Due Date:

## 5/1/2021 @11:59PM

Assignment Details

* The assignment involves creating a very simple Pizza ordering web app with two pages at minimum
* The purpose of this assignment is to familiarize you with Node Project, Node modules, ExpressJS framework and a template engine to perform server-side rendering
* Assignments must be done individually and there should be no sharing of work amongst students. If you are found in violation of this, you will receive a “0” for the assignment and will be reported to BCIT
* No sample solutions will be provided after the assignment. The second assignment makes use of some of the artifacts created in this assignment.
* Your assignment **must meet all the criteria specified** to receive full marks; however, students are encouraged to go beyond the assignment requirements. But first ensure you meet the basic requirements specified here to meet the marking criteria.

# To Hand In:

* **Before you begin your assignment create folder with your lastname\_firstname. Inside this folder create a node project. Failing to do so makes it difficult for the instructor to identify whose assignment is being graded. So, it will not be graded**
* Once you are done with your assignment, zip up your lastname\_firstname folder. Upload the zip file to the drop box folder for the Assignment 1
* Include a read me file explaining any required setup that the instructor should be aware to run your app. If you have made any assumptions (like running on a different port than 3000) and do not mention in your readme file, you may lose points
* **Do not wait till the last moment to submit your assignment as there is a possibility that D2L’s dropbox does not respond on time. Upload a version of your assignment an hour or so before the deadline and then continue to work on your assignment to upload a newer version closer to the deadline**
* When submitting the assignment to the D2L’s drop box, please leave a comment stating how many hours you spent working on this assignment
* **Assignment cannot be accepted by email submissions. Do not email zip files to the instructor.**

# Requirements:

The web app has multiple pages that perform server-side rendering with a templating engine. The main/first page that allows the user to place an order and a second page is shown upon placing an order that displays the order details and cost of the order. The second page allows the user to confirm or cancel the order. Make use of static middleware to support a CSS file and any images files you want for your web app.

## Main Page/Order Page (15)

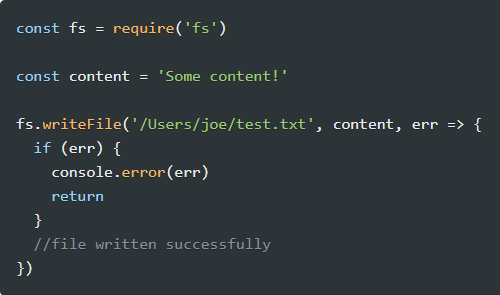
The order page displays the information about the size of the pizza, type of crust and available toppings. Use appropriate HTML controls like radio buttons and check box lists to display the information. Create a custom CSS stylesheet to style the form elements; the stylesheet is served using static middleware. The form also captures quantity, phone number and address information of the customer. When the submit button is clicked on this page it posts the data to the order confirmation route.

* The data for the size, crust type, cost details and available toppings should not be hardcoded but instead read from a JSON configuration file that is part of the project.
* Should make use of template’s looping construct display repeatable elements like toppings. Adding a new topping, for example, should just require changing the configuration file.
* Perform any data validation required – entering invalid data should not show the stack trace or crash the server.
* Make sure you use “placeholder” attribute in HTML text input elements to inform the user that the data is expected in a certain format (Eg: for phone number input, if you are expecting dashes between the numbers, then display 604-XXX-XXXX with a placeholder or if not just 604XXXXXXX in the placeholder).
* Explore NPM JS for a validation module and make use of it rather than hand coding the validation.

## Order Confirmation Page(s) (15)

The order confirmation page displays the details of the order along with the total cost of the order. The cost of the order is determined using the price calculator module described below. This page also displays two buttons – one to cancel the order and the other to confirm the order.

Cancelling the order simply returns to the previous page. Confirming the order saves the order in a JSON file on the server, and on browser displays an order confirmation message, approximate time of delivery for the pizza (to keep it simple just add a half-hour to the current time and display that as the delivery time). Save each order’s data as a JSON file to a folder in your website when the order is confirmed. Shown below is the sample code that uses fs module in Node to write the data to a file.



## Price Calculator Module (10)

Create a PriceCalculator module that computes the price (along with tax) of the order based on the size and toppings chosen**.** The data for the calculation should come from your JSON file. All the code related to the calculation of the price for the order should be contained in this module.

## JSON Configuration File

Store all the configuration data – toppings, crust etc and the corresponding prices in a JSON configuration file and use that instead of hard coding these details. You are free to come up with your own format for the JSON file.