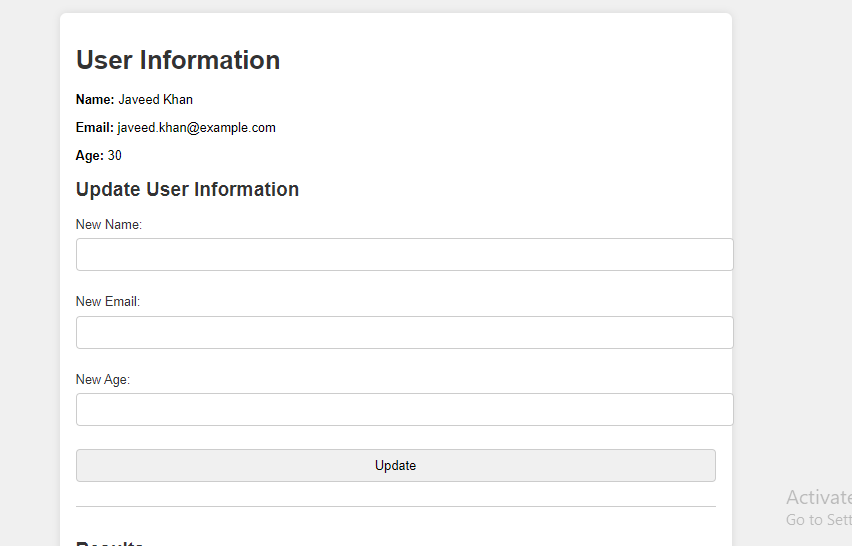
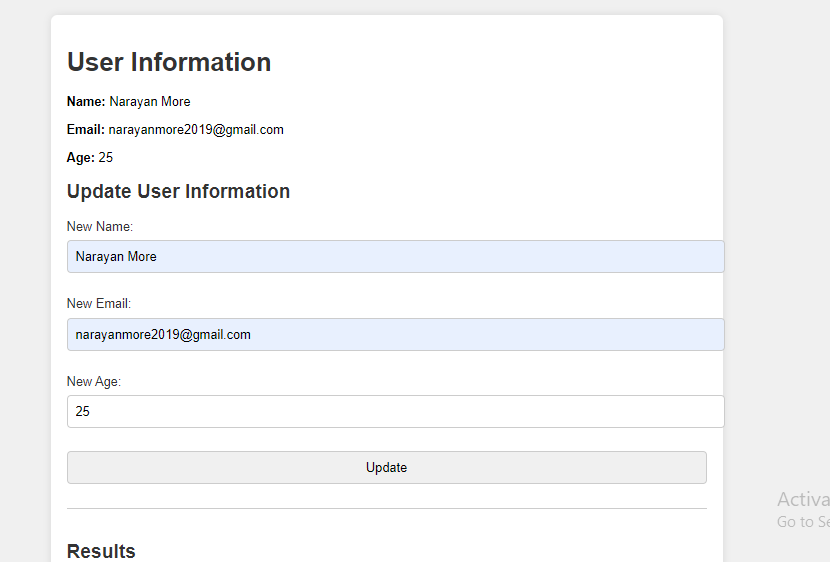
**Assignment 1: Create a JavaScript object representing a user with properties for name, email, and age. Write functions that manipulate this object, such as changing the name, updating the email, and calculating the user's birth year.**

****

**After updating info:**

****

let user = {

    name: "Javeed Khan",

    email: "javeed.khan@example.com",

    age: 30

};

function updateUserInfo(event) {

    event.preventDefault();

    let newName = document.getElementById("newName").value;

    let newEmail = document.getElementById("newEmail").value;

    let newAge = parseInt(document.getElementById("newAge").value);

    user.name = newName;

    user.email = newEmail;

    user.age = newAge;

    document.getElementById("userName").textContent = user.name;

    document.getElementById("userEmail").textContent = user.email;

    document.getElementById("userAge").textContent = user.age;

**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.**

 HTML Structure:

* The HTML file (index.html) contains a form with an input field for the email and a submit button.
* There is a paragraph (<p>) element (error-msg) to display error messages to the user.

 CSS Styling:

* CSS file (styles.css) provides basic styling to structure and format the content, ensuring readability and visual appeal.

 JavaScript:

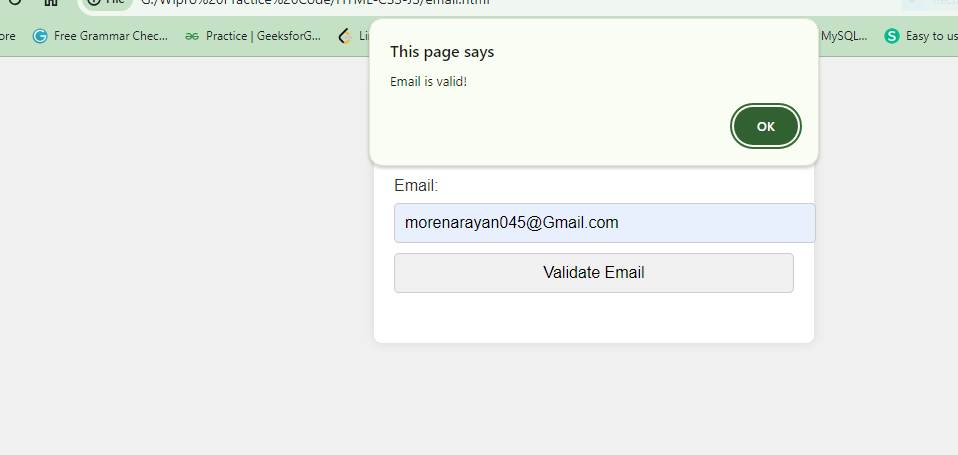
* JavaScript file (script.js) defines two main functions:
  + validateEmail(email): Uses a regular expression (pattern) to check if the email input matches the correct format.
  + handleSubmit(event): Handles form submission, prevents default behavior, validates the email using validateEmail, and displays an error message if the email format is incorrect.
* Event listeners ensure that user interactions (form submissions) trigger appropriate validation and messaging.

 Regular Expression:

* The regular expression ^[^\s@]+@[^\s@]+\.[^\s@]+$ checks for the basic structure of an email address:
  + ^[^\s@]+: Start of the string should have one or more characters that are not whitespace or '@'.
  + @[^\s@]+: Followed by '@' and then one or more characters that are not whitespace or '@'.
  + \.[^\s@]+$: Followed by '.' and then one or more characters that are not whitespace or '@', ending at the end of the string.

 Validation and Error Handling:

* If the email format is valid, it alerts the user with a message.
* If not, it displays an error message below the form input



unction validateEmail(email) {

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

    return pattern.test(email);

}

function handleSubmit(event) {

    event.preventDefault();

    const emailInput = document.getElementById("email");

    const errorMsg = document.getElementById("error-msg");

    const email = emailInput.value.trim();

    if (validateEmail(email)) {

        errorMsg.textContent = ""; // Clear error message if email is valid

        alert("Email is valid!"); // Replace with your validation success handling

    } else {

        errorMsg.textContent = "Please enter a valid email address.";

    }

}

// Add event listener to form submission

document.getElementById("emailForm").addEventListener("submit", handleSubmit);