Simplified Front End Assignment - Code and Explanations

This document contains all the code for the form, including HTML, JavaScript, and database SQL, along with a brief explanation of each.

# 1. HTML Code - Customer Contact Form

The HTML code creates the structure of the form, including input fields for name, phone, email, and password. It also handles the form submission.

## Headline:

HTML structure for a customer contact form.

This HTML creates a contact form, where users can enter their details and submit it to the system.

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Customer Contact Form</title>  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
 <link rel="stylesheet" href="style.css">   
</head>  
<body>  
 <div class="container mt-5">  
 <div id='result'></div>  
 <fieldset class="border p-4">  
 <legend class="w-auto">Customer Contact Form</legend>  
 <form name="frmContact" action="customers.php" onsubmit="return validateForm()" method="post">  
 <div class="form-group">  
 <label for="lastName">Last Name:</label>  
 <input type="text" name="lastName" class="form-control"   
 onfocus="highlightInput(this)"   
 onblur="highlightOutput(this); chgupercase()"   
 placeholder="Enter your last name" id="lastName" required>  
 </div>  
  
 <div class="form-group">  
 <label for="firstName">First Name:</label>  
 <input type="text" name="firstName" class="form-control"   
 onfocus="highlightInput(this)"   
 onblur="highlightOutput(this); chgupercase()"   
 placeholder="Enter your first name" id="firstName" required>  
 </div>  
  
 <div class="form-group">  
 <label>Gender Type:</label>  
 <div>  
 <input type="radio" id="male" name="gender" value="male" checked> Male  
 <input type="radio" id="female" name="gender" value="female"> Female  
 </div>  
 </div>  
  
 <div class="form-group">  
 <label for="CustPhone">Customer Phone #:</label>  
 <input type="text" name="CustPhone" placeholder="Enter Customer Phone #" id="CustPhone" class="form-control" required/>  
 </div>  
  
 <div class="form-group">  
 <label for="email">Email Address:</label>  
 <input type="email" id="email" class="form-control"   
 onblur="validateEmail()"   
 placeholder="Enter your email" required/>  
 <div id='result'></div>  
 </div>  
  
 <div class="form-group">  
 <label for="address">Home Address:</label>  
 <input type="text" name="address" placeholder="Address" id="address" class="form-control" required/>  
 <input type="text" name="houseNumber" placeholder="House number" id="houseNumber" class="form-control mt-2" required/>  
 <select name="country" id="country" class="form-control mt-2" required>  
 <option disabled selected value>-- Select a province --</option>  
 <option value="ON">Ontario</option>  
 <option value="NS">Nova Scotia</option>  
 <option value="BC">British Columbia</option>  
 <option value="AB">Alberta</option>  
 <option value="MB">Manitoba</option>  
 <option value="QC">Quebec</option>  
 <option value="SK">Saskatchewan</option>  
 <option value="NL">Newfoundland and Labrador</option>  
 <option value="NB">New Brunswick</option>  
 <option value="PE">Prince Edward Island</option>  
 </select>  
 </div>  
  
 <div class="form-group">  
 <label for="password">Please enter complex password:</label>  
 <input type="password" placeholder="Enter password" id="password" class="form-control" required/>  
 <input type="password" placeholder="Confirm password" id="passwordCheck" class="form-control mt-2" required/>  
 </div>  
  
 <div class="form-group">  
 <input type="checkbox" id="newsletter"/> Send me monthly newsletter  
 </div>  
  
 <div class="form-group">  
 <label>What is your question?</label>  
 <textarea id="question" rows="4" class="form-control" required></textarea>  
 </div>  
  
 <button type="submit" name="Submit" id="Submit" class="btn btn-primary">Submit</button>  
 <button type="reset" value="Reset" class="btn btn-secondary">Reset</button>  
 </form>  
 </fieldset>  
 </div>  
   
 <script src="simplified\_script.js"></script>  
</body>  
</html>

# 2. JavaScript Code - Form Validation

The JavaScript file ensures the form inputs are valid before submission. It checks the name, email, phone number, and password to ensure they meet the rules.

## Headline:

JavaScript code for validating form inputs.

This code checks that the user enters correct details before submitting the form. It ensures the email, phone, and password follow the rules.

// Change to uppercase for names  
function chgupercase() {  
 var lastName = document.getElementById("lastName");  
 var firstName = document.getElementById("firstName");  
 lastName.value = lastName.value.toUpperCase();  
 firstName.value = firstName.value.toUpperCase();  
}  
  
// Highlighting the input fields  
function highlightInput(elm) {  
 elm.style.background = "yellow";  
}  
  
// Reset background on input fields  
function highlightOutput(elm) {  
 elm.style.background = "white";  
}  
  
// Check if the email is correct  
function validateEmail() {  
 let email = document.getElementById("email").value;  
 let emailRegex = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;  
 let result = document.getElementById("result");  
 if (!emailRegex.test(email)) {  
 result.textContent = email + " is not valid";  
 result.style.color = "red";  
 return false;  
 } else {  
 result.textContent = email + " is valid";  
 result.style.color = "blue";  
 return true;  
 }  
}  
  
// Check if phone number is valid  
function validatePhone() {  
 let phone = document.getElementById("CustPhone").value;  
 let phoneRegex = /^\d{10}$/;  
 if (!phoneRegex.test(phone)) {  
 alert("Phone number must be 10 digits.");  
 return false;  
 }  
 return true;  
}  
  
// Check password to follow the rules  
function validatePassword() {  
 let password = document.getElementById("password").value;  
 let passwordRegex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{16,}$/;  
 if (!passwordRegex.test(password)) {  
 alert("Password must be at least 16 characters long and include uppercase, lowercase, numbers, and special symbols.");  
 return false;  
 }  
 return true;  
}  
  
// Check the entire form before submitting  
function validateForm() {  
 let lastName = document.forms["frmContact"]["lastName"].value;  
 let firstName = document.forms["frmContact"]["firstName"].value;  
  
 if (lastName == "" || firstName == "") {  
 alert("Both names are required.");  
 return false;  
 }  
  
 if (!validateEmail() || !validatePhone() || !validatePassword()) {  
 return false;  
 }  
  
 return true;  
}

# 3. Database Code - SQL

The SQL code creates and alters a database table for storing customer data collected from the form. This includes fields for names, phone, email, and more.

## Headline:

SQL code for creating and altering the customer table.

This SQL script creates the necessary database structure to store customer information from the form, such as name, email, and phone.

CREATE DATABASE IF NOT EXISTS tbc;  
USE tbc;  
  
CREATE TABLE IF NOT EXISTS customers (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 lastName VARCHAR(255) NOT NULL,  
 firstName VARCHAR(255) NOT NULL  
);

USE tbc;  
  
-- Create the table if it doesn't exist  
CREATE TABLE IF NOT EXISTS customers (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 lastName VARCHAR(255) NOT NULL,  
 firstName VARCHAR(255) NOT NULL  
);  
  
-- Alter the table to add additional columns  
ALTER TABLE customers   
ADD gender VARCHAR(10) AFTER firstName,  
ADD custPhone VARCHAR(20) AFTER gender,  
ADD email VARCHAR(255) AFTER custPhone,  
ADD address VARCHAR(255) AFTER email,  
ADD houseNumber VARCHAR(50) AFTER address,  
ADD country VARCHAR(50) AFTER houseNumber,  
ADD password VARCHAR(255) AFTER country,  
ADD newsletter TINYINT(1) AFTER password,  
ADD question TEXT AFTER newsletter;