**A Project Report**

on

# Online Book Store Using Spring boot

***Submitted in partial fulfillment of the***

***requirement for the award of the degree of***

Bachelor of Technology in Computer Science and Engineering

**Under The Supervision of**
**Ms. Ambika Gupta**

Submitted By

ABHISHEK SINGH
20SCSE1010500

ASHUTOSH TRIPATHI
20SCSE1010548

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**GALGOTIAS UNIVERSITY, GREATER NOIDA**
**INDIA**
**OCTOBER, 2023**

## SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
## GALGOTIAS UNIVERSITY, GREATER NOIDA

## CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"ONLINE BOOK STORE USING SPRINGBOOT"** in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of October, 2023 to June and 20223, under the supervision of Ms.Ambika Gupta, Department of Computer Science and Engineering, Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

Abhishek Singh, 20SCSE1010500

Ashutosh Tripathi, 20SCSE1010548

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Ms.Ambika Gupta

Assistant Professor

## Abstract

This Online Book Store  Project in Spring boot created based on java, Spring boot, and MYSQL Database. The Online Book Store System is a project similar like shopping cart or e-commerce but it is only for book shopping. Categories wise books available. This project built with Spring boot framework and it's allows users to search and purchase book online based on category, author and subject.

**Admin Features of Online Book Store:**

Dashboard – For the admin dashboard, you will be able to all the basic access in the whole system. Such as summary of products, orders, and the categories.
Manage Books– The admin has access to the books management information system. He can add, update and delete the books.
Manage Categories – The page where the admin can add, edit and delete categories information.
Manage Orders – As the main functions of the admin, the admin can accept or reject the order from the customers on a case to case basis and the list of customer orders are listed.
Manage User– The admin can manage the user's account. Admin can add, update and Block user in the system.
Login and Logout – By default one of the security features of this system is the secure login and logout system.

**Customer Features of Online Book Store:**

Login Page – Customer enter their website credentials on this page to gain access in order to log in.
Home Page– When customer visit the website, this is the system's default page. This page shows the books for sale in the store, or by entering a keyword in the search box above the books.
Book View Page – The page on which the product's specific information is shown, as well as the page on which the customer adds the product to his or her cart.
My Order Page – The page that lists the customer's orders.

# Table of Content

## Introduction

- A website may contain thousands of different pages including the web pages. People can access the website from anywhere using the internet. This can particularly be helpful to both the sellers and consumers.

- Using an attractive website, the products can be marketed to the potential buyers who would then have the luxury of getting the product delivered to them as fast and in the most convenient way as possible.

- Online Book Store is an e-commerce site where the users can browse from a large catalogue of books, add books in the cart and place the order, and make payment with ease.

## Literature Survey

A lot of websites and applications can be found when we search the google which are developed for learning purposes. But there is ambiguity in choosing the appropriate content in appropriate time. Some websites has been developed which consists of stories, novels,essays etc. Similarly, some personal blogs and websites are developed for studying purpose.

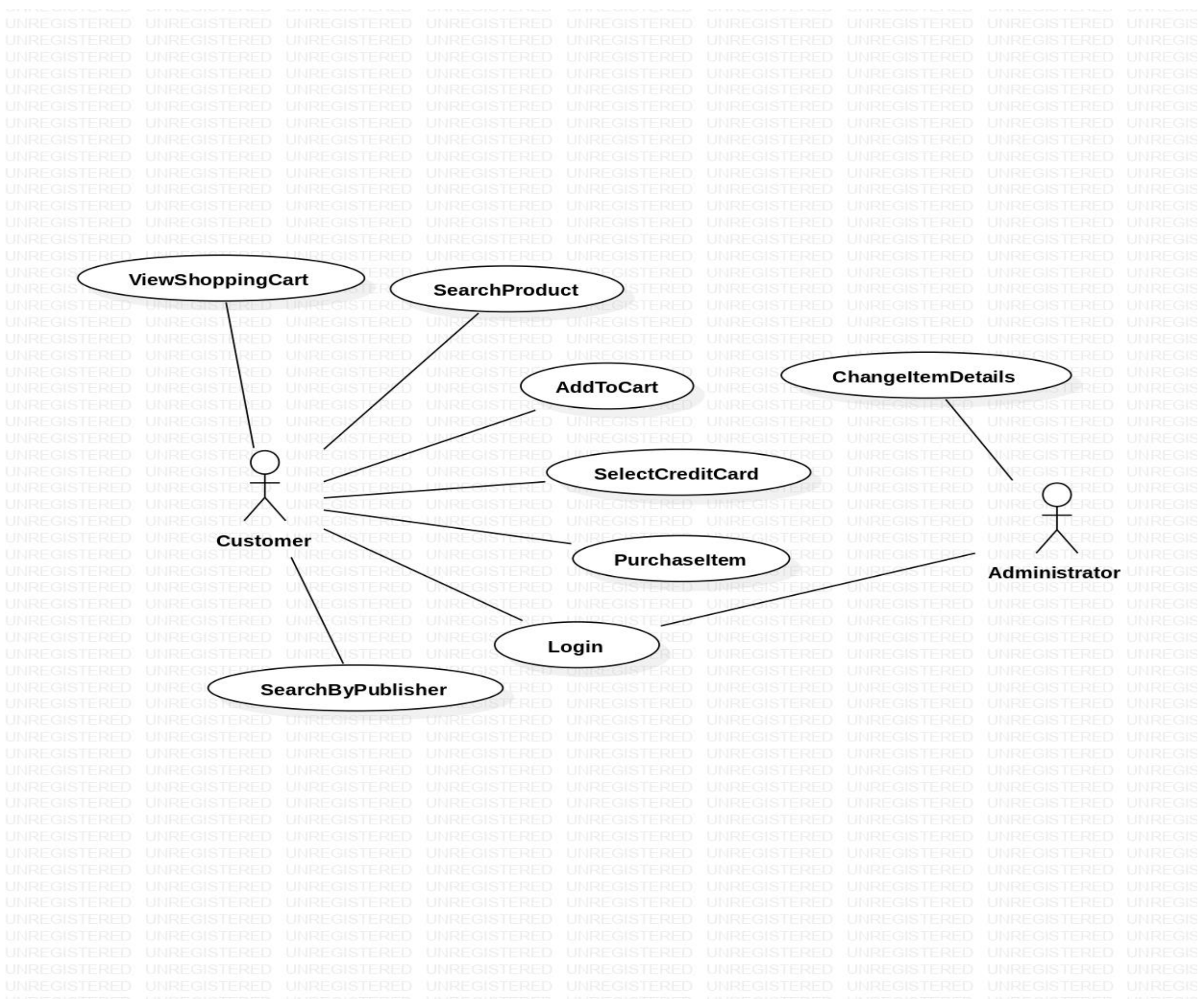
But in our project we have all types of books with the publisher name and category. If any one wants a particular publisher's books then the customer search by publishers name and the same with category also if you want books of any particular category then you can search by category. And you can search from books name also but you have type the correct spelling of books, publisher or category.

## UML Diagrams

UML stands for Unified Modelling Language. It represents a unification of the concepts and notations presented by the three amigos in their respective books. The goal is for UML to become a common language for creating models of object oriented computer software.In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

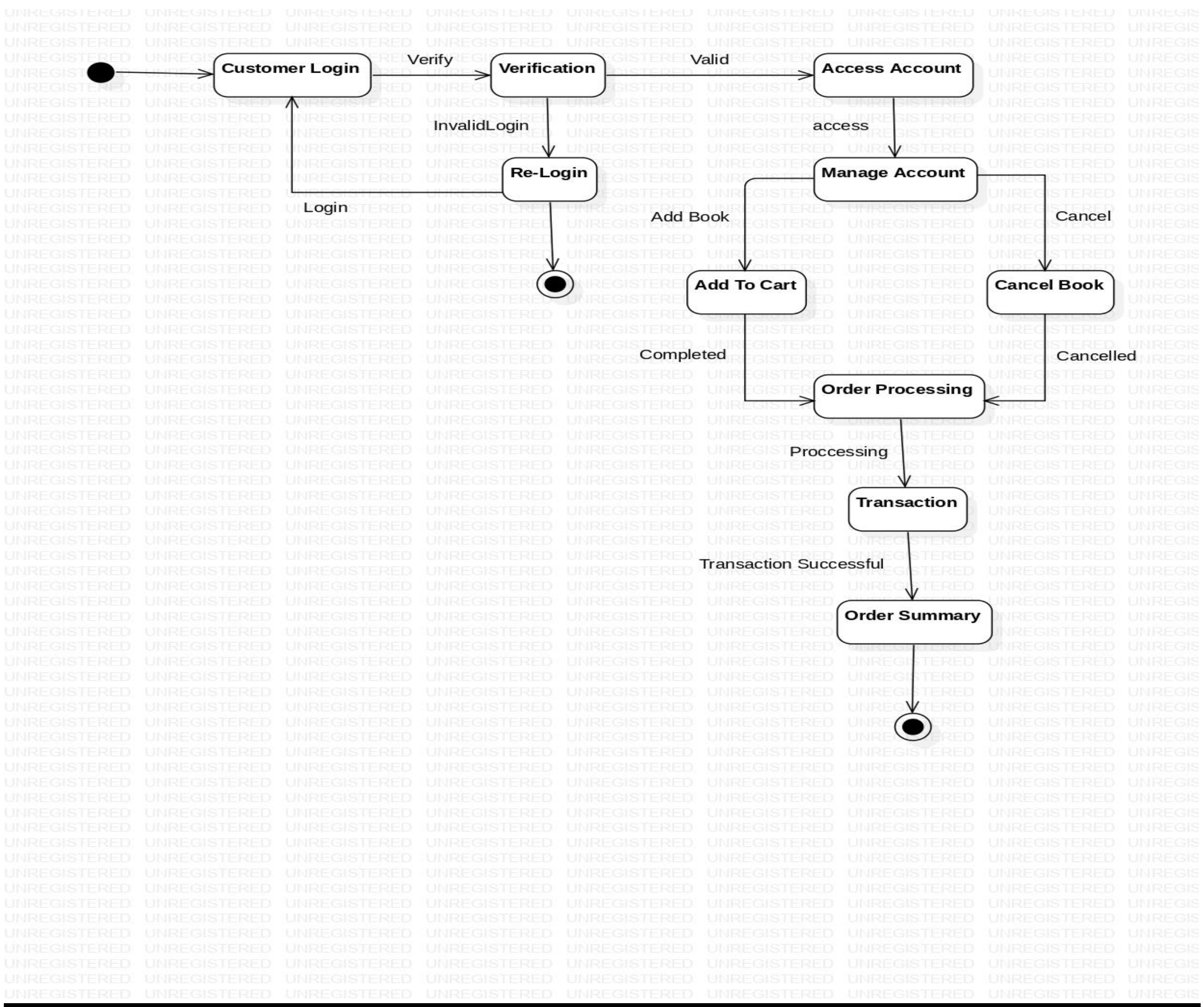
- Use-Case Diagrams
- State-Chart  Diagram
- Sequence Diagram
- Class Diagram

## USE-Case Diagram



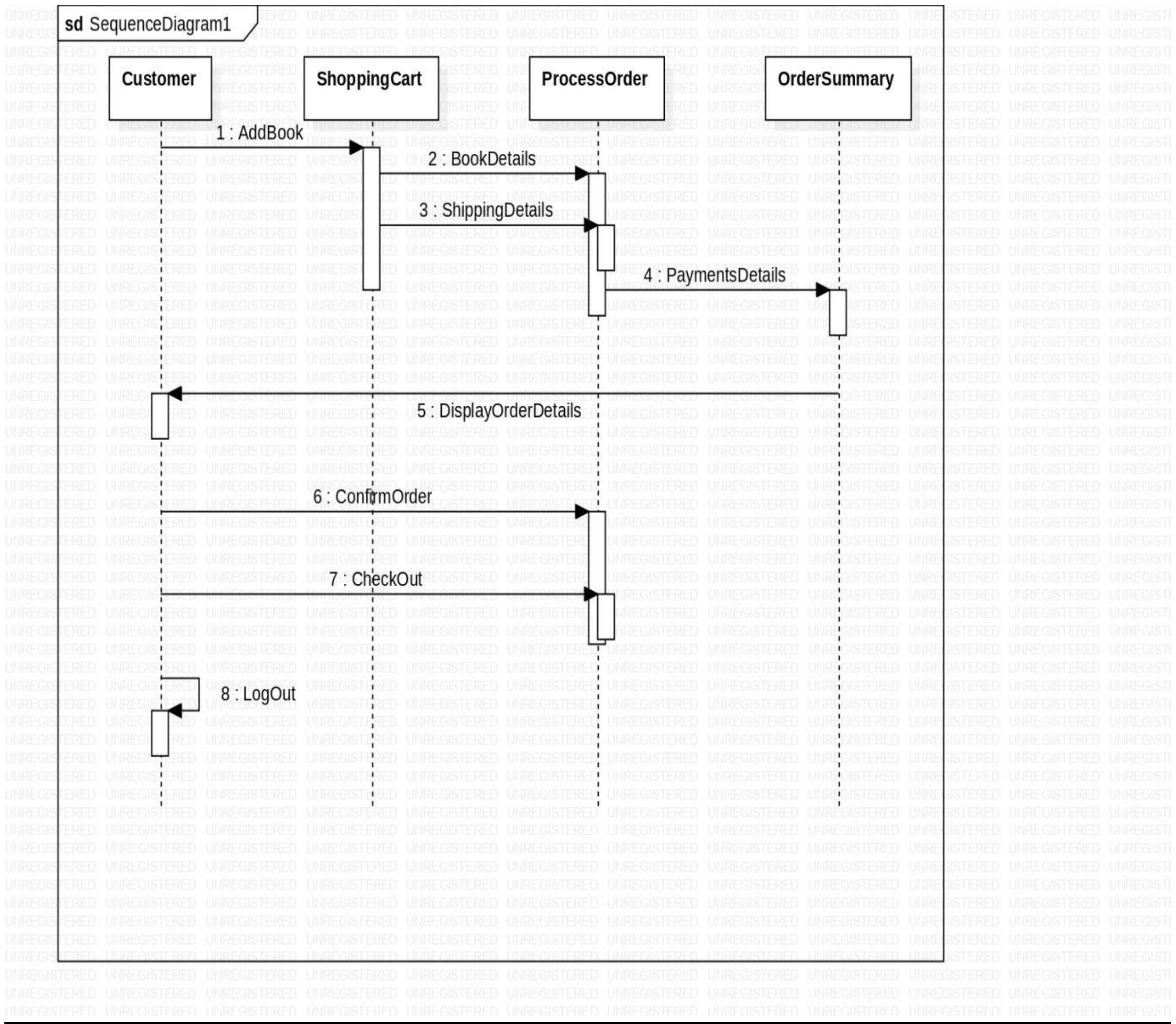
## USE-Case Diagram of Online Book Store

## State-Chart Diagram



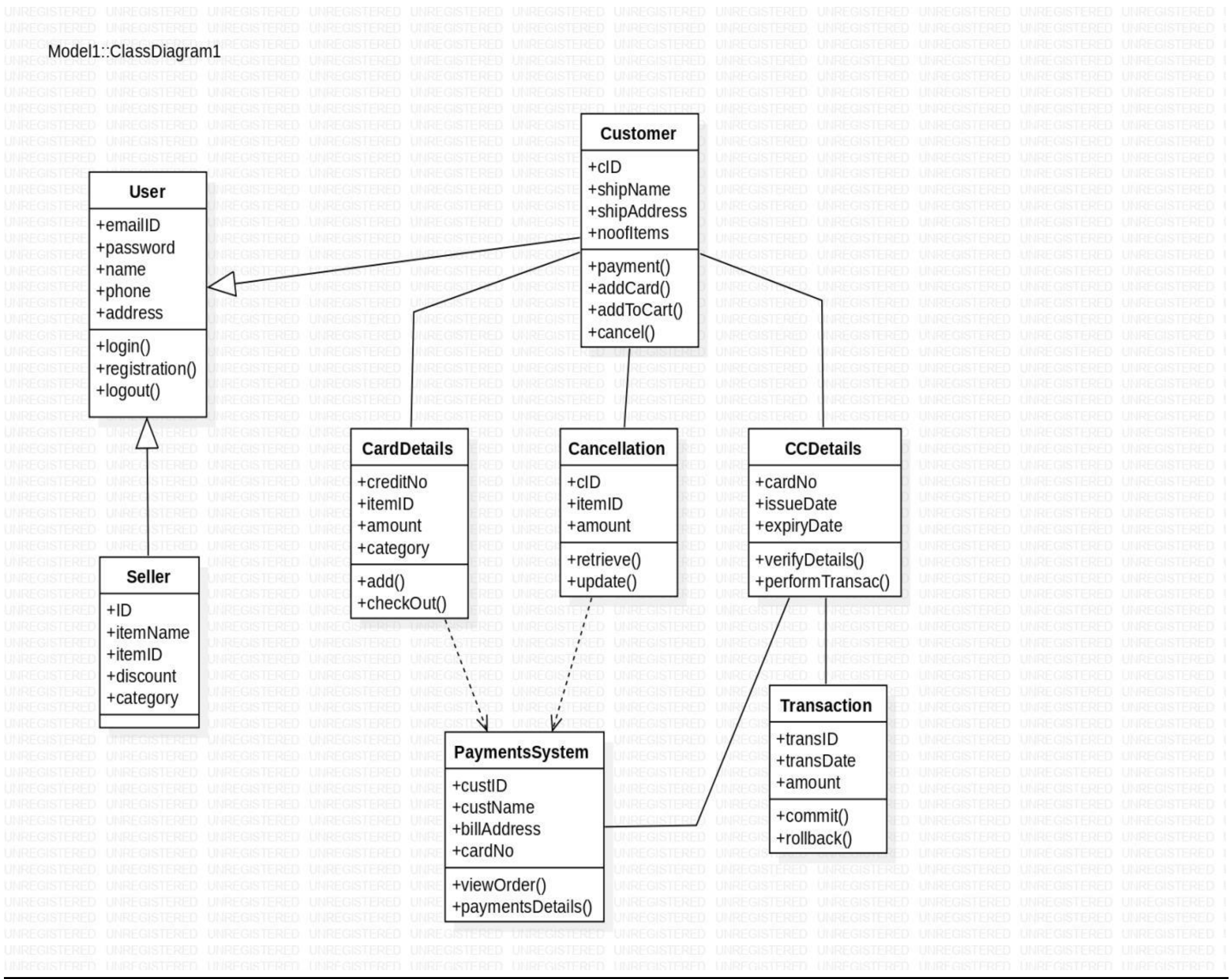
## State-Chart Diagram of Online Book Store

## Sequence Diagram



## Sequence Diagram of Online Book Store

## Class Diagram



## Class Diagram of Online Book Store

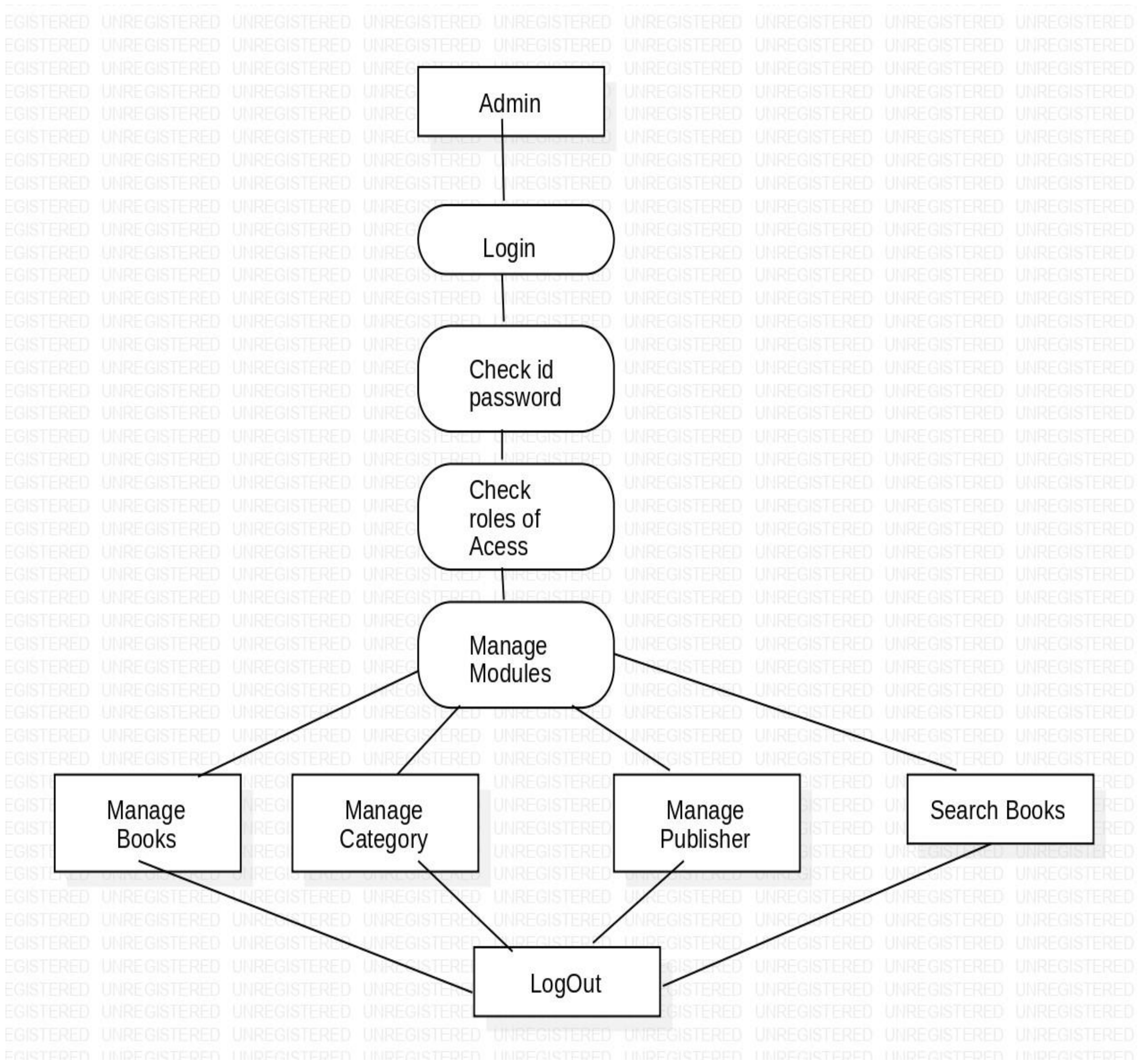
## Problem in Existing System

As with most online sites, you need a lot of trust to use online bookstores. The major problem of the online store is that there is no bookmarks as well. There are no proper application or website to where the students can find study materials for their examination under one roof. Students have to search a lot on the web to get complete study materials as well as in other sites their charge very high costs for books.

## **Proposed Work**

In this project we have to make two modules one module for admin and another module for customer. First we are making admin module in which admin can manage about all the data of books like book category and book publisher how many books are out of stocks and there is one log out button also for log out from that page. We have completed the login module and manage category module yet. In this module admin can add books there category and publishers. After completing of this module we have to complete other module which is for customer. In this we can have login and logout module same as admin but in this module customer have many features like customer can buy the book, he can see transactions details, customer can see there profile in this module and payment details etc. This is all about our project.

## Flow Chart for Proposed Work



## System Requirements

**I.** Software Requirement:

➡Spring Tool Suite(STS)
➡Spring Boot, Hibernate, Tomcat Server version 9.0 and MYSQL server
➡Thymeleaf servlet engine
➡Related Repositories & Dependencies.

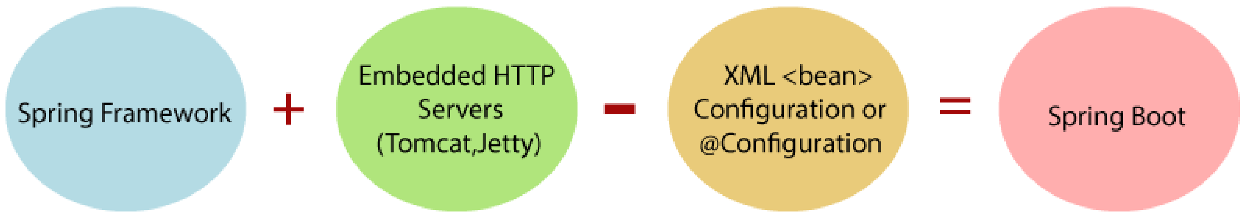
**II.** Hardware Requirement:

➡Processor: Intel i3 (or) Higher
➡Ram: 4 GB (or) Higher
➡Cache:1MB
➡Hard disk: 512 GB

## About Technology

### ➢ Spring Boot

- Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.
- It is a Spring module that provides the **RAD (Rapid Application Development)** feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration.


Spring Framework
+
Embedded HTTP Servers (Tomcat,Jetty)
-
XML <bean> Configuration or @Configuration
=
Spring Boot

- In short, Spring Boot is the combination of **Spring Framework** and **Embedded Servers**.
- In Spring Boot, there is no requirement for XML configuration (deployment descriptor). It uses convention over configuration software design paradigm that means it decreases the effort of the developer.
- We can use Spring **STS IDE** or **Spring Initializr** to develop Spring Boot Java applications.

## Advantages of Spring Boot

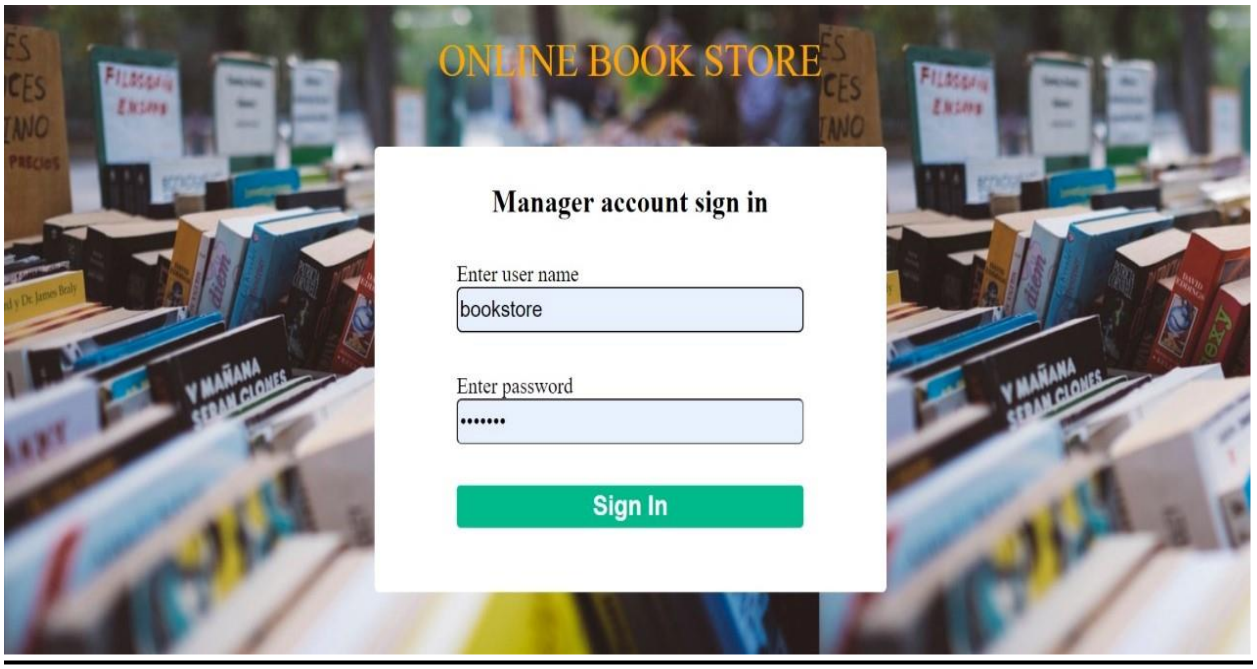
- It creates **stand-alone** Spring applications that can be started using Java **-jar**.
- There is no requirement for XML configuration.
- It tests web applications easily with the help of different Embedded HTTP servers such as Tomcat, Jetty, etc. We don't need to deploy WAR files.
- It provides opinionated 'starter' POMs to simplify our Maven configuration.
- It increases productivity and reduces development time.

## Limitations

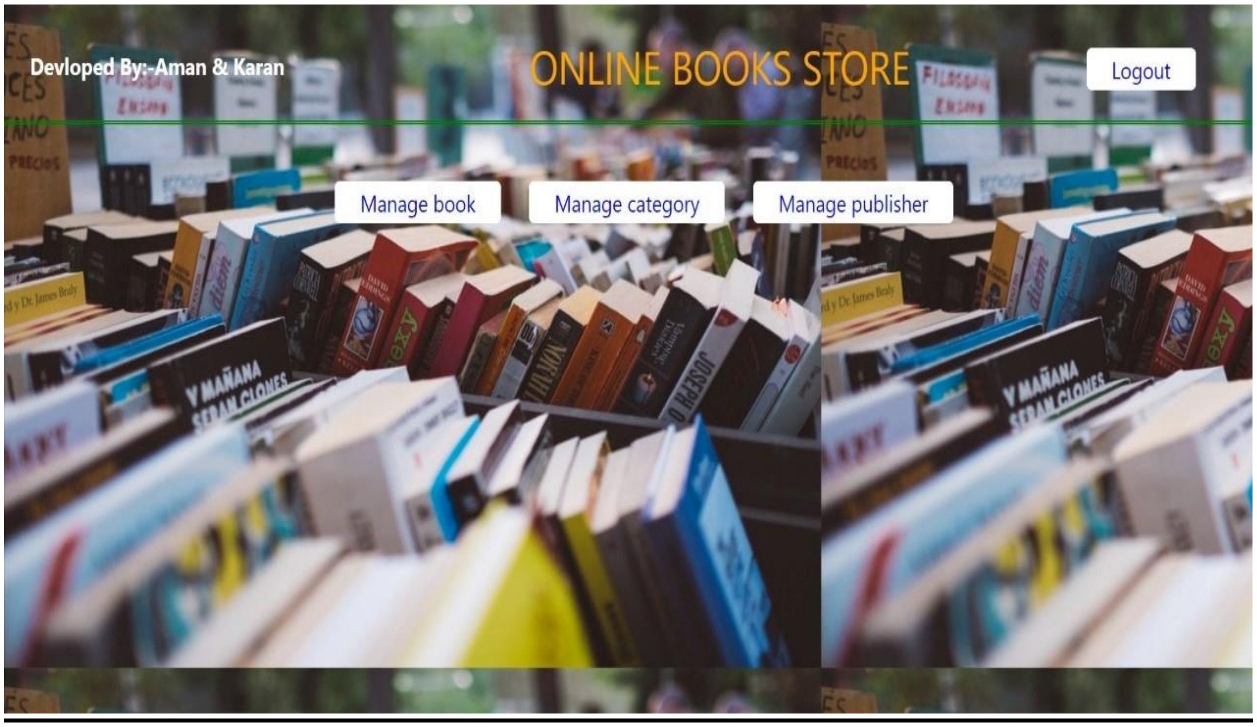
- Cannot be used without internet.
- Users cannot give feedbacks.
- Limited books are only available.
- Error in spelling by the user cannot provide the exact book they want.

## System Design

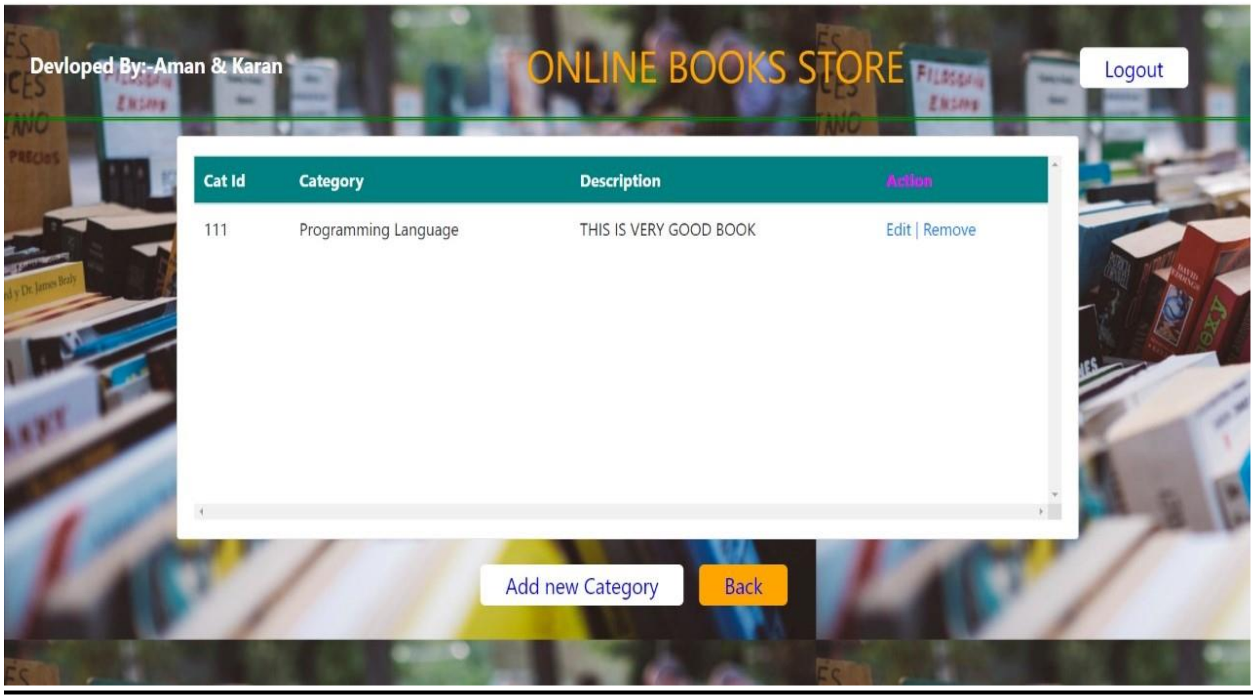
### ➢ Login Page



- **Home Page**

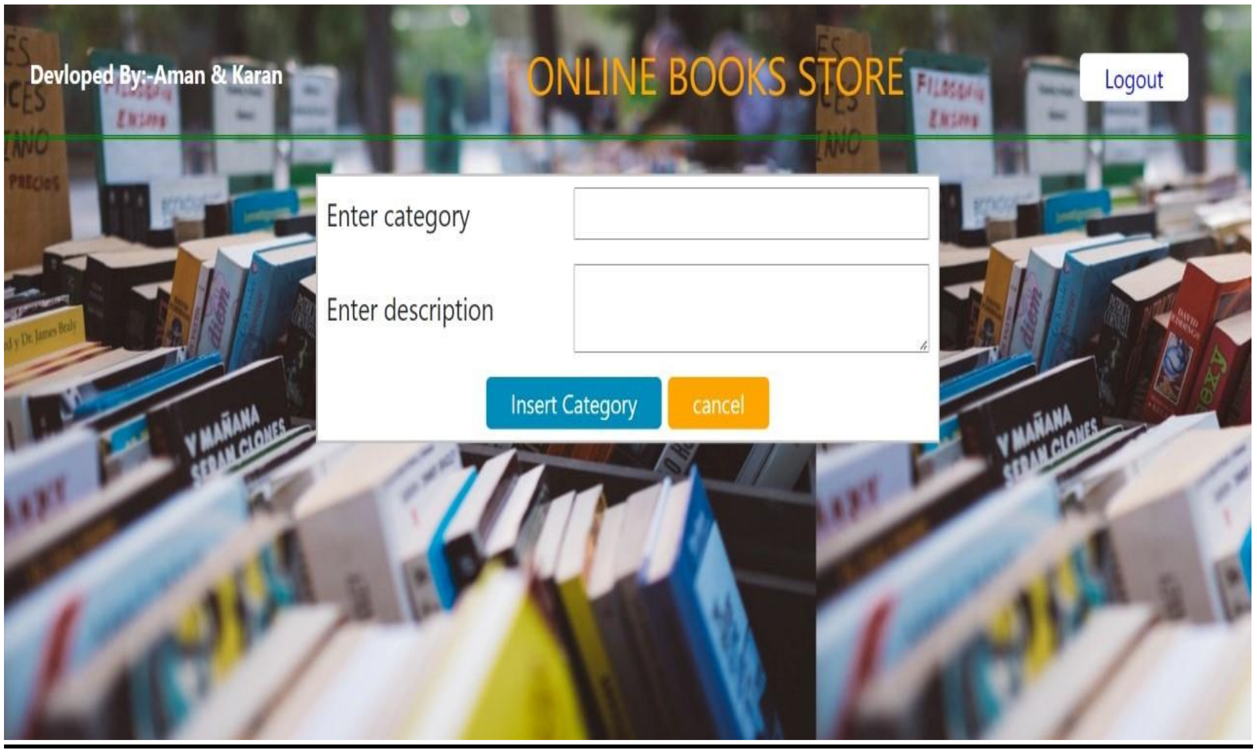

Devloped By:-Aman & Karan
ONLINE BOOKS STORE
Logout
Manage book
Manage category
Manage publisher

## ➢ **Manage Category**

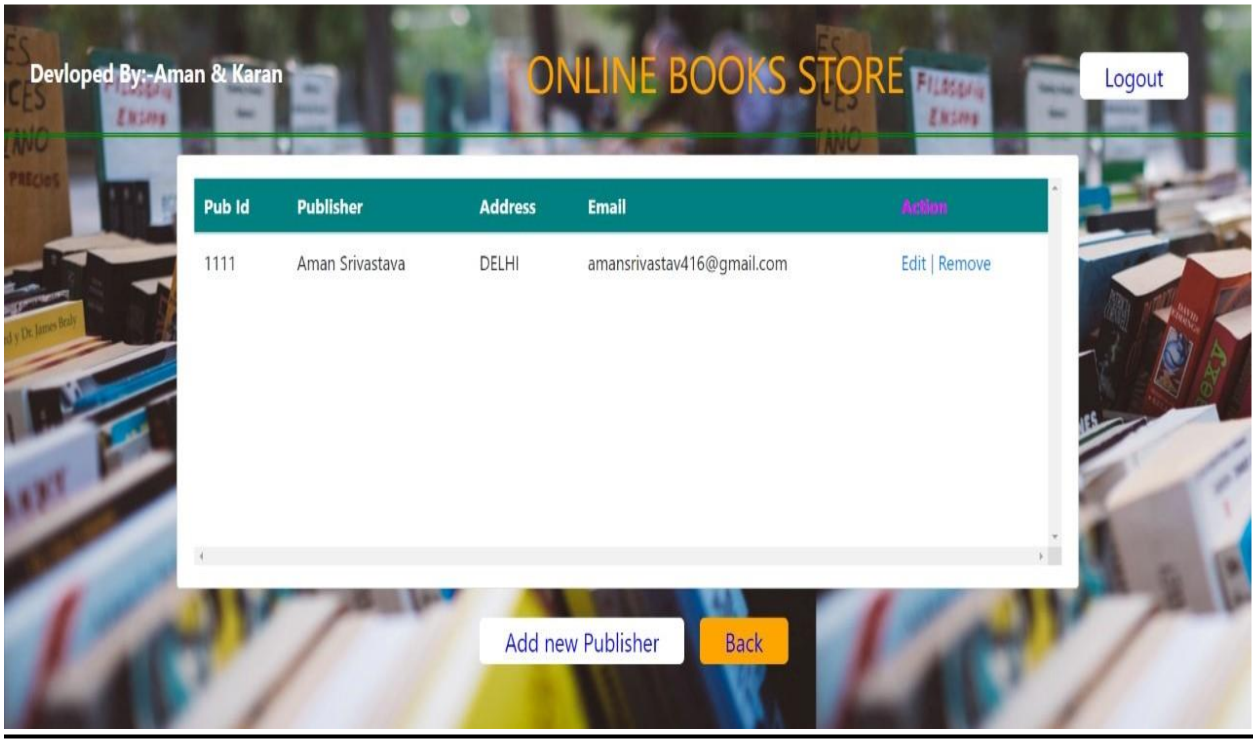


| Cat Id | Category | Description | Action |
|---|---|---|---|
| 111 | Programming Language | THIS IS VERY GOOD BOOK | Edit \| Remove |

## ➢ **Add New Category**

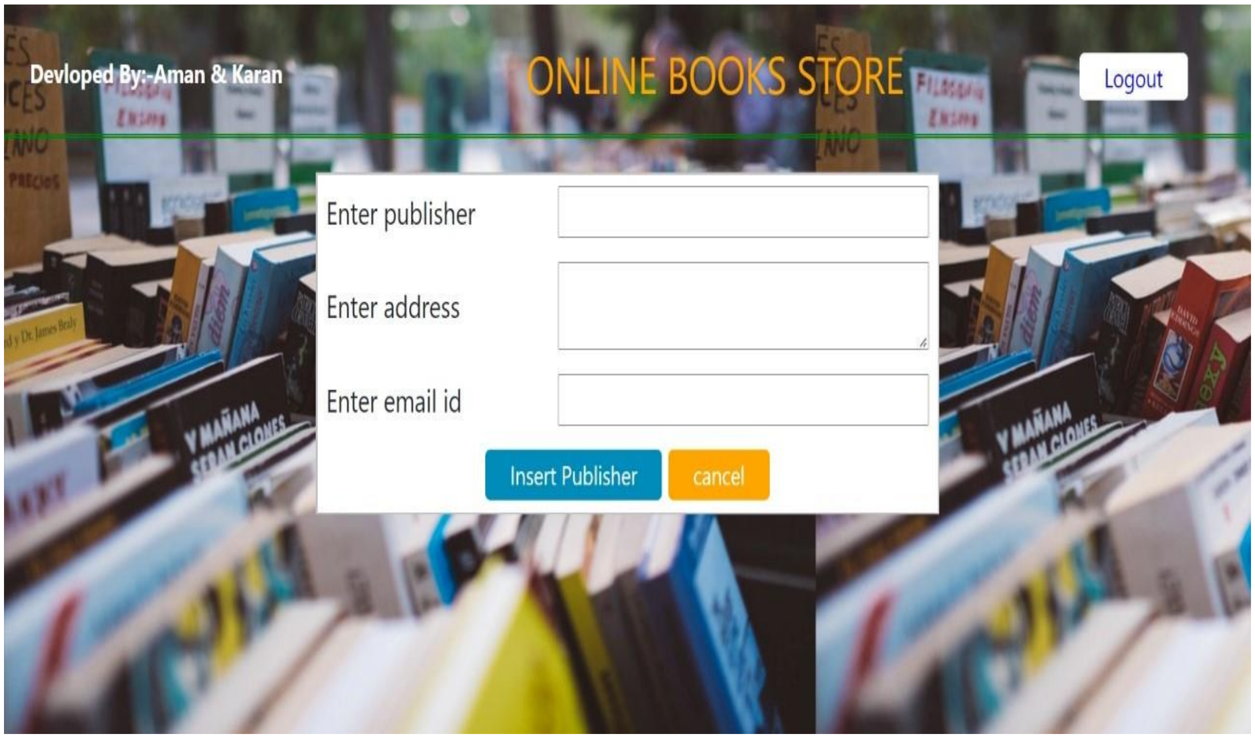


➢ **<u>Manage Publisher</u>**



| Pub Id | Publisher | Address | Email | Action |
|---|---|---|---|---|
| 1111 | Aman Srivastava | DELHI | amansrivastav416@gmail.com | Edit \| Remove |

## ➢ <u>**Add New Publisher**</u>

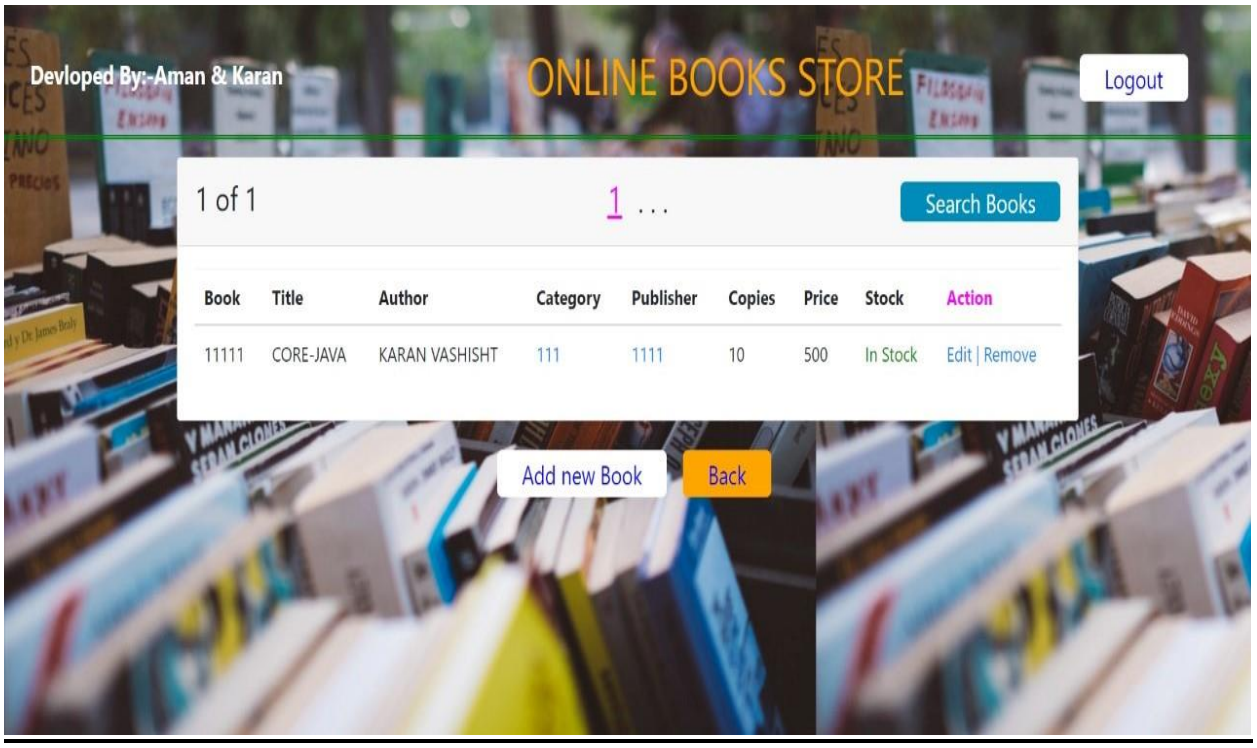


Enter publisher

Enter address

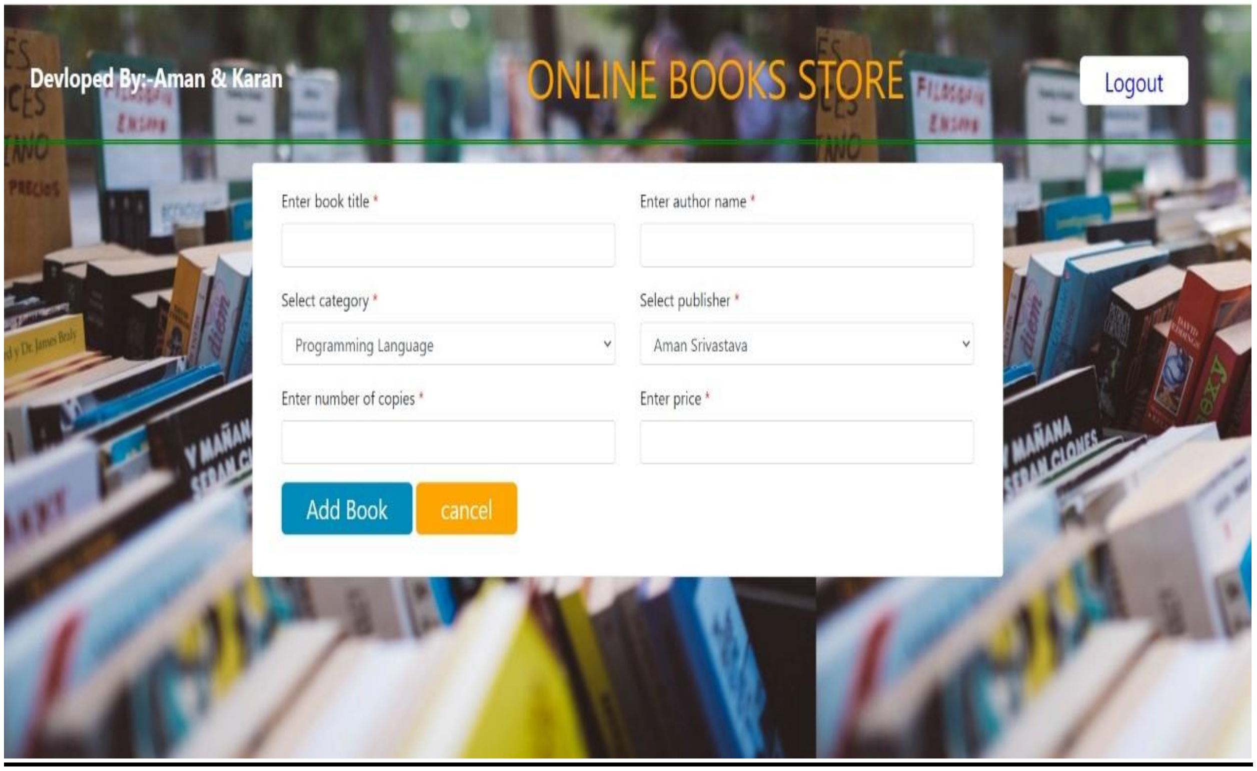
Enter email id

## ➢ **Manage Book**



| Book | Title | Author | Category | Publisher | Copies | Price | Stock | Action |
|---|---|---|---|---|---|---|---|---|
| 11111 | CORE-JAVA | KARAN VASHISHT | 111 | 1111 | 10 | 500 | In Stock | Edit \| Remove |

## ➢ <u>**Add New Publisher**</u>



Enter book title *

Enter author name *

Select category *

Programming Language

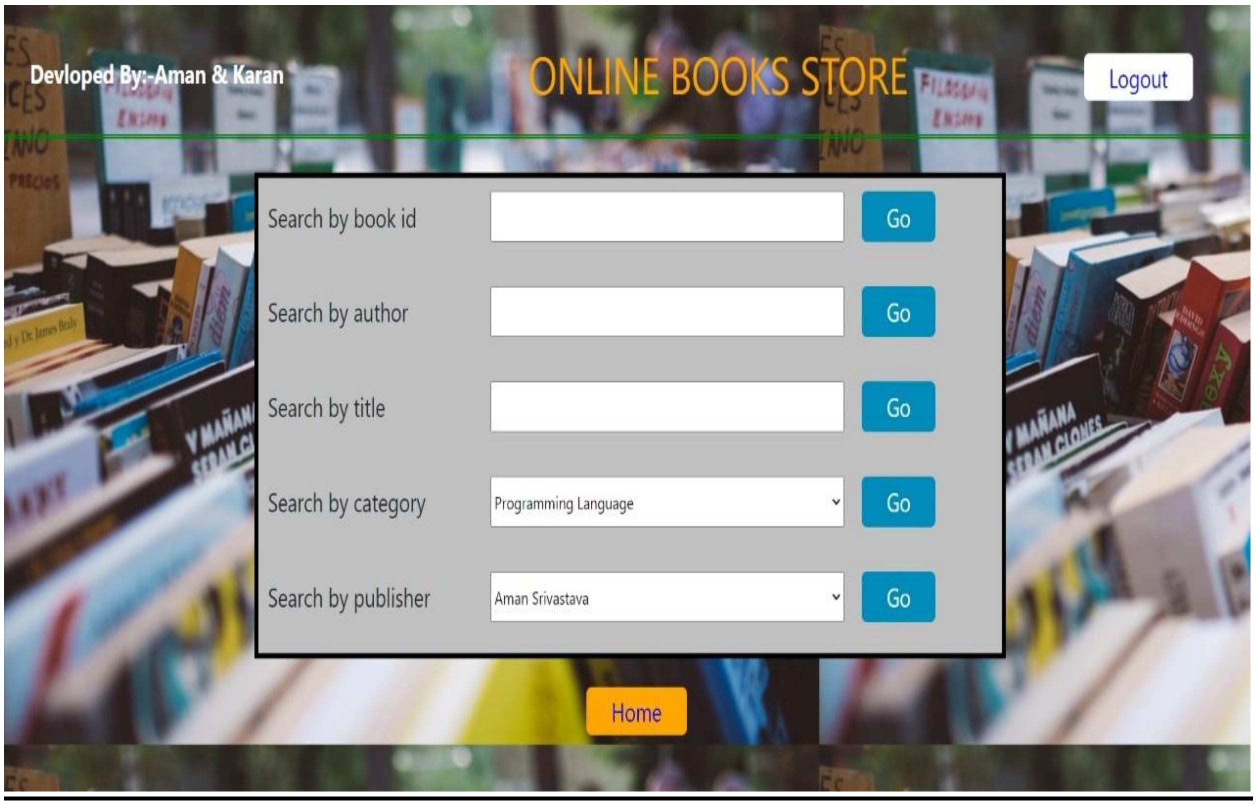
Select publisher *

Aman Srivastava

Enter number of copies *

Enter price *

## ➢ <u>**Add New Publisher**</u>



| | | |
|---|---|---|
| Search by book id | | Go |
| Search by author | | Go |
| Search by title | | Go |
| Search by category | Programming Language | Go |
| Search by publisher | Aman Srivastava | Go |

## Source Code

### ➢ Login.html

```
<html xmlns:th="http://www.thymeleaf.org">
 <head>
  <link th:href='@{/CSS/style.css}' rel='stylesheet'>
  <link th:href='@{/CSS/login.css}' rel='stylesheet'>
 </head>
 <body class='bgi'>
  <div class='dvv'>
   <label  style='color:orange;font-size:3vw'>ONLINE BOOK STORE</label>
  </div>
 <form action='/bookstore/authentication' method='post'>
  <div align='center'>
    <div class='dvadmin'>
     <h2 style='font-size:2.1vw'>Manager account sign in</h2>
     <div class='dvlog'>
       <label class='la'>Enter user name</label><br>
       <input type='text' name='uid' class='tblog'  required>
     </div>
     <div class='dvlog'>
       <label class='la'>Enter password</label><br>
       <input type='password' name='pass' class='tblog'  required>
     </div>
     <div class='dvlog'>
       <button class='btlog'>Sign In</button>
       <div style="text-align:center;margin-top:1vw" >
           <label class='la' th:text="${msg}" style='color:red'></label>
       </div>
     </div>
    </div>
  </div>
  </form>
 </body>
</html>
```

## ➢ <u>**Home.html**</u>

```
<html xmlns:th="http://www.thymeleaf.org">
<div th:insert="/top"></div>
<body class='bgi'>
  <div class='dv'>
   <a href='book/manage?pn=1' class='btl' style='margin-left:2vw'>Manage book</a>
   <a href='category/manage' class='btl' style='margin-left:2vw'>Manage category</a>
   <a href='publisher/manage' class='btl' style='margin-left:2vw'>Manage
publisher</a>
  </div>
 </body>
 </html>
```

➢ <u>**Category-list.html**</u>

```
<html   xmlns:th="http://www.thymeleaf.org">
 <body class='bgi'>
  <form>
  <div th:insert="/top"></div>
  <div class='container'>
  <div class='card'>
  <div class='card-body'>
    <div class='dvv' th:if="${list.empty}">
      <label class='lah' style='color:green'>Sorry category list is empty</label>
    </div>
    <div th:if="${!list.empty}" class='scrollit'>
    <table class='table'>
     <thead>
      <tr>
       <th class='tdr'>Cat Id</th>
       <th class='tdr'>Category</th>
       <th class='tdr'>Description</th>
       <th class='tdr' style='color:magenta'>Action</th>
      </tr>
     </thead>
     <tbody>
      <tr th:each="cat:${list}">
       <td class='tdr' th:text="${cat.catid}"></td>
       <td class='tdr' th:text="${cat.category}"></td>
       <td class='tdr' th:text="${cat.description}"></td>
       <td class='tdr'>
        <a th:href="@{edit(cid=${cat.catid})}">Edit |</a>
        <a th:href="@{remove(cid=${cat.catid})}" onclick="return confirm('Are you
sure to remove?')">Remove</a>
       </td>
        </tr>
        </tbody>
      </table>
    </div>
    </div>
   </div>
    <div class='dvv'>
     <a class='btl' href='add'>Add new Category</a>
     <a class='btl' href='/bookstore/inventory/home' style='background-
color:orange'>Back</a>
  </div>
   </div>
  </form>
 </body>
 </html>
```

## ➢ <u>**Add-category.html**</u>

```
<html xmlns:th="http://www.thymeleaf.org">
<div th:insert="/top"></div>
<body class='bgi'>
<form action="save" method="post">
<table class='tab' style='background-color:white'>
  <tr>
       <td class='tatd'>Enter category</td>
       <td class='tatd'>
         <input class='tbb' type='text' id='category' name="category" required/>
       </td>
  </tr>
  <tr>
       <td class='tatd'>Enter description</td>
       <td class='tatd'>
         <textarea class='tbb' style="height:12vh" id='description' 
name="description" required>
         </textarea>
       </td>
  </tr>
  <tr>
       <td class='tatd' colspan="2" align="center">
              <button class='btr'>Insert Category</button>
              <input type='button' value='cancel' class='btr' style='background-
color:orange' onclick='history.back()'>
    </td>
  </tr>
 </table>
</form>
</body>
</html>
```

## ➢ Publisher-list.html

```
<html   xmlns:th="http://www.thymeleaf.org">
 <body class='bgi'>
  <form>
  <div th:insert="/top"></div>
  <div class='container'>
  <div class='card'>
  <div class='card-body'>
    <div class='dvv' th:if="${list.empty}">
      <label class='lah' style='color:green'>Sorry publisher list is empty</label>
    </div>
    <div th:if="${!list.empty}" class='scrollit'>
    <table class='table'>
     <thead>
      <tr>
       <th class='tdr'>Pub Id</th>
       <th class='tdr'>Publisher</th>
       <th class='tdr'>Address</th>
       <th class='tdr'>Email</th>
       <th class='tdr' style='color:magenta'>Action</th>
      </tr>
     </thead>
     <tbody>
      <tr th:each="pub:${list}">
       <td class='tdr' th:text="${pub.pubid}"></td>
       <td class='tdr' th:text="${pub.publisher}"></td>
       <td class='tdr' th:text="${pub.address}"></td>
       <td class='tdr' th:text="${pub.email}"></td>
       <td class='tdr'>
        <a th:href="@{edit(pid=${pub.pubid})}">Edit |</a>
        <a th:href="@{remove(pid=${pub.pubid})}" onclick="return confirm('Are you
sure to remove?')">Remove</a>
       </td>
       </tr>
       </tbody>
      </table>
    </div>
    </div>
   </div>
    <div class='dvv'>
     <a class='btl' href='add'>Add new Publisher</a>
     <a class='btl' href='/bookstore/inventory/home' style='background-
color:orange'>Back</a>
  </div>
   </div>
  </form>
 </body>
 </html>
```

## ➢ Add-publisher.html

```
<html xmlns:th="http://www.thymeleaf.org">
<div th:insert="/top"></div>
<body class='bgi'>
<form action="save" method="post">
<table class='tab' style='background-color:white'>
  <tr>
       <td class='tatd'>Enter publisher</td>
       <td class='tatd'>
         <input class='tbb' type='text' name="publisher" required/>
       </td>
  </tr>
  <tr>
       <td class='tatd'>Enter address</td>
       <td class='tatd'>
         <textarea class='tbb' style="height:12vh" name="address" required>
         </textarea>
       </td>
  </tr>
  <tr>
       <td class='tatd'>Enter email id</td>
       <td class='tatd'>
         <input class='tbb' type='text' name="email" required/>
       </td>
  </tr>
  <tr>
       <td class='tatd' colspan="2" align="center">
             <button class='btr'>Insert Publisher</button>
             <input type='button' value='cancel' class='btr' style='background-
color:orange' onclick='history.back()'>
    </td>
  </tr>
 </table>
</form>
</body>
</html>
```

## ➢ Book-list.html

```
<html xmlns:th="http://www.thymeleaf.org">
<div th:insert="/top"></div>
<body class='bgi'>
  <form>
  <div class='container'>
  <div class='card'>
  <div class='card-header'>
   <table style='width:100%'>
    <tr>
     <td style='width:33%'>
       <label style='font-size:2vw' th:text="${pn+' of '+tp}"></label>
     </td>
     <td style='width:33%' align="center">
       <a th:if="${tp<=5}" th:class='${pn==i?"anc":"an"}'
th:href='@{manage(pn=${i})}' th:each="i:${#numbers.sequence( 1,tp,1)}"
th:text="${i}"></a>
       <a th:if="${tp>5 and pn<=5}" th:class='${pn==i?"anc":"an"}'
th:href='@{manage(pn=${i})}' th:each="i:${#numbers.sequence( 1,5,1)}"
th:text="${i}"></a>
       <a th:if="${tp>5 and pn>5}" th:class='${pn==i?"anc":"an"}'
th:href='@{manage(pn=${i})}' th:each="i:${#numbers.sequence( 1,4,1)}"
th:text="${i}"></a>
       <a th:if="${tp>5 and pn>5}" class='anc'  th:href='@{manage(pn=${pn})}'
th:text="${pn}"></a>
       <label style='font-size:2vw;margin-left:1vw'> . . .</label>
       <a class='an' th:href="@{manage(pn=${pn+1})}" th:if="${tp>5}"
th:onclick='|return ${pn<tp}|'>Next</a>
     </td>
     <td style='width:33%' align="right">
      <a class='btr' href='search'>Search Books</a>
     </td>
    </tr>
   </table>
  </div>
  <div class='card-body'>
     <div class='dv' th:if="${list.empty}">
      <label class='lah' style='color:red'>Book list is empty</label>
     </div>
   <div class='table'  th:if="${!list.empty}">
    <table class='table'>
     <thead>
      <tr>
       <th class='tdr'>Book</th>
       <th class='tdr'>Title</th>
       <th class='tdr'>Author</th>
       <th class='tdr'>Category</th>
       <th class='tdr'>Publisher</th>
       <th class='tdr'>Copies</th>
       <th class='tdr'>Price</th>
       <th class='tdr'>Stock</th>
       <th class='tdr' style='color:magenta'>Action</th>
       </tr>
      </thead>
      <tbody>
        <tr th:each="book:${list}">
       <td class='tdr' th:text="${book.bookid}"></td>
       <td class='tdr' th:text="${book.title}"></td>
       <td class='tdr' th:text="${book.author}"></td>
       <td class='tdr'><a
th:href='@{/bookstore/inventory/category/details(cid=${book.catid})}'
th:text="${book.catid}"></a></td>
       <td class='tdr'><a
th:href='@{/bookstore/inventory/publisher/details(pid=${book.pubid})}'
th:text="${book.pubid}"></a></td>
       <td class='tdr' th:text="${book.copies}"></td>
       <td class='tdr' th:text="${book.price}"></td>
       <td class='tdr' style='color:green' th:if="${book.copies>0}">In Stock</td>
       <td class='tdr' style='color:red' th:if="${book.copies==0}">Out of Stock</td>
       <td class='tdr'>
        <a th:href="@{edit(bid=${book.bookid})}">Edit |</a>
        <a th:href="@{remove(bid=${book.bookid})}" onclick="return confirm('Are you
sure to remove?')">Remove</a>
       </td>
        </tr>
        </tbody>
       </table>
     </div>
     </div>
    </div>
     <div class='dvv'>
      <a class='btl' href='add'>Add new Book</a>
      <a class='btl'  href='/bookstore/inventory/home' style='background-
color:orange'>Back</a>
   </div>
    </div>
   </form>
 </body>
 </html>
```

## ➢ **Add-book.html**

```
<html xmlns:th="http://www.thymeleaf.org">
<div th:insert="/top"></div>
<body class='bgi'>
<div class='container'>
 <div class='card mx-auto col-md-10'>
  <div class='card-body'>
   <form action="save" method="post">
     <div class='row form-group'>
      <div class='col-md-6'>
             <label for="title">Enter book title</label><span> *</span>
             <input type='text' name="title" id='title' class='form-control'
required/>
      </div>
      <div class='col-md-6'>
       <label for="author">Enter author name</label><span> *</span>
             <input type='text' name="author" id='author' class='form-control'
required/>
     </div>
    </div>
    <div class='row form-group'>
      <div class='col-md-6'>
             <label for="title">Select category</label><span> *</span>
             <select name="catid" class='form-control' required>
              <option th:each="cat:${clist}" th:value="${cat.catid}"
th:text="${cat.category}"></option>
             </select>
      </div>
      <div class='col-md-6'>
       <label for="author">Select publisher</label><span> *</span>
             <select name="pubid" class='form-control' required>
              <option th:each="pub:${plist}" th:value="${pub.pubid}"
th:text="${pub.publisher}"></option>
             </select>
     </div>
    </div>
    <div class='row form-group'>
     <div class='col-md-6'>
             <label for="copies">Enter number of copies</label><span> *</span>
             <input type='text'  name="copies" id='copies' class='form-control'
required/>
      </div>
      <div class='col-md-6'>
       <label for="price">Enter price</label><span> *</span>
             <input type='text' name="price" id='price' class='form-control'
required/>
     </div>
    </div>
    <div class='row form-group'>
     <div class='col-md-10'>
             <button class='btr'>Add Book</button>
             <input type='button' value='cancel' class='btr' style='background-
color:orange' onclick='history.back()'>
     </div>
    </div>
   </form>
  </div>
 </div>
</div>
</body>
</html>
```

## ➢ Search-book.html

```
<html xmlns:th="http://www.thymeleaf.org">
<div th:insert="/top"></div>
<body class='bgi'>
 <table class='searchta'>
  <tr>
  <td>
    <form action="/searchby/bookid/manager">
    <table style="width:100%">
     <tr>
                <td style="width:30%" class='tatd la'>Search by book id</td>
                <td style="width:50%" class='tatd'><input type='text' name='bookid'
class='tbb' required></td>
                <td style="width:20%" class='tatd'><button class='btr'>Go</button></td>
        </tr>
    </table>
    </form>
   </td>
   </tr>
   <tr>
   <td>
    <form action="search/byauthor">
    <table style="width:100%">
     <tr>
                <td style="width:30%" class='tatd la'>Search by author</td>
                <td style="width:50%" class='tatd'><input type='text' name='author'
class='tbb' required></td>
                <td style="width:20%" class='tatd'><button class='btr'>Go</button></td>
        </tr>
    </table>
    </form>
   </td>
  </tr>
  <tr>
   <td>
    <form action="/searchby/title/manager">
    <table style="width:100%">
     <tr>
                <td style="width:30%" class='tatd la'>Search by title</td>
                <td style="width:50%" class='tatd'><input type='text' name='title'
class='tbb' required></td>
                <td style="width:20%" class='tatd'><button class='btr'>Go</button></td>
        </tr>
    </table>
    </form>
   </td>
  </tr>
  <tr>
   <td>
    <form action="search/bycategory">
    <table style='width:100%'>
      <tr>
      <td style="width:30%" class='tatd la'>Search by category</td>
      <td style="width:50%" class='tatd'>
      <select name="catid" class='tbb' required>
                <option th:each="cat:${clist}" th:value="${cat.catid}"
th:text="${cat.category}"></option>
      </select></td>
      <td style="width:20%" class='tatd'><button class='btr'>Go</button></td>
        </tr>
       </table>
      </form>
      </td>
   </tr>
   <tr>
   <td>
    <form action="/searchby/publisher/manager">
    <table style="width:100%">
     <tr>
      <td style="width:30%" class='tatd la'>Search by publisher</td>
      <td style="width:50%" class='tatd'><select name="publisher" class='tbb'
required>
              <option th:each="pub:${plist}" th:value="${pub.pubid}"
th:text="${pub.publisher}"></option>
      </select></td>
      <td style="width:20%" class='tatd'><button class='btr'>Go</button></td>
        </tr>
        </table>
        </form>
       </td>
       </tr>
   </table>
   <div class='dvv'>
   <a class='btl' href='/bookstore/inventory/home' style='background-
color:orange'>Home</a>
   </div>
</body>
</html>
```

➢ **<u>LoginController.java</u>**

```
package com.cetpa.controllers;

import javax.servlet.http.HttpSession;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("bookstore")
public class LoginController
{
	@RequestMapping("")
	public String getAdminLoginForm()
	{
		return "login";
	}
	@RequestMapping("authentication")
	public String validateAdminLogin(String uid,String pass,Model model,HttpSession ses)
	{
		if(uid.equals("bookstore") && pass.equals("project"))
		{
			ses.setAttribute("name", "Store Manager");
			return "redirect:inventory/home";
		}
		model.addAttribute("msg","Username or password is incorrect");
		return "login";
	}
	@RequestMapping("inventory/home")
	public String getAdminHome()
	{
		return "/home/home";
	}
	@RequestMapping("inventory/logout")
	public String logout(HttpSession ses)
	{
		ses.invalidate();
		return "redirect:/bookstore";
	}
}
```

## ➢ <u>**CategoryController.java**</u>

```
package com.cetpa.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import com.cetpa.models.Book;
import com.cetpa.models.BookCategory;
import com.cetpa.models.BookPublisher;
import com.cetpa.service.BookService;
import com.cetpa.service.CategoryService;


@Controller
@RequestMapping("bookstore/inventory/category")
public class CategoryController
{
        @Autowired
        private CategoryService service;

        @RequestMapping("manage")
        public String manageCategory(Model model)
        {
                List<BookCategory> list=service.getList();
                model.addAttribute("list",list);
                return "category/category-list";
        }
        @RequestMapping("add")
        public String addCategory()
        {
     return "category/add-category";
        }
        @RequestMapping("save")
        public String saveCategory(BookCategory cat)
        {
                service.saveCategory(cat);

                return "redirect:manage";
        }
}
```

## ➢ **PublisherController.java**

```
package com.cetpa.publisher.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

import com.cetpa.models.BookCategory;
import com.cetpa.models.BookPublisher;
import com.cetpa.publisher.service.PublisherService;


@Controller
@RequestMapping("bookstore/inventory/publisher")
public class PublisherController
{
	@Autowired
	private PublisherService service;

	@RequestMapping("manage")
	public String managePublisher(Model model)
	{
		List<BookPublisher> list=service.getList();
		model.addAttribute("list",list);
		return "publisher/publisher-list";
	}
	@RequestMapping("add")
	public String getAddPublisher()
	{
		return "publisher/add-publisher";
	}
	@RequestMapping("save")
	public String savePublisher(BookPublisher pub)
	{
		service.savePublisher(pub);
		return "redirect:manage";
	}
	@RequestMapping("remove")
	public String removePublisher(int pid)
	{
		service.deletePublisher(pid);
		return "redirect:manage";
	}
	@RequestMapping("edit")
	public String getEdit(int pid,Model model)
	{
		BookPublisher pub=service.getPublisher(pid);
		model.addAttribute("cat",pub);
		return "publisher/edit-publisher";
	}
	@RequestMapping("update")
	public String updatePublisher(BookPublisher pub)
	{
		service.updatePublisher(pub);
		return "redirect:manage";
	}
	@RequestMapping("details")
	public String getPublisher(int pid,Model model)
	{
		BookPublisher pub=service.getPublisher(pid);
		model.addAttribute("pub",pub);
		return "publisher/publisher-details";
	}
}
```

➢ **<u>BookController.java</u>**

```
package com.cetpa.book.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

import com.cetpa.book.service.BookService;
import com.cetpa.category.service.CategoryService;
import com.cetpa.models.Book;
import com.cetpa.models.BookCategory;
import com.cetpa.models.BookPublisher;
import com.cetpa.publisher.service.PublisherService;

@Controller
@RequestMapping("bookstore/inventory/book")
public class BookController
{
	@Autowired
	private BookService service;
	@Autowired
	private CategoryService cservice;
	@Autowired
	private PublisherService pservice;

	@RequestMapping("manage")
	public String getList(Model model,int pn)
	{
		Page<Book> plist=service.getList(pn);
		model.addAttribute("tp",plist.getTotalPages());
		model.addAttribute("pn",pn);
		List<Book> list=plist.toList();
		model.addAttribute("list",list);
		return "book/book-list";
	}
	@RequestMapping("add")
	public String getAddBookView(Model model)
	{
		List<BookCategory> clist=cservice.getList();
		List<BookPublisher> plist=pservice.getList();
		model.addAttribute("clist",clist);
		model.addAttribute("plist",plist);
		return "book/add-book";
	}
	@RequestMapping("save")
	public String saveBook(Book book)
	{
		service.saveBook(book);
		return "redirect:manage?pn=1";
	}
	@RequestMapping("edit")
	public String getEditBookView(Model model,int bid)
	{
		Book book=service.getBook(bid);
		List<BookCategory> clist=cservice.getList();
		List<BookPublisher> plist=pservice.getList();
		model.addAttribute("book",book);
		model.addAttribute("clist",clist);
		model.addAttribute("plist",plist);
		return "book/edit-book";
	}
	@RequestMapping("update")
	public String updateBook(Book book)
	{
		service.updateBook(book);
		return "redirect:manage?pn=1";
	}
}
```

➢ **SearchBookController.java**

```
package com.cetpa.book.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;

import com.cetpa.book.service.BookService;
import com.cetpa.category.service.CategoryService;
import com.cetpa.models.Book;
import com.cetpa.models.BookCategory;
import com.cetpa.models.BookPublisher;
import com.cetpa.publisher.service.PublisherService;

@Controller
@RequestMapping("bookstore/inventory/book/search")
public class SearchBookController
{
	@Autowired
	private BookService service;
	@Autowired
	private CategoryService cservice;
	@Autowired
	private PublisherService pservice;

	@RequestMapping("")
	public String getList(Model model)
	{
		List<BookCategory> clist=cservice.getList();
		List<BookPublisher> plist=pservice.getList();
		model.addAttribute("clist",clist);
		model.addAttribute("plist",plist);
		return "search/search-books";
	}
	@RequestMapping("bycategory")
	public String getListByCategory(Model model,int catid)
	{
		List<Book> list=service.getListByCategory(catid);
		model.addAttribute("list",list);
		model.addAttribute("by", "Category");
		return "search/search-book-list";
	}
	@RequestMapping("byauthor")
	public String getListByAuthor(Model model,String author)
	{
		List<Book> list=service.getListByAuthor(author);
		model.addAttribute("list",list);
		model.addAttribute("by", "Author");
		return "search/search-book-list";
	}
}
```

# References

[1] Ms. Pragati Bagmare1 , Ms. Shraddha Girhepunje2, Ms. Priya Bisen, “Research Paper on Online Bookshop Management System”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 5., Issue 4., 2017, page no. 115-117.

[2] Fatin Najwa Binti Abdullah Sani1 , Hani Malini binti Majek2 , Umairah binti Ahmad Khairudin3 , Abdul Rahman bin Ahmad Dahlan4, “e-Bookstore: Opening Door to the Garden of Knowledge”, International Journal of Scientific and Research Publications, Volume 7, Issue 6, June 2017 ,page no. 2250-3153.

[3] Vamsi Krishna Mummaneni ,A Report Submitted in partial fulfillment of the requirements of the degree of Master of Software Engineering.'

[4] Online bookstore - A new trend in textbook sales management for services marketing Prathamesh Muzumdar The University of Texas at Arlington.

[5] Chris Richardson. Untangling enterprise Java. Queue. Volume 4, Issue 5 (June 2006). Component Technologies. Pages:36-44. 2006. ISSN: 1542-7730