Course : (WDDM120) -Web Programming Application	Assignment 2 of 4	Contribution:25% of course
Prof : Kadeem Best	Date Given: February 18th 2020	Date Due: March 2nd 2020

Notes for the Student: This Assignment is the second of 4 that is designed to give you practical experience in hosting Server-Side Web applications using a cloud based solution (Heroku) ,sending emails and sms and organizing your code according to the MVC design pattern

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.**
- **This assignment must be presented to me in class. Failure to do so would result in 0 being assigned for your mark towards the assignment. Demonstration day will be on March 3rd 2020 (during your class time)**
- **Please review Humber's policies on Academic Integrity, specifically:**

*"Academic honesty is central to the learning environment enjoyed by all members of The Humber College Institute of Technology and Advanced Learning (hereafter referred to as"Humber `` or"the College") and an expectation of all Humber applicants, students, faculty and staff. A breach of academic honesty is considered to be an offense against the academic integrity of the learning environment. This policy is intended to ensure adherence to Humber's standard of honesty and scholarly integrity in all activities relating to teaching, supervision, research, administrative and consultative work of faculty and staff and to protect the academic integrity of the institution.." **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 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**

Specifically, you are **ONLY** required to implement the following :

Features

Home Page

In addition to all the prior sections, please add a **Featured Room Section**.

This section must display a list of "**Feature Rooms**". Please note, for this Assignment, Assignment 2, the data for this section must be static. However, you are required to define the data in a separate module (Javascript file). For each room, the data should include a picture of the room, title and price.

Application Architecture

1. Your application **MUST** be structured according to the MVC Design pattern. Thus, your views, models and controllers must be separated as per the design pattern requirements.

2. All sensitive credential information **must** be stored in **environment variables**. Examples include: your sendgrid access token, MongoDB connection string, etc. You can implement environment variables locally using the dotenv 3rd party package.

Controllers

Using the Express Router, please break up your routes into multiple modules.

Models

Please be advised that you have the option of creating one separate module file to encapsulate all the static data "fake database data" for the **Feature Rooms Section** and for **the Room listing Page**. This module/JavaScript file is going to represent the Model for your MVC application.

User Registration Form (Sending Emails and Sending SMS)

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 and a welcome sms message to the user's email address and phone number. After the aforementioned is done, 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.