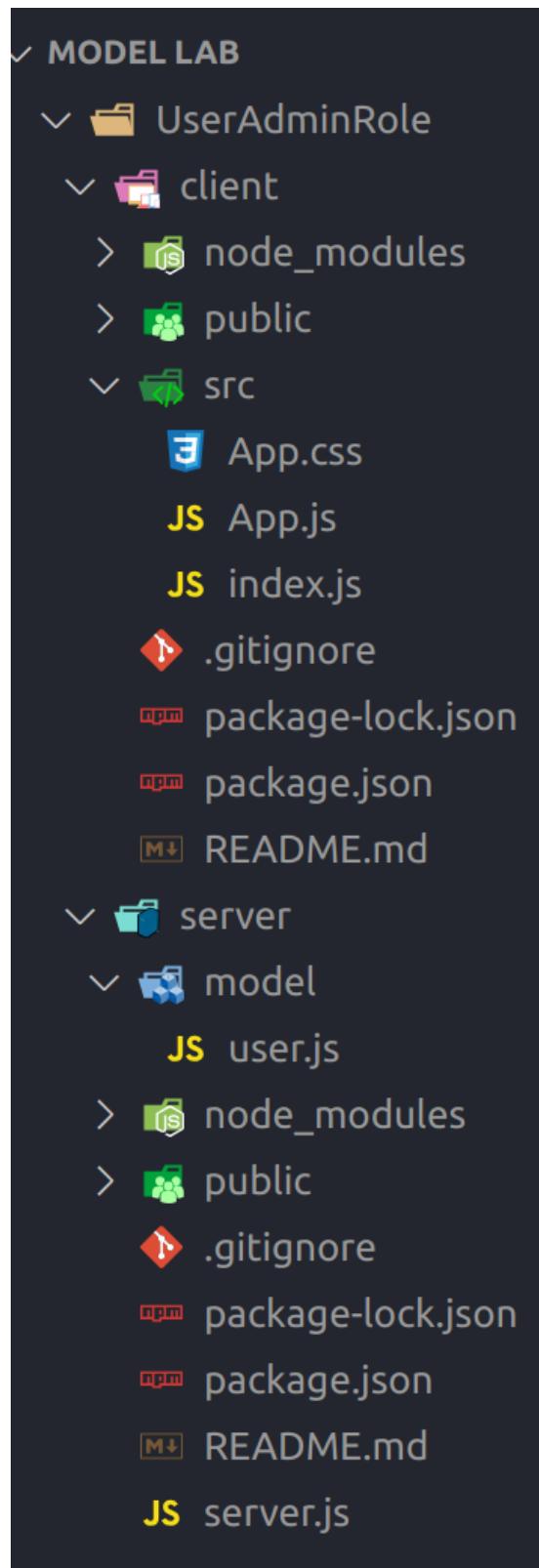


Name: M.Rohith

Model Lab

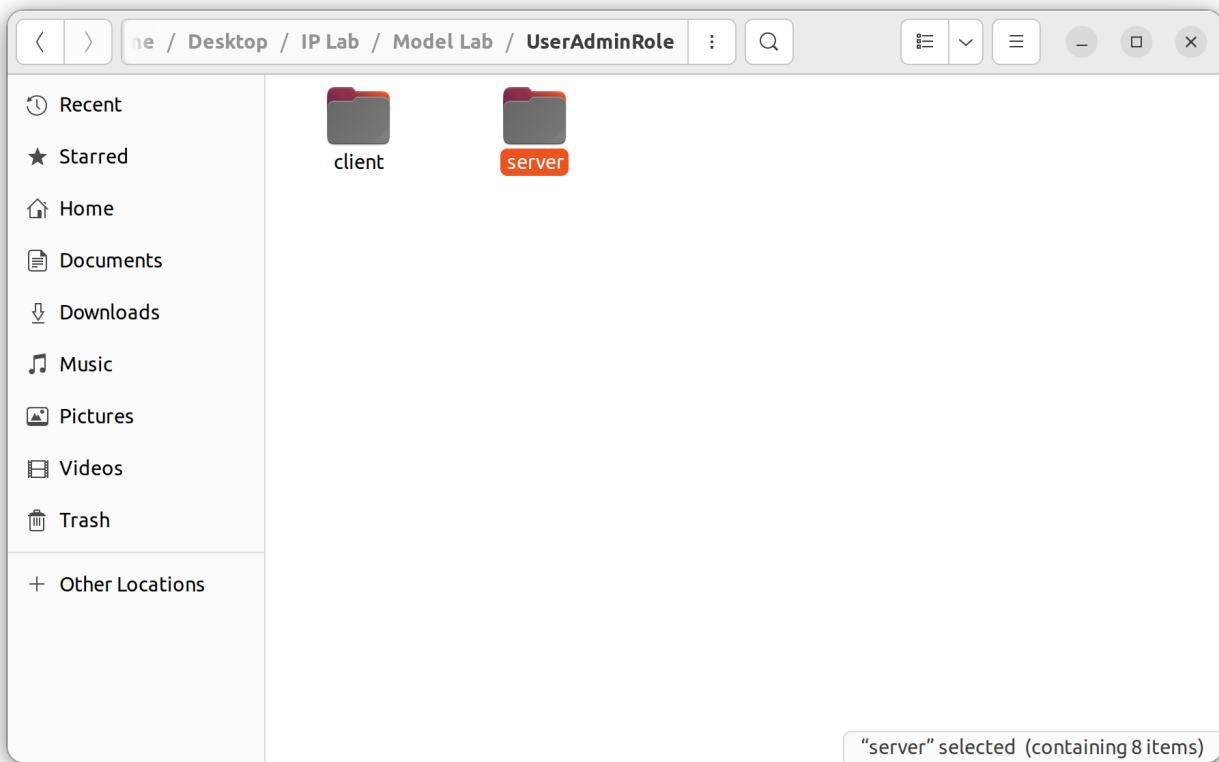
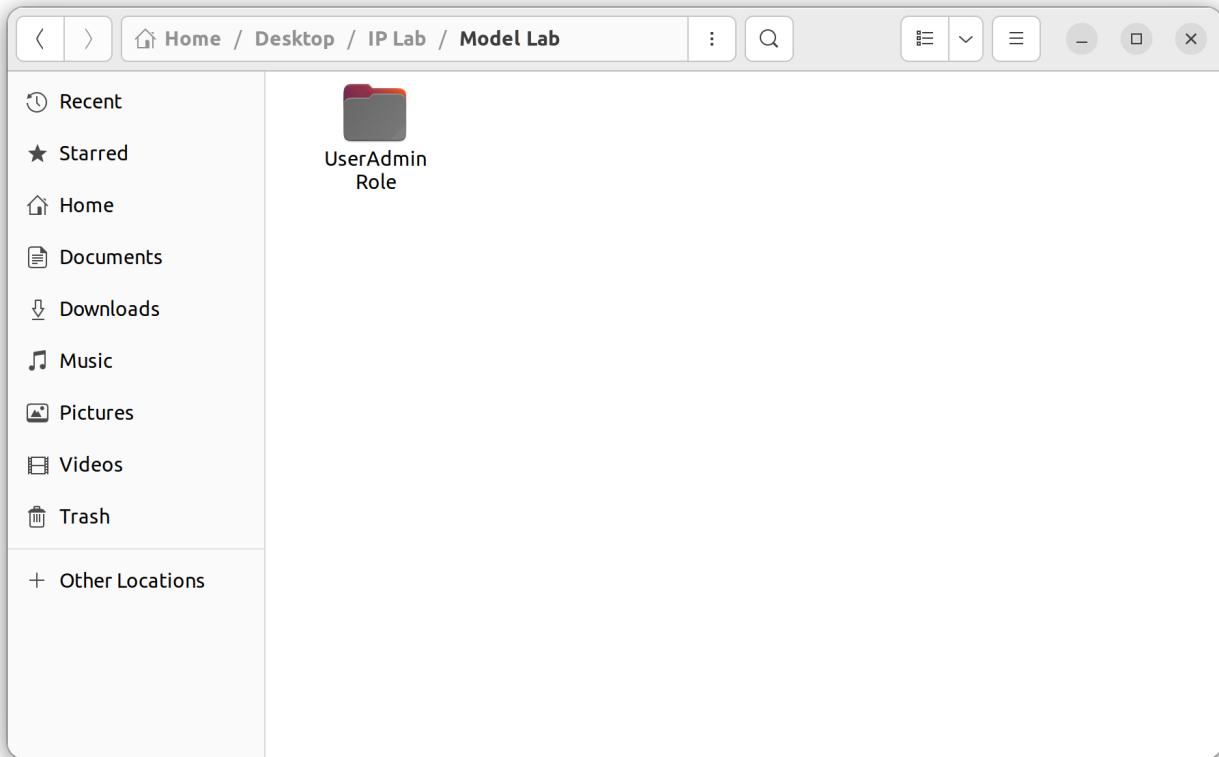
Admin: Perform CRUD operations

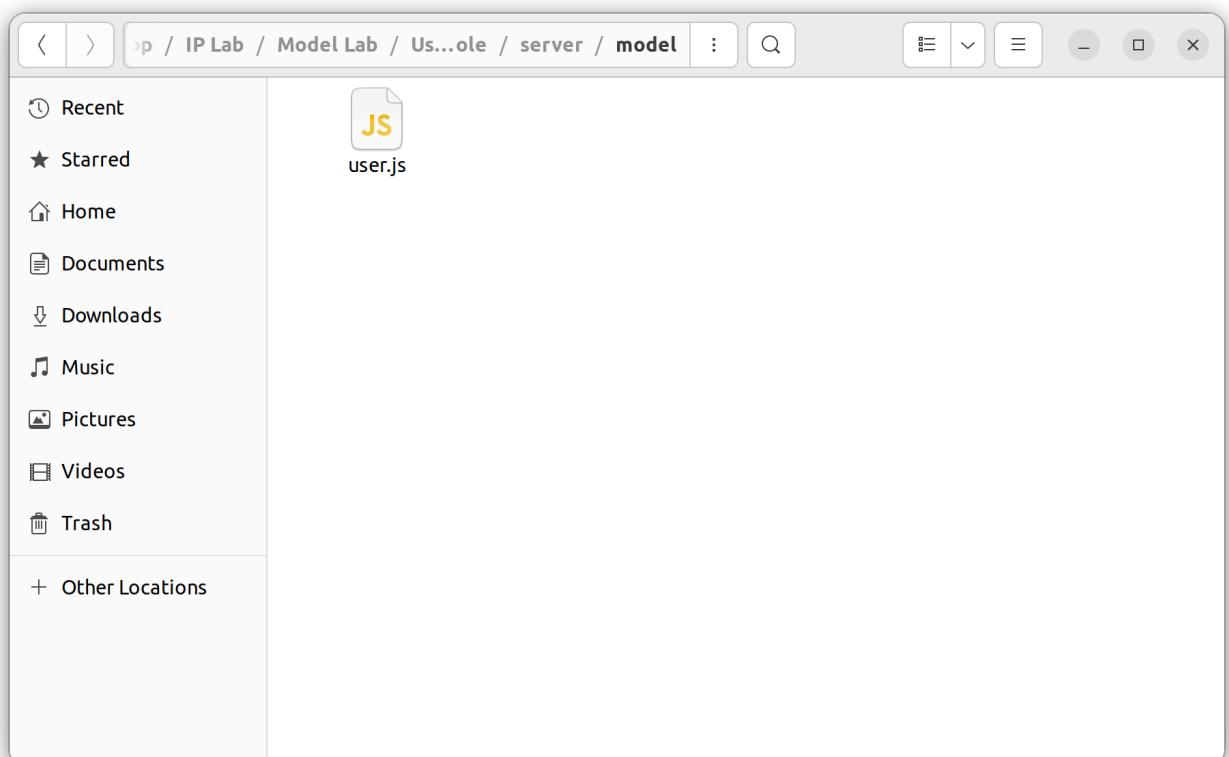
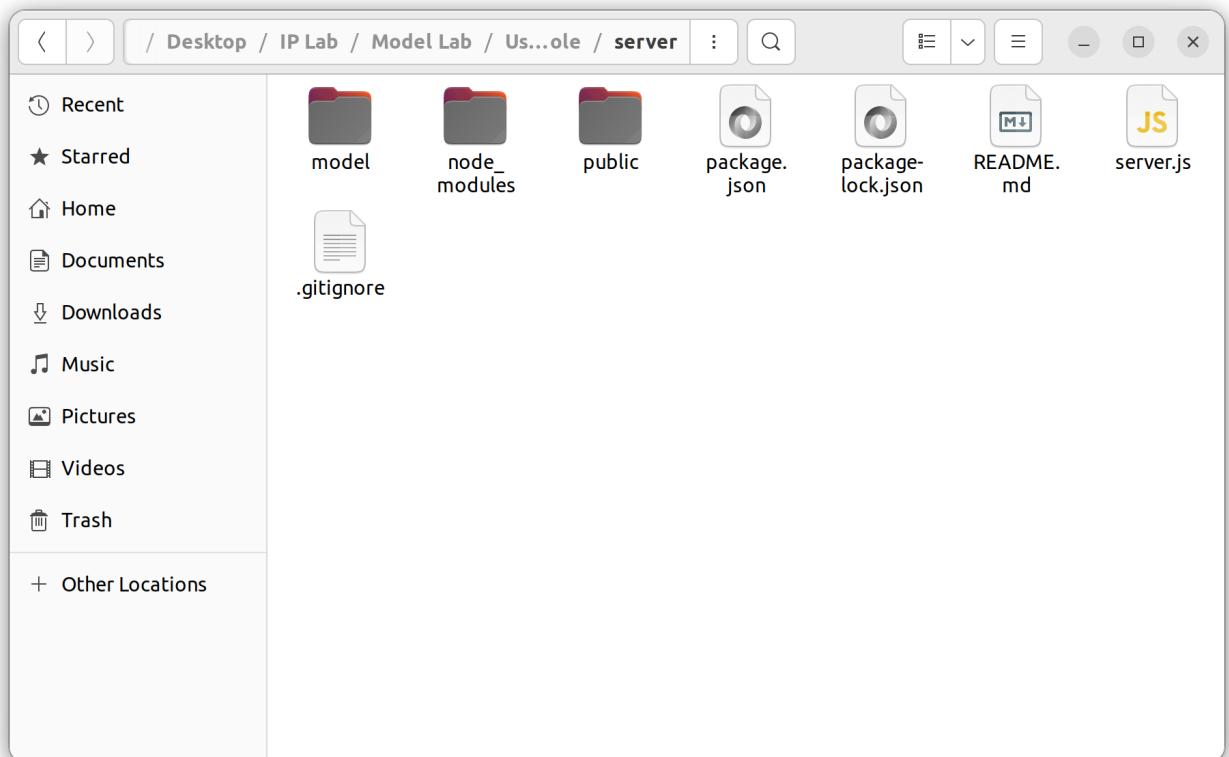
User: Perform only read operations

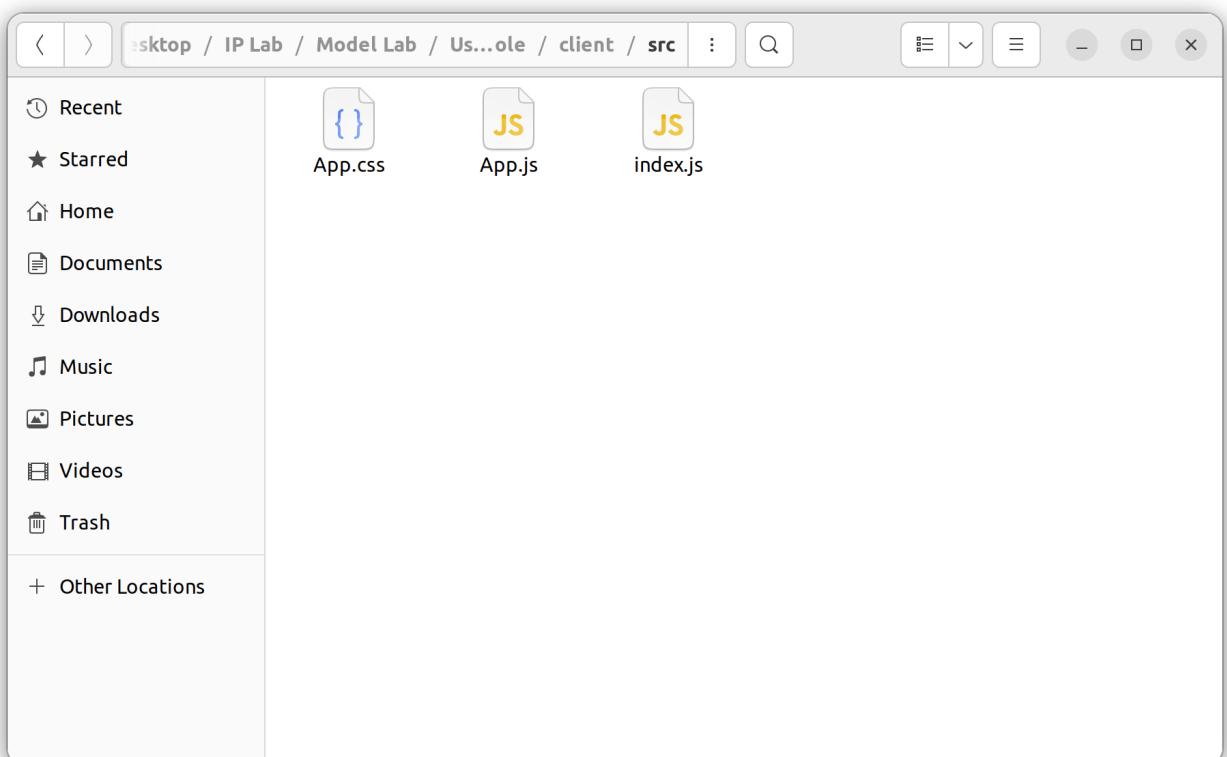
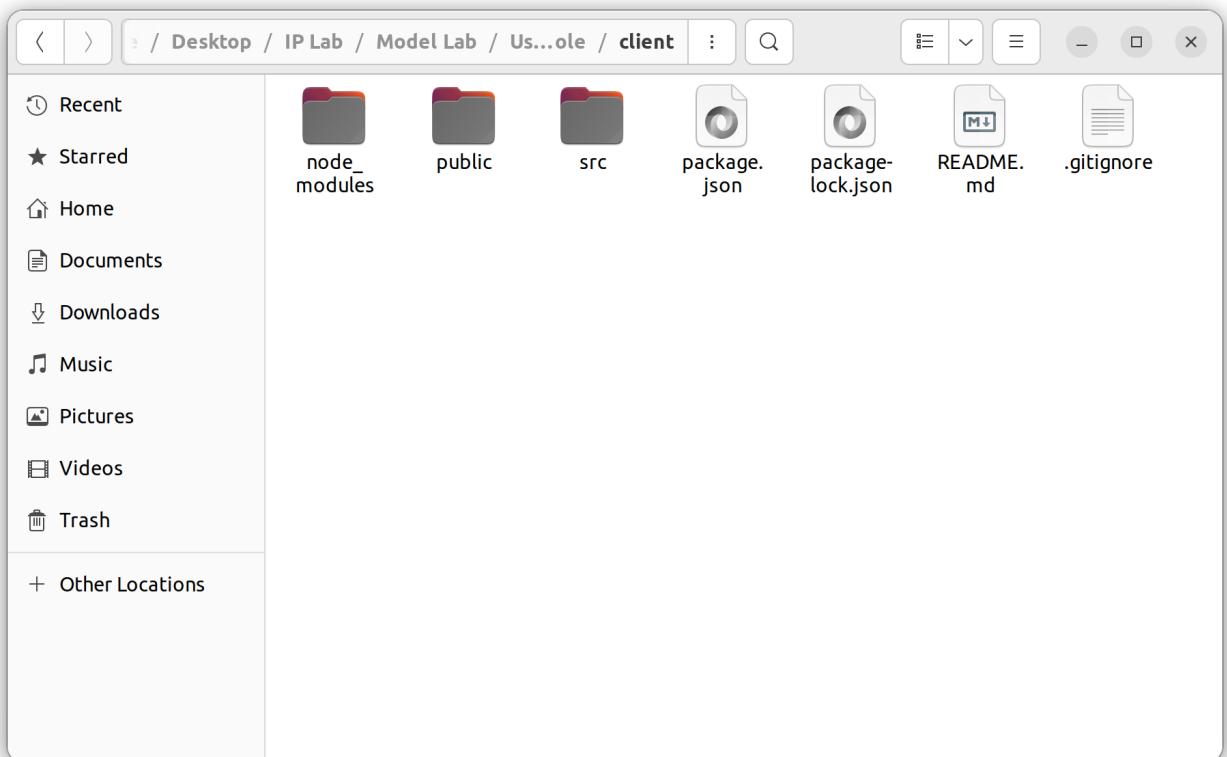


```
✓ MODEL LAB
  ✓ UserAdminRole
    ✓ client
      > node_modules
      > public
    ✓ src
      App.css
      App.js
      index.js
      .gitignore
      package-lock.json
      package.json
      README.md
  ✓ server
    ✓ model
      user.js
      node_modules
      public
      .gitignore
      package-lock.json
      package.json
      README.md
      server.js
```

Hierarchy:







Program code:

user.js

```
const mongoose=require("mongoose");

const UserSchema=new mongoose.Schema({
    username:String,
    userage:Number
});

const UserModel=mongoose.model("users",UserSchema);

module.exports=UserModel;
```

server.js

```
const mongoose=require("mongoose");
const express=require("express");
const cors=require("cors");
const UserModel=require("./model/user.js");

const app=express();
app.use(cors());
app.use(express.json());

mongoose.connect("mongodb://localhost:27017/MVCPractice");

app.post("/send",(req,res)=>{
    const {username,userage}=req.body; //LHS variable name should be same as the var name assigned and sent in App.js
    console.log(username+" "+userage);
    console.log(`Searching ${username} with ${userage}`);
    UserModel.findOne({username:username,userage:userage})
        .then(result=>res.json(result))
        .catch(err=>console.log(err))
});

app.post("/insert",(req,res)=>{
    const {username,userage}=req.body; //LHS variable name should be same as the var name assigned and sent in App.js
    console.log(`Inserting ${username} with ${userage}`);
    UserModel.create({username:username,userage:userage})
        .then(result=>res.json(result))
        .catch(err=>console.log(err))
});

app.post("/delete",(req,res)=>{
    const {username,userage}=req.body; //LHS variable name should be same as the var name assigned and sent in App.js
    console.log(`Deleting ${username} with age ${userage}`);
    UserModel.deleteOne({username:username,userage:userage})
```

```

        .then(result=>res.json(result))
        .catch(err=>console.log(err))
    });

app.post("/update", (req, res) => {
    const {username, userage, newuserName, newUserAge} = req.body; //LHS variable name should
    be same as the var name assigned and sent in App.js
    console.log(`Updating ${username} having ${userage} with ${newuserName} having $ ${newUserAge}`);
    UserModel.findOneAndUpdate({username, userage},
    {username: newuserName, userage: newUserAge}, {new: true})
        .then(result=>res.json(result))
        .catch(err=>console.log(err))
});

// New endpoint to fetch all users
app.get("/users", (req, res) => {
    UserModel.find()
        .then(users => res.json(users))
        .catch(err => console.log(err))
});

app.listen(3001, () => {
    console.log("Server is listening on port 3001");
})

```

App.js

```

import { useState } from 'react';
import axios from 'axios';
import './App.css'; // Import the CSS file

export default function App() {
    let [name, setName] = useState("");
    let [age, setAge] = useState("");
    let [newName, setNewName] = useState("");
    let [newAge, setNewAge] = useState("");
    let [display, setDisplay] = useState("");
    let [users, setUsers] = useState([]);
    let [isAdmin, setIsAdmin] = useState(false);

    function sendData() {
        console.log("Sending data to server");
        axios.post("http://localhost:3001/send", { username: name, userage: age })
            .then(res => (setDisplay(res.data ? `${res.data.username} aged ${res.data.userage} is
Present` : "User Not Found")))
            .catch(err => console.log(err))
    }

    function insertData() {
        console.log("Inserting...");
        axios.post("http://localhost:3001/insert", { username: name, userage: age })
    }
}

```

```

        .then(res => console.log(res))
        .catch(err => console.log(err))
    }

    function updateData() {
        console.log("Updating...");
        axios.post("http://localhost:3001/update", { username: name, userage: age, newUserName: newName, newuserAge: newAge })
            .then(res => console.log(res))
            .catch(err => console.log(err))
    }

    function deleteData() {
        console.log("Deleting...");
        axios.post("http://localhost:3001/delete", { username: name, userage: age })
            .then(res => console.log(res))
            .catch(err => console.log(err))
    }

    function fetchAllUsers() {
        console.log("Fetching all users...");
        axios.get("http://localhost:3001/users")
            .then(res => setUsers(res.data))
            .catch(err => console.log(err))
    }

    return (
        <div className="container">
            <center>
                <h1>MVC Architecture</h1>
                <div className="button-group">
                    <button className={isAdmin ? "" : "active"} onClick={() => setIsAdmin(false)}>User</button>
                    <button className={isAdmin ? "active" : ""} onClick={() => setIsAdmin(true)}>Admin</button>
                </div>
                <br /><br />
                {isAdmin ? (
                    <div className="admin-section">
                        <h2>Admin Operations</h2>
                        <input type="text" placeholder="username" onChange={(e) => setName(e.target.value)} />
                        <br />
                        <input type="text" placeholder="userage" onChange={(e) => setAge(e.target.value)} />
                        <br /><br />
                        <button onClick={sendData}>Search</button>
                        <br />
                        <button onClick={insertData}>Insert</button>
                        <br />
                        <button onClick={deleteData}>Delete</button>
                        <br /><br />
                ) : null}
            </center>
        </div>
    )
}

```

```

        <input type="text" placeholder="new name" onChange={(e) =>
setNewName(e.target.value)} />
        <br />
        <input type="text" placeholder='new age' onChange={(e) =>
setNewAge(e.target.value)} />
        <br /><br />
        <button onClick={updateData}>Update</button>
        <br /><br />
        <button onClick={fetchAllUsers}>Fetch All Users</button>
        <br /><br />
        <h2>{display}</h2>
        <h2>All Users:</h2>
        <ul>
            {users.map(user => (
                <li key={user._id}>{user.username} - {user.uservage}</li>
            )))
        </ul>
    </div>
):(
    <div className="user-section">
        <h2>User Operations</h2>
        <input type="text" placeholder="username" onChange={(e) =>
setName(e.target.value)} />
        <br />
        <input type="text" placeholder="uservage" onChange={(e) =>
setAge(e.target.value)} />
        <br /><br />
        <button onClick={sendData}>Search</button>
        <br /><br />
        <h2>{display}</h2>
    </div>
)
</center>
</div>
)
}

```

App.css

```

.container {
    font-family: Arial, sans-serif;
    margin: 20px;
    padding: 20px;
    background-color: beige;
}

h1 {
    color: purple;
}

.button-group {
    margin-bottom: 20px;
}

```

```
}

.button-group button {
    margin: 0 10px;
    padding: 10px 20px;
    cursor: pointer;
    border: 1px solid #ccc;
    background-color: orange;
    border-radius: 5px;
    transition: background-color 0.3s, color 0.3s;
}

.button-group button.active {
    background-color: red;
    color: #fff;
}

.button-group button:hover {
    background-color: red;
    color: #fff;
}

input {
    margin: 10px 0;
    padding: 10px;
    width: 200px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    margin: 10px;
    padding: 10px 20px;
    cursor: pointer;
    border: none;
    background-color: blue;
    color: #fff;
    border-radius: 5px;
    transition: background-color 0.3s;
}

button:hover {
    background-color: purple;
}

.admin-section, .user-section {
    text-align: left;
    display: inline-block;
}

h2 {
    color: red;
```

```
}

ul {
    list-style-type: none;
    padding: 0;
    color:purple;
}

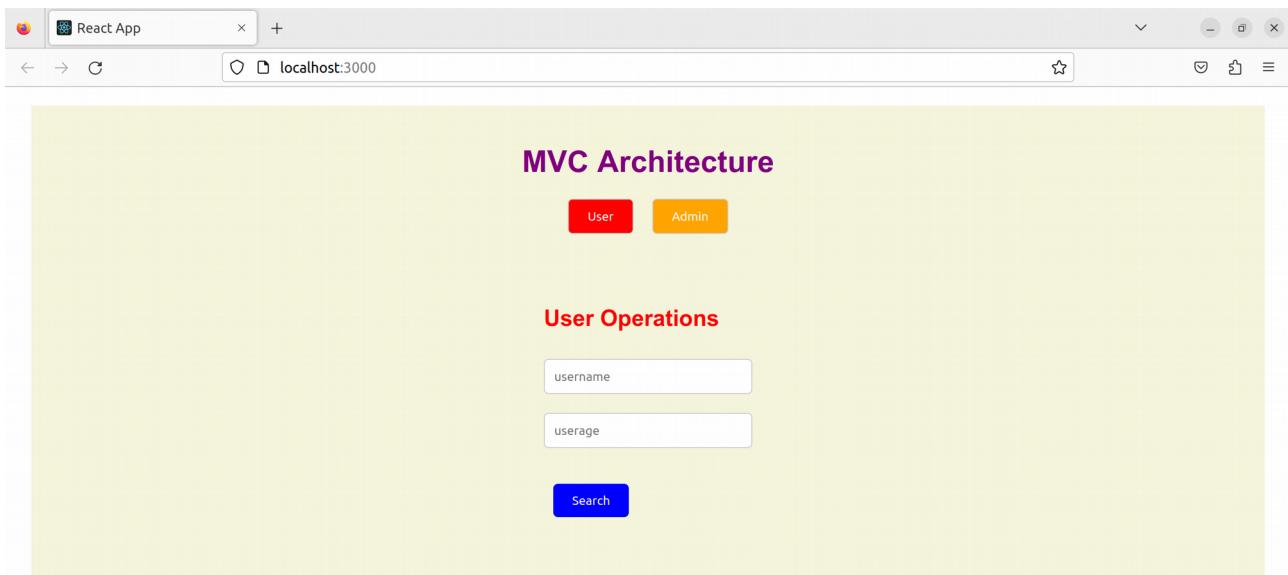
ul li {
    color:red;
    padding: 5px 0;
}
```

index.js

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

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

Output:



MVC Architecture

User Admin

Admin Operations

username

userage

Search

Insert

Delete

new name

new age

Update

Fetch All Users

All Users:

MVC Architecture

User Admin

Admin Operations

Rohith

21

Search

Insert

Delete

new name

new age

Update

Fetch All Users

Rohith aged 21 is Present

All Users:

MVC Architecture

User Admin

Admin Operations

Search
Insert
Delete

Update
Fetch All Users

Rohith aged 21 is Present

All Users:

-
Sam - 24
Nirmal - 22
Sarvesh Babu - 23
Rohith - 21

MVC Architecture

User Admin

User Operations

Search

Rohith aged 21 is Present