Module 2: CSS3 (Classes 4–8)

Module Duration: 10 Hours (5 Classes × 2 Hours)

Software: Visual Studio Code, Google Chrome

Module Learning Objectives

By the end of this module, students will be able to:

- Understand the purpose of CSS in web design.
- Apply CSS to style HTML elements.
- Work with colors, fonts, backgrounds, and borders.
- Understand the CSS Box Model.
- Create responsive layouts using Flexbox and CSS Grid.
- Add animations and transitions to web pages.
- Build responsive websites using Media Queries.

Class 4 (2 Hours)

Introduction to CSS3

Learning Outcomes

After completing this class, students will be able to:

- Understand the role of CSS.
- Apply CSS using different methods.
- Style text, colors, and backgrounds.
- Organize CSS code following best practices.

4.1 What is CSS?

CSS (Cascading Style Sheets) is a stylesheet language used to control the appearance and layout of HTML documents.

HTML provides the structure, while CSS controls the visual presentation.

Benefits of CSS

- Improves website appearance
- Separates content from design
- Makes websites responsive
- Reduces code duplication
- Ensures consistent styling across pages

4.2 Types of CSS

Inline CSS

Applied directly to an HTML element.

```
<p style="color:blue;">Welcome to CSS</p>
```

Internal CSS

Defined inside the `<style>` tag.

```
<style>
h1{
    color:blue;
}
</style>
```

External CSS (Recommended)

Stored in a separate `.css` file.

HTML

```
<link rel="stylesheet" href="style.css">
```

CSS

```
h1{
    color:blue;
}
```

4.3 CSS Syntax

```
selector{
    property:value;
}
```

Example:

```
h1{
    color:red;
    font-size:40px;
}
```

4.4 CSS Selectors

- Universal Selector (`*`)
- Element Selector (`h1`)
- Class Selector (`.title`)
- ID Selector (`#header`)
- Group Selector (`h1,p`)
- Descendant Selector (`div p`)

4.5 Colors

Methods of defining colors:

- Color Names
- HEX
- RGB
- RGBA
- HSL

Example:

```
body{
    background:#f5f5f5;
}
```

4.6 Fonts & Text Styling

Properties:

- font-family
- font-size

- font-weight
- font-style
- text-align
- text-decoration
- text-transform
- line-height

Example:

```
h1{
    font-family:Arial;
    font-size:36px;
    text-align:center;
}
```

4.7 Backgrounds

Properties:

- background-color
- background-image
- background-repeat
- background-position
- background-size

Practical Exercise

Style the HTML profile page created in Module 1 by adding:

- Background color
- Custom fonts
- Text colors
- Images
- Headings
- Paragraph styling

Assignment

Create a colorful "About Me" webpage using an external CSS file.

Class 5 (2 Hours)

CSS Box Model

Learning Outcomes

Students will be able to:

- Understand the CSS Box Model.
- Control spacing around elements.
- Design content cards and sections.
- Use borders and shadows effectively.

5.1 Box Model

Every HTML element consists of:

- Content
- Padding
- Border
- Margin

Illustration:

```
Margin
 └── Border
      └── Padding
           └── Content
```

5.2 Width & Height

```
.box{
    width:300px;
    height:200px;
}
```

5.3 Margin

Controls space outside an element.

```
margin:20px;
```

5.4 Padding

Controls space inside an element.

```
padding:15px;
```

5.5 Border

```
border:2px solid #333;
```

5.6 Border Radius

Creates rounded corners.

```
border-radius:10px;
```

5.7 Box Shadow

```
box-shadow:0 4px 10px rgba(0,0,0,.2);
```

5.8 Display Property

- block
- inline
- inline-block
- none

Practical Exercise

Create three responsive information cards containing:

- Image
- Title
- Description
- Button

Assignment

Design a simple product card.

Class 6 (2 Hours)

Flexbox & CSS Grid

Learning Outcomes

Students will be able to:

- Build flexible layouts using Flexbox.
- Create complex page structures with CSS Grid.
- Align content professionally.

6.1 Introduction to Flexbox

Flexbox is used to arrange items in one dimension (row or column).

```
.container{
    display:flex;
}
```

6.2 Flex Properties

- flex-direction
- justify-content
- align-items
- flex-wrap
- gap

Example:

```
.container{
    display:flex;
```

```
    justify-content:space-between;
    align-items:center;
}
```

6.3 CSS Grid

Grid is used for two-dimensional layouts (rows and columns).

```
.container{
    display:grid;
    grid-template-columns:repeat(3,1fr);
    gap:20px;
}
```

6.4 Position Property

- static
- relative
- absolute
- fixed
- sticky

6.5 Overflow

- visible
- hidden
- auto
- scroll

6.6 Z-index

Controls stacking order of overlapping elements.

Practical Exercise

Build a responsive webpage layout with:

- Header
- Navigation
- Sidebar
- Main Content
- Footer

Assignment

Create a responsive photo gallery using CSS Grid.

Class 7 (2 Hours)

CSS Animation & Effects

Learning Outcomes

Students will be able to:

- Add animations to web pages.
- Create smooth transitions.
- Apply modern visual effects.

7.1 CSS Transition

```
button{
    transition:.3s;
}
```

Hover example:

```
button:hover{
    background:blue;
    color:white;
}
```

7.2 CSS Transform

Properties:

- scale()
- rotate()
- translate()
- skew()

Example:

```
.card:hover{
    transform:scale(1.05);
}
```

7.3 CSS Animation

Basic animation:

```
@keyframes fadeIn{
    from{opacity:0;}
    to{opacity:1;}
}

.box{
    animation:fadeIn 2s;
}
```

7.4 Shadows

- text-shadow
- box-shadow

7.5 Gradients

Types:

- Linear Gradient
- Radial Gradient

Example:

```
background:linear-gradient(90deg,#007BFF,#00C6FF);
```

Practical Exercise

Create an animated landing page with:

- Hero section
- Call-to-action button
- Hover effects
- Animated cards

Assignment

Design a modern login page using transitions and animations.

Class 8 (2 Hours)

Responsive Web Design

Learning Outcomes

Students will be able to:

- Create websites that adapt to different screen sizes.
- Apply media queries effectively.
- Follow a mobile-first design approach.

8.1 What is Responsive Web Design?

Responsive web design ensures that websites display properly on desktops, tablets, and mobile devices.

8.2 Viewport Meta Tag

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

8.3 Media Queries

Example:

```
@media (max-width:768px){

.container{
    flex-direction:column;
}

}
```

8.4 Breakpoints

Common screen widths:

- Mobile: up to 576px
- Tablet: 577–768px
- Laptop: 769–992px
- Desktop: 993px and above

8.5 Mobile-First Design

Best practices:

- Design for small screens first.
- Use flexible layouts.
- Optimize images.
- Keep navigation simple.
- Test on multiple devices.

8.6 Responsive Images

```
img{
    max-width:100%;
    height:auto;
}
```

Practical Exercise

Convert the portfolio website into a responsive website by:

- Adding media queries
- Using Flexbox/Grid
- Adjusting typography and images for smaller screens

Module Project

Students will build a **Responsive Personal Portfolio Website** that includes:

- Header and Navigation
- About Section
- Skills Section
- Services Section
- Project Gallery
- Contact Section
- Footer
- Responsive Layout
- Animations
- Hover Effects

Module Assessment

Assessment	Marks
CSS Fundamentals Quiz	10
Practical Exercises	20
Assignments	20
Responsive Portfolio Project	40
Presentation & Viva	10
Total	**100**

Learning Outcomes

After completing **Module 2: CSS3**, students will be able to:

- Apply CSS3 to create visually appealing web pages.
- Use selectors, colors, typography, and backgrounds effectively.

- Design layouts using the CSS Box Model, Flexbox, and Grid.
- Enhance user experience with transitions, animations, and visual effects.
- Build responsive, mobile-friendly websites using media queries.
- Create modern, professional web interfaces that are ready for integration with Bootstrap and JavaScript in the next modules.