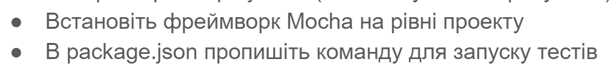
**ЛАБОРАТОРНА РОБОТА № 8**

**Розробка тестів з використанням фреймворка Mocha**

**Хід роботи:**

**Завдання 1:**



**Лістинг програми:**

**Package.json**

{  
 "name": "project04",  
 "version": "1.0.0",  
 "description": "",  
 "main": "index.js",  
 "scripts": {  
 "test": "mocha \*.test.js"  
 },  
 "keywords": [],  
 "author": "",  
 "license": "ISC",  
 "devDependencies": {  
 "mocha": "^8.4.0"  
 },  
 "dependencies": {  
 "mongoose": "^5.12.11",  
 "validator": "^13.6.0"  
 }  
}

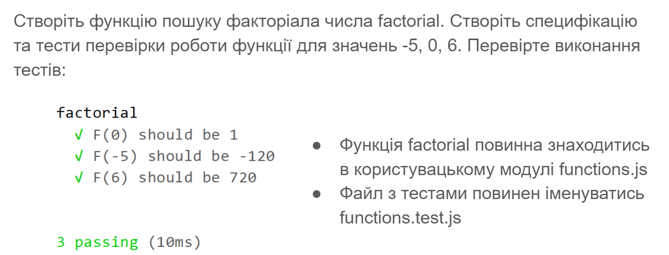
**Виконання завдання:**

Результат виконання завдання:



Мал. 1. Установка фреймворку Mocha

**Завдання 1:**



**Виконання завдання:**

**Лістинг програми:**

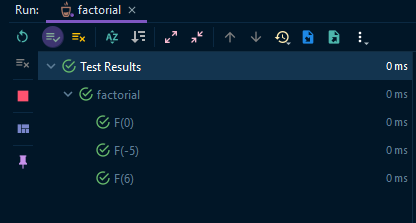
**functions.js**

*const* factorial = (n) => {  
 *if* (n === 0)  
 *return* 1;  
 *else if* (n > 0)  
 *return* (n !== 1) ? n \* factorial(n - 1) : 1;  
 *else  
 return* (n !== -1) ? n \* factorial(n + 1) : -1;  
}  
  
*const functions* = {factorial}  
  
module.exports = *functions*

**functions.test.js**

*const functions* = require('./modules/functions');  
*const* assert = require('assert');  
  
*describe*('factorial', *function* () {  
 *it*('F(0)', *function* () {  
 assert.equal(*functions*.factorial(0), 1)  
 })  
 *it*('F(-5)', *function* () {  
 assert.equal(*functions*.factorial(-5), -120)  
 })  
 *it*('F(6)', *function* () {  
 assert.equal(*functions*.factorial(6), 720)  
 })  
})

**Результат виконання програми:**

  
Мал. 2. Результат тестування функції factorial

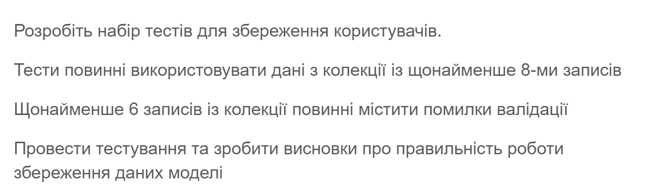
**Виконання завдання:**

**Лістинг програми:**

**user.js**

*const mongoose* = require('mongoose');  
*const validator* = require('validator');  
  
*mongoose*.*connect*(`mongodb://127.0.0.1:27017/mocha`, {  
 useNewUrlParser: *true*,  
 useCreateIndex: *true*,  
 useUnifiedTopology: *true*})  
  
*let* userSchema = *new mongoose*.Schema({  
 name: {  
 type: *String*,  
 required: *true*,  
 trim: *true*,  
 minlength: 2  
 },  
 email: {  
 type: *String*,  
 required: *true*,  
 lowercase: *true*,  
 unique: *true*,  
 validate(value) {  
 *if*(!*validator*.isEmail(value))  
 *throw new Error*("Тобою был выбран неправильный email");  
 }  
 },  
 age: {  
 type: *Number*,  
 required: *true*,  
 *default*: 0,  
 validate(value) {  
 *if* (value <= 0)  
 *throw new Error*('Возраст должен быть > 0')  
 }  
 }  
})  
  
*const User* = *mongoose*.*model*("User", userSchema);  
  
module.exports = *User*

**Завдання 2:**



**Виконання завдання:**

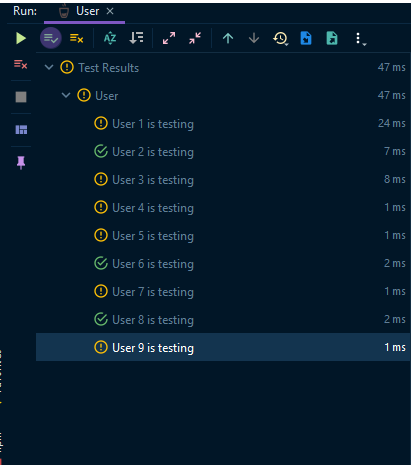
**Лістинг програми:**

**function.test.js**

*const User* = require('./models/user');

*let* users = [  
 {name: 'Ivan', age: 'sd', email: 'asdasdasdasdaag@gmail.com'}, *// неправильний вік(рядком)* {name: 'Max', age: 8, email: 'asdasdasdh@gmail.com'},  
 {name: 'Ethan', age: 23, email: 'asdasdasdh@gmail.com'}, *// дубльвонай email* {name: 'asdasdasdsd', age: -15, email: 'sd@ukr.net'}, *// неправильний вік(<=0)* {name: "Name"}, *// не вказаний вік та email* {name: 'Kevin', age: 43, email: 'kevin@gmail.com'},  
 {name: 'David', age: 12, email: 'asdasdasd'}, *// неправильний email* {name: 'Chloe', age: 8, email: 'asdasdasdasdaag@gmail.com'},  
 {name: 'M', age: 15, email: 'm@gmail.com'}, *// коротке ім'я*]  
  
*describe*('User', *function* () {  
 *for* (*let* i = 0; i < users.length; i++) {  
 *it*(`User ${i + 1} is testing`, *function* (done) {  
 *let* user = *new User*({name: users[i].name, email: users[i].email, age:users[i].age});  
 user.save(done);  
 })  
 }  
})

**Результат роботи програми:**



Мал. 3. Результат тестування

***Висновки:*** в ході виконання лабораторної роботи було отримано базові навички для роботи з Mocha, повторено роботу з mongoose