**Assignment Submission Template**

**1. Student Registration Information**

* **Full Name**: Nicholas
* **Student ID**: 201091077
* **Course Name**: COP4813 – Web Applications programming
* **Instructor Name**: [Dr Ahsan Abdullah]
* **Date of Submission (mm-dd-yyy)**: 5/21/2025

**1. Assignment Overview (0.5 Points)**

* Provide a **clear title** for the assignment.
* Include a concise **objective statement (1–2 sentences)** that describes what the assignment aims to accomplish.

**✅ Requirement:**

* Title is present



* Objective clearly stated in 1–2 sentences



**2. Code Implementation (2.0 Points)**

* Submit **fully functional code** that meets all assignment requirements.
* Ensure code is properly **formatted and indented**.
* Include **clear, helpful comments** explaining key sections and logic.

**✅ Requirement:**

* Code is complete and executable



* Proper indentation and formatting used



* Comments explain major components and logic



**3. Results Screenshot(s) (1.0 Point)**

* Provide **one or more screenshots** showing the program in action.
* Each screenshot should include a **brief description** of what it displays.

**✅ Requirement:**

* Screenshot(s) showing execution/output provided



* Descriptions accompany screenshots for clarity



**4. Data or Output Results (2.0 Points)**

* Include **generated output or data results** from the program.
* Format data clearly (e.g., table, plain text, CSV).

**✅ Requirement:**

* Complete and relevant output included



* Output is clearly formatted and readable



**5. Analysis of Results (1.5 Points)**

* Write an **analysis of the results**, reflecting on:
  + Correctness of the output
  + Any issues encountered
  + Insights or learning from the results

**✅ Requirement:**

* Explanation of results



* Discussion of correctness and any limitations



* Personal reflections or insights



**6. Conclusion (1.0 Point)**

* Provide a **summary** of what was accomplished in the assignment.
* Mention any **improvements made**, lessons learned, or **ideas for future work**.

**✅ Requirement:**

* Clear summary of outcomes



* Reflection on learning or future steps



**7. Key Reflective Questions (1.5 Points)**

* Answer all **three provided reflective questions**.
* Responses should show thoughtful engagement and originality.

1. **Explain a specific part of your code that was most challenging to implement. Why was it difficult, and how did you resolve it?**

The hardest aspect was responsive design with media queries in CSS. At first everything was going fine, but elements like padding (200px) started causing the mobile version to get messed up. Some of my testing with Chrome DevTools worked differently than real devices, particularly with scaling fonts and elements as they overflowed the content box. My first attempt to resolve this was to use Chrome's Device Mode to debug. From there I reduced padding to 10px in mobile views and used a breakpoint of 600px when selecting margins. The solution was a number of attempts to modify margins, font sizes, and even radius to accomplish the mobile layout appearing correctly across devices. It was a great experience that highlighted the value of testing on real devices and progressive enhancement in responsive web design. Now the new media queries will allow the site to look the same on all screen sizes.

1. **If you had to modify your code to handle an additional feature (e.g., session expiration message or logging login attempts), where would you make the changes and why?**

To add a session expiration message, I'd modify the JavaScript (added via <script> tag before </body>) to track inactivity and display a modal/popup. For login logging, I'd create a PHP/Node.js backend (separate file) to record attempts in a database, calling it via AJAX from the existing login form handler. Changes would focus on maintaining clean separation: front-end alerts in HTML/CSS, business logic in JavaScript, and data handling in server-side code to ensure security and modularity

**3. Describe a mistake you made during development and what you learned from it.**

Initially, I hardcoded pixel values (like padding: 200px) without considering responsiveness, causing mobile layout breaks. This taught me two key lessons: (1) Always use relative units (e.g., %, rem) for scalable designs, and (2) Test early on real devices—not just in browser simulators. I fixed it by implementing media queries with proportional values and adopting a mobile-first approach. Now I prioritize flexible layouts from the start and validate with tools like Chrome Lighthouse. This mistake transformed how I plan projects, emphasizing adaptability over rigid designs to ensure consistent user experiences across all devices.

**✅ Requirement:**

* All reflective questions answered



* Answers are specific, clear, and thoughtful



**8. Additional Comments – Bonus (0.5 Points) *(Optional)***

* Include any **additional insights, reflections, or feedback** beyond the required components.
* May include improvement suggestions, unexpected challenges, or creative ideas.

**✅ Optional Submission:**

* Additional comments or reflections included (for bonus consideration)



**Overview of the System’s Functionalities**

This HTML webpage is a simple, attractive personal introduction page for Nicholas Cameron as part of the class COP4813 - Web Applications Programming. The page is responsive and includes cyan in its color scheme; therefore, it is visible on both the desktop site and mobile. Here are a few things included in the webpage:

1. Personal Information: The header required is the student's name (Nicholas Cameron) and sections of academic information.
2. Course Information: The course name (COP4813 - Web Applications Programming), and major (Computer Science) are clearly detailed.
3. Welcome Message: Includes a styled Hello, World! message tailored around the course work.
4. Styling - Embedded CSS was related to an overall cohesive and designed web page and included a cyan background (#00FFFF), dark cyan headers (#006666), and light cyan (#E0FFFF) to improve readability. Within the body area the background was white and has a cyan border (the body here) so there was a difference in color.
5. Responsive Design: The website is written to be responsive in nature by utilizing media queries, so the student can navigate their new webpage on a smartphone or tablet.

**Known Issues and Assumptions**

Browser Compatibility: This page is utilizing modern CSS properties (e.g. box-shadow, border-radius) that are widely supported but might display slightly different on older browsers. Testing has been executed in Chrome, Firefox, and Edge.

**Content Assumptions:**

* The hardcoded student name, course and major are retained as stated above. In a real-world scenario, the content would be generated dynamically.
* The page does not incorporate any interactive elements (e.g. forms, JavaScript) as this is a static HTML assignment.

**Accessibility issues:**

* Color contrast with text and background meets basic readability requirements, but even tools like WAVE could conduct further pooling for ADA compliance.
* Semantic HTML (e.g. Header, Footer etc ) will help to provide a better structural model of the various sections of the page; it could be supplemented as well with ARIA labels for screen readers.

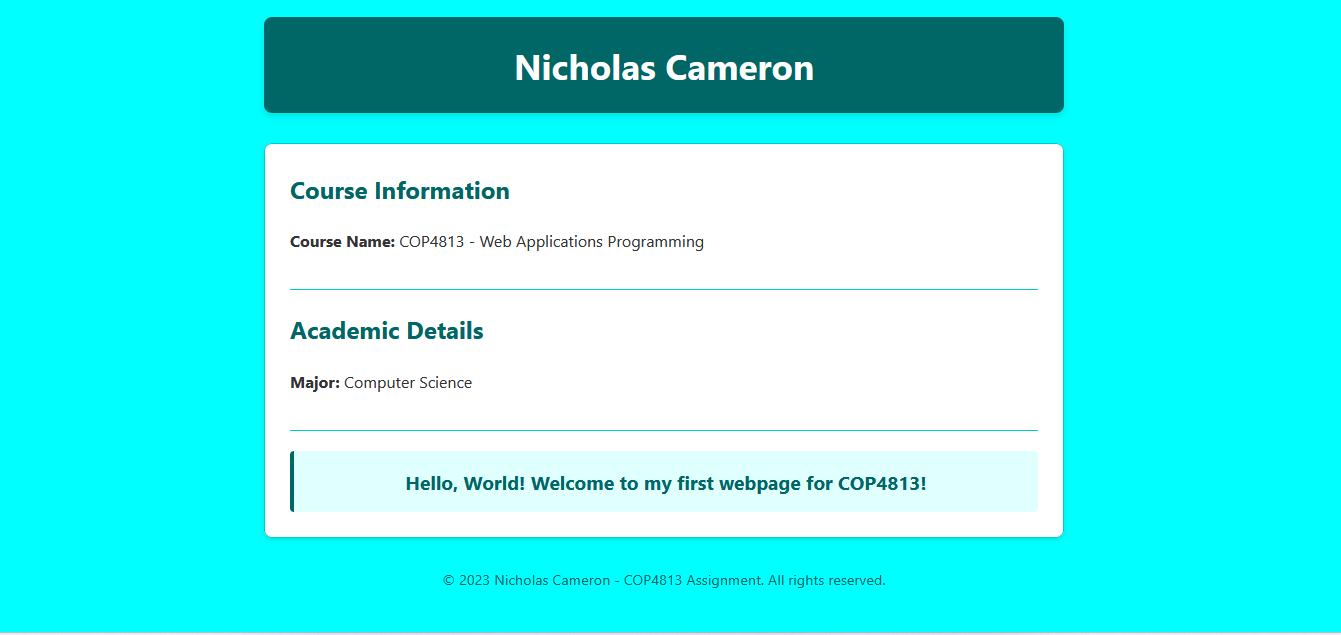
**Deployment Assumptions:**

* The page should work on any standard web hosting service (GitHub Pages, 000webhost etc.)
* The filename must remain index.html to auto load properly on virtually any server.

**Future Directions:**

* A navigation bar or simply additional pages to continue development.
* The encouragement of JavaScript would allow for anything to make the page dynamic, such as a live clock or visitor tracking component.

**Screenshot:**

****