

Course: WEB322 - Web Programming Tools and Frameworks	Assignment 1 of 5	Contribution: 10% of final grade
Instructor: Wei Song	Date Given: September 22	Date Due: October 9

Notes for the Student: This Assignment is one of five that is designed to give you practical experience in building Dynamic websites and Server-Side Web applications using Node.js and Express.

Background: You will need to have access to a code editor. You will also need a thorough understanding of HTML 5, CSS3, Javascript, "Node.JS", Express and Handlebars.

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**
- **You are not allowed to use any Front-End CSS or JavaScript Frameworks.**

Detailed App Specification

Live Fit Food (<https://livefitfood.ca/>) is a popular Canadian meal delivery company that prepares, cooks and delivers healthy meals straight to your door. You have been contracted as a Full Stack developer to develop your version of the Live Fit Food's website.

Your website does not have to look the same as theirs but you should rather use their website (or similar meal delivery websites) to inspire the look and feel for your implementation.

The implementation of the entire functionality for the website will be spanned over 5 Assignments.

In this assignment, you will **ONLY** focus on building views for your website/web application. **No database connectivity is required for Assignment 1.**

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

Express web server set up

Create an Express web server that listens to incoming HTTP requests.

Route Handlers Implementation

Ensure that you create route handlers that will direct users to specific views when they navigate to the following routes:

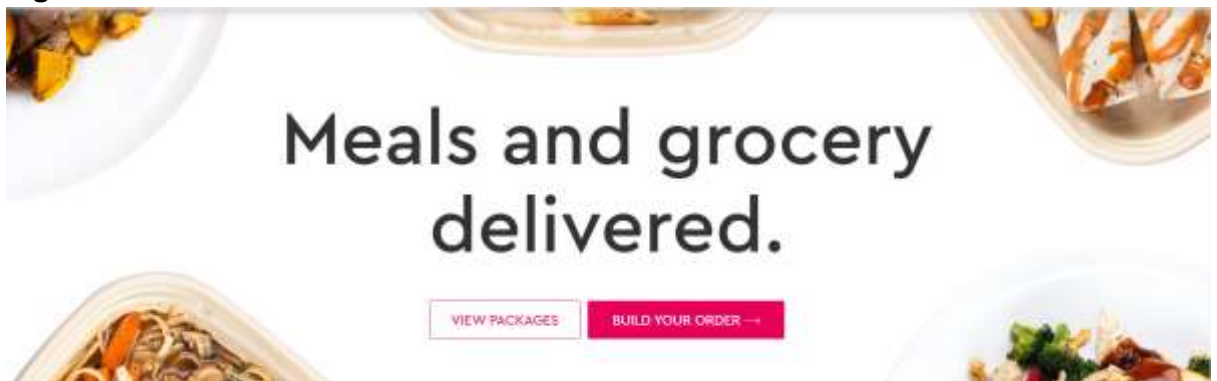
- Home
- Meals Package Page
- Customer Registration
- Login

Home Page

You are required to build a **well-designed** home page that consists of the following sections:

- **Header** - The header **MUST** contain the logo of your website and could contain the navigation bar and any other content you deem necessary.
- **A Navigation bar** - The navigation bar can be within the header or defined as an entirely new area. It **MUST** have links that navigate visitors to a sign-up page and the login page. Please note, **instead of creating a separate page for the sign-up page and the login page, you can render it as modal pop-up boxes.**
- **A Hero Section**- The section must have a prominent image or video element that is placed at the top of your pages, below the header and navigation section. Example:

Figure 1.1



- **Content Section(s)** - The sections **MUST** use a combination of grids, words, headings, and images with the sole purpose of highlighting some selling features of the website and the stellar service that they provide. See the below example:

Figure 1.2



- **Top Meal Packages** - This section must display at least four (4) top meal packages, as shown in figure 1.3. Please note, for this Assignment, Assignment 1, the data for this section must be static, i.e. it will not be pulled from a database. However, you are required to define the data in a separate module (JavaScript file). Each meal package must display an image, title, price.

Figure 1.3

Our chefs are making these right now!

You could be chowing down on any of these meals any time you choose – seriously!
Order some (or all) of these meals and see why thousands of Canadians stock us in their fridge!

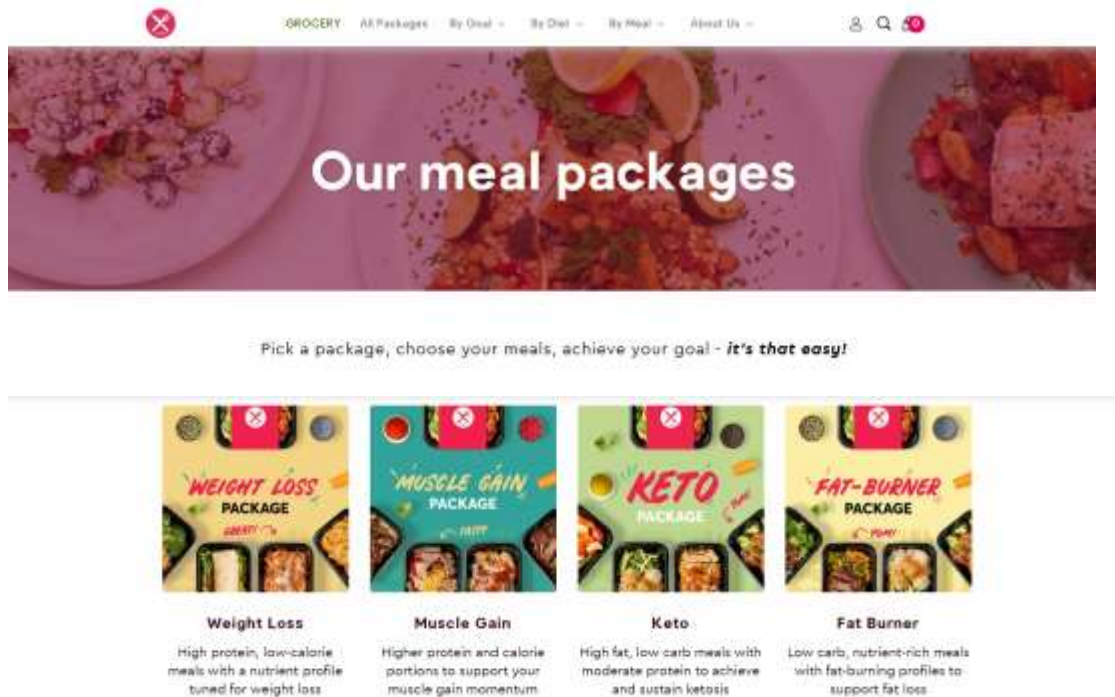


- Footer - This section **must** include footer menu items, social media links, and any other information you deem necessary.

Meal Package Listing page

You are required to build a **well-designed** meal package listing page. This page must have at least **4 sample meal packages within a 4 column grid layout**, as shown in the below image:

Figure 1.4



Note, every meal package must have an image, title, price, food category, number of meals in the package, synopsis of the content within the meals and a boolean attribute indicating if it is a top package or not. This view must also have a header, navigation, hero section, and footer like the home page.

Like the Top Meal Package section, for this Assignment, Assignment 1, the data for this section must be static, i.e. it will not be pulled from a database. However, you are required to define the data in a separate module (JavaScript file).

Please be advised, you must create one separate module file to encapsulate all the static data “fake database data” for the Meal Package Listing page, and for the Top listing. This module/JavaScript file is going to represent the Model for your MVC application.

Registration Page

You are required to build a **well-designed** user registration form as shown in **Figure1.5** . This can be implemented as an entirely new page or as a pop-up modal. In the event you implement this as a new page then you must ensure that the page maintains consistency regarding its structure, header, nav, hero section, footer, and etc as the home page.

Figure 1.5



Create Account

f G t in

or

First Name

Last Name

Email

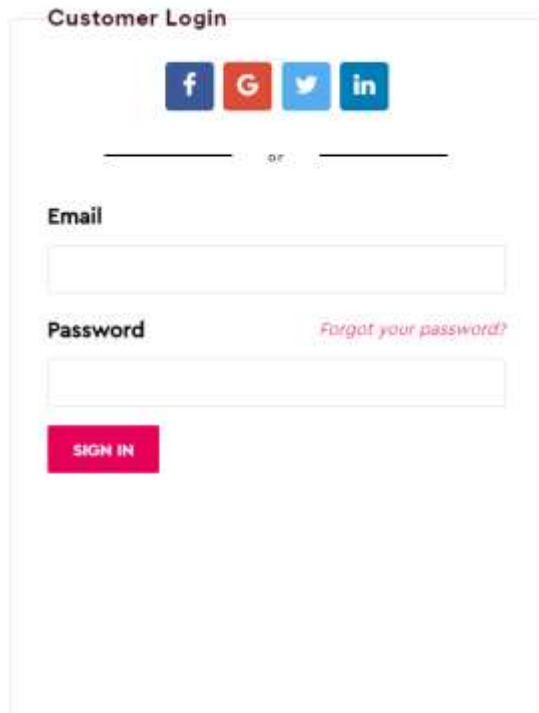
Password

SIGN UP

Login Page

You are required to build a **well-designed** login form as shown in **Figure1.6** . This can be implemented as an entirely new page or as a pop-up modal. In the event you implement this as a new page then you must ensure that the page maintains consistency regarding its structure, header, nav, footer, and etc as the home page.

Figure1.6



The image shows a 'Customer Login' form. At the top, there are four social media icons: Facebook (f), Google (G), Twitter (bird), and LinkedIn (in). Below these icons is a horizontal line with the word 'or' in the center. Underneath the line, there are two input fields: 'Email' and 'Password'. To the right of the 'Password' field, there is a link that says 'Forgot your password?'. Below the 'Password' field is a red button with the text 'SIGN IN' in white capital letters.

Reminder: All back-end functionality **MUST** be done using **Node JS and Express**. Your views **MUST** be created with **Express-Handlebars**. You must create one **Model (JavaScript file)** to encapsulate the static data (“fake database data”) for the **Meal Package Listing Page**, and the **Top Meal Package section** located on the home page.

Rubric

Criteria	Not Implemented 0	Partially Implemented 1	Fully Implemented 2
Home View <ul style="list-style-type: none">• Header• Navigation• Hero Section• Top Meal Packages Section• Footer			

Meal Product Page <ul style="list-style-type: none"> Meal Packages Listing in a grid Each meal package lists the required data on the page as per the requirements. 			
Registration /Login View or Popup <ul style="list-style-type: none"> Registration form Login form 			
Models <ul style="list-style-type: none"> The static data for the Top meal package section. Ensure it is encapsulated into a separate module (JavaScript file) The static data for the Meal Package listing. Ensure the data is encapsulated within the previously created module 			
Overall look and feel <ul style="list-style-type: none"> Overall site is polished Used pleasing typography, color palettes, and imagery Color and layout are cohesive throughout the site Appropriate use of grids 	Poor (1)	Average (3)	Exceeds (6)

Total: 28 MARKS

THE END