**1. Create a `Person` component that receives `name` and `age` props and renders a message like "John is 25 years old." Style the `Person` component using inline styles. Change the text color to blue and add a border.**

**Ans-**

import PropTypes from "prop-types";

export default function Person({ name, age }) {

return (

<div>

<h1

style={{

color: "blue",

border: "1px solid black",

margin: "auto",

textAlign: "center",

padding: "auto",

}}

>

{name} is {age} years old.

</h1>

</div>

);

}

Person.propTypes = {

name: PropTypes.string.isRequired,

age: PropTypes.number.isRequired,

};

**Output:-**



**2. Create a React component with a button. When the button is clicked, toggle between displaying and hiding a message. Use external CSS module to style the component.**

**Ans-**

import { useState } from "react";

import "../styles/toggle.css";

export default function Toggle() {

const [toggle,setToggle] = useState(true);

const handleToggle = () => {

setToggle(!toggle);

};

return (

<div>

<center>

{toggle && <h1 className="mystyle">Hello, from Aniket!! Hope u have a nice day</h1>}

<button onClick={handleToggle} className="btn btn-success">

Toggle

</button>

</center>

</div>

);

}

Css

@import url("https://fonts.googleapis.com/css2?family=Bungee+Spice&display=swap");

.mystyle {

font-family: "Bungee Spice", sans-serif;

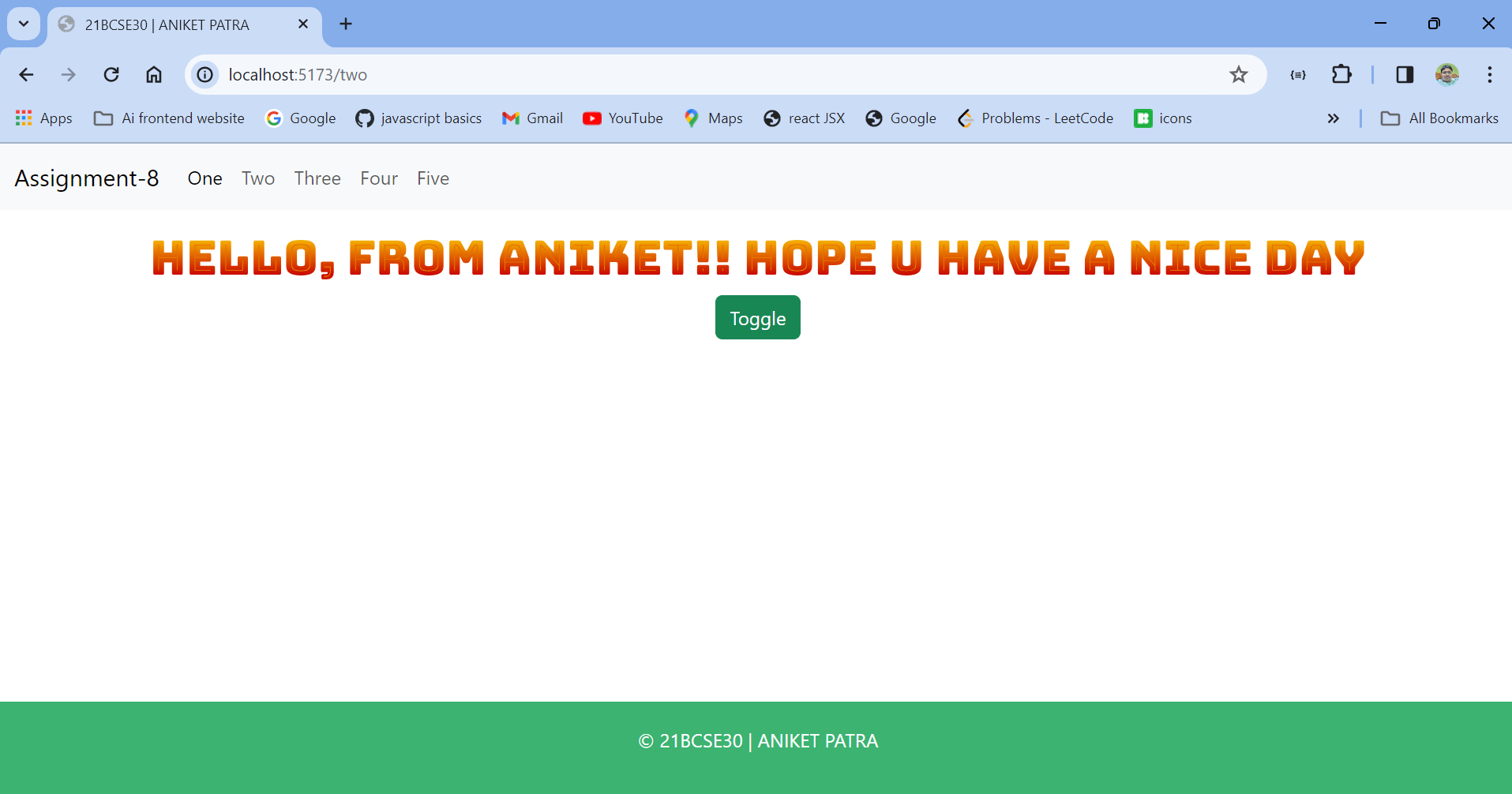
font-weight: 400;

font-style: normal;

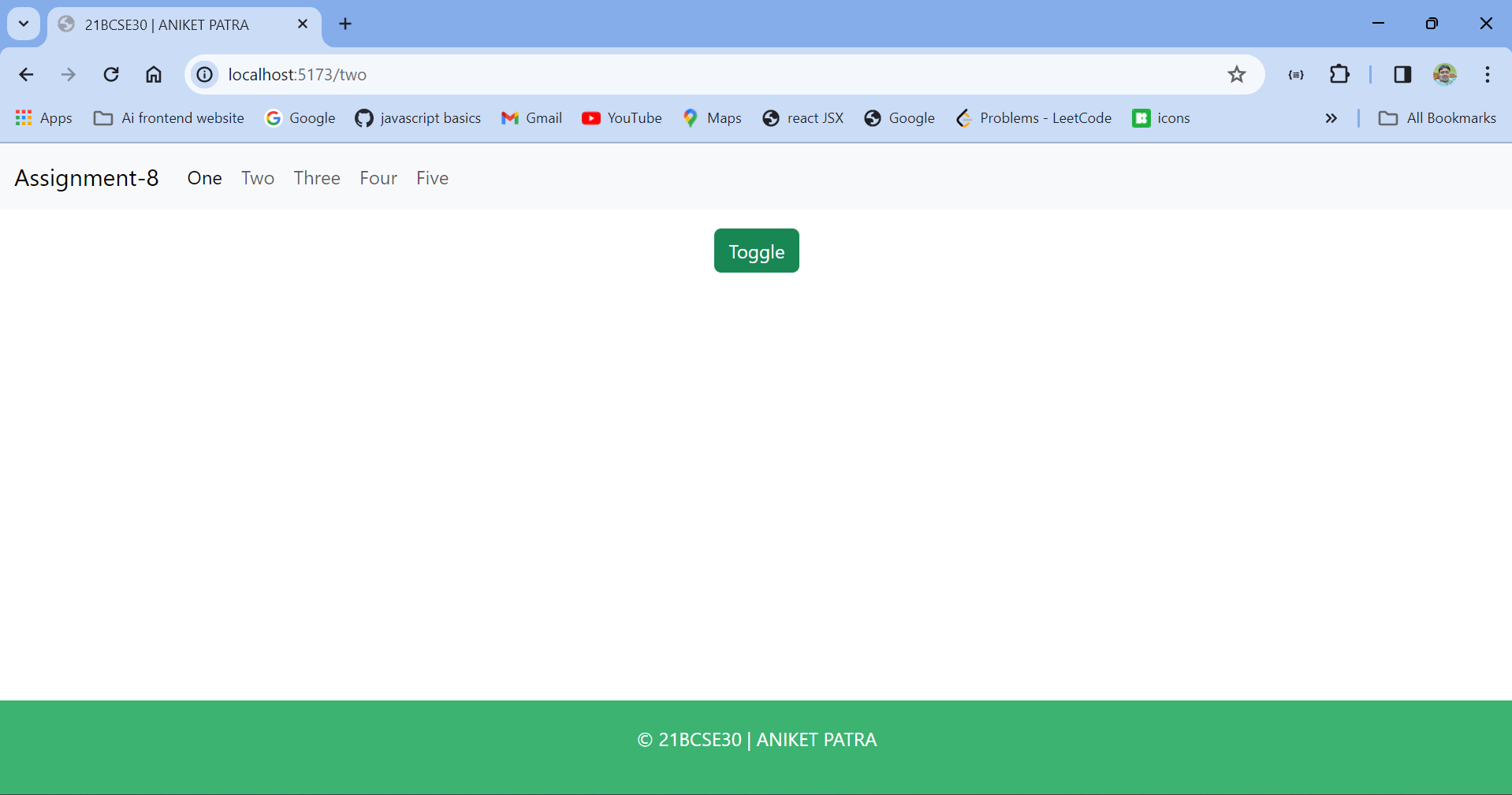
}

**Output:-**

**Message visible**

****

**Message invisible**

****

**3. Create a component named `UserGreeting` that displays "Welcome back!" if a user is logged in and "Please log in" if not. Include a button to toggle the user's login state.**

**Ans-**

import { useState } from "react";

export default function UserGreeting() {

const [status, setStatus] = useState("logout");

const handleChange = () => {

if (status === "login") {

return setStatus("logout");

} else {

return setStatus("login");

}

};

return (

<div>

<center>

<h1>{status === "login" ? "Welcome back!" : "Please Log in !"}</h1>

<button className="btn btn-primary" onClick={handleChange}>{status === "login" ? "Log Out" : "Log In"}</button>

</center>

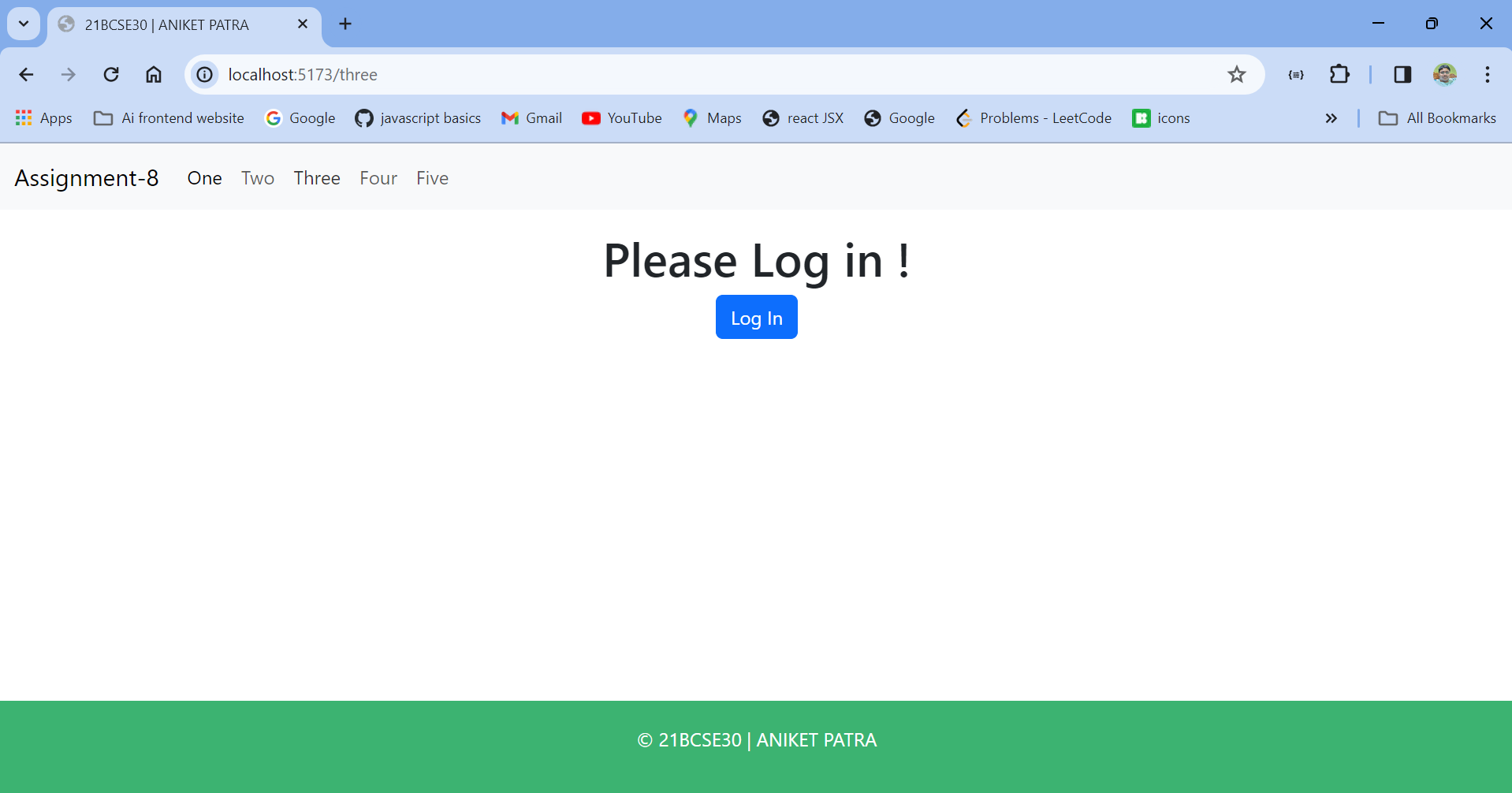
</div>

);

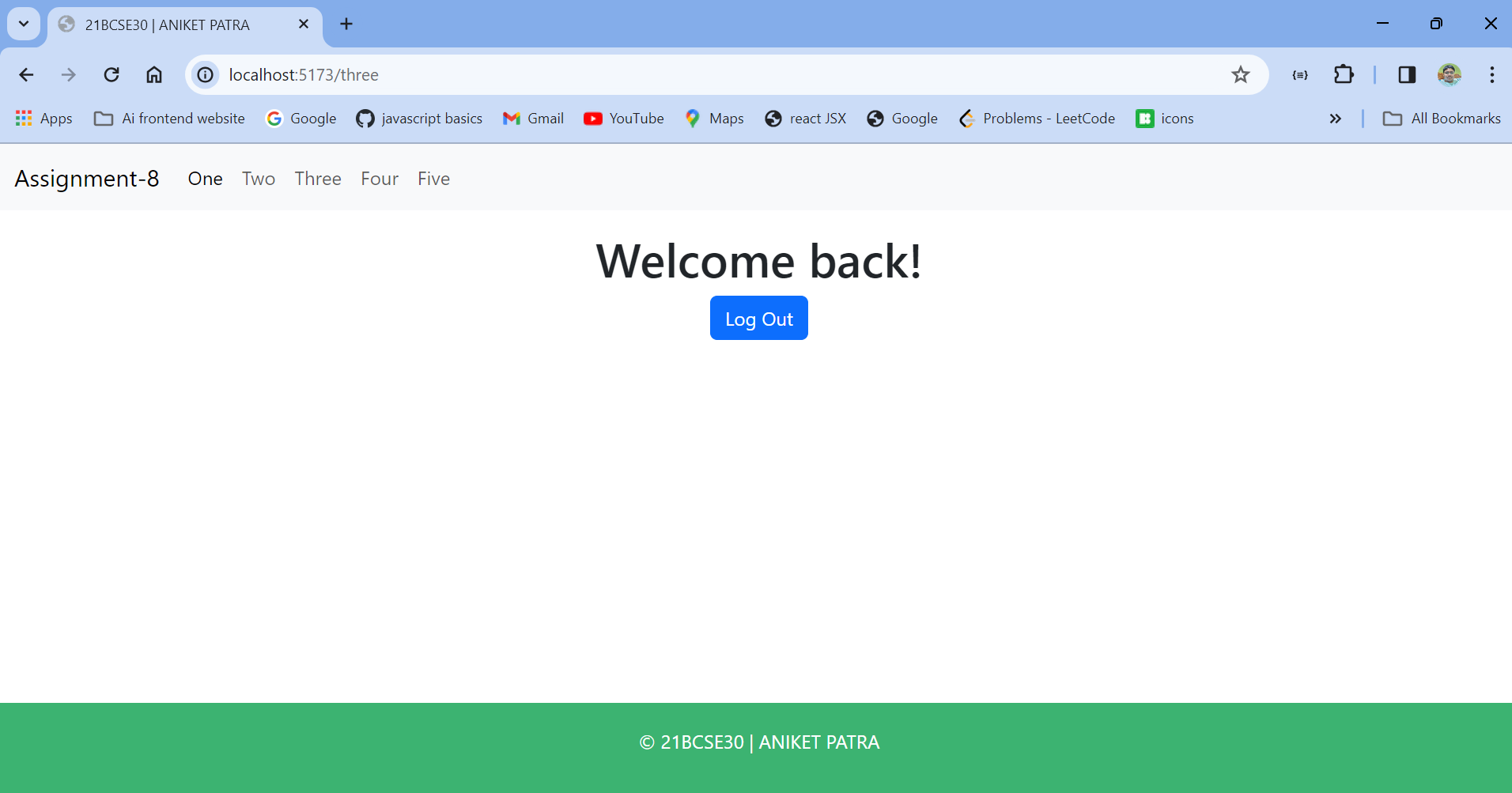
}

**Output:-**

**Initial Log In**



**After Log In**

****

**4. Create an expense tracker app with add and delete expense feature using React. Keep tract of both credit and debit. Display the total debit and total credit and also display the individual transactions. For each transaction keep tract of Description, amount, date and type of transaction.**

**Ans-**

**Expense Tracker Page**

import { useState } from 'react'

import ExpenseOverview from '../components/ExpenseOverview'

import ExpenseForm from '../components/ExpenseForm'

import ExpenseTransaction from '../components/ExpenseTransaction'

const Four = () => {

const [ expenses, setExpenses ] = useState([])

function addExpense(newExpense) {

setExpenses([newExpense, ...expenses])

}

function deleteExpense( eid ){

console.log(eid);

let newExpenses = expenses.filter (e => e.id !== eid)

setExpenses(newExpenses)

}

return (

<div className='row'>

<div className="col-md-6 mx-auto">

<h1 className='text-center my-2 display-4'>Expense Tracker</h1>

<ExpenseOverview expenses={expenses} />

<ExpenseForm addExpense={addExpense} />

<ExpenseTransaction expenses={expenses} deleteExpense={deleteExpense} />

</div>

</div>

)

}

export default Four

**ExpenseForm**

import { useRef } from 'react'

import PropTypes from "prop-types";

ExpenseForm.propTypes = {

addExpense: PropTypes.func.isRequired,

};

const ExpenseForm = ({ addExpense }) => {

const descriptionRef = useRef(null)

const amountRef = useRef(null)

const dateRef = useRef(null)

const typeRef = useRef(null)

function submitHandler(e){

e.preventDefault()

const newExpense = {

id: new Date().getTime(),

description: descriptionRef.current.value,

amount: amountRef.current.value,

date: dateRef.current.value,

type: typeRef.current.value,

}

addExpense(newExpense)

descriptionRef.current.value = ""

amountRef.current.value = ""

typeRef.current.value = ""

dateRef.current.value = ""

}

return (

<form action="" method="post" onSubmit={submitHandler}>

<div className="row g-2 my-2">

<div className="col-md-6">

<input

ref={descriptionRef}

type="text"

className="form-control"

placeholder="Description"

required

/>

</div>

<div className="col-md-6">

<input

ref={amountRef}

type="text"

className="form-control"

placeholder="Amount"

required

/>

</div>

<div className="col-md-4">

<input

ref={dateRef}

type="date"

className="form-control"

required

/>

</div>

<div className="col-md-4">

<select ref={typeRef} className='form-control' required>

<option value="debit">Debit</option>

<option value="credit">Credit</option>

</select>

</div>

<div className="col-md-4">

<input

type='submit'

value="ADD"

className="btn btn-primary w-100"

/>

</div>

</div>

</form>

)

}

export default ExpenseForm

**Expense Overview**

import PropTypes from "prop-types";

ExpenseOverview.propTypes = {

expenses: PropTypes.shape({

id: PropTypes.string,

title: PropTypes.string,

amount: PropTypes.number,

date: PropTypes.instanceOf(Date),

category: PropTypes.string,

filter: PropTypes.func

}).isRequired,

};

const ExpenseOverview = ({ expenses }) => {

const credits = expenses.filter( e => e.type === 'credit')

const debits = expenses.filter( e => e.type === 'debit')

const totalCredits = credits.reduce((tc, e)=> tc+Number(e.amount),0)

const totalDebits = debits.reduce((td, e)=> td+Number(e.amount),0)

return (

<div>

<div className="row g-2">

<div className="col-md-6 bg-success text-center py-2 text-white">

<h1>+ {totalCredits}</h1>

</div>

<div className="col-md-6 bg-danger text-center py-2 text-white">

<h1>- {totalDebits}</h1>

</div>

</div>

</div>

)

}

export default ExpenseOverview

**Expense Transaction**

import ExpenseTransactionDetails from './ExpenseTransactionDetails'

import PropTypes from "prop-types";

ExpenseTransaction.propTypes = {

expenses: PropTypes.shape({

id: PropTypes.string,

title: PropTypes.string,

amount: PropTypes.number,

date: PropTypes.instanceOf(Date),

category: PropTypes.string,

filter: PropTypes.func,

}).isRequired,

deleteExpense: PropTypes.func.isRequired,

};

const ExpenseTransaction = ({ expenses, deleteExpense }) => {

const credits = expenses.filter( e => e.type === 'credit')

const debits = expenses.filter( e => e.type === 'debit')

return (

<div>

<div className="row g-2">

<div className="col-md-6">

{

credits.map( credit => <ExpenseTransactionDetails key={credit.id} expense={credit} deleteExpense={deleteExpense} />)

}

</div>

<div className="col-md-6">

{

debits.map( debit => <ExpenseTransactionDetails key={debit.id} expense={debit} deleteExpense={deleteExpense} />)

}

</div>

</div>

</div>

)

}

export default ExpenseTransaction

**Expense Transaction Details**

import PropTypes from "prop-types";

ExpenseTransactionDetails.propTypes = {

expense: PropTypes.shape({

id: PropTypes.number,

date: PropTypes.instanceOf(Date),

amount: PropTypes.number,

type: PropTypes.string,

description: PropTypes.string

}).isRequired,

deleteExpense: PropTypes.func

};

const ExpenseTransactionDetails = ({ expense, deleteExpense }) => {

const {id, description, type, amount, date} = expense

return (

<div className={`card rounded-0 mb-1 bg-${type==='credit'? 'success' : 'danger'}-subtle`}>

<div className="card-body d-flex justify-content-between">

<div>

<p className='fw-semibold mb-0'>{description} - {amount}</p>

<p>{date}</p>

</div>

<div>

<h5 role='button' onClick={()=>deleteExpense(id)}>X</h5>

</div>

</div>

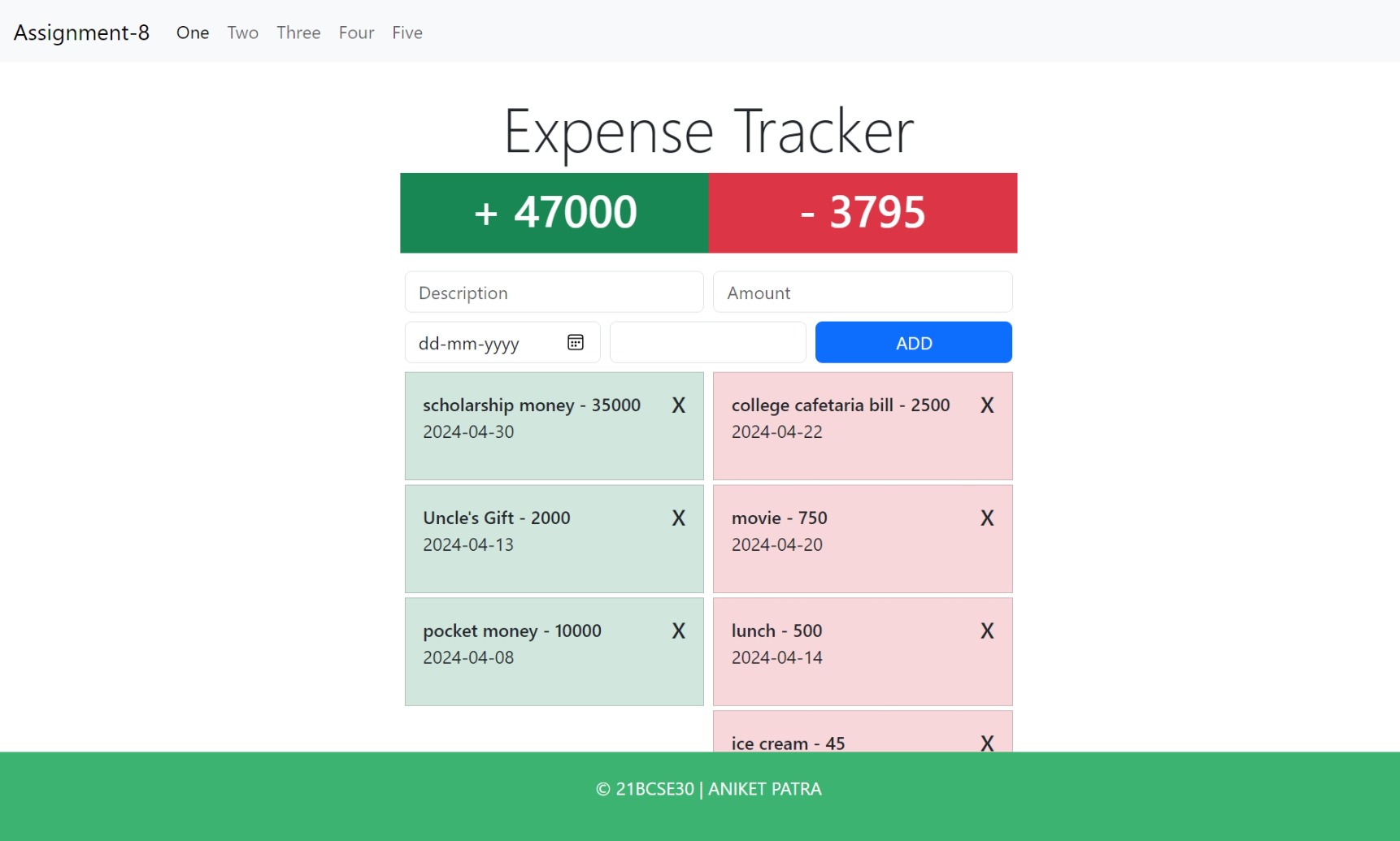
</div>

)

}

export default ExpenseTransactionDetails

**Output:-**



**5. Make an API call to https://dummyjson.com/users and display the user details like photo, name, address, mobile and email in user cards.**

**Ans-**

**UserCard**

import PropTypes from "prop-types";

export default function UserCard({ user }) {

const { image, firstName, lastName, phone, email, address } = user;

return (

<div className="card" style={{ width: "25rem", marginBottom: "2rem" }}>

<img src={image} className="card-img-top" alt="User Photo" />

<div className="card-body">

<h5 className="card-title">

{firstName}&nbsp;{lastName}

</h5>

<p className="card-text">

{address.address}

<br />

CITY {address.city}

<br />

STATE {address.state}

<br />

PIN-CODE {address.postalCode}

</p>

</div>

<ul className="list-group list-group-flush">

<li className="list-group-item">Tel No. {phone}</li>

<li className="list-group-item">Email {email}</li>

</ul>

</div>

);

}

UserCard.propTypes={

user: PropTypes.shape({

image: PropTypes.string,

firstName: PropTypes.string,

lastName: PropTypes.string,

phone: PropTypes.string,

email: PropTypes.string,

address: PropTypes.shape({

address:PropTypes.string,

street: PropTypes.string,

city: PropTypes.string,

state: PropTypes.string,

postalCode: PropTypes.string,

}).isRequired,

})

}

**Users Page**

import UserCard from "../components/UserCard";

import axios from "axios";

import { useEffect, useState } from "react";

export default function Users() {

const [users, setUsers] = useState(null);

const getUsers = async () => {

const users = await axios.get("https://dummyjson.com/users");

setUsers(users.data.users);

};

useEffect(() => {

getUsers();

}, []);

return (

<div className="row">

{ users && users.map((user) => {

return (

<div key={user.id} className="col-md-4 col-sm-6">

<UserCard user={user} />

</div>

);

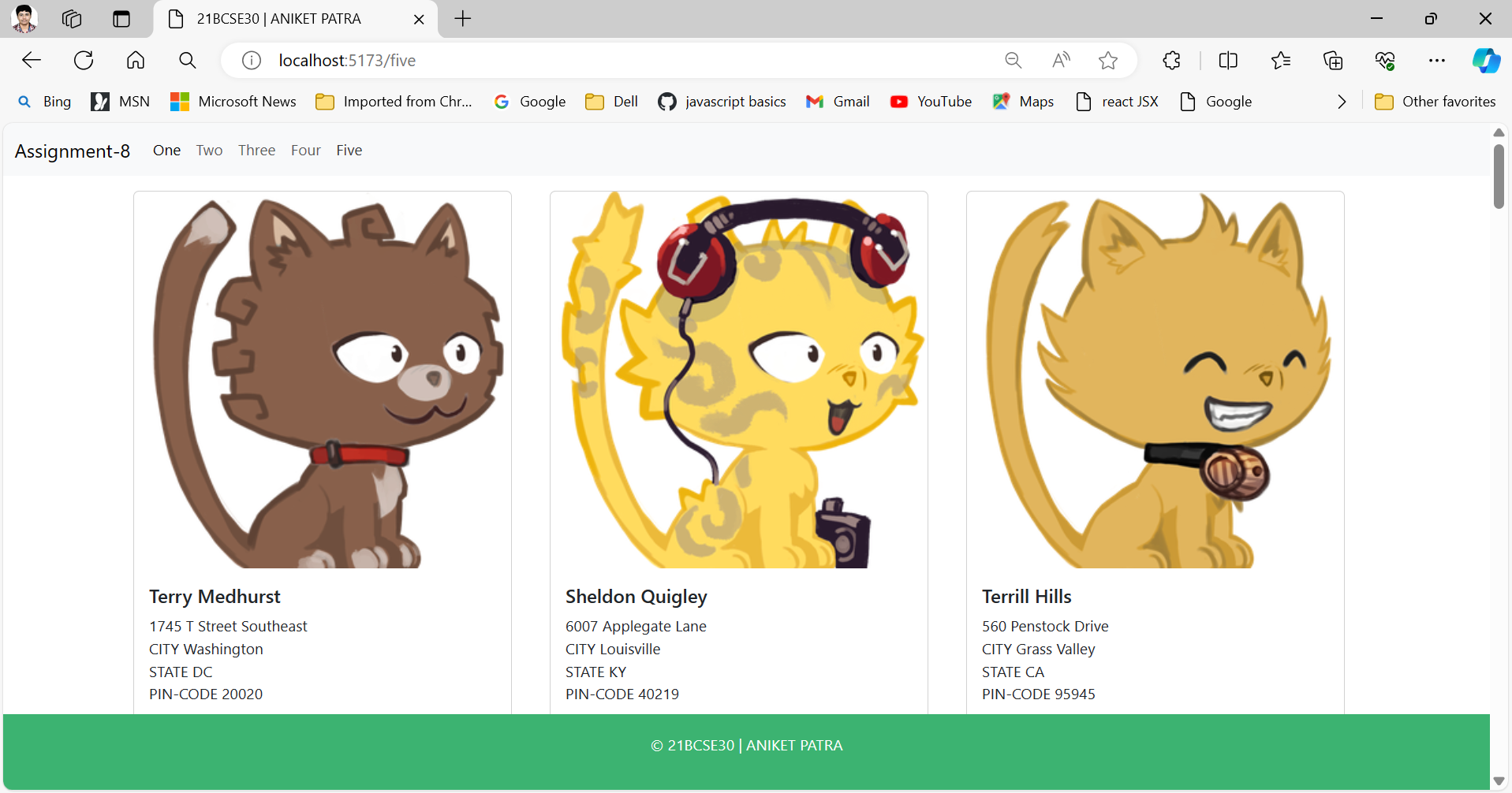
})}

</div>

);

}

**Output:-**



**6. Create a navbar to connect the apps created above with the help of react router dom. Create a layout to have navbar and footer in all the pages.**

**Ans-**

**Layout.jsx**

import { Outlet } from "react-router-dom";

import Navbar from "./Navbar";

import Footer from "./Footer";

export default function Layout() {

return (

<>

<Navbar />

<main className="container py-3">

<Outlet />

</main>

<Footer />

</>

);

}

**Navbar.jsx**

import { Link } from "react-router-dom";

export default function Navbar() {

return (

<nav className="navbar navbar-expand-lg bg-body-tertiary">

<div className="container-fluid">

<Link className="navbar-brand" to={"/"}>

Assignment-8

</Link>

<button

className="navbar-toggler"

type="button"

data-bs-toggle="collapse"

data-bs-target="#navbarNav"

aria-controls="navbarNav"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span className="navbar-toggler-icon"></span>

</button>

<div className="collapse navbar-collapse" id="navbarNav">

<ul className="navbar-nav">

<li className="nav-item">

<Link className="nav-link active" aria-current="page" to={"/one"}>

One

</Link>

</li>

<li className="nav-item">

<Link className="nav-link" to={"/two"}>

Two

</Link>

</li>

<li className="nav-item">

<Link className="nav-link" to={"/three"}>

Three

</Link>

</li>

<li className="nav-item">

<Link className="nav-link" to={"/four"}>

Four

</Link>

</li>

<li className="nav-item">

<Link className="nav-link" to={"/five"}>

Five

</Link>

</li>

</ul>

</div>

</div>

</nav>

);

}

**Footer.jsx**

import "../styles/footer.css";

export default function Footer() {

return (

<footer className="footer">

<p>&copy; 21BCSE30 | ANIKET PATRA</p>

</footer>

);

}

**Footer.css**

body {

margin: 0;

padding: 0;

height: 100%;

}

.footer {

position: fixed;

left: 0;

bottom: 0;

width: 100%;

background-color: mediumseagreen;

color: white;

text-align: center;

padding: 20px 0; /\* Adjust padding as needed \*/

}

**Output:-**

