



greatlearning
Learning for Life

SHOP FOR HOME

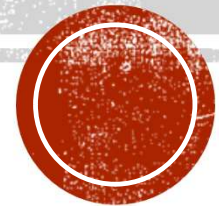
CAPSTONE PROJECT

Mentor: Praveen Tej

Batch no: G13-AWS SD APR CIII

Presented by:

- 1) Lakshmi Vineetha
- 2) Bhavya Kancharla
- 3) Sumanth
- 4) Mahima
- 5) Daneswari Aketi



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

Technologies required

We have developed this project using the below technology

- **HTML:** *Page layout has been designed in HTML*
- **CSS:** *CSS has been used for all the designing part*
- **JavaScript:** *All the validation task and animations has been developed by JavaScript*
- **Java Spring Boot:** *All the business and backend API logic has been implemented in Java Spring Boot*
- **MYSQL:** *.SQL files has been used as database for the project*
- **Angular:** *All the frontend logic has been implemented over the Angular and we used angular CLI for it & We have used this for calling all microservice API's*
- **Visual Studio Code-(VSS):** *For Angular IDE, we have used Visual Studio Code*

Objectives Sprint-1

1. Create git repository
2. Create database schema (all tables along with their relationships)
3. Create entities in Spring
4. Create Microservice based structure.
5. CRUD on User and Products
6. Create an e-commerce Template in Angular (Static only) to hold images and products lists.
7. Develop Search Functionality in Angular
8. Bulk upload implementation

1. Create git repository

The screenshot shows a web browser displaying the GitHub repository page for 'VaraPrasad08/capstone_project-g13-frontend'. The repository is public and has 0 forks and 0 stars. The page shows the repository's structure with a list of files and folders, all committed 17 minutes ago. The files include 'e2e', 'nginx', 'src', '.editorconfig', '.gitignore', 'Dockerfile', 'README.md', 'angular.json', and 'package-lock.json'. The right sidebar shows the 'About' section with no description, website, or topics provided, and the 'Releases' and 'Packages' sections with no published items.

GitHub - VaraPrasad08/capstone x +

github.com/VaraPrasad08/capstone_project-g13-frontend

Product Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

VaraPrasad08 / capstone_project-g13-frontend Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags

Go to file Code

About

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

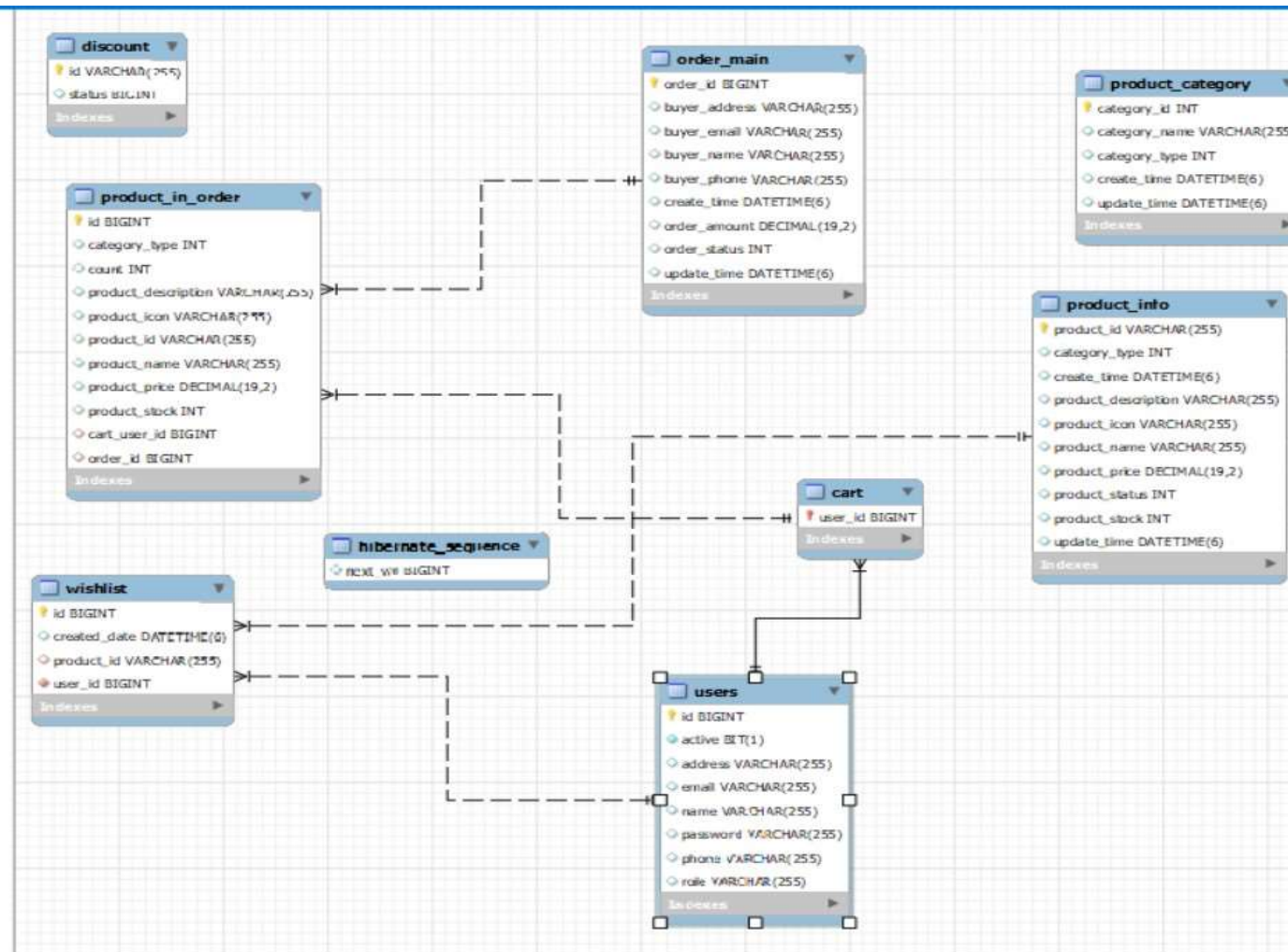
No releases published

Packages

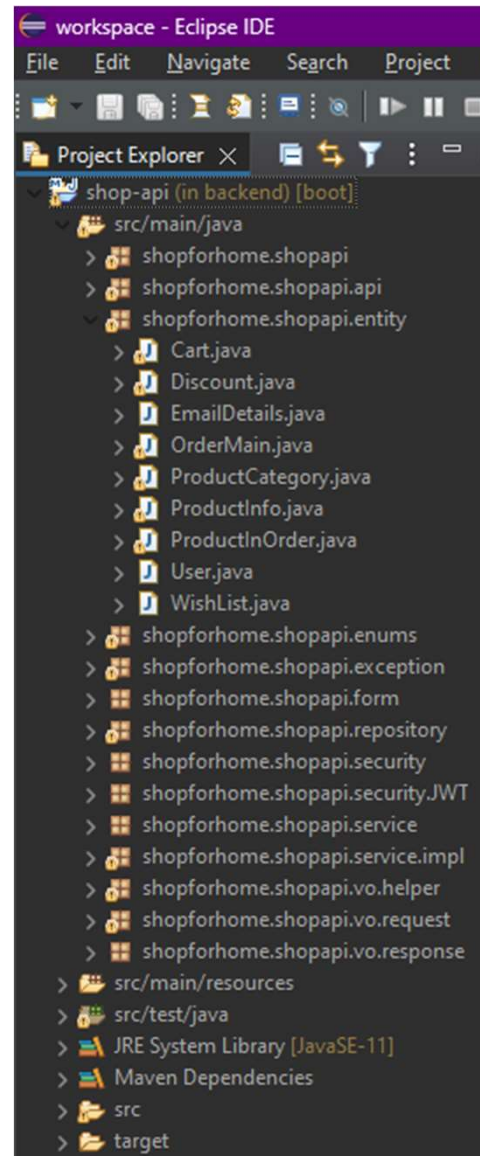
No packages published

File/Folder	Commit	Time
VaraPrasad08 first commit	724a304	17 minutes ago
e2e	first commit	17 minutes ago
nginx	first commit	17 minutes ago
src	first commit	17 minutes ago
.editorconfig	first commit	17 minutes ago
.gitignore	first commit	17 minutes ago
Dockerfile	first commit	17 minutes ago
README.md	first commit	17 minutes ago
angular.json	first commit	17 minutes ago
package-lock.json	first commit	17 minutes ago

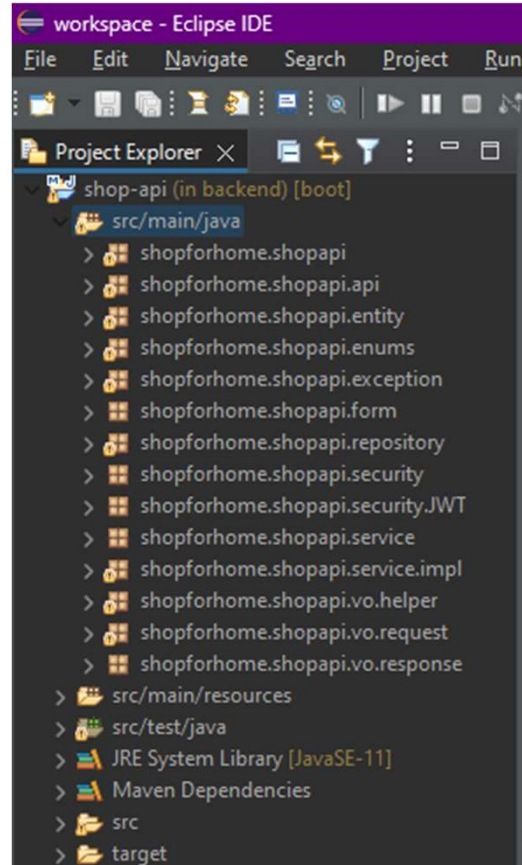
2 . Create database schema (all tables along with their relationships)



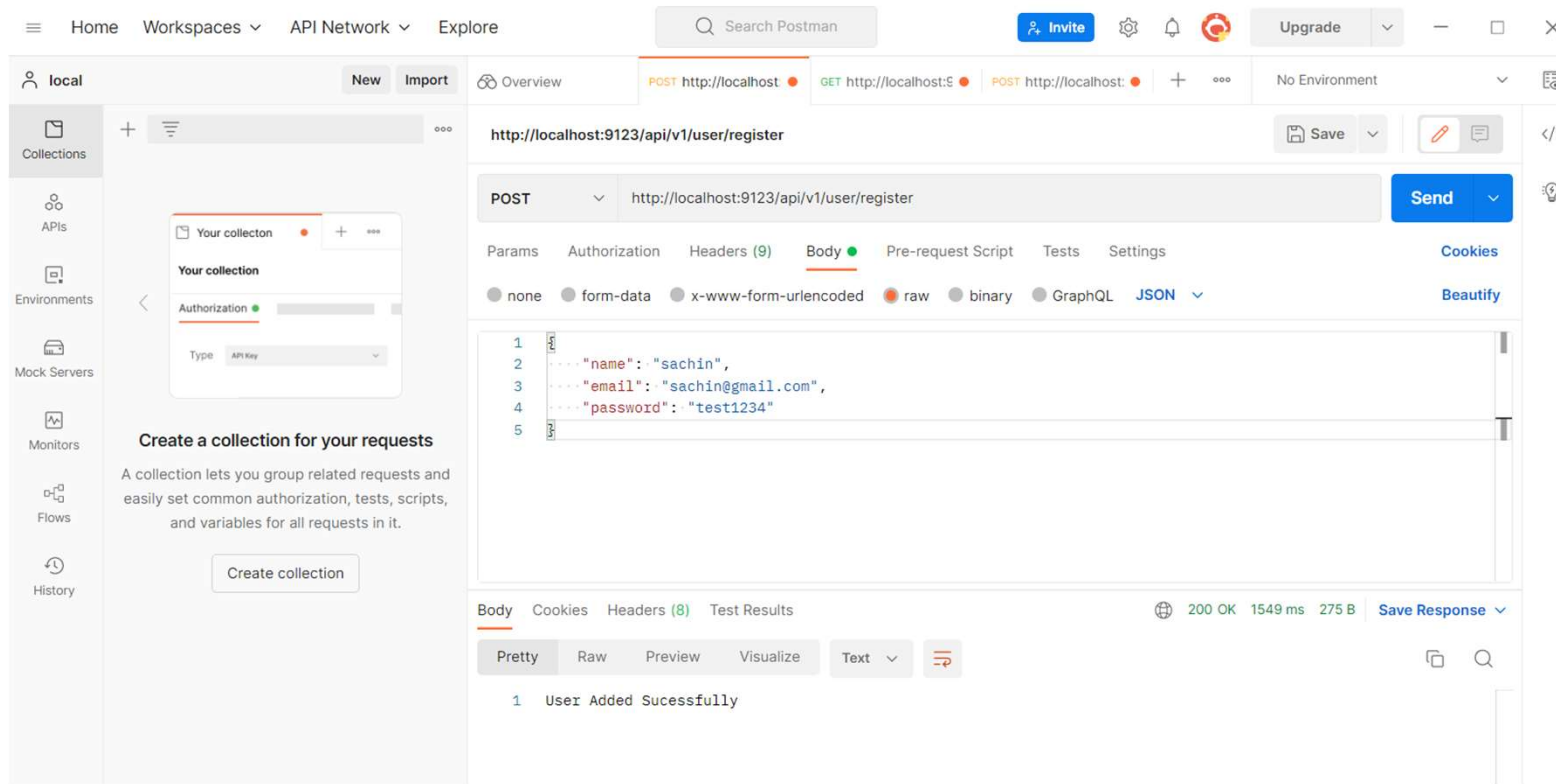
3. Create entities in Spring



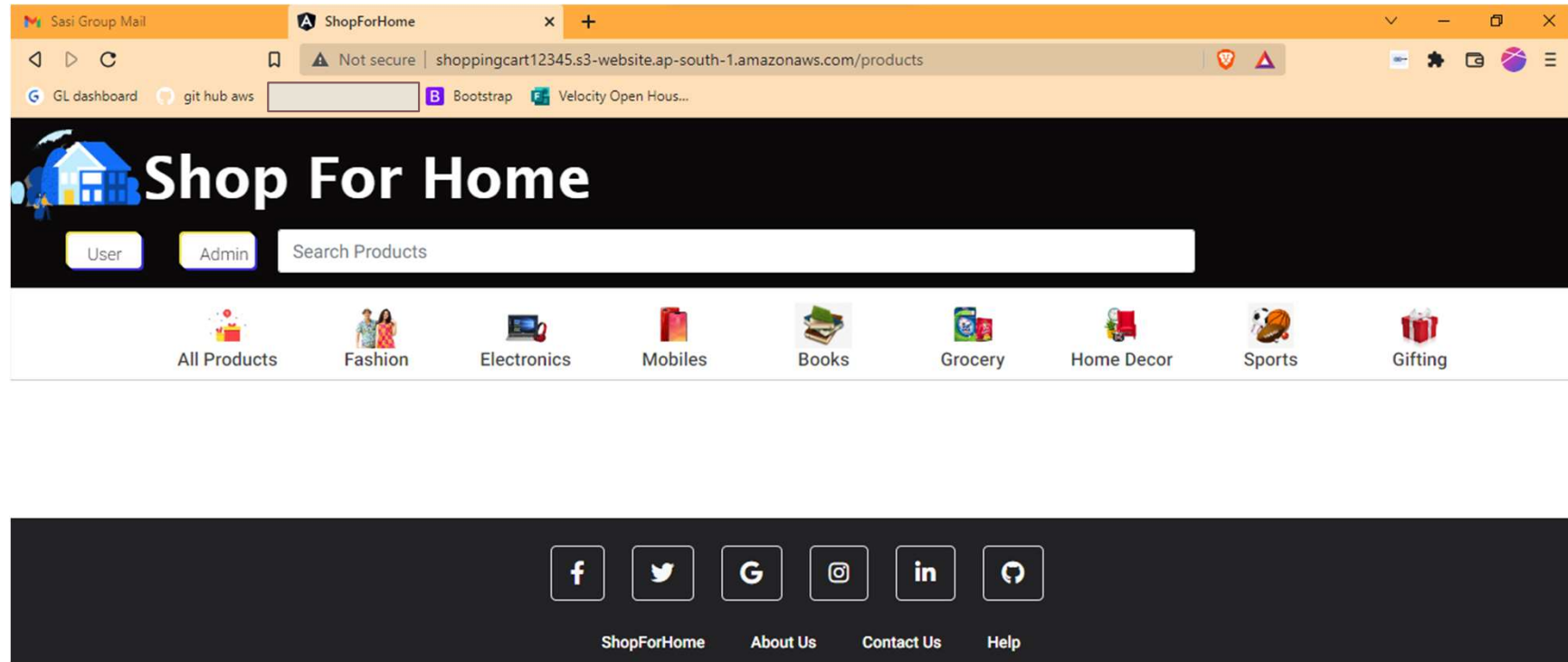
4. Create Microservice based structure



5. CRUD on User and Products



6. Create an e-commerce Template in Angular (Static only) to hold images and products lists.



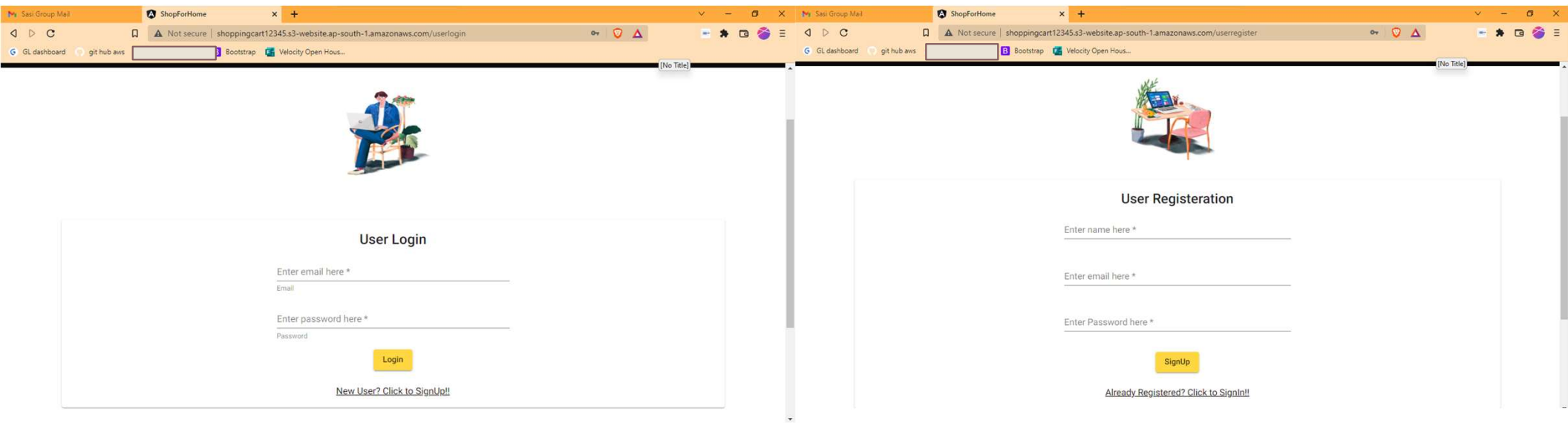
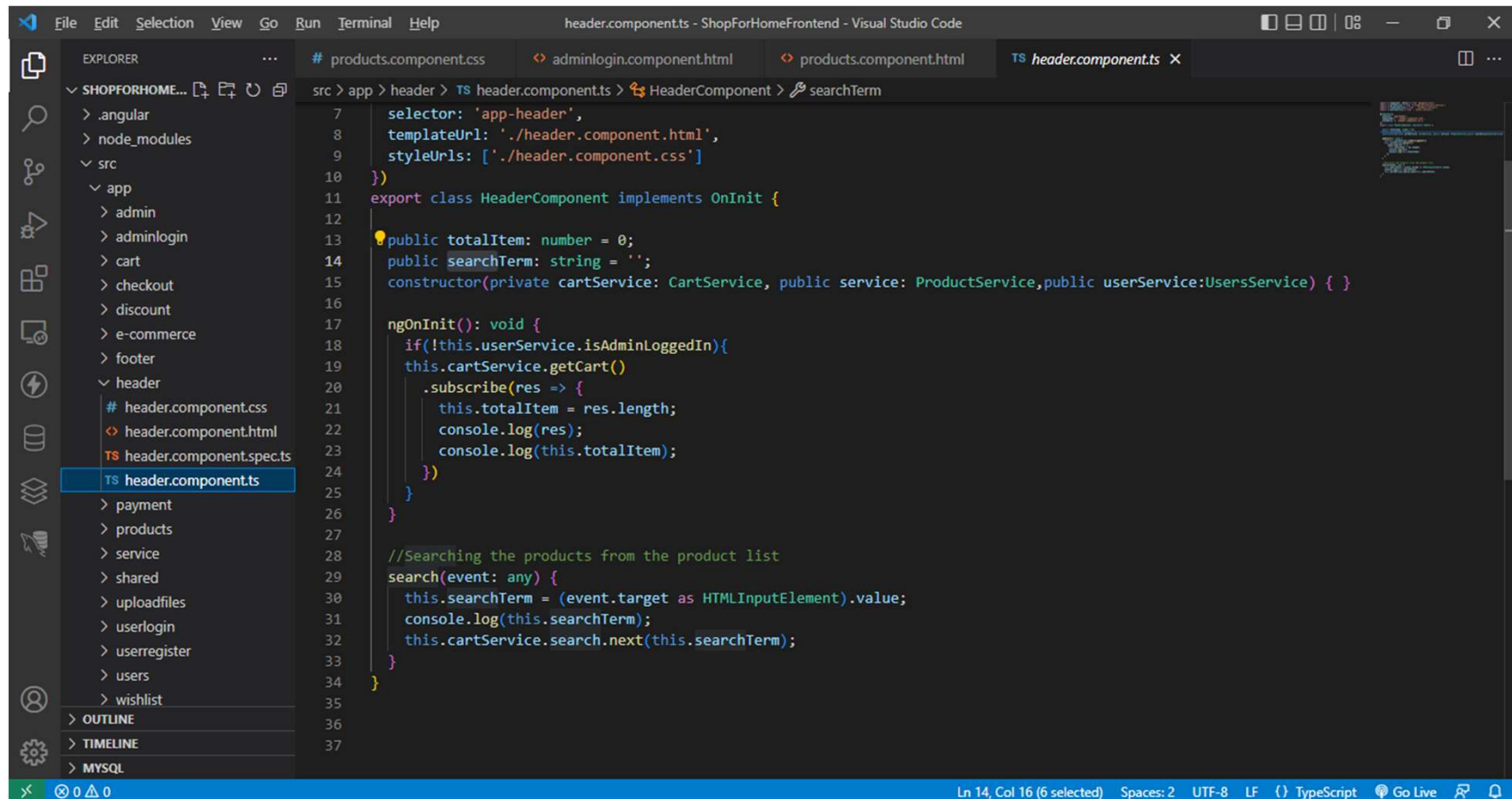


Fig- sign up &sign in pages

7. Develop Search Functionality in Angular



The screenshot displays the Visual Studio Code editor with the 'header.component.ts' file open. The Explorer sidebar on the left shows the project structure, with the 'header' folder expanded. The main editor area shows the following TypeScript code:

```
src > app > header > TS header.component.ts > HeaderComponent > searchTerm

7  selector: 'app-header',
8  templateUrl: './header.component.html',
9  styleUrls: ['./header.component.css']
10 })
11 export class HeaderComponent implements OnInit {
12
13   public totalItem: number = 0;
14   public searchTerm: string = '';
15   constructor(private cartService: CartService, public service: ProductService, public userService: UsersService) { }
16
17   ngOnInit(): void {
18     if(!this.userService.isAdminLoggedIn){
19       this.cartService.getCart()
20         .subscribe(res => {
21           this.totalItem = res.length;
22           console.log(res);
23           console.log(this.totalItem);
24         })
25     }
26   }
27
28   //Searching the products from the product list
29   search(event: any) {
30     this.searchTerm = (event.target as HTMLInputElement).value;
31     console.log(this.searchTerm);
32     this.cartService.search.next(this.searchTerm);
33   }
34 }
35
36
37
```

The status bar at the bottom indicates the cursor is at line 14, column 16, with 6 characters selected. The file encoding is UTF-8 and the line ending is LF.

Fig- search function

8. Bulk upload implementation

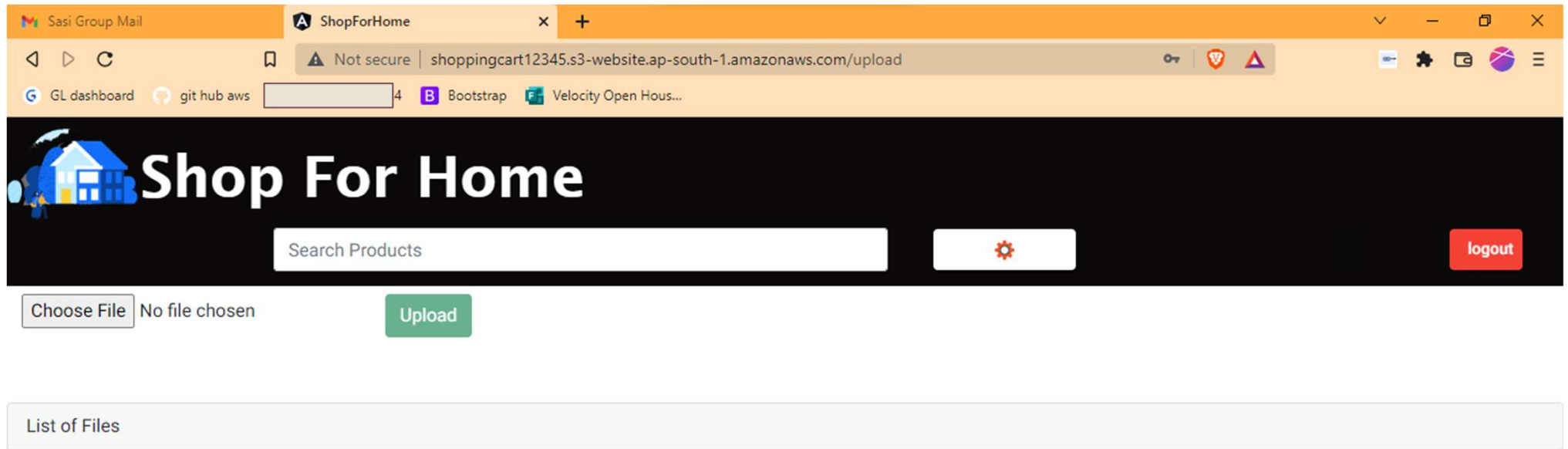



Fig- Bulkupload

Objectives Sprint-2


1. Cart option.
2. Stock and Sales reports
3. Create Data Transfer objects
4. Create repository
5. Create Service layer logic
6. Create Controller to direct rest api
7. Create Discount Microservice

1. Cart option







Shop For Home

[Wishlist](#)



[logout](#)

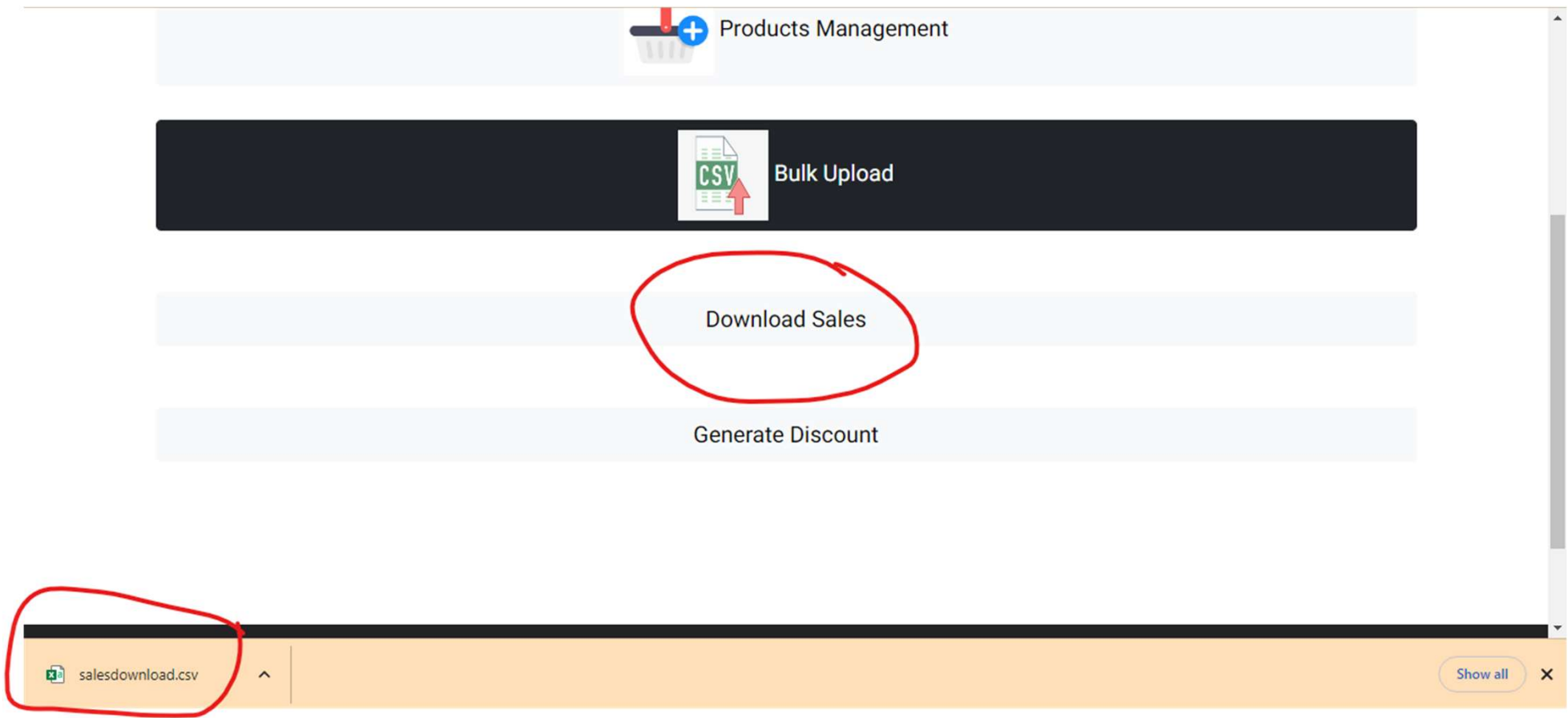
Sr.No	Product Name	Product Image	Price	Quantity	Total	Action
1	Oneplus tv		₹65,999.00	- 1 +	65999	
2	Mi 5A tv		₹45,999.00	- 1 +	45999	

[Shop More](#)

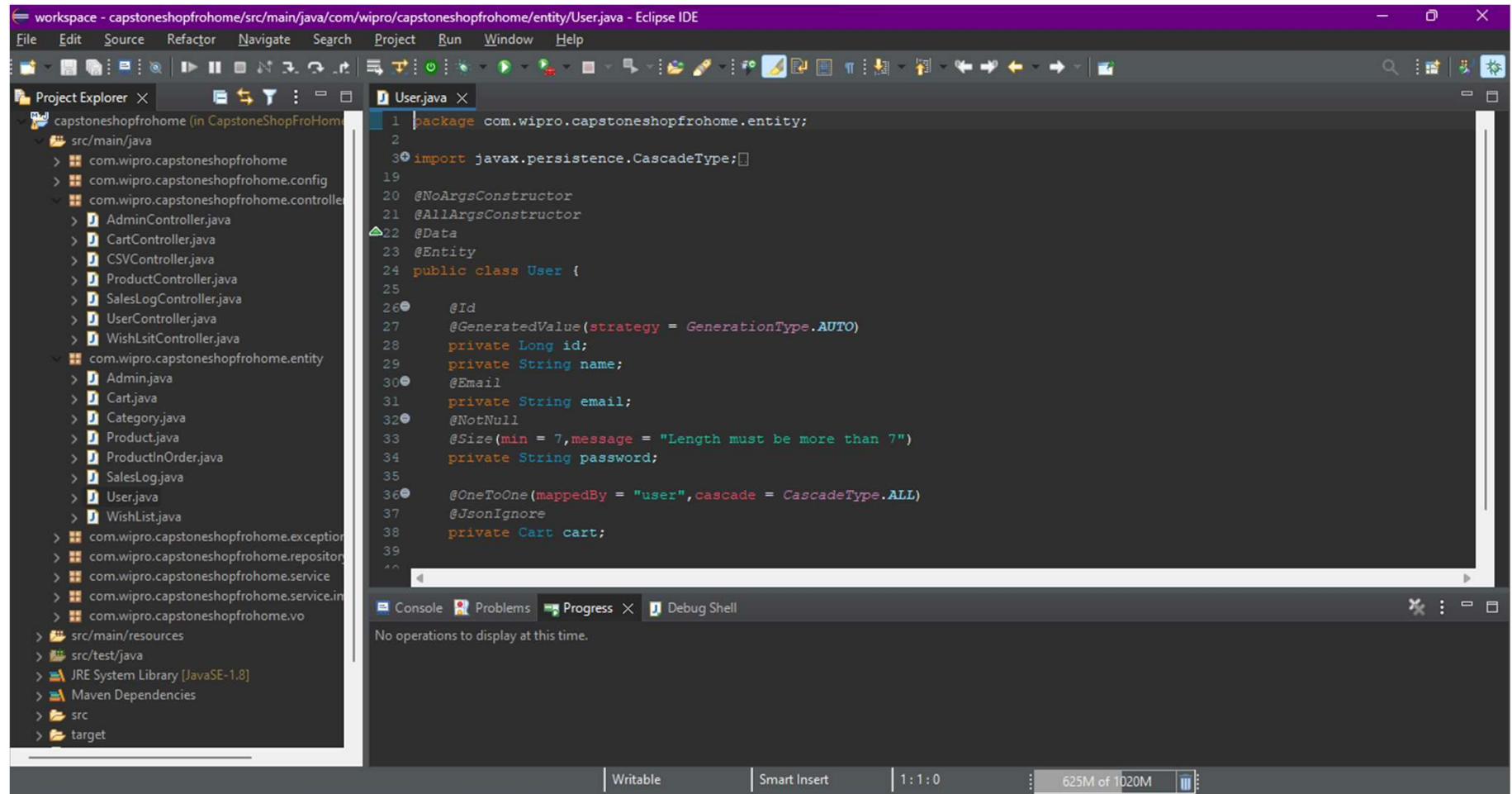
Grand Total :
₹111,998.00

[Proceed to Check Out](#)

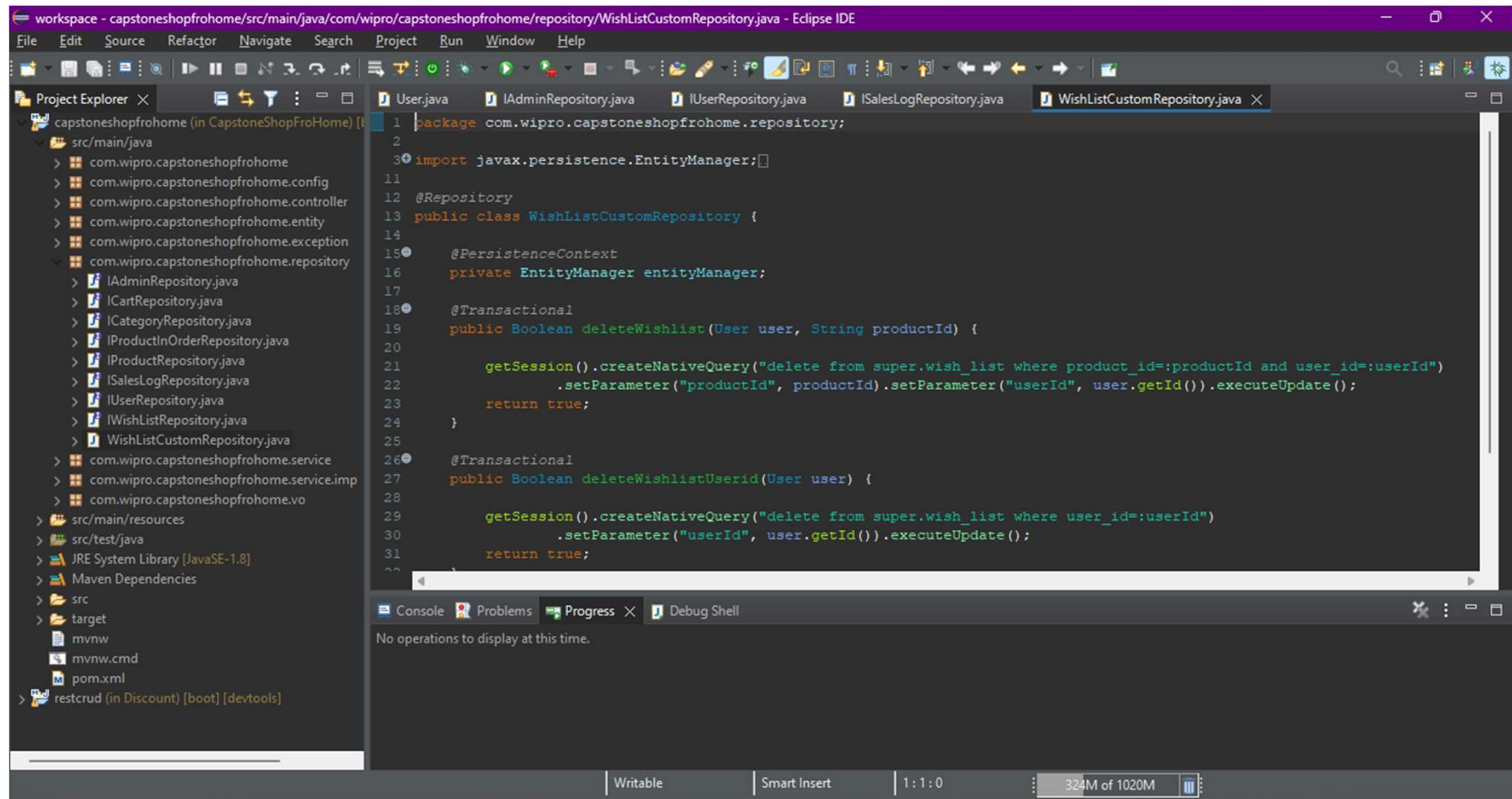
2. Stock and Sales reports



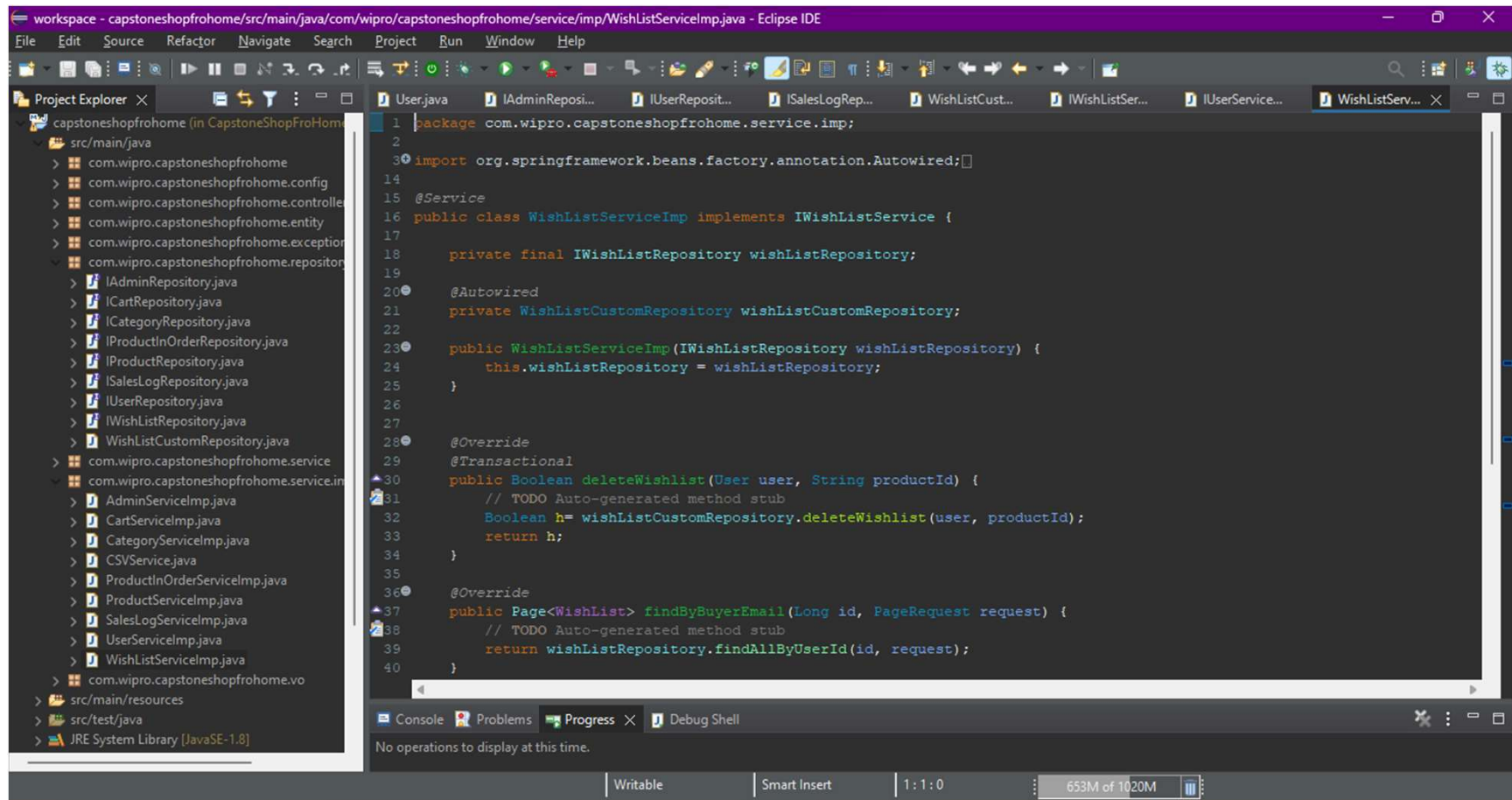
3. Create Data Transfer objects



4. Create repository



5. Create Service layer logic

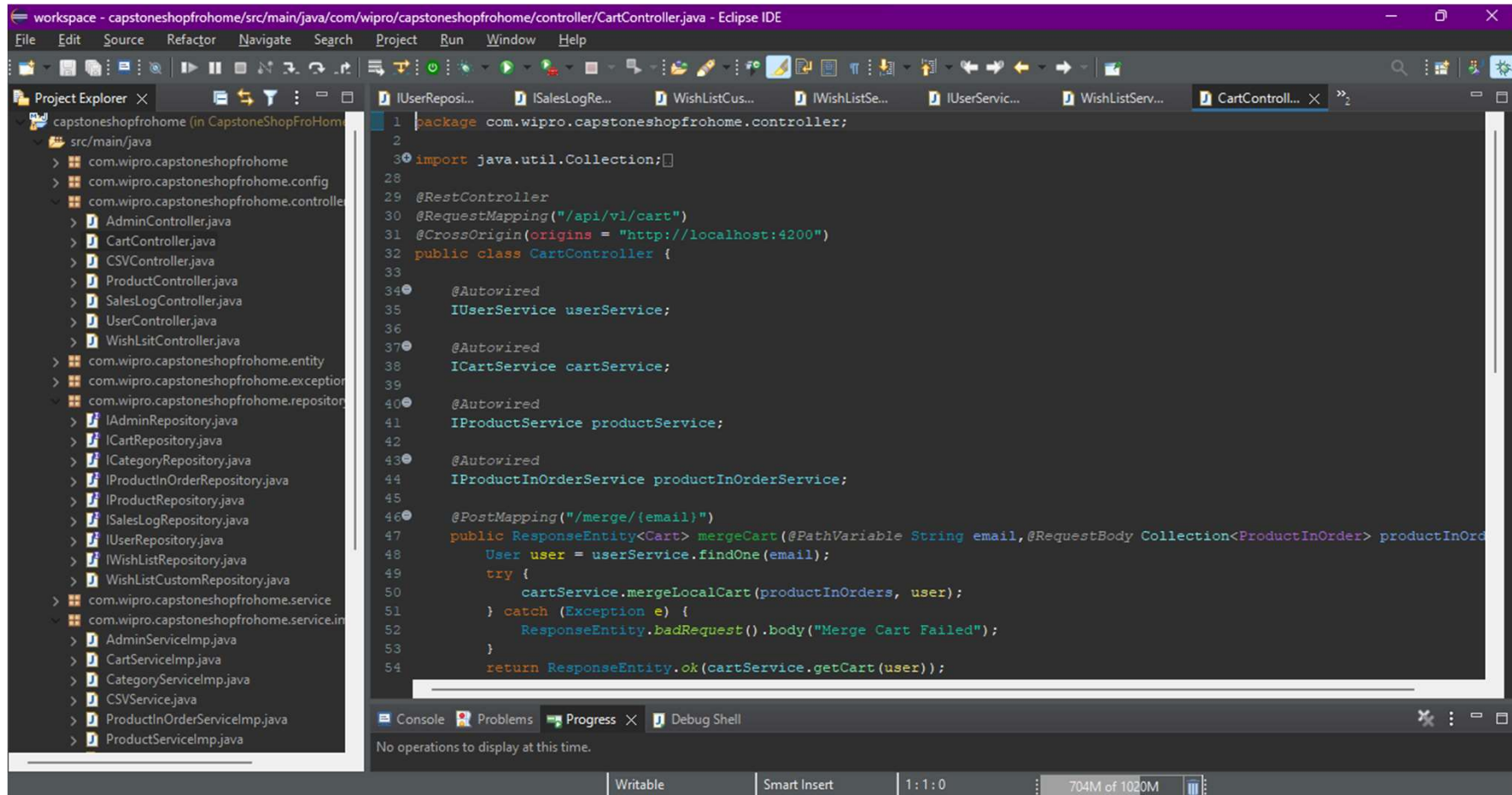


The screenshot shows the Eclipse IDE with the file `WishListServiceImp.java` open. The Project Explorer on the left shows the project structure, including the `com.wipro.capstoneshopfrohome.service` package. The main editor displays the following Java code:

```
1 package com.wipro.capstoneshopfrohome.service.imp;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
14
15 @Service
16 public class WishListServiceImp implements IWishListService {
17
18     private final IWishListRepository wishListRepository;
19
20     @Autowired
21     private WishListCustomRepository wishListCustomRepository;
22
23     public WishListServiceImp(IWishListRepository wishListRepository) {
24         this.wishListRepository = wishListRepository;
25     }
26
27
28     @Override
29     @Transactional
30     public Boolean deleteWishlist(User user, String productId) {
31         // TODO Auto-generated method stub
32         Boolean h= wishListCustomRepository.deleteWishlist(user, productId);
33         return h;
34     }
35
36     @Override
37     public Page<WishList> findByBuyerEmail(Long id, PageRequest request) {
38         // TODO Auto-generated method stub
39         return wishListRepository.findAllByUserId(id, request);
40     }
41 }
```

The bottom of the IDE shows the Console, Problems, Progress, and Debug Shell tabs. The status bar at the bottom indicates the file is writable, smart insert is enabled, and the cursor is at line 1, column 10. The memory usage is shown as 653M of 1020M.

6. Create Controller to direct rest api



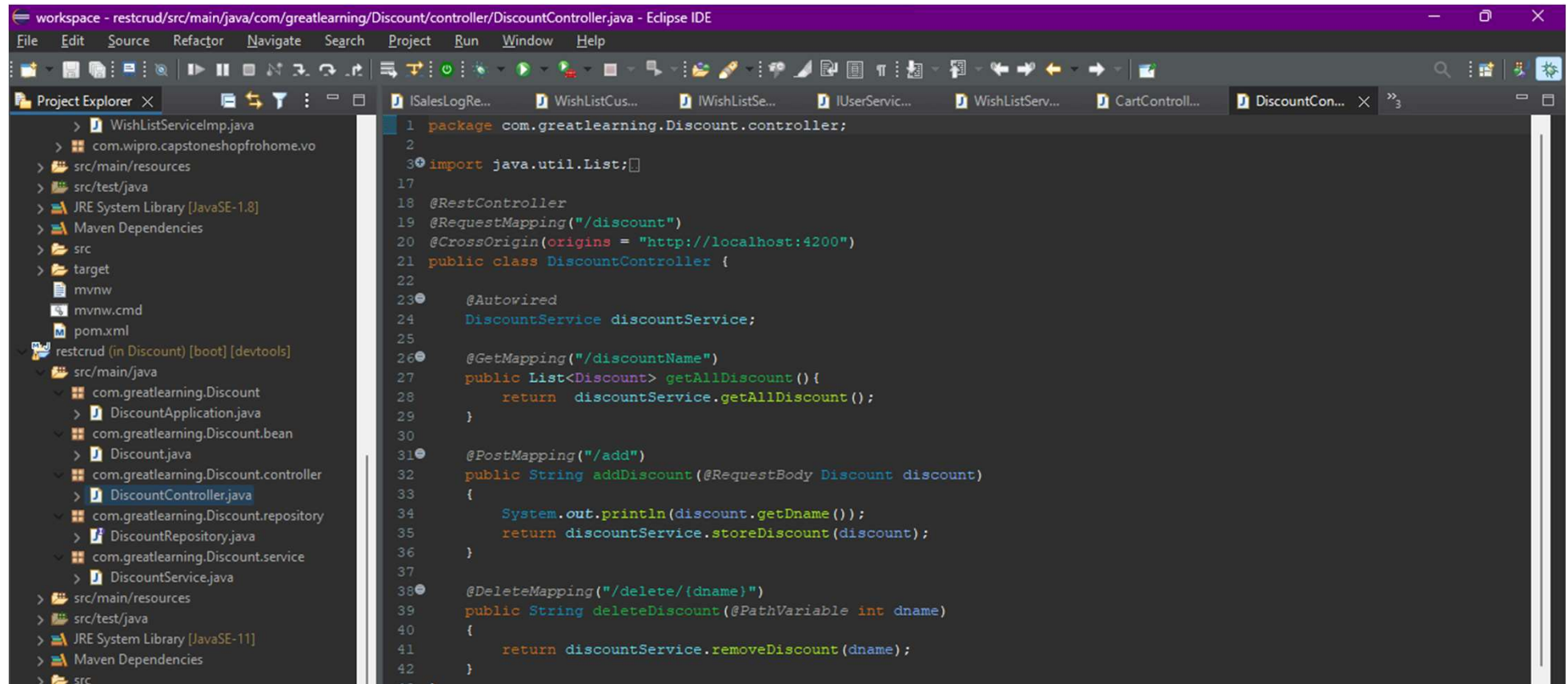
```
workspace - capstoneshopfrohome/src/main/java/com/wipro/capstoneshopfrohome/controller/CartController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer
capstoneshopfrohome (in CapstoneShopFroHome)
src/main/java
com.wipro.capstoneshopfrohome
com.wipro.capstoneshopfrohome.config
com.wipro.capstoneshopfrohome.controller
AdminController.java
CartController.java
CSVController.java
ProductController.java
SalesLogController.java
UserController.java
WishListController.java
com.wipro.capstoneshopfrohome.entity
com.wipro.capstoneshopfrohome.exception
com.wipro.capstoneshopfrohome.repository
IAdminRepository.java
ICartRepository.java
ICategoryRepository.java
IProductInOrderRepository.java
IProductRepository.java
ISalesLogRepository.java
IUserRepository.java
IWishListRepository.java
WishListCustomRepository.java
com.wipro.capstoneshopfrohome.service
com.wipro.capstoneshopfrohome.service.impl
AdminServiceImpl.java
CartServiceImpl.java
CategoryServiceImpl.java
CSVService.java
ProductInOrderServiceImpl.java
ProductServiceImpl.java

IUserReposi...
ISalesLogRe...
WishListCus...
IWishListSe...
IUserServic...
WishListServ...
CartControll...

1 package com.wipro.capstoneshopfrohome.controller;
2
3 import java.util.Collection;
4
5 @RestController
6 @RequestMapping("/api/v1/cart")
7 @CrossOrigin(origins = "http://localhost:4200")
8 public class CartController {
9
10     @Autowired
11     IUserService userService;
12
13     @Autowired
14     ICartService cartService;
15
16     @Autowired
17     IProductInOrderService productInOrderService;
18
19     @PostMapping("/merge/{email}")
20     public ResponseEntity<Cart> mergeCart(@PathVariable String email, @RequestBody Collection<ProductInOrder> productInOrder) {
21         User user = userService.findOne(email);
22         try {
23             cartService.mergeLocalCart(productInOrders, user);
24         } catch (Exception e) {
25             ResponseEntity.badRequest().body("Merge Cart Failed");
26         }
27         return ResponseEntity.ok(cartService.getCart(user));
28     }
29 }
```


7. Create Discount Microservice



```
workspace - restcrud/src/main/java/com/greatlearning/Discount/controller/DiscountController.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer
> WishlistServiceImpl.java
> com.wipro.capstoneshopfrohome.vo
> src/main/resources
> src/test/java
> JRE System Library [JavaSE-1.8]
> Maven Dependencies
> src
> target
> mvnw
> mvnw.cmd
> pom.xml
restcrud (in Discount) [boot] [devtools]
  > src/main/java
    > com.greatlearning.Discount
      > DiscountApplication.java
      > com.greatlearning.Discount.bean
        > Discount.java
        > com.greatlearning.Discount.controller
          > DiscountController.java
        > com.greatlearning.Discount.repository
          > DiscountRepository.java
        > com.greatlearning.Discount.service
          > DiscountService.java
    > src/main/resources
    > src/test/java
    > JRE System Library [JavaSE-11]
    > Maven Dependencies
    > src

1 package com.greatlearning.Discount.controller;
2
3 import java.util.List;
17
18 @RestController
19 @RequestMapping("/discount")
20 @CrossOrigin(origins = "http://localhost:4200")
21 public class DiscountController {
22
23     @Autowired
24     DiscountService discountService;
25
26     @GetMapping("/discountName")
27     public List<Discount> getAllDiscount() {
28         return discountService.getAllDiscount();
29     }
30
31     @PostMapping("/add")
32     public String addDiscount(@RequestBody Discount discount)
33     {
34         System.out.println(discount.getDname());
35         return discountService.storeDiscount(discount);
36     }
37
38     @DeleteMapping("/delete/{dname}")
39     public String deleteDiscount(@PathVariable int dname)
40     {
41         return discountService.removeDiscount(dname);
42     }
43 }
```

Fig- search funaction

Aws deployment:

Sign in

☒ **Root user**

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**

User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

— New to AWS? —

Create a new AWS account

Reinvent your business with data

Leverage data insights to drive better, faster outcomes

[LEARN MORE](#)



Creating RDS

i Ensure that general, slow query, and audit logs are turned on. Error logs are enabled by default. [Learn more](#)

Maintenance

Auto minor version upgrade [Info](#)

☒ **Enable auto minor version upgrade**
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

☐ Choose a window

☒ No preference

Deletion protection

☐ **Enable deletion protection**
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page.](#)

i You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel **Create database**

Fig- RDS creation

Creating S3 bucket

every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from accidental deletions and application failures. [Learn more](#)

Bucket Versioning

☐ Disable

☒ Enable

Tags (0) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

☒ Disable

☐ Enable

Advanced settings

After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

Fig- S3 bucket creation

Creating elastic beanstalk environment

The screenshot displays the 'Create environment' wizard in the AWS Elastic Beanstalk console. It is divided into two main sections: 'Platform' and 'Application code'.

Platform Section:

- Managed platform:** Selected by default. Description: 'Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)'. A blue border highlights this option.
- Custom platform:** Description: 'Platforms created and owned by you.' (disabled).
- Platform:** A dropdown menu with 'Java' selected.
- Platform branch:** A dropdown menu with 'Corretto 17 running on 64bit Amazon Linux 2' selected.
- Platform version:** A dropdown menu with '3.3.1 (Recommended)' selected.

Application code Section:

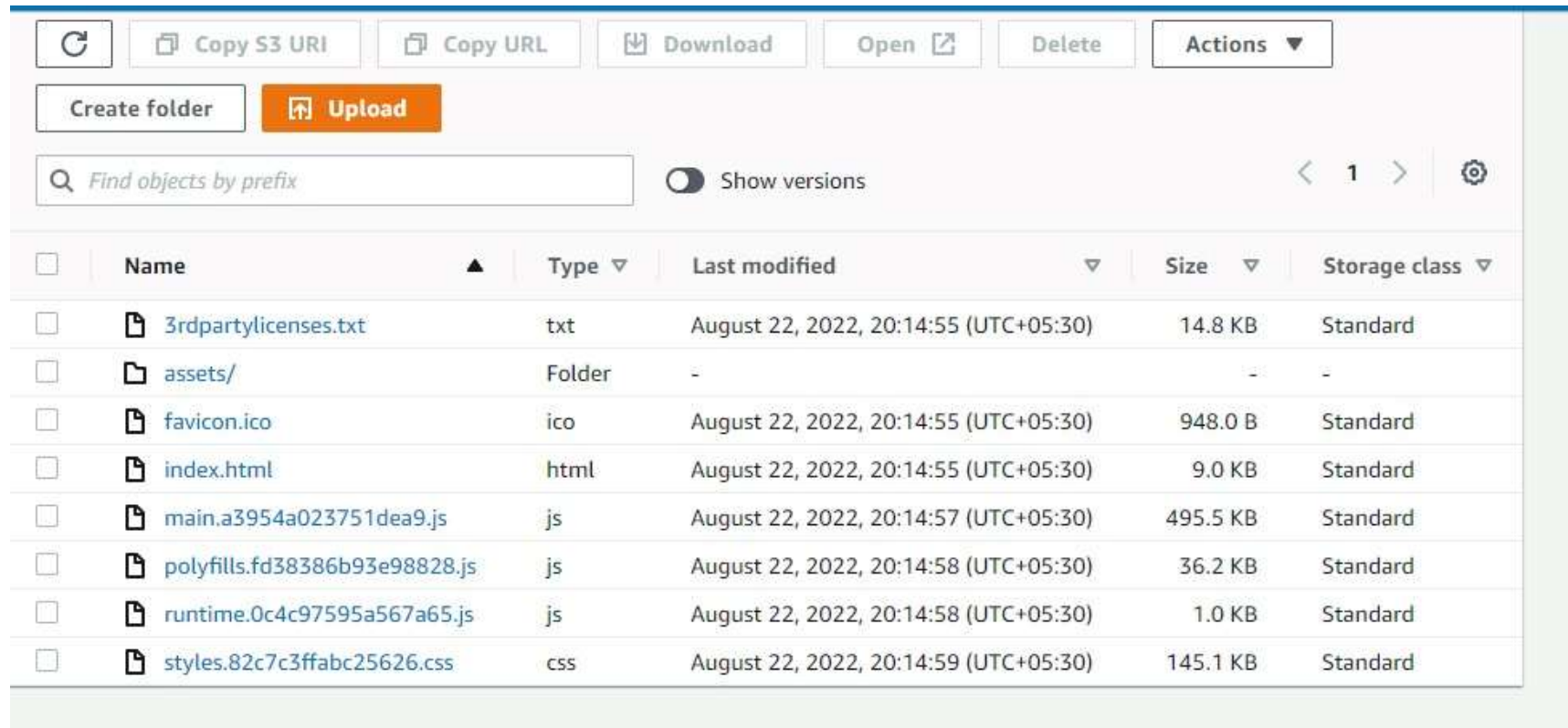
- Sample application:** Selected by default. Description: 'Get started right away with sample code.'.
- Existing version:** Description: 'Application versions that you have uploaded for shoppingcart.' Below it is a dropdown menu with the text '-- Choose a version --'.
- Upload your code:** Description: 'Upload a source bundle from your computer or copy one from Amazon S3.' (disabled).

Bottom Buttons:

- Cancel:** A light gray button.
- Configure more options:** A white button with a black border.
- Create environment:** An orange button.

Fig- creating environment

Uploading data in s3 bucket



The screenshot displays the AWS S3 console interface for a specific bucket. At the top, there is a toolbar with buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', and 'Actions'. Below this, there are buttons for 'Create folder' and 'Upload'. A search bar with the placeholder 'Find objects by prefix' and a 'Show versions' toggle are also present. The main area contains a table listing the objects in the bucket.









<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	 3rdpartylicenses.txt	txt	August 22, 2022, 20:14:55 (UTC+05:30)	14.8 KB	Standard
<input type="checkbox"/>	 assets/	Folder	-	-	-
<input type="checkbox"/>	 favicon.ico	ico	August 22, 2022, 20:14:55 (UTC+05:30)	948.0 B	Standard
<input type="checkbox"/>	 index.html	html	August 22, 2022, 20:14:55 (UTC+05:30)	9.0 KB	Standard
<input type="checkbox"/>	 main.a3954a023751dea9.js	js	August 22, 2022, 20:14:57 (UTC+05:30)	495.5 KB	Standard
<input type="checkbox"/>	 polyfills.fd38386b93e98828.js	js	August 22, 2022, 20:14:58 (UTC+05:30)	36.2 KB	Standard
<input type="checkbox"/>	 runtime.0c4c97595a567a65.js	js	August 22, 2022, 20:14:58 (UTC+05:30)	1.0 KB	Standard
<input type="checkbox"/>	 styles.82c7c3ffabc25626.css	css	August 22, 2022, 20:14:59 (UTC+05:30)	145.1 KB	Standard

Fig- data uploading in s3 bucket

Uploading data in environment

Elastic Beanstalk > Environments

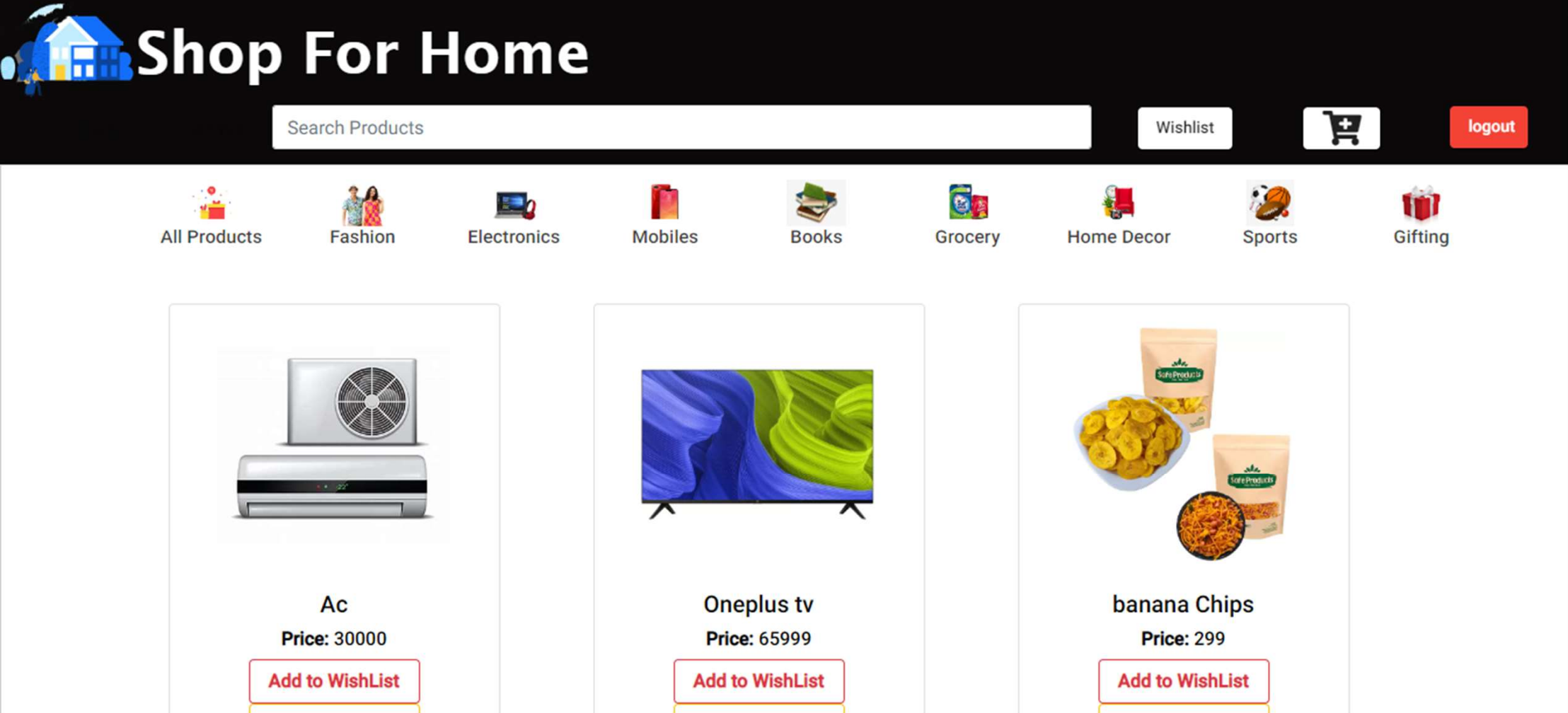
All environments Actions Create a new environment

Filter results matching the display values

Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
<input type="radio"/> ShoppingCart123-env	Severe	ShoppingCart123	2022-08-22 20:03:23 UTC+0530	2022-08-22 20:11:31 UTC+0530	main.ap-south-1.elasticbeanstalk.com	Sample Application-12	Corretto 17 running on 64bit Amazon Linux 2	Supported	WebServer

Fig- uploading data in environment


Final aws uploaded website



G13-AWS SD APR CIII

Fig- home page

Final aws uploaded website



User Login

Enter email here *


Email

Enter password here *

Password

Login

[New User? Click to SignUp!!](#)



User Registration


Enter name here *

Enter email here *

Enter Password here *

SignUp

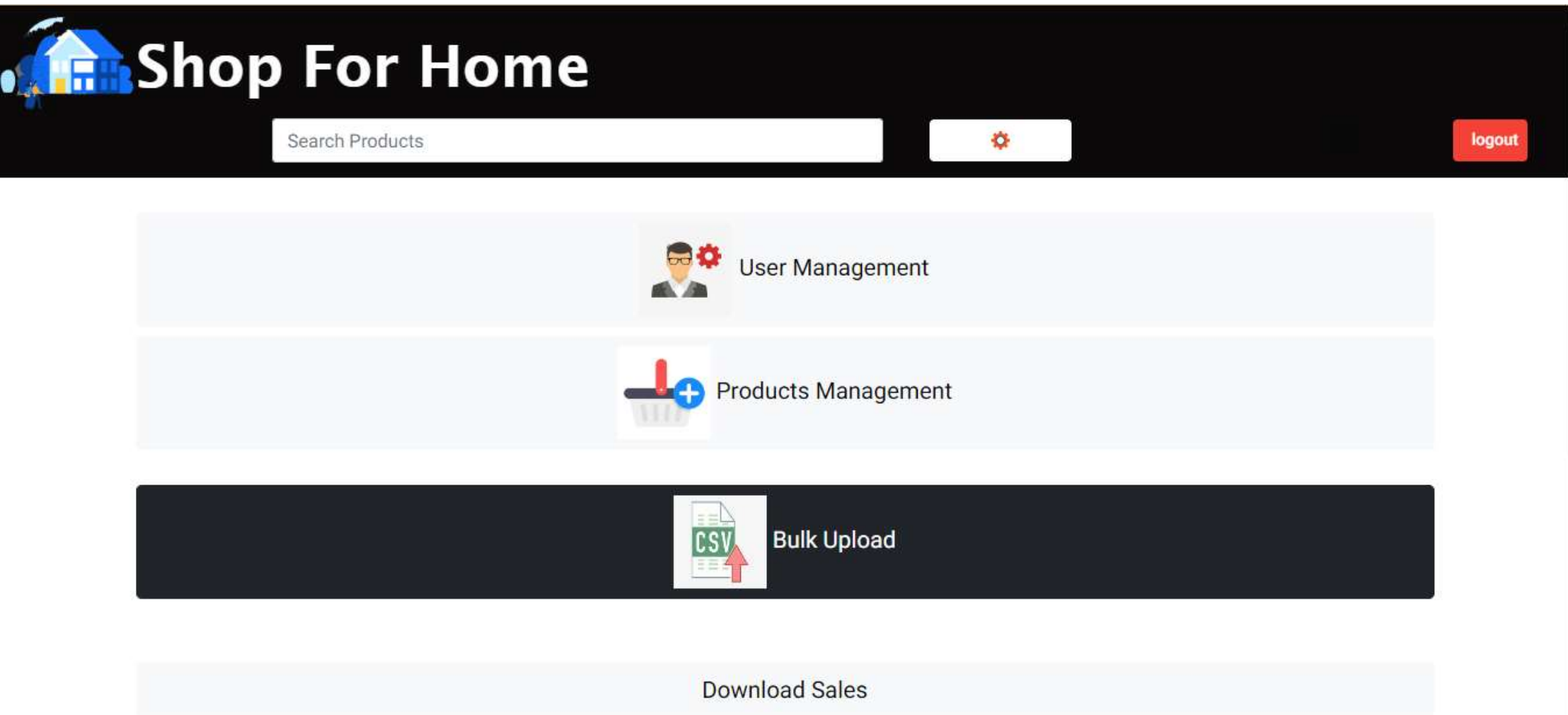
[Already Registered? Click to SignIn!!](#)



Shop For Home










Admin Login

Final aws uploaded website




Shop For Home

[Wishlist](#)[logout](#)


Sl.No	Product Name	Product Image	Price	Action
1	Eye Tshirt		₹700.00	
2	Fano shirt		₹500.00	
3	Icool Skull Cap		₹350.00	
4	Oneplus tv		₹65,999.00	
5	₹30,000.00	

Final aws uploaded website







Shop For Home

[Wishlist](#)



[logout](#)

Sr.No	Product Name	Product Image	Price	Quantity	Total	Action
1	Mi 5A tv		₹45,999.00	- 1 +	45999	
2	Oneplus tv		₹65,999.00	- 1 +	65999	

[Shop More](#)

Grand Total : ₹111,998.00

[Proceed to Check Out](#)

THANK YOU