

Assignment 2 - Web development

Introduction

In this assignment, you will build a personal webpage and enhance it with an interactive notebook function using only HTML and JavaScript. This project is designed to help you practice essential web development skills, particularly working with the Document Object Model (DOM) to create dynamic and user-interactive web pages.

Objective

This assignment is designed to help students build practical web development skills by creating a personal webpage with an interactive notebook feature. Through this hands-on task, students will apply foundational HTML and JavaScript knowledge to manipulate web content dynamically. The students will:

- Practice structuring a personal webpage using HTML.
- Learn to manipulate the DOM using JavaScript.
- Implement a basic notebook function to add and display notes.
- Understand how user interaction can dynamically change webpage content.
- Build confidence in writing client-side scripts without relying on external libraries.
- Strengthen readiness for more advanced web development tasks.

Assignment description

The assignment has two parts:

- Part 1 involves designing a simple personal webpage that introduces yourself—your name, interests, and projects—using semantic HTML to organize your content clearly.
- Part 2 is to implement a simple notebook feature using HTML and JavaScript with DOM manipulation. The notebook will allow users to add, display, and manage personal notes directly on the webpage. There is no need to save or load notes from any external file or storage; the notes exist only during the current browser session.

Part 1 – Personal website

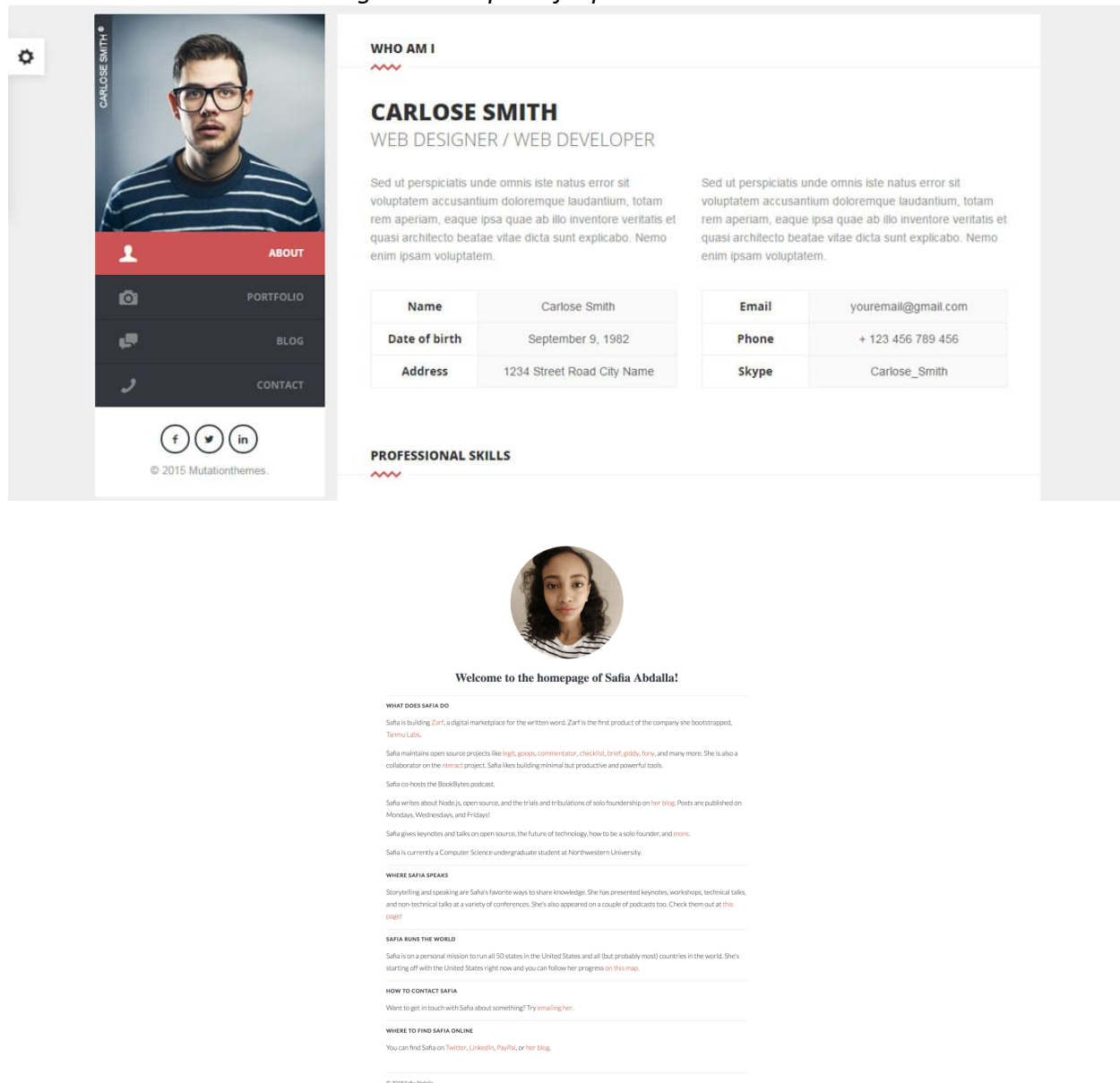
In this part, you will create your own personal website using HTML and JavaScript. Your personal website will showcase your skills, interests, projects, and achievements.

You will use HTML to add the content of your personal website. You will use HTML elements, attributes, links, images, lists, tables, etc. to organize and format your content. You will also use HTML comments to document your code. Your personal website should have at least the following information:

- Your name and a brief introduction about yourself, with your professional photo.
- A section with more details about your background, education, skills, hobbies, etc.
- A portfolio section with some examples of your projects or work samples.
- A contact section with your contact information and a form to send you a message.

Publishing your website on the Internet is a required part of the assignment. There are many free hosting services you can land on. You can choose any web host that you prefer, such as GitHub Pages, Netlify, Heroku, etc. However, some web hosts may have different requirements and limitations for hosting websites, so it is recommended that you read their documentation and follow their instructions carefully.

Figure: Examples of a personal website.



Part 2 – Notebook function

In this part, you will add a Notebook feature to your personal webpage that allows users to create and view personal notes dynamically. The notebook will be fully implemented using JavaScript

and DOM manipulation, enabling users to add notes which are immediately displayed on the page. It is acceptable for the notes to exist only during the current browser session. In other words, the notes will disappear once the browser is refreshed.

Requirements

- Provide an input area (e.g., a *textarea*) for users to type their notes.
- Include an “Add Note” button to submit the note.
- Display all added notes in a dedicated section of the page.
- Clear the input field after each note is added.
- Notes only persist while the webpage is open; reloading the page clears all notes.
- *Note:* Clicking the Add Note button while the input area is empty shows a notification to ask users to type their notes.

Instructions

- Create the Input Interface
 - Add a *textarea* or input box for note entry.
 - Add a *button* labeled “Add Note” below or beside the input.
- Add Event Handling
 - Write JavaScript to listen for the button’s click event.
 - When clicked, read the note text from the input field.
- Add Notes to the Page
 - Create a new DOM element (like a *div* or *p*) containing the note text.
 - Append this new element to the notes display area on your webpage.
- Clear the Input
 - After adding the note, clear the input field to allow entering new notes easily.
- Notes Lifecycle
 - Notes only live in the current session.
 - Reloading the page clears all notes automatically.

Figure: Example of the notebook’s interface in your website.

My Personal Notebook

Type your note below and click "Add Note". Notes will appear below and will not be saved after page reload.

Write your note here...

Text box for typing the message

Add Note ← Add-note button

Notes:

Hello, nice to meet you! Area to display the notes

You have a beautiful website.

Submission

The structure of submission folder should be organized as follows:

- <StudentID>-assignment-webdev.html: HTML file containing HTML and JavaScript code of the personal web page and the notebook, for example *ICS-2012345-assignment-webdev.html*.
- If your personal website includes multiple pages, compress all related files into a single .zip file. Name the .zip file using the same format as above (e.g., *ICS-2012345-assignment-webdev.zip*).

Please include the website link in your submission comment. This not only fulfills the project requirement but also helps you gain practical experience in web deployment and showcase your work to a wider audience.

Notes

You can use any text editor or IDE (Integrated Development Environment) that you prefer, such as Notepad++, Sublime Text, Visual Studio Code, etc. You can also use any web browser that you prefer, such as Chrome, Firefox, Safari, etc. However, some browsers may have different levels of support for certain web features, so it is recommended that you test your website on multiple browsers to ensure compatibility.

Evaluation

This rubric outlines how your submission will be evaluated based solely on functional aspects of your personal webpage and notebook features.

Table: Evaluation rubrics

Component	Functional requirements	Points
Part 1: Personal Webpage Functionality (50%)	Webpage loads correctly with no errors.	20%
	Personal content is completely displayed (e.g. name, intro, contact, professional experience and projects, etc.).	30%
Part 2: Notebook Functionality (40%)	A form or input box allows users to add a new note.	15%
	Notes are displayed immediately after being added	15%
	Interface is functional and free from errors	10%
Part 3: Website Publication	The website is successfully published, and the link is submitted*	10%
Total		100%

*To complete your project, you must publish your website online so it can be accessed and evaluated. This step demonstrates your ability to research and deploy a website, ensuring it runs correctly in a real environment. Follow the steps below to publish and submit your website link:

- Choose a free hosting platform, e.g., GitHub Pages, Netlify, Vercel, etc.
- Upload your project files to the platform.
- Complete the publishing process as guided by the platform.

- Once deployed, verify that your website works correctly online, for example, <https://myname.site>.
- A link to the website should be left as a comment for verification purpose.

Assignment Comments

0

My personal website address is
<http://myname.site>

👍 🙌 😊 🔊 🎥 📎 Submit

Deadline

Please visit Canvas for details.