Assignment #3. Advanced CSS: Flexbox & Grid

Objective

The goal of this assignment is to understand and apply advanced CSS layout techniques using Flexbox and CSS Grid. By the end, students will be able to build complex, responsive layouts without relying on floats or external frameworks. This assignment emphasizes modern layout practices, alignment, responsiveness, and design efficiency.

Assignment Instructions

1. Work in a team to complete the assignment.
2. Each team member should add a flexbox layout to all webpages they did in Assignment 2.

Rules for Submission

1. **ALL TEAM MEMBERS must submit the assignment.**
2. SUBMIT REPORT(DOCX FILE) AND YOUR PROJECT FOLDER IN ZIP FORMAT. Ensure that the project runs without errors. REPORT MUST CONTAIN:
    - Team name, member names, and group
    - All tasks with screenshots
    - Step-by-step description of achieving the goals of each task
    - URL of deployed website
3. The assignment must be submitted by the deadline, which is specified in the lms.astanait.edu.kz. Works submitted after the deadline will not be accepted! To receive a grade, you must submit by deadline and defend your work at practice lesson time.

Part 1. Flexbox

Task 1. Navigation Bar

1. Create a header section with a website (project) logo on the left and a list of links on the right.
2. Turn the header container into a flex container.
3. Align the logo and the links horizontally.
4. Apply spacing between the links using Flexbox properties.

5. Make sure the logo and links are vertically centered.

Task 2. Card Row

1. Create a container with at least three cards (each card should include an image, title, text, and a button).
2. Make the container a flex container, so the cards appear in a row.
3. Ensure all cards have equal height.
4. Add consistent gaps between the cards.
5. Add a simple hover effect (e.g., shadow, lift, or scale).

Part 2. Grid System

Task 3. Page Layout with Grid Areas

1. Set up a layout with a header, sidebar, main content, and footer.
2. Turn the parent container into a grid container.
3. Define grid rows and columns.
4. Assign grid areas so that:
    - The header spans across the top,
    - The sidebar is placed on the left,
    - The main content is on the right,
    - The footer spans across the bottom.
5. Confirm that each section fits correctly into its grid area.

Task 4. Image Gallery

1. Collect at least nine images and place them inside a gallery container.
2. Set the gallery container as a grid container.
3. Define multiple equal-width columns and rows.
4. Add consistent spacing (gaps) between images.
5. Add a hover effect - for example, display a caption overlay when the user hovers over an image.

Part 3. Final Result: Combining Flexbox & Grid

By applying Part 1 (Flexbox) and Part 2 (Grid) to your team's webpages, your project should now look like a complete responsive website.

At this stage, make sure:

- Your headers use Flexbox for navigation (logo + links).

- Your main sections use Grid for structure (e.g., content + sidebar, galleries).
- Your cards or small components use Flexbox for neat alignment of content.
- Your footers span across the bottom of the page.
- All pages have consistent spacing, alignment, and responsiveness.

Part 4. Publish Your Responsive Website

- Commit and push all changes to your project's shared repository.
- Use GitHub Pages or Netlify to rebuild and re-deploy your site.
- Submit the URL link along with your files.

Evaluation Criteria (100 points total)

➔ **Flexbox Navigation (10%)**
- ◆ Proper alignment and spacing.

➔ **Flexbox Cards (10%)**
- ◆ Equal heights, neat layout, hover effect.

➔ **Grid Layout (10%)**
- ◆ Correct use of grid areas,
- ◆ Logical placement

➔ **Image Gallery (10%)**
- ◆ Clean grid,
- ◆ Consistent spacing,
- ◆ Hover captions.

➔ **UX and Engagement (10%)**
- ◆ Integrating the project theme, ensuring user experience, and adding engaging elements.

➔ **Defense during Practice Lesson (40%)**
- ◆ Student explains assignment clearly.
- ◆ Demonstrates understanding of assignment
- ◆ Answers instructor's questions correctly.
- ◆ Completes the given task from the instructor correctly

➔ **Report (10%)**
- ◆ The report is well-written, clear, and concise.
- ◆ Includes objective, description of steps taken, screenshots of webpage, and final reflection.

- Proper formatting (headings, bullet points, or numbered sections).

Resources

1. **Abitova G.A. Web technologies Front-End Development. Part 1 (2022);**
2. **https://www.w3schools.com/css/css3_box-sizing.asp**
3. **https://www.w3schools.com/css/css3_flexbox.asp**
4. **https://css-tricks.com/snippets/css/a-guide-to-flexbox/**
5. **https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout/Basic_concepts_of_flexbox**
6. **https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout/Basic_concepts_of_grid_layout**
7. **https://flexboxfroggy.com/**

*****Questions will be based on the provided resources***

Good luck!

Note: Feel free to seek assistance from me or use online resources. You can contact me via Microsoft Teams or during Office hours if you need detailed guidance.

Please remember: contact hours are **from 9:00 AM to 8:30 PM** (Monday-Friday). Messages sent outside this time will not receive an immediate response.