Lab Assingment-14

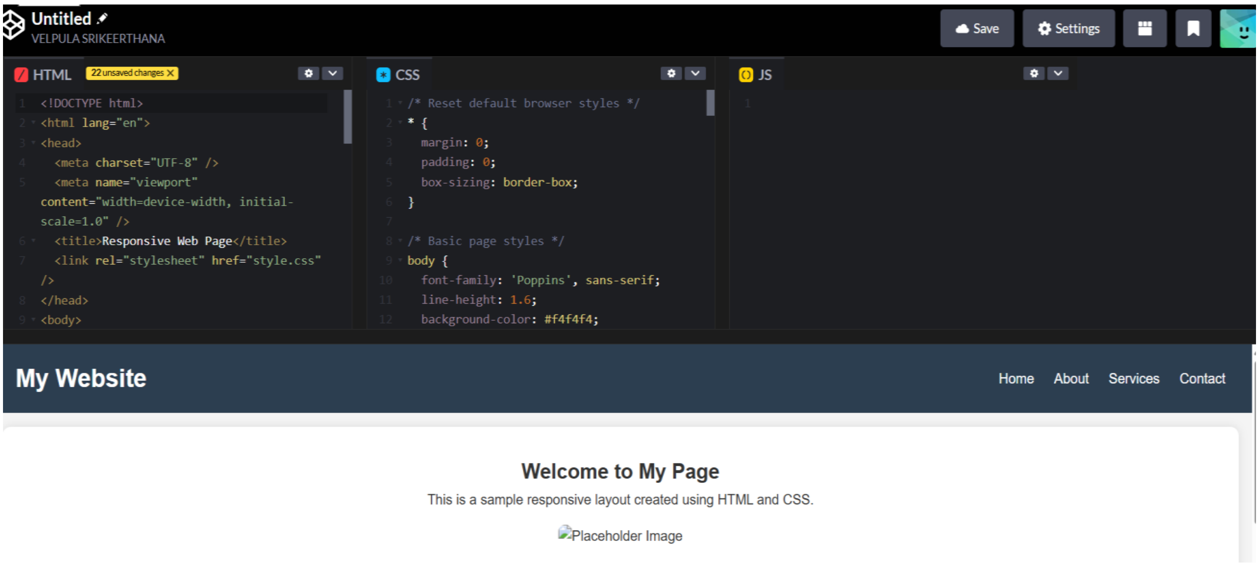
Task 1: Create a Responsive Web Page Layout

Instructions:

- Design a basic web page layout with a **header, main content area, and footer** using HTML and CSS.
- Use AI to assist in generating **responsive CSS** for different screen sizes.
- Ensure the layout is clean and visually organized.

Expected Output:

- A responsive web page with:
  - Header with navigation links
  - Main section with placeholder text/images
  - Footer with copyright or contact info
- Layout adapts correctly to desktop, tablet, and mobile screen sizes.



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0" />
  <title>Responsive Web Page</title>
  <link rel="stylesheet" href="style.css"
/>
</head>
<body>
```

```css
/* Reset default browser styles */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Basic page styles */
body {
  font-family: 'Poppins', sans-serif;
  line-height: 1.6;
  background-color: #f4f4f4;
```

Welcome to My Page

This is a sample responsive layout created using HTML and CSS.

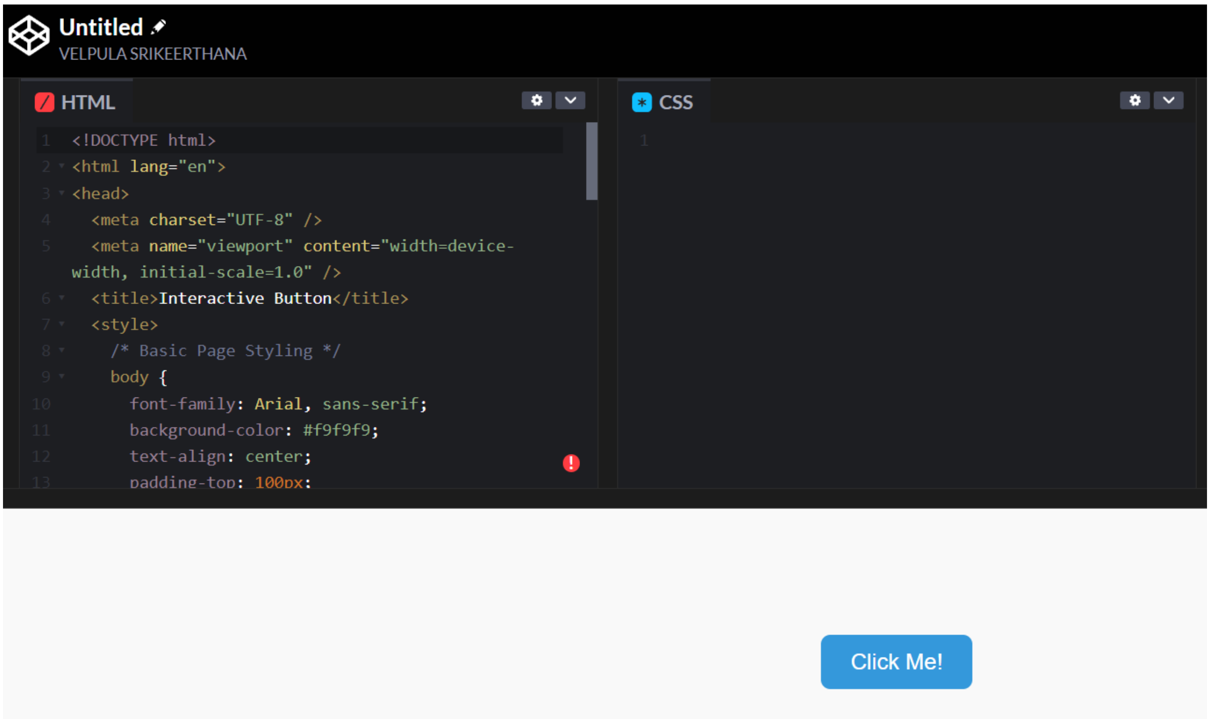
Task 2: Interactive Button with JavaScript

Instructions:

- Create a **button** on a web page.
- Use AI to generate JavaScript code that displays an **alert message** when the button is clicked.
- Ensure the code is clean and well-commented.

Expected Output:

- A web page with a button.
- Clicking the button shows a pop-up alert message, e.g., “Button clicked!”.



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-
width, initial-scale=1.0" />
  <title>Interactive Button</title>
  <style>
    /* Basic Page Styling */
    body {
      font-family: Arial, sans-serif;
      background-color: #f9f9f9;
      text-align: center;
      padding-top: 100px;
```

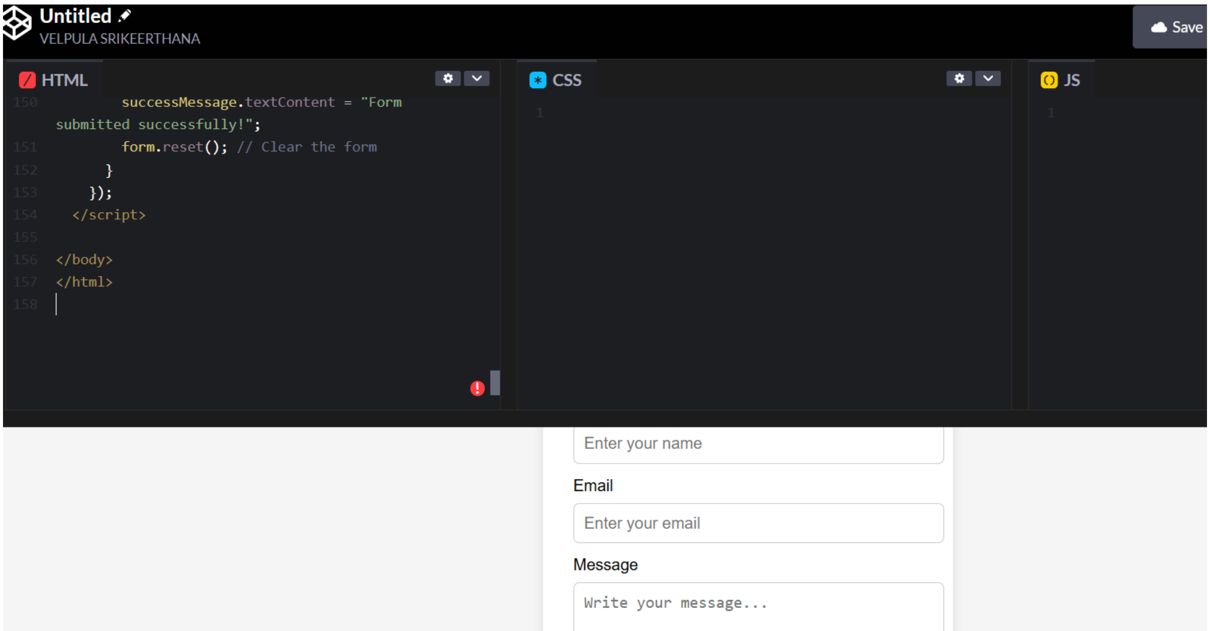
Task 3: Form with Validation

Instructions:

- Design a **contact form** with fields: Name, Email, Message.
- Use AI to generate JavaScript code for **form validation** (e.g., non-empty fields, valid email format).
- Add **inline error messages** if input is invalid.

Expected Output:

- A functional contact form where:
  - Submitting with empty fields shows error messages.
  - Submitting with valid input displays a “Form submitted successfully” message.



```
        successMessage.textContent = "Form
submitted successfully!";
        form.reset(); // Clear the form
      }
    });
  </script>

</body>
</html>
```

Enter your name

Email

Enter your email

Message

Write your message...

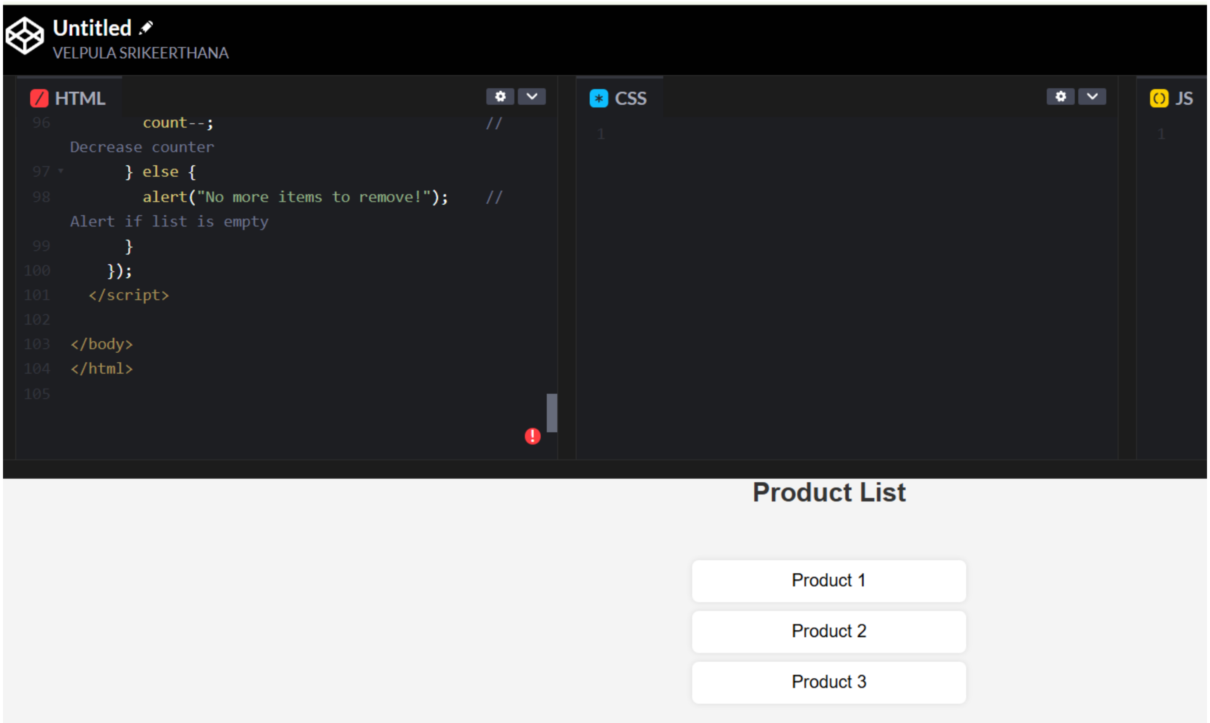
Task 4: Dynamic Content Generation

Instructions:

- Create a list of items (e.g., product names) using **HTML**.
- Use AI-generated JavaScript to **dynamically add or remove items** from the list when a button is clicked.

Expected Output:

- A web page with a list and buttons:
  - "Add Item" adds a new item to the list dynamically.
  - "Remove Item" deletes the last item.
- Updates occur without reloading the page.



```
            count--;                              //
    Decrease counter
        } else {
            alert("No more items to remove!");    //
    Alert if list is empty
        }
    });
  </script>

</body>
</html>
```

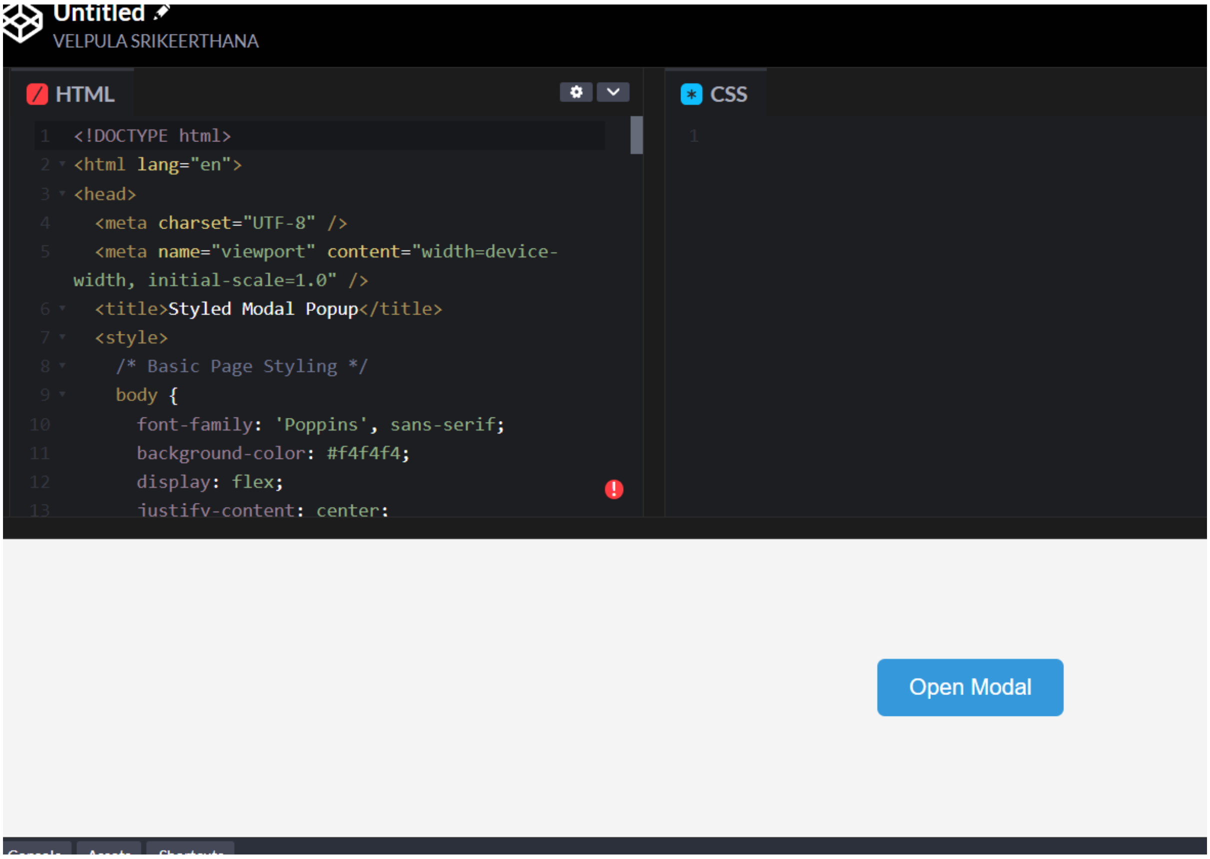
Task 5: Styled Modal Popup

Instructions:

- Use AI to generate a modal popup that opens when a button is clicked.
- Style the modal using CSS with a semi-transparent overlay.
- Include a close button that hides the modal.

Expected Output:

- A web page with a button labeled “Open Modal”.
- Clicking the button displays the modal popup with content.
- Modal can be closed by clicking the “X” button or overlay area.



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Styled Modal Popup</title>
  <style>
    /* Basic Page Styling */
    body {
      font-family: 'Poppins', sans-serif;
      background-color: #f4f4f4;
      display: flex;
      justify-content: center;
```