ENSF-381: Full Stack Web Development Laboratory

Ahmad Abdellatif and Novarun Deb
Department of Electrical & Software Engineering
University of Calgary

Objectives

Welcome to the ENSF381 course lab! Below are the detailed instructions for creating and understanding an HTML document. The main objective of this lab is to apply the fundamental concepts of HTML structure, semantic elements, and inline styling.

Groups

Lab instructions must be followed in groups with a **maximum of two students**.

Submission

You must submit the complete source code, ensuring it can be executed without any modifications. Also, if requested by the instructor, you may need to submit the corresponding documentation file (e.g., word and image). Only one member of the group needs to submit the assignment, but the submission must include the names and UCIDs of all group members at the top of the code.

Deadline

Lab exercises must be submitted by **11:55 PM on the same day as the lab session**. Submissions made within 24 hours after the deadline will receive a maximum of 50% of the mark. Submissions made beyond 24 hours will not be evaluated and will receive a grade of zero.

Academic Misconduct

Academic Misconduct refers to student behavior which compromises proper assessment of a student's academic activities and includes: cheating; fabrication; falsification; plagiarism; unauthorized assistance; failure to comply with an instructor's expectations regarding conduct required of students completing academic assessments in their courses; and failure to comply with exam regulations applied by the Registrar.

For more information on the University of Calgary Student Academic Misconduct Policy and Procedure and the SSE Academic Misconduct Operating Standard, please visit: https://schulich.ucalgary.ca/current-students/undergraduate/student-resources/policies-and-procedures

Exercise 1: UCalgary Wiki Page

Objective: The main goal of this exercise is to create UCalgary Wiki HTML page similar to the given example (output.png). You need to follow the following instructions. All the required text is taken from the UCalgary page on Wikipedia and can be found in **info.txt.**

1. **Document Structure**
   - Create an HTML file.
   - Start your file with the <!DOCTYPE html> declaration.
   - Use <html>, <head>, and <body> tags to structure your document as we covered in our class.
   - Include a <title> in the <head> section to name your webpage.
   - The title is "University of Calgary".
2. **Header and Text Content**
   - Use <h1> to define the main heading of your webpage. For this lab, it should read "University of Calgary".
   - Add a descriptive paragraph about the university using the <p> tag. Emphasize important text using <b> (bold) and <i> (italic). Refer to the output screenshot for guidance on formatting and content structure.
3. **Subheadings and Formatting**
   - Create subheadings using <h2> and <h3> tags to organize content under "History," "21st Century," and "West Campus Development Project".
   - Use inline styles such as font color for headings.
   - Draw a horizontal line after the main sections.
4. **Details on History and Administration**
   - Include a detailed description of the university's history using <p> tags. Highlight specific dates and terms with <b> and <u> (underline). Refer to the output screenshot for guidance on formatting and content structure.
   - Describe the university's governance using the <font> tag for special formatting, and mention key roles and responsibilities.
5. **Custom Styling**
   - Apply inline styles (e.g., color, font size) to distinguish sections. Example: Set the font color of headings to navy.
   - Use a <font> tag to set text styles explicitly, such as size or colour, where necessary. Refer to the output screenshot.
6. **Group Information Section**
   - Create a section for group member names and UCIDs using <pre> and <font> tags.
   - Note the vertical space between the Administration section and the group information section.
   - Add your name and the names of your group colleague along with their UCIDs.

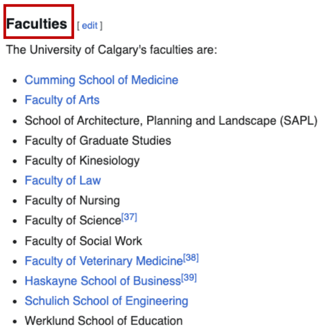
Notes:

- Ensure all opening tags have corresponding closing tags.
- Use the paragraph tag for each block of paragraph text.
- Ensure that the final output of your HTML page matches the provided output screenshot exactly.

Exercise 2: Inspecting HTML Elements

Objective: Learn how to inspect and extract specific HTML elements from a webpage using the browser's Developer Tools.

1. **Open the Wikipedia Page**
   - Open your preferred web browser (e.g., Chrome, Firefox, or Edge).
   - Go to the following URL: University of Calgary - Wikipedia.
2. **Locate the "Faculties" Section**
   - Scroll down the page to find the **"Faculties"** section. This section contains a list of faculties available at the University of Calgary.



Faculties [edit]

The University of Calgary's faculties are:

- Cumming School of Medicine
- Faculty of Arts
- School of Architecture, Planning and Landscape (SAPL)
- Faculty of Graduate Studies
- Faculty of Kinesiology
- Faculty of Law
- Faculty of Nursing
- Faculty of Science[37]
- Faculty of Social Work
- Faculty of Veterinary Medicine[38]
- Haskayne School of Business[39]
- Schulich School of Engineering
- Werklund School of Education

3. **Open the Developer Tools**
   - Right-click anywhere on the page (preferably near the "Faculties" section).
   - From the context menu, select **"Inspect"** (in Chrome and Edge) or **"Inspect Element"** (in Firefox).

   Alternatively, you can open the Developer Tools using the keyboard shortcut:

   - **Windows/Linux:** Press `Ctrl + Shift + I`
   - **Mac:** Press `Cmd + Option + I`
4. **Find the "Faculties" HTML Element**
   - In the Developer Tools panel, hover over different parts of the HTML in the **Elements** tab. The corresponding section on the webpage will be highlighted.
   - Move your mouse until the **"Faculties"** header or its content is highlighted on the webpage.
5. **Expand and View the HTML Structure**
   - If the HTML element corresponding to "Faculties" has a small triangle next to it, click the triangle to expand and see its child elements.
   - Carefully observe the HTML tags and hierarchy. Identify where the "Faculties" section begins and ends.
   - Take a screenshot of the "Faculties" section highlighted in the Developer Tools panel and submit it.

Submissions

You need to submit the following items:

1. Fill out the Answer_sheet.docx, including **the names of the group members along with their UCIDs**, a **screenshot of the output for Exercise 1**, and the **answer to Exercise 2**. You need to **<u>submit the PDF on D2L</u>**. Check the attached template for the answer sheet.
2. Upload the code for Exercise 1 to GitHub:

Part 1: Creating a GitHub Repository

1. Open <u>GitHub</u> in your browser.
2. Sign in with your GitHub account. If you do not have an account, create one.
3. Click on the + icon in the top-right corner of the GitHub page.
4. Select "New repository".
5. Fill in the repository details:
    - Repository Name: Provide a descriptive name (e.g., lab01).
    - Description (optional): Add a short description of the project.
    - Visibility: Choose Public or Private.
6. Check the box "Add a README file".
7. Click "Create repository".

Part 2: Cloning the Repository

1. Copy the Repository URL
    - On the repository page, click the green "Code" button.
    - Copy the URL under HTTPS (e.g., https://github.com/your-username/html-lab.git).
2. Open Your Terminal
    - Open a terminal or command prompt on your computer.
3. Clone the Repository
    - Navigate to the folder where you want to store the repository:

```
cd ENSF381
```

    - Run the git clone command with the repository URL:

```
git clone https://github.com/your-username/lab01.git
```

    - This will create a local copy of the repository in the specified folder.

Part 3: Adding Code and Output Image

1. Navigate to the Repository
    - In the terminal, move into the cloned repository folder:

```
cd lab01
```

2. Add Your Code and Output Image
    - Place your code file (e.g., index.html) and the output image (e.g., output.png) into the cloned repository folder.
3. Stage the Files
    - Use the git add command to stage the files for commit:

```
git add index.html output.png
```

4. Commit the Changes
    - Write a commit message describing your changes:

```
git commit -m "Add HTML code and output image"
```

5. Push the Changes to GitHub
    - Push your changes to the remote repository:

```
git push origin main
```