**1: CSS Selectors & Text Styling**

**AIM**: Create a webpage that demonstrates the use of different CSS selectors (element, class, ID, universal, attribute, pseudo-class).

**Expected Output**: Students can distinguish selector types and apply them effectively to style text and headings.

**Description**: Demonstrates CSS selectors (element, class, id, attribute, pseudo-class) for text formatting.

**Process**:

Create an HTML file with headings, paragraphs, and links.

Apply styles using different selectors.

Save as selectors.html and open in browser.

**Analysis**: The program shows how different selectors control styling. Useful for targeting elements precisely in larger projects.

**2: Colors, Borders & Backgrounds**

**AIM**: Build a profile card with customized borders, rounded corners, gradients, background images, and shadows.

**Expected Output**: Apply visual design principles to create attractive components.

**Description**: Builds a profile card with colors, rounded borders, gradients, and shadows.

**Process**:

Create a <div> with profile details.

Use CSS border, border-radius, background, box-shadow.

Save as profile.html.

**Analysis**: Teaches how CSS improves aesthetics and readability with design elements.

**3: CSS Box Model & Layouts**

**AIM**: Design a simple product showcase page using the box model (margin, padding, border, content). Add spacing between multiple product cards.

**Expected Output**: Understand how spacing and alignment work in CSS.

**Description**: Demonstrates margin, padding, border, content using product cards.

**Process**:

Create multiple <div> elements.

Apply margin (outside space), padding (inside space), border.

Save as boxmodel.html.

**Analysis**: Helps understand spacing and layout fundamentals, crucial for UI design.

**4: Flexbox Layout**

**AIM**: Create a responsive **navigation bar** and a **gallery layout** using display: flex;.

**Expected Output**: Learn responsive and dynamic layouts using flexbox properties.

**Description**: Creates a responsive navigation bar using Flexbox.

**Process**:

Add <nav> with links.

Apply display: flex; justify-content: space-around;.

Save as flexbox.html.

**Analysis**: Flexbox is powerful for aligning items in modern UIs (e.g., navbars, galleries).

**5: CSS Grid Layout**

**AIM**: Develop a **dashboard layout** (header, sidebar, main content, footer) using CSS Grid.

**Expected Output**: Apply grid properties for complex, responsive web layouts.

**Description**: Builds a dashboard layout (header, sidebar, main, footer) using Grid.

**Process**:

Create a grid-container.

Use grid-template-areas to define layout.

Save as grid.html.

**Analysis**: Grid provides control over multi-section layouts. Essential for dashboards, portals.

**6: CSS Transitions**

**AIM**: Design a **button hover effect** with smooth transitions for color, size, and shadow.

**Expected Output**: Implement interactive design with CSS transitions.

**Description**: Adds smooth hover effects to a button.

**Process**:

Create a <button>.

Apply transition: all 0.4s ease;.

Save as transitions.html.

**Analysis**: Enhances UX by making interactions smooth instead of sudden.

**7: CSS Animations**

**AIM**: Animate a bouncing ball or loading spinner using @keyframes.

**Expected Output**: Gain skills in creating dynamic, visually appealing animations.

**Description**: Creates a bouncing ball animation using @keyframes.

**Process**:

Create a <div class="ball">.

Apply animation: bounce 2s infinite;.

Save as animations.html.

**Analysis**: Animations make web pages lively and engaging without JavaScript.

**8: Responsive Web Design (Media Queries)**

**AIM**: Create a **portfolio webpage** that adjusts layout for mobile, tablet, and desktop using media queries.

**Expected Output**: Build adaptive web designs for multiple devices.

**Description**: Builds a portfolio page responsive for mobile & desktop.

**Process**:

Create cards inside .container.

Use flex-wrap and @media (max-width: 600px).

Save as responsive.html.

**Analysis**: Critical for mobile-first web design; ensures accessibility across devices.

**9: CSS Variables & Custom Properties**

**AIM**: Implement a **dark mode/light mode switcher** using CSS variables.

**Expected Output**: Understand reusable CSS values and theme management.

**10: CSS Framework Integration**

**AIM**: Rebuild the same webpage using **Bootstrap / TailwindCSS** and compare it with pure CSS implementation.

**Expected Output**: Learn the difference between custom CSS and framework-based development.

**11: CSS Shapes & Clip Paths**

**AIM**: Create a **creative landing page banner** using clip-path and CSS shapes.

**Expected Output**: Explore advanced styling techniques for modern UIs.

**12: CSS Transformations**

**AIM**: Implement a **3D rotating image gallery** using transform properties.

**Expected Output**: Build interactive 2D/3D effects.

**13: CSS Positioning**

**AIM**: Design a **sticky navigation bar** and an **absolute-positioned image overlay** on a hero banner.

**Expected Output**: Differentiate between relative, absolute, fixed, and sticky positioning.

**14: Advanced Responsive Layout**

**AIM**: Create a **multi-column news portal** that rearranges dynamically based on screen size.

**Expected Output**: Master combining grid, flexbox, and media queries.

**15: Mini Project**

**AIM**: Develop a **responsive e-commerce product page** (with cards, hover animations, navigation, footer).

**Expected Output**: Integrate all CSS skills into a real-world project.