**React js**

**Docs 10**

**Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.**

**App.js**

import logo from './logo.svg';

import './App.css';

import React from "react";

const sampleOffices = [

{

id: 1,

name: "Alpha Workspaces",

rent: 55000,

address: "12 High St, Bangalore",

image: "<https://via.placeholder.com/200x120?text=Alpha>"

},

{

id: 2,

name: "Beta Suites",

rent: 75000,

address: "99 Market Rd, Mumbai",

image: "<https://via.placeholder.com/200x120?text=Beta>"

},

{

id: 3,

name: "Gamma Offices",

rent: 60000,

address: "7 Lakeview Ave, Chennai",

image: "<https://via.placeholder.com/200x120?text=Gamma>"

}

];

function OfficeCard({ office }) {

const rentStyle = {

color: office.rent < 60000 ? "red" : "green",

fontWeight: "700"

};

const cardStyle = {

border: "1px solid #ddd",

padding: "12px",

borderRadius: 8,

width: 260,

boxShadow: "0 2px 6px rgba(0,0,0,0.06)",

margin: 8

};

return (

<div style={cardStyle}>

<img src={"https://img.freepik.com/free-photo/modern-office-space-interior\_158595-5206.jpg"} alt={office.name} style={{ width: "100%", borderRadius: 6 }} />

<h3>{office.name}</h3>

<p style={{ margin: "6px 0" }}>{office.address}</p>

<p style={rentStyle}>₹ {office.rent.toLocaleString()}</p>

</div>

);

}

function App() {

const container = { display: "flex", flexWrap: "wrap", gap: 12, padding: 16 };

return (

<div>

<header style={{ padding: 16 }}>

<h1>Office Space Rental</h1>

<p>List of offices (rent &gt;= 60000 shows green; &lt;60000 shows red)</p>

</header>

<section style={container}>

{sampleOffices.map((office) => (

<OfficeCard office={office} key={office.id} />

))}

</section>

</div>

);

}

export default App;

**Index.js**

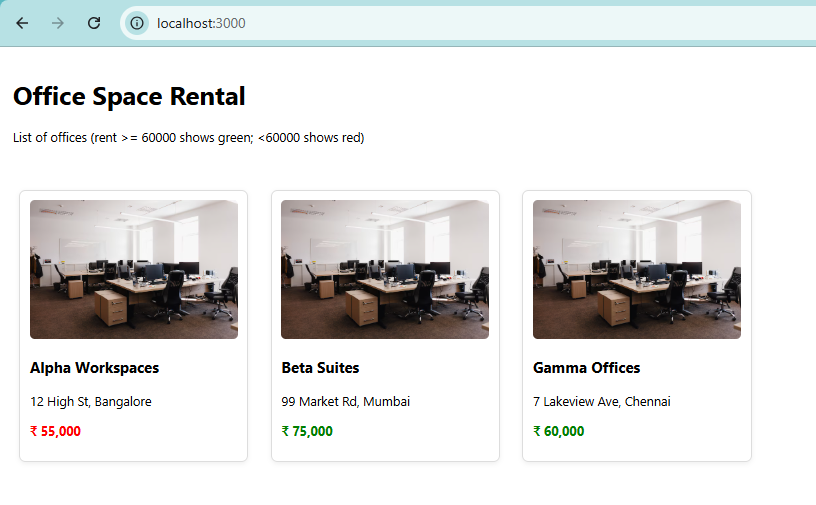
import React from "react";

import { createRoot } from "react-dom/client";

import App from "./App";

createRoot(document.getElementById("root")).render(<App />);

**Output:**



**Docs 11**

**Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.**

**Counter.jsx**

import React from 'react';

class Counter extends React.Component {

constructor(props){

super(props);

this.state = { count: 0, message: '' };

this.increment = this.increment.bind(this);

this.handleOnPress = this.handleOnPress.bind(this);

}

increment(){

this.incrementValue();

this.sayHello();

}

incrementValue(){

this.setState(prev => ({ count: prev.count + 1 }));

}

sayHello(){

this.setState({ message: 'Hello — static message' });

}

sayMessage(msg){

this.setState({ message: msg });

}

handleOnPress(e){

e.preventDefault();

this.setState({ message: 'I was clicked' });

}

render(){

return (

<div>

<h3>Counter: {this.state.count}</h3>

<button onClick={this.increment}>Increase</button>

<button onClick={() => this.setState(prev => ({ count: prev.count - 1 }))}>Decrement</button>

<button onClick={() => this.sayMessage('welcome')}>Say Welcome</button>

<button onClick={this.handleOnPress}>Synthetic OnPress (onClick)</button>

<p>{this.state.message}</p>

</div>

);

}

}

export default Counter;

**CurrencyConverter.jsx**

import React, { useState } from 'react';

export default function CurrencyConvertor(){

const [inr, setInr] = useState('');

const [euro, setEuro] = useState(null);

const RATE = 0.011;

function handleSubmit(e){

e.preventDefault();

const n = parseFloat(inr) || 0;

setEuro((n \* RATE).toFixed(2));

}

return (

<form onSubmit={handleSubmit}>

<label>

INR:

<input value={inr} onChange={e => setInr(e.target.value)} />

</label>

<button type="submit">Convert</button>

{euro !== null && <div>€ {euro}</div>}

</form>

);

}

**App.js**

import './App.css';

import React, { useState } from 'react';

function App() {

const [count, setCount] = useState(5);

const [amount, setAmount] = useState('');

const [currency, setCurrency] = useState('');

const increment = () => {

const newCount = count + 1;

setCount(newCount);

alert(`Hello Member ${newCount}`);

};

const decrement = () => {

setCount(count - 1);

};

const sayWelcome = () => {

alert('welcome');

};

const handleClick = () => {

alert('I was clicked');

};

const handleSubmit = (e) => {

e.preventDefault();

const amtInINR = parseFloat(amount) || 0;

const rates = {

USD: 0.012, EUR: 0.011, JPY: 1.74, GBP: 0.0093, CHF: 0.011,

CAD: 0.016, AUD: 0.018, CNY: 0.088, INR: 1, RUB: 1.05,

KRW: 16.0, SGD: 0.016, NZD: 0.019, HKD: 0.093, BRL: 0.066,

ZAR: 0.21, MXN: 0.21, RON: 0.056,

KWD: 0.0036, BHD: 0.0045, OMR: 0.0046, JOD: 0.0085,

LBP: 1063, IRR: 506, VND: 296.5, SLL: 254.2, LAK: 243.6,

IDR: 185.5, UZS: 146.0

};

const code = currency.toUpperCase();

if (!rates[code]) {

alert(`Unknown or unsupported currency: ${currency}`);

return;

}

const converted = amtInINR \* rates[code];

try {

const formatted = new Intl.NumberFormat('en', {

style: 'currency',

currency: code

}).format(converted);

alert(`Converting to ${code} Amount is ${formatted}`);

} catch {

alert(`Unable to format currency: ${currency}`);

}

};

return (

<div style={{ padding: '20px' }}>

<h2>{count}</h2>

<button onClick={increment}>Increment</button><br />

<button onClick={decrement}>Decrement</button><br />

<button onClick={sayWelcome}>Say welcome</button><br />

<button onClick={handleClick}>Click on me</button>

<h2 style={{ color: 'green' }}>Currency Convertor!!!</h2>

<form onSubmit={handleSubmit}>

<label>

Amount (INR):

<input

type="text"

value={amount}

onChange={(e) => setAmount(e.target.value)}

/>

</label><br />

<label>

Currency Code:

<input

type="text"

value={currency}

onChange={(e) => setCurrency(e.target.value)}

placeholder="e.g., USD, EUR, JPY"

/>

</label><br />

<button type="submit">Submit</button>

</form>

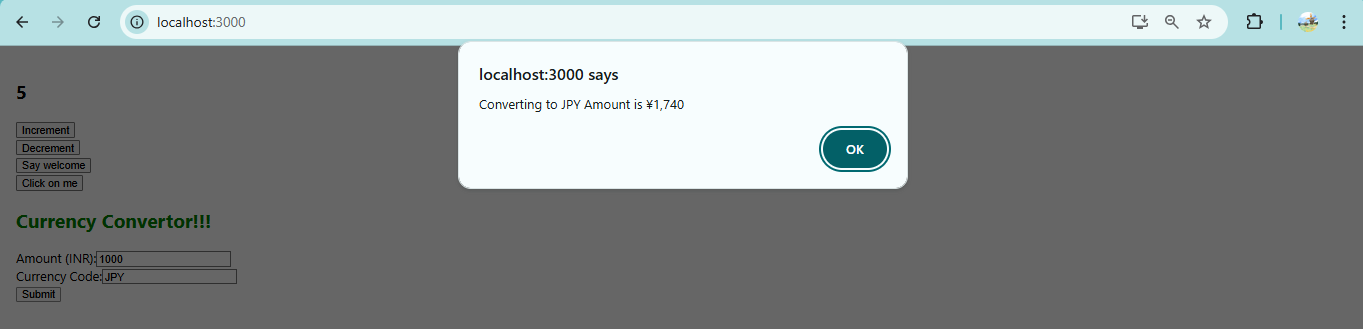
</div>

);

}

export default App;

**Output:**



**Docs 12**

**Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.**

**App.js**

import './App.css';

import React, { useState } from 'react';

function GuestPage() {

return (

<div>

<h2>Flight Details</h2>

<ul>

<li>Flight AI101 - Delhi → New York</li>

<li>Flight BA202 - London → Mumbai</li>

</ul>

<p>Please login to book tickets.</p>

</div>

);

}

function UserPage() {

return (

<div>

<h2>Welcome Back!</h2>

<h2>Flight Booking</h2>

<form>

<label>

Select Flight:

<select>

<option>AI101 - Delhi → New York</option>

<option>BA202 - London → Mumbai</option>

</select>

</label><br />

<label>

Passenger Name:

<input type="text" />

</label><br />

<button type="submit">Book Ticket</button>

</form>

</div>

);

}

export default function App() {

const [isLoggedIn, setIsLoggedIn] = useState(false);

return (

<div style={{ padding: '20px' }}>

<h1>Ticket Booking App</h1>

{isLoggedIn ? <UserPage /> : <GuestPage />}

{isLoggedIn ? (

<button onClick={() => setIsLoggedIn(false)}>Logout</button>

) : (

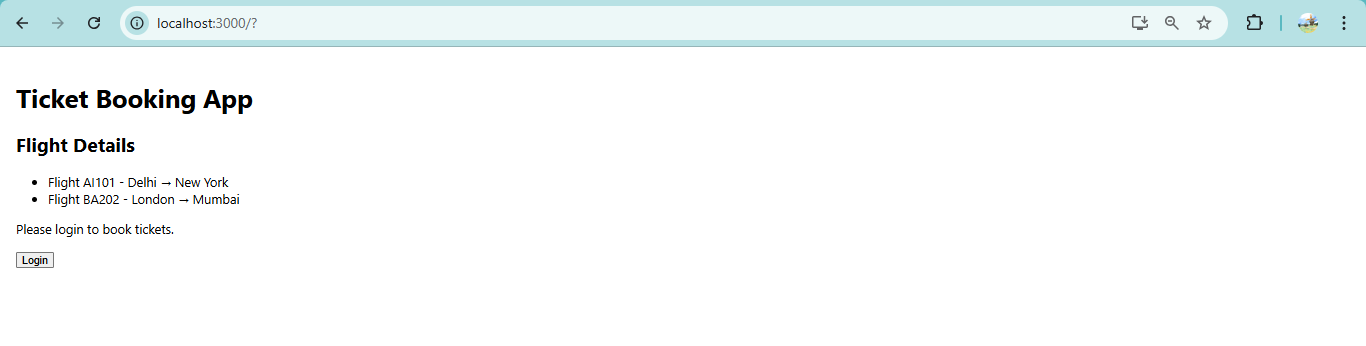
<button onClick={() => setIsLoggedIn(true)}>Login</button>

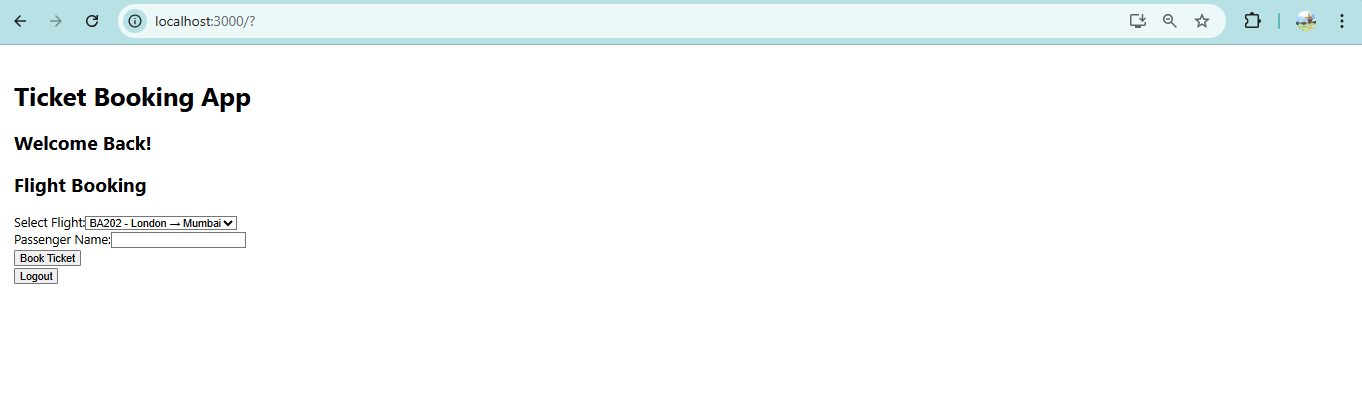
)}

</div>

);

}





**Docs 13**

**Create a React App named “bloggerapp” in with 3 components.**

⦁ **Book Details**

export default function BookDetails() {

return (

<div>

<h2>Book Details</h2>

<ul>

<li>Title: React Mastery</li>

<li>Author: XYZ</li>

<li>Price: $20</li>

</ul>

</div>

);

}

⦁ **Blog Details**

export default function BlogDetails() {

return (

<div>

<h2>Blog Details</h2>

<p>Title: "Learning React step-by-step"</p>

<p>Author: Jane Smith</p>

</div>

);

}

**⦁** **Course Details**

export default function CourseDetails() {

return (

<div>

<h2>Course Details</h2>

<p>Course: React Development Bootcamp</p>

<p>Duration: 6 weeks</p>

</div>

);

}

**App.js**

import './App.css';

import React from "react";

function CourseDetails({ courses }) {

return (

<div className="section">

<h2>Course Details</h2>

{courses.map((course, index) => (

<div key={index}>

<strong>{course.name}</strong>

<p>{course.date}</p>

</div>

))}

</div>

);

}

function BookDetails({ books }) {

return (

<div className="section">

<h2>Book Details</h2>

{books.map((book, index) => (

<div key={index}>

<strong>{book.title}</strong>

<p>{book.price}</p>

</div>

))}

</div>

);

}

function BlogDetails({ blogs }) {

return (

<div className="section">

<h2>Blog Details</h2>

{blogs.map((blog, index) => (

<div key={index}>

<strong>{blog.title}</strong>

<p>{blog.author}</p>

<p>{blog.content}</p>

</div>

))}

</div>

);

}

export default function App() {

const courses = [

{ name: "Angular", date: "4/5/2021" },

{ name: "React", date: "6/3/2021" }

];

const books = [

{ title: "Master React", price: "670" },

{ title: "Deep Dive into Angular 11", price: "800" },

{ title: "Mongo Essentials", price: "450" }

];

const blogs = [

{ title: "React Learning", author: "Stephen Biz", content: "Welcome to learning React!" },

{ title: "Installation", author: "Schwezdenier", content: "You can install React from npm." }

];

return (

<div className="container">

<CourseDetails courses={courses} />

<BookDetails books={books} />

<BlogDetails blogs={blogs} />

</div>

);

}

**App.css**

.container {

display: flex;

justify-content: space-between;

}

.section {

padding: 0 20px;

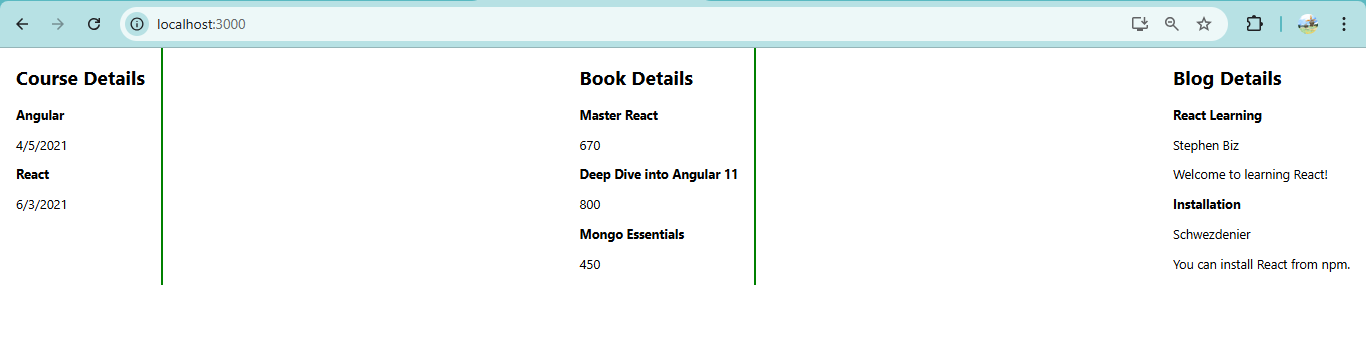
}

.section:not(:last-child) {

border-right: 3px solid green;

}

**Output:**



**Docs 14**

**Developers of Apps Centric Solutions have created an employee management application which supports light and dark themes for the buttons. The current solution uses the react state and props to provide the theme name to be used from App component to Employee List component and from there to Employee Card component. Quality assurance team analyzed the solutions and found the technique being used to be a substandard one. React architect suggested to use the react context API to share the theme name with nested child components instead of passing them down using props from the parent component.**

**ThemeContext.js**

import { createContext } from "react";

// Default : light

const ThemeContext = createContext('light');

export default ThemeContext;

**App.js**

import logo from './logo.svg';

import './App.css';

import React, { useState } from "react";

import EmployeesList from "./EmployeesList";

import ThemeContext from "./ThemeContext";

export default function App() {

const [theme, setTheme] = useState('light');

return (

<ThemeContext.Provider value={theme}>

<div className={`app-container ${theme}`}>

<h1>Employee Management</h1>

<button onClick={() => setTheme(theme === 'light' ? 'dark' : 'light')}>

Toggle Theme

</button>

{}

<EmployeesList />

</div>

</ThemeContext.Provider>

);

}

**EmployeesList.js**

import React from "react";

import EmployeeCard from "./EmployeeCard";

export default function EmployeesList() {

const employees = [

{ id: 1, name: "Prabha", role: "Developer" },

{ id: 2, name: "Lakshmi", role: "Designer" }

];

return (

<div>

{employees.map(emp => (

<EmployeeCard key={emp.id} employee={emp} />

))}

</div>

);

}

**EmployeeCard.js**

import React, { useContext } from "react";

import ThemeContext from "./ThemeContext";

export default function EmployeeCard({ employee }) {

const theme = useContext(ThemeContext);

return (

<div className={`employee-card ${theme}`}>

<h3>{employee.name}</h3>

<p>{employee.role}</p>

<button className={`btn-${theme}`}>View Profile</button>

</div>

);

}

**App.css**

.app-container.light {

background: #fff;

color: #000;

}

.app-container.dark {

background: #222;

color: #fff;

}

.btn-light {

background: #ddd;

color: #000;

}

.btn-dark {

background: #555;

color: #fff;

}

.employee-card {

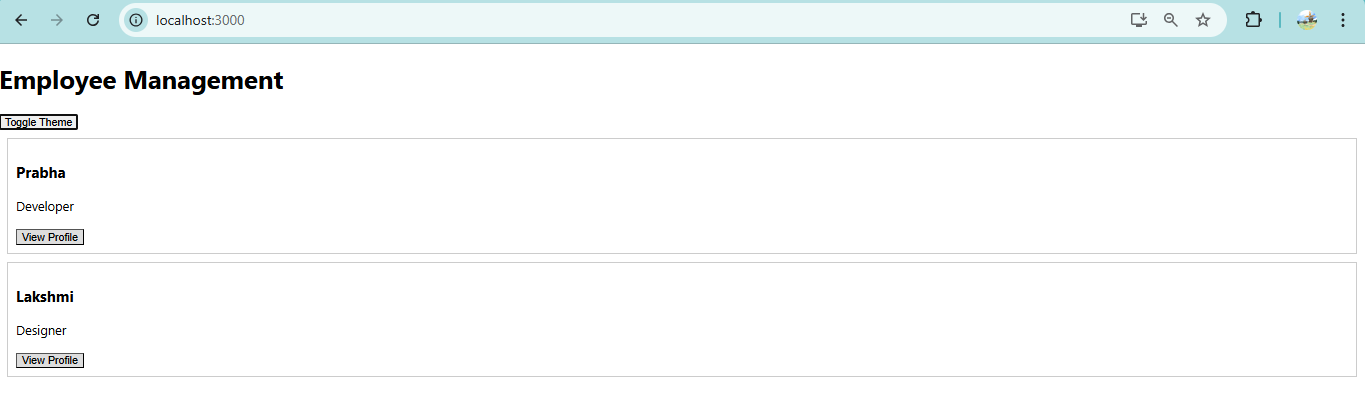
border: 1px solid #ccc;

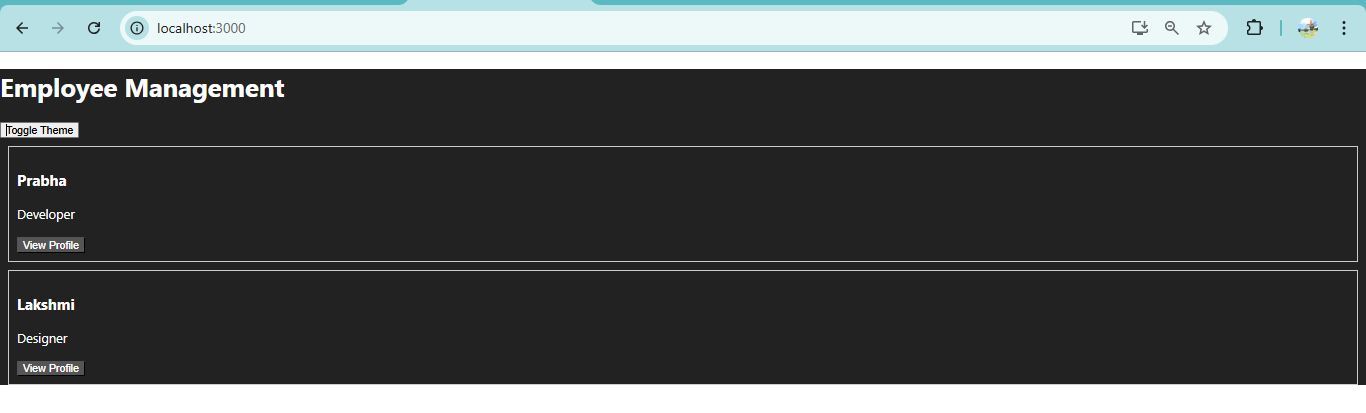
margin: 10px;

padding: 10px;

}

**Output**





**Docs 15**

**Create a React App named “ticketraisingapp” which will help to raise a complaint and get it resolved.**

**App.js**

import React from 'react';

import ComplaintRegister from './components/ComplaintRegister';

import './App.css';

function App() {

return (

<div className="App" style={{ padding: 20 }}>

<h2 style={{

color: 'red',

marginTop: '250px',

marginBottom: '40px',

textAlign: 'center'

}}>

Register your complaints here!!!

</h2>

<ComplaintRegister />

</div>

);

}

export default App;

**App.css**

.App { font-family: Arial, Helvetica, sans-serif; color: #0f0f0f; }

**ComplaintRegister.js**

import React, { useState } from 'react';

const generateRef = () => {

const timePart = Date.now().toString(36).toUpperCase();

const randPart = Math.random().toString(36).slice(2,8).toUpperCase();

return `REF-${timePart.slice(-6)}-${randPart}`;

};

export default function ComplaintRegister() {

const [name, setName] = useState('');

const [complaint, setComplaint] = useState('');

const [refNumber, setRefNumber] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (!name.trim() || !complaint.trim()) {

alert('Please enter both employee name and complaint.');

return;

}

const ref = generateRef();

alert(

`Thanks ${name}!\nYour Complaint was Submitted.\nTransaction ID is: ${ref}`

);

setRefNumber(ref);

setName('');

setComplaint('');

};

return (

<div style={{maxWidth:600, margin:'0 auto'}}>

<form onSubmit={handleSubmit} className="complaint-form">

<div style={{marginBottom:12}}>

<label htmlFor="employeeName" style={{display:'block', marginBottom:6}}>Employee Name</label>

<input

id="employeeName"

type="text"

value={name}

onChange={(e) => setName(e.target.value)}

placeholder="Enter employee name"

style={{width:'100%', padding:8, boxSizing:'border-box'}}

/>

</div>

<div style={{marginBottom:12}}>

<label htmlFor="complaint" style={{display:'block', marginBottom:6}}>Complaint</label>

<textarea

id="complaint"

value={complaint}

onChange={(e) => setComplaint(e.target.value)}

placeholder="Describe the complaint..."

rows={5}

style={{width:'100%', padding:8, boxSizing:'border-box'}}

/>

</div>

<button type="submit" style={{padding:'10px 16px'}}>Submit Complaint</button>

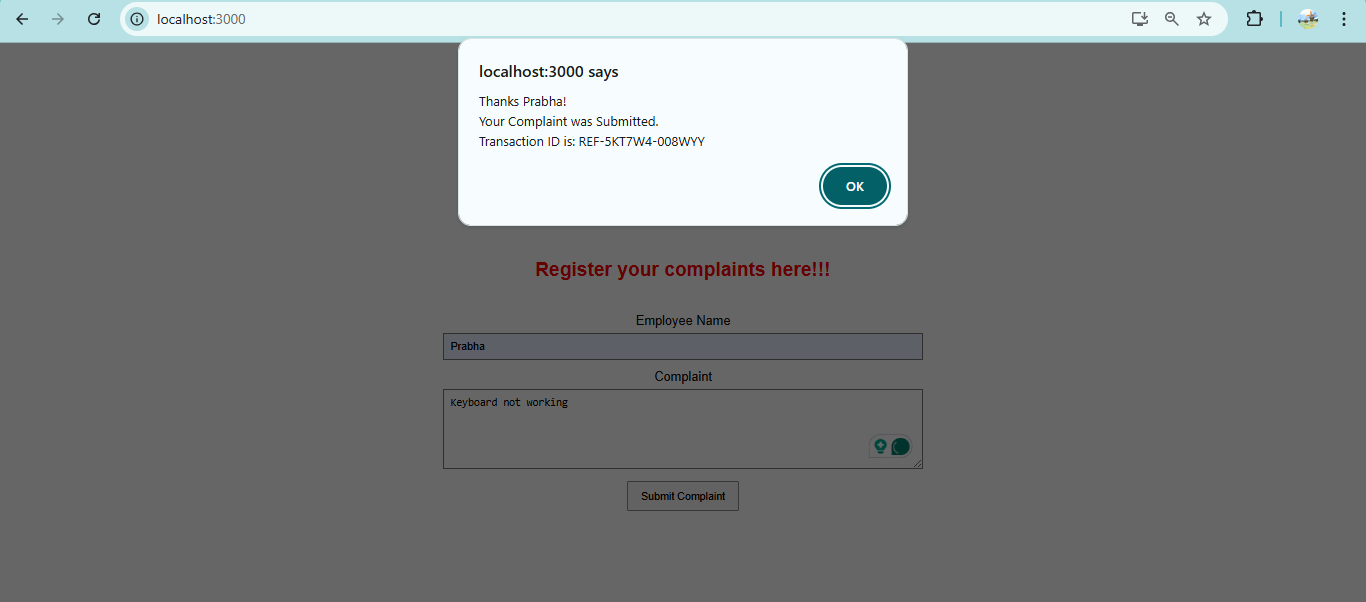
</form>

</div>

);

}

**Output:**



**Docs 16**

**Create a React App named “mailregisterapp” which will have a component named “register.js”.**

**App.js**

import logo from './logo.svg';

import './App.css';

import React from 'react';

import Register from './register';

function App() {

return (

<div>

<Register />

</div>

);

}

export default App;

**Register.js**

import React, { useState } from 'react';

export default function Register() {

const [formData, setFormData] = useState({

name: '',

email: '',

password: ''

});

const [errors, setErrors] = useState({});

const handleChange = (e) => {

setFormData({

...formData,

[e.target.name]: e.target.value

});

};

const validate = () => {

let tempErrors = {};

if (formData.name.length < 5) {

tempErrors.name = "Name should have at least 5 characters";

}

if (!formData.email.includes('@') || !formData.email.includes('.')) {

tempErrors.email = "Email should contain @ and .";

}

else if (formData.password.length < 8) {

tempErrors.password = "Password should have at least 8 characters";

}

setErrors(tempErrors);

return Object.keys(tempErrors).length === 0;

};

const handleSubmit = (e) => {

e.preventDefault();

if (validate()) {

alert(

`Form submitted successfully!`

);

} else {

let errorMsg = "";

Object.values(errors).forEach(err => {

errorMsg += `${err}\n`;

});

alert(errorMsg);

}

};

return (

<div style={{ maxWidth: "400px", margin: "auto", marginTop: "250px" }}>

<h2>Register</h2>

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

/>

{/\*errors.name && <p style={{ color: "red" }}>{errors.name}</p>\*/}

</div>

<div>

<label>Email:</label>

<input

type="text"

name="email"

value={formData.email}

onChange={handleChange}

/>

{/\*errors.email && <p style={{ color: "red" }}>{errors.email}</p>\*/}

</div>

<div>

<label>Password:</label>

<input

type="password"

name="password"

value={formData.password}

onChange={handleChange}

/>

{/\*errors.password && <p style={{ color: "red" }}>{errors.password}</p>\*/}

</div>

<button type="submit">Register</button>

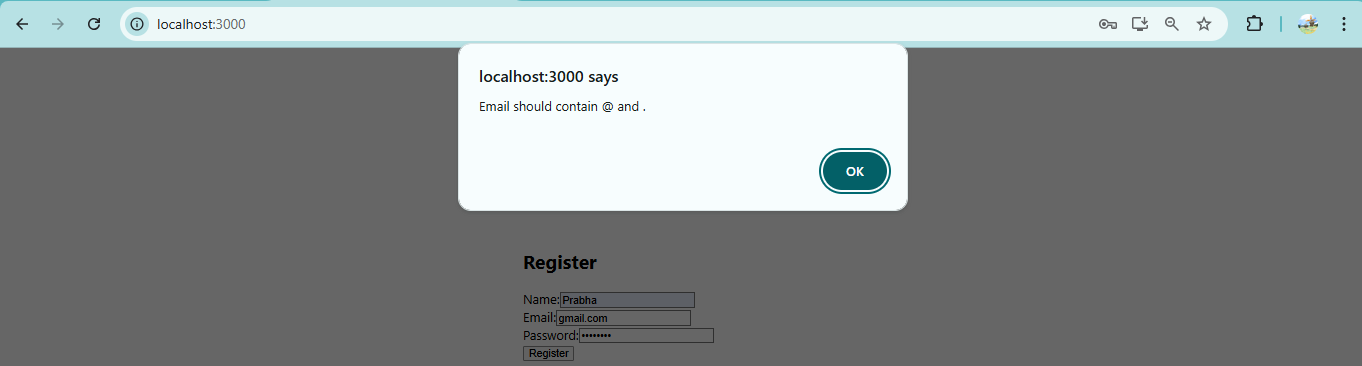
</form>

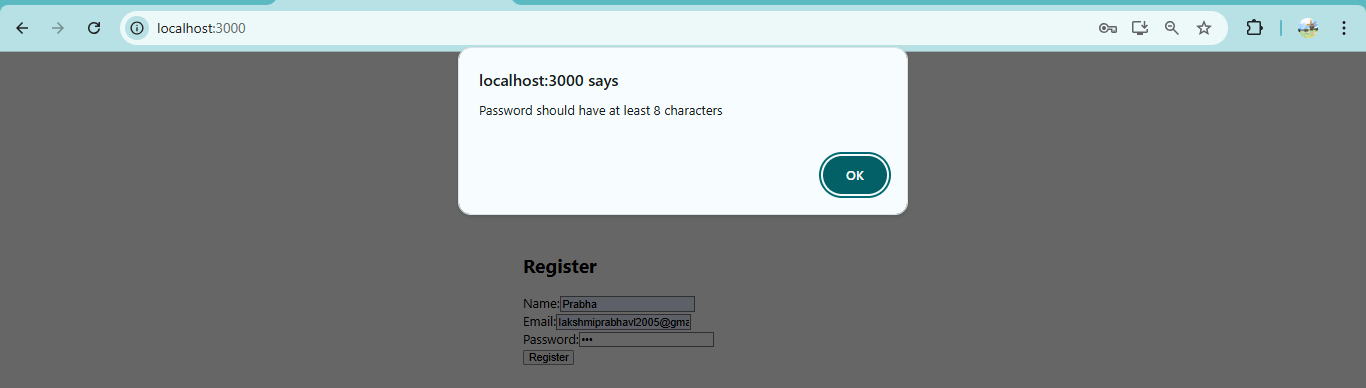
</div>

);

}

**Output:**





**Docs 17**

**Create a React Application “fetchuserapp” which will retrieve the user details from https://api.randomuser.me/ and display the title, firstname and image of a user.**

**Getuser.js**

import React, { Component } from 'react';

class Getuser extends Component {

state = {

loading: true,

error: null,

user: null,

};

async componentDidMount() {

try {

const res = await fetch('https://api.randomuser.me/');

if (!res.ok) throw new Error(`HTTP ${res.status}`);

const data = await res.json();

const user = data.results && data.results[0];

this.setState({ user, loading: false });

} catch (err) {

this.setState({ error: err.message || 'Fetch error', loading: false });

}

}

render() {

const { loading, error, user } = this.state;

if (loading) return <div>Loading user…</div>;

if (error) return <div style={{ color: 'red' }}>Error: {error}</div>;

if (!user) return <div>No user returned</div>;

const title = user.name?.title || '';

const first = user.name?.first || '';

const img = user.picture?.large || user.picture?.medium || user.picture?.thumbnail || '';

return (

<div className="getuser" style={{ textAlign: 'center', marginTop: 40 }}>

<h2>{title} {first}</h2>

<img src={img} alt={`${title} ${first}`} style={{ borderRadius: '50%', width: 150, height: 150, objectFit: 'cover' }} />

</div>

);

}

}

export default Getuser;

**App.js**

import logo from './logo.svg';

import './App.css';

import React from 'react';

import Getuser from './components/Getuser';

import './App.css';

function App() {

return (

<div className="App">

<h1 style={{ textAlign: 'center', marginTop: 20 }}>Random User</h1>

<Getuser />

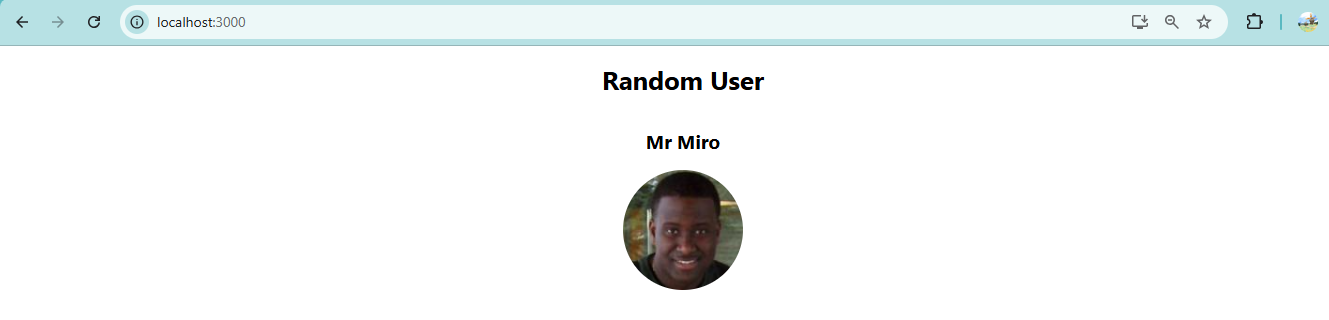
</div>

);

}

export default App;

**Output:**



**Docs 18**

**My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using React component. You are assigned the task of unit testing the component to ensure it’s free of bugs.**

**Setup.test.js**

import '@testing-library/jest-dom';

**Cohort.js**

export const CohortData = [

{ code: 'C101', name: 'React Beginners', members: 25 },

{ code: 'C102', name: 'Advanced React', members: 18 }

];

**CohortDetails.js**

import React from 'react';

export default function CohortDetails({ cohort }) {

if (!cohort) {

return <h3>No Cohort Selected</h3>;

}

return (

<div>

<h3>{cohort.code}</h3>

<p>{cohort.name}</p>

<p>Members: {cohort.members}</p>

</div>

);

}

**CohortDetails.test.js**

import React from 'react';

import { render, screen } from '@testing-library/react';

import CohortDetails from './CohortDetails';

import { CohortData } from './Cohort';

describe('Cohort Details Component', () => {

test('renders without crashing', () => {

render(<CohortDetails />);

expect(screen.getByText(/No Cohort Selected/i)).toBeInTheDocument();

});

test('renders cohort data', () => {

const sample = CohortData[0];

render(<CohortDetails cohort={sample} />);

expect(screen.getByText(sample.code)).toBeInTheDocument();

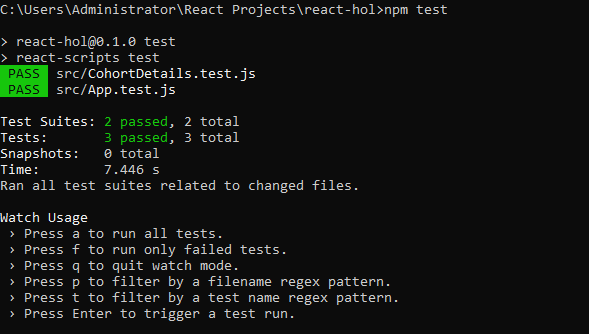
expect(screen.getByText(sample.name)).toBeInTheDocument();

expect(screen.getByText(`Members: ${sample.members}`)).toBeInTheDocument();

});

});

**Output:**



**Docs 19**

**As an intern at OpenAI you are assigned the task of creating and testing a React application which will fetch and display a list of repository names for a given user.**

**App.test.js**

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders loading message', () => {

render(<App />);

const loadingElement = screen.getByText(/loading/i);

expect(loadingElement).toBeInTheDocument();

});

**Gitclient.js**

import axios from 'axios';

export default class GitClient {

static async getRepositories(username) {

const res = await axios.get(`https://api.github.com/users/${username}/repos`);

return res.data.map(r => r.name);

}

}

**GitClient.test.js**

import axios from 'axios';

import GitClient from './GitClient';

jest.mock('axios');

test('should return repository names for techiesyed', async () => {

const fake = [{name:'repo-one'},{name:'repo-two'}];

axios.get.mockResolvedValue({ data: fake });

const names = await GitClient.getRepositories('techiesyed');

expect(axios.get).toHaveBeenCalledWith('https://api.github.com/users/techiesyed/repos');

expect(names).toEqual(['repo-one','repo-two']);

});

**Output:**

