ASSIGNMENT 9

Instructions for Students

● Create a separate HTML and CSS file for each question. ● Do not use JavaScript or any external CSS frameworks. ● Use semantic HTML5 tags (header, main, section, footer, etc.). ● Keep HTML and CSS neat, well-indented, and properly commented. ● Use only Flexbox or Grid for layout alignment where required ● Maintain proper contrast, spacing, and responsive behavior. ● Do not work like this q1.html, q1.css use filename as question name

Q1. Responsive Image Filter Gallery

Goal:

Create a responsive image gallery using CSS Grid that demonstrates different filters.

Requirements:

- Display six images in a grid layout.
- Use `grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));`.
- Apply a different CSS filter on each image (blur, grayscale, sepia, brightness, contrast, hue-rotate).
- On hover, the filter effect should be removed with a smooth transition.
- Add image captions below each image.
- Stack all images in one column when the screen width is below 600 px.

Q2. Responsive Web Design Layout

Goal:
Build a simple homepage layout that adapts to screen size using Flexbox.

Requirements:

- Header with logo on the left and navigation links on the right.
- Main area contains three feature boxes arranged horizontally.
- Footer is centered at the bottom of the page.
- Below 750 px, the feature boxes should stack vertically. ● On hover, each box should slightly scale up with a transition effect.

Q3. Responsive Cards with Media Queries

Goal:
Use media queries to make a card layout responsive at different breakpoints.

Requirements:

- Create four cards arranged in a row using Flexbox.
- Layout changes as follows:
    - Desktop: 4 columns
    - Tablet (max-width 900 px): 2 columns
    - Mobile (max-width 600 px): 1 column

- Each card changes background color on hover.
- Add equal spacing and padding between cards.

Q4. Responsive Typography

Goal:
Demonstrate how font sizes can scale automatically using relative units and the `clamp()` function.

Requirements:

- Create a section with a large heading and a paragraph.

Use:

```
h1 { font-size: clamp(1.5rem, 4vw, 3rem); }
p { font-size: clamp(1rem, 2vw, 1.5rem); }
```

- ● Align text centrally and add balanced margins and padding. ●

Below 700 px, adjust line-height and spacing for better readability.

Q5. Filter Cards with Animation

Goal:
Combine CSS filters, hover effects, and keyframe animations in a responsive layout.

Requirements:

- Create three profile cards using Flexbox.
- Each card contains an image and text; apply a filter like grayscale or sepia to the image.
- On hover, remove the filter and lift the card using `transform: translateY(-10px);`.
- Use `@keyframes` for a fade-in animation when the page loads.
- Stack cards vertically on screens smaller than 800 px.

Q6. Responsive CSS Cheat Sheet

Goal:
Design a responsive cheat sheet page that summarizes key Responsive CSS concepts.

Requirements:

- Display topics like Filters, Media Queries, Responsive Design, and Typography in separate boxes using CSS Grid.
- Each box shows a term and its description or example.
- On large screens: grid layout with 2–3 columns.
- On small screens (below 700 px): convert to stacked rows where each heading acts as a key followed by its value (text).
- Add hover and transition effects for interactivity.

Q7. Combined Responsive Web Demo

Goal:
Integrate all responsive CSS techniques into a single webpage.

Requirements:

- Header and navigation using Flexbox.
- Main content uses Grid to include three sections: an image gallery, a text explanation of responsive design, and a cheat sheet preview.
- Footer contains contact information.
- Include hover transitions and a simple fade-in animation.
- Below 700 px, all sections should stack vertically.

Q8. Filter Comparison Layout

Goal:
Create a before-and-after image comparison using filters and responsive design.

Requirements:

- Two images displayed side-by-side on desktop.
- The right image has a CSS filter applied (e.g., brightness or contrast).
- On hover, the filter fades away smoothly.

- Below 700 px, stack both images vertically with captions.

Q9. Responsive Design Principles Summary

Goal:
Present core principles of responsive design visually.

Requirements:

- Use Grid layout to create three boxes labeled Fluid Layouts, Flexible Media, and Breakpoints.
- Each box contains a short definition and a CSS code example.
- On hover, apply a background gradient and text color change.
- Below 600 px, convert to a single-column layout.

Q10. Responsive Dashboard Summary

Goal:
Build a small dashboard demonstrating responsive design concepts in one layout.

Requirements:

- Layout includes Header, Sidebar, and Main Content using CSS Grid.
- Main Content contains cards for Filters, Typography, Media Queries, and Cheat Sheet Links.

- On hover, each card scales up slightly with a shadow effect.
- Include a simple load animation using `@keyframes`.
- Below 750 px, the sidebar shifts to the top and the content stacks below it.