

CSS_Day – 3_Hands_On -- Kaveri Harish Babu

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_HarishBabuKaveri-->
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
14         <div class="box">
15             <p>Please provide the required details below.</p>
16         </div>
17         <br>
18
19         <form>
20             <label> Name:</label>
21             <input type="text" placeholder="Type your Name" required>
22             <br><br>
23             <label> Email:</label>
24             <input type="email" placeholder="Type your Email Id" required>
25             <br><br>
26             <label> Password:</label>
27             <input type="password" placeholder="Type your Password" required>
28             <br><br>
29             <label> Gender:</label>
30             <input type="radio" name="Gender"> Male
31             <input type="radio" name="Gender"> Female
32             <br><br>
33             <label for="Courses"> Select Course:</label>
34             <input type="text" id="Courses" name="Courses" list="Course">
35             <datalist id="Course">
36                 <option value="Python">
37                 <option value="Java">
38                 <option value="HTML">
39             </datalist>
40             <br><br>
41             <button class="submit" type="submit"> Submit</button>
42             <br><br>
43         </form>
44     </section>
45 </body>
46 </html>
```

File Name: index.css

```
1  /* CSS_Day3_Hands_on_Problem_Statement1_HarishBabuKaveri */
2  body {
3      font-family:Arial, Helvetica, sans-serif;
4      background-color: bisque;
5  }
6
7  h3 {
8      text-align: center;
9      text-transform: uppercase;
10     color: black;
11 }
12
13 p {
14     text-align: left;
15     font-size: 15px;
16 }
17
18 .box {
19     padding: 10px;
20     margin: 10px;
21     border: 2px solid lightgray;
22 }
23
24 .form {
25     background-color: lightseagreen;
26     border: 1px solid black;
27     border-radius: 20px;
28     margin: 50px;
29     padding: 25px;
30 }
31
32 .submit{
33     background-color: powderblue
34 }
```

Output:

The image displays two versions of a registration form, likely demonstrating the effect of different CSS styles or browser rendering.

Top Screenshot (Rounded Corners):

- Header:** REGISTRATION FORM
- Text Input:** Please provide the required details below.
- Fields:**
 - Name:
 - Email:
 - Password:
 - Gender:
 - Male
 - Female
 - Select Course:
- Buttons:** Submit

Bottom Screenshot (Rectangular):

- Header:** REGISTRATION FORM
- Text Input:** Please provide the required details below.
- Fields:**
 - Name:
 - Email:
 - Password:
 - Gender:
 - Male
 - Female
 - Select Course:
- Buttons:** Submit

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  <!--CSS_Day3_Hands_on_Problem_Statement2_HarishBabuKaveri-->
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>Personal Profile Card</title>
8
9      <style>
10         body {
11             margin:0;
12             display: flex;
13             font-family: Arial, Helvetica, sans-serif;
14             background-color: aquamarine;
15             text-align: center;
16             justify-content: center;
17             align-items: center;
18         }
19
20         .profile {
21             background-color:beige;
22             width: 350px;
23             padding: 40px;
24             text-align: center;
25             border-radius: 15px;
26         }
27
28         .role{
29             font-size: 20px;
30             font-weight:500;
31             padding: 10px;
32         }
33
34         .description{
35             font-family:sans-serif;
36             font-size: 18px;
```

```

31         padding: 10px;
32     }
33
34     .description{
35         font-family:sans-serif;
36         font-size: 18px;
37     }
38
39     .contact {
40         display: inline-block;
41         padding: 12px 25px;
42         background-color:gold;
43         color: black;
44         text-decoration:underline;
45         font-weight: bold;
46         border-radius: 8px;
47     }
48
49     .contact:hover {
50         background-color:darkgoldenrod;
51     }
52 
```

</style>

</head>

<body>

<div class="profile">

<h2>Kaveri Harish Babu</h2>

<div class="role">DotNet Developer</div>

<p class="description">

I am a passionate and dedicated learner with a strong interest in technology and software development.

I am always eager to learn new concepts and grow in the IT field.

</p>

Contact

</div>

</body>

</html>

File Name: contact.html

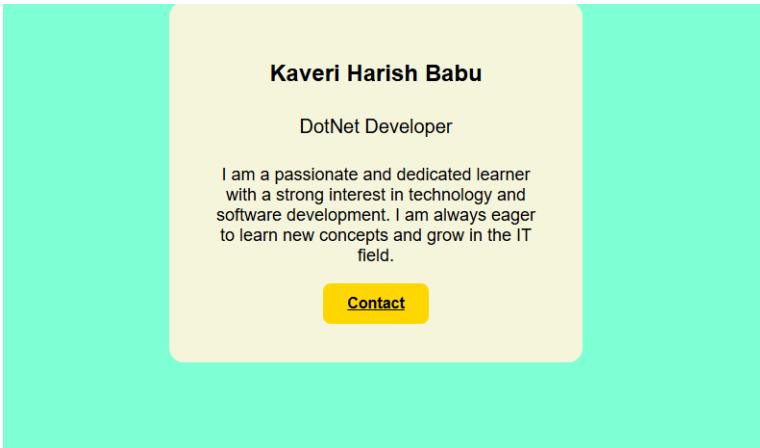
```

1 <!--CSS_Day3_Hands_on_Problem_Statement2_HarishBabuKaveri-->
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
9         <style>
10            body{
11                margin: 0;
12                display: flex;
13                font-family: Arial, Helvetica, sans-serif;
14                background-color: aquamarine;
15                text-align: center;
16                justify-content: center;
17                align-items: center;
18            }
19
20            .contact{
21                background-color:beige;
22                width: 350px;
23                padding: 40px;
24                text-align: center;
25                border-radius: 15px;
26            }
27        </style>
28    </head>
29    <body>
30        <div class="contact">
31            <h1>Contact</h1>
32            <p>Email: XYZ@abc.com</p>
33            <p>Phone: +91 9874563210</p>
34            <p>Address: Door No.: XXX, Y Street, XY City, XYZ. Pincode-XXX XXX</p>
35        </div>
36    </body>
37 </html>

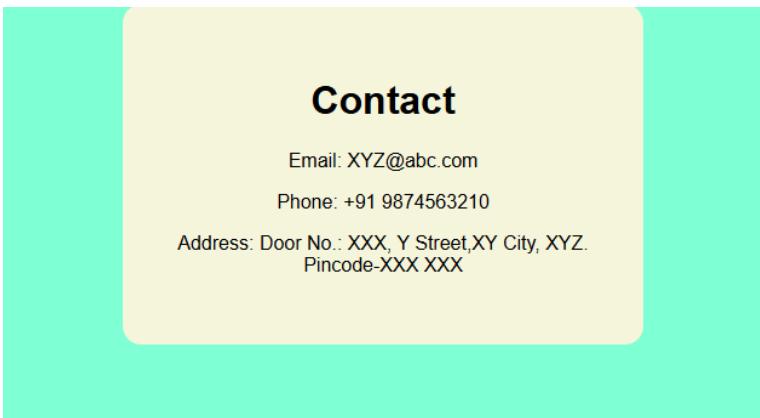
```

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

```
1  <!--CSS_Day3_Hands_on_Problem_Statement3_HarishBabuKaveri-->
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Product Features</title>
6
7      <style>
8          body {
9              font-family: "Verdana", sans-serif;
10             background-color: #eef2f3;
11             margin: 0;
12             padding: 0;
13         }
14
15         .features {
16             padding: 40px;
17             text-align: center;
18         }
19
20         .features h2 {
21             color: #2c3e50;
22             margin-bottom: 30px;
23         }
24
25         .features ul {
26             list-style-type: none;
27             padding: 0;
28         }
29
30         .features ul li {
31             background-color: #ffffff;
32             color: #333;
33             padding: 20px;
34             margin: 15px auto;
35             width: 70%;
36             box-shadow: 0 4px 8px rgba(0,0,0,0.1);
37             transition: 0.3s ease;
38         }
39
40         .features ul li:hover {
41             background-color: #28a745;
```

```

40      .features ul li:hover {
41          background-color: #28a745;
42          color: white;
43          transform: scale(1.05);
44      }
45  
```

```

46  
```

```

47 <body>
48
49     <section class="features">
50         <h2>Why Choose Our Product?</h2>
51
52         <ul>
53             <li>⚡ High Speed Performance</li>
54             <li>🔒 Advanced Security Features</li>
55             <li>📱 Responsive Design</li>
56             <li>☁️ Cloud Backup Support</li>
57             <li>⌚ 24/7 Technical Assistance</li>
58         </ul>
59     </section>
60
61 </body>
62 </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_Problem_Statement4_HarishBabuKaveri-->
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>Responsive Blog Layout</title>
8      <style>
9
10     body {
11         font-family: Arial, sans-serif;
12         background-color:beige;
13     }
14
15     header {
16         background-color: #3d89d4;
17         color:white;
18         text-align: center;
19         padding: 1px;
20     }
21
22     header h1 {
23         font-size: 2rem;
24     }
25
26     .blog-container {
27         display: flex;
28         flex-wrap: wrap;
29         justify-content: center;
30         padding: 20px;
31         gap: 20px;
32     }
33
34     .blog-card {
35         background-color:lavender;
36         border: 1px solid #ddd;
37         border-radius: 8px;
38         padding: 20px;
39         flex: 1 1 30%;
40         min-width: 250px;
41         transition: 0.3s ease;
```

```
40         min-width: 250px;
41         transition: 0.3s ease;
42     }
43
44     .blog-card h3 {
45         margin-bottom: 10px;
46         color: #2c3e50;
47     }
48
49     .blog-card p {
50         font-size: 0.95rem;
51         margin-bottom: 15px;
52         color: #555;
53     }
54
55     .blog-card button {
56         padding: 8px 15px;
57         border: none;
58         background-color: #3498db;
59         color: white;
60         cursor: pointer;
61         border-radius: 4px;
62     }
63
64     .blog-card button:hover {
65         background-color: #1f78c1;
66     }
67
68     .blog-card:hover {
69         transform: translateY(-5px);
70         box-shadow: 0 4px 12px rgba(0,0,0,0.1);
71     }
72
73     @media (max-width: 768px) {
74         .blog-card {
75             flex: 1 1 100%;
76         }
77     }
78
79     </style>
80 </head>
```

```

79      </style>
80  </head>
81  <body>
82    <header>
83      <h1>My Blog</h1>
84    </header>
85    <main class="blog-container">
86      <div class="blog-card">
87        <h3>Understanding Flexbox</h3>
88        <p>Learn how Flexbox helps in building responsive layouts easily.</p>
89        <button>Read More</button>
90      </div>
91      <div class="blog-card">
92        <h3>CSS Grid Basics</h3>
93        <p>A beginner's guide to creating modern web layouts using Grid.</p>
94        <button>Read More</button>
95      </div>
96      <div class="blog-card">
97        <h3>Responsive Design Tips</h3>
98        <p>Best practices to make your website mobile friendly.</p>
99        <button>Read More</button>
100     </div>
101   </main>
102 </body>
103 </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

```
1 | <!--CSS_Day3_Hands_on_Problem_Statement5_HarishBabuKaveri-->
2 | <!DOCTYPE html>
3 | <html>
4 | <head>
5 |   <title>Responsive Dashboard Layout</title>
6 |   <style>
7 |     * {
8 |       margin: 0;
9 |       padding: 0;
10 |      box-sizing: border-box;
11 |    }
12 |
13 |    body {
14 |      font-family: "Segoe UI", sans-serif;
15 |      min-height: 100vh;
16 |    }
17 |
18 |    .dashboard {
19 |      display: grid;
20 |      grid-template-columns: 250px 1fr;
21 |      min-height: 100vh;
22 |    }
23 |
24 |    .sidebar {
25 |      background-color: #1e272e;
26 |      color: white;
27 |      padding: 20px;
28 |    }
29 |
30 |    .sidebar h2 {
31 |      margin-bottom: 20px;
32 |    }
33 |
34 |    .sidebar ul {
35 |      list-style: none;
36 |    }
37 |
38 |    .sidebar ul li {
39 |      margin: 12px 0;
40 |    }
41 |
```

```
40 }
41
42     .sidebar ul li a {
43         text-decoration: none;
44         color: white;
45     }
46
47     .sidebar ul li a:hover {
48         color: #00cec9;
49     }
50
51     .main {
52         background-color: #f1f2f6;
53         padding: 25px;
54     }
55
56     .main h1 {
57         margin-bottom: 20px;
58     }
59
60     .card-container {
61         display: flex;
62         gap: 20px;
63         flex-wrap: wrap;
64     }
65
66     .card {
67         background: white;
68         padding: 20px;
69         border-radius: 6px;
70         border: 1px solid #ccc;
71         flex: 1 1 30%;
72         min-width: 220px;
73         transition: 0.3s;
74     }
75
76     .card:hover {
77         box-shadow: 0 4px 10px rgba(0,0,0,0.15);
78         transform: translateY(-4px);
79     }
80 }
```

```
79      }
80
81      @media (max-width: 768px) {
82
83          .dashboard {
84              grid-template-columns: 1fr;
85          }
86
87          .sidebar {
88              text-align: center;
89          }
90
91          .card-container {
92              flex-direction: column;
93          }
94      }
95
96      </style>
97  </head>
98
99  <body>
100
101     <div class="dashboard">
102
103         <aside class="sidebar">
104             <h2>Admin Panel</h2>
105             <ul>
106                 <li><a href="#dashboard">Dashboard</a></li>
107                 <li><a href="#analytics">Analytics</a></li>
108                 <li><a href="#users">Users</a></li>
109                 <li><a href="#settings">Settings</a></li>
110             </ul>
111         </aside>
112
113         <section class="main">
114             <h1>Overview</h1>
115
116             <div class="card-container">
117                 <div class="card">
118                     <h3>Visitors</h3>
119                     <p>3478</p>
```

```

116 <div class="card-container">
117   <div class="card">
118     <h3>Visitors</h3>
119     <p>3478</p>
120   </div>
121
122   <div class="card">
123     <h3>Sales</h3>
124     <p>12,300/-</p>
125   </div>
126
127   <div class="card">
128     <h3>Orders</h3>
129     <p>742</p>
130   </div>
131 </div>
132 </section>
133 </div>
134 </body>
135 </html>

```

Output:

The screenshot shows a responsive admin dashboard. On the left is a dark sidebar labeled "Admin Panel" with links for Dashboard, Analytics, Users, and Settings. The main area is titled "Overview" and contains three cards: "Visitors" (3478), "Sales" (12,300/-), and "Orders" (742).

Category	Value
Visitors	3478
Sales	12,300/-
Orders	742

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.