МІНІСТЕРСТОВ ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

**Звіт**

про виконання лабораторної роботи №3

З курсу “**Веб програмування на стороні сервера**” **“**Практика програмування на JavaScript. Використання вбудованих та сторонніх модулів**”**

Виконав:

студент групи ФЕІ-25

МАЗУРИК Богуслав

Перевірив:

ЧМИХАЛО Олександр

Львів-2025

**Мета:**

* Використати навички програмування на JavaScript та роботи з JSON
* Опанувати використання вбудованих модулів Node.js
* Навчитися створювати package.json для проекту
* Навчитися встановлювати сторонні пакети npm

**Хід роботи:**

1. Ознайомився з методичкою до лабораторної роботи.
2. Створив новий репозиторій на гітхабі та файли package.json, main.js, .gitignore у папці лабораторної роботи. Також додав до цієї папки файл weather.json та встановив модуль commander для зручної обробки аргументів командного рядка.
3. Після підключення локального репозиторію до гітхабу та коміту проекту написав код який обробляє аргументи командного рядка, перевіряє коректність їх введення, витягує потрібну інформацію з файлу weather.json та формує відповідь на основі згаданих раніше аргументів.
4. Протестував програму та написав звіт.

**Висновок:**

В цілому, лабораторною роботою та своїм виконанням я задоволений. Вона не є складною і досить інтуїтивно зрозуміла.

**Код та приклад роботи:**

Зображення, що містить текст, знімок екрана, Мультимедійне програмне забезпечення

Вміст на основі ШІ може бути неправильним.

Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Вміст на основі ШІ може бути неправильним.

import fs from 'fs';

import { program } from 'commander';

program.option('-i, --input <path>', 'шлях до вхідного json файлу');

program.option('-o, --output [path]', 'шлях для збереження результату');

program.option('-d, --display', 'виведення результату в консоль');

program.option('-h, --humidity', 'вивести вологість');

program.option('-r, --rainfall <number>', 'фільтрувати за кількістю опадів');

program.parse(process.argv);

const options = program.opts();

if (!options.input) {

  console.error("Please, specify input file");

  process.exit(1);

}

let data;

try {

  const fileContent = fs.readFileSync(options.input, 'utf8');

  data = JSON.parse(fileContent);

} catch (error) {

  console.error("Cannot find input file");

  process.exit(1);

}

let processedData = data;

if (options.rainfall) {

  const rainLevel = parseFloat(options.rainfall);

  processedData = processedData.filter(item => {

    return item.Rainfall > rainLevel;

  });

}

const resultLines = processedData.map(item => {

  let line = `rainfall: ${item.Rainfall}, pressure: ${item.Pressure3pm}`;

  if (options.humidity) {

    line += `, humadity: ${item.Humidity3pm}`;

  }

  return line;

})

const outputString = resultLines.join('\n');

if (options.display) {

  console.log(outputString);

}

if (options.output) {

  try {

    fs.writeFileSync(options.output, outputString);

    console.log('Результат успішно записано в файл:', options.output);

  } catch (error) {

    console.error('Помилка при збереженні файлу:', error);

  }

}