

CSS_Day – 3_Hands_On -- Harshitha Kamatam

Problem 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 colour 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.

Source Code:

File Name: Index.html

```
1  <!--CSS_Day3_Hands_on_Problem Statement1_Harshitha Kamatam-->
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Registration Form</title>
8      <link rel="stylesheet" href="index.css">
9  </head>
10 <body>
11     <section class="form">
12         <h3> Registration Form </h3>
13         <div class="box">
14             <p>Please provide the required details below.</p>
15         </div>
16         <br>
17         <form>
18             <label> Name: </label>
19             <input type="text" placeholder="Type your Name" required>
20             <br><br>
21             <label> Email:</label>
22             <input type="email" placeholder="Type your Email Id" required>
23             <br><br>
24             <label> Password: </label>
25             <input type="password" placeholder="Type your Password" required>
26             <br><br>
27             <label> Gender: </label>
28             <input type="radio" name="Gender"> Male
29             <input type="radio" name="Gender"> Female
30             <br><br>
31             <label for="Courses"> Select Course: </label>
32             <input type="text" id="Courses" name="Courses" list="Course">
33             <datalist id="Course">
34                 <option value="Python">
35                 <option value="Java">
36                 <option value="HTML">
37             </datalist>
38             <br><br>
39             <button class="submit" type="submit"> Submit</button>
40             <br><br>
41         </form>
42     </section>
43 </body>
44 </html>
```

File Name: index.css

```
1  /* CSS Day3 Hands on Problem Statement1 Harshitha Kamatam */
2  body {
3      font-family:Arial, Helvetica, sans-serif;
4      background-color: #bisque;
5  }
6  h3 {
7      text-align: center;
8      text-transform: uppercase;
9      color: #black;
10 }
11
12 p {
13     text-align: left;
14     font-size: 15px;
15 }
16 .box {
17     padding: 10px;
18     margin: 10px;
19     border: 2px solid #lightgray;
20 }
21
22 .form {
23     background-color: #lightseagreen;
24     border: 1px solid #black;
25     border-radius: 20px;
26     margin: 50px;
27     padding: 25px;
28 }
29 .submit{
30     background-color: #powderblue
31 }
```

Output:

REGISTRATION FORM

Please provide the required details below.

Name:

Email:

Password:

Gender: ☒ Male ☐ Female

Select Course:

REGISTRATION FORM

Please provide the required details below.

Name:

Email:

Password:

Gender: ☒ Male ☐ Female

Select Course:

Explanation:

This code creates a **Registration Form** using HTML and styles it with an external CSS file. The CSS adds background colors, spacing, borders, rounded corners, and styles the submit button to make the form look neat and attractive.

Problem 2

Problem Statement:

Personal Profile Card (Level-1)

Scenario

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

Technical Constraints

- Use only HTML + CSS
- No images or icons
- No frameworks (Bootstrap, Tailwind, etc.)
- Inline CSS not allowed (use `<style>` or external CSS)

Learning Outcome

You should be able to:

- Understand how CSS styles text and fonts
- Apply box model concepts effectively
- Style elements using colours, borders, and spacing
- Improve visual hierarchy using typography

Source Code:

File Name: index.html

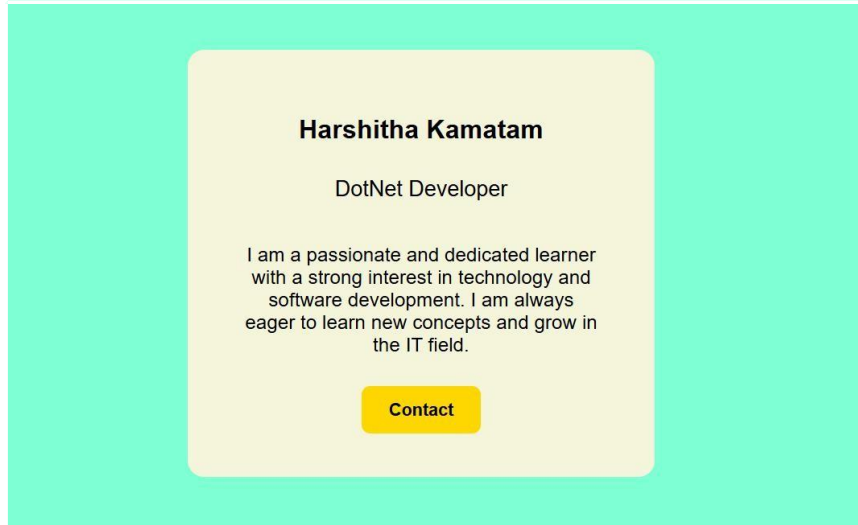
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Personal Profile Card</title>
7      <style>
8          body {
9              margin: 0;
10             display: flex;
11             font-family: Arial, Helvetica, sans-serif;
12             background-color: #a9d0d9;
13             text-align: center;
14             justify-content: center;
15             align-items: center;
16             height: 100vh;
17         }
18
19         .profile {
20             background-color: #f5f5dc;
21             width: 350px;
22             padding: 40px;
23             text-align: center;
24             border-radius: 15px;
25         }
26
27         .role {
28             font-size: 20px;
29             font-weight: 500;
30             padding: 10px;
31         }
32
33         .description {
34             font-family: sans-serif;
35             font-size: 18px;
36             padding: 10px;
37         }
38
39         .contact {
40             display: inline-block;
41             padding: 12px 25px;
42             background-color: #ffd700;
43             color: black;
44             text-decoration: none;
45             font-weight: bold;
46             border-radius: 8px;
47         }
48
49         .contact:hover {
50             background-color: #c47a3b;
51         }
52     </style>
53 </head>
54 <body>
55     <div class="profile">
56         <h2>Harshitha Kamatam</h2>
57         <div class="role">DotNet Developer</div>
58         <p class="description">
59             I am a passionate and dedicated learner with a strong interest in technology and software development.
60             I am always eager to learn new concepts and grow in the IT field.
61         </p>
62         <a href="contact.html" class="contact">Contact</a>
63     </div>
64 </body>
65 </html>
```

File Name: contact.html

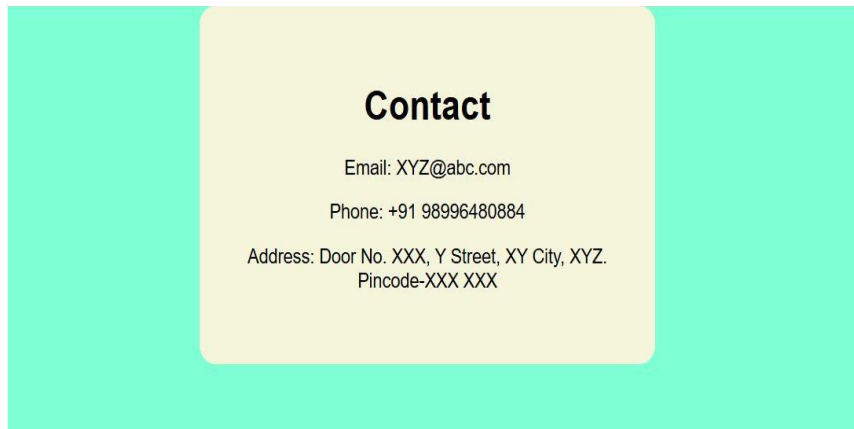
```
1  <!--CSS_Day3_Hands_on_Problem Statement2_Harshitha Kamatam-->
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Contact Page</title>
8      <style>
9          body {
10             margin: 0;
11             font-family: Arial, Helvetica, sans-serif;
12             display: flex;
13             background-color: #a9d0d9;
14             text-align: center;
15             justify-content: center;
16             align-items: center;
17         }
18         .contact{
19             background-color: #f5f5dc;
20             width: 350px;
21             padding: 40px;
22             text-align: center;
23             border-radius: 15px;
24         }
25     </style>
26 </head>
27 <body>
28     <div class="contact">
29         <h1>Contact</h1>
30         <p>Email: XYZ@abc.com</p>
31         <p>Phone: +91 98996480884</p>
32         <p>Address: Door No. XXX, Y Street, XY City, XYZ. Pincode-XXX XXX</p>
33     </div>
34 </body>
35 </html>
```


Output:

Output File: index.html



Output File: contact.html



Explanation:

This code creates a **Personal Profile Card** using HTML and CSS, and a separate Contact page linked with a button. The CSS centers the card on the screen, adds background colors, spacing, rounded corners, and a hover effect on the Contact button.

Problem 3

Problem Statement:

Product Feature List Section (Level-1)

Scenario

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

Technical Constraints

- Use unordered list ()
- No JavaScript
- No layout frameworks
- Only basic HTML structure

Learning Outcome

You will be able to:

- Practice CSS selectors
- Style lists professionally
- Improve UI consistency

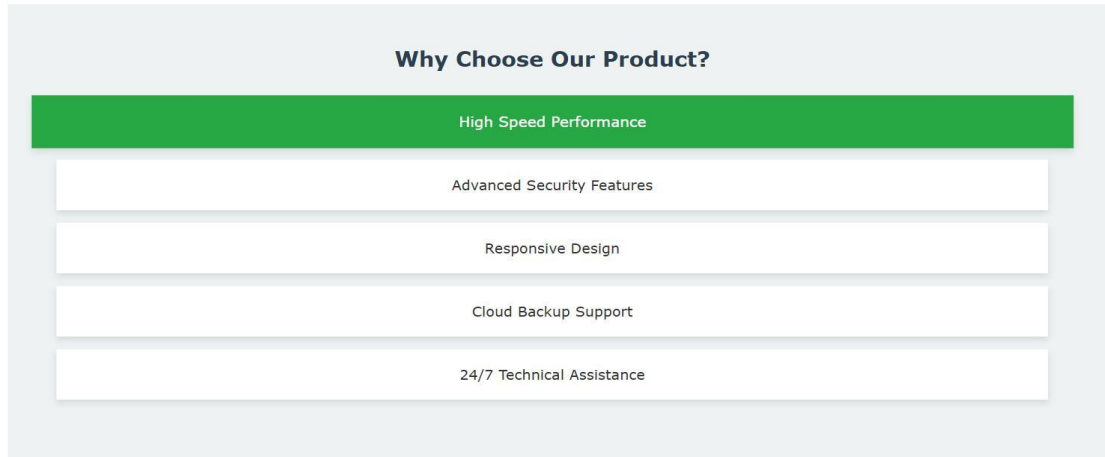
Source Code:

File Name: index.html

```
p3 > index.html > ...
1 <!--CSS Day3 Hands on Problem Statement3 Harshitha Kamatam-->
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <title>Product Features</title>
6 <style>
7 body {
8     font-family: "Verdana", sans-serif;
9     background-color: #eef2f3;
10    margin: 0;
11    padding: 0;
12 }
13 .features {
14     padding: 40px;
15     text-align: center;
16 }
17 .features h2 {
18     color: #2c3e50;
19     margin-bottom: 30px;
20 }
21 .features ul {
22     list-style-type: none;
23     padding: 0;
24 }
25 .features ul li {
26     background-color: #ffffff;
27     color: #333;
28     padding: 20px;
29     margin: 15px auto;
30     width: 70%;
31     box-shadow: 0 4px 8px rgba(0,0,0,0.1);
32     transition: 0.3s ease;
33 }
34 .features ul li:hover {
35     background-color: #28a745;
36     color: white;
37     transform: scale(1.05);
38 }
39 </style>
40 </head>
41 <body>
42 <section class="features">
43     <h2>Why Choose Our Product</h2>
44     <ul>
45         <li>High Speed Performance</li>
46         <li>Advanced Security Features</li>
47         <li>Responsive Design</li>
48         <li>Cloud Backup Support</li>
49         <li>24/7 Technical Assistance</li>
50     </ul>
51 </section>
52 </body>
53 </html>
```

Output:

Output File: index.html



Explanation:

This code creates a **Product Features** section using HTML and CSS. It displays a heading and a list of features, styled with background colors, spacing, shadows, and a hover effect that changes color and slightly enlarges each item.

Problem 4

Problem Statement:

Responsive Blog Layout (Level-2)

Scenario

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

Source Code:

File Name: index.html

```
1  <!--CSS_Day3_Hands_on_Problems_Statement4_Harshitha Kamatam-->
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5  <meta charset="UTF-8">
6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <title>Responsive Blog Layout</title>
8  <style>
9  * {
10     margin: 0;
11     padding: 0;
12     box-sizing: border-box;
13     font-family: Arial, sans-serif;
14 }
15 .header {
16     background-color: #2c3e50;
17     color: white;
18     text-align: center;
19     padding: 2rem 1rem;
20 }
21 .blog-container {
22     display: flex;
23     flex-wrap: wrap;
24     gap: 2%;
25     padding: 2rem;
26     justify-content: center;
27     background-color: #f4f4f4;
28 }
29 .blog-card {
30     background-color: white;
31     border: 1px solid #ddd;
32     border-radius: 8px;
33     padding: 1.5rem;
34     flex: 1 1 30%;
35     box-shadow: 0 4px 8px rgba(0,0,0,0.1);
36     transition: transform 0.3s ease;
37 }
38 .blog-card:hover {
39     transform: translateY(-5px);
40 }
41 .blog-card h2 {
42     margin-bottom: 1rem;
43     color: #333;
44 }
45 .blog-card p {
46     margin-bottom: 1.5rem;
47     color: #555;
48     line-height: 1.6;
49 }
50 .blog-card button {
51     padding: 0.6rem 1rem;
52     border: none;
53     background-color: #3498db;
54     color: white;
```

```

55     cursor: pointer;
56     border-radius: 4px;
57     transition: background 0.3s ease;
58 }
59 .blog-card button:hover {
60     background-color: #2980b9;
61 }
62 @media (max-width: 992px) {
63     .blog-card {
64         flex: 1 1 45%;
65     }
66 }
67 @media (max-width: 600px) {
68     .blog-card {
69         flex: 1 1 100%;
70     }
71 }
72 </style>
73 </head>
74 <body>
75 <header class="header">
76 <h1>My Blog</h1>
77 </header>
78 <main class="blog-container">
79 <div class="blog-card">
80 <h2>Web Development Basics</h2>
81 <p>Learn the fundamentals of HTML, CSS, and responsive design principles.</p>
82 <button>Read More</button>
83 </div>
84 <div class="blog-card">
85 <h2>Understanding Flexbox</h2>
86 <p>Master flexible layouts and alignment techniques using CSS Flexbox.</p>
87 <button>Read More</button>
88 </div>
89 <div class="blog-card">
90 <h2>Responsive Design</h2>
91 <p>Build websites that look great on desktop, tablet, and mobile screens.</p>
92 <button>Read More</button>
93 </div>
94 </main>
95 </body>
96 </html>

```

Output:

Output File: index.html



Explanation:

This code creates a **responsive blog layout** using HTML and CSS Flexbox. It displays blog posts as cards with titles, descriptions, and “Read More” buttons, styled with colors, shadows, and hover effects.

Problem 5

Problem Statement:

Responsive Dashboard Layout (Level-2)

Scenario

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

Technical Constraints

- HTML + CSS only
- No JavaScript
- No external UI libraries

Learning Outcome

You will be able to:

- Master page layouts using Flexbox/Grid
- Apply modern responsive web design techniques
- Understand how layout adapts across devices
- Build industry-relevant UI structures

Source Code:

File Name: index.html

```
b5 > <> index.html > html > head > style > *
1  <!--CSS_Day3_Hands_on_Problems_Statement5_Harshitha Kamatam-->
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Responsive Dashboard Layout</title>
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <style>
8          * {
9              margin: 0;
10             padding: 0;
11             box-sizing: border-box;
12         }
13         body {
14             font-family: "Segoe UI", sans-serif;
15             min-height: 100vh;
16         }
17         .dashboard {
18             display: grid;
19             grid-template-columns: 250px 1fr;
20             min-height: 100vh;
21         }
22         .sidebar {
23             background-color: #1e272e;
24             color: white;
25             padding: 20px;
26         }
27         .sidebar h2 {
28             margin-bottom: 20px;
29         }
30         .sidebar ul {
31             list-style: none;
32         }
33         .sidebar ul li {
34             margin: 12px 0;
35         }
36         .sidebar ul li a {
37             text-decoration: none;
38             color: white;
39         }
40         .sidebar ul li a:hover {
41             color: #00cec9;
42         }
43         .main {
44             background-color: #f1f2f6;
```

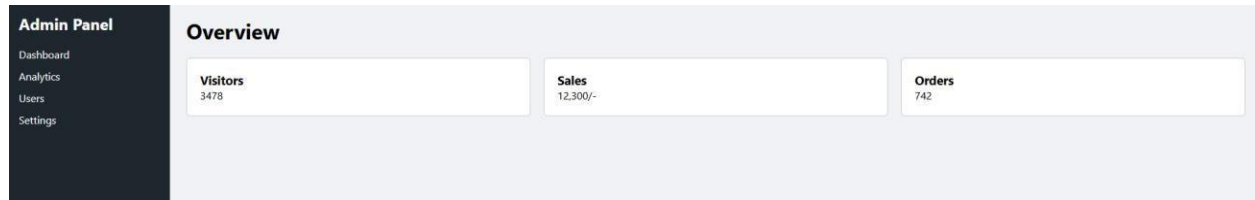
```

45         padding: 25px;
46     }
47     .main h1 {
48         margin-bottom: 20px;
49     }
50     .card-container {
51         display: flex;
52         gap: 20px;
53         flex-wrap: wrap;
54     }
55     .card {
56         background: white;
57         padding: 20px;
58         border-radius: 6px;
59         border: 1px solid #ccc;
60         flex: 1 1 30%;
61         min-width: 220px;
62         transition: 0.3s;
63     }
64     .card:hover {
65         box-shadow: 0 4px 10px rgba(0,0,0,0.15);
66         transform: translateY(-4px);
67     }
68     @media (max-width: 768px) {
69         .dashboard {
70             grid-template-columns: 1fr;
71         }
72
73         .sidebar {
74             text-align: center;
75         }
76
77         .card-container {
78             flex-direction: column;
79         }
80     }
81 </style>
82 </head>
83 <body>
84 <div class="dashboard">

```

```
84 <div class="dashboard">
85   <aside class="sidebar">
86     <h2>Admin Panel</h2>
87     <ul>
88       <li><a href="#">Dashboard</a></li>
89       <li><a href="#">Analytics</a></li>
90       <li><a href="#">Users</a></li>
91       <li><a href="#">Settings</a></li>
92     </ul>
93   </aside>
94   <section class="main">
95     <h1>Overview</h1>
96     <div class="card-container">
97       <div class="card">
98         <h3>Visitors</h3>
99         <p>3478</p>
100      </div>
101
102      <div class="card">
103        <h3>Sales</h3>
104        <p>12,300</p>
105      </div>
106
107      <div class="card">
108        <h3>Orders</h3>
109        <p>742</p>
110      </div>
111    </div>
112  </section>
113 </div>
114 </body>
115 </html>
116
```

Output:



Explanation:

This code creates a **responsive admin dashboard** using CSS Grid and Flexbox. It has a sidebar (Admin Panel with links) on the left and a main section showing cards like Visitors, Sales, and Orders.