| **Course: WEB422** | **ShopCart Project**  **1 of 6** | | **Contribution:**  **10%** |
| --- | --- | --- | --- |
| **Instructor: Kadeem Best** | **Date Given: Sept 17th, 2021 Date Due: October 10th, 2021** | | |

**Notes for the Student:** This Assignment is designed to give you practical experience in building Restful APIs using Node.js and Express.js

**Background**: You will need to have access to a code editor. You will also need a thorough understanding of JavaScript, OOP, API design, REST, Node.js & Expres.js

# Assignment Submission Requirements

* Your source code and GitHub link must be uploaded to Blackboard
* Ensure that your GitHub is private. Add me as a collaborator (see course webpage for Github username) to your remote repo
* Ensure that your Github repo has a detailed **Readme.md** file that details your application
* A link to your Heroku instance where your API was deployed

# Assignment Regulations

* This assignment must be done individually.
* A virtual “in-person” demonstration of this project is required. **The date of the presentation would be during the week of October 10th, 2021**.
* **Failure to demo would result in 0.**
* Failure to answer questions regarding foundational concepts about your project and how the said concepts were used within your code would result in 0.
* **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

* Your RESTful API must be created using Node.js & Express.js.

# Detailed App Specification

Shopmart (this is a fictional company), is one of the largest retailers in Canada with over 200 stores nationwide. Presently, they don’t have an Internet presence. However, because of all the stringent social distancing measures implemented globally, they have decided to create an online store, showcasing their wide variety of products, ranging from appliances to video games.

You have been “contracted” as a Back-End JavaScript Developer to develop a Back-End, in the form of a RESTful API, for Shopmart. The API must essentially allow administrators to manage their inventory of products, produce a catalog of their products and manage their customers.

This Assignment is the first of six (6) and will solely focus on the API functionality for the overarching online store. The other functionality will be spanned over the remaining 5 Assignments.

## Framework

Your RESTful API **MUST** be built using Express.js.

## Database

Your Restful API **MUST** be connected to a MongoDB database.

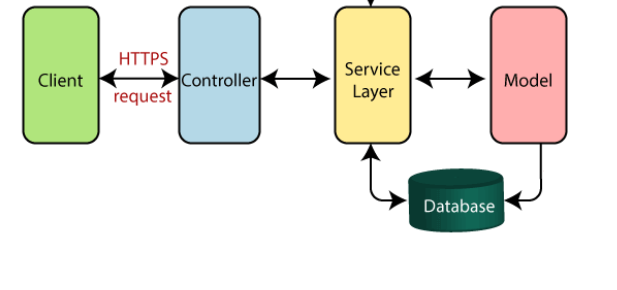
Regarding your database functionality, the following rules must be followed :

1. Setup and configure a MongoDB cloud service using MongoDB Atlas <https://www.mongodb.com/cloud/atlas>.
2. Connect your REST API to the database using an ODM called **Mongoose.**
3. Name your database and collections appropriately. Marks will be deducted if you do not.

.

## Application Architecture

Your RESTful API **MUST** be built in accordance with the MVC (with service layer) design pattern. Thus, you must create Controller files, Service files, and Model files.

* 

## Endpoints

Test all the below endpoints using Postman or any other API Client to ensure that each endpoint is functional and works.

Your API must be designed as follows. Marks will be deducted if your API does not.

1. All communication to and from the API must occur using JSON format.
2. The API expects to receive any request payload data as JSON
3. The API will always respond to a client request with JSON
4. When the API responds to client requests, the API must always return an appropriate status code in its header. You must use the codes per the discussion in class.
5. The name of the endpoints MUST adhere to the conventions discussed in class.

# 

### **None Existing Endpoints**

Requests made to any endpoint that is **not** described in the below documentation should respond with appropriate error code/status message as discussed in class.

### **Customer endpoints**

1. Create an endpoint that will allow a user to register. The below is the data to be added when a customer is created. **Asterisks indicate the fields that are mandatory**
   1. First name (\*),
   2. Last Name(\*),
   3. Email (\*),
   4. password(\*)
   5. Phone numbers - *this should be modeled as an array*

Note, passwords must not be stored in plain text in the database, thus your application must store passwords in an encrypted format. **You can use the Bcrypt Library or any encryption library of your choice.** <https://www.npmjs.com/package/bcryptjs>

This endpoint should also provide validation logic, specifically for requests that do not contain mandatory data.

1. Create an endpoint that retrieves a specific customer by id. This endpoint should also provide validation logic, specifically for requests that do not contain mandatory data.

### **Products endpoints**

1. Create an endpoint that will create products to be added to the database. The below is the data to be added when a product is created. **Asterisks indicate the fields that are mandatory**
   1. Product Name (\*),
   2. Product price (\*),
   3. Product description or details,
   4. Product category (\*)
   5. Product quantity
   6. A value that will indicate if the product is a bestseller product (or not) (\*)
   7. Product Photo URL. - For this, you should use the URL for publicly available photos. For example, below is the URL address for a stock photo of a couch :

<https://images.unsplash.com/photo-1549187774-b4e9b0445b41?ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&ixlib=rb-1.2.1&auto=format&fit=crop&w=800&q=80> You are allowed to use any photos from the Internet.

*Before using a photo, you are responsible for ensuring that the website will permit you to use said photo(s) in an academic context. Generally speaking, photos marked as Royalty-Free or Freely-Useable will prevent you from running into intellectual property and copyright issues.*

This endpoint should also provide validation logic, specifically for requests that do not contain mandatory data.

1. Create an endpoint that retrieves all the products in the database.
2. Create an endpoint that retrieves all the product categories in the database.
3. Create an endpoint that retrieves all products that belong to a specified product category.
4. Create an endpoint that will retrieve all products marked as bestsellers.
5. Create an endpoint that will retrieve a specific product by id. This endpoint should also provide validation logic, specifically for requests that do not contain a valid product id.
6. Create an endpoint that updates an existing product by id. The client should be allowed to update ANY field, except the product id**.** This endpoint should also provide validation logic, specifically for requests that do not contain a valid product id.
7. Create an endpoint that will delete an existing product by id. This endpoint should also provide validation logic, specifically for requests that do not contain a valid product id.

# Rubric

| **Criteria** | **Not Implemented** | **Fully**  **Implemented** |
| --- | --- | --- |
| **Application Architecture**   * Built-in accordance with MVC | 0 | 4 |
| **Deployment**   * Deployed to Heroku * Used environment variables (where applicable) * Pushed to Github * GitHub contains a well-detailed Readme.md | 0  0  0  0 | 4  2  2  2 |
| **Customer End Points**   * Create A Customer * Encrypting Customer’s Password * Validation for (Creating Customer) * Retrieve A Customer * Retrieve A Customer (Validation) | 0  0  0  0  0  0  0 | 3  2  2  2  3  2 |
| **Product End Point**   * Create A Product * Validation (Create A Product) * Retrieve All Products * Retrieve All Product Categories * Retrieve Products that Belong To A Specific Category * Retrieve Products That Are Marked As Bestsellers * Retrieve A Specific Product * Validation * Update A Specific Product Validation * Delete A Specific Product * Validation | 0  0  0  0  0  0  0  0  0  0  0  0  0  0 | 4  2  2  3  3  3  2  2  2  2  2  2 |
| **Non Existing Endpoints** | 0 | 3 |

Total Marks: 60 marks

# THE END