МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

|  |  |  |
| --- | --- | --- |
| ОТЧЕТ  ЗАЩИЩЕН С ОЦЕНКОЙ  ПРЕПОДАВАТЕЛЬ |  |  |
| ассистент |  | А.Э. Зянчурин |
| должность, уч. степень, звание | подпись, дата | инициалы, фамилия |

КАФЕДРА № 43

ОТЧЕТ О ЛАБОРАТОРНОМ ЗАДАНИИ №3

Применение и конфигурирование общего и системного программного обеспечения. Разработка специального программного обеспечения

по дисциплине: " Основы программной инженерии"

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 4134к Н.А. Костяков

подпись, дата инициалы, фамилия

Санкт-Петербург 2023

## Цель работы

Целью работы является формирование практических навыков разработки специального программного обеспечения с учетом специфики клиент-серверной архитектуры.

## Задание на лабораторную работу

Разработка программного обеспечения в соответствии с требованиями, предъявленными в первой лабораторной работе и поставленными задачами во второй лабораторной работе. Установка, конфигурирование и применение общего и системного программного обеспечения. Выполнение третьей лабораторной работы предполагает коллективное взаимодействие в группе из трех человек в ролях: «Специалист по дизайну графических пользовательских интерфейсов», «Разработчик Web и мультимедийных приложений», «Администратор баз данных».

## Подготовка к защите лабораторной работы

Каждый участник команды из трех человек должен по результатам проведенной работы подготовить устный доклад на 3-5 минут, в рамках которого он должен пояснить, что было сделано, какие возникли сложности, проблемы в реализации этапов и какие были сделаны выводы в процессе их реализации.

В ходе работы я не столкнулся с какими-то проблемами. Самое трудное в моей работе – создать композицию для веб интерфейса. Все остальные моменты не требуют глубоких познаний в верстке. В этом деле решает в основном практика.

Вот пример кода для взаимодействия БД и админ панели:

function list\_all\_users(){

    let data = fetch('admin/get\_users').then(response => response.json())

    .then(data => {

      console.log(data);

      const container = document.getElementById("user\_list");

      for (let i =0; i< data.length; i++){

        let newDiv = document.createElement("tr");

        newDiv.setAttribute('onclick', `fill\_lineedit\_of\_users(${data[i]["user\_id"]})`);

        for (const [key, value] of Object.entries(data[i])) {

            let obj = document.createElement("td");

            obj.innerHTML = value;

            newDiv.appendChild(obj)

          }

        container.appendChild(newDiv);

        console.log(data[i]);

    }

    });

}

**Специалист по дизайну графических пользовательских интерфейсов должен быть готов ответить на следующие вопросы.**

1. Какие элементы и атрибуты являются основными для HTML?

2. Какова структура CSS и как CSS применяется при верстке web-страниц?

3. Как следует использовать JavaScript для управления DOM и обработкой событий?

**1. Основные элементы и атрибуты HTML:**

- Элементы: Основные элементы включают в себя заголовки (<h1>, <h2>, ..., <h6>), параграфы (<p>), списки (<ul>, <ol>, <li>), ссылки (<a>), изображения (<img>), формы (<form>, <input>, <button>), таблицы (<table>, <tr>, <td>), и другие. HTML имеет много элементов для различных задач.

- Атрибуты: Каждый HTML-элемент может иметь атрибуты, которые предоставляют дополнительную информацию о элементе. Например, атрибут src используется для указания источника изображения в теге <img>, а атрибут href - для указания URL в теге <a>.

**2. Структура CSS и применение:**

- Структура: CSS (Cascading Style Sheets) представляет собой язык стилей, который определяет внешний вид элементов на веб-странице. Он работает с концепцией правил стилей, которые состоят из селекторов и объявлений стилей. Пример:

css

selector {

property: value;

}

- Применение: CSS может быть применен к HTML-документу разными способами. Это может быть встроенный (внутри тега <style> внутри <head>), внутренний (с использованием атрибута style в HTML-теге) или внешний (внешний файл стилей, подключаемый через <link> внутри <head>).

**3. Использование JavaScript для управления DOM и обработки событий:**

- Управление DOM: JavaScript позволяет изменять содержимое и структуру веб-страницы, взаимодействуя с DOM (Document Object Model). DOM представляет дерево объектов, представляющих структуру HTML-документа. Примеры операций с DOM: изменение текста элемента, добавление или удаление элементов.

- Обработка событий: JavaScript также позволяет реагировать на события, такие как щелчок мыши, нажатие клавиши и другие. Это достигается с использованием обработчиков событий. Пример:

javascript

document.getElementById('myButton').addEventListener('click', function() {

// код, выполняемый при клике на кнопку

});

- в работе для связи с беком использовалась функция fetch(); Таким образом клиентская часть получала json данные и отображала их согласно скриптам js