**INTRODUCTION**

The ShigoTo job searching portal is an innovative and comprehensive web application designed to facilitate the connection between job seekers and employers. In the modern job market, finding the right job can be a daunting task, and employers often struggle to find the right talent. ShigoTo aims to address these challenges by providing a user-friendly platform where job seekers can easily find and apply for job opportunities, and employers can efficiently post job listings and manage applications.

Developed using the React framework for the frontend, ShigoTo offers an intuitive and interactive interface that enhances the user experience. The portal includes a range of features such as advanced job search functionality, domain-specific job listings, and an employer section for job postings and hackathon management. Additionally, ShigoTo provides access to popular courses that help users upskill and stay competitive in their respective fields.

With a dedicated admin page for managing job postings, ShigoTo ensures that the process of job searching and recruitment is seamless, efficient, and effective. The platform is designed to cater to the needs of both job seekers and employers, making it a versatile tool in the employment ecosystem.

**OBJECTIVE**

The primary objective of ShigoTo is to bridge the gap between job seekers and employers by creating a seamless and efficient job search experience. The platform aims to provide job seekers with easy access to a wide range of job opportunities, tailored to their specific skills and preferences. For employers, ShigoTo offers tools to streamline the job posting process, manage applications, and organize hackathons to discover new talent. Additionally, the platform seeks to support continuous learning and professional development by offering a selection of popular courses for users to enhance their skills and improve their employability. Overall, ShigoTo is designed to foster professional connections, improve job matching, and contribute to a more dynamic and effective job market.

**SYSTEM SPECIFICATION :**

* **HARDWARE SPECIFICATION :**
* System : Pentium IV 2.4GHz.
* Hard Disk : 500GB.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 4GB.
* Keyboard : 101 Keyboard.
* **SOFTWARE SPECIFICATION :**
* Operating System : Windows 8,10,11,MAC OS.
* Front End : React JS.
* Back End : SPRING BOOT With MY SQL
* Server : SONAR CLOUD

**EXISTING SYSTEM :**

**1.** **Traditional Job Search Platforms:**

* **Complex Navigation**: Often complex interfaces that can be overwhelming for users.
* **Limited Interactions**: Insufficient communication channels between job seekers and employers.
* **Generic Listings**: Broad job categories without specialized focus
* **Inefficient Filtering**: Basic search and filter options leading to suboptimal job matching.

**2. Basic Online Job Platforms :**

* **High Fees**: Significant fees for job postings and premium services are a common drawback, deterring employers from using these platforms.
* **Limited Reach**: Some platforms do not cover a wide audience, limiting their effectiveness in job market penetration.
* **Fragmented Market**: Job listings are often spread across multiple platforms, leading to fragmented job searches and missing opportunities

**PROPOSED SYSTEM :**

**User-Friendly Interface:**

* **Intuitive Design**: A modern and straightforward design ensures ease of use for both job seekers and employers.
* **Multi-Language Support**: Features support for multiple languages to cater to a diverse user base.

**Advanced Job Search and Filtering:**

* **Real-Time Updates**: Immediate reflection of new job postings and application statuses ensures that users have access to the latest opportunities.
* **Customizable Filters**: Enhanced filters allow users to narrow down job searches by role, type, location, and experience level.

**Secure Transactions:**

* **Payment Gateway Integration**: Secure and versatile payment options are available for premium features and job postings.
* **SSL Encryption:** Ensures the protection of user data and transactions.

**Direct Communication:**

* **Built-In Messaging:** Enables seamless communication between job seekers and employers for direct engagement.
* **Real-Time Notifications:** Users receive alerts for new messages, job postings, and important updates.

**Admin Dashboard:**

* **Job Management:** Admins can efficiently post, update, and delete job listings.
* **Application Tracking:** Comprehensive tracking of applications from submission to hiring decisions.

**Course Recommendations:**

* **Popular Courses**: Users have access to a range of courses to boost their skills, with detailed information on course content, duration, and pricing.
* **Course Details:** Offers clear descriptions, ratings, and the option to purchase or enroll in courses directly through the platform.

**MODULES :**

**1. User Interface Module:**

* **Registration and Login:**
* Secure and straightforward user registration and authentication
* **Dashboard:**
* User-friendly dashboard displaying key features and updates.
* Multi-language support for a diverse user base.

**2. Job Search Module:**

* **Job Listings:**
* Real-time job listings with detailed descriptions and advanced filtering options.
* **Top Hiring Companies:**
* Spotlight on top employers and their job openings.

**3. Employer Module:**

* **Job Posting:**

Tools for employers to post job openings, manage listings, and track applications.

* **Hackathon Management:**
* Features for planning and conducting hackathons, including scheduling and participant management.

**4. Admin Module:**

* **Job Management:**
* Admin controls for posting, updating, and deleting job listings.
* **Application Review:**
* Administrative tools for reviewing and managing job applications.

**4. Course Module:**

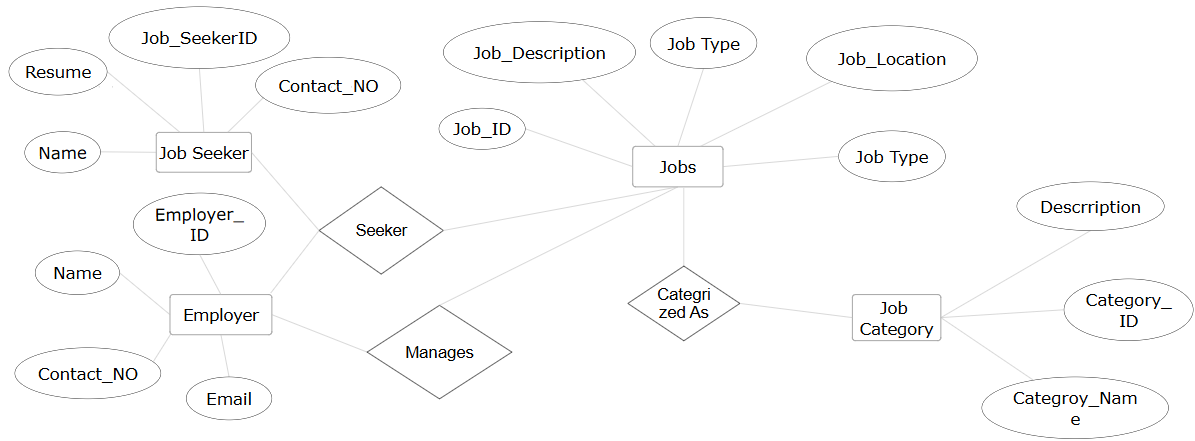
* **Popular Courses:**
* Listings of recommended courses with purchase and enrollment options.
* **Course Details:**
* Information on course content, pricing, and duration.

**4. Communication Module:**

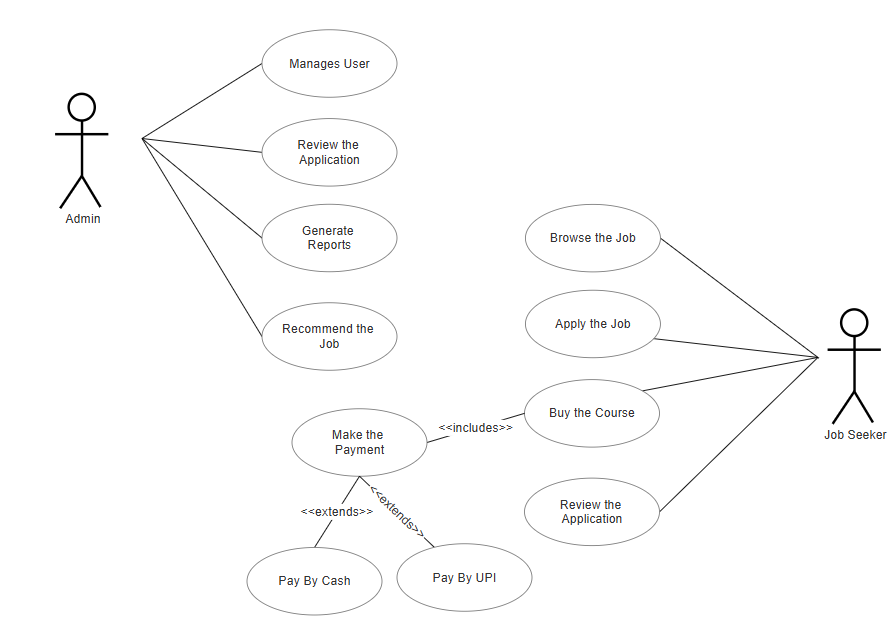
* **Messaging System:**
* Built-in chat system for direct communication between job seekers and employers.
* **Notifications:**
* Real-time alerts for important updates and interactions.

**SYSTEM DIAGRAM :**

* **ER DIAGRAM:**

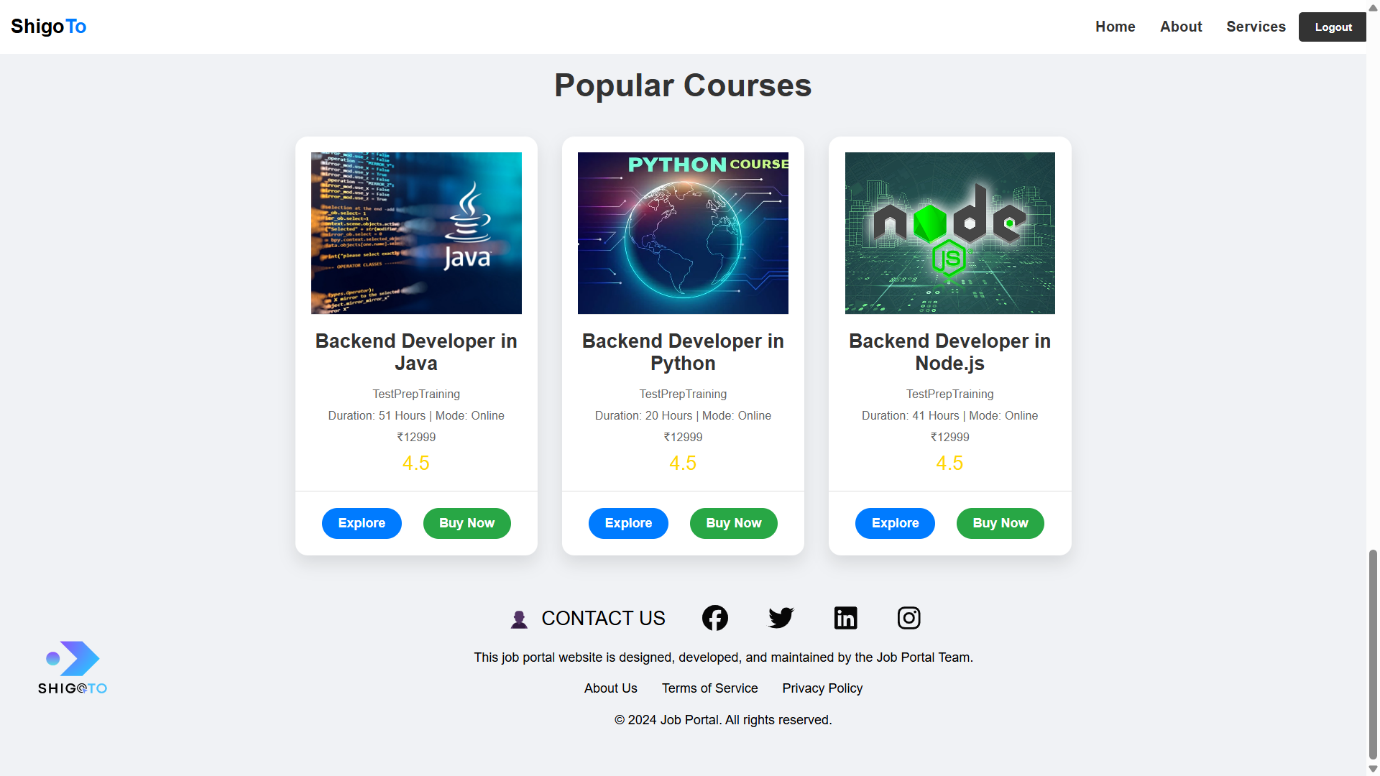
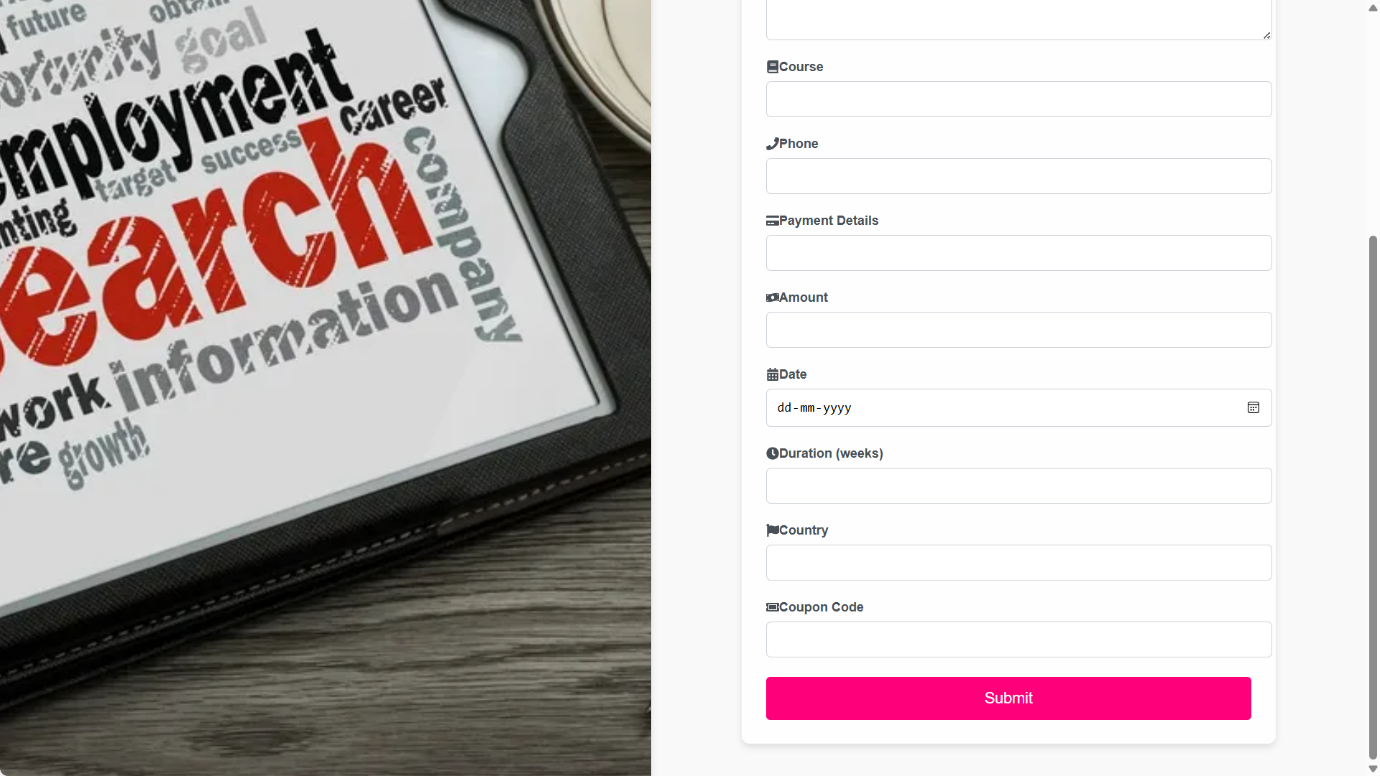
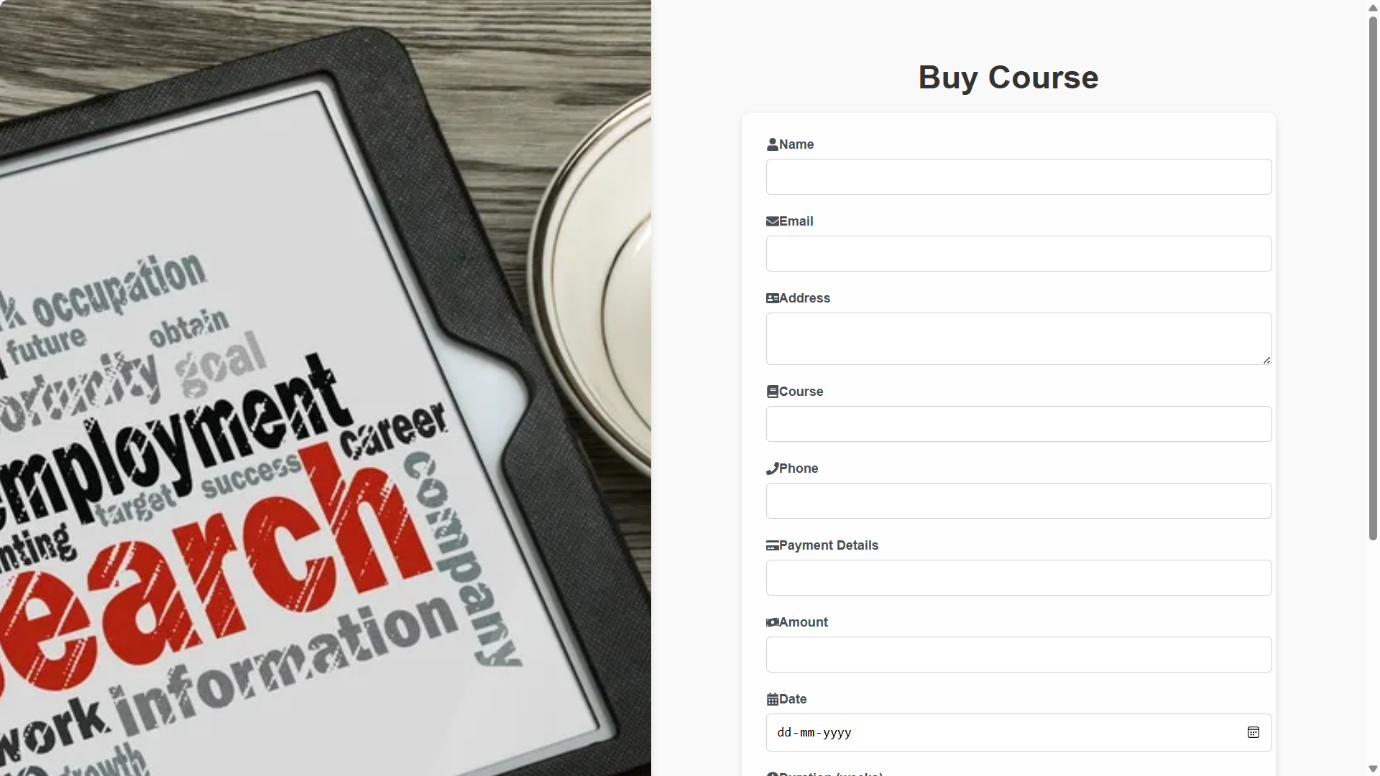
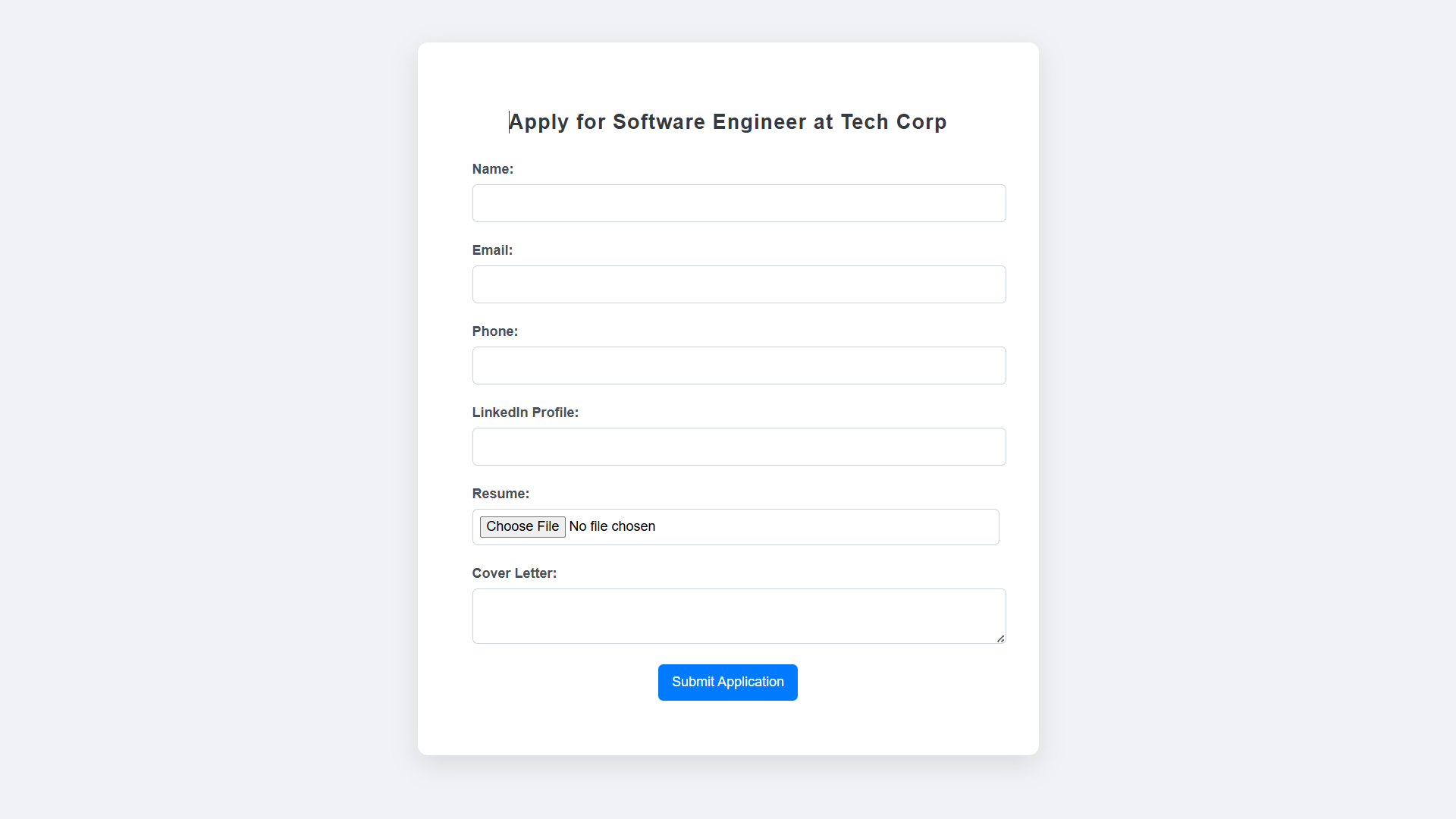
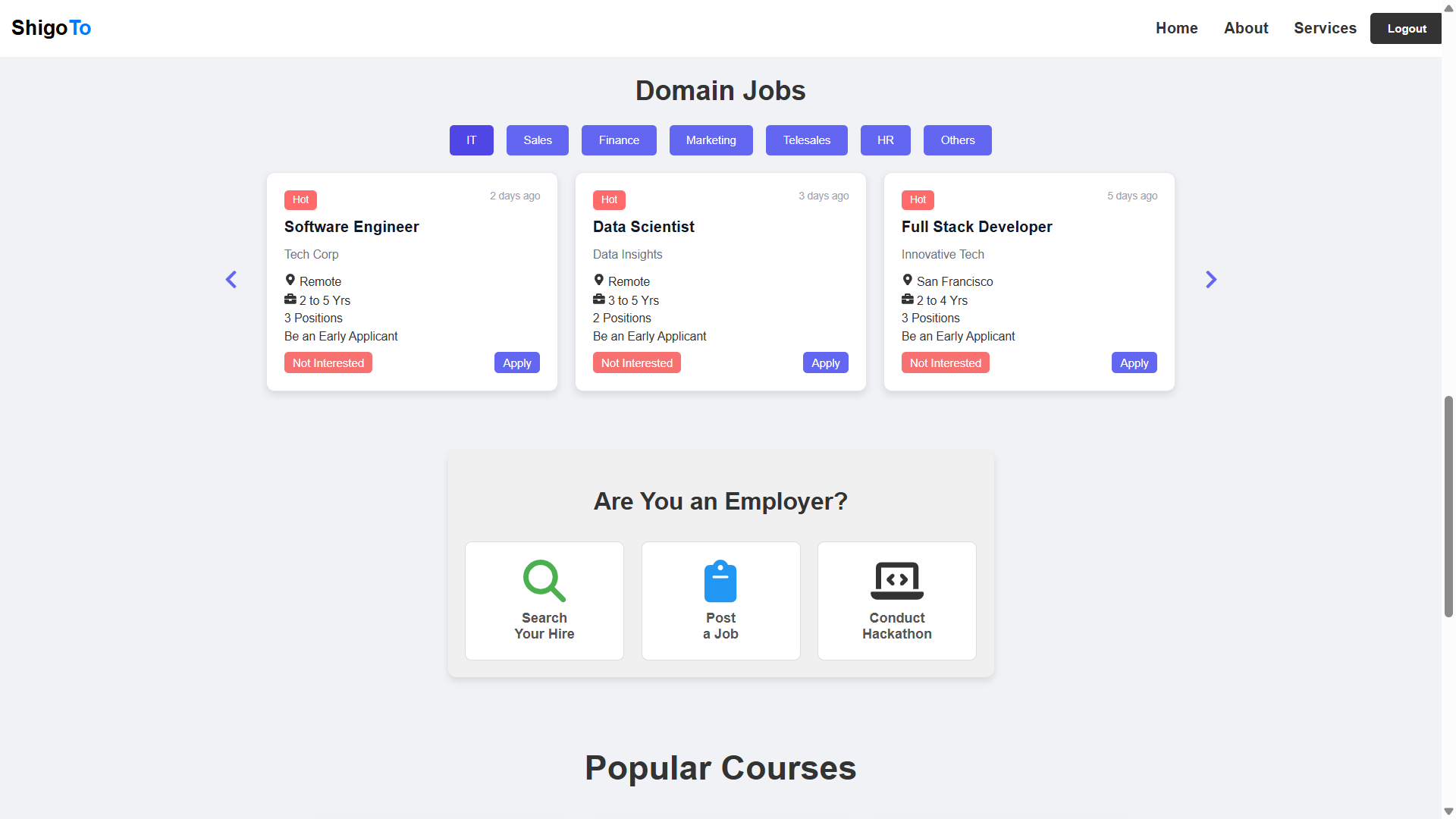
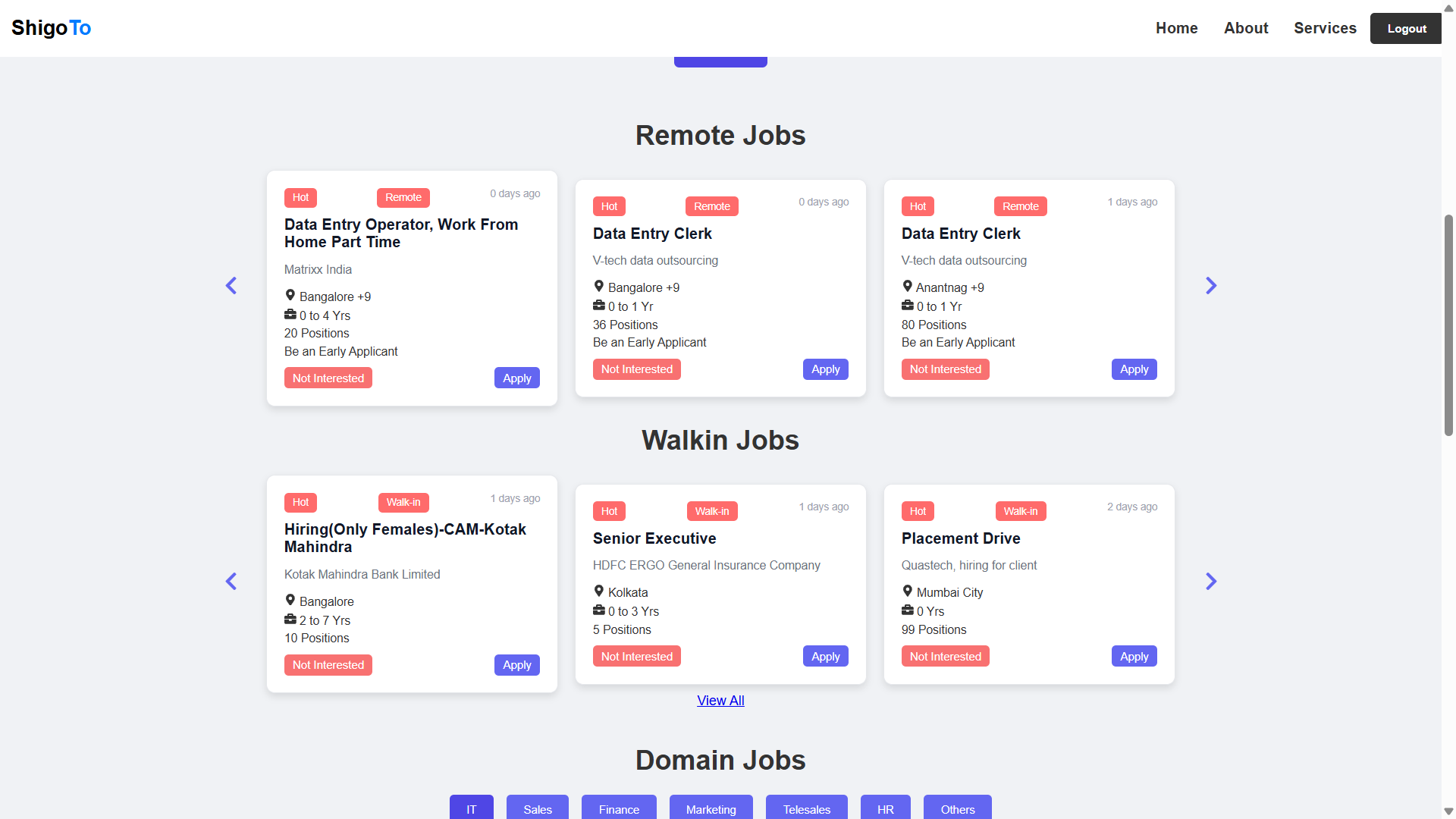


* **UML DIAGRAM:**

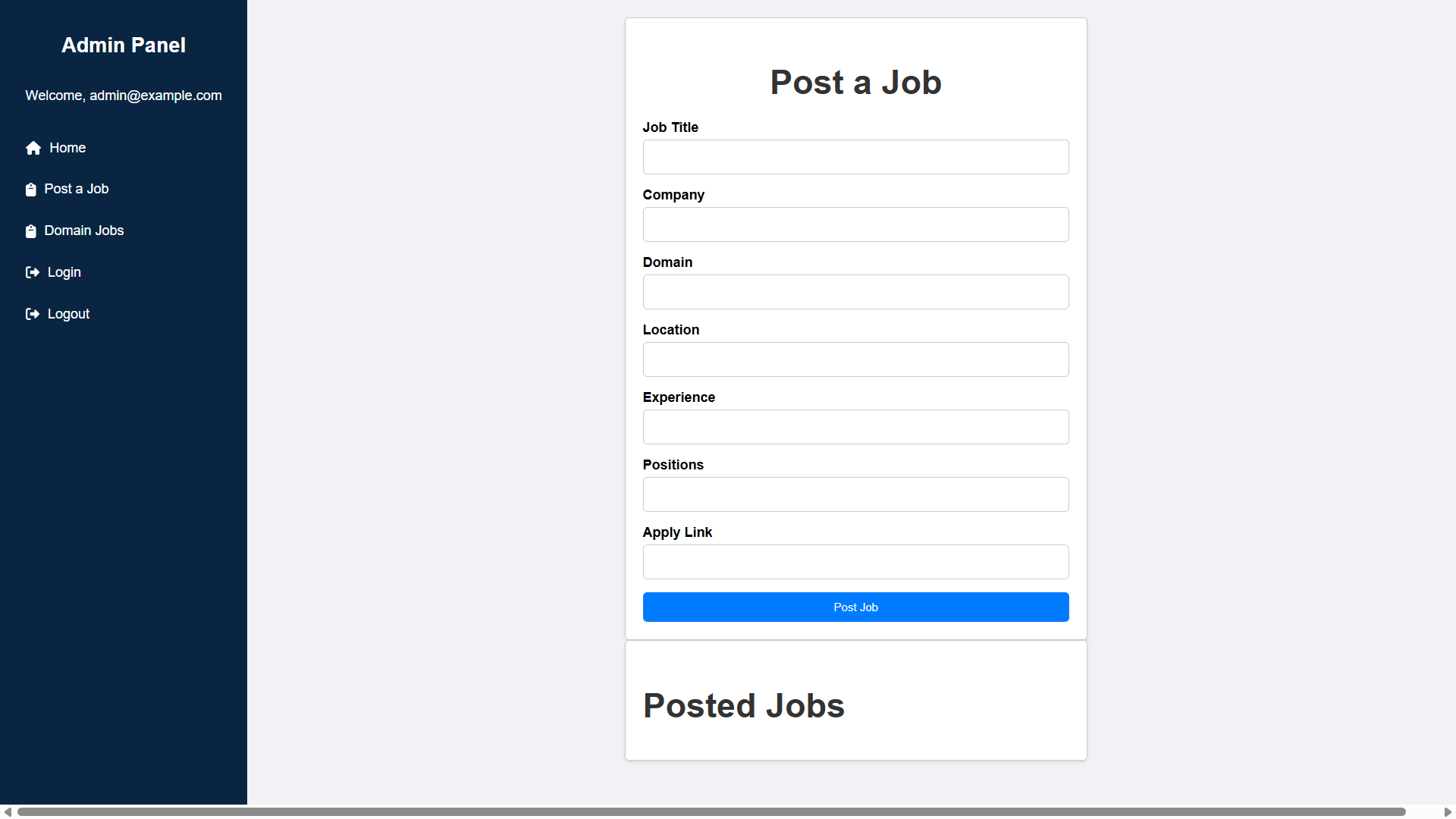
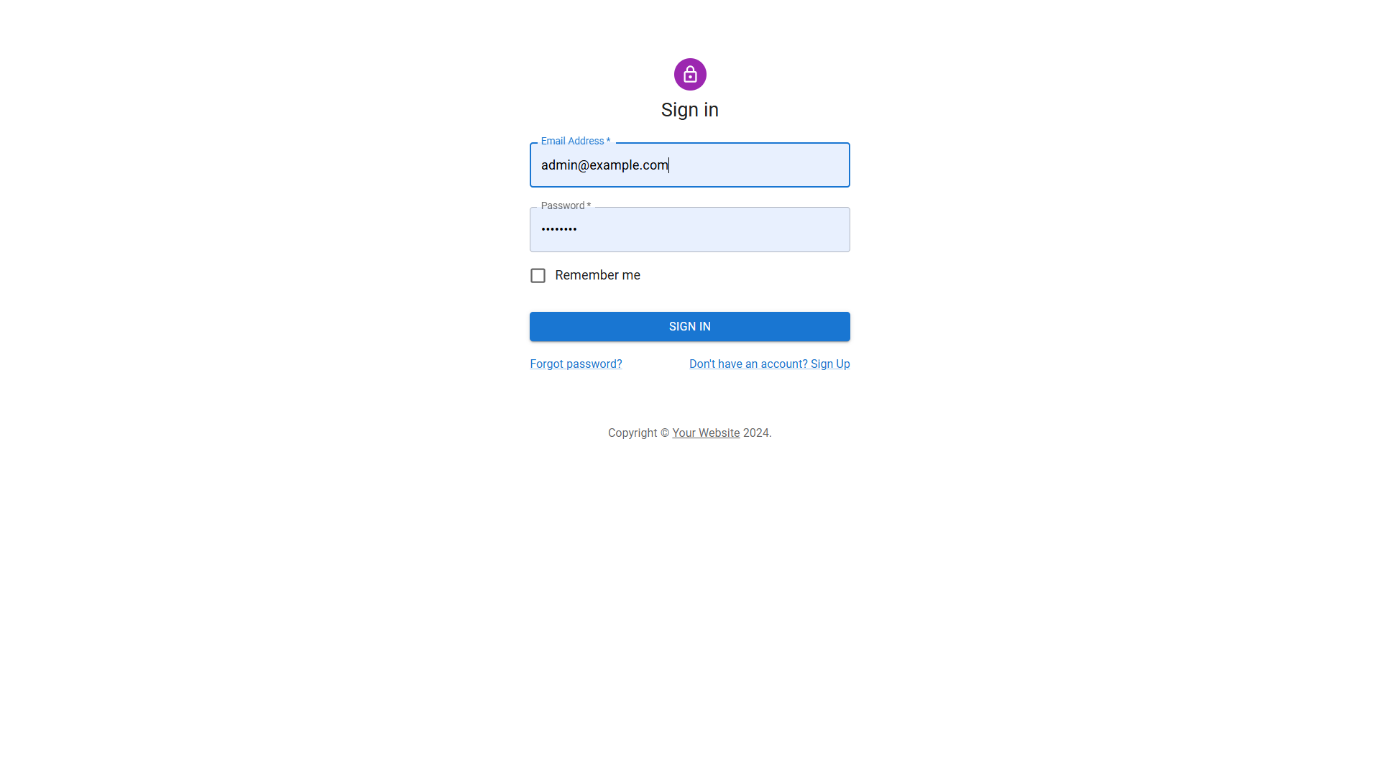


**PICTURES :**

* **USER – SIDE :**

****

* **ADMIN – SIDE :**

****

**CODING :**

* **USER SIDE**
* **Login Page :**

import React, { useState } from 'react';

import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';

import { faUserCircle } from '@fortawesome/free-solid-svg-icons';

import '../../Assest/css/Login.css';

function Login({ onLogin, switchToRegister }) {

const [username, setUsername] = useState('');

const [password, setPassword] = useState('');

const handleSubmit = async (e) => {

e.preventDefault();

const response = await fetch('http://localhost:8080/api/users/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ username, password })

});

if (response.ok) {

onLogin(username, password);

} else {

alert('Invalid username or password');

}

};

return (

<div className="login-form-container">

<h2>Login</h2>

<FontAwesomeIcon icon={faUserCircle} className="user-icon" />

<form onSubmit={handleSubmit}>

<input

type="text"

placeholder="Username"

value={username}

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

required

/>

<input

type="password"

placeholder="Password"

value={password}

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

required

/>

<center><button type="submit">Login</button></center>

</form>

<button className="switch-button" onClick={switchToRegister}>Switch to Register</button>

</div>

);

}

export default Login;

* **Register Page :**

import React, { useState } from 'react';

import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';

import { faUserPlus, faUser, faLock, faEnvelope, faPhone, faMapMarkerAlt } from '@fortawesome/free-solid-svg-icons';

import { faGoogle, faGithub } from '@fortawesome/free-brands-svg-icons';

import "../../Assest/css/Registration.css";

function Registration({ onRegister, switchToLogin }) {

const [username, setUsername] = useState('');

const [password, setPassword] = useState('');

const [email, setEmail] = useState('');

const [phone, setPhone] = useState('');

const [address, setAddress] = useState('');

const handleSubmit = async (e) => {

e.preventDefault();

const response = await fetch('http://localhost:8080/api/users/register', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ username, password, email, phone, address })

});

if (response.ok) {

onRegister(username, password, email, phone, address);

} else {

alert('Failed to register');

}

};

const handleGoogleSignIn = () => {

window.open('https://accounts.google.com/signin', '\_blank');

};

const handleGithubSignIn = () => {

window.open('https://github.com/login', '\_blank');

};

return (

<div className="registration-form-container">

<h2>Register</h2>

<FontAwesomeIcon icon={faUserPlus} className="user-icon" />

<form onSubmit={handleSubmit}>

<div className="input-container">

<FontAwesomeIcon icon={faUser} className="input-icon" />

<input

type="text"

placeholder="Username"

value={username}

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

required

/>

</div>

<div className="input-container">

<FontAwesomeIcon icon={faLock} className="input-icon" />

<input

type="password"

placeholder="Password"

value={password}

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

required

/>

</div>

<div className="input-container">

<FontAwesomeIcon icon={faEnvelope} className="input-icon" />

<input

type="email"

placeholder="Email"

value={email}

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

required

/>

</div>

<div className="input-container">

<FontAwesomeIcon icon={faPhone} className="input-icon" />

<input

type="tel"

placeholder="Phone"

value={phone}

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

required

/>

</div>

<div className="input-container">

<FontAwesomeIcon icon={faMapMarkerAlt} className="input-icon" />

<input

type="text"

placeholder="Address"

value={address}

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

required

/>

</div>

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

</form>

<button className="switch-button" onClick={switchToLogin}>Switch to Login</button>

<div className="social-signin">

<FontAwesomeIcon icon={faGoogle} className="social-icon" onClick={handleGoogleSignIn} />

<FontAwesomeIcon icon={faGithub} className="social-icon" onClick={handleGithubSignIn} />

</div>

</div>

);

}

export default Registration;

* **Welcome Page :**

import React, { useState, useEffect } from 'react';

import { toast, ToastContainer } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

import '../../Assest/css/Welcome.css';

import { AiOutlineSearch } from 'react-icons/ai';

import { CiLocationOn } from 'react-icons/ci';

import { MdWork, MdWorkOutline } from 'react-icons/md';

import { BsFillCalendarCheckFill } from 'react-icons/bs';

import Sidebar from '../sidebar/Sidebar';

import RemoteJobs from '../Jobs/RemoteJobe';

import WalkinJobs from '../WalkInJobs/WalkinJobs';

import DomainJobs from '../DomainJobs/DomainJobs';

import Recuriter from '../Recuriter/Recuriter';

import CourseCard from '../Course/CourseCard';

import JobCard from '../Course/JobCard';

import Footer from '../Footer/Footer';

const districts = [

'Coimbatore',

'Chennai',

'Salem',

'Nammakal',

'Trichy',

// Add more districts as needed

];

const Welcome = () => {

const [timer, setTimer] = useState(10);

const [intervalId, setIntervalId] = useState(null);

const handleLogout = () => {

const id = setInterval(() => {

setTimer(prevTimer => {

if (prevTimer <= 1) {

clearInterval(id);

setTimer(0);

} else {

return prevTimer - 1;

}

});

}, 1000);

setIntervalId(id);

toast.info(

<div>

<p>Are you sure you want to logout?</p>

<p>Time remaining: {timer} seconds</p>

<div className="toast-button-container">

<button

onClick={() => {

clearInterval(id);

toast.success('Logged out successfully');

setTimeout(() => {

window.location.href = '/login';

}, 1000);

}}

className="toast-button toast-button-yes"

>

Yes

</button>

<button

onClick={() => {

clearInterval(id);

toast.error('Logout canceled');

}}

className="toast-button toast-button-no"

>

No

</button>

</div>

</div>,

{

autoClose: false,

closeOnClick: false,

draggable: false,

onClose: () => {

clearInterval(intervalId);

},

}

);

};

useEffect(() => {

return () => {

clearInterval(intervalId);

};

}, [intervalId]);

const [isScrolled, setIsScrolled] = useState(false);

useEffect(() => {

const handleScroll = () => {

if (window.scrollY > 0) {

setIsScrolled(true);

} else {

setIsScrolled(false);

}

};

window.addEventListener('scroll', handleScroll);

return () => {

window.removeEventListener('scroll', handleScroll);

};

}, []);

return (

<div className="welcome-page">

<header className={`header ${isScrolled ? 'scrolled' : ''}`}>

<div className="logo"><span className='logoo'>Shigo</span><span className='logoo1'>To</span></div>

<nav className="nav">

<a href="#home">Home</a>

<a href="#about">About</a>

<a href="#services">Services</a>

<button className="logout-button" onClick={handleLogout}>Logout</button>

</nav>

</header>

<div className="main-content">

<div className="text-content">

<h1>Welcome to Our Website!</h1>

<p>Discover amazing opportunities and connect with experts.</p>

</div>

<div className="search-bar">

<div className="search-input-group">

<MdWork className="search-icon" />

<select className="search-select">

<option value="">Job Role</option>

<option value="developer">Developer</option>

<option value="designer">Designer</option>

<option value="manager">Manager</option>

{/\* Add more roles as needed \*/}

</select>

</div>

<div className="search-input-group">

<MdWorkOutline className="search-icon" />

<select className="search-select">

<option value="">Job Type</option>

<option value="full-time">Full-Time</option>

<option value="part-time">Part-Time</option>

<option value="internship">Internship</option>

{/\* Add more types as needed \*/}

</select>

</div>

<div className="search-input-group">

<CiLocationOn className="search-icon" />

<select className="search-select">

<option value="">Location</option>

{districts.map((district, index) => (

<option key={index} value={district}>{district}</option>

))}

</select>

</div>

<div className="search-input-group">

<BsFillCalendarCheckFill className="search-icon" />

<select className="search-select">

<option value="">Experience Level</option>

<option value="junior">Junior</option>

<option value="mid">Mid-Level</option>

<option value="senior">Senior</option>

{/\* Add more levels as needed \*/}

</select>

</div>

<button className="search-button">

<AiOutlineSearch className="search-button-icon" />

Search

</button>

</div>

<div>

<TopHiringCompanies />

</div>

<div className="content">

{/\* <Sidebar /> \*/}

<RemoteJobs />

</div>

<div className='context1'>

<WalkinJobs />

</div>

<div className='context2'>

<DomainJobs />

</div>

<div className='context5'>

<Recuriter />

</div>

<div>

<JobCard />

</div>

<div className="footer">

<Footer />

</div>

</div>

<ToastContainer />

</div>

);

};

export default Welcome;

* **BuyNow Page :**

import React, { useState } from 'react';

import { useNavigate } from 'react-router-dom';

import { ToastContainer, toast } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

import { FaUser, FaEnvelope, FaAddressCard, FaBook, FaPhone, FaCreditCard, FaMoneyBillWave, FaCalendarAlt, FaClock, FaFlag, FaTicketAlt } from 'react-icons/fa';

import emailjs from 'emailjs-com';

import '../../Assest/css/BuyNow.css'; // Import the CSS file

*const* BuyNow = () *=>* {

*const* navigate = useNavigate();

*const* [formData, setFormData] = useState({

    name: '',

    email: '',

    address: '',

    course: '',

    phone: '',

    payment: '',

    amount: '',

    date: '',

    duration: '',

    country: '',

    coupon: ''

  });

*const* handleChange = (*e*) *=>* {

*const* { name, value } = *e*.target;

    setFormData({ ...formData, [name]: value });

  };

*const* handleSubmit = async (*e*) *=>* {

*e*.preventDefault();

    // Send email using EmailJS

    emailjs.send('service\_o8t50xl', 'template\_1yljwv8', formData, 'e4myIqx69FAyxz7Kl')

      .then((*response*) *=>* {

        toast.success('Purchase successful! You will receive an email soon.');

        setTimeout(() *=>* {

          navigate('/');

        }, 4000); // Delay navigation for 4 seconds

      })

      .catch((*error*) *=>* {

        toast.error('Something went wrong! Please try again.');

      });

  };

  return (

    <div className="buy-now-container">

      <div className="left-side"></div>

      <div className="right-side">

        <h2 className="fade-in">Buy Course</h2>

        <form onSubmit={handleSubmit} className="buy-form fade-in">

          <div className="form-group">

            <label className="label-group" htmlFor="name"><*FaUser* /> Name</label>

            <input

              type="text"

              id="name"

              name="name"

              value={formData.name}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="email"><*FaEnvelope* /> Email</label>

            <input

              type="email"

              id="email"

              name="email"

              value={formData.email}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="address"><*FaAddressCard* /> Address</label>

            <textarea

              id="address"

              name="address"

              value={formData.address}

              onChange={handleChange}

              required

            ></textarea>

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="course"><*FaBook* /> Course</label>

            <input

              type="text"

              id="course"

              name="course"

              value={formData.course}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="phone"><*FaPhone* /> Phone</label>

            <input

              type="tel"

              id="phone"

              name="phone"

              value={formData.phone}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="payment"><*FaCreditCard* /> Payment Details</label>

            <input

              type="text"

              id="payment"

              name="payment"

              value={formData.payment}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="amount"><*FaMoneyBillWave* /> Amount</label>

            <input

              type="number"

              id="amount"

              name="amount"

              value={formData.amount}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="date"><*FaCalendarAlt* /> Date</label>

            <input

              type="date"

              id="date"

              name="date"

              value={formData.date}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="duration"><*FaClock* /> Duration (weeks)</label>

            <input

              type="number"

              id="duration"

              name="duration"

              value={formData.duration}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="country"><*FaFlag* /> Country</label>

            <input

              type="text"

              id="country"

              name="country"

              value={formData.country}

              onChange={handleChange}

              required

            />

          </div>

          <div className="form-group">

            <label className="label-group" htmlFor="coupon"><*FaTicketAlt* /> Coupon Code</label>

            <input

              type="text"

              id="coupon"

              name="coupon"

              value={formData.coupon}

              onChange={handleChange}

            />

          </div>

          <button type="submit" className="submit-btn">Submit</button>

        </form>

      </div>

      <*ToastContainer* />

    </div>

  );

};

export default BuyNow;

* **ADMIN SIDE**
* **RecruiterLogin Page :**

import \* as React from 'react';

import Avatar from '@mui/material/Avatar';

import Button from '@mui/material/Button';

import CssBaseline from '@mui/material/CssBaseline';

import TextField from '@mui/material/TextField';

import FormControlLabel from '@mui/material/FormControlLabel';

import Checkbox from '@mui/material/Checkbox';

import Link from '@mui/material/Link';

import Grid from '@mui/material/Grid';

import Box from '@mui/material/Box';

import LockOutlinedIcon from '@mui/icons-material/LockOutlined';

import Typography from '@mui/material/Typography';

import Container from '@mui/material/Container';

import { createTheme, ThemeProvider } from '@mui/material/styles';

import { useNavigate } from 'react-router-dom';

import { AuthContext } from '../../Components/Authentication/AuthContext'

*function* Copyright(*props*) {

  return (

    <*Typography* variant="body2" color="text.secondary" align="center" {...props}>

      {'Copyright © '}

      <*Link* color="inherit" href="https://mui.com/">

        Your Website

      </*Link*>{' '}

      {new *Date*().getFullYear()}

      {'.'}

    </*Typography*>

  );

}

*const* defaultTheme = createTheme();

export default *function* RecruiterLogin() {

*const* navigate = useNavigate();

*const* { login } = React.useContext(AuthContext);

*const* handleSubmit = (*event*) *=>* {

*event*.preventDefault();

*const* data = new *FormData*(*event*.currentTarget);

*const* email = data.get('email');

*const* password = data.get('password');

    // Hardcoded credentials

    if (email === 'admin@example.com' && password === 'password') {

      login(email);

      navigate('/post-job');

    } else {

      alert('Invalid credentials');

    }

  };

  return (

    <*ThemeProvider* theme={defaultTheme}>

      <*Container* component="main" maxWidth="xs">

        <*CssBaseline* />

        <*Box*

          sx={{

            marginTop: 8,

            display: 'flex',

            flexDirection: 'column',

            alignItems: 'center',

          }}

        >

          <*Avatar* sx={{ m: 1, bgcolor: 'secondary.main' }}>

            <*LockOutlinedIcon* />

          </*Avatar*>

          <*Typography* component="h1" variant="h5">

            Sign in

          </*Typography*>

          <*Box* component="form" onSubmit={handleSubmit} noValidate sx={{ mt: 1 }}>

            <*TextField*

              margin="normal"

              required

              fullWidth

              id="email"

              label="Email Address"

              name="email"

              autoComplete="email"

              autoFocus

            />

            <*TextField*

              margin="normal"

              required

              fullWidth

              name="password"

              label="Password"

              type="password"

              id="password"

              autoComplete="current-password"

            />

            <*FormControlLabel*

              control={<*Checkbox* value="remember" color="primary" />}

              label="Remember me"

            />

            <*Button*

              type="submit"

              fullWidth

              variant="contained"

              sx={{ mt: 3, mb: 2 }}

            >

              Sign In

            </*Button*>

            <*Grid* container>

              <*Grid* item xs>

                <*Link* href="#" variant="body2">

                  Forgot password?

                </*Link*>

              </*Grid*>

              <*Grid* item>

                <*Link* href="#" variant="body2">

                  {"Don't have an account? Sign Up"}

                </*Link*>

              </*Grid*>

            </*Grid*>

          </*Box*>

        </*Box*>

        <*Copyright* sx={{ mt: 8, mb: 4 }} />

      </*Container*>

    </*ThemeProvider*>

  );

}

* **SideBar Page :**

import React from 'react';

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

import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';

import { faHome, faClipboard, faSignOutAlt } from '@fortawesome/free-solid-svg-icons';

import "../../Assest/css/Sidebar1.css"; // Create and import the CSS file

const Sidebar = ({ username }) => {

return (

<div className="sidebar">

<h2 className="sidebar-title"><span className='admin'>Admin Panel</span></h2>

<div className="sidebar-user">

<p>Welcome, {username}</p>

</div>

<ul className="sidebar-menu">

<li>

<Link to="/">

<FontAwesomeIcon icon={faHome} className="sidebar-icon" /> Home

</Link>

</li>

<li>

<Link to="/post-job">

<FontAwesomeIcon icon={faClipboard} className="sidebar-icon" /> Post a Job

</Link>

</li>

<li>

<Link to="/domain-jobs">

<FontAwesomeIcon icon={faClipboard} className="sidebar-icon" /> Domain Jobs

</Link>

</li>

<li>

<Link to="/login">

<FontAwesomeIcon icon={faSignOutAlt} className="sidebar-icon" /> Login

</Link>

</li>

<li>

<Link to="/">

<FontAwesomeIcon icon={faSignOutAlt} className="sidebar-icon" /> Logout

</Link>

</li>

</ul>

</div>

);

};

export default Sidebar;

**CONCLUSION :**

The ShigoTo job searching portal is designed to modernize the job search process through a user-friendly and feature-rich platform. By integrating advanced job search functionalities, direct communication channels, and comprehensive administrative tools, the portal aims to provide a seamless experience for job seekers and employers alike. The platform’s ability to cater to various needs—whether through detailed job listings, employer management, or skill-enhancing courses—ensures that it stands out in the competitive job search market. This project highlights the effective use of contemporary web technologies to address real-world challenges in job searching and professional development.