|  |  |
| --- | --- |
| C:\Users\HP\OneDrive\Desktop\WhatsApp Image 2022-11-17 at 12.25.02.jpeg | **Online BookShop** |
| **JAVA PROJECT NOVEMBER 2025** | JAVA PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF **BACHELOR ENGINEERING** IN  Computer and communication  OF THE ANNA UNIVERSITY |
|  |  |
| **PROJECT WORK** | Submitted by  **SUGIRTHAN S-722824134055** |
| **BATCH 2024 – 2028** |  |
|  | Under the Guidance of  **MS.P.MEGALA ASSOCIATE PROFESSOR** |
| **Department of Computer Communication**  **Sri Eshwar College of Engineering**  (An Autonomous Institution – Affiliated to Anna University)  **COIMBATORE – 641 202** | |

Sri Eshwar College of Engineering

(An Autonomous Institution – Affiliated to Anna University)

**COIMBATORE – 641 202**

**BONAFIDE CERTIFICATE**

Certified that this Report titled **“Online BookShop”** is the Bonafide work of **SUGIRTHAN S - 722824134055** who carried out the project work under my supervision.

|  |  |
| --- | --- |
|  |  |
| **SIGNATURE** | **SIGNATURE** |
| **DR.C.VIVEK M.TECH,(PH.D) HEAD OF THE DEPARTMENT**  Professor, Department of Computer and communication engineering  Sri Eshwar College of Engineering, Coimbatore – 641 202. | **MS.P.MEGALA SUPERVISOR**  Associate Professor,  Department of Computer and communication engineering  Sri Eshwar College of Engineering,  Coimbatore – 641 202. |
|  |  |

|  |  |
| --- | --- |
| Submitted for the **Autonomous Semester End Mini Project Viva-Voce** held on  **30.10.2025.** | |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

**DECLARATION**

**SUGIRTHAN S [722824134055]**

To declare that the project entitled **“ Online BookShop”** submitted in partial fulfilment to the University as the project work of Bachelor of Engineering (Computer and commmunication engineering) Degree, is a record of original work done by me under the supervision and guidance of **Ms.P.Megala,** Associate Professor, Department of Computer And communication engineering, Sri Eshwar College of Engineering, Coimbatore.

**Place:** Coimbatore

# Date:

[**SUGIRTHAN S**]

Project Guided by,

# Ms.P. Megala

Associate Professor / CCE

**ACKNOWLEDGEMENT**

****

The success of a work depends on a team and cooperation. I take this opportunityto express my gratitude and thanks to everyone who helped me in my project. I would like to thank the management for the constant support provided by them to complete this project.

It is indeed our great honor bounded duty to thank our beloved **Chairman Mr. R. Mohanram,** for his academic interest shown towards the students.

We are indebted to our **Director Mr. R. Rajaram,** for motivating and providing me with all facilities.

I wish to express my sincere regards and deep sense of gratitude to **Dr. Sudha Mohanram, M.E, Ph.D. Principal**, for the excellent facilities and encouragement provided during the course of the study and project.

We are indebted to **Dr.c.Vivek M.Tech(Ph.d).,** Professor and Head of the Computer and communication engineering Department for having permitted me to carry out this project and giving the complete freedom to utilize the resources of the department.

I express my sincere thanks to my guide **Ms.P.Megala** Associate Professor of Computer and communication Department for the valuable guidance and encouragement given to me for completing this project.

I solemnly express our thanks to all the teaching and nonteaching staff of the Computer and communication Department, family and friends for their valuable support which inspired me to work on this project.

**TABLE OF CONTENTS**

****

|  |  |
| --- | --- |
| **CONTENT** | **PAGE NO** |
| **ABSTRACT** | **6** |
| 1. **INTRODUCTION**    1. **BACKGROUND AND MOTIVATION**    2. **PROBLEM STATEMENT AND OBJECTIVES** | **7**  **7** |
| 1. **SYSTEM REQUIREMENTS & SPECIFICATION**    1. **TECHNICAL ENVIRONMENT**    2. **FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS** | **8**  **9** |
| 1. **LOW-LEVEL DESIGN (LLD)**    1. **MODULE/COMPONENT OVERVIEW**    2. **CLASS DESIGN AND RELATIONSHIPS** | **10**  **11** |
| 1. **MODELING (UML DIAGRAMS)**    1. **ER DIAGRAM**    2. **USE CASE DIAGRAM** | **12**  **13** |
| 1. **DATABASE DESIGN**    1. **ENTITIES AND TABLES**    2. **RELATIONSHIPS AND CONSTRAINTS** | **14**  **14** |
| 1. **IMPLEMENTATION**    1. **CODING**    2. **OUTPUT AND FLOWCHART** | **15**  **20** |
| 1. **CONCLUSION AND FUTURE SCOPE**    1. **CONCLUSION**    2. **FUTURE SCOPE** | **22**  **23** |
| 1. **APPENDICES AND REFERENCES**    1. **REFERENCES**    2. **APENDIX A**    3. **APENDIX B** | **24**  **25**  **26** |

**ABSTRACT**

****

This project implements an **Online Book Shop System** to simplify and automate the process of book purchasing, rental, and management. Motivated by the need to replace traditional manual book sales and record-keeping, the system features a user-friendly web interface developed using **Java, MySQL, and Servlet/JSP technologies**. Key functionalities include book browsing, searching, purchasing or renting, user registration, and secure authentication. The application efficiently stores book details, user information, and transaction records in a well-structured database for easy access and management. The system design follows a modular architecture to ensure scalability, maintainability, and ease of future updates. Testing focused on validating key operations such as book addition, order placement, and rental return. The project enhances convenience, reduces manual errors, and promotes efficient online transactions. Future enhancements may include integrating online payment gateways and personalized recommendation systems. Overall, the system provides a reliable and practical solution for managing online book sales and rentals effectively.

**INTRODUCTION**

****

# Background and Motivation

The growing demand for books and reading materials in educational institutions and among the general public has created a pressing need for digitized and automated book management systems. Traditionally, book purchasing and rental processes involved manual record-keeping, in-person transactions, and physical storage of data, which are time-consuming and prone to errors or data loss. Such inefficiencies make it difficult to maintain accurate inventory, manage customer orders, and ensure timely service. Moreover, manual systems often lack transparency in transactions and require significant administrative effort and physical resources, leading to reduced operational efficiency and customer satisfaction.

# Problem Statement and Objectives

The fundamental problem addressed by this project is the inefficiency and limitations of traditional manual book sales and rental management systems. Manual handling of book records, customer details, and transactions often leads to data inconsistencies, misplaced records, and delays in order processing. Additionally, the lack of automation makes it difficult to track book availability, manage inventory, and generate timely sales reports. These challenges collectively affect customer satisfaction, reduce operational efficiency, and hinder effective management of online book transactions.

**SYSTEM ANALYSIS AND DESIGN**

****

# 2.1. Technical Environment

This section outlines the fundamental hardware and software requirements necessary for the deployment and operation of the Receipt Collection and Management System.

# Hardware Requirements:

The system can operate on any standard desktop or laptop computer capable of running Java-supported applications. Minimum specifications include a dual-core processor, 4GB RAM, and at least 20GB of free disk space to accommodate the application and receipt files storage.

# Software/OS Prerequisites:

The project is developed using Java Spring Boot and Thymeleaf, requiring Java Development Kit (JDK) 17 or higher installed on the host machine. The backend runs on a server supporting Java applications, while the frontend is browser-based with no special client software needed. Database management uses MySQL or PostgreSQL as the data store.

# Browser or Device Compatibility:

Users can access the system on modern web browsers such as Chrome, Firefox, Edge, and Safari. The interface is responsive and optimized for use on desktop and tablet devices; mobile compatibility may be limited based on screen size.

# 2.2 Functional & Non-Functional Requirements

* + - **Users, Roles, and Permissions:**

The system supports multiple user roles, including **Admin** and **Customer**. Each role has specific access rights and responsibilities. **Admins** can manage book records, categories, and user accounts, as well as view and update transaction details. **Customers** can register, browse available books, view details, add books to their cart, and purchase or rent them securely. Authentication and authorization are implemented through a secure login system to ensure data privacy and role-based access control.

# Expected Functionalities:

# Key features include adding and managing book details such as title, author, price, and category, browsing and searching books by users, viewing detailed book information, purchasing or renting books online, and managing orders efficiently. The system also supports editing and deleting book records with proper permission checks. Additionally, it generates exportable reports summarizing sales and rentals by category over specific periods.

# Performance:

The system should handle simultaneous operations from multiple users without significant delay. Book browsing, purchasing, and rental processes are optimized to ensure fast response times and efficient database management for smooth user experience.

# Security:

All user interactions require authentication with encrypted password storage. Role-based access controls prevent unauthorized data access. Book management operations validate input data and sanitize user inputs to prevent injection attacks and ensure secure transactions.

# Usability:

The interface follows user-friendly design principles, offering clear navigation and feedback during actions such as uploads or deletions. Form validation reduces entry errors, and confirmation prompts prevent accidental data loss.

**LOW-LEVEL DESIGN (LLD)**

****

# Module/Component Overview:

# The Online Book Shop System follows the MVC (Model-View-Controller) architecture for better structure, scalability, and maintainability.

# Controller Layer:

Handles user requests, navigation, and view rendering (e.g., BookController, UserController, OrderController).

# Service Layer:

Implements the business logic, interacting with repositories (e.g., BookService, UserService, OrderService).

# Repository Layer:

Manages database operations using **Spring Data JPA**, handling CRUD operations for books, users, and orders.

# Model Layer:

Defines entities (User, Book, and Order) and their relationships.

# View Layer:

# Developed using ****Thymeleaf templates**** (e.g., books.html, orders.html, users.html) to create dynamic and interactive web pages.

# Class Design and Relationships:

The system includes three core entities with clear relationships:

# User: Contains details such as userId, username, email, and password. One user can place multiple orders.

# Book: Stores book details (bookId, title, author, price, category, availability). A book can appear in many orders.

**Order:**

Represents purchase or rental details (orderId, orderDate, totalAmount) and connects to both the user and the books purchased or rented.

# Relationships:

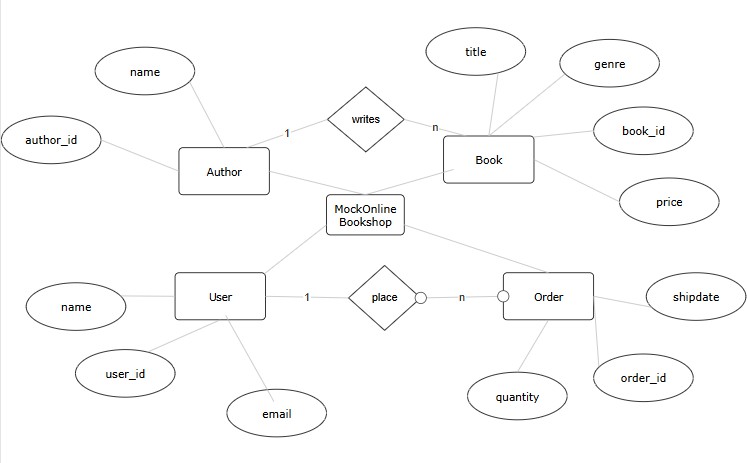
# One User → Many Orders

* One Order→ Many Books

**UMODELING(UML DIAGRAMS)**

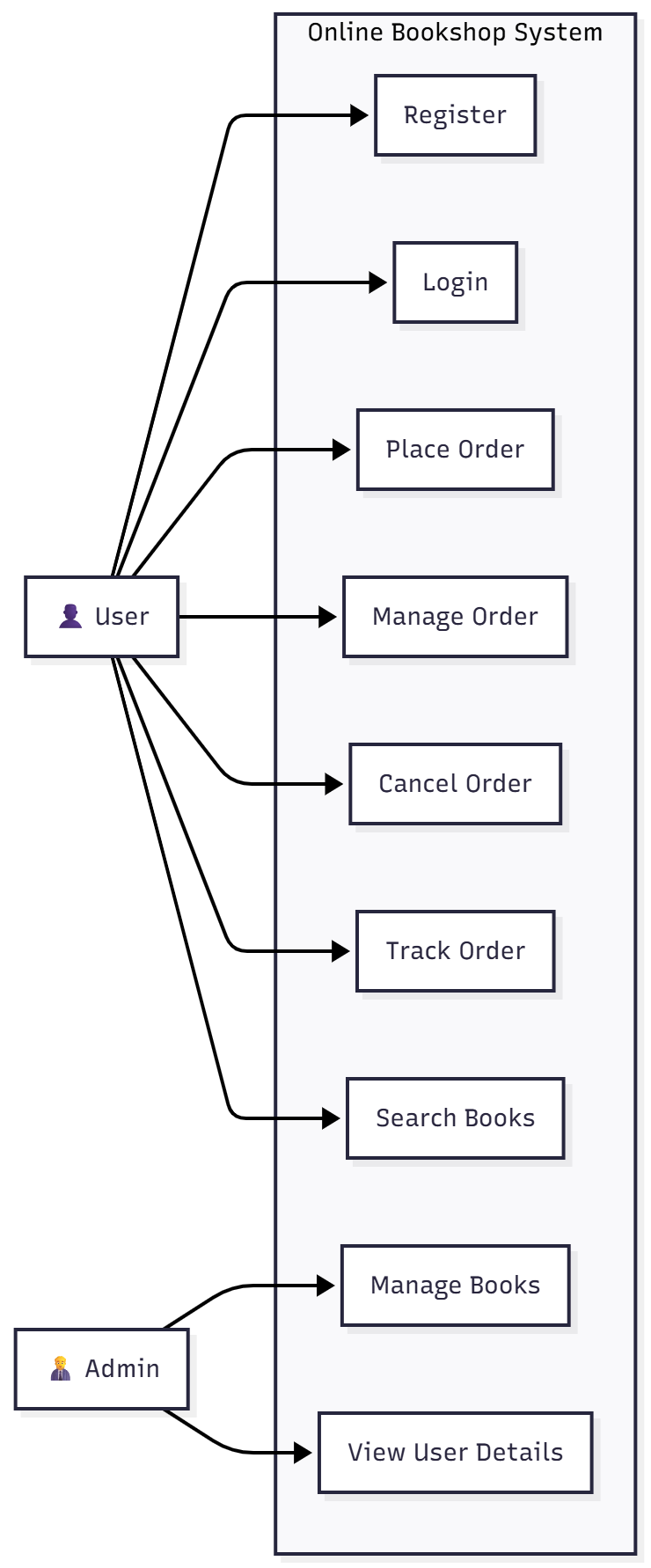
****

* 1. **ER DIAGRAM:**



**Fig:4.1.1 ER DIAGRAM**

* 1. **USE CASE DIAGRAM:**



**4.2.1 USECASE FOR ONLINE BOOKSHOP**

**DATABASE DESIGN**

****

* 1. **ENTITIES AND TABLES**
* **User**

(user\_id, username, email, password)

# Book (book\_id, title, author, price, category, availability)

* + - **Order**(order\_id, order\_date, total\_amount, user\_id FK → User)
  1. **RELATIONSHIPS AND CONSTRAINTS:**
     + **User →** Order: One-to-Many (a user can place multiple orders)
     + **Order →** Book: Many-to-Many (an order can include multiple books, and a book can appear in multiple orders)

# Constraints:

* + - Auto-increment IDs (user\_id, book\_id, order\_id, category\_id)
    - Foreign key constraint from order.user\_id → user.user\_id
    - Foreign key constraint from order\_details.book\_id → book.book\_id
    - Email and username unique in User
    - book.title and book.author must not be empty

**IMPLEMENTATION AND RESULT**

****

* 1. **CODE**

**Models**

# Online Bookshop.java

package com.example.onlinebookshop.model;

import jakarta.persistence.\*;

import lombok.Data;

@Entity

@Table(name = "book\_categories")

@Data

public class BookCategory {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "category\_id")

private Integer categoryId;

@Column(length = 50, nullable = false)

private String name;

@Column(columnDefinition = "TEXT")

private String description;

// Getters and Setters

public Integer getCategoryId() {

return categoryId;

}

public void setCategoryId(Integer categoryId) {

this.categoryId = categoryId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

# User.java

# package com.example.onlinebookshop.model;

# import jakarta.persistence.\*;

# import lombok.Data;

# import java.sql.Timestamp;

# @Entity

# @Table(name = "users", uniqueConstraints = {@UniqueConstraint(columnNames = "email")})

# @Data

# public class User {

# @Id

# @GeneratedValue(strategy = GenerationType.IDENTITY)

# @Column(name = "user\_id")

# private Integer userId;

# @Column(length = 50, nullable = false)

# private String username;

# @Column(length = 100, nullable = false, unique = true)

# private String email;

# public Timestamp getJoinDate() {

# return joinDate;

# }

# public void setJoinDate(Timestamp joinDate) {

# this.joinDate = joinDate;

# }

# }

# Controllers

# package com.example.onlinebookshop.controller;

# import com.example.onlinebookshop.model.Book;

# import com.example.onlinebookshop.service.BookService;

# import org.springframework.beans.factory.annotation.Autowired;

# import org.springframework.ui.Model;

# import org.springframework.web.bind.annotation.\*;

# import java.util.List;

# import java.util.Optional;

# @RestController

# @RequestMapping("/api/books")

# public class BookController {

# @Autowired

# private BookService bookService;

# public List<Book> getBooksByCategory(@PathVariable Integer category\_id) {

# return bookService.getBoo

# UserController.java

package com.example.receiptcollector.controller; import com.example.receiptcollector.model.User;

import com.example.receiptcollector.service.UserService;

import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController @RequestMapping("/api/users") public class UserController {

@Autowired

private UserService userService;

@PostMapping

public User addUser(@RequestBody User user) { return userService.addUser(user);

}

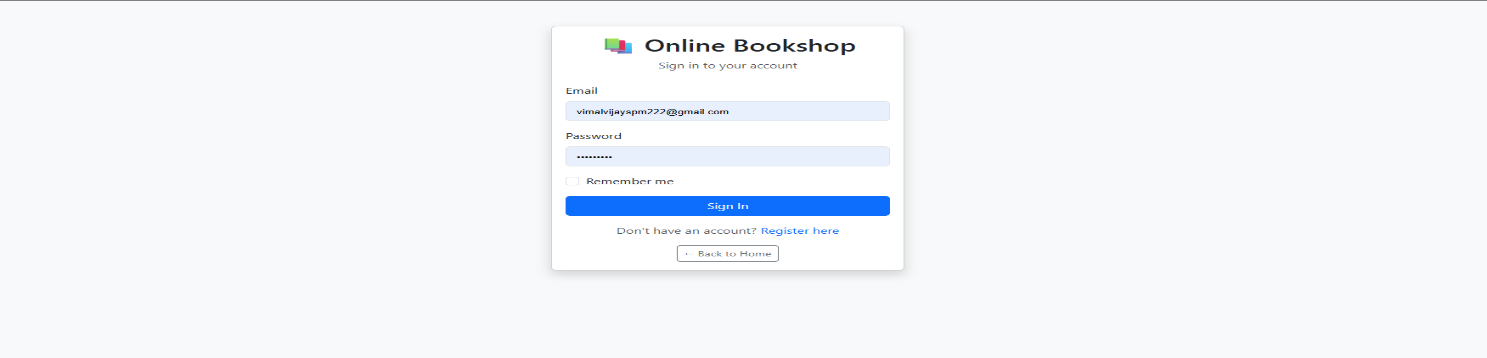
@GetMapping

public List<User> getAllUsers() { return userService.listUsers();

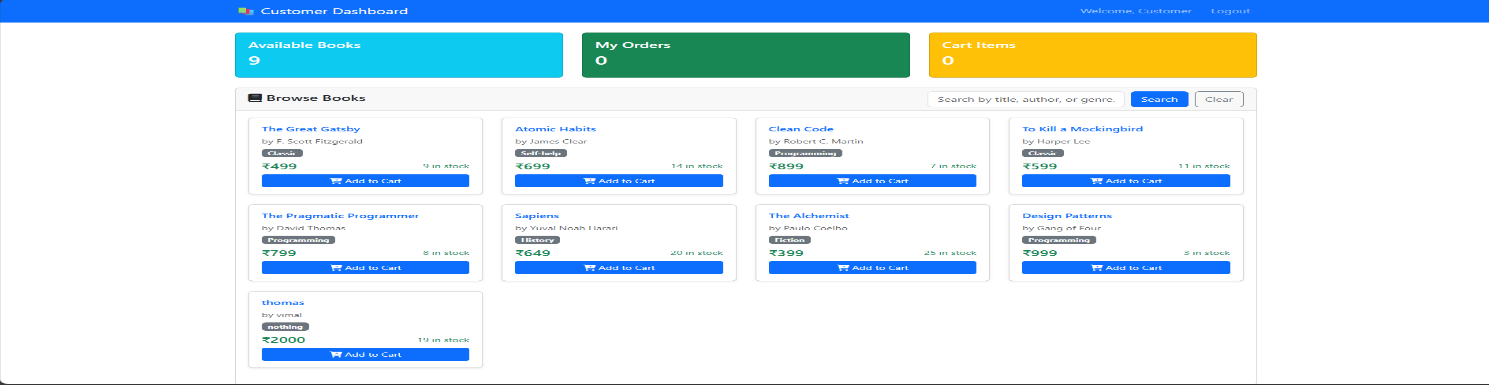
}

}

* 1. **OUTPUT**

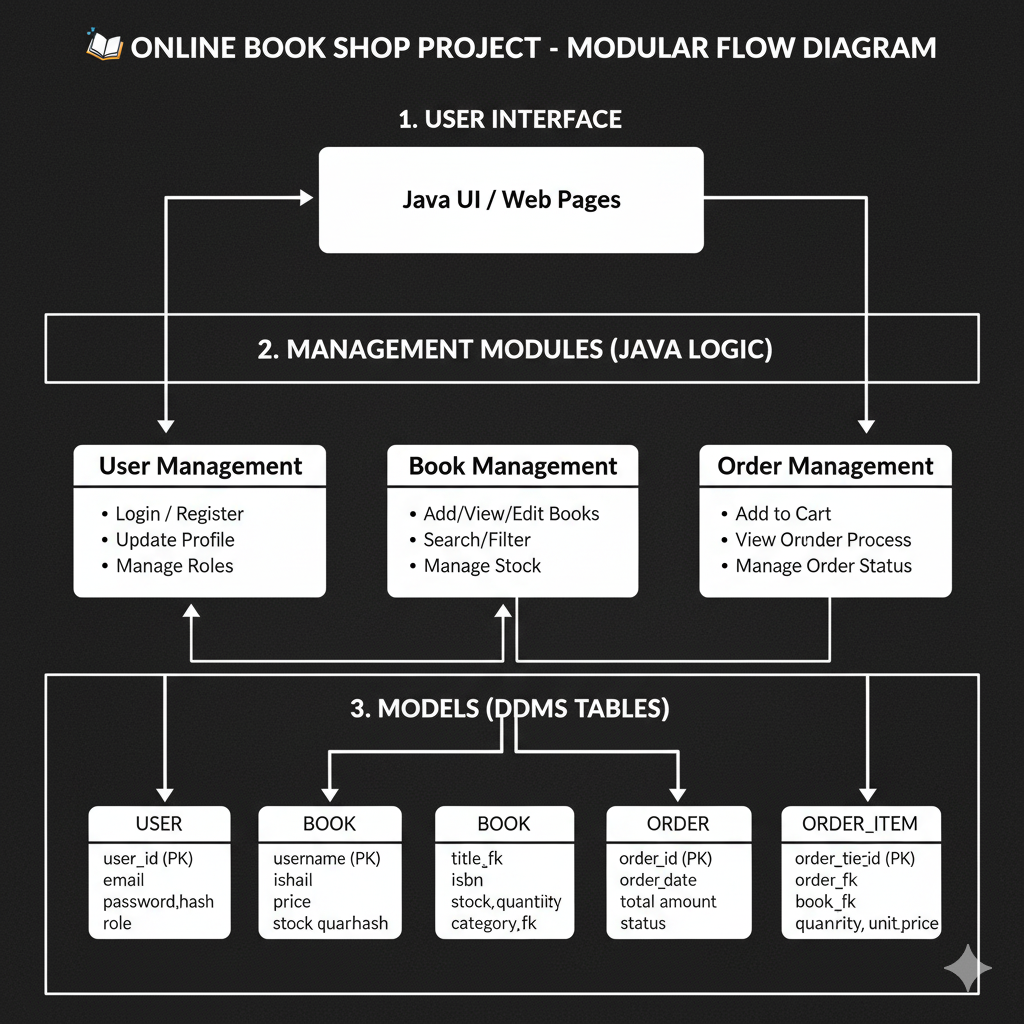


**6.2.1 ONLINE BOOKSHOP LOGIN PAGE**



**6.2.2 ONLINE BOOKSHOP OUTPUT**

**6.2 FLOW CHART:**

****

**6.2.3 ONLINE BOOKSHOP FLOWCHART**

**CONCLUSION**

****

The **Online Book Shop System** was developed to automate and simplify the process of purchasing and renting books through a secure and user-friendly digital platform. By replacing traditional, manual methods with an interactive web-based solution, the project enhances efficiency, reduces errors, and ensures accurate management of book inventory and transactions.

Using the **Model-View-Controller (MVC)** architecture, the system maintains a clear separation between presentation, business logic, and data management layers, thereby improving scalability and maintainability. Core features such as secure user authentication, book categorization, and order management provide users and administrators with reliable tools for smooth operation and effective book management.

The system not only streamlines book purchasing and rental operations but also improves user convenience by offering easy search, instant access, and real-time order tracking. Overall, it contributes to better resource utilization, faster transactions, and enhanced satisfaction for both customers and administrators.

Overall, the project not only fulfills its objective of simplifying the bookshop management process but also lays a strong foundation for future enhancements—such as integrating online payment gateways, AI-based book recommendations, and cloud-based deployment. Hence, the *Online Book Shop System* stands as a reliable, efficient, and scalable solution that bridges the gap between technology and modern book retail management.

**FUTURE SCOPE**

****

The *Online Book Shop System* has ample opportunities for future enhancement and scalability. One of the major improvements can be the integration of a **secure online payment gateway**, enabling users to make instant payments through credit/debit cards, UPI, and digital wallets. This will make the entire process of purchasing and renting books fully automated and user-friendly. Additionally, **AI-based recommendation systems** can be introduced to analyze user preferences and browsing history to suggest relevant books, enhancing the personalization and engagement of the platform.

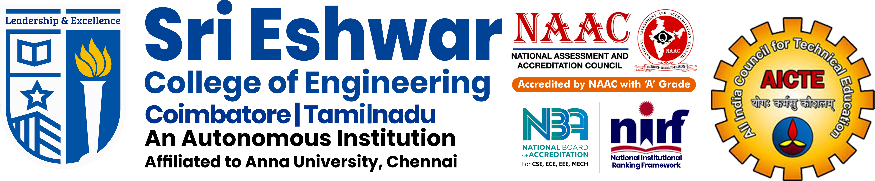
Another major development area lies in **mobile application development**, which would allow users to access the system conveniently from their smartphones. The platform can also be upgraded to support **multi-vendor listings**, where different publishers and sellers can register and manage their own inventories. Features like **customer reviews and ratings**, as well as **integration with e-book and audiobook services**, would make the system more interactive and appealing to a wider range of users, creating a richer and more dynamic reading environment.

From a technical perspective, the system can be migrated to **cloud-based infrastructure** to ensure better scalability, performance, and data security. The use of **data analytics dashboards** can help administrators track sales performance, user behavior, and market trends in real time, supporting informed decision-making. In the long term, adopting **microservices architecture**, implementing **machine learning models** for trend prediction, and utilizing **blockchain for secure transactions** could transform this project into a professional, enterprise-level online bookstore capable of competing with modern e-commerce solutions.

**REFERENCES**

****

* Akash Panicker. "Online BookShop." 2022. <https://www.akashpanicker.com/receipt-management-system>
* **Plain English.** *"Building an Online Bookstore Application with Java and Spring Boot."* 2025. <https://plainenglish.io/building-an-online-bookstore-application-with-java-and-spring-boot>
* **Octo.** *"E-Commerce Platform Architecture Overview."* 2022. https://blog.octo.com/ecommerce-platform-architecture-overview
* **Indeed Career Guide.** *"How to Write a Project Conclusion (Including Tips and Examples)."* 2025. <https://uk.indeed.com/career-advice/career-development/how-to-write-a-project-conclusion>
* **Amazon Web Services.** *"Build an E-Commerce Application Using AWS and Java."* 2024. https://aws.amazon.com/blogs/compute/build-an-ecommerce-application-using-aws-and-java

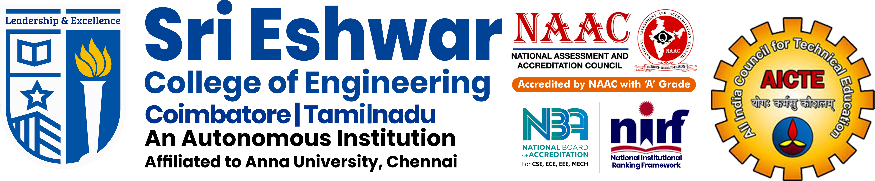


|  |  |  |
| --- | --- | --- |
| **PROJECT TITLE** | **ONLINE BOOKSHOP** | |
| **PROGRAM** | **B.E Computer and communication Engineering** | |
| **PROJECT NUMBER** |  | |
| **REGISTER NUMBER & NAME** | **722824134055** | **SUGIRTHAN S** |
| **NAME OF THE SUPERVISOR** | **Ms.P Megala , Associate Professor, CCE Department** | |
| **NAME OF THE SDG GOALS MAPPED** | **Responsible Consumption and Production, Climate Action,**  **Industry, Innovation and Infrastructure** | |
| **MENTION THE SDG GOALS NUMBER** | **SDG 12, SDG 13, SDG 9** | |
| **NAME OF THE TRL LEVEL** | **Prototype demonstrated in relevant environment (pre- commercial phase)** | |
| **MENTION THE TRL LEVEL** | **Level: 6** | |

**POs & PSOs Mapping (Put a tick mark in the mapped PO’s & PSO’s):**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program Outcomes** | | | | | | | | | | | **Program Specific**  **Outcomes** | | |
| **PO 1** | **P**  **O 2** | **P**  **O 3** | **P**  **O 4** | **P**  **O 5** | **P**  **O 6** | **P**  **O 7** | **P**  **O 8** | **P**  **O 9** | **P**  **O 10** | **P**  **O 11** | **PS O 1** | **PS O 2** | **PS O 3** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Signature of the Supervisor (Ms.P Megala)**



**VENUE AND EXPENDITURE STATEMENT FOR THE PROJECT WORK**

|  |  |
| --- | --- |
| **Laboratory details where**  **the project is carried out** | **CYBER SECURITY, CCE** |
| **Software / Hardware details** |  |

**Details of the Component and Expenditure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Name of the Component** | **Qty** | **Price / Unit in (Rs.)** | **Amount (Rs.)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**(\*Include any other charges which includes fabrication cost and others)**

**Signature of the student Signature of the Supervisor**

**(Sugirthan S) (Ms.P Megala)**