Project

In this assignment, you will create a web server with endpoints that execute server side logic.

Coding Guidelines

- When creating your solution, you must use the coding practices and conventions demonstrated in class. A solution that does not reflect what was taught in class will not be accepted (0 grade) and/or be subject to an academic integrity review.

- Javascript syntax rules:
    - Variables must be declared using let/const, not var
    - Functions must be declared using arrow function syntax, not function() syntax
    - When checking equality, use strict equality (triple equals ===), **not** double equals (==)
    - Do **NOT** use higher order array functions: forEach, map, reduce, filter, closest, etc.

Submission Requirements:

- ❏ Create a zip file containing your entire project folder. Name your file PROJECT_Gxx.zip, where you replace xx with your group number. Example: PROJECT_G04.zip

- ❏ A screen recording as described in the instructions. If your screen recording is too large for the submission dropbox, then you should upload the screen recording to Microsoft OneDrive and share the link in the submission comments. Ensure you update the link sharing settings so that it is accessible by anyone in the college with the link.

Academic Integrity

- You are responsible for familiarizing yourself with the college's Academic Integrity Policy.

- This is an individual assessment

- Situations which often cause academic integrity issues:

    - Reposting any part of the assessment to online forums or homework help websites
    - Contract plagiarism: Purchasing a solution, or completing a solution for financial compensation
    - Sharing or receiving source code, references, or assistance from others

Problem Description:

Create a server-side application for a restaurant to accept, fulfill, and deliver customer orders.

Specifically, the application must consist of a:
- Restaurant website: Customers use this website to view menu items, order items, and view their order history.
- Order processing website: The restaurant owners use this website to view incoming orders, update the status of an order, and assign an order to a delivery driver.
- Delivery driver website: Delivery drivers working for the restaurant use this website to manage delivering the order to the customer.

You must create both the backend and frontend for this website. As you work in your group, please ensure that each group member has opportunities to demonstrate their skills in both **server side** and **user interface**.

- Server side skills: database programming, endpoint logic, etc
- User interface: handlebars, html, css, etc

Database:

Database must be implemented using **Mongo Atlas**.
At minimum, your database must consist of the following collections:

orders_collection
- Each order must have a customer name, delivery address, items ordered, date/time of order, and status
- Order statuses include:
  - RECEIVED: this means the order was received by the restaurant, but not prepared yet.
  - READY FOR DELIVERY: the restaurant has finished preparing the order and is waiting for the delivery driver to pickup the order.
  - IN TRANSIT: The delivery driver has picked up the order and is delivering it to the customer
  - DELIVERED: The order is delivered to the customer.

drivers_collection
- Each driver has a username, password, full name, vehicle model, color, and license plate.

menu items_collection:
- A collection of items sold by the restaurant.
- Each menu item has a name, image, description, and price.

If needed, you may add additional collections or properties to your database.

User Interface Guidelines

All user interfaces must be created with Express Handlebars.

UI must be reasonably polished, with pleasing typography, fonts, colors, images, and layouts.

3rd party icon libraries are permitted, but CSS styling frameworks like Bootstrap are NOT allowed.

1. Restaurant Website:

Create a website that customers use to view information about the restaurant and order items. Y
- At minimum, the website must show:
  - Name of restaurant
  - Main / banner section showing a featured menu item
  - Restaurant location and hours
  - Menu with items and pricing.
  - Order status form
    → Used by customers to check the status of their order
  - Order form
    → Used by customers to place an order with the restaurant
    → You may assume that customers will only ever order ONE of each menu item. (assume quantity = 1)

Menu:
- At minimum, your restaurant must have 4 menu items
- Menu items must be stored in your database
- Website must programmatically display the menu items based on the database. For example, if the price of a menu item in the database changes, then the website should show the updated price.

Order form:
- Form is used by customers to place an order with the restaurant
- Customer will only ever order maximum ONE of each menu item (quantity = 1)
- After an order is placed, the application must provide a receipt and an **order confirmation number**.

Order status form:
- Form is used by customers to check the status of the order
- Customer check status by entering their order id into a form
- If order id can be located, display its current status
- If no order is found, display an error message.

2. Ordering Processing Website

Create a website that is used by the restaurant to manage orders, update order status, and assign an order to a delivery driver.

The website must provide user interface/logic for:
- History of orders
- Current orders

List of orders
- Displays a list of all orders received by the restaurant, sorted by date/time.
    - The most recent orders should be shown first.
    - Each item in the list should show:
        - Customer name
        - Customer delivery address
        - Date of order
        - Number of items ordered
        - Order total
        - Date of order
        - Driver name and license plate.
        - If the order was delivered, provide a way to view the photo of delivery.
- The user can search for all orders from a specific customer.
- For each order, provide a way for the user to update the status of the order

3. Driver Delivery website

Create a website that allows users to register as a delivery driver for the restaurant.
- Provide appropriate user account and login mechanisms.

After registration / login, drivers can view:
- A list of open deliveries
- Delivery fulfillment page

List of orders
- Displays a list of orders available to be delivered (status = READY FOR DELIVERY)
- Driver can select an order to deliver. Once a driver has selected an order, the order status should update to IN TRANSIT.
    - The order should be removed from the list of orders available
    - Other logged in drivers will no longer be able to select the order.
    - Changes to the order must be reflected on the Order Processing website.

Delivery fulfillment page
- When the order is delivered, the driver must update the order status to DELIVERED and take a photo as proof of order delivery.
- Provide a page for the drive to complete these actions.
    - Changes to the order must be reflected on the Order Processing website.

END OF ASSESSMENT