WEB322 Assignment 3

Submission Deadline:

Sunday, October 17th, 2021 @ 11:59 PM

Assessment Weight:

9% of your final course Grade

**Background**: You will need to have access to a code editor. You will also need a thorough understanding of HTML 5, CSS3, Javascript, “Node.JS” and Express.

**Assignment Submission Requirements**

* **Submit the link of your hosted Heroku project in Blackboard.**
* **Submit a zipped folder on Blackboard containing only the html, js and css files. (Please do not include node\_modules and git folder)**

## **Assignment Regulations**

* **This assignment must be done individually.**
* **Please review Seneca’s policies on Academic Integrity, specifically:**

*“Each student should be aware of the College's policy regarding Cheating and Plagiarism. Seneca's Academic Policy will be strictly enforced.To support academic honesty at Seneca College, all work submitted by students may be reviewed for authenticity and originality, utilizing software tools and third party services. Please visit the Academic Honesty site on http://library.senecacollege.ca for further information regarding cheating and plagiarism policies and procedures.  
.”* ***Thus, ensure that your code or any part of it is not duplicated by another student(s). This will result in a percentage of zero (0%) assigned to all parties involved.***

**Detailed App Specification**

This assignment is a continuation of Assignment 2 thus all the requirements for this assignment is to be made “on top” of your assignment 2.

Note, **no database connectivity is required for Assignment 2.**

## Registration Page

You are required to build a **well designed** user registration form as shown in **Figure1.1**. This can be implemented as an entirely new page or as a pop-up modal. In the event you implement this as a new page then you must ensure that the page maintains consistency regarding its structure, header, nav, hero section, footer, etc as the home page.

**Figure 1.1**

**Graphical user interface, application

Description automatically generated**

## Login Page

You are required to build a **well designed** login form as shown in **Figure1.2**. This can be implemented as an entirely new page or as a pop-up modal. In the event you implement this as a new page then you must ensure that the page maintains consistency regarding its structure, header, nav, footer, etc as the home page.

**Figure1.2**

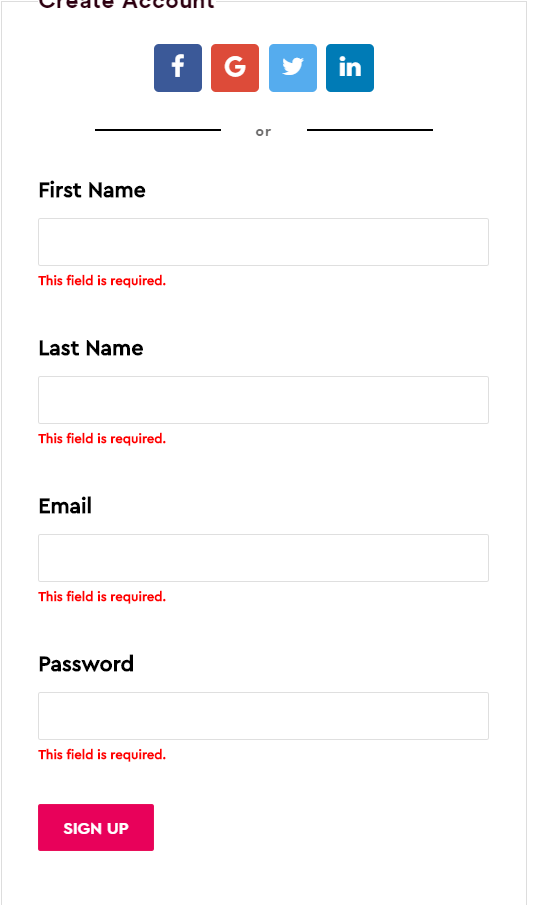
Graphical user interface, application, Teams

Description automatically generated

# Server Side Validation

* For the login form, you are required to ONLY check for nulls (i.e, check to see if the user entered a value in the respective text fields). However, for the registration form, you have to check for nulls AND implement at least 2 complex validation criteria using regular expressions on two separate fields (For example, enforcing that the user must enter a password that is 6 to 12 characters, and the password must have letters and numbers only).
* All error messages must be rendered on their respective pages or areas and must be styled properly as shown below.

**Figure 1.3**



## User Registration Form

When a user fills out the registration form and then hits the submit button, provided that all the validation criteria were not violated, your website must then **redirect the user to a dashboard page.** For now, the dashboard page should contain information welcoming the user and should be properly styled.