

Course : (WEB322) -Web Programming Tools and Frameworks	Assignment 4,5,& 6	Contribution:25% of course
Prof : Sharmin Ahmed	Date Given: November 11th 2019	Date Due: December 8th 2019

Notes for the Student: This Project corresponds to Assignment 3,4 5,; and, is a continuation of the work you have already completed in Assignments 1-2.

Assignment Submission Requirements

- An electronic copy of your work (both the web application and a written report) should be uploaded to Blackboard by 11:59pm on the date of submission.
- Submissions must be made through Blackboard. **Email submissions will not be accepted.**
- The written report **must** be in pdf format and must contain the following sections:
 - A cover page.
 - A table of contents.
 - Overview of web application-For this section, you are required to explain what your web application does, all the functionality implemented or not implemented and (if any) additional libraries and/or frameworks used.
 - A link to your deployed application (Heroku URL)
 - Screenshots of your website, with relevant descriptions, detailing the functionality.

Assignment Regulations

- This assignment must be done individually.
- **Because this project is due on the last day of class, no late submissions will be accepted.**
- **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 1 and 2, thus all the requirements for this assignment is to be made "on top" of your Assignment 1 and 2.

Assignment 3 (10%)

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.

User Registration Module

You are required to implement database functionality for your registration page that was previously implemented in Assignment 1 and 2 . Thus, 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 web application must then redirect the user to a dashboard page.

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

1. Add a Postgre Database to a new/existing Heroku Application as an Add-on and save the database credentials
2. Connect your web application to your Postgre database using add both the sequelize and the pg / pg-hstore.
3. Name your database and collections appropriately.
4. Ensure that the email field in your registration form **is unique**, thus your application must prohibit different users from having the same email in the database.
5. Passwords **must not** be stored in plain text in the database, thus your application must store passwords in an encrypted format. You can use a 3rd party package called [bcryptjs](#) to do the aforementioned.

Authentication Module

You are required to implement a fully functional authentication module with the following features:

- Your application must allow an administrator or a regular user to **log-in via the login form created in Assignment 3.**
- Upon a successful authentication(entering a username and password pair that exists in the database) **a session must be created to maintain the user state until they logout of the application. To implement sessions** in an Express app you can use <https://github.com/expressjs/session>
- Upon an unsuccessful authentication, the application **must** display an appropriate message (Example: **Sorry, you entered the wrong username and/or password**)

- Also after successfully authenticating, the application must determine if the person logging in is an administrator or a regular user and will be redirected to the appropriate dashboard.
- A regular user will be directed to a user dashboard and an administrator will be directed to an administrator dashboard
- Both dashboards, must show the user's name (first name and lastname) and a logout link
- The logout link must destroy the session created when the user initially authenticated.

Assignment 4 (10%)

Administrator Module

You are required to implement an Administrator Module that will do the following :

- **Create a room:** The administrator must be able to create a room with the following data:
 - room title,
 - price,
 - description/details,
 - location; and,
 - upload a photo of the room.
- **All created rooms must be populated on the front-end of the web application, specifically on the room listing page that was created in Assignment 1, for users to view. Please note, a visitor to the web application does not need to be logged in to view the rooms that were created by the administrator.**
- View a list of all created rooms.
- Edit and change room details for a selected room.

Search Module

Visitors of the web application should be able to search for rooms via a location. This can be easily implemented by having a drop down list of at least 3 locations and a search button. When the user selects a location and clicks the search button, the application must present a list of all the rooms associated with the selected location. Please note, a user does not have to be logged in to search for rooms.

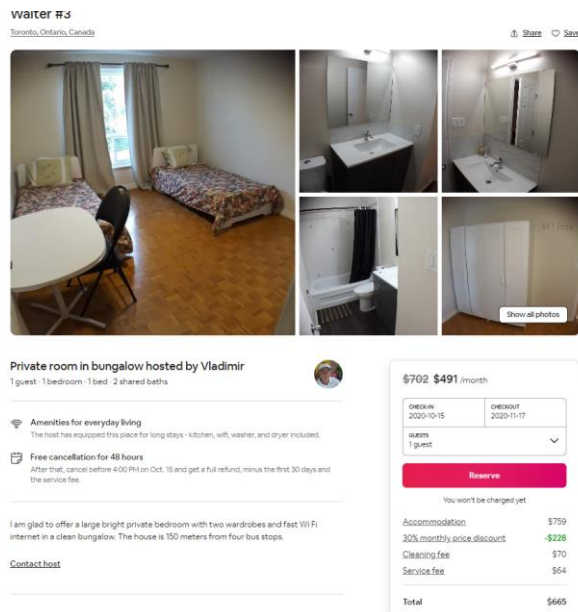
Assignment 5 (10%)

Room Description Page and Book Room Module

Only logged in users should be able to “book” a room.

From the Rooms listing page, when a user clicks on a particular room, they should be navigated to the **Room Description page** of the clicked room and from that page they can select a date range and “book” the room for the selected period.

Sample of a Room Description Page Example :



The Room Description page of any room should list the following:

- Room Image
- Room title
- Room description
- Room price per night
- Start Date
- End Date
- Book Room Button

When the user selects a date range (start date and end date), your application should calculate the number of days the room will be booked for based on the range selected. Then your application should calculate and display the final price the customer must pay (no of days X room price per night)

When the user clicks the “**Book Room**” button, the given product will be booked and the application should email the user indicating the given room that was booked with all relevant details.

THE END