NANYANG TECHNOLOGICAL UNIVERSITY

IN6225
Enterprise Application Development

Individual Assignment Report:
Full Stack E-commerce Web Application Development

Written by:

Ngoh Jie Rui (G2203664C)

Table of Contents

1. Introduction

As of 2022, there are over five billion internet users worldwide, which corresponded with an increased amount of people making purchases online. In 2022, retail e-commerce sales exceed 5.7 trillion U.S. dollars worldwide and are projected to increase in the coming years. (Statista , 2023) Rise of online shopping has made it imperative for businesses to develop ecommerce web applications in order to remain competitive. This has also led to an increasing demand for ecommerce web application development which offer convenience and seamless shopping experience for the customers. This paper will discuss the development of an ecommerce web application, JRPotter, developed using Angular, Spring Boot and mySQL.

1.1 Product Description

1.1.1 Purpose

JRPotter is a web application developed to provide a one-stop solution for customers to purchase tour packages online at discounted prices. JRPotter web application aims to provide an interactive,seamless, and secured shopping application which allows users to view, search and purchase the variety of products sold.

1.1.2 Scope

JRPotter is a web application that enables system administrators to manage the list of products. In addition, system administrators are able to track the orders placed, confirmed and delivered. JRPotter also includes functionalities that allow customers to view, search and place their orders online.

1.1.3 Users and Stakeholders

The users of the JRPotter web application are consumers (users) and the system administrator (admin).

1. Admin : The admin is responsible for maintaining the operations of the web application. Functions performed by admin includes:-
    - Updating the list of products sold and deleting products that are no longer on sales
    - Viewing and updating the status of purchases made by users

2. Users: The users are customers who visited the web application. The users should be able to perform the following functions:
    - View, add, and update items in shopping cart
    - Sign up, log in, log out using username and password
    - Track the order placed

1.1.4 Assumptions and constraints

The application is built using the following programming languages and technologies:-

- Java
- Javascript
- HTML
- CSS
- TypeScript
- Spring Boot v 2.4.5
- Angular v 10.1.4
- MySQL v 8

Assumptions of JRPotter includes:-

- Separation of concerns: Front end development is developed using Angular, which separates the presentation layer from application logic. Back end development is developed using Spring Boot, where application logic is distinct from data layer and presentation layer.
- RESTful API: Backend of Spring Boot will expose RESTful web API for communication with the frontend. Thus, enables loose coupling between the front and back ends, making the application more modular and scalable.

Constraints of software development of JRPotter includes:-

- Programming languages: As JRPotter is developed using Spring Boot and Angular. Developers shall have a good understanding of Java and Typescript for the Continuous Improvement (CI) and Continuous Development(CD) of JRPotter.
- Development environments: Node.js version 14.20 and npm version 9.5 are required to be installed for Angular project and Maven will need to be in built for Spring Boot projects
- Learning curve: Mastering Angular and Spring Boots come with a steep learning curve. I.e. a significant amount of time is required to master the technologies and their associated tools and frameworks.

1.2 Software Development Lifecycle Model (SDLC)

The agile software development life cycle (SDLC) model was adopted for this project.

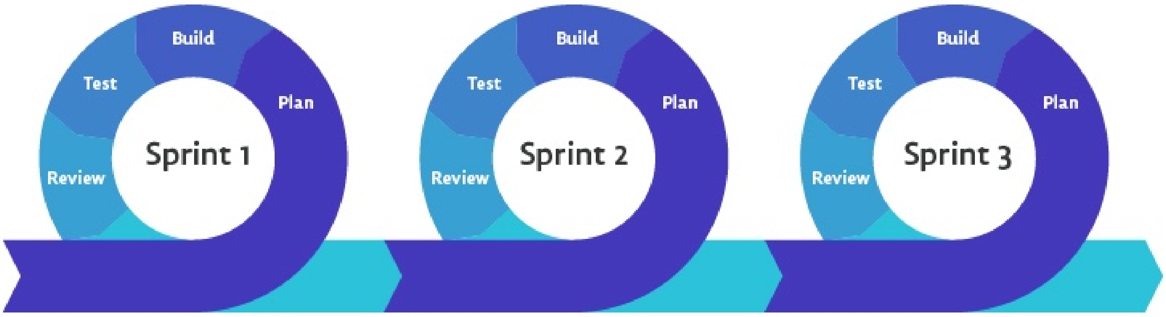


Figure 1: Illustration of agile model (Bit Bean, n.d.)

Agile SDLC was selected as it allowed features to be developed, tested and verified incrementally and stacked upon completion of each feature as follow:-

- Iteration 1 - Back end application and database development i.e. Data Access Object (DAO), REST APIs of products , JSON Web Token (JWT) server authentication for admin and users
- Iteration 2 - Front and back end development - Display list of products sold
- Iteration 3- Front and back end development -Create, Remove, Update, Delete (CRUD) features for admin
- Iteration 4- Front and back end development -Create, Remove, Update, Delete (CRUD) features for user
- Iteration 5 - Front and back end development- User Interface (UI)/User Experience (UX) development i.e. Header, Footer and Carousel

2. Functional Requirements

This section highlight the functional requirements for the different components of JRPotter:-

1. Home Page
   1.1 The system must be able to display the list of products.
   1.2 The system must allow users to use the search function.
   1.3 The system must allow users to register their accounts.
   1.4 The system must allow users to login to their accounts.
   1.5 The user must be able to select and retrieve more details of the products.

2. Search
   2.1 The user must be able to search and filter the product based on products' names and/or descriptions.

3. Registration
   3.1 The user must be able to register for a new account via the system registration system.
      3.1.1 Information include:-
         3.1.1.1 Username
         3.1.1.2 First Name
         3.1.1.3 Last Name
         3.1.1.4 Password

4. Login/ logout
   4.1 The user must be able to login using the username and password.
   4.2 The system must be able to redirect users to the application's home page.
   4.3 The user must be able to logout from the application.

5. Shopping Cart
   5.1 The user must be able to add products into the virtual shopping cart.
   5.2 The user must be able to view the products that are added into the virtual shopping cart.
   5.3 The user must be able to delete the products in the shopping cart.

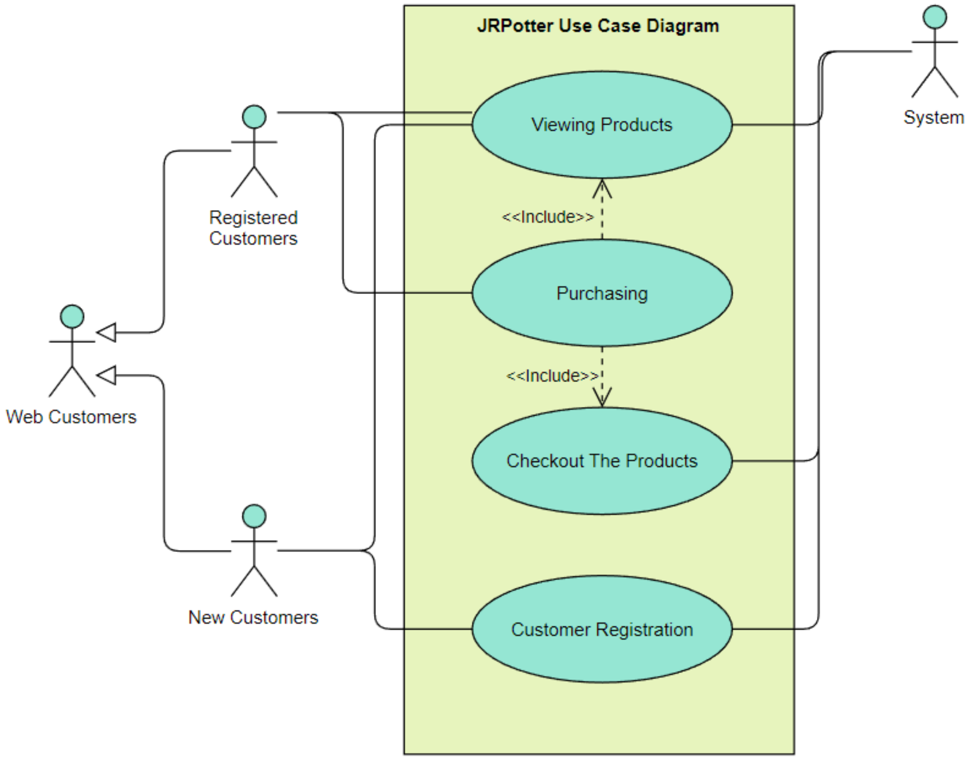
6. Order
   6.1 The user must be able to place an order by completing the online order form.
   6.2 The system must be able to view/ process the orders made by users.

7. Product Catalog

7.1 The system must be able to add/update/delete list of products sold.

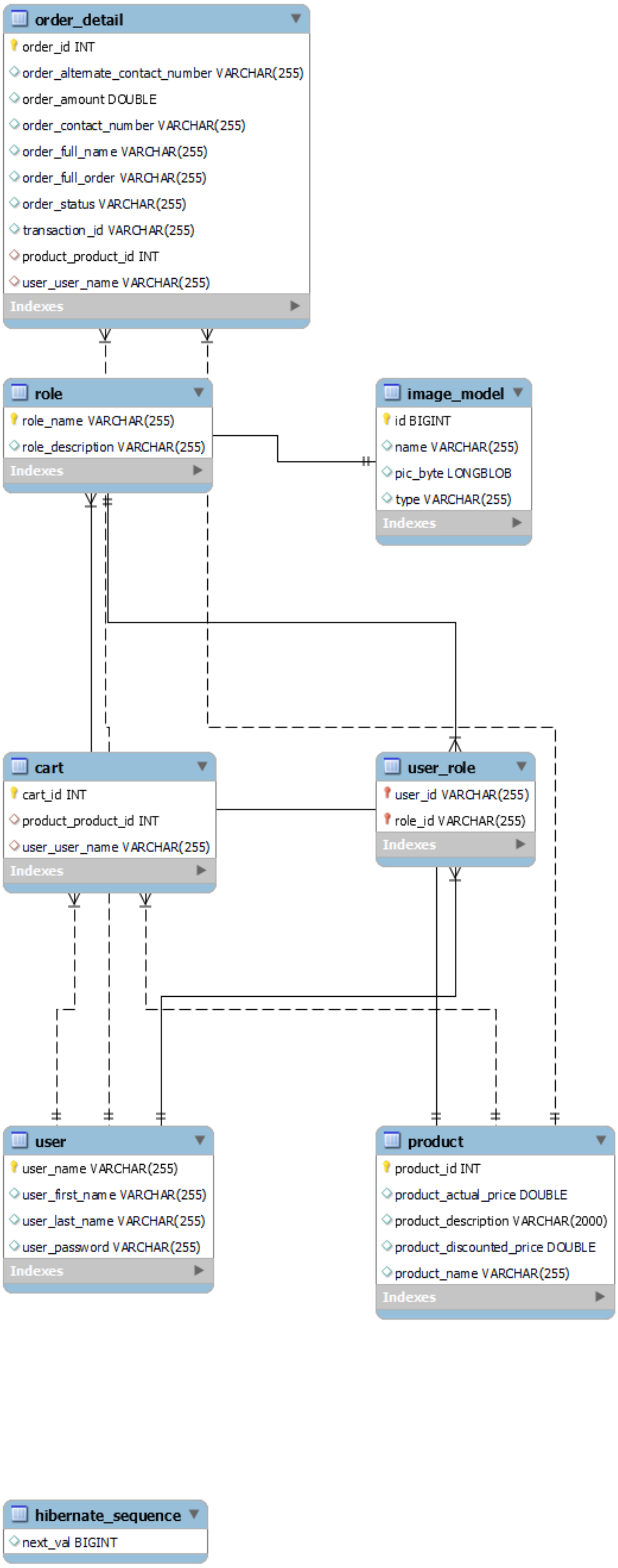
2.1 User Case Diagram

The below diagram illustrates the user case diagram for JRPotter:-



2.2 Entity Relationship (ER) Diagram

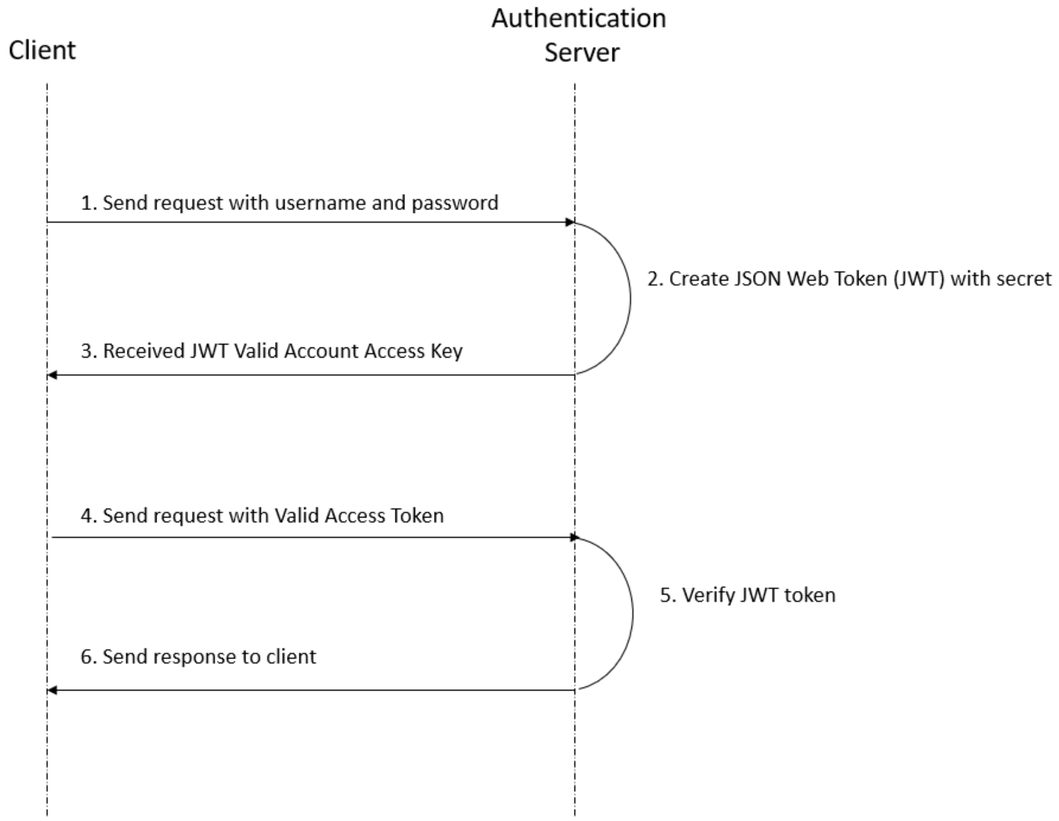
The below diagram illustrates the ER diagram of the database of JRPotter:-



order_detail
order_id INT
order_alternate_contact_number VARCHAR(255)
order_amount DOUBLE
order_contact_number VARCHAR(255)
order_full_name VARCHAR(255)
order_full_order VARCHAR(255)
order_status VARCHAR(255)
transaction_id VARCHAR(255)
product_product_id INT
user_user_name VARCHAR(255)
Indexes
role
role_name VARCHAR(255)
role_description VARCHAR(255)
Indexes
image_model
id BIGINT
name VARCHAR(255)
pic_byte LONGBLOB
type VARCHAR(255)
Indexes
cart
cart_id INT
product_product_id INT
user_user_name VARCHAR(255)
Indexes
user_role
user_id VARCHAR(255)
role_id VARCHAR(255)
Indexes
user
user_name VARCHAR(255)
user_first_name VARCHAR(255)
user_last_name VARCHAR(255)
user_password VARCHAR(255)
Indexes
product
product_id INT
product_actual_price DOUBLE
product_description VARCHAR(2000)
product_discounted_price DOUBLE
product_name VARCHAR(255)
Indexes
hibernate_sequence
next_val BIGINT

2.3 JSON Web Token (JWT) Server Authentication Sequence Diagram

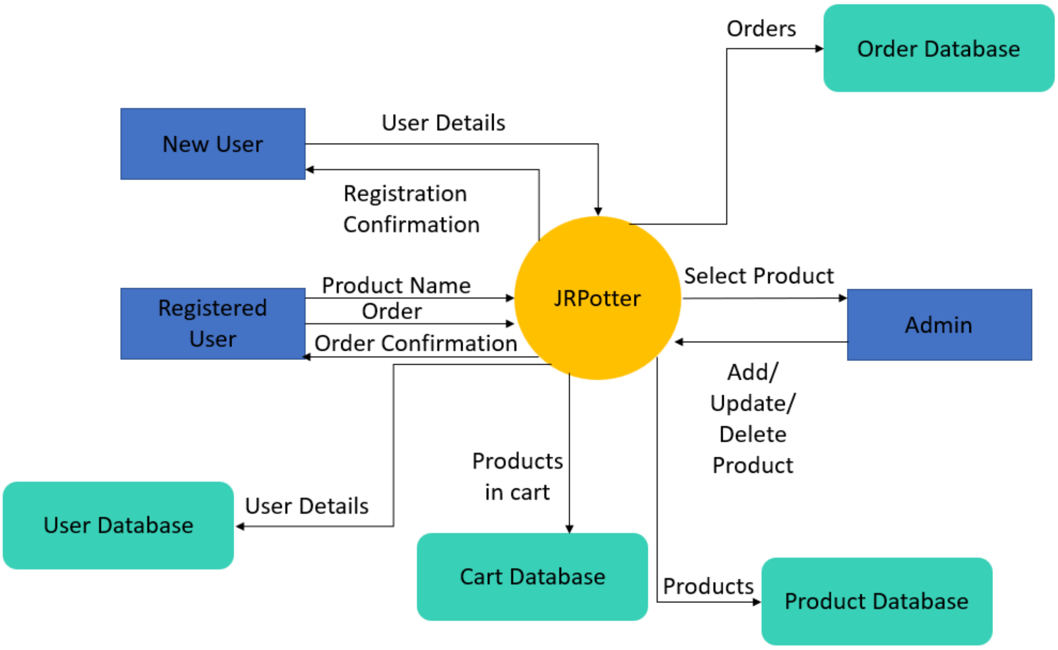
The below diagram illustrates the JWT server authentication sequence diagram for admin and users:-


Client
Authentication Server
1. Send request with username and password
2. Create JSON Web Token (JWT) with secret
3. Received JWT Valid Account Access Key
4. Send request with Valid Access Token
5. Verify JWT token
6. Send response to client

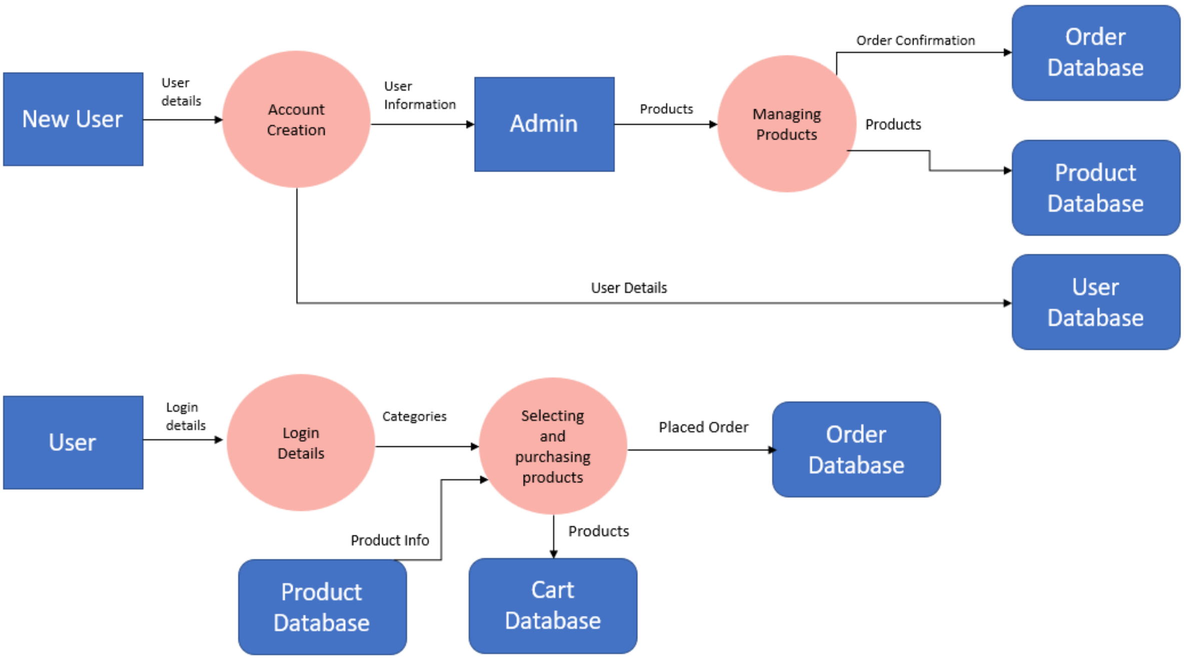
2.4 Data Flow Diagram (DFD)

The below diagrams illustrates the DFD of JRPotter:-

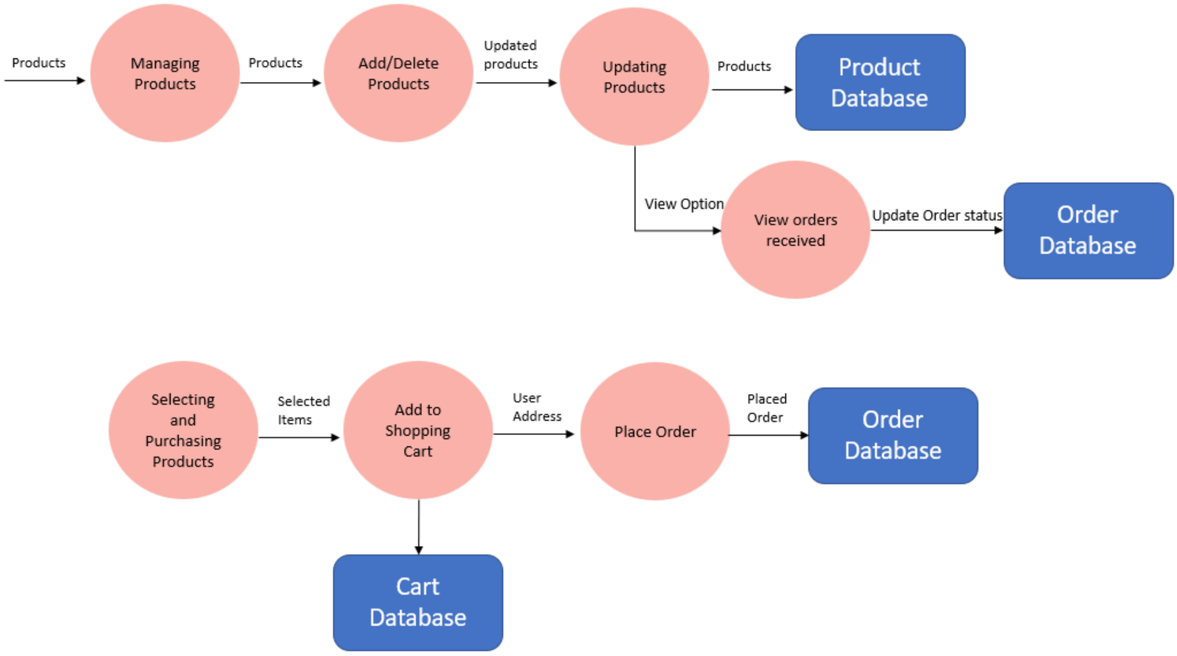
0-Level DFD



1-Level DFD



2-Level DFD



3. Non-functional Requirements

3.1 Usability Requirements

I. The system must be easy to navigate, with a user-friendly interface that is intuitive and simple to understand.
   A. User Interface to be simple and allows users to retrieve their list of products easily.

II. The system must allow easy reversal of actions.
   A. The user must be able to add and remove their selected products.

III. The system must provide informative feedback.
   A. To display error messages when certain requests fail.
   B. To provide necessary feedback to the user when invalid inputs are detected.

IV. The system must strive for consistency.
   A. A consistent actions is required for similar situations
   B. A consistent visual layout must be adopted in the application (e.g. labels, fonts, and color).

3.2 Performance Requirements

I. The system must be able to handle a high volume of traffic not crash when the user opens the application.

II. The system must be able to load quickly and return the display results to the user within 3 seconds.

III. The system must be able to process requests efficiently and maintain with little or no downtime occurring.

3.3 Security Requirements

I. The system must be secured, with user authentication and authorization, secured communication channels, and protection against common security threats.

II. The system will implement Secure Hash Algorithm (SHA) to perform salt-hashing on all the passwords before it is stored in the database.

3.4 Extensibility Requirements

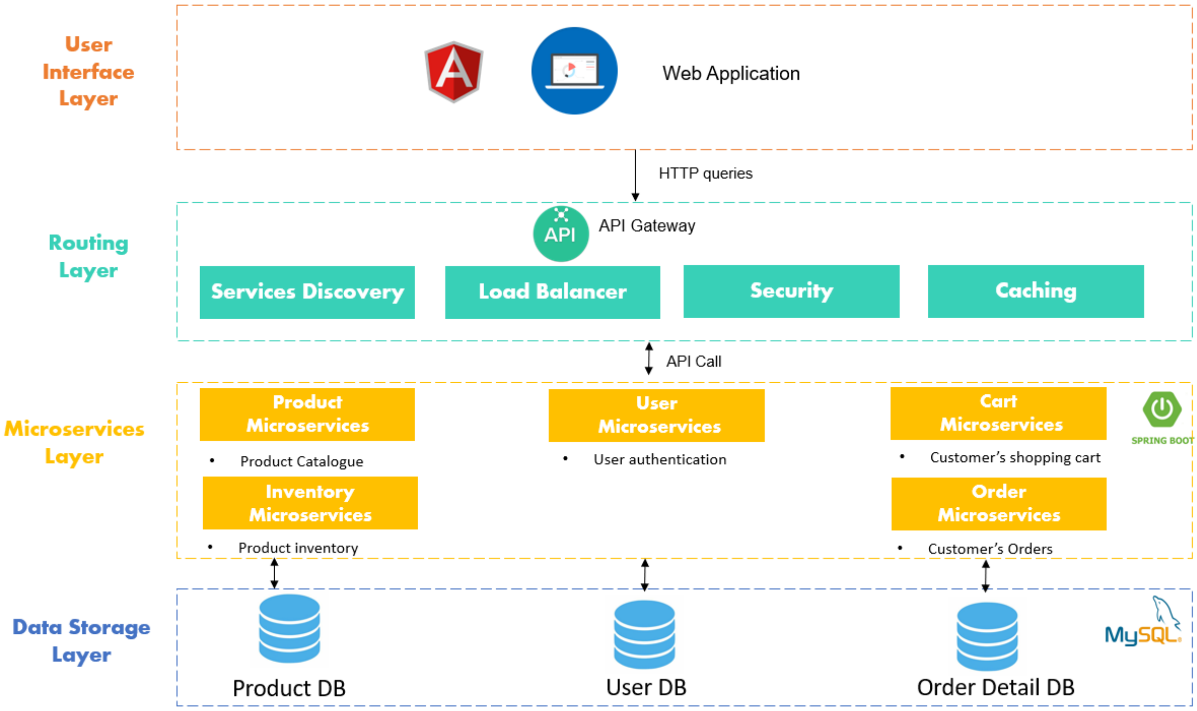
I. The system must be designed with Backend by Spring Boot and Front End by Angular to support future enhancements and promote extensibility.

II. To facilitate data storage and analysis, data collected are synced with mySQL

4. Architecture Design

4.1 System Architecture Diagram

The below diagram illustrates the system architecture diagram of JRPotter:-



Microservices and MVC architecture and design pattern was adopted for the development of JRPotter with the following features:-

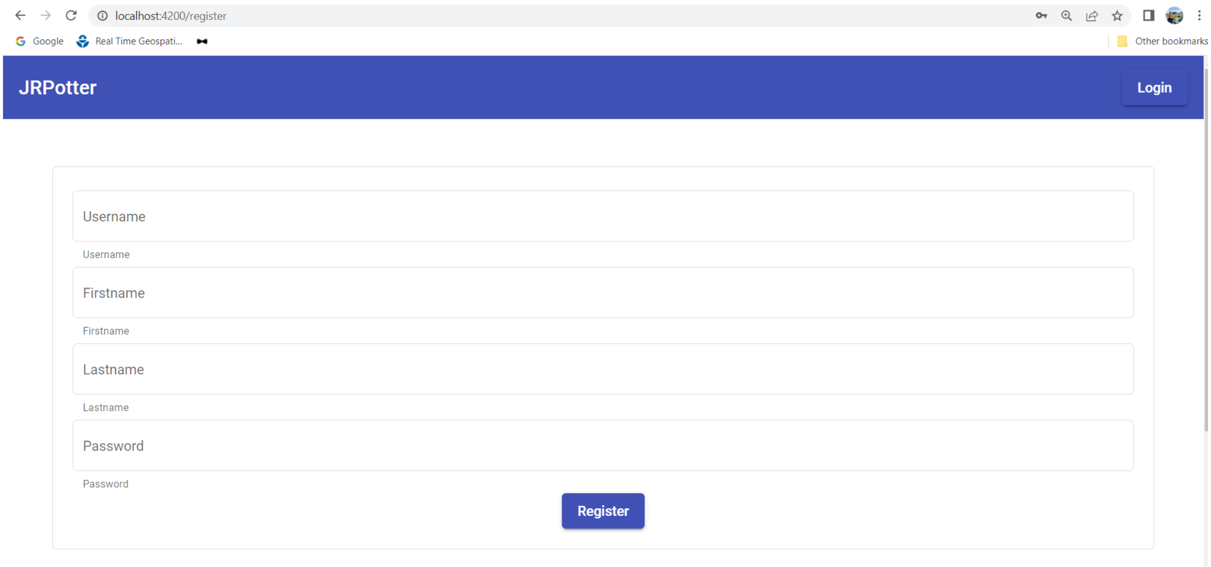
- User Interface (UI) layer consist of the front end layer by Angular to access the different microservices
- Routing layer consists of routing the different HTTPS queries to the corresponding microservices via REST API generated by backend application, Spring Boot, including:-
  - ➢ Service discovery which dynamically relate microservices instances
  - ➢ Load balancer which distributes REST API among microservices
  - ➢ Caching which stores and return static data to be load web pages faster
  - ➢ Security which safeguard microservices from cyber threats
- Microservices layer consist of microservices as follows:

  - ➢ Product-service: REST API of catalog information of products.
  - ➢ Inventory -service: REST API of product inventory.
  - ➢ Cart-services: REST API of customer's shopping cart
  - ➢ Order-services: REST API of customer's orders
  - ➢ User-service: REST API of users information
- Data storage layer consists of the stored databases for the respective microservices in mySQL

5. User Guide

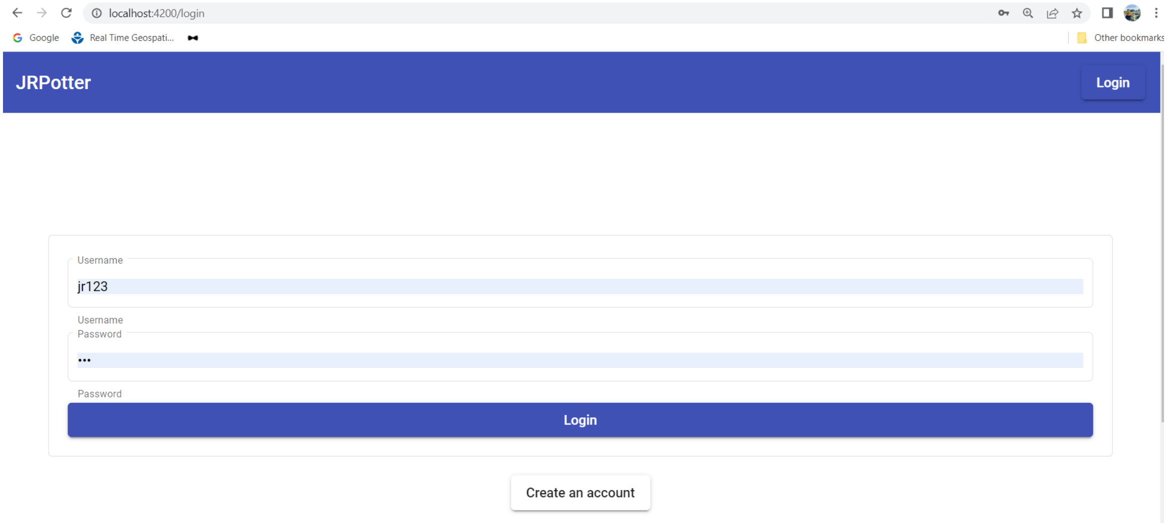
This section details the user guide for JRpotter developed using Angular and Spring Boot which enable users to navigate and use the features of JRPotter.

Registration



1. To enjoy the features of JRPotter, new users will need to register for an account. New users can fill in their particulars, i.e. username, first name, last name and password and submit the registration form.

Login



Username

jr123

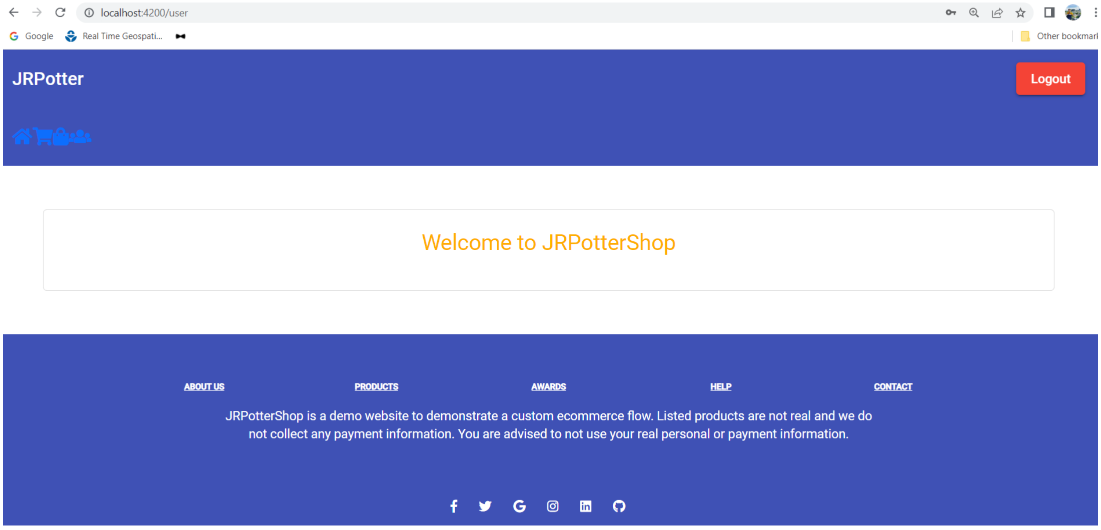
Username

Password

•••

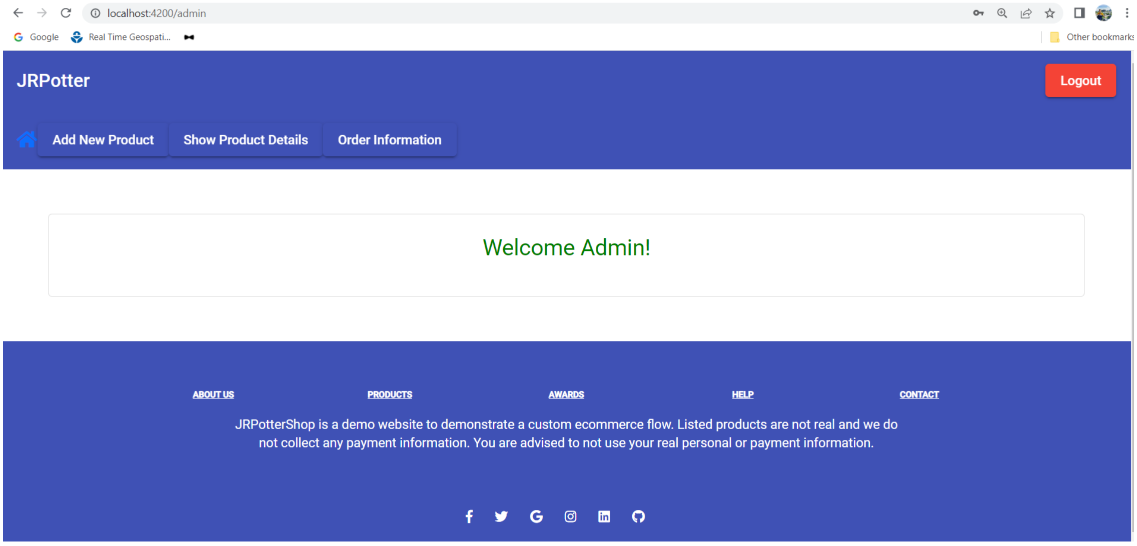
Password

2. After registering an account, users can log in by clicking on the “Login” button by entering their registered user name and password. Upon successful login by the user, the user will be able to view the user tab.



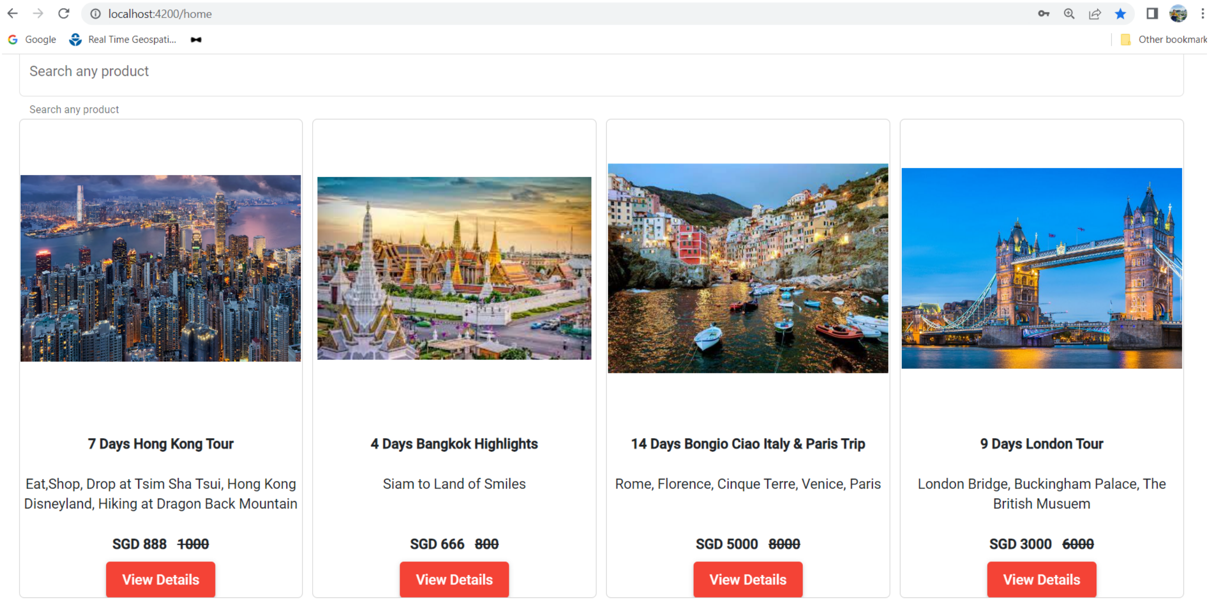
JRPotterShop is a demo website to demonstrate a custom ecommerce flow. Listed products are not real and we do not collect any payment information. You are advised to not use your real personal or payment information.

3. Upon successful login by the admin, the admin will be able to view the admin tab.



JRPotterShop is a demo website to demonstrate a custom ecommerce flow. Listed products are not real and we do not collect any payment information. You are advised to not use your real personal or payment information.

Homepage



Search any product

Search any product

7 Days Hong Kong Tour

Eat,Shop, Drop at Tsim Sha Tsui, Hong Kong Disneyland, Hiking at Dragon Back Mountain

SGD 888 ~~1000~~

4 Days Bangkok Highlights

Siam to Land of Smiles

SGD 666 ~~800~~

14 Days Bongio Ciao Italy & Paris Trip

Rome, Florence, Cinque Terre, Venice, Paris

SGD 5000 ~~8000~~

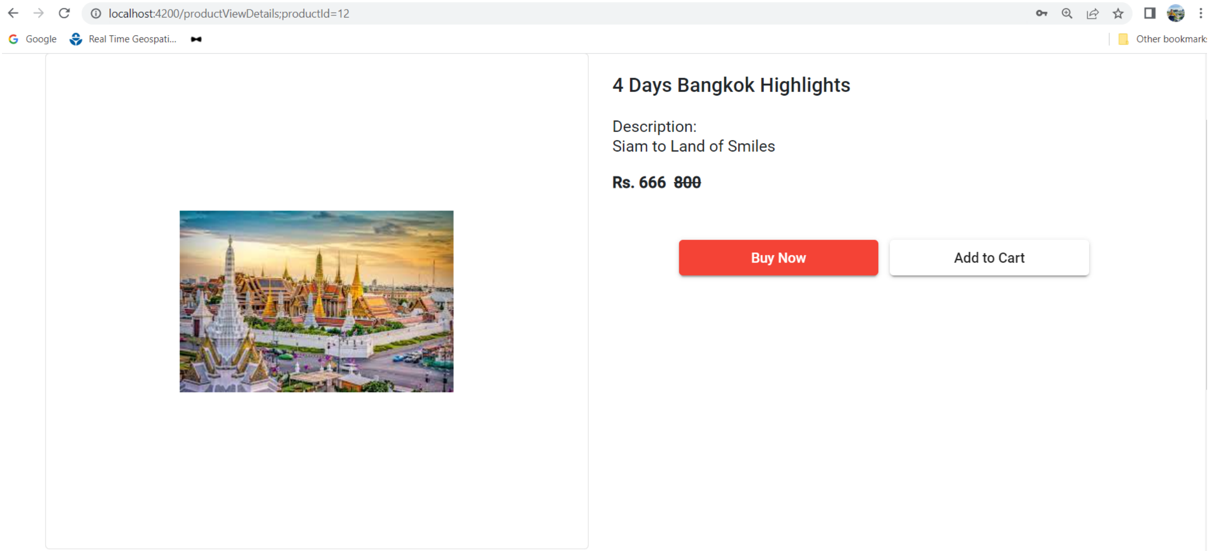
9 Days London Tour

London Bridge, Buckingham Palace, The British Musuem

SGD 3000 ~~6000~~

4. Once a user/ new user is logged in, the user will be redirected to the home page. The home page contains the list of products that users can view,search and purchase.

Product page



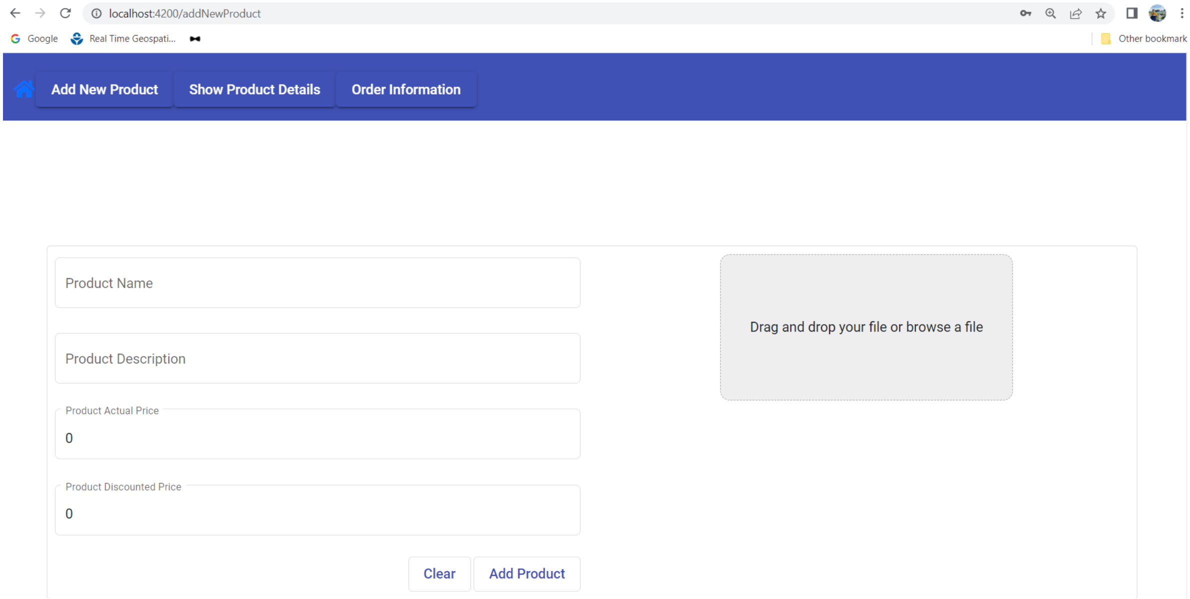
4 Days Bangkok Highlights

Description:
Siam to Land of Smiles

Rs. 666 ~~800~~

5. To view more details about a product, the user can click on the “View details” button. On the product detail’s page, the user can view the product’s details, e.g. image, description and price.

Adding/modifying/deleting products

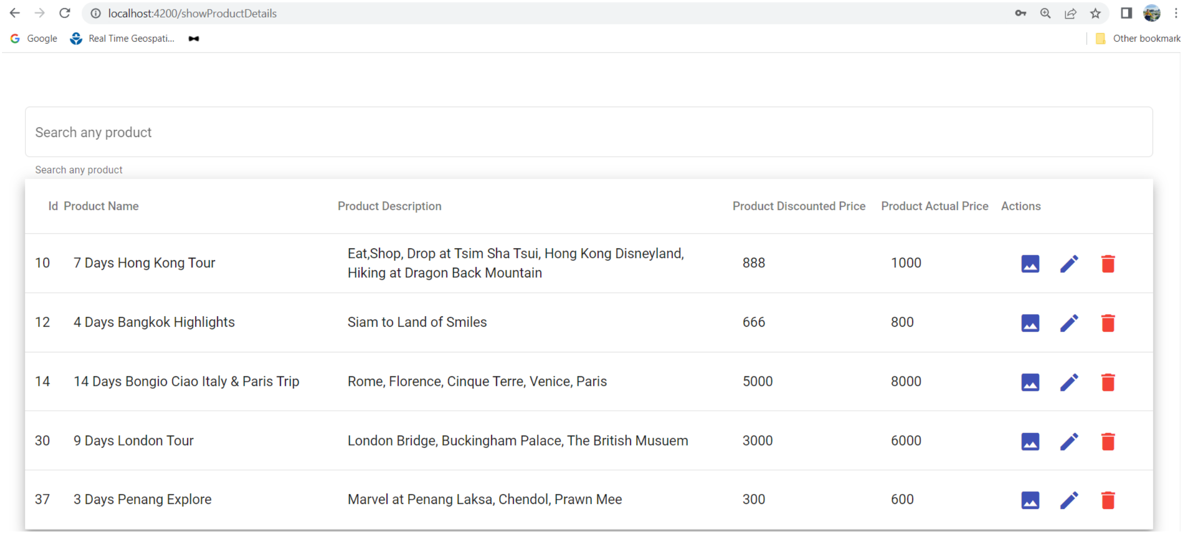


Product Name

Product Description

Product Actual Price: 0

Product Discounted Price: 0



Search any product

Search any product

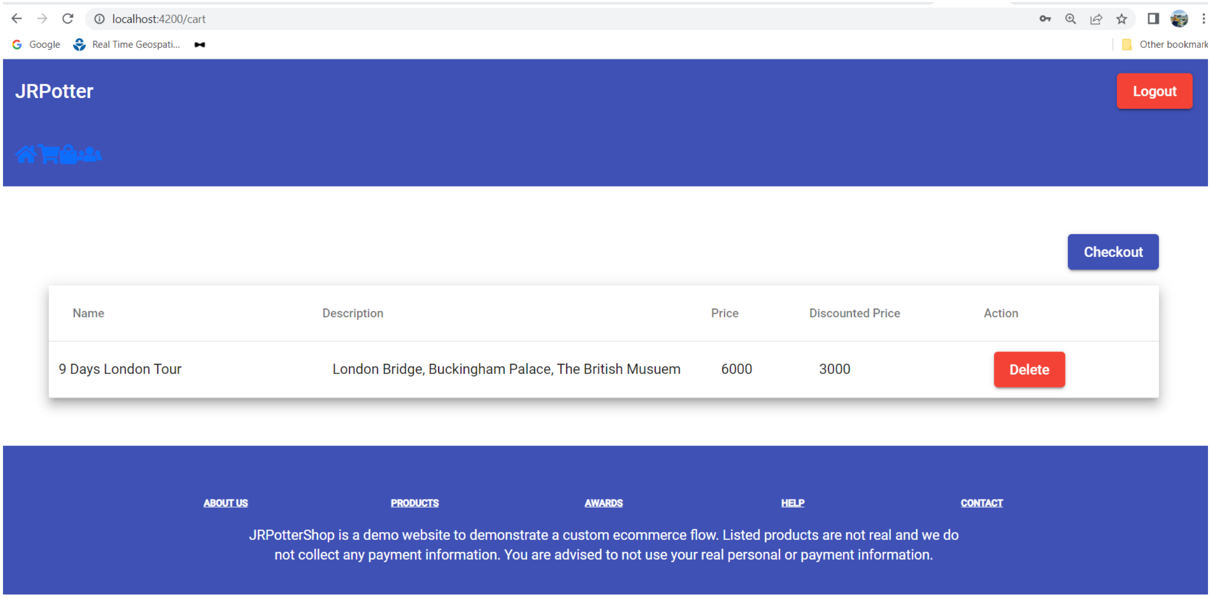
Id	Product Name	Product Description	Product Discounted Price	Product Actual Price	Actions
10	7 Days Hong Kong Tour	Eat,Shop, Drop at Tsim Sha Tsui, Hong Kong Disneyland, Hiking at Dragon Back Mountain	888	1000	
12	4 Days Bangkok Highlights	Siam to Land of Smiles	666	800	
14	14 Days Bongio Ciao Italy & Paris Trip	Rome, Florence, Cinque Terre, Venice, Paris	5000	8000	
30	9 Days London Tour	London Bridge, Buckingham Palace, The British Musuem	3000	6000	
37	3 Days Penang Explore	Marvel at Penang Laksa, Chendol, Prawn Mee	300	600	

6. The admin can add/modify/delete products under the products details tab.

Adding Items to cart

7. To purchase the product, the user can click on the “Add to Cart” button, which will add the product to the shopping cart.

Shopping cart

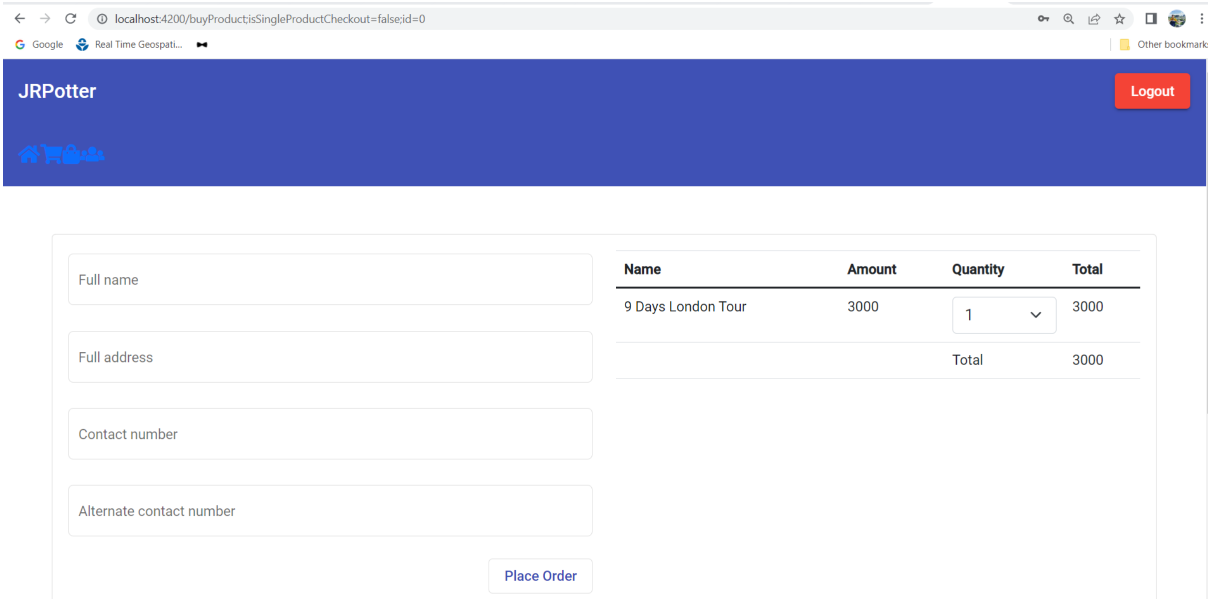


Name	Description	Price	Discounted Price	Action
9 Days London Tour	London Bridge, Buckingham Palace, The British Musuem	6000	3000	Delete

JRPotterShop is a demo website to demonstrate a custom ecommerce flow. Listed products are not real and we do not collect any payment information. You are advised to not use your real personal or payment information.

8. Users can view their shopping cart by clicking on the cart icon in the top left corner of the page.

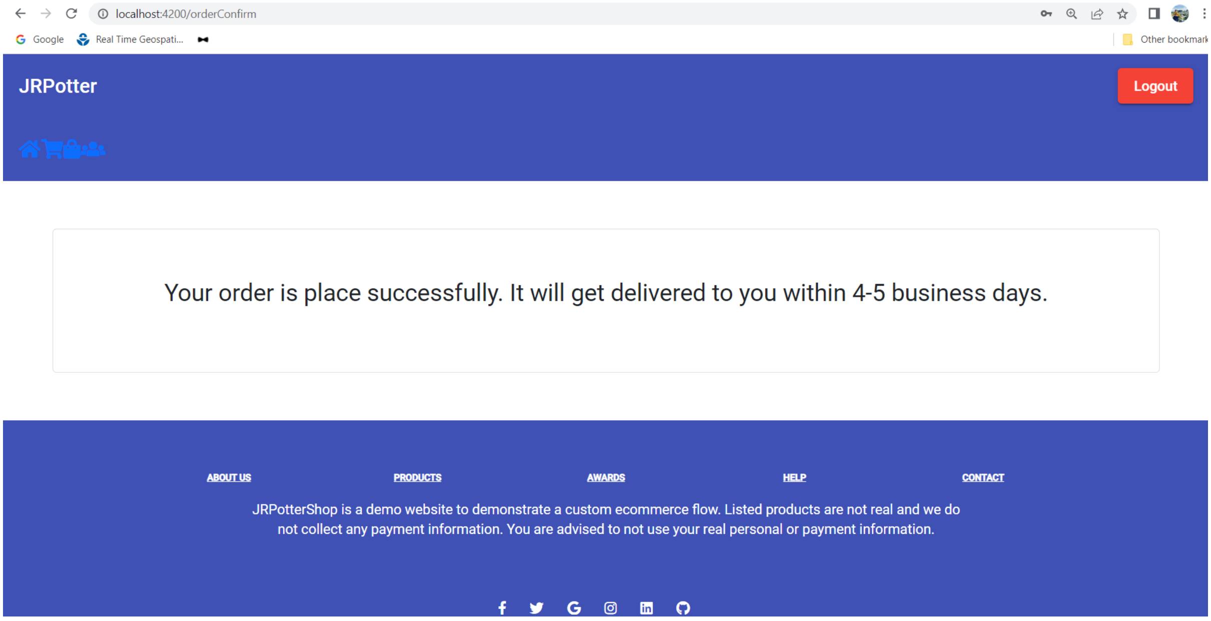
Place Order



Name	Amount	Quantity	Total
9 Days London Tour	3000	1	3000
		Total	3000

9. In the shopping cart, the user can view the items added and the total cost of their purchase. On the checkpage, users will need to enter their personal details and billing address.To proceed to checkout, users can click on the “Place Order” button in the shopping cart.

Order confirmation



Your order is place successfully. It will get delivered to you within 4-5 business days.

JRPotterShop is a demo website to demonstrate a custom ecommerce flow. Listed products are not real and we do not collect any payment information. You are advised to not use your real personal or payment information.

10. After submitting an order, users will receive a confirmation notification and view their order history and track the status of their order.

Logout

11. To logout of JRPotter, users can click on the “Logout” button.

6. Source Code

Source code of developed web application, JRpotter can be accessed via below link:- https://github.com/ngohjierui/JRPotter_2

7. References

1. T. L. P. Y. (2021). *GitHub - team-learn-programming-yourself/jwt-youtube*. Retrieved from https://github.com/team-learn-programming-yourself/jwt-youtube
2. T. L. P. Y. (2021). GitHub - team-learn-programming-yourself/jwt-youtube-ui. Retrieved from https://github.com/team-learn-programming-yourself/jwt-youtube-ui
3. Smashing Magazine. (2020) *Create A Responsive Dashboard With Angular Material And ng2-Charts*. Retrieved from https://www.smashingmagazine.com/2020/07/responsive-dashboard-angular-material-ng2-charts-schematics/
4. Techiediaries. (2022) *Build an Angular 14 CRUD Example & Tutorial* . Retrieved from https://www.techiediaries.com/angular/angular-9-8-crud-example-and-tutorial/
5. BezKorder. (2022). *Spring Boot + Angular 14 + MySQL: CRUD example*. Retrieved from https://www.bezkoder.com/spring-boot-angular-14-mysql/
6. Jason Watmore’s Blog (2019) *Angular 7 Tutorial Part 3 - Add Routing & Multiple Pages*. Retrieved from

https://jasonwatmore.com/post/2019/04/29/angular-7-tutorial-part-3-add-routing-multiple-pages

7. Statista (2023) *E-commerce worldwide*.Retrieved from: https://www.statista.com/topics/871/online-shopping/
8. EdxGroup (2021) *Incremental Model in SDLC: Use, Advantage & Disadvantage*. Retrieved from https://edxgroup.vn/2021/04/11/incremental-model-in-sdlc-use-advantage-disadvantage/
9. Visual Paradigm (n.d.) *Agile Development: Iterative and Incremental*. Retrieved from https://www.visual-paradigm.com/scrum/agile-development-iterative-and-incremental/
10. Bitbean. (n.d.).*Unlock the Potential of the Software Development Life Cycle*. Retrieved from https://www.bitbean.com/resources/unlocking-potential-of-software-development-life-cycle/