

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ОСТРОЗЬКА АКАДЕМІЯ”
Кафедра економіко-математичного моделювання та інформаційних
технологій

ЗВІТ
Лабораторна робота №8
Підключення сторінки користувачів
до бекену використовуючи axios

здобувачки вищої освіти
першого (бакалаврського) рівня
третього року навчання групи
КН-31
спеціальності 122 Комп’ютерні
науки
ОПП «Комп’ютерні науки»

Лайтер Ярини Семенівни
(прізвище, ім’я, по батькові)

База лабораторної - кафедра економіко-математичного моделювання
та інформаційних технологій

Керівник лабораторної від університету

(науковий ступінь, учене звання керівника)

(Власне ім’я та ПРІЗВИЩЕ)

Члени комісії:

Зміст

Зміст	2
Опис	3
Виконання	4
1. Додаємо новий сервіс:	4
2. Бекенд:	4
3. Видозмінюємо компонент:	5
1. UI/UX:	6
Висновок:	8

Опис

Тема: Підключення сторінки користувачів до бекенду використовуючи axios

Мета: Ознайомити студентів зі підключенням сторінок до бекенду з використанням axios

Завдання

1. Підготуйте свій React проект. Якщо у вас ще немає проекту, створіть його за допомогою Create React App або іншого інструменту на ваш вибір.
2. Використайте компонент `UserList`, який відповідає за відображення списку користувачів.
3. Використовуючи Axios, зробіть запит до бекенду для отримання списку користувачів. Бекенд може бути симульованим (наприклад, використовуючи JSON-сервер) або реальним, якщо ви маєте доступ до бекенду.
4. Після успішного отримання даних, відобразіть список користувачів на сторінці, використовуючи компонент `UserList`.
5. Додайте можливість пагінації або фільтрації користувачів (за ім'ям, електронною поштою тощо)
6. Зробіть додаток візуально привабливим та зручним для користувача, використовуючи CSS та інші стилізаційні засоби.
7. Перевірте, чи правильно обробляється помилковий стан (наприклад, якщо запит до бекенду не вдається виконати).
8. Використовуючи Axios, додайте можливість додавання ,видалення та редагування користувачів через веб-інтерфейс.

Виконання

1. Додаємо новий сервіс:

Використовуємо наш попередній проект з 7 лабораторної роботи.

Туди додаємо наш новий сервіс

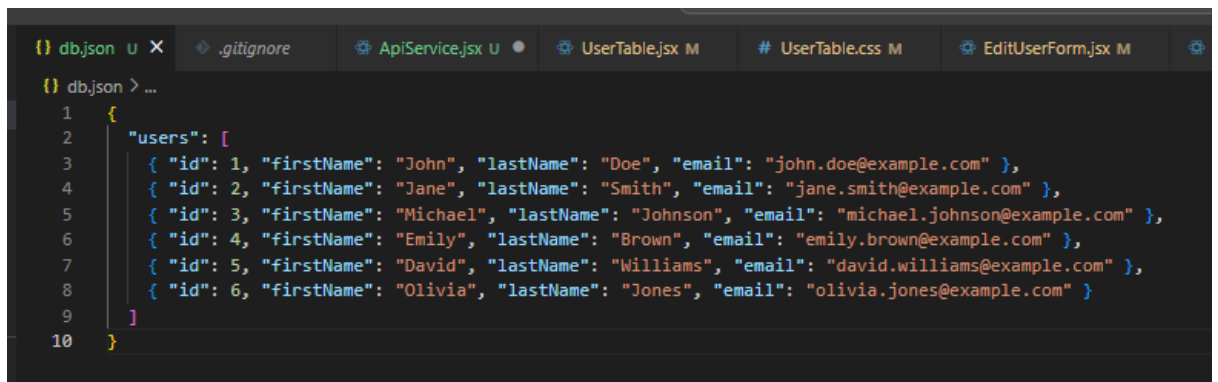


```
src > services > ApiService.jsx > [default]
1  import axios from 'axios';
2
3  class ApiService {
4    constructor(baseUrl) {
5      this.api = axios.create({
6        baseUrl,
7      });
8    }
9
10   getUsers() {
11     return this.api.get('/users');
12   }
13
14   getUserById(userId) {
15     return this.api.get(`/users/${userId}`);
16   }
17
18   addUser(newUser) {
19     return this.api.post('/users', newUser)
20   }
21
22   editUser(userId, updatedUser) {
23     return this.api.put(`/users/${userId}`, updatedUser);
24   }
25
26   deleteUser(userId) {
27     return this.api.delete(`/users/${userId}`);
28   }
29 }
30
31 export default ApiService;
```

Рис 1.1 Компонент ApiService.jsx

2. Бекенд:

Створюємо наш стимульований бекенд (використовуючи json-сервер):

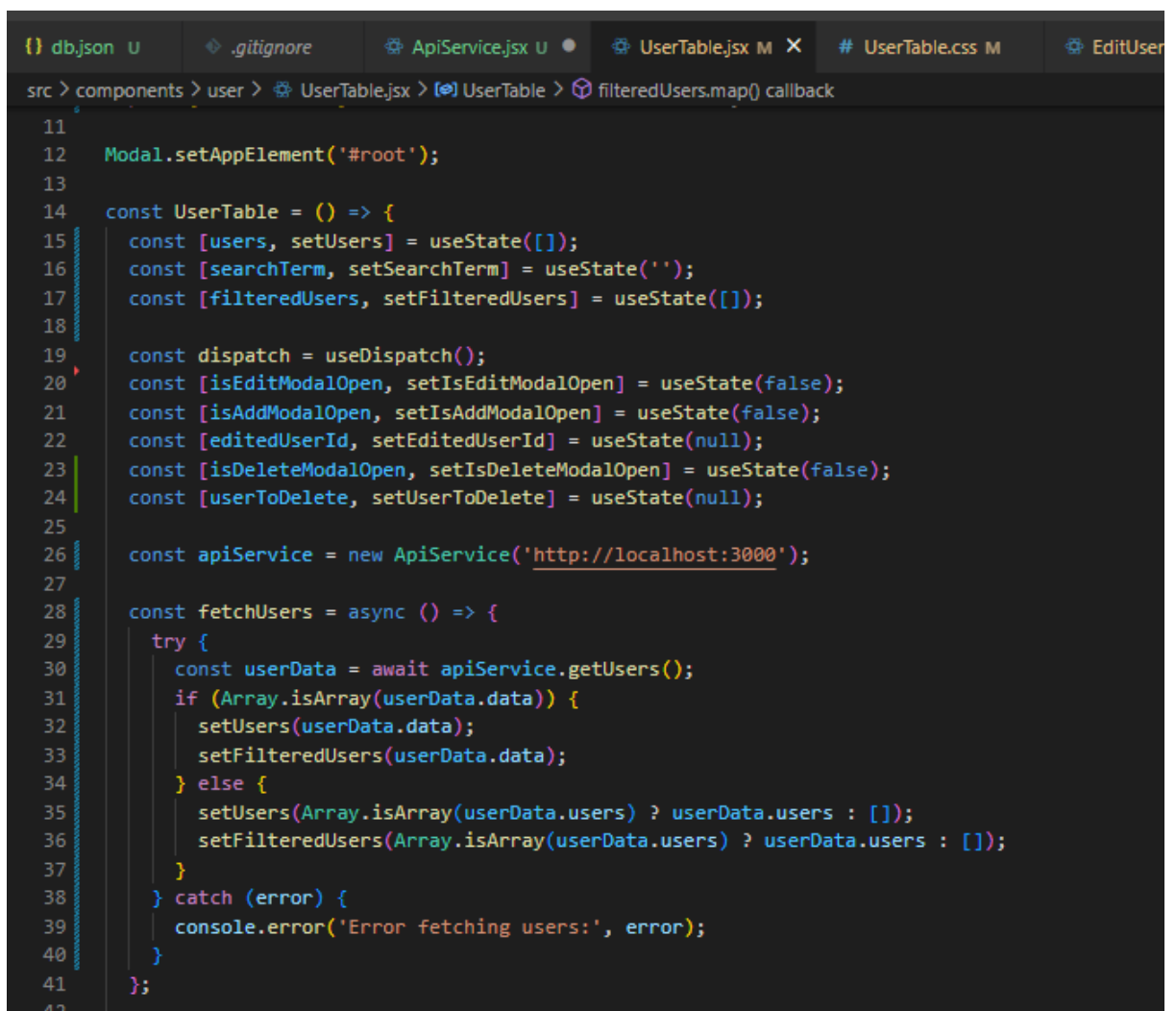


```
{
  "users": [
    { "id": 1, "firstName": "John", "lastName": "Doe", "email": "john.doe@example.com" },
    { "id": 2, "firstName": "Jane", "lastName": "Smith", "email": "jane.smith@example.com" },
    { "id": 3, "firstName": "Michael", "lastName": "Johnson", "email": "michael.johnson@example.com" },
    { "id": 4, "firstName": "Emily", "lastName": "Brown", "email": "emily.brown@example.com" },
    { "id": 5, "firstName": "David", "lastName": "Williams", "email": "david.williams@example.com" },
    { "id": 6, "firstName": "Olivia", "lastName": "Jones", "email": "olivia.jones@example.com" }
  ]
}
```

Рис 1. 2 Бекенд додатку

3. Видозмінюємо компонент:

Далі підключаємо наш новостворений компонент та вказуємо посилання на бекенд у нашому компоненті для виведення списку користувачів:



```
src > components > user > UserTable.jsx > UserTable > filteredUsers.map() callback
11
12 Modal.setAppElement('#root');
13
14 const UserTable = () => {
15   const [users, setUsers] = useState([]);
16   const [searchTerm, setSearchTerm] = useState('');
17   const [filteredUsers, setFilteredUsers] = useState([]);
18
19   const dispatch = useDispatch();
20   const [isEditModalOpen, setIsEditModalOpen] = useState(false);
21   const [isAddModalOpen, setIsAddModalOpen] = useState(false);
22   const [editedUserId, setEditedUserId] = useState(null);
23   const [isDeleteModalOpen, setIsDeleteModalOpen] = useState(false);
24   const [userToDelete, setUserToDelete] = useState(null);
25
26   const apiService = new ApiService('http://localhost:3000');
27
28   const fetchUsers = async () => {
29     try {
30       const userData = await apiService.getUsers();
31       if (Array.isArray(userData.data)) {
32         setUsers(userData.data);
33         setFilteredUsers(userData.data);
34       } else {
35         setUsers(Array.isArray(userData.users) ? userData.users : []);
36         setFilteredUsers(Array.isArray(userData.users) ? userData.users : []);
37       }
38     } catch (error) {
39       console.error('Error fetching users:', error);
40     }
41   };
42 }
```

Рис 1. 4 Код сторінки UserTable.jsx,

Також додали валідацію та фільтрацію до юзерів

1. UI/UX:

В результаті отримуємо:

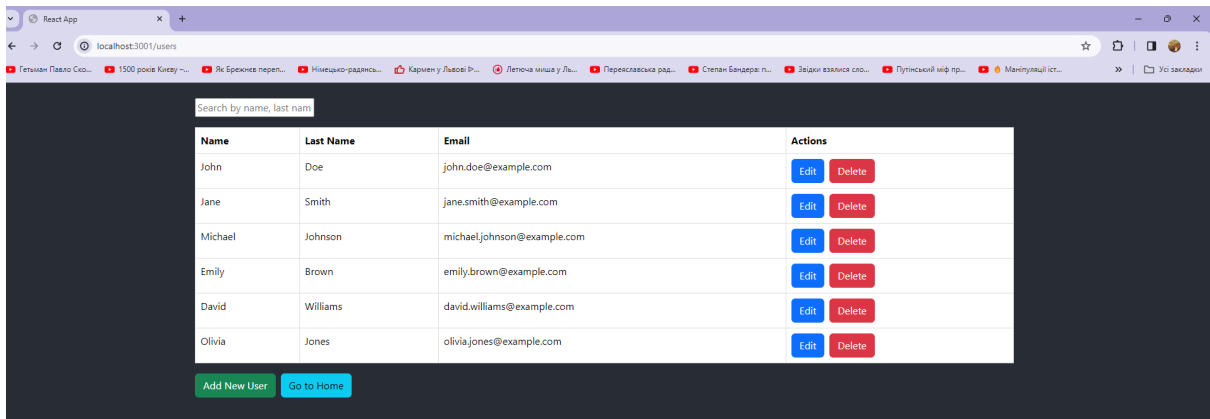


Рис 6.1 Результат запуску програми

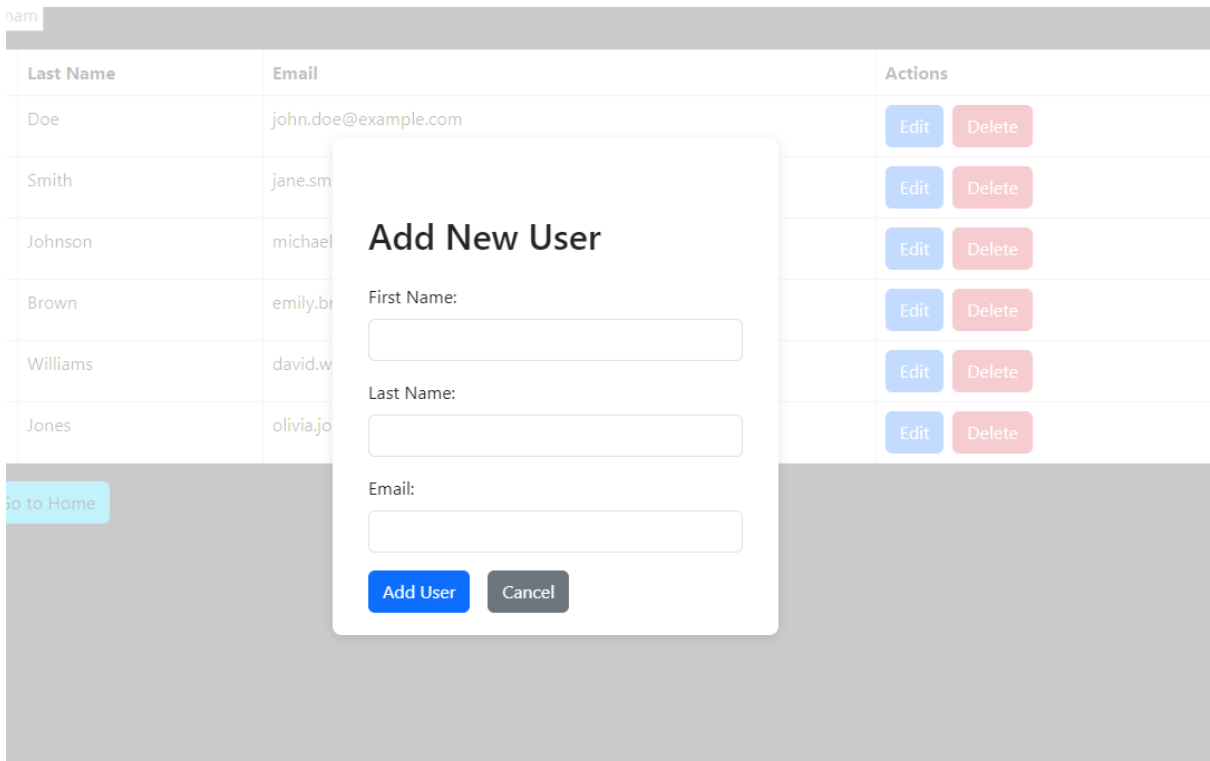


Рис 6.2 Результат запуску програми

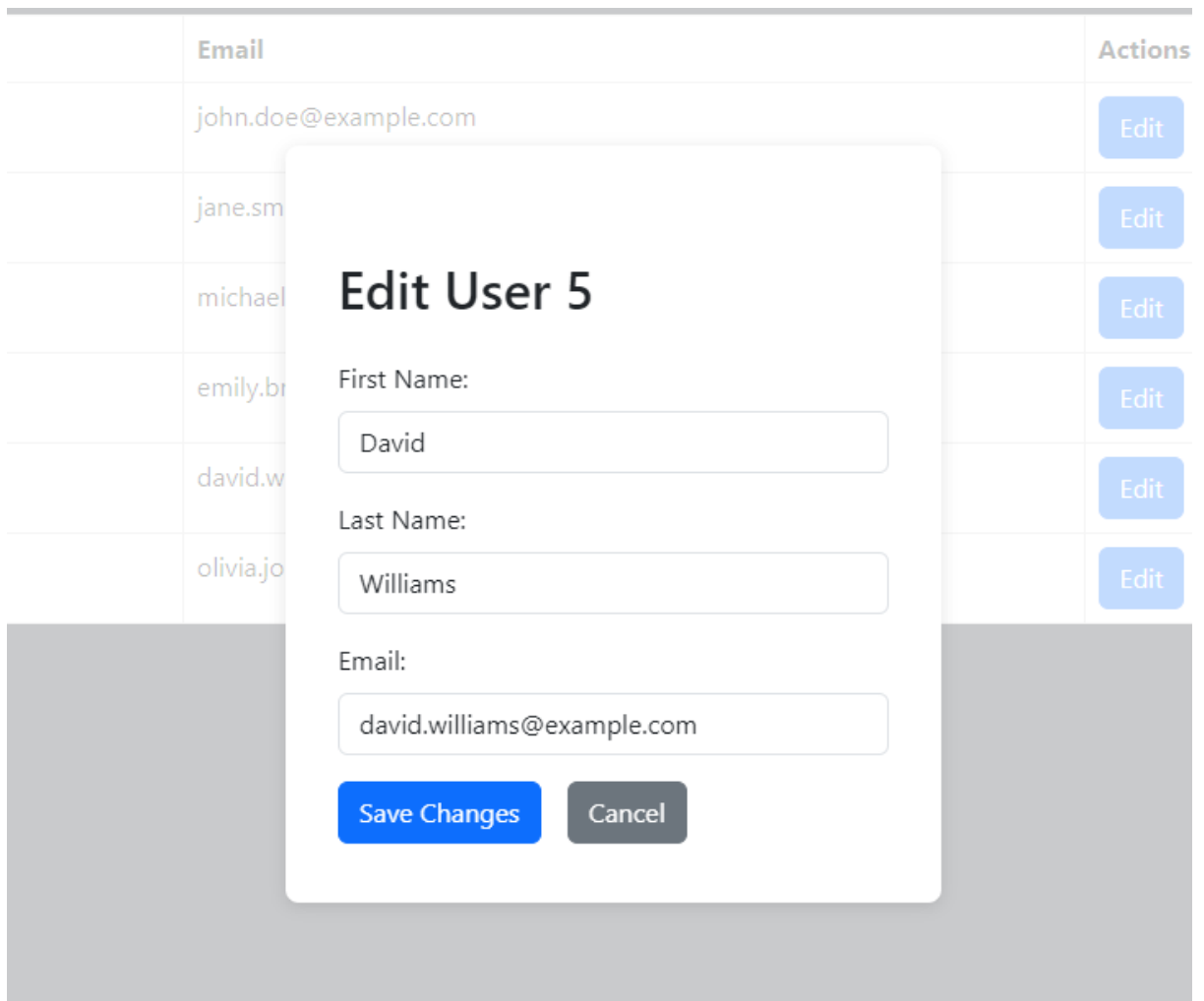


Рис 6.3 Результат запуску програми

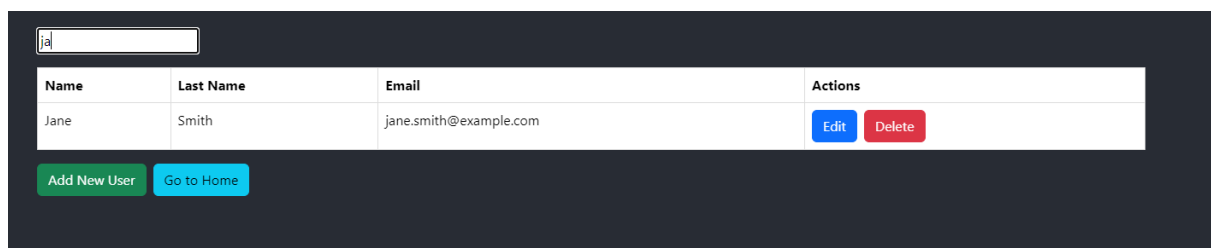


Рис 6.4 Результат запуску програми

Висновок:

На цій лабораторній роботі ми створили стимульований бекенд для нашого додатку та підключили його для сторінки користувачів.

Реалізували додавання, видалення та редагування юзерів за допомогою арі, а також додали валідацію та фільтрацію.

Зробили додаток візуально привабливим та перевірили на правильність обробки помилковий стан.