|  |  |  |
| --- | --- | --- |
| **Course : (WEB322) -Web Programming Tools and Frameworks** | **Assignment 3 of 6** | **Contribution:10% of course** |
| **Prof : Kadeem Best** | **Date Given: October 16th 2019** | **Date Due: November 3rd 2019** |

**Notes for the Student:** This Assignment is three of six and is a continuation of Assignment 1 & 2.

**Background**: You will need to have access to an IDE or good text editor. You will also need a thorough understanding of CSS3, Node JS, Express, Handlebar, SendGrid, MongoDB, Mongoose, Git and Heroku.

**Assignment Submission Requirements**

* An electronic copy of your work (both the web application and a written report) should be uploaded to the Blackboard by midnight on the deadline date.
* **This Assignment WILL NOT BE ACCEPTED VIA EMAIL.**
* The written report must be in pdf format and must contain the following :-
  + A cover page.
  + A table of contents.
  + Screenshots of your website, with relevant descriptions, detailing whether the functionality was met or not met, as specified below.

## **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**

# **Detailed App Specification**

This assignment is a continuation of Assignment 1 & Assignment 2, thus all the requirements for this assignment is to be made “on top” of your assignment 1 & 2 .

Please implement the below features :

# Login Form

You must add a login form to your website. The login form must contain fields that will allow a user to enter their username and password. You can choose how you want this to be rendered and implemented on your website. You can implement it as a modal or within an area on your home page or on an entirely separate page.

**You are not required to implement any login/authentication database functionality.**

# Server Side Validation

You are required to implement **Server-Side validation** for both the login and registration form.

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**(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.**

# 

# User Account Creation

You are required to implement database functionality for your registration page. 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 create a user account in your database.

**Once the user account is created, your website MUST send a welcome email message to the user’s email address and then redirect the user to a dashboard page.** For now, the dashboard should contain static data.

Regarding your database and email functionality, the following rules must be followed :

1. Setup and config a MongoDB cloud service using MongDB Atlas <https://www.mongodb.com/cloud/atlas>.
2. Connect your website/web-app to your mongoDB database using an ODM called **Mongoose.**
3. Name your database and collection appropriately.
4. With respect to the email functionality, do the following:
   1. Set up an account with Sendgrid [**https://sendgrid.com**](https://sendgrid.com)
   2. Install the **nodemailer** and **nodemailer-sendgrid-transport** packages within your website.
   3. Config your website to communicate with the sendgrid service to send email messages.

## Responsive Design

Ensure that your entire website renders well on a variety of devices and window or screen sizes, specifically on desktops, tablets and smart phones.,

## Rubric

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Not Implemented**  **0** | **Partially**  **Implemented**  **1** | **Fully**  **Implemented**  **2** |
| **Login form**   * Website contains a login form with the appropriate labels and widgets. * Username validation(checking for nulls) * Password validation(checking for nulls) * Error messages are styled |  |  |  |
| **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) * Advance Validation Criteria 1 * Advance Validation Criteria 2 * Error messages are styled |  |  |  |
| **Database/Mail**   * MongoDB cloud service is setup (take a screenshot of the dashboard and include in report) * Database and collection have appropriate names * User’s data is inserted into the database when the user fills out then form and hits the submit button. * 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 |  |  |  |
| **Response Design**   * Overall site looks polished on desktop and laptops. * Overall site looks polished on tablets. * Overall site looks polished on smart phone devices   **Heroku (push all new changes )** |  |  |  |

**Total : 40 MARKS THE END**