

Course : (WEB322) -Web Programming Tools and Frameworks	Assignment 2 of 5	Contribution:10% of course
Instructor : Kadeem Best	Date Given: February 16th 2020	Date Due March 4th 2020

Notes for the Student: This Assignment is the second of five that is designed to give you practical experience in hosting Server-Side Web applications using a cloud based solution (Heroku) ,sending emails and applying server side form validation to your web application .

Background: You will need to have access to an IDE or text editor and have a thorough understanding of Node.js, Express,Git ,(GitHub or Bitbucket) and Heroku.

Assignment Submission Requirements

- Submit the link of web application to your GitHub or BitBucket repo
- Submit the link of your hosted Heroku project in Blackboard.
- If your application is not pushed to GitHub or BitBucket, it will not be marked.
- If your web application does not function or work on Heroku, it will not be marked.
- Marks will be deducted for any projects committed to GitHub or BitBucket after the deadline date.

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.

Technical Requirements

- All back-end functionality **MUST** be done using **Node JS and Express**.
- Your views **MUST** be created with **Express-Handlebars**
- You are not allowed to use any **Front-End CSS or JavaScript Frameworks**.

Detailed App Specification

This assignment is a continuation of Assignment 1 thus all the requirements for this assignment is to be made “on top” of your assignment 1.

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

REMINDER:

All back-end functionality **MUST** be done using **Node JS and Express**.
Your views **MUST** be created with **Express-Handlebars**

Features

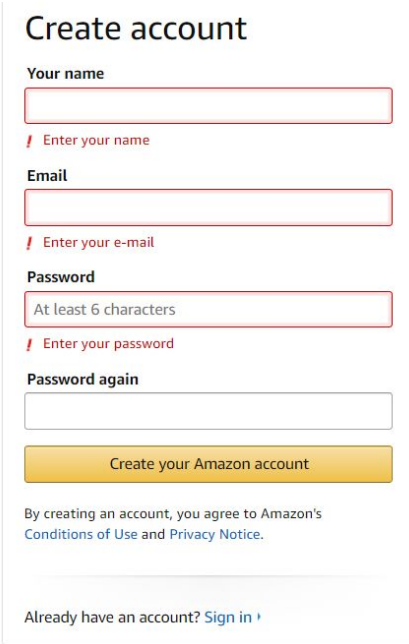
Responsive Design

Ensure that your entire website renders well on a variety of devices, specifically on desktops, tablets and smartphones. To accomplish this you will need to use CSS Media Queries in combination with a modern CSS Layout Module (CSS Grids , or Flexbox or both)

Server Side Validation

- You are required to implement Server-Side validation for both the login and registration form. **NO HTML5 VALIDATION OR CLIENT SIDE JS VALIDATION IS ALLOWED**
- 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) .
- Your form should 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, similarly to See Figure 1.1

Figure 1.1



The image shows a screenshot of the Amazon 'Create account' form. It features four input fields: 'Your name', 'Email', 'Password', and 'Password again'. Each field has a red border and a red error message below it: 'Enter your name', 'Enter your e-mail', 'At least 6 characters', and 'Enter your password'. A yellow 'Create your Amazon account' button is positioned below the fields. At the bottom, there is a link to 'Sign in' for users who already have an account. The form is titled 'Create account' at the top.

User Registration Form (Sending Emails)

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 **send a welcome email message to the user's email address and then redirect the user to a dashboard page**. For now, the dashboard page should contain information welcoming the user and should be properly styled.

Git, GitHub and Heroku

Your web application must be pushed to a remote GitHub or BitBucket repository on your account. Please note that you must set your remote repository to private and then add me as a collaborator so I can view your web application. Lastly, you are required to deploy the working web application to Heroku.

Rubric

Criteria	Not Implemented 0	Partially Implemented 1	Fully Implemented 2
Login form Validation <ul style="list-style-type: none"> Username validation(checking for nulls) Password validation(checking for nulls) Error messages are styled and are styled properly 			
Registration form Validation <ul style="list-style-type: none"> First Name validation (checking for nulls) Last Name validation (checking for nulls) Email validation(checking for nulls) Password validation(checking for nulls) Advance Validation Criteria 1 Advance Validation Criteria 2 Error messages are styled and are styled properly 			
Email <ul style="list-style-type: none"> Email is sent to the user's email when the user fills out the registration form and hits the submit button. User is redirected to a dashboard page 			

<ul style="list-style-type: none"> Dashboard page is styled 			
GitHub <ul style="list-style-type: none"> You made at least 4 reasonable commits Website was deployed to heroku 			
Responsive Design <ul style="list-style-type: none"> Overall site is looks polished on all devices , specifically on smartphones, tablets and large screens 	0	5	10

Total : 40 MARKS

THE END