



GINA CODY SCHOOL OF ENGINEERING AND COMPUTER SCIENCE
COURSE CODE: SOEN-287 (Section S)
COURSE TITLE: Web Programming
SEMESTER: Winter 2023

INSTRUCTOR:	Dr. Bonaventure Chidube Molokwu
ASSESSMENT TITLE:	Assignment 02
ASSESSMENT CODE:	SOEN-287-S-A02
DURATION/DEADLINE:	11.59pm, Monday, March 6

Question (20 points):

With regard to your initial task where you designed a (*static*) personal website for yourself; this task will require you to create an administrative website for managing the display content of your personal website. To this end, you shall be storing the display content populated from your administrative website into local (*.txt*) files.

Furthermore, this task is based on the concepts and knowledge of HTML, CSS, and JavaScript. This will require you to employ the concepts of managing Sessions (i.e. `sessionStorage` property with respect to the `window()` object) and working with files in JavaScript. Kindly design each *.html* webpage of your administrative website as required and as stated below. To this effect, your administrative website folder MUST be renamed to: **admin_LastName_StudentNumber** (e.g. *admin_Molokwu_12345678*), and each webpage of the administrative website MUST be styled using either **Style-A** or **Style-B** layouts as defined below.

Table 1: **Style-A** layout (banner menu layout)

Home	Resume	Projects	Contact	Social	Logout
Body content					

Table 2: **Style-B** layout (left-panel menu layout)

Home	Body content
Resume	
Projects	
Contact	
Social	
Logout	

Home (*adminIndex.html*) - 3.0 points:

The `adminIndex.html` page will act as an entry point for storing dynamic content which shall be displayed onto your `index.html` page (on your personal website). At the moment, you aim is to implement and accomplish the following, viz:

- (i) A content box, using the `<textarea>` tag, which should be labeled *Professional Statement* - (0.5 point).
- (ii) A content box, using the `<textarea>` tag, which should be labeled *Brief Biography* - (0.5 point).
- (iii) A submit button, using the `<input />` tag, which should be labeled *Submit* - (0.5 point).
- (iv) A JavaScript function identified as: `submitIndex`. This function should be activated/triggered when the administrator clicks on the *Submit* button defined above. Also, this function should copy the contents of (i) and (ii) above; and store them in one or more (*.txt*) file(s) on your local computer/PC - (1.5 points).
- (v) Note: Alignment positions, with respect to the textual content and webpage layout, are based on your discretion.

Resume (*adminResume.html*) - 5.5 points:

The `adminResume.html` page will act as an entry point for storing dynamic content which shall be displayed onto your `resume.html` page (on your personal website). At the moment, you aim is to implement and accomplish the following, viz:

- (i) A content box, using the `<textarea>` tag, which should be labeled *Educational Qualifications* - (0.5 point).
- (ii) A content box, using the `<textarea>` tag, which should be labeled *Skill Set* - (0.5 point).
- (iii) A content box, using the `<textarea>` tag, which should be labeled *Awards/Recognition* - (0.5 point).
- (iv) A content box, using the `<textarea>` tag, which should be labeled *Work Experience* - (0.5 point).
- (v) A content box, using the `<textarea>` tag, which should be labeled *Referees* - (0.5 point).
- (vi) A submit button, using the `<input />` tag, which should be labeled *Submit* - (0.5 point).
- (vii) A JavaScript function identified as: `submitResume`. This function should be activated/triggered when the administrator clicks on the *Submit* button defined above. Also, this function should copy the contents of (i) - (v) above; and store them in one or more (*.txt*) file(s) on your local computer/PC - (2.5 points).
- (viii) Note: Alignment positions, with respect to the textual content and webpage layout, are based on your discretion.

Projects (*adminProjects.html*) - 3.0 points:

The `adminProjects.html` page will act as an entry point for storing dynamic content which shall be displayed onto your `projects.html` page (on your personal website). At the moment, you aim is to implement and accomplish the following, viz:

- (i) A content box, using the `<textarea>` tag, which should be labeled *My Projects* - (0.5 point).
- (ii) A submit button, using the `<input />` tag, which should be labeled *Submit* - (0.5 point).
- (iii) A JavaScript function identified as: `submitProjects`. This function should be activated/triggered when the administrator clicks on the *Submit* button defined above. Also, this function should copy the content of (i) above; and store it in a *(.txt)* file on your local computer/PC - (2.0 points).
- (iv) Note: Alignment positions, with respect to the textual content and webpage layout, are based on your discretion.

Contact (*adminContact.html*) - 3.0 points:

The `adminContact.html` page will act as a display page for reading existing messages from the `contact.html` page of your personal website. At the moment, your aim is to implement and accomplish the following, viz:

- (i) A submit button, using the `<input />` tag, which should be labeled *ReadMessage* - (0.5 point).
- (ii) A JavaScript function identified as: `readMessages`. This function should be activated/triggered when the administrator clicks on the *ReadMessage* button defined above. Also, this function should be capable of loading a *(.txt)* file, from your local computer/PC; and display its content on the *adminContact.html* page. Optionally, you may choose to format the display of the file *(.txt)* contents in a tabular format - (2.5 points).
- (iii) Note: Alignment positions, with respect to the textual content and webpage layout, are based on your discretion.

Social (*adminSocial.html*) - 4.0 points:

The `adminSocial.html` page will act as an entry point for storing dynamic content which shall be displayed onto your `social.html` page (on your personal website). At the moment, your aim is to implement and accomplish the following, viz:

- (i) A textbox, using the `<input />` tag, which should be labeled *Social Link 1* - (0.5 point).
- (ii) A textbox, using the `<input />` tag, which should be labeled *Social Link 2* - (0.5 point).
- (iii) A textbox, using the `<input />` tag, which should be labeled *Social Link 3* - (0.5 point).
- (iv) A submit button, using the `<input />` tag, which should be labeled *Submit* - (0.5 point).
- (v) A JavaScript function identified as: `submitSocial`. This function should be activated/triggered when the administrator clicks on the *Submit* button defined above. Also, this function should copy the contents of (i) - (iii) above; and store them in one or more *(.txt)* file(s) on your local computer/PC - (2.0 points).
- (vi) Note: Alignment positions, with respect to the textual content and webpage layout, are based on your discretion.

Logout (*logout.html*) - 1.5 points:

The `adminLogout.html` page will act as a prompt page which an administrator gets to view upon log-out or sign-out action. At the moment, your aim is to implement and accomplish the following, viz:

- (i) A submit button, using the `<input />` tag, which should be labeled *Logout* - (0.5 point).
- (ii) A JavaScript function identified as: `processLogout`. This function should be activated/triggered when the administrator clicks on the *Logout* button defined above. Also, this function should be capable of clearing all JavaScript-defined sessions on your website/webpage. In this regard, you should consider using the `sessionStorage` property, of the `window()` object, which exposes you to the `Storage()` object in JavaScript - (1.0 points).
- (iii) Note 1: Alignment positions, with respect to the textual content and webpage layout, are based on your discretion.
- (iv) Note 2: <https://developer.mozilla.org/en-US/docs/Web/API/Window/sessionStorage>

Rules for Acceptable Submission

- (1) Every valid submission MUST be submitted as a compressed directory/folder (.zip) file via MoodleTM.
- (2) Validate each of your webpage via the [W3C Markup Validator](#); and ensure that all errors are fixed prior to submission on MoodleTM. Each validation error is mapped to -1.0 point.
- (3) You MUST use and maintain a consistent filenames standard with respect to the constituent webpages of your website directory/folder. Note that you MUST rename your website folder to: “**admin_LastName_StudentNumber**” prior to upload and submission via MoodleTM.
- (4) All XHTML tags MUST be in lowercase standard. Avoid spelling and grammatical errors.
- (5) Maintain a uniform background color throughout your webpage.
- (6) Any form of advertisement whatsoever is unacceptable.
- (7) Any type of pornographic or nude content whatsoever is unacceptable.
- (8) Since a webpage cannot be dismembered; it MUST always have one (1) `<head>` section and one (1) `<body>` section in that order.
- (9) No commercial/public webpage(s) or website(s) of any kind will be accepted. A score of zero (0) will be awarded if this rule is broken.
- (10) This is strictly an individual project/task. Thus, each student’s submission MUST be unique such that the content of your website and webpages MUST reflect and be your original work. Plagiarism of any form or kind will result in a score of zero (0).
- (11) All the aforementioned rules are associated with *penalty points* when/where broken. Thus, those *penalty points* which have not been explicitly stated herein are subject to the instructor’s discretion; and they will be reported to any defaulting student after grading, if/where necessary.
- (12) Kindly review the course outline for the university’s policies regarding Academic Integrity and Student Code of Conduct - <https://www.concordia.ca/academics/undergraduate/calendar/current/17-10.html>