

CSS3– Day-3 Hands On – Srinu

Problem 1. Selectors and Box model Demo

1. Problem Statement

Assessment Goal: Check ability to apply styling and understand selectors and box model.

Hands-on Tasks:

1. Create an external CSS file and link it to HTML
2. Apply different text styles to headings and paragraphs
3. Add background color and borders to a section
4. Demonstrate margin and padding on a div
5. Style the form created on Day 2

Expected Outcome:

A visually styled webpage using external CSS and proper selectors.

2. Code

Index.html

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Student Portal - External CSS Demo</title>
    <!-- Requirement 1: External CSS -->
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <!-- Header -->
    <header class="header">
      <h1>Student Portal</h1>
      <p class="tagline">This is a demo of external CSS styling and validation form</p>
    </header>
    <!-- Requirement 2 & 3: Section -->
    <section class="content-section">
```

```
<h2 class="section-heading">Welcome to CSS Styling</h2>
<p id="highlight-text">
    This paragraph uses ID selector with special styling.
</p>
<p>
    This paragraph uses element selector. This section has background
    color and border applied using external CSS.
</p>
</section>
<!-- Requirement 4: Div with margin and padding -->
<div class="for-div">
    This DIV demonstrates margin, padding, border, and background color.
</div>
<!-- Requirement 5: Styled Form -->
<form class="form-style">
    <h2 class="form-heading">Registration Form</h2>
    <label>Full Name</label>
    <input type="text" placeholder="Enter your name" required>
    <label>Email Address</label>
    <input type="email" placeholder="Enter your email" required>
    <label>Password</label>
    <input type="password" placeholder="Enter password" required>
    <label>Gender</label>
    <div class="gender-group">
        <label>
            <input type="radio" name="gender" required>
            Male
        </label>
    </div>
</form>
```

```
<label>
    <input type="radio" name="gender" required>
    Female
</label>
</div>

<label>Select Course</label>
<select required>
    <option value="">Select Course</option>
    <option>HTML</option>
    <option>CSS</option>
    <option>JavaScript</option>
    <option>React</option>
</select>
<button type="submit">Register</button>
</form>
</body>
</html>
```

Style.css

```
/* Global Fix */
* {
    box-sizing: border-box;
}

/* Body */
body {
    font-family: Arial, sans-serif;
    background-color: #f4f6f8;
    margin: 0;
}
```

```
/* Header Styling */
.header {
    background-color: #2c3e50;
    color: white;
    text-align: center;
    padding: 20px;
}

.tagline {
    font-size: 16px;
    margin-top: 5px;
}

/* Requirement 2 & 3: Section Styling */
.content-section {
    background-color: rgb(176, 166, 166);
    border: 3px solid #3498db;
    margin: 30px auto;
    padding: 20px;
    width: 80%;
    border-radius: 8px;
}

/* Text Styling */
.section-heading {
    color: #2c3e50;
}

p {
    font-size: 16px;
    line-height: 1.5;
}

/* ID Selector */
```

```
#highlight-text {
    color: #4d3ce7;
    font-weight: bold;
}

.for-div {
    background-color: #dff9fb;
    border: 3px solid #0984e3;
    margin: 30px auto;
    padding: 30px;
    width: 60%;
    text-align: center;
    font-weight: bold;
    border-radius: 8px;
}

/* Requirement 5: Form Styling */

.form-style {
    background-color: white;
    border: 3px solid #27ae60;
    margin: 30px auto;
    padding: 25px;
    width: 350px;
    border-radius: 10px;
}

/* Form Heading */

.form-heading {
    text-align: center;
    color: #2c3e50;
}
```

```
/* Labels */
.form-style label {
    display: block;
    margin-top: 12px;
    font-weight: bold;
}

/* Inputs */
.form-style input,
.form-style select {
    width: 100%;
    padding: 10px;
    margin-top: 5px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

/* Gender Alignment */
.gender-group {
    display: flex;
    gap: 20px;
    margin-top: 5px;
}

.gender-group input {
    width: auto;
}

/* Button */
.form-style button {
    width: 100%;
    padding: 12px;
    margin-top: 20px;
```

```

background-color: #2c3e50;
color: white;
border: none;
border-radius: 5px;
font-size: 16px;

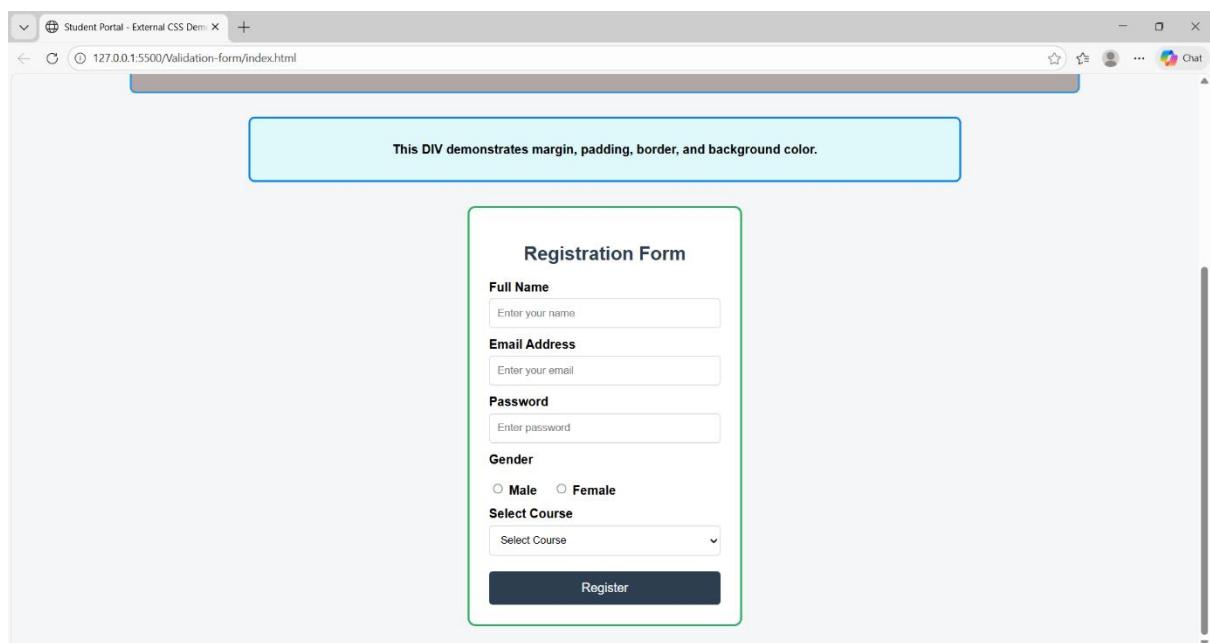
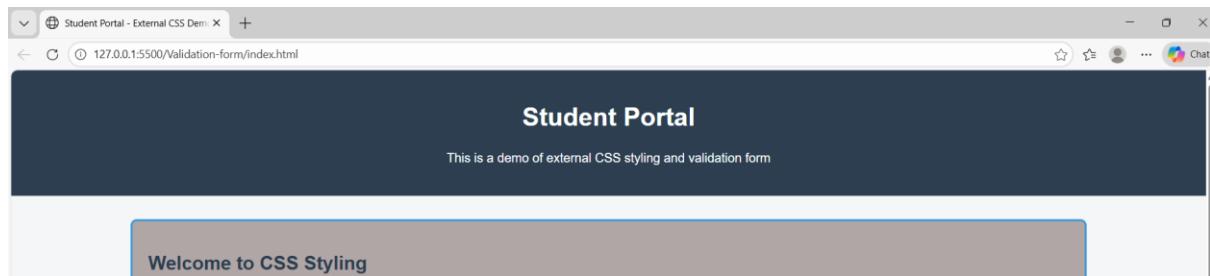
}

/* Hover */

.form-style button:hover {
background-color: #27ae60;
}

```

3.Output Screenshot



4.Code Explanation

<link rel="stylesheet" href="style.css"> -->Links the external CSS file to apply styles to the HTML document.

* { box-sizing: border-box; } -->Universal selector applies border-box model so padding and border are included in element width.

body -->Styles the entire webpage by setting font family, background color, and removing default margin.

.header -->Class selector used to style the header section with background color, text color, alignment, and padding.

.tagline -->Class selector used to style the tagline text by adjusting font size and spacing.

.content-section -->Class selector used to style the section with background color, border, margin, padding, and width.

.section-heading -->Class selector used to apply color styling to the section heading.

p -->Element selector used to apply font size and line spacing to all paragraph elements.

#highlight-text -->ID selector used to apply unique color and bold styling to a specific paragraph.

.for-div -->Class selector used to demonstrate margin, padding, border, and background color for box model spacing.

.form-style -->Class selector used to style the form container with border, padding, margin, and width.

.form-heading -->Class selector used to align and style the form heading text.

.form-style label -->Descendant selector used to style label elements inside the form.

.form-style select -->Descendant selector used to style dropdown select elements.

.gender-group -->Class selector using flexbox to align radio buttons horizontally.

.gender-group input -->Targets radio inputs and adjusts their width automatically.

margin -->Creates space outside the element border.

padding -->Creates space inside the element between border and content.

border -->Defines the boundary around an element.

background-color -->Sets the background color of an element.

width -->Specifies the horizontal size of an element.

border-radius -->Creates rounded corners on element borders.

Problem 2: Personal Profile Card (Level-1)

1. Problem Statement

You are building a simple personal profile card for a portfolio website. The card will display a user's name, role, short bio, and contact button, styled purely using CSS.

Requirements Create a profile card layout containing: Name (heading) Role/title Short description paragraph Contact button

Apply: Text formatting (alignment, decoration) Font styles (font-family, size, weight)
Typography enhancements (line-height, letter-spacing)

Use the CSS Box Model: Padding for spacing inside the card Border to define the card boundary Margin to separate the card from the page edges

Add: Background color Border radius Hover effect on the button

2. Code

Index.html

```
<!DOCTYPE html>

<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Personal Profile Card</title>

        <!-- Requirement: External CSS file -->
        <link rel="stylesheet" href="style.css">
    </head>

    <body>

        <!-- Profile Card Container -->
        <div class="card">
```

```
<!-- Name (Requirement: Name heading) -->
<h1 class="name">Dhanamreddy Srinu</h1>

<!-- Role/Title -->
<h3 class="role">Software Developer</h3>

<!-- Short Description -->
<p class="bio">
    Passionate and motivated software developer with experience in building
    web applications using HTML, CSS, JavaScript, React, and Node.js.
    Interested in creating user-friendly interfaces and learning modern
    full-stack technologies.
</p>

<!-- Contact Button -->
<button class="btn">Contact Me</button>

</div>
</body>
</html>
```

Style.css

```
body {
    font-family: Arial, sans-serif; /* font-family requirement */
    background-color: #eef2f7; /* background color requirement */
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
```

```
margin: 0;  
}  
  
.card {  
width: 350px;  
background-color: white; /* background color */  
padding: 25px; /* Box Model: padding */  
border: 2px solid #2c3e50; /* Box Model: border */  
margin: 20px; /* Box Model: margin */  
border-radius: 12px; /* border radius */  
text-align: center; /* text alignment */  
box-shadow: 0 4px 10px rgba(0,0,0,0.2);  
}
```

```
.name {  
font-size: 26px; /* font size */  
font-weight: bold; /* font weight */  
letter-spacing: 1px; /* typography enhancement */  
text-decoration: underline; /* text decoration */  
margin-bottom: 10px;  
}
```

```
.role {  
font-size: 18px;  
font-weight: 600;  
color: #2980b9;  
letter-spacing: 0.5px;  
margin-bottom: 15px;  
}  
  
.bio {
```

```

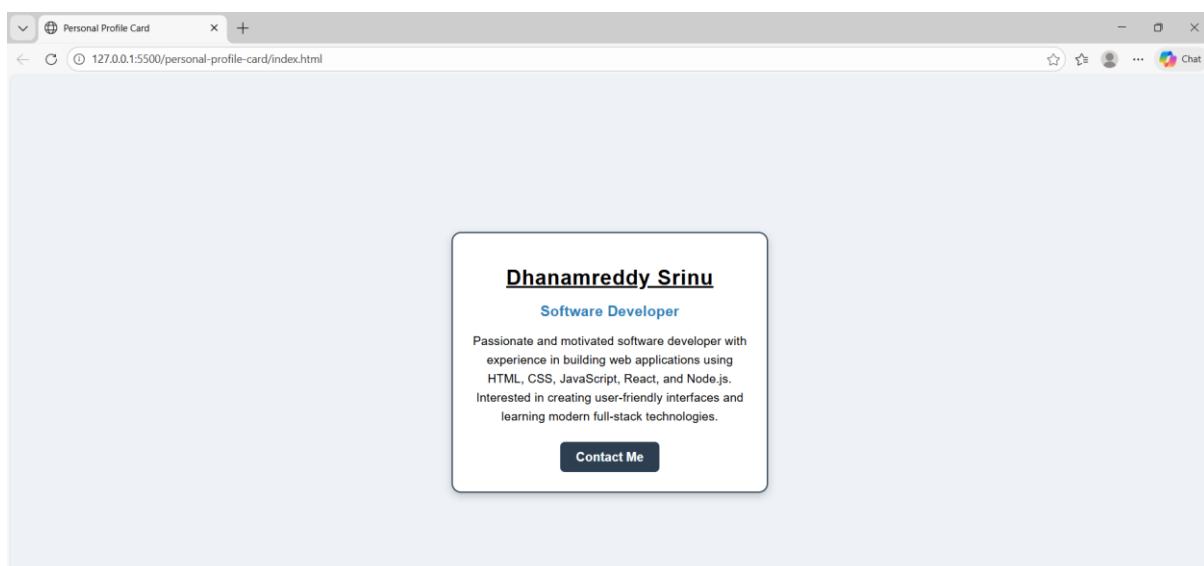
font-size: 15px;
line-height: 1.6; /* typography requirement */
letter-spacing: 0.3px; /* typography requirement */
margin-bottom: 20px;
}

.btn {
padding: 10px 20px; /* padding */
font-size: 16px;
font-weight: bold;
background-color: #2c3e50;
color: white;
border: none;
border-radius: 6px;
cursor: pointer;
}

.btn:hover {
background-color: #27ae60;
transition: 0.3s;
}

```

3. Output Screenshot



4. Code Explanation

<link rel="stylesheet" href="style.css"> -->Links the external CSS file to apply styles to the profile card.

body -->Styles the entire webpage by setting font family, background color, and using flexbox for center alignment.

display: flex -->Enables flexbox layout to align elements horizontally or vertically.

justify-content: center -->Horizontally centers the profile card inside the webpage.

align-items: center -->Vertically centers the profile card inside the webpage.

height: 100vh -->Sets the body height equal to the full viewport height.

margin: 0 -->Removes default browser margin to prevent unwanted spacing.

.card -->Class selector used to style the profile card container.

width: 350px -->Defines the fixed width of the profile card.

background-color: white -->Sets the background color of the card.

padding: 25px -->Adds space inside the card between content and border.

border: 2px solid #2c3e50 -->Defines the boundary around the card.

margin: 20px -->Creates space outside the card from other elements.

border-radius: 12px -->Rounds the corners of the card.

text-align: center -->Aligns all text content to the center.

box-shadow -->Adds shadow effect around the card for better visual appearance.

.name -->Class selector used to style the name heading.

font-size: 26px -->Sets the size of the name text.

font-weight: bold -->Makes the name text bold.

letter-spacing: 1px -->Adds space between characters for better readability.

text-decoration: underline -->Adds underline styling to the name text.

.role -->Class selector used to style the role/title text.

font-size: 18px -->Sets the size of the role text.

font-weight: 600 -->Applies semi-bold styling to the role text.

color: #2980b9 -->Changes the text color.

letter-spacing -->Improves readability by adding space between letters.

.bio -->Class selector used to style the description paragraph.

font-size: 15px -->Sets the paragraph text size.

line-height: 1.6 -->Controls vertical spacing between lines.

letter-spacing -->Controls spacing between characters.

.btn -->Class selector used to style the contact button.

padding -->Adds space inside the button for better click area.

background-color -->Sets the button background color.

color: white -->Sets the text color.

border: none -->Removes default button border.

border-radius -->Creates rounded corners on the button.

cursor: pointer -->Changes mouse cursor to pointer on hover.

.btn:hover -->Pseudo-class used to apply styles when mouse hovers over the button.

background-color: #27ae60 -->Changes button color during hover for user interaction feedback.

Problem 3: Product Feature List Section (Level-1)

1. Problem Statement

You are designing a feature highlight section for a product landing page that lists key features in a visually appealing way.

Requirements

Create a section with:

- Section heading
- List of 4–5 features

Style the list using:

- Custom fonts
- Text color and background color
- Border and padding for each feature item

Use:

- CSS selectors (element, class, descendant)

Apply hover effects on feature items

2. Code

Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">
    <title>Product Features Section</title>
    <!-- External CSS File -->
    <link rel="stylesheet" href="style.css">
</head>

<body>

    <!-- Feature Section -->
    <section class="features-section">
```

```
<!-- Section Heading (Requirement) -->
<h2 class="section-heading">Product Key Features</h2>
<!-- Unordered List (Requirement: must use ul) -->
<ul class="feature-list">
    <li class="feature-item">
        Fast and reliable performance for daily tasks
    </li>
    <li class="feature-item">
        User-friendly interface with modern design
    </li>
    <li class="feature-item">
        Secure data protection and privacy features
    </li>
    <li class="feature-item">
        Cross-platform compatibility
    </li>
    <li class="feature-item">
        24/7 customer support available
    </li>
</ul>
</section>
</body>
</html>
```

Style.css

```
body {
    font-family: "Segoe UI", Arial, sans-serif; /* custom font */
    background-color: #f4f6f8;
    margin: 0;
```

```
}

.features-section {
    width: 60%;
    margin: 40px auto;
    padding: 20px;
    background-color: #ffffff;
    border: 2px solid #2c3e50;
    border-radius: 10px;
}

h2 {
    text-align: center;
    color: #2c3e50;
    letter-spacing: 1px;
}

.feature-list {
    list-style-type: none; /* removes default bullets */
    padding: 0;
}

.feature-item {
    background-color: #e8f0fe;
    color: #333;
    border: 1px solid #3498db;
    padding: 15px; /* padding requirement */
    margin: 10px 0;
    border-radius: 6px;
    font-size: 16px;
}

.features-section ul li {
    letter-spacing: 0.5px;
```

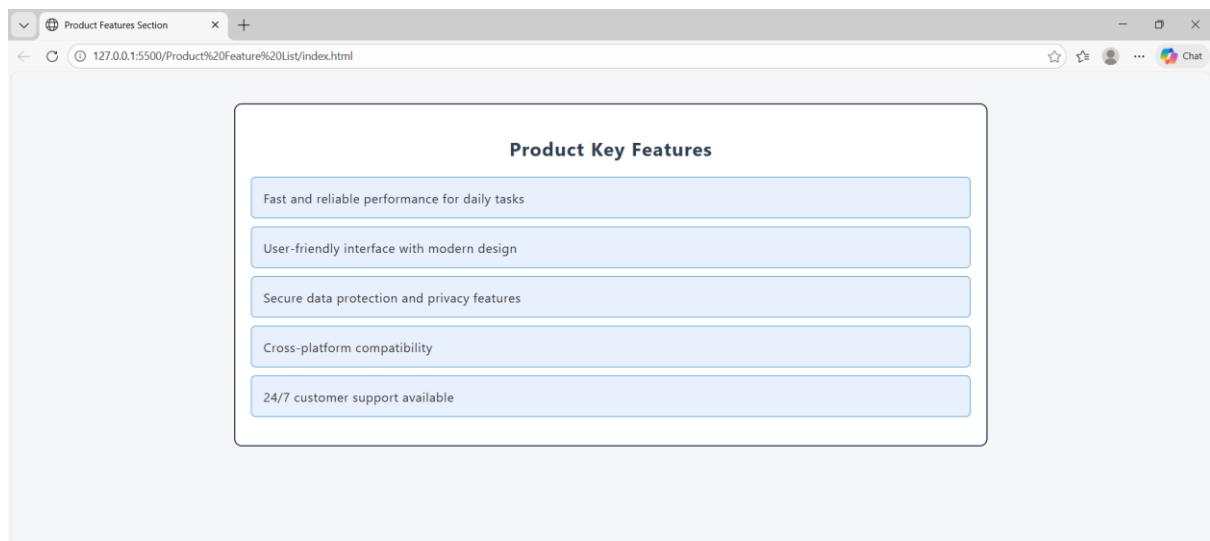
```

}

.feature-item:hover {
    background-color: #3498db;
    color: white;
    cursor: pointer;
    transition: 0.3s;
}

```

3. Output Screenshot



4. Code Explanation

<link rel="stylesheet" href="style.css"> -->Links the external CSS file to apply styling to the product feature section.

body -->Styles the entire webpage by setting custom font, background color, and removing default margin.

font-family: "Segoe UI", Arial, sans-serif -->Applies custom fonts to improve typography and appearance.

background-color: #f4f6f8 -->Sets the background color of the webpage.

.features-section -->Class selector used to style the main feature section container.

width: 60% -->Defines the width of the feature section relative to the page.

margin: 40px auto -->Adds vertical spacing and centers the section horizontally.

padding: 20px -->Adds internal spacing between content and border.

border: 2px solid #2c3e50 -->Creates a visible boundary around the feature section.

border-radius: 10px -->Adds rounded corners to the section.

background-color: #ffffff -->Sets the section background color.

h2 -->Element selector used to style the section heading.

text-align: center -->Aligns the heading text to the center.

color: #2c3e50 -->Changes the heading text color.

letter-spacing -->Adds space between letters for better readability.

.feature-list -->Class selector used to style the unordered list container.

list-style-type: none -->Removes default bullet points from the list.

padding: 0 -->Removes default spacing inside the list.

.feature-item -->Class selector used to style each feature list item.

background-color: #e8f0fe -->Sets background color for each feature item.

color: #333 -->Sets text color of feature items.

border: 1px solid #3498db -->Adds border around each feature item.

padding: 15px -->Adds internal spacing inside each feature item.

margin: 10px 0 -->Adds spacing between feature items.

border-radius: 6px -->Rounds the corners of each feature item.

font-size: 16px -->Sets the text size of feature items.

.features-section ul li -->Descendant selector used to apply letter spacing to list items inside the feature section.

letter-spacing: 0.5px -->Improves readability by spacing characters.

.feature-item:hover -->Pseudo-class used to apply styles when mouse hovers over a feature item.

background-color: #3498db -->Changes background color on hover.

color: white -->Changes text color on hover.

cursor: pointer -->Changes cursor to pointer for interactive effect.

transition: 0.3s -->Creates smooth animation during hover effect.

Problem 4: Responsive Blog Layout (Level-2)

1. Problem Statement

You are creating a blog homepage that displays articles in a modern responsive layout suitable for desktop, tablet, and mobile screens.

Requirements

Create a page layout with: Header Main content area with multiple blog cards

Use: **Flexbox or CSS Grid** for layout Box model for spacing Each blog card should contain:
Title Short description Read More button

Apply: Background colors Borders Typography enhancements

Use **media queries** to: Display multiple columns on desktop Switch to single column on mobile

Technical Constraints Only HTML + CSS No CSS frameworks No fixed widths (use percentages or flexible units)

Learning Outcome

Learners should be able to:Understand Flexbox/Grid layouts Apply responsive design principles Write effective media queries Build real-world page layouts

2. Code

Index.html

```
<!DOCTYPE html>

<html lang="en">

    <head>
        <meta charset="UTF-8">
        <!-- Requirement: Responsive design support -->
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Responsive Blog Layout</title>
        <!-- Requirement: External CSS -->
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <!-- Requirement: Header -->
```

```
<header class="header">
  <h1>My Tech Blog</h1>
  <p class="tagline">Latest articles on Web Development</p>
</header>

<!-- Requirement: Main content area -->
<!-- Contains multiple blog cards -->

<main class="blog-container">

  <!-- Blog Card 1 -->
  <article class="blog-card">
    <h2 class="blog-title">Understanding HTML Basics</h2>
    <p class="blog-description">
      Learn the fundamentals of HTML and how to structure web pages effectively.
    </p>
    <button class="read-btn">Read More</button>
  </article>

  <!-- Blog Card 2 -->
  <article class="blog-card">
    <h2 class="blog-title">CSS Flexbox Guide</h2>
    <p class="blog-description">
      Master Flexbox to create responsive and flexible layouts easily.
    </p>
    <button class="read-btn">Read More</button>
  </article>

  <!-- Blog Card 3 -->
  <article class="blog-card">
    <h2 class="blog-title">JavaScript for Beginners</h2>
    <p class="blog-description">
      Start your JavaScript journey and build interactive web applications.
    </p>
  </article>
</main>
```

```
<button class="read-btn">Read More</button>
</article>
<!-- Blog Card 4 -->
<article class="blog-card">
  <h2 class="blog-title">Responsive Web Design</h2>
  <p class="blog-description">
    Learn how to create websites that work perfectly on all devices.
  </p>
  <button class="read-btn">Read More</button>
</article>
</main>
</body>
</html>
```

Style.css

```
body {
  font-family: Arial, sans-serif; /* typography */
  margin: 0;
  background-color: #83bfffa;
}

.header {
  background-color: #2c3e50;
  color: white;
  text-align: center;
  padding: 2%;
}

.tagline {
  letter-spacing: 1px; /* typography enhancement */
```

```
}

.blog-container {
    display: flex; /* Flexbox requirement */
    flex-wrap: wrap; /* allows responsive wrapping */
    gap: 2%; /* spacing between cards */
    padding: 2%;
    justify-content: center;
}

.blog-card {
    background-color: white; /* background color */
    border: 1px solid #ccc; /* border */
    padding: 2%; /* box model padding */
    margin: 1%; /* box model margin */
    flex: 1 1 30%; /* flexible width */
    border-radius: 8px;
}

/* Typography Enhancements */

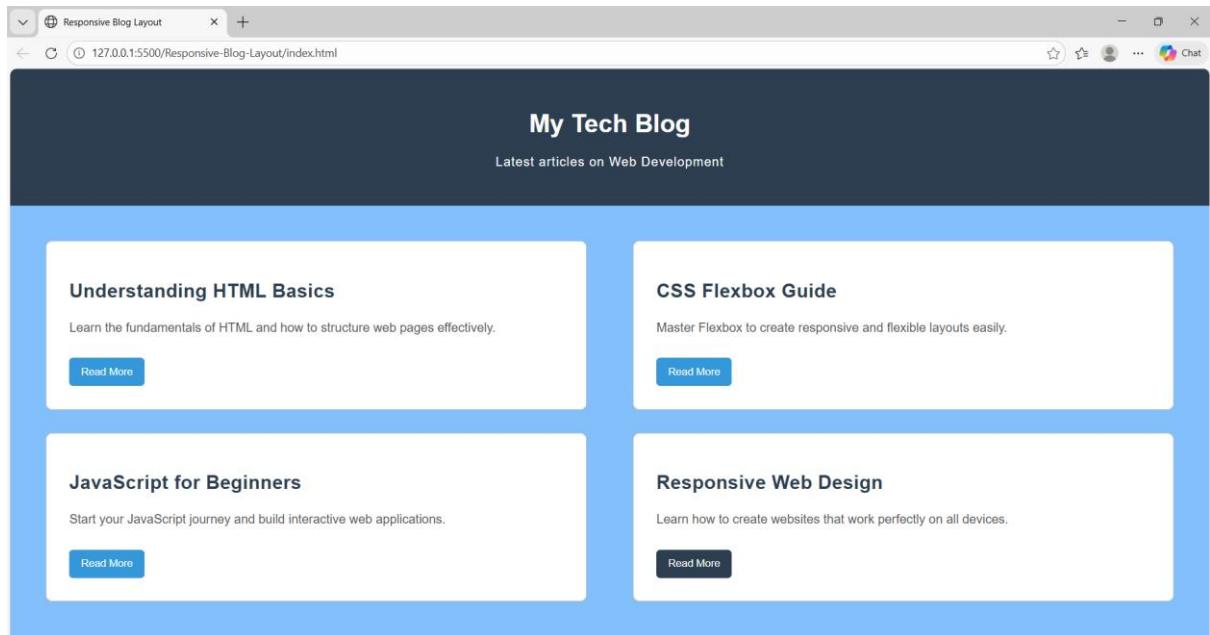
.blog-title {
    color: #2c3e50;
    letter-spacing: 0.5px;
}

.blog-description {
    line-height: 1.6;
    color: #555;
}

.read-btn {
    background-color: #3498db;
    color: white;
}
```

```
border: none;  
padding: 10px 15px;  
margin-top: 10px;  
border-radius: 5px;  
cursor: pointer;  
}  
/* Hover effect */  
.read-btn:hover {  
background-color: #2c3e50;  
}  
/* Requirement: Media Query */  
/* Tablet View */  
@media (max-width: 768px) {  
.blog-card {  
flex: 1 1 45%; /* two columns */  
}  
}  
/* Requirement: Media Query */  
/* Mobile View */  
/* Switch to single column */  
@media (max-width: 480px) {  
.blog-card {  
flex: 1 1 100%; /* single column */  
}  
}
```

3. Output Screenshot



4. Code Explanation

<meta name="viewport" content="width=device-width, initial-scale=1.0"> -->Enables responsive design by adjusting layout based on device screen size.

<link rel="stylesheet" href="style.css"> -->Links the external CSS file to apply styles to the blog layout.

body -->Styles the entire webpage by setting font family, removing default margin, and applying background color.

font-family: Arial, sans-serif -->Sets the font style for better readability.

margin: 0 -->Removes default browser spacing.

background-color: #83bffa -->Sets the page background color.

.header -->Class selector used to style the header section.

background-color: #2c3e50 -->Sets header background color.

color: white -->Sets header text color.

text-align: center -->Aligns header text to center.

padding: 2% -->Adds internal spacing inside the header.

.tagline -->Class selector used to style the tagline text.

letter-spacing: 1px -->Adds space between letters for typography enhancement.

.blog-container -->Class selector used to style the main blog content container.

display: flex -->Enables Flexbox layout for responsive arrangement.

flex-wrap: wrap -->Allows blog cards to move to the next row when needed.

gap: 2% -->Creates spacing between blog cards.

padding: 2% -->Adds internal spacing inside the container.

justify-content: center -->Aligns blog cards horizontally.

.blog-card -->Class selector used to style each blog card.

background-color: white -->Sets background color of the card.

border: 1px solid #ccc -->Creates boundary around the card.

padding: 2% -->Adds internal spacing inside the card.

margin: 1% -->Adds external spacing between cards.

flex: 1 1 30% -->Defines flexible width allowing multiple columns.

border-radius: 8px -->Creates rounded corners.

.blog-title -->Class selector used to style blog title text.

color: #2c3e50 -->Changes title text color.

letter-spacing -->Improves readability by spacing characters.

.blog-description -->Class selector used to style blog description text.

line-height: 1.6 -->Controls vertical spacing between lines.

color: #555 -->Sets description text color.

.read-btn -->Class selector used to style the Read More button.

background-color: #3498db -->Sets button background color.

color: white -->Sets button text color.

padding -->Adds space inside the button.

border-radius -->Creates rounded button corners.

cursor: pointer -->Changes cursor to pointer on hover.

.read-btn:hover -->Pseudo-class used to apply hover effect on button.

background-color: #2c3e50 -->Changes button color when mouse hovers over it.

@media (max-width: 768px) -->Media query used to apply styles for tablet screen sizes.

.blog-card { flex: 1 1 45%; } -->Displays blog cards in two columns on tablet screens.

@media (max-width: 480px) -->Media query used to apply styles for mobile screen sizes.

Problem 5: Responsive Dashboard Layout (Level-2)

1. Problem Statement

You are designing a simple admin dashboard layout for a web application.

Requirements

Create a layout with: Sidebar navigation Main content area Info cards inside the main section

Use: display properties Flexbox or Grid for page structure

Style: Sidebar with background color Cards with borders, padding, and margins

Apply **media queries** to: Convert sidebar into top navigation on mobile Stack cards vertically on small screens

2. Code

```
<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <!-- Requirement: Responsive layout support -->
        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Responsive Admin Dashboard</title>

        <!-- Requirement: External CSS (HTML + CSS only) -->
        <link rel="stylesheet" href="style.css">

    </head>

    <body>

        <!-- Requirement: Dashboard Layout      -->
```

```
<!-- Uses display and Flexbox structure -->

<div class="dashboard">

    <!-- Requirement: Sidebar Navigation -->
    <aside class="sidebar">

        <h2 class="logo">Admin Dashboard</h2>

        <ul class="nav-menu">

            <li> Dashboard</li>
            <li> Users</li>
            <li> Analytics</li>
            <li> Settings</li>
            <li> Logout</li>

        </ul>

    </aside>

    <!-- Requirement: Main Content Area -->
    <!-- Contains dashboard overview, stats -->
    <!-- and info cards -->

    <main class="main-content">

        <!-- Dashboard Title -->
        <h1 class="page-title">Dashboard Overview</h1>

    </main>

</div>
```

```
<!-- Dashboard Description -->
<p class="page-description">

    Welcome back, Admin! This dashboard provides a quick summary of your system
    performance,
        user activity, sales statistics, and real-time application insights. You can monitor
        important metrics and manage your platform efficiently from here.

</p>

<!-- Requirement: Info Cards          -->
<!-- Each card shows important statistics -->
<div class="cards-container">

    <!-- Card 1 -->
    <div class="card">
        <h3>Total Users</h3>

        <p>
            1,250 registered users are currently using the platform.
            This includes both active and inactive accounts.
        </p>

        <small>
            8% increase compared to last month
        </small>
    </div>

    <!-- Card 2 -->
    <div class="card">
        <h3>Total Revenue</h3>
```

<p>

The platform has generated \$12,500 in total revenue this month
from product sales and subscriptions.

</p>

<small>

12% increase compared to last month

</small>

</div>

<!-- Card 3 -->

<div class="card">

<h3>New Orders</h3>

<p>

320 new orders have been placed recently. This shows steady
customer engagement and business growth.

</p>

<small>

5% increase compared to last week

</small>

</div>

<!-- Card 4 -->

<div class="card">

<h3>Active Sessions</h3>

<p>

Currently, 85 users are actively using the application.

This reflects real-time user engagement.

</p>

<small>

Real-time monitoring enabled

</small>

</div>

<!-- Card 5 -->

<div class="card">

<h3>System Performance</h3>

<p>

System performance is operating at 92% efficiency with
optimal response time and minimal downtime.

</p>

<small>

Status: Stable

</small>

</div>

<!-- Card 6 -->

<div class="card">

<h3>Support Requests</h3>

<p>

18 support requests are currently pending. Admin can review

and resolve them to maintain service quality.

```
</p>
<small>
    Needs attention
</small>
</div>
</div>
</main>
</div>
</body>
</html>
```

Style.css

```
/* Requirement: Display Property      */
/* Global styling                  */
body {

    margin: 0;

    font-family: Arial, sans-serif;

    background-color: #a2a4a6;

}

/* Requirement: Flexbox Page Structure */
/* Uses display:flex for layout      */
/* Sidebar and main content alignment */
.dashboard {
```

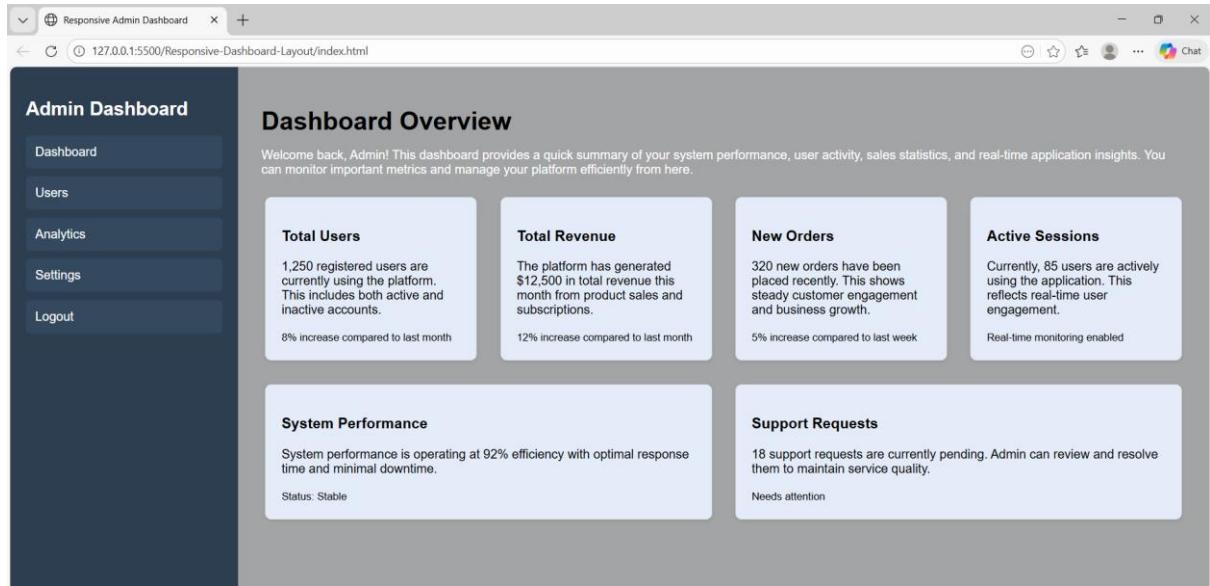
```
display: flex; /* Requirement: display property */  
  
min-height: 100vh;  
  
}  
  
/* Requirement: Sidebar Styling */  
/* Background color applied */  
.sidebar {  
  
background-color: #2c3e50; /* Requirement */  
  
color: white;  
width: 250px;  
padding: 20px;  
}  
/* Sidebar logo */  
.logo {  
margin-bottom: 20px;  
}  
/* Navigation menu */  
.nav-menu {  
list-style: none;  
padding: 0;  
}  
  
/* Navigation items */  
.nav-menu li {
```

```
padding: 12px;  
margin-bottom: 10px;  
background-color: #34495e;  
border-radius: 5px;  
cursor: pointer;  
}  
  
/* Hover effect */  
.nav-menu li:hover {  
background-color: #1abc9c  
}  
/* Requirement: Main Content Styling */  
.main-content {  
flex: 1; /* flexible layout */  
padding: 30px;  
}  
/* Typography */  
.page-title {  
margin-bottom: 10px;  
}  
  
.page-description {  
color: white;  
margin-bottom: 20px;  
}  
/* Requirement: Cards Container */  
/* Uses Flexbox layout */  
.cards-container {  
display: flex; /* Requirement: Flexbox */
```

```
flex-wrap: wrap;  
gap: 20px;  
}  
/* Requirement: Cards Styling */  
/* Border, padding, margin applied */  
.card {  
background-color: rgb(228, 236, 249);  
border: 1px solid #ddd; /* Requirement */  
padding: 20px; /* Requirement */  
margin: 5px; /* Requirement */  
border-radius: 8px  
flex: 1 1 200px;  
box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
}  
/* Hover effect */  
.card:hover {  
transform: translateY(-5px);  
  
transition: 0.3s;  
}  
/* Requirement: Media Query */  
/* Convert sidebar into top navigation */  
@media (max-width: 768px) {  
.dashboard {  
flex-direction: column;  
}  
.sidebar {
```

```
width: 100%;  
display: flex;  
justify-content: space-between;  
align-items: center;  
}  
  
.nav-menu {  
display: flex;  
gap: 10px;  
}  
  
.nav-menu li {  
margin: 0;  
}  
}  
  
/* Requirement: Media Query */  
/* Stack cards vertically on mobile */  
  
@media (max-width: 480px) {  
 .cards-container {  
flex-direction: column;  
}  
}
```

3. Output Screenshot



4. Code Explanation

<meta name="viewport" content="width=device-width, initial-scale=1.0"> -->Enables responsive design by adjusting layout according to device screen size.

<link rel="stylesheet" href="style.css"> -->Links the external CSS file to apply styling to the dashboard layout.

body -->Styles the entire webpage by setting font family, removing default margin, and applying background color.

margin: 0 -->Removes default browser spacing around the page.

font-family: Arial, sans-serif -->Sets the font style for readability.

background-color: #a2a4a6 -->Sets background color of the dashboard page.

.dashboard -->Class selector used as the main container for sidebar and main content.

display: flex -->Enables Flexbox layout to arrange sidebar and main content horizontally.

min-height: 100vh -->Ensures the dashboard takes full screen height.

.sidebar -->Class selector used to style the sidebar navigation section.

background-color: #2c3e50 -->Sets sidebar background color.

color: white -->Sets sidebar text color.

width: 250px -->Defines fixed width of sidebar.

padding: 20px -->Adds internal spacing inside sidebar.

.logo -->Class selector used to style the dashboard logo heading.
margin-bottom: 20px -->Adds spacing below the logo.

.nav-menu -->Class selector used to style the navigation menu list.
list-style: none -->Removes default bullet points from list.
padding: 0 -->Removes default spacing.

.nav-menu li -->Class selector used to style each navigation item.
padding: 12px -->Adds internal spacing inside menu item.
margin-bottom: 10px -->Adds spacing between menu items.
background-color: #34495e -->Sets background color for menu items.
border-radius: 5px -->Adds rounded corners.
cursor: pointer -->Changes cursor to pointer for interactivity.

.nav-menu li:hover -->Pseudo-class used to apply hover effect on navigation items.
background-color: #1abc9c -->Changes background color when mouse hovers.

.main-content -->Class selector used to style the main content area.
flex: 1 -->Allows main content to take remaining available space.
padding: 30px -->Adds internal spacing inside main content.

.page-title -->Class selector used to style the dashboard title.
margin-bottom -->Adds spacing below the title.

.page-description -->Class selector used to style dashboard description text.
color: white -->Changes text color.
margin-bottom -->Adds spacing below the description.

.cards-container -->Class selector used to hold all info cards.
display: flex -->Enables Flexbox layout for cards.
flex-wrap: wrap -->Allows cards to move to next row if needed.
gap: 20px -->Creates spacing between cards.

.card -->Class selector used to style individual dashboard cards.
background-color -->Sets card background color.
border: 1px solid #ddd -->Creates border around card.
padding: 20px -->Adds internal spacing inside card.

margin: 5px --> Adds external spacing between cards.

border-radius: 8px --> Adds rounded corners.

flex: 1 1 200px --> Allows flexible card width.

box-shadow --> Adds shadow effect for visual enhancement.

.card:hover --> Pseudo-class used to apply hover effect on cards.

transform: translateY(-5px) --> Moves card upward on hover.

transition: 0.3s --> Creates smooth animation effect.

@media (max-width: 768px) --> Media query used for tablet screen layout.

.dashboard { flex-direction: column; } --> Converts sidebar from vertical to horizontal layout.

.sidebar { width: 100%; } --> Makes sidebar full width.

.nav-menu { display: flex; } --> Displays navigation items horizontally.

@media (max-width: 480px) --> Media query used for mobile screen layout.

.cards-container { flex-direction: column; } --> Stacks cards vertically on mobile devices.