OOADJ Mini Project Report

Name: Bhagyesh(PES1UG20CS632)

V Poovarasan(PES1UG21CS836)

Vallabha N Kowshik(PES1UG21CS839)

Problem Statement:

The problem statement for the Java project "Online Book Store Management" revolves around the need for a modern, efficient and user-friendly solution for managing the operations of an online book store. With the increasing popularity of online shopping and the growth of the e-commerce industry, there is a growing demand for a system that can handle the book inventory, customer information, and order processing efficiently. The project aims to develop a robust web-based application that utilizes the Model-View-Controller (MVC) architecture to ensure separation of concerns, code reusability, and scalability. The application will provide an easy-to-use interface for the customers to browse through the book catalog, place orders, make payments, and track their orders. Additionally, the system will allow the store administrators to manage the book inventory, track sales, generate reports, and process orders. The project aims to address the challenges faced by online book stores in managing their operations and provide a reliable and secure solution for their business needs.

Design Patterns Used:

1) Singleton pattern:

In Java, the Singleton pattern is a design pattern that restricts the instantiation of a class to only one instance and provides a global point of access to that instance. This pattern is useful when a single instance of a class is required to coordinate actions across the system, such as managing a shared resource or controlling a set of actions that need to be synchronized.

The Singleton pattern ensures that a class has only one instance by providing a static method that returns the same instance every time it is called. The constructor of the Singleton class is made private, so that no other object can create a new instance of the class.

Singleton pattern has several advantages, including improved resource utilization, efficient memory management, and easy testing. However, it can also introduce a global state, which can make the system harder to understand and maintain. Therefore, it is important to use the Singleton pattern judiciously and consider the trade-offs involved.

2) Factory Pattern:

The Factory pattern is a widely used design pattern in Java that falls under the category of creational patterns. It provides an interface for creating objects in a super class, but allows the subclasses to alter the type of objects that will be created. The pattern promotes loose coupling between objects and enhances code maintainability and flexibility.

The Factory pattern works by creating a factory class that encapsulates object creation logic. This factory class creates objects by invoking a factory method which returns an object of a specified class. The factory method can be overridden by subclasses to create objects of different types. This makes the Factory pattern a great choice when there are multiple subclasses with different object creation logic.

One of the benefits of the Factory pattern is that it allows the client code to be decoupled from the concrete implementation of the objects. The client code interacts only with the factory class to create objects, and does not need to worry about how the objects are created. This promotes code reusability and enables easier maintenance.

3) Decorator Pattern:

The Decorator pattern is a popular design pattern in Java that falls under the category of structural patterns. It allows behavior to be added to an individual object, either statically or dynamically, without affecting the behavior of other objects in the same class.

The Decorator pattern works by creating a decorator class that wraps around an existing class and provides additional functionality. The decorator class has the same interface as the original class, and the object of the original class is passed to the decorator class through its constructor. The decorator class then adds its own functionality to the object of the original class.

One of the benefits of the Decorator pattern is that it promotes open-closed principle, as it allows new behavior to be added to an object without modifying the original code. It also enables flexible behavior modification at runtime, and eliminates the need for creating many subclasses to add different behaviors to an object.

4) MVC Architecture Pattern:

The Model-View-Controller (MVC) Architecture pattern is a widely used design pattern in software engineering, particularly in web and GUI applications. The pattern separates an application into three interconnected components: the Model, View, and Controller.

The Model represents the application's data and business logic. It contains the application's core functionality and data storage. The Model component interacts with the database, file systems, or any other data sources to fetch or update the data.

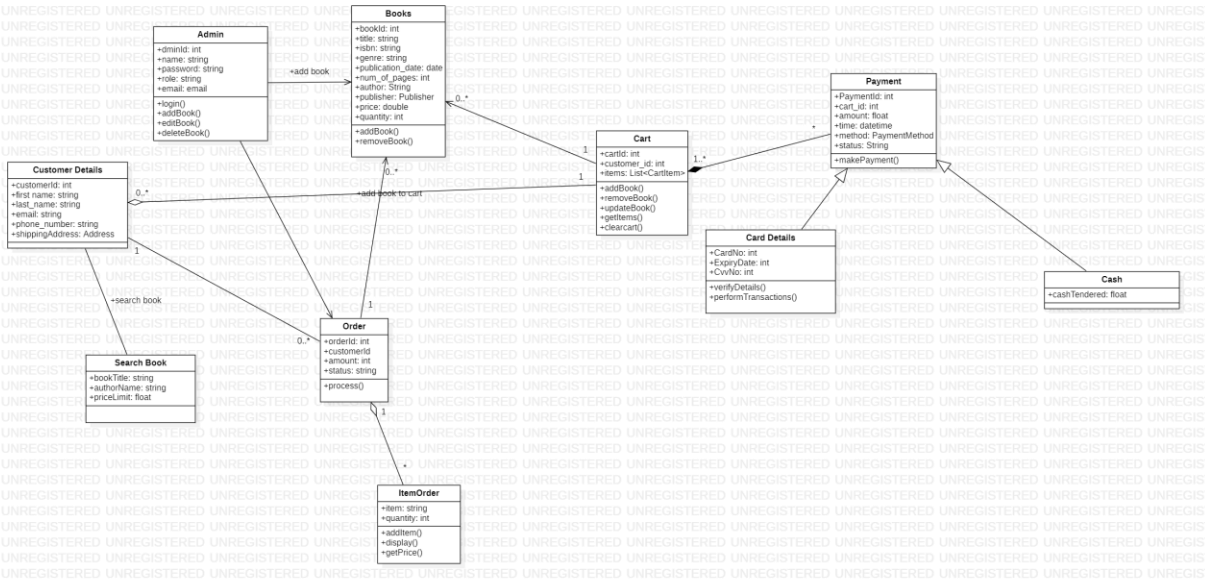
The View is responsible for the presentation of data and the user interface. It is the component that the user interacts with directly. The View receives the data from the Model and displays it to the user in a human-readable format.

The Controller acts as an intermediary between the Model and the View. It receives input from the user via the View and processes it. The Controller then interacts with the Model to fetch or update the data and sends the updated data to the View for presentation to the user.

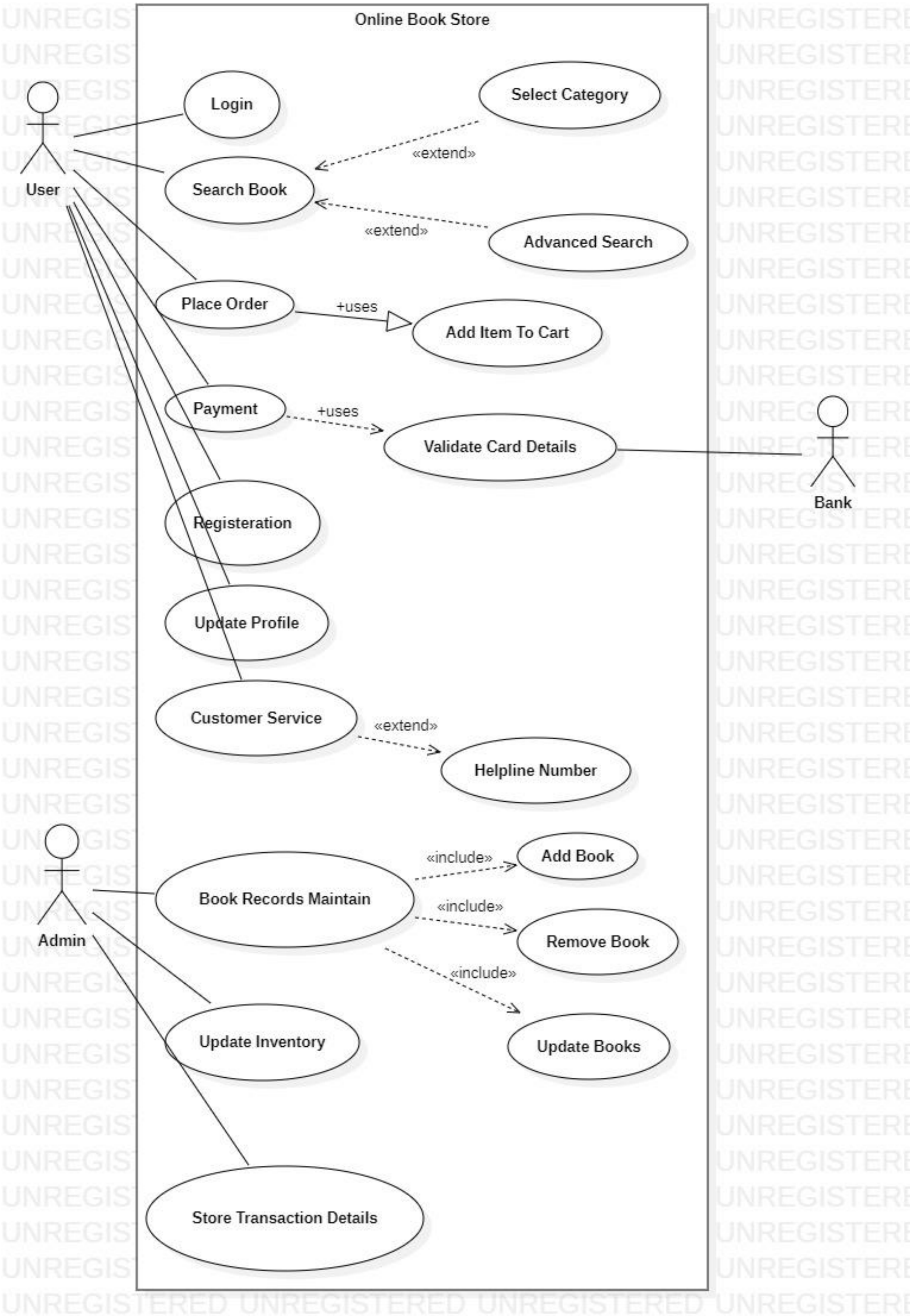
The benefits of using the MVC Architecture pattern are that it promotes code reusability, code maintainability, and scalability. It enables the separation of concerns between the application's data, business logic, and user interface. It also allows the individual components to be developed and tested independently, making the development process more efficient and effective.

<u>UML Diagrams:</u>

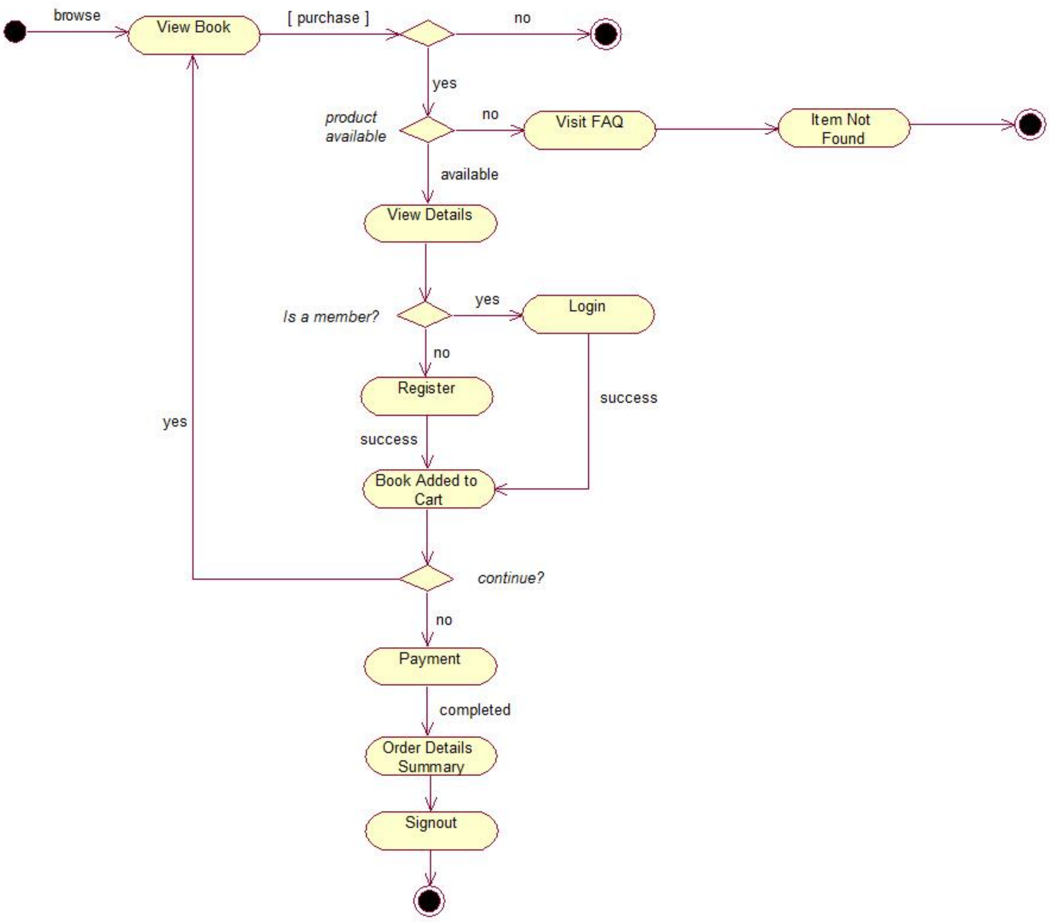
1) Class Diagram:



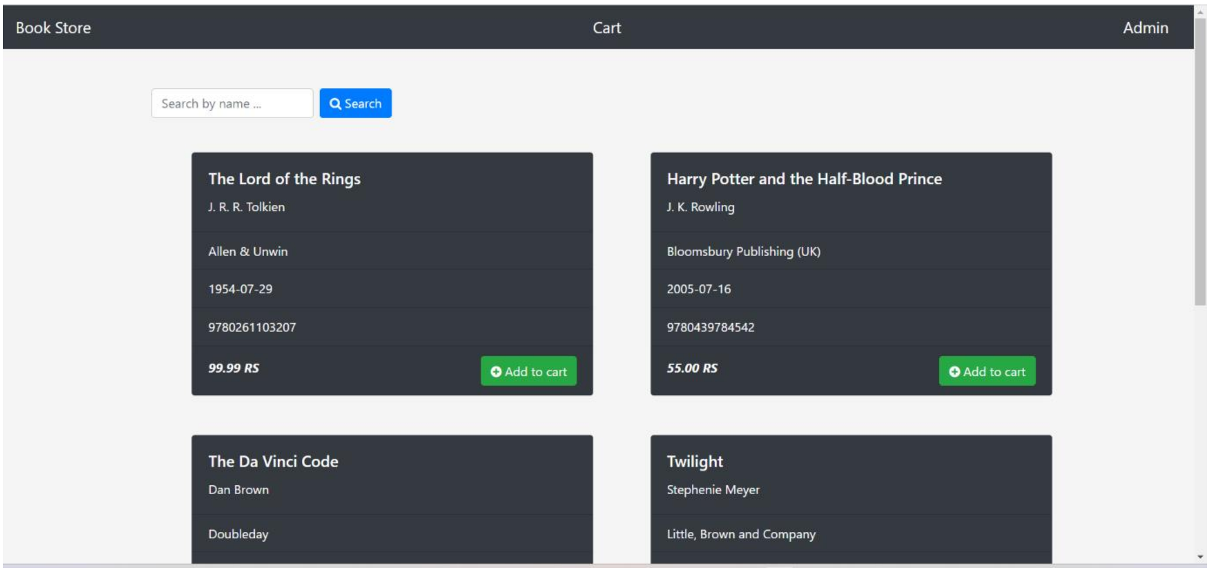
2) UseCase Diagram:



3) Activity Diagram:



Front End Screenshots:


Book Store   Cart   Admin

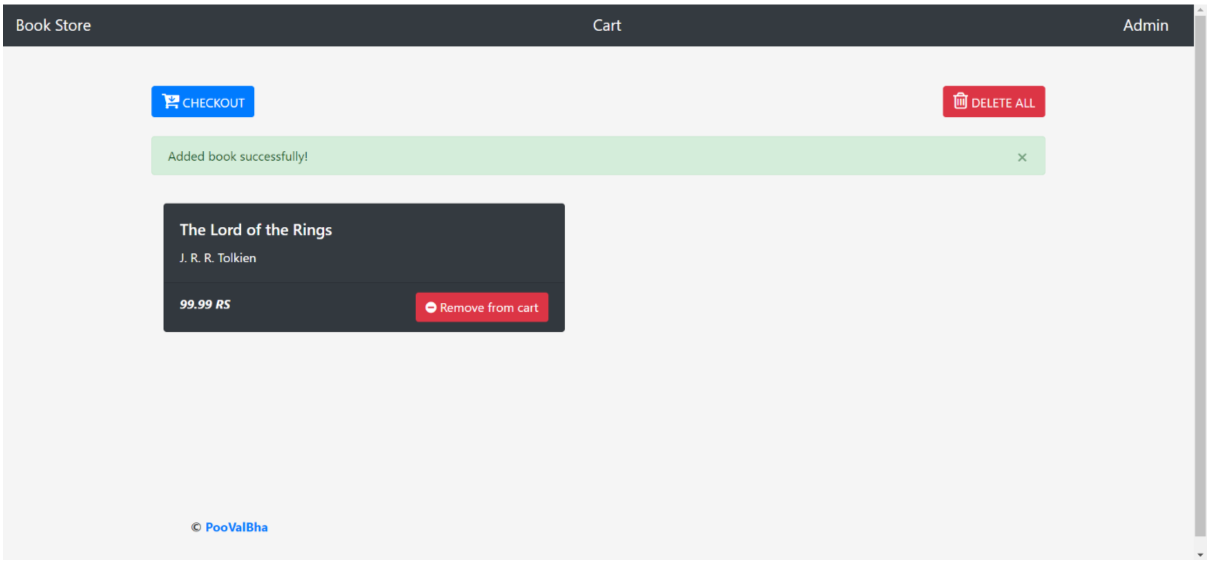
Search by name ...   Search

The Lord of the Rings
J. R. R. Tolkien
Allen & Unwin
1954-07-29
9780261103207
99.99 RS   Add to cart

Harry Potter and the Half-Blood Prince
J. K. Rowling
Bloomsbury Publishing (UK)
2005-07-16
9780439784542
55.00 RS   Add to cart

The Da Vinci Code
Dan Brown
Doubleday

Twilight
Stephenie Meyer
Little, Brown and Company



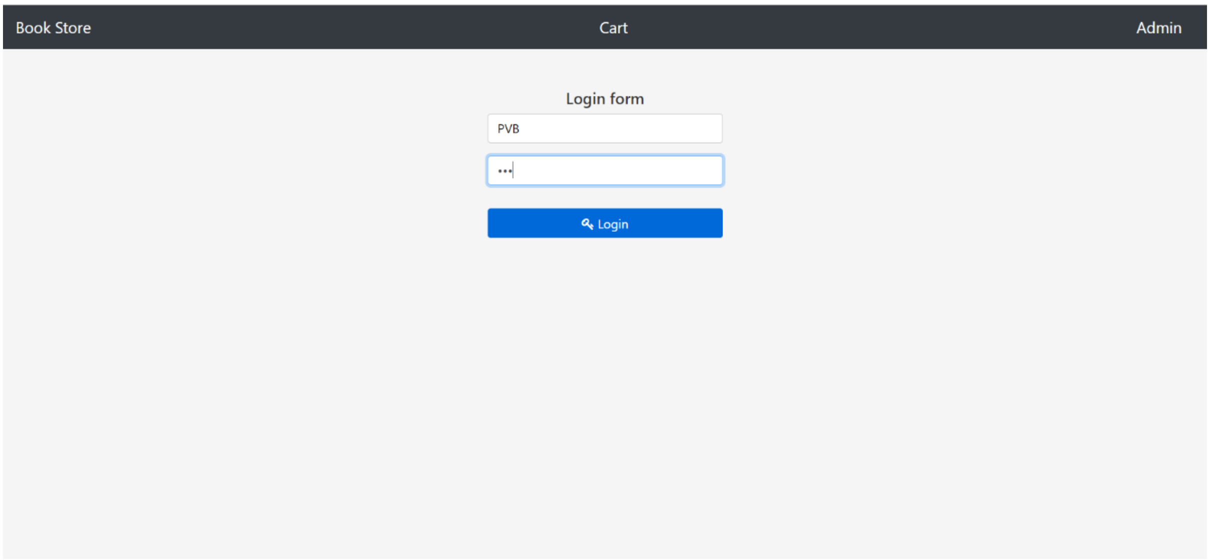
Added book successfully!

The Lord of the Rings

J. R. R. Tolkien

99.99 RS

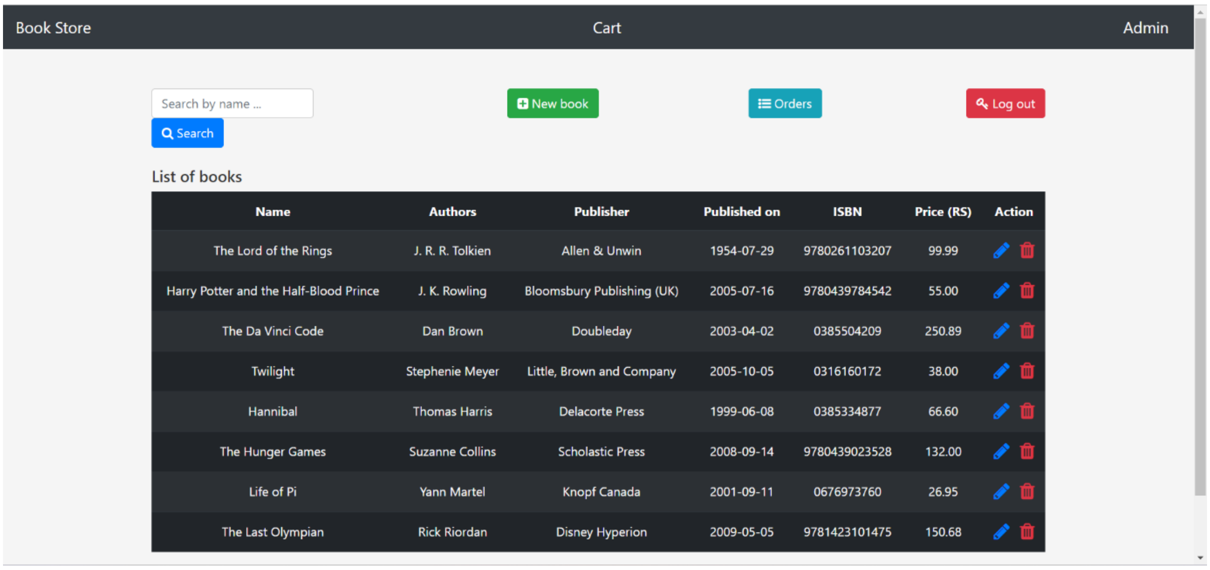
© PooValBha



Login form

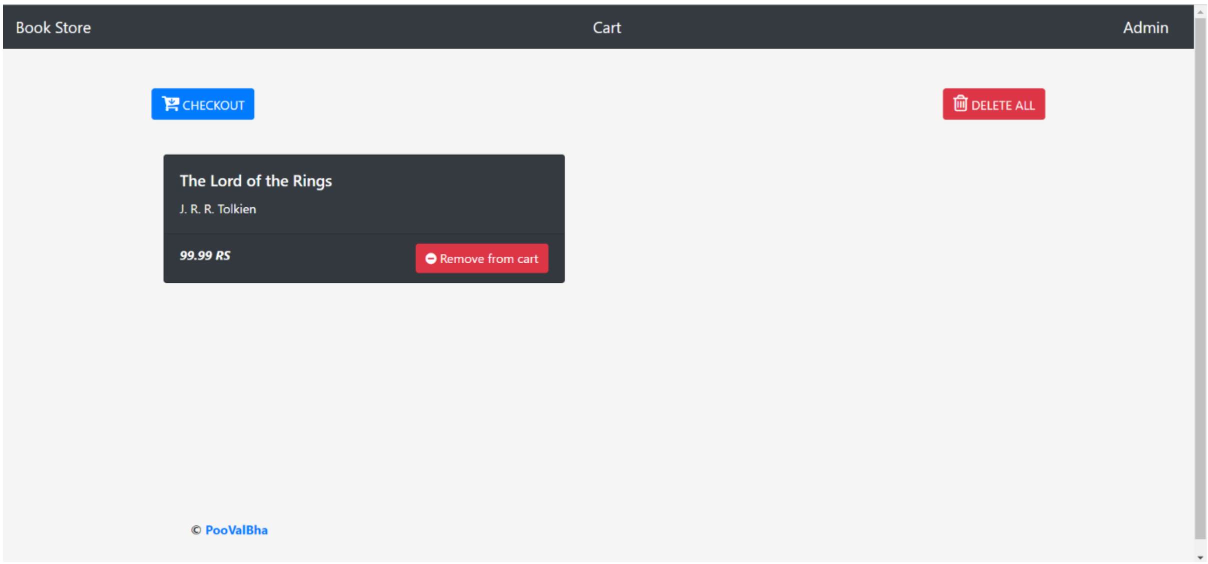
PVB

•••



List of books

Name	Authors	Publisher	Published on	ISBN	Price (RS)	Action
The Lord of the Rings	J. R. R. Tolkien	Allen & Unwin	1954-07-29	9780261103207	99.99	
Harry Potter and the Half-Blood Prince	J. K. Rowling	Bloomsbury Publishing (UK)	2005-07-16	9780439784542	55.00	
The Da Vinci Code	Dan Brown	Doubleday	2003-04-02	0385504209	250.89	
Twilight	Stephenie Meyer	Little, Brown and Company	2005-10-05	0316160172	38.00	
Hannibal	Thomas Harris	Delacorte Press	1999-06-08	0385334877	66.60	
The Hunger Games	Suzanne Collins	Scholastic Press	2008-09-14	9780439023528	132.00	
Life of Pi	Yann Martel	Knopf Canada	2001-09-11	0676973760	26.95	
The Last Olympian	Rick Riordan	Disney Hyperion	2009-05-05	9781423101475	150.68	

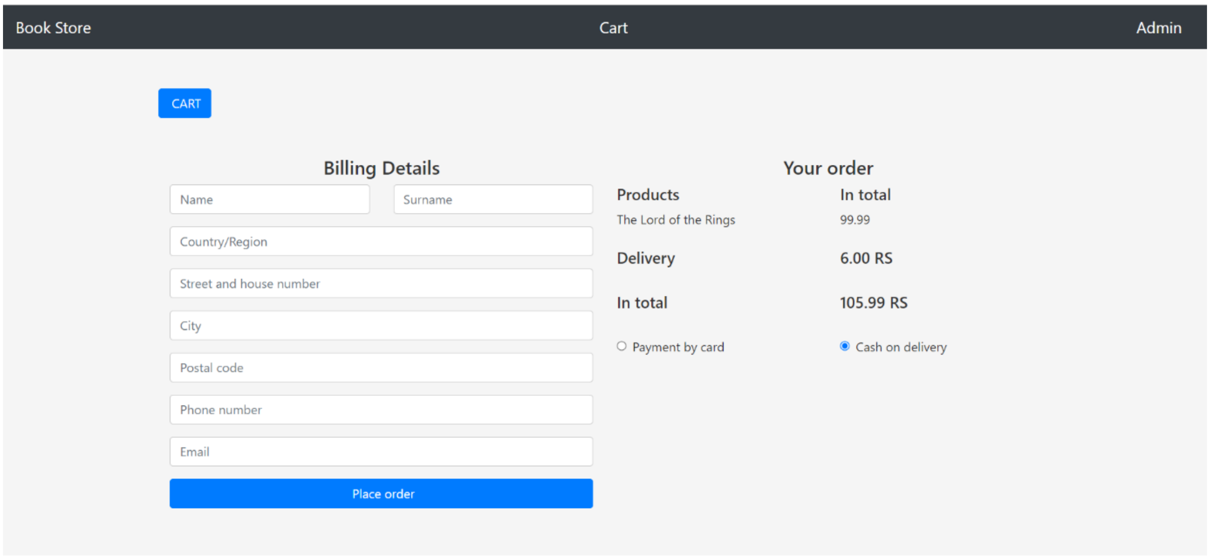


The Lord of the Rings

J. R. R. Tolkien

99.99 RS

© PooValBha



Billing Details

Your order

Products	In total
The Lord of the Rings	99.99
Delivery	6.00 RS
In total	105.99 RS

○ Payment by card ● Cash on delivery

Github Link: https://github.com/Vallabha-Kowshik/OOADJ_Mini-Project