ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Н. И. Мирошниченко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3 |
| Практическое задание №3 по курсу: ОСНОВЫ FRONTEND-РАЗРАБОТКИ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4217 |  |  |  | В. А. Милованов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

**Цель работы**

Изучение принципов адаптивной верстки веб-сайтов.

**Ход работы**

1-я часть задания - использование технологии Flexbox

Требуется на выбранную вами самостоятельно тему создать веб-страницу, на которой расположить 3 flex-контейнера.

1) В первом flex-контейнере должна быть помещена горизонтальная навигационная панель с 4 гиперссылками, которые должны быть равномерно распределены по всей строке с одинаковым пространством вокруг них, пустое пространство перед первой и после последней должно быть равно половине пространства между двумя соседними элементами.



Рисунок 1 – Первый flex-контейнер



Рисунок 2 – CSS правила первого контейнера

2) Во втором flex-контейнере необходимо расположить 3 блока с текстом, причем в центральном блоке должно быть изображение и обтекающий его с правой стороны текст.

Левый и правый блоки должны иметь ширину примерно в 2 раза меньшую, чем ширина центрального блока. Блоки должны иметь такой размер по вертикали, чтобы занять всё доступное пространство flex-контейнера вдоль поперечной оси.

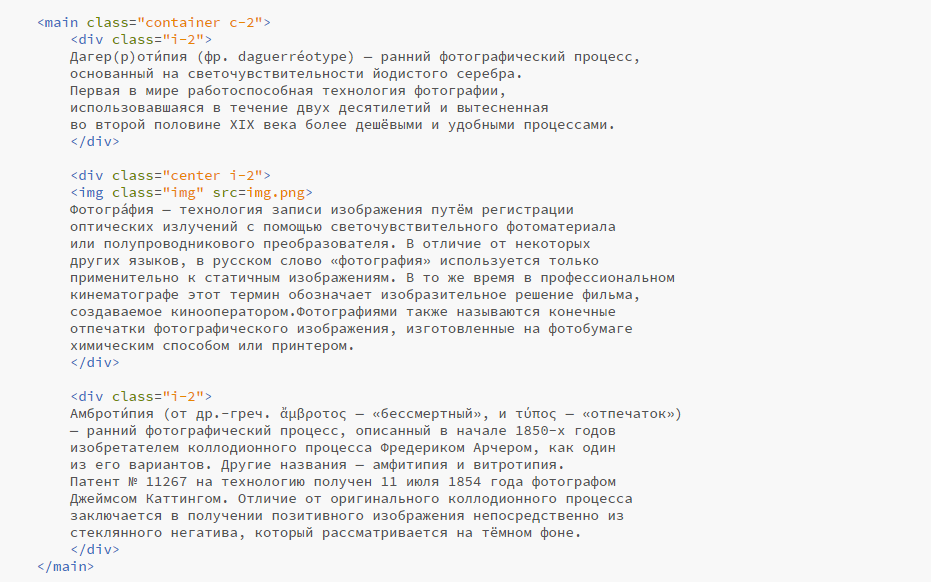


Рисунок 3 – Второй flex-контейнер

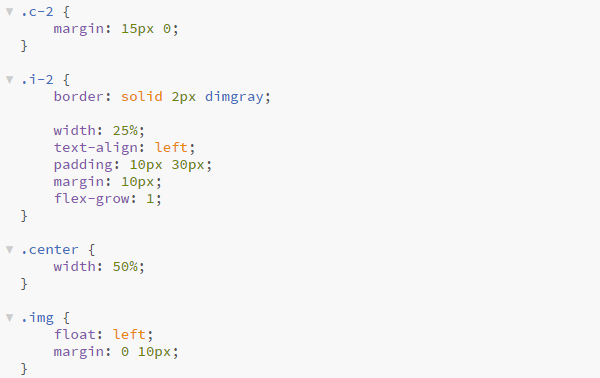


Рисунок 4 – CSS правила второго контейнера

3) В третьем flex-контейнере должен находиться подвал с контактной информацией, расположенной в двух блоках: слева телефон, справа – адрес.



Рисунок 5 – Третий flex-контейнер



Рисунок 6 – CSS правила третьего контейнера

Требования к адаптивности страницы следующие.

- При изменении ширины экрана до величины меньшей 800 px гиперссылки 1-го flex-контейнера и блоки 2-го flex-контейнера должны расположиться друг по другом, ширина блоков при этом становится одинаковой.

- Блоки подвала при сужении экрана перестраиваться не должны, они могут менять только свою ширину.

- Ширина изображения не должна становиться меньше, чем 200 px.



Рисунок 7 – Требования к адаптивности

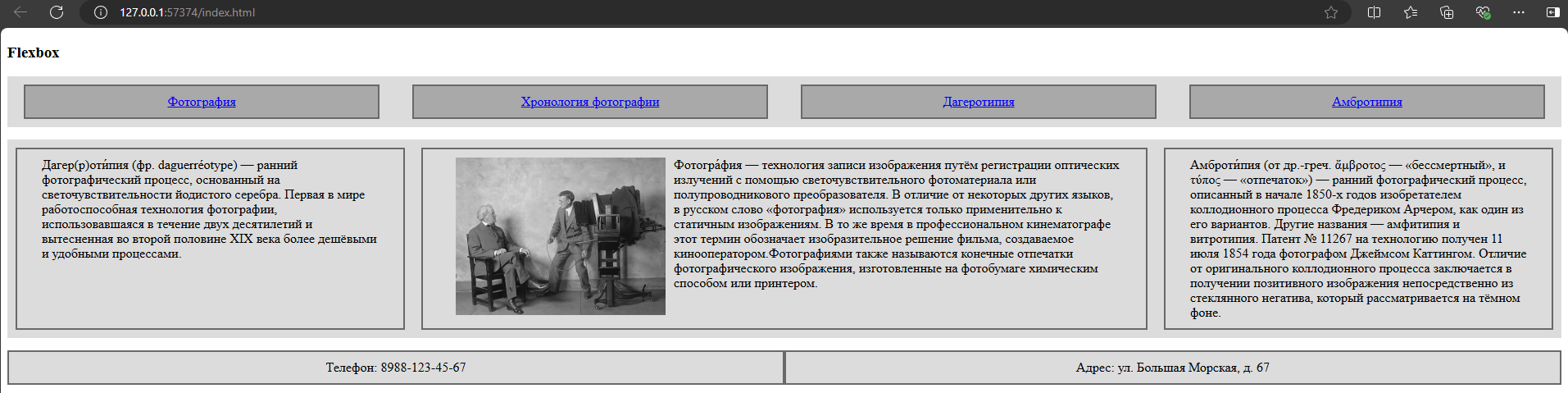


Рисунок 8 – Отображение страницы в полноэкранном режиме



Рисунок 9 – Отображение страницы >800px

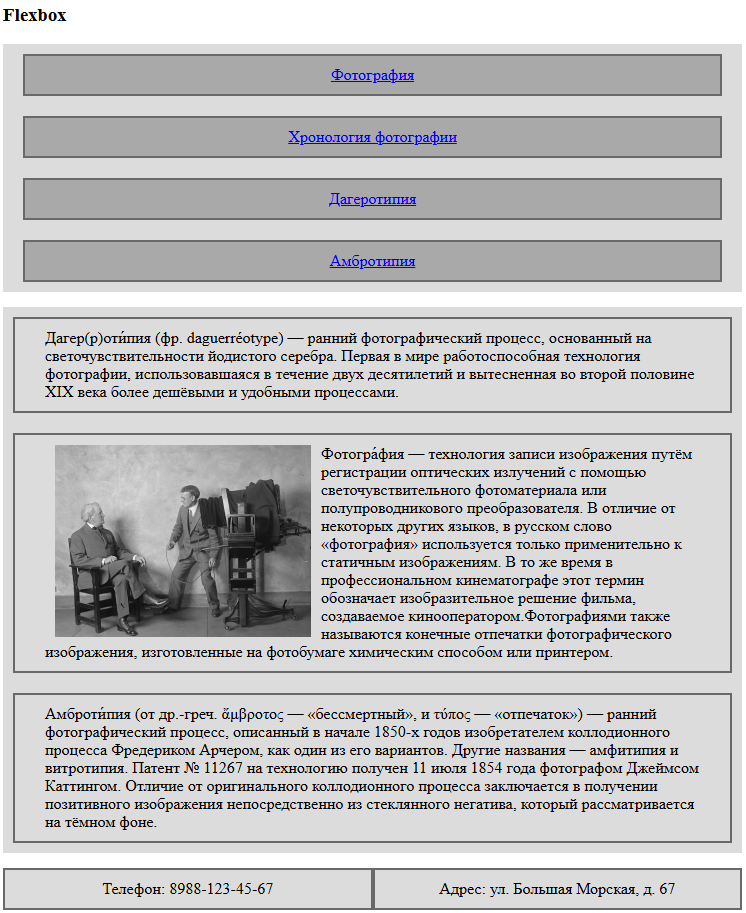


Рисунок 10 – Отображение страницы <800px

2-я часть задания - использование технологии CSS Grid

На базе технологии CSS Grid с использованием имен для областей сетки (с применением свойства grid-template-areas) требуется создать web-страницу.

Внутри блоков можно ничего не размещать кроме их названий.



Рисунок 11 – Grid-контейнер



Рисунок 12 – CSS правила grid-контейнера

Требования к адаптивности страницы – при ширине экрана меньшей 700 px все элементы страницы должны выстроиться в одну колонку друг под другом.

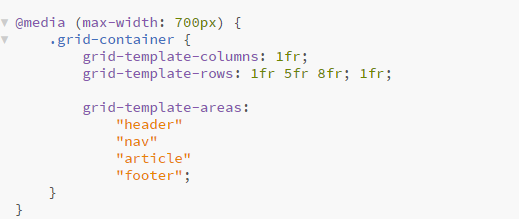


Рисунок 13 – Требования к адаптивности grid-контейнера

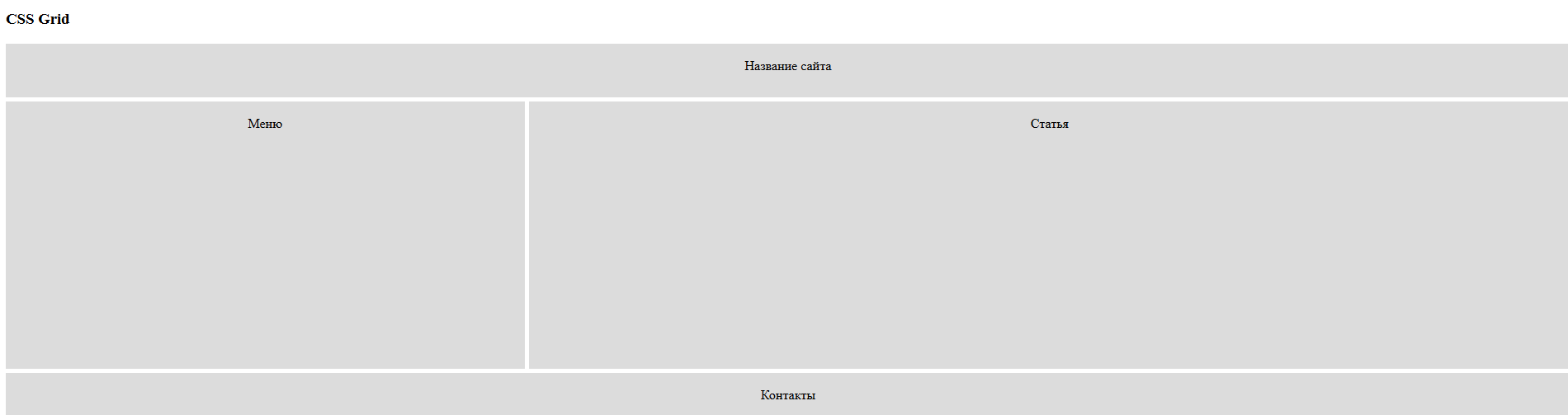


Рисунок 14 – Результат в полноэкранном режиме

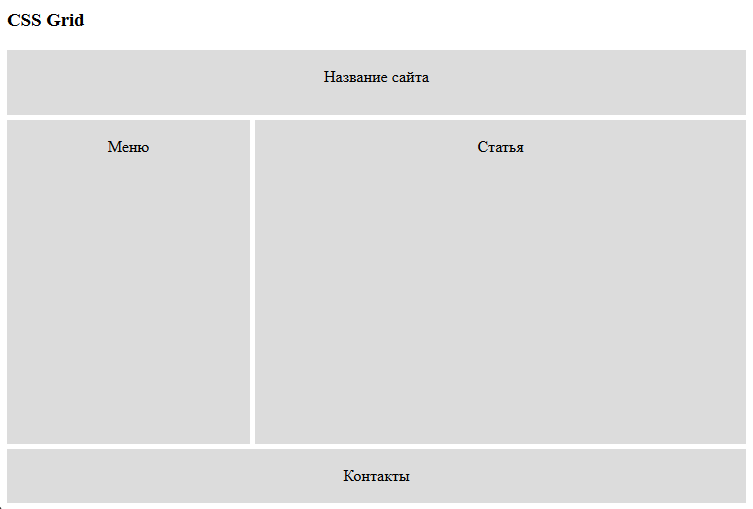


Рисунок 15 – Результат страницы >700px

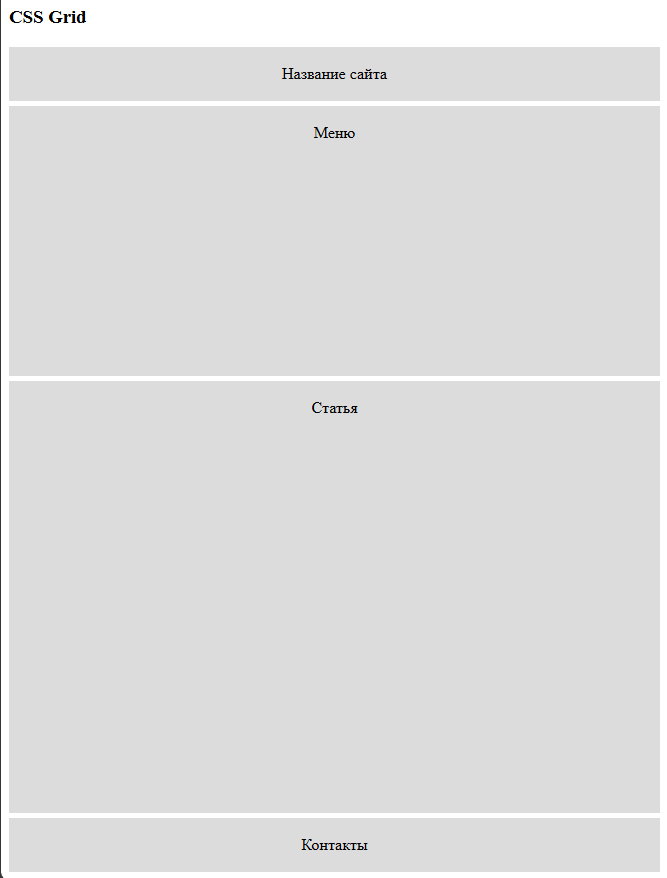


Рисунок 16 – Результат страницы <700px

**Вывод**

В ходе выполнения практической работы мной были освоены технологии flexbox и CSS grid. Были изучены и использованы следующие семантические элементы: nav для навигационной панели, main для центрального раздела, header для шапки сайта, article для раздела статьи, footer для подвала сайта.

Для работы с упомянутыми технологиями использовался CSS. В роли селекторов выступали классы. Были добавлены медиавыражения. С их помощью происходила настройка адаптивности контента при уменьшении размера окна браузера. Использовались следующие свойства для стилизации flex и grid контейнеров: flex-grow, flex-direction, grid-gap, grid-template-columns, grid-template-rows, grid-template-areas, grid-area.