c

Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

Lab Report NO: 02
Course Title: Web Programming Lab
Course Code: CSE 302 Section: 231_D3

Lab Experiment Name: Introduction to CSS & Implementation of Page Layout in CSS

Student Details

Name		ID
1.	Promod Chandra Das	231002005

Lab Date: 03.03.2025
Submission Date: 11.03.2025
Course Teacher's Name: Mr.Mozdaher Abdul Quader

Lab Report Status

Marks: …………………………………
Comments:...

Signature:.....................
Date:.............................

c

➢ TITLE OF THE LAB REPORT EXPERIMENT:

Introduction to CSS & Implementation of Page Layout in CSS

➢ OBJECTIVES/AIM

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of HTML documents. It controls layout, colors, fonts, and responsiveness, enabling visually appealing and consistent designs across web pages. CSS separates content from design, enhancing maintainability. Key concepts include selectors, properties, and the box model. For page layout, CSS offers tools like Flexbox and Grid, which simplify creating responsive and dynamic designs. Flexbox focuses on one-dimensional layouts, while Grid handles two-dimensional structures. Media queries further enhance responsiveness by adapting layouts to different screen sizes. Mastering CSS is essential for modern web development and user-friendly interfaces.

1. PROCEDURE / ANALYSIS / DESIGN

1. **Procedure**:
   A procedure is a series of well-defined steps or instructions to accomplish a specific task. It ensures consistency, efficiency, and repeatability in executing processes, whether in programming, manufacturing, or daily workflows.
2. **Analysis**:
   Analysis involves breaking down a problem or system into smaller components to understand its structure, behavior, and requirements. It includes identifying inputs, outputs, constraints, and objectives, which are crucial for designing effective solutions.
3. **Design**:
   Design is the process of creating a plan or blueprint for a system, product, or solution. It translates analysis findings into a structured framework, incorporating functionality, usability, and scalability. In software development, design includes architecture, algorithms, and user interfaces.

➢ IMPLEMENTATION

➢ Problem: 01

Design a form for following cases:

• Five fields where three are text input fields, one is radio button, and one is select box

• Among three input field two will be mandatory and another will be hidden

➢ Answer To The Q No: 01

```
   <!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

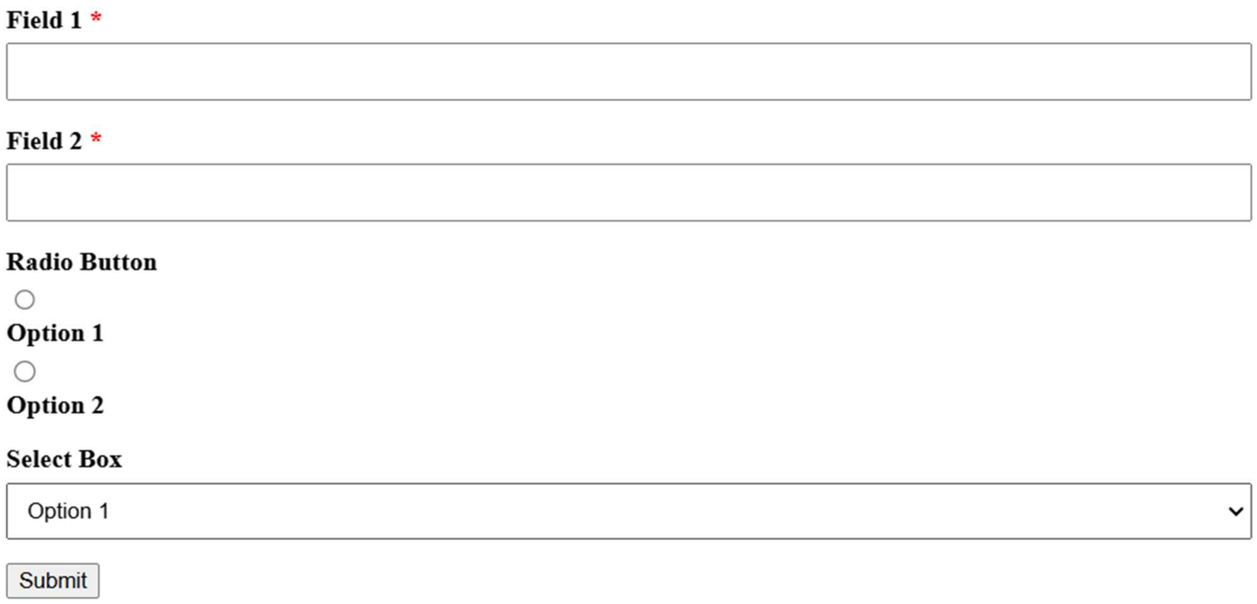
c

```
    <title>Form Design</title>
    <style>
        .hidden {
            display: none;
        }
        .form-group {
            margin-bottom: 15px;
        }
        label {
            display: block;
            margin-bottom: 5px;
            font-weight: bold;
        }
        input[type="text"], select {
            width: 100%;
            padding: 8px;
            box-sizing: border-box;
        }
        .required {
            color: red;
        }
    </style>
</head>
<body>
    <form>
        <!-- Text Input Field 1 (Mandatory) -->
        <div class="form-group">
            <label for="field1">Field 1 <span class="required">*</span></label>
            <input type="text" id="field1" name="field1" required>
        </div>

        <!-- Text Input Field 2 (Mandatory) -->
        <div class="form-group">
            <label for="field2">Field 2 <span class="required">*</span></label>
            <input type="text" id="field2" name="field2" required>
        </div>

        <!-- Text Input Field 3 (Hidden) -->
        <div class="form-group hidden">
            <label for="field3">Field 3</label>
            <input type="text" id="field3" name="field3">
        </div>

        <!-- Radio Button -->
        <div class="form-group">
            <label>Radio Button</label>
            <input type="radio" id="radio1" name="radio" value="option1">
            <label for="radio1">Option 1</label>
            <input type="radio" id="radio2" name="radio" value="option2">
            <label for="radio2">Option 2</label>
        </div>

        <!-- Select Box -->
        <div class="form-group">
            <label for="select">Select Box</label>
            <select id="select" name="select">
                <option value="option1">Option 1</option>
                <option value="option2">Option 2</option>
                <option value="option3">Option 3</option>
```

c

```
          </select>
        </div>

        <!-- Submit Button -->
        <div class="form-group">
          <button type="submit">Submit</button>
        </div>
    </form>
</body> </html>
```

- **Output:**



Field 1 *

Field 2 *

Radio Button

○ **Option 1**

○ **Option 2**

Select Box

Option 1

Submit

- **Problem: 02**
- **Answer To The Question NO: 02**

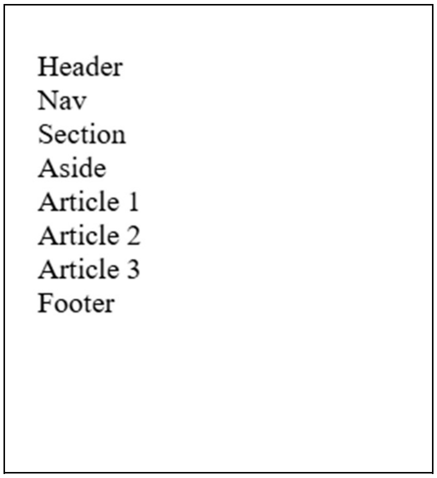
1. CSS Grid Layout

```
               <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CSS Grid Layout</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="grid-container">
        <header>Header</header>
```

c

```
        <nav>Nav</nav>
        <section>Section</section>
        <aside>Aside</aside>
        <article>Article 1</article>
        <article>Article 2</article>
        <article>Article 3</article>
        <footer>Footer</footer>
    </div>
</body>
</html>
```

Output:



```
Header
Nav
Section
Aside
Article 1
Article 2
Article 3
Footer
```

2. CSS (styles.css):

```
         body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.grid-container {
    display: grid;
    grid-template-areas:
        "header header"
        "nav nav"
        "section aside"
        "article article"
        "footer footer";
    grid-gap: 10px;
    padding: 10px;
}

header {
    grid-area: header;
    background-color: #f4a261;
    padding: 20px;
    text-align: center;
}
```

c

```
nav {
    grid-area: nav;
    background-color: #2a9d8f;
    padding: 20px;
    text-align: center;
}

section {
    grid-area: section;
    background-color: #e9c46a;
    padding: 20px;
}

aside {
    grid-area: aside;
    background-color: #e76f51;
    padding: 20px;
}

article {
    grid-area: article;
    background-color: #f4a261;
    padding: 20px;
}

footer {
    grid-area: footer;
    background-color: #264653;
    color: white;
    padding: 20px;
    text-align: center;
}

/* Responsive Design */
@media (max-width: 768px) {
    .grid-container {
        grid-template-areas:
            "header"
            "nav"
            "section"
            "aside"
            "article"
            "footer";
    }
}
```

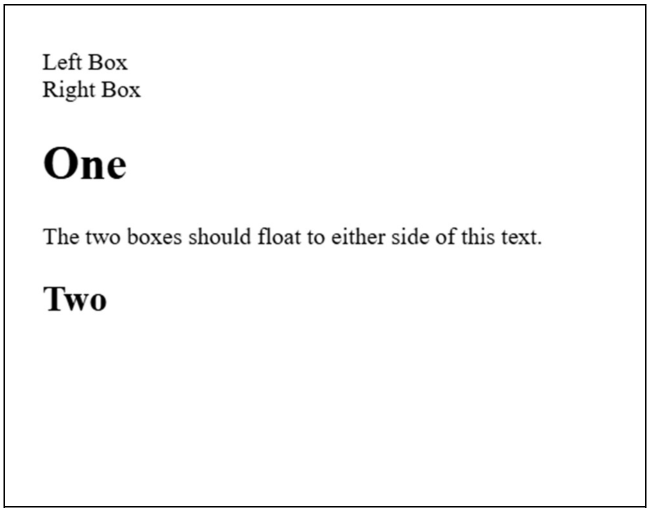
2. Float Layout

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Float Layout</title>
```

c

```
    <link rel="stylesheet" href="float-styles.css">
</head>
<body>
    <div class="container">
        <div class="box left">Left Box</div>
        <div class="box right">Right Box</div>
        <div class="content">
            <h1>One</h1>
            <p>The two boxes should float to either side of this text.</p>
            <h2>Two</h2>
        </div>
    </div>
</body>
</html>
```

➢ **Output:**



Left Box
Right Box

One

The two boxes should float to either side of this text.

Two

➢ **CSS (float-styles.css):**

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.container {
    width: 100%;
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
}

.box {
```

c

```
    width: 45%;
    padding: 20px;
    background-color: #e9c46a;
    text-align: center;
    margin-bottom: 20px;
}

.left {
    float: left;
}

.right {
    float: right;
}

.content {
    clear: both;
    padding: 20px;
    background-color: #f4a261;
}

/* Responsive Design */
@media (max-width: 768px) {
    .box {
        width: 100%;
        float: none;
    }
}
```

➢ ANALYSIS AND DISCUSSION :

- CSS is a fundamental tool for web development, enabling developers to create visually appealing and responsive designs.
- CSS Grid and Flexbox are the preferred layout techniques for modern web development due to their flexibility and responsiveness.
- Float layouts, while still relevant for simpler designs, are being phased out in favor of more advanced techniques.
- Implementing responsive design principles ensures that web pages are accessible and functional across all devices, enhancing user experience and engagement.

c

c

c

c