Assignment: **02**

Students Name: **Ghulam Mujtaba, Karim Khellaf, Hiten Sharma**

Lab Professor Name: **Mr. Alemeseged Legesse**

Lab Section Number: **CST8285 - 313**

Due Date: **Aug 10, 2025 11:59 PM**

Project Overview:

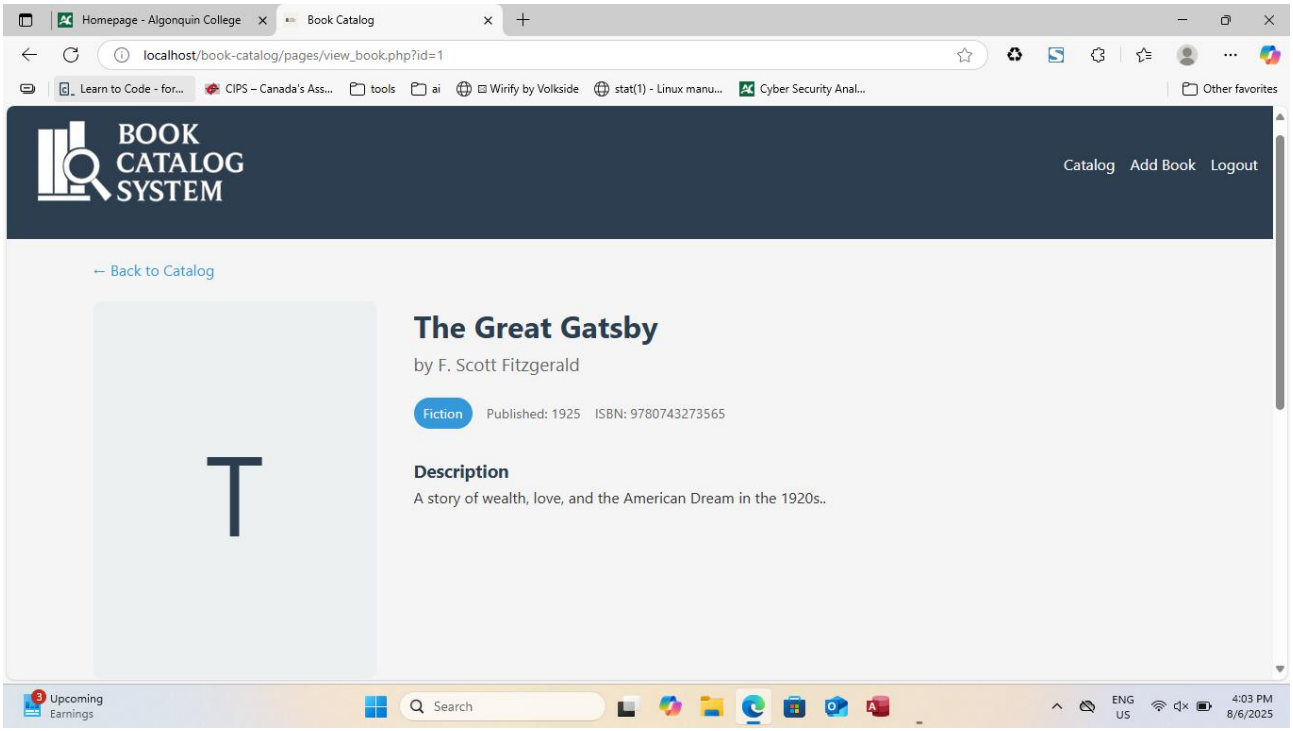
The Book Catalog System is a dynamic web application that enables users to register, log in, and manage a collection of books. Users can add, view, edit, and delete books, as well as search and filter through the catalog. The application uses HTML and CSS for structure and styling, JavaScript for client-side validation and interactivity, PHP for server-side processing, and MySQL for backend database management.

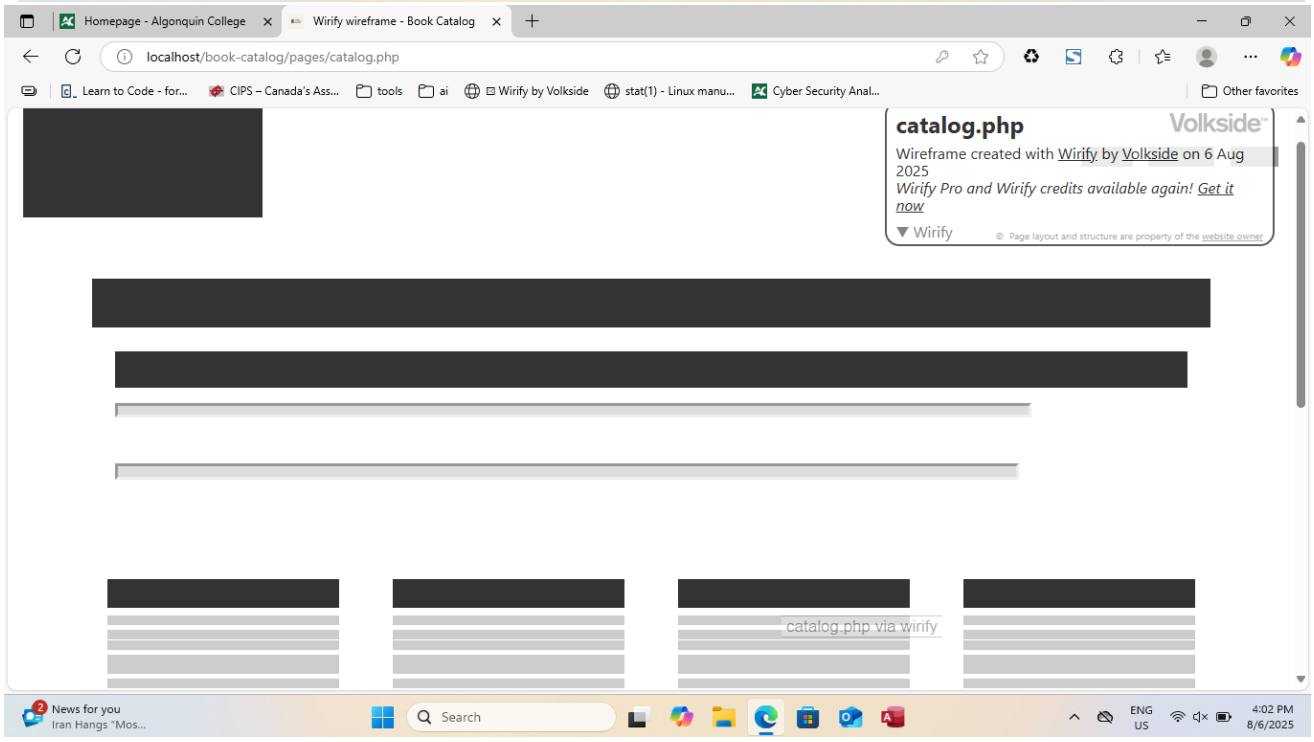
Web Map:

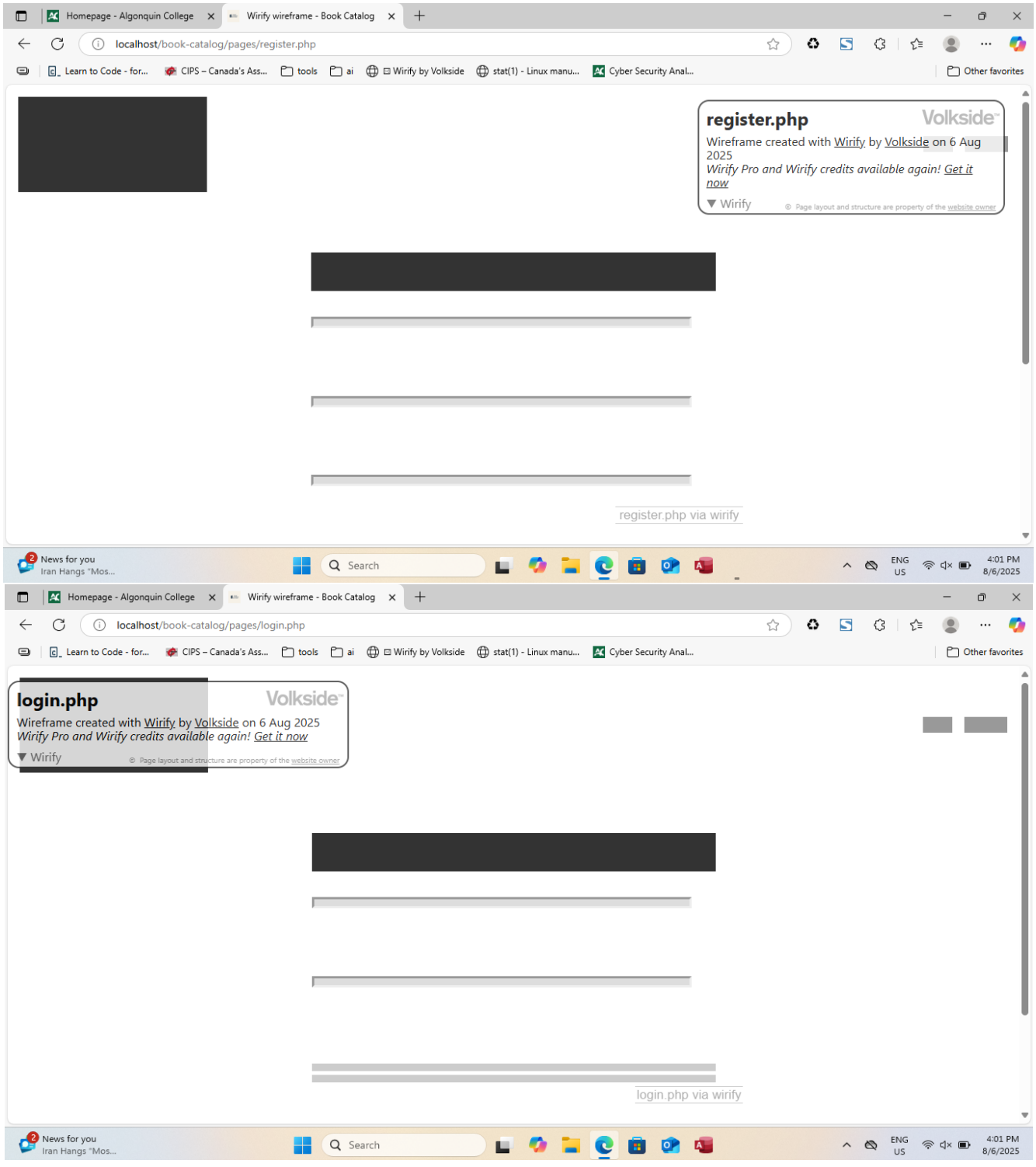
Website navigation structure:

Home → Catalog → View Book
↘ Add Book → Catalog
Register → Login → Catalog

Wireframe Screenshots:






Homepage - Algonquin College
Wirify wireframe - Book Catalog
localhost/book-catalog/pages/register.php
register.php
Volkside
Wireframe created with Wirify by Volkside on 6 Aug 2025
Wirify Pro and Wirify credits available again! Get it now
Wirify
Page layout and structure are property of the website owner
register.php via wirify
localhost/book-catalog/pages/login.php
login.php
Wireframe created with Wirify by Volkside on 6 Aug 2025
Wirify Pro and Wirify credits available again! Get it now
Wirify
Page layout and structure are property of the website owner
login.php via wirify
Search
4:01 PM
8/6/2025

Database Description:

The database contains the following tables:

- **users** – Stores user account details (id, username, email, password hash).
- **books** – Stores book information (id, title, author, genre, year, description).

Primary keys ensure unique records, and foreign keys define relationships between tables.

Functionality Guide:

1. Registration – Users complete the registration form, validated by JavaScript, and the data is stored in the database.
2. Login – Users log in with credentials verified against the database.
3. Adding a Book – Logged-in users can add a new book using a form.
4. Viewing the Catalog – Displays all books with search and filter options.
5. Editing/Deleting Books – Users can update or remove existing entries.
6. Searching/Filtering – Search by title, author, or genre, and apply filters.

Diagram:


Done By : Ghulam Mujtaba , Karim Khellaf, Hiten Sharma
User
Browser
HTML
CSS
PHP
SQL
User opens webpage (launches the form)" — The user starts the interaction by opening the site.
Browser requests HTML, content is loaded" — The browser fetches the HTML structure of the page.
Browser detects <link>, requests CSS file" — The browser sees the CSS reference and retrieves the styling.
"User submits the form" — The user enters information and sends it.
"Browser sends form data to PHP server" — The browser passes form inputs to the server-side script.
PHP sends SQL query (INSERT, SELECT, UPDATE, DELETE)" — The PHP script communicates with the database.
"Database returns results" — The database sends back the requested data or confirmation.
PHP processes form data and returns updated HTML" — The server sends the updated page content.
Browser displays updated content" — Final output is shown to the user.

1. **User → Browser:** *"User opens webpage (launches the form)"* – The user begins the process by opening the web application in their browser. This action triggers the browser to start retrieving the resources needed to display the page.

2. **Browser → HTML:** *"Browser requests HTML, content is loaded"* – The browser sends a request to the server for the HTML file. Once received, the HTML provides the basic structure and content for the webpage.

3. **HTML → CSS:** *"Browser detects <link>, requests CSS file"* – While reading the HTML, the browser finds a <link> tag pointing to a CSS file. The browser then requests this CSS file to apply styling, colors, and layout to the page.

4. **User → Browser:** *"User submits the form"* – After interacting with the page (for example, registering a new account or adding a book), the user clicks the submit button to send their input.

5. **Browser → PHP:** *"Browser sends form data to PHP server"* – The browser packages the form data and sends it to the server-side PHP script via an HTTP POST or GET request.

6. **PHP → SQL:** *"PHP sends SQL query (INSERT, SELECT, UPDATE, DELETE)"* – The PHP script processes the form input and sends the appropriate SQL query to the MySQL database. This could involve inserting new data, retrieving records, updating existing information, or deleting entries.

7. **SQL → PHP:** *"Database returns results"* – The database executes the query and returns the results or a success/failure status to the PHP script.

8. **PHP → Browser:** *"PHP processes form data and returns updated HTML"* – Based on the database results, the PHP script generates updated HTML or JSON and sends it back to the browser.

9. **Browser → User:** *"Browser displays updated content"* – The browser receives the updated content and displays it to the user, showing either the requested data (e.g., a list of books) or feedback messages (e.g., confirmation of successful registration).

Task Breakdown:

Task	Person Responsible
HTML structure	Ghulam Mujtaba
CSS styling	Hiten Sharma
PHP backend	Karim Khellaf
Database design	Karim Khellaf