

Course Title & Number:  
**WEB CLIENT DEVELOPMENT**

**420-DW2-AS**

**FINAL PROJECT**

Fall 2023

**- - - - - CONDITIONS - - - - -**

* Team of **3 students**.
* Create a GitHub **repository** in the beginning of the project and regularly update it to include all the files of your Web app at the end.
* Weighting: **30%** of the final grade.

**- - - - - PROJECT STATEMENT - - - - -**

1. Use the notions learnt in this class related to HTML, CSS and JavaScript (including jQuery), and your additional notions and experiences, in order to the **develop the client-side web application** described below. This application includes the following components:
   1. A **login form** shown the front page (homepage) that allows users to authenticate with a username and password. After a successful login, the questionnaire form described below in b. is displayed. Otherwise, a significant error message is shown.
   2. A **questionnaire form** that includes “The True Colors Personality Test” attached, including appropriate input fields (only “checkbox to select, input to type”, or drag and drop to order, must be used) that allow users to answer each of the 11 questions. Each single question must be displayed in a single webpage page. All the fields (checkbox, input, drag and drop) in a webpage must be filled-in first to be able to submit the form it includes. Otherwise, a significant error message is shown.
   3. When all the questions are submitted, the **result of the test** is displayed within a HTML table, including:

* The score for each of the 4 colors for each of the 11 questions.
* The total score for each of the 4 colors for all the 11 questions.
* A focus on the 2 highest score that represent the first and second color (for example shown in bold, with a higher font size, with a different color, with animation)
  1. When the result of the test is shown, **a description of the 2 colors** (first and second color) that score the highest score, or a hyperlink to access their description, is shown. In the situation where **there are no 2 colors** with that score the highest score, an appropriate message is shown.

1. **Test web client-side application** developed by another team using the functional test sheet provided. And make your application available to be tested by another team. (Peer review)
2. **Deliver an oral presentation** of your project to:
   1. Demonstrate how the application works (back-end and front-end);
   2. Explain how you developed it;
   3. Answer questions and discuss.

**- - - - - TECHNICAL SPECIFICATIONS - - - - -**

* + 1. The .pdf attached includes an introduction to the test “The True Colors Personality Test”. Read it carefully to understand how it works and how to generate its results. Use its content to include all the necessary information in your webpages in order to improve the user experience (i.e. Description of the test in the About Us page).
    2. Each webpage of the application must include images, background and other style elements to make it attractive. For example, the components (i.e.; form, result…) must be placed in the middle of the pages to leave more space to display background.
    3. Do not use the BOM to display the result of the test but use the DOM.
    4. Use a way, corresponding to the topic to be learned (client-side web development) to identify a user after a successful authentication and then personalize the test. For example Use JavaScript session storage (window.sessionStorage).
    5. The client-side username is “user1” and the password is “passAdmin01#”
    6. You are free to decide the html structure of the Web pages, but each webpage must respect the HTML standard and include at least a <header>, <body>, <nav>, <article>, and <footer> element.
    7. The navigation menu must include the following elements: Home, Login, and About that consecutively allow to access:

1. The Homepage
2. The Login form
3. The About Us page (that describe the test)
4. The Logout feature
   * 1. Include the name of the developers (team members) in each webpage of the application. For example, in the footer.
     2. When there’s an error in the information written in a form (validation failed), display a significant error message to allow the user to easily modify it and submit the form again, when applicable.
     3. Do not create isolated pages, that is, pages that do not contain at least one hyperlink that allows the user to navigate to another page, without manually changing the URL in the address bar of the web browser.
     4. To make your website's code easy to review and maintain, it’s mandatory to format it appropriately by:

* Adding indents (a larger left margin indicating what element is imbedded inside others);
* Use lowercase letters, except when this is not recommended (e.g. DOCTYPE);
* Adding at least 5 significant comments in each file.

For example, you can use an appropriate extension for Visual Studio Code to automatically format your program code.

* + 1. Use standard file names, such as index.php and style.css.
    2. Use standard directory names to group files, such as css, image, js, and font
    3. Include a readme.txt file in the root of your program that indicates:

1. The full names of the developers (team members) and the contribution of each to the program built.
2. Any interesting indications about the program built, for example the version of php used, how it works, and how to use it.

**- - - - - ADDITIONAL DOCUMENTS - - - - -**

2.**True Colors Personality Test** (Description and operation)

3.**Task assignment and Timeline** of the project

4.**Functional Test sheet** of the application

5.**Evaluation grid** of the project