**Assignment #3. Advanced CSS ( Flexbox & Grid)**

Student: Zhaskairat Ayaulym

Group: SE-2433

**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.

**Rules for Submission:**

1. The assignment must be completed individually. If you have the same paperwork with other students, you will have 0 to this work.
2. 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.
3. SUBMIT REPORT(DOC FILE) AND YOUR PROJECT FOLDER IN ZIP FORMAT. Ensure that the project runs without errors.

IN REPORT SHOULD BE:

* YOUR NAME, GROUP
* **All parts with screens(write part, task and then add screen)**
* Summary of your work process

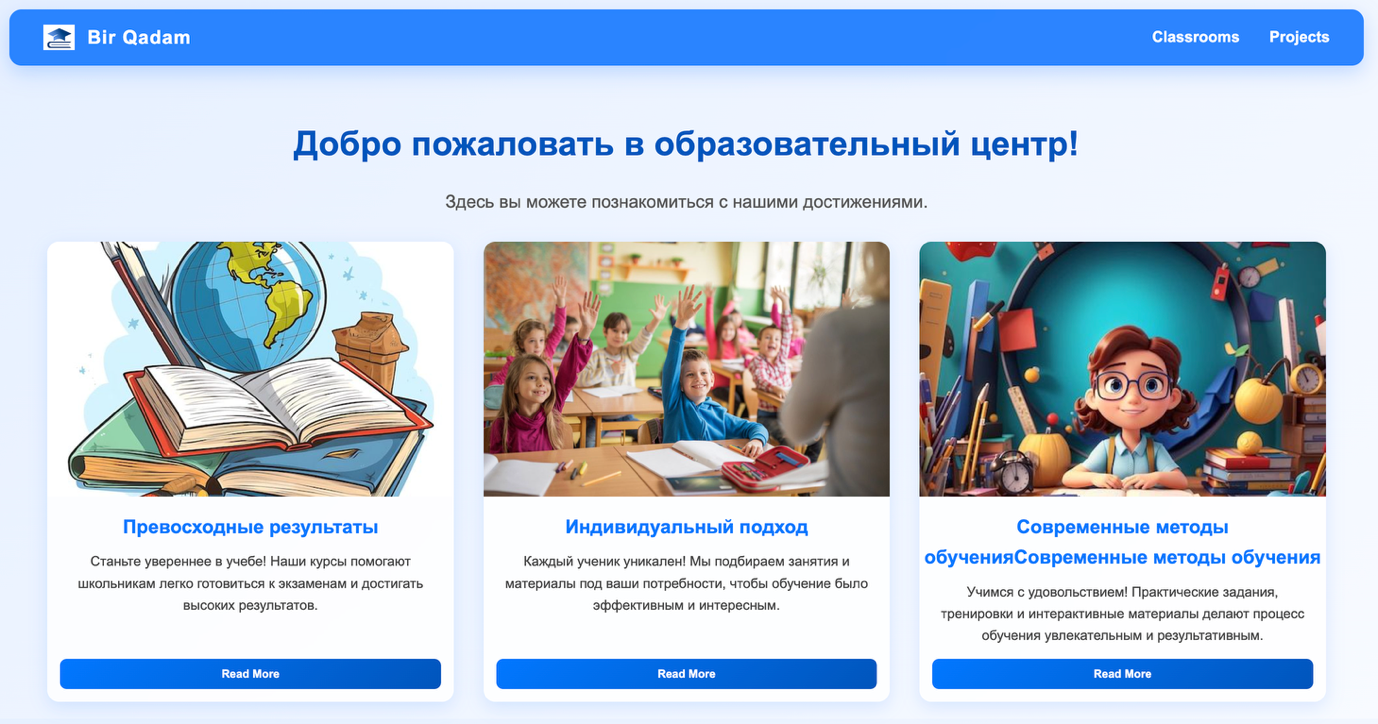
**Part 1. Flexbox**

**Task 0. Navigation Bar**

1. Create a header section with a 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 1. Card Row**

1. Create a container with at least three cards (each card should include an image, title, text, and 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).

****

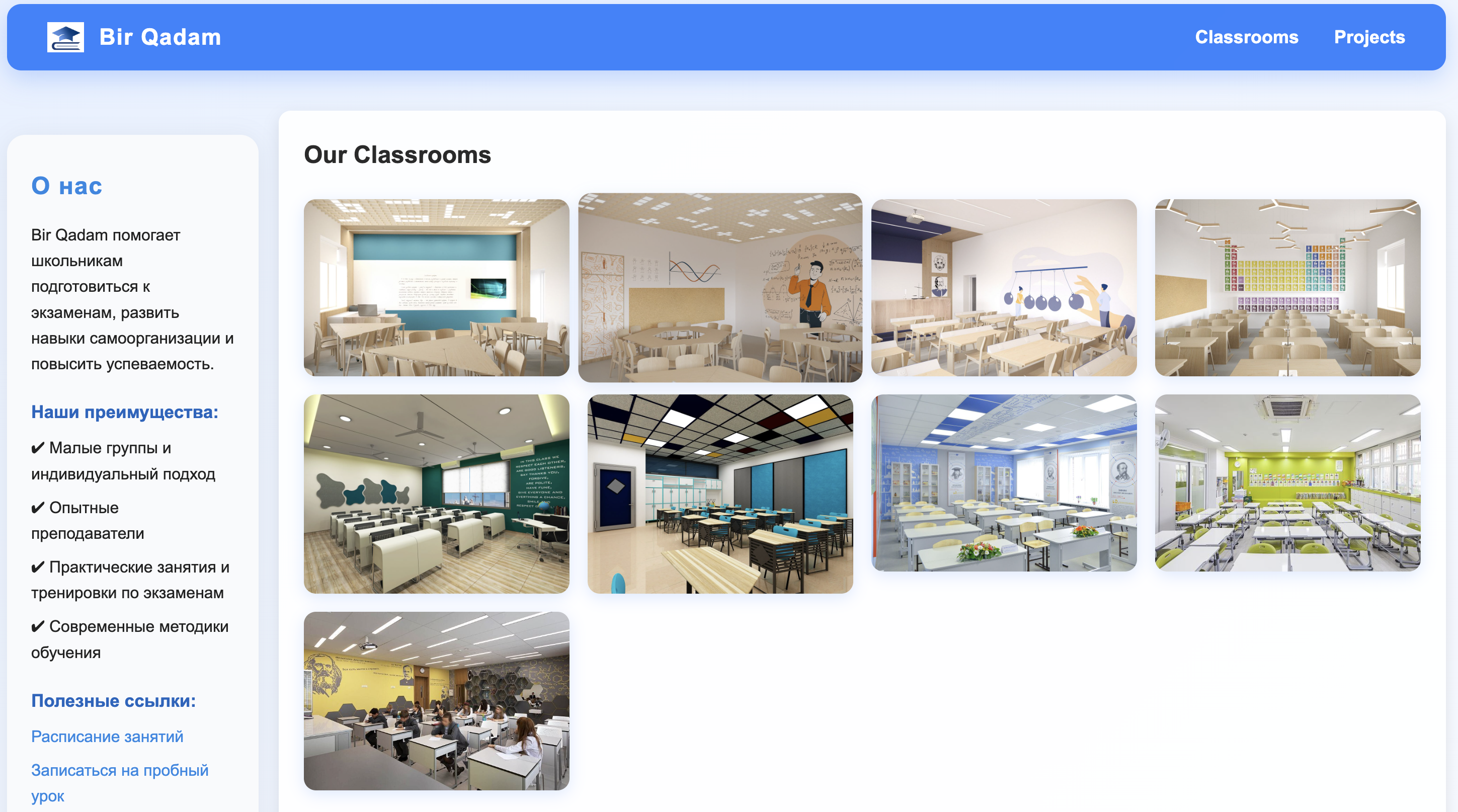
**Part 2. Grid System**

**Task 2. 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 3. Image Gallery**

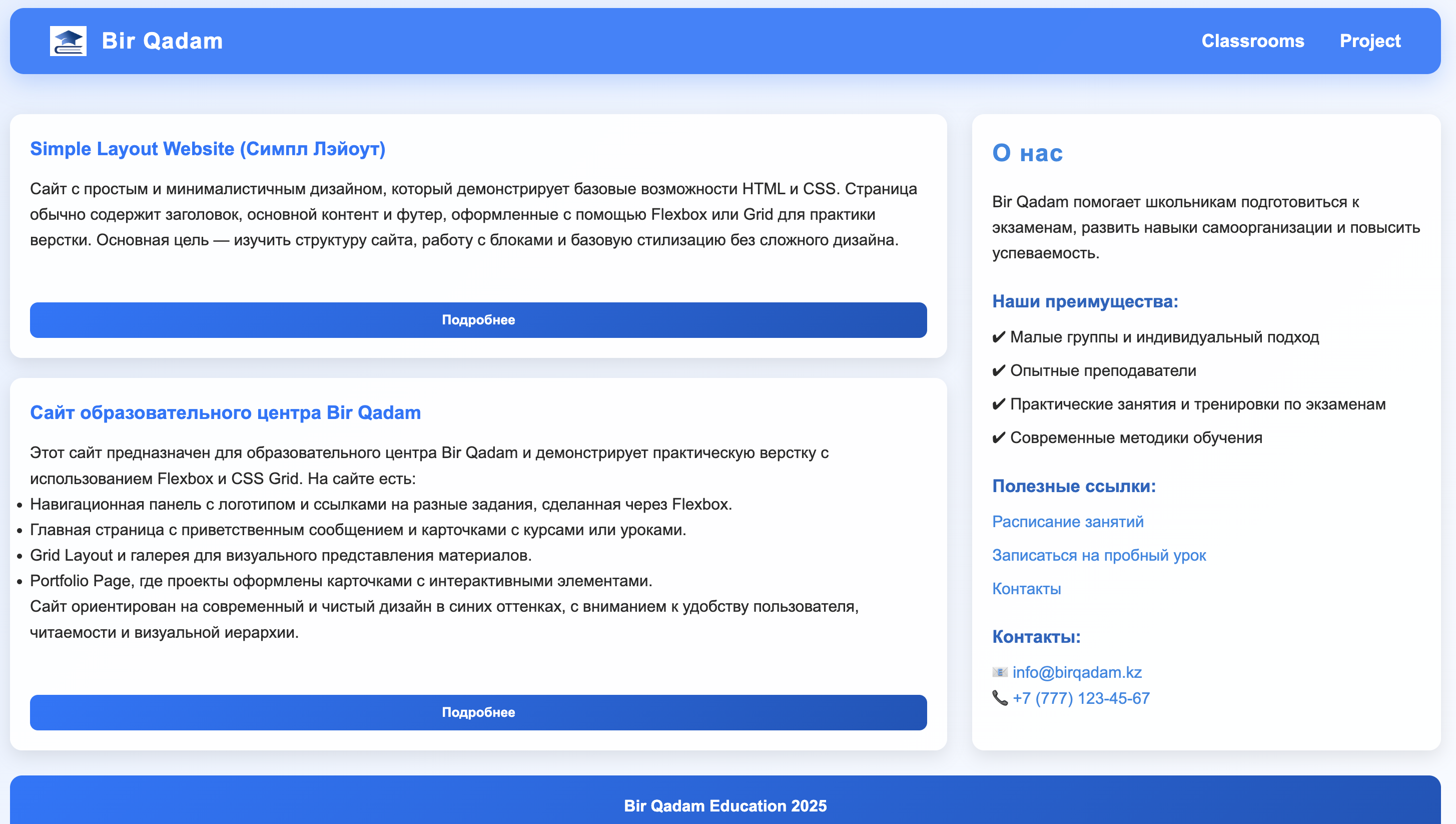
1. Collect at least nine images and place them inside a gallery container.
2. Set the gallery container as a grid container.
3. Define multiple equal-width columns and rows.
4. Add consistent spacing (gaps) between images.
5. Add a hover effect — for example, display a caption overlay when the user hovers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



**Part 3. Combining Flexbox & Grid**

**Task 4. Portfolio Page**

1. Create a page structure with: header, main section, sidebar, and footer.
2. Use **Flexbox** in the header for the navigation bar.
3. Use **CSS Grid** for the main section:
   * Projects area on the left,
   * Info or sidebar on the right.
4. Inside each project card, use Flexbox to arrange content (title, description, button).
5. Ensure the footer spans across the bottom of the page.
6. Review the page for consistency in spacing and alignment.



**Evaluation Criteria (100 points total):**

1. **Flexbox Navigation (10 points)**
   * proper alignment and spacing
2. **Flexbox Cards (10 points)**
   * equal heights, neat layout, hover effect
3. **Grid Layout (10 points)**
   * correct use of grid areas, logical placement
4. **Image Gallery (10 points)**
   * clean grid, consistent spacing, hover captions
5. **Portfolio Page (10 points)**

* effective combination of Flexbox & Grid

1. **Design & Code Quality (10 points)**

* clean structure, readability, consistent style

1. **Defense during Practice Lesson (30 points)**
   * Student explains assignment clearly.
   * Demonstrates understanding of assignment
   * Answers instructor’s questions correctly.
   * Completes the given task from the instructor correctly
2. **Report (10 points)**
   * Well-written report with all parts, screens and summary

**Resources:**

1. **Abitova G.A. Web technologies Front-End Development. Part 1 (2022);**
2. [**https://www.youtube.com/playlist?list=PLPT6CF0r4E3rkvy1rLUKdDHf\_HmWZeURW**](https://www.youtube.com/playlist?list=PLPT6CF0r4E3rkvy1rLUKdDHf_HmWZeURW)
3. [**https://www.google.com/search?client=safari&rls=en&q=flexbox+froggy&ie=UTF-8&oe=UTF-8**](https://www.google.com/search?client=safari&rls=en&q=flexbox+froggy&ie=UTF-8&oe=UTF-8)
4. [**https://www.w3schools.com/css/css3\_box-sizing.asp**](https://www.w3schools.com/css/css3_box-sizing.asp)
5. [**https://www.w3schools.com/css/css3\_flexbox.asp**](https://www.w3schools.com/css/css3_flexbox.asp)

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

**Good luck!**