1) Write a program to configure routing with authentication with sharing information with and without login?

Program;

App.js ;

```
import { BrowserRouter } from 'react-router-dom';
import './App.css';
import { AuthWrapper } from './AuthWrapper';


function App() {
  return (
    <div className="App">
      <BrowserRouter>
        <AuthWrapper />
      </BrowserRouter>
    </div>
  );
}
```

```
export default App;
```

Home.js ;

```
import React from 'react';

export const Home = () => {

    return (
         <div className="page">
              <h2>Home page</h2>
              <p>This is the text for the home page</p>
         </div>
    )
}
```

AuthWrapper.js ;

```
import { createContext, useContext, useState } from "react"
import { RenderHeader } from "../components/structure/Header";
import { RenderMenu, RenderRoutes } from "../components/structure/RenderNavigation";

const AuthContext = createContext();
export const AuthData = () => useContext(AuthContext);
```

```
export const AuthWrapper = () => {
```

```
    const [ user, setUser ] = useState({name: "", isAuthenticated: false})
```

```
    const login = (userName, password) => {
```

```
         // Make a call to the authentication API to check the username
```

```
          return new Promise((resolve, reject) => {
```

```
               if (password === "password") {
                    setUser({name: userName, isAuthenticated: true})
                    resolve("success")
               } else {
                    reject("Incorrect password")
               }
          })


     }
     const logout = () => {
          setUser({...user, isAuthenticated: false})
     }
     return (

               <AuthContext.Provider value={{user, login, logout}}>
                    <>
                         <RenderHeader />
                         <RenderMenu />
                         <RenderRoutes />
                    </>

               </AuthContext.Provider>

     )
}
```

Login.js ;

```
import { useReducer, useState } from "react";
import { useNavigate } from "react-router-dom";
import { AuthData } from "../../auth/AuthWrapper"

export const Login = () => {
```

```
     const navigate = useNavigate();
     const { login } = AuthData();
     const [ formData, setFormData ] = useReducer((formData, newItem) => { return
( {...formData, ...newItem} )}, {userName: "", password: ""})
     const [ errorMessage, setErrorMessage ] = useState(null)

     const doLogin = async () => {
```

```
          try {

               await login(formData.userName, formData.password)
               navigate("/account")
```

```
          } catch (error) {
```

```
               setErrorMessage(error)

          }
```

```
    }
```

```
    return (
        <div className="page">
            <h2>Login page</h2>
            <div className="inputs">
                <div className="input">
                    <input value={formData.userName} onChange={(e) =>
setFormData({userName: e.target.value}) } type="text"/>
                </div>
                <div className="input">
                    <input value={formData.password} onChange={(e) =>
setFormData({password: e.target.value}) } type="password"/>
                </div>
                <div className="button">
                    <button onClick={doLogin}>Log in</button>
                </div>
                {errorMessage ?
                <div className="error">{errorMessage}</div>
                : null }
            </div>
        </div>
    )
}
```

About.js ;

```
import { useAuth } from '../contexts/AuthContent';
import { useNavigate } from 'react-router-dom'; export const About = () => {

    return (
        <div className="page">
            <h2>About page</h2>
            <p>This is the text for the about page</p>
        </div>
    )
}
```

Private.js;

```
export const Private = () => {

    return (
        <div className="page">
            <h2>Members area</h2>
            <p>This is the private members area</p>
        </div>
    )
}
```

App.css ;

```
.header {
  background-color: #f1f1f1;
  padding: 20px;
}

.header h1 {
  text-align: center;
  margin-bottom: 0;
  font-size: 20px;
}
```

```
.header p {
  text-align: center;
}
```

```
.header .logo {
  text-align: center;
}
```

```
.header .logo img {
  cursor: pointer;
  width: 100px;
}
```

```
.menu {
  background-color: #000;
  text-align: center;
  padding: 15px 0;
}
```

```
.menu .menuItem {
  display: inline-block;
  margin: 0 30px;
}
```

```
.menu .menuItem a {
  color: #fff;
  text-decoration: none;
}
```

```
.page {
  padding: 30px;
}
```

```
.page h2 {
  text-align: center;
}
```

```
.page p {
  text-align: center;
}
```

```
.inputs .input {
  text-align: center;
```

```
}
```

```
.inputs .input input {
  height: 30px;
  width: 300px;
  margin: 5px 0;
}
```

```
.inputs .button {
  text-align: center;
}
```

```
.inputs .button button {
  background-color: #555;
  border: none;
  padding: 15px 30px;
  color: #fff;
  width: 308px;
  margin: 10px 0;
}
```

```
.error {
  background-color: red;
  width: 280px;
  margin: 5px 0;
  margin: 0 auto;
  padding: 15px;
  color: #fff;
}
```

App.Scss ;

```
.header {
    background-color: #f1f1f1;
    padding: 20px;
```

```
    h1 {
         text-align: center;
         margin-bottom: 0;
         font-size: 20px;
    }
    p {
         text-align: center;
    }
```

```
    .logo {
         img {
              cursor: pointer;
              width: 100px;
         }

         text-align: center;
    }
```

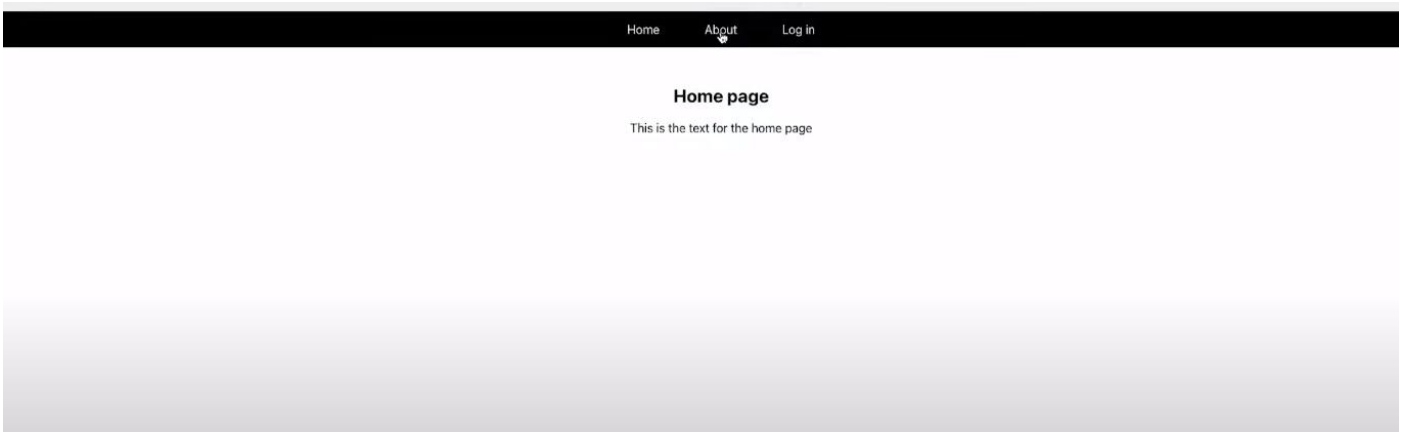
```
}
.menu {
    background-color: #000;
    text-align: center;
    padding: 15px 0;
    .menuItem {
        display: inline-block;
        margin: 0 30px;
        a {
            color: #fff;
            text-decoration: none;

        }
    }
}
.page {
    padding: 30px;
    h2 {
        text-align: center;
    }
    p {
        text-align: center;
    }
}
.inputs {
    .input {
        text-align: center;
        input {
            height: 30px;
            width: 300px;
            margin: 5px 0;
        }
    }
    .button {
        text-align: center;
        button {
            background-color: #555;
            border: none;
            padding: 15px 30px;
            color: #fff;
            width: 308px;
            margin: 10px 0;
        }
    }
}
```

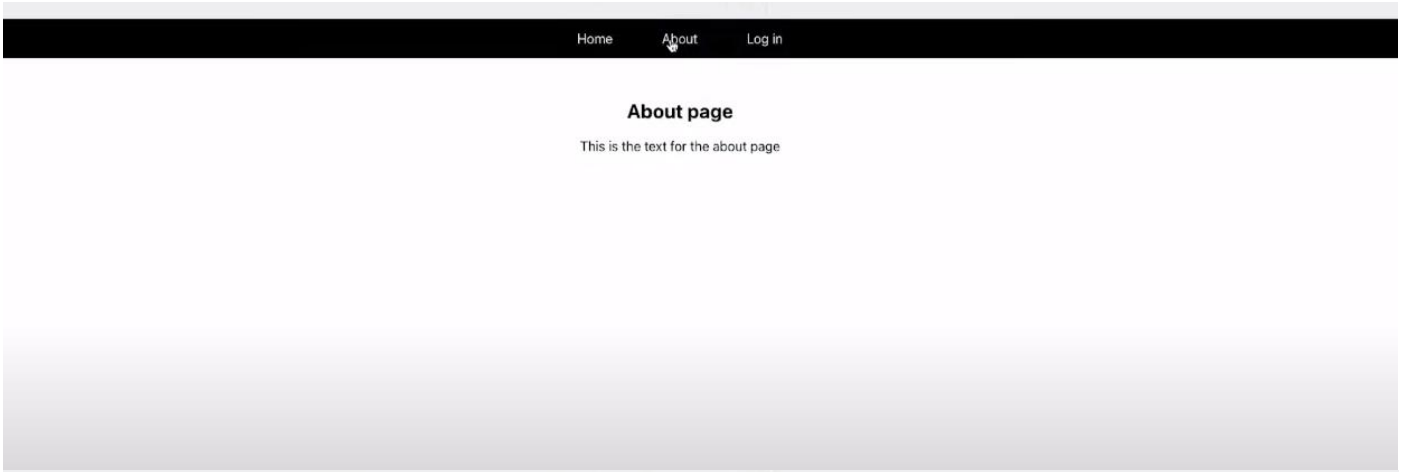
```css
.error {
    background-color: red;
    width: 280px;
    margin: 5px 0;
    margin: 0 auto;
    padding: 15px;
    color: #fff;
}
```

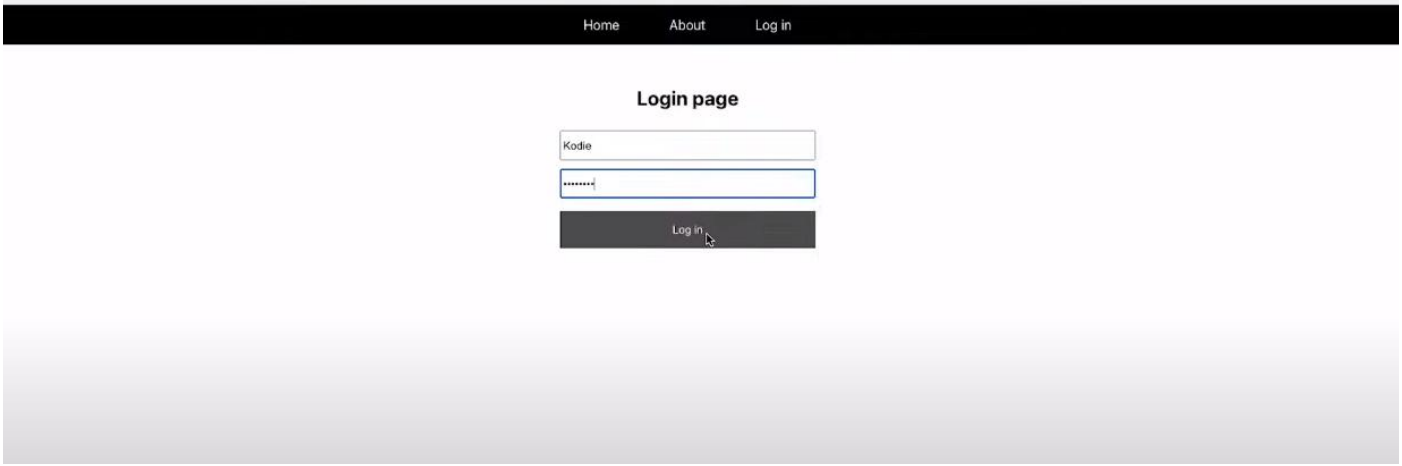
Index.js ;

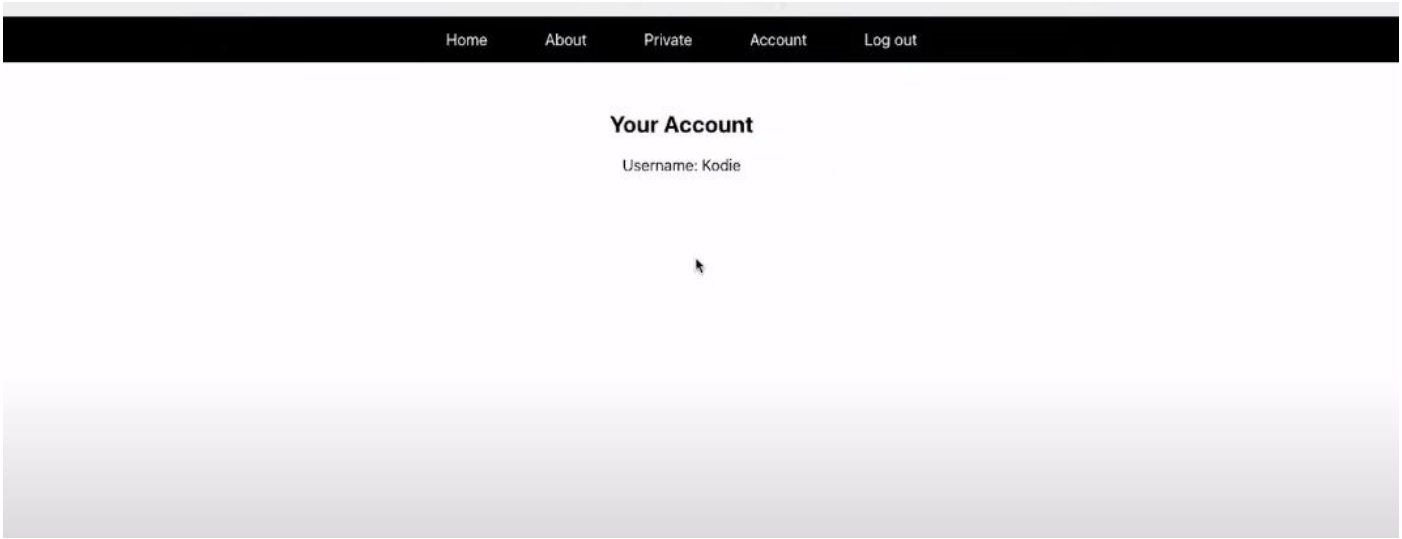
```jsx
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
    <App />
);
```

Output ;




Home
About
Log in
About page
This is the text for the about page


Home
About
Log in
Login page
Kodie
Log in


Home
About
Private
Account
Log out
Your Account
Username: Kodie

2) Write a program to create login and signup forms and on successful login need to show the dashboard with all logged in users?

Program:

App.js ;

```
import React, { useState } from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Navbar from './Navbar';
import Home from './Home';
import Login from './Login';
import Signup from './Signup';

const App = () => {
  const [users, setUsers] = useState([]);
```

```
  return (
    <Router>
      <Navbar />
      <Routes>
        <Route path="/" element={<Home users={users} />} />
        <Route path="/login" element={<Login setUsers={setUsers} />} />
        <Route path="/signup" element={<Signup setUsers={setUsers} />} />
      </Routes>
    </Router>
  );
};
```

```
export default App;
```

Home.Js ;

```
import React from 'react';
import './App.css';

const Home = ({ users }) => {
  return (
    <div className="container">
      <h1>Home Page</h1>
      <h2>Logged In Users:</h2>
      <ul>
        {users.map((user, index) => (
          <li key={index}>{user.username}</li>
        ))}
      </ul>
    </div>
  );
};
```

```
export default Home;
```

Login.js;

```
import React, { useState } from 'react';
import './App.css';

const Login = ({ onLogin }) => {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
```

```
  const handleSubmit = (e) => {
    e.preventDefault();
    onLogin({ username, password });
  };
```

```
  return (
    <div className="container">
      <h1>Login</h1>
      <form onSubmit={handleSubmit}>
        <div>
          <label>Username:</label>
          <input type="text" value={username} onChange={(e) => setUsername(e.target.value)}
required />
        </div>
        <div>
          <label>Password:</label>
          <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required />
        </div>
        <button type="submit">Login</button>
      </form>
    </div>
  );
};
```

```
export default Login;
```

Signup.js;

```
import React, { useState } from 'react';
import './App.css';

const Signup = ({ onSignup }) => {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const handleSubmit = (e) => {
    e.preventDefault();
    onSignup({ username, password });
  };
```

```
  return (
    <div className="container">
      <h1>Signup</h1>
      <form onSubmit={handleSubmit}>
        <div>
          <label>Username:</label>
```

```
          <input type="text" value={username} onChange={(e) => setUsername(e.target.value)}
required />
        </div>
        <div>
          <label>Password:</label>
          <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required />
        </div>
        <button type="submit">Signup</button>
      </form>
    </div>
  );
};
```

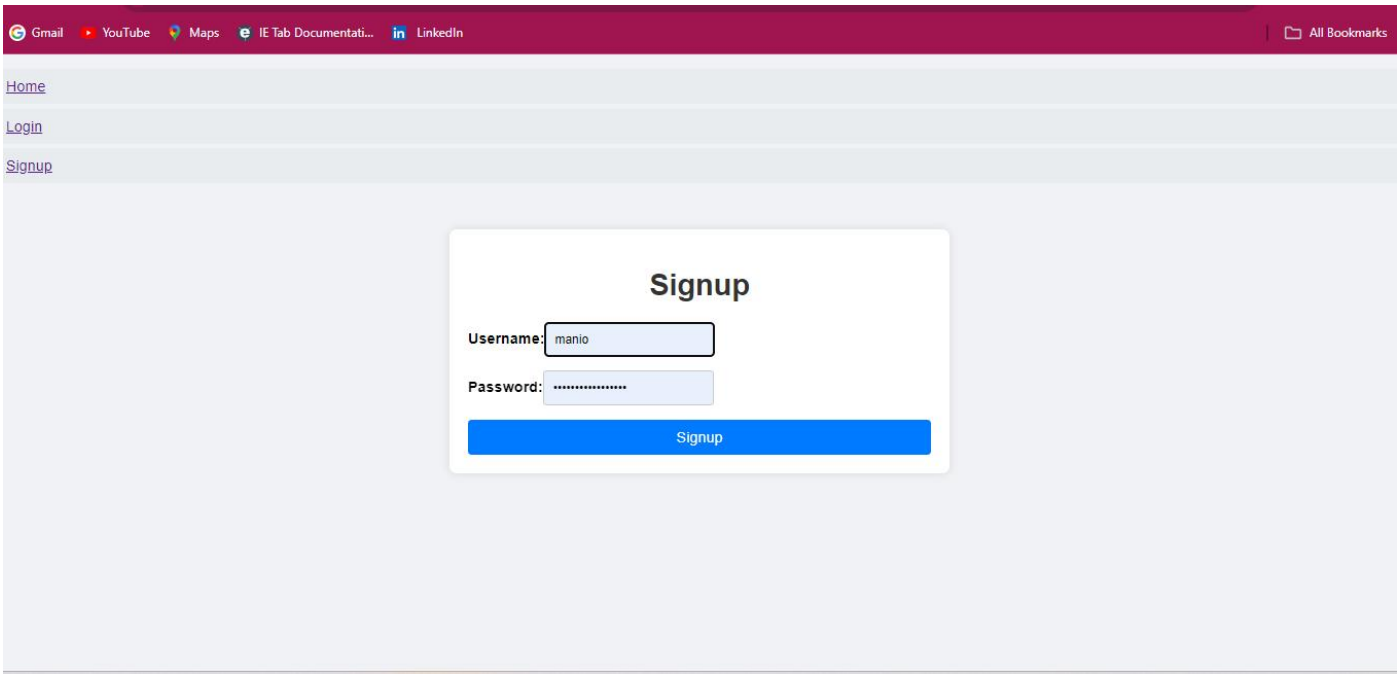
```
export default Signup;
```

Navbar.js ;

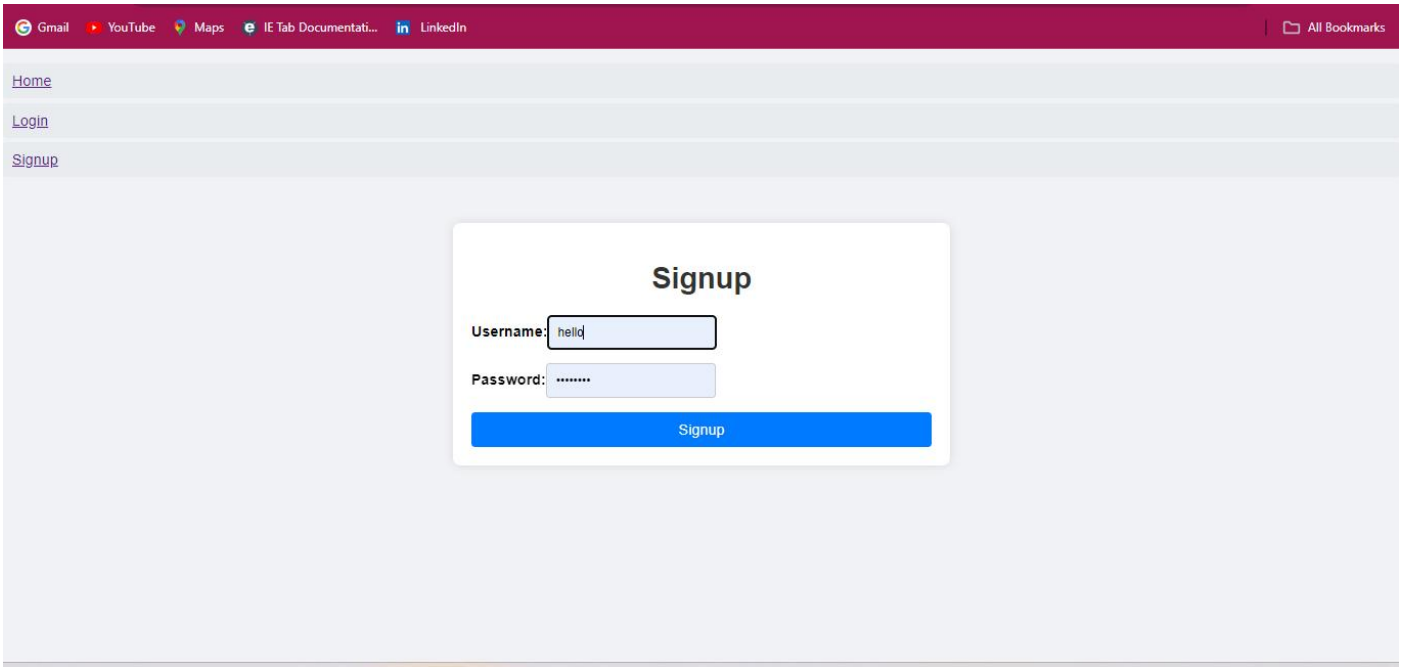
```
// src/components/Navbar.js
import React from 'react';
import { Link } from 'react-router-dom';

const Navbar = () => {
  return (
    <nav>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/login">Login</Link></li>
        <li><Link to="/signup">Signup</Link></li>
      </ul>
    </nav>
  );
};
```

```
export default Navbar;
```

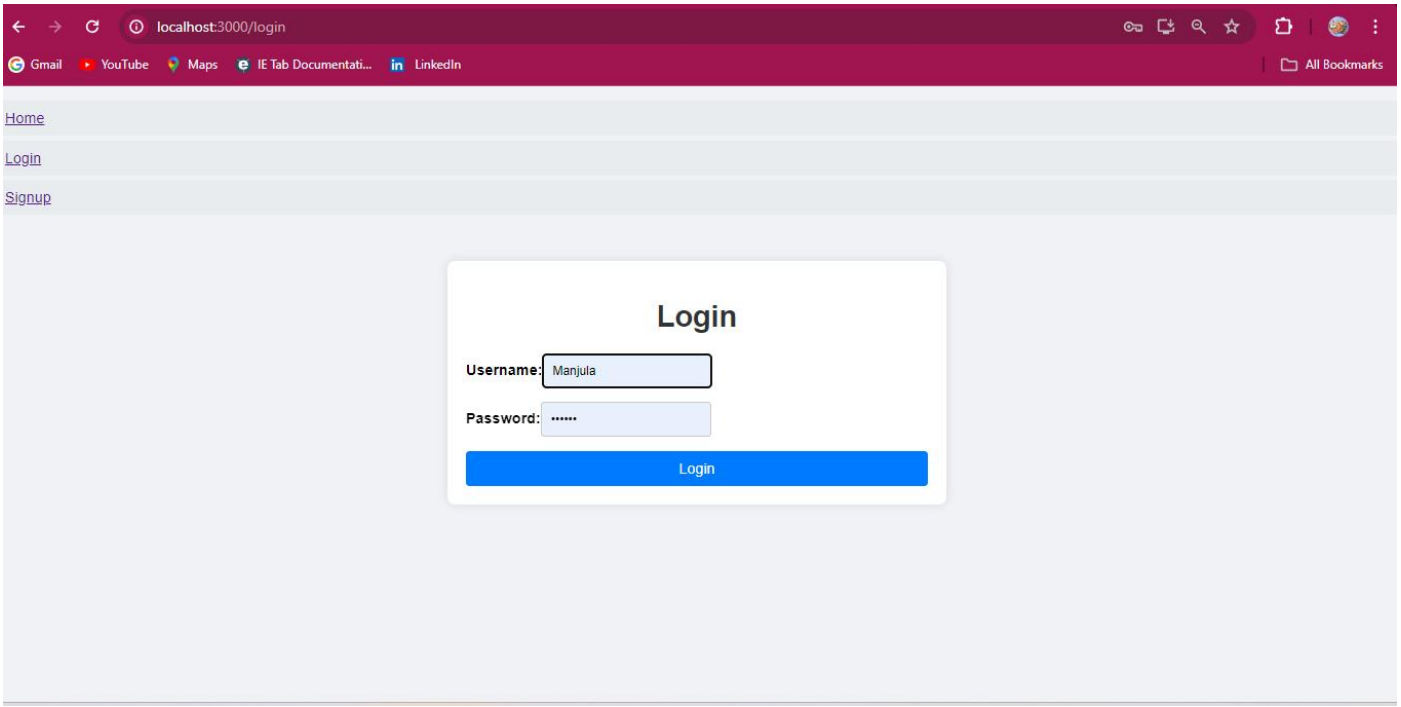
Output:

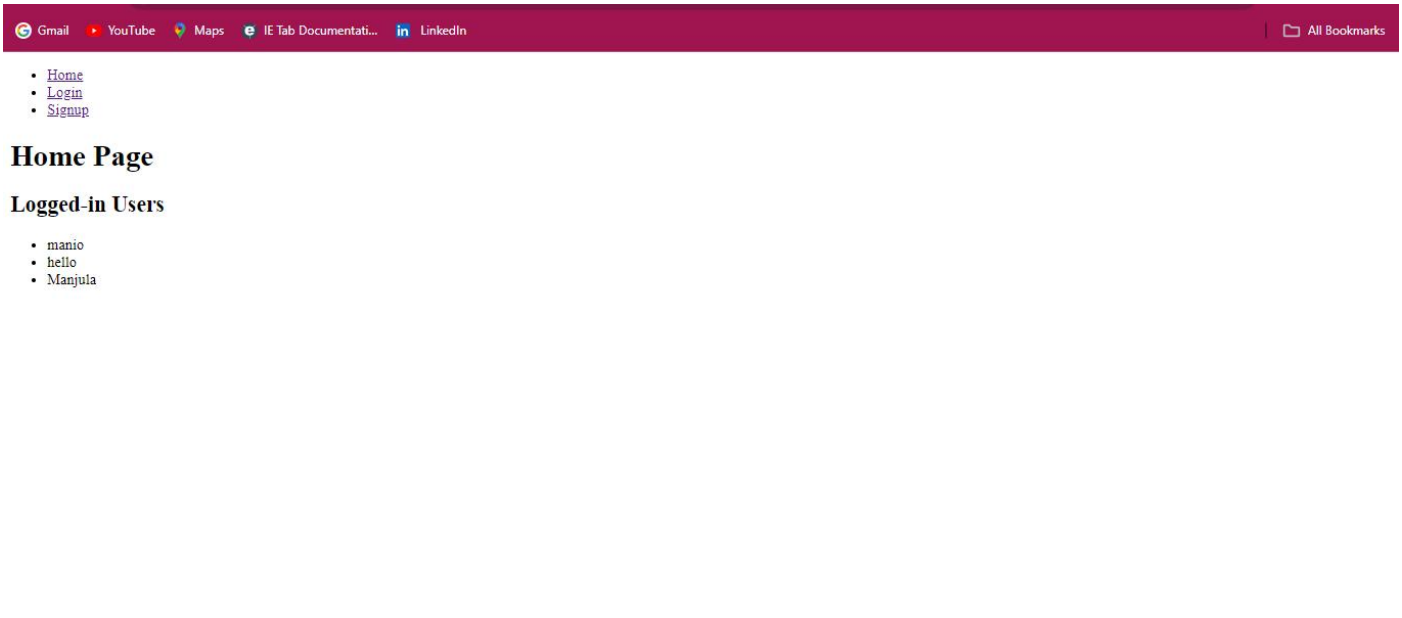

Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Signup
Username:
manio
Password:
Signup

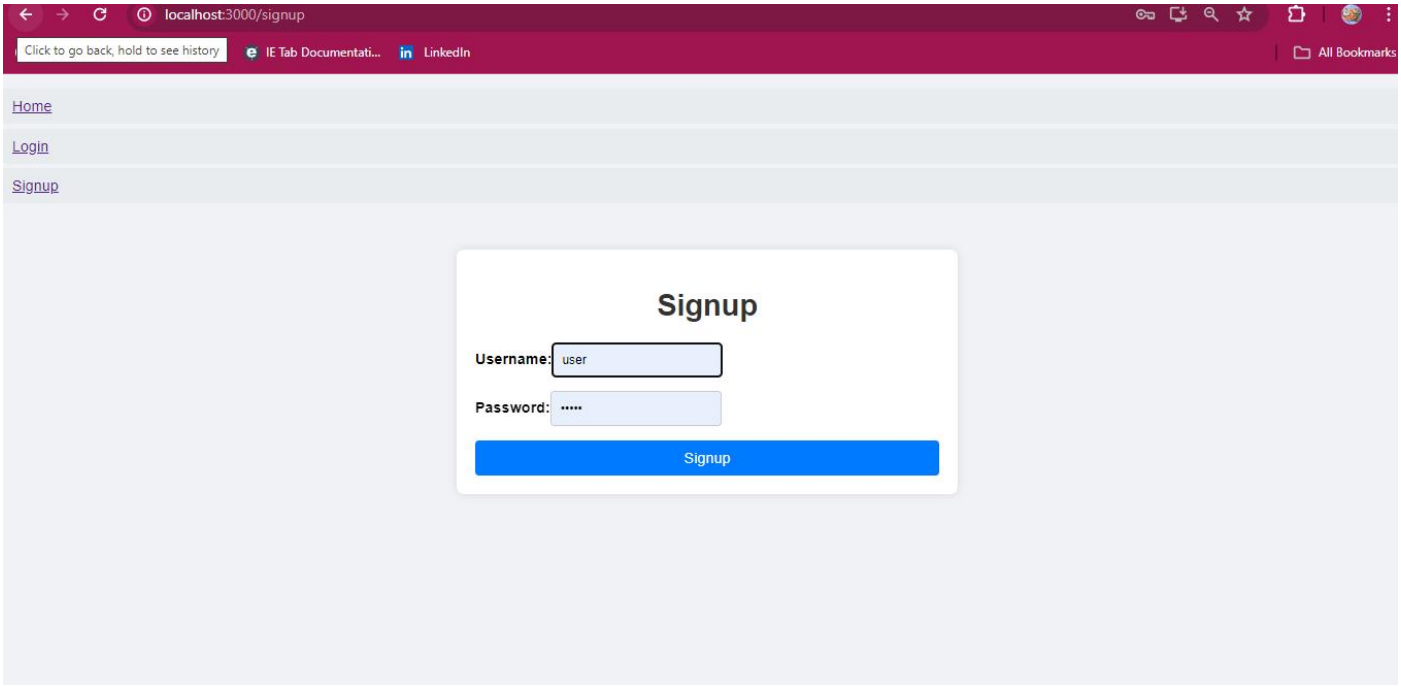

Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Home Page
Logged-in Users
manio


Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Signup
Username:
hello
Password:
Signup


Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Home Page
Logged-in Users
manio
hello


localhost:3000/login
Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Login
Username:
Manjula
Password:
Login


Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Home Page
Logged-in Users
manio
hello
Manjula


localhost:3000/signup
Click to go back, hold to see history
IE Tab Documentati...
LinkedIn
All Bookmarks
Home
Login
Signup
Signup
Username:
user
Password:
Signup

3) Write a program to execute react life cycle hooks?

A) The useContext hook

useContext allows functional components to access context values. It's a way to pass data down the component tree without explicitly passing props

Create UserContext.js

First, create a new file UserContext.js to define the context and provider component.

```
mport React, { createContext, useState } from 'react';


// Create a Context for the user
const UserContext = createContext()


// Create a provider component
const UserProvider = ({ children }) => {
  const [user, setUser] = useState({ name: 'John Doe', age: 30 });
  return (
    <UserContext.Provider value={{ user, setUser }}>
      {children}
    </UserContext.Provider>
  );
};
export { UserContext, UserProvider };
```

UserContext: Created using createContext.

UserProvider: A provider component that uses useState to manage the user state and provides this state to its children via the UserContext.Provider.UserProfile.js:

Create UserProfile.js

Next, create a new file UserProfile.js to consume the context value using useContext.

```
import React, { useContext } from 'react';
import { UserContext } from './UserContext';


const UserProfile = () => {
  const { user, setUser } = useContext(UserContext);
```

```
  const updateAge = () => {
    setUser(prevUser => ({ ...prevUser, age: prevUser.age + 1 }));
  };
  return (
    <div className="UserProfile">
      <h2>User Profile</h2>
      <p>Name: {user.name}</p>
      <p>Age: {user.age}</p>
      <button onClick={updateAge}>Increase Age</button>
    </div>

  );
};
export default UserProfile;
```

useContext(UserContext): Consumes the context value.

updateAge: A function that updates the user's age by one year using the setUser function provided by the context.

2) Usestate() and 3)Useeffect():

The following code for the Timer component. This code will use useState to manage the count state and useEffect to handle side effects such as setting up and clearing the interval.

```
import React, { useState, useEffect } from 'react';

function Timer() {
  const [count, setCount] = useState(0);
```

```
  useEffect(() => {
    // This function runs after the component mounts
    console.log('Component did mount');
    const interval = setInterval(() => {
      setCount(prevCount => prevCount + 1);
    }, 1000);
```

```
    // This return function runs before the component unmounts
    return () => {
      console.log('Component will unmount');
      clearInterval(interval);
    };
  }, []); // Empty dependency array ensures this useEffect runs only once on mount and
unmount
```

```
  useEffect(() => {
    // This function runs every time the count updates
    console.log(`Component did update - count is ${count}`);
  }, [count]); // Dependency array with count to run this useEffect on count updates
```

```
  return (
    <div className="Timer">
      <h1>Timer: {count} seconds</h1>
    </div>
  );
}
```

```
export default Timer;
```

useState:const [count, setCount] = useState(0);

initializes a state variable count with a value of 0.

useEffect for Mounting and Unmounting:

The first useEffect with an empty dependency array ([]) runs only once after the initial render, simulating **componentDidMount.**

Inside this useEffect, we set up an interval to increment the count every second.

The return function inside this useEffect simulates **componentWillUnmount** and clears the interval.

useEffect for Updating:

The second useEffect with [count] as a dependency runs every time the count state changes, simulating **componentDidUpdate.**

It logs the current count to the console whenever the count updates.

Update App.js

App.js

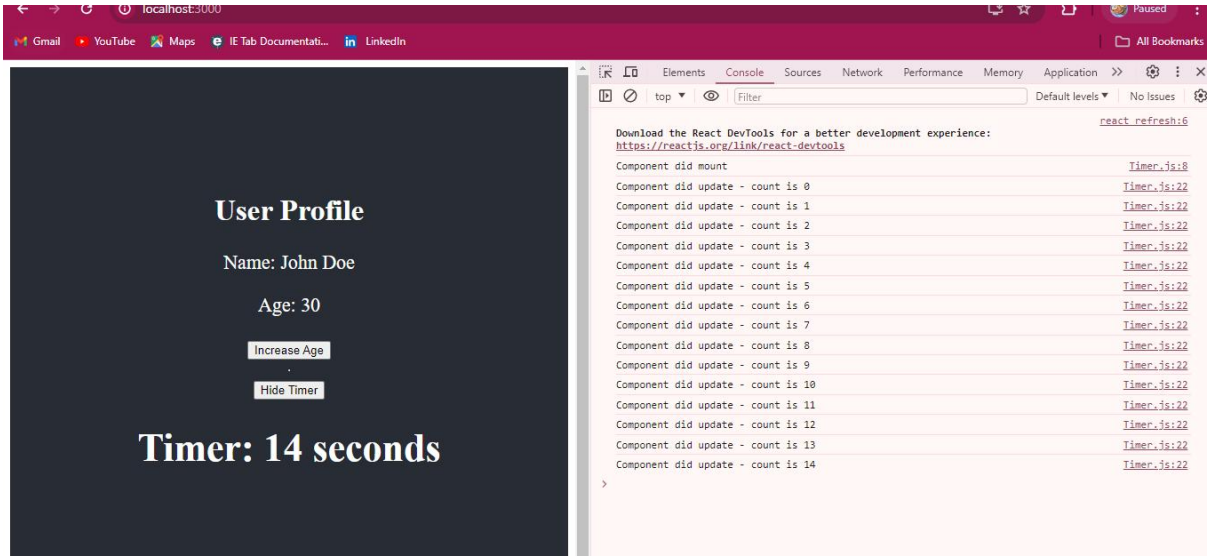
```
import React, { useState } from 'react';
import './App.css';
import Timer from './Timer';
import { UserProvider } from './UserContext';
import UserProfile from './UserProfile';


function App() {
  const [showTimer, setShowTimer] = useState(true);
  const toggleTimer = () => {
    setShowTimer(!showTimer);
  };
  return (
    <div className="App">
      <header className="App-header">
      <UserProvider>
         <UserProfile />
        </UserProvider>
        <hr></hr>
        <button onClick={toggleTimer}>
          {showTimer ? 'Hide' : 'Show'} Timer
        </button>
        {showTimer && <Timer />}
      </header>
    </div>
  );
}
export default App;
```

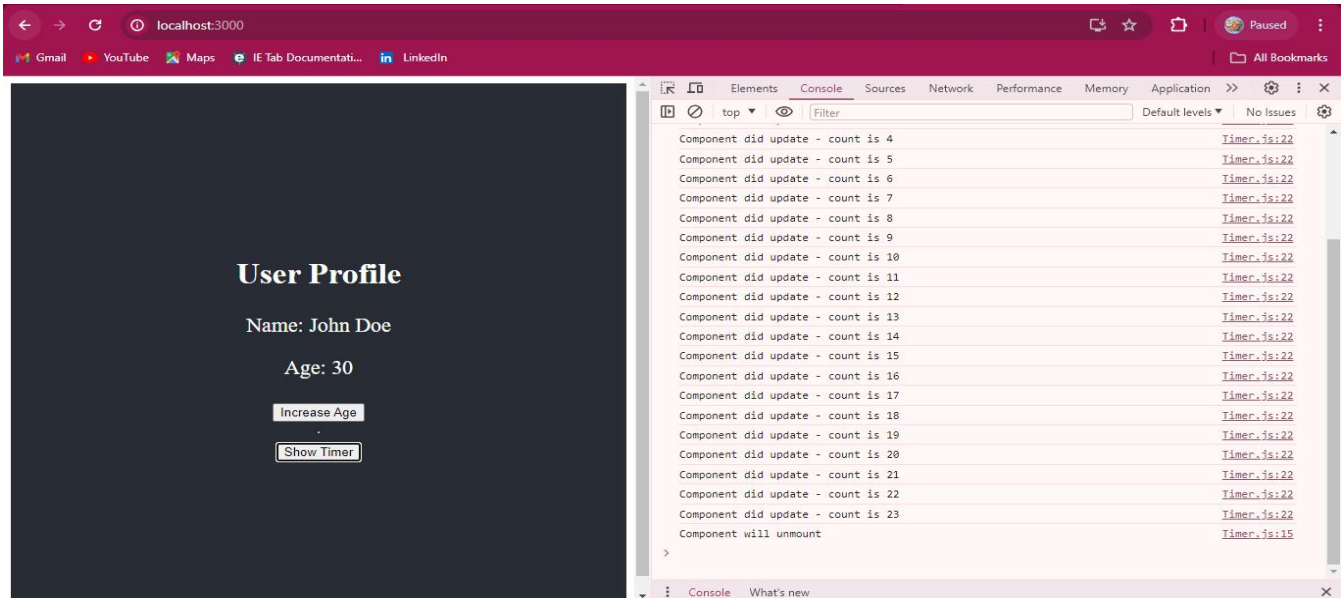
Finally, update the App.js file to use the UserProvider and UserProfile components.

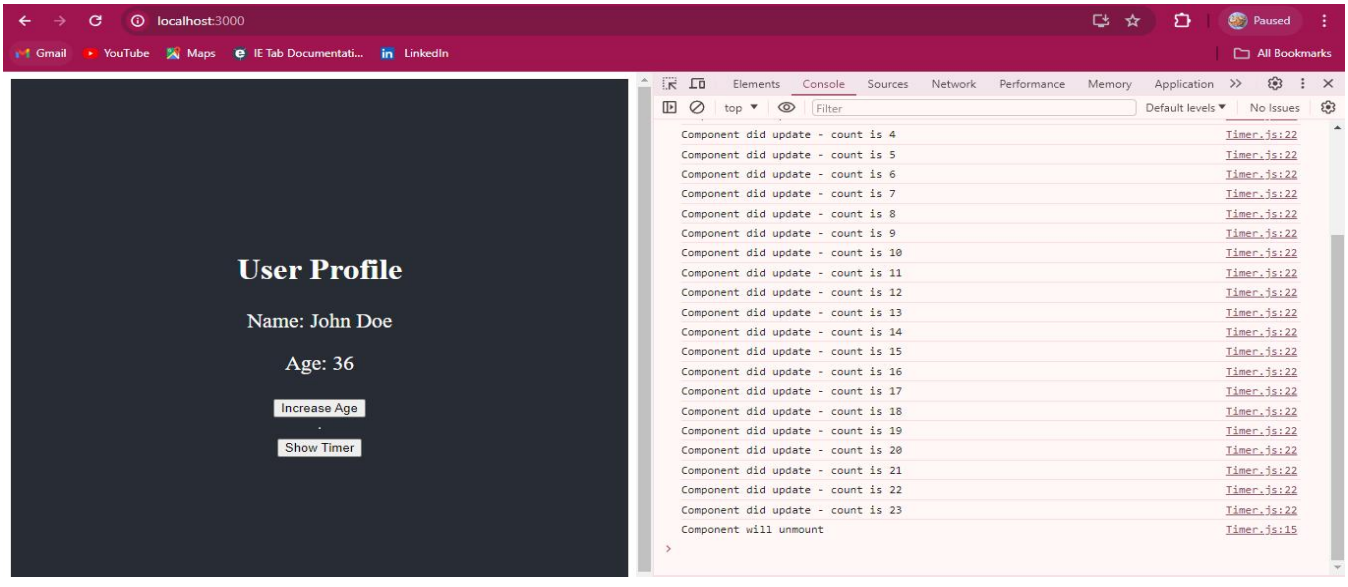
Uses a state showTimer to conditionally render the Timer component.

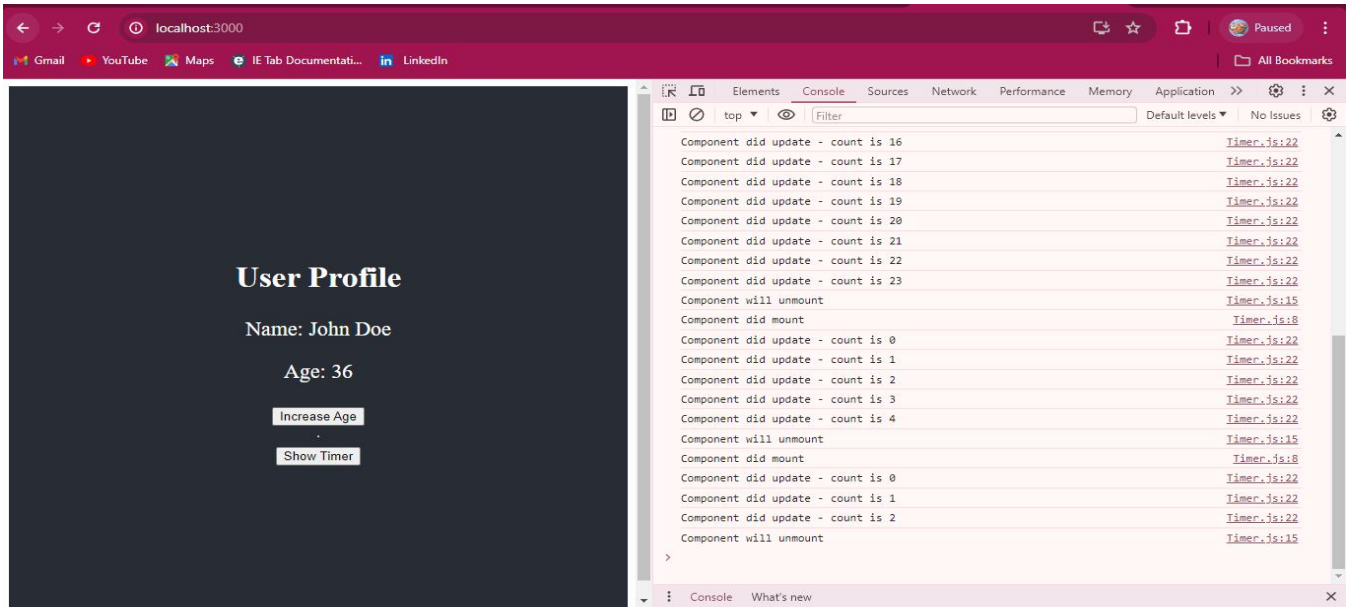
Provides a button to toggle the visibility of the Timer component, allowing us to see the mounting and unmounting lifecycle hooks in action.

Output :









4) Write a program to exchange product information from parent to child components in react?

Program ;

App.js ;

```
import React from 'react';
import GadgetList from './GadgetList';


const App = () => {
  const gadgets = [
    {
      id: 1,
      name: 'Smartphone',
      description: 'A high-end smartphone',
      price: '$699',
      image: '.\img 1'
    },
    {
      id: 2,
      name: 'Laptop',
      description: 'A powerful laptop',
      price: '$1299',
      image: '.\img 2'
```

```
    },

    {
      id: 3,
      name: 'Smartwatch',
      description: 'A smartwatch with various features',
      price: '$199',
      image: '.\img 3'
    },
    {
      id: 4,
      name: 'Bluetooth Speaker',
      description: 'A Bluetooth with high frequency',
      price: '$399',
      image:'.\img 4'
    },
    {
      id: 5,
      name: 'LED TV',
      description: 'A LED Tv with quality display ',
      price: '$1000',
      image:'.\img 5'
    }
  ];
```

```
  return (
    <div>
      <h1>Gadget Information</h1>
      <GadgetList gadgets={gadgets} />
    </div>
  );
};
```

```
export default App;
```

GadgetCard.js;

```
import React from 'react';


const GadgetCard = ({ gadget }) => {
  return (
    <div style={styles.card}>
      <img src={gadget.image} alt={gadget.name} style={styles.image} />
      <h2>{gadget.name}</h2>
      <p>{gadget.description}</p>
      <p><strong>{gadget.price}</strong></p>
    </div>
  );
};
```

```
const styles = {
  card: {
    border: '1px solid #ccc',
    borderRadius: '8px',
    padding: '16px',
    width: '200px',
    boxShadow: '2px 2px 12px rgba(0,0,0,0.1)',
    textAlign: 'center',
  },
  image: {
    width: '100%',
    height: 'auto',
    borderRadius: '8px 8px 0 0',
  }
};
```

```
export default GadgetCard;
```

GadgetList.js;

```jsx
import React from 'react';
import GadgetCard from './GadgetCard';


const GadgetList = ({ gadgets }) => {
  return (
    <div style={styles.container}>
      {gadgets.map(gadget => (
        <GadgetCard key={gadget.id} gadget={gadget} />
      ))}
    </div>
  );
};
```

```jsx
const styles = {
  container: {
    display: 'flex',
    flexWrap: 'wrap',
    gap: '20px',
    justifyContent: 'center',
  },
};
```

```jsx
export default GadgetList;
```

Output;

Gadget Information

Smartphone

A high-end smartphone

$699

Laptop

A powerful laptop

$1299

Smartwatch

A smartwatch with various features

$199

Bluetooth Speaker

A Bluetooth with high frequency

$399

LED TV

A LED Tv with quality display

$1000

5) Write a program to do post request in Axios and Fetch methods?

App.js ;

```
import React from 'react';
import PostWithAxios from './PostWithAxios';
import FetchPostExample from './FetchPostExample';

const App = () => {
  return (
    <div>
      <PostWithAxios />
      <FetchPostExample />
    </div>
  );
};
```

```
export default App;
```

App.css;

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f5f0;
  margin: 0;
  padding: 0;
}

.container {
  width: 100%;
  max-width: 500px;
  margin: 50px auto;
  padding: 20px;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
}
```

```
h1 {
  text-align: center;
  color: #333;
}
```

```
form {
  display: flex;
  flex-direction: column;
}
```

```
form div {
  margin-bottom: 15px;
}
```

```
label {
```

```
  font-weight: bold;
  margin-bottom: 5px;
}
```

```
input {
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
}
```

```
button {
  padding: 10px;
  background-color: #065bcaa8;
  color: #fff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
}
button:hover {
  background-color: #04608a;
}
ul {
  list-style-type: none;
  padding: 0;
}
li {
  background-color: #e9ecef;
  margin: 5px 0;
  padding: 10px;
  border-radius: 4px;
}
```

Fetch.js ;

```
// FetchPostExample.js
import React, { useState } from 'react';

const FetchPostExample = () => {
  const [data, setData] = useState(null);
  const [error, setError] = useState(null);
```

```
  const handleSubmit = async (event) => {
    event.preventDefault();
    try {
      const response = await fetch('https://jsonplaceholder.typicode.com/posts', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          title: 'foo',
          body: 'bar',
```

```
          userId: 1,
        }),
      });
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      const result = await response.json();
      setData(result);
    } catch (error) {
      setError(error.message);
    }
  };
```

```
  return (
    <div>
      <h2>Fetch POST Request</h2>
      <form onSubmit={handleSubmit}>
        <button type="submit">Submit</button>
      </form>
      {data && <div>Response: {JSON.stringify(data)}</div>}
      {error && <div>Error: {error}</div>}
    </div>
  );
};
```

```
export default FetchPostExample;
```

Axios.Js;

```
// AxiosPostExample.js
import React, { useState } from 'react';
import axios from 'axios';

const PostWithAxios = () => {
  const [data, setData] = useState(null);
  const [error, setError] = useState(null);
```

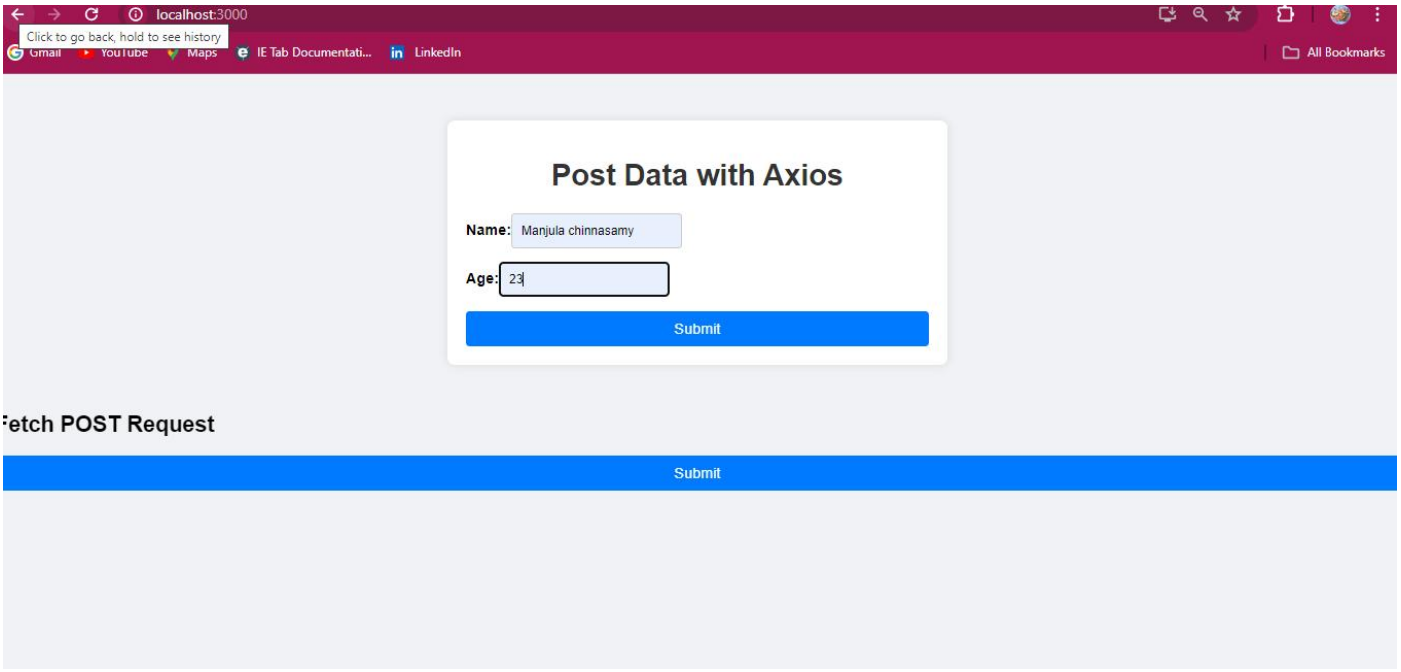
```
  const handleSubmit = async (event) => {
    event.preventDefault();
    try {
      const response = await axios.post('https://jsonplaceholder.typicode.com/posts', {
        title: 'foo',
        body: 'bar',
        userId: 1,
      });
      setData(response.data);
    } catch (error) {
      setError(error.message);
    }
  };
```

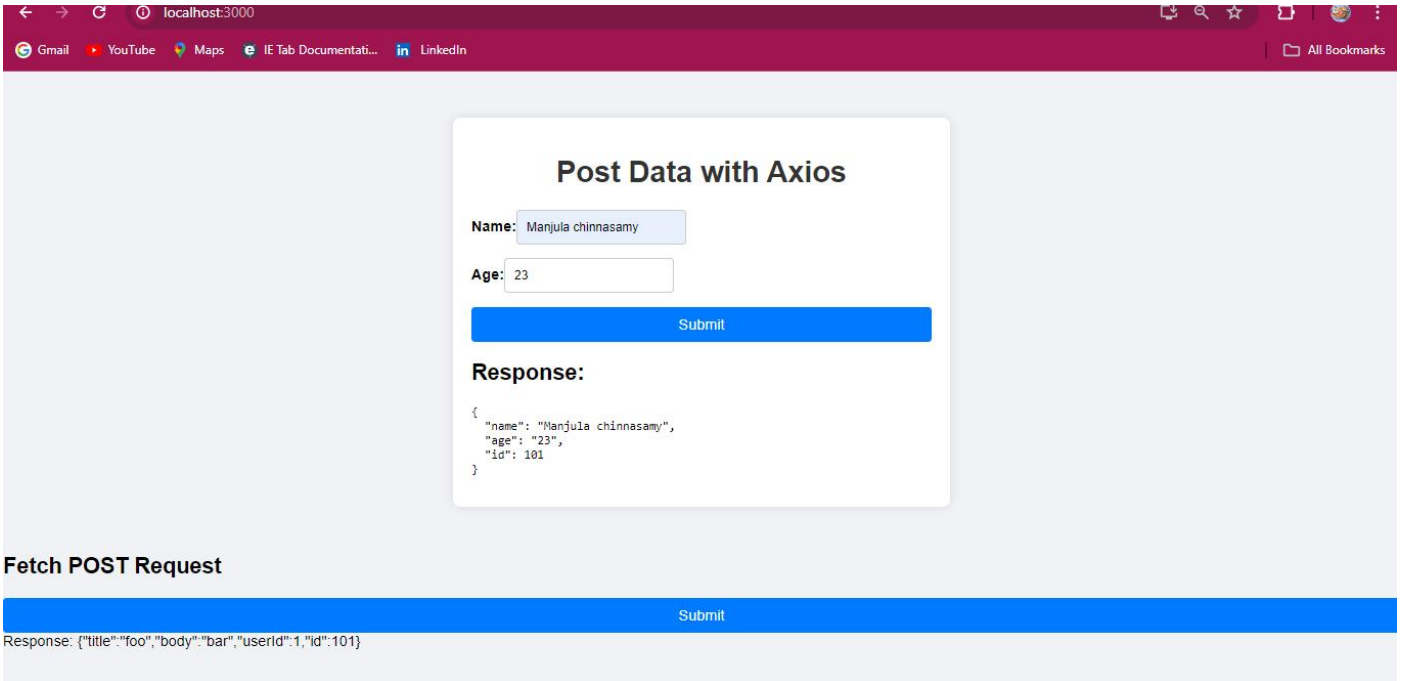
```
  return (
    <div>
```

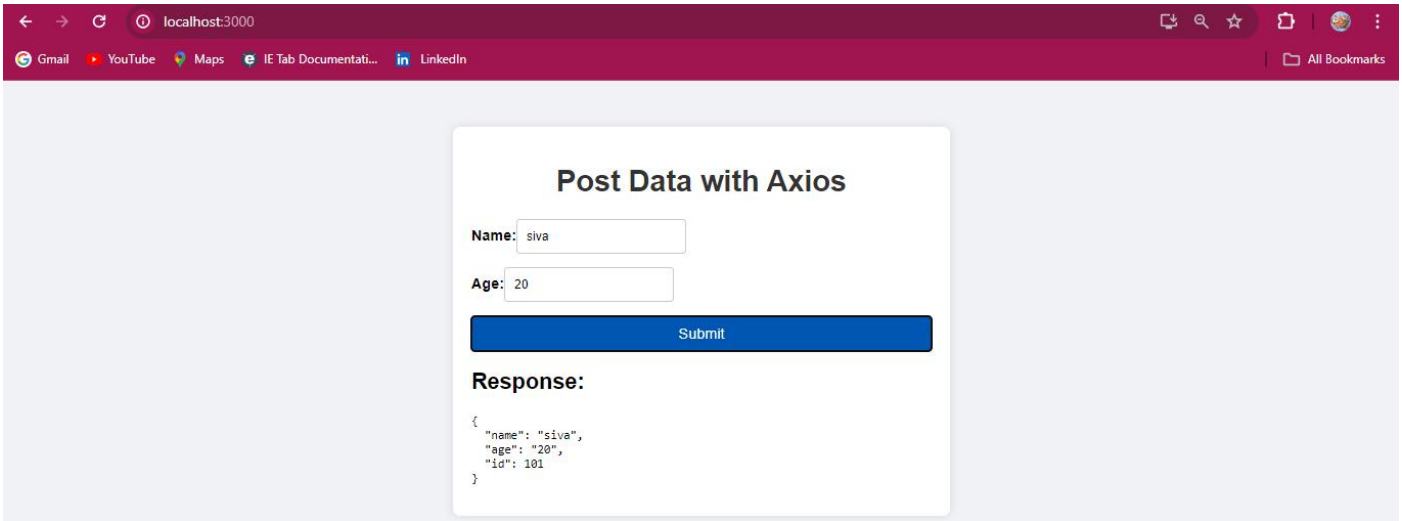
```
      <h2>Axios POST Request</h2>
      <form onSubmit={handleSubmit}>
        <button type="submit">Submit</button>
      </form>
      {data && <div>Response: {JSON.stringify(data)}</div>}
      {error && <div>Error: {error}</div>}
    </div>
  );
};
```

```
export default PostWithAxios;
```

Output ;

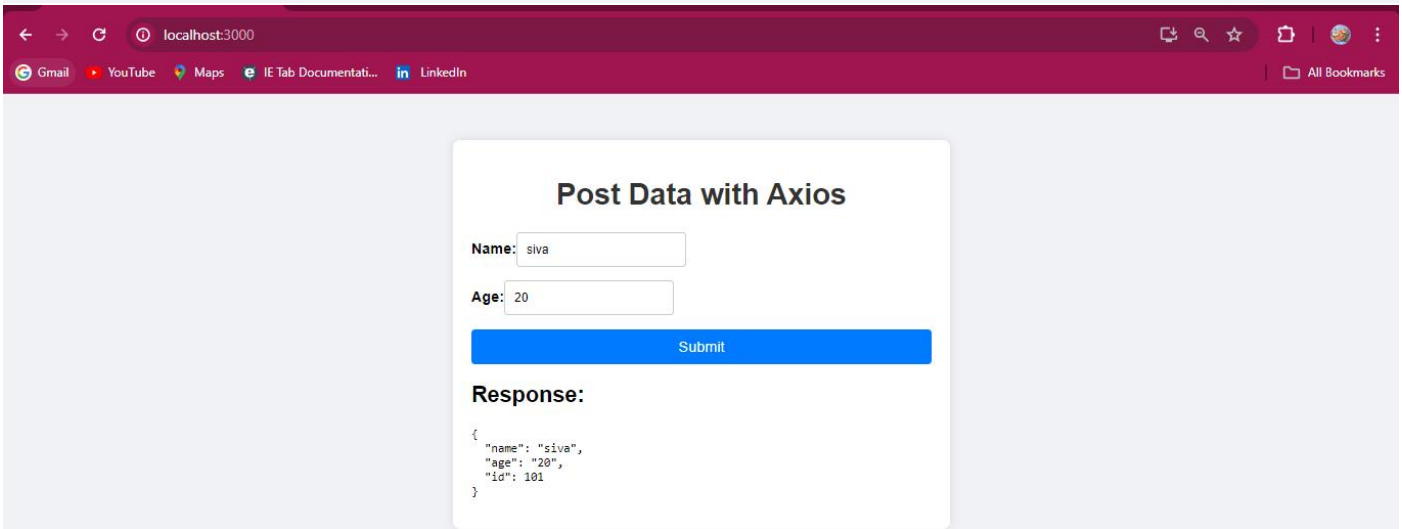






Fetch POST Request

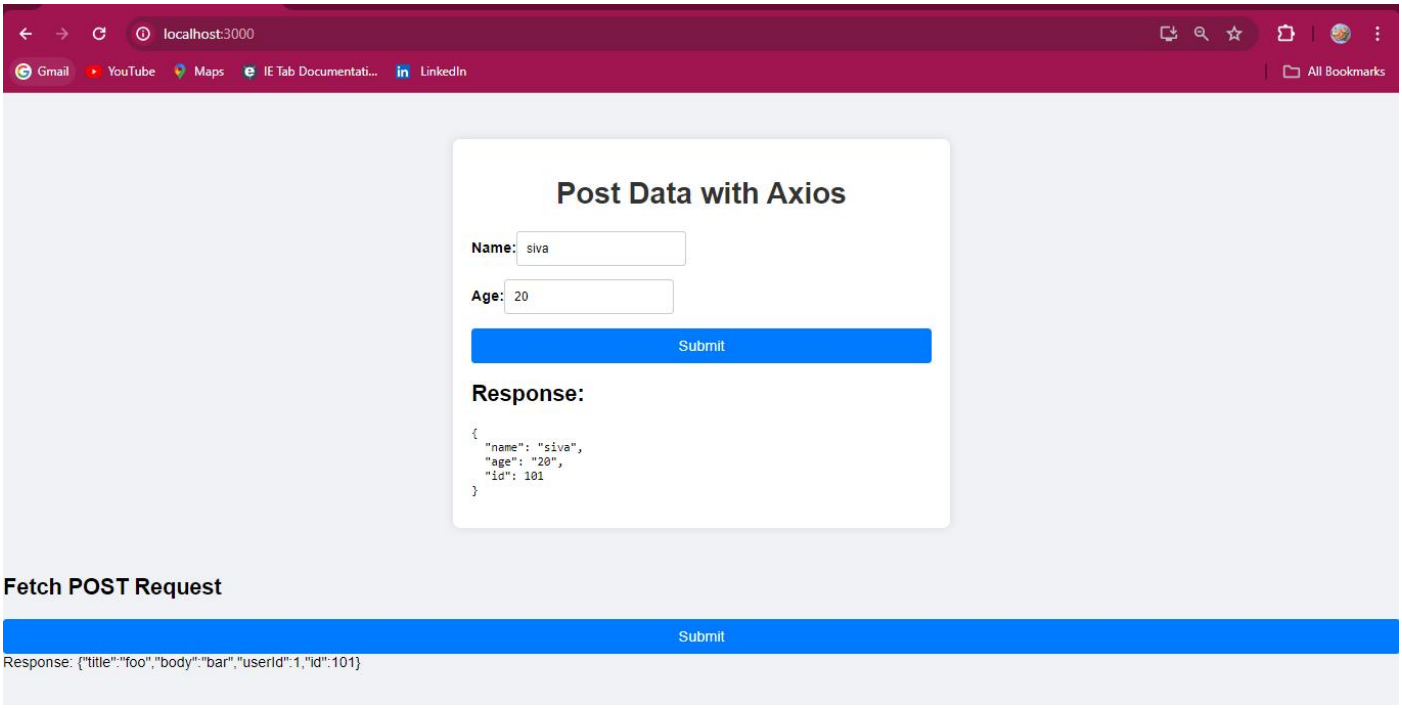
Submit

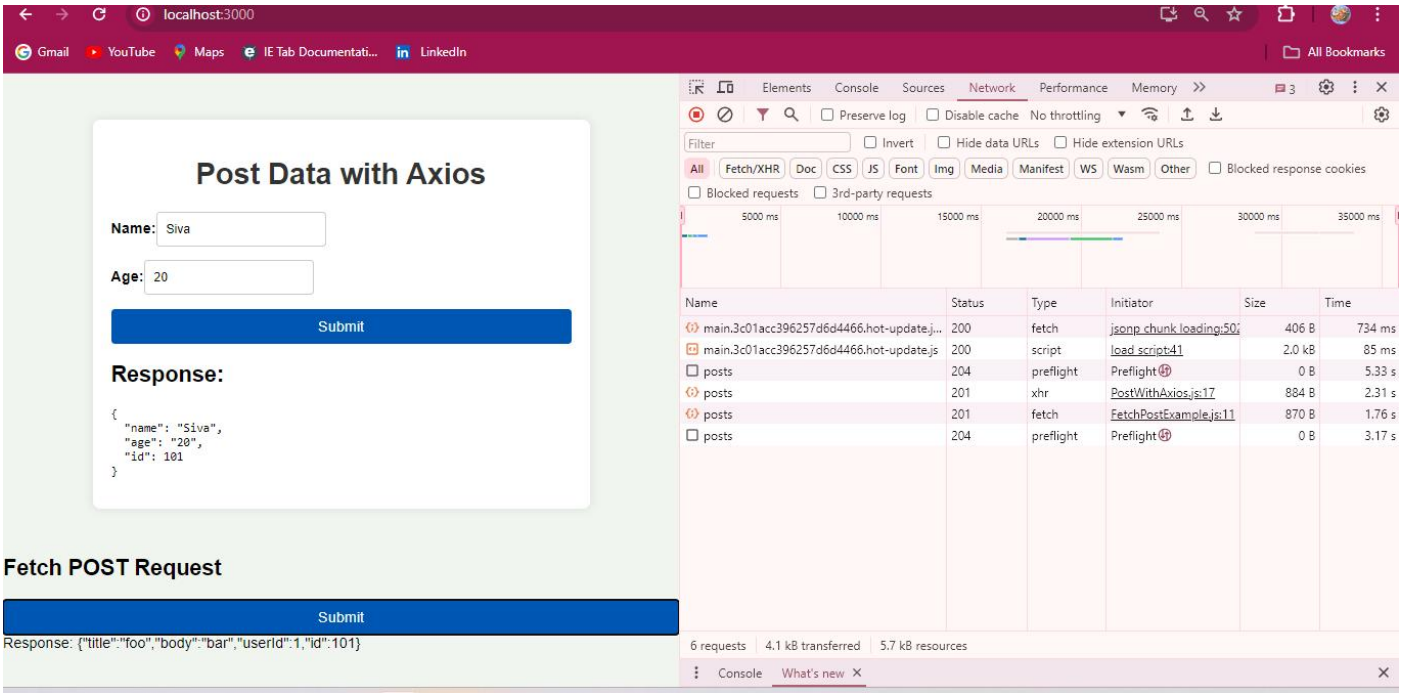


Fetch POST Request

Submit

Response: {"title":"foo","body":"bar","userId":1,"id":101}


localhost:3000
Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Post Data with Axios
Name: siva
Age: 20
Submit
Response:
{
"name": "siva",
"age": "20",
"id": 101
}
Fetch POST Request
Submit
Response: {"title":"foo","body":"bar","userId":1,"id":101}


localhost:3000
Gmail
YouTube
Maps
IE Tab Documentati...
LinkedIn
All Bookmarks
Post Data with Axios
Name: Siva
Age: 20
Submit
Response:
{
"name": "Siva",
"age": "20",
"id": 101
}
Fetch POST Request
Submit
Response: {"title":"foo","body":"bar","userId":1,"id":101}
Elements
Console
Sources
Network
Performance
Memory
Preserve log
Disable cache
No throttling
Filter
Invert
Hide data URLs
Hide extension URLs
All
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Manifest
WS
Wasm
Other
Blocked response cookies
Blocked requests
3rd-party requests
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
Name
Status
Type
Initiator
Size
Time
main.3c01acc396257d6d4466.hot-update.j... 200 fetch jsonp chunk loading:50 406 B 734 ms
main.3c01acc396257d6d4466.hot-update.js 200 script load script:41 2.0 kB 85 ms
posts 204 preflight Preflight 0 B 5.33 s
posts 201 xhr PostWithAxios.js:17 884 B 2.31 s
posts 201 fetch FetchPostExample.js:11 870 B 1.76 s
posts 204 preflight Preflight 0 B 3.17 s
6 requests
4.1 kB transferred
5.7 kB resources
Console
What's new