**ЛАБОРАТОРНА РОБОТА № 1**

**Робота з файлами, JSON, об’єктами, модулями, npm, lodash, nodemon, yargs**

Хід роботи

Завдання 1: В виконавчому файлі sandbox/task01.js виведіть на консоль “Hello, World!”

Лістинг коду:

console.log("Hello World");

Результат виконання програми:



Рис.1. Результат виконання програми

Завдання 2: Створити додаток, що приєднує до файлу рядок

Лістинг коду:

const fs = require('fs');

const newText = 'Hello World';

if (!fs.existsSync('output.txt')) {

fs.writeFileSync('output.txt', '');

}

fs.appendFileSync('output.txt', `${newText}\n`);

Результат виконання програми:

A screenshot of a computer

Description automatically generated

Рис.2. Результат виконання програми

Завдання 3: Вивести інформацію про користувача операційної системи

Лістинг коду:

const os = require('os');

const userSystemName = os.userInfo().username;

console.log(`Hello, ${userSystemName}`);

Результат виконання програми:



Рис.3. Результат виконання роботи

Завдання 4: Використання модуля lodash

Лістинг коду:

const \_ = require('lodash');

const capitalizedString = \_.capitalize('hello world');

const lowerCaseString = \_.toLower('Hello World');

const upperCaseString = \_.toUpper('Hello World');

const upperFirstString = \_.upperFirst('hello world');

const lowerFirstString = \_.lowerFirst('Hello World');

console.log("Capitalized string: ", capitalizedString, "\nLower case string: ", lowerCaseString, "\nUpper case string: ", upperCaseString, "\nUnderlined string: ", upperFirstString, "\nLower first string: ", lowerFirstString);

Результат виконання коду:

A screen shot of a computer

Description automatically generated

Рис.4. Результат виконання програми

Завдання 5: Створіть додаток для додавання, перегляду, видалення мов користувача

Лістинг коду:

app.js:

const yargs = require('yargs');

const user = require('./user.js');

yargs.command({

command: 'add',

describe: 'Додати нову мову',

builder: {

title: {

describe: 'Назва мови',

demandOption: true,

type: 'string'

},

level: {

describe: 'Рівень володіння мовою',

demandOption: true,

type: 'string'

}

},

handler(argv) {

user.addLanguage(argv.title, argv.level);

}

})

.command({

command: 'remove',

describe: 'Видалити мову',

builder: {

title: {

describe: 'Назва мови',

demandOption: true,

type: 'string'

}

},

handler(argv) {

user.removeLanguage(argv.title);

}

})

.command({

command: 'list',

describe: 'Переглянути всі мови',

handler() {

user.listLanguages();

}

})

.command({

command: 'read',

describe: 'Переглянути деталі мови',

builder: {

title: {

describe: 'Назва мови',

demandOption: true,

type: 'string'

}

},

handler(argv) {

user.readLanguage(argv.title);

}

})

.help()

.argv;

user.js:

const fs = require('fs');

const loadUser = () => {

try {

const dataBuffer = fs.readFileSync('user.json');

const dataJSON = dataBuffer.toString();

return JSON.parse(dataJSON);

} catch (e) {

return {

"FirstName": "Ivan",

"LastName": "Potiienko",

"Languages": []

};

}

};

const saveUser = (user) => {

const dataJSON = JSON.stringify(user);

fs.writeFileSync('user.json', dataJSON);

};

const addLanguage = (title, level) => {

const user = loadUser();

const duplicateLanguage = user.Languages.find((language) => language.title === title);

if (!duplicateLanguage) {

user.Languages.push({

title: title,

level: level

});

saveUser(user);

console.log('Мову додано');

} else {

console.log('Мова вже існує');

}

};

const removeLanguage = (title) => {

const user = loadUser();

const languagesToKeep = user.Languages.filter((language) => language.title !== title);

if (user.Languages.length > languagesToKeep.length) {

console.log('Мову видалено');

user.Languages = languagesToKeep;

saveUser(user);

} else {

console.log('Мова не знайдена');

}

};

const listLanguages = () => {

const user = loadUser();

console.log('Ваші мови:');

user.Languages.forEach((language) => {

console.log(`${language.title}: ${language.level}`);

});

};

const readLanguage = (title) => {

const user = loadUser();

const language = user.Languages.find((language) => language.title === title);

if (language) {

console.log(`${language.title}: ${language.level}`);

} else {

console.log('Мова не знайдена');

}

};

module.exports = {

addLanguage: addLanguage,

removeLanguage: removeLanguage,

listLanguages: listLanguages,

readLanguage: readLanguage

};

Результат виконання програми:

A screen shot of a computer program

Description automatically generated

Рис.5. До виконання



A screenshot of a computer program

Description automatically generated

Рис.6. Результат виконання програми