**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗВІТ**

*з лабораторної роботи № 4*

**«Редагування та додавання компонентiв React»**

**Варіант № 28**

Дисципліна «Web development»

Спеціальність «Інженерія програмного забезпечення»

121–ЛР.4–309.22110728

***Cтудент*** *\_\_\_\_\_\_\_\_\_\_****К. О. Школьний***

*(підпис)*

*\_\_21.10.2023 \_\_*

*(дата)*

***Викладач*** *\_\_\_\_\_\_\_\_\_\_****Є. Д. Стоєв***

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв – 2023**

**Лабораторна робота №4**

***Тема роботи:***Редагування та додавання компонентiв React

***Мета роботи:***Створення веб-додатка з різними компонентами React.

***Завдання:***

* Перетворіть будь-який класовий компонент на функціональний
* Створіть 2 власні хуки, наприклад для одержання статусу та для виведення даних зі стану в консоль
* \*Використайте useEffect
* \*Продемонструйте роботу ефекту з залежністю

***Виконання завдання:***

<https://github.com/DreadLord025/WebDev-4>

***Висновки:***

В даннiй роботi було досягнуто мети створення веб-додатка з різними компонентами React. Існуючий класовий компонент був успішно перетворений на функціональний, і використано власні хуки для керування станом та ефектами, у тому числі "useEffect" з залежністю.

**Контрольнi питання:**

1. Hook - функція в React, яка дозволяє використовувати стан та інші функціональності React в функціональних компонентах.

2. Базові Hook’s:

- «useState» - використовується для створення та оновлення стану компонента.

- «useEffect» - дозволяє виконувати певний код після рендерингу компонента.

- «useContext» - дозволяє отримувати доступ до контексту React.

- «useReducer» - використовується для керування станом за допомогою функції редуктора.

3. Додаткові Hook’s:

- «useRef» - для зберігання посилань на DOM-елементи та інші значення, які залишаються незмінними між рендерами.

- «useMemo» - для мемоізації обчислення дорогих обчислень та виразів.

- «useCallback» - повертає мемоізовану версію callback-функції.

- «useLayoutEffect» - подібний до «useEffect», але виконується синхронно після всіх змін перед наступним рендером.

4. Власний хук можна створити, створивши функцію, яка починається з "use", і використовуючи інші хуки в її тілі. Наприклад, «useCustomHook» використовує інші хуки та повертає значення для використання в компонентах.

5. Мемоізація - це оптимізація, яка полягає в зберіганні попередніх результатів обчислень, щоб уникнути повторних обчислень у разі, коли вхідні дані не змінилися.

6. «useCallback» повертає мемоізовану версію функції, яка не змінюється між рендерами, якщо її залежності не змінилися. Звичайна callback функція створюється при кожному рендері, навіть якщо її залежності не змінилися, що може призвести до надмірного перерендеру компонентів.