

## **Assignment 6**

### Implementation Guidelines

- Server-side functionality must be implemented using NodeJS and Express
- Views must be created using server-side rendering and Express-Handlebars. You may use should use layouts and partials as appropriate.
- Database collections must be implemented using MongoDB and the Mongoose driver.
- CSS should be written using CSS selectors in an external stylesheet and placed in a static resources folder

### Submission Checklist

For your submission to be graded, you must provide a **zip** file of your project, and a **screen recording** demonstrating the functionality you implemented.

#### **1. Creating Your Project**

- When creating your project, name the project: **LibraryFirstName**, where you replace FirstName with your first name. Example: **LibraryDavid**
- Ensure your project is configured as a NodeJS project. You can do this by running “**npm init -y**” in your project folder
- After completing your code, create a zip file containing all project code. Ensure your zip file is named **LibraryFirstName.zip**, where FirstName is your first name. **No 7zip or rar files are accepted.**

#### **3. Creating Your Screen Recording**

- In the screen recording, demonstrate the app functionalities you implemented
- During the screen recording, you must verbally narrate/explain what you are doing while you are doing it (do not assume the instructor will understand what you are doing simply by watching you click things on the screen)
  - You are not required to explain your code
  - Max 5 min
- After creating your screen recording, then upload the screen recording to **your college Microsoft OneDrive account** and ensure that the link is set to: “Anyone with the link can view”. Paste a link to the recording in the **submission comments**.

#### **4. In the assignment dropbox:**

- Submit your zip file to the dropbox
- Paste a link to your screen recording to the dropbox.

### Academic Integrity

- This is an individual assessment.
- Permitted activities: Usage of Internet to search for syntax only; usage of course materials
- Not permitted:
  - Communication with others (both inside and outside the class)
  - Discussion of solution or approaches with others; sharing/using a “reference” from someone
  - Searching the internet for full or partial solutions
  - Sharing of resources, including links, computers, accounts

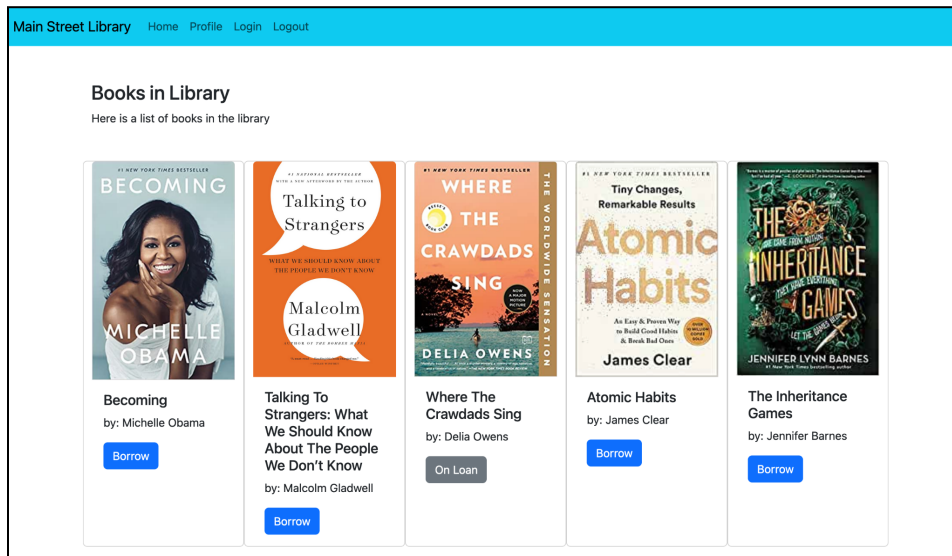
## Problem Description

In this assessment, you will build a web application for a library

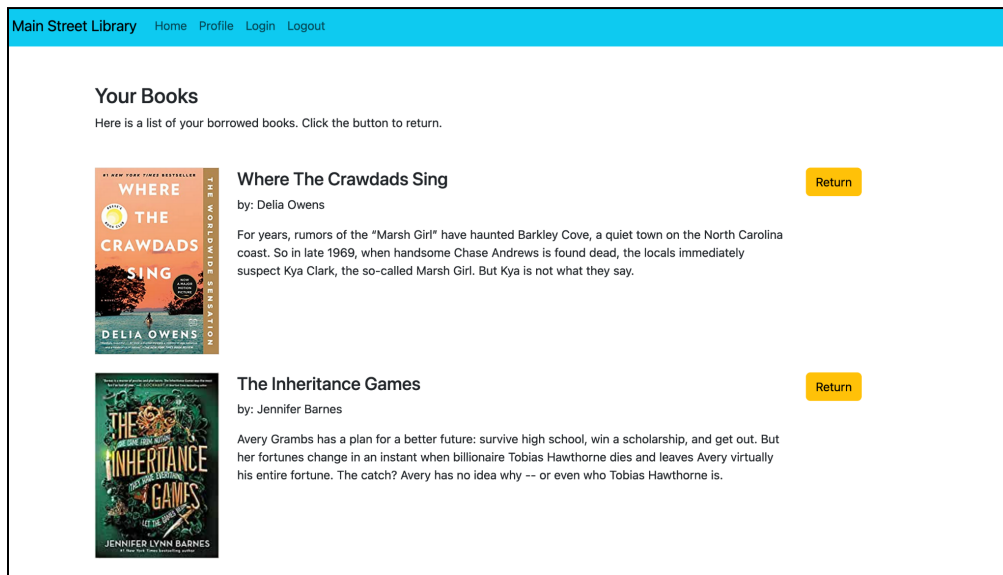
The user interface of your application must be constructed using **Bootstrap 5.3** and **Handlebars**:

- Home Page template
- Profile Page template
- Login Page template
- Error Page template

### Example of Home Page

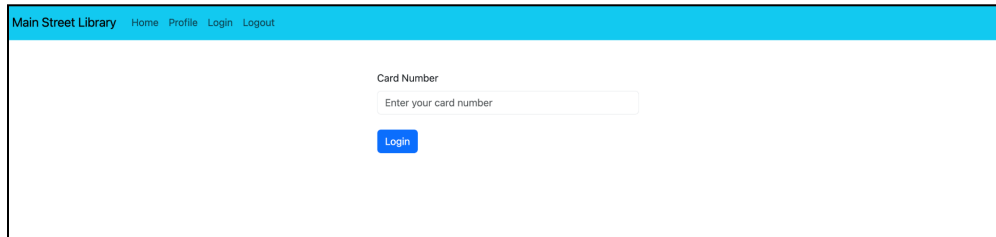


### Example of Profile Page

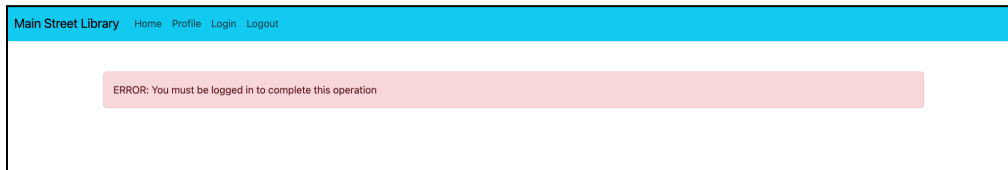


The content in this document is subject to copyright and other intellectual property rights. Reposting, distributing, or modification is not permitted.

### Example of Login Page



### Example of error page



### Endpoint Access Control

Page	Access
Home Page	Anyone can view the page, but only logged in users can borrow a book
Profile Page	If user is logged in, show the profile page. If user not logged in, display error message
Logout Page	Any user can click on this
Login Page	Any user can view this page

### MongoDb Database

Prepopulate your database with the following collections. You must have 5 books and 2 users.

#### 1/ Books collection:

- Author (string)
- Title (string)
- borrowedBy (string) → the library card number of the user who borrowed the book. By default, this should be set to an empty string.
- img (string) → url or filename of a photo of the book
- desc (string) → short description of the book

You may select any books for your database collection.

#### 2/ Users collection:

The content in this document is subject to copyright and other intellectual property rights. Reposting, distributing, or modification is not permitted.

- Library card number (string)

Use the following library card numbers:

- 0000
- 1234

## Headers and Footers

All pages must have a *header* that contains a navigation menu. The navigation menu must be implemented using the Bootstrap **NavBar** component. Show these menu items:

- Name of library (this is the “brand item”)
- Home → when clicked, redirects to the / endpoint
- Profile → when clicked, displays the user’s profile page
- Login → when clicked, displays the login page
- Logout → when clicked, logs the user out

## Login Page

Provide a form (label, input box, button) that allows the user to enter a library card number. If the card is valid, then redirect the user to the home page (Use `res.redirect()`)

If the card number is invalid, then display the **Error Page** template with an appropriate error message.

You may assume the user will always enter a library card (ie: the library card box will never be empty).

The form and its components (label, input box, button) must be styled using Bootstrap:

<https://getbootstrap.com/docs/5.3/forms/form-control/>

## Home Page

Displays the list of books in the library.

Each book should be displayed as a Bootstrap **Card** component.

<https://getbootstrap.com/docs/5.3/components/card/#example>

You may need to adjust the sample code to meet your needs.

For each book, display:

- The image of the book
- Book title
- Book author
- Button
  - If the book can be borrowed, display a BORROW button
  - If the book cannot be borrowed, display an ON LOAN button.

### On Loan

If user clicks on the **On Loan** button, display a Bootstrap Popover component with an appropriate error message. (<https://getbootstrap.com/docs/5.3/components/popovers>)

HINT: You must include this Javascript in your layout file for the popover to work properly:

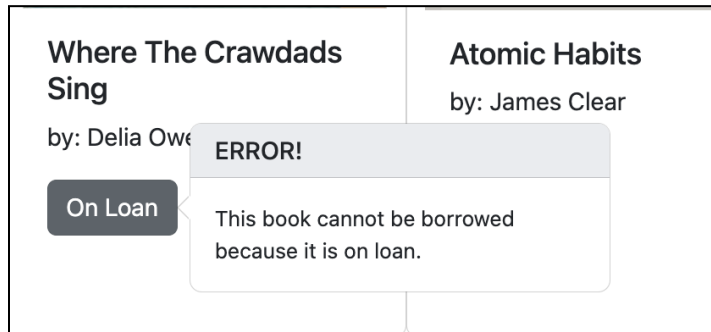
<https://getbootstrap.com/docs/5.3/components/popovers/#enable-popovers>

*Example javascript:*

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z"
crossorigin="anonymous">
</head>
<body>
  {{{ body }}}

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYN57NTKfAdvQSZ"
  crossorigin="anonymous"></script>
  <script type="text/javascript">
    const popoverTriggerList = document.querySelectorAll('[data-bs-toggle="popover"]')
    const popoverList = [...popoverTriggerList].map(popoverTriggerEl => new
bootstrap.Popover(popoverTriggerEl))
  </script>
</body>
</html>
```

*Example:*



### Borrow Button

When pressed, check if the user is logged in.

If logged in, update the selected Book with the card number of the currently logged in user.

If NOT logged in, display the Error Page template with an appropriate message.

### Profile Page

This page displays a list of books borrowed by the user. The layout of the user interface elements must be constructed using **Bootstrap Grids**.

This page can only be displayed to logged in users.

If user is not logged in, and attempts to access this page, then display the Error Page template with an appropriate message.

If user IS logged in:

- Retrieve the list of books that are borrowed by this user
- Display the list of books.
- For each book, show:
  - Book image
  - Title
  - Author
  - Description
  - A RETURN button

### Return Button

- When return button is pressed, update the book's borrowedBy field to its default value (empty string)
- Display a success message.

The content in this document is subject to copyright and other intellectual property rights. Reposting, distributing, or modification is not permitted.

## Error Page

This page template consists of a Bootstrap Alert component.

<https://getbootstrap.com/docs/5.3/components/alerts>

The message displayed by the alert must be programmatically set by the server.

**END OF ASSESSMENT**