CMPS 350 Web Development Fundamentals Spring 2022

Final Lab Exam

Student Name	
Student Id	
Email	

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1) The exam duration is **120** minutes. So, read the exam questions carefully and plan your time accordingly.
2) Push your code to GitHub regularly to avoid unpleasant surprises, as your computer might hang or shutdown ☹.
3) The Exam is an open book; however, in case of plagiarism, both parties will receive **- 0 points**. Hence do not share or receive any code from anyone.
4) Once you complete the Exam,
   a. Demo your work before leaving the Exam.
   b. You should add the screenshots of the output to the testing sheet provided to you
   c. You should push your code and testing sheet to your GitHub repo under the Final Exam

GOOD LUCK ON YOUR EXAMINATION!

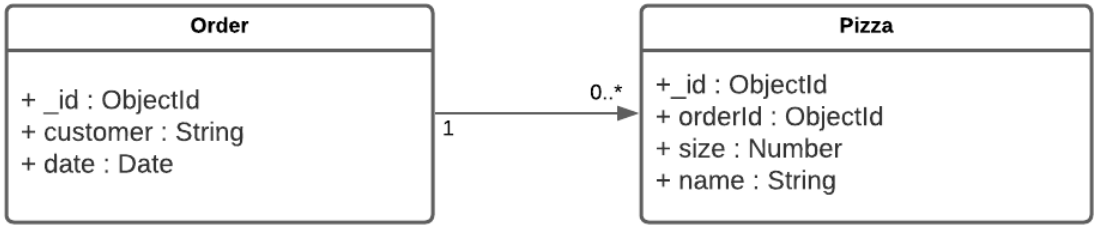
Preparation

1. Sync cmps350-lab repo to get the **pizzaApp** inside the Final Exam folder
2. The project has the **package.json** file, so do not create a new one.
3. Run **npm install** to get all the dependencies needed to complete the Exam. You should not install any extra packages or remove the package.json file.
4. Make sure you run the Mongo DB server [mongod / mongo] on your terminal.

1. Creating the Server and Pizza APIs [80%]

1.1. **Creating the models** : **[10%]**

1.1.1. You should create two models named [**Order** and **Pizza**] with the following properties.



1.2.1 Make sure you write proper validation for all the fields. All the fields are required. The possible pizza names are ['Margherita', 'Marinara' ,'Quattro', 'Napoletana']. Also, the pizza's size has a minimum value of 1 and a maximum value of 3.

1.2. **Creating the repository**: **[40%]** Create a class named OrderRepo under **repository/order-repo.js** file and implement the following methods

Method	Description
getOrder()	Return all orders inside the orders collection
addOrder(order)	Adds new order to the database orders collection
deleteOrder(orderId)	Removes the matching order from the orders collection
getOrderPizzas(orderId)	Returns all the Pizza that are related to the given order. You should open the pizza collection and filter all the pizzas with the given order Id
addPizza(pizza)	Adds new pizza to the database pizzas collection
updatePizza(updatedPizza)	Updates specific pizza
deletePizza(pizzaId)	Deletes specific pizza from the pizzas collection
getPizza(pizzaId)	Return a single pizza that has the same pizza id

1.3. **Create the services**: **[20%]** Create a **service-service.js** file and implement **OrderService** class with methods to handle the requests listed in the table 1 below. The handlers should use the provided methods from the OrderRepo class.

1.4. **Create the router** : **[10%]** Create a **router.js** file and implement handlers for the routes listed in the table below. The route handlers should use the methods from the **OrderService** class you implemented above.

HTPP Verb	URL	Functionality
GET	/api/orders	Return all orders
POST	/api/orders	Adds new order
DELETE	/api/orders/:orderId	Deletes specific order from the orders collection that has the same order Id
GET	/api/orders/:orderId/pizzas	Gets all the pizzas that have the given order Id
POST	/api/orders/:orderId/pizzas	Adds new pizza to the pizzas collection
PUT	/api/orders/:orderId/pizzas	Updates specific pizza
GET	/api/orders/pizzas/:pizzaId	Return a single pizza that has the same pizza id
DELETE	/api/orders/pizzas/:pizzaId	Deletes specific pizza from the pizzas collection

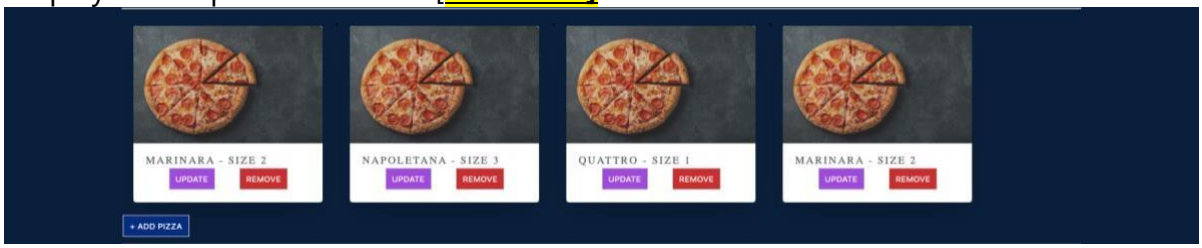
1.5. **Create the Server:** Finally, create **app.js** **[5]** file, and import express package, instantiate an express app, configure it, then start it at port 9090.

Important: Test your API's using Postman before moving to the next Step

2. Connecting the Client App to the Server

Your task is to create the client interface for the **pizzaApp** using handlebars. The application demo is shown below.

Display all the pizzas as cards [**15 Points]**

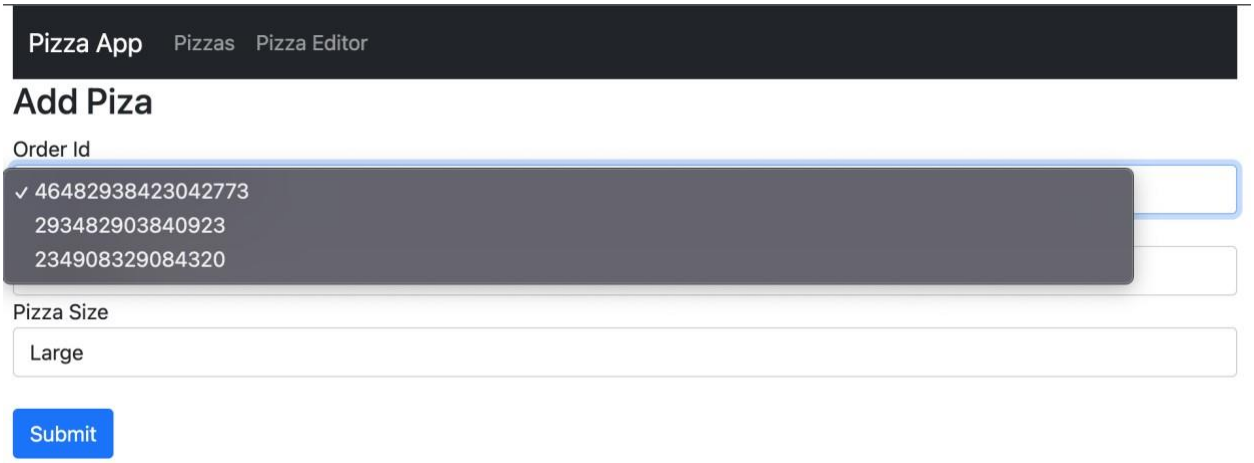


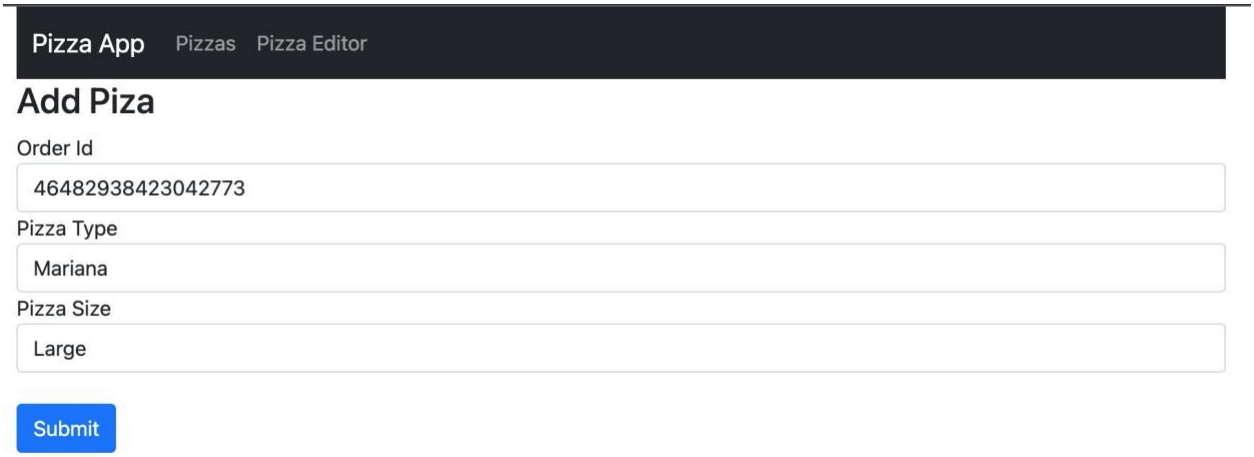
Bonus 10 Points

Implement the following two features

Add Pizza

Note: All of the three inputs are dropdown





Delete Pizza