

AI ASSISTED CODING.

LAB ASSIGNMENT -14.3

Program	: B-TECH
Specialization	: AIML
Name of Student	: AKULA MANOJ
Enrollment-No	: 2403A52031
Batch No	: 02
Date	: 22-10-2025

TASK DESCRIPTION-1

Ask AI to generate a simple HTML homepage for a "Student Info Portal" with a header, navigation menu, and footer

PROMPT:-

GENERATE A SIMPLE HTML homepage for a "Student Info Portal" with a header, navigation menu, and footer.

Expected Output:

- HTML code with <header>, <nav>, <footer>.
- Clean indentation, proper tags, and comments

GENERATED CODE:

```
AIAC-14.3-Task-1.html > html > head > style > footer
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Student Info Portal</title>
7   <style>
8     /* Basic CSS for layout */
9     body {
10       margin: 0;
11       font-family: Arial, sans-serif;
12     }
13     header {
14       background-color: #2c3e50;
15       color: white;
16       text-align: center;
17       padding: 1rem;
18     }
19     nav {
20       background-color: #34495e;
21       padding: 1rem;
22     }
23     nav a {
24       color: white;
25       text-decoration: none;
26       margin: 0 1rem;
27     }
28     footer {
29       background-color: #2c3e50;
30       color: white;
31       text-align: center;
32       padding: 1rem;
33       position: fixed;
34       bottom: 0;
35       width: 100%;
36     }
37   </style>
```

```

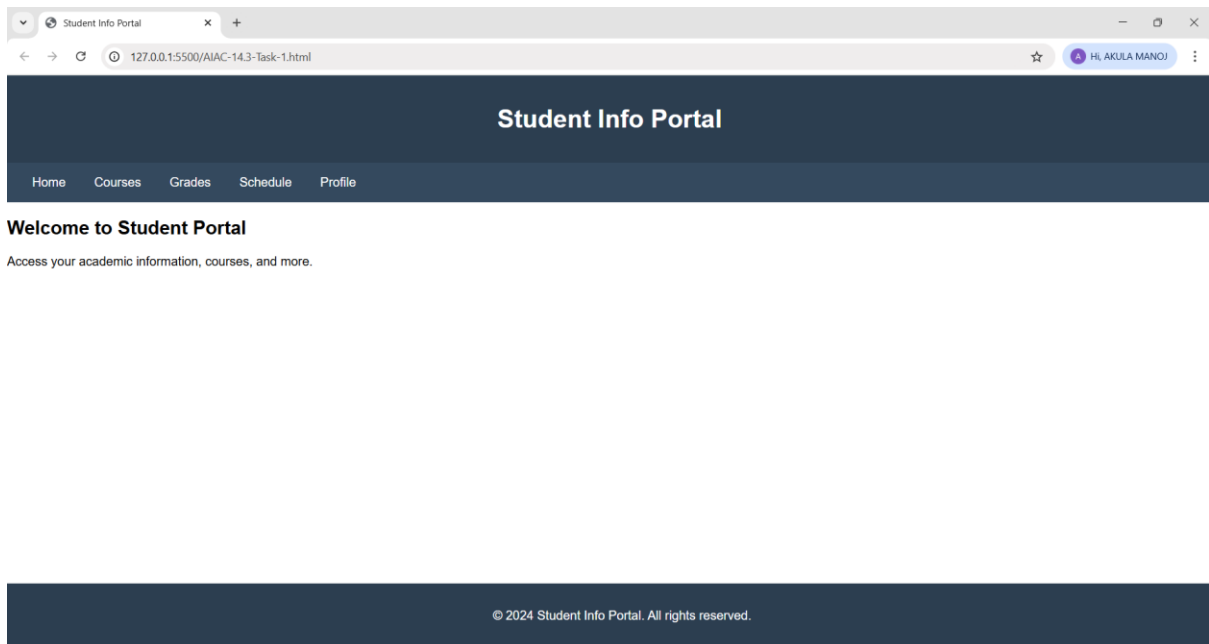
    }
  </style>
</head>
<body>
  <!-- Header Section -->
  <header>
    <h1>Student Info Portal</h1>
  </header>

  <!-- Navigation Menu -->
  <nav>
    <a href="#home">Home</a>
    <a href="#courses">Courses</a>
    <a href="#grades">Grades</a>
    <a href="#schedule">Schedule</a>
    <a href="#profile">Profile</a>
  </nav>

  <!-- Main Content Area -->
  <main>
    <h2>Welcome to Student Portal</h2>
    <p>Access your academic information, courses, and more.</p>
  </main>

  <!-- Footer Section -->
  <footer>
    <p>&copy; 2024 Student Info Portal. All rights reserved.</p>
  </footer>
</body>
</html>
```

OUTPUT:-



TASK -2

CSS Styling

PROMPT:-

CSS Styling add CSS styling to homepage for:

Responsive navigation bar.

Centered content section.

Footer with light gray background.

Expected Output:

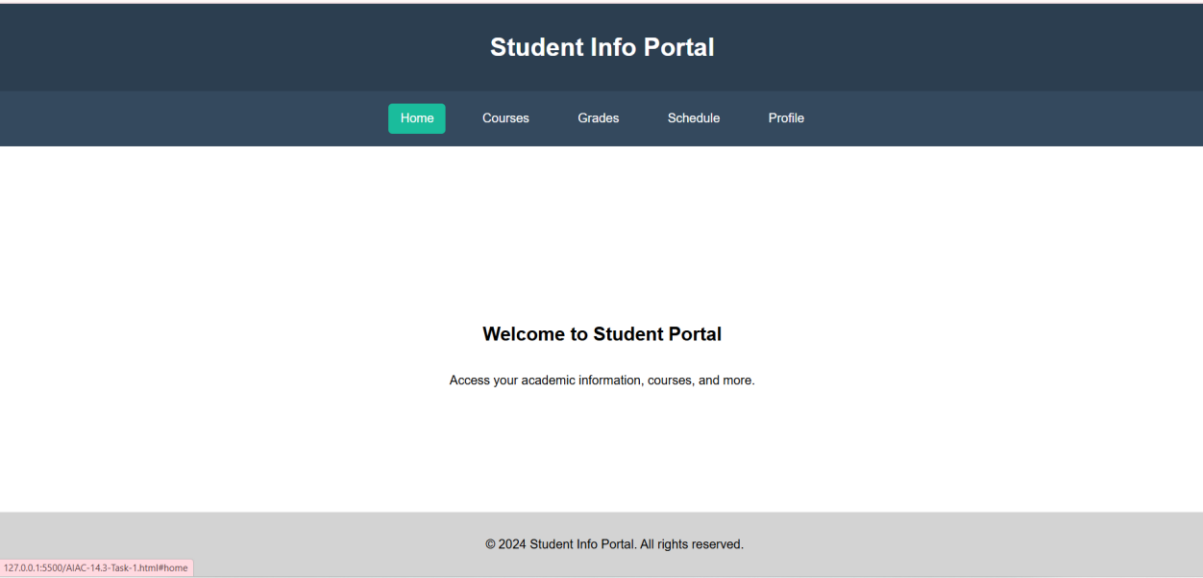
HTML + CSS combined.APPLY CLEAN CSS AND MAKE IT MORE ATTRACTIVE

GENERATED CODE:-

ENHANCED CODE:

```
3 </html>
4
5 <style>
6   /* Enhanced CSS for a more attractive layout */
7
8   nav {
9     display: flex;
10    justify-content: center;
11    flex-wrap: wrap;
12  }
13
14  nav a {
15    padding: 10px 15px;
16    border-radius: 5px;
17    transition: background-color 0.3s;
18  }
19
20  nav a:hover {
21    background-color: #1abc9c;
22  }
23
24  main {
25    display: flex;
26    flex-direction: column;
27    align-items: center;
28    justify-content: center;
29    height: calc(100vh - 200px); /* Adjust height based on header and footer */
30    text-align: center;
31  }
32
33  footer {
34    background-color: #d3d3d3; /* Light gray background */
35    color: black;
36  }
37 </style>
```

OUTPUT:-



TASK DESCRIPTION-3

PROMPT: generate a javascript for the html for functioning Working on submit JS validation.

Clear error messages if inputs are empty.

GENERATED CODE:

```

    footer {
      background-color: #d3d3d3; /* Light gray background */
      color: black;
    }
  }
</style>
<!-- Login overlay + JS to handle first-time login using localStorage -->
<div id="login-overlay" style="position:fixed;inset:0;display:flex;align-items:center;justify-content:center;background:rgba(0,0,0,0.5);z-index:1000;">
  <form id="login-form" style="background:#fff;padding:1.25rem;border-radius:8px;min-width:320px;box-shadow:0 4px 20px rgba(0,0,0,0.2);">
    <h3 style="margin:0 0 0.5rem 0;text-align:center;">Student Portal Login</h3>

    <label style="display:block;margin-bottom:0.5rem;">
      <span style="display:block;font-size:0.9rem;margin-bottom:0.25rem;">Username</span>
      <input id="username" name="username" type="text" style="width:100%;padding:0.5rem;border:1px solid #ccc;border-radius:4px;">
    </label>
    <div id="username-error" class="error" style="color:#e74c3c;font-size:0.85rem;height:18px;margin-bottom:0.5rem;"></div>

    <label style="display:block;margin-bottom:0.5rem;">
      <span style="display:block;font-size:0.9rem;margin-bottom:0.25rem;">Password</span>
      <input id="password" name="password" type="password" style="width:100%;padding:0.5rem;border:1px solid #ccc;border-radius:4px;">
    </label>
    <div id="password-error" class="error" style="color:#e74c3c;font-size:0.85rem;height:18px;margin-bottom:0.75rem;"></div>

    <div style="display:flex;justify-content:space-between;align-items:center;">
      <input id="remember" type="checkbox" style="margin-right:6px;">Remember me</div>
    <button type="submit" style="background:#2c3e50;color:#fff;padding:0.45rem 0.9rem;border:none;border-radius:4px;cursor:pointer;">Login</button>
  </form>
</div>

<script>
  // Simple "first login" flow using localStorage
  (function () {
    const overlay = document.getElementById('login-overlay');
    const form = document.getElementById('login-form');

    // Clear errors
    function clearErrors() {
      document.querySelectorAll('.error').forEach(e => e.textContent = '');
    }

    // Show welcome message
    function showWelcome(name) {
      main.innerHTML = `
        <h2>Welcome, ${escapeHtml(name)}!</h2>
        <p>You are now logged in to the Student Info Portal.</p>
        <button id="logout-btn" style="margin-top:1rem;padding:0.5rem 0.9rem;border:none;border-radius:4px;background:#e74c3c;color:#fff;cursor:pointer;">Logout</button>
      `;
      const logoutBtn = document.getElementById('logout-btn');
      logoutBtn.addEventListener('click', () => {
        localStorage.removeItem('studentPortalUser');
        location.reload();
      });
    }

    // Basic HTML escape to avoid injection if username used in DOM
    function escapeHtml(str) {

```

```

    }

    // Basic HTML escape to avoid injection if username used in DOM
    function escapeHtml(str) {
        return String(str).replace(/[<>"]/g, s => ({'&': '&amp;', '<': '&lt;', '>': '&gt;', '"': '&quot;', "'": '&#39;'}[s]));
    }

    // If user previously logged in and chose to remember, skip login
    const stored = localStorage.getItem('studentPortalUser');
    if (stored) {
        overlay.style.display = 'none';
        showWelcome(stored);
    } else {
        overlay.style.display = 'flex';
    }

    // Validate and "login"
    form.addEventListener('submit', function (e) {
        e.preventDefault();
        clearErrors();
        let valid = true;
        const username = usernameInput.value.trim();
        const password = passwordInput.value.trim();

        if (!username) {
            document.getElementById('username-error').textContent = 'Username is required';
            valid = false;
        }
        if (!password) {
            document.getElementById('password-error').textContent = 'Password is required';
            valid = false;
        }

        if (!valid) return;
    });

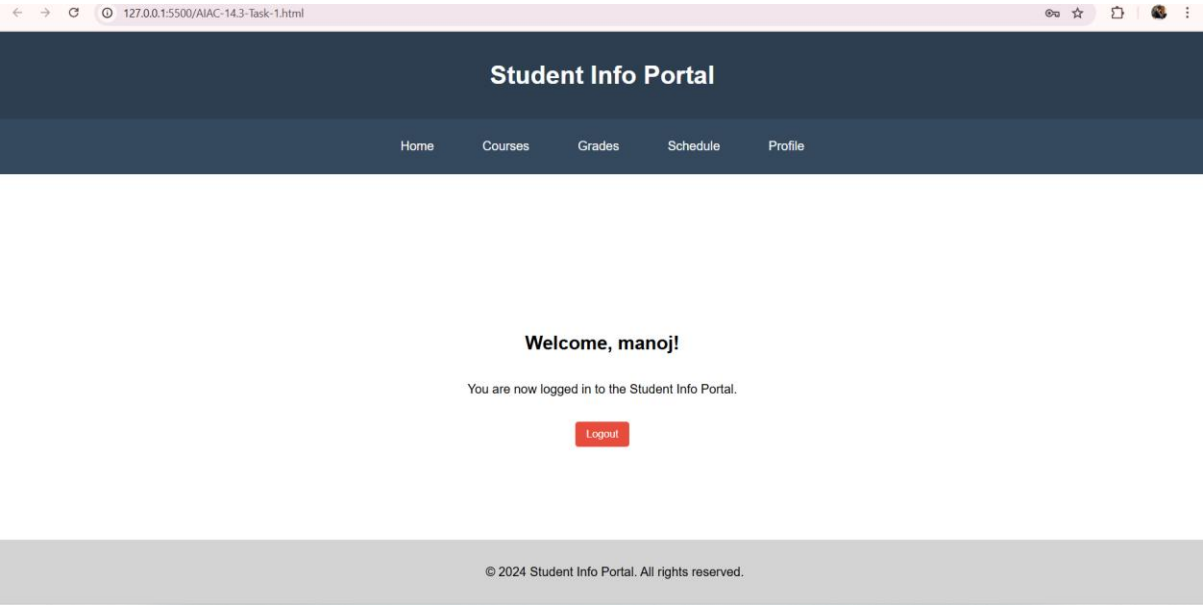
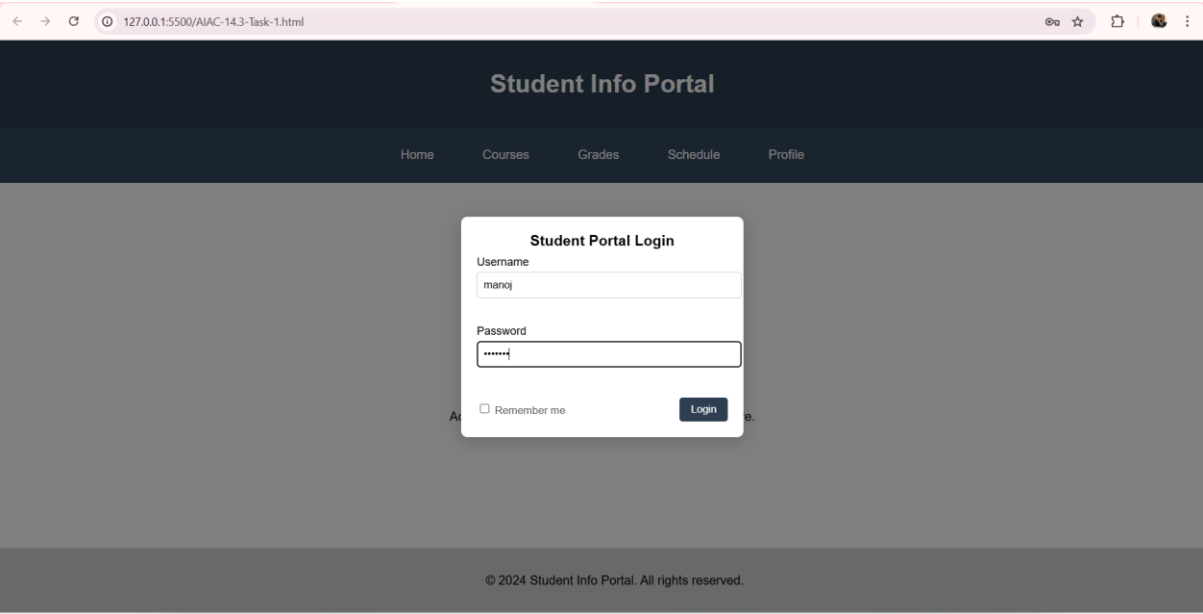
```

```

18 <script>
19     (function () {
20         form.addEventListener('submit', function (e) {
21             document.getElementById('username-error').textContent = 'Username is required';
22             valid = false;
23         });
24         if (!password) {
25             document.getElementById('password-error').textContent = 'Password is required';
26             valid = false;
27         }
28         if (!valid) return;
29
30         // NOTE: This is a demo. Do not use client-side auth for real apps.
31         // Simulate successful login and optionally remember the user.
32         if (rememberCheckbox.checked) {
33             localStorage.setItem('studentPortalUser', username);
34         } else {
35             // store in sessionStorage for this tab only
36             sessionStorage.setItem('studentPortalUser', username);
37         }
38
39         overlay.style.display = 'none';
40         showWelcome(username);
41     });
42
43     // Clear error when typing
44     [usernameInput, passwordInput].forEach(input => {
45         input.addEventListener('input', () => {
46             const err = document.getElementById(input.id + '-error');
47             if (err) err.textContent = '';
48         });
49     });
50
51 })();
52 </script>

```

OUTPUT:



THANK YOU