default.

1. Build backend project

bash cd backend mvn

package

2. Build fontend project

bash cd frontend npm install

alias

ng="node_modules/@angular

/cli/b in/ng" ng build---prod.

2.5 GOOGLE CLOUD PLATFORM:

Google Cloud Platform (**GCP**), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, Google Drive, and YouTube.

Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.

COMPUTE

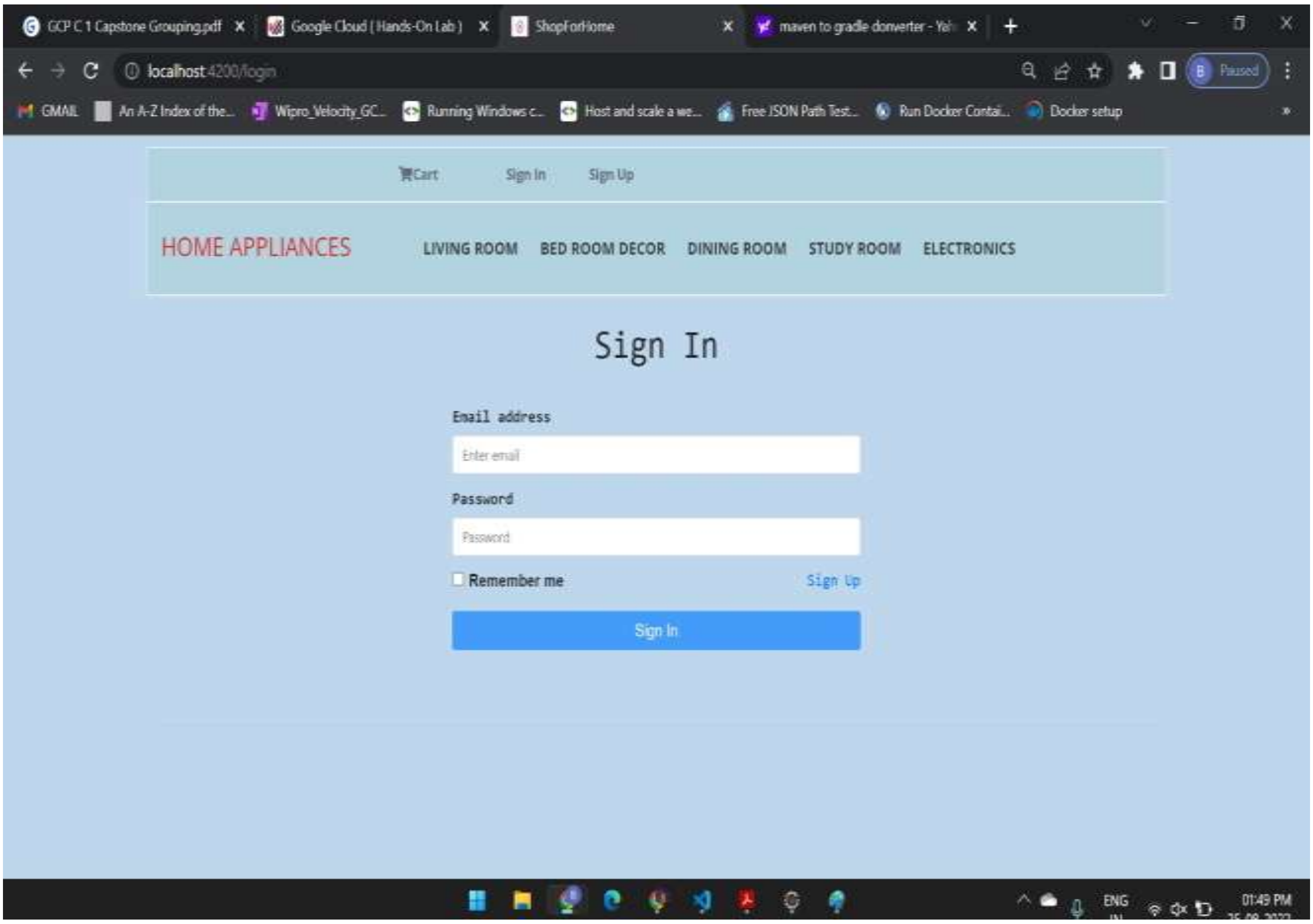
- App Engine – Platform As A Service to deploy Java , PHF, Node.js, Python, C#, .Net, Ruby and Go applications.
- Compute Engiine – Infrastructure as a Service to run Microsoft Windows and Linux virtual machines.
- Google Kubernetes Engine (GKE) or GKE on-prem offered as part of Anthos platform- Containers as a Service based on Kubernetes.

1.Customer login

Default log in details:

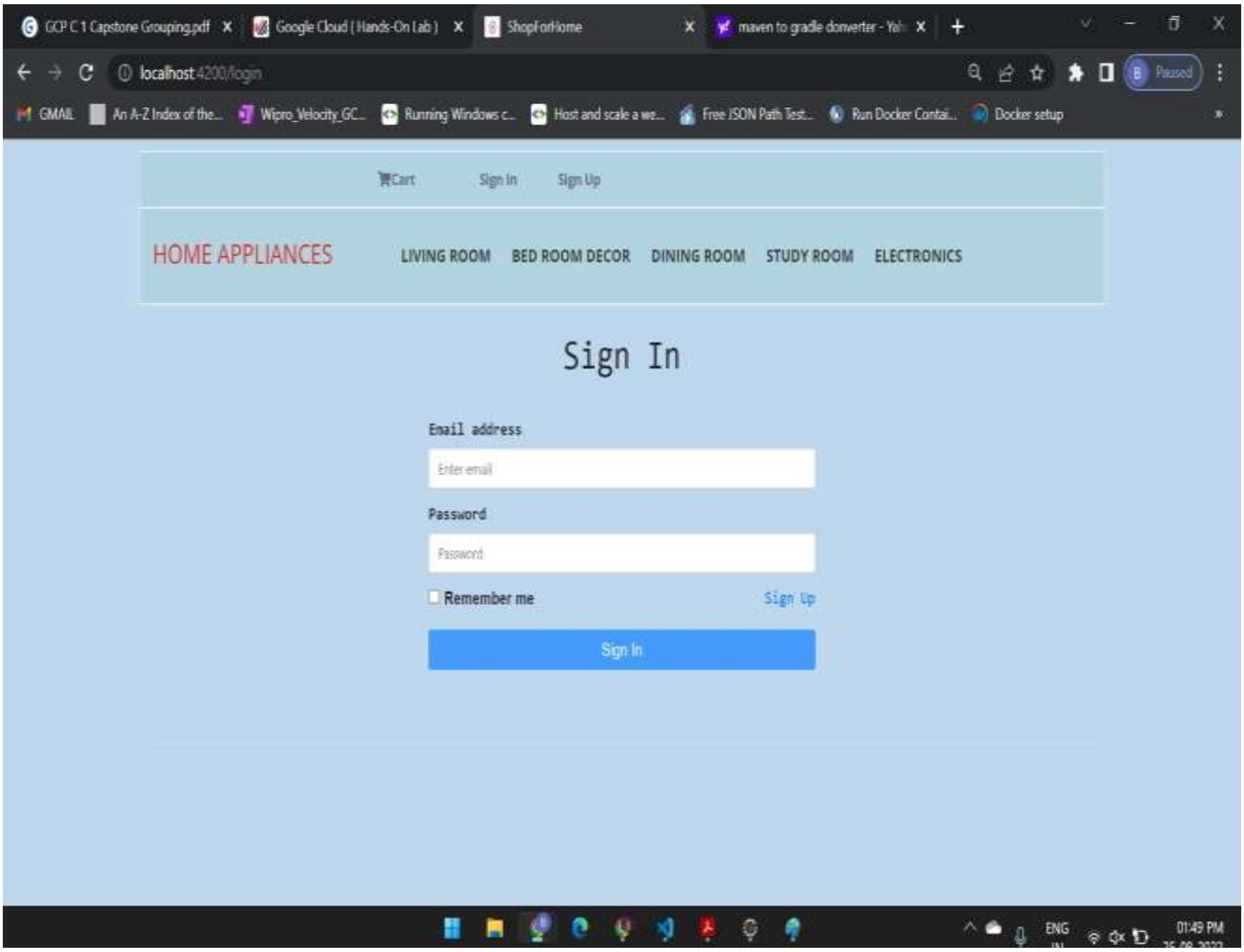
User Name: anilup@gmail.com

Password: anil123



2.Admin login

- Default Admin Login
  User Name :brohithreddy.160301@gmail.com
  Password - apgxxzsajemrpymm



Sign In

Email address

Enter email

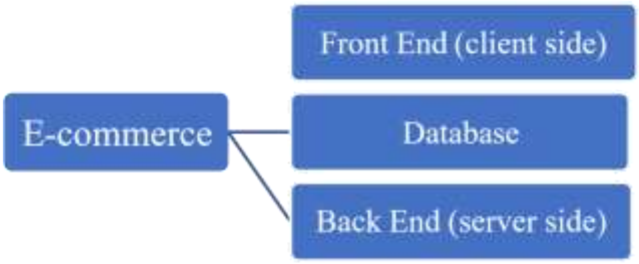
Password

Password

□ Remember me   Sign Up

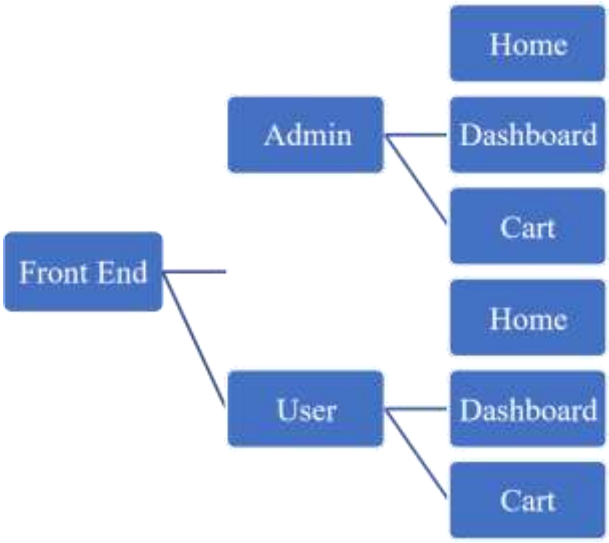
Sign In

CHAPTER-4
WEB APPLICATION STRUCTURE



Creation of an E-commerce application includes Front End(client side) operations and Database for storing the Data and Back End (server side) operations.

4.1 FRONT END (CLIENT SIDE) OPERATIONS :



4.1.1 Home Page:

The home page of the web Application mainly contains a list of the products which are saved in the database. And there some options that will be in the menu bar if the user does not sign in/login in yet then "Sign in" and "Sign up" options will be there. The home page will show you all types of products and they will be displayed to customers, for example, this homepage has types of décor items such as sofa, Table lamp, If we want to add another type of product we can simply add a wide range of products, we can give different prices for different products based upon their quality, customers have the facility to add the product to the cart which they like, the customer will have another facility to

contact the admin if he has any issues regarding products, prices, and any other issues all these things have appeared preview image is mentioned below:

Sign in and Sign up:

These two-option redirect to the page where the user can find a form to fill either to create an account or to sign in to an account.

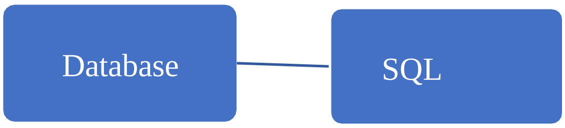
4.1.2 Dashboard:

This page will be different for the user. Admin Dashboard will have a chance to create some categories and can add products to those categories, as well as admin can delete products and he can change prices also.

4.1.3 Cart

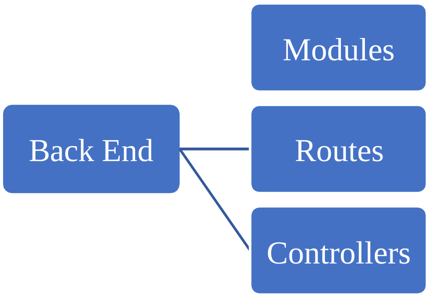
After selecting any product users can see their product on this page and here the payment will be carried on. Stripe Payment is included for the cart which was a third-party tool that helps users to done payment by some debit cards, credit cards, UPI's.

4.2 DATABASE:



Data that is entered by the users will be stored in the database. There are so many databases are been used nowadays. In this project, we have MongoDB as a database. Using the mongoose library, we can connect to SQL. There are so many methods in this library to create schema and also to save the datail the database.

4.3 BACK-END:



Models:

Here we define the structure of the data that should in the database. By using some models which help to store the data in the database like mongoose, it is one of the famous libraries in NodeJS. Creating the schemas by database Schema with Connecting the names and type

Routers:

All the work related to the routing of the pages was done here. Angular ,JS is a popular library form routing. CURD operations and routing-related code are saved in this folder.

Controllers:

In controllers, the definitions of the functions which are declared in the routing will be stored and also the codes of the middleware are stored in this folder.

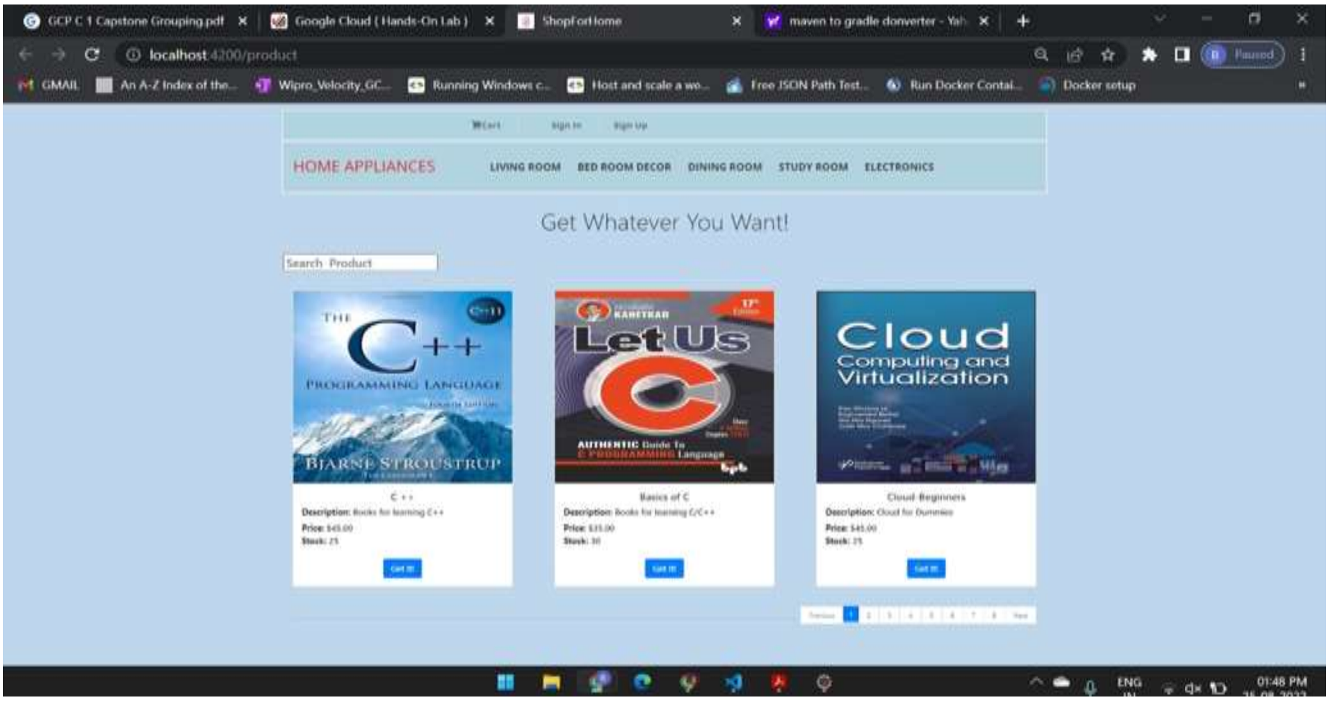
In the controller phase, the function definitions of the function which are declared in the Routers will be done. We are having some middleware also defined here.

5. ONLINE SHOPPING APPLICATION:

Anyone can view Online Shopping portal and available products, but every user must login by his/her Username and password in order to purchase or order products. Unregistered members can register by navigating to registration page. Only Admin will have access to modify roles, by default developer can only be an 'Admin'. Once user register site, his default role will be 'User'.

5.1 HOMEPAGE:

The Home Screen will consist of screen were one can Login Into the application.

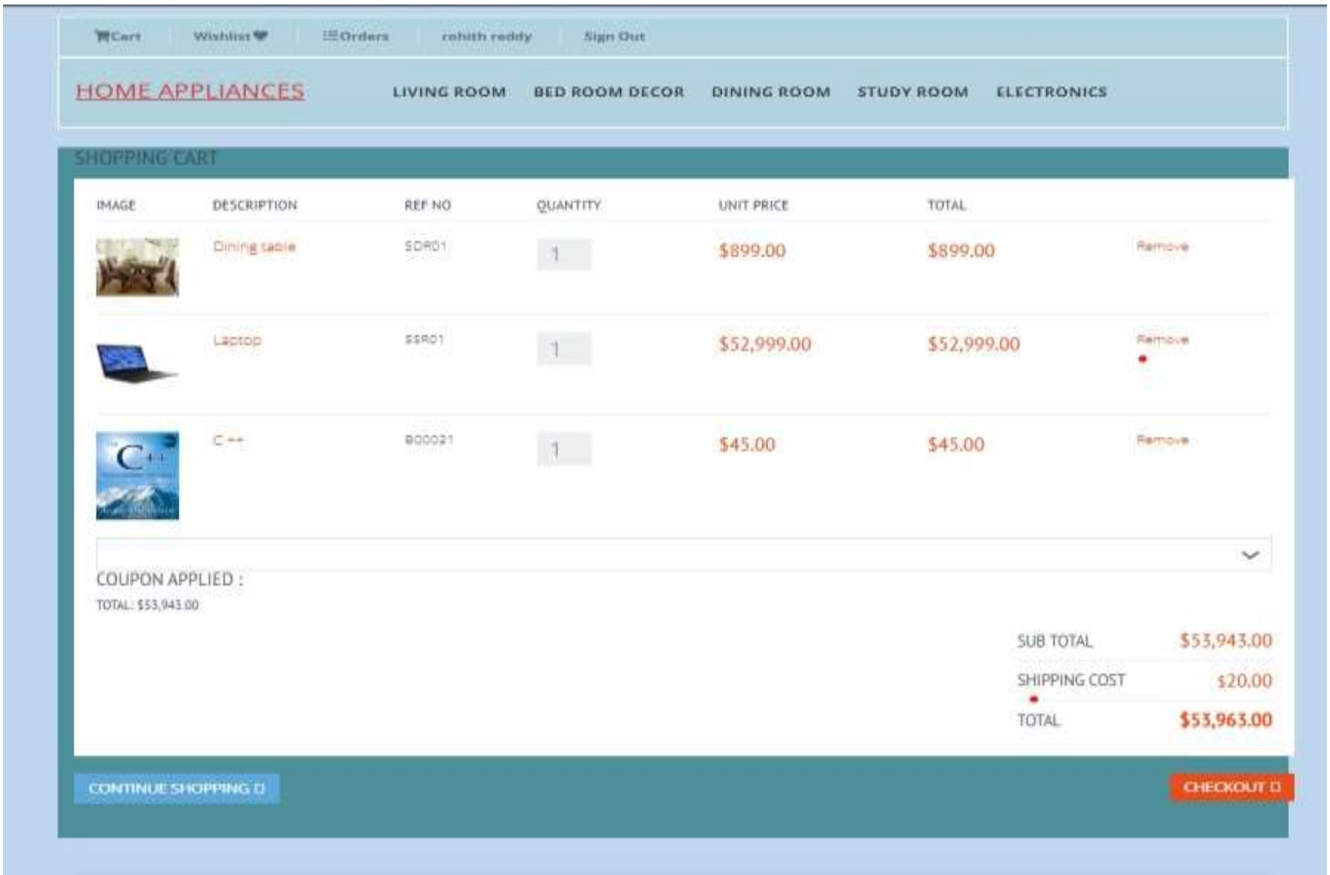


C ++
Description: Books for learning C++
Price: $45.00
Stock: 25

Basics of C
Description: Books for learning C/C++
Price: $35.00
Stock: 30

Cloud Beginners
Description: Cloud for Dummies
Price: $45.00
Stock: 25

5.2 SHOPPING CART : Here visitors can go through the toys available in the store. And even search in the search bar for a specific item.



SHOPPING CART

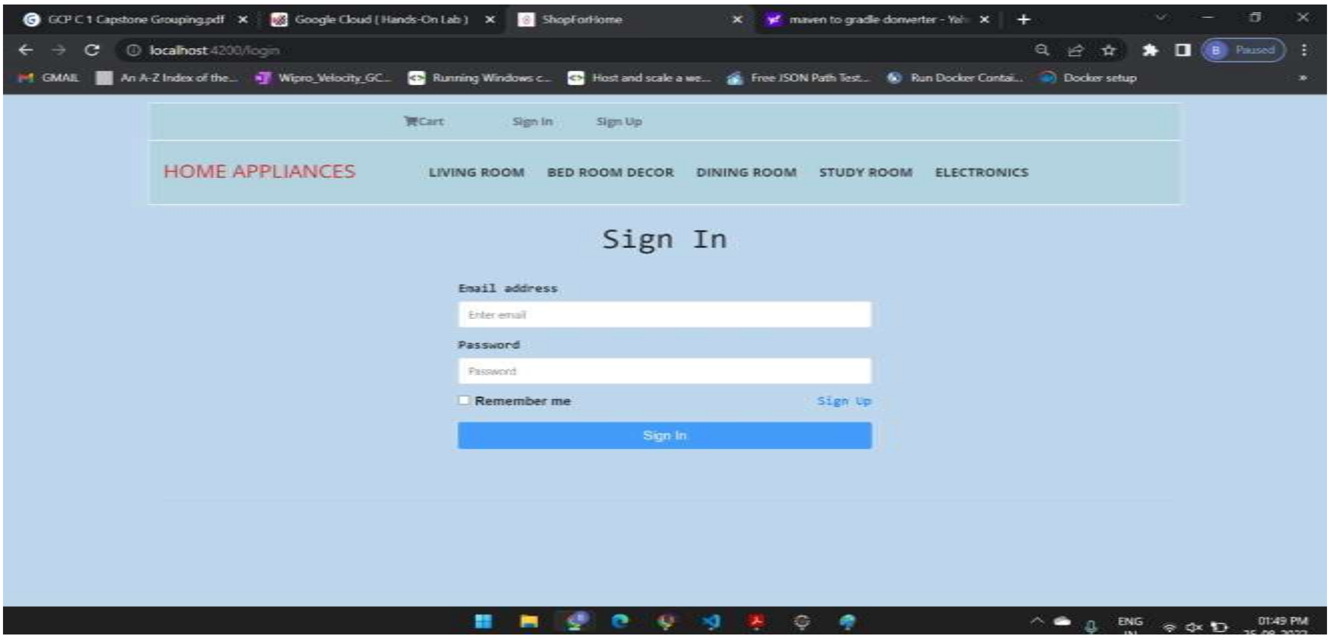
IMAGE	DESCRIPTION	REF NO	QUANTITY	UNIT PRICE	TOTAL	
	Dining table	SDR01	1	$899.00	$899.00	Remove
	Laptop	SSR01	1	$52,999.00	$52,999.00	Remove
	C++	B00021	1	$45.00	$45.00	Remove

COUPON APPLIED :

TOTAL: $53,943.00

SUB TOTAL	$53,943.00
SHIPPING COST	$20.00
TOTAL	$53,963.00

5.3 SIGN-IN PAGE: Here a user can login into his account. If the user have Admin privileges, he will login in to Admin Panel else, he will be login into User Panel.



Sign In

Email address

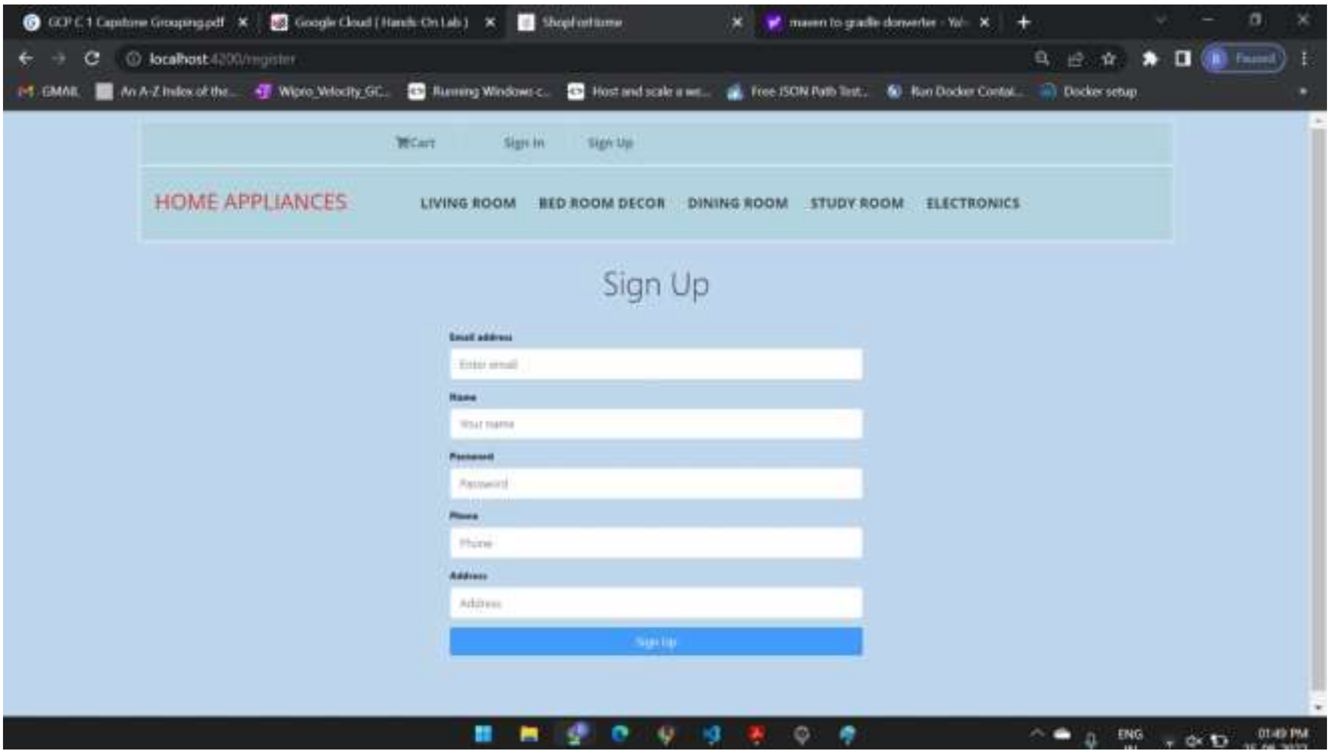
Enter email

Password

Password

□ Remember me Sign Up

5.4 SIGN UP PAGE: If the user does not have a account he can create a account by providing his/her credentials. The account created will be a user account



Sign Up

Email address: Enter email

Name: Your name

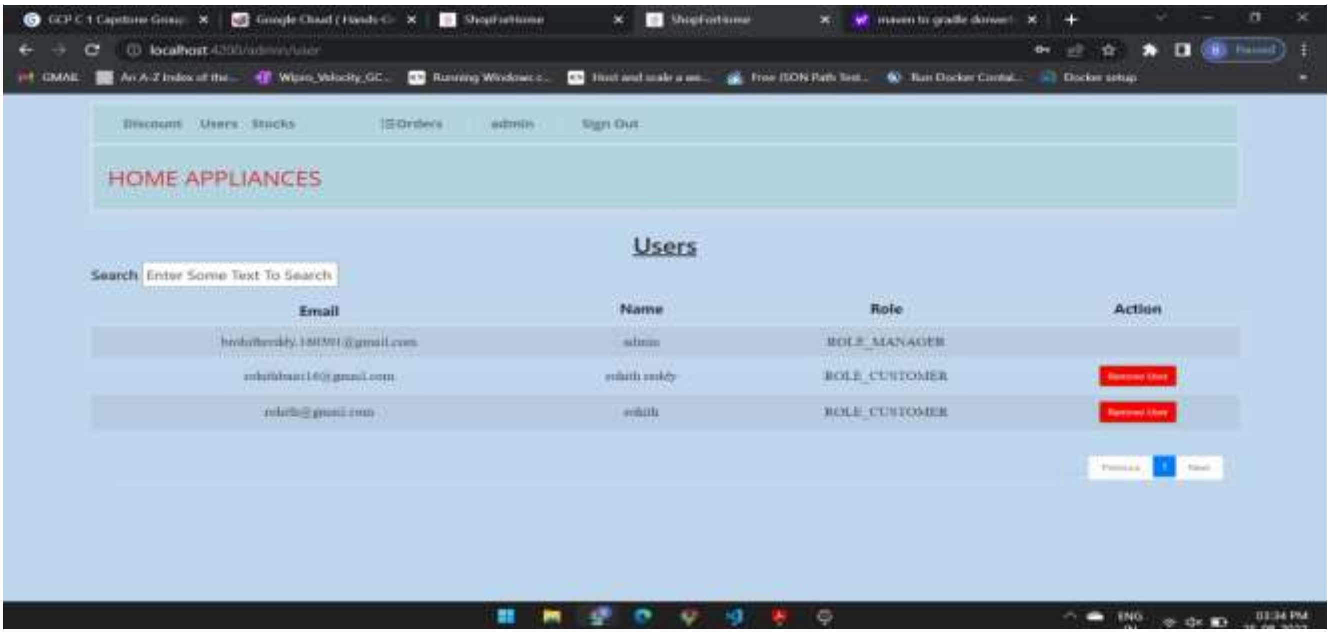
Password: Password

Phone: Phone

Address: Address

Sign Up

5.5 ADMIN VIEW: It has all the additional functionality a admin have.

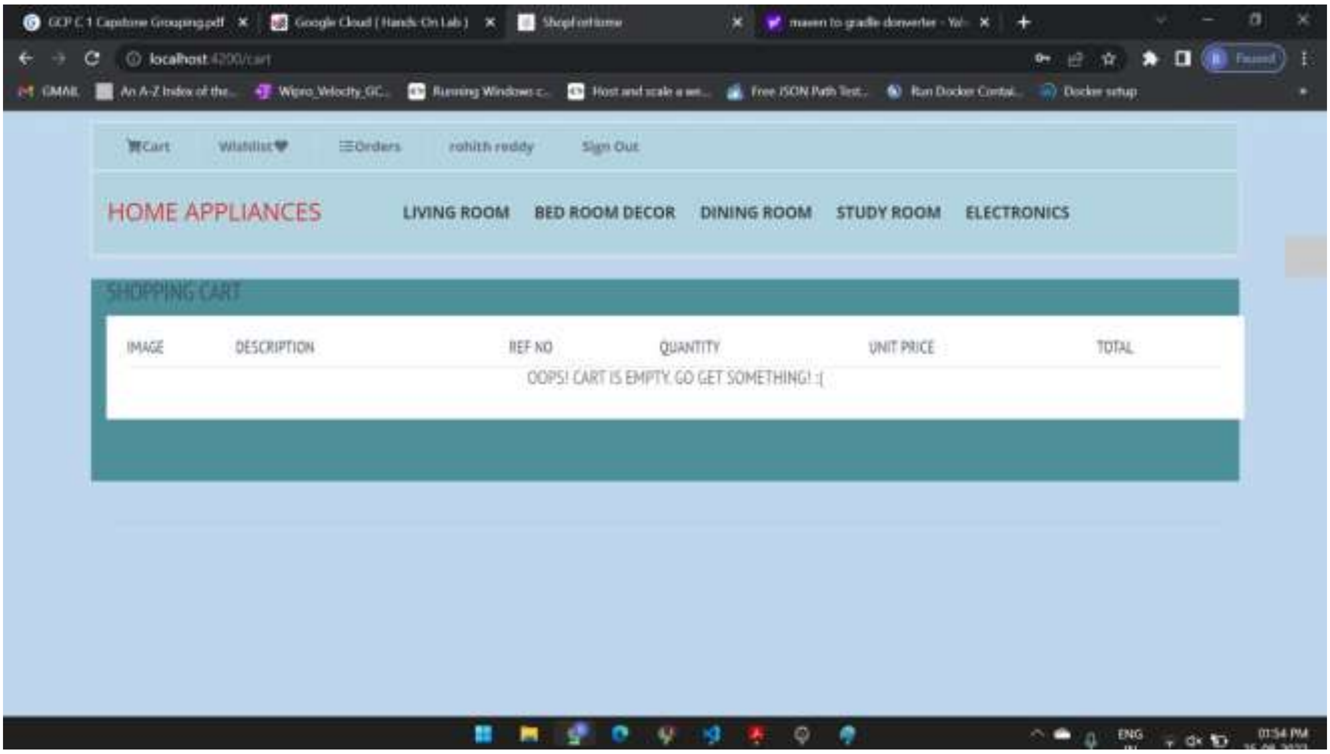


Users

Search: Enter Some Text To Search

Email	Name	Role	Action
[illegible]@gmail.com	admin	ROLE_MANAGER	
[illegible]@gmail.com	rohith reddy	ROLE_CUSTOMER	Remove User
rohith@gmail.com	rohith	ROLE_CUSTOMER	Remove User

5.6 EMPTY CART:



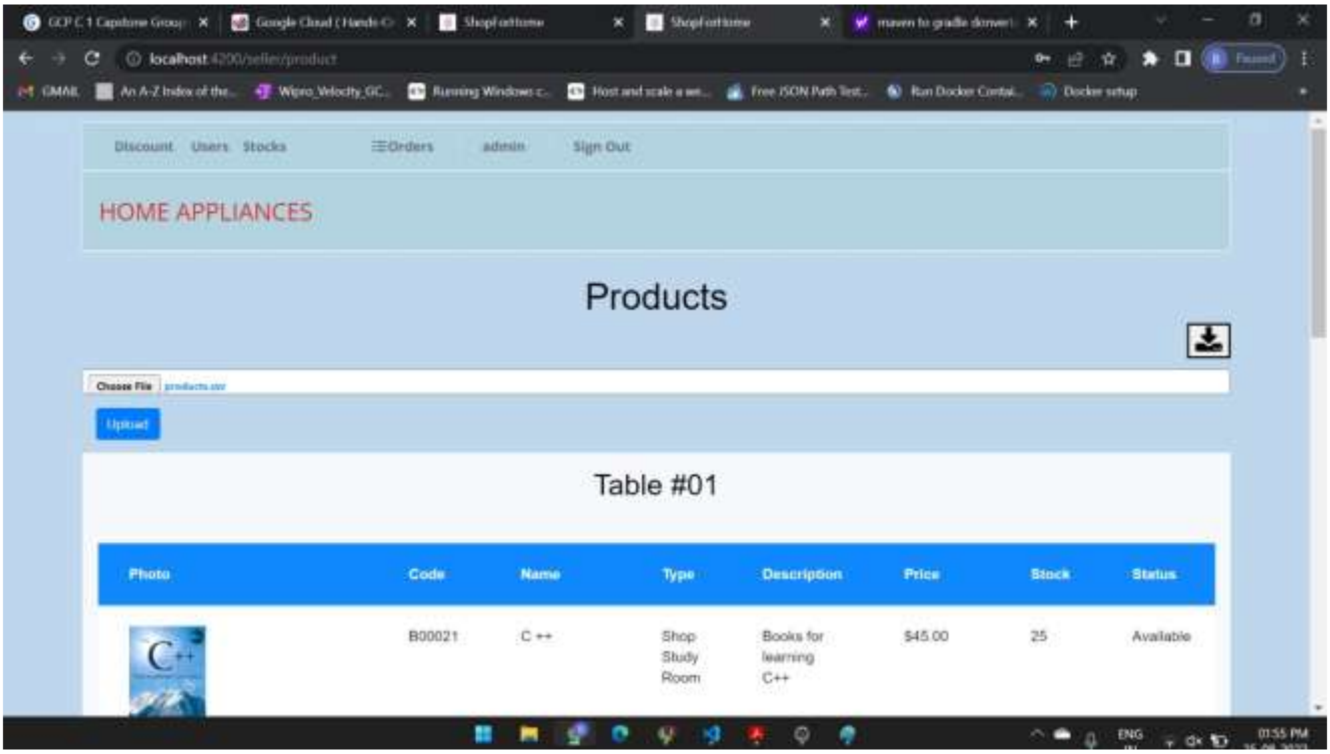
HOME APPLIANCES

LIVING ROOM BED ROOM DECOR DINING ROOM STUDY ROOM ELECTRONICS

SHOPPING CART

IMAGE	DESCRIPTION	REF NO	QUANTITY	UNIT PRICE	TOTAL
OOPS! CART IS EMPTY. GO GET SOMETHING! :(					

5.6 BULK UPLOAD :



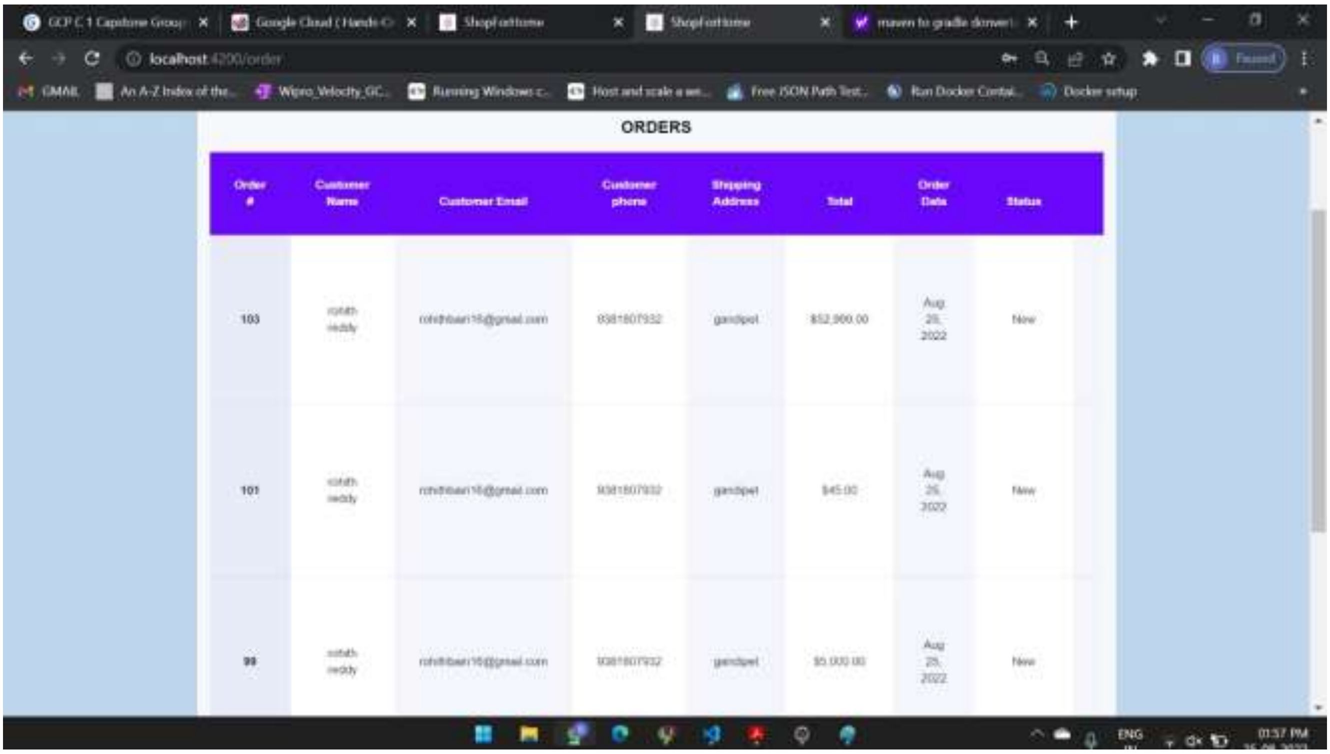
HOME APPLIANCES

Products

Table #01

Photo	Code	Name	Type	Description	Price	Stock	Status
	B00021	C ++	Shop Study Room	Books for learning C++	$45.00	25	Available

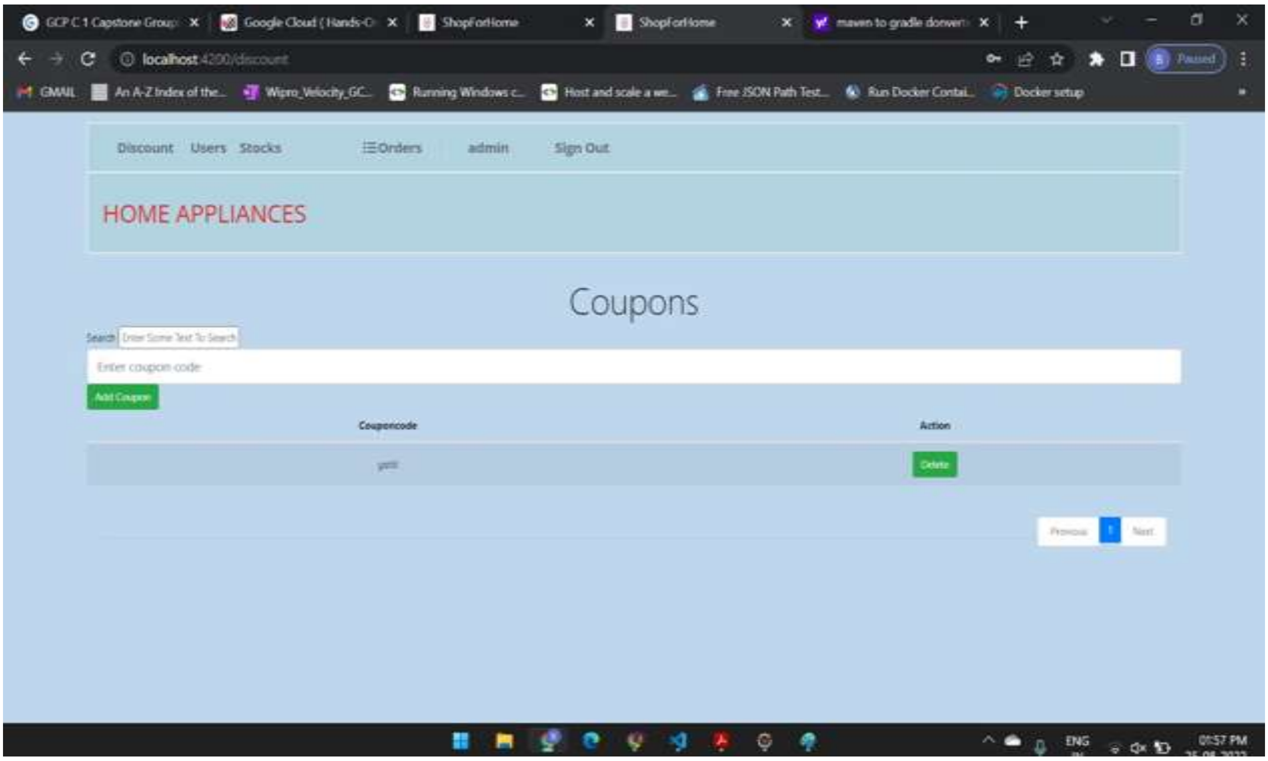
5.7 ORDERS: Here is the orders list that customer have ordered.



ORDERS

Order #	Customer Name	Customer Email	Customer phone	Shipping Address	Total	Order Date	Status
103	rohith reddy	rohithbairi16@gmail.com	9381807932	gandipet	$52,900.00	Aug 25, 2022	New
101	rohith reddy	rohithbairi16@gmail.com	9381807932	gandipet	$45.00	Aug 25, 2022	New
99	rohith reddy	rohithbairi16@gmail.com	9381807932	gandipet	$5,000.00	Aug 25, 2022	New

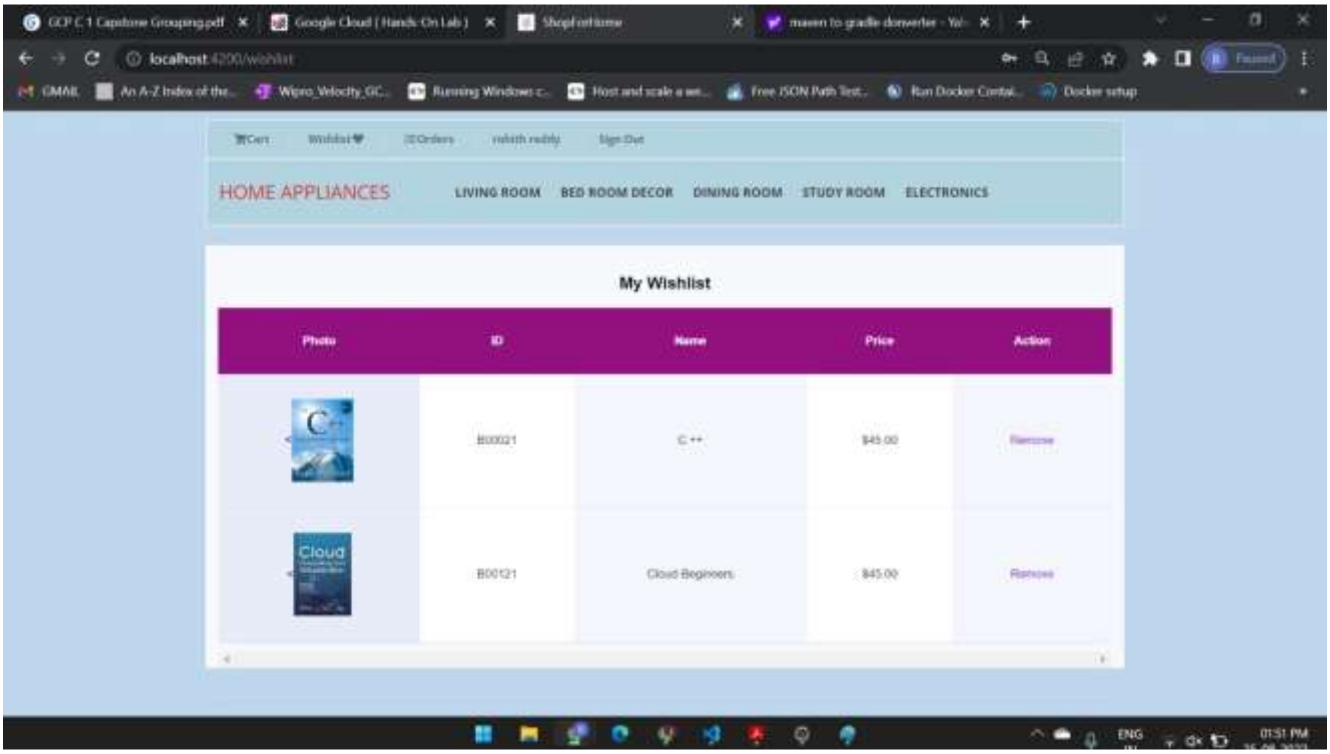
5.8 COUPONS :



Coupons

Couponcode	Action
yetti	Delete

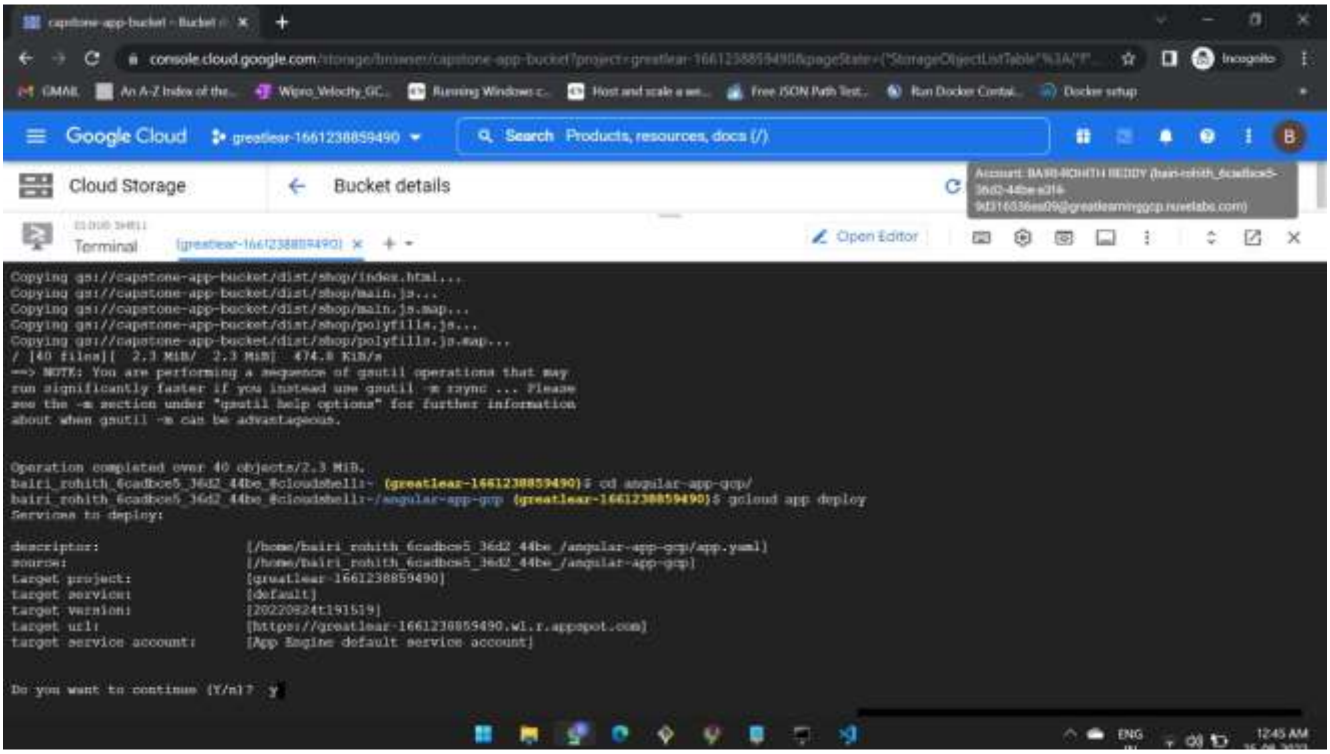
5.9 WISHLIST: Here user can see record of all the items in wish list



My Wishlist

Photo	ID	Name	Price	Action
	B00021	C ++	$45.00	Remove
	B00121	Cloud Beginners	$45.00	Remove

Deploying into GCP:



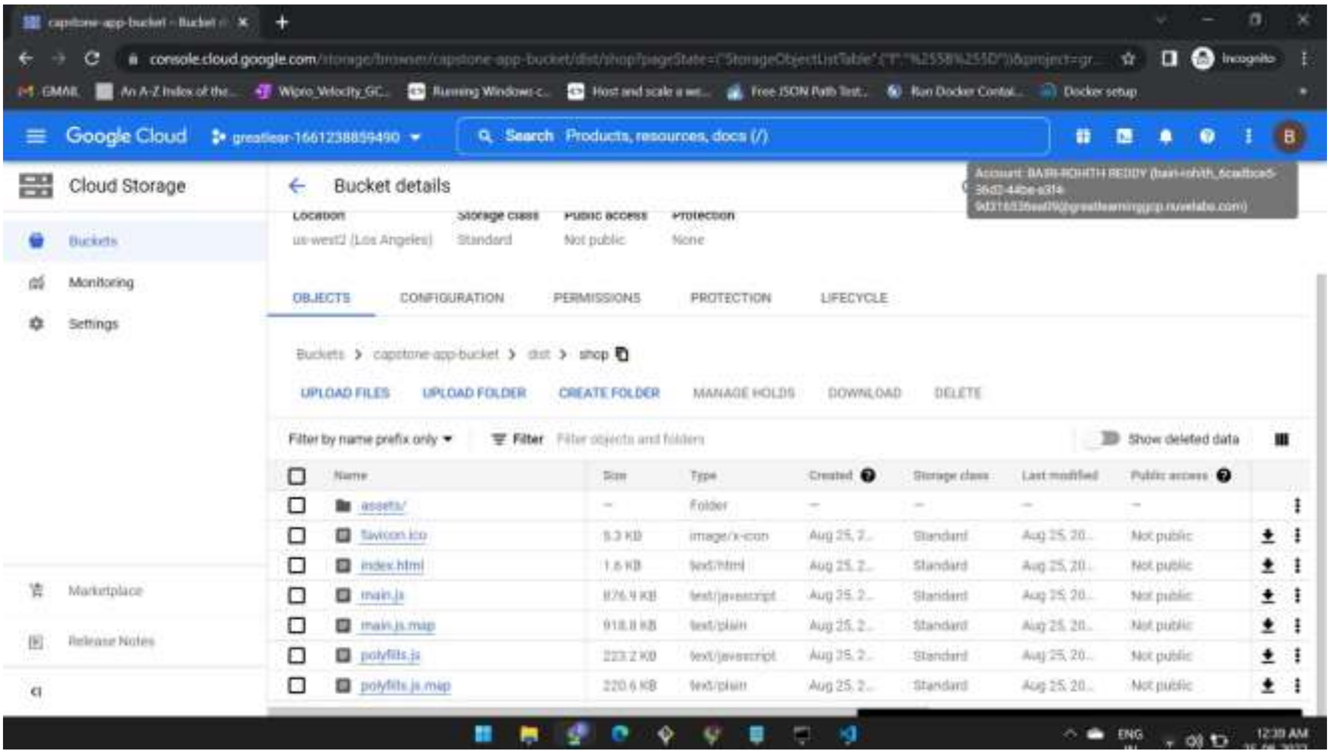
capstone-app-bucket – Bucket d
console.cloud.google.com/storage/browser/capstone-app-bucket?project=greatlear-1661238859490&pageState=("StorageObjectListTable":(%3A("f"...
Google Cloud
greatlear-1661238859490
Search Products, resources, docs (/)
Cloud Storage
Bucket details
Account: BAIRI-ROHITH REDDY (bairi-rohith_6cadbce5-36d2-44be-a314-9d316536ea09@greatlearninggcp.nuvelabs.com)
CLOUD SHELL
Terminal
(greatlear-1661238859490)
Open Editor

```
Copying gs://capstone-app-bucket/dist/shop/index.html...
Copying gs://capstone-app-bucket/dist/shop/main.js...
Copying gs://capstone-app-bucket/dist/shop/main.js.map...
Copying gs://capstone-app-bucket/dist/shop/polyfills.js...
Copying gs://capstone-app-bucket/dist/shop/polyfills.js.map...
/ [40 files][  2.3 MiB/  2.3 MiB]  474.8 KiB/s
==> NOTE: You are performing a sequence of gsutil operations that may
run significantly faster if you instead use gsutil -m rsync ... Please
see the -m section under "gsutil help options" for further information
about when gsutil -m can be advantageous.


Operation completed over 40 objects/2.3 MiB.
bairi_rohith_6cadbce5_36d2_44be_@cloudshell:~ (greatlear-1661238859490)$ cd angular-app-gcp/
bairi_rohith_6cadbce5_36d2_44be_@cloudshell:~/angular-app-gcp (greatlear-1661238859490)$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/bairi_rohith_6cadbce5_36d2_44be_/angular-app-gcp/app.yaml]
source:                      [/home/bairi_rohith_6cadbce5_36d2_44be_/angular-app-gcp]
target project:              [greatlear-1661238859490]
target service:              [default]
target version:              [20220824t191519]
target url:                  [https://greatlear-1661238859490.wl.r.appspot.com]
target service account:      [App Engine default service account]


Do you want to continue (Y/n)?  y
```

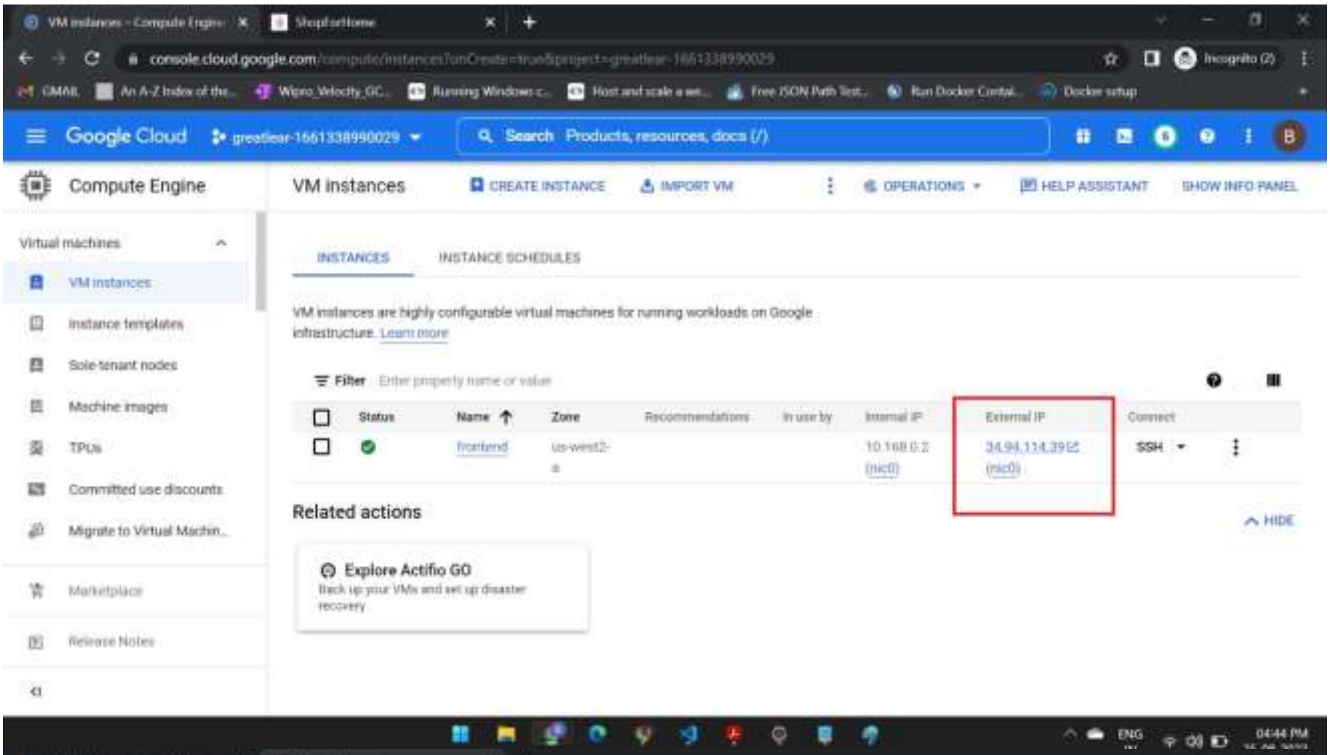


capstone-app-bucket – Bucket d
console.cloud.google.com/storage/browser/capstone-app-bucket/dist/shop;pageState=("StorageObjectListTable":("f":"%255B%255D"))&project=gr...
Google Cloud
greatlear-1661238859490
Search Products, resources, docs (/)
Account: BAIRI-ROHITH REDDY (bairi-rohith_6cadbce5-36d2-44be-a314-9d316536ea09@greatlearninggcp.nuvelabs.com)
Cloud Storage
Buckets
Monitoring
Settings
Marketplace
Release Notes
Bucket details

Location	Storage class	Public access	Protection
us-west2 (Los Angeles)	Standard	Not public	None

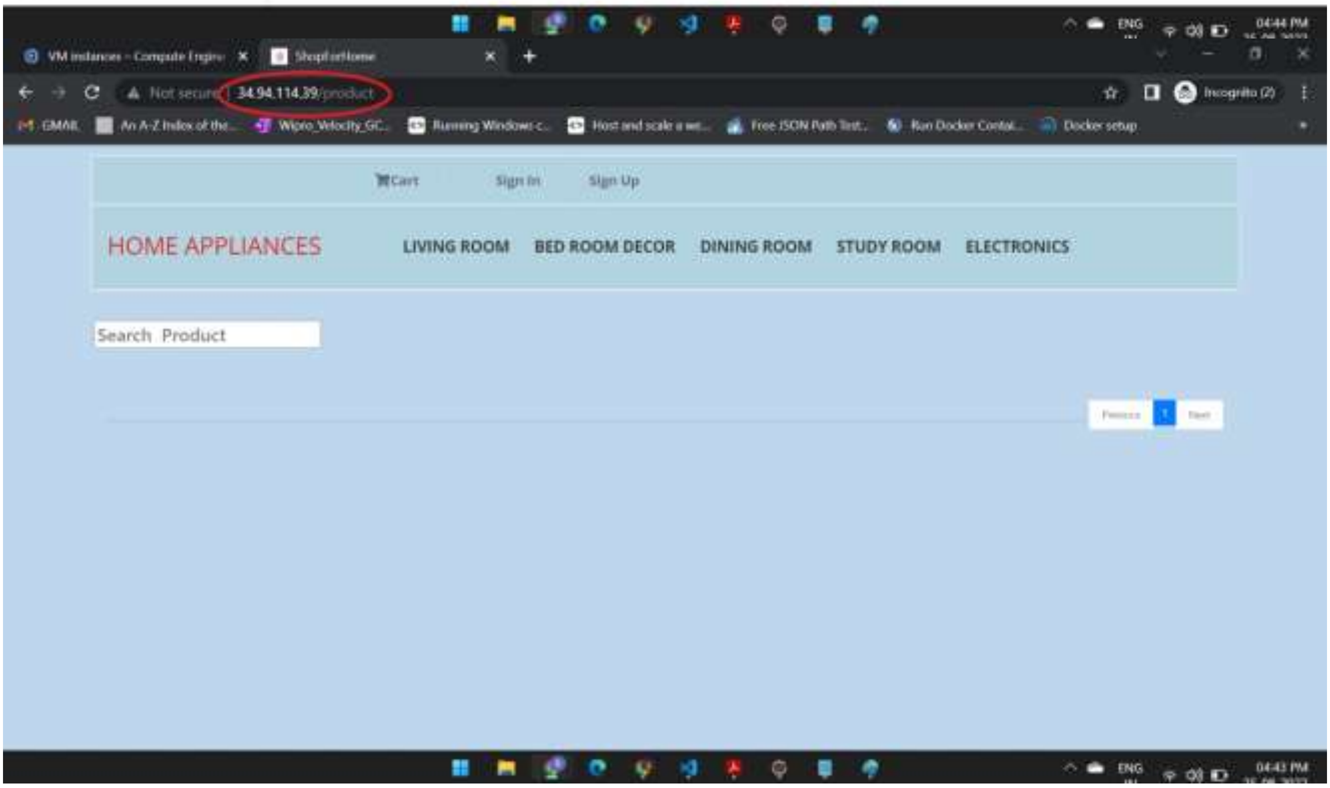
OBJECTS
CONFIGURATION
PERMISSIONS
PROTECTION
LIFECYCLE
Buckets > capstone-app-bucket > dist > shop
UPLOAD FILES
UPLOAD FOLDER
CREATE FOLDER
MANAGE HOLDS
DOWNLOAD
DELETE
Filter by name prefix only
Filter
Filter objects and folders
Show deleted data

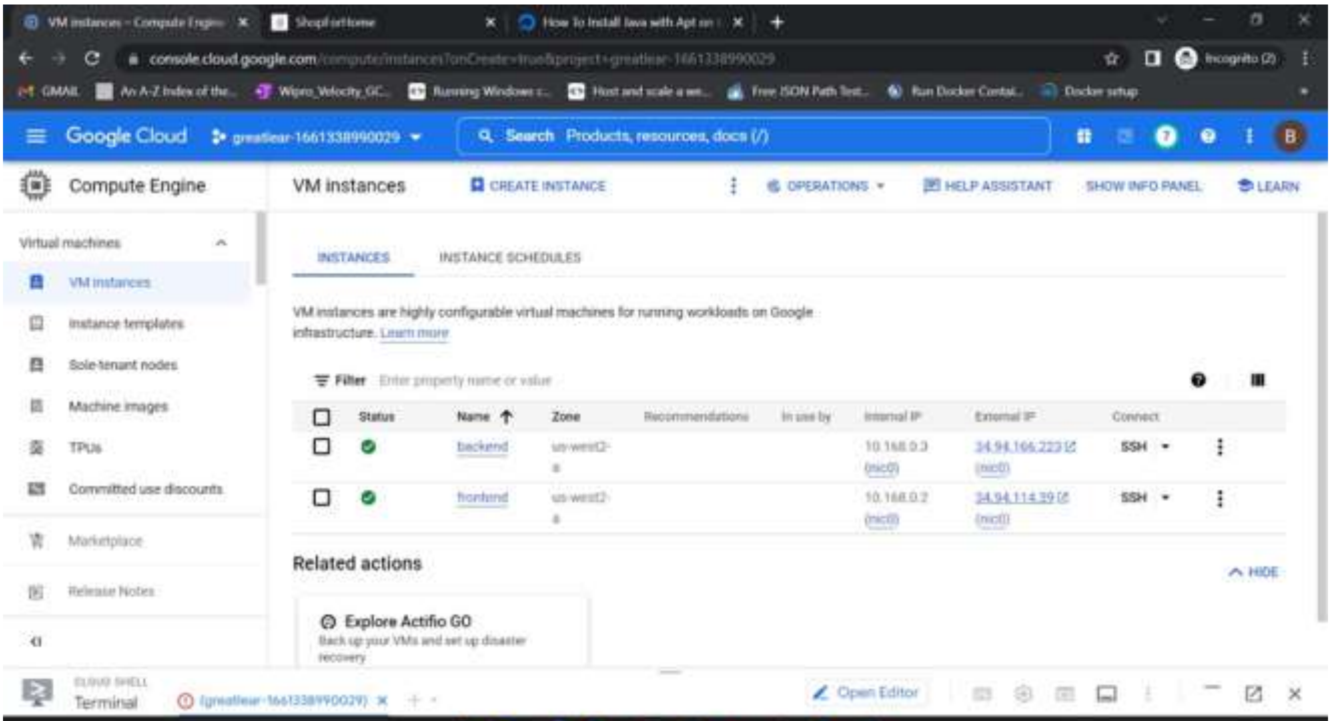
Name	Size	Type	Created	Storage class	Last modified	Public access
assets/	–	Folder	–	–	–	–
favicon.ico	5.3 KB	image/x-icon	Aug 25, 2...	Standard	Aug 25, 20...	Not public
index.html	1.6 KB	text/html	Aug 25, 2...	Standard	Aug 25, 20...	Not public
main.js	876.9 KB	text/javascript	Aug 25, 2...	Standard	Aug 25, 20...	Not public
main.js.map	918.8 KB	text/plain	Aug 25, 2...	Standard	Aug 25, 20...	Not public
polyfills.js	223.2 KB	text/javascript	Aug 25, 2...	Standard	Aug 25, 20...	Not public
polyfills.js.map	220.6 KB	text/plain	Aug 25, 2...	Standard	Aug 25, 20...	Not public



VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

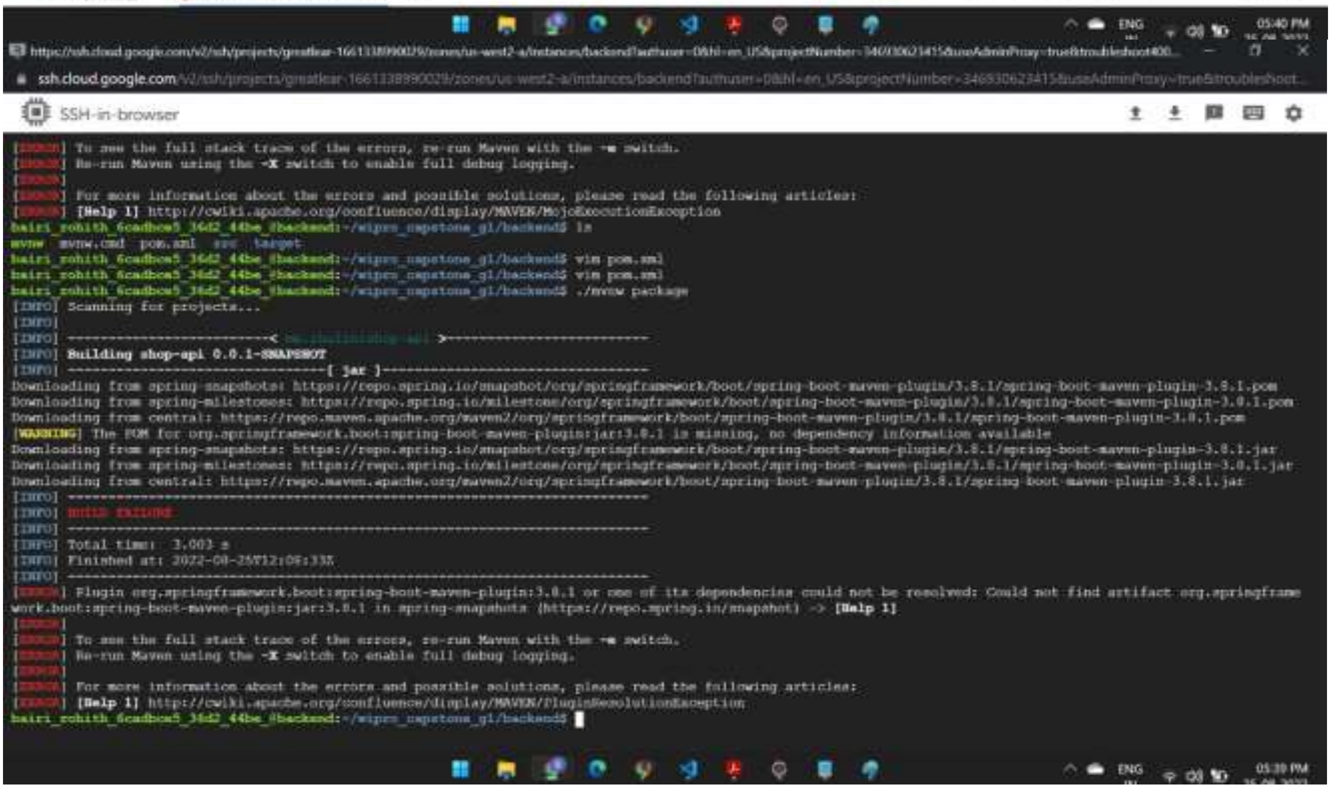
□	Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	frontend	us-west2-a			10.168.0.2 (nic0)	34.94.114.39 (nic0)	SSH





VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	backend	us-west2-a			10.168.0.3 (nic0)	34.94.166.223 (nic0)	SSH
□	✓	frontend	us-west2-a			10.168.0.2 (nic0)	34.94.114.39 (nic0)	SSH



https://ssh.cloud.google.com/v2/ssh/projects/greatlear-1661338990029/zones/us-west2-a/instances/backend?authuser=0&hl=en_US&projectNumber=346930623415&useAdminProxy=true&troubleshoot400...
ssh.cloud.google.com/v2/ssh/projects/greatlear-1661338990029/zones/us-west2-a/instances/backend?authuser=0&hl=en_US&projectNumber=346930623415&useAdminProxy=true&troubleshoot...
SSH-in-browser
ENG
05:39 PM

```
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionException
bairi_rohith_6cadbce5_36d2_44be_@backend:~/wipro_capstone_g1/backend$ ls
mvnw  mvnw.cmd  pom.xml  src  target
bairi_rohith_6cadbce5_36d2_44be_@backend:~/wipro_capstone_g1/backend$ vim pom.xml
bairi_rohith_6cadbce5_36d2_44be_@backend:~/wipro_capstone_g1/backend$ vim pom.xml
bairi_rohith_6cadbce5_36d2_44be_@backend:~/wipro_capstone_g1/backend$ ./mvnw package
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------------< [illegible] >---------------------------
[INFO] Building shop-api 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
Downloading from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-maven-plugin/3.0.1/spring-boot-maven-plugin-3.0.1.pom
Downloading from spring-milestones: https://repo.spring.io/milestone/org/springframework/boot/spring-boot-maven-plugin/3.0.1/spring-boot-maven-plugin-3.0.1.pom
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-maven-plugin/3.0.1/spring-boot-maven-plugin-3.0.1.pom
[WARNING] The POM for org.springframework.boot:spring-boot-maven-plugin:jar:3.0.1 is missing, no dependency information available
Downloading from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-maven-plugin/3.0.1/spring-boot-maven-plugin-3.0.1.jar
Downloading from spring-milestones: https://repo.spring.io/milestone/org/springframework/boot/spring-boot-maven-plugin/3.0.1/spring-boot-maven-plugin-3.0.1.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/springframework/boot/spring-boot-maven-plugin/3.0.1/spring-boot-maven-plugin-3.0.1.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.003 s
[INFO] Finished at: 2022-08-25T12:05:33Z
[INFO] ------------------------------------------------------------------------
[ERROR] Plugin org.springframework.boot:spring-boot-maven-plugin:3.0.1 or one of its dependencies could not be resolved: Could not find artifact org.springframe
work.boot:spring-boot-maven-plugin:jar:3.0.1 in spring-snapshots (https://repo.spring.io/snapshot) -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/PluginResolutionException
bairi_rohith_6cadbce5_36d2_44be_@backend:~/wipro_capstone_g1/backend$ 
```