**PROBLEM STATEMENT – 4**

**Configuring Spring Boot Actuator Endpoints – Product – ready services.**

**pom.xml:**

<dependencies>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-data-jpa</artifactId>

    </dependency>

    <dependency>

        <groupId>mysql</groupId>

        <artifactId>mysql-connector-java</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-thymeleaf</artifactId>

    </dependency>

    <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-actuator</artifactId>

    </dependency>

</dependencies>

**application.properties:**

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/Book\_Details

spring.datasource.username=root

spring.datasource.password=Password@12

spring.jpa.hibernate.ddl-auto=update

spring.exposure.include=\*

**Book.java:**

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Book {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    private String bookTitle;

    private String bookPublisher;

    private String bookIsbn;

    private int bookNumberOfPages;

    private int bookYear;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getBookTitle() {

return bookTitle;

}

public void setBookTitle(String bookTitle) {

this.bookTitle = bookTitle;

}

public String getBookPublisher() {

return bookPublisher;

}

public void setBookPublisher(String bookPublisher) {

this.bookPublisher = bookPublisher;

}

public String getBookIsbn() {

return bookIsbn;

}

public void setBookIsbn(String bookIsbn) {

this.bookIsbn = bookIsbn;

}

public int getBookNumberOfPages() {

return bookNumberOfPages;

}

public void setBookNumberOfPages(int bookNumberOfPages) {

this.bookNumberOfPages = bookNumberOfPages;

}

public int getBookYear() {

return bookYear;

}

public void setBookYear(int bookYear) {

this.bookYear = bookYear;

}

}

**BookStoreRepository.java:**

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

import java.util.List;

public interface BookStoreRepository extends JpaRepository<Book, Long> {

    List<Book> findByBookTitle(String bookTitle);

    List<Book> findByBookPublisherLike(String bookPublisher);

    List<Book> findByBookYear(int bookYear);

}

**BookStoreService.java:**

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class BookStoreService {

    @Autowired

    private BookStoreRepository repository;

    public Book addBook(Book book) {

        return repository.save(book);

    }

    public Book updateBook(Book book) {

        return repository.save(book);

    }

    public List<Book> getAllBooks() {

        return repository.findAll();

    }

    public Book getBookById(Long id) {

        return repository.findById(id).orElse(null);

    }

    public void deleteBookById(Long id) {

        repository.deleteById(id);

    }

    public List<Book> getBooksByTitle(String title) {

        return repository.findByBookTitle(title);

    }

    public List<Book> getBooksByPublisher(String publisher) {

        return repository.findByBookPublisherLike(publisher);

    }

    public List<Book> getBooksByYear(int year) {

        return repository.findByBookYear(year);

    }

}

**BookStoreRestController.java:**

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

import java.util.List;

@Controller

@RequestMapping("/books")

public class BookStoreRestController {

    @Autowired

    private BookStoreService service;

    @GetMapping

    public String getAllBooks(Model model) {

        List<Book> books = service.getAllBooks();

        model.addAttribute("books", books);

        return "books";

    }

    @GetMapping("/add")

    public String showAddBookForm(Model model) {

        model.addAttribute("book", new Book());

        return "add-book";

    }

    @PostMapping

    public String addBook(@ModelAttribute Book book) {

        service.addBook(book);

        return "redirect:/books";

    }

    @GetMapping("/edit/{id}")

    showEditBookForm(@PathVariable Long id, Model model) {

        Book book = service.getBookById(id);

        model.addAttribute("book", book);

        return "edit-book";

    }

    @PostMapping("/edit")

    public String updateBook(@ModelAttribute Book book) {

        service.updateBook(book);

        return "redirect:/books";

    }

    @GetMapping("/delete/{id}")

    public String deleteBook(@PathVariable Long id) {

        service.deleteBookById(id);

        return "redirect:/books";

    }

}

**books.html:**

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

    <title>Book List</title>

</head>

<body>

    <h1>Book List</h1>

    <table>

        <thead>

            <tr>

                <th>ID</th>

                <th>Title</th>

                <th>Publisher</th>

                <th>ISBN</th>

                <th>Pages</th>

                <th>Year</th>

                <th>Actions</th>

            </tr>

        </thead>

        <tbody>

            <tr th:each="book : ${books}">

                <td th:text="${book.id}"></td>

                <td th:text="${book.bookTitle}"></td>

                <td th:text="${book.bookPublisher}"></td>

                <td th:text="${book.bookIsbn}"></td>

                <td th:text="${book.bookNumberOfPages}"></td>

                <td th:text="${book.bookYear}"></td>

                <td>

                    <a th:href="@{/books/edit/{id}(id=${book.id})}">Edit</a>

                    <a th:href="@{/books/delete/{id}(id=${book.id})}">Delete</a>

                </td>

            </tr>

        </tbody>

    </table>

    <a href="/books/add">Add New Book</a>

</body>

</html>

**add-book.html:**

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

    <title>Add Book</title>

</head>

<body>

    <h1>Add Book</h1>

    <form action="#" th:action="@{/books}" th:object="${book}" method="post">

        <label>Title: <input type="text" th:field="\*{bookTitle}" /></label><br/>

        <label>Publisher: <input type="text" th:field="\*{bookPublisher}" /></label><br/>

        <label>ISBN: <input type="text" th:field="\*{bookIsbn}" /></label><br/>

        <label>Pages: <input type="number" th:field="\*{bookNumberOfPages}" /></label><br/>

        <label>Year: <input type="number" th:field="\*{bookYear}" /></label><br/>

        <button type="submit">Add Book</button>

    </form>

    <a href="/books">Back to Book List</a>

</body>

</html>

**edit-book.html:**

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

    <title>Edit Book</title>

</head>

<body>

    <h1>Edit Book</h1>

    <form action="#" th:action="@{/books}" th:object="${book}" method="post">

        <input type="hidden" th:field="\*{id}" />

        <label>Title: <input type="text" th:field="\*{bookTitle}" /></label><br/>

        <label>Publisher: <input type="text" th:field="\*{bookPublisher}" /></label><br/>

        <label>ISBN: <input type="text" th:field="\*{bookIsbn}" /></label><br/>

        <label>Pages: <input type="number" th:field="\*{bookNumberOfPages}" /></label><br/>

        <label>Year: <input type="number" th:field="\*{bookYear}" /></label><br/>

        <button type="submit">Update Book</button>

    </form>

    <a href="/books">Back to Book List</a>

</body>

</html>