

Державний вищий навчальний заклад  
Ужгородський національний університет  
Факультет інформаційних технологій  
Кафедра програмного забезпечення систем

## **ЛАБОРАТОРНА РОБОТА №8**

**Тема:** Юніт (компонент) тести.

Виконав:  
студент 3 курсу  
спеціальності 121 «Інженерія  
програмного забезпечення 3»  
Лях Іван Іванович

**Ужгород 2024**

**Мета роботи:** Створити юніт (компонент) тести для тестування проекту.

**Завдання до роботи:**

1. Створити компонент тести для тестування компонентів проекту.

### **Хід роботи**

1. Створення компонент тестів:

Для написання та виконання тестів було обрано бібліотеку cypress, яка надає гнучкі рішення для component та end-2-end тестування проекту. Для того щоб дізнатися для яких компонентів було створено тестування, та який відсоток тестів зачепило весь компонент, було використано бібліотеку code coverage, яка створює звіт про покриття тестами коду проекту. Таким чином ми завжди будемо знати, які частини коду залишилися без тестів.

Перед початком роботи, потрібно налаштувати відповідні бібліотеки для написання тестів, та створення звітності. Для налаштування cypress потрібно виконати команду “cypress run”, яка запускає графічний додаток бібліотеки cypress. Після першого запуску програми, відбувається налаштування цієї бібліотеки: створюються відповідні директорії з файлами-конфігураторами в них.

Основним файлом конфігурації cypress є файл cypress.config.js, в якому відбувається явне налаштування тестування програми. Повертаючись до code coverage, в цьому файлі імпортується відповідна бібліотека та додаються налаштування для збору інформації про тести.

У випадку використання ES6, налаштування для code coverage у файлі конфігурації cypress, мають такий вигляд:

```
setupNodeEvents(on, config) {  
  codeCoverageTask(on, config);  
  return config;  
}
```

Для роботи з jsx файлами та реактом потрібно створити відповідні налаштування:

```
export default {
  presets: ['@babel/preset-env', '@babel/preset-react'],
  plugins: ['istanbul', '@babel/plugin-syntax-jsx'],
};
```

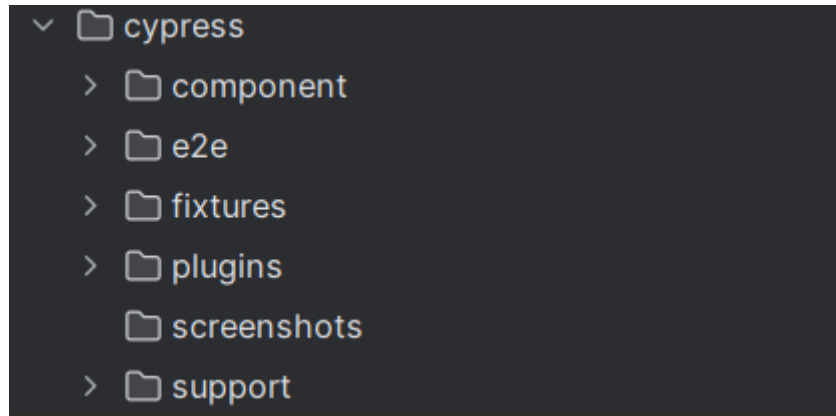


Рис. 1. Вигляд директорії cypress після всіх налаштувань.

Можна побачити, що директорія містить декілька інших директорій, які містять компонент та end-2-end тести у відповідних їм папках.

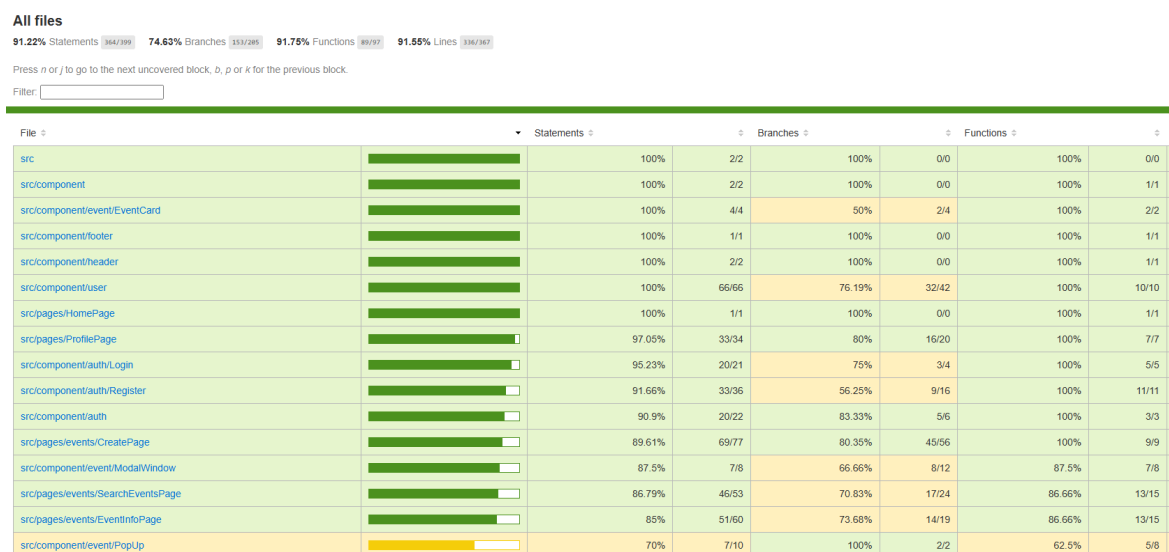


Рис. 2. Звіт про покриття коду.

Після створення тестів та їх виконання створюється папка coverage, в якій наявний html файл, після запуску якого, у браузері відкривається файл зі звітністю про покриття тестами коду проекту.

Як можна побачити на фото, було охоплено всі можливі файли для їх тестування.

**Висновок:** на даній лабораторній роботі було створено компонент-тести проекту. Для написання тестів було використано бібліотеку `supress`, а для збору інформації та створення звіту про покриття тестами коду відповідну бібліотеку `“code coverage”`.