**Assignment 1:** Implement a navigation bar using an unordered list with links to different sections of your HTML page. Use CSS to style the list as a horizontal menu and highlight the current page or section.

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Navigation Bar Example</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<nav>

<ul class="navbar">

<li class="nav-item"><a href="#home" class="nav-link">Home</a></li>

<li class="nav-item"><a href="#about" class="nav-link">About</a></li>

<li class="nav-item"><a href="#services" class="nav-link">Services</a></li>

<li class="nav-item"><a href="#contact" class="nav-link">Contact</a></li>

</ul>

</nav>

<section id="home">

<h1>Home</h1>

<p>Welcome to the home page.</p>

</section>

<section id="about">

<h1>About</h1>

<p>Learn more about us on this page.</p>

</section>

<section id="services">

<h1>Services</h1>

<p>Discover our services here.</p>

</section>

<section id="contact">

<h1>Contact</h1>

<p>Get in touch with us.</p>

</section>

</body>

</html>

**CSS**

body {

font-family: Arial, sans-serif;

}

nav {

background-color: #333;

}

.navbar {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

display: flex;

justify-content: center;

}

.nav-item {

float: left;

}

.nav-link {

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

.nav-link:hover {

background-color: #111;

}

.nav-link.active {

background-color: #4CAF50;

}

### **Explanation**

* **HTML**: The navigation bar is created using an unordered list inside a nav element. Each list item contains a link to a section of the page.
* **CSS**: The CSS styles the unordered list as a horizontal menu. The .nav-link.active class is used to highlight the current page or section.

—----------------------------------------------------------------------------------------------------------------------------

**Assignment 2:** Create a simple HTML page that includes the use of headings, paragraphs, and at least two semantic tags like <article> or <section>. Add a table with data of your choice and a form with fields for a user's name, email, and a submit button.

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Simple HTML Page</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header, section, article, footer {

padding: 20px;

margin: 10px;

}

header {

background-color: #f4f4f4;

}

table {

width: 100%;

border-collapse: collapse;

margin: 20px 0;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

form {

margin: 20px 0;

}

form div {

margin-bottom: 10px;

}

form label {

display: block;

margin-bottom: 5px;

}

form input[type="text"], form input[type="email"] {

width: 100%;

padding: 8px;

box-sizing: border-box;

}

form input[type="submit"] {

padding: 10px 20px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

form input[type="submit"]:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<header>

<h1>Welcome to My Simple HTML Page</h1>

<p>This page includes various HTML elements for demonstration purposes.</p>

</header>

<section>

<h2>About This Page</h2>

<p>This section provides information about the content and structure of the page.</p>

</section>

<article>

<h2>Sample Article</h2>

<p>This is a sample article to show the use of the <code>&lt;article&gt;</code> semantic tag. Articles are used to represent a self-contained piece of content that could be distributed independently.</p>

</article>

<section>

<h2>Sample Data Table</h2>

<table>

<tr>

<th>Name</th>

<th>Age</th>

<th>City</th>

</tr>

<tr>

<td>John Doe</td>

<td>30</td>

<td>New York</td>

</tr>

<tr>

<td>Jane Smith</td>

<td>25</td>

<td>Los Angeles</td>

</tr>

<tr>

<td>Sam Johnson</td>

<td>35</td>

<td>Chicago</td>

</tr>

</table>

</section>

<section>

<h2>Contact Form</h2>

<form action="#" method="post">

<div>

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

</div>

<div>

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

<input type="email" id="email" name="email" required>

</div>

<div>

<input type="submit" value="Submit">

</div>

</form>

</section>

<footer>

<p>&copy; 2024 My Simple HTML Page</p>

</footer>

</body>

</html>

### **Explanation**

* **Header**: Contains the main title of the page and an introductory paragraph.
* **Section "About This Page"**: Provides information about the page.
* **Article "Sample Article"**: Demonstrates the use of the <article> tag.
* **Section "Sample Data Table"**: Includes a table with sample data.
* **Section "Contact Form"**: Contains a form with fields for a user's name, email, and a submit button.
* **Footer**: Provides a footer with a copyright notice.

This simple HTML page illustrates the use of various HTML elements, including headings, paragraphs, semantic tags, a table, and a form

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 3:** Apply the CSS box model to ensure that your page content has appropriate margins and padding. Create a layout using div tags and style them to arrange content in a multi-column format using floats or flexbox.

**HTML**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

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

**<title>Multi-Column Layout</title>**

**<link rel="stylesheet" href="styles.css">**

**</head>**

**<body>**

**<header>**

**<h1>Welcome to My Multi-Column Layout Page</h1>**

**<p>This page uses the CSS box model and flexbox to create a responsive layout.</p>**

**</header>**

**<div class="container">**

**<nav class="sidebar">**

**<h2>Navigation</h2>**

**<ul>**

**<li><a href="#home">Home</a></li>**

**<li><a href="#about">About</a></li>**

**<li><a href="#services">Services</a></li>**

**<li><a href="#contact">Contact</a></li>**

**</ul>**

**</nav>**

**<main class="content">**

**<section id="home">**

**<h2>Home</h2>**

**<p>Welcome to the home page.</p>**

**</section>**

**<section id="about">**

**<h2>About</h2>**

**<p>Learn more about us on this page.</p>**

**</section>**

**<section id="services">**

**<h2>Services</h2>**

**<p>Discover our services here.</p>**

**<article>**

**<h2>Sample Article</h2>**

**<p>This is a sample article to show the use of the <code>&lt;article&gt;</code> semantic tag. Articles are used to represent a self-contained piece of content that could be distributed independently.</p>**

**</article>**

**<section>**

**<h2>Sample Data Table</h2>**

**<table>**

**<tr>**

**<th>Name</th>**

**<th>Age</th>**

**<th>City</th>**

**</tr>**

**<tr>**

**<td>John Doe</td>**

**<td>30</td>**

**<td>New York</td>**

**</tr>**

**<tr>**

**<td>Jane Smith</td>**

**<td>25</td>**

**<td>Los Angeles</td>**

**</tr>**

**<tr>**

**<td>Sam Johnson</td>**

**<td>35</td>**

**<td>Chicago</td>**

**</tr>**

**</table>**

**</section>**

**<section>**

**<h2>Contact Form</h2>**

**<form action="#" method="post">**

**<div>**

**<label for="name">Name:</label>**

**<input type="text" id="name" name="name" required>**

**</div>**

**<div>**

**<label for="email">Email:</label>**

**<input type="email" id="email" name="email" required>**

**</div>**

**<div>**

**<input type="submit" value="Submit">**

**</div>**

**</form>**

**</section>**

**</section>**

**</main>**

**<aside class="extra">**

**<h2>Extra Content</h2>**

**<p>This section can be used for additional content like advertisements or links.</p>**

**</aside>**

**</div>**

**<footer>**

**<p>&copy; 2024 My Multi-Column Layout Page</p>**

**</footer>**

**</body>**

**</html>**

**CSS**

/\* Apply CSS box model to all elements \*/

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header, footer {

background-color: #f4f4f4;

padding: 20px;

text-align: center;

}

.container {

display: flex;

flex-wrap: wrap;

margin: 20px;

}

/\* Sidebar \*/

.sidebar {

flex: 1 1 20%;

padding: 20px;

background-color: #e2e2e2;

}

.sidebar h2, .sidebar ul {

margin-bottom: 10px;

}

.sidebar ul {

list-style-type: none;

padding-left: 0;

}

.sidebar ul li {

margin-bottom: 10px;

}

.sidebar ul li a {

text-decoration: none;

color: #333;

}

/\* Main Content \*/

.content {

flex: 3 1 60%;

padding: 20px;

background-color: #fff;

}

.content h2 {

margin-bottom: 10px;

}

.content section, .content article {

margin-bottom: 20px;

}

/\* Extra Content \*/

.extra {

flex: 1 1 20%;

padding: 20px;

background-color: #e2e2e2;

}

.extra h2, .extra p {

margin-bottom: 10px;

}

/\* Table Styles \*/

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

/\* Form Styles \*/

form {

margin-bottom: 20px;

}

form div {

margin-bottom: 10px;

}

form label {

display: block;

margin-bottom: 5px;

}

form input[type="text"], form input[type="email"] {

width: 100%;

padding: 8px;

box-sizing: border-box;

}

form input[type="submit"] {

padding: 10px 20px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

form input[type="submit"]:hover {

background-color: #45a049;

}

footer {

padding: 10px;

background-color: #f4f4f4;

text-align: center;

}

### **Explanation**

* **CSS Box Model**: Applied universally using \* { box-sizing: border-box; margin: 0; padding: 0; }.
* **Flexbox Layout**: The .container uses flexbox to create a responsive layout with three columns: a sidebar, main content, and an extra content section.
* **Responsive Design**: The flex properties ensure that the layout adjusts based on screen size and available space.
* **Semantic HTML**: Proper use of semantic tags (<section>, <article>, <nav>, <aside>) improves the readability and structure of the HTML document.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 4:** Enhance the page by adding CSS3 animations to the menu and form elements. Also, use media queries to make the page responsive, ensuring it looks good on both desktop and mobile screen sizes.

/\* Apply CSS box model to all elements \*/

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header, footer {

background-color: #f4f4f4;

padding: 20px;

text-align: center;

}

.container {

display: flex;

flex-wrap: wrap;

margin: 20px;

}

/\* Sidebar \*/

.sidebar {

flex: 1 1 20%;

padding: 20px;

background-color: #e2e2e2;

animation: slideIn 1s ease-in-out;

}

.sidebar h2, .sidebar ul {

margin-bottom: 10px;

}

.sidebar ul {

list-style-type: none;

padding-left: 0;

}

.sidebar ul li {

margin-bottom: 10px;

}

.sidebar ul li a {

text-decoration: none;

color: #333;

display: block;

padding: 10px;

transition: background-color 0.3s;

}

.sidebar ul li a:hover {

background-color: #ddd;

}

/\* Main Content \*/

.content {

flex: 3 1 60%;

padding: 20px;

background-color: #fff;

}

.content h2 {

margin-bottom: 10px;

}

.content section, .content article {

margin-bottom: 20px;

}

/\* Extra Content \*/

.extra {

flex: 1 1 20%;

padding: 20px;

background-color: #e2e2e2;

}

.extra h2, .extra p {

margin-bottom: 10px;

}

/\* Table Styles \*/

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

/\* Form Styles \*/

form {

margin-bottom: 20px;

animation: fadeIn 1.5s ease-in-out;

}

form div {

margin-bottom: 10px;

}

form label {

display: block;

margin-bottom: 5px;

}

form input[type="text"], form input[type="email"] {

width: 100%;

padding: 8px;

box-sizing: border-box;

transition: border-color 0.3s;

}

form input[type="text"]:focus, form input[type="email"]:focus {

border-color: #4CAF50;

}

form input[type="submit"] {

padding: 10px 20px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

transition: background-color 0.3s;

}

form input[type="submit"]:hover {

background-color: #45a049;

}

footer {

padding: 10px;

background-color: #f4f4f4;

text-align: center;

}

/\* CSS Animations \*/

@keyframes slideIn {

from {

opacity: 0;

transform: translateX(-100%);

}

to {

opacity: 1;

transform: translateX(0);

}

}

@keyframes fadeIn {

from {

opacity: 0;

}

to {

opacity: 1;

}

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.container {

flex-direction: column;

}

.sidebar, .content, .extra {

flex: 1 1 100%;

margin-bottom: 20px;

}

.sidebar ul li {

text-align: center;

}

}

@media (max-width: 480px) {

header, footer {

padding: 10px;

}

.sidebar ul li a {

padding: 8px;

}

form input[type="text"], form input[type="email"], form input[type="submit"] {

padding: 6px;

}

}

### **Explanation**

1. **Animations**:
   * The slideIn animation is applied to the sidebar to create a sliding effect when it appears.
   * The fadeIn animation is applied to the form to create a fading effect when it appears.
2. **Responsive Design**:
   * Media queries are used to adjust the layout for different screen sizes.
   * For screens with a maximum width of 768px, the flex container's direction changes to column, making the layout single-column and stacking the sidebar, content, and extra content sections vertically.
   * For screens with a maximum width of 480px, padding for the header, footer, sidebar links, and form inputs is reduced to optimize the layout for small screens.

This updated CSS ensures that the page is both visually appealing with animations and responsive for various screen sizes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 5:** Take the static HTML page from Day 1 and integrate Bootstrap. Refactor the navigation bar, table, and form to use Bootstrap components. Ensure the page is responsive using Bootstrap's grid system."

**HTML**

Integrating Bootstrap into the HTML page will enhance its design and responsiveness with minimal effort. Below is the refactored HTML page using Bootstrap components.

### **HTML with Bootstrap Integration**

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Bootstrap Integrated Page</title>

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

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<a class="navbar-brand" href="#">My Page</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item active">

<a class="nav-link" href="#home">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#about">About</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#services">Services</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#contact">Contact</a>

</li>

</ul>

</div>

</nav>

<div class="container mt-5">

<div class="row">

<div class="col-md-3">

<div class="bg-light p-3">

<h2>Navigation</h2>

<ul class="list-group">

<li class="list-group-item"><a href="#home">Home</a></li>

<li class="list-group-item"><a href="#about">About</a></li>

<li class="list-group-item"><a href="#services">Services</a></li>

<li class="list-group-item"><a href="#contact">Contact</a></li>

</ul>

</div>

</div>

<div class="col-md-6">

<section id="home" class="mb-5">

<h2>Home</h2>

<p>Welcome to the home page.</p>

</section>

<section id="about" class="mb-5">

<h2>About</h2>

<p>Learn more about us on this page.</p>

</section>

<section id="services" class="mb-5">

<h2>Services</h2>

<p>Discover our services here.</p>

<article class="mb-5">

<h2>Sample Article</h2>

<p>This is a sample article to show the use of the <code>&lt;article&gt;</code> semantic tag. Articles are used to represent a self-contained piece of content that could be distributed independently.</p>

</article>

<section class="mb-5">

<h2>Sample Data Table</h2>

<table class="table table-striped">

<thead>

<tr>

<th>Name</th>

<th>Age</th>

<th>City</th>

</tr>

</thead>

<tbody>

<tr>

<td>John Doe</td>

<td>30</td>

<td>New York</td>

</tr>

<tr>

<td>Jane Smith</td>

<td>25</td>

<td>Los Angeles</td>

</tr>

<tr>

<td>Sam Johnson</td>

<td>35</td>

<td>Chicago</td>

</tr>

</tbody>

</table>

</section>

<section>

<h2>Contact Form</h2>

<form>

<div class="form-group">

<label for="name">Name:</label>

<input type="text" class="form-control" id="name" name="name" required>

</div>

<div class="form-group">

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

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

</div>

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

</form>

</section>

</section>

</div>

<div class="col-md-3">

<div class="bg-light p-3">

<h2>Extra Content</h2>

<p>This section can be used for additional content like advertisements or links.</p>

</div>

</div>

</div>

</div>

<footer class="bg-dark text-white text-center p-3">

<p>&copy; 2024 My Multi-Column Layout Page</p>

</footer>

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

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"></script>

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

</body>

</html>

### **Explanation**

1. **Bootstrap Integration**: The HTML page includes the Bootstrap CSS and JavaScript libraries from a CDN.
2. **Navigation Bar**: The navbar uses Bootstrap's navbar, navbar-expand-lg, navbar-dark, and bg-dark classes to create a responsive, dark-themed navigation bar.
3. **Grid System**: The content is organized using Bootstrap's grid system with container, row, and col-md-\* classes.
4. **Sidebar and Extra Content**: These sections are placed in col-md-3 divs to create a three-column layout.
5. **Table**: The table is styled with Bootstrap's table and table-striped classes.
6. **Form**: The form uses Bootstrap's form-group, form-control, and btn btn-primary classes to style the form elements.
7. **Responsive Design**: The Bootstrap grid system ensures the page is responsive and looks good on both desktop and mobile screens.

This refactored page leverages Bootstrap components to create a responsive, modern design with minimal custom CSS.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Day 20**

**Assignment 1:** Create a JavaScript object representing a user with properties for name, email, and age. Write functions that manipulate this object, such as changing the name, updating the email, and calculating the user's birth year.

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Info</title>

<script src="script.js" defer></script>

</head>

<body>

<h1>User Information</h1>

<p>Name: <span id="name">John Doe</span></p>

<p>Email: <span id="email">john.doe@example.com</span></p>

<p id="birthYear">Birth Year: 1994</p>

<h2>Update User Info</h2>

<label for="newName">New Name:</label>

<input type="text" id="newName"><br>

<label for="newEmail">New Email:</label>

<input type="text" id="newEmail"><br>

<button onclick="updateUserInfo()">Update Info</button>

</body>

</html>

JS

// Define the user object

const user = {

name: 'John Doe',

email: 'john.doe@example.com',

age: 30,

// Function to change the name

changeName: function(newName) {

this.name = newName;

document.getElementById('name').innerText = this.name;

},

// Function to update the email

updateEmail: function(newEmail) {

this.email = newEmail;

document.getElementById('email').innerText = this.email;

},

// Function to calculate the birth year

getBirthYear: function() {

const currentYear = new Date().getFullYear();

return currentYear - this.age;

}

};

function updateUserInfo() {

const newName = document.getElementById('newName').value;

const newEmail = document.getElementById('newEmail').value;

user.changeName(newName);

user.updateEmail(newEmail);

document.getElementById('birthYear').innerText = `Birth Year: ${user.getBirthYear()}`;

}

### **Explanation:**

* **HTML File (index.html)**: This file contains the structure and content of the webpage. The <script src="script.js" defer></script> tag includes the JavaScript file and ensures that it loads after the HTML content is fully parsed.
* **JavaScript File (script.js)**: This file contains the JavaScript code for manipulating the user object and handling interactions.

Make sure both files are in the same directory, or adjust the src path in the <script> tag accordingly if they are in different locations.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Assignment 2:** Use regular expressions in JavaScript to validate the email address entered in the form. It should check for the correct format of the email and display a message to the user if the format is incorrect."

To validate the email address using regular expressions in JavaScript, you can enhance the updateUserInfo function to include email validation. If the email format is incorrect, you can display an error message to the user.

Here's how you can update the JavaScript file to include email validation:

// Define the user object

const user = {

name: 'John Doe',

email: 'john.doe@example.com',

age: 30,

// Function to change the name

changeName: function(newName) {

this.name = newName;

document.getElementById('name').innerText = this.name;

},

// Function to update the email

updateEmail: function(newEmail) {

this.email = newEmail;

document.getElementById('email').innerText = this.email;

},

// Function to calculate the birth year

getBirthYear: function() {

const currentYear = new Date().getFullYear();

return currentYear - this.age;

}

};

// Function to validate email address using regex

function validateEmail(email) {

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

return emailRegex.test(email);

}

function updateUserInfo() {

const newName = document.getElementById('newName').value;

const newEmail = document.getElementById('newEmail').value;

// Validate email

if (!validateEmail(newEmail)) {

alert('Please enter a valid email address.');

return; // Exit the function if email is invalid

}

user.changeName(newName);

user.updateEmail(newEmail);

document.getElementById('birthYear').innerText = `Birth Year: ${user.getBirthYear()}`;

}

### **Explanation:**

* **validateEmail(email)**: This function uses a regular expression to check if the provided email address is in a valid format. The regex ^[^\s@]+@[^\s@]+\.[^\s@]+$ checks for a typical email pattern.
* **updateUserInfo()**: This function now includes a check to see if the email is valid before proceeding. If the email is not valid, it displays an alert message and exits the function.

With this setup, users will be prompted with an alert if they enter an invalid email address, and the email will only be updated if it passes validation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_