Introduction

This assignment is the fifth of six assignments. In this assignment you will upload image files as part of a multi-part form submission.

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.

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.
- You will use the **express-session** module to handle user session state information.
- You will use **bcrypt.js** to encrypt user passwords.
- 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.
- You **must** use MongoDB as your database engine.

Objectives

Data Loading

Create a throw-away "load data" controller but do not delete the controller; the controller must be submitted so it can be marked by your professor.

The "load data" controller must contain a route that will populate meal kit data. The following are the requirements for the route:

1. A data clerk can load meal kit data to the database by accessing the URL "/load-data/meal-kits".
2. This functionality must be placed in its own controller and must be coded using the MVC design patterns learned in class.
3. You must check to ensure there are no meal kits in the database before adding new meal kits; this is it to prevent adding of duplicate entries.
4. You must populate a minimum of three "top meals" and four "on the menu" meals.
   a. The "on the menu" page should contain at least two different categories.
   b. It is okay for "top meals" to appear on both the home page and the "on the menu" page.
5. After accessing a load data route, display a Handlebars view with a message describing the outcome:
   a. If data was loaded to the database, display a message like "Added meal kits to the database".
   b. If data has already been loaded to the database, display a message like "Meal kits have already been added to the database".
   c. Only data clerks can load data. If the user is not a data clerk, display a message like "You are not authorized to add meal kits".
   d. The message is displayed on a page showing the proper header and navigation bar.

Data Entry Clerk Module

You are required to implement a Data Entry Clerk Module that allows a data entry clerk to perform the following tasks:

1. Create meal kits. The data entry clerk must be able to add new meal kits to the database. At minimum, the following data can be specified when adding a new meal kit:
   a. Title – *Sautéed Ground Pork over Jasmine Rice*
   b. What is included? – *Toasted Peanuts & Quick-Pickled Cucumber Salad*
   c. Description (This is a text area, not an text input) – *Gingery pork, crunchy cucumbers and toasty peanuts make for a classic culinary …*
   d. Category – *Classic Meals*
   e. Price – *$19.99*

   f. Cooking Time – *25 minutes*

   g. Servings – *2*

   h. Calories Per Serving – *890*

   i. Is it a Top Meal? – *true*

   j. Upload a photo of the meal kit *(to keep the project simple, only upload one image per meal kit).*

2. Ensure that all new meal kits that were entered into the database are populated on the front-end of the web application.

   a. All meal kits must appear on the "on-the-menu" page.

   b. Only the meal kits that were set as "Top Meals" should be populated in the "Top Meals" section of the home page.

   c. A visitor does not need to be logged in to view the meal kits that were created by the data entry clerk.

3. Ensure that a user can only upload images stored in an appropriate file type. For example: jpg, gif, or png.

4. View a list of all the meal kits.

5. Edit and change meal kit details for a selected meal kit. For example: price, title, etc.

A data entry clerk must be logged-in to the web application to perform the tasks mentioned above. That is, a data entry clerk must log in to create meal kits, view all meal kits, and change meal kit details.

GitHub

You will continue to 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.

A realistic view of your progress must be showed by looking at your commits.

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.
Data Clerk Module • The data entry clerk can view a list of all meal kits. • A data entry clerk can create meal kits supplying necessary data. • The data entry clerk can edit all meal kit details. • A data entry clerk can upload a photo for a new meal kit or an existing meal kit. • Only images files can be uploaded as a photo. • New meal kits appear on the "on-the-menu" page with the correct category. • Meal kits that are set as top meals are shown on the home page.			

Load Data Module			
• Functionality is stored in a dedicated controller. Only data clerks can load the data by navigating to "/load-data/meal-kits". • Controller will not add duplicate meal kits if run multiple times. • At least three "top meals" and four "on the menu" meal kits are added. There are at least two categories in the "on the menu" page. • Correct message is displayed, and the message is formatted with the header and navigation bar present.			

Total: 22 Marks

Note: Half marks may be awarded.

Submitting your work

Make sure you submit your assignment <u>before the due date and time</u>. 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 <u>three</u> 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.