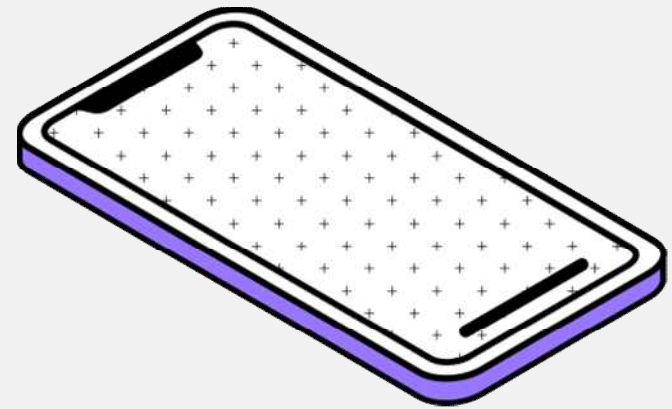




Backend Projects

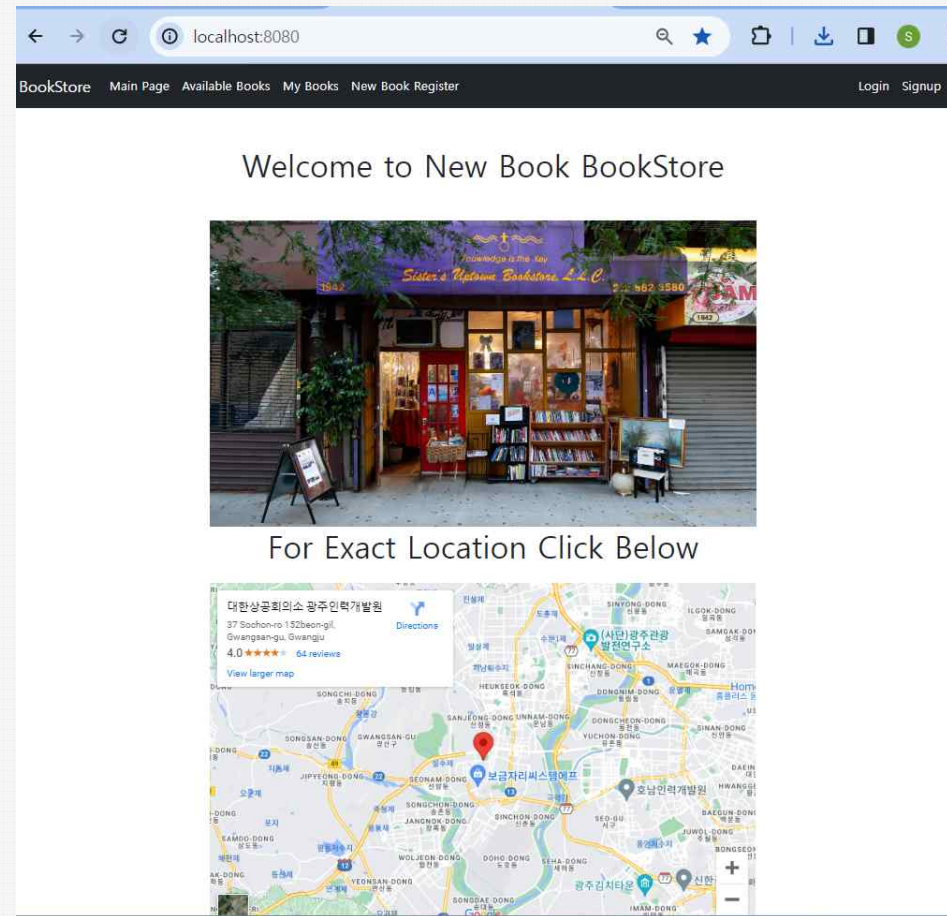
Sara Kim



Main Page Book-Store Project

Welcome Message +

Current address of the store



1- SignUp

```
@GetMapping("/signup")
public String signup() {
    return "signup";
}

@PostMapping("/signup")
public String signupPost(@ModelAttribute Userb user) {
    String userPwd = user.getPwd();
    String encodedPwd = passwordEncoder.encode(userPwd);
    user.setPwd(encodedPwd);
    userRepository.save(user);
    System.out.println(encodedPwd);
    return "redirect:/signin";
}

@GetMapping("/logout")
public String logout(HttpSession session) {
    session.invalidate();
    return "redirect:/";
}
```

The screenshot displays a web application interface for a bookstore. At the top, a dark navigation bar contains the following links: "BookStore", "Main Page", "Available Books", "My Books", "New Book Register", and "Login". The main content area is titled "Signup for BookStore". Below the title, there are four input fields, each with a label above it: "Email address", "Password", "Name", and "Role". Each input field is a simple white box with a thin border. Below the "Role" input field is a blue button with the text "Signup" in white. The entire form is centered on a light gray background.

2- Login

[BookStore](#) [Main Page](#) [Available Books](#) [My Books](#) [New Book Register](#) [Login](#) [Signup](#)

Login to BookStore

Email address

Password

Login

```
@PostMapping("/react/signin")
@ResponseBody
public Map<String, Object> reactSigninPost(
    @ModelAttribute Userb user,
    HttpSession session,
    @RequestParam(required = false) String referer
) {
    Userb existingUser = userRepository.findByEmail(user.getEmail());
    String encodedPwd = existingUser.getPwd();
    String userPwd = user.getPwd();
    boolean isMatch = passwordEncoder.matches(userPwd, encodedPwd);

    Map<String, Object> map = new HashMap<>();

    if (isMatch) {
        session.setAttribute(name:"user", existingUser);
        map.put(key:"msg", value:"Success");
    } else {
        map.put(key:"msg", value:"Fail");
    }
    return map;
}
```

3- New Book Register

Only Admin can Register the book

New Book Register

Name

test

Author

sara

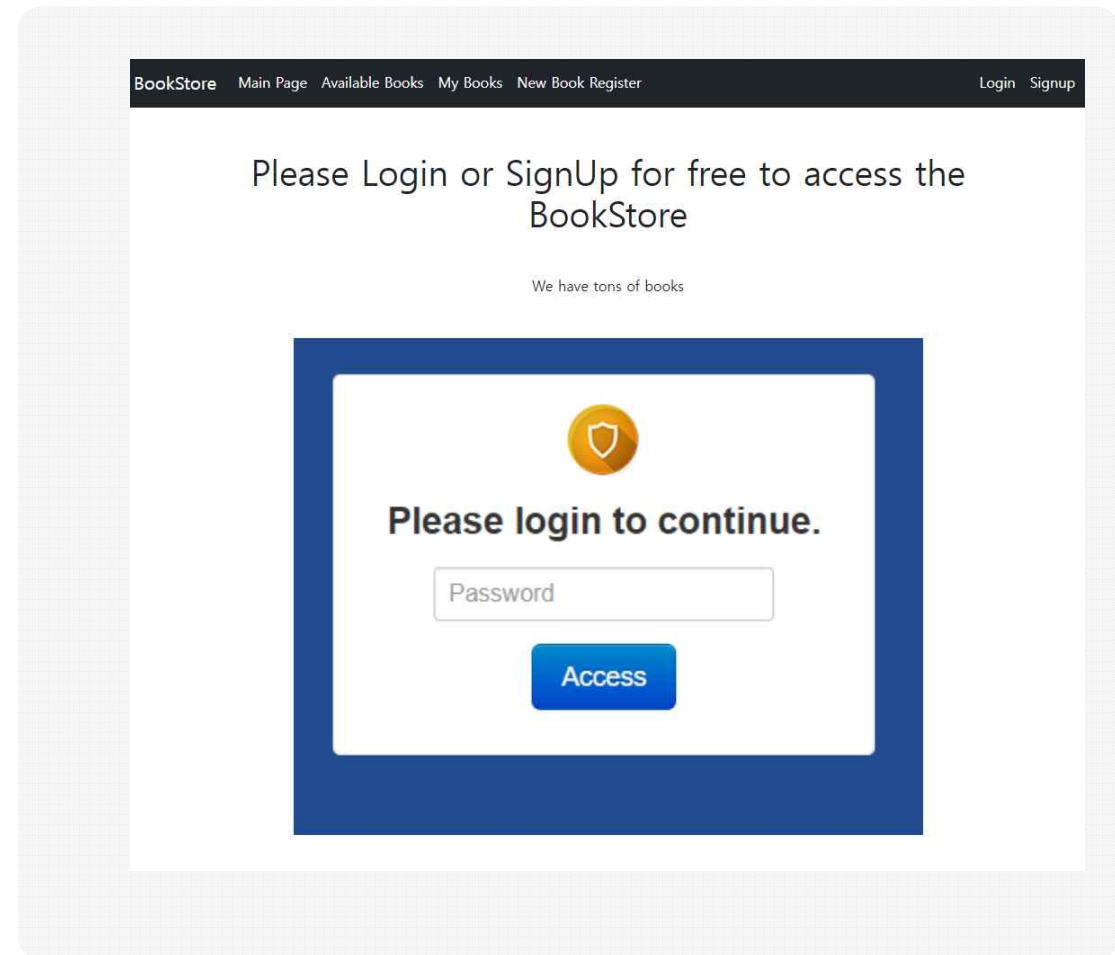
Price

1000

Submit

```
@GetMapping("/book_register")
public String bookRegister() {
    return "bookRegister";
}
```

```
@PostMapping("/save")
public String addBook(@ModelAttribute Book bookb) {
    bookRepository.save(bookb);
    return "redirect:/available_books";
}
```



4- Available Books

Everyone Can see Book List (But Just Admin can Delete and Modify books)

BookStore

Main Page

Available Books























My Books

New Book Register

김사라

welcome

Logout

Id	Name	Author	Price	Action			
3	"To Kill a Mockingbird"	Harper Lee	1132 \$	Add To Mybook	View Details		
5	"The Great Gatsby"	F. Scott Fitzgerald	9.99 \$	Add To Mybook	View Details		
6	"One Hundred Years of Solitude"	Gabriel Garcia Marquez	12.99 \$	Add To Mybook	View Details		
7	"Moby Dick"	Herman Melville	11.50 \$	Add To Mybook	View Details		
8	"War and Peace"	Leo Tolstoy	15.95 \$	Add To Mybook	View Details		
9	"Pride and Prejudice"	Jane Austen	7.95 \$	Add To Mybook	View Details		
10	"Crime and Punishment"	Fyodor Dostoevsky	14.50 \$	Add To Mybook	View Details		
11	"Ulysses"	James Joyce	16.00 \$	Add To Mybook	View Details		
12	"The Catcher in the Rye"	J.D Salinger	10.00 \$	Add To Mybook	View Details		
34	how to think	sara kim	56 \$	Add To Mybook	View Details		
67	test	sara	1000 \$	Add To Mybook	View Details		

```
@GetMapping("/available_books")
public String getAllBooks(Model model) {
    List<Book> book1 = bookRepository.findAll();
    model.addAttribute(attributeName:"book", book1);
    return "bookList";
}
```

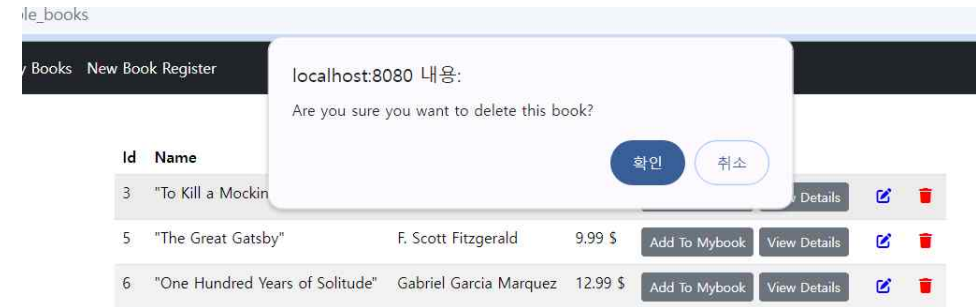

5- Book Delete/Edit

Just Admin can Delete and Modify Books

```
@GetMapping("/deleteMyList/{id}")
public String deleteBook(@PathVariable int id) {
    myBookRepository.deleteById(id);
    return "redirect:/my_books";
}

@GetMapping("/editBook/{id}")
public String editBook(@PathVariable int id, Model model) {
    Optional<Book> optionalBook = bookRepository.findById(id);

    if (optionalBook.isPresent()) {
        Book b = optionalBook.get();
        model.addAttribute(attributeName:"book", b);
        return "bookEdit";
    } else {
        return "redirect:/available_books";
    }
}
```



Edit Book Here

Name

Author

Price



6- My Books

Login -> Add To Mybook -> My Books

```
@GetMapping("/my_books")
public String getMyBooks(Model model) {
    List<MyBookList> list = myBookRepository.findAll();
    model.addAttribute(attributeName:"book", list);
    return "myBooks";
}
```


```
@GetMapping("/mylist/{id}")
public String addBookToMyList(@PathVariable("id") int id) {
    Optional<Book> optionalBook = bookRepository.findById(id);
    if (optionalBook.isPresent()) {
        Book b = optionalBook.get();
        MyBookList mb = new MyBookList();
        mb.setId(b.getId());
        mb.setName(b.getName());
        mb.setAuthor(b.getAuthor());
        mb.setPrice(b.getPrice());
        myBookRepository.save(mb);
    }

    return "redirect:/my_books";
}
```

Id	Name	Author	Price	Action
2	test1	sara	1000 \$	<button>Add To Mybook</button> <button>View Details</button>  

BookStore Main Page Available Books My Books New Book Register

김사라 welcome Logout

Id	Name	Author	Price	Action
2	test1	sara	1000	

7- Book Details

```
@GetMapping("/bookDetail/{id}")
public String bookDetail(@PathVariable("id") int id, Model model) {
    Optional<Book> optionalBook = bookRepository.findById(id);

    if (optionalBook.isPresent()) {
        Book book = optionalBook.get();
        model.addAttribute(attributeName:"book", book);
        List<Comment> comments = commentRepository.findByBook(book);
        model.addAttribute(attributeName:"comments", comments);

        Userb user = (Userb) session.getAttribute(name:"user");
        if (user != null) {
            model.addAttribute(attributeName:"user", user);
        }

        return "bookDetail";
    } else {
        return "error";
    }
}
```

BookStore Main Page Available Books My Books New Book Register 김사라 welcome Logout

Book Details

ID: 2

Name: test1

Author: sara

Price: \$1000

Title:

Content:

Upload Files: 선택된 파일 없음

Submit

Writer	Title	Content
김사라	Tesst	Test1

8- Book Details/Comment

```
@PostMapping("/comment")
public String comment2(
    @RequestParam("title") String title,
    @RequestParam("content") String content,
    @RequestParam("files") MultipartFile[] files,
    @RequestParam int bookId) {

    Userb currentUser = (Userb) session.getAttribute(name:"user");

    if (currentUser == null) {
        return "redirect:/signin";
    }

    Comment comment = new Comment();
    comment.setTitle(title);
    comment.setContent(content);
    comment.setWriter(currentUser.getName());
    comment.setUser(currentUser);
    Book book = new Book();
    book.setId(bookId);
    comment.setBook(book);

    Comment savedComment = commentRepository.save(comment); // save comment
```

```
for(MultipartFile file : files) {
    String fileName = file.getOriginalFilename();
    try {

        file.transferTo(new File("c:/files/" + fileName));

        FileAttach f = new FileAttach();
        f.setOriginalName(fileName);
        f.setSaveName(fileName);
        f.setComment(savedComment);
        fileAttachRepository.save(f);

    } catch (IllegalStateException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

List<FileAttach> fileAttaches = new ArrayList<>();
// }

return "redirect:/bookDetail/" + bookId;
}
```

Thanks
you

