requestQuote.html:

<!DOCTYPE html>  
<html lang="en" dir="ltr">  
  
<head>  
 <meta charset="utf-8">  
 <link rel="stylesheet" href="general.css" type="text/css">  
 <title>Request A Quote Page</title>  
</head>  
  
<body>  
<div class="mainHeader">  
 <h1>Request a Quote</h1>  
</div>  
<div class="mainNav">  
 <ul>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="requestQuote.html">Request a Quote</a></li>  
 <li><a>Account Info</a>  
 <ul>  
 <li><a href="clientInfo.html">Client Information</a></li>  
 <li><a href="quoteHistory.html">Quote History</a></li>  
 </ul>  
 </li>  
 <li><a>About Us</a>  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="ourMission.html">Mission Statement</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 </li>  
 </ul>  
</div>  
  
  
<div id="formMainContent">  
 <form id="mainForm">  
 <fieldset>  
 <legend><b> Please Enter the Information to Recieve your Quote </b></legend>  
 <label>Gallons Requested:</label>  
 <input id="gallonsRequested" type="text" name="double"> <br> <!-- TextBox Generation -->  
 <P id="gallonsError" class="errorMessage"></P>  
  
 <label>Delivery Date:</label>  
 <input id="deliveryDate" type="text" name="date"> <br> <!-- TextBox Generation -->  
 <P id="dDateError" class="errorMessage"></P>  
  
 <label>Delivery Location:</label>  
 <input id="deliveryLocation" type="text" name="string"> <br>  
 <P id="dLocationError" class="errorMessage"></P>  
  
 <label>Delivery Contact Person Name:</label>  
 <input id="deliveryContactPersonName" type="text" name="name"> <br>  
 <P id="dContactNameError" class="errorMessage"></P>  
  
 <label>Delivery Contact Person Phone:</label>  
 <input id="deliveryContactPersonPhone" type="text" name="phone"> <br>  
 <P id="dContactPhoneError" class="errorMessage"></P>  
  
 <label>Delivery Contact Person Email:</label>  
 <input id="deliveryContactPersonEmail" type="text" name="email"> <br>  
 <P id="dContactEmailError" class="errorMessage"></P>  
  
 <label>Suggested Price Per Gallon:</label>  
 <input id="suggestedPricePerGallon" type="text" name="double"> <br>  
 <P id="priceGalError" class="errorMessage"></P>  
  
 <label>Total Amount Due:</label>  
 <input id="totalAmountDue" type="text" name="double"> <br>  
 <P id="totAmtError" class="errorMessage"></P>  
  
 <input type="button" id="submit" value="Submit" onclick="createQuote()">  
  
  
 <p id="subStatus" class="errorMessage"></p>  
  
 </fieldset>  
 </form>  
</div>  
  
<footer class="mainFooter">  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="./clientInfo.html">Client Information</a></li>  
 <li><a href="./quoteHistory.html">Quote History</li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 <p>Copyright &copy; 2018 Samuel Pugh, William Fallin. All rights reserved</p>  
  
 <script src="./controller.js" type="text/javascript">  
  
 </script>  
  
</footer>  
  
</body>  
  
</html>

clientInfo.html:

<!DOCTYPE html>  
<html lang="en" dir="ltr">  
  
<head>  
 <meta charset="utf-8">  
 <link rel="stylesheet" href="general.css" type="text/css">  
 <title>Client Information</title>  
</head>  
  
<body>  
<div class="mainHeader">  
 <h1>Client Information</h1>  
</div>  
<div class="mainNav">  
 <ul>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="requestQuote.html">Request a Quote</a></li>  
 <li><a>Account Info</a>  
 <ul>  
 <li><a href="clientInfo.html">Client Information</a></li>  
 <li><a href="quoteHistory.html">Quote History</a></li>  
 </ul>  
 </li>  
 <li><a>About Us</a>  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="ourMission.html">Mission Statement</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 </li>  
 </ul>  
</div>  
<div class="mainContent">  
 <table id="clientInfoBlock">  
 <th colspan="2">Client Information</th>  
 <tr>  
 <td>Name:</td>  
 <td id="clientInfoName"></td>  
 </tr>  
 <tr>  
 <td>Address:</td>  
 <td id="clientInfoAddress"></td>  
 </tr>  
 <tr>  
 <td>Phone:</td>  
 <td id="clientInfoPhone"></td>  
 </tr>  
 <tr>  
 <td>Email:</td>  
 <td id="clientInfoEmail"></td>  
 </tr>  
 </table>  
 <!--<form id="idForm">-->  
 <!--<label for="">Search An Id Number:</label>-->  
 <!--<input type="text" name="id"> <br>-->  
 <!--<input id="clientIdSubmit" type="submit" value="Submit" onclick="handler()">-->  
 <!--</form>-->  
  
</div>  
  
<footer class="mainFooter">  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="requestQuote.html">Request A Quote</a></li>  
 <li><a href="./quoteHistory.html">Quote History</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 <p>Copyright &copy; 2018 Samuel Pugh, William Fallin. All rights reserved</p>  
  
 <script src="./controller.js" type="text/javascript">  
  
 </script>  
 <script>displayClient()</script>  
  
</footer>  
  
</body>  
  
</html>

quoteHistory.html:

<!DOCTYPE HTML>  
<html lang="en" dir="ltr">  
  
<head>  
 <meta charset="utf-8">  
 <link rel="stylesheet" href="general.css" type="text/css">  
 <title>Quote History</title>  
</head>  
  
<body>  
<div class="mainHeader">  
 <h1>Quote History</h1>  
</div>  
<div class="mainNav">  
 <ul>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="requestQuote.html">Request a Quote</a></li>  
 <li><a>Account Info</a>  
 <ul>  
 <li><a href="clientInfo.html">Client Information</a></li>  
 <li><a href="quoteHistory.html">Quote History</a></li>  
 </ul>  
 </li>  
 <li><a>About Us</a>  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="ourMission.html">Mission Statement</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 </li>  
 </ul>  
</div>  
  
  
<div class="mainContent">  
 <table id="quoteHistoryTable">  
 <tr id="quoteHistoryHeaders">  
 <!--Header Elements for the Tables -->  
 <th>Item #</th>  
 <th>Request Date</th>  
 <th>Delivery Date</th>  
 <th>Gallons</th>  
 <th>Rate</th>  
 <th>Total Price</th>  
 </table>  
</div>  
  
<footer class="mainFooter">  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="./clientInfo.html">Client Information</a></li>  
 <li><a href="./requestQuote.html">Request A Quote</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 <p>Copyright &copy; 2018 Samuel Pugh, William Fallin. All rights reserved</p>  
  
 <!-- Tie the javascript to the HTML file -->  
 <script src="controller.js">  
 </script>  
 <script>populateQuoteHistory()</script>  
  
</footer>  
  
</body>  
  
</html>

modifyClient.html:

<!DOCTYPE html>  
<html lang="en" dir="ltr">  
  
<head>  
 <meta charset="utf-8">  
 <link rel="stylesheet" href="general.css" type="text/css">  
 <title>Modify Current Client</title>  
</head>  
  
<body>  
<div class="mainHeader">  
 <h1>Modify Current Client</h1>  
</div>  
<div class="mainNav">  
 <ul>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="requestQuote.html">Request a Quote</a></li>  
 <li><a>Account Info</a>  
 <ul>  
 <li><a href="clientInfo.html">Client Information</a></li>  
 <li><a href="modifyClient.html">Modify Current Client</a></li>  
 <li><a href="quoteHistory.html">Quote History</a></li>  
 </ul>  
 </li>  
 <li><a>About Us</a>  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="ourMission.html">Mission Statement</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 </ul>  
 </li>  
 </ul>  
</div>  
  
  
<div id="formMainContent">  
 <form id="modClientForm">  
 <fieldset>  
 <legend><b> Please Enter the modified client info </b></legend>  
  
 <label>Client Name:</label>  
 <input id="fullName" type="text" name="name"> <br>  
 <P id="fullNameError" class="errorMessage"></P>  
  
 <label>Client Address:</label>  
 <input id="address" type="text" name="string"> <br>  
 <P id="addressError" class="errorMessage"></P>  
  
 <label>Client Phone:</label>  
 <input id="phone" type="text" name="phone"> <br>  
 <P id="phoneError" class="errorMessage"></P>  
  
 <label>Client Email:</label>  
 <input id="email" type="text" name="email"> <br>  
 <P id="emailError" class="errorMessage"></P>  
  
 <input type="button" id="submit" value="Submit" onclick="modifyClient()">  
  
 <p id="subStatus" class="errorMessage"></p>  
  
 </fieldset>  
 </form>  
</div>  
  
<footer class="mainFooter">  
 <ul>  
 <li><a href="contactUs.html">Contact Us</a></li>  
 <li><a href="./homePage.html">Home Page</a></li>  
 <li><a href="./clientInfo.html">Client Information</a></li>  
 <li><a href="./quoteHistory.html">Quote History</a></li>  
 <li><a href="socialMedia.html">Social Media</a></li>  
 <li><a href="requestQuote.html">Request a Quote</a></li>  
 </ul>  
 <p>Copyright &copy; 2018 Samuel Pugh, William Fallin. All rights reserved</p>  
  
 <script src="./controller.js" type="text/javascript">  
 </script>  
 <script>loadClientIntoModifyForm()</script>  
  
</footer>  
  
</body>  
  
</html>

Controller.js:

// FUNCTIONAL \\  
**class** ClientInformation {  
  
 constructor(clientId = 0, fullName = 'DefaultName', address = 'DefaultAddress',  
 city = 'DefaultCity', state = 'DefaultState', zipCode = 'DefaultZipCode',  
 phone = 'DefaultPhone', email = 'DefaultEmail') {  
 **this**.clientId = clientId;  
 **this**.fullName = fullName;  
 **this**.address = address;  
 **this**.city = city;  
 **this**.state = state;  
 **this**.zipCode = zipCode;  
 **this**.phone = phone;  
 **this**.email = email;  
 }  
  
}  
  
// FUNCTIONAL \\  
**class** Quote {  
  
 constructor(id, clientId, gallonsRequested, requestDate, deliveryDate, deliveryAddress, deliveryCity, deliveryState, deliveryZipCode, deliveryContactName, deliveryContactPhone, deliveryContactEmail, suggestedPrice, totalAmountDue) {  
 **this**.quoteId = id;  
 **this**.clientId = clientId;  
 **this**.gallonsRequested = gallonsRequested;  
 **this**.requestDate = requestDate;  
 **this**.deliveryDate = deliveryDate;  
 **this**.deliveryAddress = deliveryAddress;  
 **this**.deliveryCity = deliveryCity;  
 **this**.deliveryState = deliveryState;  
 **this**.deliveryZipCode = deliveryZipCode;  
 **this**.deliveryContactName = deliveryContactName;  
 **this**.deliveryContactPhone = deliveryContactPhone;  
 **this**.deliveryContactEmail = deliveryContactEmail;  
 **this**.suggestedPrice = suggestedPrice;  
 **this**.totalAmountDue = totalAmountDue;  
 }  
}  
  
// FUNCTIONAL \\  
**class** quoteHistory {  
 constructor() {  
 // Create and array of quotes  
 **this**.quoteList = [];  
 }  
}  
  
{ // Functions that are called when on the requestQuote page  
 // Function is called when the submit button is clicked on requestQuote Page  
 **function** createQuote() {  
 // grab all of the elements of the main form.  
 **var** gallons = document.getElementById("gallonsRequested").value;  
 **var** devDate = document.getElementById("deliveryDate").value;  
 **var** devPerName = document.getElementById("deliveryContactPersonName").value;  
 **var** devPerPhone = document.getElementById("deliveryContactPersonPhone").value;  
 **var** devPerEmail = document.getElementById("deliveryContactPersonEmail").value;  
 **var** sugPrice = document.getElementById("suggestedPricePerGallon").value;  
 **var** totAmountDue = document.getElementById("totalAmountDue").value;  
 **var** devLocation = document.getElementById("deliveryLocation").value;  
  
 **var** validFieldList = [];  
 validFieldList.push(devLocation);  
  
 // JS considers an empty string to be false, if none are empty -> do validation  
 **if** (gallons && devDate && devLocation && devPerName  
 && devPerPhone && devPerEmail && sugPrice && totAmountDue) {  
 { // Double fields validation  
 **if** (isDoubleValid(gallons)) {  
 console.log("gallonsRequested:" + gallons);  
 validFieldList.push(gallons);  
 document.getElementById("gallonsError").innerHTML = "";  
 }  
 **else** {  
 document.getElementById("gallonsError").innerHTML = "Gallons requested must be a number e.g. '##.##'";  
 }  
  
 **if** (isDoubleValid(totAmountDue)) {  
 console.log("totalAmountDue:" + totAmountDue);  
 validFieldList.push(totAmountDue);  
 document.getElementById("totAmtError").innerHTML = "";  
 }  
 **else** {  
 document.getElementById("totAmtError").innerHTML = "Total amount due must be a number e.g. '##.##'";  
 }  
  
 **if** (isDoubleValid(sugPrice)) {  
 console.log("suggestedPricePerGallon:" + sugPrice);  
 validFieldList.push(sugPrice);  
 document.getElementById("priceGalError").innerHTML = "";  
 }  
 **else** {  
 document.getElementById("priceGalError").innerHTML = "Suggested rate must be a number e.g. '##.##'";  
 }  
 }  
  
 { // Testing the name for special characters  
 **var** a = /^[a-zA-Z ]\*$/i;  
 **if** (a.test(devPerName)) {  
 console.log("deliveryPersonName: " + devPerName);  
 validFieldList.push(devPerName);  
 document.getElementById("dContactNameError").innerHTML = "";  
 }  
 **else** {  
 document.getElementById("dContactNameError").innerHTML = "Delivery contact name must only contain alphabet characters and cannot be blank";  
 }  
 }  
  
 { // Testing the date for correct format  
 **var** dateForm = /^\d{1,2}\/\d{1,2}\/\d{4}$/;  
 **if** (dateForm.test(devDate)) {  
 console.log("deliveryDate: " + devDate);  
 realDevDate = devDate.substr(6, 10) + '-' + devDate.substr(0, 2) + '-' + devDate.substr(3, 2) + ' 00:00:00';  
 validFieldList.push(realDevDate);//devDate.replace(/\//g, '-'));  
 document.getElementById("dDateError").innerHTML = "";  
 }  
 **else** {  
 document.getElementById("dDateError").innerHTML = "Date must be formatted MM/DD/YYYY";  
 }  
 }  
  
 { // Testing the phone number for correct format  
 **var** phoneForm = /^\d{3}\-\d{3}\-\d{4}$/;  
  
 **if** (phoneForm.test(devPerPhone)) {  
 console.log("deliveryPersonPhone: " + devPerPhone);  
 validFieldList.push(devPerPhone.replace(/-/g, ''));  
 document.getElementById("dContactPhoneError").innerHTML = ""  
 }  
 **else** {  
 document.getElementById("dContactPhoneError").innerHTML = "Phone number must be formatted ###-###-####"  
 }  
  
 }  
  
 { // Testing the email for correct formatting  
 **var** delimiter = /[@]/;  
 **var** length = devPerEmail.length;  
 **var** subString = devPerEmail.substring(length - 4, length);  
  
 **if** (delimiter.test(devPerEmail) && (subString === ".com" || subString === ".edu" || subString === ".gov")) {  
 console.log("deliveryPersonEmail: " + devPerEmail);  
 validFieldList.push(devPerEmail);  
 document.getElementById("dContactEmailError").innerHTML = ""  
 }  
 **else** {  
 document.getElementById("dContactEmailError").innerHTML = "Email must be formatted as XX@XX{.com, .gov, .edu}";  
 }  
 }  
  
 // Done with error checking  
 document.getElementById("subStatus").innerHTML = "";  
  
 **if** (validFieldList.length === 8) {  
 // Getting today's date  
 **var** today = **new** Date();  
 **var** dd = today.getDate();  
 **var** mm = today.getMonth()+1; //January is 0!  
 **var** yyyy = today.getFullYear();  
 **if**(dd<10) {  
 dd = '0'+dd  
 }  
 **if**(mm<10) {  
 mm = '0'+mm  
 }  
 // SQL wants it like this 2018-11-15 00:00:00  
 today = yyyy + '-' + mm + '-' + dd + ' 00:00:00';  
  
 quoteToSend = **new** Quote('NextID', sessionStorage.clientId, validFieldList[1], today, validFieldList[5], validFieldList[0],  
 'City', 'TX', '78666', validFieldList[4], validFieldList[6], validFieldList[7],  
 validFieldList[3], validFieldList[2]);  
  
 fetch('http://localhost:3000/api/fuelquotes/addquote', {  
 method: 'POST',  
 body: JSON.stringify(quoteToSend),  
 headers: {'Content-Type': 'application/json'}  
 });  
  
 }  
 }  
 // The else to our top if; there are fields that are empty  
 **else** {  
 document.getElementById("subStatus").innerHTML = "Errors in form, all fields must be filled out.";  
 }  
 } // End of createQuote  
  
 // Function takes a double and checks it to make sure its a number that we can use  
 **function** isDoubleValid(dToValidate) {  
 dToValidate = parseFloat(dToValidate).toFixed(2);  
  
 **var** decRegex = /^\d\*\.\d{2}$/;  
  
 **return** (decRegex.test(dToValidate));  
  
 } // End isDoubleValid  
}  
  
{ // Functions that are called by clientInfo.html  
 // Gets client based on ID from db and displays it  
 **function** displayClient(id = sessionStorage.clientId) {  
 **const** Http = **new** XMLHttpRequest();  
 **const** url = 'http://localhost:3000/api/clients/' + id;  
 Http.open("GET", url);  
 Http.send();  
  
 **var** parsedJSON = '';  
 **var** currentClient = '';  
  
 Http.onreadystatechange = **function** () {  
 **if** (**this**.readyState == 4 && parsedJSON === '') {  
 parsedJSON = JSON.parse(Http.responseText);  
 currentClient = **new** ClientInformation(parsedJSON[0].clientId, parsedJSON[0].fullName,  
 parsedJSON[0].address, parsedJSON[0].city, parsedJSON[0].state, parsedJSON[0].zipCode,  
 parsedJSON[0].phone, parsedJSON[0].email);  
 document.getElementById("clientInfoName").innerHTML = parsedJSON[0].fullName;  
 document.getElementById("clientInfoAddress").innerHTML = parsedJSON[0].address;  
 document.getElementById("clientInfoPhone").innerHTML = formatPhone(parsedJSON[0].phone);  
 document.getElementById("clientInfoEmail").innerHTML = parsedJSON[0].email;  
 }  
 };  
 }  
 // Changes client and reflects that in session storage  
 **function** changeClient() {  
 sessionStorage.clientId = document.getElementById("newClientId").value;  
 displayClient();  
 }  
}  
  
{ // Function that is run when quoteHistory is loaded  
 **function** populateQuoteHistory(id = sessionStorage.clientId) {  
 **const** Http = **new** XMLHttpRequest();  
 **const** url = 'http://localhost:3000/api/fuelquotes/' + id;  
 Http.open("GET", url);  
 Http.send();  
  
 **var** parsedJSON = '';  
 **var** currentClient = '';  
  
 Http.onreadystatechange = **function** () {  
 **if** (**this**.readyState == 4 && parsedJSON === '') {  
 parsedJSON = JSON.parse(Http.responseText);  
 displayQuoteHistory(parsedJSON);  
 }  
 };  
 }  
  
  
 // This will eventually retrieve the data from the DB  
 **function** displayQuoteHistory(parsedJSON) {  
 historyOfQuotes = **new** quoteHistory();  
 **for** (**var** i = 0; i < parsedJSON.length; i++) {  
 **var** tempQuote = **new** Quote(parsedJSON[i].quoteId, parsedJSON[i].clientId, parsedJSON[i].gallonsRequested,  
 parsedJSON[i].requestDate, parsedJSON[i].deliveryDate, parsedJSON[i].deliveryAddress,  
 parsedJSON[i].deliveryCity, parsedJSON[i].deliveryState, parsedJSON[i].deliveryZipCode,  
 parsedJSON[i].deliveryContactName, parsedJSON[i].deliveryContactPhone,  
 parsedJSON[i].deliveryContactEmail, parsedJSON[i].suggestedPrice, parsedJSON[i].totalAmountDue);  
  
 historyOfQuotes.quoteList.push(tempQuote);  
 }  
  
 // for each outer array row  
 **for** (**var** i = 0; i < historyOfQuotes.quoteList.length; i++) {  
 **var** tr = document.createElement("tr");  
  
 // for each inner array cell  
 // create td then text, append  
 **var** td = document.createElement("td");  
 **var** txt = document.createTextNode(historyOfQuotes.quoteList[i].quoteId);  
 td.appendChild(txt);  
 tr.appendChild(td);  
  
 **var** td = document.createElement("td");  
 **var** txt = document.createTextNode(historyOfQuotes.quoteList[i].requestDate.substr(0,10));  
 td.appendChild(txt);  
 tr.appendChild(td);  
  
 **var** td = document.createElement("td");  
 **var** txt = document.createTextNode(historyOfQuotes.quoteList[i].deliveryDate.substr(0, 10));  
 td.appendChild(txt);  
 tr.appendChild(td);  
  
 **var** td = document.createElement("td");  
 **var** txt = document.createTextNode(historyOfQuotes.quoteList[i].gallonsRequested);  
 td.appendChild(txt);  
 tr.appendChild(td);  
  
 **var** td = document.createElement("td");  
 **var** txt = document.createTextNode(historyOfQuotes.quoteList[i].suggestedPrice);  
 td.appendChild(txt);  
 tr.appendChild(td);  
  
 **var** td = document.createElement("td");  
 **var** txt = document.createTextNode(historyOfQuotes.quoteList[i].totalAmountDue);  
 td.appendChild(txt);  
 tr.appendChild(td);  
  
 // append row to table  
 document.getElementById("quoteHistoryTable").appendChild(tr);  
 }  
 }  
}  
  
{ // Function thats called by the modify client page  
 **function** modifyClient() {  
 **let** newFullName = document.getElementById("fullName").value;  
 **let** newAddress = document.getElementById("address").value;  
 **let** newPhone = (document.getElementById("phone").value).replace(/-/g, '');  
 **let** newEmail = document.getElementById("email").value;  
  
 **let** clientJSON = JSON.stringify(**new** ClientInformation(sessionStorage.clientId, newFullName, newAddress,  
 'city', 'state', 'zip', newPhone, newEmail));  
  
 fetch('http://localhost:3000/api/clients', {  
 method: 'PUT',  
 body: clientJSON,  
 headers: {'Content-Type': 'application/json'}  
 });  
 }  
  
 **function** loadClientIntoModifyForm(id = sessionStorage.clientId) {  
 **const** Http = **new** XMLHttpRequest();  
 **const** url = 'http://localhost:3000/api/clients/' + id;  
 Http.open("GET", url);  
 Http.send();  
  
 **var** parsedJSON = '';  
 **var** currentClient = '';  
  
 Http.onreadystatechange = **function** () {  
 **if** (**this**.readyState == 4 && parsedJSON === '') {  
 parsedJSON = JSON.parse(Http.responseText);  
 currentClient = **new** ClientInformation(parsedJSON[0].clientId, parsedJSON[0].fullName,  
 parsedJSON[0].address, parsedJSON[0].city, parsedJSON[0].state, parsedJSON[0].zipCode,  
 parsedJSON[0].phone, parsedJSON[0].email);  
 document.getElementById("fullName").value = parsedJSON[0].fullName;  
 document.getElementById("address").value = parsedJSON[0].address;  
 document.getElementById("phone").value = formatPhone(parsedJSON[0].phone);  
 document.getElementById("email").value = parsedJSON[0].email;  
 }  
 };   
 }  
  
}  
  
// Additional functions  
**function** formatPhone(phoneNum) {  
 phoneNum = phoneNum.toString();  
 formattedPhone = phoneNum.substr(0,3) + '-' + phoneNum.substr(3,3) + '-' + phoneNum.substr(6,4);  
 **return** formattedPhone;  
}  
  
**if**(**typeof** sessionStorage.clientId === 'undefined'){  
 sessionStorage.clientId = '1';  
}

queryServer.js:

**const** mysql = require('mysql');  
**const** express = require('express');  
**const** bodyParser = require('body-parser');  
**const** SqlString = require('sqlstring');  
  
**const** app = express();  
  
// Add headers  
app.use(**function** (req, res, next) {  
  
 // Website you wish to allow to connect  
 res.setHeader('Access-Control-Allow-Origin', 'http://localhost:63342');  
  
 // Request methods you wish to allow  
 res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT');  
  
 // Request headers you wish to allow  
 res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type');  
  
 // Set to true if you need the website to include cookies in the requests sent  
 // to the API (e.g. in case you use sessions)  
 res.setHeader('Access-Control-Allow-Credentials', **true**);  
  
 // Pass to next layer of middleware  
 next();  
});  
  
app.use(bodyParser.json());  
app.use(bodyParser.urlencoded({extended: **false**}));  
  
app.get('/api/clients/:id', (req, res) => {  
 **var** con = mysql.createConnection({  
 host: "localhost",  
 user: "root",  
 password: "password",  
 database: 'CS3320',  
 });  
 con.connect(**function** (err) {  
 **if** (err) **throw** err;  
 con.query("SELECT \* FROM clientinformation WHERE clientId=" + req.params.id, **function** (err, result, fields) {  
 **if** (err) **throw** err;  
 res.send(result);  
 });  
 });  
});  
  
app.get('/api/fuelquotes/:id', (req, res) => {  
 **var** con = mysql.createConnection({  
 host: "localhost",  
 user: "root",  
 password: "password",  
 database: 'CS3320',  
 });  
 con.connect(**function** (err) {  
 **if** (err) **throw** err;  
 con.query("SELECT \* FROM fuelquote WHERE clientId=" + req.params.id, **function** (err, result, fields) {  
 **if** (err) **throw** err;  
 res.send(result);  
 });  
 });  
});  
  
app.post('/api/fuelquotes/addquote', **function** (req, res) {  
 //Retrieving data:  
 **let** parsedJSON = req.body;//JSON.parse(req.body);  
 **let** con = mysql.createConnection({  
 host: "localhost",  
 user: "root",  
 password: "password",  
 database: 'CS3320',  
 });  
 **let** sql = ("INSERT INTO fuelQuote (clientId, gallonsRequested, requestDate, deliveryDate, deliveryAddress, " +  
 "deliveryCity, deliveryState, deliveryZipCode, deliveryContactName, deliveryContactPhone, " +  
 "deliveryContactEmail, suggestedPrice, totalAmountDue) VALUES (" + SqlString.escape(req.body.clientId) + ', '  
 + SqlString.escape(req.body.gallonsRequested) + ', ' + SqlString.escape(req.body.requestDate) + ', '  
 + SqlString.escape(req.body.deliveryDate) + ', ' + SqlString.escape(req.body.deliveryAddress) + ', '  
 + SqlString.escape(req.body.deliveryCity) + ', ' + SqlString.escape(req.body.deliveryState) + ', '  
 + SqlString.escape(req.body.deliveryZipCode) + ', ' + SqlString.escape(req.body.deliveryContactName) + ', '  
 + SqlString.escape(req.body.deliveryContactPhone) + ', ' + SqlString.escape(req.body.deliveryContactEmail) + ', '  
 + SqlString.escape(req.body.suggestedPrice) + ', ' + SqlString.escape(req.body.totalAmountDue) + ')');  
 con.connect(**function** (err) {  
 **if** (err) **throw** err;  
 con.query(sql, **function** (err, result) {  
 **if** (err) **throw** err;  
 });  
 });  
  
 res.send();  
});  
  
app.put('/api/clients', **function** (req, res) {  
 // Retrieving new client:  
 **let** con = mysql.createConnection({  
 host: "localhost",  
 user: "root",  
 password: "password",  
 database: 'CS3320',  
 }); // fullName address phone email  
 **let** sql = ("UPDATE clientinformation SET fullName= " + SqlString.escape(req.body.fullName)  
 + ", address= " + SqlString.escape(req.body.address)  
 + ", phone= " + SqlString.escape(req.body.phone)  
 + ", email= " + SqlString.escape(req.body.email)  
 + " WHERE clientId = " + SqlString.escape(req.body.clientId));  
 con.connect(**function** (err) {  
 **if** (err) **throw** err;  
 con.query(sql, **function** (err, result) {  
 **if** (err) **throw** err;  
 });  
 });  
  
});  
  
app.listen(3000, () => console.log('Listening on 3000...'));  
  
// *TODO: Get City, State, and Zip from user in request quote page*

Screenshots:

