

CS381: FULL STACK DEVELOPMENT

PRACTICAL 01

AIM: Install & Configure Visual Studio Code as an IDE, setup an environment for React by downloading Node.js and install React framework along with its packages. Understand the folder structure used by React and develop a “Hello World Application”.

Source Code of App.js

```
import './App.css';

function App() {

  return (

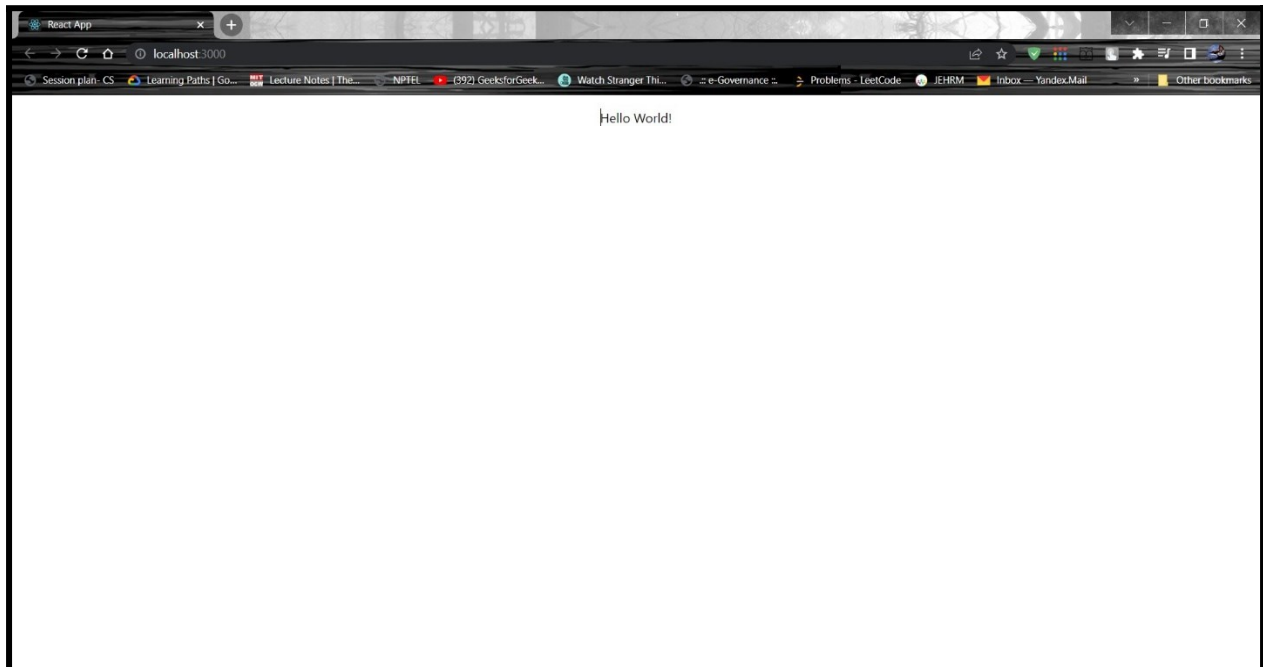
    <div className="App">

      <p>Hello World!</p>

    </div>

  );
}

export default App;
```

OUTPUT:**CONCLUSION:**

Through this practical we get to know the installation process of Node.js and React. We also learn about the basic folder structure of a React App, along with building a Hello World application in React.

PRACTICAL 02

AIM: Implement React Routing for navigating 5 different web pages. Design a responsive navbar in ReactJS by using media queries, Material-UI/React-Bootstrap.

Source Code of App.js

```
import React from 'react';

import './App.css';

import Navbar from './components/Navbar';

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

import Home from './pages';

import About from './pages/about';

import Services from './pages/services';

import Contact from './pages/contact';

import SignUp from './pages/signup';

function App() {

  return (

    <Router>

      <Navbar />

      <Routes>

        <Route path="/" exact component={Home} />

        <Route path="/about" component={About} />

        <Route path="/services" component={Services} />

        <Route path="/contact-us" component={Contact} />

        <Route path="/sign-up" component={SignUp} />

      </Routes>

    </Router>

  );
}
```

```
    </Routes>

    </Router>

  );
}

export default App;
```

Source Code of Home.js

```
import React from 'react';

const Home = () => {

  return (

    <div

      style={{

        display: 'flex',

        justifyContent: 'center',

        alignItems: 'center',

        height: '90vh'

      }}

    >

      <h1>Home</h1>

    </div>

  );

};

export default Home;
```

Source Code of About.js

```
import React from 'react';

const About = () => {

  return (

    <div

      style={{

        display: 'flex',

        justifyContent: 'center',

        alignItems: 'center',

        height: '90vh'

      }}

    >

    <h1>about</h1>

    </div>

  );

};

export default About;
```

Source Code of Contact.js

```
import React from 'react';

const Contact = () => {

  return (

    <div

      style={{
```

```
        display: 'flex',

        justifyContent: 'center',

        alignItems: 'center',

        height: '90vh'

    }}

  >

    <h1>Contact Us</h1>

  </div>

);

};

export default Contact;
```

Source Code of Service.js

```
import React from 'react';

const Services = () => {

  return (

    <div

      style={{

        display: 'flex',

        justifyContent: 'center',

        alignItems: 'center',

        height: '90vh'

      }}

    >
```

```
    <h1>Services</h1>

    </div>

  );
};

export default Services;
```

Source Code of Signup.js

```
import React from 'react';

const SignUp = () => {

  return (

    <div

      style={{

        display: 'flex',

        justifyContent: 'center',

        alignItems: 'center',

        height: '90vh'

      }}

    >

      <h1>Sign Up</h1>

    </div>

  );
};

export default SignUp;
```

```
import React from 'react';

import {
  Nav,
  NavLink,
  Bars,
  NavMenu,
  NavBtn,
  NavBtnLink
} from './NavbarElements';

const Navbar = () => {
  return (
    <>
      <Nav>
        <NavLink to='/'>
          /* <img src={require('../images/logo.svg')} alt='Logo' />
*/
          Home
        </NavLink>
        <Bars />
        <NavMenu>
          <NavLink to='/about' activeStyle>
            About
          </NavLink>
          <NavLink to='/services' activeStyle>
```



```
        Services

        </NavLink>

        <NavLink to='/contact-us' activeStyle>

            Contact Us

        </NavLink>

        <NavLink to='/sign-up' activeStyle>

            Sign Up

        </NavLink>

        { /* Second Nav */ }

        { /* <NavBtnLink to='/sign-in'>Sign In</NavBtnLink> */ }

    </NavMenu>

    <NavBtn>

        <NavBtnLink to='/signin'>Sign In</NavBtnLink>

    </NavBtn>

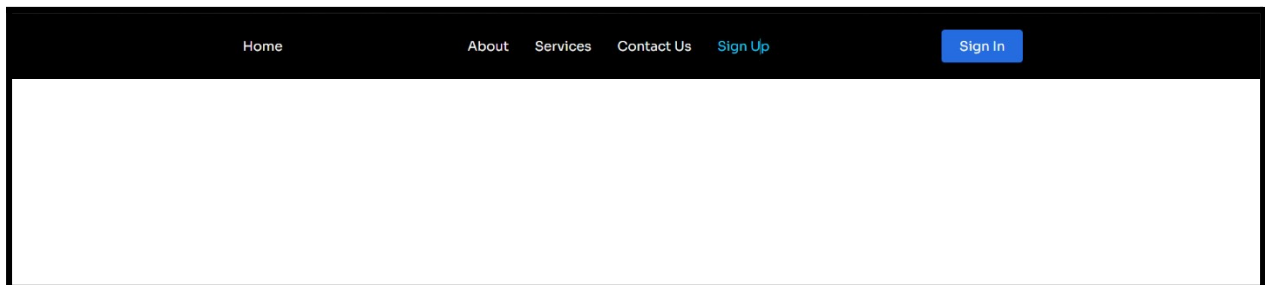
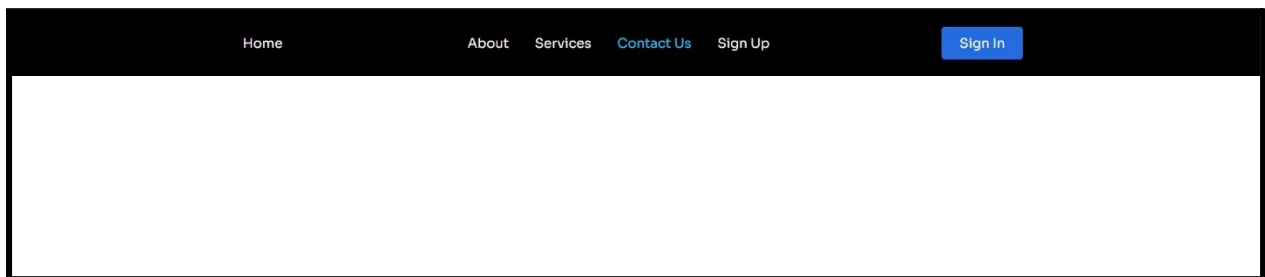
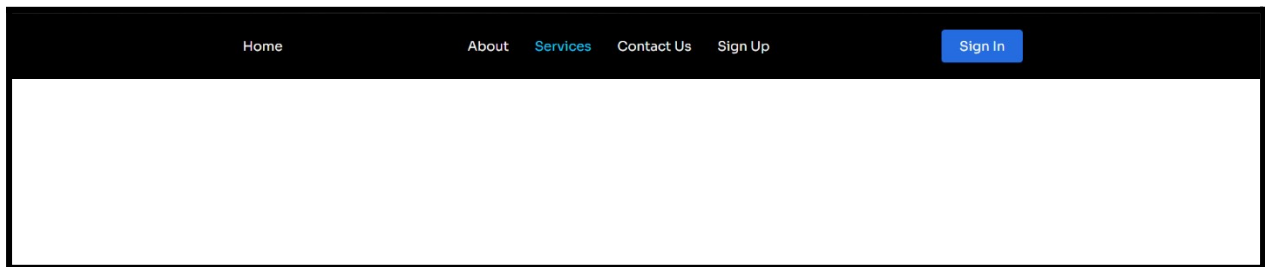
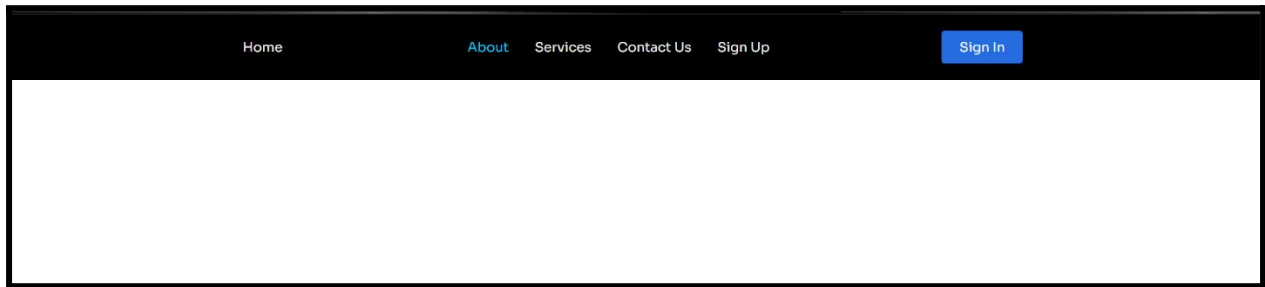
</Nav>

</>

);

};

export default Navbar;
```

OUTPUT:**CONCLUSION:**

Through this practical we learn about react routing.

PRACTICAL 03

AIM: Create React Sidebar Navigation Menu for a webpage. Implement the concept of React Hooks for backward compatibility and faster user experience.

Source Code of App.js

```
import Sidebar from './components/Sidebar'

function App() {

  return (

    <div className="main">

      <Sidebar />

      <div className="container">

        <h1 className="title">My React App</h1>

        <p className="info">Demo Application</p>

        <button className="btn">Explore now</button>

      </div>

    </div>

  )

}

export default App
```

Source Code of index.js

```
import React from 'react'

import ReactDOM from 'react-dom'

import './index.css'

import App from './App'
```

```
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
)
```

Source Code of Sidebar.js

```
import SidebarItem from "../SidebarItem"  
  
import items from "../data/sidebar.json"  
  
export default function Sidebar(){  
  return (  
    <div className="sidebar">  
      { items.map((item, index) => <SidebarItem key={index}  
item={item} />) }  
    </div>  
  )  
}
```

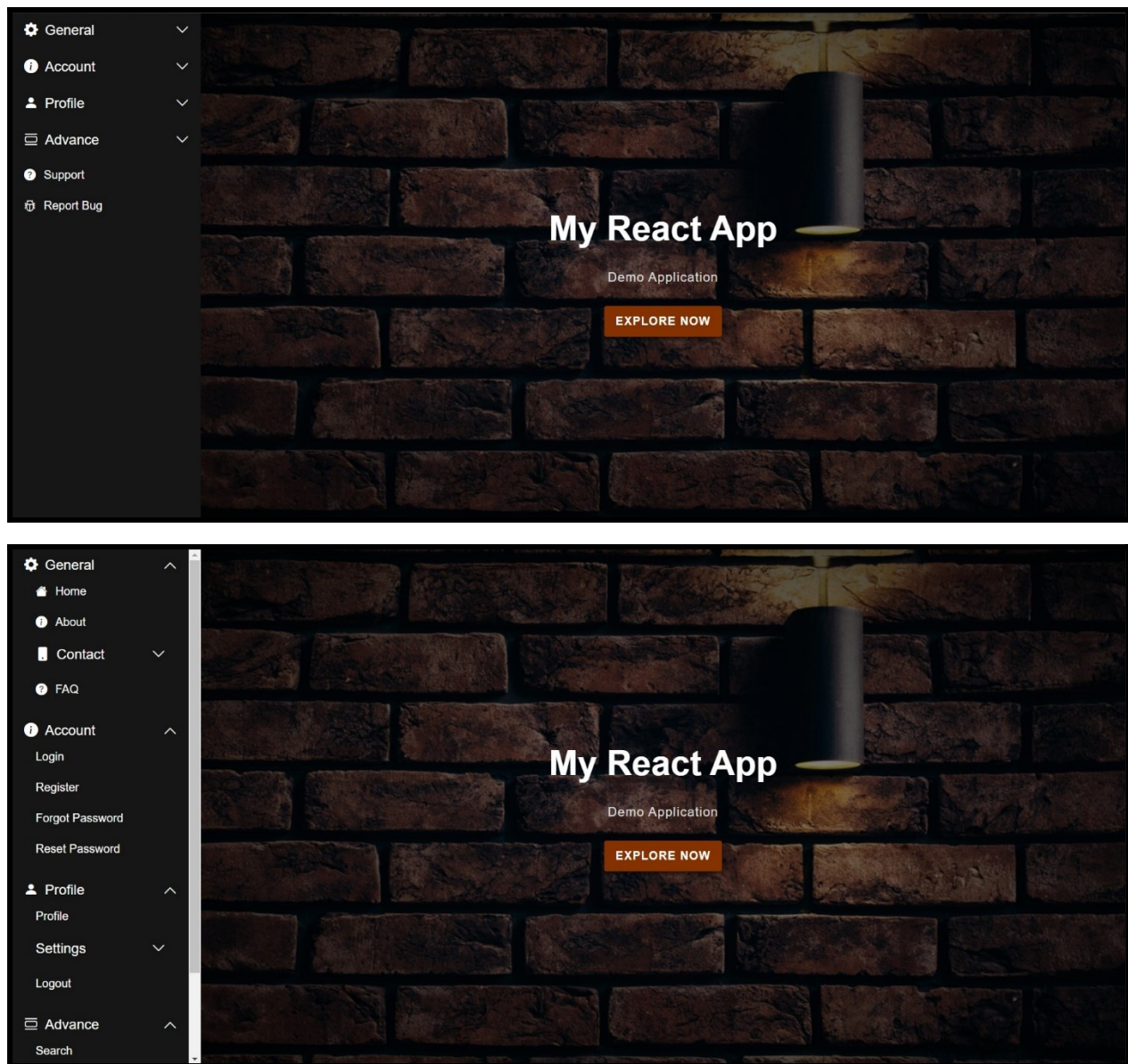
Source Code of SidebarItems.js

```
import { useState } from "react"  
  
export default function SidebarItem({item}){  
  const [open, setOpen] = useState(false)  
  
  if(item.childrens){
```

```

    return (
      <div className={open ? "sidebar-item open" : "sidebar-item"}>
        <div className="sidebar-title">
          <span>
            { item.icon && <i className={item.icon}></i> }
            {item.title}
          </span>
          <i className="bi-chevron-down toggle-btn" onClick={()
=> setOpen(!open)}></i>
        </div>
        <div className="sidebar-content">
          { item.childrens.map((child, index) => <SidebarItem
key={index} item={child} />) }
        </div>
      </div>
    )
  }else{
    return (
      <a href={item.path || "#"} className="sidebar-item plain">
        { item.icon && <i className={item.icon}></i> }
        {item.title}
      </a>
    )
  }
}

```

OUTPUT:**CONCLUSION:**

Through this practical we learn the concept of react hooks and implement it to create a side navigation bar using react.

PRACTICAL 04

AIM: Create simple Application of React Todo List. Implement the concepts of useState, useEffect, props, useRefs. Also use JavaScript functions for event Handling.

Source Code of app.js

```
import logo from './logo.svg';
import './App.css';
import TodoList from './components/TodoList'
import 'bootstrap/dist/css/bootstrap.min.css'
import 'font-awesome/css/font-awesome.min.css'
function App() {
  return (
    <div className="App">
      <TodoList />
    </div>
  );
}
export default App;
```

Source Code of index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
reportWebVitals();
```

Source Code of Card.js

```
import React, {useState} from 'react';
import EditTask from '../modals/EditTask'

const Card = ({taskObj, index, deleteTask, updateListArray}) => {
  const [modal, setModal] = useState(false);
  const colors = [
    { primaryColor : "#5D93E1",
      secondaryColor : "#ECF3FC" },
    { primaryColor : "#F9D288",
      secondaryColor : "#FEFAF1" },
    { primaryColor : "#5DC250",
      secondaryColor : "#F2FAF1" },
    { primaryColor : "#F48687",
      secondaryColor : "#FDF1F1" },
    { primaryColor : "#B964F7",
      secondaryColor : "#F3F0FD" }
  ]
  const toggle = () => {
    setModal(!modal); }
  const updateTask = (obj) => {
    updateListArray(obj, index) }
  const handleDelete = () => {
    deleteTask(index) }
  return (
    <div class = "card-wrapper mr-5">
      <div class = "card-top" style={{ "background-color":
colors[index%5].primaryColor }}></div>
      <div class = "task-holder">
        <span class = "card-header" style={{ "background-color":
colors[index%5].secondaryColor, "border-radius":
"10px" }}>{taskObj.Name}</span>
        <p className = "mt-3">{taskObj.Description}</p>
        <div style={{ "position": "absolute", "right" : "20px",
"bottom" : "20px" }}>
          <i class = "far fa-edit mr-3" style={{ "color" :
```



```

colors[index%5].primaryColor, "cursor" : "pointer"}} onClick = {() =>
setModal(true)}></i>
      <i class="fas fa-trash-alt" style = {"color" :
    colors[index%5].primaryColor, "cursor" : "pointer"}} onClick =
{handleDelete}></i>
    </div>
  </div>
  <EditTask modal = {modal} toggle = {toggle} updateTask =
{updateTask} taskObj = {taskObj}/>
</div>
);
};
export default Card;

```

Source Code of TodoList.js

```

import React, {useEffect, useState} from 'react';
import CreateTask from '../modals/CreateTask'
import Card from './Card';
const TodoList = () => {
  const [modal, setModal] = useState(false);
  const [taskList, setTaskList] = useState([])
  useEffect(() => {
    let arr = localStorage.getItem("taskList")
    if(arr){
      let obj = JSON.parse(arr)
      setTaskList(obj)
    }
  }, [])
  const deleteTask = (index) => {
    let tempList = taskList
    tempList.splice(index, 1)
    localStorage.setItem("taskList", JSON.stringify(tempList))
    setTaskList(tempList)
    window.location.reload()
  }
}

```

```

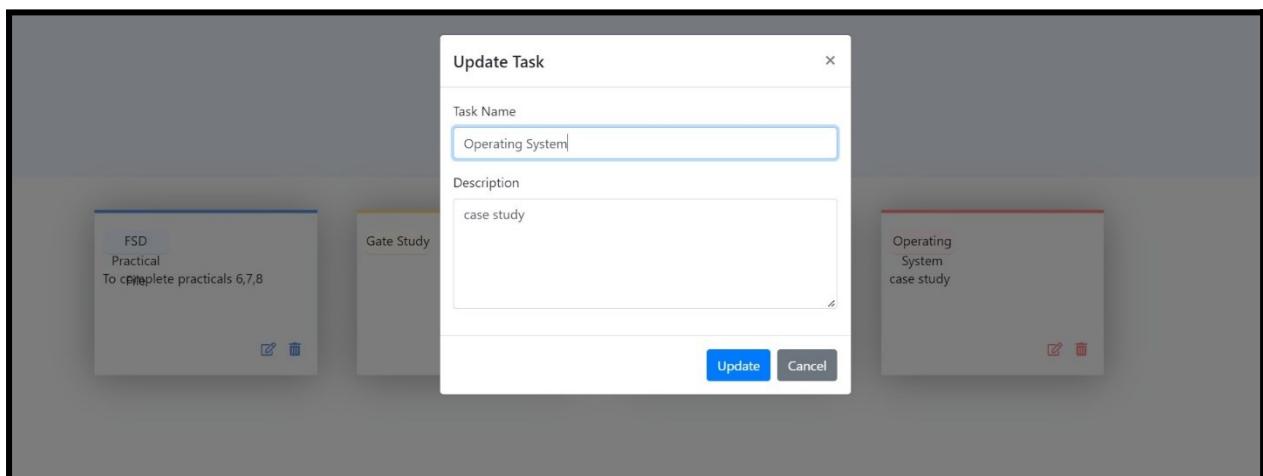
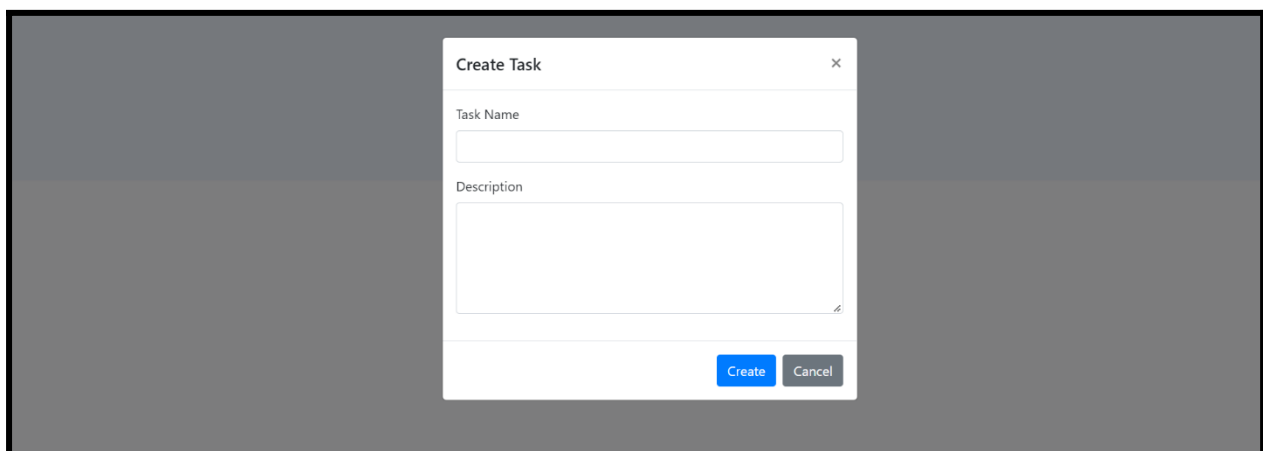
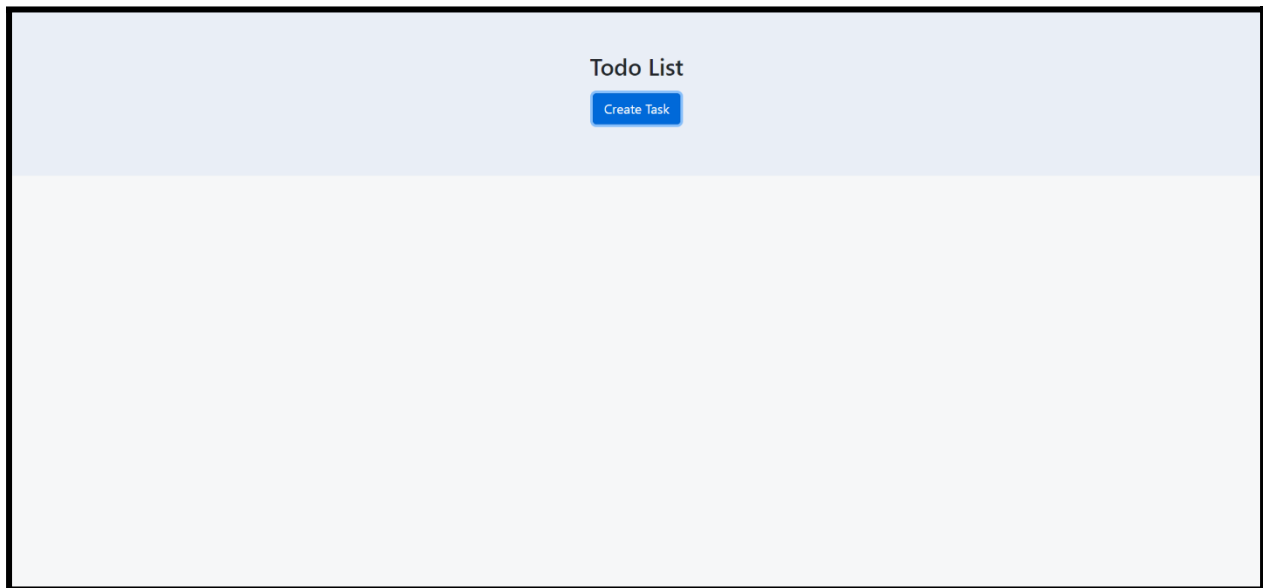
    const updateListArray = (obj, index) => {
      let tempList = taskList
      tempList[index] = obj
      localStorage.setItem("taskList", JSON.stringify(tempList))
      setTaskList(tempList)
      window.location.reload()
    }

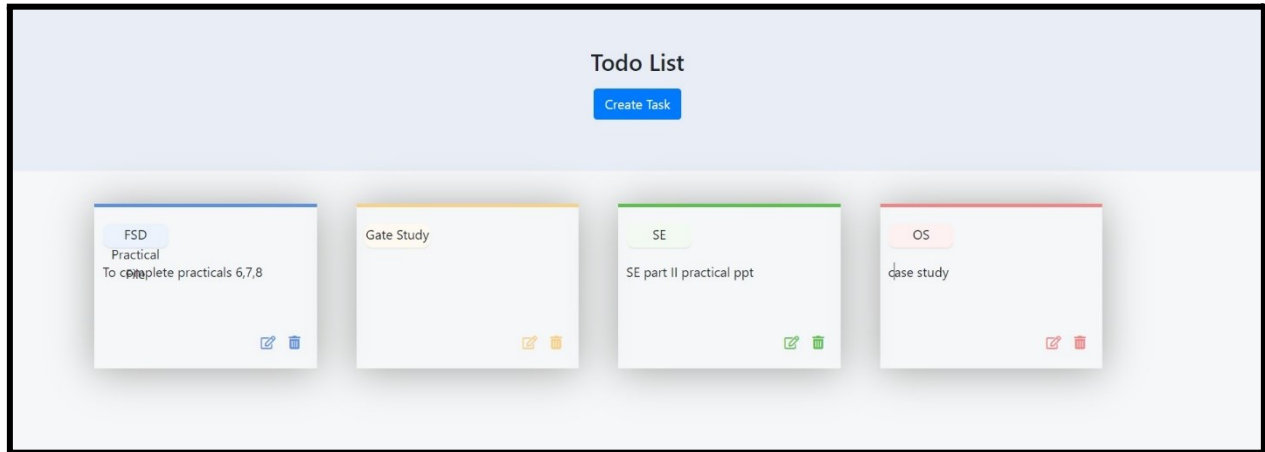
    const toggle = () => {
      setModal(!modal);
    }

    const saveTask = (taskObj) => {
      let tempList = taskList
      tempList.push(taskObj)
      localStorage.setItem("taskList", JSON.stringify(tempList))
      setTaskList(taskList)
      setModal(false)
    }

    return (
      <>
        <div className = "header text-center">
          <h3>Todo List</h3>
          <button className = "btn btn-primary mt-2" onClick = {()
=> setModal(true)} >Create Task</button>
        </div>
        <div className = "task-container">
          {taskList && taskList.map((obj , index) => <Card taskObj =
{obj} index = {index} deleteTask = {deleteTask} updateListArray =
{updateListArray}/> ) }
        </div>
        <CreateTask toggle = {toggle} modal = {modal} save =
{saveTask}/>
      </>
    );
  };
};
export default TodoList;

```

OUTPUT:



CONCLUSION:

Through the implementation of To-Do list we learn the concepts of `useState`, `useEffect`, `props` and `useRefs`.

PRACTICAL 05

AIM: Design an attractive Login and Registration form using Material-UI. Use validators to validate the errors and prompt the errors on the webpage.

Source Code of App.js

```
import './App.css';
import LoginPage from './components/LoginPage/LoginPage';
function App() {
  return (
    <div className="App">
      { <LoginPage/> }
    </div>  );}
export default App;
import './App.css';
import LoginPage from './components/LoginPage/LoginPage';
function App() {
  return (
    <div className="App">
      { <LoginPage/> }
    </div>  );}
export default App;
```

Source Code of LoginPage.js

```
import React, { Component } from 'react';
import './LoginPage.css'
import Grid from '@material-ui/core/Grid';
import insta_image from '../images/9364675fb26a.svg';
import insta_logo from '../images/logoinsta.png';
import fb from '../images/fb.png';
import appstore from '../images/app.png';
import playstore from '../images/play.png';
import SignIN from './SignIn/SignIn';
import SignUp from './SignUp/SignUp';
class LoginPage extends Component {
```

```

    constructor(props) {
      super(props);
      this.state = { isLogin: true }
    }
    changeLogin={() => {
      if (this.state.isLogin)
        this.setState({ isLogin: false });
      else
        this.setState({ isLogin: true });
    }}
    render() {
      return (
        <div>
          <Grid container>
            <Grid item xs={3}>
            </Grid>
            <Grid item xs={6}>
              <div className="loginpage__main">
                <div>
                  <img src={inst_image} width="454px" />
                </div>
                <div>
                  <div className="loginpage_rightcomponent">
                    <img className="loginpage__logo"
src={insta_logo} />
                    <div className="loginPage__signin">
                      {
                        this.state.isLogin ? <SignIN/> :
<SignUp/>
                      }
                    <div className="login__ordiv">
                      <div
className="login__dividor"></div>
                      <div
className="login__or">OR</div>
                      <div
className="login__dividor"></div>
                    </div>
                    <div className="login__fb">
                      <img src={fb} width="15px"
style={{ "marginRight": "5px" }} />Log in with Facebook

```

```

        <div className="login_forgt">
Forgot password?</div>
        </div>
    </div>
    <div className="loginpage__signuptoption">
        {
            this.state.isLogin ?
        <div
className="loginPage__signin">
            <span onClick={this.changeLogin} style={{ "fontWeight":"bold",
"color":"#0395F6"}}>Sign up</span>
            </div> :
        <div
className="loginPage__signup">
            <span onClick={this.changeLogin} style={{ "fontWeight":"bold",
"color":"#0395F6"}}>Sign in</span>
            </div>
        }
    </div>
    <div
className="loginPage__downloadSection">
        <div>
            Get the app.
        </div>
        <div className="loginPage__option">
            <img className="loginPage_dwimg"
src={appstore} width="136px" />
            <img className="loginPage_dwimg"
src={playstore} width="136px" />
        </div>
    </div>
</div>
</div>
</div>
</Grid>
<Grid item xs={3}>
</Grid>

```

```
);    }}
export default LoginPage;
```

Source Code of SignIn.js

```
import React, { Component } from 'react';
import "../LoginPage/LoginPage.css"
class SignIN extends Component {
  constructor(props) {
    super(props);
    this.state = {
      emailId : null,
      password: null    }
  }
  render() {    return (
    <div>
      <input className="logipage_text"
onChange={ (event)=>{this.state.emailId=event.currentTarget.value}}
type="text" placeholder="Phone number, username, or email" />
      <input className="logipage_text"
onChange={ (event)=>{this.state.password=event.currentTarget.value}}
type="password" placeholder="Password" />
      <button className="login__button" onClick={this.login}>Log
In</button>
    </div>    );    }}
export default SignIN;
```

Source Code of SignUp.js

```
import React, { Component } from 'react';
class SignUp extends Component {
  constructor(props) {
    super(props);
    this.state = {
      emailId : null,
      name: null,
      userName: null,
      password: null    }
  }
  render() {    return (
    <div>
```



```

    <input className="logipage_text"
    onChange={ (event)=>{this.state.emailId=event.currentTarget.value;}}
    type="text" placeholder="Mobile number or Email" />

    <input className="logipage_text"
    onChange={ (event)=>{this.state.name=event.currentTarget.value;}}
    type="text" placeholder="Full Name" />

    <input className="logipage_text"
    onChange={ (event)=>{this.state.userName=event.currentTarget.value;}}
    type="text" placeholder="Username" />

    <input className="logipage_text"
    onChange={ (event)=>{this.state.password=event.currentTarget.value;}}
    type="password" placeholder="Password" />

    <button className="login_button" onClick={this.newSignUp}
    >Sign up</button>

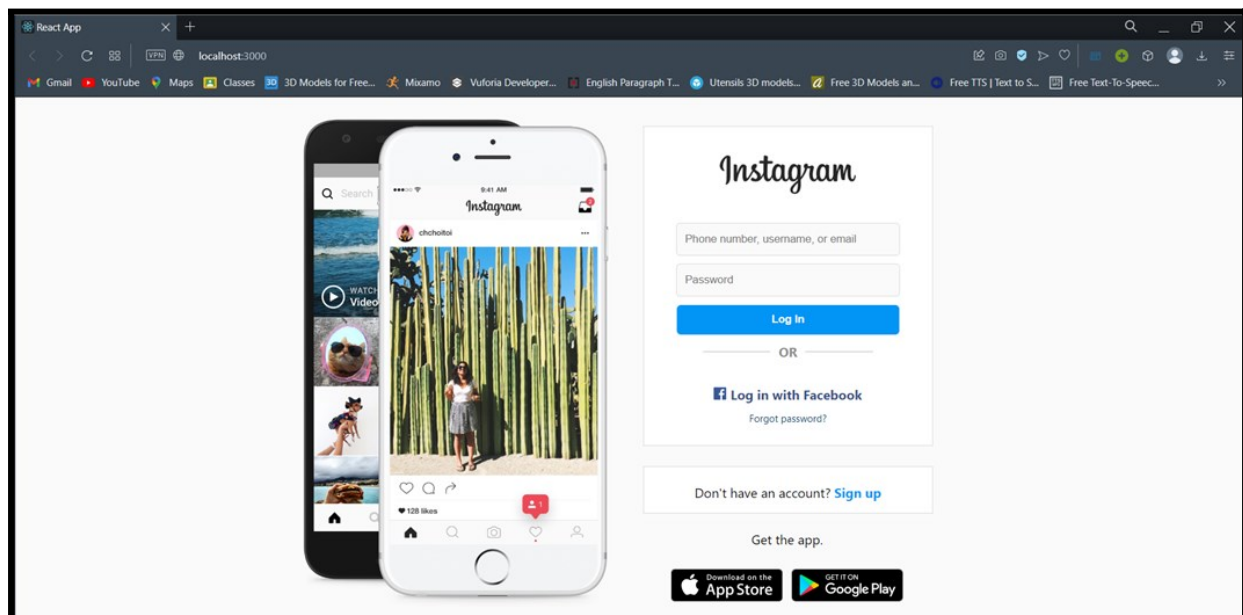
  </div>

  );  }}

export default SignUp;

```

OUTPUT:



CONCLUSION:

Through this practical we learn about Material UI and how to use validators to validate the errors and prompt the errors on the webpage

PRACTICAL 06

AIM: Perform Create, Read, Update and Delete Operations using Node.js, Express server and MongoDB. Use React Framework for developing your user interface. Use Rest APIs, CORS for client-server communication.

FRONTEND:

Source Code of Client/App.js

```
import logo from './logo.svg';
import './App.css';
import NavBar from './components/NavBar';
import AllUsers from './components/AllUsers';
import CRUD from './components/CRUD';
import AddUser from './components/AddUser';
import EditUser from './components/EditUser';
import {BrowserRouter, Routes, Route} from 'react-router-dom';
import "../node_modules/bootstrap/dist/css/bootstrap.min.css";
function App() {
  return (
    <BrowserRouter>
      <NavBar/>
      <Routes>
        <Route path = "/" element = {<CRUD/>} />
        <Route path = "/all" element = {<AllUsers/>} />
        <Route path = "/add" element = {<AddUser/>} />
        <Route path = "/edit/:id" element = {<EditUser/>} />
      </Routes>
    </BrowserRouter>
  );
}
export default App;
```

Source Code of Client/Service/Api.js

```
import axios from 'axios';
const URL = 'http://localhost:8000';
export const addUser = async (data) => {
  try {
    return await axios.post(`${URL}/add`, data);
  } catch (error) {
    console.log('Error while calling add User Api', error);
  }
}
export const getUsers = async () => {
  try {
    return await axios.get(`${URL}/all`);
  } catch (error) {
    console.log('Error while calling getUsers API', error);
  }
}
export const getUser = async (id) => {
  try {
    return await axios.get(`${URL}/${id}`);
  } catch (error) {
    console.log('Error while calling getUser api', error);
  }
}
export const editUser = async (user, id) => {
  try {
    return await axios.post(`${URL}/${id}`, user);
  } catch (error) {
    console.log("Error while calling editUser API", error);
  }
}
export const deleteUser = async (id) => {
  try {
    return await axios.delete(`${URL}/${id}`);
  } catch (error) {
    console.log("Error while calling deleteUser API", error);
  }
}
```

Source Code of Client/Components/AddUser.jsx

```
import { useState } from "react";
import { FormControl, FormGroup, InputLabel, Input, Typography, styled, Button } from "@mui/material";
import { addUser } from '../service/api';
import { useNavigate } from 'react-router-dom';
const Container = styled(FormGroup) `
  width: 50%;
  margin: 5% auto 0 auto;
  & > div { margin-top: 20px;}`
const defaultValue = {
  name: '',
  username: '',
  email: '',
  phone: ''
}
const AddUser = () => {
  const [user, setUser] = useState(defaultValue);
  const navigate = useNavigate();
  const onValueChange = (e) => {
    console.log(e.target.name, e.target.value)
    setUser({ ...user, [e.target.name]: e.target.value })
    console.log(user);
  }
  const addUserDetails = async () => {
    await addUser(user);
    navigate('/all'); }
  return (
    <Container>
      <Typography variant="h2"> Add User</Typography>
      <FormControl>
        <InputLabel>Name</InputLabel>
        <Input onChange={ (e) => onValueChange(e) } name="name"/>
      </FormControl>
      <FormControl>
```

```

        <InputLabel>Username</InputLabel>
        <Input onChange={ (e) => onValueChange(e) }
name="username"/>
      </FormControl>
      <FormControl>
        <InputLabel>Email</InputLabel>
        <Input onChange={ (e) => onValueChange(e) } name="email"/>
      </FormControl>
      <FormControl>
        <InputLabel>Phone</InputLabel>
        <Input onChange={ (e) => onValueChange(e) } name="phone"/>
      </FormControl>
      <FormControl>
        <Button variant="contained" onClick= { () =>
addUserDetails() }>Add User</Button>
      </FormControl>
    </Container>    )
  }
export default AddUser;

```

Source Code of Client/Components/AllUsers.jsx

```

import { useState } from "react";
import { FormControl, FormGroup, InputLabel, Input, Typography, styled,
Button } from "@mui/material";
import { addUser } from '../service/api';
import { useNavigate } from 'react-router-dom';
const Container = styled(FormGroup) `
  width: 50%;
  margin: 5% auto 0 auto;
  & > div { margin-top: 20px; } `
const defaultvalue = {
  name: '',
  username: '',
  email: '',
  phone: ''
}
const AddUser = () => {

```

```

const [user, setUser] = useState(defaultValue);
const navigate = useNavigate();
const onValueChange = (e) => {
  console.log(e.target.name, e.target.value)
  setUser({ ...user, [e.target.name]: e.target.value })
  console.log(user);
}
const addUserDetails = async () => {
  await addUser(user);
  navigate('/all');
}
return (
  <Container>
    <Typography variant="h2"> Add User</Typography>
    <FormControl>
      <InputLabel>Name</InputLabel>
      <Input onChange={ (e) => onValueChange(e) } name="name"/>
    </FormControl>
    <FormControl>
      <InputLabel>Username</InputLabel>
      <Input onChange={ (e) => onValueChange(e) }
name="username"/>
    </FormControl>
    <FormControl>
      <InputLabel>Email</InputLabel>
      <Input onChange={ (e) => onValueChange(e) } name="email"/>
    </FormControl>
    <FormControl>
      <InputLabel>Phone</InputLabel>
      <Input onChange={ (e) => onValueChange(e) } name="phone"/>
    </FormControl>
    <FormControl>
      <Button variant="contained" onClick= { () =>
addUserDetails() }>Add User</Button>
    </FormControl>
  </Container>
) }
export default AddUser;

```

Source Code of Client/Components/EditUsers.jsx

```

import { useState } from "react";
import { FormControl, FormGroup, InputLabel, Input, Typography, styled, Button } from "@mui/material";

import { addUser } from '../service/api';
import { useNavigate } from 'react-router-dom';

const Container = styled(FormGroup) `
  width: 50%;
  margin: 5% auto 0 auto;
  & > div { margin-top: 20px; } `

const defaultValue = {
  name: '',
  username: '',
  email: '',
  phone: ''
}

const AddUser = () => {
  const [user, setUser] = useState(defaultValue);
  const navigate = useNavigate();
  const onValueChange = (e) => {
    console.log(e.target.name, e.target.value)
    setUser({ ...user, [e.target.name]: e.target.value })
    console.log(user);
  }

  const addUserDetails = async () => {
    await addUser(user);
    navigate('/all');
  }

  return (
    <Container>
      <Typography variant="h2"> Add User</Typography>
      <FormControl>
        <InputLabel>Name</InputLabel>
        <Input onChange={(e) => onValueChange(e)} name="name"/>
      </FormControl>
      <FormControl>
        <InputLabel>Username</InputLabel>
        <Input onChange={(e) => onValueChange(e)}
name="username"/>

```

```

        </FormControl>
        <FormControl>
            <InputLabel>Email</InputLabel>
            <Input onChange={ (e) => onValueChange(e) } name="email"/>
        </FormControl>
        <FormControl>
            <InputLabel>Phone</InputLabel>
            <Input onChange={ (e) => onValueChange(e) } name="phone"/>
        </FormControl>
        <FormControl>
            <Button variant="contained" onClick= { () =>
addUserDetails() }>Add User</Button>
        </FormControl>
    </Container>
  )}
export default AddUser;

```

Source Code of Client/Components/Navbar.jsx

```

import {AppBar, Toolbar, styled} from '@mui/material';
import { NavLink } from 'react-router-dom';
const Header = styled(AppBar) `
  background: #111111
`;
const Tabs = styled(NavLink) `
  font-size: 20px;
  margin-right: 20px;
  color: inherit;
  text-decoration: none;`
const NavBar = () => {
  return(
    <Header position="static">
      <Toolbar>
        <Tabs to="/">CRUD</Tabs>
        <Tabs to="/all">All Users</Tabs>
        <Tabs to="/add">Add User</Tabs>

```



```
        </Toolbar>
      </Header>
    })
  export default NavBar;
```

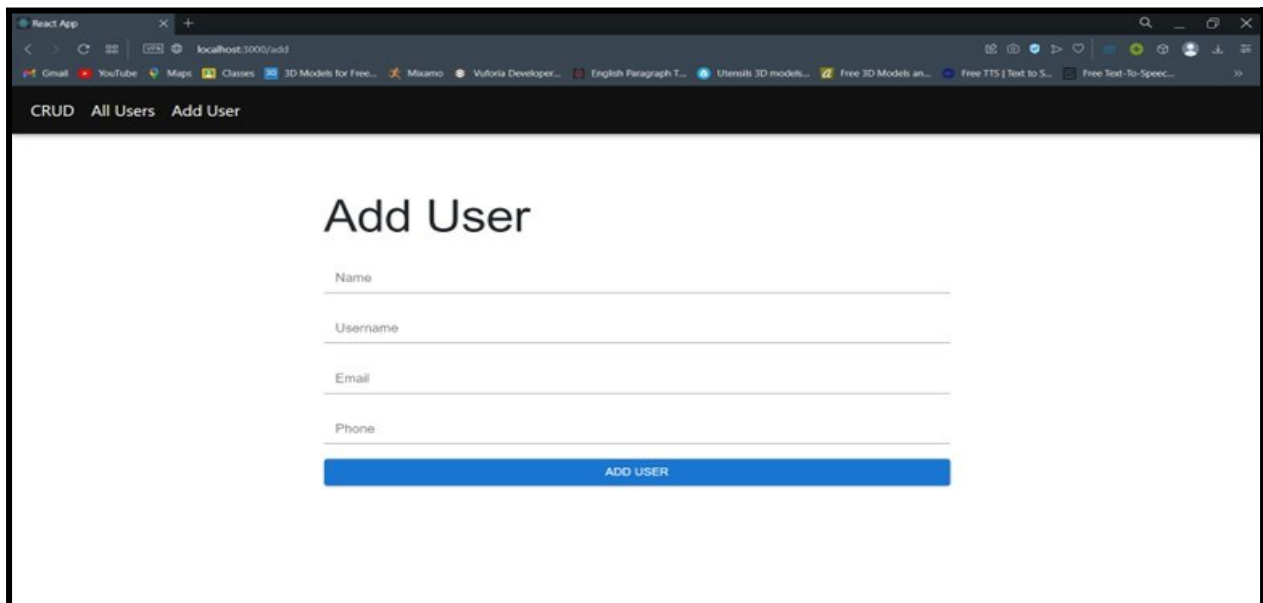
BACKEND:

Source Code for Server/controller/user-controller.js

```
import user from '../schema/user-schema.js';
import User from '../schema/user-schema.js'
export const addUser = async (request, response) => {
  const user = request.body;
  const newUser = new User(user);
  try{
    await newUser.save();
    response.status(201).json(newUser);
  } catch (error) {
    response.status(409).json({message: error.message });
  }
}
export const getUsers = async (request, response) => {
  try {
    const users = await User.find({});
    response.status(200).json(users);
  } catch (error) {
    response.status(404).json({ message: error.message });
  }
}
export const getUser = async (request, response) => {
  try {
    const user = await User.findById(request.params.id);
    response.status(200).json(user);
  } catch (error) {
    response.status(404).json({ message: error.message });
  }
}
export const editUser = async (request, response) => {
```

```
let user = request.body;
const editUser = new User(user);
try {
  await User.updateOne({ _id: request.params.id }, editUser);
  response.status(201).json(editUser);
} catch (error) {
  response.status(409).json({ message: error.message });
}}
export const deleteUser = async (request, response) => {
  try {
    await User.deleteOne({ _id: request.params.id });
    response.status(200).json({ message: "User deleted uccessfully" });
  } catch (error) {
    response.status(409).json({ message: error.message });
  }}
}
```

OUTPUT:



CONCLUSION:

Through this practice we learn to implement CRUD operations using MERN stack. We also learn to use Rest APIs, CORS for client-server communication.

PRACTICAL 07

AIM: MERN stands for MongoDB, Express, React, Node and these four things added to make the stack. Create a project with this stack and implement a Register and Login system for your project.

Source Code of Register.js

```
import React , {useState} from 'react'
import axios from "axios";
const Register = () => {
  const [user,setUser] = useState({
    name:"",
    email:"",
    password: ""
  })
  const handleChange = e =>{
    const {name,value} = e.target
    setUser({
      ...user,//spread operator
      [name]:value
    })
  }
  const egister = ()=>{ //register function
    const {name,email,password} = user
    if (name && email && password){
      axios.post("http://localhost:6969/Register",user )
        .then(res=>console.log(res))
    }
    else{
      alert("invalid input")
    }
  };
  return (
    <>
    Create a new account
```

```

    </div>
    Already have an account ?
    <a href="#" target="_blank" class="text-sm text-blue-500 underline
hover:text-blue-700">
    Sign in
    </a>
  </span>
  <div class="p-6 mt-8">
    <form action="#">
      <div class="flex flex-col mb-2">
        <div class="relative">
          </div>
        </div>
        <div class="flex gap-4 mb-2">
          <div class="relative">
            </div>
            </div>
            <div class="flex flex-col mb-2">
              Register
            </div>
          </div>
        </div>
      </div>
    </form>
  </div>
}
export default Register

```

Source Code of Login.js

```

import React,{useState} from 'react'
import axios from 'axios';
import {useHistory} from "react-router-dom"
const Login = ({setLoginUser}) => {
const history = useHistory()
  const [user,setUser] = useState({
    name:"",
    password: ""
  })
  const handleChange = e =>{
    const {name,value} = e.target
    setUser({
      ...user,//spread operator

```

```

    [name]:value
  })
}

const login =()=>{
  axios.post("http://localhost:6969/Login",user)
    .then(res=>{alert(res.data.message)
    setLoginUser(res.data.user)
    history.push("/")})
}

return (
  <>
<div class="flex flex-col w-full max-w-md px-4 py-8 bg-white rounded-lg
shadow dark:bg-gray-800 sm:px-6 md:px-8 lg:px-10">

  <div class="self-center mb-6 text-xl font-light text-gray-600
sm:text-2xl dark:text-white">

    Login To Your Account

  </div>

  <div class="mt-8">

    <form action="#" autoComplete="off">

      <div class="flex flex-col mb-2">

        <div class="flex relative ">

          <span class="rounded-l-md inline-flex items-center
px-3 border-t bg-white border-l border-b border-gray-300 text-gray-500
shadow-sm text-sm">

            <svg width="15" height="15" fill="currentColor"
viewBox="0 0 1792 1792" xmlns="http://www.w3.org/2000/svg">

              <path d="M1792 710v794q0 66-47 113t-113
47h-1472q-66 0-113-47t-47-113v-794q44 49 101 87 362 246 497 345 57 42 92.5
65.5t94.5 48 110 24.5h2q51 0 110-24.5t94.5-48 92.5-65.5q170-123 498-345
57-39 100-87zm0-294q0 79-49 151t-122 123q-376 261-468 325-10 7-42.5
30.5t-54 38-52 32.5-57.5 27-50 9h-2q-23
0-50-9t-57.5-27-52-32.5-54-38-42.5-30.5q-91-64-262-182.5t-205-142.5q-62-42
-117-115.5t-55-136.5q0-78 41.5-130t118.5-52h1472q65 0 112.5 47t47.5 113z">

            </path>

          </svg>

        </span>

        <input type="text" id="sign-in-email" class="
rounded-r-lg flex-1 appearance-none border border-gray-300 w-full py-2
px-4 bg-white text-gray-700 placeholder-gray-400 shadow-sm text-base
focus:outline-none focus:ring-2 focus:ring-purple-600
focus:border-transparent" name="email" value={user.email}
onChange={handleChange} placeholder="Your email"/>

      </div>

    </div>

  </div>

```

```

<div class="flex flex-col mb-6">
  <div class="flex relative ">
    <span class="rounded-l-md inline-flex
items-center px-3 border-t bg-white border-l border-b border-gray-300
text-gray-500 shadow-sm text-sm">
      <svg width="15" height="15"
fill="currentColor" viewBox="0 0 1792 1792"
xmlns="http://www.w3.org/2000/svg">
        <path d="M1376 768q40 0 68 28t28 68v576q0
40-28 68t-68 28h-960q-40 0-68-28t-28-68v-576q0-40
28-68t68-28h32v-320q0-185 131.5-316.5t316.5-131.5 316.5
316.5q0 26-19 45t-45 19h-64q-26 0-45-19t-19-45q0-106-75-181t-181-75-181
75-75 181v320h736z">
      </path>
    </svg>
  </span>
  <input type="password" id="sign-in-email" class="
rounded-r-lg flex-1 appearance-none border border-gray-300 w-full py-2
px-4 bg-white text-gray-700 placeholder-gray-400 shadow-sm text-base
focus:outline-none focus:ring-2 focus:ring-purple-600
focus:border-transparent" name="password" value={user.password}
onChange={handleChange} placeholder="Your password"/>
</div>
</div>
<div class="flex items-center mb-6 -mt-4">
  <div class="flex ml-auto">
    <a href="#" class="inline-flex text-xs
font-thin text-gray-500 sm:text-sm dark:text-gray-100 hover:text-gray-700
dark:hover:text-white">
      Forgot Your Password?
    </a>
  </div>
</div>
<div class="flex w-full">
  <button type="submit" class="py-2 px-4
bg-purple-600 hover:bg-purple-700 focus:ring-purple-500
focus:ring-offset-purple-200 text-white w-full transition ease-in
duration-200 text-center text-base font-semibold shadow-md
focus:outline-none focus:ring-2 focus:ring-offset-2 rounded-lg "
onClick={login}>
    Login
  </button>
</div>
</form>
</div>
<div class="flex items-center justify-center mt-6">

```

```

    <a href="#" target=" blank" class="inline-flex
items-center text-xs font-thin text-center text-gray-500
hover:text-gray-700 dark:text-gray-100 dark:hover:text-white"
onClick={history.push("/Register")}>

        <span class="ml-2">

            You don't have an account?

        </span>

    </a>

</div>

</div>

</>

)

}

export default Login

import React from 'react'

const Homepage = () => {

    return (

        <>

            <h1>Welcome to Homepage which is only visible when you are
logged in </h1>

            </>

        )

    )

}

export default Homepage

```

Source Code of App.js

```

import './App.css';

import Homepage from "../components/homepage/Homepage"
import Login from "../components/login/Login"
import Register from "../components/register/Register"

import {

    BrowserRouter as Router,

    Switch,

    Route

} from "react-router-dom";

import {useState} from 'react';

function App() {

```

```

    const [user, setLoginUser] = useState({
    })
    return (
      <div className="App">
        <Router>
<Switch>
      <Route exact path="/">
        {
          user && user._id ? <Homepage/>:<Login/>
        }<Homepage/></Route>
      <Route path="/Login"><Login setLoginUser={setLoginUser}/></Route>
      <Route path="/Register"><Register/></Route>
</Switch>
        </Router>
      </div>
    );
  }
}
export default App;

```

Source Code of Index.js

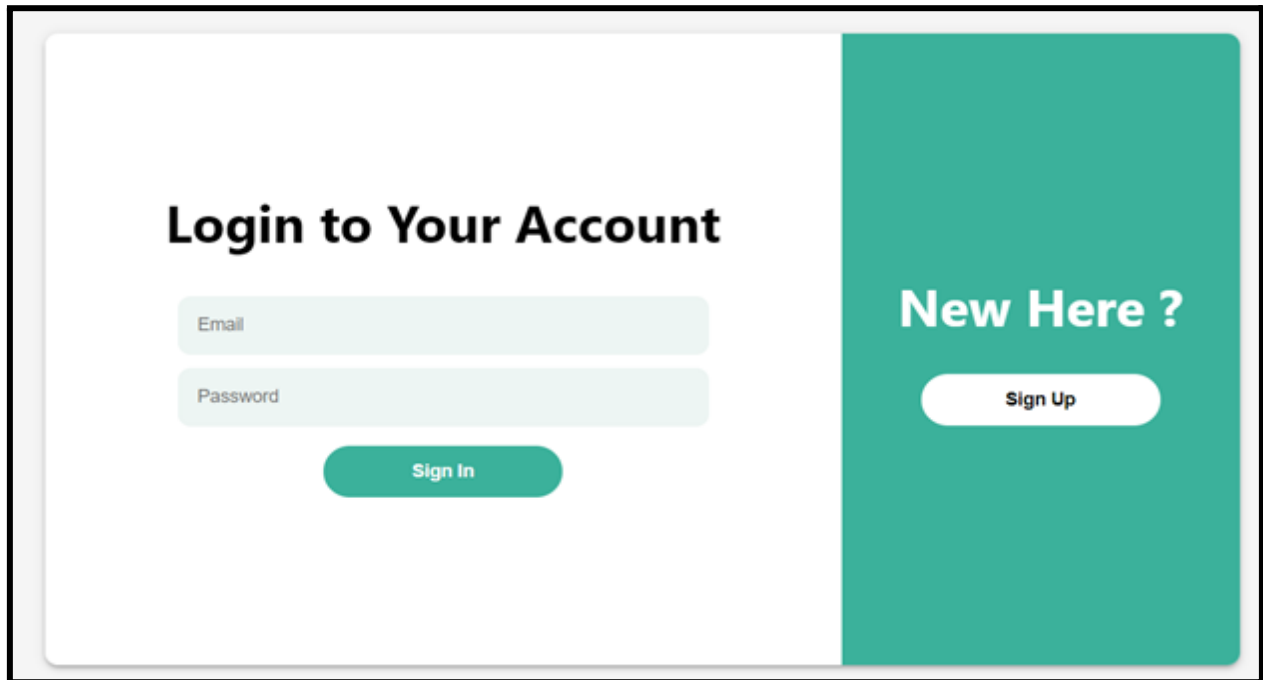
```

import express from "express";
import cors from "cors";
import mongoose from "mongoose";
const app = express();
app.use(express.json());
app.use(express.urlencoded());
app.use(cors());
mongoose.connect("mongodb://localhost:27017/auth",{
  useNewUrlParser:true,
  useUnifiedTopology:true
});()=>{
  console.log("connected to DB")
}
const userSchema = new mongoose.Schema({ //user schema
  name: String,

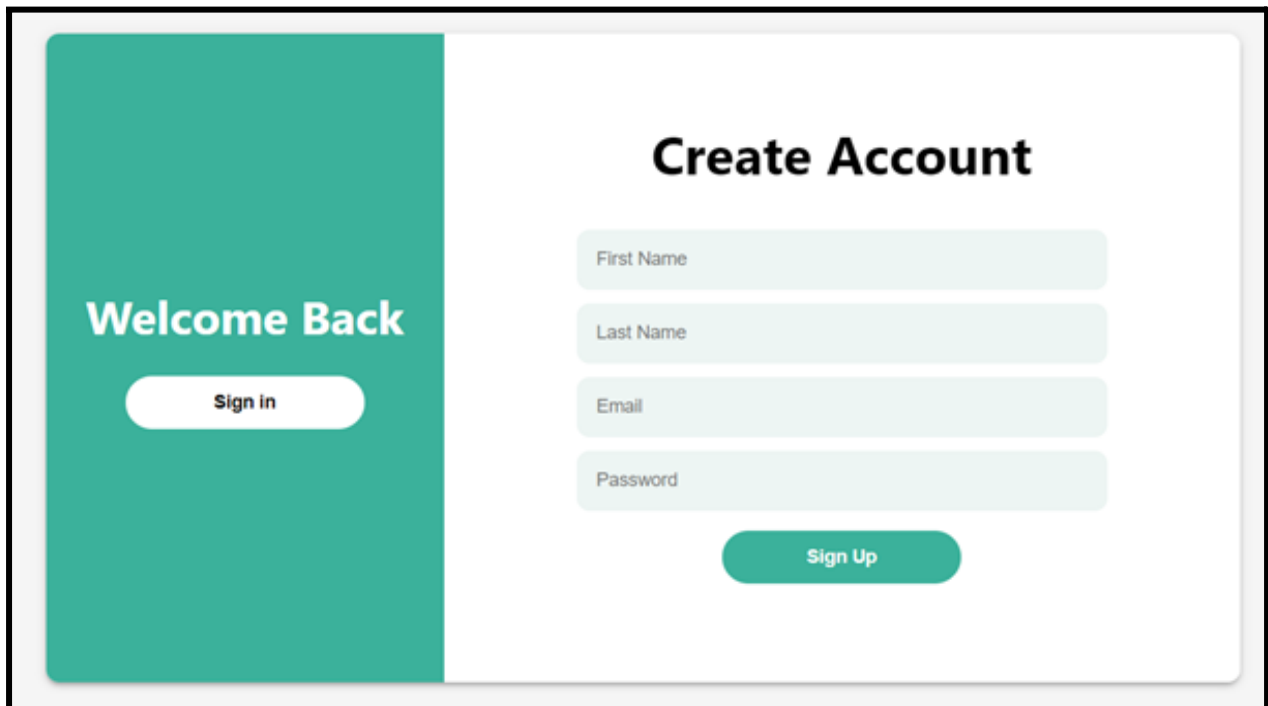
```



```
    email: String,
    password: String}))
const User = new mongoose.model("User", userSchema)
app.post("/Login", (req, res) => { //routes routes
    const {email, password} = req.body;
    User.findOne({email: email}, (err, user) => {
        if (user) {
            if (password === user.password) {
                res.send({message: "login success", user: user})
            } else {
                res.send({message: "wrong credentials"})
            }
        } else {
            res.send("not register")
        }
    })
})
app.post("/Register", (req, res) => {
    console.log(req.body)
    const {name, email, password} = req.body;
    User.findOne({email: email}, (err, user) => {
        if (user) {
            res.send({message: "user already exist"})
        } else {
            const user = new User({name, email, password})
            user.save(err => {
                if (err) {
                    res.send(err)
                } else {
                    res.send({message: "successful"})
                }
            })
        }
    })
})
app.listen(6969, () => {
    console.log("started")
})
```

OUTPUT:

The image shows a login form titled "Login to Your Account". It features two input fields: "Email" and "Password", both with light blue borders and placeholder text. Below these fields is a teal "Sign In" button. To the right of the form is a teal sidebar with the text "New Here ?" in white, and a white "Sign Up" button below it.



The image shows a registration form titled "Create Account". It features four input fields: "First Name", "Last Name", "Email", and "Password", all with light blue borders and placeholder text. Below these fields is a teal "Sign Up" button. To the left of the form is a teal sidebar with the text "Welcome Back" in white, and a white "Sign in" button below it.

CONCLUSION:

In this practical we implement a Register and Login system using MERN Stack

PRACTICAL 08

AIM: Build an exact similar clone of any video streaming web application like Netflix. Your web application should contain following things:

1. Get all the Movies/TV Shows on the Home page through TMDB API Key.
2. Build the animated rows to display the contents cards.
3. Build the Navbar.
4. Build the Banner.
5. Add popups to show the trailer of Movies/TV Shows when clicked on it.

Finally, host your web application on Google Firebase and push the code to your GitHub Repository to enhance your GitHub Profile.

Source Code of netflix-clone/components/BackgroundImage.jsx

```
import React from "react";
import styled from "styled-components";
import background from "../assets/login.jpg";
export default function BackgroundImage() {
  return (
    <Container>
      <img src={background} alt="background" />
    </Container>
  );
}
const Container = styled.div`
  height: 100vh;
  width: 100vw;
  img {
    height: 100vh;
    width: 100vw;
  }
`;
```

Source Code of netflix-clone/components/Card.jsx

```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import styled from "styled-components";
import { IoPlayCircleSharp } from "react-icons/io5";
import { AiOutlinePlus } from "react-icons/ai";
import { RiThumbUpFill, RiThumbDownFill } from "react-icons/ri";
import { BiChevronDown } from "react-icons/bi";
import { BsCheck } from "react-icons/bs";
import axios from "axios";
import { onAuthStateChanged } from "firebase/auth";
import { firebaseAuth } from "../utils/firebase-config";
import { useDispatch } from "react-redux";
import { removeMovieFromLiked } from "../store";
import video from "../assets/video.mkv";
export default React.memo(function Card({ index, movieData, isLiked =
false }) {
  const navigate = useNavigate();
  const dispatch = useDispatch();
  const [isHovered, setIsHovered] = useState(false);
  const [email, setEmail] = useState(undefined);
  onAuthStateChanged(firebaseAuth, (currentUser) => {
    if (currentUser) {
      setEmail(currentUser.email);
    } else navigate("/login");
  });
  const addToList = async () => {
    try {
      await axios.post("http://localhost:5000/api/user/add", {
        email,
        data: movieData,
      });
    } catch (error) {
      console.log(error);
    }
  };
  return (
```

```
<Container
  onMouseEnter={() => setIsHovered(true)}
  onMouseLeave={() => setIsHovered(false)}
>
  <img
    src={`https://image.tmdb.org/t/p/w500${movieData.image}`}
    alt="card"
    onClick={() => navigate("/player")}
  />
  {isHovered && (
    <div className="hover">
      <div className="image-video-container">
        <img
          src={`https://image.tmdb.org/t/p/w500${movieData.image}`}
          alt="card"
          onClick={() => navigate("/player")}
        />
        <video
          src={video}
          autoPlay={true}
          loop
          muted
          onClick={() => navigate("/player")}
        />
      </div>
      <div className="info-container flex column">
        <h3 className="name" onClick={() => navigate("/player")}>
          {movieData.name}
        </h3>
        <div className="icons flex j-between">
          <div className="controls flex">
            <IoPlayCircleSharp
              title="Play"
              onClick={() => navigate("/player")}
            />
          </div>
        </div>
      </div>
    </div>
  )}
</Container>
```

```

        <RiThumbDownFill title="Dislike" />
        {isLiked ? (
          <BsCheck
            title="Remove from List"
            onClick={() =>
              dispatch(
                removeMovieFromLiked({ movieId: movieData.id,
email })
              )
            }
          />
        ) : (
          <AiOutlinePlus title="Add to my list"
onClick={addToList} />
        )}
      </div>
      <div className="info">
        <BiChevronDown title="More Info" />
      </div>
    </div>
    <div className="genres flex">
      <ul className="flex">
        {movieData.genres.map((genre) => (
          <li>{genre}</li>
        ))}
      </ul>
    </div>
  </div>
</div>
))
</Container>
);
});
const Container = styled.div`

```

```
width: 230px;
height: 100%;
cursor: pointer;
position: relative;
img {
  border-radius: 0.2rem;
  width: 100%;
  height: 100%;
  z-index: 10;
}
.hover {
  z-index: 99;
  height: max-content;
  width: 20rem;
  position: absolute;
  top: -18vh;
  left: 0;
  border-radius: 0.3rem;
  box-shadow: rgba(0, 0, 0, 0.75) 0px 3px 10px;
  background-color: #181818;
  transition: 0.3s ease-in-out;
  .image-video-container {
    position: relative;
    height: 140px;
    img {
      width: 100%;
      height: 140px;
      object-fit: cover;
      border-radius: 0.3rem;
      top: 0;
      z-index: 4;
      position: absolute;
    }
    video {
      width: 100%;
```

```
        height: 140px;
        object-fit: cover;
        border-radius: 0.3rem;
        top: 0;
        z-index: 5;
        position: absolute;
    }
}
.info-container {
    padding: 1rem;
    gap: 0.5rem;
}
.icons {
    .controls {
        display: flex;
        gap: 1rem;
    }
    svg {
        font-size: 2rem;
        cursor: pointer;
        transition: 0.3s ease-in-out;
        &:hover {
            color: #b8b8b8;
        }
    }
}
.genres {
    ul {
        gap: 1rem;
        li {
            padding-right: 0.7rem;
            &:first-of-type {
                list-style-type: none; } } } } }
```

Source Code of netflix-clone/components/CardSlider.jsx


```

import React, { useRef, useState } from "react";
import styled from "styled-components";
import { AiOutlineLeft, AiOutlineRight } from "react-icons/ai";
import Card from "../Card";
export default React.memo(function CardSlider({ data, title }) {
  const listRef = useRef();
  const [sliderPosition, setSliderPosition] = useState(0);
  const [showControls, setShowControls] = useState(false);
  const handleDirection = (direction) => {
    let distance = listRef.current.getBoundingClientRect().x - 70;
    if (direction === "left" && sliderPosition > 0) {
      listRef.current.style.transform = `translateX(${230 + distance}px)`;
      setSliderPosition(sliderPosition - 1);
    }
    if (direction === "right" && sliderPosition < 4) {
      listRef.current.style.transform = `translateX(${ -230 +
distance}px)`;
      setSliderPosition(sliderPosition + 1);
    }
  };
  return (
    <Container
      className="flex column"
      showControls={showControls}
      onMouseEnter={() => setShowControls(true)}
      onMouseLeave={() => setShowControls(false)}
    >
      <h1>{title}</h1>
      <div className="wrapper">
        <div
          className={`slider-action left ${
            !showControls ? "none" : ""
          } flex j-center a-center`}
        >
          <AiOutlineLeft onClick={() => handleDirection("left")} />
        </div>

```

```

    {data.map((movie, index) => {
      return <Card movieData={movie} index={index} key={movie.id}
    />;
    })}
  </div>
  <div
    className={`slider-action right ${
      !showControls ? "none" : ""
    } flex j-center a-center`}
  >
    <AiOutlineRight onClick={() => handleDirection("right")} />
  </div>
</div>
</Container>
);
});
const Container = styled.div`
  gap: 1rem;
  position: relative;
  padding: 2rem 0;
  h1 { margin-left: 50px; }
  .wrapper {
    .slider {
      width: max-content;
      gap: 1rem;
      transform: translateX(0px);
      transition: 0.3s ease-in-out;
      margin-left: 50px;
    }
    .slider-action {
      position: absolute;
      z-index: 99;
      height: 100%;
      top: 0;
      bottom: 0;

```

```

    width: 50px;
    transition: 0.3s ease-in-out;

    svg {
        font-size: 2rem;
    }
}

.none { display: none; }
.left { left: 0; }
.right { right: 0; }
}
`;

```

Source Code of netflix-clone/components/Header.jsx

```

import React from "react";
import { useNavigate } from "react-router-dom";
import styled from "styled-components";
import logo from "../assets/logo.png";
export default function Header(props) {
    const navigate = useNavigate();
    return (
        <StyledHeader className="flex a-center j-between">
            <div className="logo">
                <img src={logo} alt="logo" />
            </div>
            <button onClick={() => navigate(props.login ? "/login" :
"/signup")}>
                {props.login ? "Log In" : "Sign Up"}
            </button>
        </StyledHeader>
    );
}

const StyledHeader = styled.header`
padding: 0 4rem;

.logo {
    img { height: 5rem; }
}

```

```
}  
button {  
  padding: 0.5rem 1rem;  
  background-color: #e50914;  
  border: none;  
  cursor: pointer;  
  color: white;  
  border-radius: 0.2rem;  
  font-weight: bolder;  
  font-size: 1.05rem;  
}  
`;  
`;
```

Source Code of netflix-clone/components/Navbar.jsx

```
import { signOut } from "firebase/auth";  
import React, { useState } from "react";  
import { Link } from "react-router-dom";  
import styled from "styled-components";  
import logo from "../assets/logo.png";  
import { firebaseAuth } from "../utils/firebase-config";  
import { FaPowerOff, FaSearch } from "react-icons/fa";  
export default function Navbar({ isScrolled }) {  
  const [showSearch, setShowSearch] = useState(false);  
  const [inputHover, setInputHover] = useState(false);  
  const links = [  
    { name: "Home", link: "/" },  
    { name: "TV Shows", link: "/tv" },  
    { name: "Movies", link: "/movies" },  
    { name: "My List", link: "/mylist" },  
  ];  
  return (  
    <Container>  
      <nav className={` ${isScrolled ? "scrolled" : ""} flex`}>  
        <div className="left flex a-center">  
          <div className="brand flex a-center j-center">
```

```

        <img src={logo} alt="Logo" />
      </div>
      <ul className="links flex">
        {links.map(({ name, link }) => {
          return (
            <li key={name}>
              <Link to={link}>{name}</Link>
            </li>
          );
        })}
      </ul>
    </div>
    <div className="right flex a-center">
      <div className={`search ${showSearch ? "show-search" : ""}`}>
        <button
          onFocus={() => setShowSearch(true)}
          onBlur={() => {
            if (!inputHover) {
              setShowSearch(false);
            }
          }}
        >
          <FaSearch />
        </button>
        <input
          type="text"
          placeholder="Search"
          onMouseEnter={() => setInputHover(true)}
          onMouseLeave={() => setInputHover(false)}
          onBlur={() => {
            setShowSearch(false);
            setInputHover(false);
          }}
        />
      </div>
    </div>

```

```
        <button onClick={() => signOut(firebaseAuth)}>
          <FaPowerOff />
        </button>
      </div>
    </nav>
  </Container>
);
}

const Container = styled.div`
  .scrolled {
    background-color: black;
  }
  nav {
    position: sticky;
    top: 0;
    height: 6.5rem;
    width: 100%;
    justify-content: space-between;
    position: fixed;
    top: 0;
    z-index: 2;
    padding: 0 4rem;
    align-items: center;
    transition: 0.3s ease-in-out;
    .left {
      gap: 2rem;
      .brand {
        img {
          height: 4rem;
        }
      }
      .links {
        list-style-type: none;
        gap: 2rem;
        li {
```

```
      a {
        color: white;
        text-decoration: none; } } }
    }
    .right {
      gap: 1rem;
      button {
        background-color: transparent;
        border: none;
        cursor: pointer;
        &:focus {
          outline: none;
        }
        svg {
          color: #f34242;
          font-size: 1.2rem;
        }
      }
    }
    .search {
      display: flex;
      gap: 0.4rem;
      align-items: center;
      justify-content: center;
      padding: 0.2rem;
      padding-left: 0.5rem;
      button {
        background-color: transparent;
        border: none;
        &:focus {
          outline: none;
        }
        svg {
          color: white;
          font-size: 1.2rem;
        }
      }
    }
  }
}
```

```

    }

    input {
      width: 0;
      opacity: 0;
      visibility: hidden;
      transition: 0.3s ease-in-out;
      background-color: transparent;
      border: none;
      color: white;
      &:focus {
        outline: none;
      }
    }
  }
}

.show-search {
  border: 1px solid white;
  background-color: rgba(0, 0, 0, 0.6);
  input {
    width: 100%;
    opacity: 1;
    visibility: visible;
    padding: 0.3rem; } } } }`;
import { signOut } from "firebase/auth";
import React, { useState } from "react";
import { Link } from "react-router-dom";
import styled from "styled-components";
import logo from "../assets/logo.png";
import { firebaseAuth } from "../utils/firebase-config";
import { FaPowerOff, FaSearch } from "react-icons/fa";
export default function Navbar({ isScrolled }) {
  const [showSearch, setShowSearch] = useState(false);
  const [inputHover, setInputHover] = useState(false);
  const links = [
    { name: "Home", link: "/" },
    { name: "TV Shows", link: "/tv" },

```



```

    { name: "Movies", link: "/movies" },
    { name: "My List", link: "/mylist" },
  ];
  return (
    <Container>
      <nav className={` ${isScrolled ? "scrolled" : ""} flex`} >
        <div className="left flex a-center">
          <div className="brand flex a-center j-center">
            <img src={logo} alt="Logo" />
          </div>
          <ul className="links flex">
            {links.map(({ name, link }) => {
              return (
                <li key={name}>
                  <Link to={link}>{name}</Link>
                </li>
              );
            })}
          </ul>
        </div>
        <div className="right flex a-center">
          <div className={` search ${showSearch ? "show-search" : ""}`} >
            <button
              onFocus={() => setShowSearch(true)}
              onBlur={() => {
                if (!inputHover) {
                  setShowSearch(false);
                }
              }}
            >
              <FaSearch />
            </button>
            <input
              type="text"
              placeholder="Search"

```

```

        onMouseEnter={() => setInputHover(true)}
        onMouseLeave={() => setInputHover(false)}
        onBlur={() => {
            setShowSearch(false);
            setInputHover(false);
        }}
    />
</div>
<button onClick={() => signOut(firebaseAuth)}>
    <FaPowerOff />
</button>
</div>
</nav>
</Container>
);
}

const Container = styled.div`
    .scrolled {
        background-color: black;
    }
    nav {
        position: sticky;
        top: 0;
        height: 6.5rem;
        width: 100%;
        justify-content: space-between;
        position: fixed;
        top: 0;
        z-index: 2;
        padding: 0 4rem;
        align-items: center;
        transition: 0.3s ease-in-out;
        .left {
            gap: 2rem;
            .brand {

```

```
    img { height: 4rem; }
  }
  .links {
    list-style-type: none;
    gap: 2rem;
    li {
      a {
        color: white;
        text-decoration: none;
      }
    }
  }
}
.right {
  gap: 1rem;
  button {
    background-color: transparent;
    border: none;
    cursor: pointer;
    &:focus {
      outline: none;
    }
    svg {
      color: #f34242;
      font-size: 1.2rem;
    }
  }
  .search {
    display: flex;
    gap: 0.4rem;
    align-items: center;
    justify-content: center;
    padding: 0.2rem;
    padding-left: 0.5rem;
    button {
```



```
export default function NotAvailable() {
  return (
    <h1 className="not-available">
      No Movies available for the selected genre. Please select a
different
      genre.
    </h1>
  );
}
```

Source Code of netflix-clone/components/SelectGenre.jsx

```
import React from "react";
import { useDispatch } from "react-redux";
import styled from "styled-components";
import { fetchDataByGenre } from "../store";
export default function SelectGenre({ genres, type }) {
  const dispatch = useDispatch();
  return (
    <Select
      className="flex"
      onChange={(e) => {
        dispatch(
          fetchDataByGenre({
            genres,
            genre: e.target.value,
            type,
          })
        );
      }}
    >
      {genres.map((genre) => {
        return (
          <option value={genre.id} key={genre.id}>
            {genre.name}
          </option>
        );
      })}
    </Select>
  );
}
```

```

}
const Select = styled.select`
  margin-left: 5rem;
  cursor: pointer;
  font-size: 1.4rem;
  background-color: rgba(0, 0, 0, 0.4);
  color: white;`;

```

Source Code of netflix-clone/components/Slider.jsx

```

import React from "react";
import styled from "styled-components";
import CardSlider from "../CardSlider";
export default function Slider({ movies }) {
  const getMoviesFromRange = (from, to) => {
    return movies.slice(from, to);
  };
  return (
    <Container>
      <CardSlider data={getMoviesFromRange(0, 10)} title="Trending Now" />
      <CardSlider data={getMoviesFromRange(10, 20)} title="New Releases"
    />
      <CardSlider
        data={getMoviesFromRange(20, 30)}
        title="Blockbuster Movies"
      />
      <CardSlider
        data={getMoviesFromRange(30, 40)}
        title="Popular on Netflix"
      />
      <CardSlider data={getMoviesFromRange(40, 50)} title="Action Movies"
    />
      <CardSlider data={getMoviesFromRange(50, 60)} title="Epics" />
    </Container>
  );
}
const Container = styled.div``;

```

Source Code of netflix-clone/pages/Login.jsx

```

import React, { useState } from "react";

```

```

import styled from "styled-components";
import logo from "../assets/logo.png";
import background from "../assets/login.jpg";
import { useNavigate } from "react-router-dom";
import BackgroundImage from "../components/BackgroundImage";
import Header from "../components/Header";
import { onAuthStateChanged, signInWithEmailAndPassword } from
"firebase/auth";
import { firebaseAuth } from "../utils/firebase-config";
function Login() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const navigate = useNavigate();
  const handleLogin = async () => {
    try {
      await signInWithEmailAndPassword(firebaseAuth, email, password);
    } catch (error) {
      console.log(error.code);
    }
  };
  onAuthStateChanged(firebaseAuth, (currentUser) => {
    if (currentUser) navigate("/");
  });
  return (
    <Container>
      <BackgroundImage />
      <div className="content">
        <Header />
        <div className="form-container flex column a-center j-center">
          <div className="form flex column a-center j-center">
            <div className="title">
              <h3>Login</h3>
            </div>
            <div className="container flex column">
              <input
                type="text"
                placeholder="Email"
                onChange={ (e) => setEmail(e.target.value)}

```

```

        value={email}
      />
      <input
        type="password"
        placeholder="Password"
        onChange={ (e) => setPassword(e.target.value) }
        value={password}
      />
      <button onClick={handleLogin}>Login to your account</button>
    </div>
  </div>
</div>
</div>
</Container> );}
const Container = styled.div`
  position: relative;
  .content {
    position: absolute;
    top: 0;
    left: 0;
    height: 100vh;
    width: 100vw;
    background-color: rgba(0, 0, 0, 0.5);
    grid-template-rows: 15vh 85vh;
    .form-container {
      gap: 2rem;
      height: 85vh;
      .form {
        padding: 2rem;
        background-color: #000000b0;
        width: 25vw;
        gap: 2rem;
        color: white;
        .container {
          gap: 2rem;

```



```
    input {
      padding: 0.5rem 1rem;
      width: 15rem;
    }
    button {
      padding: 0.5rem 1rem;
      background-color: #e50914;
      border: none;
      cursor: pointer;
      color: white;
      border-radius: 0.2rem;
      font-weight: bolder;
      font-size: 1.05rem; } } } } }
`;
export default Login;
```

```
import React, { useEffect, useState } from "react";
import styled from "styled-components";
import Navbar from "../components/Navbar";
import CardSlider from "../components/CardSlider";
import { onAuthStateChanged } from "firebase/auth";
import { firebaseAuth } from "../utils/firebase-config";
import { useNavigate } from "react-router-dom";
import { useSelector, useDispatch } from "react-redux";
import { fetchMovies, getGenres } from "../store";
import SelectGenre from "../components/SelectGenre";
import Slider from "../components/Slider";
import NotAvailable from "../components/NotAvailable";
function MoviePage() {
  const [isScrolled, setIsScrolled] = useState(false);
  const movies = useSelector((state) => state.netflix.movies);
  const genres = useSelector((state) => state.netflix.genres);
  const genresLoaded = useSelector((state) => state.netflix.genresLoaded);
  const navigate = useNavigate();
```

```

const dispatch = useDispatch();
useEffect(() => {
  dispatch(getGenres());
}, []);
useEffect(() => {
  if (genresLoaded) {
    dispatch(fetchMovies({ genres, type: "movie" }));
  }, [genresLoaded]);
const [user, setUser] = useState(undefined);
onAuthStateChanged(firebaseAuth, (currentUser) => {
  if (currentUser) setUser(currentUser.uid);
  else navigate("/login");
});
window.onscroll = () => {
  setIsScrolled(window.pageYOffset === 0 ? false : true);
  return () => (window.onscroll = null); };
return (
  <Container>
    <div className="navbar">
      <Navbar isScrolled={isScrolled} />
    </div>
    <div className="data">
      <SelectGenre genres={genres} type="movie" />
      {movies.length ? <Slider movies={movies} /> : <NotAvailable />}
    </div>
  </Container>
);}
const Container = styled.div`
.data {
  margin-top: 8rem;
  .not-available {
    text-align: center;
    color: white;
    margin-top: 4rem;
  }
}

```

```
    }  
  `;  
  export default MoviePage;
```

Source Code of netflix-clone/pages/Netflix.jsx

```
import React, { useEffect, useState } from "react";  
import styled from "styled-components";  
import Navbar from "../components/Navbar";  
import backgroundImage from "../assets/home.jpg";  
import MovieLogo from "../assets/homeTitle.webp";  
import { onAuthStateChanged } from "firebase/auth";  
import { firebaseAuth } from "../utils/firebase-config";  
import { useNavigate } from "react-router-dom";  
import { useSelector, useDispatch } from "react-redux";  
import { fetchMovies, getGenres } from "../store";  
import { FaPlay } from "react-icons/fa";  
import { AiOutlineInfoCircle } from "react-icons/ai";  
import Slider from "../components/Slider";  
function Netflix() {  
  const [isScrolled, setIsScrolled] = useState(false);  
  const movies = useSelector((state) => state.netflix.movies);  
  const genres = useSelector((state) => state.netflix.genres);  
  const genresLoaded = useSelector((state) => state.netflix.genresLoaded);  
  const navigate = useNavigate();  
  const dispatch = useDispatch();  
  useEffect(() => {  
    dispatch(getGenres());  
  }, []);  
  useEffect(() => {  
    if (genresLoaded) {  
      dispatch(fetchMovies({ genres, type: "all" }));  
    }  
  }, [genresLoaded]);  
  onAuthStateChanged(firebaseAuth, (currentUser) => {  
    if (!currentUser) navigate("/login");  
  });  
}
```

```

});

window.onscroll = () => {
  setIsScrolled(window.pageYOffset === 0 ? false : true);
  return () => (window.onscroll = null);
};

return (
  <Container>
    <Navbar isScrolled={isScrolled} />
    <div className="hero">
      <img
        src={backgroundImage}
        alt="background"
        className="background-image"
      />
      <div className="container">
        <div className="logo">
          <img src={MovieLogo} alt="Movie Logo" />
        </div>
        <div className="buttons flex">
          <button
            onClick={() => navigate("/player")}
            className="flex j-center a-center"
          >
            <FaPlay />
            Play
          </button>
          <button className="flex j-center a-center">
            <AiOutlineInfoCircle />
            More Info
          </button>
        </div>
      </div>
    </div>
    <Slider movies={movies} />
  </Container>

```

```
);  
}  
const Container = styled.div`  
  background-color: black;  
  .hero {  
    position: relative;  
    .background-image {  
      filter: brightness(60%);  
    }  
    img {  
      height: 100vh;  
      width: 100vw;  
    }  
    .container {  
      position: absolute;  
      bottom: 5rem;  
      .logo {  
        img {  
          width: 100%;  
          height: 100%;  
          margin-left: 5rem;  
        }  
      }  
    }  
    .buttons {  
      margin: 5rem;  
      gap: 2rem;  
      button {  
        font-size: 1.4rem;  
        gap: 1rem;  
        border-radius: 0.2rem;  
        padding: 0.5rem;  
        padding-left: 2rem;  
        padding-right: 2.4rem;  
        border: none;  
        cursor: pointer;  
      }  
    }  
  }  
`;
```

```

        transition: 0.2s ease-in-out;
        &:hover {
            opacity: 0.8;
        }
        &:nth-of-type(2) {
            background-color: rgba(109, 109, 110, 0.7);
            color: white;
            svg {
                font-size: 1.8rem; } } } } }
    }
`;
export default Netflix;

```

Source Code of netflix-clone/pages/Player.jsx

```

import React from "react";
import styled from "styled-components";
import { BsArrowLeft } from "react-icons/bs";
import { useNavigate } from "react-router-dom";
import video from "../assets/video.mkv";
export default function Player() {
    const navigate = useNavigate();
    return (
        <Container>
            <div className="player">
                <div className="back">
                    <BsArrowLeft onClick={() => navigate(-1)} />
                </div>
                <video src={video} autoPlay loop controls muted />
            </div>
        </Container>
    );
}
const Container = styled.div`
    .player {
        width: 100vw;
    }

```

```
height: 100vh;
.back {
  position: absolute;
  padding: 2rem;
  z-index: 1;
  svg {
    font-size: 3rem;
    cursor: pointer;
  }
}
video {
  height: 100%;
  width: 100%;
  object-fit: cover; } } `;
```

Source Code of netflix-clone/pages/Signup.jsx

```
import {
  createUserWithEmailAndPassword,
  onAuthStateChanged,
} from "firebase/auth";
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import styled from "styled-components";
import BackgroundImage from "../components/BackgroundImage";
import Header from "../components/Header";
import { firebaseAuth } from "../utils/firebase-config";
function Signup() {
  const [showPassword, setShowPassword] = useState(false);
  const [formValues, setFormValues] = useState({
    email: "",
    password: "",
  });
  const navigate = useNavigate();
  const handleSignIn = async () => {
    try {
```

```

    const { email, password } = formValues;
    await createUserWithEmailAndPassword(firebaseAuth, email, password);
  } catch (error) {
    console.log(error);
  }
};

onAuthStateChanged(firebaseAuth, (currentUser) => {
  if (currentUser) navigate("/");
});

return (
  <Container showPassword={showPassword}>
    <BackgroundImage />
    <div className="content">
      <Header login />
      <div className="body flex column a-center j-center">
        <div className="text flex column">
          <h1>Unlimited movies, TV shows and more.</h1>
          <h4>Watch anywhere. Cancel anytime.</h4>
          <h6>
            Ready to watch? Enter your email to create or restart
membership.
          </h6>
        </div>
        <div className="form">
          <input
            type="email"
            placeholder="Email address"
            onChange={ (e) =>
              setFormValues({
                ...formValues,
                [e.target.name]: e.target.value,
              })
            }
            name="email"
            value={formValues.email}
          />

```



```

        {showPassword && (
            <input
                type="password"
                placeholder="Password"
                onChange={ (e) =>
                    setFormValues({
                        ...formValues,
                        [e.target.name]: e.target.value,
                    })
                }
                name="password"
                value={formValues.password}
            />
        )}
        {!showPassword && (
            <button onClick={ () => setShowPassword(true) }>Get
Started</button>
        )}
    </div>
    {showPassword && <button onClick={handleSignIn}>Sign
Up</button>}
    </div>
</div>
</Container>
);
}

const Container = styled.div`
    position: relative;
    .content {
        position: absolute;
        top: 0;
        left: 0;
        background-color: rgba(0, 0, 0, 0.5);
        height: 100vh;
        width: 100vw;
        display: grid;

```

```
grid-template-rows: 15vh 85vh;

.body {
  gap: 1rem;
  .text {
    gap: 1rem;
    text-align: center;
    font-size: 2rem;
    h1 {
      padding: 0 25rem;
    }
  }
}

.form {
  display: grid;
  grid-template-columns: ${({ showPassword }) =>
    showPassword ? "1fr 1fr" : "2fr 1fr"};
  width: 60%;
  input {
    color: black;
    border: none;
    padding: 1.5rem;
    font-size: 1.2rem;
    border: 1px solid black;
    &:focus {
      outline: none;
    }
  }
  button {
    padding: 0.5rem 1rem;
    background-color: #e50914;
    border: none;
    cursor: pointer;
    color: white;
    font-weight: bolder;
    font-size: 1.05rem;
  }
}
```

```
    }  
    button {  
      padding: 0.5rem 1rem;  
      background-color: #e50914;  
      border: none;  
      cursor: pointer;  
      color: white;  
      border-radius: 0.2rem;  
      font-weight: bolder;  
      font-size: 1.05rem; } } }  
`;  
export default Signup;
```

Source Code of netflix-clone/pages/TVShows.jsx

```
import React, { useEffect, useState } from "react";  
import styled from "styled-components";  
import Navbar from "../components/Navbar";  
import CardSlider from "../components/CardSlider";  
import { onAuthStateChanged } from "firebase/auth";  
import { firebaseAuth } from "../utils/firebase-config";  
import { useNavigate } from "react-router-dom";  
import { useSelector, useDispatch } from "react-redux";  
import { fetchMovies, getGenres } from "../store";  
import SelectGenre from "../components/SelectGenre";  
import Slider from "../components/Slider";  
function TVShows() {  
  const [isScrolled, setIsScrolled] = useState(false);  
  const movies = useSelector((state) => state.netflix.movies);  
  const genres = useSelector((state) => state.netflix.genres);  
  const genresLoaded = useSelector((state) => state.netflix.genresLoaded);  
  const dataLoading = useSelector((state) => state.netflix.dataLoading);  
  const navigate = useNavigate();  
  const dispatch = useDispatch();  
  useEffect(() => {  
    if (!genres.length) dispatch(getGenres());  
  });  
}
```

```

    }, []);

    useEffect(() => {
      if (genresLoaded) {
        dispatch(fetchMovies({ genres, type: "tv" }));
      }
    }, [genresLoaded]);

    const [user, setUser] = useState(undefined);
    onAuthStateChanged(firebaseAuth, (currentUser) => {
      if (currentUser) setUser(currentUser.uid);
      else navigate("/login");
    });

    window.onscroll = () => {
      setIsScrolled(window.pageYOffset === 0 ? false : true);
      return () => (window.onscroll = null);
    };

    return (
      <Container>
        <Navbar isScrolled={isScrolled} />
        <div className="data">
          <SelectGenre genres={genres} type="tv" />
          {movies.length ? (
            <>
              <Slider movies={movies} />
            </>
          ) : (
            <h1 className="not-available">
              No TV Shows available for the selected genre. Please select a
              different genre.
            </h1>
          )}
        </div>
      </Container>
    );
  }

  const Container = styled.div`

```

```
.data {  
  margin-top: 8rem;  
  .not-available {  
    text-align: center;  
    margin-top: 4rem;  
  }  
}  
`;  
  
export default TVShows;
```

Source Code of netflix-clone/pages/UserListedMovies.jsx

```
import axios from "axios";  
import { onAuthStateChanged } from "firebase/auth";  
import React, { useEffect, useState } from "react";  
import { useNavigate } from "react-router-dom";  
import { firebaseAuth } from "../utils/firebase-config";  
import Card from "../components/Card";  
import styled from "styled-components";  
import Navbar from "../components/Navbar";  
import { getUsersLikedMovies } from "../store";  
import { useDispatch, useSelector } from "react-redux";  
export default function UserListedMovies() {  
  const movies = useSelector((state) => state.netflix.movies);  
  const dispatch = useDispatch();  
  const navigate = useNavigate();  
  const [isScrolled, setIsScrolled] = useState(false);  
  const [email, setEmail] = useState(undefined);  
  onAuthStateChanged(firebaseAuth, (currentUser) => {  
    if (currentUser) setEmail(currentUser.email);  
    else navigate("/login");  
  });  
  useEffect(() => {  
    if (email) {  
      dispatch(getUsersLikedMovies(email));  
    }  
  })  
}
```

```

    }, [email]);

    window.onscroll = () => {
      setIsScrolled(window.pageYOffset === 0 ? false : true);
      return () => (window.onscroll = null);
    };

    return (
      <Container>
        <Navbar isScrolled={isScrolled} />
        <div className="content flex column">
          <h1>My List</h1>
          <div className="grid flex">
            {movies.map((movie, index) => {
              return (
                <Card
                  movieData={movie}
                  index={index}
                  key={movie.id}
                  isLiked={true}
                />
              );
            })}
          </div>
        </div>
      </Container>
    );
  }
}

const Container = styled.div`
  .content {
    margin: 2.3rem;
    margin-top: 8rem;
    gap: 3rem;
    h1 {
      margin-left: 3rem;
    }
    .grid {

```

```
    flex-wrap: wrap;
    gap: 1rem;
  }
}
`;
```

Source Code of netflix-clone/store/index.js

```
import {
  configureStore,
  createAsyncThunk,
  createSlice,
} from "@reduxjs/toolkit";
import axios from "axios";
import { API_KEY, TMDB_BASE_URL } from "../utils/constants";
const initialState = {
  movies: [],
  genresLoaded: false,
  genres: [],
};
export const getGenres = createAsyncThunk("netflix/genres", async () => {
  const {
    data: { genres },
  } = await axios.get(
    "https://api.themoviedb.org/3/genre/movie/list?api_key=3d39d6bfe362592e6aa293f01fbcf9b9"
  );
  return genres;
});
const createArrayFromRawData = (array, moviesArray, genres) => {
  array.forEach((movie) => {
    const movieGenres = [];
    movie.genre_ids.forEach((genre) => {
      const name = genres.find(({ id }) => id === genre);
      if (name) movieGenres.push(name.name);
    });
  });
};
```

```

        if (movie.backdrop_path)
            moviesArray.push({
                id: movie.id,
                name: movie?.original_name ? movie.original_name :
movie.original_title,
                image: movie.backdrop_path,
                genres: movieGenres.slice(0, 3),
            });
    });
};

const getRawData = async (api, genres, paging = false) => {
    const moviesArray = [];
    for (let i = 1; moviesArray.length < 60 && i < 10; i++) {
        const {
            data: { results },
        } = await axios.get(`${api}${paging ? `&page=${i}` : ""}`);
        createArrayFromRawData(results, moviesArray, genres);
    }
    return moviesArray;
};

export const fetchDataByGenre = createAsyncThunk(
    "netflix/genre",
    async ({ genre, type }, thunkAPI) => {
        const {
            netflix: { genres },
        } = thunkAPI.getState();
        return getRawData(
            `https://api.themoviedb.org/3/discover/${type}?api_key=3d39d6bfe362592e6aa
293f01fbcf9b9&with_genres=${genre}`,
            genres
        );
    }
);

export const fetchMovies = createAsyncThunk(
    "netflix/trending",
    async ({ type }, thunkAPI) => {

```



```
const {
  netflix: { genres },
} = thunkAPI.getState();
return getRawData(
  `${TMDB_BASE_URL}/trending/${type}/week?api_key=${API_KEY}`,
  genres,
  true
);
}
);
export const getUsersLikedMovies = createAsyncThunk(
  "netflix/getLiked",
  async (email) => {
    const {
      data: { movies },
    } = await axios.get(`http://localhost:5000/api/user/liked/${email}`);
    return movies;
  }
);
export const removeMovieFromLiked = createAsyncThunk(
  "netflix/deleteLiked",
  async ({ movieId, email }) => {
    const {
      data: { movies },
    } = await axios.put("http://localhost:5000/api/user/remove", {
      email,
      movieId,
    });
    return movies;
  }
);
const NetflixSlice = createSlice({
  name: "Netflix",
  initialState,
  extraReducers: (builder) => {
```

```
builder.addCase(getGenres.fulfilled, (state, action) => {
  state.genres = action.payload;
  state.genresLoaded = true;
});

builder.addCase(fetchMovies.fulfilled, (state, action) => {
  state.movies = action.payload;
});

builder.addCase(fetchDataByGenre.fulfilled, (state, action) => {
  state.movies = action.payload;
});

builder.addCase(getUsersLikedMovies.fulfilled, (state, action) => {
  state.movies = action.payload;
});

builder.addCase(removeMovieFromLiked.fulfilled, (state, action) => {
  state.movies = action.payload;
});
},
});

export const store = configureStore({
  reducer: {
    netflix: NetflixSlice.reducer,
  },
});

export const { setGenres, setMovies } = NetflixSlice.actions;
```

Source Code of netflix-clone/utils/firebase-config.js

```
import { initializeApp } from "firebase/app";
import { getAuth } from "firebase/auth";
const firebaseConfig = {
  apiKey: "AIzaSyDDzWaaVTxfvyfxpPeS8ztJsaiOaw4RkOQ",
  authDomain: "netflix-clone-2120e.firebaseio.com",
  projectId: "netflix-clone-2120e",
  storageBucket: "netflix-clone-2120e.appspot.com",
  messagingSenderId: "848027499523",
```

```

    appId: "1:848027499523:web:d7f40a1e01751007912236",
    measurementId: "G-YG64LJ4M8K"
  };

const app = initializeApp(firebaseConfig); // Initialize Firebase
export const firebaseAuth = getAuth(app);

```

Source Code of netflix-clone/App.js

```

import React from "react";
import { BrowserRouter, Route, Routes } from "react-router-dom";
import Login from "./pages/Login";
import MoviePage from "./pages/Movies";
import Netflix from "./pages/Netflix";
import Player from "./pages/Player";
import Signup from "./pages/Signup";
import TVShows from "./pages/TVShows";
import UserListedMovies from "./pages/UserListedMovies";
export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route exact path="/login" element={<Login />} />
        <Route exact path="/signup" element={<Signup />} />
        <Route exact path="/player" element={<Player />} />
        <Route exact path="/tv" element={<TVShows />} />
        <Route exact path="/movies" element={<MoviePage />} />
        <Route exact path="/new" element={<Player />} />
        <Route exact path="/mylist" element={<UserListedMovies />} />
        <Route exact path="/" element={<Netflix />} />
      </Routes>
    </BrowserRouter>
  );
}

```

Source Code of netflix-clone/index.js

```

import React from "react";

```

```
import ReactDOM from "react-dom/client";
import { Provider } from "react-redux";
import "./index.css";
import App from "./App";
import { store } from "./store";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <Provider store={store}>
      <App />
    </Provider>
  </React.StrictMode>
);
```

NETFLIX-API

Source Code of netflix-clone/controllers/UserController.js

```
const User = require("../models/UserModel");
module.exports.getLikedMovies = async (req, res) => {
  try {
    const { email } = req.params;
    const user = await await User.findOne({ email });
    if (user) {
      return res.json({ msg: "success", movies: user.likedMovies });
    } else return res.json({ msg: "User with given email not found." });
  } catch (error) {
    return res.json({ msg: "Error fetching movies." });
  }
};

module.exports.addToLikedMovies = async (req, res) => {
  try {
    const { email, data } = req.body;
    const user = await await User.findOne({ email });
    if (user) {
      const { likedMovies } = user;
```

```

    const movieAlreadyLiked = likedMovies.find(({ id }) => id ===
data.id);

    if (!movieAlreadyLiked) {
      await User.findByIdAndUpdate(
        user._id,
        {
          likedMovies: [...user.likedMovies, data],
        },
        { new: true }
      );
    } else return res.json({ msg: "Movie already added to the liked
list." });

    } else await User.create({ email, likedMovies: [data] });
    return res.json({ msg: "Movie successfully added to liked list." });
  } catch (error) {
    return res.json({ msg: "Error adding movie to the liked list" });
  }
};

module.exports.removeFromLikedMovies = async (req, res) => {
  try {
    const { email, movieId } = req.body;
    const user = await User.findOne({ email });
    if (user) {
      const movies = user.likedMovies;
      const movieIndex = movies.findIndex(({ id }) => id === movieId);
      if (!movieIndex) {
        res.status(400).send({ msg: "Movie not found." });
      }
      movies.splice(movieIndex, 1);
      await User.findByIdAndUpdate(
        user._id,
        {
          likedMovies: movies,
        },
        { new: true }
      );
    }
  }
};

```

```
        return res.json({ msg: "Movie successfully removed.", movies });
    } else return res.json({ msg: "User with given email not found." });
} catch (error) {
    return res.json({ msg: "Error removing movie to the liked list" });
}
};
```

Source Code of netflix-clone/routes/UserRoutes.js

```
const {
    addToLikedMovies,
    getLikedMovies,
    removeFromLikedMovies,
} = require("../controllers/UserController");
const router = require("express").Router();
router.get("/liked/:email", getLikedMovies);
router.post("/add", addToLikedMovies);
router.put("/remove", removeFromLikedMovies);
module.exports = router;
```

Source Code of netflix-clone/Server.js

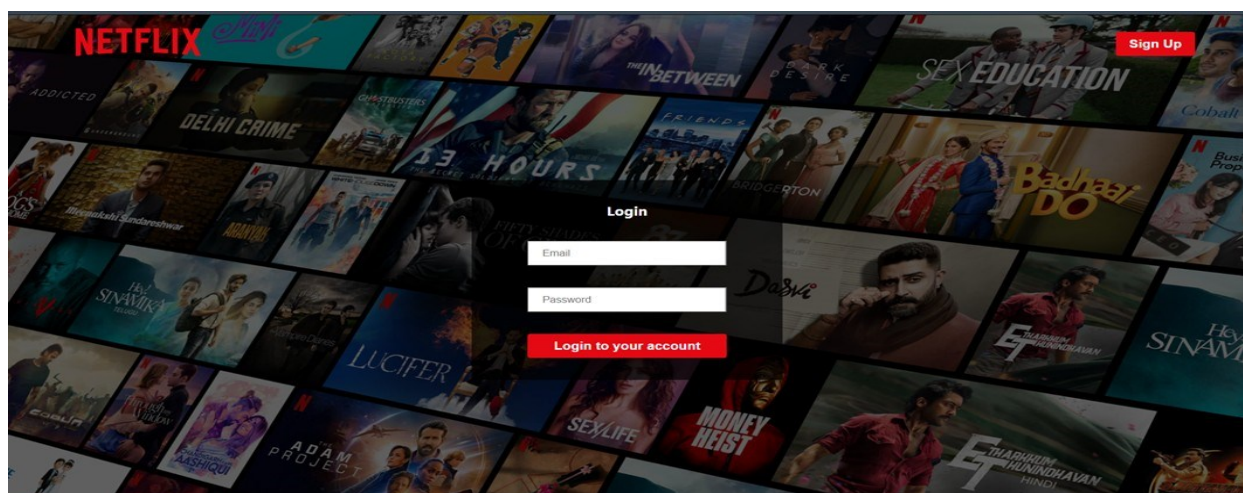
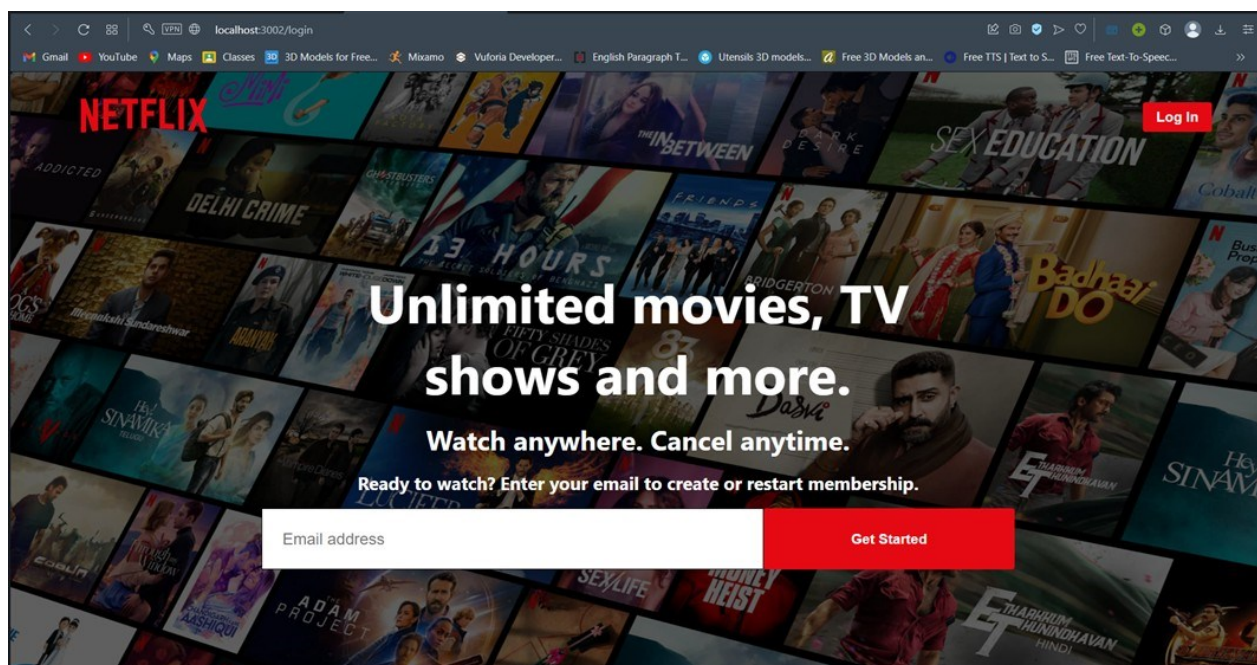
```
const express = require("express");
const cors = require("cors");
const userRoutes = require("../routes/UserRoutes");
const mongoose = require("mongoose");
const app = express();
app.use(cors());
app.use(express.json());
mongoose
    .connect("mongodb://localhost:27017/netflix", {
        useNewUrlParser: true,
        useUnifiedTopology: true,
    })
    .then(() => {
        console.log("DB Connection Successfull");
    })
```

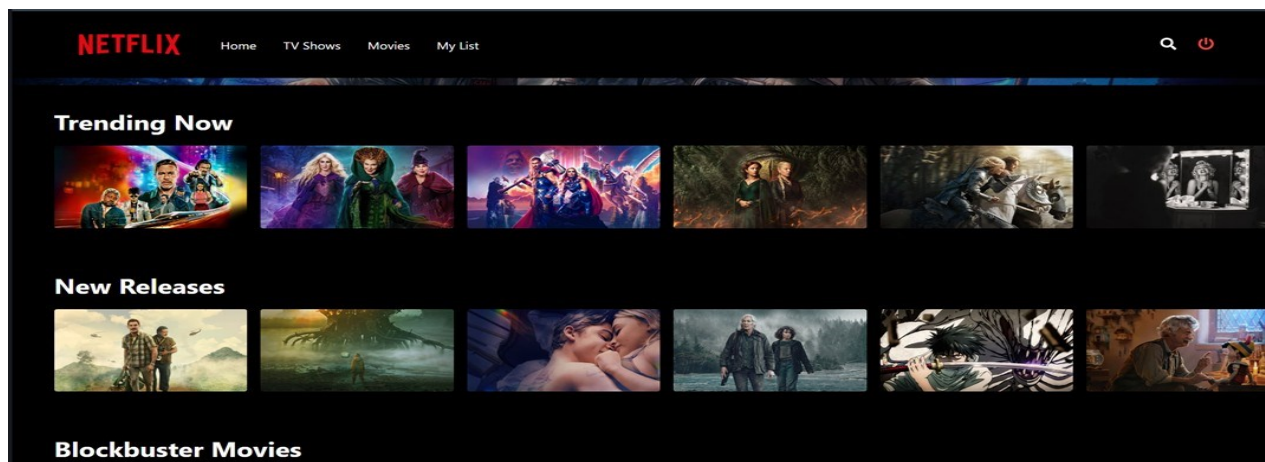
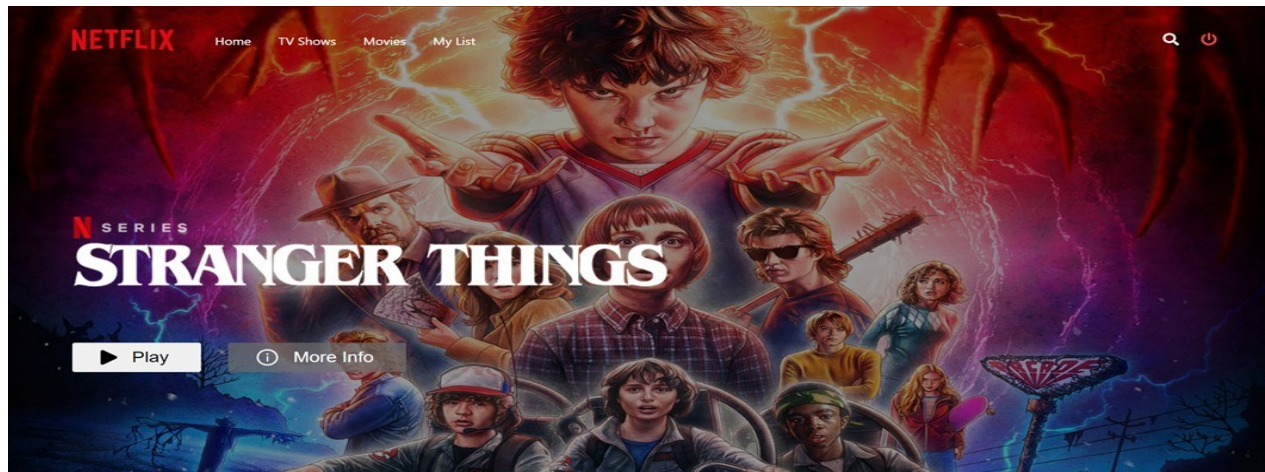
```

    .catch((err) => {
      console.log(err.message);
    });
app.use("/api/user", userRoutes);
app.listen(5000, () => {
  console.log("server started on port 5000");
});

```

OUTPUT:





CONCLUSION: In this practice we implement the clone of NETFLIX video streaming web application.

PRACTICAL 09

AIM: Build an exact similar clone of any social media web application like Instagram/LinkedIn/WhatsApp/Twitter etc. Your web application should contain following things,

1. Build a Login/Signup.
2. Build the sidebar.
3. Chatting can be done between 2 users.
4. Build the rows to display the recent chats.
5. Build the sidebar.

Finally, host your web application on Google Firebase and push the code to your GitHub Repository to enhance your GitHub Profile.

FRONT-END

Source Code of client/src/app.css

```
.App {
  text-align: center;
}
.App-logo {
  height: 40vmin;
  pointer-events: none;
}
@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}
.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
```

```

    flex-direction: column;
    align-items: center;
    justify-content: center;
    font-size: calc(10px + 2vmin);
    color: white;
  }
  .App-link {
    color: #61dafb;
  }
  @keyframes App-logo-spin {
    from {
      transform: rotate(0deg);
    }
    to {
      transform: rotate(360deg);
    }
  }

```

Source Code of client/src/app.js

```

import { lazy, Suspense } from 'react';
import { GoogleAuthProvider } from '@react-oauth/google';
import UserProvider from './context/UserProvider';
import AccountProvider from './context/AccountProvider';
import Loader from './components/loader/Loader';
const Messenger = lazy(() => import('./components/Messenger'));
function App() {
  const clientId =
    '357591291047-de4g78bkkqddj4i3isfe310jos75tlpa.apps.googleusercontent.com';
  return (
    <GoogleAuthProvider clientId={clientId}>
      <UserProvider>
        <AccountProvider>
          <Suspense fallback={<Loader />}>
            <Messenger/>
          </Suspense>
        </AccountProvider>
      </UserProvider>
    </GoogleAuthProvider>
  );
}

```

```

    </AuthProvider>
    </GoogleOAuthProvider>
  );
}
export default App;

```

Source Code of client/src/components/account/LoginDialog.jsx

```

import { useContext } from 'react';
import { Dialog, Typography, List, ListItem, Box, styled } from
 '@mui/material';
import { GoogleLogin } from '@react-oauth/google';
import jwt_decode from "jwt-decode";
import { addUser } from '../../service/api';
import { AccountContext } from '../../context/AccountProvider';
import { qrCodeImage } from '../../constants/data';
const Component = styled(Box) `
  display: flex;`;
const Container = styled(Box) `
  padding: 56px 0 56px 56px;`;
const QRCode = styled('img') ({
  margin: '50px 0 0 50px',
  height: 264,
  width: 264
});
const Title = styled(Typography) `
  font-size: 26px;
  margin-bottom: 25px;
  color: #525252;
  font-family: Segoe UI, Helvetica Neue, Helvetica, Lucida
Grande, Arial, Ubuntu, Cantarell, Fira Sans, sans-serif;
  font-weight: 300;`;
const StyledList = styled(List) `
  & > li {
    padding: 0;
    margin-top: 15px;
    font-size: 18px;

```

```

        line-height: 28px;
        color: #4a4a4a;
    }`;

const dialogStyle = {
    marginTop: '12%',
    height: '95%',
    width: '60%',
    maxWidth: '100%',
    maxHeight: '100%',
    borderRadius: 0,
    boxShadow: 'none',
    overflow: 'hidden'}

const LoginDialog = () => {
    const { setAccount, showloginButton, setShowloginButton,
    setShowlogoutButton } = useContext(AccountContext);

    const onLoginSuccess = async (res) => {
        let decoded = jwt_decode(res.credential);
        setAccount(decoded);
        setShowloginButton(false);
        setShowlogoutButton(true);
        await addUser(decoded);    };

    const onLoginFailure = (res) => {
        console.log('Login Failed:', res);    };

    const onSignoutSuccess = () => {
        alert("You have been logged out successfully");
        console.clear();
        setShowloginButton(true);
        setShowlogoutButton(false);    };

    return (
        <Dialog
            open={true}
            BackdropProps={{style: {backgroundColor: 'unset'}}}
            maxWidth={'md'}
            PaperProps={{ sx: dialogStyle }}
        >

        <Component>

```

```

        <Container>
          <Title>To use WhatsApp on your computer:</Title>
          <StyledList>
            <ListItem>1. Open WhatsApp on your
phone</ListItem>
            <ListItem>2. Tap Menu Settings and select WhatsApp
Web</ListItem>
            <ListItem>3. Point your phone to this screen to
capture the code</ListItem>
          </StyledList>
        </Container>
        <Box style={{position:'relative'}}>
          <QRCode src={qrCodeImage} alt="QR Code" />
          <Box style={{position: 'absolute', top: '50%',
transform: 'translateX(25%) translateY(-25%)'}}>
            { showloginButton ?
              <GoogleLogin
                buttonText=""
                onSuccess={onLoginSuccess}
                onError={onLoginFailure}
              /> : null}
          </Box>
        </Box>
      </Component>
    </Dialog>
  )}
export default LoginDialog;

```

Source Code of client/src/components/chat/ChatBox.jsx

```

import { useContext, useState, useEffect } from 'react';
import { Box } from '@mui/material';
import { UserContext } from '../../../context/UserProvider';
import { AccountContext } from '../../../context/AccountProvider';
import { getConversation } from '../../../service/api';
import ChatHeader from './ChatHeader';
import Messages from './Messages';
const ChatBox = () => {

```

```

    const { person } = useContext(UserContext);
    const { account } = useContext(AccountContext);
    const [conversation, setConversation] = useState({});
    useEffect(() => {
        const getConversationDetails = async () => {
            let data = await getConversation({ senderId: account.sub,
receiverId: person.sub });
            setConversation(data);
        }
        getConversationDetails();
    }, [person.sub]);
    return (
        <Box style={{height: '75%'}}>
            <ChatHeader person={person} />
            <Messages person={person} conversation={conversation} />
        </Box>
    )
}
export default ChatBox;

```

Source Code of client/src/components/chat/ChatHeader.jsx

```

import { useContext } from 'react';
import { Box, Typography, styled } from '@mui/material';
import { Search, MoreVert } from '@mui/icons-material';
import { AccountContext } from '../../../context/AccountProvider';
import { defaultProfilePicture } from '../../../constants/data';
const Header = styled(Box)`
    height: 44px;
    background: #ededed;
    display: flex;
    padding: 8px 16px;
    align-items: center;`;
const Image = styled('img')({
    width: 40,
    height: 40,
    objectFit: 'cover',

```

```

        borderRadius: '50%'))
const Name = styled(Typography) `
    margin-left: 12px !important;`;
const RightContainer = styled(Box) `
    margin-left: auto;
    & > svg {
        padding: 8px;
        font-size: 22px;
        color: #000; }`;
const Status = styled(Typography) `
    font-size: 12px !important;
    color: rgb(0, 0, 0, 0.6);
    margin-left: 12px !important;`;
const ChatHeader = ({ person }) => {
    const url = person.picture || defaultProfilePicture;
    const { activeUsers } = useContext(AccountContext);
    return (
        <Header>
            <Image src={url} alt="display picture" />
            <Box>
                <Name>{person.name}</Name>
                <Status>{activeUsers?.find(user => user.sub ===
person.sub) ? 'Online' : 'Offline'}</Status>
            </Box>
            <RightContainer>
                <Search />
                <MoreVert />
            </RightContainer>
        </Header>
    )
}
export default ChatHeader;

```

Source Code of client/src/components/chat/EmptyChat.jsx

```
import { Box, styled, Typography, Divider } from '@mui/material';
```

```

import { emptyChatImage } from '../../../constants/data';

const Component = styled(Box) `
  background: #f8f9fa;
  padding: 30px 0;
  text-align: center;
  height: 100%;`;

const Container = styled(Box) `
  padding: 0 200px;`;

const Image = styled('img') ({
  marginTop: 100,
  width: 400
})

const Title = styled(Typography) `
  font-size: 32px;
  font-family: inherit;
  font-weight: 300;
  color: #41525d;
  margin-top: 25px 0 10px 0;`;

const SubTitle = styled(Typography) `
  font-size: 14px;
  color: #667781;
  font-weight: 400;
  font-family: inherit;`;

const StyledDivider = styled(Divider) `
  margin: 40px 0;
  opacity: 0.4;`;

const EmptyChat = () => {

  return (
    <Component>
      <Container>
        <Image src={emptyChatImage} alt="empty" />
        <Title>WhatsApp Web</Title>
        <SubTitle>Now send and receive messages without keeping
your phone online.</SubTitle>

```



```

        <SubTitle>Use WhatsApp on up to 4 linked devices and 1
phone at the same time. </SubTitle>

        <StyledDivider />

    </Container>

</Component>

)
}

export default EmptyChat;

```

Source Code of client/src/components/chat/Message.jsx

```

import { useContext } from 'react';
import { Box, styled, Typography } from '@mui/material';
import { GetApp as GetAppIcon } from '@mui/icons-material';
import { AccountContext } from '../../../context/AccountProvider';
import { downloadMedia, formatDate } from '../../../utils/common-utils';
import { iconPDF } from '../../../constants/data';

const Wrapper = styled(Box) `
    background: #FFFFFF;
    padding: 5px;
    max-width: 60%;
    width: fit-content;
    display: flex;
    border-radius: 10px;
    word-break: break-word;`;

const Own = styled(Box) `
    background: #dcf8c6;
    padding: 5px;
    max-width: 60%;
    width: fit-content;
    margin-left: auto;
    display: flex;
    border-radius: 10px;
    word-break: break-word;`;

const Text = styled(Typography) `
    font-size: 14px;

```

```

padding: 0 25px 0 5px; ;
const Time = styled(Typography) `
  font-size: 10px;
  color: #919191;
  margin-top: 6px;
  word-break: keep-all;
  margin-top: auto;`;
const Message = ({ message }) => {
  const { account } = useContext(AccountContext);
  return (
    <>
    { account.sub === message.senderId ?
      <Own>
        { message.type === 'file' ? <ImageMessage
message={message} /> : <TextMessage message={message} />
        }
      </Own>
      :
      <Wrapper>
        { message.type === 'file' ? <ImageMessage
message={message} /> : <TextMessage message={message} />
        }
      </Wrapper>
    }
    </>
  )}
const TextMessage = ({ message }) => {
  return (
    <>
    <Text>{message.text}</Text>
    <Time>{formatDate(message.createdAt)}</Time>
    </>
  )}
const ImageMessage = ({ message }) => {
  return (
    <div style={{ position: 'relative' }}>

```

```

        message?.text?.includes('.pdf') ?
            <div style={{ display: 'flex' }}>
                <img src={iconPDF} alt="pdf-icon" style={{ width:
80 }} />
                <Typography style={{ fontSize: 14 }}>
>{message.text.split("/").pop()}</Typography>
            </div>
        :
            <img style={{ width: 300, height: '100%', objectFit:
'cover' }} src={message.text} alt={message.text} />
        }
        <Time style={{ position: 'absolute', bottom: 0, right: 0 }}>
            <GetAppIcon
                onClick={(e) => downloadMedia(e, message.text)}
                fontSize='small'
                style={{ marginRight: 10, border: '1px solid grey',
borderRadius: '50%' }}
            />
            {formatDate(message.createdAt)}
        </Time>
    </div> )}
export default Message;

```

Source Code of client/src/components/chat/Messages.jsx

```

import { useState, useEffect, useContext, useRef } from 'react';
import { Box, styled } from '@mui/material';
import { io } from 'socket.io-client';
import { getMessages, newMessages } from '../../../service/api';
import { AccountContext } from '../../../context/AccountProvider';import
Message from './Message';
import Footer from './Footer';
const Wrapper = styled(Box)`
    background-image:
url(${`https://user-images.githubusercontent.com/15075759/28719144-86dc0f7
0-73b1-11e7-911d-60d70fcded21.png`});
    background-size: 50%;`;
const StyledFooter = styled(Box)`
    height: 55px;
    background: #ededed;

```

```

width: 100%; ;
const Component = styled(Box) `
  height: 80vh;
  overflow-y: scroll;;
const Container = styled(Box) `
  padding: 1px 80px;`;
const Messages = ({ person, conversation }) => {
  const [messages, setMessages] = useState([]);
  const [incomingMessage, setIncomingMessage] = useState(null);
  const [value, setValue] = useState();
  const [file, setFile] = useState();
  const [image, setImage] = useState();
  const scrollRef = useRef();
  const { account, socket, newMessageFlag, setNewMessageFlag } =
    useContext(AccountContext);
  useEffect(() => {
    socket.current.on('getMessage', data => {
      setIncomingMessage({
        ...data,
        createdAt: Date.now()
      })
    })
  }, []);
  useEffect(() => {
    const getMessageDetails = async () => {
      let data = await getMessages(conversation?._id);
      setMessages(data);
    }
    getMessageDetails();
  }, [conversation?._id, person._id, newMessageFlag]);
  useEffect(() => {
    scrollRef.current?.scrollIntoView({ transition: "smooth" })
  }, [messages]);
  useEffect(() => {
    incomingMessage &&
    conversation?.members?.includes(incomingMessage.senderId) &&

```

```

        setMessages((prev) => [...prev, incomingMessage]);
    }, [incomingMessage, conversation]);

    const receiverId = conversation?.members?.find(member => member !==
account.sub);

    const sendText = async (e) => {
        let code = e.keyCode || e.which;
        if(!value) return;
        if(code === 13) {
            let message = {};
            if (!file) {
                message = {
                    senderId: account.sub,
                    receiverId: receiverId,
                    conversationId: conversation._id,
                    type: 'text',
                    text: value
                };
            } else {
                message = {
                    senderId: account.sub,
                    conversationId: conversation._id,
                    receiverId: receiverId,
                    type: 'file',
                    text: image
                };
            }
            socket.current.emit('sendMessage', message);
            await newMessages(message);
            setValue('');
            setFile();
            setImage('');
            setNewMessageFlag(prev => !prev);
        } }

    return (
        <Wrapper>
            <Component>

```

```

        {
            messages && messages.map(message => (
                <Container ref={scrollRef}>
                    <Message message={message} />
                </Container>
            ))
        }
    </Component>
    <Footer
        sendText={sendText}
        value={value}
        setValue={setValue}
        setFile={setFile}
        file={file}
        setImage={setImage}
    />
</Wrapper>
)}
export default Messages;

```

Source Code of client/src/components/menu/Conversation.jsx

```

import { useContext, useEffect, useState } from 'react';
import { styled, Box, Typography } from "@mui/material";
import { UserContext } from '../../../context/UserProvider';
import { AccountContext } from '../../../context/AccountProvider';
import { setConversation, getConversation } from '../../../service/api';
import { emptyProfilePicture } from '../../../constants/data';
import { formatDate } from '../../../utils/common-utils';
const Component = styled(Box)`
    height: 45px;
    display: flex;
    padding: 13px 0;
    cursor: pointer;`;
const Image = styled('img') ({
    width: 50,

```

```

    objectFit: 'cover',
    borderRadius: '50%',
    padding: '0 14px'
  });

const Container = styled(Box)`
  display: flex;`;
const Timestamp = styled(Typography)`
  font-size: 12px;
  margin-left: auto;
  color: #00000099;
  margin-right: 20px;`;
const Text = styled(Typography)`
  display: block;
  color: rgba(0, 0, 0, 0.6);
  font-size: 14px;`;

const Conversation = ({ user }) => {
  const url = user.picture || emptyProfilePicture;
  const { setPerson } = useContext(UserContext);
  const { account, newMessageFlag } = useContext(AccountContext);
  const [message, setMessage] = useState({});
  useEffect(() => {
    const getConversationMessage = async () => {
      const data = await getConversation({ senderId: account.sub,
receiverId: user.sub });
      setMessage({ text: data?.message, timestamp: data?.updatedAt
});
    }
    getConversationMessage();
  }, [newMessageFlag]);

  const getUser = async () => {
    setPerson(user);
    await setConversation({ senderId: account.sub, receiverId:
user.sub });
  }
}

```

```

    <Component onClick={() => getUser()}>
      <Box>
        <Image src={url} alt="display picture" />
      </Box>
      <Box style={{width: '100%'}}>
        <Container>
          <Typography>{user.name}</Typography>
          {
            message?.text &&
<Timestamp>{formatDate(message?.timestamp)}</Timestamp>
          }
        </Container>
        <Box>
          <Text>{message?.text?.includes('localhost') ? 'media'
: message.text}</Text>
        </Box>
      </Box>
    </Component>
  )}
export default Conversation;

```

Source Code of client/src/components/menu/Header.jsx

```

import { useContext, useState } from 'react';
import { Box, styled } from '@mui/material';
import { Chat as MessageIcon } from '@mui/icons-material';
import { AccountContext } from '../../../../context/AccountProvider';
import HeaderMenu from './HeaderMenu';
import InfoDrawer from '../../../../drawer/Drawer';
const Component = styled(Box) `
  height: 44px;
  background: #ededed;
  display: flex;
  padding: 8px 16px;
  align-items: center;`;
const Wrapper = styled(Box) `
  margin-left: auto;

```



```

& > * {
  margin-left: 2px;
  padding: 8px;
  color: #000;    };
& :first-child {
  font-size: 22px;
  margin-right: 8px;
  margin-top: 3px;    }`;
const Image = styled('img') ({
  height: 40,
  width: 40,
  borderRadius: '50%'})
const Header = () => {
  const [openDrawer, setOpenDrawer] = useState(false);
  const { account } = useContext(AccountContext);
  const toggleDrawer = () => {
    setOpenDrawer(true);    }
  return (
    <>
      <Component>
        <Image src={account.picture} onClick={() =>
toggleDrawer()} />
        <Wrapper>
          <MessageIcon />
          <HeaderMenu/>
        </Wrapper>
      </Component>
      <InfoDrawer open={openDrawer} setOpen={setOpenDrawer}
profile={true} />
    </>    )}
export default Header;

```

Source Code of client/src/components/menu/HeaderMenu.jsx

```

import { useState, useContext } from 'react';
import { MoreVert } from '@mui/icons-material';
import { Menu, MenuItem, styled } from '@mui/material';

```

```

import { googleLogout } from '@react-oauth/google';
import { AccountContext } from '../../../context/AccountProvider';
import { UserContext } from '../../../context/UserProvider';
import { clientId } from '../../../constants/data';
import InfoDrawer from '../../../drawer/Drawer';
const MenuOption = styled(MenuItem) `
  font-size: 14px
  padding: 15px 60px 5px 24px;
  color: #4A4A4A;`;
const Logout = styled(googleLogout) `
  border: none;
  box-shadow: none;`;
const HeaderMenu = () => {
  const [open, setOpen] = useState(false);
  const [openDrawer, setOpenDrawer] = useState(false);
  const { setAccount, setShowloginButton, showlogoutButton,
setShowlogoutButton } = useContext(AccountContext);
  const { setPerson } = useContext(UserContext);
  const handleClick = (event) => {
    setOpen(event.currentTarget);    };
  const handleClose = () => {
    setOpen(null);    };
  const onSignoutSuccess = () => {
    alert("You have been logged out successfully");
    console.clear();
    setShowlogoutButton(false);
    setShowloginButton(true);
    setAccount('');
    setPerson({});    };
  const toggleDrawer = () => {
    setOpenDrawer(true);    }
  return (
    <>
      <MoreVert onClick={handleClick} />
      <Menu
        anchorEl={open}

```

```

        keepMounted
        open={open}
        onClose={handleClose}
        getContentAnchorEl={null}
        anchorOrigin={{
          vertical: 'bottom',
          horizontal: 'center',
        }}
        transformOrigin={{
          vertical: 'top',
          horizontal: 'right',
        }}
      >
      <MenuOption onClick={() => { handleClose();
toggleDrawer() }}>Profile</MenuOption>
      <MenuOption onClick={() => { handleClose(); }}>
      /* { showlogoutButton ?
        <Logout
          clientId={clientId}
          buttonText="Logout"
          onLogoutSuccess={onSignoutSuccess}
        >
        </Logout> : null
      } */
      </MenuOption>
    </Menu>
    <InfoDrawer open={openDrawer} setOpen={setOpenDrawer}
profile={true} />
  </>
  )}
export default HeaderMenu;

```

Source Code of client/src/components/menu/Search.jsx

```

import { Box, InputBase, styled } from '@mui/material';
import { Search as SearchIcon } from '@mui/icons-material';
const Component = styled(Box)`

```

```

    background: #fff;
    height: 45px;
    display: flex;
    align-items: center;
    border-bottom: 1px solid #F2F2F2;`;

const Wrapper = styled(Box)`
    position: relative;
    border-radius: 10px;
    background-color: #f0f2f5;
    margin: 0 13px;
    width: 100%;
`;

const Icon = styled(Box)`
    color: #919191;
    padding: 8px;
    height: 100%;
    position: absolute;`;

const InputField = styled(InputBase)`
    width: 100%;
    padding: 16px;
    padding-left: 65px;
    font-size: 14px;
    height: 15px;
    width: 100%;`;

const Search = ({ setText }) => {
    return (
        <Component>
            <Wrapper>
                <Icon>
                    <SearchIcon fontSize="small"/>
                </Icon>
                <InputField
                    placeholder="Search or start new chat"
                    inputProps={{ 'aria-label': 'search' }}
                    onChange={(e) => setText(e.target.value)}
                />
            </Wrapper>
        </Component>
    )
}
```

```

        />
      </Wrapper>
    </Component>
  )
}
export default Search;

```

Source Code of client/src/components/menu/ChatDialog.jsx

```

import { useContext } from 'react';
import { Dialog, styled, Box } from '@mui/material';
import { UserContext } from '../../../context/UserProvider';
import Menu from './menu/Menu';
import ChatBox from './chat/ChatBox';
import EmptyChat from './chat/EmptyChat';
const Component = styled(Box) `
  display: flex;`;
const LeftComponent = styled(Box) `
  min-width: 450px;`;
const RightComponent = styled(Box) `
  width: 73%;
  min-width: 300px;
  height: 100%;
  border-left: 1px solid rgba(0, 0, 0, 0.14);`;
const dialogStyle = {
  height: '95%',
  width: '100%',
  margin: '20px',
  maxWidth: '100%',
  maxHeight: '100%',
  borderRadius: 0,
  boxShadow: 'none',
  overflow: 'hidden'};
const ChatDialog = () => {
  const { person } = useContext(UserContext);
  return (

```

```

    <Dialog
      open={true}
      BackdropProps={{style: {backgroundColor: 'unset'}}}
      PaperProps={{ sx: dialogStyle }}
      maxWidth={'md'}
    >
      <Component>
        <LeftComponent>
          <Menu/>
        </LeftComponent>
        <RightComponent>
          {
            Object.keys(person).length ? <ChatBox/> :
EmptyChat />
          }
        </RightComponent>
      </Component>
    </Dialog>
  )}
export default ChatDialog;

```

Source Code of client/src/components/drawer/Drawer.jsx

```

import { styled, Drawer, Box, Typography } from '@mui/material';
import { ArrowBack } from '@mui/icons-material';
import Profile from './Profile';
const Header = styled(Box) `
  background: #008069;
  height: 107px;
  color: #FFFFFF;
  display: flex;
  & > svg, & > p {
    margin-top: auto;
    padding: 15px;
    font-weight: 600;`;
const Component = styled(Box) `

```

```

    background: #ededed;
    height: 85%;`
const Text = styled(Typography)`
    font-size: 18px;`
const drawerStyle = {
    left: 20,
    top: 17,
    height: '95%',
    width: '30%',
    boxShadow: 'none'}
const InfoDrawer = ({ open, setOpen, profile }) => {
    const handleClose = () => {
        setOpen(false);
    };
    return (
        <Drawer
            open={open}
            onClose={handleClose}
            PaperProps={{ sx: drawerStyle }}
            style={{ zIndex: 1500 }}
        >
            <Header>
                <ArrowBack onClick={() => setOpen(false)} />
                <Text>Profile</Text>
            </Header>
            <Component>
                {profile && <Profile />}
            </Component>
        </Drawer>    );}
export default InfoDrawer;

```

Source Code of client/src/components/drawer/Profile.jsx

```

import { useContext } from "react";
import { Box, styled, Typography } from "@mui/material"
import { AccountContext } from "../../context/AccountProvider";

```

```

const ImageContainer = styled(Box)
  display: flex;
  justify-content: center;`;
const Image = styled('img') ({
  width: 200,
  height: 200,
  borderRadius: '50%',
  padding: '25px 0'
});
const BoxWrapper = styled(Box)`
  background: #FFFFFF;
  padding: 12px 30px 2px;
  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.08);
  & :first-child {
    font-size: 13px;
    color: #009688;
    font-weight: 200;    };
  & :last-child {
    margin: 14px 0;
    color: #4A4A4A;    }`;
const DescriptionContainer = styled(Box)`
  padding: 15px 20px 28px 30px;
  & > p {
    color: #8696a0;
    font-size: 13px;    }`;
const Profile = () => {
  const { account } = useContext(AccountContext);
  return (
    <>
      <ImageContainer>
        <Image src={account.picture} alt="displaypicture" />
      </ImageContainer>
      <BoxWrapper>
        <Typography>Your name</Typography>
        <Typography>{account.name}</Typography>
    </>
  )
}

```



```

        </BoxWrapper>
        <DescriptionContainer>
          <Typography>This is not your username or pin. This name
will be visible to your WhatsApp contacts.</Typography>
        </DescriptionContainer>
        <BoxWrapper>
          <Typography>About</Typography>
          <Typography>Eat! Sleep! Code! Repeat</Typography>
        </BoxWrapper>
      </>
    )}
  export default Profile;

```

Source Code of client/src/components/loader/Messenger.jsx

```

import { useContext } from 'react';
import { AppBar, Toolbar, styled, Box } from '@mui/material';
import { AccountContext } from '../../context/AccountProvider';
import ChatDialog from '../../chat/ChatDialog';
import LoginDialog from '../../account/LoginDialog';
const Component = styled(Box) `
  height: 100vh;
  background: #DCDCDC;`;
const Header = styled(AppBar) `
  background-color: #00A884;
  height: 125px;
  box-shadow: none;`;
const LoginHeader = styled(AppBar) `
  background: #00bfa5;
  height: 200px;
  box-shadow: none;`;
const Messenger = () => {
  const { account } = useContext(AccountContext);
  return ( <Component> { account ?
    <>
      <Header>
        <Toolbar></Toolbar>

```

```

        </Header>
        <ChatDialog />
    </>
    :
    <>
        <LoginHeader>
            <Toolbar></Toolbar>
        </LoginHeader>
        <LoginDialog />
    </> }
</Component>    )}
export default Messenger;

```

Source Code of client/src/components/context/AccountProvider.jsx

```

import { createContext, useState, useEffect, useRef } from 'react';
import { io } from 'socket.io-client';
export const AccountContext = createContext(null);
const AccountProvider = ({children}) => {
    const [ account, setAccount ] = useState();
    const [showloginButton, setShowloginButton] = useState(true);
    const [showlogoutButton, setShowlogoutButton] = useState(false);
    const [activeUsers, setActiveUsers] = useState([]);
    const [newMessageFlag, setNewMessageFlag] = useState(false);
    const socket = useRef();
    useEffect(() => {
        socket.current = io('ws://localhost:9000'); }, [])
    return (
        <AccountContext.Provider value={{
            account,
            setAccount,
            showloginButton,
            setShowloginButton,
            showlogoutButton,
            setShowlogoutButton,
            socket,

```

```

        activeUsers,
        setActiveUsers,
        newMessageFlag,
        setNewMessageFlag }}>
        {children}
    </AccountContext.Provider>
  )}
export default AccountProvider;

```

Source Code of client/src/components/context/UserProvider.jsx

```

import { createContext, useState } from 'react';
export const UserContext = createContext(null);
const UserProvider = ({children}) => {
  const [ person, setPerson ] = useState({});
  return (
    <UserContext.Provider value={{ person, setPerson }}>
      {children}
    </UserContext.Provider>
  )}
export default UserProvider;

```

Source Code of client/src/components/Service/Api.js

```

import axios from 'axios';
const url = 'http://localhost:8000';
export const addUser = async (data) => {
  try {
    let response = await axios.post(`${url}/add`, data);
    return response.data;
  } catch (error) {
    console.log('Error while calling addUser API ', error);
  }
}
export const getUsers = async () => {
  try {

```

```
        let response = await axios.get( `${url}/users` );
        return response.data
      } catch (error) {
        console.log('Error while calling getUsers API ', error);
      }
    }
  }
export const setConversation = async (data) => {
  try {
    await axios.post(`${url}/conversation/add`, data);
  } catch (error) {
    console.log('Error while calling setConversation API ', error);
  }
}
export const getConversation = async (users) => {
  try {
    let response = await axios.post(`${url}/conversation/get`, users);
    return response.data;
  } catch (error) {
    console.log('Error while calling getConversation API ', error);
  }
}
export const getMessages = async (id) => {
  try {
    let response = await axios.get(`${url}/message/get/${id}`);
    return response.data
  } catch (error) {
    console.log('Error while calling getMessages API ', error);
  }
}
export const newMessages = async (data) => {
  try {
    return await axios.post(`${url}/message/add`, data);
  } catch (error) {
    console.log('Error while calling newConversations API ', error);
  }
}
```

```
}  
export const uploadFile = async (data) => {  
  try {  
    return await axios.post(`${url}/file/upload`, data);  
  } catch (error) {  
    console.log('Error while calling newConversations API ', error);  
  }  
}
```

BACK-END

Source Code of server/index.js

```
import express from 'express';  
import bodyParser from 'body-parser';  
import cors from 'cors';  
import dotenv from 'dotenv';  
import Connection from './database/db.js';  
import Routes from './routes/Routes.js';  
const app = express();  
const PORT = 8000;  
const username = process.env.DB_USERNAME;  
const password = process.env.DB_PASSWORD;  
Connection(username, password);  
app.listen(PORT, () => console.log(`Server is running successfully on PORT  
${PORT}`));  
app.use(bodyParser.json({ extended: true }));  
app.use(bodyParser.urlencoded({ extended: true }));  
app.use(cors());  
app.use('/', Routes);
```

Source Code of server/controller/conversation-controller.js

```
import Conversation from "../modal/Conversation.js";  
export const newConversation = async (request, response) => {  
  let senderId = request.body.senderId;
```

```
    let receiverId = request.body.receiverId;

    const exist = await Conversation.findOne({ members: { $all:
[receiverId, senderId]  }})

    if(exist) {

        response.status(200).json('conversation already exists');

        return;

    }

    const newConversation = new Conversation({

        members: [senderId, receiverId]    });

    try {

        const savedConversation = await newConversation.save();

        response.status(200).json(savedConversation);

    } catch (error) {

        response.status(500).json(error);

    }}

export const getConversation = async (request, response) => {

    try {

        const conversation = await Conversation.findOne({ members: { $all:
[ request.body.senderId, request.body.receiverId]  }});

        response.status(200).json(conversation);

    } catch (error) {

        response.status(500).json(error);

    }}

}
```

Source Code of server/controller/image-controller.js

```
import grid from 'gridfs-stream';
import mongoose from 'mongoose';
const url = 'http://localhost:8000';
let gfs, gridfsBucket;
const conn = mongoose.connection;
conn.once('open', () => {

    gridfsBucket = new mongoose.mongo.GridFSBucket(conn.db, {

        bucketName: 'fs'

    });

    gfs = grid(conn.db, mongoose.mongo);
    gfs.collection('fs');});
```

```
export const uploadImage = (request, response) => {
  if(!request.file)
    return response.status(404).json("File not found");
  const imageUrl = `${url}/file/${request.file.filename}`;
  response.status(200).json(imageUrl);
}
export const getImage = async (request, response) => {
  try {
    const file = await gfs.files.findOne({ filename:
request.params.filename });
    const readStream = gridfsBucket.openDownloadStream(file._id);
    readStream.pipe(response);
  } catch (error) {
    response.status(500).json({ msg: error.message });
  }
}
```

Source Code of server/controller/message-controller.js

```
import Message from "../modal/Message.js";
import Conversation from '../modal/Conversation.js';
export const newMessage = async (request, response) => {
  const newMessage = new Message(request.body);
  try {
    await newMessage.save();
    await Conversation.findByIdAndUpdate(request.body.conversationId,
{ message: request.body.text });
    response.status(200).json("Message has been sent successfully");
  } catch (error) {
    response.status(500).json(error);
  }
}
export const getMessage = async (request, response) => {
  try {
    const messages = await Message.find({ conversationId:
request.params.id });
    response.status(200).json(messages);
  } catch (error) {
    response.status(500).json(error);
  }
}
```

Source Code of server/controller/user-controller.js

```
import User from "../modal/User.js";
export const addUser = async (request, response) => {
  try {
    let exist = await User.findOne({ sub: request.body.sub });
    if(exist) {
      response.status(200).json('user already exists');
      return;
    }
    const newUser = new User(request.body);
    await newUser.save();
    response.status(200).json(newUser);
  } catch (error) {
    response.status(500).json(error);
  }
}
export const getUser = async (request, response) => {
  try {
    const user = await User.find({});
    response.status(200).json(user);
  } catch (error) {
    response.status(500).json(error);
  }
}
```

Source Code of server/modal/Conversation.js

```
import mongoose from 'mongoose';
const ConversationSchema = new mongoose.Schema({
  members: { type: Array },
  message: { type: String },
  { timestamps: true });
const conversation = mongoose.model('Conversation', ConversationSchema);
export default conversation;
```

Source Code of server/modal/Message.js

```
import mongoose from 'mongoose';
```



```
const MessageSchema = new mongoose.Schema({
  conversationId: {      type: String    },
  senderId: {          type: String    },
  receiverId: {        type: String    },
  text: {              type: String    },
  type: {              type: String    }
},
{
  timestamps: true})
const message = mongoose.model('Message', MessageSchema);
export default message;
```

Source Code of server/modal/User.js

```
import mongoose from 'mongoose';
const userSchema = new mongoose.Schema({
  aud: {      type: String,
    required: true  },
  azp: {      type: String,
    required: true  },
  email: {    type: String,
    required: true  },
  email_verified: {
    type: Boolean,
    required: true
  },
  exp: {      type: Number,
    required: true  },
  family_name: {      type: String,
    required: true  },
  given_name: {       type: String,
    required: true  },
  iat: {         type: Number,
    required: true  },
  iss: {         type: String,
    required: true  },
  jti: {         type: String,
```

```

        required: true    },
    name: {                type: String,
        required: true    },
    nbf: {                 type: Number,
        required: true    },
    picture: {             type: String,
        required: true    },
    sub: {                 type: String,
        required: true    }}}
const user = mongoose.model('user', userSchema);
export default user;

```

Source Code of server/Routes/Routes.js

```

import express from 'express';
import { newConversation, getConversation } from
'../controller/conversation-controller.js';
import { addUser, getUser } from '../controller/user-controller.js';
import { newMessage, getMessage } from
'../controller/message-controller.js';
import { uploadImage, getImage } from '../controller/image-controller.js';
import upload from '../utils/upload.js';
const route = express.Router();
route.post('/add', addUser);
route.get('/users', getUser);
route.post('/conversation/add', newConversation);
route.post('/conversation/get', getConversation);
route.post('/message/add', newMessage);
route.get('/message/get/:id', getMessage);
route.post('/file/upload', upload.single('file'), uploadImage);
route.get('/file/:filename', getImage);
export default route;

```

Source Code of socket/index.js

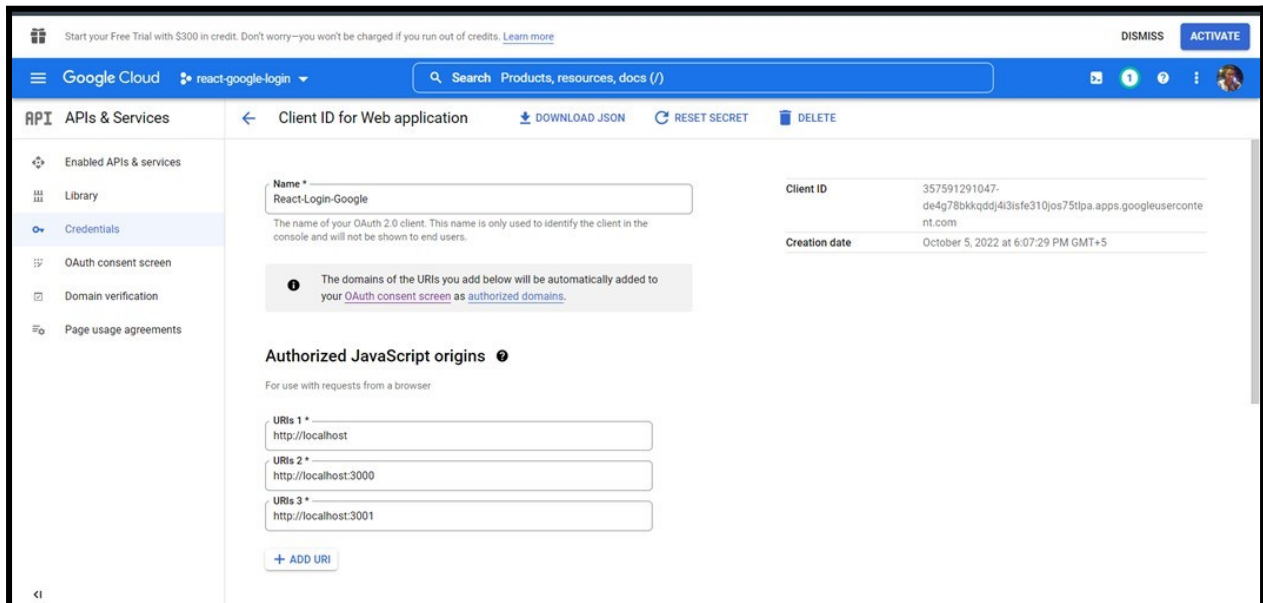
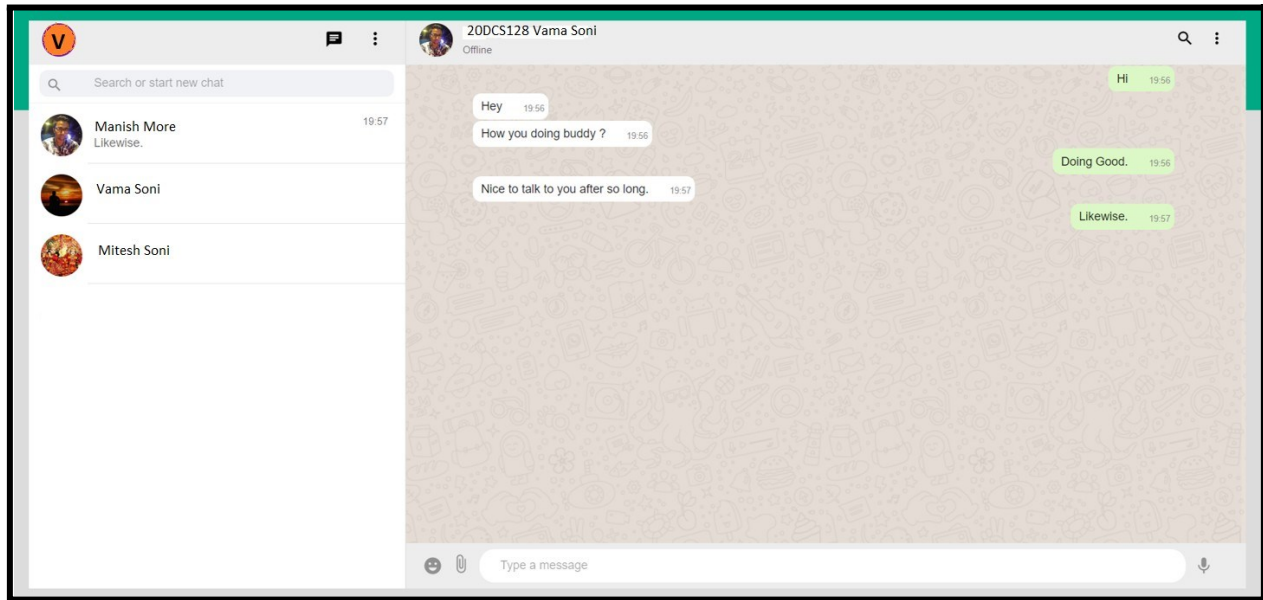
```

import { Server } from 'socket.io';
const io = new Server(9000, {

```

```
    cors: {
      origin: 'http://localhost:3000',
    },
  })
let users = [];
const addUser = (userData, socketId) => {
  !users.some(user => user.sub === userData.sub) && users.push({
    ...userData, socketId });
}
const removeUser = (socketId) => {
  users = users.filter(user => user.socketId !== socketId);
const getUser = (userId) => {
  return users.find(user => user.sub === userId);
}
io.on('connection', (socket) => {
  console.log('user connected')
  socket.on("addUser", userData => {
    addUser(userData, socket.id);
    io.emit("getUsers", users);
  })
  socket.on('sendMessage', (data) => {
    const user = getUser(data.receiverId);
    io.to(user.socketId).emit('getMessage', data)
  })
  socket.on('disconnect', () => {
    console.log('user disconnected');
    removeUser(socket.id);
    io.emit('getUsers', users);  })
}
```

OUTPUT:



CONCLUSION:

Through this practical we learn to implement a WhatsApp clone using MERN Stack