PRACTICAL 3

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 React from 'react';
import './App.css';
import Sidebar from './Sidebar';
import Home from './Home/Home';
import Desserts from './Desserts/Desserts';
import Pizzas from './Pizzas/Pizzas';
import Salads from './Salads/Salads';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
function App() {
  return (
    <div className="App" id="outer-container">

      <BrowserRouter>
        <Sidebar />
        <div id="page-wrap">
          <h1>Cool Restaurant</h1>
          <h2>Check out our offerings in the sidebar!</h2>
        </div>
        <Routes>
          <Route exact path='/' element={<Home />}>
          </Route>
          <Route exact path='/Desserts' element={<Desserts />}>
          </Route>
          <Route exact path='/Pizzas' element={<Pizzas />}>
          </Route>
          <Route exact path='/Salads' element={<Salads />}>
          </Route>

        </Routes>
      </BrowserRouter>
    </div>
  );
}

export default App;
```

Source code of Home.js and Home.css: -

```
import React from 'react';
export default function Home() {
   return <h2>Home</h2>;
}
```

Source code of Salad.js and Salads.css: -

```
import React from 'react';

export default function Salads() {
   return <h2>Salads</h2>;
}
```

Source code of Pizzas.js and Pizzas.css: -

```
import React from 'react';

export default function Pizzas() {
   return <h2>Pizzas</h2>;
}
```

Source code of Pizzas.js and Pizzas.css: -

```
import React from 'react';

export default function Desserts() {
   return <h2>Desserts</h2>;
}
```

Source code of Sidebar.js and Sidebar.css: -

```
import React, {useState} from 'react';
import { elastic as Menu } from 'react-burger-menu';
import './Sidebar.css';
import {
   NavLink,
} from './NavbarElements';

export default function Sidebar() {
   const [Sidebar, setSidebar] = useState(false);
   const showSidbar = () => setSidebar(!Sidebar);
   //const [Count, setCount] = useState(0);
   return (
      <Menu>
         <NavLink to='/' onclick={() => setSidebar(!Sidebar)}>
            Home
         </NavLink>
         <NavLink to='/Desserts' onclick={() => setSidebar(!Sidebar)}>
            Desserts
         </NavLink>
         <NavLink to='/Pizzas' onclick={() => setSidebar(!Sidebar)}>
```

```
                Pizzas
            </NavLink>
            <NavLink to='/Salads' onclick={() => setSidebar(!Sidebar)}>
                Salads
            </NavLink>
            {/* <div>
                <button onClick={() => setCount(Count + 1)}>
                    Click here
                </button>
                Clicked {Count} times!
            </div> */}
        </Menu>
    );
};

-----------------------------------------------------------------------------------------------

/* Position and sizing of burger button */
.bm-burger-button {
    position: fixed;
    width: 36px;
    height: 30px;
    left: 36px;
    top: 36px;
  }

  /* Color/shape of burger icon bars */
  .bm-burger-bars {
    background: #373a47;
  }

  /* Color/shape of burger icon bars on hover*/
  .bm-burger-bars-hover {
    background: #a90000;
  }

  /* Position and sizing of clickable cross button */
  .bm-cross-button {
    height: 24px;
    width: 24px;
  }

  /* Color/shape of close button cross */
  .bm-cross {
    background: #bdc3c7;
  }

  /*
  Sidebar wrapper styles
  Note: Beware of modifying this element as it can break the animations - you should not need
to touch it in most cases
```

```
  */
  .bm-menu-wrap {
    position: fixed;
    height: 100%;
  }

  /* General sidebar styles */
  .bm-menu {
    background: #373a47;
    padding: 2.5em 1.5em 0;
    font-size: 1.15em;
  }

  /* Morph shape necessary with bubble or elastic */
  .bm-morph-shape {
    fill: #373a47;
  }

  /* Wrapper for item list */
  .bm-item-list {
    color: #b8b7ad;
    padding: 0.8em;
  }

  /* Individual item */
  /* Individual item */
.bm-item {
    display: inline-block;

    color: #d1d1d1;
    margin-bottom: 10px;
    text-align: left;
    text-decoration: none;
    transition: color 0.2s;
  }

  .bm-item:hover {
    color: #ffffff;
  }

  /* Styling of overlay */
  .bm-overlay {
    background: rgba(0, 0, 0, 0.3);
  }
```

Source code of NavbarElements.js: -

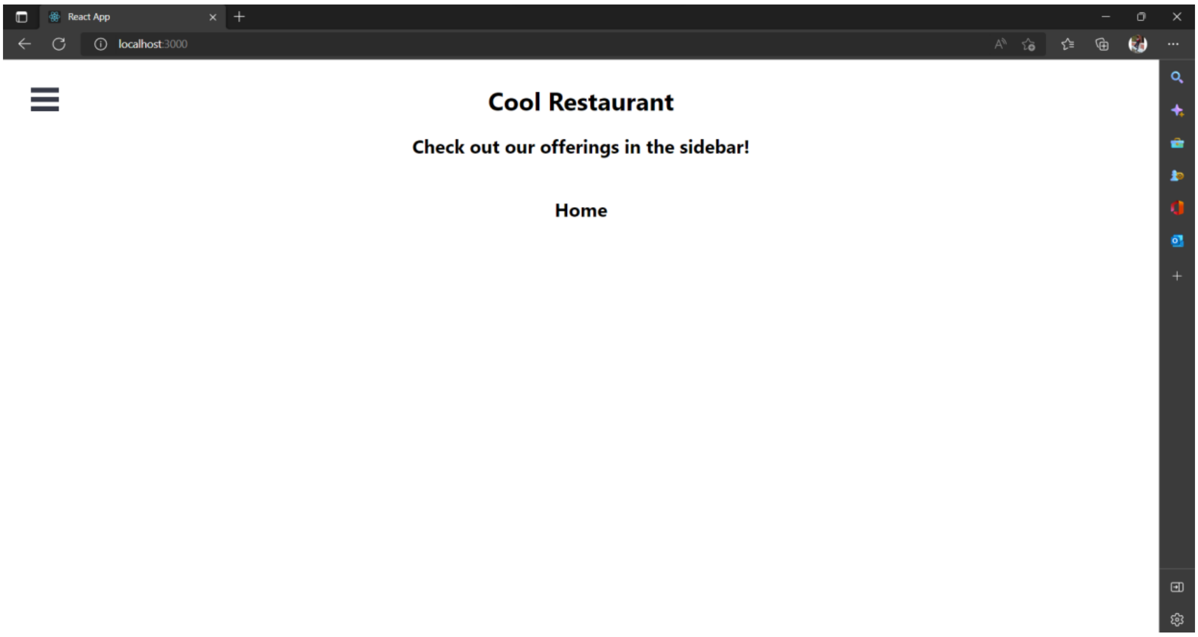
```
import { FaBars } from 'react-icons/fa';
import { NavLink as Link } from 'react-router-dom';
import styled from 'styled-components';
```

```
export const Nav = styled.nav`
background: #63D471;
height: 85px;
display: flex;
justify-content: space-between;
padding: 0.2rem calc((100vw - 1000px) / 2);
z-index: 12;
/* Third Nav */
/* justify-content: flex-start; */
`;

export const NavLink = styled(Link)`
color: white;
display: flex;
align-items: center;
text-decoration: none;
padding: 0 1rem;
height: 20px;
cursor: pointer;
&.active {
	color: grey;
}
`;

export const Bars = styled(FaBars)`
display: none;
color: #808080;
@media screen and (max-width: 768px) {
	display: block;
	position: absolute;
	top: 0;
	right: 0;
	transform: translate(-100%, 75%);
	font-size: 1.8rem;
	cursor: pointer;
}
`;

export const NavMenu = styled.div`
display: flex;
align-items: center;
margin-right: -24px;
/* Second Nav */
/* margin-right: 24px; */
/* Third Nav */
/* width: 100vw;
white-space: nowrap; */
@media screen and (max-width: 768px) {
	display: none;
}
`;
```

```
export const NavBtn = styled.nav`
display: flex;
align-items: center;
margin-right: 24px;
/* Third Nav */
/* justify-content: flex-end;
width: 100vw; */
@media screen and (max-width: 768px) {
	display: none;
}
`;

export const NavBtnLink = styled(Link)`
border-radius: 4px;
background: #808080;
padding: 10px 22px;
color: #000000;
outline: none;
border: none;
cursor: pointer;
transition: all 0.2s ease-in-out;
text-decoration: none;
/* Second Nav */
margin-left: 24px;
&:hover {
	transition: all 0.2s ease-in-out;
	background: #fff;
	color: #808080;
}
`;
```

Output Screenshot: -

Home Page:-

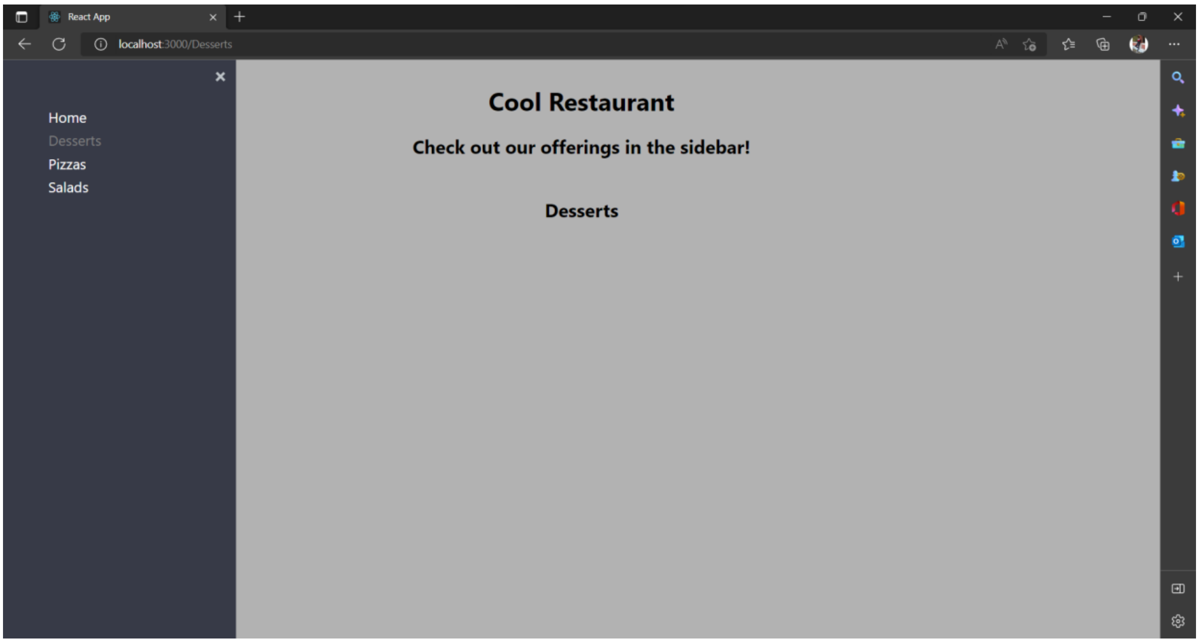


Cool Restaurant

Check out our offerings in the sidebar!

Home

Salads Page:-

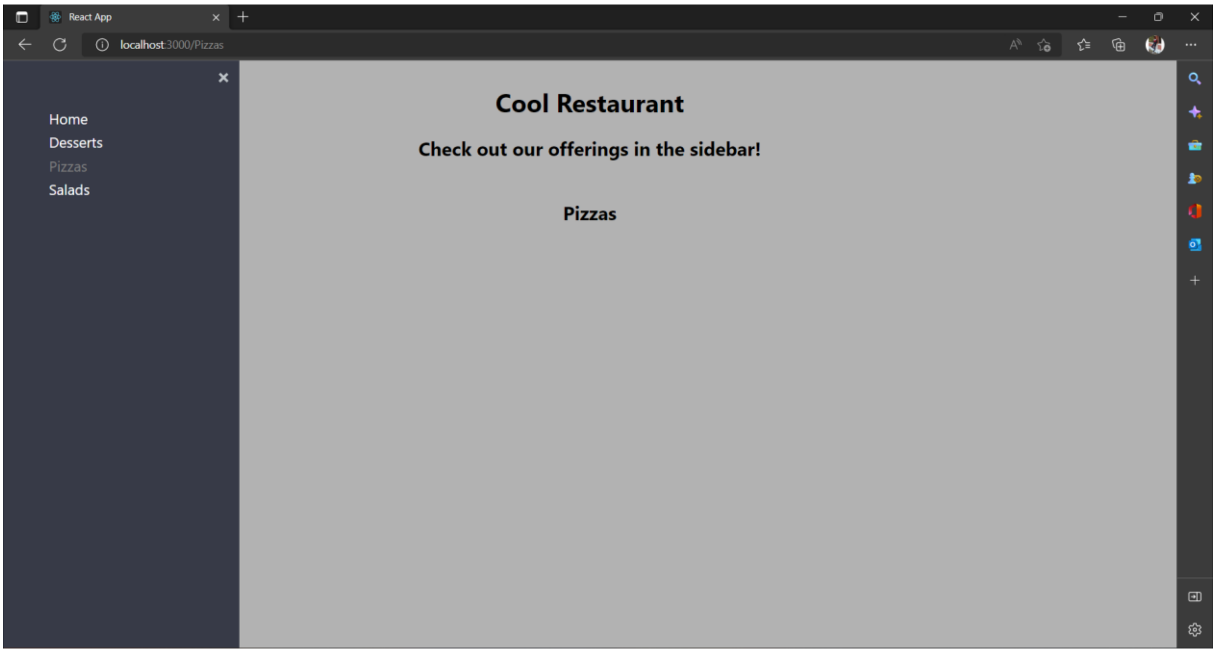


Cool Restaurant

Check out our offerings in the sidebar!

Desserts

Pizzas Page:-

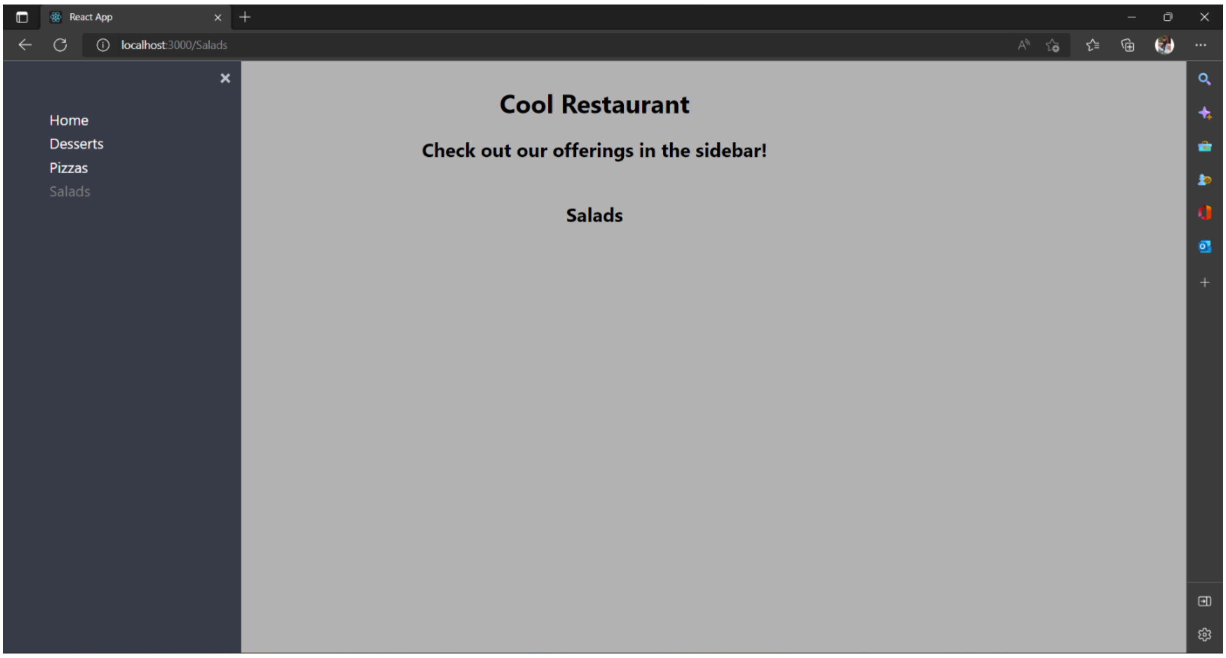


Cool Restaurant

Check out our offerings in the sidebar!

Pizzas

Desserts Page:-



Cool Restaurant

Check out our offerings in the sidebar!

Salads

PRACTICAL 4

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 Todo.js: -

```
import React, { useState } from "react";
import TodoForm from "./TodoForm";
import { RiCloseCircleLine } from "react-icons/ri";
import { TiEdit } from "react-icons/ti";

const Todo = ({ todos, completeTodo, removeTodo, updateTodo }) => {
  const [edit, setEdit] = useState({
    id: null,
    value: "",
  });

  const submitUpdate = (value) => {
    updateTodo(edit.id, value);
    setEdit({
      id: null,
      value: "",
    });
  };
  if (edit.id) {
    return <TodoForm edit={edit} onSubmit={submitUpdate} />;
  }

  return todos.map((todo, index) => (
    <div
      className={todo.isComplete ? "todo-row complete" : "todo-row"}
      key={index}
    >
      <div key={todo.id} onClick={() => completeTodo(todo.id)}>
        {todo.text}
      </div>
      <div className="icons">
        <RiCloseCircleLine
          onClick={() => removeTodo(todo.id)}
          className="delete-icon"
        />
        <TiEdit
          onClick={() => setEdit({ id: todo.id, value: todo.text })}
          className="edit-icon"
        />
      </div>
    </div>
  ));
};

export default Todo;
```

Source code of TodoForm.js:-

```
import React, { useState, useEffect, useRef } from "react";

function TodoForm(props) {
  const [input, setInput] = useState(props.edit ? props.edit.value : "");

  const inputRef = useRef(null);

  useEffect(() => {
    inputRef.current.focus();
  });

  const handleChange = (e) => {
    setInput(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();

    props.onSubmit({
      id: Math.floor(Math.random() * 10000),
      text: input,
    });
    setInput("");
  };

  return (
    <form onSubmit={handleSubmit} className="todo-form">
      {props.edit ? (
        <>
          <input
            placeholder="Update your item"
            value={input}
            onChange={handleChange}
            name="text"
            ref={inputRef}
            className="todo-input edit"
          />
          <button onClick={handleSubmit} className="todo-button edit">
            Update
          </button>
        </>
      ) : (
        <>
          <input
            placeholder="Add a todo"
            value={input}
            onChange={handleChange}
            name="text"
            className="todo-input"
            ref={inputRef}
          />
          <button onClick={handleSubmit} className="todo-button">
            Add todo
          </button>
        </>
      )}
    </form>
  );
}

export default TodoForm;
```

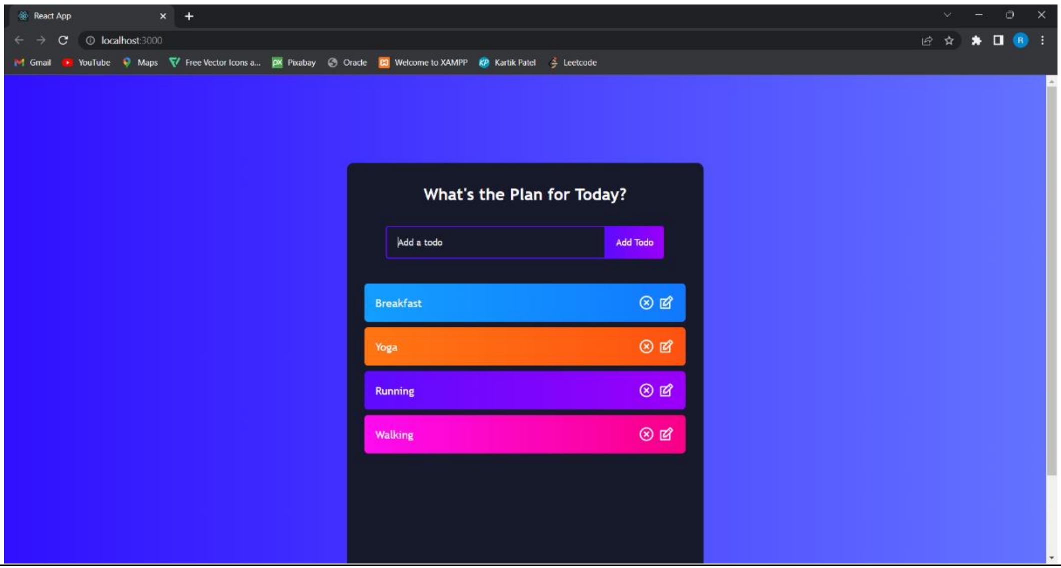
Source code of TodoList.js:-

```
import React, { useState } from "react";
import TodoForm from "./TodoForm";
import Todo from "./Todo";

function TodoList() {
 const [todos, setTodos] = useState([]);

 const addTodo = (todo) => {
  if (!todo.text || /^\s*$/.test(todo.text)) {
   return;
  }

  const newTodos = [todo, ...todos];

  setTodos(newTodos);
  console.log(...todos);
 };

 const updateTodo = (todoId, newValue) => {
  if (!newValue.text || /^\s*$/.test(newValue.text)) {
   return;
  }

  setTodos((prev) =>
   prev.map((item) => (item.id === todoId ? newValue : item))
  );
 };

 const removeTodo = (id) => {
  const removedArr = [...todos].filter((todo) => todo.id !== id);

  setTodos(removedArr);
 };
```

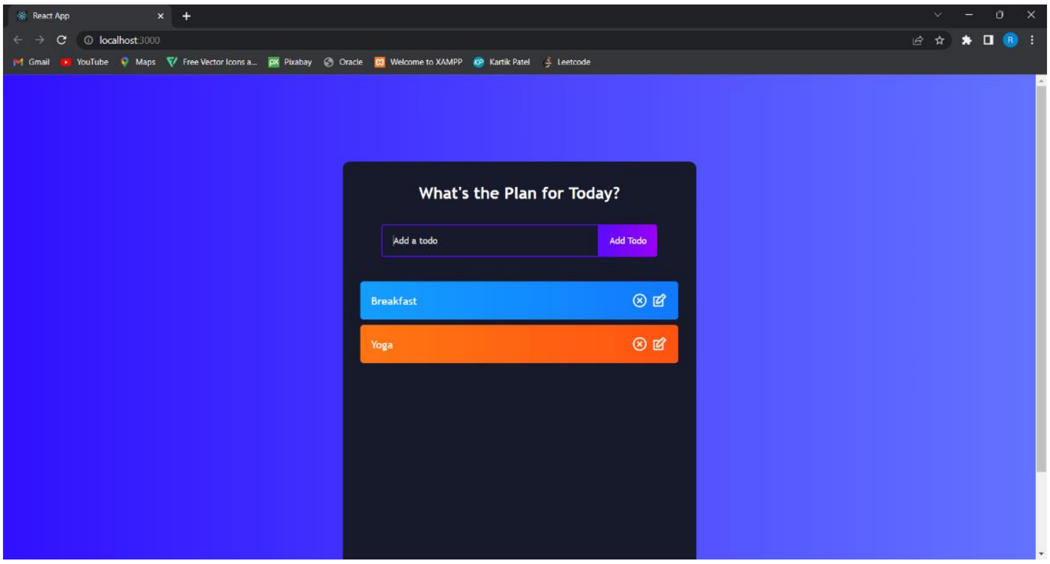
```
  const completeTodo = (id) => {
    let updatedTodos = todos.map((todo) => {
      if (todo.id === id) {
        todo.isComplete = !todo.isComplete;
      }
      return todo;
    });
    setTodos(updatedTodos);
  };

  return (
    <>
      <h1>What's the Plan for Today?</h1>
      <TodoForm onSubmit={addTodo} />
      <Todo
        todos={todos}
        completeTodo={completeTodo}
        removeTodo={removeTodo}
        updateTodo={updateTodo}
      />
    </>
  );
}

export default TodoList;
```

<u>Output: -</u>

1) Adding todo in the list: -



2) Removing todo from the list: -



Practical-5

Aim	**Design an attractive Login and Registration form using Material-UI. Use Validations to validate the errors and prompt the errors on the webpage.**

Source code of app.js

```
import logo from './logo.svg';
import './App.css';
import Instalogin from './Instalogin/Instalogin'
import Fbreg from './Fbreg/Fbreg';
import Navbar from './Navbar/Navbar';
import {BrowserRouter,Routes,Route} from 'react-router-dom'
function App() {
  return (
    <div className="App">
    <BrowserRouter>
    <Navbar/>
    <Routes>
      <Route excat path='/' element={<Instalogin/>}></Route>
      <Route excat path='/FacebookRegistration' element={<Fbreg/>}></Route>
    </Routes>
    </BrowserRouter>
    </div>
  );
}

export default App;
```

Source code of instalogin.js

```
import React from 'react'
import { TextField } from '@mui/material'
import { Container } from '@mui/system'
import { useFormik } from 'formik'
import { Alert } from '@mui/material';
import './Instalogin.css'
import {FormControlLabel} from '@mui/material'
import{ FormControl} from '@mui/material'
import {Checkbox} from '@mui/material'
import{FormGroup} from '@mui/material'
import {Button} from '@mui/material'
import { Snackbar } from '@mui/material';
import { spacing } from '@mui/system';
export default function Form() {
  const formik = useFormik({
    initialValues:{
      name:"",
      password:"",
      checked:[]
    },
    onSubmit:(values)=>{
      console.log(values);
    }
  })

  return (
    <div className='insta'>
    <Container  maxWidth="xs">
    <h1 className='logo' >Instagram</h1>
   <form >
    <TextField id="outlined-basic" label='Phone number, username or email' name='name'
value={formik.values.name} onChange={formik.handleChange} variant="outlined"
fullWidth  InputProps={{ disableUnderline: true }}/><br></br>
    <TextField
          id="outlined-password-input"
          name="password"
          value={formik.values.password}
          label="Password"
          type="password"
          autoComplete="current-password"
          InputProps={{ disableUnderline: true }}
          fullWidth
          style={{marginTop:"10px"}}
          onChange={formik.handleChange}
        />

      <FormControlLabel control={<Checkbox name='checked'
value={formik.values.checked}/>} label="Save login info" /><br />
      <Button variant="contained"
disabled={formik.values.name==""||formik.values.password==""} id='btn' fullWidth
onClick={formik.handleSubmit} >Log in</Button>
      <Button variant="text" style={{display:"block",margin:"auto"}}><p
style={{color:"black"}}>Forgot Password?</p></Button>
      </form>
    </Container>

    </div>
  )
}
```

Source code of Fbreg.js

```
import React from 'react'
import './Fbreg.css'
import { Radio,RadioGroup } from '@mui/material'
import {FormControl} from '@mui/material'
import {FormControlLabel} from '@mui/material'
import { useFormik } from 'formik'
export default function Fbreg() {
    const formik=useFormik({
        initialValues:{
          name:"",
          Surname:"",
          mpe:"",
          password:"",
          select:[],
          male:false,
          female:false,
          other:false,
        },
        onSubmit:(values)=>{
          console.log(values);
        }
    })
  return (
    <div className='container'>
    <h1 class='text-center my-2'style={{color:"blue",fontWeight:"bolder"}}>facebook</h1>
      <div class="shadow-lg mb-5 bg-body rounded" id='shad'>
      <div class="container py-2">
      <h2 class='text-center my-2'>Create a new account</h2>
      <h4 class='text-center my-2'id='para'>It's quick and easy.</h4>
      </div>
      <hr/>
      <div className="container px-3 py-2">
      <div class="input-group mb-3" id='fname'>
      <input type="text" class="form-control" placeholder="First name" name='name' aria-
label="Username" aria-describedby="basic-addon1" />
      </div>
      <div class="input-group mb-3" id='lname'>
      <input type="text" class="form-control" placeholder="Surname"  name='Surname'aria-
label="Username"   aria-describedby="basic-addon1" />
      </div>
      <div class="input-group mb-3" >
      <input type="text" class="form-control" placeholder="Mobile number or email address"
name='mpe' aria-label="Username"  aria-describedby="basic-addon1" />
      </div>
      <div class="input-group mb-3" >
      <input type="password" class="form-control" placeholder="New password" aria-
label="Username" name='password' aria-describedby="basic-addon1" />
      </div>
      <div className="d-flex justify-content-between">
      <select class="form-select" aria-label="Default select example" id="mar1"
name='select1' >
          <option value={1} selected>1</option>
          <option value={2}>2</option>
          <option value={3}>3</option>
          <option value={4}>4</option>
          <option value={5}>5</option>
          <option value={6}>6</option>
          <option value={7}>7</option>
          <option value={8}>8</option>
          <option value={9}>9</option>
          <option value={10}>10</option>
          <option value={11}>11</option>
          <option value={12}>12</option>
          <option value={13}>13</option>
          <option value={14}>14</option>
          <option value={15}>15</option>
          <option value={16}>16</option>
          <option value={17}>17</option>
          <option value={18}>18</option>
          <option value={19}>19</option>
          <option value={20}>20</option>
          <option value={21}>21</option>
          <option value={22}>22</option>
          <option value={23}>23</option>
          <option value={24}>24</option>
          <option value={25}>25</option>
          <option value={26}>26</option>
          <option value={27}>27</option>
          <option value={28}>28</option>
          <option value={29}>29</option>
          <option value={30}>30</option>
        </select>
        <select class="form-select" aria-label="Default select example"  id="mar2"
name='select2'>
          <option value={1} selected>January</option>
          <option value={2}>February</option>
          <option value={3}>March</option>
          <option value={4}>April</option>
          <option value={5}>May</option>
          <option value={6}>June</option>
          <option value={7}>July</option>
          <option value={8}>August</option>
          <option value={9}>September</option>
          <option value={10}>October</option>
          <option value={11}>November</option>
          <option value={12}>December</option>
        </select>
        <select class="form-select" aria-label="Default select example" id="mar3"
name='select3'>
          <option value={2001} selected>2001</option>
          <option value={2002}>2002</option>
          <option value={2003}>2003</option>
          <option value={2004}>2004</option>
          <option value={2005}>2005</option>
          <option value={2006}>2006</option>
          <option value={2007}>2007</option>
          <option value={2008}>2008</option>
          <option value={2009}>2009</option>
          <option value={2010}>2010</option>
          <option value={2011}>2011</option>
          <option value={2012}>2012</option>
          <option value={2013}>2013</option>
          <option value={2014}>2014</option>
          <option value={2015}>2015</option>
          <option value={2016}>2016</option>
          <option value={2017}>2017</option>
          <option value={2018}>2018</option>
          <option value={2019}>2019</option>
          <option value={2020}>2020</option>
          <option value={2021}>2021</option>
          <option value={2022}>2022</option>
        </select>
       </div>
       <div className="d-flex justify-content-between">
       <FormControlLabel value="male" control={<Radio name='male'/>} label="Male" />
    <FormControlLabel value="female" control={<Radio name='female'/>} label="Female" className='bord'/>
    <FormControlLabel value="other" control={<Radio name='other'/>} label="Other"  />
       </div>
       <p className='sm'>People who use our service may have uploaded your contact information to Facebook.<a href="#" class="link-primary"> Learn more.</a></p>
       <p className='sm'>By clicking Sign Up, you agree to our <a href="#" class="link-primary">Terms</a>, <a href="#" class="link-primary">Privacy Policy</a> and <a href="#" class="link-primary">Cookies Policy</a>. You may receive SMS notifications from us and can opt out at any time.</p>
       <button type="button" onClick={formik.handleSubmit} class="btn btn-success  w-50 d-block mx-auto">Sign Up</button>
      <p class='text-center my-2'><a href="#">Already have an account?</a></p>

       </div>
       </div>
    </div>
  )
}
```

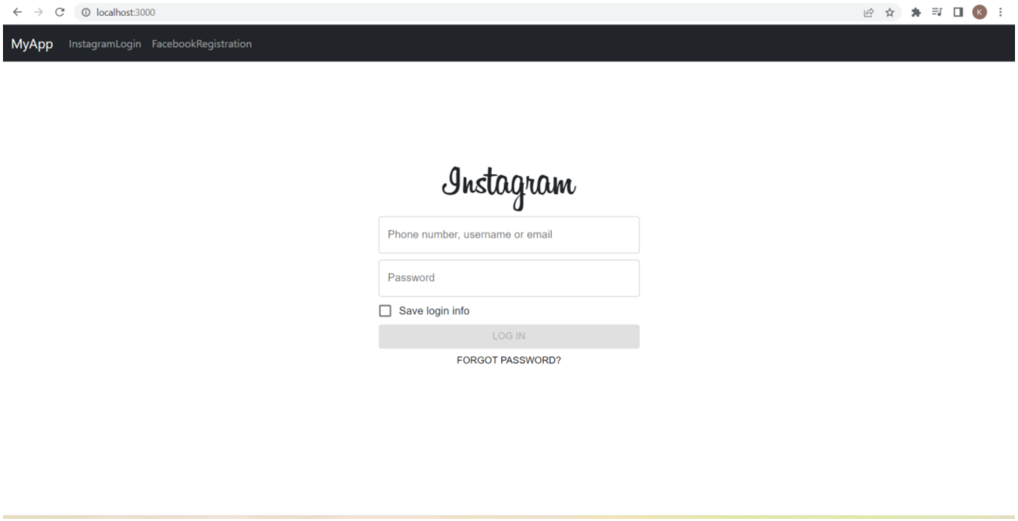
Source code of navbar.js

```
import React from 'react'
import { Link } from 'react-router-dom'
export default function Navbar() {
  return (
    <div>
        <nav class="navbar navbar-expand-lg bg-dark navbar-dark">
  <div class="container-fluid">
    <Link class="navbar-brand" to="#">Navbar</Link>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <Link class="nav-link" aria-current="page" to="/">InstagramLogin</Link>
        </li>
        <li class="nav-item">
```

```
          <Link class="nav-link" to="/FacebookRegistration">FacebookRegistration</Link>
        </li>
      </ul>
    </div>
  </nav>
      </div>
  )
}
```

Output: -

1. Output of Instagram login page: -



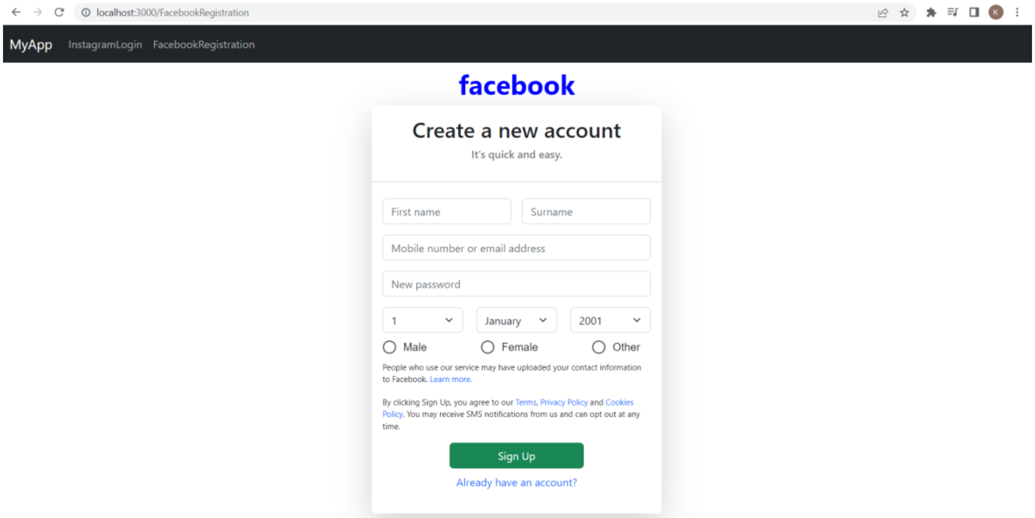
localhost:3000

Phone number, username or email

Password

□ Save login info

2. Output of Facebook registration page: -



localhost:3000/FacebookRegistration

Create a new account

It's quick and easy.

First name | Surname

Mobile number or email address

New password

1 | January | 2001

○ Male ○ Female ○ Other

People who use our service may have uploaded your contact information to Facebook. Learn more.

By clicking Sign Up, you agree to our Terms, Privacy Policy and Cookies Policy. You may receive SMS notifications from us and can opt out at any time.

Practical-6

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

Source code of db.js: -

```
const mongoose=require('mongoose');
const
mongouri="mongodb://localhost:27017/prac6?readPreference=primary&appname=MongoDB%20Compass&ssl=false";
const connectmongo=()=>{
   mongoose.connect(mongouri,()=>{
      console.log("connected to database");
   })
}
module.exports=connectmongo;
```

Source code of index.js: -

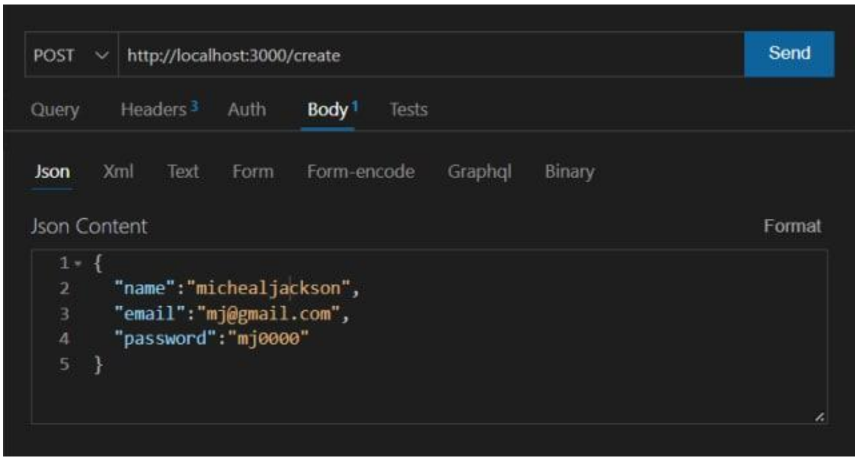
```
const mongoose=require('mongoose');
const express=require('express');
const connectmongo=require('./db');
const user=require('./mymodel');
const port=3000;
connectmongo();
const app=express();
app.use(express.json());
app.post('/create',async(req,res)=>{
   const data=await user.create({
      name:req.body.name,
      email:req.body.email,
      password:req.body.password,
   }).then(data=>res.json(data)).catch((error)=>res.send(error));
});
app.get('/read',async(req,res)=>{
   const data=await
user.findById(req.body.id).then(data=>res.send(data)).catch(error=>res.send(error));
})
app.get('/update',async(req,res)=>{
   const data=await
user.updateOne({id:req.body.id},{$set:{name:req.body.name}}).then(data=>res.json(data)).catch(error=>res.send(error));
})
app.get('/delete',async(req,res)=>{
   const data=await user.deleteOne({name:req.body.name}).then(data=>res.send("deleted
successfully")).catch(error=>res.send(error));
})
app.listen(port,()=>{
    console.log(`app listening on port ${port}`);
})
```

Source code of model.js: -

```
const mongoose=require('mongoose');
const {Schema}=mongoose;
const Userschema=new Schema({
    name:{
        type:String,
    },
    email:{
        type:String,
    },
    password:{
        type:String,
    }
})
const User=mongoose.model('user',Userschema);
module.exports=User
```

Results:-

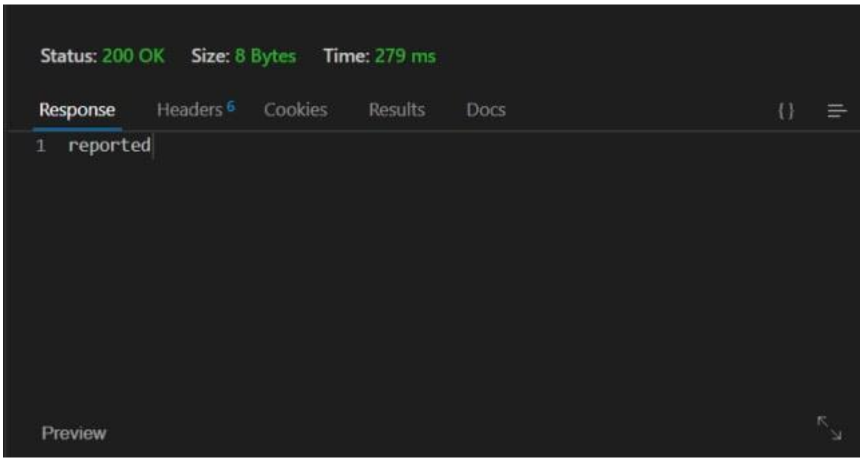
1. **Creating new user: -**



POST http://localhost:3000/create

Json Content

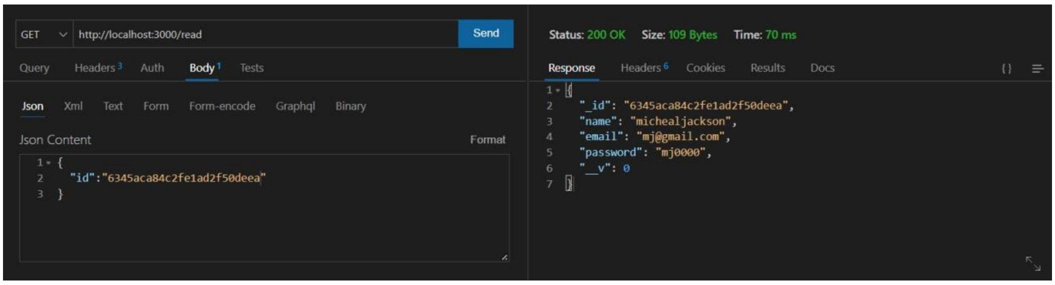
```
{
    "name":"michealjackson",
    "email":"mj@gmail.com",
    "password":"mj0000"
}
```



Status: 200 OK Size: 8 Bytes Time: 279 ms

```
reported
```

2. Reading the user data: -



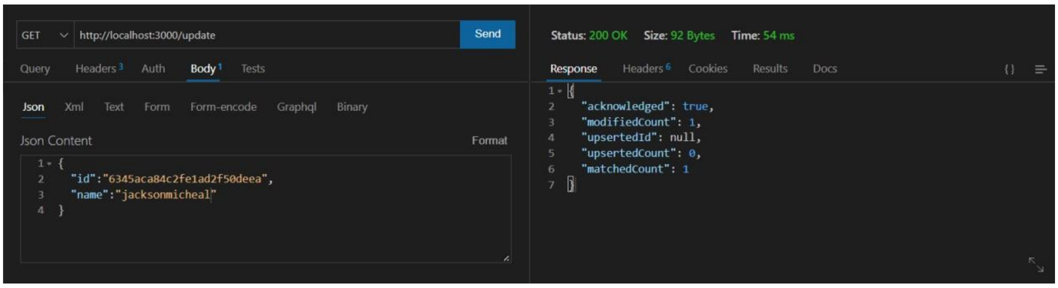
GET http://localhost:3000/read

```
{
  "id":"6345aca84c2fe1ad2f50deea"
}
```

Status: 200 OK Size: 109 Bytes Time: 70 ms

```
{
  "_id": "6345aca84c2fe1ad2f50deea",
  "name": "michealjackson",
  "email": "mj@gmail.com",
  "password": "mj0000",
  "__v": 0
}
```

3. Updating the created user data: -



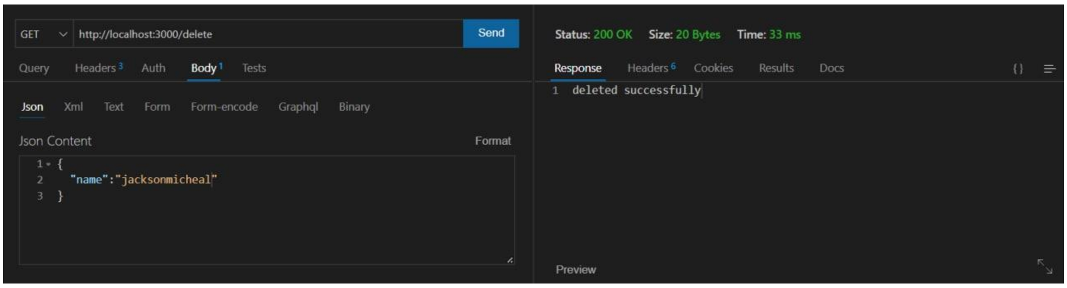
GET http://localhost:3000/update

```
{
  "id":"6345aca84c2fe1ad2f50deea",
  "name":"jacksonmicheal"
}
```

Status: 200 OK Size: 92 Bytes Time: 54 ms

```
{
  "acknowledged": true,
  "modifiedCount": 1,
  "upsertedId": null,
  "upsertedCount": 0,
  "matchedCount": 1
}
```

4. Deleting the created user: -



GET http://localhost:3000/delete

```
{
  "name":"jacksonmicheal"
}
```

Status: 200 OK Size: 20 Bytes Time: 33 ms

```
deleted successfully
```

Practical-7

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 Register and Login system for your project.**

LandingPage.js:-

```
import React from 'react'
import { Link } from 'react-router-dom'
import '../../App.css'
import BackgroundImage from '../../assets/images/bg.png'
export default function LandingPage() {
return (
<header style={ HeaderStyle }>
<h1 className="main-title text-center">login / register page</h1>
<p className="main-para text-center">join us now and don't waste time</p>
<div className="buttons text-center">
<Link to="/login">
<button className="primary-button">log in</button>
</Link>
<Link to="/register">
<button className="primary-button" id="reg_btn"><span>register
</span></button>
</Link>
</div>
</header>
) }
const HeaderStyle = {
width: "100%",
height: "100vh",
background: `url(${BackgroundImage})`,
backgroundPosition: "center",
backgroundRepeat: "no-repeat",
backgroundSize: "cover"
}
```

LoginPage.js:-

```
import React from 'react'
import { Link } from 'react-router-dom'
import '../../App.css'
export default function SignUpPage() {
return (
<div className="text-center m-5-auto">
<h2>Join us</h2>
<h5>Create your personal account</h5>
<form action="/home">
<p>
    <label>Username</label><br/>
    <input type=”text” name=”first_name” required />
</p>
<p>
    <label>Email</label><br/>
    <input type=”email” name=”email” required />
</p>
<p>
    <label>Password</label><br/>
    <input type=”password” name=”pswd” required />
</p>
<p>
    <button id="sub_btn" type="submit">Register</button>
</p>
</form>
<footer>
<p><Link to="/">Back to Homepage</Link>.</p>
</footer>
</div> ) }
```

HomePage.js:-

```
import React from 'react'
import { Link } from 'react-router-dom'
export default function HomePage() {
return (
<div className="text-center">
<h1 className="main-title home-page-title">You are logged in</h1>
<Link to="/">
<button className="primary-button">Log out</button>
</Link>
</div> ) }
```

App.js:-

```
import React from 'react'
import { BrowserRouter as Router, Switch, Route } from 'react-router-dom'
import LandingPage from './components/pages/LandingPage'
import LoginPage from './components/pages/LoginPage'
import RegisterPage from './components/pages/RegisterPage'
import ForgetPasswordPage from './components/pages/ForgetPasswordPage'
import HomePage from './components/pages/HomePage'
import './App.css'
export default function App() {
return (
<Router>
<div>
<Switch>
<Route exact path="/" component={ LandingPage } />
<Route path="/login" component={ LoginPage } />
<Route path="/register" component={ RegisterPage } />
<Route path="/forget-password" component={ ForgetPasswordPage } />
<Route path="/home" component={ HomePage } />
</Switch>
</div>
</Router> ) }
```

Code for server/db.js file: -

```
const mongoose = require("mongoose");
module.exports = () => {
const connectionParams = {
useNewUrlParser: true,
useUnifiedTopology: true,
};
try {
mongoose.connect(process.env.DB, connectionParams);
console.log("Connected to database successfully");
} catch (error) {
console.log(error);
console.log("Could not connect database!");
} };
```

Code for server/index.js file: -

```
require("dotenv").config();
const express = require("express");
const app = express();
const cors = require("cors");
const connection = require("./db");
const userRoutes = require("./routes/users");
const authRoutes = require("./routes/auth");
connection();
app.use(express.json());
app.use(cors());
app.use("/api/users", userRoutes);
app.use("/api/auth", authRoutes);
const port = process.env.PORT || 8080;
app.listen(port, console.log(`Listening on port ${port}...`));
```

Code for server/models/user.js file: -

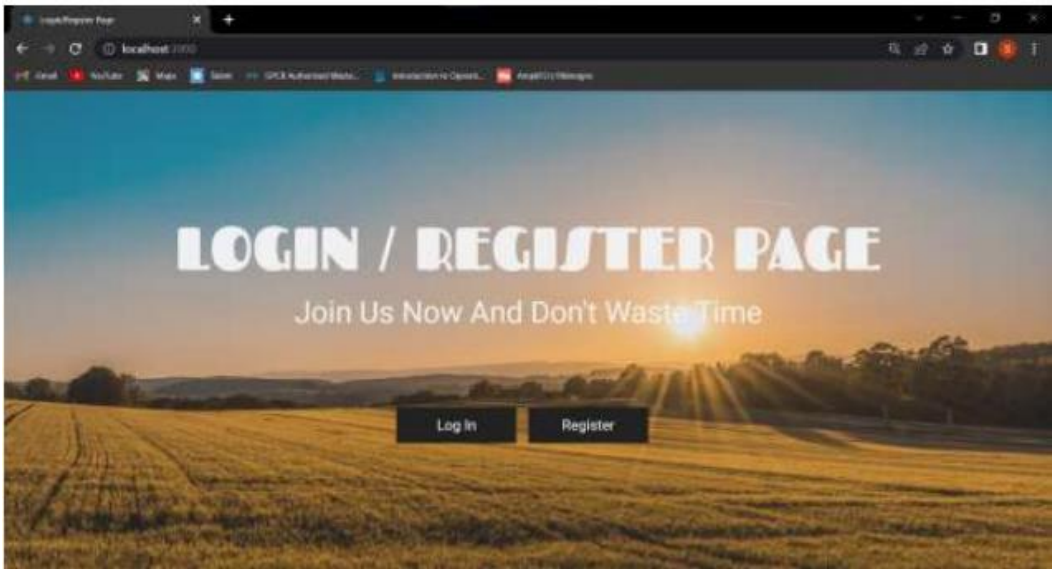
```
const mongoose = require("mongoose");
const jwt = require("jsonwebtoken");
const Joi = require("joi");
const passwordComplexity = require("joi-password-complexity");
const userSchema = new mongoose.Schema({
Name: { type: String, required: true },
email: { type: String, required: true },
password: { type: String, required: true },
});
```

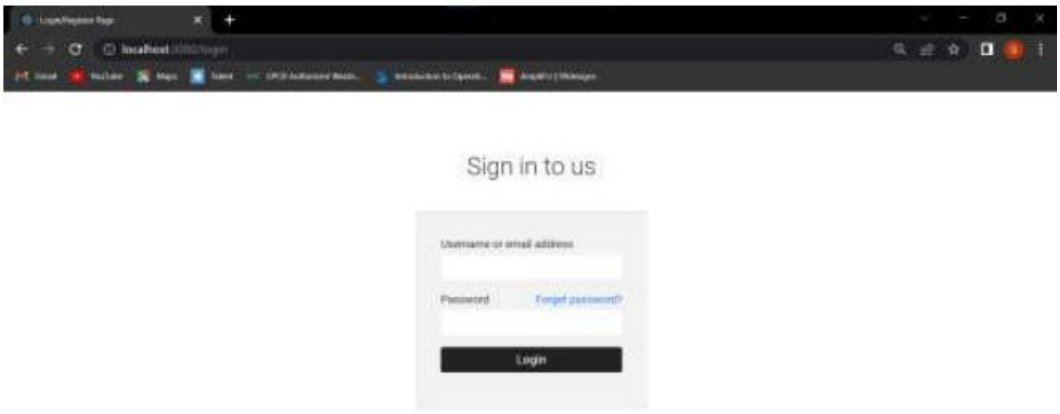
```
userSchema.methods.generateAuthToken = function () {
const token = jwt.sign({ _id: this._id }, process.env.JWTPRIVATEKEY, {
expiresIn: "7d",
});
return token;
};
const User = mongoose.model("user", userSchema);
const validate = (data) => {
const schema = Joi.object({
Name: Joi.string().required().label("First Name"),
email: Joi.string().email().required().label("Email"),
password: passwordComplexity().required().label("Password"),
});
return schema.validate(data);
};
module.exports = { User, validate };
```

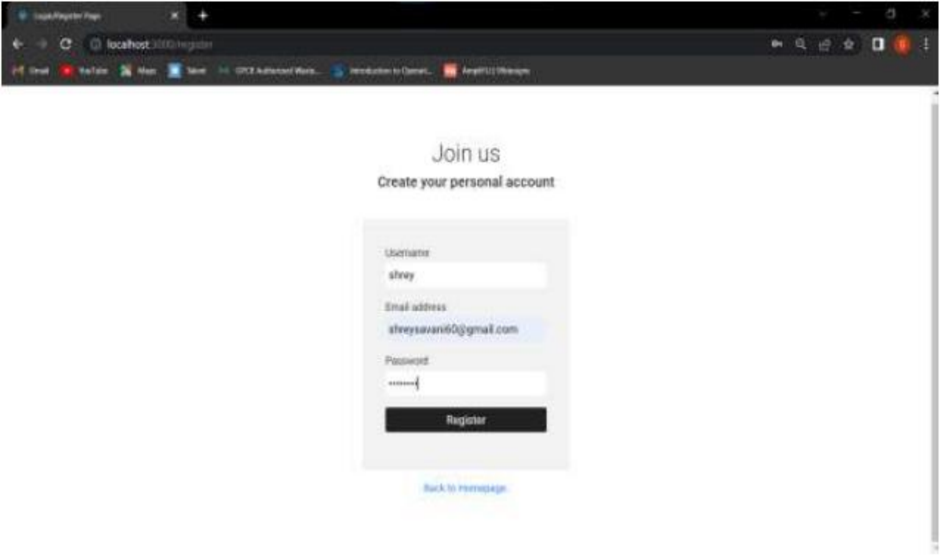
Code for server/routes/auth.js file: -

```
const router = require("express").Router();
const { User } = require("../models/user");
const bcrypt = require("bcrypt");
const Joi = require("joi");
router.post("/", async (req, res) => {
try {
const { error } = validate(req.body);
if (error)
return res.status(400).send({ message: error.details[0].message });
const user = await User.findOne({ email: req.body.email });
if (!user)
return res.status(401).send({ message: "Invalid Email or Password" });
const validPassword = await bcrypt.compare(
req.body.password,
user.password
);
if (!validPassword)
return res.status(401).send({ message: "Invalid Email or Password" });
const token = user.generateAuthToken();
res.status(200).send({ data: token, message: "Logged in successfully" });
} catch (error) {
res.status(500).send({ message: "Internal Server Error" });
} });
const validate = (data) => {
const schema = Joi.object({
email: Joi.string().email().required().label("Email"),
password: Joi.string().required().label("Password"),
});
return schema.validate(data);
};
module.exports = router;
```

Results: -


localhost
LOGIN / REGISTER PAGE
Join Us Now And Don't Waste Time
Log In
Register


localhost
Sign in to us
Username or email address
Password
Forget password?
Login



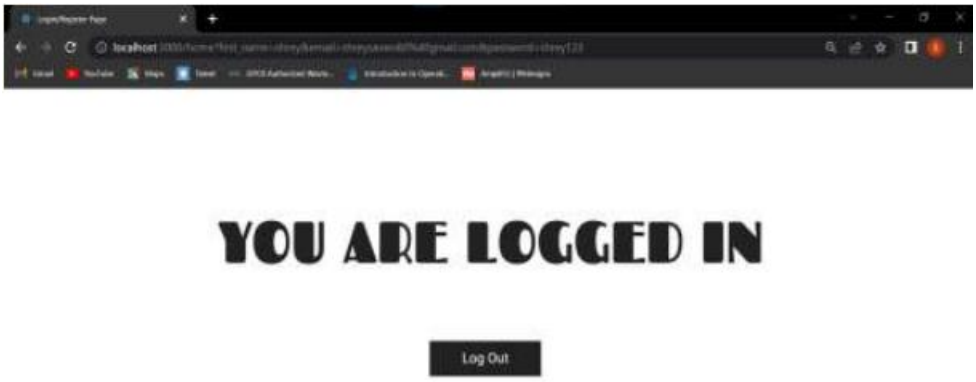
Join us

Create your personal account

Username: shrey

Email address: shreysavani60@gmail.com

Password: ••••••••



YOU ARE LOGGED IN

Practical-8

Aim **Build an exact similar clone of any video streaming web application like Netflix.**

Source code of App.js: -

```
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;
}
```

Code for netflix-ui/components/Card.jsx file: -

```
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 "reacticons/ai";
import { RiThumbUpFill, RiThumbDownFill } from "reacticons/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")}
 />
 <RiThumbUpFill title="Like" />
 <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` max-width:
230px; 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: maxcontent; 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-styletype: none;
 } } }
```

Code for netflix-ui/components/CardSlider.jsx file: -

```
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>
 <div className="slider flex" ref={listRef}>
 {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>
 );
});
```

Code for netflix-ui/components/Header.jsx file: -

```
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>
 );
}
```

Code for netflix-api/routes/UserRoutes.js file: -

```
const {
 addToLikedMovies, getLikedMovies,
 removeFromLikedMovies,} = require("../controllers/UserController"); const
router =
require("express").Router(); router.get("/liked/:email", getLikedMovies);
Code for netflix-api/models/UserModel.js file: -
const mongoose = require("mongoose"); const userSchema = new
mongoose.Schema({ email: {
type: String, required: true, unique: true,
 max: 50,
 },
 likedMovies: Array,
});
```

netflix-api:

Code for netflix-api/controllers/UserController.js file: -

```
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" });
 }
};
```

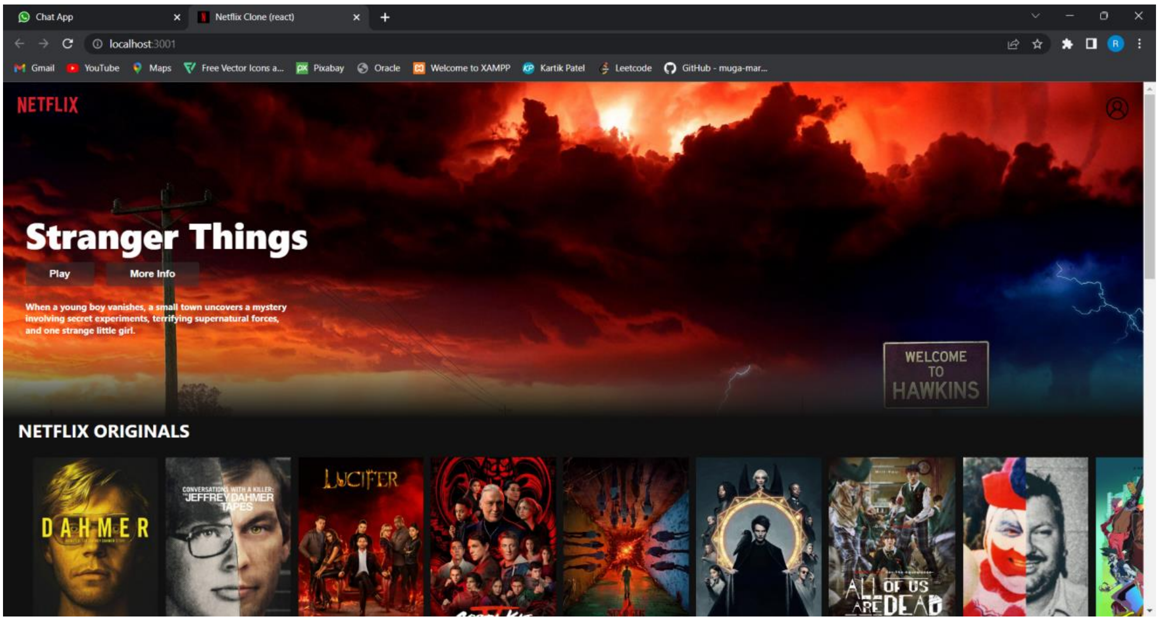
Code for netflix-api/models/UserModel.js file: -

```
const mongoose = require("mongoose"); const userSchema = new
mongoose.Schema({ email: {
type: String, required: true, unique: true,
 max: 50,
 },
 likedMovies: Array,
});
module.exports = mongoose.model("users", userSchema);
```

Code for netflix-api/routes/UserRoutes.js file: -

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

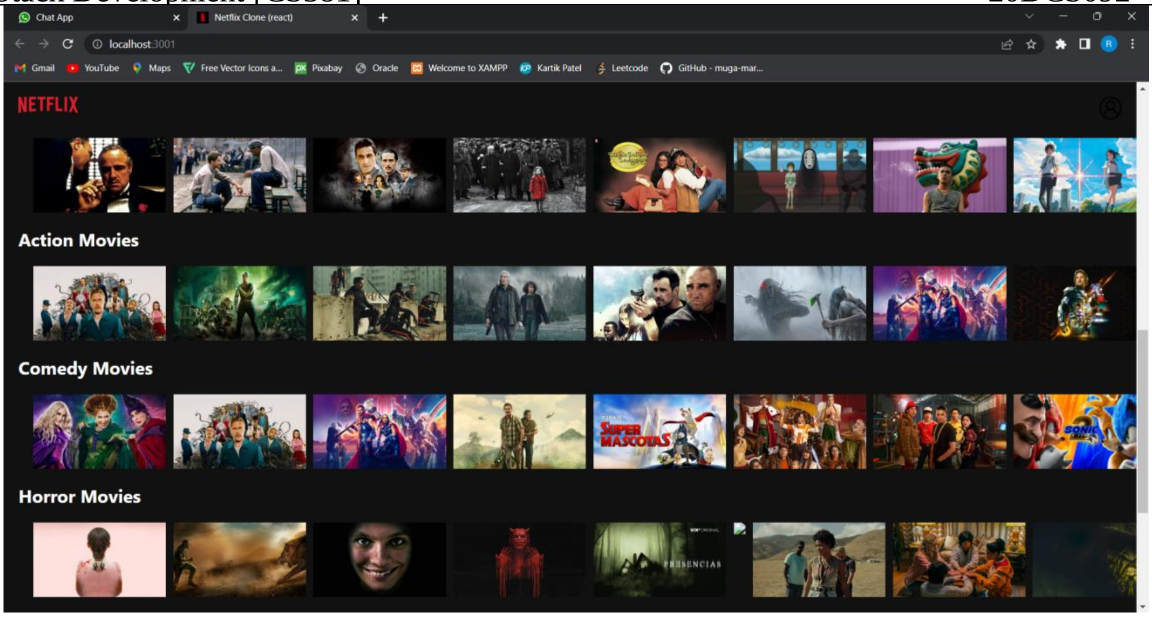
Output:-



Stranger Things

When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces, and one strange little girl.

NETFLIX ORIGINALS



Chat App
Netflix Clone (react)
localhost:3001
Gmail
YouTube
Maps
Free Vector Icons a...
Pixabay
Oracle
Welcome to XAMPP
Kartik Patel
Leetcode
GitHub - muga-mar...
NETFLIX
Action Movies
Comedy Movies
Horror Movies

Practical-9

Aim	**Build an exact similar clone of any social media web application like Instagram/LinkedIn/WhatsApp/Twitter etc.**

Source code of chat.js: -

```
import { Avatar, IconButton } from "@material-ui/core";
import { useParams } from "react-router-dom";
import { SearchOutlined } from "@material-ui/icons"; import React, { useState,
useEffect,
useRef } from "react";
import AttachFileIcon from "@materialui/icons/AttachFile";
import MoreVertIcon from "@materialui/icons/MoreVert";
import MicIcon from "@material-ui/icons/Mic";
import InsertEmoticonIcon from "@materialui/icons/InsertEmoticon";
import "./../CSS/Chat.css"; import db from "./../firebase";
import { useStateValue } from "./StateProvider";
import firebase from "firebase"; import Picker from "emoji-picker-react";
import CloseIcon from "@material-ui/icons/Close";
function Chat() {
 const [input, setInput] = useState(""); const [SEED, setSEED] =
useState(""); const { roomId } = useParams();
 const [roomName, setRoomName] = useState({ name: "",
photoURL: "" }); const [messages, setMessages] = useState([]);
const [{ user }, dispatch] = useStateValue();
 const [isPreviewShown, setPreviewShown] = useState(false);
const [isCloseShown, setCloseShown] = useState(false);
 const myRef = React.createRef(); const messagesEndRef =
React.createRef(); const openEmoji = () => {
 setPreviewShown(!isPreviewShown);
 setCloseShown(!isCloseShown);
 };
 const scrollToBottom = () => {
 messagesEndRef.current?.scrollIntoView({ behavior: "smooth" });
};
 useEffect(() => { scrollToBottom();
 }, [messages]);
 const onEmojiClick = (event, emojiObject) => { const position =
myRef.current.selectionStart; var output = [ input.slice(0,
position), emojiObject.emoji, input.slice(position),
 ].join("");
 setInput(output);
 };
 useEffect(() => { if (roomId) {
 db.collection("rooms")
 .doc(roomId)
 .onSnapshot((snapshot) => setRoomName(
 user.displayName === snapshot.data().firstUserName
 ? {
 name: snapshot.data().secondUserName,
photoURL: snapshot.data().secondUserDP,
 } : {
 name: snapshot.data().firstUserName,
photoURL: snapshot.data().firstUserDP,
 }
 )
 );
 db.collection("rooms")
 .doc(roomId)
 .collection("messages")
 .orderBy("timestamp", "asc")
 .onSnapshot((snapshot) =>
 setMessages(snapshot.docs.map((doc) => doc.data()))
 );
 }
 }, [roomId]);
 useEffect(() => {
 setSEED(Math.floor(Math.random() * 5000));
}, []);
 const sendMessage = (e) => { e.preventDefault();
console.log("You typed >>>>", input); setInput("");
 db.collection("rooms").doc(roomId).collection("messages").add({
message: input, name: user.displayName,
 timestamp: firebase.firestore.FieldValue.serverTimestamp(),
 });
 };
 return (
 <div className="chat">
 <div className="chat__header">
 <Avatar
 src={
 roomName.photoURL ? roomName.photoURL
 : `https://avatars.dicebear.com/api/human/${SEED}.svg`
 }
 />
 <div className="chat__headerinfo">
 <h3>{roomName.name}</h3>
 <p>
 Last Messaage:{" "}
 {messages.length ? new Date(
 messages[messages.length - 1]?.timestamp?.toDate()
).toLocaleString("hi-IN")
 : "none"}
 </p>
 </div>
 <div className="chat__headerRight"> <IconButton>
 <SearchOutlined />
 </IconButton>
 <IconButton>
 <AttachFileIcon />
 </IconButton>
 <IconButton>
 <MoreVertIcon />
 </IconButton>
 </div>
</div>
 <div className="chat__body">
 {messages.map((message) => (
 <p
 className={`chat__message ${
 message.name === user.displayName && "chat_reciever"
 } `}
 >
 <span className="chat__name">{message.name}</span>
 {message.message}
 <span className="chat__timestamp">
 {new
Date(message.timestamp?.toDate()).toLocaleString("hi-IN")}
</span>
 </p>
 ))}
 {isPreviewShown && (
 <Picker
 pickerStyle={{ position: "absolute", bottom: 0, left: 0 }}
onEmojiClick={onEmojiClick}
 />
 )}
 <div ref={messagesEndRef} />
 </div>
 <div className="chat__footer">
 {!isCloseShown && (
 <IconButton onClick={openEmoji}>
 <InsertEmoticonIcon />
 </IconButton>
 )}
 {isCloseShown && (
 <IconButton onClick={openEmoji}> <CloseIcon />
 </IconButton>
 )}
<form>        <input value={input}
ref={myRef}
 onChange={(e) => setInput(e.target.value)}
placeholder="Type a message" type="text"
 ></input>
 <button onClick={sendMessage} type="submit">
 Send a messaage
 </button>
 </form>
 <IconButton>
 <MicIcon />
 </IconButton>
 </div>
 </div>
);
}
export default Chat;
```

Login.js :-

```
import { Button } from "@material-ui/core";
import React from "react";
import { auth, provider } from "../firebase";
import "./../CSS/Login.css";
import { useStateValue } from "./StateProvider";
import db from "./../firebase";
import { actionType } from "./Reducer";
function Login() {
 const [{}, dispatch] = useStateValue(); const signIn = () => {
 auth
 .signInWithPopup(provider)
 .then((result) => { dispatch({
 type: actionType.SET_USER, user: result.user,
 });
 console.log(result.user.email);
 var docRef = db.collection("users").doc(result.user.email);
 docRef
.get()
 .then((doc) => { if (doc.exists) {
 } else {
 db.collection("users").doc(result.user.email).set({
name: result.user.displayName,
 email: result.user.email,
 photoURL: result.user?.photoURL,
 });
 console.log("No such document!");
 }
 })
 .catch((error) => {
 console.log("Error getting document:", error);
 });
 })
 .catch((error) => alert(error.message));
 }; return (
 <div className="login">
 <div className="login__container">
 <img

src="http://pngimg.com/uploads/whatsapp/whatsapp_PNG95169.png
" alt=""
 />
 <div className="login__text">
 <h1>Sign in to Chat App</h1>
 </div>
 <Button type="submit" onClick={signIn}>
 Sign in with Google
 </Button>
 </div>
 </div>
 );
}
export default Login;
```

SidebarChat.js:-

```
import React, { useEffect, useState } from "react";
import "./../CSS/SidebarChat.css";
import { Avatar } from "@materialui/core";
import db from "./../firebase";
import { Link } from "react-router-dom";
import { useStateValue } from "./StateProvider";
function SidebarChat({ id, name, photoURL, addNewChat }) {
const [SEED, setSEED] = useState(""); const [messages,
setMessages] = useState(""); const [{ user }, dispatch] =
useStateValue(); useEffect(() => {
 setSEED(Math.floor(Math.random() * 5000));
 }, []);
 useEffect(() => { if (id) {
 db.collection("rooms")
 .doc(id)
 .collection("messages")
 .orderBy("timestamp", "desc") .onSnapshot((snapshot) =>
 setMessages(snapshot.docs.map((doc) => doc.data()))
 );
 }
 }, [id]);
 const createChat = () => {
 const email = prompt("Please enter email for chat"); if (email) {
 db.collection("users")
 .doc(email)
 .get()
 .then((doc) => { if (doc.exists) {
 db.collection("rooms").add({
 firstUser: email, secondUser: user.email,
firstUserName: doc.data().name, secondUserName:
user.displayName, firstUserDP: doc.data().photoURL,
secondUserDP: user?.photoURL,
 });
 } else {
 alert(`No user found with ${email}`);
 }
});
 } else {
 }
 };
 return !addNewChat ? ( <Link to={`/rooms/${id}`}>
 <div className="sidebarChat">
 <Avatar src={ photoURL ? photoURL
 : `https://avatars.dicebear.com/api/human/${SEED}.svg`
 }
 />
 <div className="sidebarChat__info">
 <h2>{name}</h2>
 <p>{messages[0]?.message}</p>
 </div>
 </div>
 </Link>
 ) : (
 <div onClick={createChat} className="sidebarChat">
 <h2>Add New Chat</h2>
 </div>
 );
}
export default SidebarChat;
```

App.js :-

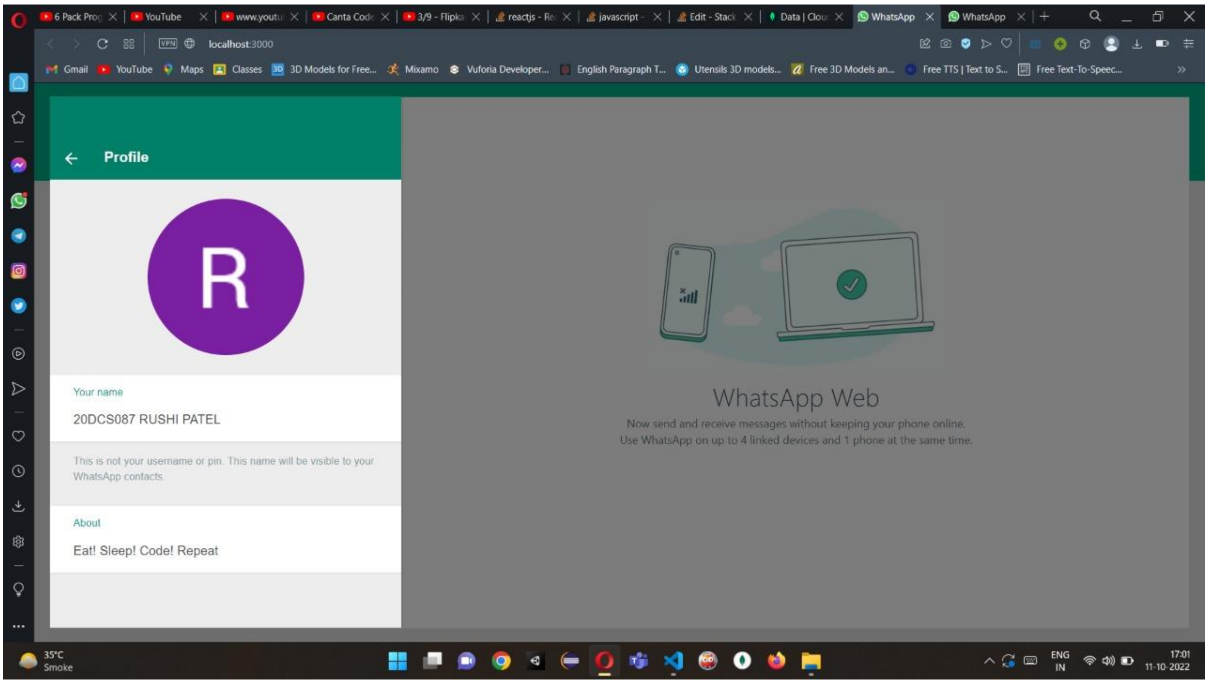
```
import "./App.css";
import Chat from "./Components/Chat";
import Sidebar from "./Components/Sidebar";
import { useState } from "react";
import { BrowserRouter as Router, Switch, Route } from "reactrouter-dom";
import Login from "./Components/Login";
import { useStateValue } from "./Components/StateProvider";
function App() {
 const [{ user }, dispatch] = useStateValue();
return (
 <div className="app">
 {!user ? (
 <Login />
 ) : (
 <div className="app__body">
 <Router>
 <Sidebar />
 <Switch>
 <Route path="/rooms/:roomId">
 <Chat />
 </Route>
 <Route path="/" />
 </Switch>
 </Router>
 </div>
 )}
 </div>
 );}
```

```
export default App;
```

Firebase.js :-

```
import firebase from "firebase";
const firebaseConfig = {
 apiKey: "AIzaSyBUxrxJBJcRu2tMnE7OmBgTqhsstc8Eb1k",
 authDomain: "chat-app-b190a.firebaseapp.com", projectId: "chatapp-b190a",
 storageBucket: "chat-app-b190a.appspot.com",
messagingSenderId: "807646752429",
 appId: "1:807646752429:web:d685c08ea96daed472e024",
measurementId: "G-CLL655MRQX"
};
const firebaseApp = firebase.initializeApp(firebaseConfig); const db
= firebaseApp.firestore(); const auth = firebaseApp.auth();
const provider = new firebase.auth.GoogleAuthProvider();
export { auth, provider }; export default db;
```

Output:-



Your name

20DCS087 RUSHI PATEL

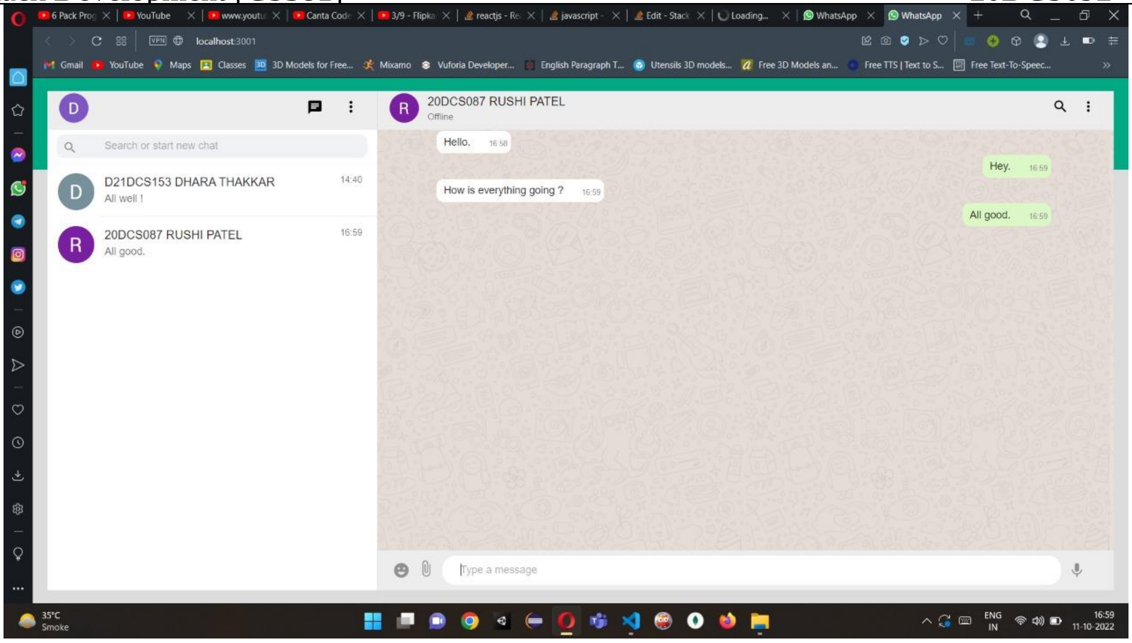
This is not your username or pin. This name will be visible to your WhatsApp contacts.

About

Eat! Sleep! Code! Repeat

WhatsApp Web

Now send and receive messages without keeping your phone online.
Use WhatsApp on up to 4 linked devices and 1 phone at the same time.



D21DCS153 DHARA THAKKAR 14:40
All well !

20DCS087 RUSHI PATEL 16:59
All good.

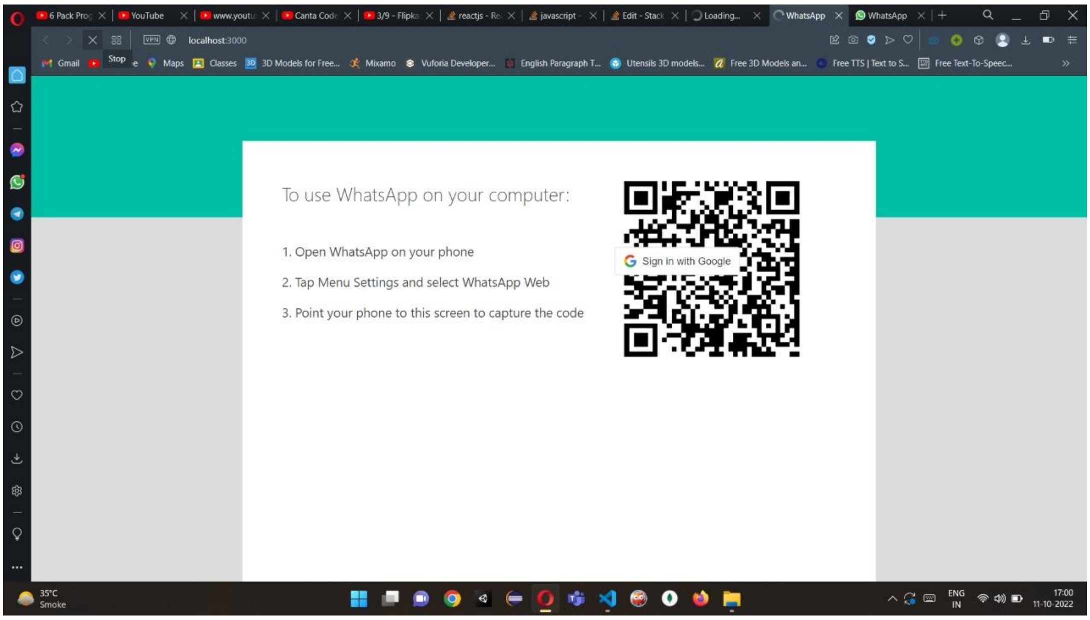
20DCS087 RUSHI PATEL
Offline

Hello. 16:58

Hey. 16:59

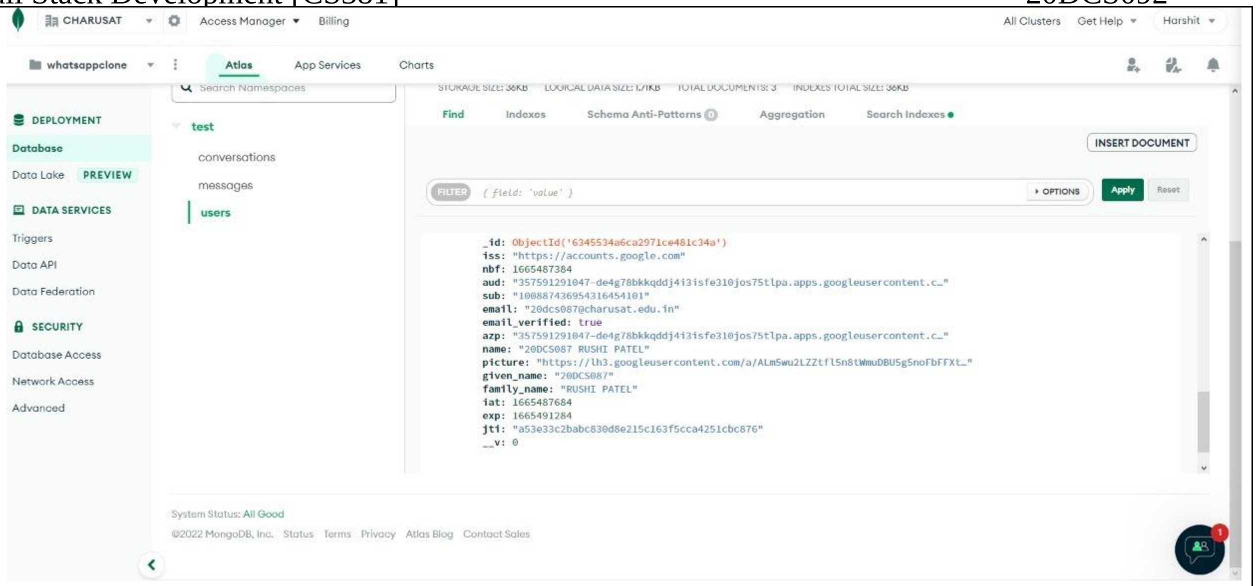
How is everything going ? 16:59

All good. 16:59



To use WhatsApp on your computer:

1. Open WhatsApp on your phone
2. Tap Menu Settings and select WhatsApp Web
3. Point your phone to this screen to capture the code



```
_id: ObjectId('6345534a6ca2971ce481c34a')
iss: "https://accounts.google.com"
nbf: 1665487384
aud: "357591291047-de4g78bkkqddj4i3isfe310jos75tlpa.apps.googleusercontent.c…"
sub: "100887436954316454101"
email: "20dcs087@charusat.edu.in"
email_verified: true
azp: "357591291047-de4g78bkkqddj4i3isfe310jos75tlpa.apps.googleusercontent.c…"
name: "20DCS087 RUSHI PATEL"
picture: "https://lh3.googleusercontent.com/a/ALm5wu2LZZtfl5n8tWmuDBU5g5noFbFFXt…"
given_name: "20DCS087"
family_name: "RUSHI PATEL"
iat: 1665487684
exp: 1665491284
jti: "a53e33c2babc830d8e215c163f5cca4251cbc876"
__v: 0
```