**Moldova State University**

**Faculty of Mathematics and Informatics**

**Department of Informatics**

Лабораторная работа №2  
По предмету : “Java Script / TypeScript”

Учащийся: Rusanov Nichita

Группа: IAfr2302

Дата: 25.05.2024

**Задание:**

**Структура проекта**

1. Создайте файл index.html с основной структурой веб-страницы.
2. Создайте файл index.css для определения стилей страницы.
3. Создайте директорию /src, где будут размещены файлы JavaScript.
4. В директории /src создайте файл index.js (или main.js) с основным кодом JavaScript.
5. В директории /src создайте файл activity.js, в котором будет содержаться логика для получения данных со сторонних ресурсов.

**Создание HTML-страницы**

1. Создайте HTML-страницу со следующим содержанием.

**Получение данных с сервера**

1. Напишите функцию getRandomActivity(), которая будет делать запрос и получать данные со стороннего ресурса: <http://www.boredapi.com/api/activity/>.
2. Отобразите полученную активность (текст активности) на странице index.html  
   1. *Какое значение возвращает функция fetch?*
   2. *Что представляет собой Promise?*
   3. *Какие методы доступны у объекта Promise?*
3. Добавьте обработку ошибок в функцию getRandomActivity(). В случае ошибки добавьте следующий текст в файл index.html: "К сожалению, произошла ошибка".
4. Измените функцию getRandomActivity() так, чтобы она использовала ключевые слова async / await. *Каковы основные различия между использованием async / await и Promise?*
5. Добавьте функционал обновления данных каждую минуту. Используйте функцию setTimeout().
6. **Дополнительное задание**: измените функцию getRandomActivity() так, чтобы она возвращала данные, и добавьте функцию updateActivity(), которая будет отображать полученные данные.

**Описание цели и основные этапы работы**

Основной целью данной индивидуальной работы было использование полученных на паре теоретических данных. В данной работе были затронуты темы асинхронного JavaScript, модулей и обработки ошибок. Более того, данная работа предполагает взаимодействие с внешними API. Среди основных этапов можно выделить три этапа. Первый – создание структуры проекта, включающей HTML, CSS и JavaScript файлы. Второй – написание функции для получения данных со стороннего ресурса. Третий – реализация обработки ошибок и автоматического обновления данных.

**Практическая часть**

В данной работе были реализованы многие аспекты языка JavaScript. Среди них всех я бы хотел выделить работу с асинхронным кодом, использование модулей и обработку ошибок. Одно из самых уместных было использование fetch() для выполнения HTTP-запросов и применение async/await для более удобного написания асинхронного кода. Также был реализован функционал автоматического обновления данных с использованием setTimeout().

**Вывод**

В ходе данной работы я столкнулся с некоторыми проблемами, что раскрыло некоторые недоученные мной темы, но я успешно их решил и подтянул свои знания JavaScript. Данная работа была не столь сложной и произвела на меня исключительно положительные впечатления. Все выполненные задания были интересными и, что самое главное, закрепили теоретические знания, полученные на парах.

**Ссылка на github:**<https://github.com/n1kry/js-lab2>

**Ответы на контрольные вопросы:**

 **Какое значение возвращает функция fetch?**

Функция fetch возвращает объект Promise, который разрешается объектом Response.

 **Что представляет собой Promise?**

Promise представляет собой объект, который используется для обработки асинхронных операций. Он может находиться в одном из трех состояний: ожидание (pending), выполнен (fulfilled), отклонен (rejected).

 **Какие методы доступны у объекта Promise?**

У объекта Promise доступны методы then(), catch(), и finally().

 **Основные различия между использованием async/await и Promise?**

async/await предоставляет более удобный и синхронно выглядящий способ работы с асинхронным кодом, в то время как Promise требует использования цепочек then и catch.

**Использованные источники:**

1. <https://chat.openai.com> – исключительно для разбора ошибок и искоренения их причин
2. <https://www.w3schools.com/>
3. <https://stackoverflow.com/>
4. <https://www.freecodecamp.org/>