Code :-

App.tsx :-

```tsx
import { Route, Routes, NavLink } from 'react-router-dom';
import UsersPage from './pages/UsersPage';

export default function App() {
  return (
    <div className="container">
      <nav className="navbar">
        <h1>User Admin</h1>
        <NavLink to="/" end>Users</NavLink>
      </nav>
      <Routes>
        <Route path="/" element={<UsersPage />} />
      </Routes>
    </div>
  );
}
```
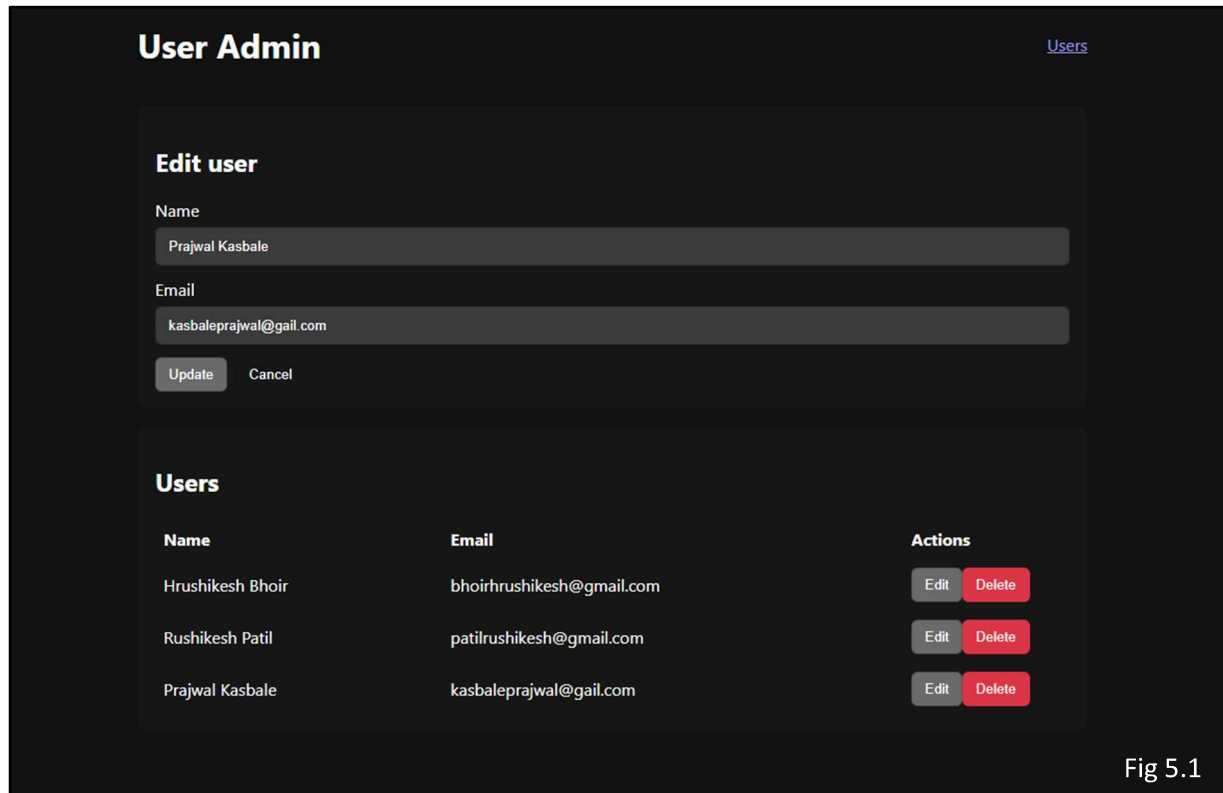
Style.css :-

```css
:root { color-scheme: light dark; }
* { box-sizing: border-box; }
body { margin: 0; font-family: system-ui, Arial, sans-serif; }
.container { max-width: 960px; margin: 0 auto; padding: 24px; }
.navbar { display: flex; align-items: center; justify-content: space-between; margin-bottom: 16px; }
.page { display: grid; gap: 16px; }
.card { border: 1px solid rgba(0,0,0,.15); border-radius: 8px; padding: 16px; background: rgba(255,255,255,.02); }
.form { display: grid; gap: 12px; }
label { display: grid; gap: 6px; }
input { padding: 10px 12px; border: 1px solid rgba(0,0,0,.2); border-radius: 6px; }
.actions { display: flex; gap: 8px; }
button { padding: 8px 12px; border-radius: 6px; border: 1px solid rgba(0,0,0,.2); cursor: pointer; }
button.secondary { background: transparent; }
button.danger { background: #dc3545; color: #fff; border-color: #dc3545; }
.table { width: 100%; border-collapse: collapse; }
.table th, .table td { padding: 8px; border-top: 1px solid rgba(0,0,0,.1); text-align: left; }
.error { color: #dc3545; }
```
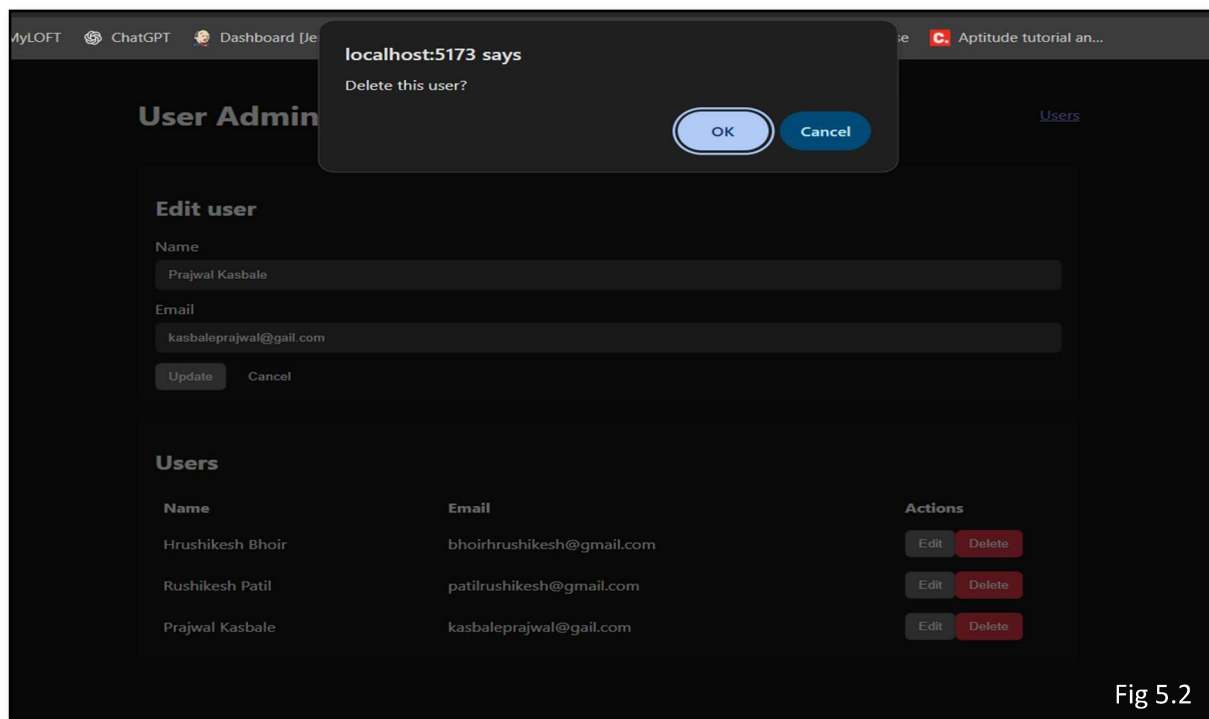
Pages/UserPage.tsx :-

```tsx
1   import { useEffect, useMemo, useState } from 'react';
2   import { createUser, deleteUser, fetchUsers, updateUser, type User } from '../api/client';
3
4   type FormState = { name: string; email: string };
5
6   export default function UsersPage() {
7     const [users, setUsers] = useState<User[]>([]);
8     const [loading, setLoading] = useState(true);
9     const [error, setError] = useState<string | null>(null);
10    const [editingId, setEditingId] = useState<string | null>(null);
11    const [form, setForm] = useState<FormState>({ name: '', email: '' });
12
13    async function load() {
14      setLoading(true);
15      setError(null);
16      try {
17        const data = await fetchUsers();
18        setUsers(data);
19      } catch (e: any) {
20        setError(e?.message ?? 'Failed to load users');
21      } finally {
22        setLoading(false);
23      }
24    }
25
26    useEffect(() => {
27      load();
28    }, []);
29
30    function onEdit(user: User) {
31      setEditingId(user._id ?? null);
32      setForm({ name: user.name, email: user.email });
33    }
34
35    function onCancel() {
36      setEditingId(null);
37      setForm({ name: '', email: '' });
38    }
39
40    async function onSubmit(e: React.FormEvent) {
41      e.preventDefault();
42      try {
43        if (editingId) {
44          const updated = await updateUser(editingId, form);
45          setUsers(prev => prev.map(u => (u._id === editingId ? updated : u)));
46        } else {
```

Output :-

Fig 5.1

Delete User :-

Fig 5.2

Add User :-

Fig 5.3