**Student: Sabyrov Sanzhar  
Group: SE-2425  
Link to the GitHub:**

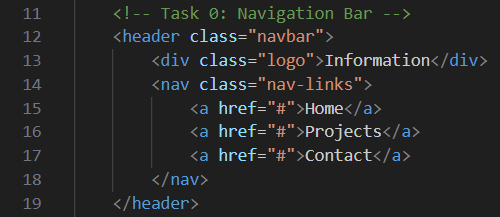
**https://github.com/SanzharS1/WEB\_ASS\_3.git**

**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.**

**Task 0: Navigation Bar**

1. **Create a header section with a logo on the left and a list of links on the right:  
   I created a header containing a logo on the left and navigation links (Home, Projects, Contact) on the right.**
2. **Turn the header container into a flex container:  
   I applied Flexbox to the header by using display: flex; to align the logo and the links horizontally.**
3. **Align the logo and the links horizontally:  
   I used justify-content: space-between; to make the logo appear on the left and links on the right.**
4. **Apply spacing between the links using Flexbox properties:  
   I added margin to the links and used justify-content to control spacing effectively.**
5. **Make sure the logo and links are vertically centered:  
   I used align-items: center; to center the items vertically within the header.**

****

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.**

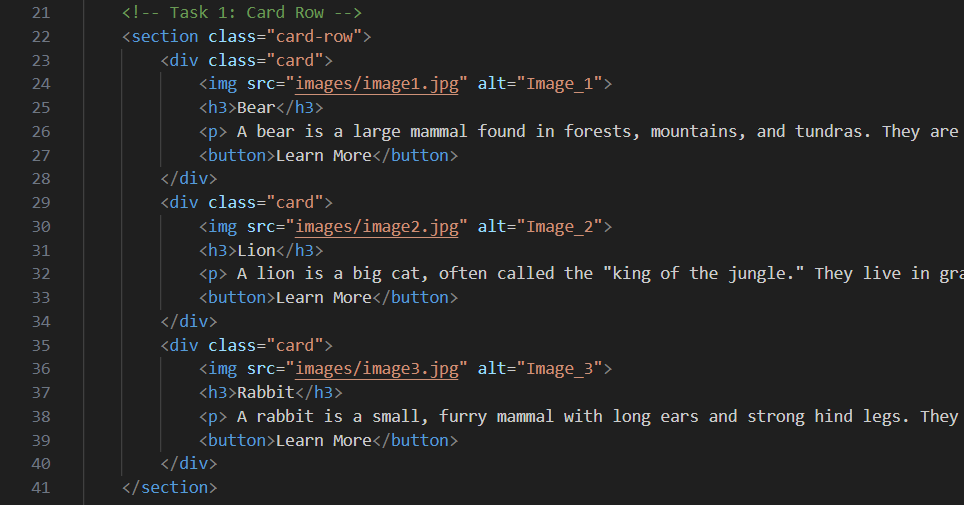
**Step 1: Card Row**

1. **Create a container with at least three cards:  
   I created a container with three cards, each including an image, title, description, and a button.**
2. **Make the container a flex container so the cards appear in a row:  
   I applied display: flex; to the container, which arranges the cards in a row.**
3. **Ensure all cards have equal height:  
   I used height: 100% and adjusted the padding to make sure all cards are of equal height.**
4. **Add consistent gaps between the cards:  
   I used gap: 20px; to create consistent spacing between the cards.**
5. **Add a simple hover effect (e.g., shadow, lift, or scale):  
   I added a hover effect using transform: scale(1.05);, which slightly enlarges the card when hovered over.**

**👇**

**👇**

**👇**

****

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Step 2: Page Layout with Grid Areas**

1. **Set up a layout with a header, sidebar, main content, and footer:  
   I used CSS Grid to structure the page into a header, sidebar, main content, and footer.**
2. **Turn the parent container into a grid container:  
   I applied display: grid; to the parent container to enable grid layout.**
3. **Define grid rows and columns:  
   I used grid-template-columns and grid-template-rows to define the layout.**
4. **Assign grid areas:  
   The header spans the top, sidebar is on the left, main content on the right, and footer spans the bottom, using grid-template-areas.**
5. **Confirm that each section fits correctly into its grid area:  
   I verified that each section aligns correctly within its defined grid area.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Step 3: Image Gallery**

1. **Collect at least nine images and place them inside a gallery container:  
   I selected and placed nine images into a gallery container.**
2. **Set the gallery container as a grid container:  
   I applied display: grid; to the gallery to create a grid layout.**

**👇**

**👇**

1. **Define multiple equal-width columns and rows:  
   I used grid-template-columns: repeat(3, 1fr); to create three equal-width columns.**
2. **Add consistent spacing (gaps) between images:  
   I used gap: 20px; to ensure consistent spacing between the images.**
3. **Add a hover effect:  
   I added a hover effect that zooms in the image when the user hovers over it, using transform: scale(1.1);.**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Step 4: Portfolio Page**

1. **Create a page structure with: header, main section, sidebar, and footer:  
   I created a full portfolio page layout using a header, main content area, sidebar, and footer.**
2. **Use Flexbox in the header for the navigation bar:  
   I applied Flexbox in the header to align the navigation links horizontally.**
3. **Use CSS Grid for the main section:  
   I used CSS Grid for the main section with a projects area on the left and an info/sidebar on the right.**
4. **Inside each project card, use Flexbox to arrange content (title, description, button):  
   Inside each project card, I applied Flexbox to arrange the title, description, and button.**
5. **Ensure the footer spans across the bottom of the page:  
   I confirmed that the footer spans the full width of the page at the bottom.**
6. **Review the page for consistency in spacing and alignment:  
   I checked the alignment and spacing to ensure consistency and neatness across the entire page.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как текст, снимок экрана, Шрифт, число

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Conclusion:**

**By completing this assignment, I gained a deeper understanding of Flexbox and CSS Grid, two modern layout techniques. Using these techniques, I was able to create flexible, responsive layouts for various web elements such as navigation bars, image galleries, and portfolio pages. The project helped me improve my ability to build structured and visually appealing web pages without relying on external frameworks, using only native CSS for layout control.**

**This experience has significantly enhanced my skills in layout design, making it easier to work with complex and responsive designs. Flexbox and CSS Grid are powerful tools that will help me tackle more advanced web design projects in the future.**