

Лабораторна робота №7

Розробка тестів з використанням фреймворка Mocha

Мета: навчитись розробляти тести з використанням фреймворка Mocha

Установка Mocha

- Створіть проект mocha (або наступний за рахунком)
- Встановіть фреймворк Mocha на рівні проекту
- В package.json пропишіть команду для запуску тестів

```
package.json
{
  "dependencies": {
    "yargs-parser": "^20.2.4",
    "yargs-unparser": "^2.0.0",
    "yocto-queue": "^0.1.0"
  },
  "scripts": {
    "test": "mocha tests/*.test.js"
  },
  "author": ""
}
```

```
PS C:\Users\PC\4 semester\node.js\Lab6_ChatApp> cd "C:\Users\PC\4 semester\node.js\Lab7_Mocha"
PS C:\Users\PC\4 semester\node.js\Lab7_Mocha> npm install --save-dev mocha

added 72 packages in 3s

20 packages are looking for funding
  run `npm fund` for details
PS C:\Users\PC\4 semester\node.js\Lab7_Mocha> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (lab7_mocha)
version: (1.0.0)
```

Рис. 1. Результат

					ДУ «Житомирська політехніка».24.121.13.000 - Лр7			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Клосович І.А.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Сидорчук В.О.					1	11
Керівник							ФІКТ Гр. ІІЗ-22-1[1]	
Н. контр.								
Зав. каф.								

Завдання 1.

Створіть функцію пошуку факторіала числа factorial. Створіть специфікацію та тести перевірки роботи функції. Перевірте виконання тестів:

n	factorial(n)
5	120
6	720
0	1
-5	null
-6	null

- Функція factorial повинна знаходитись в користувацькому модулі functions.js
- Файл з тестами повинен іменуватись functions.test.js

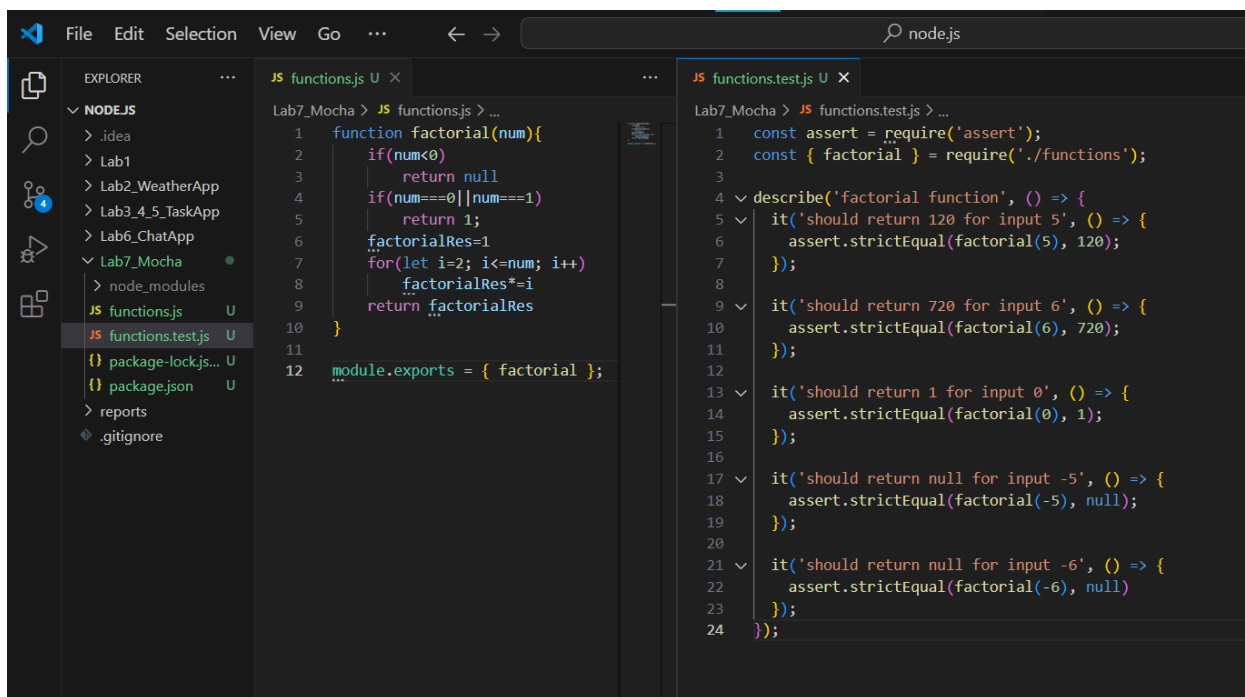


Рис. 2. Результат

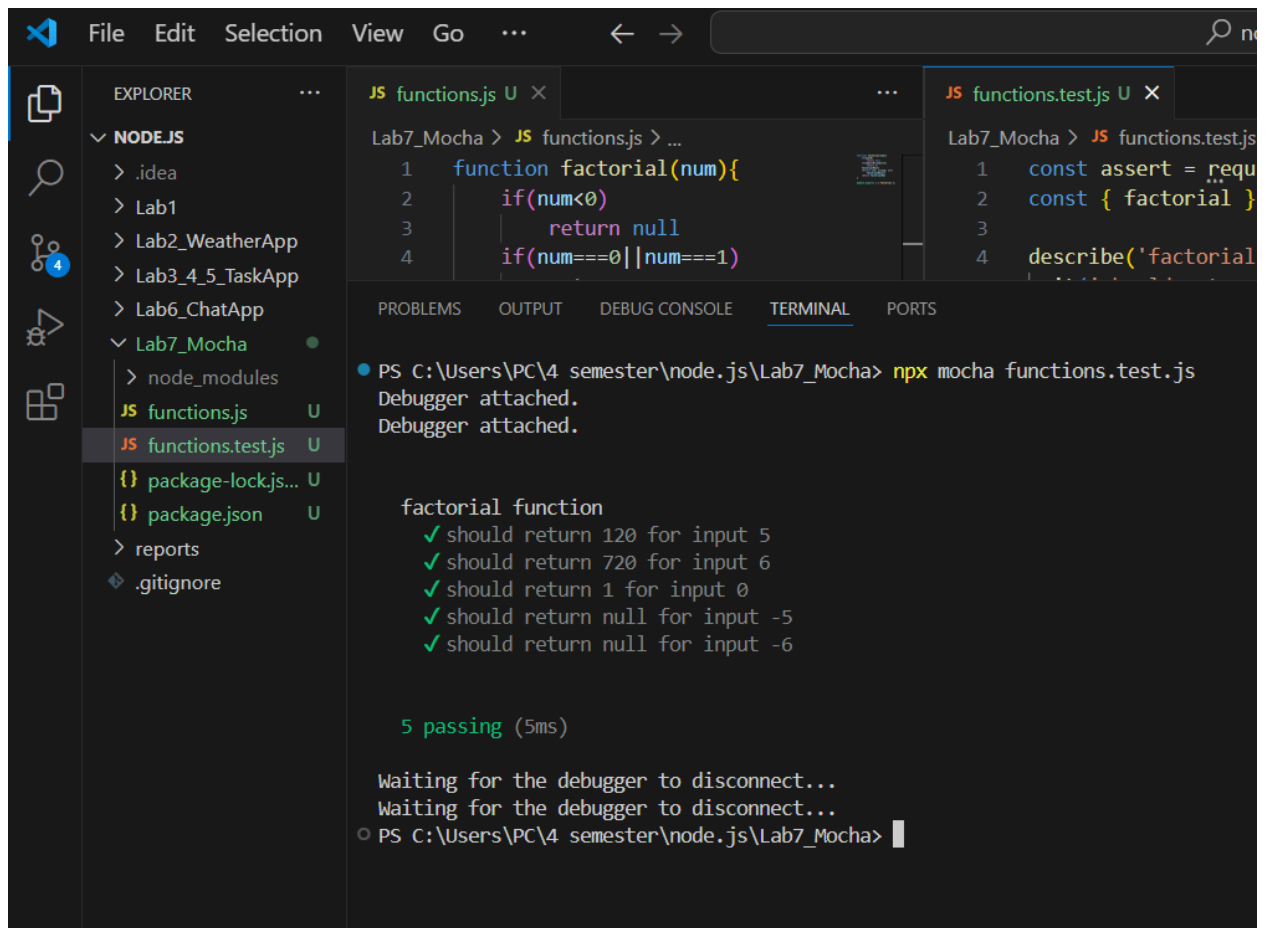


Рис. 3. Результат

Завдання 2. Розробка тестів для REST-застосунку (TaskApp)

Перед виконанням тестування відбувається очистка колекцій *users* і *tasks*;

#	Тест	Результат проходження тесту
1	Реєстрація користувача User1 з помилкою валідації	Статус 401 та назва помилки
2	Реєстрація користувача User1 без помилок	Статус 200. Отриманий об'єкт user з властивістю <i>_id</i>
3	Реєстрація користувача User2 без помилок	Статус 200 і отриманий об'єкт user з ідентифікатором

4	Вхід під User1 з вірними даними	Статус 200 і отримане повідомлення success
5	Додавання задачі Task1	Статус 200 і отриманий об'єкт task з ідентифікатором
6	Додавання задачі Task2	Статус 200 і отриманий об'єкт task з ідентифікатором
7	Отримання задач користувача User1	Статус 200. Довжина 2
8	Отримуємо задачу Task1 по ідентифікатору	Статус 200. Об'єкт task з властивостями title і completed.
9	Вихід	Повідомлення "logout success"
10	Вхід під User2 з вірними даними	Статус 200 і отримане повідомлення success
11	Додавання задачі Task3	Статус 200 і отриманий об'єкт task з ідентифікатором

12	Отримання задач користувача User2	Статус 200. Довжина 1.
13	Отримуємо задачу Task1 по ідентифікатору	Статус 404. Повідомлення "Not Found"
14	Вихід	Статус 200. Повідомлення "logout success"
15	Отримуємо задачу Task1 по її ідентифікатору	Статус 403. Повідомлення "Forbidden Access"

В звіті після запуску тестування всі 15 тестів повинні бути пройдені

В ході даного завдання було частково модернізовано код з 3-5 лабораторних робіт і написано 15 функцій для тестування.

		Клосович І.А.			ДУ «Житомирська політехніка».24.121.13.000 - Лр7	Арк.
		Сидорчук В.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

1 process.env.NODE_ENV = 'test';
2
3 // let mongoose = require("mongoose");
4 require('.././Lab3_4_5_TaskApp/db/mongoose');
5 require('dotenv').config({ path: '.././Lab3_4_5_TaskApp/.env' });
6 let User = require('.././Lab3_4_5_TaskApp/models/user');
7 let Task = require('.././Lab3_4_5_TaskApp/models/task');
8
9 let chai = require('chai');
10 let chaiHttp = require('chai-http');
11 const expect = chai.expect;
12 let server = require('.././Lab3_4_5_TaskApp/src/app');
13 let should = chai.should();
14
15
16 chai.use(chaiHttp);

```

Рис. 4. Деякі підключення

```

17 describe('Users and Tasks', () => {
18
19   let userId;
20   let user2Id;
21   let AuthTokenUser;
22   let user1Task1Id;
23   let user1Task2Id;
24   let user2Task1Id;
25
26   before(async () => {
27     try {
28       await User.deleteMany({});
29       await Task.deleteMany({});
30     } catch (error) {
31       throw new Error(error.message);
32     }
33   });
34
35   it('1. it should not POST a user without data(validation error)', (done) => {
36     let user = {}
37     chai.request(server)
38       .post('/users')
39       .send(user)
40       .end((err, res) => {
41         res.should.have.status(400); //код 400 описує зазвичай помилку валідації
42         res.body.should.be.a('object');
43         res.body.should.have.property('errors');
44         expect(res.body).to.have.property('name').to.equal('ValidationError');
45         done();
46       });
47   });
48 });

```

Рис. 5. Приклад деякого коду функцій

```

JS user.js 232
node_modules 233
src 234
  middleware 235
    JS auth.js M 236
    routers 237
      JS task.js M 238
      JS user.js M 239
      JS app.js M 240
    .env 241
    JS eslint.config.mjs 242
    {} package-lock.json 243
    {} package.json 244
  > Lab6_ChatApp 245

```

```

done();
});
});
it('15. it shouldn\'t GET Task1 for unauthorized user by id', (done) => {
  chai.request(server)
    .get(`/tasks/${user1Task1Id}`)
    // .set('Authorization', 'Bearer ' + AuthTokenUser)
    .end((err, res) => {
      res.should.have.status(403); // хоча 401 - неавтентифікований теж могло б підійти
      res.text.should.equal('Forbidden Access');
      done();
    });
});
});

```

Рис. 6. Приклад коду

```

Lab7_Mocha > tests > JS TaskApp.test.js > describe('Users and Tasks') callback
1R describe('Users and Tasks'. () => {
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
  PS C:\Users\PC\4 semester\node.js> cd "C:\Users\PC\4 semester\node.js\Lab7_Mocha"
  PS C:\Users\PC\4 semester\node.js\Lab7_Mocha> npx mocha tests/taskapp.test.js
  Server is listening on 4000

  Users and Tasks
  Connected to database
  ✓ 1. it should not POST a user without data(validation error) (39ms)
  ✓ 2. should register User1 without errors
  ✓ 3. should register User2 without errors
  ✓ 4. should login User1 with correct credentials
  Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfcmFpbGUiOiI2NjM3YTQ3YjczOGYxMDdkYzVhMTJmN2IiLCJpYXQiOiJlZ3MTQ5MjI1MT19.
  Decoded: [object Object]
  User: {
    _id: new ObjectId('6637a47b738f107dcea12f7b'),
    name: 'Iryna',
    password: '$2b$08$MgyhqgCTG08cG/LxsaDk/ePt7kKAFTTOY4NzvqeBsrK.RrUp/YCd2',
    age: 19,
    email: 'iryna@gmail.com',
    tokens: [
      {
        token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfcmFpbGUiOiI2NjM3YTQ3YjczOGYxMDdkYzVhMTJmN2IiLCJpYXQiOiJlZ3MTQ5MjI1MT19.
        _id: new ObjectId('6637a47b738f107dcea12f80'),
        id: '6637a47b738f107dcea12f80'
      }
    ],
    __v: 1,
    id: '6637a47b738f107dcea12f7b'
  }
  ✓ 5. should add Task1 for User1 without errors
  Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfcmFpbGUiOiI2NjM3YTQ3YjczOGYxMDdkYzVhMTJmN2IiLCJpYXQiOiJlZ3MTQ5MjI1MT19.
  Decoded: [object Object]
  User: {
    _id: new ObjectId('6637a47b738f107dcea12f7b'),
    name: 'Iryna',
    password: '$2b$08$MgyhqgCTG08cG/LxsaDk/ePt7kKAFTTOY4NzvqeBsrK.RrUp/YCd2',

```

Рис. 7. Результат

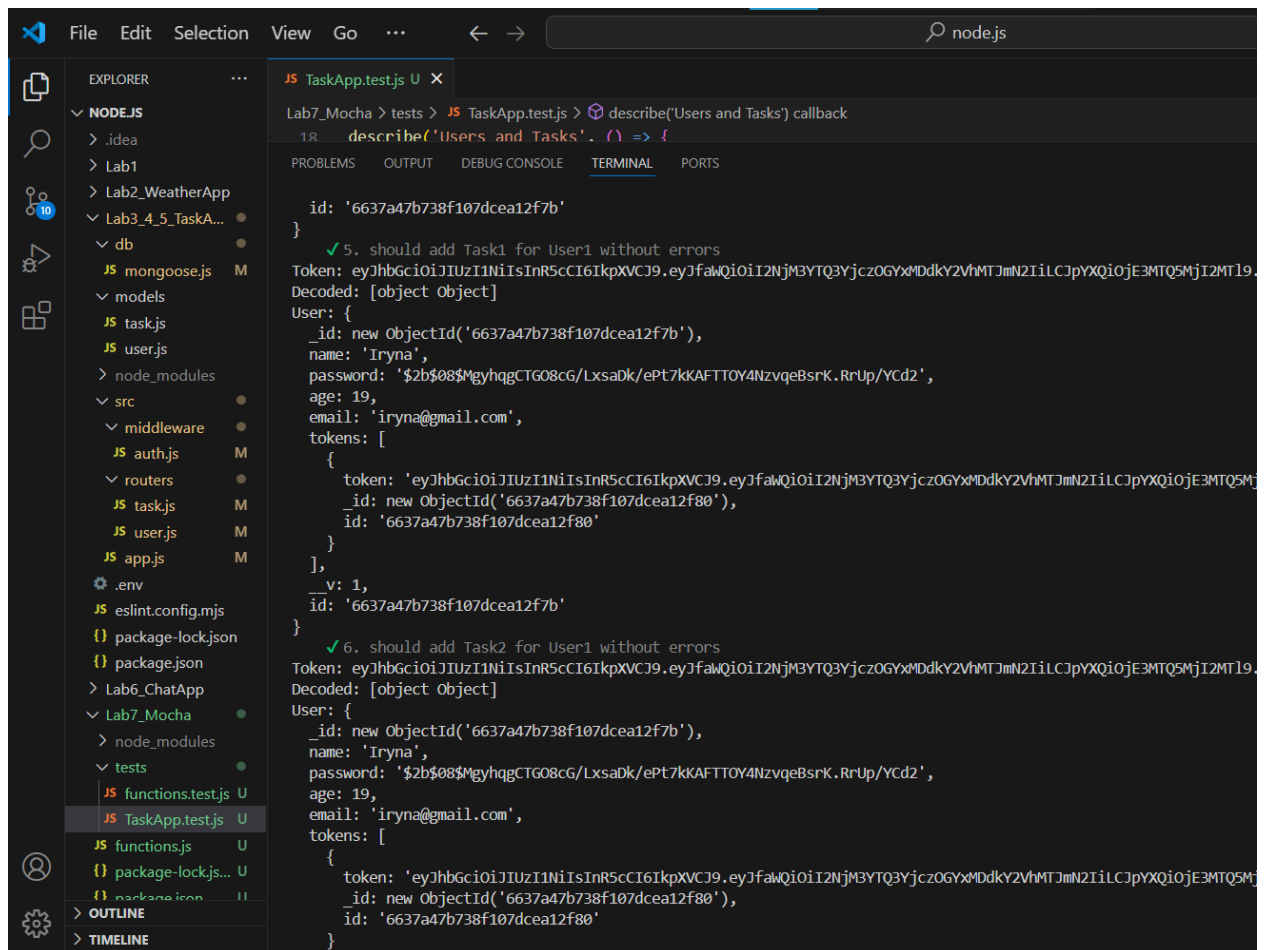


Рис. 8. Результат

		Клосович І.А.			ДУ «Житомирська політехніка».24.121.13.000 - Лр7	Арк.
		Сидорчук В.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

VS Code interface showing a Node.js application. The Explorer sidebar on the left displays the project structure, including files like `db`, `models`, `tasks`, `users`, and `app.js`. The main editor area shows a REST client request to `describe('Users and Tasks')` with a JSON body. The response shows a user object with fields like `id`, `name`, `password`, `age`, `email`, and `tokens`.

Рис. 9. Результат

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying a project structure for 'Lab7_Mocha'. The main editor window shows the file 'TaskApp.test.js' with the following content:

```

Lab7_Mocha > tests > JS TaskApp.test.js > describe('Users and Tasks') callback
18 describe('Users and Tasks'. () => {
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

  __v: 1,
  id: '6637a47b738f107dcea12f7b'
}
  ✓ 9. should logout User1 successfully
  ✓ 10. should login User2 with correct credentials
Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfawQioiI2NjM3YTQ3YjczOGYxMDdkY2VhMTJmN2QilCJp
Decoded: [object Object]
User: {
  _id: new ObjectId('6637a47b738f107dcea12f7d'),
  name: 'Taras',
  password: '$2b$08$E.5PK2P/2HFLjSDtjvw1auMJf8aTydD2tRFecQmRSr0ZanPieB6lq',
  age: 19,
  email: 'taras@gmail.com',
  tokens: [
    {
      token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfawQioiI2NjM3YTQ3YjczOGYxMDdkY2VhMTJmN2QilCJp
      _id: new ObjectId('6637a47b738f107dcea12f96'),
      id: '6637a47b738f107dcea12f96'
    }
  ],
  __v: 1,
  id: '6637a47b738f107dcea12f7d'
}
  ✓ 11. should add Task3 for User2 without errors
Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfawQioiI2NjM3YTQ3YjczOGYxMDdkY2VhMTJmN2QilCJp
Decoded: [object Object]
User: {
  _id: new ObjectId('6637a47b738f107dcea12f7d'),
  name: 'Taras',
  password: '$2b$08$E.5PK2P/2HFLjSDtjvw1auMJf8aTydD2tRFecQmRSr0ZanPieB6lq',
  age: 19,
  email: 'taras@gmail.com',
  tokens: [
    {
      token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfawQioiI2NjM3YTQ3YjczOGYxMDdkY2VhMTJmN2QilCJp
      _id: new ObjectId('6637a47b738f107dcea12f96'),

```

Рис. 10. Результат

Рис. 11. Результат

```

User: {
  _id: new ObjectId('6637a47b738f107dcea12f7d'),
  name: 'Taras',
  password: '$2b$08$E.5PK2P/2HFLjSDtjvwlauM3f8aTydD2tRFecQmRSr0ZanPieB6lq',
  age: 19,
  email: 'taras@gmail.com',
  tokens: [
    {
      token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfawQioiI2NjM3YTQ3Yjc',
      _id: new ObjectId('6637a47b738f107dcea12f96'),
      id: '6637a47b738f107dcea12f96'
    }
  ],
  __v: 1,
  id: '6637a47b738f107dcea12f7d'
}

✓ 14. should logout User2 successfully
✓ 15. it shouldn't GET Task1 for unauthorization user by id

15 passing (253ms)

```

Рис. 12. Результат

Висновок: на лабораторному занятті ми навчились розробляти тести з використанням фреймворка Mocha