



A project report on
SHOP FOR HOME

*Submitted in
partial fulfilment of the requirements for the internship of*

WIPRO JAVA-AWS

by

SRAVAN ARIMADLA
CHANDANA SRI KANDI
KORADA KAVYA SRI
KUMARI KAJAL SINGH
HRISHIKESH KIRAN DHAYAFULE

under the guidance of
NEERAJ KUMAR KHERIA
Great Learning
August 2022

ABSTRACT

E-Commerce is a process of doing business through computer networks. A person sitting on his chair in front of a computer can access all the facilities of the Internet to buy or sell the products. Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. Also in today's fast-changing business environment, it's extremely important to be able to respond to client needs in the most effective and timely manner. If your customers wish to see your business online and have instant access to your products or services. Here Online Shopping is a shop for home decor store, just like e-commerce web application. This project allows viewing various products available, enables registered users to purchase desired product. In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, frontend and backend implementation, technologies such as Angular, Spring Tool Suite, programming language (such as Java) and relational databases (sql). This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to learn about each technologies used for creating and implementing an e-commerce website.

ACKNOWLEDGEMENT

We are profoundly grateful to NEERAJ KUMAR KHERIA for his expert guidance and encouragement throughout the project. We would like to extend our sincere thanks to our institution for providing us with this opportunity. We are highly indebted to our guide for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project. The work is a team effort minus which the completion of this project was not possible.

SRAVAN ARIMDALA

CHANDANA SRI KANDI

KORADA KAVYA SRI

KUMARI KAJAL SINGH

HRISHIKESH KIRAN DHAYAFULE

TABLE OF CONTENTS

| | |
|--------------------------------------|-----|
| Abstract | ii |
| Acknowledgement | iii |
| Table of Contents | iv |
| CHAPTER 1: Introduction | 1 |
| CHAPTER2: System Requirement | 4 |
| 2.1 Angular | |
| 2.2 Spring Boot | |
| 2.3 SQL | |
| 2.4 Amazon Web Services | |
| 2.5 GitHub | |
| CHAPTER 3: Shop For Home Application | 10 |
| 3.1 Home Page | |
| 3.2 Registration Page | |
| 3.3 Login Page | |
| 3.4 Product Categories | |
| 3.5 Cart View | |
| 3.6 Discount | |
| 3.7 Wishlist | |
| 3.8 Admin Dashboard | |
| CHAPTER 4: Code Snapshots | 24 |
| 4.1 Angular | |
| 4.2 SQL | |
| 4.3 Spring Tool Suite | |
| 4.4 AWS | |
| 4.5 Git Repository | |
| CHAPTER 5: Conclusion | 34 |

CHAPTER 1: Introduction

From startups to small businesses right through to huge brands, there are a huge number of companies that can benefit from their own ecommerce website, where they can sell their own products or services. In todays competitive and convenience focused society, no longer do consumers want to venture to the high street in order to buy items, instead consumers want to shop from their own homes, making ecommerce a flexible solution for both businesses and buyers.. Consumers can get full information about the product with its reviews being passed by the existing users. If one wants to buy a product he/she is no longer limited to asking the friends and families because there are many products reviews on the web which gives opinions of the existing users of the product. Online shopping sites contain wide variety of goods both high quality and mild quality keeping in mind the level of people.

The goal of this application is to develop a web based interface for online retailers. The system would be easy to use and hence make the shopping experience pleasant for the users.

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.

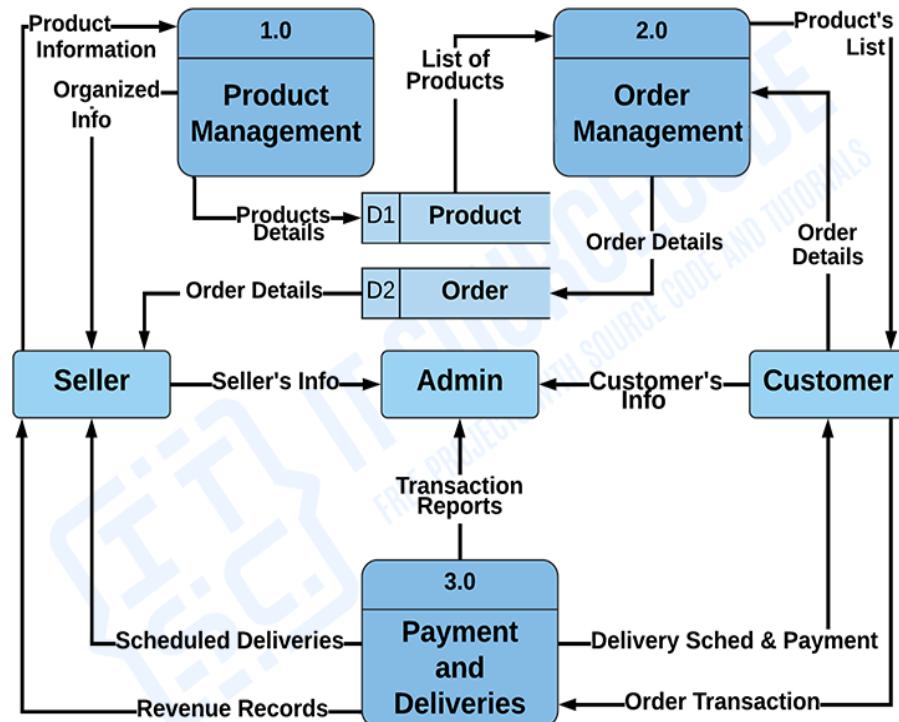
Admin Stories:

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

DATA FLOW DIAGRAM:

From startups to small businesses right through to huge brands, there are a huge number of companies that can benefit from their own ecommerce website, where they can sell their own products or services. In today's competitive and convenience focused society, no longer do consumers want to venture to the high street in order to buy items, instead consumers want to shop from their own homes, making ecommerce a flexible solution for both businesses and buyers.. Consumers can get full information about the product with its reviews being passed by the existing users. If one wants to buy a product he/she is no longer limited to asking the friends and families because there are many products reviews on the web which gives opinions of the existing users of the product. Online shopping sites contain wide variety of goods both high quality and mild quality keeping in mind the level of people.

The goal of this application is to develop a web based interface for online retailers. The system would be easy to use and hence make the shopping experience pleasant for the users.



CHAPTER 2 : SYSTEM REQUIREMENTS

- Frontend : Angular(Visual Studio Code)
- Backend : Spring Tool Suite
- Database : SQL Workbench
- Deployment : Amazon Web Services
- Creating repository : GitHub

2.1 ANGULAR

Angular (commonly referred to as "Angular 2+" or "Angular CLI") is a TypeScriptbased free and open-source web application framework led by the Angular Team at

Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS. Angular. Developer(s).

2.1.1 Applications of Angular :

As mentioned before, Angular 7 is one of the most widely used language over the web. I'm going to list few of them here:

Google Supported Community - Google actively supports Angular and its development. Angular is used in various Google Apps.

POJO based development - Angular heavily used Plain Old JavaScript Object and it helps in learning Angular in an easier way.

Declarative User Interface - Angular uses HTML as view language and extends its functionality. It helps in handling UI vs code differentiation and UI is loosely coupled with code.

TypeScript - TypeScript is super set of javascript and is easy to debug. It is highly secure and is object oriented.

Modular Structure - Angular development is highly modular, is component based and is highly maintainable.

Multi-platform support - Angular code works well on all platforms without much change in code.

2.2 SPRING TOOL SUITE

STS is an is a java IDE tailored for developing Spring-based enterprise applications. It is a free, open-source, and powered by VMWare. Spring Tools 4 is the next generation of Spring tooling for the favorite coding environment. Largely rebuilt from scratch.

It is an Eclipse-based development environment that is customized for the development of Spring applications. It provides a ready-to-use environment to implement, debug, run and deploy your applications. It also includes integration for Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven and AspectJ. STS is built as an addition on top of the latest Eclipse releases.

2.2.1 Advantages :

- Easy to understand and develop spring applications
- Increases productivity
- Reduces the development time

2.3 SQL

Mysql workbench is the visual and graphic tool that can be used for the development of the database, modeling, configuring, designing, creating, and maintaining the database including backup and restoration using interactive graphics.. For example, you can define your own data types, build out custom functions, even write code from different programming languages without recompiling your database.

2.3.1 Advantages:

- The user interface is very easy to use, user-friendly, and intuitive.
- It is licensed under GPL(General Public License) and is free for downloading.
- Workbench is cross-platform supportive and can be used on different operating systems including Windows, Linux, and mac os.
- Schema and query formation and optimization can be done using graphical visualization tools.
- It contains all the features that are updated now and then with changes in the MySQL database server to make itself completely supportive to the MySQL database server.

EveSQL has developed a plugin that can be used with a workbench that allows the optimization of the queries inside the IDE itself.

2.4 AMAZON WEB SERVICES

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers

and servers, you can access technology services, such as computing power, storage, and

databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

2.4.1 Benefits of cloud computing

- Agility

The cloud gives you easy access to a broad range of technologies so that you can innovate faster and build nearly anything that you can imagine. You can quickly spin up resources as you need them—from infrastructure services, such as compute, storage, and databases, to Internet of Things, machine learning, data lakes and analytics, and much more.

- Elasticity

With cloud computing, you don't have to over-provision resources up front to handle peak levels of business activity in the future. Instead, you provision the amount of resources that you actually need. You can scale these resources up or down to instantly grow and shrink capacity as your business needs change.

2.4.2 Why AWS & Benefits:

- Easy to use

AWS is designed to allow application providers, ISVs, and vendors to quickly and securely host your applications – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform.

- Scalable and high-performance

Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

- Secure.

AWS utilizes an end-to-end approach to secure and harden our infrastructure, including physical, operational, and software measures. For more information, see the AWS Security Center and many more.

2.5 GitHub

Git is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers.

GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans.

Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

It Makes It Easy to Contribute to Your Open Source Projects To be honest, nearly every open-source project uses GitHub to manage their project. Using GitHub is free if your project is open-source, and GitHub includes a wiki and issue tracker that makes it easy to include more in-depth documentation and get feedback about your project. If you want to contribute, you just fork a project, make your changes and then send them a pull request using GitHub web interface.

CHAPTER 3: ShopForHome Application

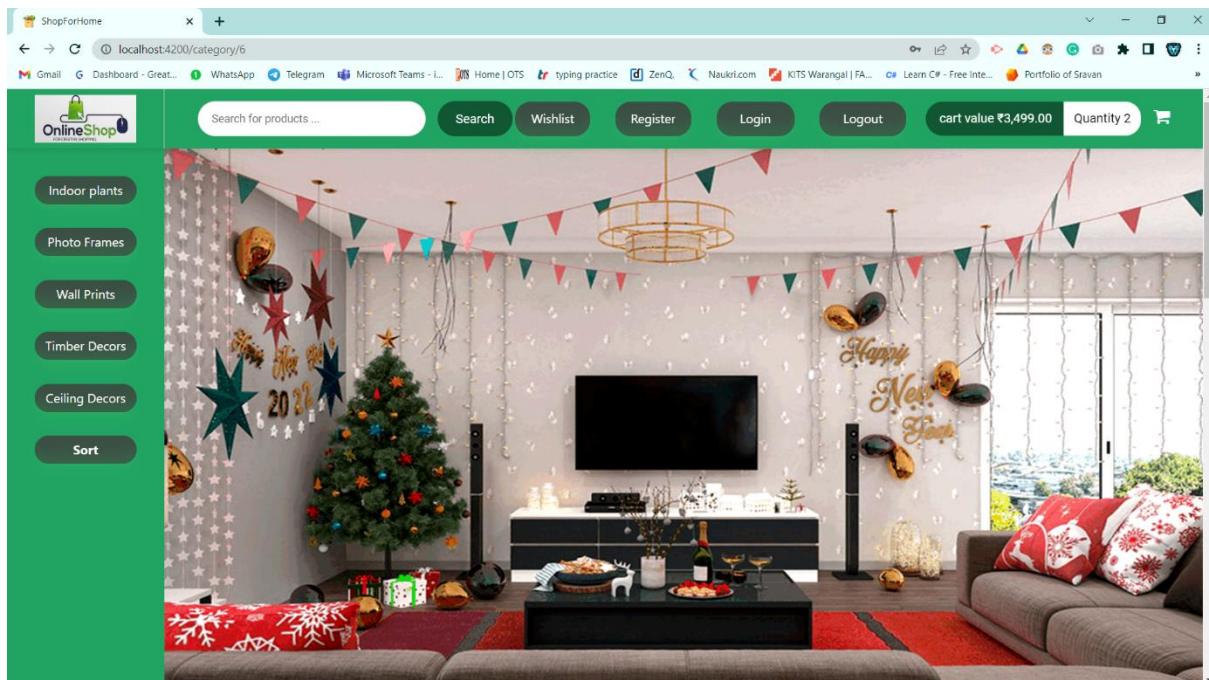
3.1 Home Page :

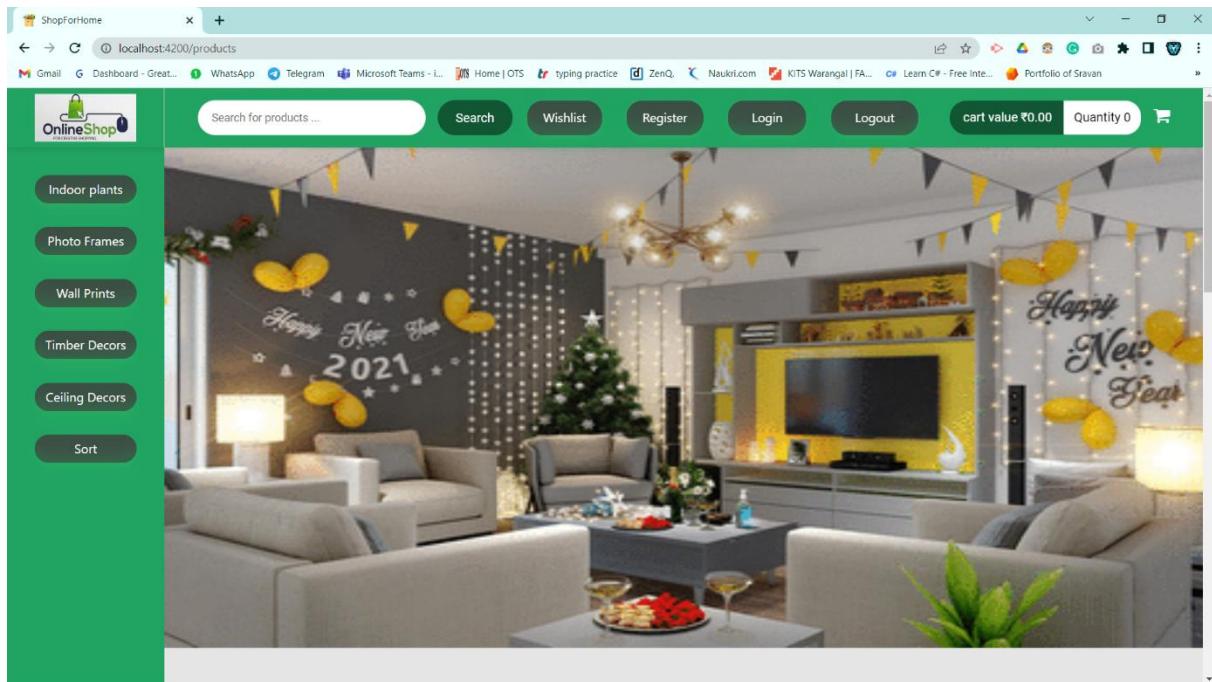
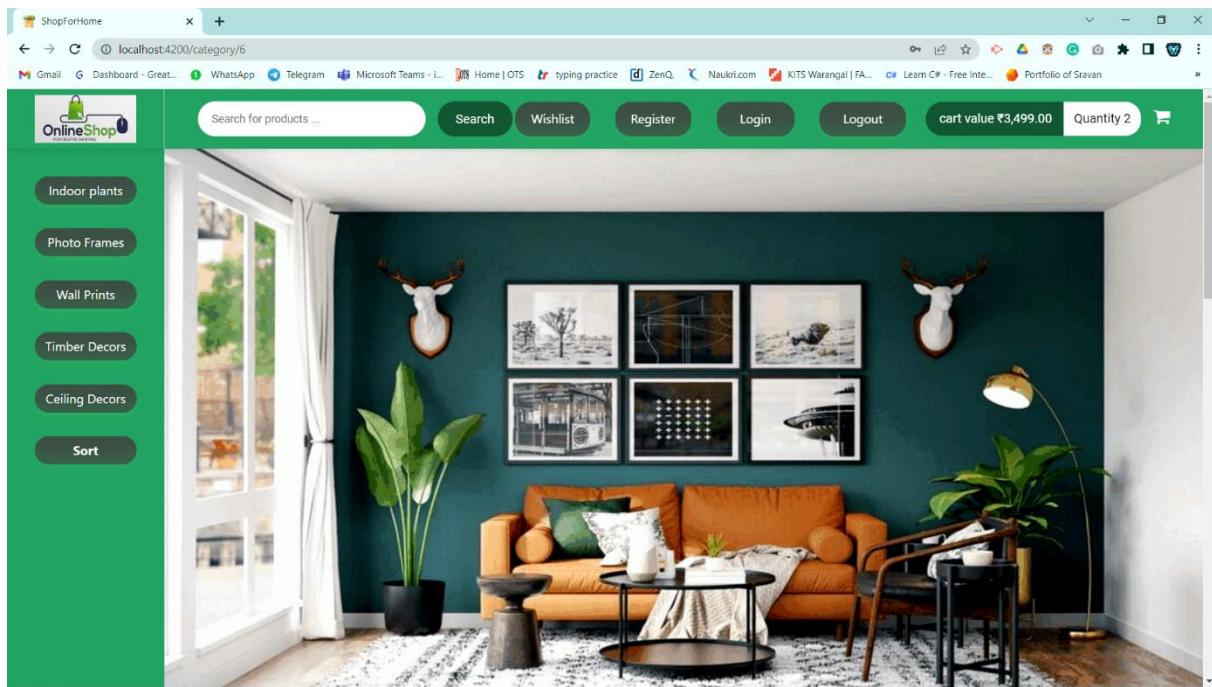
The main function of a homepage is to direct users around the site, so it is important that guests can do so with ease. Clearly marks the difference between the choices, and have a good sense of what lies beneath the links.

Website design is one of the most powerful tools for the business to make a good first impression. But in order to do that, it needs a great web design.

While a great website involves much more than just the design, customers want and expect an attractive visual layout. And they'll judge the business on looks before they read a single word.

So we have the following Designed Home Page which attracts the customer.



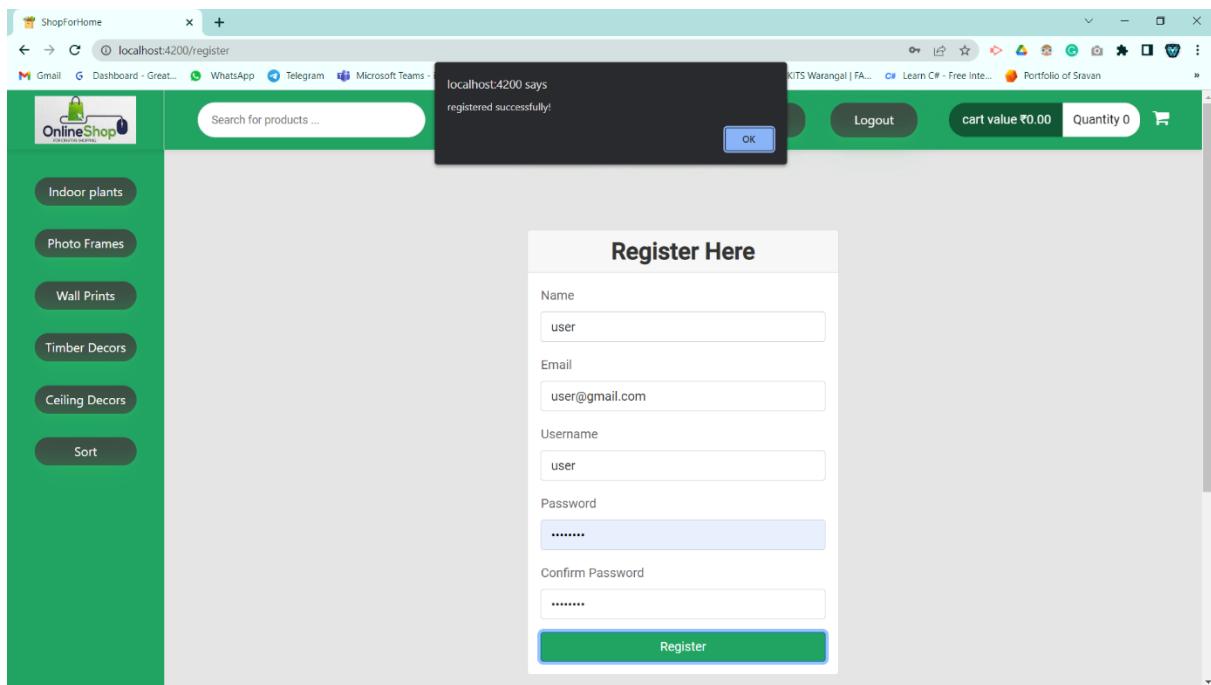


3.2 Registration Page

One of the most stressing parts about joining a website is to provide detailed information on a Log-in form, just to have complete access to that particular website. On one hand where log-in forms are essential for website owners, they can also be cumbersome and often take people off of a website. This is because they lookout for other easier log-in options on similar kind of websites where it is not always a hassle to log in to a website to use it.

With a fast-paced generation, youngsters and adults need one click options to log in to a website. Their time is precious and they do not have the energy in them to always type in their complete email address or name for access to these websites.

So we have created Registration Page with minimum data required and a simple username option.

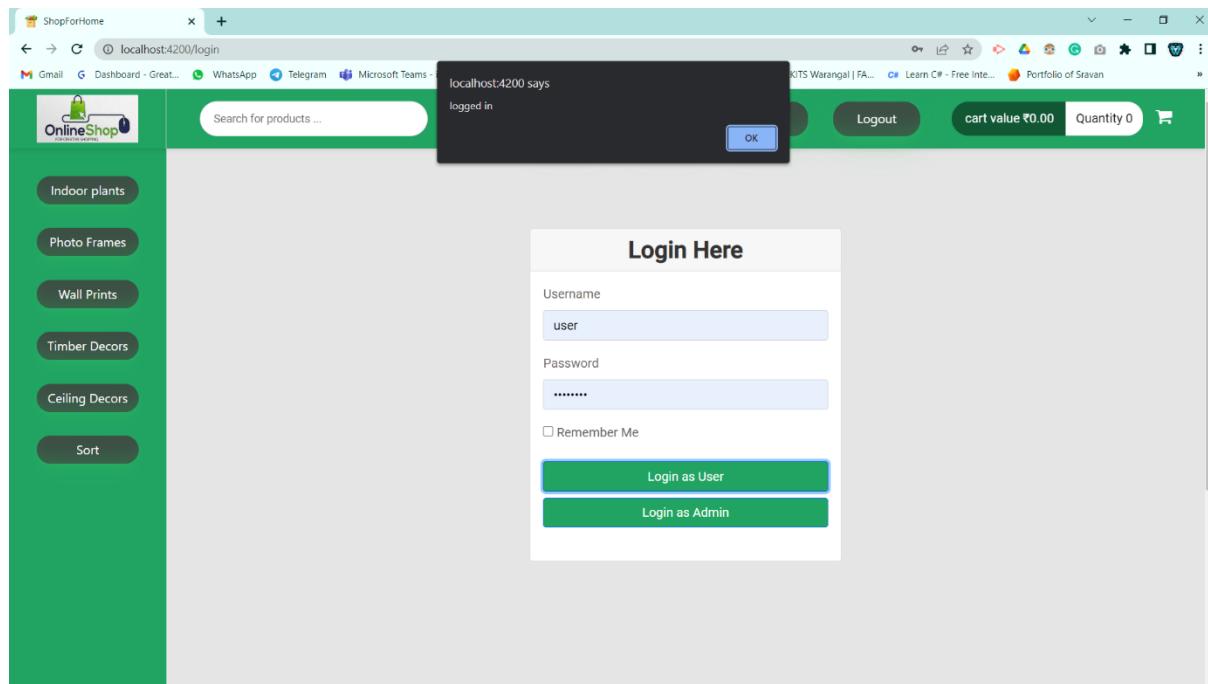


3.3 Login Page

Logging in not only provides site access for the user, but also allows the website to track user actions and behavior. Logging off a webpage or site may be manual by the user or they can occur automatically when certain conditions (such as closing the page, turning off the computer, a long time delay, etc.)

A Login form is used to enter authentication credentials to access a restricted page or form. The login form contains a field for the username and another for the password. When the login form is submitted its underlying code checks that the credentials are authentic, giving the user can access the restricted page.

So we have a Two login options Login as User and Login as Admin.



3.4 Product Categories

Different types of products are made available viz. Indoor Plants, Photo Frames, Wall Prints, Timber Decors and Ceiling Decors.

The screenshot shows the Timber Decors category page. The left sidebar has buttons for Indoor plants, Photo Frames, Wall Prints, Timber Decors (which is selected and highlighted in green), Ceiling Decors, and Sort. The main content area displays eight timber decor items arranged in two rows of four. Each item includes a small image, the name, and a price. Below each name are two buttons: 'Add to Cart' (green) and 'Add to Wishlist' (red).

| Timber Decor | Price | Add to Cart | Add to Wishlist |
|----------------|-----------|-------------|-----------------|
| Timber Decor-1 | ₹4,999.00 | Add to Cart | Add to Wishlist |
| Timber Decor-2 | ₹5,999.00 | Add to Cart | Add to Wishlist |
| Timber Decor-3 | ₹7,999.00 | Add to Cart | Add to Wishlist |
| Timber Decor-4 | ₹5,999.00 | Add to Cart | Add to Wishlist |
| Timber Decor-5 | ₹8,888.00 | Add to Cart | Add to Wishlist |
| Timber Decor-6 | ₹6,000.00 | Add to Cart | Add to Wishlist |
| Timber Decor-7 | ₹3,000.00 | Add to Cart | Add to Wishlist |
| Timber Decor-8 | ₹3,999.00 | Add to Cart | Add to Wishlist |

The screenshot shows the Wall Prints category page. The left sidebar has buttons for Indoor plants, Photo Frames, Wall Prints (selected and highlighted in green), Timber Decors, Ceiling Decors, and Sort. The main content area displays eight wall print items arranged in two rows of four. Each item includes a small image, the name, and a price. Below each name are two buttons: 'Add to Cart' (green) and 'Add to Wishlist' (red).

| Wall Print | Price | Add to Cart | Add to Wishlist |
|--------------|-----------|-------------|-----------------|
| Wall Print-1 | ₹599.00 | Add to Cart | Add to Wishlist |
| Wall Print-2 | ₹899.00 | Add to Cart | Add to Wishlist |
| Wall Print-3 | ₹1,500.00 | Add to Cart | Add to Wishlist |
| Wall Print-4 | ₹1,600.00 | Add to Cart | Add to Wishlist |
| Wall Print-5 | ₹2,000.00 | Add to Cart | Add to Wishlist |
| Wall Print-6 | ₹999.00 | Add to Cart | Add to Wishlist |
| Wall Print-7 | ₹1,600.00 | Add to Cart | Add to Wishlist |
| Wall Print-8 | ₹1,789.00 | Add to Cart | Add to Wishlist |

Product Details after selecting a product.

The screenshot shows a web browser window for 'ShopForHome' at localhost:4200/products/20. The page displays a product named 'Plant-20' with a price of ₹499.00. The product image shows several green plants in pots on a wooden surface. The left sidebar has categories like Indoor plants, Photo Frames, Wall Prints, Timber Decor, Ceiling Decor, and Sort. The top navigation bar includes links for Search, Wishlist, Register, Login, Logout, cart value ₹0.00, and Quantity 0.

Product Details:

- Name:** Plant-20
- Price:** ₹499.00
- Description:** Blooming blossoming; flowering; flourishing; thriving in vigor, health and beauty. Blossoming the process and time of budding and unfolding of blossoms. Blushing having a delicate and warm color like some flowers; roseate; blooming

[Back to Product List](#)

3.5 Cart View

The integration of a shopping cart into a website enhances the flexibility and working of the portal, which act as a magnet and pull maximum customers.

Customers find it super-easy to track and process the multiple orders with the presence of an shopping cart. No unnecessary repetitions, which are a big turn-off for the customers. A business is likely to attract more customers if it is making a better choice of merging a digital cart.

So we have created this View Cart feature for the benefit of customers.

The screenshot shows a web browser window for 'ShopForHome' at 'localhost:4200/cart-details'. The header includes a logo, a search bar, and navigation links for 'Search', 'Wishlist', 'Register', 'Logout', and a shopping cart icon indicating 'cart value ₹7,898.00' and 'Quantity 5'. On the left, a sidebar lists categories: 'Indoor plants', 'Photo Frames', 'Wall Prints', 'Timber Decors', 'Ceiling Decors', and 'Sort'. The main content area displays a table of items in the cart:

| Product Image | Product Detail | Order Details |
|---------------|----------------------|--|
| | Frame-1 ₹1,000.00 | Product Quantity: 1 Subtotal: ₹1,000.00 |
| | Frame-2 ₹2,000.00 | Product Quantity: 1 Subtotal: ₹2,000.00 |
| | Frame-3 ₹4,000.00 | Product Quantity: 1 Subtotal: ₹4,000.00 |
| | Plant-1 ₹399.00 | Product Quantity: 1 Subtotal: ₹399.00 |

3.6 Discount Page

The customer acquisition process starts with building awareness and interest, and discounts are a proven way to do just that. Consumers are attracted by discounts and tend to share them via word-of-mouth—an efficient way to reach new customers while keeping customer acquisition costs in check.

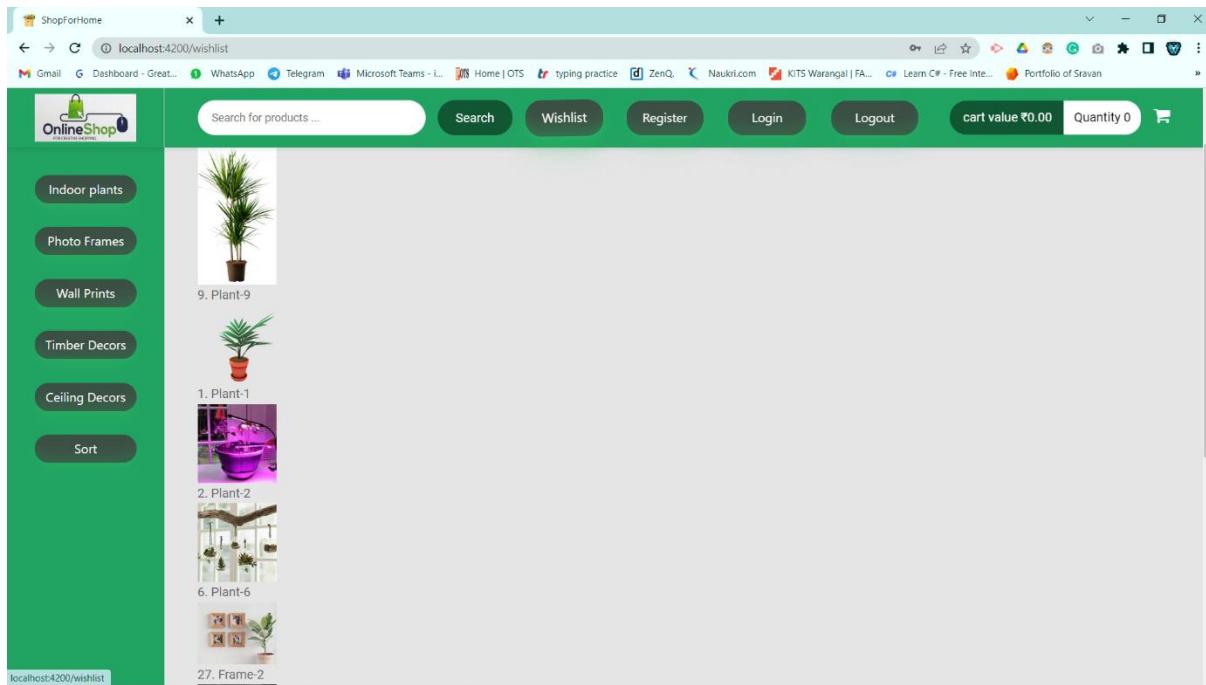
The screenshot shows a web browser window for 'ShopForHome' at 'localhost:4200/cart-details'. The page displays a shopping cart with two items:

| Product Image | Product Detail | Order Details |
|--|-----------------------------|---|
| | Plant-4 ₹499.00 | Product Quantity: 1 Subtotal: ₹499.00 |
| | Timber Decor-7 ₹3,000.00 | Product Quantity: 1 Subtotal: ₹3,000.00 |
| <input type="text" value="C3BATCH2"/> <input type="button" value="Apply"/> | | Total Items Quantity: 2 Shipping: Free Total Price: ₹1,749.50 |

A message at the bottom indicates: "Hurrey..! 50% Discount Applied". A red 'Check Out' button is visible at the bottom right.

3.7 Wishlist

A wishlist allows shoppers to create personalized collections of products they want to buy and save them in their user account for future reference. Wishlists signify a customer's interest in a product without an immediate intent to purchase.



3.8 Admin Dashboard

The dashboard is the front page of the Administration UI. It provides convenient shortcuts for common management tasks, some server information, and Review Board project news updates.

The abbreviation CRUD stands for create, read, update, and delete in computer programming. We have added CRUD operators for the admin dashboard.

CRUD on USER :

- Show all user.

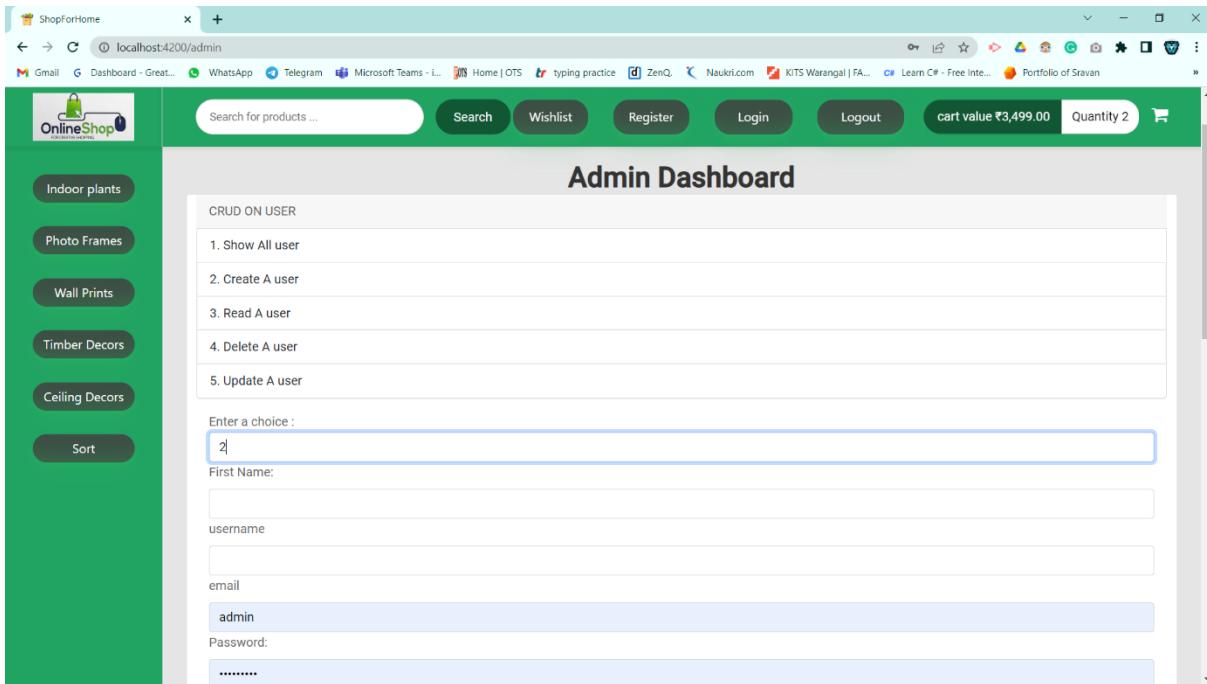
The screenshot shows a web browser window titled "ShopForHome" with the URL "localhost:4200/admin". The page has a green header bar with a search bar, a navigation bar with "Search", "Wishlist", "Register", "Login", and "Logout" buttons, and a cart summary showing "cart value ₹3,499.00" and "Quantity 2". On the left, there's a sidebar with categories: Indoor plants, Photo Frames, Wall Prints, Timber Decors, Ceiling Decors, and a "Sort" button. The main content area is titled "Admin Dashboard" and contains a section titled "CRUD ON USER" with the following list:

- 1. Show All user
- 2. Create A user
- 3. Read A user
- 4. Delete A user
- 5. Update A user

Below this, there's a form field labeled "Enter a choice :" with the number "1" selected. A table below shows two user entries:

| ID | Username | Name | email |
|----|----------|-------|-----------------|
| 1 | admin | admin | admin@gmail.com |
| 3 | user | user | user@gmail.com |

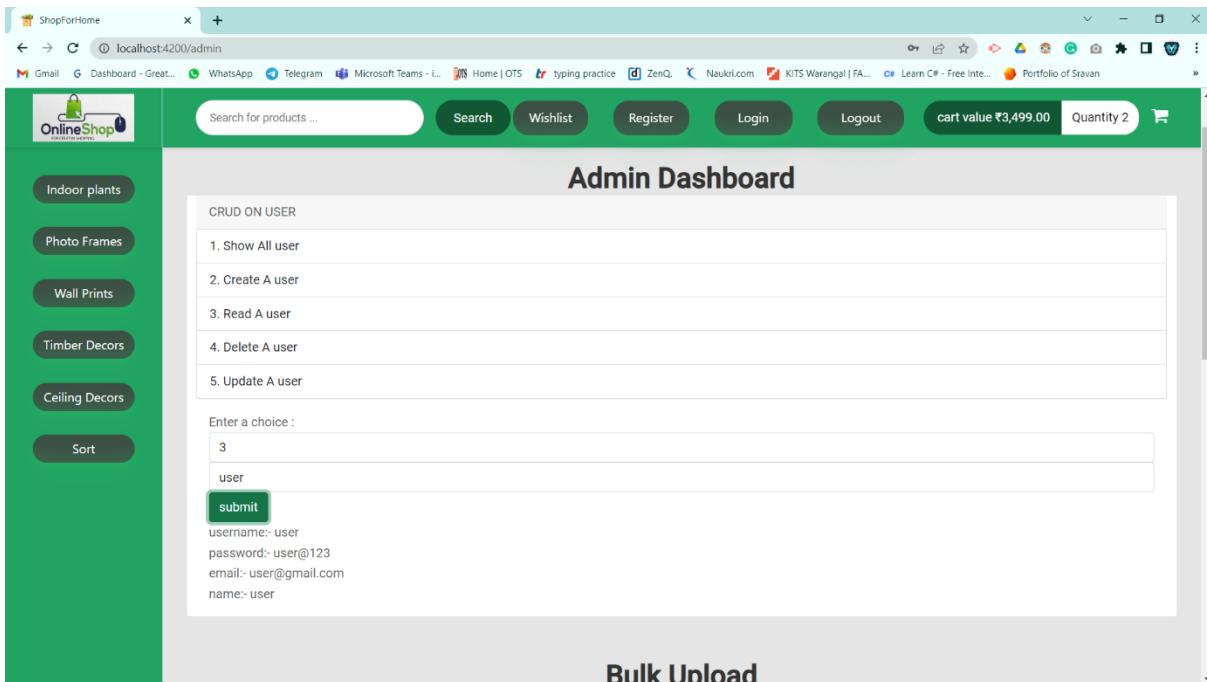
b. Create A User



The screenshot shows the Admin Dashboard with the following details:

- Header:** Search for products ..., Search, Wishlist, Register, Login, Logout, cart value ₹3,499.00, Quantity 2.
- Sidebar:** Indoor plants, Photo Frames, Wall Prints, Timber Decors, Ceiling Decors, Sort.
- Main Content:**
 - Section:** CRUD ON USER
 - List:** 1. Show All user, 2. Create A user, 3. Read A user, 4. Delete A user, 5. Update A user.
 - Text:** Enter a choice :
 - Form Fields:** First Name: (empty), username: (empty), email: (empty), Password: (admin).

c. Read A User



The screenshot shows the Admin Dashboard with the following details:

- Header:** Search for products ..., Search, Wishlist, Register, Login, Logout, cart value ₹3,499.00, Quantity 2.
- Sidebar:** Indoor plants, Photo Frames, Wall Prints, Timber Decors, Ceiling Decors, Sort.
- Main Content:**
 - Section:** CRUD ON USER
 - List:** 1. Show All user, 2. Create A user, 3. Read A user, 4. Delete A user, 5. Update A user.
 - Text:** Enter a choice :
 - Form Fields:** user.
 - Buttons:** submit.
 - Output:** username:- user, password:- user@123, email:- user@gmail.com, name:- user.
 - Section:** Bulk Upload

CRUD on PRODUCTS :

a. Show all Products

The screenshot shows the 'ShopForHome' admin dashboard. On the left, there's a sidebar with categories: Indoor plants, Photo Frames, Wall Prints, Timber Decors, Ceiling Decors, and Sort. The main area has a header with a search bar, 'Search', 'Wishlist', 'Register', 'Login', 'Logout', and a shopping cart icon showing 'cart value ₹3,499.00' and 'Quantity 2'. Below the header, a section titled 'CRUD ON PRODUCTS' lists six options: 1. Show All Products, 2. Create A Prouct, 3. Read A Product, 4. Delete A Product, 5. get all stocks, and 6. get sales report. An input field labeled 'Enter a choice:' contains the number '1'. A table below shows product details:

| Name | sku | unit price |
|---------|-----------------|------------|
| Plant-1 | SHOP-FOR-HOME00 | 399 |
| Plant-2 | SHOP-FOR-HOME01 | 599 |
| Plant-3 | SHOP-FOR-HOME02 | 799 |
| Plant-4 | SHOP-FOR-HOME03 | 499 |
| Plant-5 | SHOP-FOR-HOME04 | 799 |
| Plant-6 | SHOP-FOR-HOME05 | 699 |

b. Get all Stocks

This screenshot is similar to the previous one but shows the result of selecting option 5 ('get all stocks'). The main area displays the same 'CRUD ON PRODUCTS' menu and sidebar. The input field now contains '5'. The table below shows the same products as before, but includes an additional column for 'units in stock'.

| Name | sku | unit price | units in stock |
|---------|-----------------|------------|----------------|
| Plant-1 | SHOP-FOR-HOME00 | 399 | 100 |
| Plant-2 | SHOP-FOR-HOME01 | 599 | 100 |
| Plant-3 | SHOP-FOR-HOME02 | 799 | 100 |
| Plant-4 | SHOP-FOR-HOME03 | 499 | 100 |
| Plant-5 | SHOP-FOR-HOME04 | 799 | 100 |
| Plant-6 | SHOP-FOR-HOME05 | 699 | 100 |
| Plant-7 | SHOP-FOR-HOME06 | 999 | 100 |

c. Get Sales Report

The screenshot shows the ShopForHome admin dashboard. On the left sidebar, there are several categories: Indoor plants, Photo Frames, Wall Prints, Timber Decor, Ceiling Decor, and Sort. A 'Search for products ...' bar is at the top. The main area has a green header 'CRUD ON PRODUCTS' containing numbered options: 1. Show All Products, 2. Create A Prouct, 3. Read A Product, 4. Delete A Product, 5. get all stocks, and 6. get sales report. The number '6' is highlighted in blue. Below this is a table with columns: Name, sku, unit price, units in stock, and Date Created. The table lists seven products named Plant-1 through Plant-7, each with a unique SKU and creation date.

| Name | sku | unit price | units in stock | Date Created |
|---------|-----------------|------------|----------------|-------------------------------|
| Plant-1 | SHOP-FOR-HOME00 | 399 | 100 | 2022-08-04T11:52:47.000+00:00 |
| Plant-2 | SHOP-FOR-HOME01 | 599 | 100 | 2022-08-04T11:52:47.000+00:00 |
| Plant-3 | SHOP-FOR-HOME02 | 799 | 100 | 2022-08-04T11:52:47.000+00:00 |
| Plant-4 | SHOP-FOR-HOME03 | 499 | 100 | 2022-08-04T11:52:47.000+00:00 |
| Plant-5 | SHOP-FOR-HOME04 | 799 | 100 | 2022-08-04T11:52:47.000+00:00 |
| Plant-6 | SHOP-FOR-HOME05 | 699 | 100 | 2022-08-04T11:52:47.000+00:00 |
| Plant-7 | SHOP-FOR-HOME06 | 999 | 100 | 2022-08-04T11:52:47.000+00:00 |

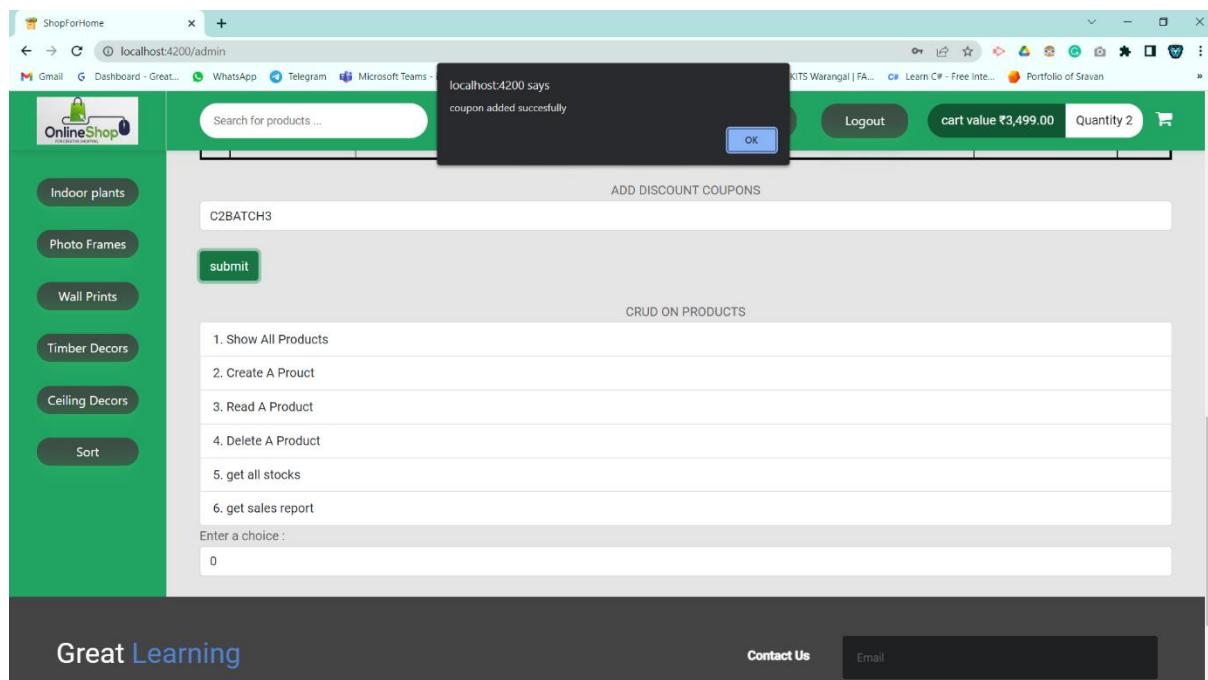
d. Bulk Upload

Bulk Upload feature allows sellers to easily manage large inventories. This feature allows the use of a .csv file, which can be downloaded directly from our export feature. Sellers have the ability to Add new listings, Change listing details, or Delete listing details, for all items in the exported inventory file.

The screenshot shows the 'Bulk Upload' page. It features a 'Choose File' button with the path 'BulkUpload1.csv'. Below this is a table with columns: ID, Product Name, Product Description, Stock Available, and Price. The table contains 10 rows, each representing a Photo Frame with a detailed description and specific stock and price values.

| ID | Product Name | Product Description | Stock Available | Price |
|----|--------------|---|-----------------|-------|
| 1 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 699 |
| 2 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 599 |
| 3 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 1999 |
| 4 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 2099 |
| 5 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 699 |
| 6 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 599 |
| 7 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 1899 |
| 8 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 2099 |
| 9 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 1999 |
| 10 | Photo Frame | The photo frame is a home decor item which highlights our memories on a wall or table | 100 | 1699 |

As an Admin, one can set the discount coupons for the specific set of users. So we have this **ADD DISCOUNT COUPONS** option for the Admin in Admin Dashboard.



CHAPTER 4: Code Snapshots

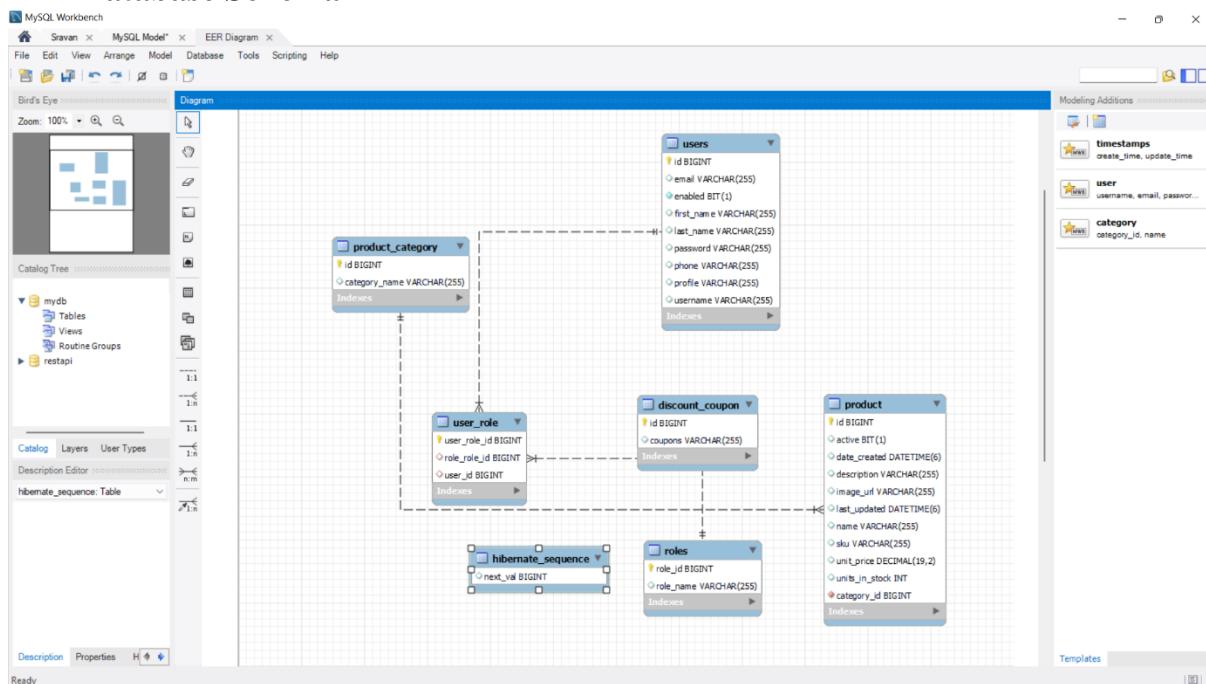
4.1 ANGULAR

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under the "FRONTEND" folder, including files like "wishlist.component.html", "wishlist.component.ts", "corousel", "corousel.component.css", "corousel.component.html", "corousel.component.ts", "services", "cart.service.spec.ts", "cart.service.ts", "products.service.spec.ts", "products.service.ts", "wishlist-service.service.spec.ts", "wishlist-service.service.ts", and "app.component.css".
- Code Editor:** The "app.component.html" file is open, displaying the HTML template for the application. It includes meta tags for charset, http-equiv, and viewport, followed by a title and a link to a CSS file. The template features a header section with a logo and navigation links.
- Terminal:** The terminal shows the output of a build command: "PS C:\Users\sree\Desktop\Wipro\C3_Capstone_Project_Group2\frontend> ng build --prod" followed by "Compiling module 0 of 1... Compiled successfully." and session restoration information.
- Status Bar:** Shows the current file is "app.component.html - frontend - Visual Studio Code", the line count is "Ln 65, Col 107", and the character count is "Spaces: 2 LF: 108 HTML".

4.2 SQL

Database Schema



Database Product Categories

```

create database if not exists restapi;
use restapi;
INSERT INTO product_category(category_name) VALUES ('Indoor plants');
INSERT INTO product_category(category_name) VALUES ('Photo Frames');
INSERT INTO product_category(category_name) VALUES ('Wall Prints');
INSERT INTO product_category(category_name) VALUES ('Timber Decors');
INSERT INTO product_category(category_name) VALUES ('Ceiling Decors');
INSERT INTO product_category(category_name) VALUES ('Sort');

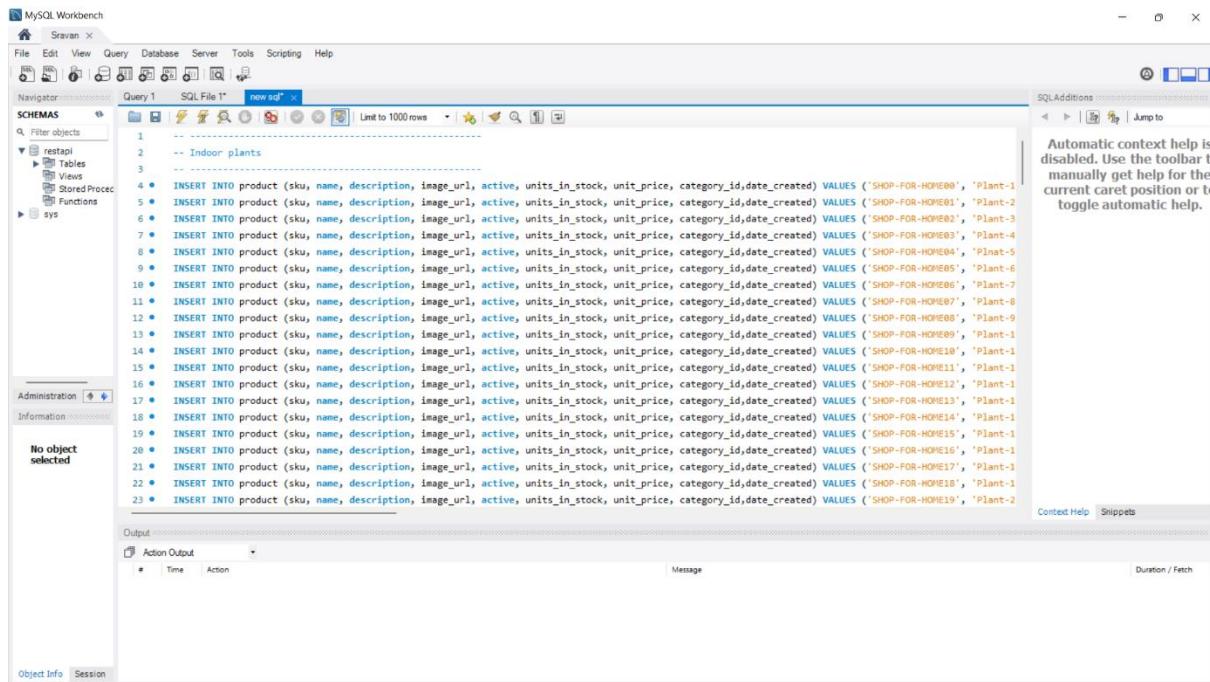
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (sys selected), Filter objects.
- Query Editor:** Title: unconnected, Schemas: Schemas, Query 1: new sql*, Content: The above SQL code.
- Output:** Action Output table showing the execution history of the queries.
- SQL Additions:** A note: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

| # | Action | Time | Action | Message | Duration / Fetch |
|----|-------------|--|--------|-------------------|------------------|
| 1 | 1 11:52:15 | DROP DATABASE `restapi`; | | 7 row(s) affected | 0.172 sec |
| 2 | 11:52:21 | create database if not exists restapi | | 1 row(s) affected | 0.000 sec |
| 3 | 11:52:21 | use restapi | | 0 row(s) affected | 0.000 sec |
| 4 | 4 11:52:32 | INSERT INTO product_category(category_name) VALUES ('Indoor plants') | | 1 row(s) affected | 0.000 sec |
| 5 | 5 11:52:32 | INSERT INTO product_category(category_name) VALUES ('Photo Frames') | | 1 row(s) affected | 0.000 sec |
| 6 | 6 11:52:32 | INSERT INTO product_category(category_name) VALUES ('Wall Prints') | | 1 row(s) affected | 0.000 sec |
| 7 | 7 11:52:32 | INSERT INTO product_category(category_name) VALUES ('Timber Decors') | | 1 row(s) affected | 0.000 sec |
| 8 | 8 11:52:32 | INSERT INTO product_category(category_name) VALUES ('Ceiling Decors') | | 1 row(s) affected | 0.000 sec |
| 9 | 9 11:52:32 | INSERT INTO product_category(category_name) VALUES ('Sort') | | 1 row(s) affected | 0.000 sec |
| 10 | 10 11:52:47 | INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id, date_creat... | | 1 row(s) affected | 0.015 sec |
| 11 | 11 11:52:47 | INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id, date_creat... | | 1 row(s) affected | 0.000 sec |

Database Product Details



The screenshot shows the MySQL Workbench interface with a query editor containing 23 INSERT statements for a 'product' table. The statements are as follows:

```
1 -- -----
2 -- Indoor plants
3 --
4 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME00', 'Plant-1
5 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME01', 'Plant-2
6 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME02', 'Plant-3
7 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME03', 'Plant-4
8 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME04', 'Plant-5
9 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME05', 'Plant-6
10 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME06', 'Plant-7
11 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME07', 'Plant-8
12 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME08', 'Plant-9
13 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME09', 'Plant-1
14 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME10', 'Plant-1
15 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME11', 'Plant-1
16 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME12', 'Plant-1
17 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME13', 'Plant-1
18 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME14', 'Plant-1
19 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME15', 'Plant-1
20 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME16', 'Plant-1
21 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME17', 'Plant-1
22 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME18', 'Plant-1
23 • INSERT INTO product (sku, name, description, image_url, active, units_in_stock, unit_price, category_id,date_created) VALUES ('SHOP-FOR-HOME19', 'Plant-2
```

4.3 Spring Tool Suite

Running Spring Boot

The screenshot shows the Spring Tool Suite interface with the following details:

- Package Explorer:** Shows the project structure for "demo [boot]".
- DemoApplication.java:** The main entry point of the application.
- application.properties:** Configuration file for the application.
- Console:** Displays the application logs. The logs show the application starting up, bootstrapping Spring Data JPA repositories, initializing Spring embedded WebApplicationContext, and starting the Tomcat service. It also shows the configuration of Hibernate, including the dialect, version, and persistence unit.

```

package com.example.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}

```

```

2022-08-04 11:52:56,555 INFO 1228 --- [           main] com.example.demo.DemoApplication      : Starting DemoApplication using Java 17.0.3 on Sravan with PID 1
2022-08-04 11:52:56,555 INFO 1228 --- [           main] com.example.demo.DemoApplication      : No active profile set, falling back to 1 default profile: "default"
2022-08-04 11:52:57,421 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-08-04 11:52:57,421 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 66 ms. Found 5 JPA
2022-08-04 11:52:57,621 INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-08-04 11:52:58,559 INFO 1228 --- [           main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.63]
2022-08-04 11:52:58,559 INFO 1228 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1936 ms
2022-08-04 11:52:58,294 INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-08-04 11:52:58,294 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000412: Hibernate ORM core version 5.6.9.Final
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000001: Hibernate Commons Annotations 5.1.2.Final
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000408: Using dialect: org.hibernate.dialect.MySQL8Dialect
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.enve

```

Spring Boot Application Properties

The screenshot shows the Spring Tool Suite interface with the following details:

- Package Explorer:** Shows the project structure for "demo [boot]".
- DemoApplication.java:** The main entry point of the application.
- application.properties:** Configuration file for the application.
- Console:** Displays the application logs. The logs show the application starting up, bootstrapping Spring Data JPA repositories, initializing Spring embedded WebApplicationContext, and starting the Tomcat service. It also shows the configuration of Hibernate, including the dialect, version, and persistence unit.

```

spring.datasource.url=jdbc:mysql://localhost:3306/restapi?useSSL=false&useUnicode=yes&characterEncoding=UTF-8&allowPublicKeyRetrieval=true&serverTimezone=UTC
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.showSql=true
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true

```

```

2022-08-04 11:52:56,255 INFO 1228 --- [           main] com.example.demo.DemoApplication      : Starting DemoApplication using Java 17.0.3 on Sravan with PID 1
2022-08-04 11:52:56,255 INFO 1228 --- [           main] com.example.demo.DemoApplication      : No active profile set, falling back to 1 default profile: "default"
2022-08-04 11:52:57,441 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-08-04 11:52:57,441 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 66 ms. Found 5 JPA
2022-08-04 11:52:57,621 INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2022-08-04 11:52:58,294 INFO 1228 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-08-04 11:52:58,294 INFO 1228 --- [           main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.63]
2022-08-04 11:52:58,294 INFO 1228 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1936 ms
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000412: Hibernate ORM core version 5.6.9.Final
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000001: Hibernate Commons Annotations 5.1.2.Final
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000408: Using dialect: org.hibernate.dialect.MySQL8Dialect
2022-08-04 11:52:58,311 INFO 1228 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.enve

```

Microservices

The screenshot shows the Spring Tool Suite interface with the following details:

- Project Explorer:** Shows the project structure for "demo" with packages like com.example.demo, com.example.demo.config, com.example.demo.controller, com.example.demo.entity, com.example.demo.repo, com.example.demo.service, and com.example.demo.service.impl.
- Code Editor:** Displays the content of `UserController.java`:


```

1 package com.example.demo.controller;
2
3 import java.util.HashSet;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.*;
7
8 @RestController
9 @CrossOrigin("*")
10 @RequestMapping("/user")
11 public class UserController {
12
13     @Autowired
14     private UserService userService;
15
16     @PostMapping("/")
17     public ResponseEntity<UserRole> save(@RequestBody User user) {
18         Set<UserRole> roles = new HashSet<>();
19
20         Role role = new Role();
21         role.setRoleId(45L);
22         role.setRoleName("NORMAL");
23
24         UserRole userRole = new UserRole();
25         userRole.setUser(user);
26         userRole.setRole(role);
27
28         roles.add(userRole);
29
30         return ResponseEntity.ok().body(userService.save(roles));
31     }
32
33 }
      
```
- Console:** Shows the application logs starting up:


```

2022-08-04 11:52:56.555 INFO 1228 --- [           main] com.example.demo.DemoApplication      : Starting DemoApplication using Java 17.0.3 on Sravan with PID 1
2022-08-04 11:52:56.568 INFO 1228 --- [           main] com.example.demo.DemoApplication      : No active profile set, falling back to 1 default profile: "defau
2022-08-04 11:52:57.541 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-08-04 11:52:57.621 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 66 ms. Found 5 JPA r
2022-08-04 11:52:58.294 INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2022-08-04 11:52:58.511 INFO 1228 --- [           main] o.apache.catalina.core.StandardService : Starting service Tomcat
      
```

Spring Entities

The screenshot shows the Spring Tool Suite interface with the following details:

- Project Explorer:** Shows the project structure for "demo" with packages like com.example.demo, com.example.demo.repo, com.example.demo.service, and com.example.demo.service.impl.
- POM.xml:** Displays the Maven configuration:


```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POX/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.0</version>
    <relativePath><!-- lookup parent from repository --&gt;
  &lt;/parent&gt;
  &lt;groupId&gt;com.example&lt;/groupId&gt;
  &lt;artifactId&gt;demo&lt;/artifactId&gt;
  &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;
  &lt;name&gt;demo&lt;/name&gt;
  &lt;description&gt;Demo project for Spring Boot&lt;/description&gt;
  &lt;properties&gt;
    &lt;java.version&gt;17&lt;/java.version&gt;
  &lt;/properties&gt;
  &lt;dependencies&gt;
    &lt;dependency&gt;
      &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;
      &lt;artifactId&gt;spring-boot-starter&lt;/artifactId&gt;
      &lt;version&gt;2.7.0&lt;/version&gt;
    &lt;/dependency&gt;
    &lt;dependency&gt;
      &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt;
      &lt;artifactId&gt;spring-boot-starter-web&lt;/artifactId&gt;
      &lt;version&gt;2.7.0&lt;/version&gt;
    &lt;/dependency&gt;
  &lt;/dependencies&gt;
&lt;/project&gt;
      </pre>

```
- Console:** Shows the application logs starting up, identical to the first screenshot:


```

2022-08-04 11:52:56.555 INFO 1228 --- [           main] com.example.demo.DemoApplication      : Starting DemoApplication using Java 17.0.3 on Sravan with PID 1
2022-08-04 11:52:56.568 INFO 1228 --- [           main] com.example.demo.DemoApplication      : No active profile set, falling back to 1 default profile: "defau
2022-08-04 11:52:57.541 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-08-04 11:52:57.621 INFO 1228 --- [           main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 66 ms. Found 5 JPA r
2022-08-04 11:52:58.294 INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8081 (http)
2022-08-04 11:52:58.511 INFO 1228 --- [           main] o.apache.catalina.core.StandardService : Starting service Tomcat
      
```

4.4 AWS

Uploading Files in S3 Bucket

The screenshot shows the AWS S3 console interface. The left sidebar has sections for Buckets, Storage Lens, Feature spotlight, and AWS Marketplace. The main area displays the properties of the bucket 'c3.capstoneproject.batch2.com'. The 'Properties' tab is selected. Key details shown include:

- AWS Region: US East (N. Virginia) us-east-1
- Amazon Resource Name (ARN): arn:aws:s3:::c3.capstoneproject.batch2.com
- Creation date: August 4, 2022, 13:28:31 (UTC+05:30)

The 'Bucket Versioning' section indicates it is disabled.

EC2 Instance

The screenshot shows the AWS EC2 Management Console. The left sidebar includes sections for EC2 Dashboard, Global View, Events, Tags, Limits, Instances (with sub-options like Instances, Images, and Elastic Block Store), and Feedback. The main area displays a list of instances:

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 |
|-----------------|---------------------|----------------|---------------|-------------------|--------------|-------------------|-------------|
| EC2S3-instance | i-0e0c321c18d3a6d85 | Running | t2.micro | 2/2 checks passed | No alarms | us-east-1d | ec2-54-14- |
| Userservice-env | i-008ba59287c9fdd2 | Running | t2.micro | 2/2 checks passed | No alarms | us-east-1a | ec2-54-23- |

Details for the instance 'i-0e0c321c18d3a6d85' are shown in the bottom panel, including its summary, public and private IP addresses, and monitoring status.

Elastic Load Balancer

The screenshot shows the AWS Management Console EC2 Management Console interface for creating a new load balancer. The left sidebar shows navigation options like EC2 Dashboard, Instances, and Images. The main area has tabs for Create Load Balancer and Actions. A search bar at the top right says "Search for services, features, blogs, docs, and more". Below it is a table showing one load balancer entry:

| Name | DNS name | State | VPC ID | Availability Zones | Type | Create |
|--------------|----------------------------|--------------|-----------------------|-----------------------------|-------------|--------|
| LoadBalancer | LoadBalancer-1989575933... | Provisioning | vpc-0d88fe7079490c263 | us-east-1c, us-east-1d, ... | application | August |

Below the table are tabs for Description, Listeners, Monitoring, Integrated services, and Tags. Under Basic Configuration, the following details are shown:

- Name: LoadBalancer
- ARN: arn:aws:elasticloadbalancing:us-east-1:559242741107:loadbalancer/app/LoadBalancer/bdd9594c15dbbc32
- DNS name: LoadBalancer-1989575933.us-east-1.elb.amazonaws.com (A Record)
- State: Provisioning
- Type: application
- Scheme: internet-facing
- IP address type: ipv4
- VPC: vpc-0d88fe7079490c263

At the bottom, there are links for Feedback, Unified Settings, and a copyright notice: © 2022, Amazon Internet Services Private Ltd. or its affiliates.

Autoscaling

The screenshot shows the AWS Management Console EC2 Management Console interface for Auto Scaling Groups. The left sidebar includes sections for Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area displays a success message: "GL-ASG, 1 Scaling policy created successfully. Group metrics collection is enabled." Below this is a table for Auto Scaling groups:

| Name | Launch template/configuration | Instances | Status | Desired capacity | Min | Max | Available |
|--------|-------------------------------|-----------|--------|------------------|-----|-----|-----------|
| GL-ASG | GL-instance Version Default | 2 | - | 2 | 1 | 10 | us-east-1 |

Below the table are tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The Details tab shows Group details:

| Desired capacity | Auto Scaling group name |
|------------------|---|
| 2 | GL-ASG |
| Minimum capacity | Date created |
| 1 | Thu Aug 04 2022 15:31:51 GMT+0530 (India Standard Time) |

At the bottom, there are links for Feedback, Unified Settings, and a copyright notice: © 2022, Amazon Internet Services Private Ltd. or its affiliates.

Connecting EC2 using Command Prompt

```
C:\Users\sree>cd Downloads
C:\Users\sree\Downloads>ssh -i "capstoneproject.pem" ec2-user@ec2-54-226-128-176.compute-1.amazonaws.com
The authenticity of host 'ec2-54-226-128-176.compute-1.amazonaws.com (54.226.128.176)' can't be established.
ECDSA key fingerprint is SHA256:LtxEf13yUYse9dWyaZACYNzGzR27EgA75hALFZnnz4Qw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-226-128-176.compute-1.amazonaws.com,54.226.128.176' (ECDSA) to the list of known hosts.

[ec2-user@ip-172-31-31-145 ~]$ sudo
usage: sudo -h | -k | -K | -V
usage: sudo -v [-Akns] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-Akns] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user]
      [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-Akns] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...
[ec2-user@ip-172-31-31-145 ~]$ sudo su
[root@ip-172-31-31-145 ec2-user]# yum
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to give some command
Usage: yum [options] COMMAND

List of Commands:

check          Check for problems in the rpmdb
check-update   Check for available package updates
clean          Remove cached data
depplist       List a package's dependencies
distribution-synchronization Synchronize installed packages to the latest available versions
downgrade     downgrade a package
erase          Remove a package or packages from your system
fs             Acts on the filesystem data of the host, mainly for removing docs/languages for minimal hosts.
fssnapshot    Creates filesystem snapshots, or lists/deletes current snapshots.
groups         Display, or use, the groups information
help          Display a helpful usage message
```

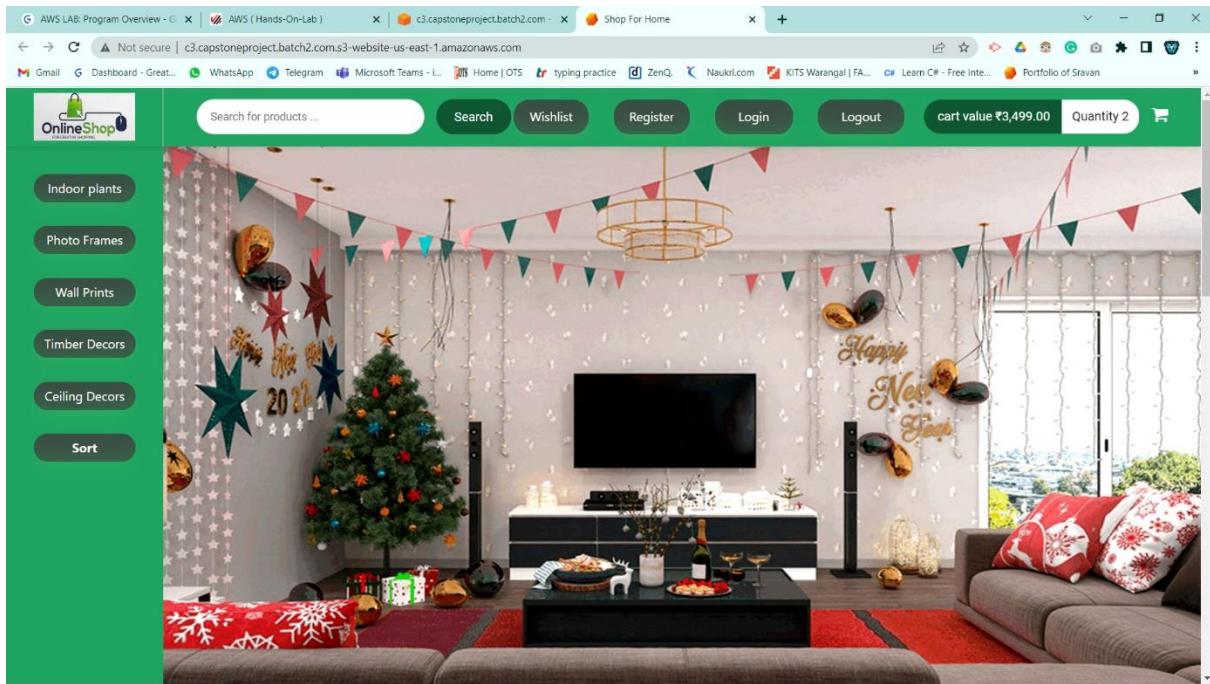
Installing Server Application

```
[root@ip-172-31-31-145 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httdp for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httdp.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.54-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.54-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

+---+---+---+---+---+
| ch | Version | Repository | Size | Package | Ar |
+---+---+---+---+---+
| httpd | x86_64 | 2.4.54-1.amzn2 | amzn2-core | 1.4 M | Installing: |
+---+---+---+---+---+
| apr | x86_64 | 1.7.0-9.amzn2 | amzn2-core | 122 k |
| apr-util | x86_64 | 1.6.1-5.amzn2.0.2 | amzn2-core | 99 k |
| apr-util-bdb | x86_64 | 1.6.1-5.amzn2.0.2 | amzn2-core | 19 k |
| generic-logos-httdp | noarch | 18.0.0-4.amzn2 | amzn2-core | 19 k |
| httpd-filesystem | noarch | 2.4.54-1.amzn2 | amzn2-core | 24 k |
| httpd-tools | x86_64 | 2.4.54-1.amzn2 | amzn2-core | 88 k |
| mailcap | noarch | 2.1.41-2.amzn2 | amzn2-core | 31 k |
| mod_http2 | x86_64 | 1.15.19-1.amzn2.0.1 | amzn2-core | 149 k |
```

AWS Website Hosting on S3



4.5 Git Repository

The screenshot shows a GitHub repository page for 'A-Sravan / C3CapstoneProjectBatch2'. The repository is private. The 'Code' tab is selected, showing the 'master' branch. The branch status is '1 commit ahead, 1 commit behind main'. The commit history lists several files: 'backend/demo', 'frontend', 'BulkUpload1.csv', 'Capstone batch2 ppt.pptx', 'Database Schema.png', and 'data.sql'. All commits are from 3 hours ago. The 'About' section indicates no description, website, or topics provided. It has 0 stars, 1 watching, and 0 forks. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published.

A-Sravan / C3CapstoneProjectBatch2 · Private

Code Issues Pull requests Actions Projects Security Insights Settings

This branch is 1 commit ahead, 1 commit behind main.

A-Sravan capstoneproject

| backend/demo | capstoneproject | 3 hours ago |
|--------------------------|-----------------|-------------|
| frontend | capstoneproject | 3 hours ago |
| BulkUpload1.csv | capstoneproject | 3 hours ago |
| Capstone batch2 ppt.pptx | capstoneproject | 3 hours ago |
| Database Schema.png | capstoneproject | 3 hours ago |
| data.sql | capstoneproject | 3 hours ago |

Add a README with an overview of your project. Add a README

About

No description, website, or topics provided.

0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

CHAPTER 5: CONCLUSION

Conclusion:

We have created a ShopForHome JAVA-AWS core application using the Angular, Spring Tool Suite and AWS (EC2 Instances, Load Balancer, Auto Scaling, S3), MySQL database. Hence, we are successful in deploying the application with all the requirements mentioned in the given problem statement.

The internet has opened so many opportunities for doing business online, and e-commerce is one of the most popular ones. Not only does it require low investment, it actually is a type of business that does not require a full-time commitment, especially if you are selling a limited number of products. You just make an online presence and promote the business here and there using both free and paid methods. However, as each business, it has its positive and negative sides. If you want to achieve success, you will have to create an organized strategy which is based on realistic goals and comprehensive analysis of the market.