

Development Scenario: Insurance Claim Processing System

Day 1: HTML, CSS, and JavaScript - User Authentication and Profile Setup

Task 1: Design and code the HTML forms for user registration and login, ensuring accessibility standards are met.

Registration Form:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Registration Form</title>

  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

</head>

<body>

  <div class="container mt-5">

    <h2>Register</h2>

    <form id="registrationForm">

      <div class="form-group">

        <label for="username">Username</label>

        <input type="text" class="form-control" id="username" placeholder="Enter username" aria-
describedby="usernameHelp" required>

        <small id="usernameHelp" class="form-text text-muted">Your username must be
unique.</small>

      </div>

      <div class="form-group">

        <label for="email">Email address</label>

        <input type="email" class="form-control" id="email" placeholder="Enter email" required>

      </div>

      <div class="form-group">

        <label for="password">Password</label>
```

```

        <input type="password" class="form-control" id="password" placeholder="Password" required>
    </div>

    <div class="form-group">
        <label for="confirmPassword">Confirm Password</label>

        <input type="password" class="form-control" id="confirmPassword" placeholder="Confirm
Password" required>
    </div>

    <button type="submit" class="btn btn-primary">Register</button>
</form>
</div>

```

```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

Login Form:

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login Form</title>

    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

</head>

<body>

    <div class="container mt-5">

        <h2>Login</h2>

        <form id="loginForm">

            <div class="form-group">

```

```
<label for="loginEmail">Email address</label>

<input type="email" class="form-control" id="loginEmail" placeholder="Enter email" required>

</div>

<div class="form-group">

  <label for="loginPassword">Password</label>

  <input type="password" class="form-control" id="loginPassword" placeholder="Password"
required>

</div>

<button type="submit" class="btn btn-primary">Login</button>

</form>

</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>

</html>
```

Task 2: Apply CSS to style the forms for a consistent look and feel that aligns with the company's branding.

```
body {

  font-family: Arial, sans-serif;

  background-color: #f8f9fa;

  color: #333;

}
```

```
.container {

  max-width: 500px;

  margin: auto;

}
```

```

h2 {
  margin-bottom: 20px;
  color: #007bff;
}

.form-group {
  margin-bottom: 15px;
}

.btn-primary {
  background-color: #007bff;
  border-color: #007bff;
}

.btn-primary:hover {
  background-color: #0056b3;
  border-color: #0056b3;
}

.form-text {
  color: #6c757d;
}

```

Task 3: Implement JavaScript form validations to provide immediate feedback on user input errors before submission.

```
<script>
```

```

document.getElementById('registrationForm').addEventListener('submit', function(event) {
  event.preventDefault();

```

```

  const username = document.getElementById('username').value;
  const email = document.getElementById('email').value;
  const password = document.getElementById('password').value;
  const confirmPassword = document.getElementById('confirmPassword').value;

  let valid = true;

  document.querySelectorAll('.error').forEach(el => el.remove());

  if (username.length < 3) {

```

```

    valid = false;

    showError('username', 'Username must be at least 3 characters long.');
```

}

```

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(email)) {

    valid = false;

    showError('email', 'Please enter a valid email address.');
```

}

```

if (password.length < 6) {

    valid = false;

    showError('password', 'Password must be at least 6 characters long.');
```

}

```

if (password !== confirmPassword) {

    valid = false;

    showError('confirmPassword', 'Passwords do not match.');
```

}


```

if (valid) {

    alert('Registration successful!');
```

}

```

});
```



```

function showError(inputId, message) {

    const inputElement = document.getElementById(inputId);

    const errorElement = document.createElement('div');

    errorElement.className = 'error text-danger';

    errorElement.textContent = message;

    inputElement.parentNode.appendChild(errorElement);

}
```



```

document.getElementById('loginForm').addEventListener('submit', function(event) {
```

```

event.preventDefault();

const email = document.getElementById('loginEmail').value;
const password = document.getElementById('loginPassword').value;
let valid = true;

document.querySelectorAll('.error').forEach(el => el.remove());

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
if (!emailRegex.test(email)) {
    valid = false;
    showError('loginEmail', 'Please enter a valid email address.');
```

```

}

if (password.length < 6) {
    valid = false;
    showError('loginPassword', 'Password must be at least 6 characters long.');
```

```

}

if (valid) {
    alert('Login successful!');
}

});

function showError(inputId, message) {
    const inputElement = document.getElementById(inputId);
    const errorElement = document.createElement('div');
    errorElement.className = 'error text-danger';
    errorElement.textContent = message;
    inputElement.parentNode.appendChild(errorElement);
}

</script>

```