

CS312 Internet Concepts
Term Project Assignment, V.1

Learning Objectives:

- Develop a basic website using common website languages.
- Apply principles of proper authentication to control access to a website.
- Apply principles of accessibility in constructing a website.

Overview: In this project, you will complete a functional website for a hypothetical, fictional organization. The website will provide information (passive content) as well as some forms of functionality (active content).

The coding will be done exclusively on your account on the CS Department linux servers, using exclusively the tools studied in class (no unauthorized external libraries). All code must be neatly formatted, readable, and properly documented. All generated web pages must be W3C compliant and must meet accessibility guidelines. The website must also follow all university rules for use of information technology equipment. For the sake of privacy, no personal images may be displayed without express permission of the subject.

This is an **individual** project requiring **original** work. All coding and all media (photos, video, text, etc.) used in this website must be original for this project. Nothing may be copied from other websites. Nothing may be shared with other students. All work must be your own.

Specifications: There are four separate deliverables for the project, which are listed below. Specifications for the first deliverable is provided here. The remaining specifications will be provided later in the semester.

- Proposal: Your website should describe a **fictitious** organization, e.g., a club, affinity group, church, charity, business, etc. It must, however, be authentic and realistic in nature (i.e., nothing funny or absurd such as “Tidewater Zombie Association”). It must not name any existing organizations (e.g., nothing with “ODU” in the title), and must not imply anything offensive (e.g., “Stupid XXXX People’s Club”). It can be political or religious in nature, just not in any offensive way.

Your proposal should be exactly one page long (single spaced), about 350 words. It should include the following:

1. Title: Your name, the name of the class, the name of the assignment
2. A description of the organization that your website will represent. Provide a thorough description that describes the mission and work of the organization and how the website will serve its mission
3. A list of 7-10 public web pages that the website will contain, with a one-line description of each page
4. A list of 3-5 members-only pages that will be hidden from the public, with a one-line description of each page

These lists are intended to ensure that your proposal is well thought-out. You are not bound to implementing only this content in the project – it can change later.

- Part 1: For this part of the project, you will create several static web pages, a common layout (with CSS), and a common template (header, menu, footer) that is used for all of the pages. The pages should be nicely formatted and attractive in form and color, but they need not be artistic or of commercial quality.
 - Page 1: index.html (home page): This page should include a welcome message, a logo or image, and general information about the organization.
 - Page 2: about.html (About Us): This page should include more detailed information about the organization, such as its mission, goals, leadership, location, etc.. The information should be fictional but realistic in tone.
 - Page 3: events.html (Events): Include a grid calendar or a chronological listing of events sponsored by the organization. Include at least 10 events, with a date, time, and brief description of each.
 - Pages 4-5: Create two additional pages that are relevant to your group. Be sure that these pages are included in the standard menu.
 - Style: The structure of all pages should include a standard header at the top, two or three columns for content, and a standard footer at the bottom. A standard menu should be placed below the heading, or on the left.

All web pages should use a common external style sheet. The style sheet should have rules for color, layout, and other visual attributes of the major page elements. It should contain at least 10 rules. All style elements should be placed in this sheet. The layout and style should be easy to read.

- General Requirements:
 - All of the files should be located in the directory /secure_html/cs312/project.
 - File names should be in lower case, spelled *exactly* as indicated here.
 - All files should have public read access.
 - All HTML and CSS code must pass W3C validation.
 - All HTML and CSS code must be hand-coded (not generated by any tools).
 - All code must be properly formatted and easy to read.
- Part 2: In this part of the assignment, you will add to your project two entry forms, one for registration and one for events.
 - registration.php: This page should include a registration form with the following fields. The page should be accessible via a link or button placed on the home page.
 - Userid
 - Password
 - Password confirmation
 - Name
 - Email address
 - 5 other fields
 - new_event.php: This page should accept the following event information. It should be accessible from a link or button on the events.html page.
 - Event name
 - Sponsor
 - Description (use a text area)
 - Date and Time (use appropriate input types)
 - Each field should have the appropriate HTML type, and should use validation patterns if possible.
 - The dual password fields should be validated for a match using Javascript before submitting the form.
 - The event date should be validated to eliminate past (prior to current date) events using Javascript before submitting the form.
 - If any validation fails, the form should not be submitted.
 - Appropriate, meaningful error messages should be provided for all validation test failures.
 - registration_action.php:
 - new_event_action.php: These modules should handle the form submits. They should validate each input field using regex patterns, and should return appropriate error messages if any errors are found. If no errors are found, they should write the data to a file (registrations.txt / events.txt), and return a message to confirm that the transaction was successful. Messages should appear to the user on a page by themselves, using the standard page format. Both data files should have public read access.
 - events.php should list all of the events contained in events.txt, in chronological order.
 - No Surprises Rule: All pages should work in an intuitive fashion, without presenting puzzling or confusing behavior to the user under any circumstances.
- Part 3: User Authentication / Registration; Database (TBA)

Grading:

- Criteria:
 - Completeness: The artifacts produced must meet all specifications given in the assignment.
 - Correctness: All artifacts must be correctly implemented, including correct coding and adherence to W3C and other appropriate standards.
 - Authenticity: The web site must provide a complete, meaningful, and seemingly authentic description of the (fictional) organization that it supports.

- Coding style: The code must be neat and readable, with proper comments and indentation.
- Breakdown:
 - Proposal: 10%
 - Part 1: 30%
 - Part 2: 30%
 - Part 3: 30%