

CST8285: Web Programming

Assignment 2: Dynamic Web Application

HTML, CSS, JavaScript & PHP

Objectives

- Allows you to use your creativity, to develop a dynamic web application. You will choose what you want to build, with the project requirements constraining the features that must be used.
- Have hands-on experience with several technologies and can make these components interact with each other to provide services.
- Work as a team, collaborate with your project partner(s) professionally and contribute fairly.

Read the entire instruction before starting. If at any time you are unsure or are having problems, consult your lab instructor.

Required Equipment: XAMPP, HTML editor, Collaboration Tool (GitHub recommended)

Part I: Choosing Your Team (Due: November, 4)



This is a team project; Teams will consist of maximum **three students** When you have decided on your team configuration register your team on the Brightspace.

Part II: Choosing Your Project



For this project you will create a dynamic web application which meets the following minimum criteria: ***Please consider the following as minimum requirements only — you are encouraged to get creative and do more than this, but this is what will be required to receive full marks.***

1. **Backend Database Functionality:** Implement backend database functionality for seamless content management on the website. Users should be able to add, display, update and remove content through the website interface.
2. **Search and Filtering:** Enable users to search and filter content effectively.
3. **HTML Structure:** Use semantically correct HTML to ensure proper document structure.
4. **User Registration Form:** Develop a user registration form allowing individuals to sign up as new users. Implement functionality to store user information in a database.
5. **Form Validation:** Implement client-side form validation using JavaScript and DOM manipulation. No standard HTML form validation is allowed; all validation will be done with JavaScript. Provide user-appropriate error messages for a smooth user experience.
6. **Dynamic Behavior:** Incorporate dynamic behavior with client-side input validation using JavaScript and DOM manipulation. Modify pages according to user inputs or the application's current states.
7. **Responsive Design:** Create a responsive and user-friendly interface using HTML and CSS. Utilize one external CSS file for styling.
8. **Adhere to good coding practices.**
 - Consistent Indentation:
 - Meaningful Variable and Function Names:
 - Include comments for clarity.

Examples of different Project Ideas

1. Online Recipe Manager:

- **Backend Database Functionality:** Allow users to add, display, and remove recipes through a web interface. Store recipe information in a backend database.

- **Search and Filtering:** Enable users to search for recipes based on ingredients, cuisine, or dietary preferences.
- **HTML Structure:** Use semantically correct HTML to structure recipe pages and ingredients.
- **User Registration Form:** Create a user registration form for individuals to sign up and save their favorite recipes.
- **Form Validation:** Implement client-side form validation for the registration form using JavaScript, providing clear error messages.
- **Dynamic Behavior:** Modify the recipe pages dynamically based on user interactions, such as favoriting or rating recipes.
- **Responsive Design:** Ensure a responsive and user-friendly design for easy access on various devices.

2. Book Cataloging System:

- **Backend Database Functionality:** Implement a backend to manage a catalog of books. Users can add, display, and remove books through the web interface.
- **Search and Filtering:** Allow users to search for books by title, author, genre, or other criteria.
- **HTML Structure:** Use semantically correct HTML for book listings and details.
- **User Registration Form:** Develop a user registration form for book enthusiasts to create accounts and manage their book collections.
- **Form Validation:** Apply client-side form validation for the registration form using JavaScript, providing user-friendly error messages.
- **Dynamic Behavior:** Implement dynamic features like book ratings, reviews, and personalized recommendations.
- **Responsive Design:** Design a responsive layout for optimal viewing on different devices.

3. Blog Platform:

- **Backend Database Functionality:** Implement a backend for a blogging platform. Users can add, display, and remove blog posts through the website.
- **Search and Filtering:** Enable users to search for blog posts by keywords, categories, or authors.
- **HTML Structure:** Use semantically correct HTML for blog post layouts.
- **User Registration Form:** Develop a user registration form for individuals to create accounts and write their own blog posts.
- **Form Validation:** Implement client-side form validation for the registration form using JavaScript, providing clear error messages.
- **Dynamic Behavior:** Allow users to comment on blog posts and dynamically update the page without refreshing.
- **Responsive Design:** Design a responsive and visually appealing layout for an optimal reading experience.

4. Event Planning System:

- **Backend Database Functionality:** Implement a backend for managing events. Users can add, display, and remove events through the website.
- **Search and Filtering:** Enable users to search for events based on date, location, or type.
- **HTML Structure:** Use semantically correct HTML for event details and listings.
- **User Registration Form:** Develop a user registration form for event organizers and attendees.
- **Form Validation:** Apply client-side form validation for the registration form using JavaScript, providing user-friendly error messages.
- **Dynamic Behavior:** Allow users to RSVP to events, and dynamically update event details based on user interactions.

- **Responsive Design:** Ensure a responsive design for users accessing event information on various devices.

5. Task Management Application:

- **Backend Database Functionality:** Implement a backend for a task management system. Users can add, display, and remove tasks through the website.
- **Search and Filtering:** Enable users to search for tasks and filter them based on priority, due date, or status.
- **HTML Structure:** Use semantically correct HTML for task details and listings.
- **User Registration Form:** Develop a user registration form for individuals to create accounts and manage their tasks.
- **Form Validation:** Implement client-side form validation for the registration form using JavaScript, providing clear error messages.
- **Dynamic Behavior:** Implement dynamic features such as task prioritization, completion status, and real-time updates.
- **Responsive Design:** Design a responsive layout for easy task management on different devices.

Remember, these are just examples to guide you. Feel free to choose a project idea that you're passionate about. If you have an idea that meets the requirements, go for it! This project is an opportunity for you to express your creativity and apply your skills in a way that excites you.

Part III: Project Submission



When the project is completed and tested you will package up the deliverables as follows:

- All website collateral (code, images, SQL, etc.) arranged in folders so that they can be installed in a subdirectory on the lab instructor's XAMPP.
- All supporting documentation that contains at a minimum a web map, a wireframe, a functionality guide, a description of the database, and any special coding considerations.
- All code written for the project in any language. Identify **who wrote what portions of the code**.
- Database definition code (DDL for the tables).
- Database scripts.
- A breakdown of the tasks assigned to each team member.
- Designate your main page as "index.html".
- Place all your HTML files in a dedicated folder. Name this folder something like "pages"
- Organize all your JavaScript files into a separate folder. You might name "scripts".
- Create a distinct folder for PHP files. Name this folder "server"
- For any database-related files create a "database" folder.
- Your slides for presentation

Each team submits only one copy. Upload the zip file to group assignment on Brightspace.

Part IV: Final Project Demo (Full Team)

All partners must be present at the demo.

Presentations

Prepare a 3-6 slides that represent your project.

Each participant should be able to describe and demo the features of the website that they have worked on. The team leader will schedule this demo. **If any team members do not attend the demo, they will be graded as zero in Assignment 2.**

You will have to sign-up for a demonstration slot in order to do this. A sign-up page will be available for you to use. An announcement with details will be sent out when a demo sign-up sheet becomes available. Please only **sign up for one slot and do not remove the slot of another group**. Sign up early so that you get the slot you want.

- The work will be graded zero if you do not demo it on time, even if uploaded.
- A demo is limited to 15 minutes (10 minutes demo + 5 minutes Q&A).

Grading

Criteria	Points	Description
Backend Database Functionality (20 point)		
Database Implementation	10	Successfully implements backend database functionality. Allows users to add, display, update and remove content through the website interface. Uses appropriate database management
Content Management	5	Users can seamlessly manage content through the website. Content is properly stored and retrieved from the backend.
Error Handling	5	Implements proper error handling for database operations. Provides clear error messages to users.
Search and Filtering (10 points)		
Search Functionality	5	Enables users to search for content effectively. Implements a robust search algorithm.

Filtering Options	5	Provides effective content filtering options. Allows users to filter content based on relevant criteria.
HTML Structure (5 points)		
Semantic HTML	5	Uses semantically correct HTML to ensure proper document structure. Properly utilizes HTML5 semantic elements.
User Registration Form (10 points)		
Form Creation	5	Develops a user registration form. Includes necessary fields for user registration.
Database Integration	5	Implements functionality to store user information in a database. Ensures secure storage of user data.
Form Validation (10 points)		
Client-Side Validation	10	Implements client-side form validation using JavaScript and DOM manipulation. Provides user-appropriate error messages for a smooth user experience.
Dynamic Behavior (15 points)		
Page Modification	10	Modifies pages according to user inputs or the application's current states.
Responsive Design (10 points)		
Responsive Interface	5	Creates a responsive and user-friendly interface using HTML and CSS. Ensures a consistent and pleasant user experience across different devices.
External CSS File	5	Utilizes at least one external CSS file for styling.
Adherence to Good Coding Practices (10 points)		
	10	Consistent Indentation: Meaningful Variable and Function Names: Include comments for clarity.
Adherence all the Submission requirements (10 points)		
	10	All the Requirement of Part III: Project Submission.
Total	100	

Suggested Weekly Schedule

- Week 9 – Form a team (max 3 members), decide on a project concept (be creative!). Design a user interface, including all necessary elements.
- Week 10 – Client-side Design (HTML/CSS)
- Week 11– Client-side Scripting (JavaScript)
- Week 12– Server-side Components (PHP/SQL)

Due Date

- Sunday. 24th Nov. Midnight (**no extension**)