Name: Abdurrahman Qureshi

Roll No: 242466

Practical No: 9

Date Of Performance: 25/09/2025

Aim: Implementation of React useState, useEffect, useContext

THEORY:

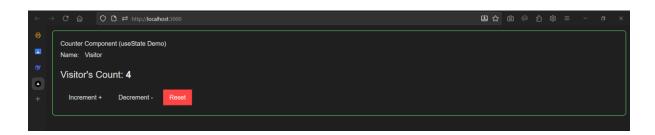
Q) Create functional components to demonstrate useState

CODE (Quote):

```
"use client";
import React, { useEffect, useState } from 'react';
const Counter: React.FC = () => {
    const [count, setCount] = useState<number>(0);
    const [name, setName] = useState<string>('Visitor');
    useEffect(() => {
        console.log(`Counter mounted or updated: ${name}'s count is ${count}`);
        return () => {
            console.log('Cleanup for counter effect');
    }, [count, name]);
    const increment = () => setCount(prev => prev + 1);
    const decrement = () => setCount(prev => prev - 1);
    const reset = () => setCount(0);
    return (
        <div style={{ padding: '20px', border: '2px solid #4CAF50', margin: '10px',</pre>
borderRadius: '8px' }}>
            <h2>Counter Component (useState Demo)</h2>
            <div style={{ marginBottom: '15px' }}>
                <label>
```

```
Name:
                    <input
                        type="text"
                         value={name}
                         onChange={(e) => setName(e.target.value)}
                         style={{ marginLeft: '10px', padding: '5px' }}
                    />
                </label>
            </div>
            <div style={{ fontSize: '24px', marginBottom: '15px' }}>
                {name}'s Count: <strong>{count}</strong>
            </div>
            <div>
                <button onClick={increment} style={{ margin: '5px', padding: '8px</pre>
16px' }}>
                    Increment +
                </button>
                <button onClick={decrement} style={{ margin: '5px', padding: '8px</pre>
16px' }}>
                     Decrement -
                </button>
                <button onClick={reset} style={{ margin: '5px', padding: '8px</pre>
16px', backgroundColor: '#ff4444', color: 'white' }}>
                </button>
            </div>
        </div>
    );
};
export default Counter;
CODE (Rendering):
import Counter from "./components/Counter";
export default function Home() {
  return (
      <Counter />
    </>
  );
}
```

OUTPUT:





Q) Create functional components to demonstrate useContext

CODE (Gang):

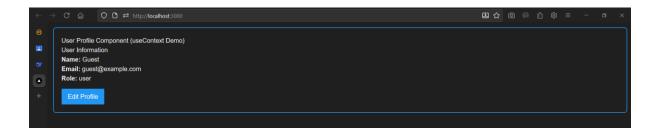
```
"use client";
import React, { createContext, useContext, useState } from 'react';
interface User {
    name: string;
    email: string;
    role: string;
}
const defaultUser: User = {
    name: 'Guest',
    email: 'guest@example.com',
    role: 'user'
};
const UserContext = createContext<User>(defaultUser);
const UserProfile: React.FC = () => {
    const [user, setUser] = useState<User>(defaultUser);
    const [isEditing, setIsEditing] = useState<boolean>(false);
    const [tempUser, setTempUser] = useState<User>(user);
    const handleSave = () => {
        setUser(tempUser);
        setIsEditing(false);
    };
    const handleCancel = () => {
        setTempUser(user);
        setIsEditing(false);
    };
    return (
        <UserContext.Provider value={user}>
            <div style={{ padding: '20px', border: '2px solid #2196F3', margin:</pre>
'10px', borderRadius: '8px' }}>
                <h2>User Profile Component (useContext Demo)</h2>
                {isEditing ? (
```

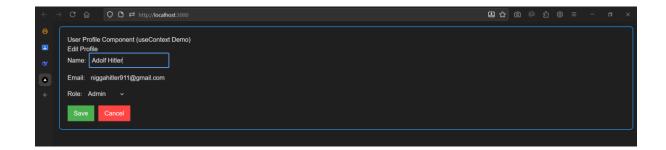
```
<EditForm
                        user={tempUser}
                        onUserChange={setTempUser}
                        onSave={handleSave}
                        onCancel={handleCancel}
                    />
                ):(
                    <ProfileDisplay
                        onEdit={() => setIsEditing(true)}
                    />
                )}
            </div>
        </UserContext.Provider>
    );
};
const ProfileDisplay: React.FC<{ onEdit: () => void }> = ({ onEdit }) => {
    const user = useContext(UserContext);
    return (
        <div>
            <div style={{ marginBottom: '15px' }}>
                <h3>User Information</h3>
                <strong>Name:</strong> {user.name}
                <strong>Email:</strong> {user.email}
                <strong>Role:</strong> {user.role}
            </div>
            <button
                onClick={onEdit}
                style={{ padding: '8px 16px', backgroundColor: '#2196F3', color:
'white' }}
                Edit Profile
            </button>
        </div>
    );
};
const EditForm: React.FC<{</pre>
    user: User;
    onUserChange: (user: User) => void;
    onSave: () => void;
    onCancel: () => void;
}> = ({ user, onUserChange, onSave, onCancel }) => {
    const handleChange = (field: keyof User, value: string) => {
        onUserChange({ ...user, [field]: value });
    };
    return (
        <div>
            <h3>Edit Profile</h3>
            <div style={{ marginBottom: '10px' }}>
                <label>
                    Name:
                    <input
                        type="text"
```

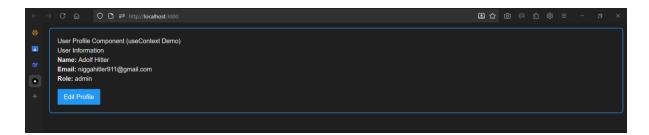
```
value={user.name}
                        onChange={(e) => handleChange('name', e.target.value)}
                        style={{ marginLeft: '10px', padding: '5px', width: '200px'
}}
                    />
                </label>
            </div>
            <div style={{ marginBottom: '10px' }}>
                <label>
                    Email:
                    <input
                        type="email"
                        value={user.email}
                        onChange={(e) => handleChange('email', e.target.value)}
                        style={{ marginLeft: '10px', padding: '5px', width: '200px'
}}
                    />
                </label>
            </div>
            <div style={{ marginBottom: '15px' }}>
                <label>
                    Role:
                    <select
                        value={user.role}
                        onChange={(e) => handleChange('role', e.target.value)}
                        style={{ marginLeft: '10px', padding: '5px' }}
                        <option value="user">User</option>
                        <option value="admin">Admin</option>
                         <option value="moderator">Moderator</option>
                    </select>
                </label>
            <button onClick={onSave} style={{ marginRight: '10px', padding: '8px</pre>
16px', backgroundColor: '#4CAF50', color: 'white' }}>
                Save
            <button onClick={onCancel} style={{ padding: '8px 16px',</pre>
backgroundColor: '#ff4444', color: 'white' }}>
                Cancel
            </button>
        </div>
    );
};
export default UserProfile;
CODE (Rendering):
import UserProfile from "./components/UserProfile ";
export default function Home() {
  return (
      <UserProfile />
```

```
</>
```

OUTPUT:







Q) Create functional components to demonstrate useEffect

CODE (Animals):

```
"use client";
import React, { useEffect, useState } from 'react';
interface Post {
    id: number;
    title: string;
    body: string;
}
```

```
const [posts, setPosts] = useState<Post[]>([]);
    const [loading, setLoading] = useState<boolean>(true);
    const [error, setError] = useState<string | null>(null);
    const [userId, setUserId] = useState<number>(1);
    const [fetchCount, setFetchCount] = useState<number>(0);
    useEffect(() => {
        console.log('Fetching posts for user:', userId);
        const fetchData = async () => {
            try {
                setLoading(true);
                setError(null);
                const response = await fetch(
                    `https://jsonplaceholder.typicode.com/posts?userId=${userId}`
                );
                if (!response.ok) {
                    throw new Error('Failed to fetch posts');
                }
                const data = await response.json();
                setPosts(data.slice(0, 3));
                setFetchCount(prev => prev + 1);
            } catch (err) {
                setError(err instanceof Error ? err.message : 'An error occurred');
            } finally {
                setLoading(false);
            }
        };
        fetchData();
        return () => {
            console.log('Cleanup: cancelling any ongoing requests');
        };
    }, [userId]);
    useEffect(() => {
        document.title = `User ${userId} - ${posts.length} posts`;
        return () => {
            document.title = 'React App';
        };
    }, [userId, posts.length]);
    const changeUser = (newUserId: number) => {
        if (newUserId >= 1 && newUserId <= 10) {
            setUserId(newUserId);
        }
    };
        <div style={{ padding: '20px', border: '2px solid #FF9800', margin: '10px',</pre>
borderRadius: '8px' }}>
            <h2>Data Fetcher Component (useEffect Demo)</h2>
```

```
<div style={{ marginBottom: '15px' }}>
               <label>
                   Select User ID (1-10):
                   <input
                       type="number"
                       value={userId}
                       onChange={(e) => changeUser(parseInt(e.target.value) || 1)}
                       min="1"
                       max="10"
                       style={{ marginLeft: '10px', padding: '5px', width: '60px'
}}
                   />
               </label>
               <span style={{ marginLeft: '15px', color: '#666' }}>
                   Fetched {fetchCount} time(s)
               </span>
            </div>
            {loading && <div style={{ padding: '10px', backgroundColor: '#fff3cd'
}}>Loading posts...</div>}
            {error && (
               <div style={{ padding: '10px', backgroundColor: '#f8d7da', color:</pre>
'#721c24' }}>
                   Error: {error}
               </div>
            )}
            {!loading && !error && (
               <div>
                   <h3>Posts for User {userId}:</h3>
                   {posts.length === 0 ? (
                       No posts found for this user.
                   ):(
                       <div style={{ display: 'grid', gap: '10px' }}>
                           {posts.map(post => (
                               <div key={post.id} style={{
                                   padding: '10px',
                                   border: '1px solid #ddd',
                                   borderRadius: '4px',
                               }}>
                                   <h4 style={{ margin: '0 0 5px 0'
}}>{post.title}</h4>
                                   }}>{post.body}
                               </div>
                           ))}
                       </div>
                   )}
               </div>
           )}
        </div>
    );
};
export default DataFetcher;
```

CODE (Rendering):

OUTPUT:



