Introduction

This assignment is the third of six assignments. It has been designed to give you practical experience capturing form-submissions and validating forms using server-side JavaScript. You will have the opportunity to store your source-code on a GitHub repository and host your web app on Heroku. Finally, you will use the SendGrid service to send emails.

Before you begin this assignment, you must finish your previous assignment. All objectives listed for this assignment are to be made "on top" of your previous assignment.

This assignment is worth 9% of your final grade.

Note: Database connectivity is **not** required for this assignment.

Reminder about academic integrity

You must comply with Seneca College's Academic Integrity Policy. Although you may interact and collaborate with others, this assignment must be worked on individually and you must submit your own work.

You are responsible to ensure that your solution, or any part of it, is not duplicated by another student. If you choose to push your source code to a source control repository, such as GIT, ensure that you have made that repository private.

A suspected violation will be filed with the Academic Integrity Committee and may result in a grade of zero on this assignment or a failing grade in this course.

Technical Requirements

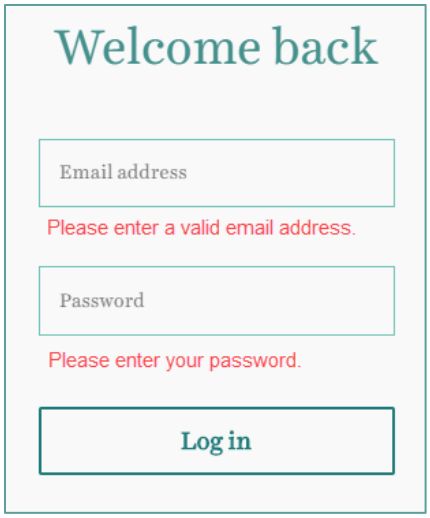
- All back-end functionality **must** be done using **Node.js** and **Express**.
- You will use the **body-parser** module to handle form submissions.
- Your views **must** be created with **Express-Handlebars**.
- You **can use** a front-end CSS framework such as Bootstrap, Bulma or Materialize CSS to make your website responsive and aesthetically pleasing.
- You are **not allowed** to use any Front-End JavaScript Frameworks. For example, you may not use React, Vue, or Angular.

Objectives

Server-Side Validation

You are required to implement server-side validation for both the login and registration form.

- HTML5 validation and client-side JavaScript validation are **not** allowed.
- You must create vanilla JS to perform server-side validation. The use of a Node.js module is **not** allowed.
- For the login form, you are only required to check for nulls or empty values (i.e., check to see if the user entered a value in the respective text fields).
- For the registration form, you must check for nulls (or empty values) **and** implement two complex validation criteria using regular expressions:
    - **Email address validation:** Ensure the email address is not malformed.
    - **Password validation:** Enforce a password that is between 6 to 12 characters and contains at least one lowercase letter, uppercase letter, number and symbol.
- You **must not** clear the data entered in the form if there are validation errors.
- All error messages must be rendered on their respective pages (or areas) and must be styled properly. Here is an example:



User Registration Form

When a user fills out the registration form and clicks on the submit button, provided that all the validation criteria were met, your website must then send a **welcome email message** to the user's email address and then redirect the user to a **welcome page**.

Email Message:

- Must contain a message welcoming the user.
- Must contain the user's first and/or last name.
- Must contain your name and website name.
- Does not need to be styled.

Welcome page:

- Must be configured with its own route.
- Must contain the header, navigation bar, and footer.
- Must contain a message welcoming the user.
- May contain (but does not have to) images, videos, etc.
- Markup must appear in a Handlebars view.
- Have styling applied that matches the other pages of your web app.

Do not forget to protect your SendGrid API key. Create a file to store environment variables and use the **dotenv** module to add the API key to the **process.env** object.

GitHub

You must push your web application to a remote GitHub repository in your own account. **Do not forget to set your remote repository to private.** Add your professor as a collaborator so he/she can view your web application.

Include a *.gitignore* file to prevent pushing the *node_modules* folder and the environment variables file to GitHub.

You must ensure you show at least four reasonable commits. A realistic view of your progress will be determined by looking at your commits.

Heroku

You are required to deploy the working web application to Heroku. See the "Heroku Guide" on the web322 website for help on this topic. Do not forget to provide the URL to your professor.

README.md

You must add a *README.md* file in the root folder of your assignment. Copy and paste the following text in an empty text file called *README.md* and remember to personalize the content. When the repository is accessed on GitHub, this readme file should be displayed by default.

```
# WEB322 Project (Fall 2021)

I declare that this assignment is my own work in accordance with
the Seneca Academic Policy. No part of this assignment has been
copied manually or electronically from any other source
(including web sites) or distributed to other students.

Name:
Student ID:
Course/Section:

## Project URLs

GitHub Repository: https://github.com/<repo_name>/
Heroku URL: https://<app_name>.herokuapp.com/
```

Rubric

Criteria	**Not Implemented (0)** Little or no work done. Unacceptable attempt.	**Partially Implemented (1)** Work is minimally acceptable but is incomplete or needs significant modification.	**Fully Implemented (2)** Work is complete and done perfectly.
Login form validation • Username validation (checking for nulls). • Password validation (checking for nulls). • Error messages are styled and are styled properly.			
Registration form validation • First Name validation (checking for nulls). • Last Name validation (checking for nulls). • Email validation (checking for nulls). • Password validation (checking for nulls). • Email address regular expression validation. • Password regular expression validation. • Error messages are styled and are styled properly.			

Registration Email • Email is sent to the user's email when the user fills out the registration form and hits the submit button. It contains the student's name and website name. • Contains the name of the user that registered on the website. • API key is stored as an environment variable and dotenv is used to read the key.			
Welcome Page • User is redirected to a welcome page. • Contains the name of the user that registered on the website. • Contains the header, navigation bar and footer. Styling matches the rest of the website.			

Cloud Services • A private GitHub repository has been configured and the URL supplied in the readme.md file. • You made at least four 4 reasonable commits (before the due date) and "ignored" the node_modules folder and environment variables file. • Website was successfully deployed to Heroku and the URL supplied in the readme.md file.			

Total: 38 Marks **Note:** Half marks may be awarded.

Submitting your work

Make sure you submit your assignment before the due date and time. It will take a few minutes to package up your project so make sure you give yourself a bit of time to submit the assignment.

1. Locate the folder that holds your solution files.
2. Compress the copied folder into a zip file. **You must use ZIP compression, do not use 7z or other compression algorithms or your assignment will not be marked.**
3. Login to My.Seneca.
4. Open the **Web Programming Tools and Frameworks** course area and click the **Project** link on the left-side navigator. Follow the link for this assignment.
5. Submit/upload your zip file. The page will accept three submissions so you may re-upload the project if you need to make changes. Make sure you make all your changes before the due date. Only the latest submission will be marked.