Introduction

This assignment is the last of six assignments. In this assignment, you will implement a shopping cart experience for your customers.

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

Most of the materials posted in this course are protected by copyright. It is a violation of Canada's Copyright Act and Seneca's Copyright Policy to share, post, and/or upload course material in part or in whole without the permission of the copyright owner. This includes posting materials to third-party file-sharing sites such as assignment-sharing or homework help sites. Course material includes teaching material, assignment questions, tests, and presentations created by faculty, other members of the Seneca community, or other copyright owners.

It is also prohibited to reproduce or post to a third-party commercial website work that is either your own work or the work of someone else, including (but not limited to) assignments, tests, exams, group work projects, etc. This explicit or implied intent to help others may constitute a violation of Seneca's Academic Integrity Policy and potentially involve such violations as cheating, plagiarism, contract cheating, etc.

These prohibitions remain in effect both during a student's enrollment at the college as well as withdrawal or graduation from Seneca.

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.
- You **must** use MongoDB as your database engine.
- 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.

Objectives

Meal Kit Description Page

Logged in users must be able to "purchase" meal kits by adding selected meal kits to their shopping cart.

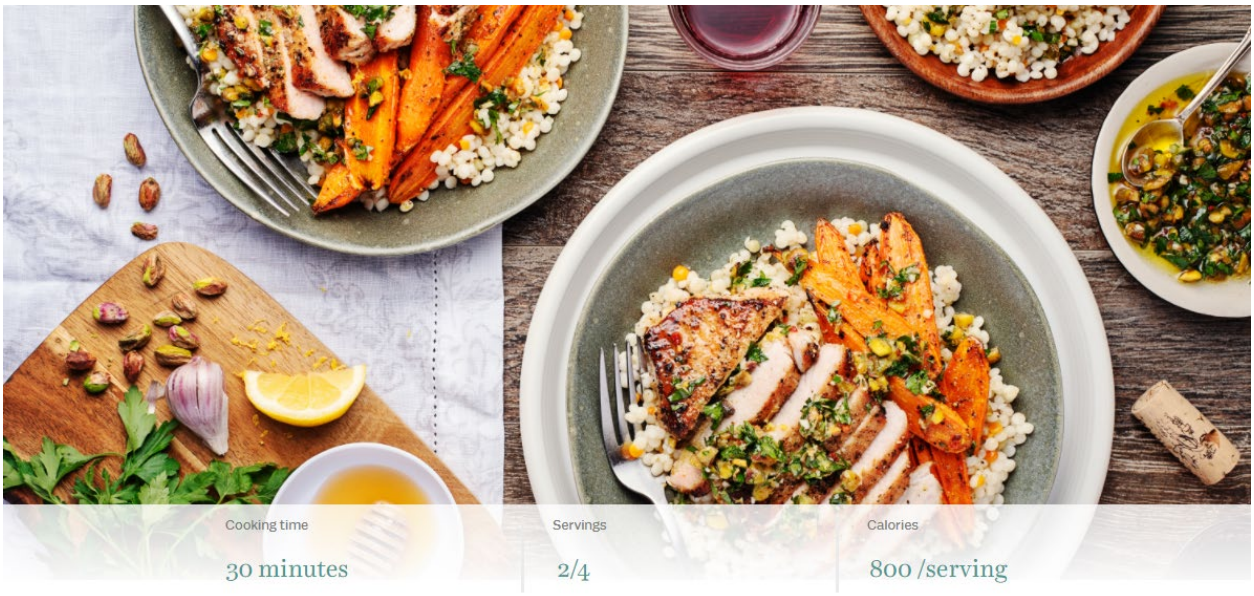
From the "on the menu" page, when a user clicks on a particular meal kit, they should be navigated to the meal kit description page (of the clicked kit). From this description page, users can add the meal package to their shopping cart.

The meal kit description page should include most of the information associated to each meal kit, at minimum:

- Meal kit image
- Title
- Description
- Price
- Cooking time
- Number of servings
- Add to order button

When the user clicks an "Add to Order" button, the kit will be added to the user's shopping cart.

An example of the meal kit description page follows:



Seared Pork Chops with Pistachio-Parsley Gremolata

Honey-Roasted Carrots & Pearl Couscous

We love the challenge of taking a classic meat 'n' veg recipe to the next level. We think we've done just that here, adding a twist while still keeping all the things we love about pork chops and roasted carrots. Exhibit A: tender seared chops, garnished with a savoury and crunchy pistachio, parsley and lemon topping. Exhibit B: vibrant carrots roasted to perfection and tossed with a lip-smacking combo of garlic and honey. Exhibit C: plump pearl couscous infused with hints of citrus and fresh herbs. Assembled all together on one plate: pure joy!

Buy for $19.99

Note: You do not have calories per serving in your database.

Shopping Cart Module

The customer's dashboard should have a link to the logged-in user's shopping cart. This page must display all the meal kits added to the shopping cart, the order total, and a "Place your order" button. Each meal kit should display the name, quantity, price, and extended price. The extended price equal to the quantity multiplied by the price. There must be a way to change the quantity.

When the "Place your order" button is pressed, your web application will send an email to the logged-in user's email, showing all the packages purchased along with the quantity for each. The email must include the customer's order total but do not worry about taxes.

After the email is sent, clean *(zero-out)* the shopping cart.

GitHub

You will continue to push your web application to a remote GitHub repository in your own account. Continue to use the same repository you set up in assignment 3.

If you haven't already done so, **set your remote repository to private.** Remember to 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.

Cyclic

You are required to deploy the working web application to Cyclic. See the "Cyclic Guide" on the web322 website for help on this topic.

You can continue to use (and overwrite) the Cyclic app you created in assignment 3. Do not forget to provide the URL to your professor (in the README.md file).

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.
Meal Kit Details • Clicking a meal kit will show the details of the meal kit with all necessary fields. There is an "Add to Order" button.			
Ordering Module • Only logged in customers can add meal kits to their shopping cart. • A shopping cart displays the purchased meal kits. The name, price, quantity, and extended price of each meal kit is displayed, along with the shopping cart total. • When the customer clicks the "Place Order" button, the application "clears" the shopping cart and sends an email with the order information.			
Cloud Services • Website is deployed to Cyclic and works correctly.			

Criteria	**Not Implemented (0)** Little or no work done. Unacceptable attempt.	**Partially Implemented (5)** Work is minimally acceptable but is incomplete or needs significant modification.	**Fully Implemented (10)** Work is complete and done perfectly.
Overall Behaviour and Look • Overall site looks polished on all devices, follows the specifications as outlined in all assignments, and works well.			

Total: 20 Marks

Note: Half marks may be awarded.

Submitting your work

Make sure you submit your assignment before the due date and time. 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. You may choose to delete the node_modules folder but do not delete any other files or folders.

2. Compress the copied folder into a zip file. **You must use ZIP compression, do not use 7z, RAR, or other compression algorithms or your assignment will not be marked.**

3. Login to My.Seneca, open the **Web Programming Tools and Frameworks** course area, then click the **Project** link on the left-side navigator. Follow the link for this assignment.

4. Submit/upload your zip file. The page will accept unlimited 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.