NAGARJUNA COLLEGE OF INFORMATION TECHNOLOGY

AFFILIATED TO TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

A Mini Project on:

E-Library (A Book Management System)

Advanced Database(CSC-461)

Submitted To:

Department of Computer Science

Nagarjuna College of Information Technology

Submitted By:

Isha Shrestha (15798/074)

Jebina Maharjan(15799/074)

Utsav Maarjan (15819/074)

BSc.CSIT Eighth Semester

Abstract:

As the number of books grows in one's book collection, it would be convenient to have a software utility to manage them. In this project, we propose a software system called eLibrary to manage personal book collections. It stores book information in a local database; book information can be easily added, deleted, updated in the book database. Most of the book's information, such as book title, author, publisher are stored in the database.

Table of Content

1. Introduction

1.1 Introduction

A book management system, termed as e-library is the software that enables the users to keep the record of the book with the details of the books like its name, author, ISBN number and so on. This system enables the users to add the books to the list, remove them to the list, update the details of the books and also view the list of the books. The actions performed for managing the books are directly connected to the database. This system plays the major role where one has a huge collection of the books and is having difficulty in handling the book by remembering all the details the book possesses.

1.2 Background :

Everybody has some books; some people may have a lot. As the number of books grows in one's collection, it would be convenient to have a software utility to manage them by generating reports on the books currently in collection, keeping track of book loans, quickly finding the books he or she needs from the bookshelves, etc. Also, many books now include their electronic version in CD-ROMs (normally in PDF), and some authors even offer free download of their books. Furthermore, researchers may have acquired a vast collection of electronic research documents over the years, thanks to the organisations offering full-text download (usually in PDF). People also need an efficient way to manage these electronic documents, since the file names of these documents are often meaningless and therefore it becomes extremely hard to find the exact one needed in a collection of hundreds or thousands of electronic documents.

The proposed book management system is a software application called eLibrary. The goal of this project is to build a full-featured software package to help people manage their books (either printed or electronic).

1.3 Objectives

- The main objective of this system is to provide a flexible platform for the users to keep the record of the book as well as manage them in accordance.
- To be able to add, remove and update the details of the list of books.

1.4 Scope

The proposed book management system, eLibrary, is a web-based system that helps people catalogue their books. It should be able to retrieve book information, such as book titles. The product should provide a friendly user interface; an average user should have no problem using it without being specially trained. This system is applicable in the scenario where one possesses numerous books and has to manage them to keep logs of them. This system, being a web based system, uses different web technologies that are very familiar to the users providing them with a user-friendly system. It can be used by the individual, a bookstore or as a module of the library system. It can be a greater area of scope because the world is turning into the digital world where everything is based on technology. The system is also applicable because there is always risk of tearing and losing the records if kept manually.

2. Theoretical background of mini project

The project is to enable efficient management of the book through the digital process. The digitalised database is used to keep the records of the books. The facility of a digitalised database is achieved through the use of MongoDB.

Since the system is web-based, the clients send the URL request to the host computer through the HTTP Protocol. The request is sent in the form of HTTP quest through the port. The request is received by the web server, locates the document on its local file system, and sends it back to the browser and the browser interprets the sent document from the web server into the correct format and displays it in the web browser.

The system is built in with the help of HTML, CSS, Bootstrap for frontend and for the backend Java language is used with the SpringBoot framework. The data is stored and maintained by MongoDB.

3. Features of E-library:

The book management system, e-library, is designed with the main motive of proper management and recording of the books for the individuals having enormous amount of books. This system possesses following features:

a. Adding a Book:
   The user can easily add the books in the list along with their details like author, type, publisher and many more.

b. Update the Details of Book:
   The user is able to update the details of the books whenever necessary.

c. Deleting book from the list:
   The user has the facility to delete the book from the list whenever he wants to delete it.

d. Listing the available Book:
   The user has the facility to list all the books he has that are stored in the database.

e. Book Database:
   It is the central repository where all the details of the books are stored. The retrieval of a list of books, updating them or deleting them are directly concerned with the database.

4. Technology Used:

For the completion of the entire project, different technologies are used in different aspects of the project.

a. HTML:

HTML(HyperText Markup Language) is used to create the framework of the User Interface. It is used in the front end part.

b. CSS:

CSS(Cascading Style Sheet) is used to enhance the beauty and styling of the framework that is created by the HTML.

c. Bootstrap:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

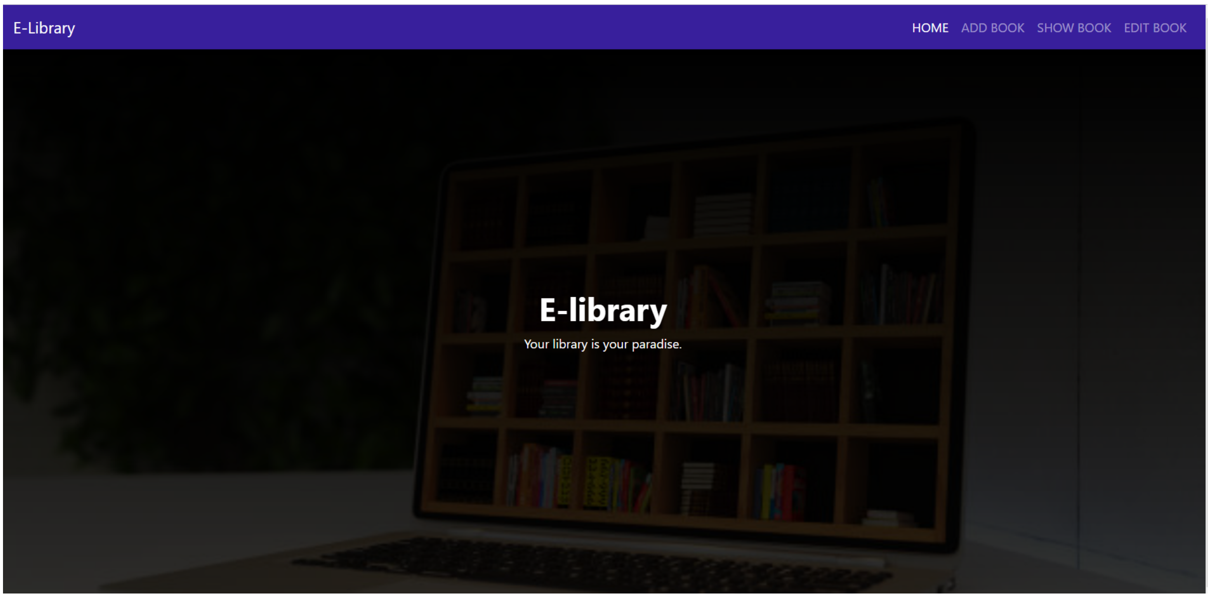
d. SpringBoot:

SpringBoot is the Java framework that follows the Java language. It is used in the backend of this system.

e. MongoDB:

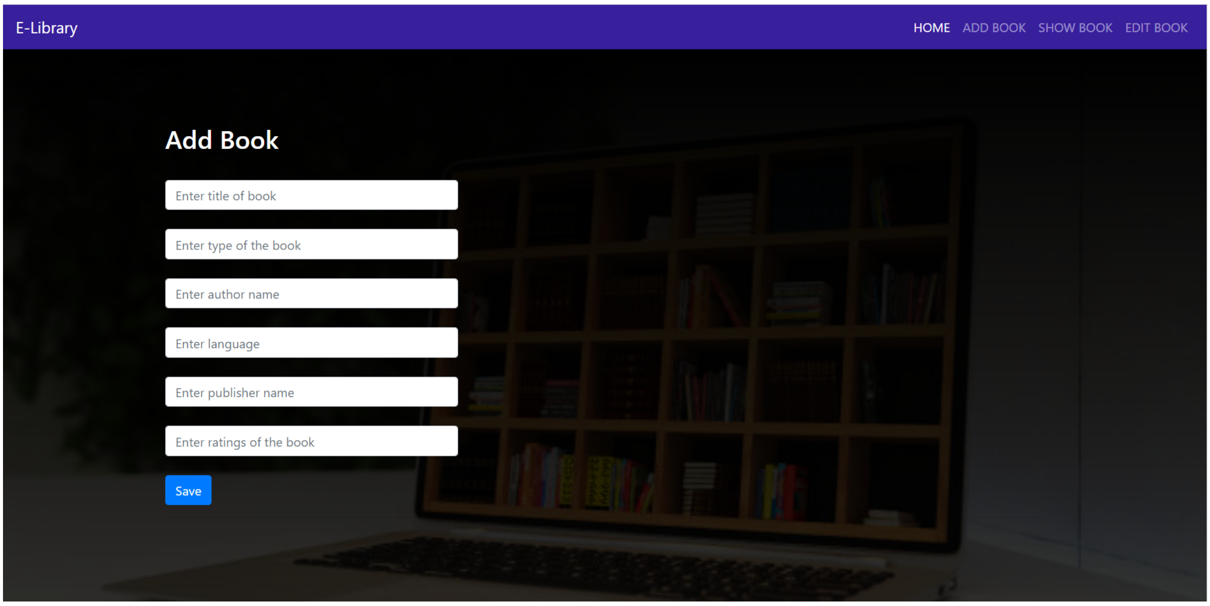
MongoDB is an open source NoSQL database management program. NoSQL is used as an alternative to traditional relational databases. MongoDB is a tool that is used to manage document-oriented information, store or retrieve information.

5. Output



E-library

Your library is your paradise.



Add Book

Enter title of book

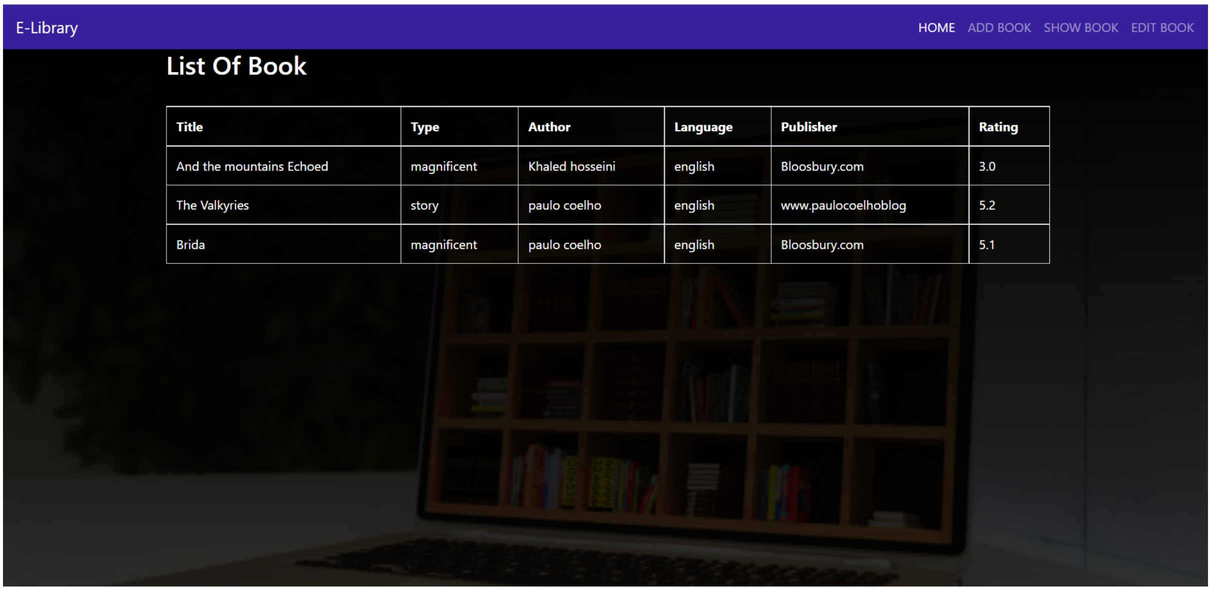
Enter type of the book

Enter author name

Enter language

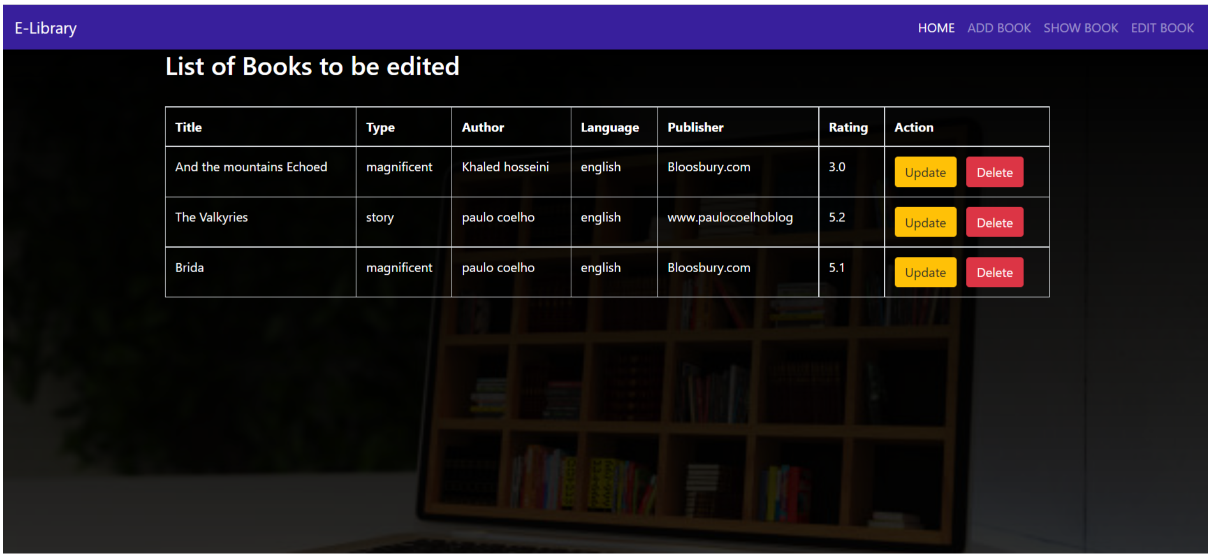
Enter publisher name

Enter ratings of the book



List Of Book

Title	Type	Author	Language	Publisher	Rating
And the mountains Echoed	magnificent	Khaled hosseini	english	Bloosbury.com	3.0
The Valkyries	story	paulo coelho	english	www.paulocoelhoblog	5.2
Brida	magnificent	paulo coelho	english	Bloosbury.com	5.1



List of Books to be edited

Title	Type	Author	Language	Publisher	Rating	Action
And the mountains Echoed	magnificent	Khaled hosseini	english	Bloosbury.com	3.0	Update Delete
The Valkyries	story	paulo coelho	english	www.paulocoelhoblog	5.2	Update Delete
Brida	magnificent	paulo coelho	english	Bloosbury.com	5.1	Update Delete

6. Conclusion

It is nearly impossible for the individual to memorise all the names and details of the book he has; especially when there is a massive volume of the books. Besides, the manual records in papers always have the risk of tearing and losing. So e-library or the book management system is important.

In sumup, the system is able to achieve its primary goal of providing a flexible platform for the users to keep the record of the book as well as manage them in accordance. The system is able to add, update and delete the list of books.

7.Annexes

Repository:

```
package com.bookmanagement.repository;

        import org.springframework.data.mongodb.repository.MongoRepository;
        import org.springframework.stereotype.Repository;

import com.bookmanagement.entity.Book;
@Repository
        public interface BookRepository extends MongoRepository<Book, Integer> {

}
```

Controller :

```
package com.bookmanagement.controller;


        import java.util.List;



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

        import org.springframework.web.bind.annotation.GetMapping;
        import org.springframework.web.bind.annotation.ModelAttribute;
        import org.springframework.web.bind.annotation.PostMapping;
        import org.springframework.web.bind.annotation.RequestParam;
        import org.springframework.web.servlet.ModelAndView;

import com.bookmanagement.entity.Book;
import com.bookmanagement.repository.BookRepository;
import com.bookmanagement.service.SequenceGeneratorService;
```

```
@Controller
public class BookController {
        @Autowired
        private SequenceGeneratorService services;
        @Autowired
        private BookRepository eRepo;

        @GetMapping({"/home","/login","/"})
        public ModelAndView home() {
                ModelAndView mv=new ModelAndView("home");
                return mv;
                }

        @GetMapping({"/showBook","/list"})
        public ModelAndView showBook() {
                ModelAndView mv=new ModelAndView("list-book");
                List<Book> book = eRepo.findAll();
                mv.addObject("book",book);
                return mv;
                }
        @GetMapping({"/actionBook"})
        public ModelAndView actionBook() {
                ModelAndView mv=new ModelAndView("Action");
                List<Book> book = eRepo.findAll();
                mv.addObject("book",book);
                return mv;
                }
        @GetMapping("/addBookForm")
         public ModelAndView addBookForm() {
                ModelAndView mv= new ModelAndView("add-book-form");
```

```
			Book newBook= new Book();
			mv.addObject("book",newBook);
			return mv;
		}

		@PostMapping("/saveBook")
		public String saveBook( @ModelAttribute Book book) {


book.setId(services.getSequenceNumber(Book.SEQUENCE_NAME));
			eRepo.save(book);
			return"redirect:/list";



		}
		@GetMapping("/showUpdateForm")
		public ModelAndView showUpdateForm(@RequestParam int bookId) {
			ModelAndView mv=new ModelAndView("update-book-form");
			Book book = eRepo.findById(bookId).get();
			mv.addObject("book",book);
			return mv;

		}

		@PostMapping("/saveBook1")
		public String saveBook1(@ModelAttribute Book book) {


			eRepo.save(book);
			return"redirect:/list";



		}
		@GetMapping("/deleteBook")
```

```
        public String deleteBook(@RequestParam int bookId) {
                eRepo.deleteById(bookId);
                return "redirect:/list";
        }


}
```

Dependency :

```
<dependencies>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-mongodb</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
```

```
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <excludes>
                    <exclude>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
                    </exclude>
                </excludes>
            </configuration>
        </plugin>
```

```
                </plugins>
        </build>
```

Html:

```
<!DOCTYPE html>
<html xmlns="http://www.thymeleaf.org"th:replace="base::layout(~{::section})">
<head>
<meta charset="ISO-8859-1">
<title>Update Book</title>
<link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgF
AW/dAiS6JXm"
    crossorigin="anonymous">
</head>
<body>
<section class="banner d-flex text-white">
<div class="container">
<h2>Update Book</h2>
<hr/>
<form th:action="@{/saveBook1}" th:object="${book}" th:method="POST">
<input type="text" th:field="*{title}" placeholder="Enter title of book" class="form-control
col-4 mb-4"/>
<input type="text" th:field="*{type}" placeholder="Enter type of the book"
class="form-control col-4 mb-4"/>
<input type="text" th:field="*{author}" placeholder="Enter author name"
class="form-control col-4 mb-4"/>
<input type="text" th:field="*{language}" placeholder="Enter language"
class="form-control col-4 mb-4"/>
<input type="text" th:field="*{publisher}" placeholder="Enter publisher name"
class="form-control col-4 mb-4"/>
```

```
<input type="text" th:field="*{rating}" placeholder="Enter ratings of the book " class="form-control col-4 mb-4"/>
<button class="btn btn-primary" type="submit">Save</button>
<input type="hidden" th:field="*{id}"/>

</form>
<hr/>


</div>
</section>


</body>
</html>
```