**Техническая документация информационной системы**

**Техническое задание**

**Задача:**

1. Разработать лендинговую (целевую, главную) страницу
2. Адаптивность сайта (минимальная)
3. Разработать скрипт JS для нескольких блоков

**Цель:** получение знаний в процессе разработки ИС

**Разработать целевую страницу**

Первоначально, рекомендуется (необязательно) использовать редактор кода Visual Studio для более удобной работы.

Далее приступаем к написанию кода и преображению нашей целевой страницы.

Чтобы появилась адаптивность сайта к разным диагоналям монитора используем CSS GRID. GRID (Grid Layout) – представляет собой двумерную систему сеток в CSS. GRID’ы подойду и для вёрстки основных областей страницы, и небольших элементов пользовательского интерфейса. GRID представляет собой пересекающийся набор горизонтальных и вертикальных линий, образующих колонки и строки. Элементы могут быть помещены в грид в пределах линий этих колонок и строк. GRID имеет следующие особенности:

1. **Фиксированные и гибкие размеры полос**

Вы можете создать грид с фиксированным размером полос, например, использую пиксели. Это установит грид на определённый пиксель, соответствующим желаемому макету. Вы также можете создать грид с гибкими размерами, используя проценты или новую единицу измерения – «fr», разработанную для этой цели.

1. **Расположение элементов**

Вы можете размещать элементы в заданном месте на гриде использую номера строк, имена или путём привязки в области грида. Грид также содержит алгоритм управления размещением элементов, не имеющих явной позиции на гриде.

1. **Создание дополнительных полос для хранения контента**

Вы можете определить явную сетку с помощью грид-раскладки. Спецификация грид–раскладки достаточно гибкая, чтобы добавить при необходимости дополнительные строки и колонки. Также в неё включены такие возможности как, например, добавление «стольких колонок, сколько будет помещено в контейнер».

1. **Управлением перекрывающимся контентом**

В ячейку или область грида может быть помещено несколько элементов;

Эти элементы могут частично перекрывать друг друга. Такое наложение можно контролировать с помощью свойств «z-index».

Грид – это мощная спецификация, и в сочетании с другими частями CSS, таким как flexbox, поможет вам создать макеты, которые ранее невозможно было построить в CSS. Всё начинается с создания сетки в вашем грид-контейнере

**Grid – контейнер**

Мы создаём *grid* контейнер, объявляя на элементе display: grid или display: inline-grid. Как только мы это сделаем, всё прямые потомки этого элемента станут элементами сетки.

В этом примере у меня есть контейнер div с классом – container и семью дочерними элементами внутри.

Я сделал .container грид-контейнером, а также написал несколько свойств для контейнера

Все прямые потомки теперь являются грид-элементами. В браузере вы не увидите разницы с тем, как элементы отображались до помещения их в грид, поскольку грид сделан как одно колоночная сетка. На это стадии вам, возможно, будет удобнее работать с инструментом «Грид-инспектор», который является частью «Инструментов веб-разработчика». Если вы просмотрите этот пример в Yandex или Chrome и проинспектируйте грид, вы увидите маленький значок рядом с значением grid. Нажмите на него и сетка на этом элементе будет наложена в окне браузера.

На *рисунке 1* вы можете наблюдать пример с использованием GRID разметки.

По ширине, контейнер имеет 3 столбца, а в высоту 5 столбца. Несмотря на то, что в контейнере показано всего 3 элемента в высоту, 5 столбцов нужно для корректной разметки.

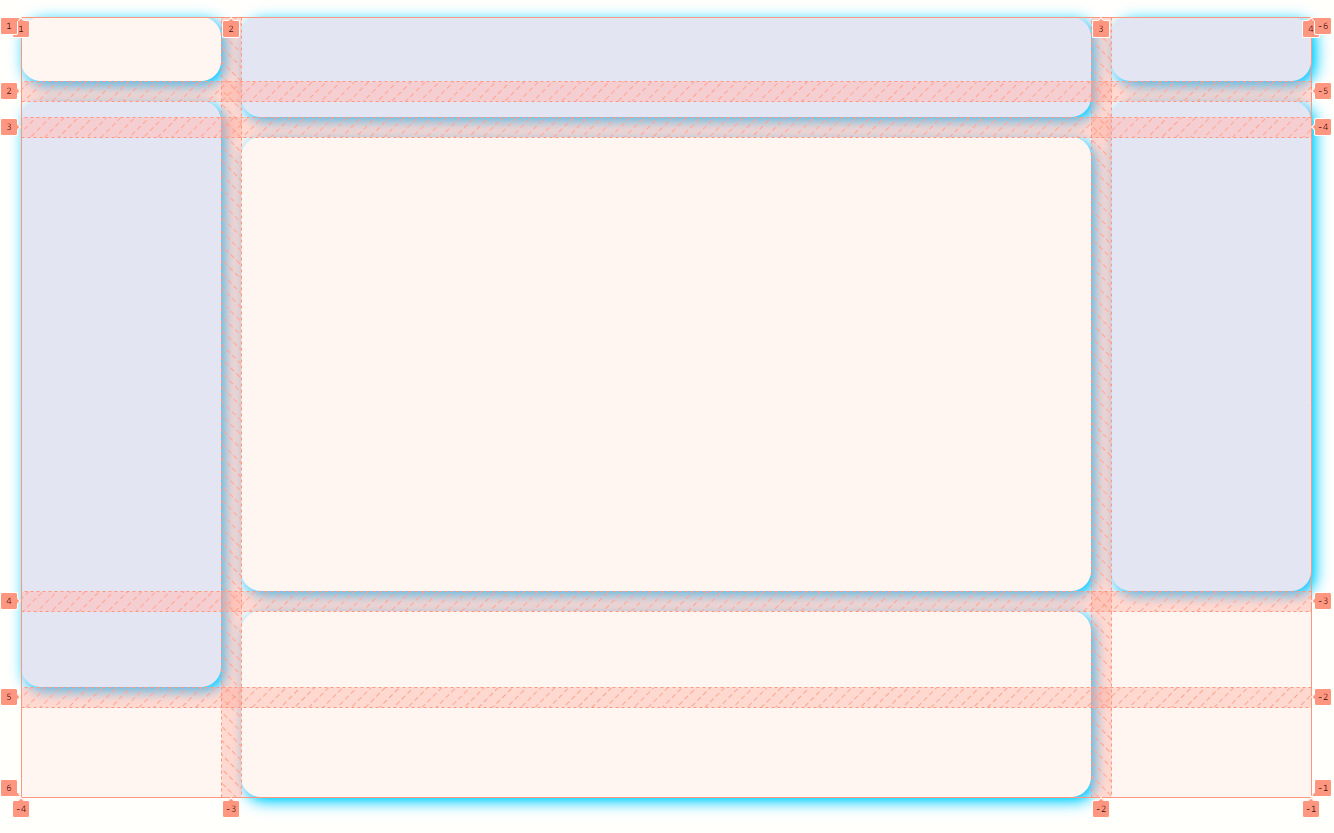


Рисунок 1.Пример GRID разметки.

**CSS – стили**

Cascading Style Sheets (CSS) – язык иерархических правил, используемый для представления внешнего вида документа, написанного на HTML. CSS описывает, каким образом элемент должен отображаться на экране, на бумаге, голосом или с использованием других медиа средств.

CSS является одним из основных языков свободной веб-разработки, который стандартизирован спецификацией W3C. Стандарт CSS делится на уровни: CSS1 в настоящее время устарел, CSS2.1 – рекомендован для применения, а CSS3, разбитый на более мелкие модули, развивается на пути стандартизации. Всю полную документацию по CSS вы можете посмотреть в интернете.

Для примера, покажу CSS стили на готовой GRID разметке. Ознакомитесь с *рисунком 2*

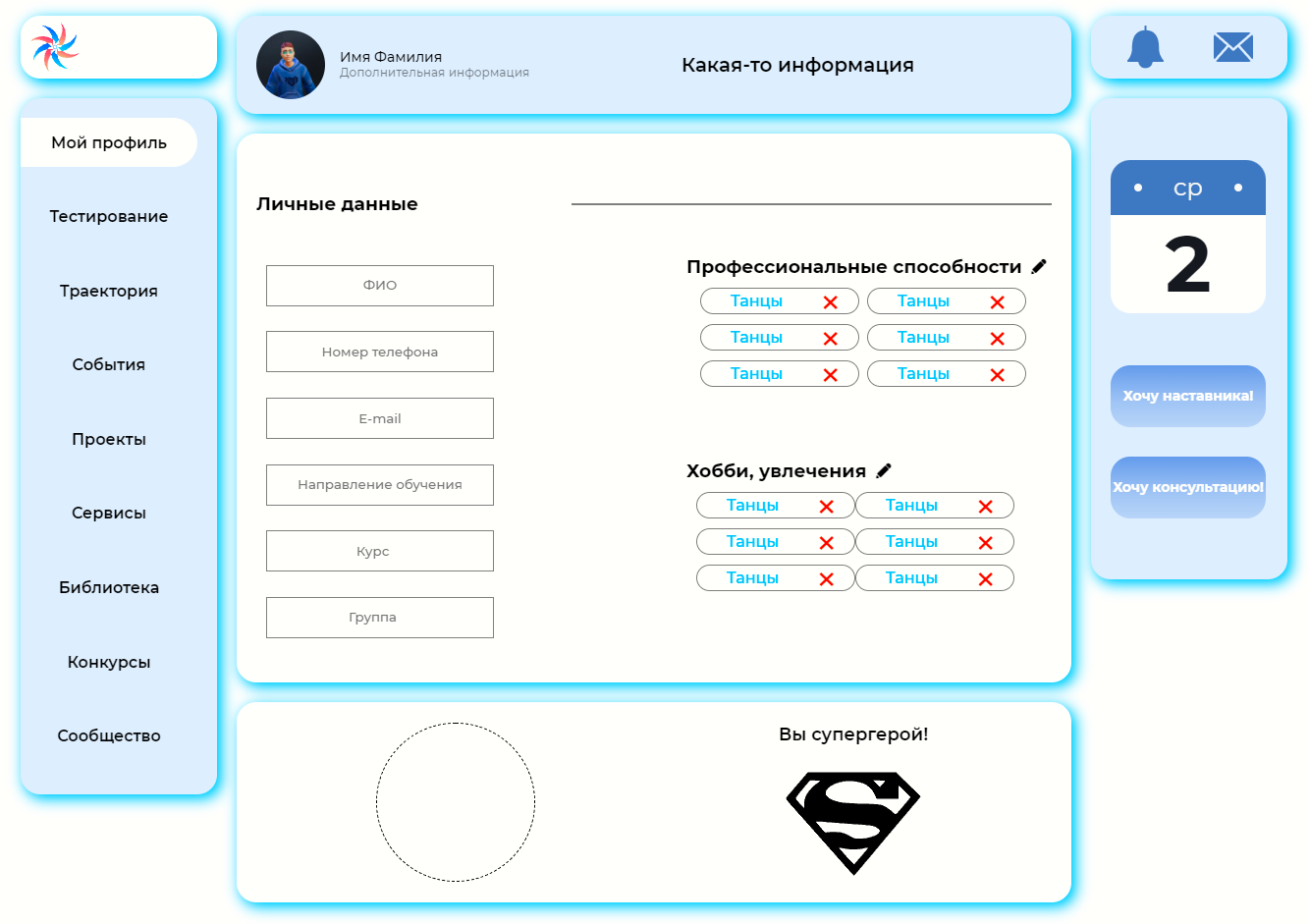
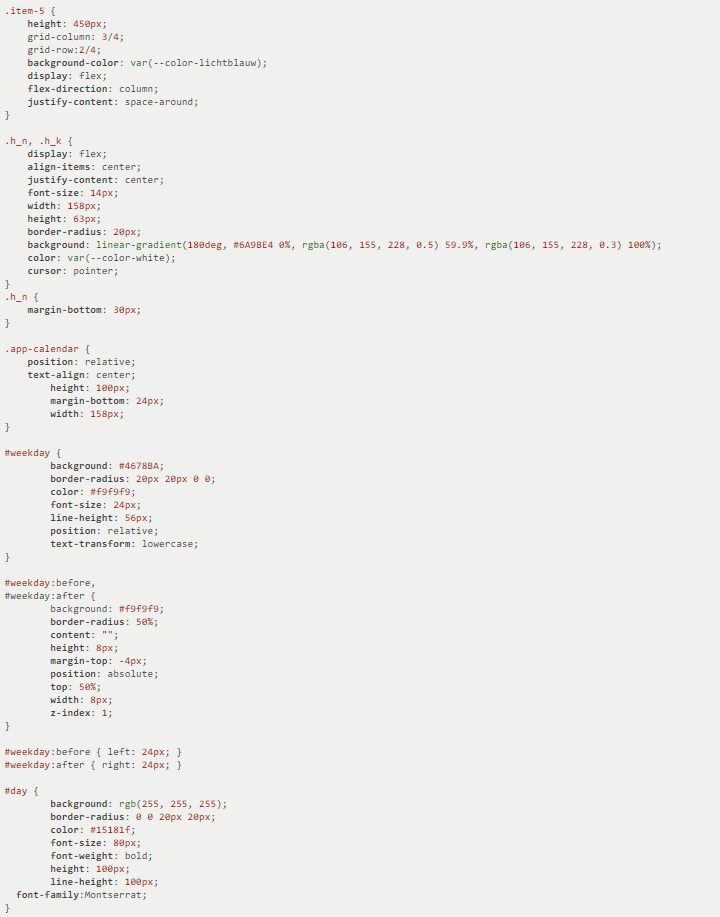
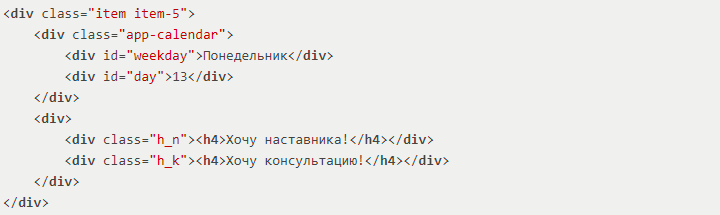
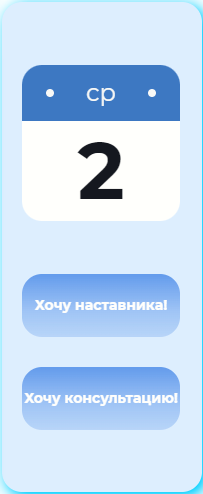


Рисунок 2.CSS стили

С помощью CSS мы указали цвет текста, цвет фона, шрифт, размер текста и т.д. Грубо говоря, вы сами решаете, как будет выглядит ваш сайт. Чтобы детально узнать, какое бывает волшебство с помощью CSS, ознакомитесь с макетами на разных сайтах.

Также, покажу пример CSS стилей на одном из блоков.



**JavaScript**

JavaScript (JS) – легковесный интерпретируемый или JIT – компилируемый, объектно-ориентированный язык с функциями первого класса. Наиболее широкое применение находит как язык сценариев веб-страницы, но также используется и в других программных продуктах, например, node.js или Apache CoucheDB. JavaScript это прототипно-ориентированный, мультипарадигменный язык с динамической типизацией, который поддерживает объектно-ориентированный, императивный и декларированный стиль программирования.

Для сайта, я также написал несколько сценариев, которые создают функционал.

На *рисунке 3* вы можете наблюдать сценарий, при котором происходит событие клика на карандаш, после которого открывается PopUp окно.

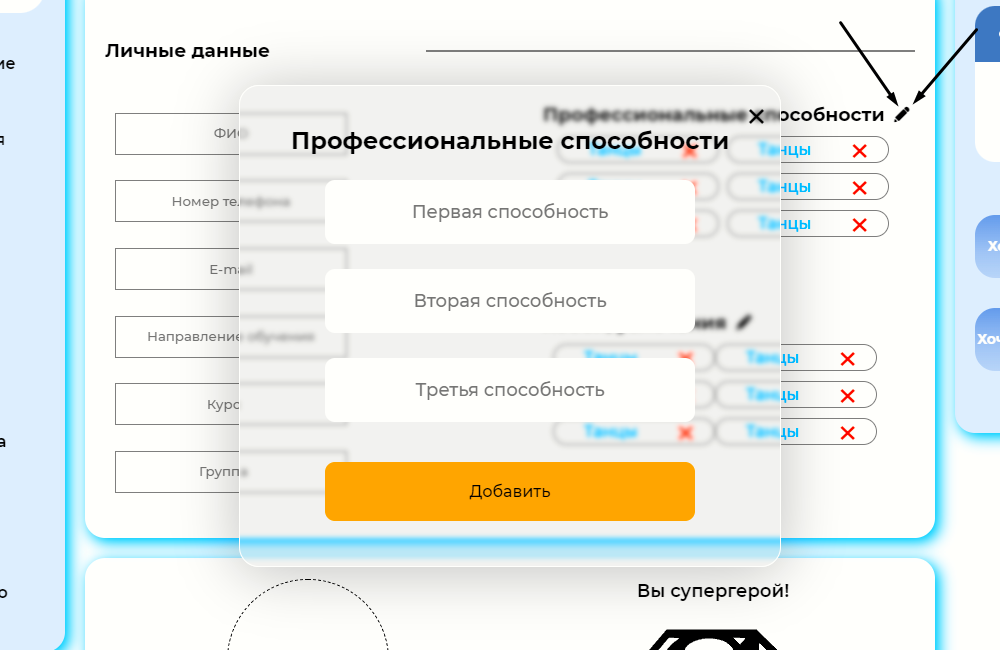


Рисунок 3.JS

Для верстки всего сайта, я использовал такие теги, стили и скрипты:

Теги: div, ul, li, button, img, h2, h3, a, input, p

Стили: style.css, root.css, modal.css, input.css, center.css, calendar.css, btn.css

Скрипты: modal.js, script.js