**Лабораторна робота № 9**

**Тема:** Основи роботи із бібліотекою React.js. Testing

**Мета:** Навчитись працювати з бібліотекою React.js. Testing.

**Завдання**

1. Створити тести до розробленого проекту на попередній практичній роботі.

**Хід роботи**

App.js

import PhoneBook from "./PhoneBook";

function App() {  
 return (  
 <div className="App">  
 <PhoneBook/>  
 </div>  
 );  
}  
  
export default App;

PhoneBook.js

import React, {useCallback, useState} from 'react';

import './PhoneBook.css';

function PhoneBook() {

    let [contacts, setContacts] = useState([]);

    const submitForm = useCallback((e) => {

        e.preventDefault();

        const surnameInput = document.getElementById('surname');

        const nameInput = document.getElementById('name');

        const middle\_nameInput = document.getElementById('middle\_name');

        const telephone\_numberInput = document.getElementById('telephone\_number');

        const cityInput = document.getElementById('city');

        setContacts([...contacts, {

            "surname": surnameInput.value,

            "name": nameInput.value,

            "middle\_name": middle\_nameInput.value,

            "telephone\_number": telephone\_numberInput.value,

            "city": cityInput.value,

        }])

    }, [contacts, setContacts]);

    return (

        <div className="Student flex justify-content\_space-evenly align-items\_flex-start m\_2rem">

            <form role="form" className="form flex flex-direction\_column" onSubmit={submitForm}>

                <label htmlFor="surname">Surname</label>

                <input id="surname" type="text" required/>

                <label htmlFor="name">Name</label>

                <input id="name" type="text" required/>

                <label htmlFor="middle\_name">Middle name</label>

                <input id="middle\_name" type="text" required/>

                <label htmlFor="telephone\_number">Telephone number</label>

                <input id="telephone\_number" type="tel" required/>

                <label htmlFor="city">City</label>

                <select id="city">

                    <option value="Khmelnytskyi">Khmelnytskyi</option>

                    <option value="Vinnitsa">Vinnitsa</option>

                    <option value="Shepetivka">Shepetivka</option>

                    <option value="Dunaivtsi">Dunaivtsi</option>

                </select>

                <button type="submit">Submit</button>

            </form>

            <Contacts contacts={contacts}/>

        </div>

    );

}

function Contacts(props) {

    return (

        <div className="Contacts">

            {

                props.contacts.length > 0 ?

                    <table role="table">

                        <thead>

                        <tr>

                            <th>Surname</th>

                            <th>Name</th>

                            <th>Middle name</th>

                            <th>Telephone number</th>

                            <th>City</th>

                        </tr>

                        </thead>

                        <tbody>

                        {

                            props.contacts.map((s, index) =>

                                <tr key={index}>

                                    <td>{s.surname}</td>

                                    <td>{s.name}</td>

                                    <td>{s.middle\_name}</td>

                                    <td>{s.telephone\_number}</td>

                                    <td>{s.city}</td>

                                </tr>

                            )

                        }

                        </tbody>

                    </table> : ''}

        </div>

    );

}

export default PhoneBook;

PhoneBook.test.js

import userEvent from "@testing-library/user-event";

import { render, screen, fireEvent } from '@testing-library/react';

import App from './App';

import PhoneBook from './PhoneBook';

import renderer from 'react-test-renderer';

test('document should contain form', () => {

    render(<PhoneBook />);

    const form = screen.getByRole('form');

    expect(form).toBeInTheDocument();

});

test("should submit", () => {

    render(<PhoneBook/>);

    const form = screen.getByRole('form');

    const nameInput = screen.getByLabelText("Name");

    userEvent.type(nameInput, "Liza");

    fireEvent.submit(form);

    const table = screen.getByRole('table');

    console.log(table.getElementsByTagName('td')[table.getElementsByTagName('td').length-4].textContent);

    expect(table.getElementsByTagName('td')[table.getElementsByTagName('td').length-4].textContent).toBe('Liza');

});

test("should submit and add to form", () => {

    render(<PhoneBook />);

    const form = screen.getByRole('form');

    const nameInput = screen.getByLabelText("Name");

    userEvent.type(nameInput, "Liza");

    fireEvent.submit(form);

    const table = screen.getByRole('table');

    const oldLength = table.getElementsByTagName('td').length / 5;

    userEvent.type(nameInput, "Ania");

    fireEvent.submit(form);

    const newLength = table.getElementsByTagName('td').length / 5;

    expect(newLength - oldLength).toBe(1);

});

test("should match snapshot", () => {

    const phoneBook = renderer

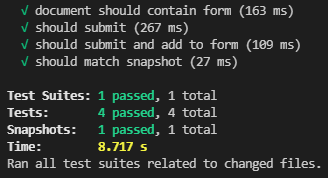
        .create(<PhoneBook />)

        .toJSON();

    expect(phoneBook).toMatchSnapshot();

});

Результат:



**Висновок:** Навчилась працювати із бібліотекою React.js. Testing.