**Cardiff Metropolitan University**

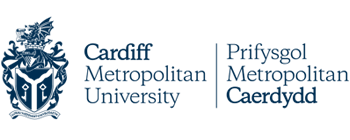
**B.Sc. (Hons) in Business Information Systems**

**Assignment Cover Sheet**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Details (Student should fill the content)** | | | | | | | | | |
| Name | | | Madura Prasad | | | | | | |
| Student ID | | | CL/BSCSD/24/71 | | | | | | |
| **Scheduled unit details** | | | | | | | | | |
| Unit code | | | CIS6003 | | | | | | |
| Unit title | | | Advanced Programming | | | | | | |
| Unit enrolment details | | | Year | | 3 | | | | |
| Study period | | 2022 | | | | |
| Lecturer | | | Assignment prepared by Tharik Kanaka | | | | | | |
| Mode of delivery | | | Full Time | | | | | | |
| **Assignment Details** | | | | | | | | | |
| Nature of the Assessment | | | Coursework (a report) | | | | | | |
| Topic of the Case Study | | | Online loan offering system to buy items and it in installments | | | | | | |
| Learning Outcomes covered | | | LO2, LO3, LO4 | | | | | | |
| Word count | | | 4000(report) | | | | | | |
| Due date / Time | | | 03rd March, 2023 | | | | | | |
| Extension granted? | | | Yes | No | Extension Date | | |  | |
| Is this a resubmission? | | | Yes | No | Resubmission Date | | |  | |
| **Declaration** | | | | | | | | | |
| I certify that the attached material is my original work. No other person’s work or ideas have been used without acknowledgement. Except where I have clearly stated that I have used some of this material elsewhere, I have not presented it for examination / assessment in any other course or unit at this or any other institution | | | | | | | | | |
| Name/Signature | | |  | | | Date | |  | |
| **Submission** | | | | | | | | | |
| Return to: | | |  | | | | | | |
| **Result** | | | | | | | | | |
| Marks by 1st Assessor |  | Name & Signature of the 1st Assessor | | | | |  | | **Agreed Mark** |
| Marks by2nd Assessor |  | Name & Signature of the 2nd Assessor | | | | |  | |
| Comments on the Agreed mark |  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STUDENT NAME:**  **CMU B.Sc. (HONS) BIS - ASSIGNMENT FEEDBACK SHEET –ICBT CAMPUS** | | | | | | | | | | **STUDENT NUMBER:** | | |
| **Module Number & Title**: Advanced Programming | | | | | | | | | | **Semester: II** | | |
| **Assignment Type & Title:** Coursework: | | | | | | | | | | | | |
| **For student use: *Critical feedback on the individual progression towards achieving the assignment outcomes*** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **For the Assessors’ feedback**  **Indicate the Task number strength and Weaknesses and the marks for each task** | | | | | | | | | | | | |
| **Task No/Question No** | **Strengths (1st Assessor)** | | | | | | **Strengths (2nd Assessor)** | | | | | |
| **Task No / Question No** | **Weaknesses (1st Assessor)** | | | | | | **Weaknesses (2nd Assessor)** | | | | | |
| **Areas for future improvement** | | | | | | | | | | | | |
| **Comments by 1st Assessor** | | | | | | **Comments by 2nd Assessor** | | | | | | |
| **Marks** | | | | | | | | | | | | |
| **Task /Question No** | | **Marks by 1st Assessor** | **Marks by 2nd Assessor** | | **Marks by IV (if any)** | | | **IV comments (If Any)** | | | | |
|  | |  |  | |  | | |  | | | | |
|  | |  |  | |  | | |
|  | |  |  | |  | | |
| **Total Marks** | |  |  | |  | | |
| **Name and the Signature of the 1st Assessor** | | | |  | | | | | **Date:** | |  | |
| **Name & Signature of the 2ndAssessor:** | | | |  | | | | | **Date:** | |  | |
| **Name & Signature of the IV: (If any)** | | | |  | | | | | **Date:** | | |  |

**Advanced Programming**



**Student ID:** Madura Prasad

**Student Name:** CL/BSCSD/24/71

# **Acknowledgement**

We had to enlist the assistance and counsel of a few respected individuals in order to complete the project, and we owe them our sincerest appreciation. We would like to express our thanks to the Course Instructor Mr. Tharik Kanaka on the ICBT Campus for providing us with an excellent assignment guidance through multiple discussions.

Furthermore, my classmates, in particular, have offered helpful remark ideas on my work, which has inspired me to increase the quality of the project.

**Table of Contents**

[Acknowledgement 5](#_Toc131767638)

[Introduction 11](#_Toc131767639)

[Task A 12](#_Toc131767640)

[Requirement specification for the proposed system 12](#_Toc131767641)

[1) Introduction 12](#_Toc131767642)

[2) Purpose of the system 12](#_Toc131767643)

[3) Scope of the system 13](#_Toc131767644)

[4) Objective 15](#_Toc131767645)

[5) Functional requirements 15](#_Toc131767646)

[6) Non-functional requirements 16](#_Toc131767647)

[Task B 18](#_Toc131767648)

[UML Diagrams 18](#_Toc131767649)

[1) Use case Diagram 18](#_Toc131767650)

[2) Class Diagram 20](#_Toc131767651)

[3) Admin - Sequence Diagram 22](#_Toc131767652)

[4) Customer - Sequence Diagram 24](#_Toc131767653)

[Task C 26](#_Toc131767654)

[Design Patterns 26](#_Toc131767655)

[1) Creational design patterns 26](#_Toc131767656)

[2) The singleton patterns 27](#_Toc131767657)

[3) The factory patterns 28](#_Toc131767658)

[4) The abstract factory patterns 29](#_Toc131767659)

[5) Creational design patterns 30](#_Toc131767660)

[6) The Factory Design Pattern 31](#_Toc131767661)

[Task D 32](#_Toc131767662)

[1) Add Product 32](#_Toc131767663)

[2) Update Product 35](#_Toc131767664)

[3) Delete Product 38](#_Toc131767665)

[4) Read Product 41](#_Toc131767666)

[5) Add Customer 45](#_Toc131767667)

[6) Update Customer 48](#_Toc131767668)

[7) Delete Customer 51](#_Toc131767669)

[8) Read Customer 54](#_Toc131767670)

[9) Read Instalment Details 57](#_Toc131767671)

[10) Customer Registration 60](#_Toc131767672)

[11) Over 18 Installment Plan 62](#_Toc131767673)

[12) Admin Login 65](#_Toc131767674)

[Task E 67](#_Toc131767675)

[Test Plan and test-driven development 67](#_Toc131767676)

[1) Customer Unit Testing 67](#_Toc131767677)

[2) Product Unit Testing 68](#_Toc131767679)

[3) Payment Unit Testing 69](#_Toc131767681)

[Test Plan 70](#_Toc131767683)

[Test Cases 72](#_Toc131767684)

[Task F 93](#_Toc131767685)

[User manual and technical documentation 93](#_Toc131767686)

[Admin Panel 93](#_Toc131767687)

[1) Login 93](#_Toc131767688)

[2) Admin Panel 94](#_Toc131767689)

[3) Customer Function 95](#_Toc131767690)

[4) Product Functions 97](#_Toc131767691)

[5) Customer 99](#_Toc131767692)

[6) Product Page 100](#_Toc131767693)

[7) Cart Page 101](#_Toc131767694)

[8) Check-Out Page 102](#_Toc131767695)

[9) Installment Plan Function 103](#_Toc131767696)

[Task G 105](#_Toc131767697)

[GitHub Repository 105](#_Toc131767698)

[Conclusion 106](#_Toc131767699)

[References 107](#_Toc131767700)

**Table of Figures**

[Figure 1 : Use Case Diagram 18](#_Toc131766647)

[Figure 2 : Class Diagram 20](#_Toc131766648)

[Figure 3 : Admin Sequence Diagram 22](#_Toc131766649)

[Figure 4 : Customer Sequence Diagram 24](#_Toc131766650)

[Figure 5 Creational Design Pattern 26](#_Toc131766651)

[Figure 6 Database Creation Code 27](#_Toc131766652)

[Figure 7 ResourseNotFoundException Code 27](#_Toc131766653)

[Figure 8 Factory Design Pattern 28](#_Toc131766654)

[Figure 9 Abstract Factory Pattern 29](#_Toc131766655)

[Figure 10 Creational Design Pattern 30](#_Toc131766656)

[Figure 11 Factory Design Pattern 31](#_Toc131766657)

[Figure 12 Admin Add Product Front-End 32](#_Toc131766658)

[Figure 13 Admin Add Product Front-End Code 33](#_Toc131766659)

[Figure 14 Admin Add Product Front-End API Connect Code 33](#_Toc131766660)

[Figure 15 Admin Add Product Spring Bot Code 34](#_Toc131766661)

[Figure 16 Add Product Update Front-Code 35](#_Toc131766662)

[Figure 17 Update Product Front-End Code 36](#_Toc131766663)

[Figure 18 Update Product API Connected Code 36](#_Toc131766664)

[Figure 19 Update Product Spring Bot Code 37](#_Toc131766665)

[Figure 20 Delete Product Front-End 38](#_Toc131766666)

[Figure 21 Delete Product Front-End Code 39](#_Toc131766667)

[Figure 22 Delete Product API Connected Code 39](#_Toc131766668)

[Figure 23 Delete Product Spring Bot Code 40](#_Toc131766669)

[Figure 24 View Product Front-End 41](#_Toc131766670)

[Figure 25 View Product Front-End Code 42](#_Toc131766671)

[Figure 26 View Product API Code Connect Code 43](#_Toc131766672)

[Figure 27 View Product Spring-Bot Code 43](#_Toc131766673)

[Figure 28 Add Customer Front-End 45](#_Toc131766674)

[Figure 29 Add Customer Front-End Code 46](#_Toc131766675)

[Figure 30 Add Customer API Connected Code 46](#_Toc131766676)

[Figure 31 Add Customer Spring Bot Code 47](#_Toc131766677)

[Figure 32 Update Customer Front-End 48](#_Toc131766678)

[Figure 33 Update Customer Front-End Code 49](#_Toc131766679)

[Figure 34 Update Customer API Connected Code 49](#_Toc131766680)

[Figure 35 Update Customer Spring-Bot Code 50](#_Toc131766681)

[Figure 36 Delete Customer Front-End 51](#_Toc131766682)

[Figure 37 Delete Customer Front-End Code 52](#_Toc131766683)

[Figure 38 Delete Customer API Connected Code 52](#_Toc131766684)

[Figure 39 Delete Customer Spring-Bot Code 53](#_Toc131766685)

[Figure 40 View Customer Front-End 54](#_Toc131766686)

[Figure 41 View Customer Front-End Code 54](#_Toc131766687)

[Figure 42 View Product API Connected Code 55](#_Toc131766688)

[Figure 43 View Product Spring-Bot Code 55](#_Toc131766689)

[Figure 44 View Installment Plan Font-End 57](#_Toc131766690)

[Figure 45 View Installment Front-End Code 57](#_Toc131766691)

[Figure 46 View Installment API Connected Code 58](#_Toc131766692)

[Figure 47 View Installment Spring-Bot Code 58](#_Toc131766693)

[Figure 48 Register Front-Code 60](#_Toc131766694)

[Figure 49 Register Front-End Code 60](#_Toc131766695)

[Figure 50 Register API Connected Code 61](#_Toc131766696)

[Figure 51 Registered Spring-Bot Code 61](#_Toc131766697)

[Figure 52 Over 18 Installment Front-End Code 63](#_Toc131766698)

[Figure 53 Over 18 Installment Divided amount code 63](#_Toc131766699)

[Figure 54 Over 18 Installment Front-End Code 63](#_Toc131766700)

[Figure 55 Admin Login Front-End 65](#_Toc131766701)

[Figure 56 Admin Login Front-End Code 65](#_Toc131766702)

[Figure 57 Unit Testing Customer 67](#_Toc131766703)

[Figure 58 Unit Testing Product 68](#_Toc131766704)

[Figure 59 Unit Testing Payment 69](#_Toc131766705)

[Figure 60 Login User-Manual 93](#_Toc131766706)

[Figure 61 Admin panel user-Manual 94](#_Toc131766707)

[Figure 62 Customer User-Manual 95](#_Toc131766708)

[Figure 63 Product User-Manual 97](#_Toc131766709)

[Figure 64 Customer User-Manual 99](#_Toc131766710)

[Figure 65 Product User-Manual 100](#_Toc131766711)

[Figure 66 Car User-Manual 101](#_Toc131766712)

[Figure 67 Checkout User-Manual 102](#_Toc131766713)

[Figure 68 Installment User-Manual 103](#_Toc131766714)

[Figure 69 Git Hub Repository 105](#_Toc131766715)

# **Introduction**

Bumble Bee is an e-commerce site that provides a unique and convenient shopping experience to its users. The site is designed to cater to the needs of individuals who want to purchase products online but don't have the necessary funds to do so. Bumble Bee's "Buy first and pay later" service allows users to make purchases online within a maximum entry budget of up to 15,000 LKR, without having to pay for it upfront. Instead, users can pay back the loan amount in three interest-free installments. Bumble Bee's user-friendly platform is built using cutting-edge technology. The site is developed using Spring Boot and React, two powerful and widely used technologies in the software development industry. The site also uses a REST API (Representational State Transfer Application Programming Interface) to enable seamless communication between different systems and applications.

One of the unique features of Bumble Bee is its focus on providing a hassle-free shopping experience. The site is designed to be easy to use, with intuitive navigation and a straightforward checkout process. Users can browse through a wide range of products, from electronics to home appliances, and add them to their cart with just a few clicks. Once the user is ready to make the purchase, they can select the "Buy now, pay later" option and complete the checkout process.

Bumble Bee's loan approval process is fast and straightforward. Users can apply for a loan within minutes by filling out a simple application form. The loan is approved instantly, and users can start shopping right away. Bumble Bee's payment system is secure and reliable, ensuring that users' personal and financial information is kept safe at all times.

# **Task A**

## **Requirement specification for the proposed system**

### **Introduction**

With Bumble Bee's innovative "Purchase first and pay later" method, customers can shop online without worrying about making a down payment of up to 15,000 LKR. Those above the age of 18 who want to reap the benefits of online shopping without the associated risk of making instant full payments can do so with the help of our application. Interest-free payments spread out over three months make the loan repayment process even more appealing. With Bumble Bee, customers have the freedom to choose how and when they pay back their loans, quick and simple access to a wide variety of products, and the peace of mind that comes from knowing that their personal and financial data is safe and secure. Our app is dedicated to giving you a stress-free, hassle-free, and time-saving purchasing experience. In order to ensure that Bumble Bee satisfies its users' expectations and advances the goals of the eCommerce sector, the requirements outlined in this paper have been meticulously crafted.

### **Purpose of the system**

The proposed system for the well-known online loan application Bumble Bee is meant to make online shopping easier for people over the age of 18 who want to enjoy its benefits without having to worry about making full payments right away. The system lets people buy things online up to a limit of 15,000 LKR without having to pay anything up front. The app is made to be easy to use and find your way around. Users can make an account on the platform by giving basic information like their name, address, and contact information. After making an account, users can look at the many products on the platform and add them to their shopping cart. At checkout, users can choose to pay for their purchases with a Bumble Bee loan. The total cost of the items in the cart and the user's maximum entry budget will be used to figure out how much of a loan to give.

The loan can be paid back in three interest-free payments, which makes it easy for users to pay back the loan without having to spend more money. The app uses advanced security measures like SSL encryption and two-factor authentication to make sure that user information and financial transactions are safe. The platform also has a customer service system to help users with problems or questions about their purchases or loan payments. In short, the proposed system for Bumble Bee is a safe and easy-to-use online shopping platform that gives users the option to get loans for their purchases without interest. This makes it easy for them to enjoy the benefits of online shopping without having to worry about money.

### **Scope of the system**

Scope for Bumble Bee E-commerce application that lets anyone over 18 buy things online with a maximum entry budget of up to 15,000LKR and pay back the loan in 3 interest-free installments can be defined as follows:

**Registering as a user**: This e-commerce application's first step is to register as a user. The user should be able to sign up for an account by giving their name, email address, and phone number.

**Age Verification**: Since the app is only for people over 18, it needs a way to check the user's age. The application can check the person's age with government-issued IDs like a National ID or Passport.

**Product Catalog**: The e-commerce app should have a wide range of products that users can buy with a maximum budget of 15,000LKR. The product catalogue should be well-organized and divided into categories so that it is easy to use.

**Product Details**: The product details page should have all the information about the product, such as its price, features, specifications, and description. The app should also let users look at pictures and videos of the products.

**Add to Cart**: The app should let users put items they want to buy in their shopping cart. The app should also show the total price of the cart, which includes any taxes, shipping costs, and other fees.

**Checkout**: The process of checking out should be easy and simple. Users should be able to choose how they want to pay and enter their shipping and billing information in the app.

**Loan Option**: The app should have a loan option that lets users pay for their purchases in 3 interest-free installments. The loan option should be easy to use, and the loan's terms and conditions should be clear on the application.

**Security**: The app should have strong security measures in place to protect sensitive information, such as personal and financial information, about users. The app should use encryption to keep user data safe and stop unauthorized people from getting to it.

Overall, an e-commerce app that lets anyone over the age of 18 buy things online with a maximum entry budget of up to 15,000LKR and the option to pay the loan back in 3 interest-free installments should have a wide range of products, safe payment options, and great customer service.

### **Objective**

* Give people over the age of 18 an easy-to-use interface that lets them look at and buy products online with a maximum entry budget of LKR. 15000.
* Sell a lot of different kinds of products, like electronics, home appliances, fashion, beauty products, and more.
* Make sure that all of the products on the platform come from reputable and trustworthy suppliers. This will give users high-quality, real products.
* Give users a unique way to pay for items without having to pay for them all at once. This could be a loan that can be paid back in 3 interest-free payments.
* Make it easy to pay back loans by automating the process and sending regular reminders and notifications so that users don't forget to pay.

### **Functional requirements**

This web app was developed to manage user functions and give a clean interface for inventory and sales management. All data entered into the system is saved in a database and can be retrieved at any time. All the new features are very simple to use, such as the ability to easily add, delete, update, and screen information on sales and branches. These are the features that may be found in the online programmed.

**Customers**

* Customers Can Register
* Manage cart
* Buy products

**Admin**

* CRUD Products
* CRUD Customers
* View Payment instalments details

### **Non-functional requirements**

**Performance**: These criteria can be thought of as a set of benchmarks against which a system's actual performance can be evaluated. The quality of a software system is defined by these non-functional requirements. If the web app fails to meet the non-functional requirements, it will also fail to meet the needs of its customers.

**Usability**: Even with significant traffic demand, the website should load in under 3 seconds. This ensures that users do not have to wait extended periods of time or face delays while browsing or making purchases on the platform.

**Usability**: The website should be simple to use, with detailed product descriptions, pricing, and availability information. This guarantees that users can locate what they're looking for fast and make informed purchases.

**Security**: To secure user data and transactions, the website should be constructed with security in mind, employing industry-standard encryption and authentication mechanisms. This prevents consumers' personal and financial information from being compromised.

**Reliability**: The website should always be available and accessible, with little downtime or service interruptions. This ensures that consumers can make purchases and log into their accounts at any time.

**Scalability**: The website should be designed to accommodate a rising number of visitors and transactions over time while maintaining performance, security, and dependability. This ensures that the platform can withstand increased demand and usage without failing or becoming unstable.

**Compatibility**: The website should be accessible to users regardless of their preferred technology or operating system.

**Accessibility**: The website should be developed with screen reader compatibility, alternative text descriptions for images, and keyboard navigation choices in mind. This ensures that disabled users may access and use the platform.

**Compliance**: The website should adhere to all applicable legal and regulatory standards, such as data protection, privacy, and consumer protection legislation. This guarantees that the platform functions in an ethical and responsible manner, protecting both users and the company. (ieeexplore, 2020)

# **Task B**

## **UML Diagrams**

1. **Use case Diagram**

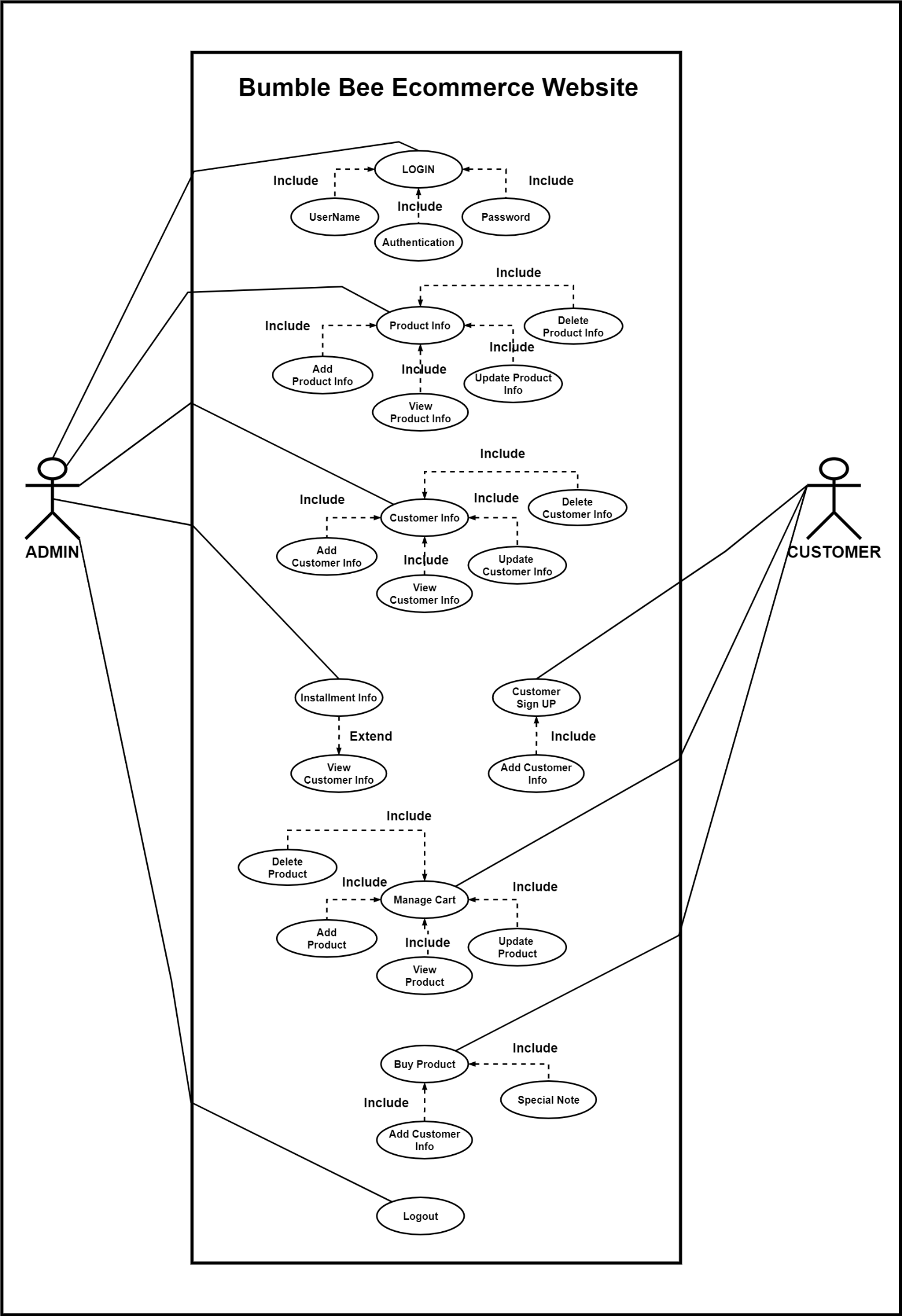


Figure 1 : Use Case Diagram

A use case diagram is a visual representation of the functional requirements of a system and shows the interactions between the system and its actors. Here's an example of a use case diagram for a Bumble Bee Ecommerce website that has two actors: admin and customer. The use case diagram consists of two main components: actors and use cases.

**Actors:**

* Admin: A person who is responsible for managing the products and customers of the ecommerce website.
* Customer: A person who is interested in buying products from the ecommerce website.

**Use cases:**

* Login: Both admin and customer can log into the system by their user credentials.
* Logout: Both admin and customer can log out of the system after they have completed their activities.
* Manage Products: Admin can add, update, and delete products from the website.
* Manage Customers: Admin can view and manage customer information such as personal details and order history.
* Manage Cart: Customer can add, remove, and update products in their shopping cart.
* Buy Products: Customer can purchase products from the ecommerce website by adding them to their cart and completing the checkout process.

1. **Class Diagram**

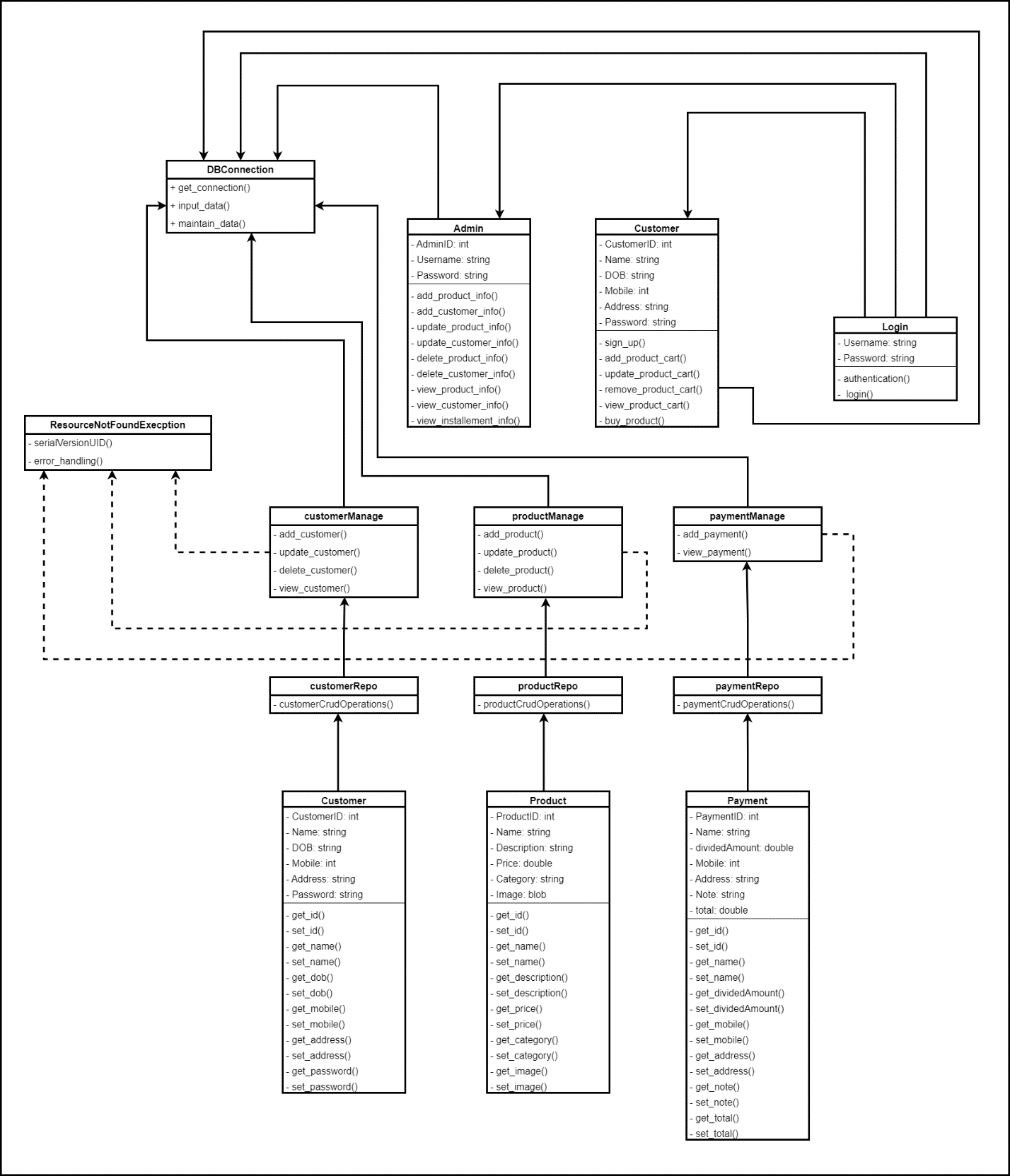


Figure 2 : Class Diagram

A class diagram is a graphical representation of the classes, interfaces, and their relationships to one another in a system. It shows the static structure of the system and helps in understanding the system's architecture. Now, let's dive into the classes of the Bumble Bee E-commerce website:

* Admin: The Admin class represents the administrators of the website who have the authority to manage the products, customers, and payments. The class includes the main function of managing products, managing customers, and managing payments.
* Product: The Product class represents the products available on the website. It includes attributes such as product name, description, price, and quantity. The Manage Product function in the admin class is responsible for creating, updating, and deleting products.
* Payment: The Payment class represents the different payment methods available on the website. It includes attributes such as payment method name, description, and availability. The Manage Payment function in the admin class is responsible for adding, updating, and deleting payment methods.
* Customer: The Customer class represents the users who visit the website and purchase products. It includes attributes such as customer name, address, email, and phone number. The Manage Customer function in the admin class is responsible for managing customer accounts, including adding, updating, and deleting customer information.

The class diagram for the Bumble Bee E-commerce website includes the Admin, Product, Payment, and Customer classes, where the main functions of managing products, managing customers, and managing payments are included in the admin class. This class diagram provides a visual representation of the static structure of the system and helps in understanding the relationships between the classes.

1. **Admin - Sequence Diagram**

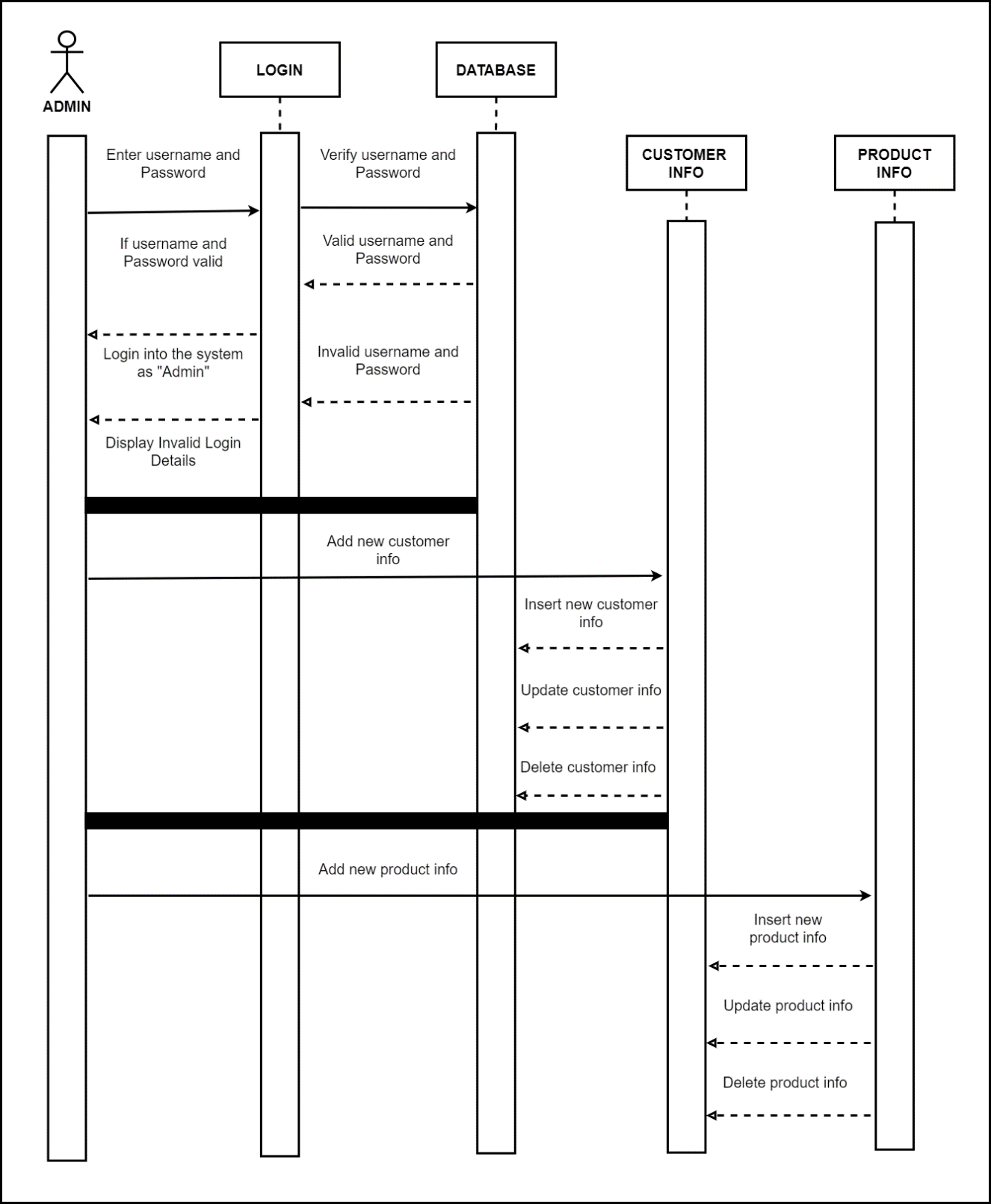


Figure 3 : Admin Sequence Diagram

A sequence diagram is a type of UML diagram that shows how objects interact with each other in a system over time. In this case, the sequence diagram will show the interaction between the admin and the various entities in the Bumble Bee e-commerce website. Here's a step-by-step breakdown of the sequence diagram for the admin in the Bumble Bee e-commerce website:

* The admin opens the website and selects the login option.
* The website prompts the admin to enter their login credentials.
* The admin enters their login credentials and submits them to the website.
* The website verifies the admin's login credentials with the database.
* If the login credentials are valid, the website displays the admin dashboard.
* From the dashboard, the admin can choose to manage either products or customers.
* If the admin chooses to manage products, the website retrieves the product data from the database.
* The admin can then add, edit, or delete products as necessary.
* If the admin chooses to manage customers, the website retrieves the customer data from the database.
* The admin can then add, edit, or delete customers as necessary.

Overall, the sequence diagram for the admin in the Bumble Bee e-commerce website shows how the admin interacts with the website and the various entities within it to manage products and customers. It also shows how the website retrieves and stores data in the database to facilitate this interaction.

1. Customer - Sequence Diagram

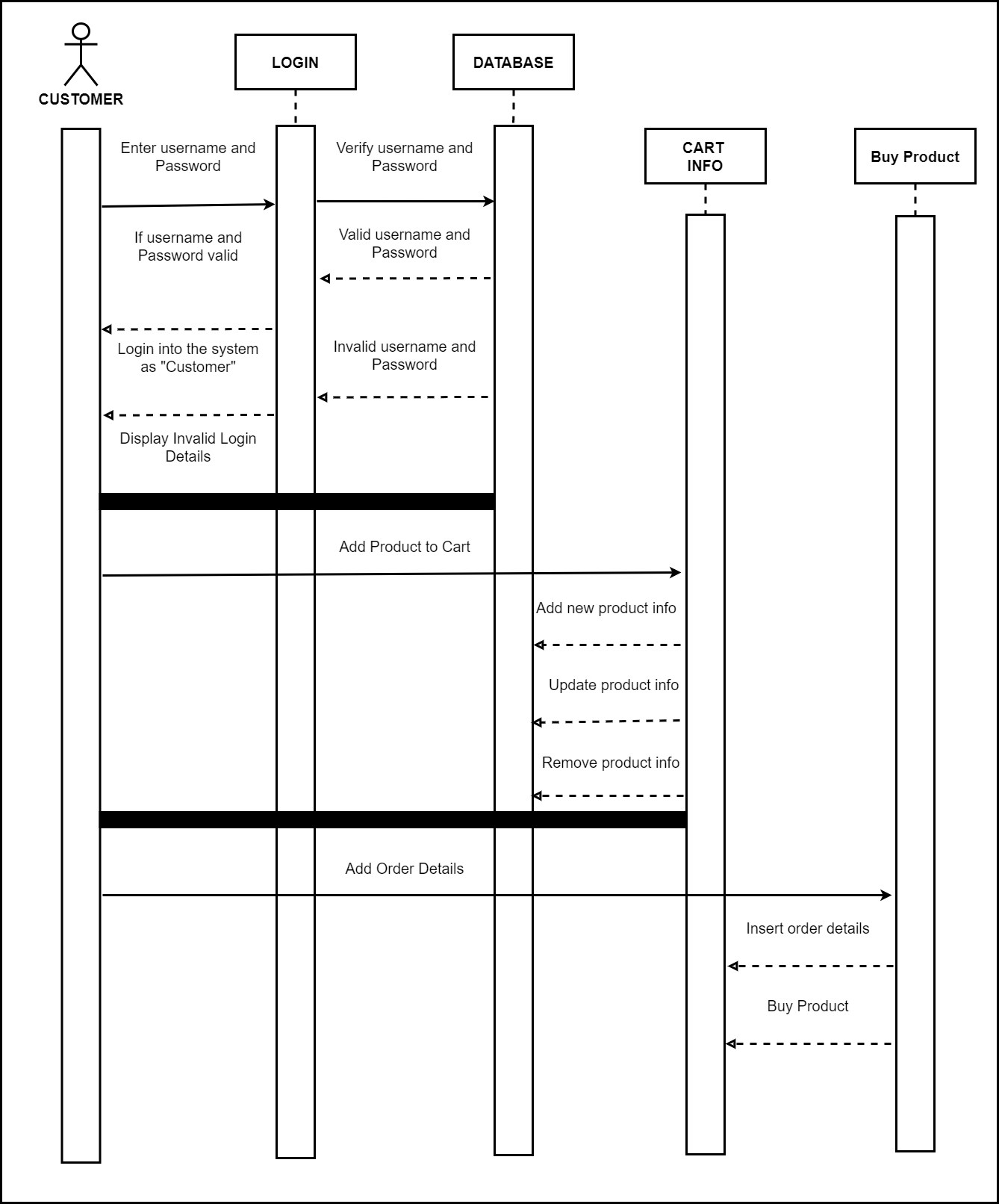


Figure 4 : Customer Sequence Diagram

In customer sequence diagram entities interact with each other in a typical customer sequence diagram. The customer logs in to their account using their username and password and login entity validates the credentials and grants access to the customer's profile. The customer browses the product catalog and selects the items they want to buy. The database entity retrieves the product details and displays them on the screen. The customer adds the selected products to their cart. The cart entity updates the cart contents and shows the updated total price. The customer reviews the cart items and proceeds to checkout. The buy product entity prompts the customer to enter their shipping and payment details. The customer confirms the order and submits the payment. The buy product entity processes the payment and updates the order status. The customer receives the order confirmation and the shipping details.

**Here's a brief explanation of each entity's main function:**

* Login: This entity is responsible for handling user authentication and authorization. It allows customers to log in to their accounts and access their personal information, saved addresses, and order history.
* Database: This entity stores all the relevant information about the products, customers, orders, and transactions. It ensures the accuracy and consistency of the data and provides efficient retrieval and update operations.
* Cart: This entity enables customers to add and remove products from their shopping cart. It calculates the total price, applies discounts or promotions, and shows the checkout options.
* Buy Product: This entity processes the customer's order and payment details. It updates the inventory and shipping status, generates the order confirmation, and sends the notification to the customer.

# **Task C**

## **Design Patterns**

A design pattern offers a typically reusable solution to the common problems in software design. The pattern typically illustrates links and relationships between groups of objects or artefacts. By providing a development model that has been tried and established, the goal is to hasten the growth process. Application patterns are specific programming strategies for a certain issue. So, a design pattern is a concept rather than a particular realization. By adopting design patterns, you may increase the adaptability, reuse, and maintenance of your code. You can't always use design patterns in every project. The creation of programmers is not the purpose of fashion trends. Design patterns are created to solve common issues. Choose a good pattern wherever you can to avoid such issues in the future. select the appropriate pattern. We need to make an effort to understand the objectives of the concept models. After that, we may decide which option is best. (tutorialspoint, 2021)

### **Creational design patterns**

Object creation mechanisms are the focus of the design pattern category known as creational design patterns. In addition to ensuring that the objects are made in a fashion that satisfies specific requirements, they offer means to generate things in a flexible and reusable way.

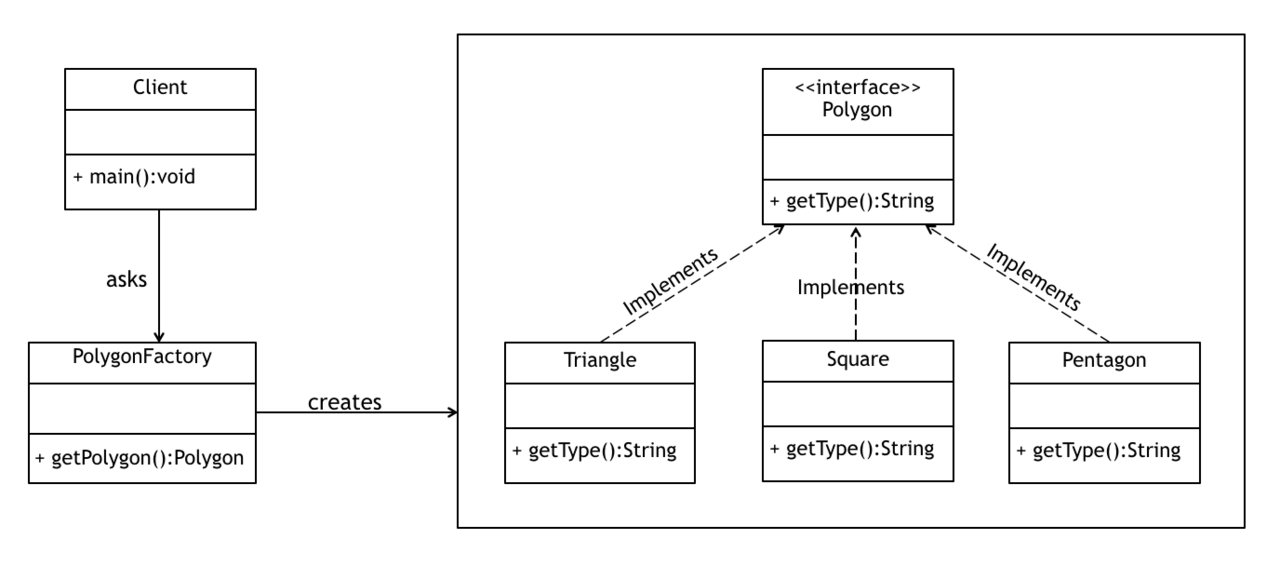


Figure Creational Design Pattern

### **The singleton patterns**

The singleton patterns make the guarantee that there is only ever one instance of a class produced. It is helpful when a single object needs to coordinate operations across a whole system since it offers a global point of access to that instance.



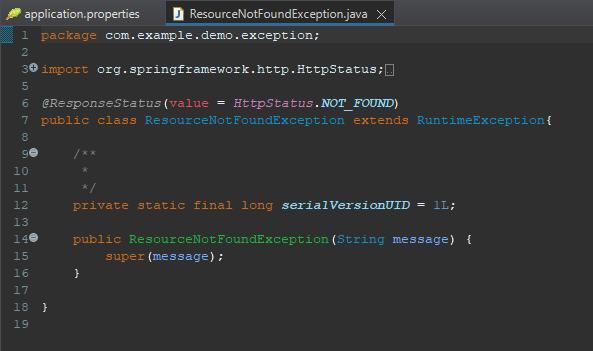
Figure Database Creation Code

Figure ResourseNotFoundException Code

### **The factory patterns**

The factory patterns give users a way to create objects, but it lets subclasses choose which class to instantiate. It is helpful in instances when there are several objects that need to be produced with various parameters or settings since it gives the means to assign the duty of object instantiation to various subclasses.

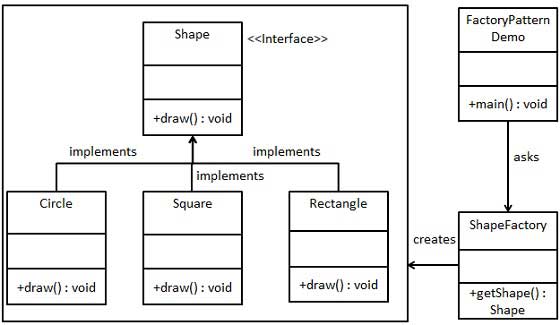


Figure Factory Design Pattern

### **The abstract factory patterns**

Abstract patterns offer a way to build object families that are dependent or connected without having to describe their specific classes. It is helpful when numerous linked items need to be generated concurrently since it enables the generation of related objects without requiring the precise type of each object to be specified.

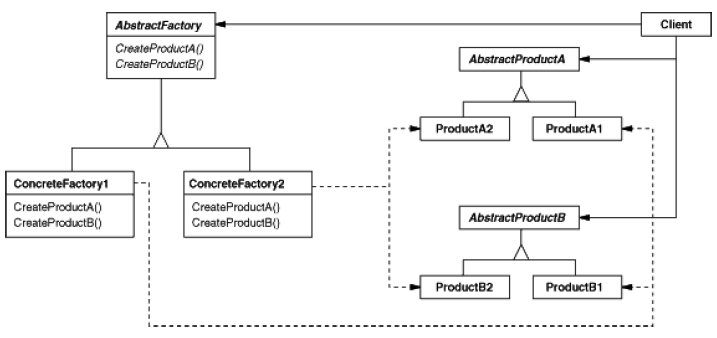


Figure Abstract Factory Pattern

### **Creational design patterns**

Creational design pattern can be helpful in making sure that objects are made in a fashion that satisfies particular needs and in making objects that are flexible and reusable. Developers can increase the overall quality and dependability of their product while also writing code that is simpler to maintain, alter, and expand by utilizing these patterns.

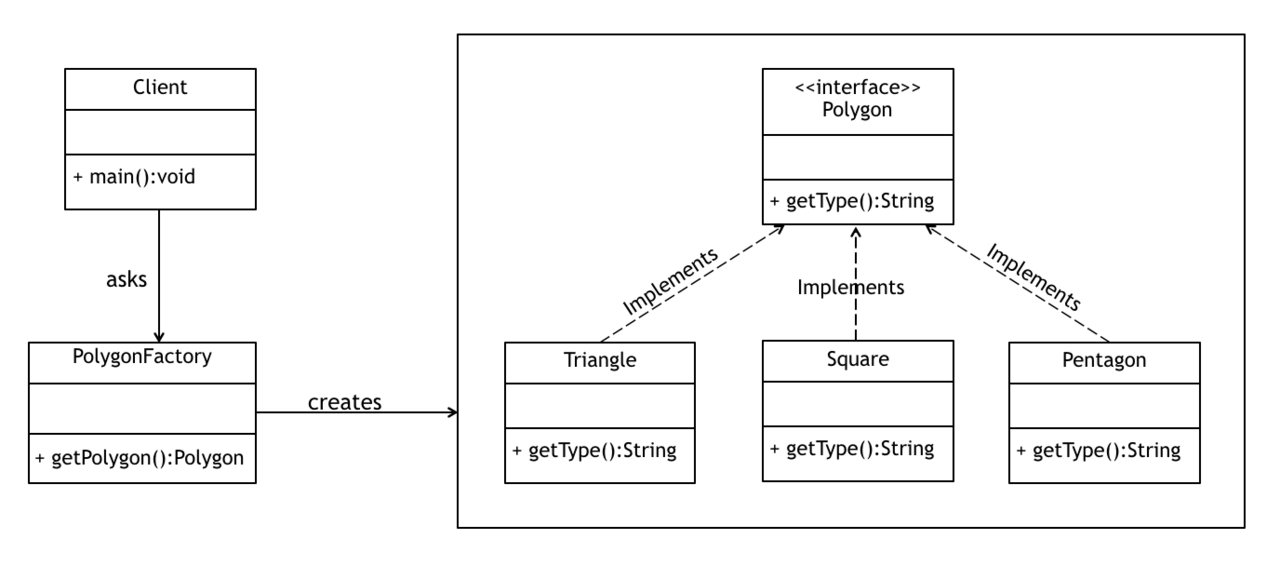


Figure Creational Design Pattern

### **The Factory Design Pattern**

The Factory Design Pattern, sometimes known as the Factory method, is one of the most widely used design patterns in Java. In spite of the fact that this pattern "defines an interface to generate an object, subclasses are free to choose which class they want to instantiate," The Factory process can be subclassed using a class deferral scheme. Using this pattern, a particular factory class receives the responsibility of first creating a virtual function Object () {[native code]} type on behalf of the client. To finish that, we rely on a plant that provides the artefacts while concealing the precise details of the implementation. The produced objects are reachable via a well-known interface.

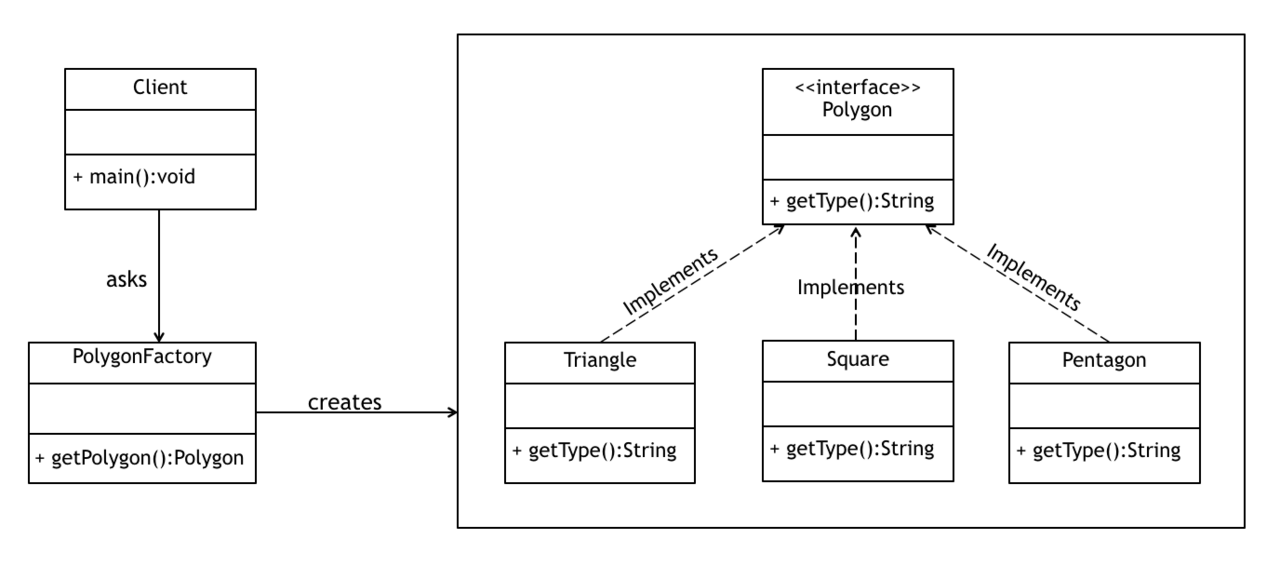


Figure Factory Design Pattern

# **Task D**

1. **Add Product**

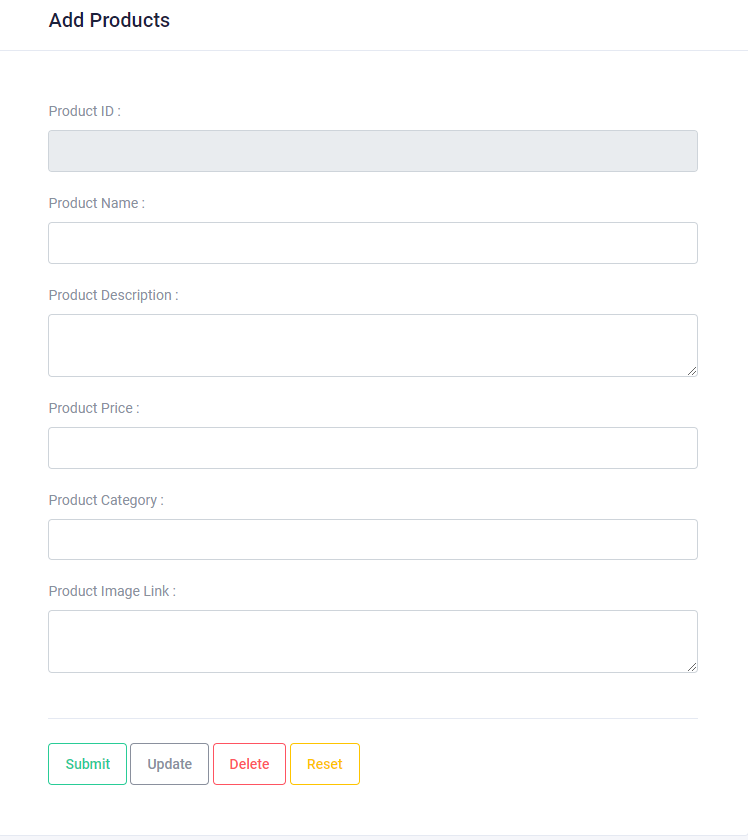


Figure Admin Add Product Front-End

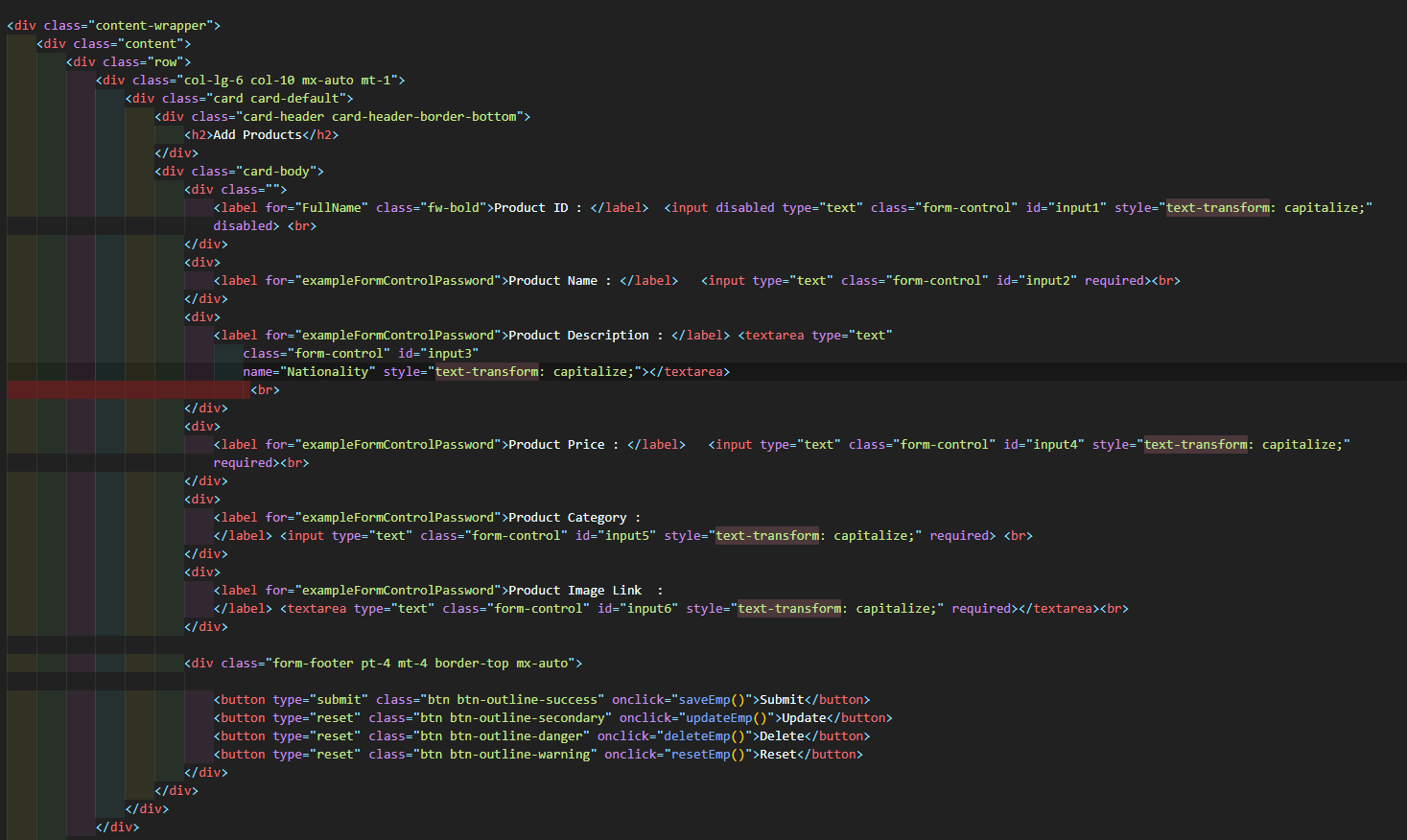


Figure Admin Add Product Front-End Code



Figure Admin Add Product Front-End API Connect Code

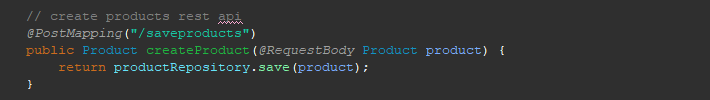


Figure Admin Add Product Spring Bot Code

E-commerce websites and applications frequently have the "Add product" functionality. Users can do this to add a new product to the website's stock or catalog.

Here's an overview of how the "add product" function typically works:

Logging into their account on the website or application and going to the inventory or catalog management area will allow users to use the "add product" capability. Enter product information: After selecting the "add product" option, the user is requested to enter the new product's information. The name of the item, its description, category, price, accompanying photographs, and any other pertinent information are examples of these specifics. Set inventory levels: If the website or application keeps track of inventory levels, the user might need to enter the number of units that are currently on sale for the product. Save product information: The user can save the new product to the inventory or catalog after providing all the required information. At this moment, buyers will be able to view and buy the new product. Before confirming the entry of the new product, users may also have the ability to examine and update the product details.

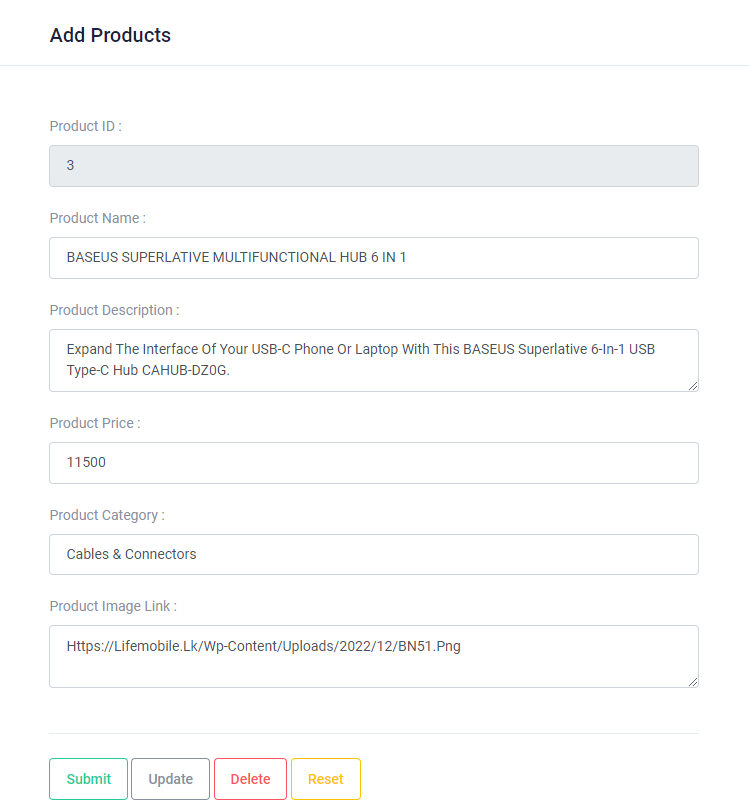
1. **Update Product**

Figure Add Product Update Front-Code

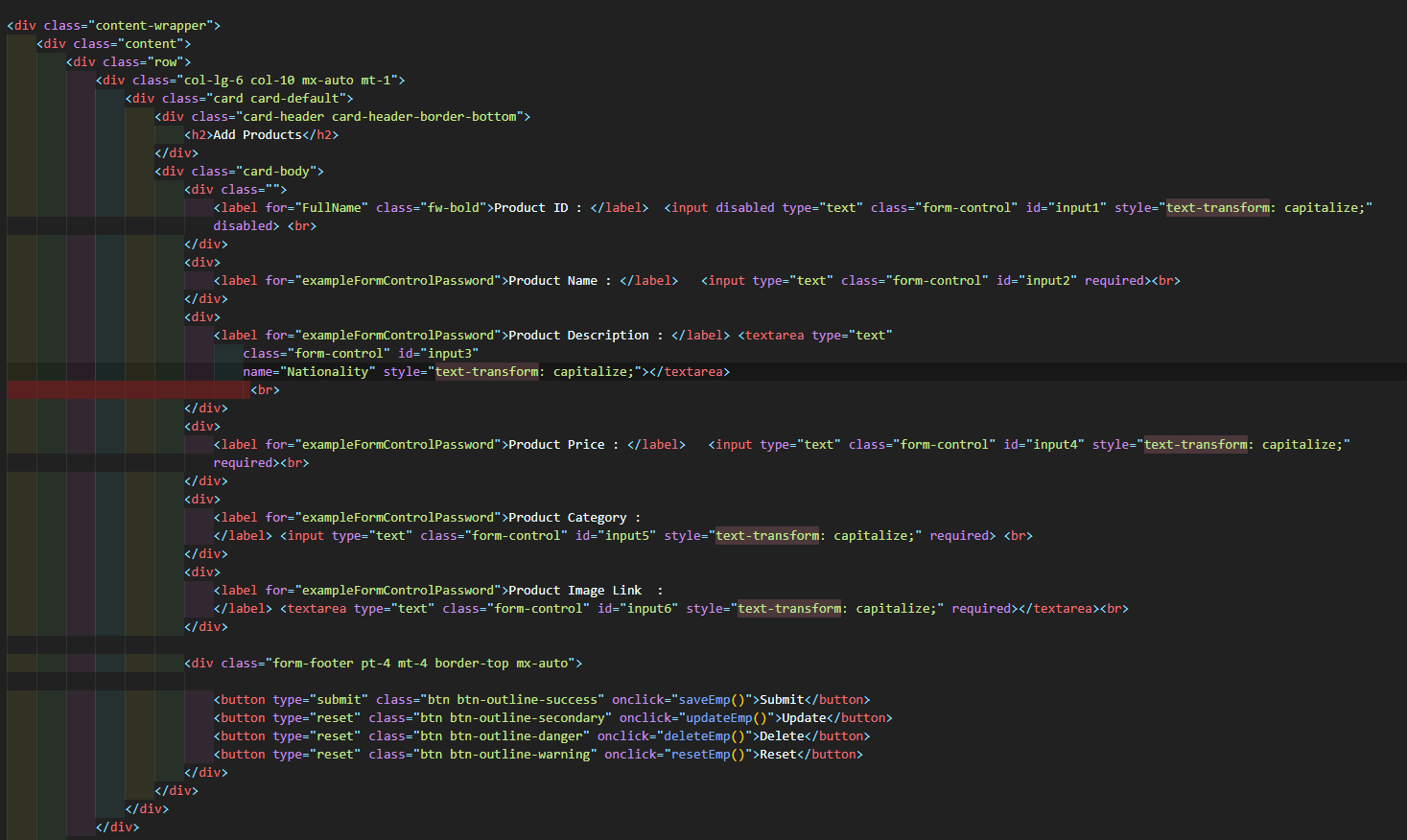


Figure Update Product Front-End Code



Figure Update Product API Connected Code

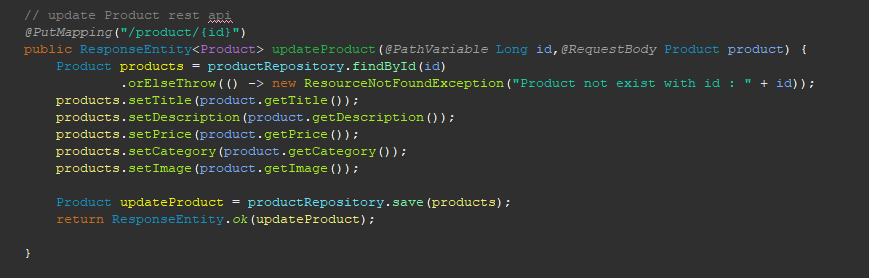


Figure Update Product Spring Bot Code

The "**Update product**" function is a feature commonly found in e-commerce websites or applications. It allows users to modify the details of an existing product in the website's catalog.

Here's an overview of how the "update product" function typically works:

Logging into their account on the website or application and going to the inventory or catalog management area will allow users to use the "update product" option. Pick the product you want to update: The user must choose the product they wish to update from the list of currently available items. Change product information: The user may change the product information as necessary after choosing the product they wish to edit. The name of the item, its description, category, price, accompanying photographs, and any other pertinent information are examples of these specifics. Save changes: The user can save the modified product information to the inventory or catalog after making the appropriate adjustments. Display alterations When the modified product details are saved, the modifications

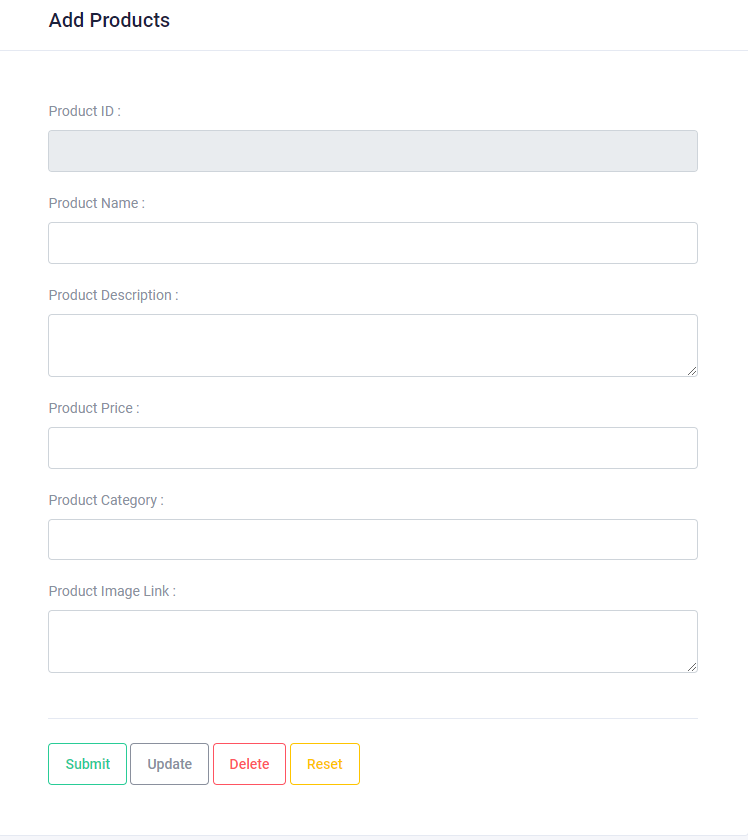
1. **Delete Product**

Figure Delete Product Front-End

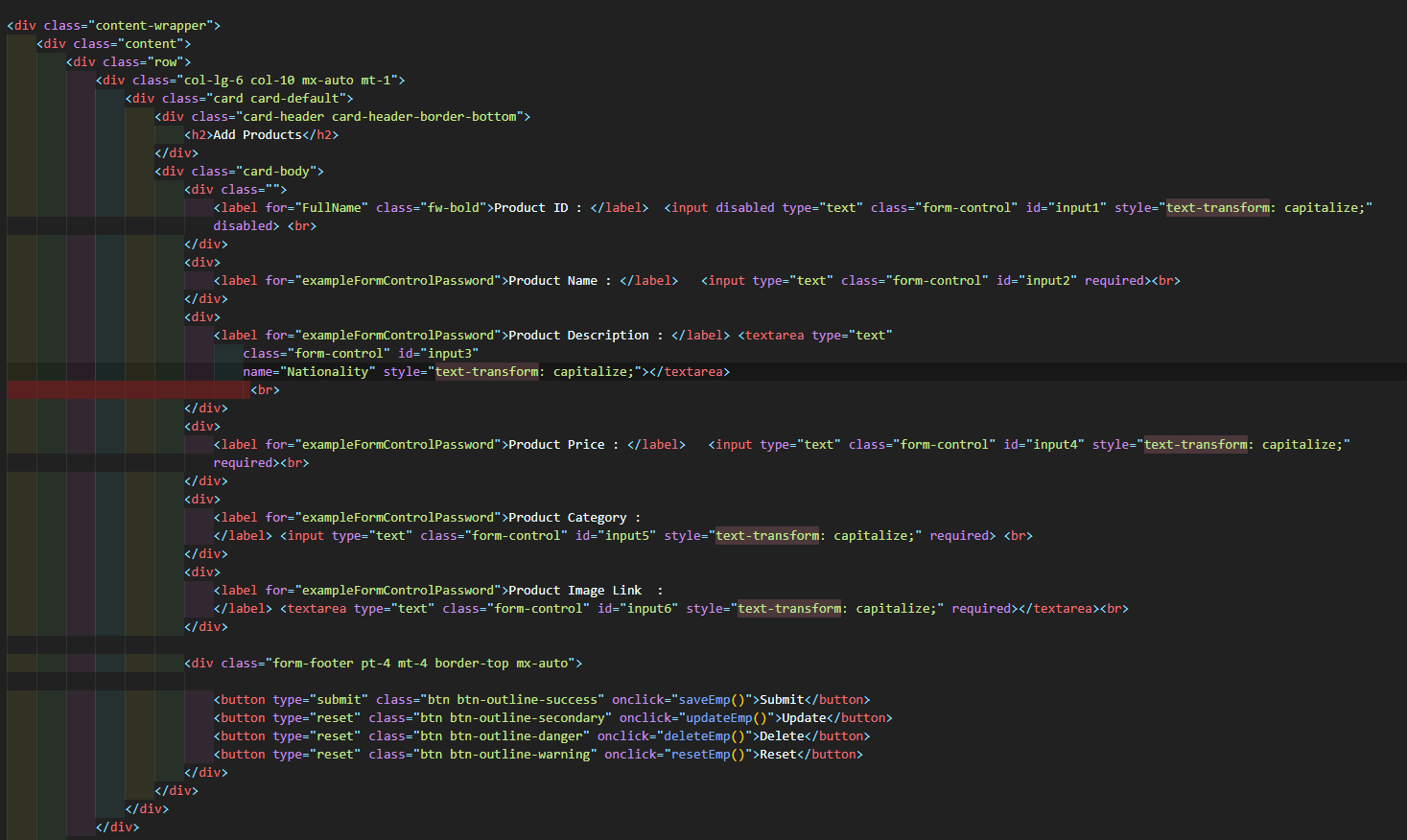


Figure Delete Product Front-End Code

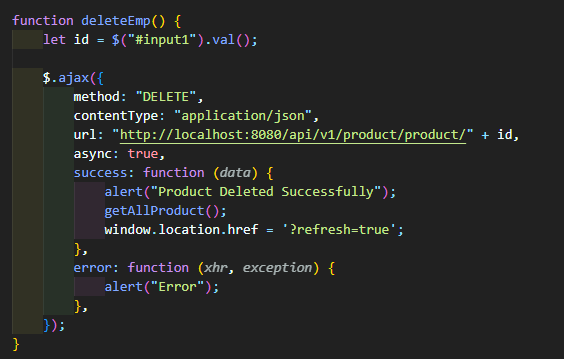


Figure Delete Product API Connected Code

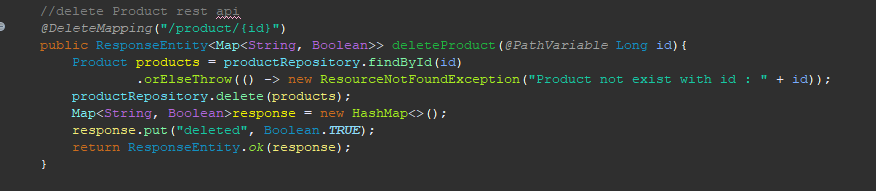


Figure Delete Product Spring Bot Code

The "**Delete product**" function is a feature commonly found in e-commerce websites or applications. It allows users to remove a product from the website's inventory or catalog.

Here's an overview of how the "delete product" function typically works:

Logging into their account on the website or application and going to the inventory or catalog management area will allow users to use the "remove product" option. Choose the item you want to remove: From the list of already-existing items, the user must choose the item they wish to remove. Confirm deletion: The customer will be asked to confirm their deletion of the product after selecting the one they wish to get rid of. This is a crucial step since removing a product from the website's inventory or catalog will result in its permanent removal. Delete the product: The website or application will delete the item from the inventory if the user verifies that they wish to do so.

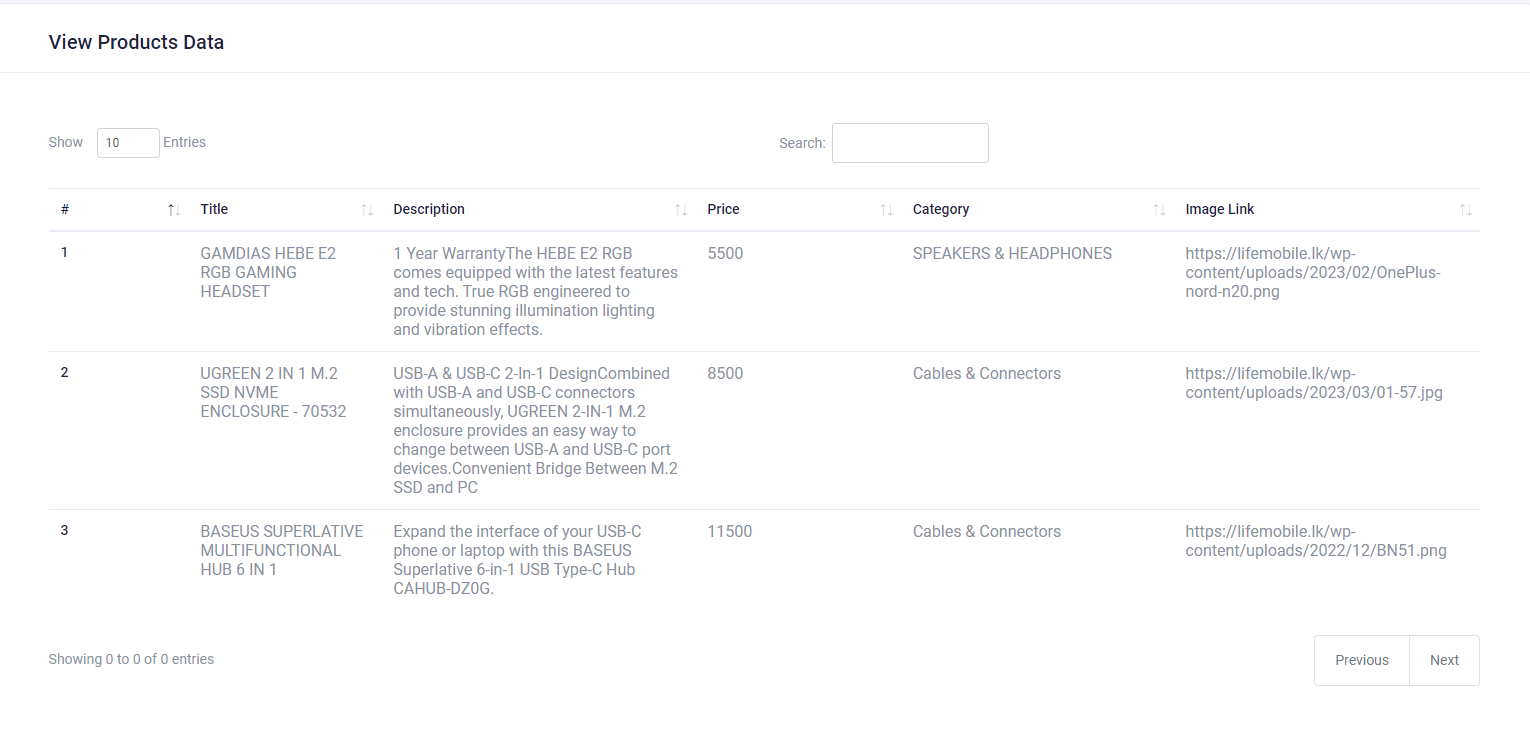
1. **Read Product**

Figure View Product Front-End

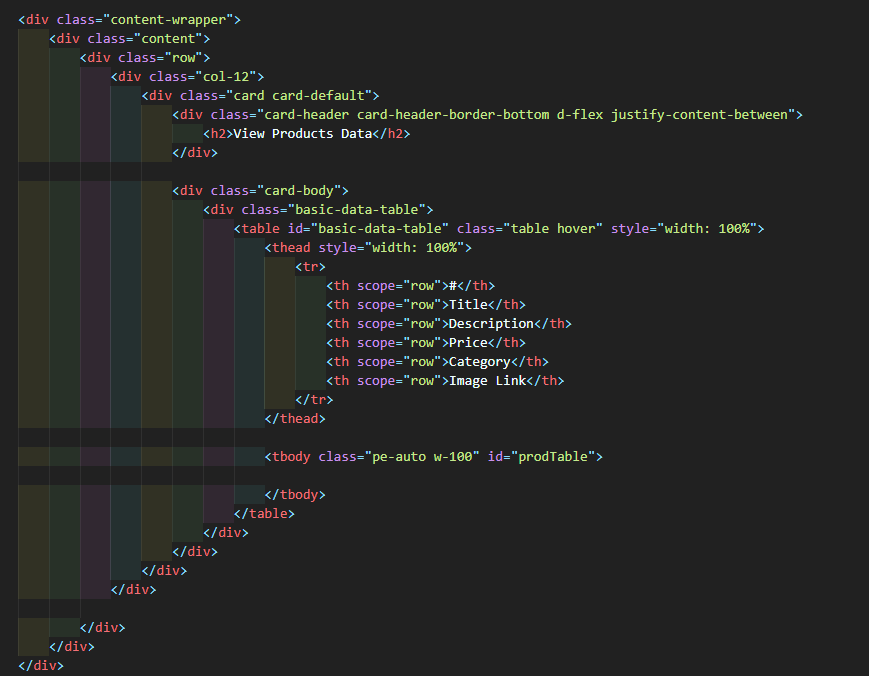


Figure View Product Front-End Code



Figure View Product API Code Connect Code

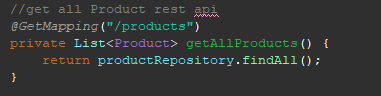


Figure View Product Spring-Bot Code

The "**Read product**" function is a feature commonly found in e-commerce websites or applications. It allows users to view the details of a specific product in the website's inventory or catalog.

Here's an overview of how the "read product" function typically works:

Logging into their account on the website or application and going to the inventory or catalog management part will allow users to use the "read product" capability. To view the product, choose: From the list of available items, the user must choose the item they wish to view. View product information: The user will be able to access the product information once they have chosen the item they wish to view. The name of the item, its description, category, price, accompanying photographs, and any other pertinent information are examples of these specifics. Edit product information: The "update product" feature can be used by the user to make modifications if they need to be made to the product's specifications. Add to shopping basket or buy: If the user so chooses

1. **Add Customer**

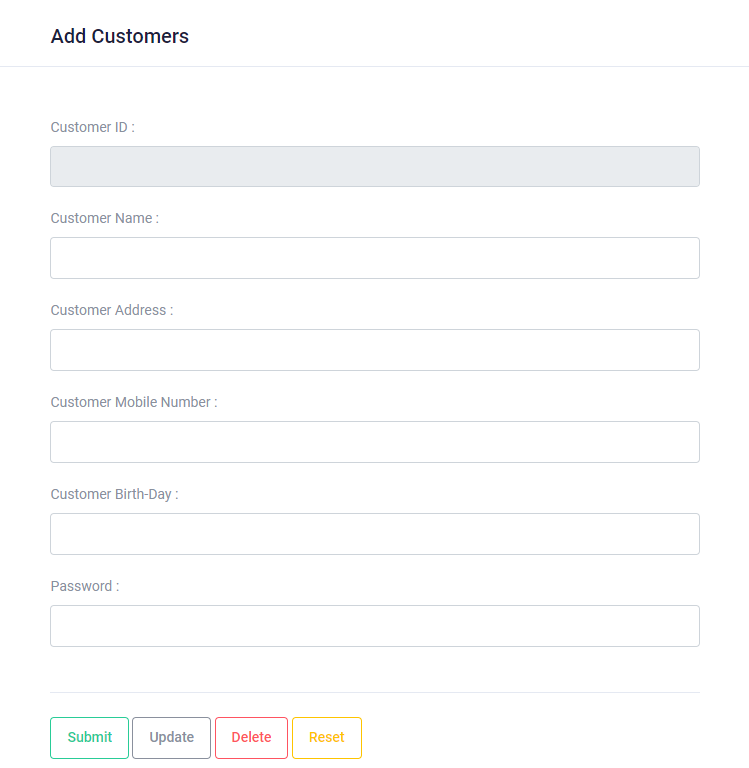


Figure Add Customer Front-End

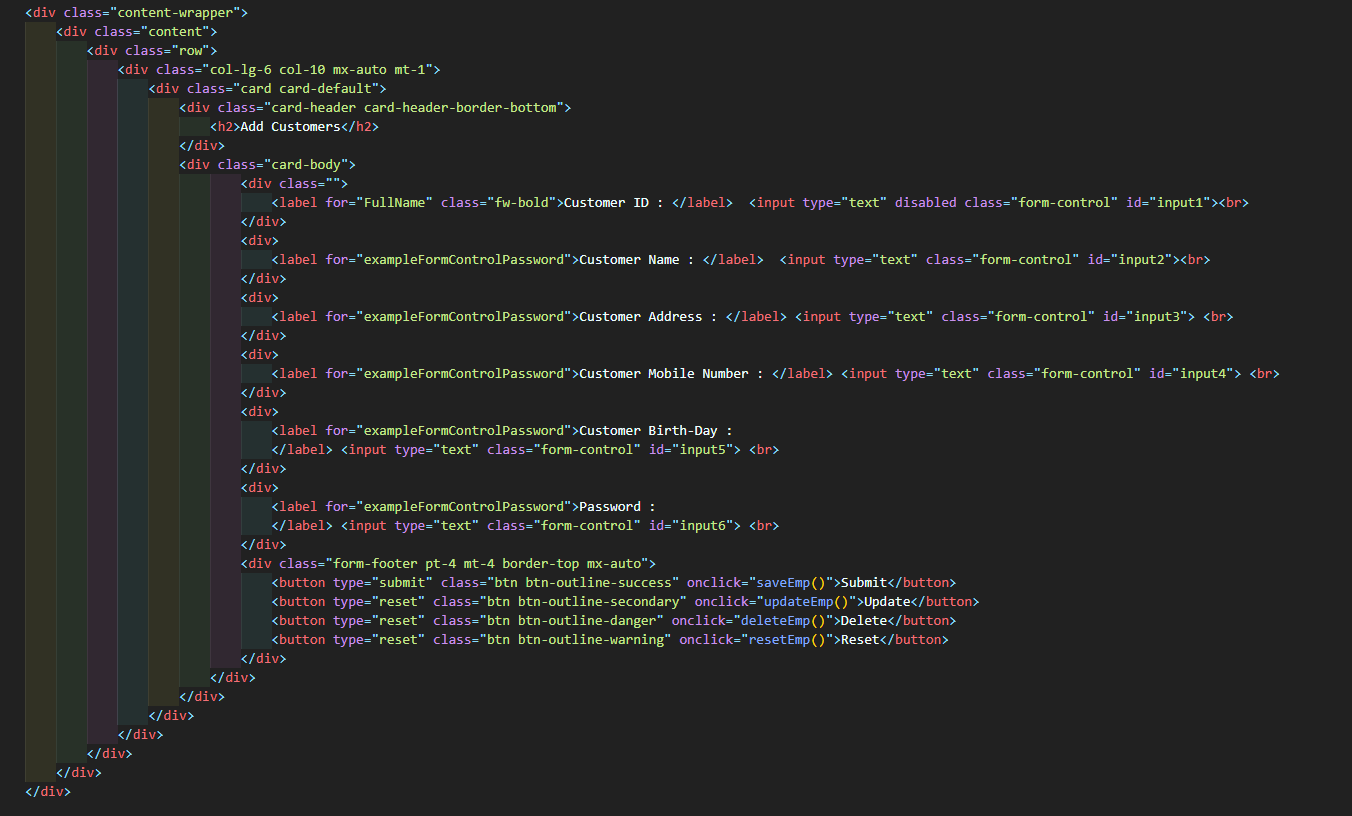


Figure Add Customer Front-End Code



Figure Add Customer API Connected Code

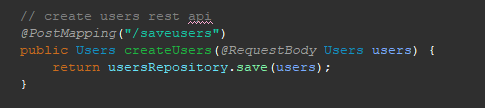


Figure Add Customer Spring Bot Code

The "**Add customer**" function is a feature commonly found in customer relationship management (CRM) software or e-commerce websites. It allows businesses to create and store customer records for individuals who interact with their company.

Here's an overview of how the "add customer" function typically works:

Users may use the "add customer" option by entering into their accounts on the CRM application or e-commerce website, then going to the customer management area and selecting "Add Customer." Enter the client's details here: Information about the customer must be entered into the system by the user. The name, contact details (such as a phone number and email address), shipping address, and any other pertinent data of the customer may be included. Save customer information: The user can save the record to the CRM program or an e-commerce website after entering the customer's details. As a result, a client profile will be created that may later be accessed and modified. Add categories or tags: The user may be able to assign categories to their client data if they want to categorize them.

1. **Update Customer**

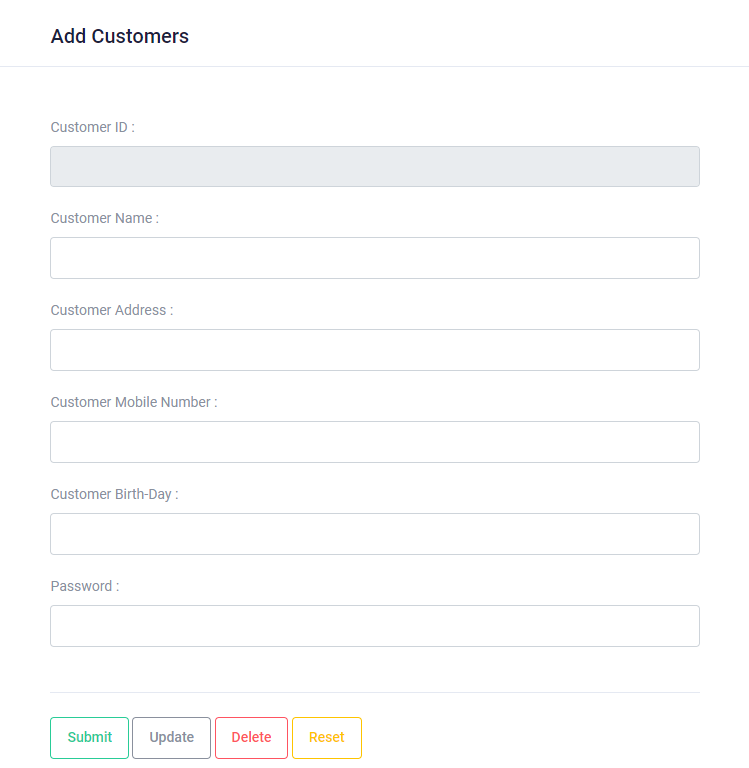


Figure Update Customer Front-End

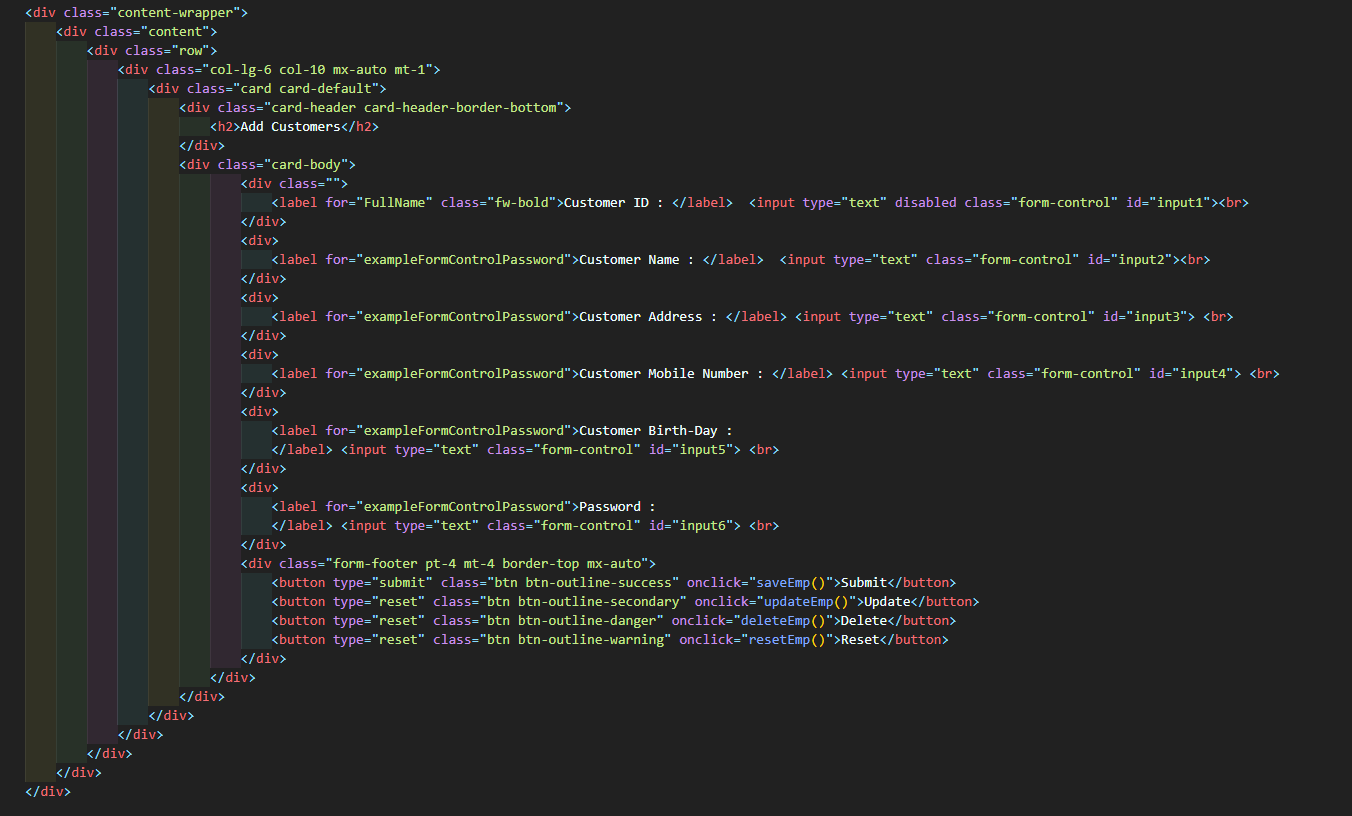


Figure Update Customer Front-End Code

Figure Update Customer API Connected Code

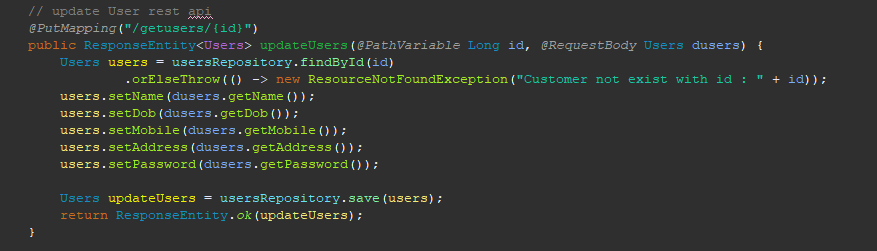


Figure Update Customer Spring-Bot Code

The "**Update Customer**" function is a feature commonly found in customer relationship management (CRM) software or e-commerce websites. It allows businesses to modify and update the information stored in a customer's profile.

Here's an overview of how the "update customer" function typically works:

By login into their accounts on the CRM program or e-commerce website and going to the customer management area, users can use the "update customer" option. Choose the client to update: From the list of current customers, the user must choose the client they wish to update. Client information may be changed as necessary when the user has chosen the customer whose information they wish to update. This might involve amending the client's address or payment information, adding notes or tags to the customer record, or updating contact information. Save the changes: After making the necessary adjustments, the user may save the changes to the customer record in the CRM program or on the online store.

1. **Delete Customer**

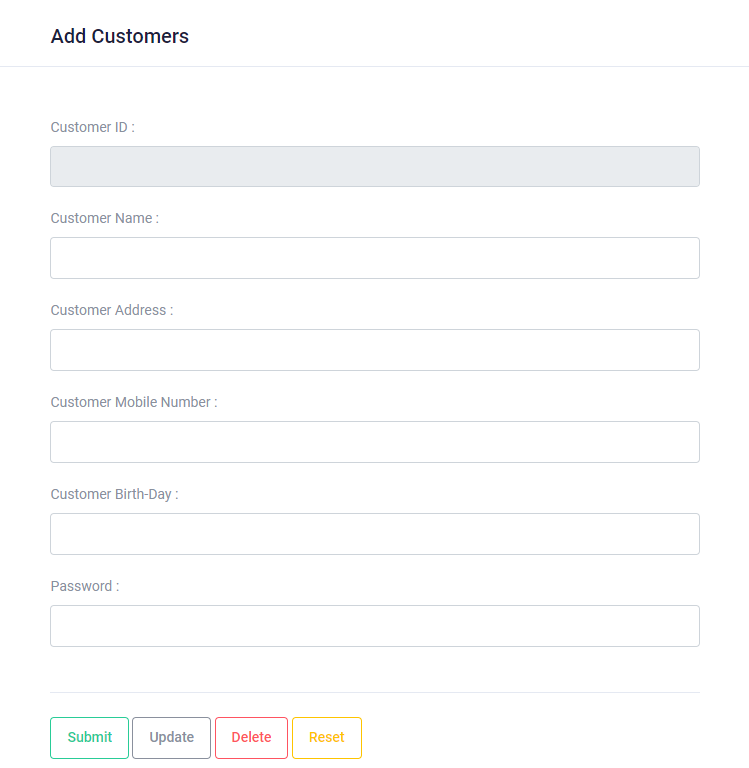


Figure Delete Customer Front-End

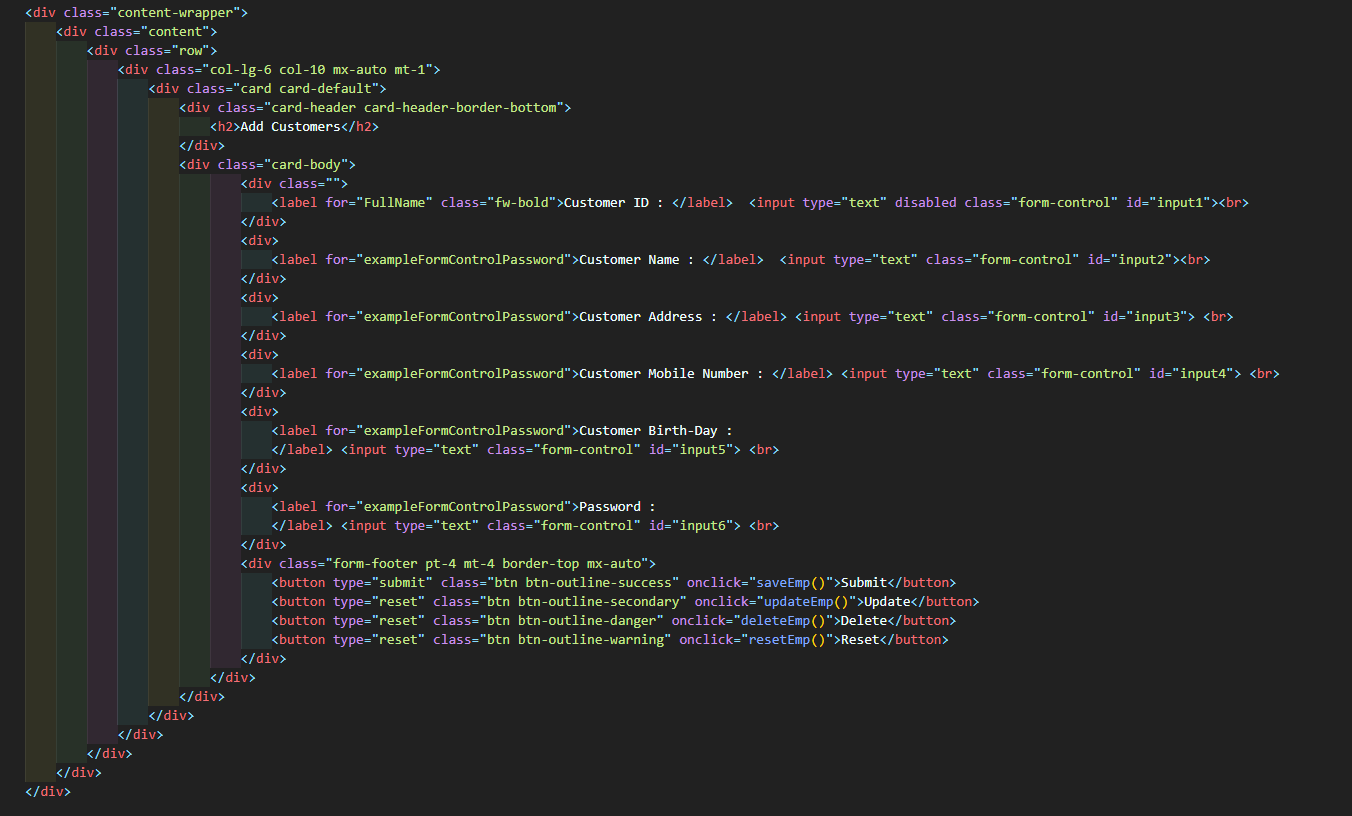


Figure Delete Customer Front-End Code

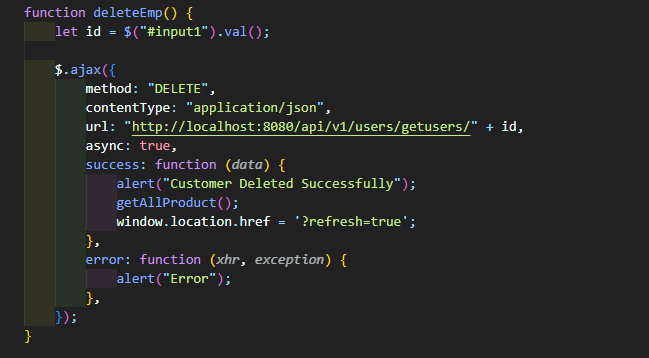


Figure Delete Customer API Connected Code

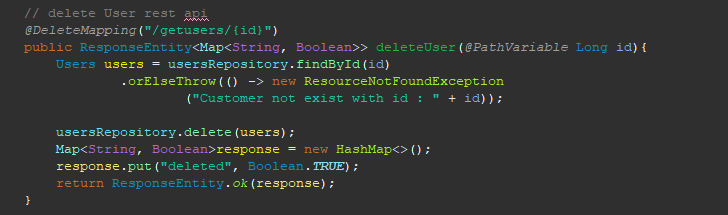


Figure Delete Customer Spring-Bot Code

The "**Delete Customer**" function is a feature commonly found in customer relationship management (CRM) software or e-commerce websites. It allows businesses to remove customer records from their system.

Here's an overview of how the "delete customer" function typically works:

By login into their account on the CRM application or e-commerce website and going to the customer management area, users can use the "remove customer" option. Choose the client you want to remove: From the list of current customers, the user must choose the one they wish to eliminate. Confirm the deletion: The user will normally be required to confirm that they want to remove the customer record after selecting the client they wish to delete. This is done to stop client records from being accidentally deleted. Delete the customer record: The customer record will be deleted from the CRM program or e-commerce website if the user confirms their desire to do so. Examine and confirm the

1. **Read Customer**

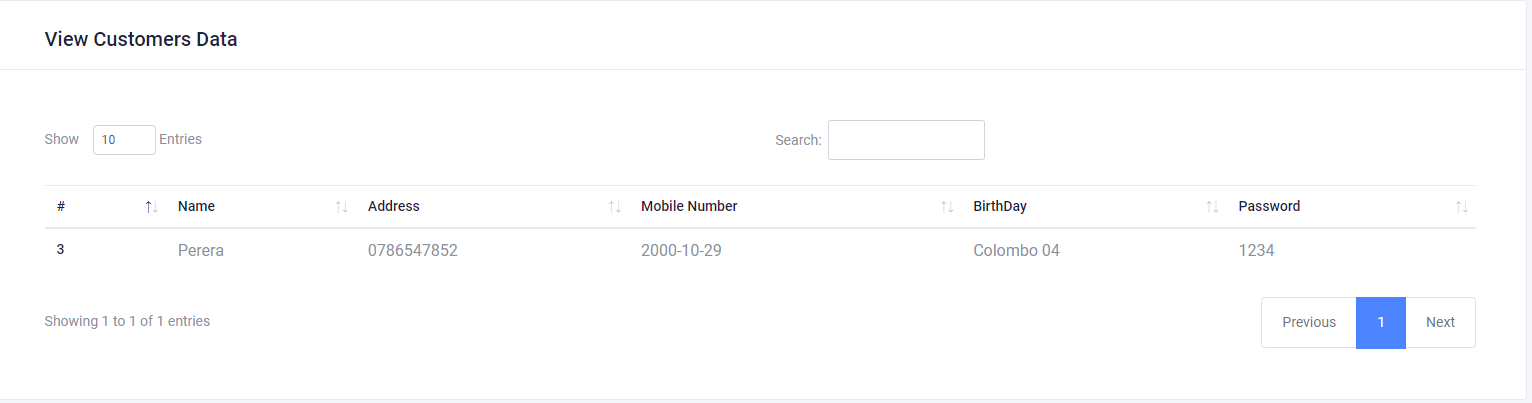


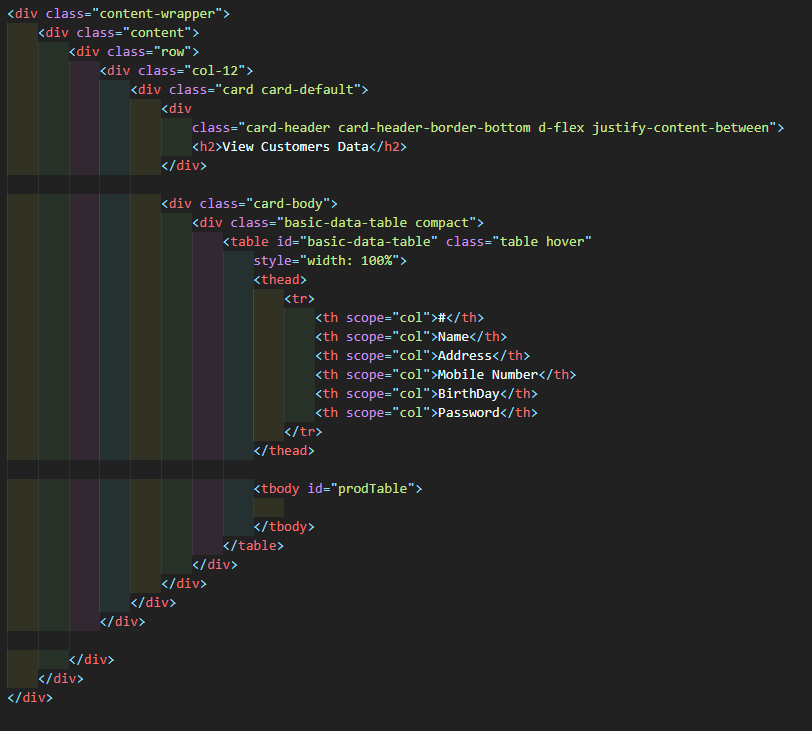
Figure View Customer Front-End

Figure View Customer Front-End Code



Figure View Product API Connected Code

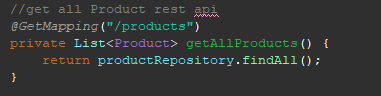


Figure View Product Spring-Bot Code

The "**Read customer**" function is a feature commonly found in customer relationship management (CRM) software or e-commerce websites. It allows businesses to view the information stored in a customer's profile.

Here's an overview of how the "read customer" function typically works:

By entering into their account on the CRM application or e-commerce website and going to the customer management area, users can use the "read customer" function. To view the customer, choose: From the list of current customers, the user must choose the client they wish to view. Customer data viewable the user will be able to see the data kept in the customer's profile once they have chosen the customer they wish to view. Name, contact details, shipping address, previous purchases, and any other pertinent data about the consumer may be included. if required, update client information The "edit customer" feature allows the user to make changes to the customer's profile if they find any mistakes or out-of-date information.

1. **Read Instalment Details**

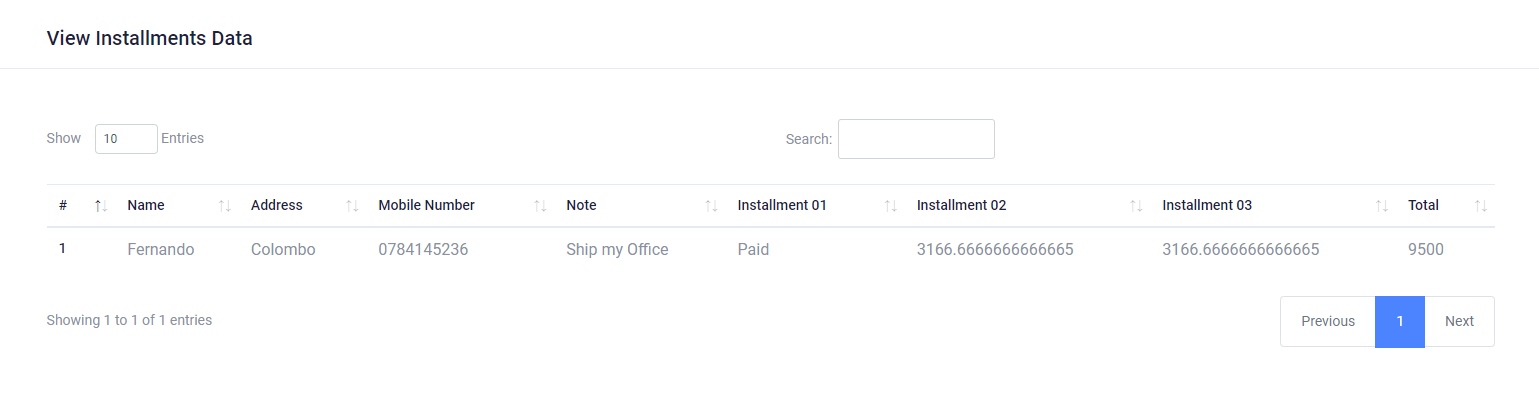


Figure View Installment Plan Font-End

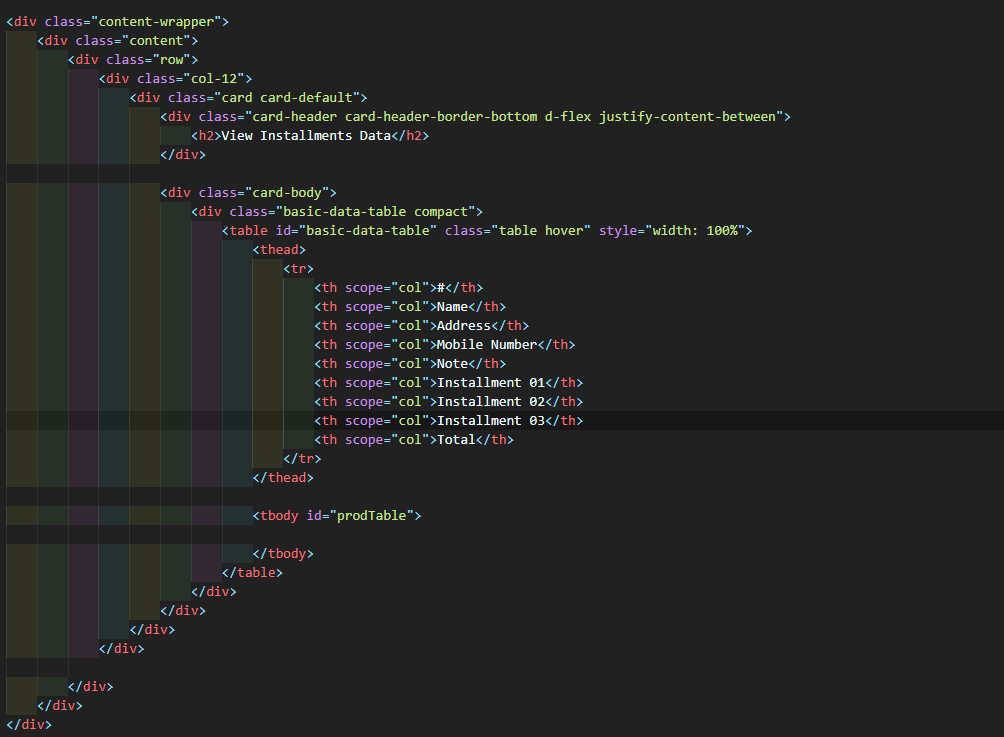


Figure View Installment Front-End Code

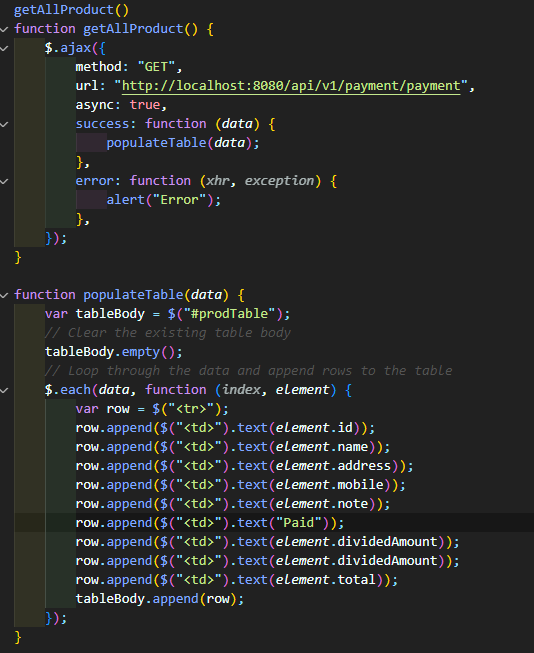


Figure View Installment API Connected Code

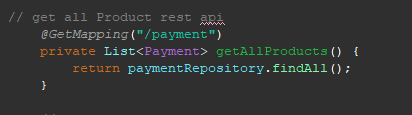


Figure View Installment Spring-Bot Code

The "**Read Instalment Details**" function is a feature commonly found in instalment payment systems used by e-commerce websites or other businesses that allow customers to pay for products or services in installments. This function allows businesses to view the details of the installment payments made by a customer.

Here's an overview of how the "read installment details" function typically works:

Logging into their account on the installment payment system and going to the installment management area will allow users to access the "read installment details" option. Choose the client and the payment schedule: The client whose installment information the user wishes to view must be chosen, as well as the particular payment plan related to the customer's account. View information about installment payments: The user will be able to view the specifics of each installment payment made by the customer after choosing the customer and installment plan. This could include the due date, payment sum, payment status, and any fees or penalties that might be owed. Take action based on the installment payment details: After reviewing the installment payment details, the user may

1. **Customer Registration**

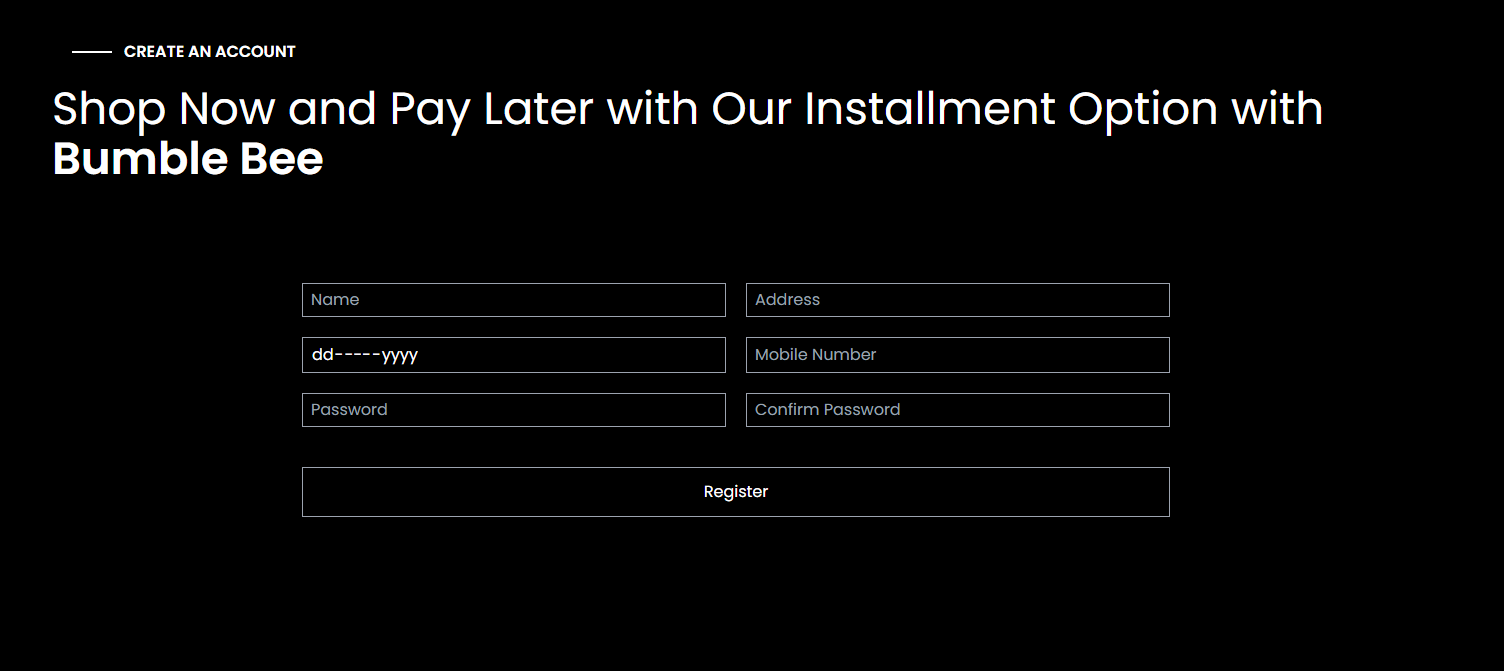


Figure Register Front-Code

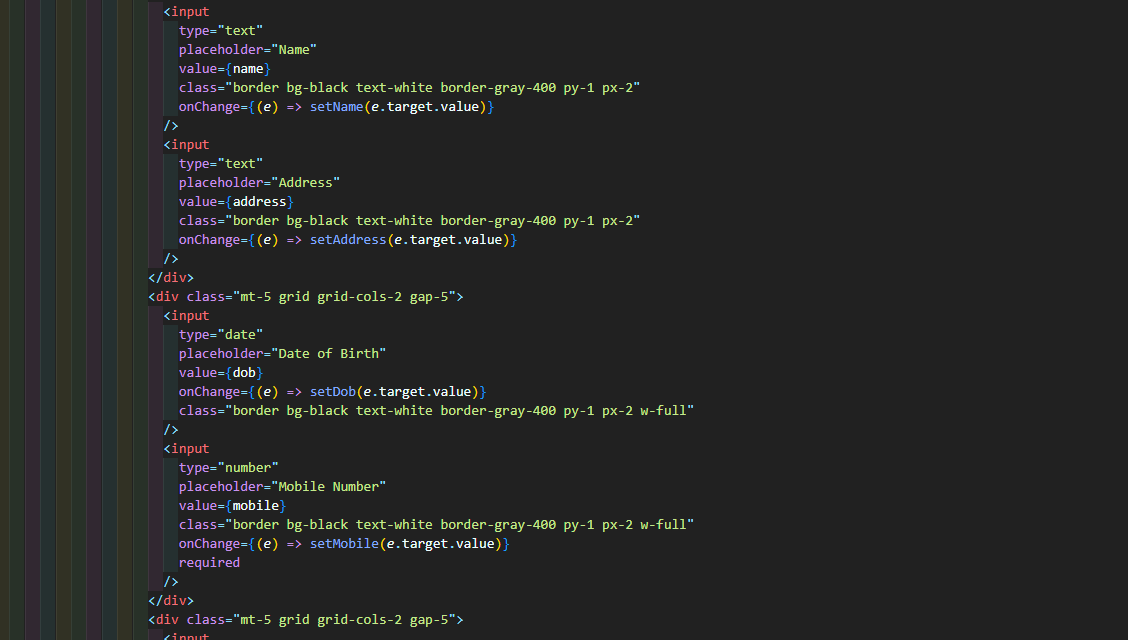


Figure Register Front-End Code

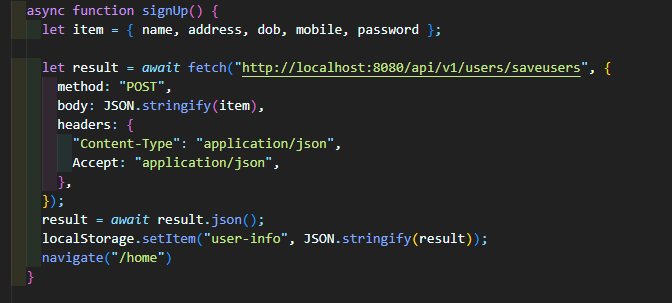


Figure Register API Connected Code

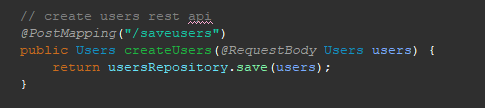


Figure Registered Spring-Bot Code

The "**Customer registration**" function is a feature commonly found on e-commerce websites or other online platforms that allow customers to create an account to access certain features or services. Here's an overview of how the customer registration function typically works:

The registration page can be accessed by customers either by going straight to the registration page or by clicking on a "Register" or "Sign up" button on the website. Customers must submit certain personal information, such as their name, email address, and password, in order to create an account. Verify their email address: Before users can access their accounts on many websites, they must verify their email address. Usually, this is accomplished by sending an email including a link to confirm the email address to the address entered during registration.

Customers may personalize their accounts by providing extra information, such as their shipping address, billing information, or communication preferences, after having their email address verified. Access their account: After completing the website's registration process, customers can access their accounts. They could then be able to access other options accessible to registered users, examine their order history, or add products to a wish list.

1. **Over 18 Installment Plan**

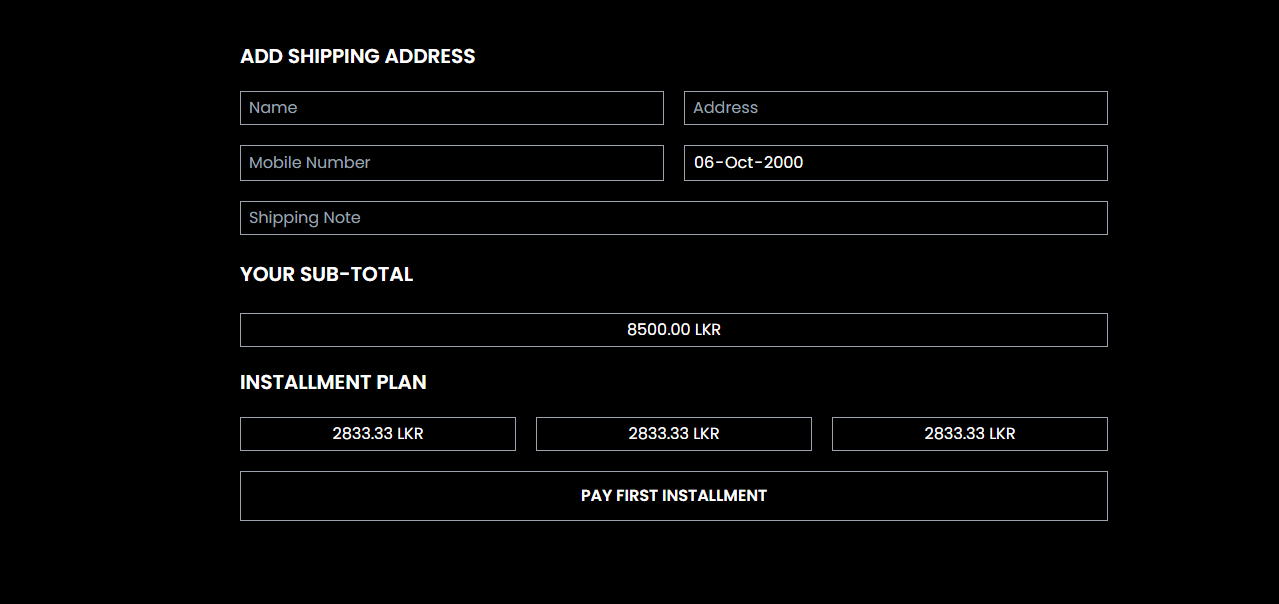


Figure Over 18 Installment Front-End Code

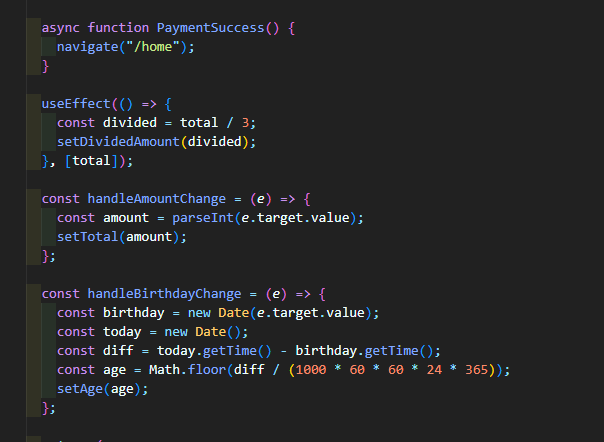


Figure Over 18 Installment Divided amount code

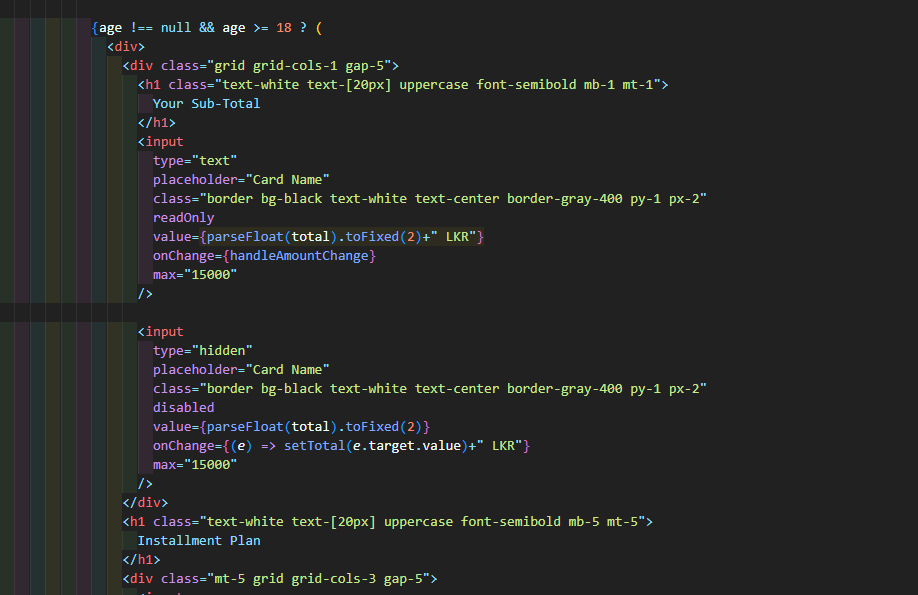


Figure Over 18 Installment Front-End Code

The "**Over 18 Installment Plan**" function is a feature commonly found in installment payment systems used by e-commerce websites or other businesses that allow customers to pay for products or services in installments. This function is typically designed to ensure that customers who use the installment plan are legally eligible to enter into a binding contract.

Here's an overview of how the "over 18 installment plan" function typically works:

Customer submits age information: When choosing the payment plan option, a customer can be prompted to enter their date of birth or other details to verify that they are older than 18 years old. System confirms age-related data: The customer's age may be verified via the installment payment system in a number of ways, including by comparing it to public records or by employing a third-party verification service. eligibility was established: The consumer will be qualified to utilize the payment plan if their age is confirmed and it is found that they are older than 18 years old. If the customer's age cannot be confirmed or it is found that they are under 18, they could not be able to use the payment plan. Installment

1. **Admin Login**

****

Figure Admin Login Front-End

****

Figure Admin Login Front-End Code

**The Admin Login Function** is a common feature in web applications and is used to provide restricted access to certain functionalities or resources that are only available to authorized users. Here is an overview of how admin login function typically works:

Admin login page: The first step is to create an admin login page where the admin user can enter their login credentials, such as a username and password.

Authentication: The program should authenticate the user by verifying the entered credentials with the values held in a database or another sort of data store when the admin user enters their login credentials. The program should establish a session or a token to identify the authorized user if the credentials are correct. Redirect to admin dashboard: The application should refer the user to the admin dashboard or the required page if the admin user has permission to access the requested resources or functionality. The program should show an error message or reroute the user to a different page if the user is not allowed. The last step is to enable a logout feature that enables the admin user to terminate their session and logout.

# **Task E**

## **Test Plan and test-driven development**

1. **Customer Unit Testing**

****

Figure Unit Testing Customer

1. **Product Unit Testing**

****

Figure Unit Testing Product

1. **Payment Unit Testing**

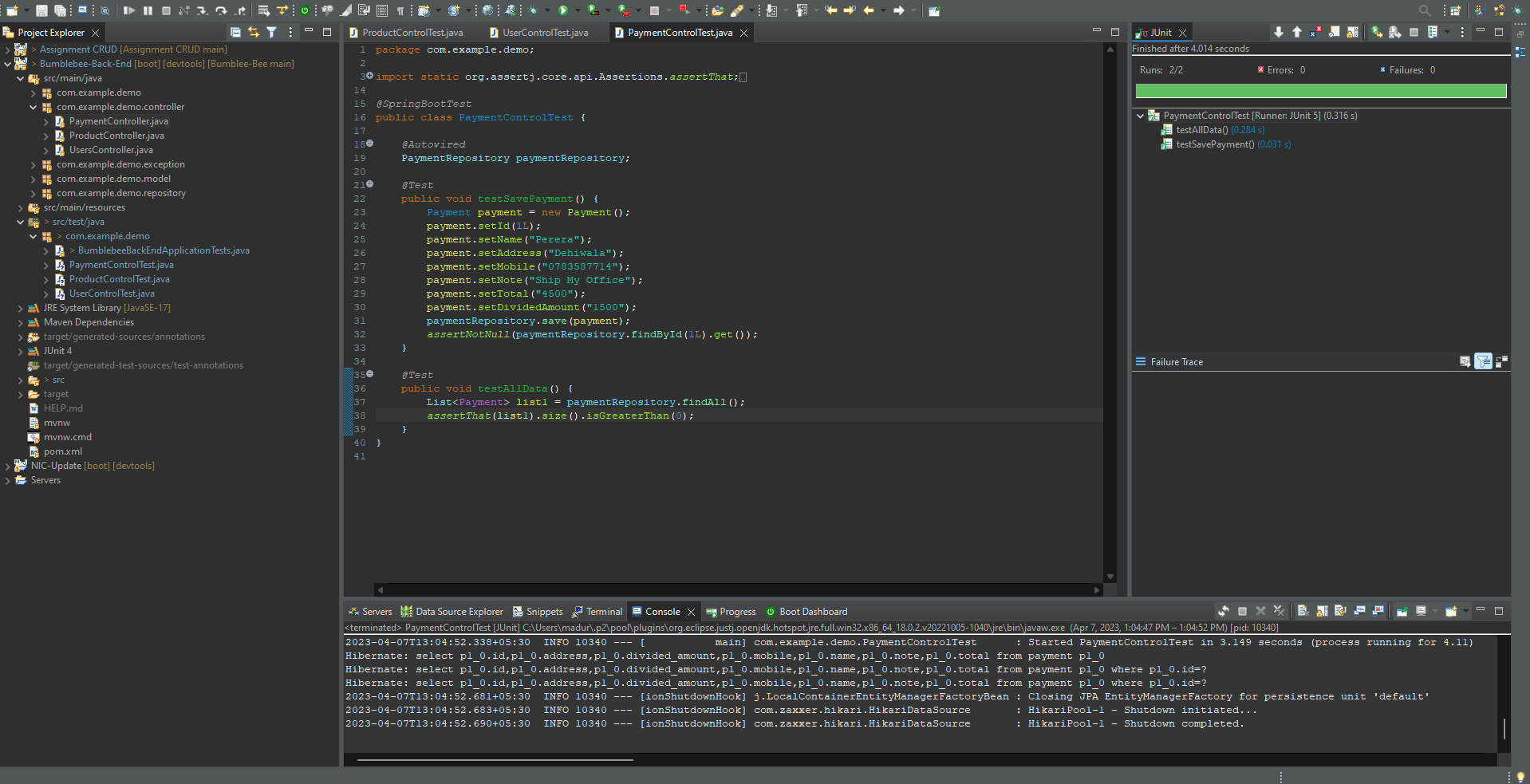
****

Figure Unit Testing Payment

## **Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Case Name** | **Scenario** | **Expected Results** |
| TC 01 | Checking Login Function Works. | Enter password For Admin. | Login Admin Panel Successfully. |
| TC 02 | Check the Login Work. | Enter Wrong Password for Admin. | Show Alert Message. |
| TC 03 | Check the Add Product Working. | Enter Product Details. | Successfully Added Product. |
| TC 04 | Check the Update Product Working. | Enter Product Update details. | Successfully Updated Product. |
| TC 05 | Check the Delete Product Working. | Find and Delete Product. | Successfully Deleted Product. |
| TC 06 | Check the Add Customer Working. | Enter Customer Details. | Successfully Added Customer. |
| TC 07 | Check the Update Customer Working. | Enter Customer Update details. | Successfully Updated Customer. |
| TC 08 | Check the Delete Customer Working. | Find and Delete Customer. | Successfully Deleted Customer. |
| TC 09 | Check the Payment Installments Working. | View Payment Installments. | Successfully Viewed Payment Installment. |
| TC 10 | Check the Customer Register Working. | Enter the Customer Details. | Successfully Redirect Bumble bee home page. |
| TC 11 | Check Product API working. | Home Page Product Display. | Successfully Displayed Products. |
| TC 12 | Check Product View Function. | Display Product Description Details. | Successfully Displayed Product Details. |
| TC 13 | Check Product Add Cart Function. | Display Cart Sidebar Add Item. | Successfully Displayed Cart Add Items. |
| TC 14 | Check Product Increase Function Cart Working. | Display Product Increase and Total Increase. | Successfully Displayed Product Increase and Total. |
| TC 15 | Check Product Remove Function on Cart. | Remove the Cart Item. | Successfully Remove the Products. |
| TC 16 | Check Checkout Page Working. | Display Cart details on Checkout page. | Successfully Displayed Checkout Page Cart Details. |
| TC 17 | Check Customer Enter Date of Birth calculate Age Function Working. | Enter the Birthday. | Successfully Displayed. |
| TC 18 | Check Customer Enter Age Under 18 Birthday. | Enter th Birthday. | Successfully Displayed Pay Now Option. |
| TC 19 | Check Customer Enter Age Over 18 Birthday. | Enter the Birthday. | Successfully Displayed Installment Details. |
| TC 20 | Check Customer Enter Age Over 18 Birthday total Amount Divide Prices. | Enter the Birthday. | Successfully Displayed Installments Amount Details. |
| TC 21 | Check Pay Now Button Working. | Checkout Page Click Pay Now Button. | Successfully Data Added On database. |

## **Test Cases**

|  |  |
| --- | --- |
| Test Case ID | TC 01 |
| Test Objective | Checking Login Function Works. |
| Test Scenario | Enter password For Admin. |
| Expected Result | Login Admin Panel Successfully. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 02 |
| Test Objective | Check the Login Work. |
| Test Scenario | Enter Wrong Password for Admin. |
| Expected Result | Show Alert Message. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 03 |
| Test Objective | Check the Add Product Working. |
| Test Scenario | Enter Product Details. |
| Expected Result | Successfully Added Product. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 04 |
| Test Objective | Check the Update Product Working. |
| Test Scenario | Enter Product Update details. |
| Expected Result | Successfully Updated Product. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 05 |
| Test Objective | Check the Delete Product Working. |
| Test Scenario | Find and Delete Product. |
| Expected Result | Successfully Deleted Product. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 06 |
| Test Objective | Check the Add Customer Working. |
| Test Scenario | Enter Customer Details. |
| Expected Result | Successfully Added Customer. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 07 |
| Test Objective | Check the Update Customer Working. |
| Test Scenario | Enter Customer Update details. |
| Expected Result | Successfully Updated Customer. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 08 |
| Test Objective | Check the Delete Customer Working. |
| Test Scenario | Find and Delete Customer. |
| Expected Result | Successfully Deleted Customer. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 09 |
| Test Objective | Check the Payment Installments Working. |
| Test Scenario | View Payment Installments. |
| Expected Result | Successfully Viewed Payment Installment. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 10 |
| Test Objective | Check the Customer Register Working. |
| Test Scenario | Enter the Customer Details. |
| Expected Result | Successfully Redirect Bumble bee home page. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 11 |
| Test Objective | Check Product API working. |
| Test Scenario | Home Page Product Display. |
| Expected Result | Successfully Displayed Products. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 12 |
| Test Objective | Check Product View Function. |
| Test Scenario | Display Product Description Details. |
| Expected Result | Successfully Displayed Product Details. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 13 |
| Test Objective | Check Product Add Cart Function. |
| Test Scenario | Display Cart Sidebar Add Item. |
| Expected Result | Successfully Displayed Cart Add Items. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 14 |
| Test Objective | Check Product Increase Function Cart Working. |
| Test Scenario | Display Product Increase and Total Increase. |
| Expected Result | Successfully Displayed Product Increase and Total. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 15 |
| Test Objective | Check Product Remove Function on Cart. |
| Test Scenario | Remove the Cart Item. |
| Expected Result | Successfully Remove the Products. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 16 |
| Test Objective | Check Checkout Page Working. |
| Test Scenario | Display Cart details on Checkout page. |
| Expected Result | Successfully Displayed Checkout Page Cart Details. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 17 |
| Test Objective | Check Customer Enter Date of Birth calculate Age Function Working. |
| Test Scenario | Enter the Birthday. |
| Expected Result | Successfully Displayed. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 18 |
| Test Objective | Check Customer Enter Age Under 18 Birthday. |
| Test Scenario | Enter th Birthday. |
| Expected Result | Successfully Displayed Pay Now Option. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 19 |
| Test Objective | Check Customer Enter Age Over 18 Birthday. |
| Test Scenario | Enter the Birthday. |
| Expected Result | Successfully Displayed Installment Details. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 20 |
| Test Objective | Check Customer Enter Age Over 18 Birthday total Amount Divide Prices. |
| Test Scenario | Enter the Birthday. |
| Expected Result | Successfully Displayed Installments Amount Details. |
| Actual Result |  |
| Conclusion | Working Properly. |

|  |  |
| --- | --- |
| Test Case ID | TC 21 |
| Test Objective | Check Pay Now Button Working. |
| Test Scenario | Checkout Page Click Pay Now Button. |
| Expected Result | Successfully Data Added On database. |
| Actual Result |  |
| Conclusion | Working Properly. |

# **Task F**

# **User manual and technical documentation**

## **Admin Panel**

1. **Login**

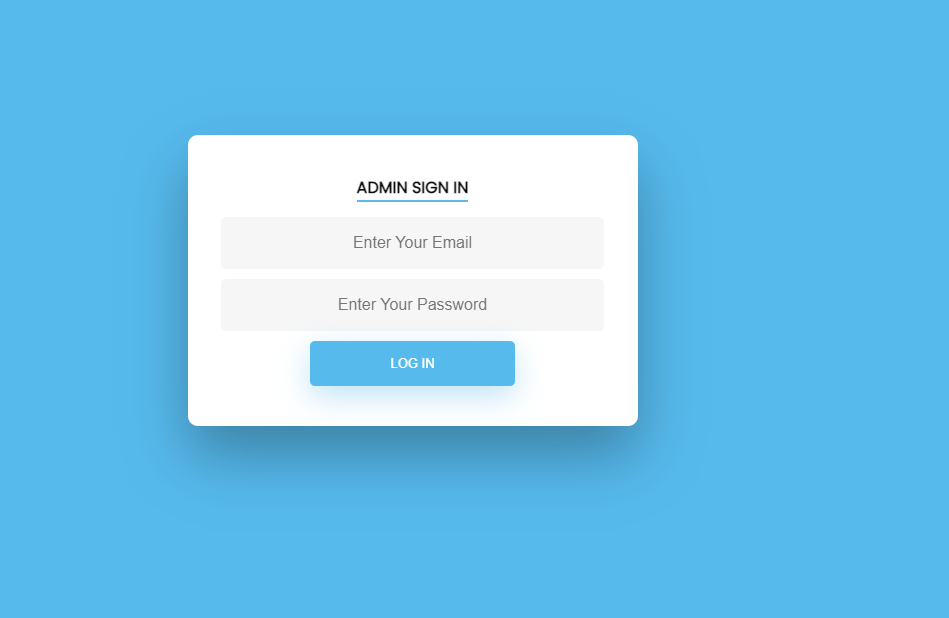


Figure Login User-Manual

* Go to the login page for your e-commerce site by using your web browser.
* Fill up the relevant boxes with the admin username and password. You must use the default login information supplied by your ecommerce platform or pre-configured during installation if this is your first-time checking in.
* Enter your login information and access the admin dashboard by clicking the "Login" button.

Once you are logged in to the admin dashboard, you can perform various functions depending on your access level and permissions.

1. **Admin Panel**

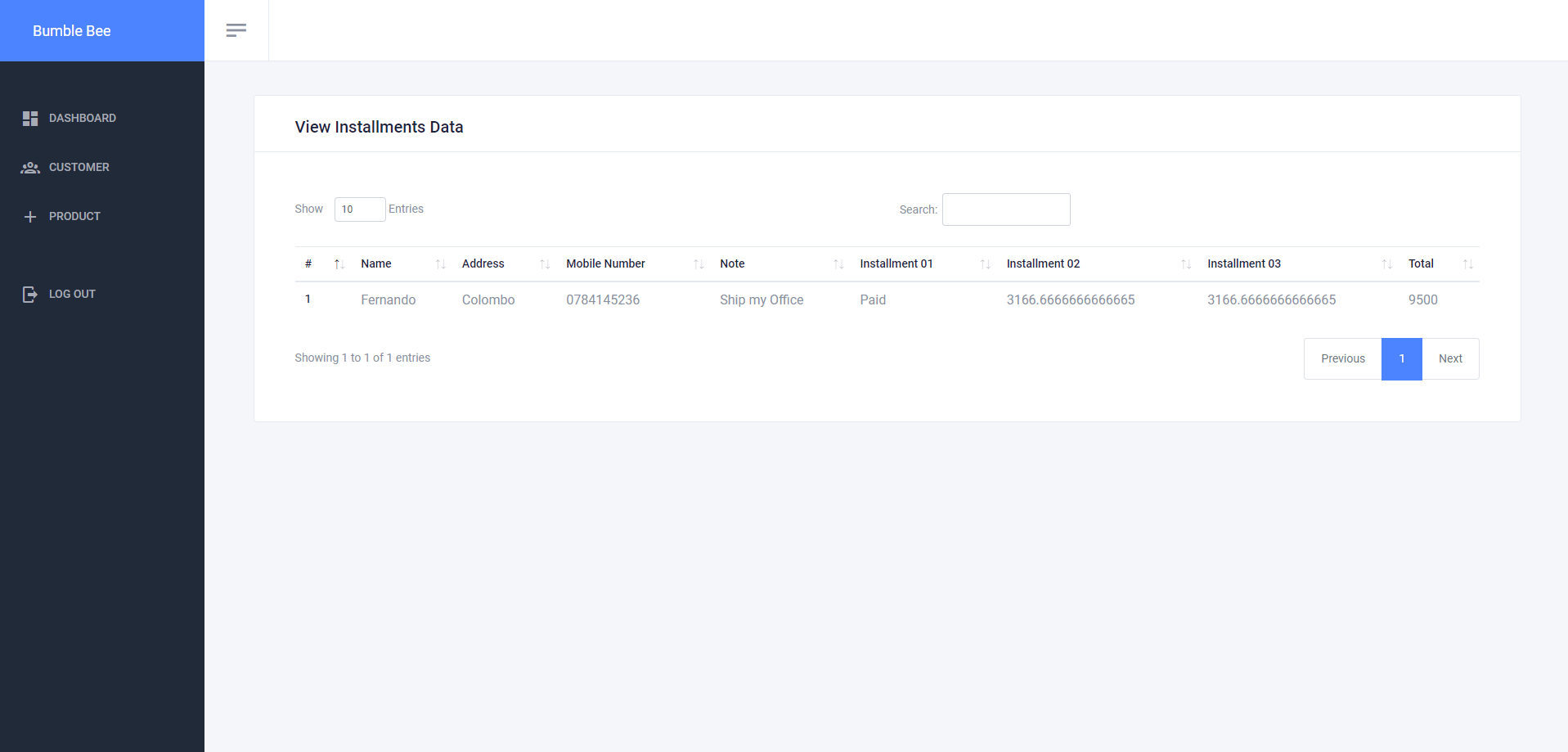
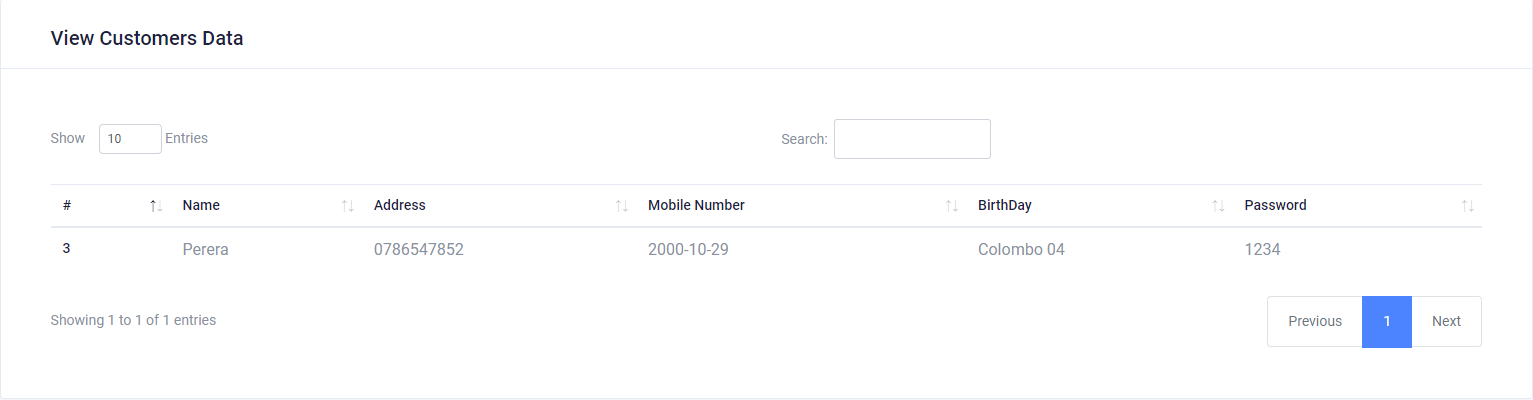


Figure Admin panel user-Manual

* You will be sent to the admin dashboard after entering into the online store.
* The dashboard gives a summary of the site's products, customers, and installations.

1. **Customer Function**

******

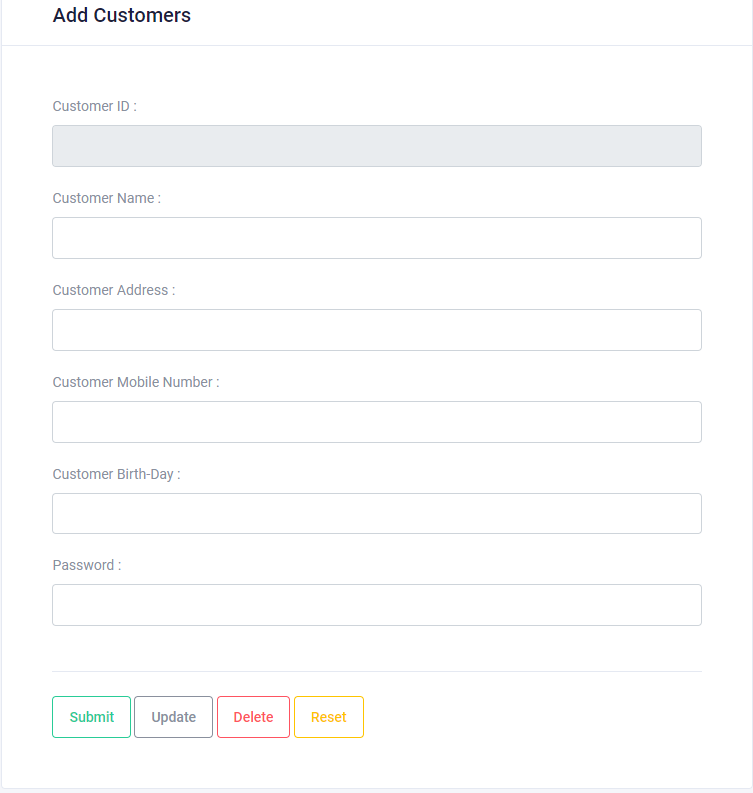


Figure Customer User-Manual

* You will be sent to the admin dashboard after entering into the online store.
* Click the "Customers" option in the navigation bar at the top of the page to manage customers.
* You may see a list of all of your clients here. You may filter the list by name, email, or status or use the search field to look up a specific client.
* Click the "Add Customer" button in the upper right corner to add a new customer. Include the customer's name, email address, phone number, shipping and billing addresses, as well as any other pertinent information.
* Click on the customer in the customer list and make the required adjustments to the customer information to alter an existing customer.
* Click the client in the customer list and choose "Delete" to remove them. Please be aware that eliminating a customer also eliminates all orders linked to that customer.

1. **Product Functions**

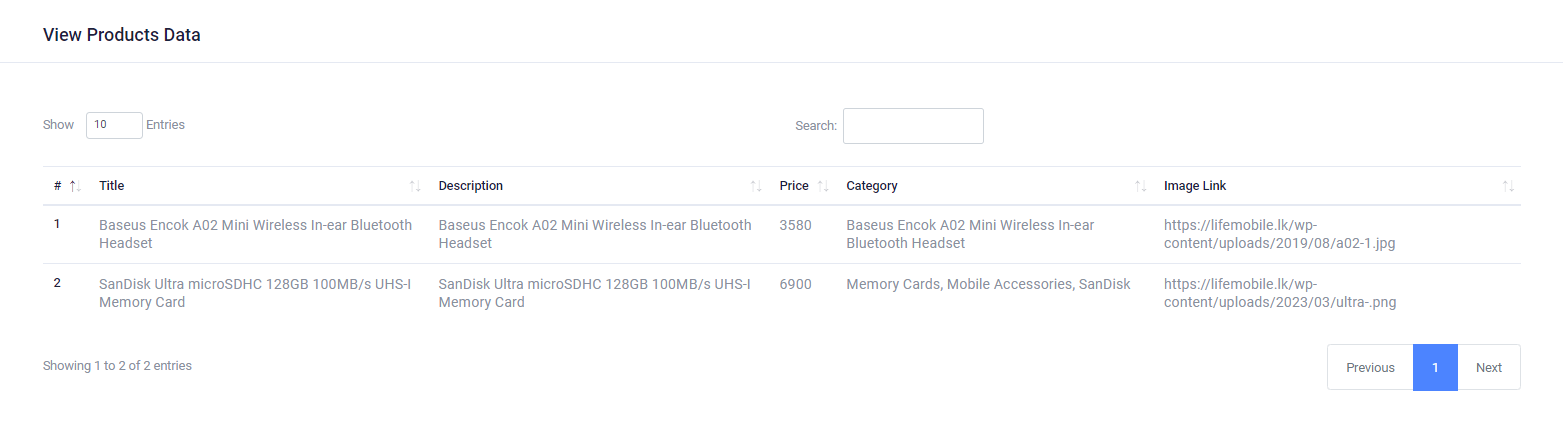
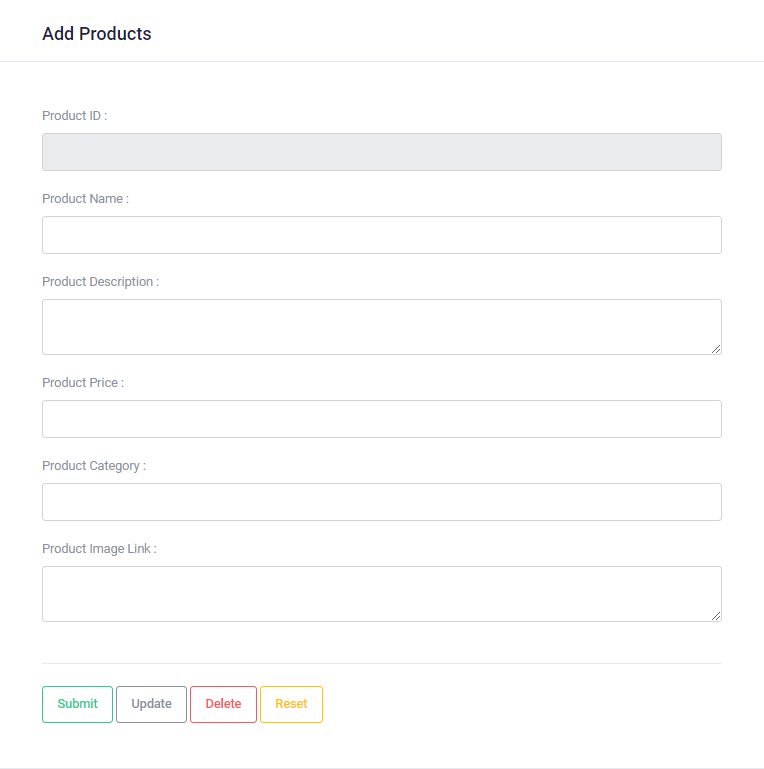
****

Figure Product User-Manual

* You will be sent to the admin dashboard after entering into the online store.
* Click the "Items" area in the navigation bar at the top of the page to manage products.
* View a list of all your items here. Using the search bar, you may narrow down the selection of products by name, category, or status.
* Click on the "Add Product" button in the upper right corner to add a new product. Enter the product's name, description, picture, price, and any further pertinent information. Additionally, you may enter the product's inventory details and choose the product category.
* Click on the item in the product list to edit it, then make the required adjustments to the product's details.
* Click the item in the product list and choose "Delete" to remove it. Please be aware that removing a product from your site permanently removes it from search results.

1. **Customer**

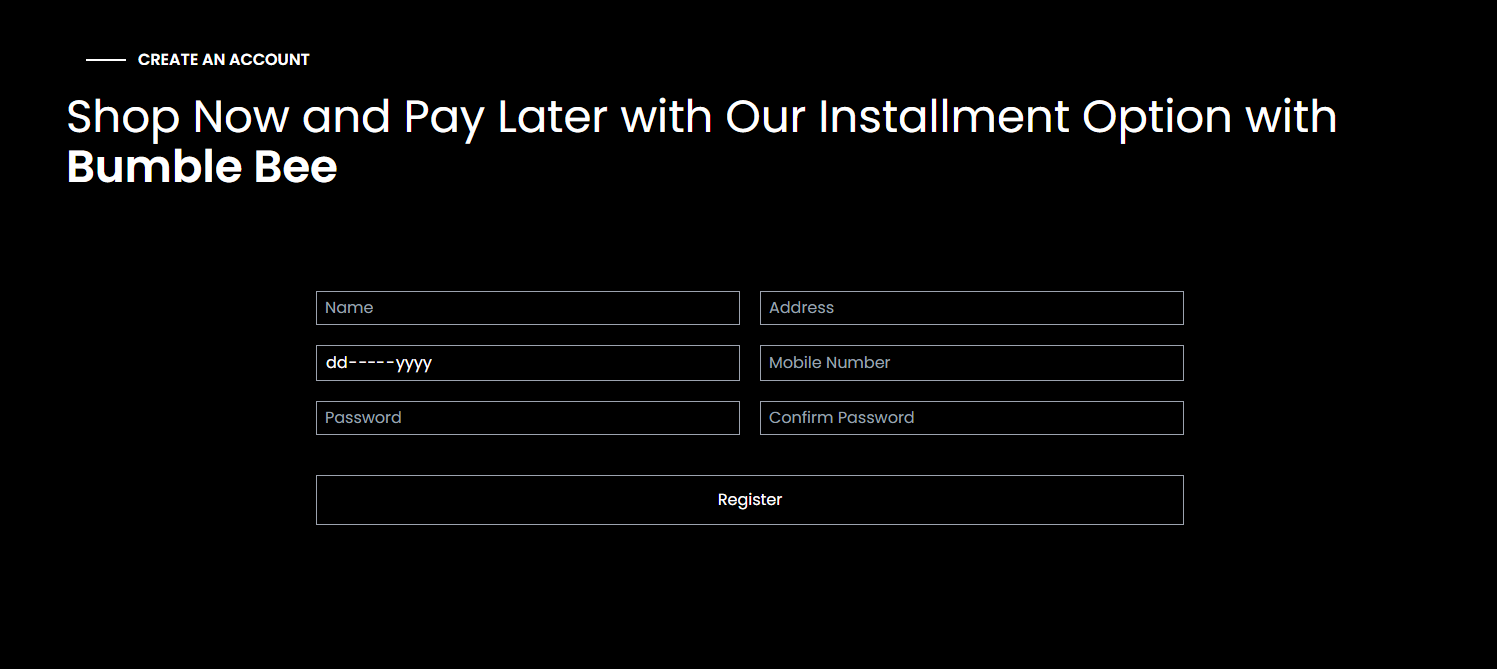


Figure Customer User-Manual

* Click the "Register" or "Sign Up" button located on the e-commerce website's home page.
* This will direct you to the page where customers register. Input the necessary information, including your name, address, date of birth, and password. Select a strong password that is at least 8 characters long, contains both letters and digits, and is unique.
* Fill out any other information, such as the shipping and billing addresses, if there are any.
* To finish the registration procedure, click the "Register" button.

### **Product Page**

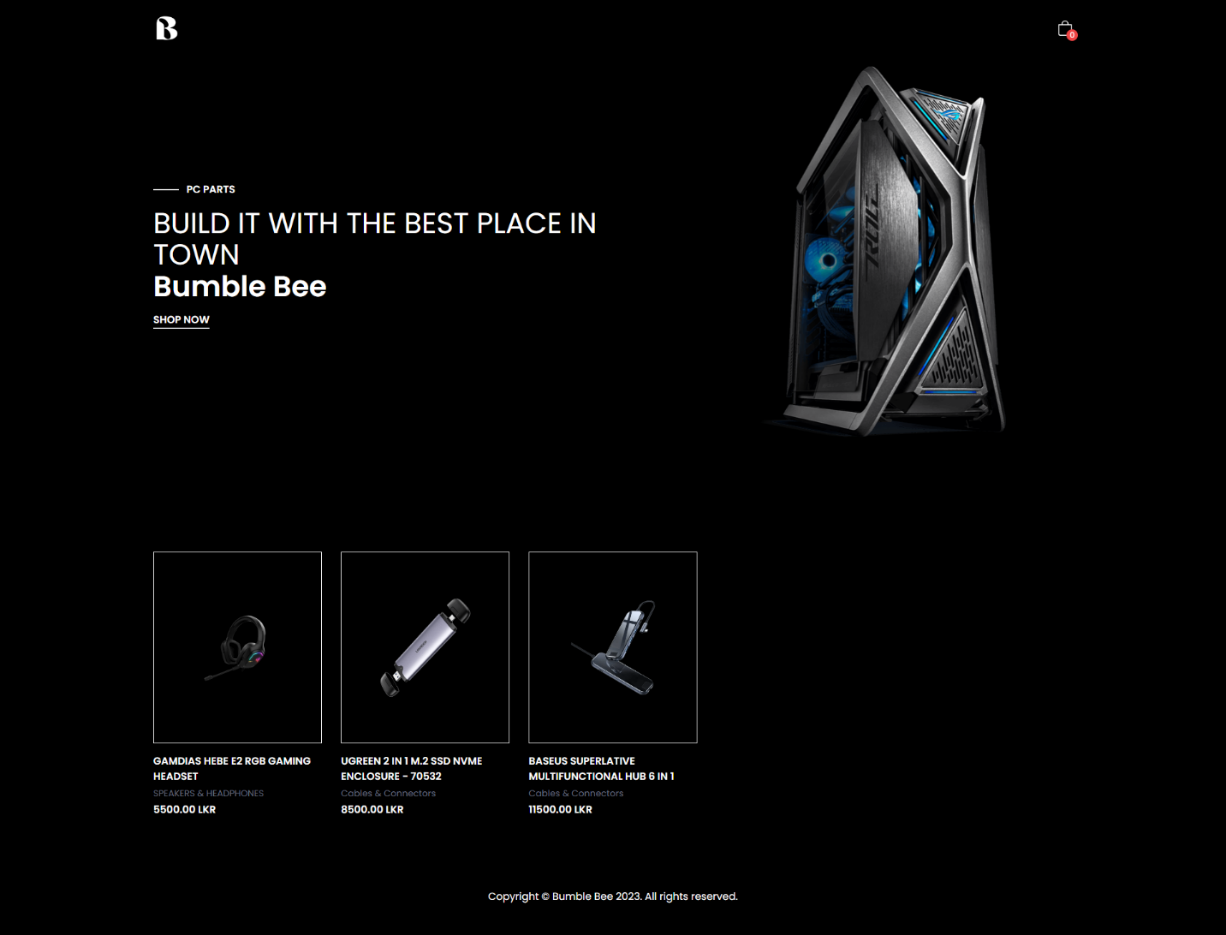


Figure Product User-Manual

* On the homepage of the ecommerce site, you can browse products by category or search for a specific product using the search bar.
* To access the product details page, click on the product title or image.
* You may examine the product picture, description, price, and any other pertinent facts on the product details page.
* To add the item to your cart, click the "Add to Cart" button. You may either go to checkout or keep shopping and add additional items to your cart.
* Click the shopping cart symbol at the top of the page and choose "Checkout" to continue.

### **Cart Page**

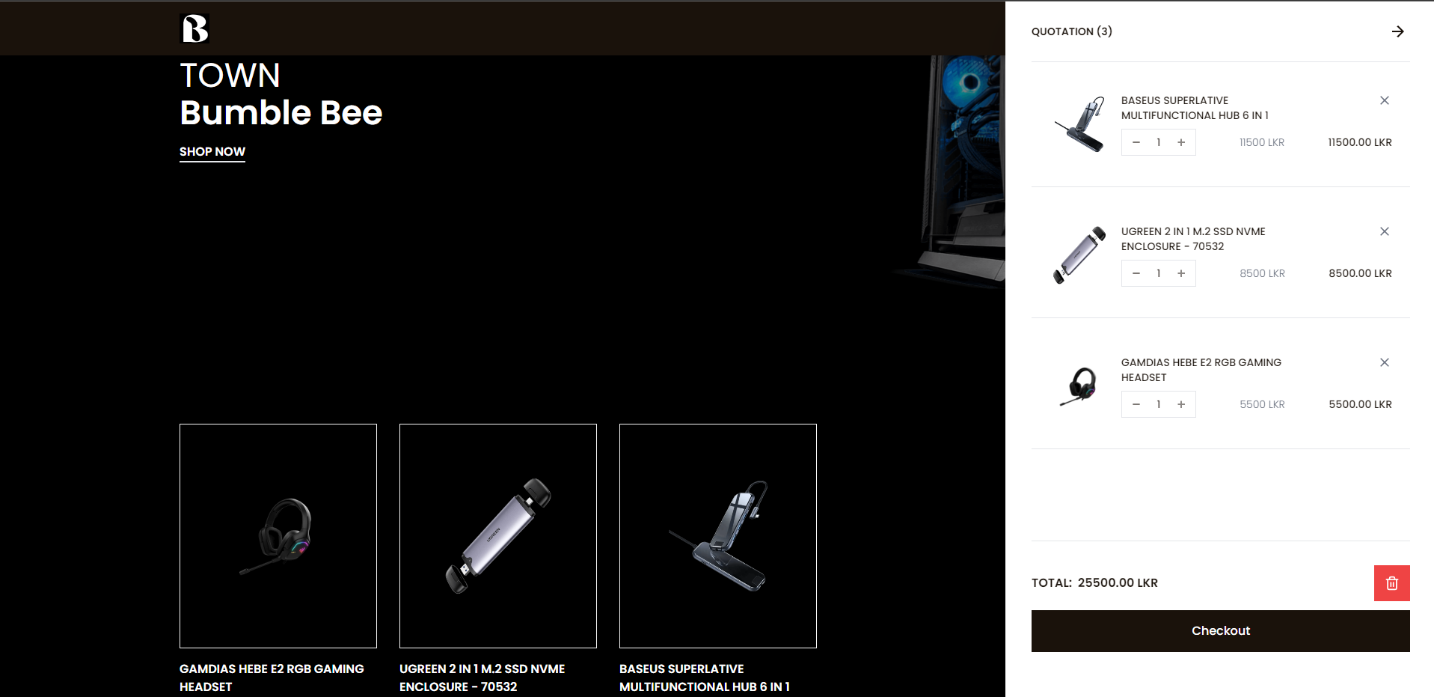


Figure Car User-Manual

* By selecting the "Add to Basket" button on each product page on the e-commerce site's main page, you may add items to your shopping cart.
* When you've finished adding items to your basket, you can view it by clicking the shopping cart symbol at the top of the page.
* You may examine the items you've added to the cart along with their name, quantity, price, and total cost on the cart page.
* Click on the plus or minus buttons next to the quantity area to change the amount of a product. Based on the new amount, the price will update itself automatically.
* Click the "Remove" button next to a product to remove it from your basket.
* Click the "Checkout" button to go to the checkout page and finalize your purchase after making the necessary changes to your basket.

1. **Check-Out Page**

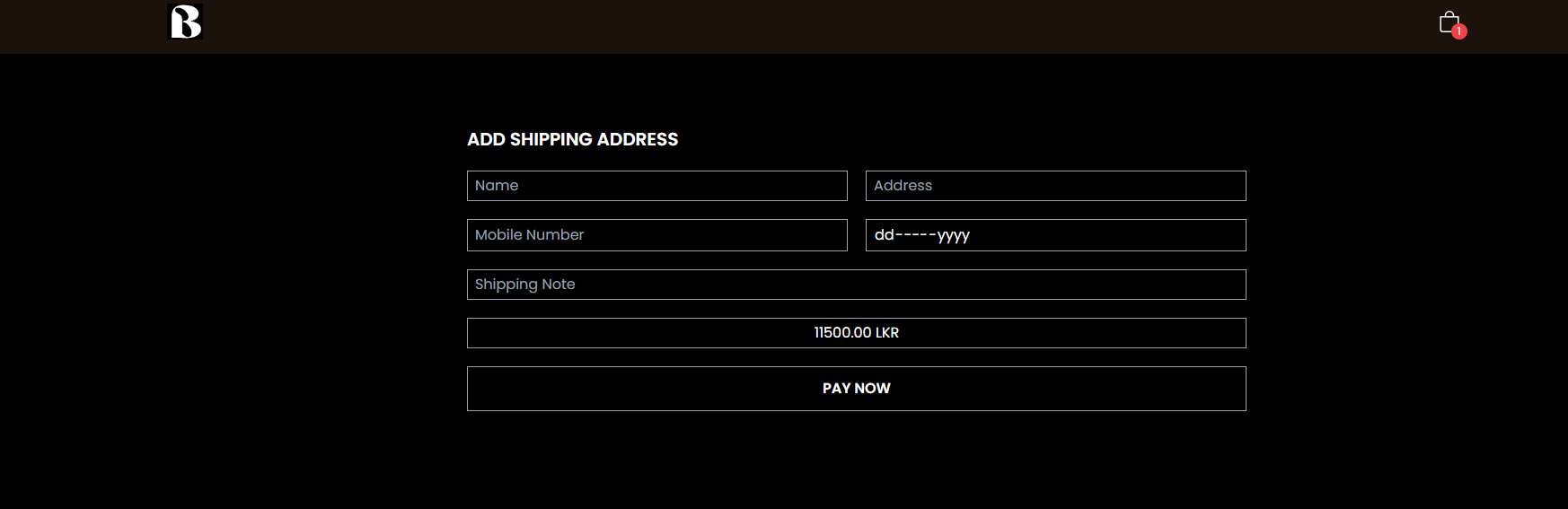
****

Figure Checkout User-Manual

* Review the specifics of your order on the cart page, then click "Checkout" to go to the payment page.
* Enter your shipping information, including your name, address, and contact information, on the checkout page. If you have an account on the online store, you might also be able to choose a stored address.
* Review the estimated delivery cost and choose your desired shipping option.
* Enter your name, address, and payment information along with your billing information. If you have an account, you might also be able to choose to utilize a stored billing address.
* Review the information on your order, including the items you are buying and the overall cost.

1. **Installment Plan Function**

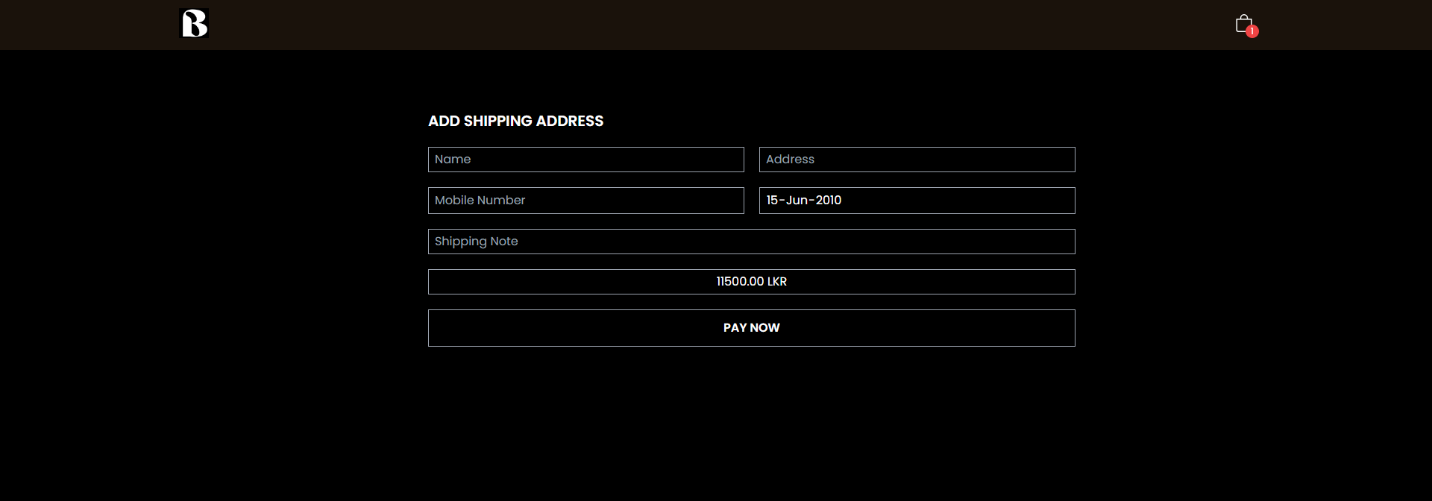
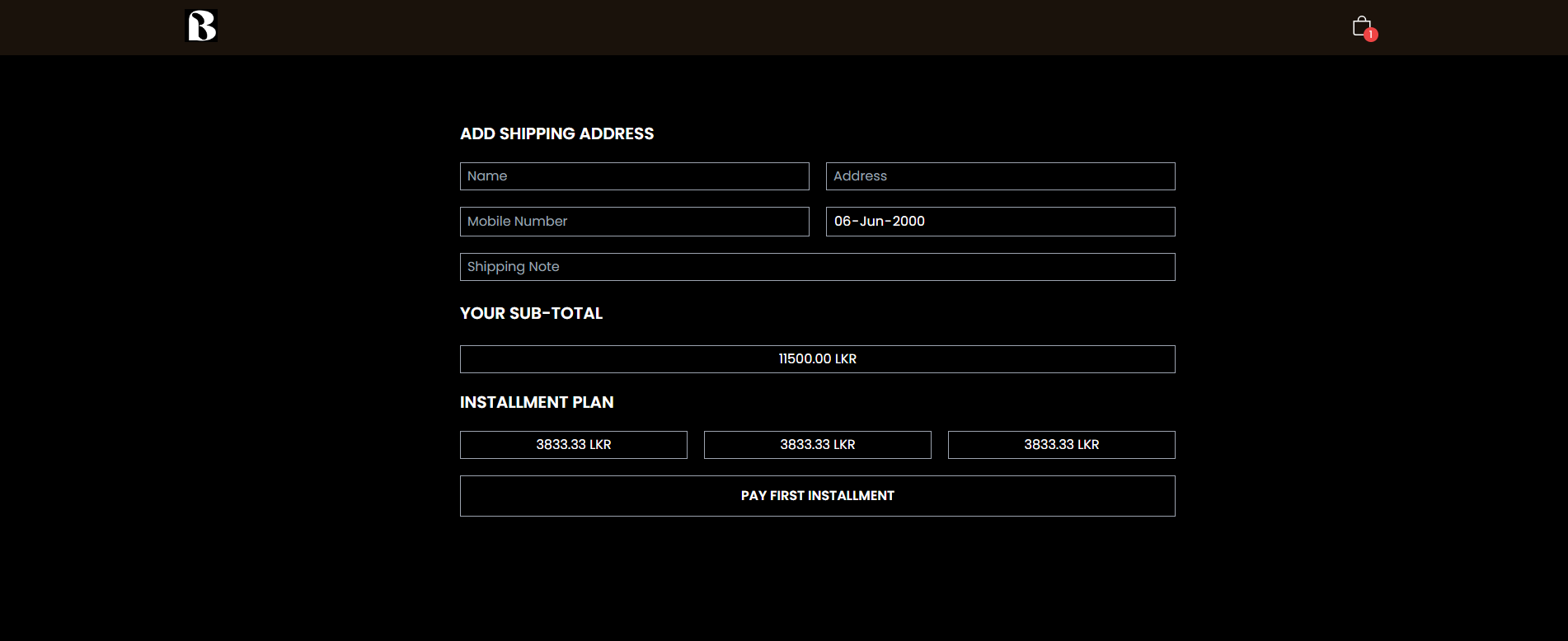
****

Figure Installment User-Manual

* Your birthdate will be requested on the checkout page after your shipping and billing details have been entered.
* You will see two payment options if you are over the age of 18: "Pay Now" and "Installment Plan."
* If you choose "Pay Now," you will be taken to the payment page where you can finish making your purchase.
* You will be asked for more information, such as your income and job status, if you choose "Installment Plan," in order to ascertain your eligibility for the installment plan.
* You will be given a payment schedule, which will include the down payment and following installment payments, if you are approved for the installment plan. A contract explaining the terms and circumstances of the payment plan will also need to be signed.
* You can proceed to finish your purchase after reading and accepting the conditions of the payment plan.
* You can use the "Pay Now" option to finish your purchase if you are not qualified for the payment plan or would like to pay the entire amount at once.

# **Task G**

## **GitHub Repository**

Link: <https://github.com/Madura-Prasad/Bumble-Bee-Loan-System.git>

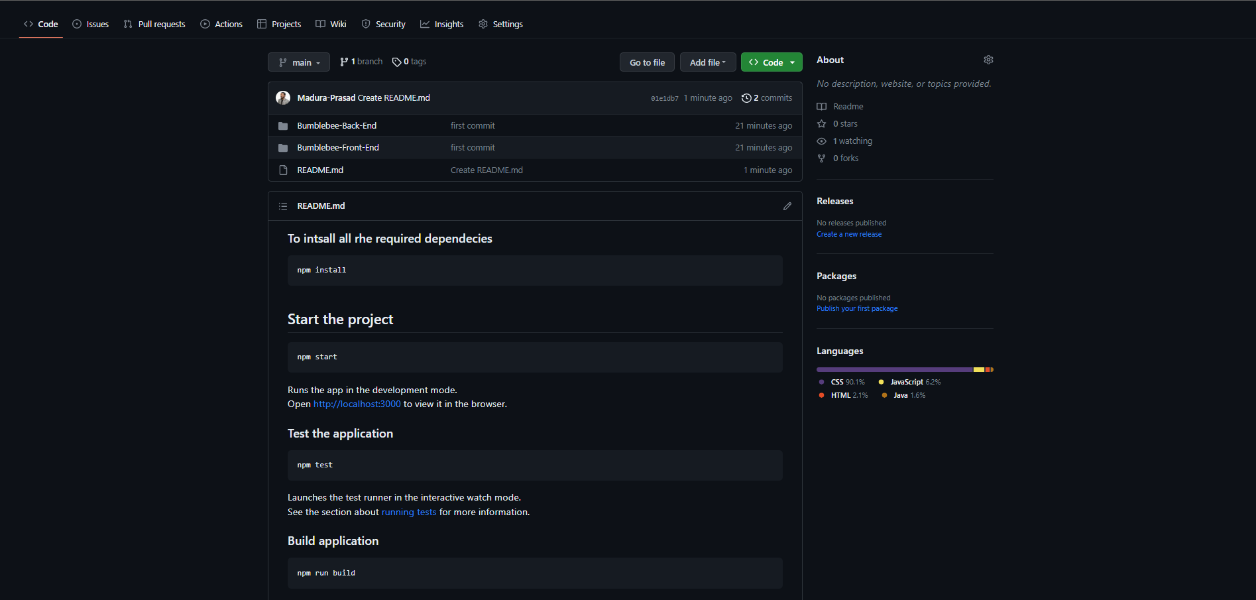


Figure Git Hub Repository

# **Conclusion**

In conclusion, Bumble Bee is an excellent e-commerce site that offers a unique and convenient service of buying first and paying later. With its user-friendly interface, customers can easily purchase products online within their maximum budget, up to 15,000 LKR, without worrying about immediate payment. The site is developed using the Spring Boot and React frameworks, which ensure seamless performance and an optimized user experience. The REST API used by the site enables secure and efficient communication between the front-end and back-end of the application, resulting in faster load times and smoother browsing.

Bumble Bee's offering of interest-free installments for repayment of loans is a testament to their customer-centric approach. This feature ensures that customers have enough time to pay off their loans without incurring any additional interest charges, making it an affordable option for those who are on a tight budget.

Overall, Bumble Bee is a well-reputed online loan providing application that offers a unique and convenient service to its customers. With its efficient technology and customer-centric approach, it is a great option for those looking for a hassle-free shopping experience with flexible payment options.

# **References**

ieeexplore, 2020. *ieeexplore.* [Online]   
Available at: https://ieeexplore.ieee.org/document/8080002  
[Accessed 01 04 2023].

tutorialspoint, 2021. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/design\_pattern/design\_pattern\_overview.htm  
[Accessed 02 04 2023].