Assignments & Project

Assignment 1

In this assignment we will come up with initial design for a software application that you will build in this semester. **We will not be writing any code in this assignment,** but only looking at some initial design ideas and high level architecture.

Description:
A partner of your company has requested to build a software application that will predict the rate of the fuel based on the following criteria:
- Client Location (in-state or out-of-state)
- Client history (existing customer with previous purchase or new)
- Gallons requested
- Company profit margin (%)

Software must include following components:
- Login (Allow Client to register if not a client yet)
- Client Registration (Initially only username and Password)
- Client Profile Management (after client registers they should login first to complete profile)
- Fuel Quote Form with Pricing module (Once user enters all required information pricing module calculates the rate provides total cost)
- Fuel Quote History

Answer these questions:

1. Discuss your initial thoughts in details on how you will design this application? (2 points)
2. Discuss what development methodology you will use and why? (2 points)
3. Provide high level design / architecture of your solution that you are proposing? (6 points)
4. **IMPORTANT: list who did what within the group. TAs should be able to validate in GitHub, otherwise team members who didn't contribute will receive a ZERO.**

Fill in this table, provide as much details possible:

Group Member Name	What is your contribution?	Discussion Notes
1.		
2.		

3.		

NOTE: Include a separate file (may be a scan of hand drawn picture if you like, or a PDF, tiff, word document, etc.) which shows a UML diagram of your design.
Here are some architecture examples: https://www.edrawsoft.com/software-architecture-example.php
What to turn in:
- Only soft copy uploaded to blackboard before due date.
- Only one submission per group.
- No extensions.
- All group members must contribute equally.

Assignment 2

In this assignment you will build the front end for the web application that you designed in assignment 1.
Remember, **we are only building front end in this assignment.**

Description:
Same as assignment 1.

Additional Details:
Front end must include following components:
- Login (Allow Client to register if not a client yet)
- Client Registration (Initially only username and Password)
- Client Profile Management (After client registers they should login first to complete the profile). Following fields will be on Profile page / form:
    - Full Name (50 characters, required)
    - Address 1 (100 characters, required)
    - Address 2 (100 characters, optional)
    - City (100 characters, required)
    - State (Drop Down, selection required) DB will store 2 character state code
    - Zipcode (9 characters, at least 5 character code required)

- Fuel Quote Form with following fields: (We are not building pricing module yet)
    - Gallons Requested (numeric, required)
    - Delivery Address (Non-editable, comes from client profile)
    - Delivery Date (Calender, date picker)
    - Suggested Price / gallon (numeric non-editable, price will be calculated by Pricing Module - we are not building pricing module yet)
    - Total Amount Due (numeric non-editable, calculated (gallons * price))

- Fuel Quote History
    - Tabular display of all client quotes in the past. All fields from Fuel Quote are displayed.

- You should have validations in place for required fields, field types, and field lengths.

- **NOTE:** Only provide a word / pdf doc. You should use GitHub for your group collaboration and code.

Answer these questions:
1. Provide link to GitHub repository for TAs to view the code? (1 point)
2. Discuss if your design and development methodology has changed since assignment 1 and why? (1 point)
3. List what front end technologies you are using and why. List who is responsible of doing what in your group? (2 points)
4. Provide screen shots of your front end, each page? (5 points)
5. IMPORTANT: list who did what within the group. TAs should be able to validate in GitHub, otherwise team members who didn't contribute will receive a ZERO. (1 point)

Fill in this table, provide as much details possible:

Group Member Name	What is your contribution?	Discussion Notes
1.		
2.		
3.		

What to turn in:
- Only soft copy uploaded to BlackBoard before due date.
- **DO NOT SUBMIT CODE to BlackBoard.**
- Only one submission per group.
- No extensions.
- **All group members must contribute equally.**

Assignment 3

In this assignment you will build the back end for the web application that you designed in assignment 1.
Remember, we are only building back end in this assignment.

Description:
Same as assignment 1.

Additional Details:

Back end must include following components/modules:

- Login module
- Client Profile Management module
- Fuel Quote module, includes list of quote history for a client.
- Pricing module. Only create a class. You will implement this in last assignment.

Important deliverables:
- You should have validations in place for required fields, field types, and field lengths in backend code as well.
- All backend code should be covered by unit tests. Code coverage should be grater than 80%.
- Research how to run the code coverage reports. Each IDE has plugins to generate reports. **Here are few pointers.** **https://stackify.com/code-coverage-tools/**
- All front end should be connected to back end. Form data should be populated from backend. Backend should receive data from front end, validate, and prepare it to persist to DB.
- WE ARE NOT IMPLEMENTING DB yet. For this assignment you can hard code the values.

NOTE: Only provide a word / pdf doc. You should use GitHub for your group collaboration and code.

Answer these questions:
1. Provide link to GitHub repository for TAs to view the code. Code should include unit tests.(5 points)
2. List what backend technologies you are using and why? (2 points)
3. IMPORTANT: list who did what within the group. TAs should be able to validate in GitHub, otherwise team members who didn't contribute will receive a ZERO.

Fill in this table, provide as much details possible:

Group Member Name	What is your contribution?	Discussion Notes
1.		
2.		
3.		

What to turn in: - Only soft copy uploaded to BlackBoard before due date.
- **DO NOT SUBMIT CODE to BlackBoard.**
- Only one submission per group.
- No extensions.
- **All group members must contribute equally.**

Assignment 4

In this assignment you will create the database and connect it to your web application.

Description:
Same as assignment 1.

Additional Details:

Database must include following tables:
- UserCredentials (ID & password), password should be encrypted.
- ClientInformation
- FuelQuote
- Any additional tables you feel, like States.

Important deliverables:
- You should have validations in place for required fields, field types, and field lengths.
- Backend should retrieve data from DB and display it to front end.
- Form data should be populated from backend. Backend should receive data from front end, validate, and persist to DB.
- Any new code added should be covered by unit tests. Keep code coverage above 80%.

- **NOTE:** Only provide a word / pdf doc. You should use GitHub for your group collaboration and code.

Answer these questions:
1. Provide link to GitHub repository for TAs to view the code. (5 points)
2. Provide SQL statements to create database.(3 points)
3. Rerun the code coverage report and provide it. (2 points)
4. IMPORTANT: list who did what within the group. TAs should be able to validate in GitHub, otherwise team members who didn't contribute will receive a ZERO.

Fill in this table, provide as much details possible:

Group Member Name	What is your contribution?	Discussion Notes
1.		
2.		
3.		

What to turn in:
- Only soft copy uploaded to BlackBoard before due date.
- DO NOT SUBMIT CODE to BlackBoard.

- Only one submission per group.
- No extensions.
- **All group members must contribute equally.**

Complete Project with Demo

In this assignment you will create the pricing module for Fuel Quote Form and put together the complete project that you have built so far.

Description:
Same as assignment 1.

Additional Details:

Create a pricing module that should calculate the price per gallon based on this formula.

Suggested Price = Current Price + Margin

Where,

Current price per gallon = $1.50 (this is the price what distributor gets from refinery and it varies based upon crude price. But we are keeping it constant for simplicity)
Margin = Current Price * (Location Factor - Rate History Factor + Gallons Requested Factor + Company Profit Factor)

Consider these factors:
Location Factor = 2% for Texas, 4% for out of state.
Rate History Factor = 1% if client requested fuel before, 0% if no history (you can query fuel quote table to check if there are any rows for the client)
Gallons Requested Factor = 2% if more than 1000 Gallons, 3% if less
Company Profit Factor = 10% always

Example:
1500 gallons requested, in state, does have history (i.e. quote history data exist in DB for this client)

Margin => (.02 - .01 + .02 + .1) * 1.50 = .195
Suggested Price/gallon => 1.50 + .195 = $1.695
Total Amount Due => 1500 * 1.695 = $2542.50

Additional Validations:
- Make suggested price and total amount fields in your Quote form read-only, i.e. user cannot enter these values.
- Create another button on Quote Form before Submit, call it "Get Quote".
- After user enters all other fields in the form other than Suggested Price and Total Amount, allow user to click on "Get Quote", i.e. Get Quote and Submit Quote buttons should be disabled if there are no values entered in the form.
- When user clicks on "Get Quote" button make a call to Pricing Module and populate the suggested price and total.

- Display Suggested Price and Total Amount once you get the values from pricing module.
- Make sure you do not lose any form values when you make a call to Pricing module.
- You can use AJAX call to achieve this i.e. partial form submission.
- Then user clicks on Submit Quote and you save the quote.

Deliverables:

1. Provide link to GitHub repository for TAs to view the code. (5 points)
2. Rerun the code coverage report. Code coverage must be above 80%. (2 points)
3. Demo prior to project due date. (3 points)
4. **IMPORTANT: list who did what within the group. TAs should be able to validate in GitHub, otherwise team members who didn't contribute will receive a ZERO.**

Fill in this table, provide as much details possible:

Group Member Name	What is your contribution?	Discussion Notes
1.		
2.		
3.		

IMPORTANT:

All group members must be present for demo. TAs will validate who did what.

What to turn in:

- Only soft copy uploaded to BlackBoard before due date.
- **DO NOT SUBMIT CODE to BlackBoard.**
- Only one submission per group.
- No extensions.
- All group members must contribute equally.