Assignment 4

Caleb Corrigeux, Michael Pham, Tarun Appannagari

1. Project Link: <https://github.com/Corrigeux/Software-Design-4353-Group-17/tree/main/FuelProject>
2. For this project we used MongoDB and its platform MongoDB Atlas, which uses an the API mongoose for managing NoSQL data. Our model is based on a single document which holds the user information including UserCredentials, ClientInformation, and FuelQuote requests. The User MongoDB model has a username of type String, email of type String, encrypted password of type String, userdetails of type Object holding client information including: Full Name, Address 1, Address 2, City, State, and Zipcode. The fuelquotes item is an array which holds objects, these objects contain information of the fuel quote requests such as gallons, Delivery Address, Delivery Date, Suggested Price, and Total Amount which we did not do because it is for the next assignment. We also made sure the information was properly saved with the character lengths of the fields. MongoDB Atlas uses a Web-App to create the Cluster, which we can connect to and create documents there. We just connect to the cluster and update our content using the MongoDB Library Mongoose. Here is some of the code we used:
   1. findOneAndUpdate() – update user content based on given id
   2. mongoose.connect() – used for connecting to MongoDB
   3. findOne() – used for finding user content based on given id or email
   4. .save() – for saving data to the user document
3. For our unit testing, we mostly focused on making sure our functions such as checking for information validation is covered. Most of the validation is done within the API calls when we pass information to our POST routes. For these information validation functions we worked on tests to see whether they are covered. The App is a little hard to test because we are already doing validations when the user submits information to the server, but it we are doing unit tests using JestJS for our functions too.
4. Tarun ensured that the Backend retrieved data from the DB and that it displayed it to the front end. Mike and Caleb ensured that the Backend was receiving data from front end, validating, and persisting to DB.

As far as the code, we all used visual studio live share which enabled us to collaborate/code together and to help each other out.