4. Develop a web application using Spring Boot to perform CRUD operations on Book information. The application should use Thymeleaf for the view layer, Spring MVC for the controller layer, Spring Data JPA and Hibernate for data access, and MySQL as the database. The application should manage book attributes such as title, authors, edition, publication and price

Book.java

```
package com.example.demo;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Book {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	Long id;
	String name;
	String author;
	int price;

	public Book() {
		// TODO Auto-generated constructor stub
	}

	public Long getId() {
		return id;
	}

	public void setId(Long id) {
		this.id = id;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getAuthor() {
		return author;
	}

	public void setAuthor(String author) {
		this.author = author;
	}

	public int getPrice() {
```

```
		return price;
	}

	public void setPrice(int price) {
		this.price = price;
	}


}
```

BookRepository.java

```
package com.example.demo;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {

}
```

BookManager.java

```
package com.example.demo;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class BookManager {

	@Autowired
	BookRepository repository;

	public List<Book> getAllBook(){
		return repository.findAll();
	}

	public Book getBookById(Long id) {
		return repository.findById(id).get();
	}

	public void saveBook(Book book) {
		repository.save(book);
	}

	public void deleteBook(Long id) {
		repository.deleteById(id);
	}
}
```

AppController.java

```
package com.example.demo;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class AppController {

	@Autowired
	BookManager manager;

	@GetMapping("/")
	public String showHomePage(Model model) {
		model.addAttribute("booklist",manager.getAllBook());
		return "index";
	}

	@GetMapping("/new")
	public String showCreatePage() {
		return "create_book_form";
	}

	@PostMapping("/save")
	public String saveBook(@ModelAttribute Book book) {
		manager.saveBook(book);
		return "redirect:/";
	}

	@GetMapping("/edit/{id}")
	public String editBook(@PathVariable Long id, Model model) {
		model.addAttribute("staff", manager.getBookById(id));
		return "edit_form";

	}

	@GetMapping("/delete/{id}")
	public String deleteBook(@PathVariable Long id, Model model) {
		manager.deleteBook(id);
		return "redirect:/";

	}


}
```

Edit the **application.properties** file available under **/src/main/resources** to add the following three lines (change the database name, username and password for MySQl server accordingly)

```
spring.datasource.url=jdbc:mysql://localhost:3306/spring
spring.datasource.username=root
spring.datasource.password=password
```

Thyemeleaf Pages to implement the View

index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Book List</title>
</head>
<body>
<div align="center">
<h1>Book List</h1>
<br/>
<a href="/new">Create New Book</a>
<table border="1" cellpadding="10" cellspacing="0">
<tr>
      <th>ID</th>
      <th>Name</th>
      <th>Author</th>
      <th>Price</th>
      <th>Action</th>
</tr>
<tr th:each ="book:${booklist}">
      <td th:text="${book.id}"/>
      <td th:text="${book.name}"/>
      <td th:text="${book.author}"/>
      <td th:text="${book.price}"/>
      <td>
            <a th:href="@{'/edit/' + ${book.id}}">Edit</a>
            <a th:href="@{'/delete/' + ${book.id}}">Delete</a>
      </td>
      </tr>
</table>
</div>
</body>
</html>
```

create_book_form.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Create New Staff</title>
</head>
<body>
<div align="center">
<h1>Create New Staff</h1>
<br/>
<form action="/save" method="post">
<table border="0" cellpadding="10" cellspacing="10">
      <tr>
            <td>Name</td>
            <td><Input type="text" name="name"/></td>
      </tr>
      <tr>
            <td>Author</td>
            <td><Input type="text" name="author"/></td>
      </tr>
      <tr>
            <td>Price</td>
            <td><Input type="text" name="price"/></td>
      </tr>
      <tr>
            <td colspan="2" align="center">
                  <button type="submit">Save</button>
            </td>
      </tr>
</table>
</form>
</div>
</body>
</html>
```

edit_form.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Edit Staff Details</title>
</head>
<body>
<div align="center">
<h1>Edit Staff Details</h1>
<br/>
<form action="#" th:action="@{/save}" th:object="${staff}"
method="post">
<table border="0" cellpadding="10" cellspacing="0">
      <tr>
            <td>ID</td>
```

```
            <td><input type="text" th:field="*{id}"
readonly="readonly"/></td>
      </tr>
      <tr>
            <td>Name</td>
            <td><input type="text" th:field="*{name}"/></td>
      </tr>
      <tr>
            <td>Author</td>
            <td><input type="text" th:field="*{author}"/></td>
      </tr>
      <tr>
            <td>Price</td>
            <td><input type="text" th:field="*{price}"/></td>
      </tr>
      <tr>
            <td colspan="2" align="center">
                  <button type="submit">Save</button>
            </td>
      </tr>
</table>

</form>
</div>
</body>
</html>
```

MySql server database

- **Logging in to mysql**

  ```
  mysql -u -p student

  password: student
  ```

- **select the database for use**

  ```
  use student;
  ```

- **Query to create the table in MySql**

  ```
  create table book (id integer(4) primary key auto_increment,
  name varchar(50), author varchar(50), price integer(5));
  ```

- **Query to describe the table schema**

  ```
  desc book;
  ```

- **Query to select and display all the rows and columns of the table**

  ```
  select * from book;
  ```