

CPCS 403 – Final Project Description

Submit Wednesday, 10 December 2025 @ 11:59 pm

Submit via BlackBoard

Once completed, you should submit the zip file on the BlackBoard

Objective

The result of the final project will be a fully developed web application containing both front- and back-end that will be available publicly online, hosted by a domain.

1. Number of Pages

- For the project idea, you can adopt the idea you submitted in assignment 1 (if it requires both front-end and back-end development). Alternatively, you can choose a new idea.
- The web application must have a minimum of **10** total pages (each page requires both front- and back-end). The pages must be different and not identical/similar to each other in purpose.
- All pages must have proper content with appropriate titles/headings.
- Your website must contain the following types of content (discussed in this document):
 - Home page
 - At least one page must contain a table, such as a Schedule or any table suitable for your website idea. The table must be dynamically created and filled using data in your database.
 - At least one page must contain a picture gallery.
 - At least one page must contain a video.
 - A feedback form or another kind of form suitable to your website idea
 - Resume
 - Contact Us
 - An account creation page and a login page. Furthermore, at least one page must be protected such that only authenticated users can access them.
- Interactive function (get input from user and website display appropriate action/content)
- Database
- Add any other page(s) of your own choice with proper content according to your website idea
- Make sure that your website has good usability (the presentation of information and choices in a clear and concise way, a lack of ambiguity and the placement of important items in appropriate areas)

2. PHP

- The backend of the web application must be developed in PHP. The use of specialized/advanced PHP frameworks (e.g., Laravel) is not allowed.
- ALL pages must be PHP pages (saved as *.php) in order to use PHP *includes*.
- ALL pages MUST pass W3C validation.
- ALL pages must use at least TWO PHP includes for the header and footer.
 - **Header:** all pages should have a common “header” using PHP *includes*
 - The header should have your logo and the links of the main menu.
 - This header should be an include page within all other pages of the Web site
 - **Footer:** all pages should have a common “footer” section using PHP *includes*

- Your footer should have BOTH W3C validation images, and both images should be “clickable”. For HTML, you can use the validation service of HTML5.



3. Directory Structure

- All files for your website should be organized into appropriate directories:
 - All images should be saved in an “image” folder.
 - All PHP *include* files should be saved in “includes” folder/
 - All internal videos should be saved in a “videos” folder/
 - All HTML/PHP files should be saved in a “pages” folder.
 - All JavaScript files should be saved in a “script” folder.
 - All CSS files should be saved in a “css” folder.
- You can choose the names of these folders at your own choice, just make sure your files are organized appropriately.

4. Look (HTML, layout, and CSS)

- All pages should have a unified theme and color scheme, which is set in external CSS files. The use of inline or embedded CSS is not allowed.
- You MUST use the DIV approach to layout different components of your website (such as header, footer, content area, etc.) using CSS grid or flexbox.
- Your website should follow the hierarchical approach to website design
 - Depending on the content/purpose of your site, some of the internal pages may use a linear approach
- All layout/design should be done in CSS.
- You should have at least one CSS file for your entire site (i.e., shared by pages).

5. Table

- You should have at least one page with a table.
- This table should not be used for the layout of the page, but should be used to display some tabular data, such as a schedule or some other type of content that is best laid out within a table suitable to your website idea.
- The table must be filled dynamically using PHP code and data stored in your database.
- Your table should have multiple columns and rows, and it MUST use *colspan* and *rowspan*
- Printable table: In addition to the main CSS file, create a separate CSS (**print.css**) for customized print options. It should print only **table contents** (with caption) and nothing else.

6. Picture Gallery

- You should include a page containing an image gallery consisting of some thumbnails. Clicking one of the thumbnails causes the corresponding image to be displayed in a bigger size. Use JavaScript to implement this feature.

7. Video

- You should have a page with a Video. You have the option to display a local video or an external video (e.g., from YouTube).

8. Feedback Form (or another Form suitable to your idea)

1. Look

- The web application should have a form. The form should be divided into at least two sections. Use *fieldset*s to divide the sections.
- Proper labels must be assigned to all form elements.
- Your form should contain AT LEAST the following elements:
 - 4 textboxes (including email and name) and depending on your website idea.
 - 2 radio/option inputs.
 - 2 checkboxes.
 - 1 dropdown selection lists.
 - 1 Textarea box for feedback.
- Apply CSS for the form elements.

2. JavaScript Validation

- Validation must be applied by using external JavaScript on the form, which includes:
 - Input validation: the form should not submit with invalid data
 - Mandatory fields: the form should not submit with empty mandatory fields.
- Add all validation code in a separate file **validation.js**
- The use of advanced JavaScript frameworks and libraries is not allowed.

3. Server Processing and Validation

- Store all data on the server in the database - create a designated table(s) for this form.
- Apply validation on email ID so that there should be an only single entry for an email ID.
- In case the email ID is new, store the feedback data; otherwise give an error message to user without saving it.

9. Resume

- Include PDF version of your resume. The pdf resume must be displayed within a page without the need to download it or open it in a new tab by using the <object> tag for it.

10. Database (PHP + MySQL)

- Create a MySQL database on your hosting service.
- Add at least two tables to the database (other than “form” table, as mentioned in section 7).
 - Add appropriate tables that are related to your project idea.
- There must be a relationship between these two tables (primary/foreign keys).
- Ensure that you have proper PHP pages that add data to these tables.
- Ensure that you have proper PHP pages that fetch/display data from these tables.

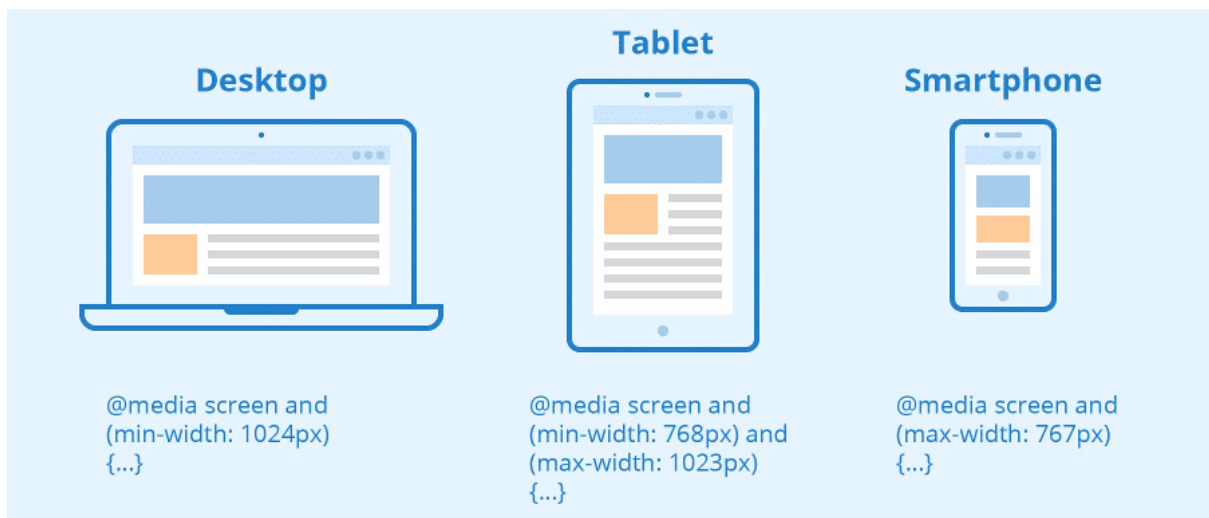
11. Security

- Ensure that all validations done at the client-side (via JavaScript) are also performed at the server side via PHP.

- Ensure that SQL injection attacks are handled properly.
- Ensure that Cross-Site Scripting attacks are handled properly.
- Ensure that you incorporate a secure design.

Responsive Design

- Your website must be responsive, supporting at least the dimensions shown below.
- The layout and content of your pages must be displayed correctly on these devices.
- You can rearrange the content or hide elements(e.g., table columns) based on their importance to fit the page on smaller screens.



Deliverables (what to submit)

- You must submit one ZIP file containing all the files and folders of your Web Application to Blackboard.
- The project must also be uploaded to your domain so that the web application becomes live and accessible via a URL. The project (including backend) must work correctly on the host.
- Word file that explains your website idea and design. You can build on the file submitted for assignment 1.
- Submit a script for creating the various tables used in the project.

Deadline

Submit on Wednesday, 10 December 2025, @ 11:59 pm.