

# Online Job Portal

## **Abstract:**

The Job Portal Website is a web-based application designed to simplify and optimize the recruitment process by connecting job seekers with potential employers through a secure and interactive platform. The system enables job seekers to register, create profiles, upload resumes, browse available job listings, and apply for positions. Employers can register their companies, post job openings, review applications, and shortlist candidates efficiently.

Developed using the MEAN Stack (MongoDB, Express, Angular, and Node.js), the platform ensures smooth interaction between the frontend and backend components. The backend RESTful APIs handle functionalities such as user authentication, job posting, and application tracking. MongoDB is used as the primary database to store user profiles, job listings, and application data, while JWT (JSON Web Token) is implemented for secure session management and route protection.

This system streamlines the hiring workflow, reducing manual effort and communication gaps between job seekers and recruiters. The responsive and intuitive UI provides a seamless experience across devices. Additionally, the system is scalable and designed for easy integration with future enhancements such as resume parsing, job recommendations, and interview scheduling. The Job Portal Website offers a smart and efficient solution for modern recruitment challenges, creating a centralized platform for career opportunities and talent acquisition.

## **Keywords:**

MongoDB, User authentication, Application tracking, Secure session management, Route protection, Job Portal Website, Recruitment platform, Job seekers, User registration, Employers, Profile creation, Resume upload.

## **Introduction :**

The Job Portal Website is a web-based application designed to simplify and optimize the job recruitment process by connecting job seekers with employers in a seamless and interactive environment. In today's digital age, traditional recruitment methods often involve inefficiencies, communication gaps, and a lack of centralized platforms. This system addresses those issues by providing a centralized, user-friendly portal that streamlines job discovery, application management, and candidate hiring.

Developed using HTML, CSS, JavaScript, and Node.js, the portal provides a responsive and intuitive user interface along with powerful backend processing. The frontend ensures a visually appealing and accessible layout for both candidates and employers, while the backend manages user authentication, job postings, and application tracking.

The system allows job seekers to register, search for job opportunities, and apply directly through the platform. Employers can post job listings, manage applicants, and communicate with potential candidates. The solution is designed to ensure a secure, reliable, and scalable job matching experience.

Job Search Portal is a web application, which serves jobseekers to find available job vacancies and Employers to identify eligible job seekers with the prospect of selecting the most qualified candidates. The only way to select the best-qualified candidate is to have a pool of eligible applicants, which is possible by drawing the interest of individuals in the market.

Job search portals best serve this purpose. E-recruitment has become the standard means for employers and job seekers to meet their respective objectives. The traditional methods for recruitment includes Job fairs, University career employment services, Employee referrals, advertising in the newspapers, televisions etc. With the advancement in technology and growth of internet usage, the e-recruitment has revolutionised the way organisations hire and candidates search for jobs. With the Online Job search portals, the recruitment process is speeded up at every stage from job postings, to receiving applications from candidates, and the interviewing process. The cost of searching/posting jobs will be much less compared to the traditional way of advertising. Job search portal stands as an active means for Employers to outline the job vacancies, responsibilities and qualifications to attract jobseekers. Using the portal jobseekers can extensively search for jobs in companies, organisations and regions they may otherwise have not learnt. In addition, candidates/Employers can write a review about an organisation, which might help them to change the way things are done.

## **Design:**

### **System Architecture Overview**

#### **Frontend:**

- Built with HTML5, CSS3, and JavaScript, the frontend provides an interactive UI for users. Responsive design ensures compatibility across devices, while JavaScript handles client-side logic and API requests.

#### **Backend:**

- The backend is developed using Node.js and Express.js, providing RESTful API services for user authentication, job management, and application handling.

#### **Database:**

- A NoSQL database like MongoDB or a JSON-based file system can be used to store user profiles, job listings, and application data.

#### **API Layer:**

- RESTful APIs enable communication between the frontend and backend. All user and job-related interactions are managed through well-structured endpoints.

#### **User Interface Design:**

##### **1. Login & Registration Page**

- Simple and clean forms.
- Client-side input validation.
- Secure login using token-based authentication (JWT recommended).

##### **2. Job Seeker Dashboard**

- View available jobs by category, title, or company.
- Apply for jobs and track application history.

##### **3. Employer Dashboard**

- Post and manage job listings.
- View and shortlist applicants.
- Edit or remove job posts.

#### **4. Profile Pages**

- Job Seekers: Add resume, experience, skills.
- Employers: Add company profile and contact information.

#### **5. Search and Filter Module**

- Keyword and location-based job search.
- Filter by job type (Full-time, Part-time, Remote).

### **Modules:**

#### **Authentication Module**

- Manages user registration and login.
- Roles: Job Seeker, Employer.
- API Endpoints:
  - POST /api/register
  - POST /api/login
  - GET /api/logout

#### **User Management Module**

- Handles user profile creation and updates.
- Validates sessions using tokens.
- API Endpoints:
  - GET /api/profile
  - POST /api/update-profile

#### **Job Posting Module (Employer Side)**

- Allows employers to post, update, or delete job listings.
- API Endpoints:
  - POST /api/jobs
  - GET /api/jobs
  - DELETE /api/jobs/:id

#### **Job Application Module (Job Seeker Side)**

- Enables job seekers to apply for jobs.
- Tracks application history.

- API Endpoints:
  - POST /api/apply
  - GET /api/applications

## Search & Filter Module

- Users can search for jobs based on keywords and filters.
- Job listings displayed dynamically using JavaScript.

## Architecture:

### 1. Presentation Layer (Frontend)

- Technologies Used: HTML5, CSS3, JavaScript

#### Features:

- Responsive pages with Bootstrap or custom CSS.
- Real-time form validation and dynamic content loading using AJAX or Fetch API.
- Session-based UI rendering.

### 2. Application Layer (Backend)

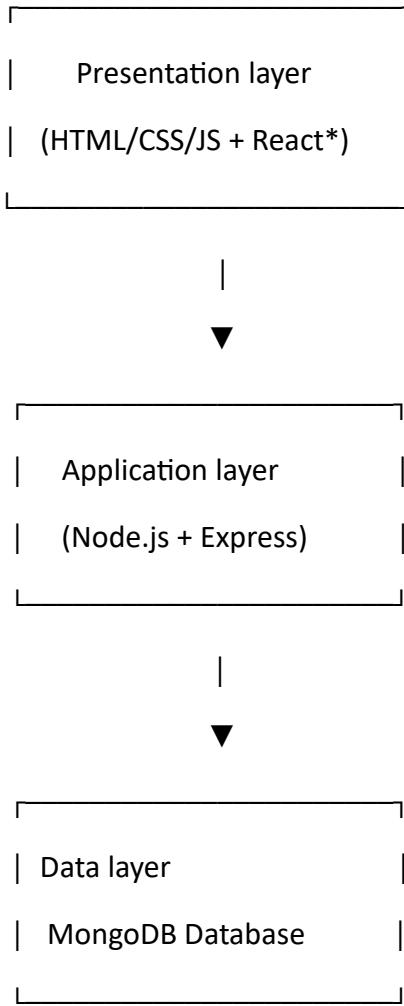
- Technologies Used: Node.js, Express.js

#### Features:

- Route handling for job, user, and application operations.
- Middleware for authentication and authorization.
- Business logic for application workflows and role-based access control.

### 3. Data Layer (Database)

- Technologies Used: MongoDB (or JSON-based storage)
- Collections:
  - Users (with roles)
  - Job Listings
  - Applications



## Technologies:

### 1. Frontend (Client-Side)

- HTML5 – for structuring web pages
- CSS3 / SCSS / Tailwind CSS / Bootstrap – for styling
- JavaScript – for dynamic content and interactivity
- React.js / Angular / Vue.js – for building responsive UI (optional but powerful for SPAs).

#### Key Feature:

- Structures core elements like forms, job listings, and user profiles
- Creates responsive, mobile-friendly, and visually appealing UI
- Adds interactivity like search filters, real-time form validation, and dynamic content

### 2. Backend (Server-Side)

- **Node.js** – JavaScript runtime for building backend services
- **Express.js** – Web framework for Node.js
- **Java (Spring Boot) or Python (Django/Flask)** – alternatives for server-side logic
- **PHP (Laravel)** – another common backend choice

**Key Feature:**

- Handles server-side logic, processes job applications, and manages user sessions
- Routes API requests for authentication, job posting, and user management
- Facilitates communication between frontend and backend with structured data.

### 3. Database

• **SQL Databases:**

- **MySQL / PostgreSQL / Microsoft SQL Server** – relational databases

• **NoSQL Databases:**

- **MongoDB / Firebase** – for more flexible, scalable data structures

**Key Feature:**

- Stores flexible user profiles, job posts, and messages in JSON-like documents.
- Manages relational data like application history, roles, and job categories.

### 4. Hosting & Deployment Optional (Advanced Features)

- **Vercel / Netlify** – for frontend deployment
- **Render / Railway / Heroku / AWS / Azure** – for full stack or backend deployment
- **Docker** – containerization for easier deployment
- **Nginx / Apache** – for web server setup

**Key Feature:**

- Quickly deploys the frontend with global CDN support
- Hosts and scales backend APIs and database services
- Packages the entire app for consistent development and deployment environments.

## **5. Authentication & Security**

- **JWT (JSON Web Tokens)** – for session and API security
- **OAuth 2.0** – for social logins (e.g., Google, LinkedIn)
- **bcrypt** – for password hashing

### **Key Feature:**

- Maintains secure, stateless user authentication after login
- Hashes passwords to securely store user credentials
- Enables social login (Google, LinkedIn) for a faster signup process.

## **6. Optional (Advanced Features)**

- **ElasticSearch** – for powerful search/filter functionality
- **Redis** – for caching and session management
- **Socket.io** – for real-time chat between employers and applicants
- **AI/ML** – for job recommendations or resume parsing.

### **Key Feature:**

- Tracks code versions and supports collaboration among developers
- Tests APIs for functionality before frontend integration
- Streamlines development with debugging, extensions, and project management.

The Job Portal System leverages a MEAN-like stack (MongoDB, Express.js, Node.js) combined with modern frontend technologies to deliver a seamless and user-friendly platform for job seekers and employers. With secure authentication and a robust backend architecture, the system ensures efficient job postings, application tracking, and user management, making it a reliable solution for modern recruitment needs.

### **Coding:**

#### **Frontend:**

#### **Index.html:**

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Job Portal</title>

<link rel="stylesheet" href="index.css">

<style>

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #e4d2d2;

}

header {

    background-color: #0055ff;

    padding: 15px;

    color: rgb(229, 216, 216);

    text-align: center;

}

nav ul {

    list-style: none;

    padding: 0;

    margin: 0;

}
```

```
        display: flex;  
        justify-content: center;  
        gap: 15px;  
    }  
  
nav ul li {  
    display: inline;  
}  
  
nav ul li a {  
    text-decoration: none;  
    color: white;  
    font-weight: bold;  
}  
  
.hero {  
    text-align: center;  
    padding: 40px;  
    background-color: #423f3f;  
}  
  
.search-box {  
    padding: 10px;  
    width: 250px;  
    border-radius: 5px;  
    border: 1px solid #ccc;  
}  
  
button {  
    padding: 10px 15px;
```

```
background-color: #16d368;  
color: white;  
border: none;  
cursor: pointer;  
border-radius: 5px;  
}  
  
.job-listings {  
display: flex;  
flex-wrap: wrap;  
gap: 15px;  
justify-content: center;  
padding: 20px;  
}  
  
.job-card {  
background: white;  
padding: 15px;  
border-radius: 10px;  
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
width: 250px;  
text-align: center;  
}  
  
.job-card h3 {  
margin: 10px 0;  
}  
  
.job-card button {
```

```
margin-top: 10px;  
}  
  
footer {  
    text-align: center;  
    padding: 15px;  
    background-color: #01060c;  
    color: white;  
    margin-top: 20px;  
}  
  
</style>  
  
</head>  
  
<body>  
  
<header>  
  
<div class="logo">  
    <h1>Job Portal</h1>  
</div>  
  
<nav>  
    <ul>  
        <li><a href="#">Home</a></li>  
        <li><a href="job.html">Jobs</a></li>  
        <li><a href="about page.html">About us</a></li>  
        <li><a href="contact.html">Contact</a></li>  
        <li><a href="Apply.html">Apply a Job</a></li>  
        <li><a href="login.html">Login</a></li>  
        <li><a href="register.html">Register</a></li>  
    </ul>  
</nav>
```

```
<li><a href="profile.html">Profile</a></li>

</ul>

</nav>

</header>

<script src="index.js"></script>

<section class="hero">

  <h2>Find Your Dream Job</h2>

  <p>Search for jobs and opportunities that match your skills</p>

  <input type="text" placeholder="Search for jobs..." class="search-box">

  <button>Search</button>

</section>

<section class="job-listings">

  <div class="job-card">

    <h3>Software Engineer</h3>

    <p><strong>Company:</strong> Tech Corp</p>

    <p><strong>Location:</strong> New York, NY</p>

    <button>Apply</button>

  </div>

  <div class="job-card">

    <h3>Full Stack Developer</h3>

    <p><strong>Company:</strong> StartupX</p>

    <p><strong>Location:</strong> Mumbai, India</p>

    <button>Apply</button>

  </div>

</div>
```

```
<div class="job-card">  
  <h3>Frontend Developer</h3>  
  <p><strong>Company:</strong> CodeCrafters Inc.</p>  
  <p><strong>Location:</strong> Bangalore, India</p>  
  <button>Apply</button>  
</div>  
  
<div class="job-card">  
  <h3>Data Analyst</h3>  
  <p><strong>Company:</strong> Data Solutions</p>  
  <p><strong>Location:</strong> Los Angeles, CA</p>  
  <button>Apply</button>  
</div>  
  
<div class="job-card">  
  <h3>Backend Developer</h3>  
  <p><strong>Company:</strong> DevSolutions</p>  
  <p><strong>Location:</strong> Pune, India</p>  
  <button>Apply</button>  
</div>  
  
<div class="job-card">  
  <h3>Cybersecurity Analyst</h3>  
  <p><strong>Company:</strong> IBM</p>  
  <p><strong>Location:</strong> Austin, TX</p>  
  <button>Apply</button>  
</div>  
  
<div class="job-card">
```

```
<h3>Machine Learning Engineer</h3>
<p><strong>Company:</strong> OpenAI</p>
<p><strong>Location:</strong> Remote</p>
<button>Apply</button>
</div>

<div class="job-card">
<h3>Product Manager</h3>
<p><strong>Company:</strong> Tesla</p>
<p><strong>Location:</strong> Palo Alto, CA</p>
<button>Apply</button>
</div>

<div class="job-card">
<h3>UI/UX Designer</h3>
<p><strong>Company:</strong> Creative Studios</p>
<p><strong>Location:</strong> Chicago, IL</p>
<button>Apply</button>
</div>

<div class="job-card">
<h3>Digital Marketing Executive</h3>
<p><strong>Company:</strong> MarketWise</p>
<p><strong>Location:</strong> Jaipur, India</p>
<button>Apply</button>
</div>

<div class="job-card">
<h3>Cloud Engineer</h3>
```

```
<p><strong>Company:</strong> CloudBridge</p>
<p><strong>Location:</strong> Gurgaon, India</p>
<button>Apply</button>
</div>

<div class="job-card">
    <h3>Game Developer</h3>
    <p><strong>Company:</strong> Ubisoft</p>
    <p><strong>Location:</strong> Montreal, Canada</p>
    <button>Apply</button>
</div>

<div class="job-card">
    <h3>IT Support Specialist</h3>
    <p><strong>Company:</strong> Dell Technologies</p>
    <p><strong>Location:</strong> Houston, TX</p>
    <button>Apply</button>
</div>

<div class="job-card">
    <h3>SEO Specialist</h3>
    <p><strong>Company:</strong> Shopify</p>
    <p><strong>Location:</strong> Toronto, Canada</p>
    <button>Apply</button>
</div>

</section>
<footer>
    <p>&copy; 2025 Job Portal. All rights reserved.</p>
```

```
</footer>

</body>

</html>

/* General Styles */

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-size: cover;

    background-position: center;

    color: #8b8484;

}

/* Header */

header {

    background: #007bff;

    color: rgb(23, 21, 21);

    padding: 15px 20px;

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.logo h1 {

    margin: 0;

}

nav ul {
```

```
list-style: none;  
margin: 0;  
padding: 0;  
display: flex;  
}  
  
nav ul li {  
margin: 0 15px;  
}  
  
nav ul li a {  
color: rgb(221, 214, 214);  
text-decoration: none;  
font-size: 16px;  
}  
  
nav ul li a:hover {  
text-decoration: underline;  
}  
  
/* Hero Section */  
  
.hero {  
text-align: center;  
padding: 50px 20px;  
background: rgba(0, 0, 0, 0.6);  
color: rgb(233, 212, 212);  
}  
  
.hero h2 {  
font-size: 28px;
```

```
}

.search-box {
    padding: 10px;
    width: 60%;
    margin: 10px 0;
    border-radius: 5px;
    border: 1px solid #ccc;
}

button {
    padding: 10px 20px;
    border: none;
    background: #28a745;
    color: rgb(221, 211, 211);
    cursor: pointer;
    border-radius: 5px;
}

button:hover {
    background: #218838;
}

/* Job Listings */

.job-listings {
    padding: 20px;
    text-align: center;
}

.job-card {
```

```
background: rgb(209, 195, 195);
padding: 15px;
margin: 10px auto;
width: 80%;
border-radius: 5px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.job-card h3 {
margin-bottom: 5px;
}

/* Footer */

footer {
background: #4a4b4c;
color: rgb(236, 222, 222);
text-align: center;
padding: 10px;
margin-top: 20px;
}

document.addEventListener("DOMContentLoaded", () => {
// Select all Apply buttons inside job cards
const applyButtons = document.querySelectorAll(".job-card button");
applyButtons.forEach((button) => {
button.addEventListener("click", () => {
// Redirect to the Apply page
window.location.href = "Apply.html"; // Adjust path if needed

```

```

    });

});

// Optional: Search functionality

const searchBox = document.querySelector(".search-box");

const searchButton = document.querySelector(".hero button");

searchButton.addEventListener("click", () => {

    const query = searchBox.value.trim().toLowerCase();

    const jobs = document.querySelectorAll(".job-card");

    let found = false;

    jobs.forEach((job) => {

        const title = job.querySelector("h3").textContent.toLowerCase();

        const location = job.querySelector("p:nth-child(3)").textContent.toLowerCase();

        if (title.includes(query) || location.includes(query)) {

            job.style.display = "block";

            found = true;

        } else {

            job.style.display = "none";

        }

    });

    if (!found) {

        alert("No jobs found for: " + query);

    }

});

});

```

O/P:

The screenshot shows a web browser window with a job portal interface. At the top, there's a blue header bar with the title "Job Portal" and navigation links for Home, Jobs, About us, Contact, Apply a Job, Login, Register, and Profile. Below the header is a dark grey banner with the text "Find Your Dream Job" and a search bar placeholder "Search for jobs and opportunities that match your skills". The main content area displays ten job listings arranged in two rows of five. Each listing includes the job title, company name, location, and an "Apply" button.

Job Title	Company	Location	Action
Software Engineer	Tech Corp	New York, NY	<button>Apply</button>
Full Stack Developer	StartupX	Mumbai, India	<button>Apply</button>
Frontend Developer	CodeCrafters Inc.	Bangalore, India	<button>Apply</button>
Data Analyst	Data Solutions	Los Angeles, CA	<button>Apply</button>
Backend Developer	DevSolutions	Pune, India	<button>Apply</button>
Cybersecurity Analyst	IBM	Austin, TX	<button>Apply</button>
Machine Learning Engineer	OpenAI	Remote	<button>Apply</button>
Product Manager	Tesla	Palo Alto, CA	<button>Apply</button>
UI/UX Designer	Creative Studios	Chicago, IL	<button>Apply</button>
Digital Marketing Executive	MarketWise	Jaipur, India	<button>Apply</button>

## Login.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;
```

```
justify-content: center;  
align-items: center;  
height: 100vh;  
background-color: #f4f4f4;  
margin: 0;  
}  
  
.container {  
background: white;  
padding: 20px;  
border-radius: 10px;  
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
text-align: center;  
width: 300px;  
}  
  
h1 {  
margin-bottom: 20px;  
}  
  
label {  
display: block;  
text-align: left;  
font-weight: bold;  
margin-top: 10px;  
}  
  
input {  
width: 100%;
```

```
padding: 8px;  
margin-top: 5px;  
border: 1px solid #ccc;  
border-radius: 5px;  
}  
  
button {  
width: 100%;  
padding: 10px;  
background: #007BFF;  
border: none;  
color: white;  
font-size: 16px;  
border-radius: 5px;  
cursor: pointer;  
margin-top: 15px;  
}  
  
button:hover {  
background: #0056b3;  
}  
  
a {  
display: block;  
margin-top: 15px;  
color: #007BFF;  
text-decoration: none;  
}
```

```
a:hover {  
    text-decoration: underline;  
}  
  
p {  
    margin-top: 15px;  
    font-size: 14px;  
    color: #666;  
}  
  
</style>  
  
<script>  
  
function redirectToNextPage(event) {  
  
    event.preventDefault(); // Prevent actual form submission  
  
    window.location.href = "index.html"; // Redirect to the next page  
  
}  
  
</script>  
  
</head>  
  
<body>  
  
<div class="container">  
  
    <h1>Login</h1>  
  
    <form action="/login" method="POST" onsubmit="redirectToNextPage(event)">  
  
        <label for="email">Email:</label>  
  
        <input type="email" id="email" name="email" required>  
  
        <br>  
  
        <label for="password">Password:</label>  
  
        <input type="password" id="password" name="password" required>  
  
</div>
```

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

</form>

<a href="/register.html">Register</a>

<p>Admin? Use <strong>admin@example.com</strong> with your password.</p>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Signup Page</title>

<style>

* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: Arial, sans-serif;

}

body {

display: flex;

justify-content: center;
```

```
    align-items: center;  
    height: 100vh;  
    background: blueviolet;  
}  
  
.signup-container {  
    background: white;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    width: 300px;  
    text-align: center;  
}  
  
h2 {  
    margin-bottom: 20px;  
}  
  
.input-group {  
    margin-bottom: 15px;  
    text-align: left;  
}  
  
label {  
    display: block;
```

```
margin-bottom: 5px;  
font-weight: bold;  
}
```

```
input {  
width: 100%;  
padding: 8px;  
border: 1px solid #ccc;  
border-radius: 5px;  
}
```

```
button {  
width: 100%;  
padding: 10px;  
background: blueviolet;  
color: white;  
border: none;  
border-radius: 5px;  
cursor: pointer;  
font-size: 16px;  
}
```

```
button:hover {  
background: #218838;  
}
```

```
.error {  
    color: red;  
    font-size: 12px;  
    display: block;  
    height: 15px;  
}  
  
.login-link {  
    margin-top: 10px;  
    font-size: 14px;  
}  
  
</style>  
</head>  
<body>  
  
<div class="signup-container">  
    <h2>Sign Up</h2>  
    <form id="signupForm">  
        <div class="input-group">  
            <label for="username">Full Name</label>  
            <input type="text" id="username" name="username" placeholder="Enter your name" required>  
            <small class="error" id="nameError"></small>  
        </div>  
    </form>  
</div>
```

```
<div class="input-group">  
    <label for="email">Email</label>  
    <input type="email" id="email" name="email" placeholder="Enter your email" required>  
    <small class="error" id="emailError"></small>  
</div>  
  
<div class="input-group">  
    <label for="password">Password</label>  
    <input type="password" id="password" name="password" placeholder="Enter password" required>  
    <small class="error" id="passwordError"></small>  
</div>  
  
<div class="input-group">  
    <label for="confirmPassword">Confirm Password</label>  
    <input type="password" id="confirmPassword" placeholder="Confirm password" required>  
    <small class="error" id="confirmPasswordError"></small>  
</div>  
  
<button type="submit">Sign Up</button>  
  
<p class="login-link">Already have an account? <a href="login.html">Login</a></p>  
</form>  
</div>
```

```
<script>

    document.getElementById("signupForm").addEventListener("submit", async
function(event) {

    event.preventDefault(); // Prevent form submission


    let username = document.getElementById("username").value.trim();

    let email = document.getElementById("email").value.trim();

    let password = document.getElementById("password").value.trim();

    let confirmPassword = document.getElementById("confirmPassword").value.trim();


    let nameError = document.getElementById("nameError");

    let emailError = document.getElementById("emailError");

    let passwordError = document.getElementById("passwordError");

    let confirmPasswordError = document.getElementById("confirmPasswordError");


    // Reset errors

    nameError.textContent = "";

    emailError.textContent = "";

    passwordError.textContent = "";

    confirmPasswordError.textContent = "";


    let valid = true;

    // Name validation

    if (username.length < 3) {

        nameError.textContent = "Name must be at least 3 characters.";
```

```
    valid = false;

}

// Email validation

if (!email.includes("@") || !email.includes(".")) {

    emailError.textContent = "Enter a valid email.";

    valid = false;

}

// Password validation

if (password.length < 6) {

    passwordError.textContent = "Password must be at least 6 characters.";

    valid = false;

}

// Confirm Password validation

if (password !== confirmPassword) {

    confirmPasswordError.textContent = "Passwords do not match.";

    valid = false;

}

if (valid) {

    try {

        let response = await fetch("/register", {

            method: "POST",

```

```

headers: {
    "Content-Type": "application/json"
},
body: JSON.stringify({ username, email, password })
});

let result = await response.json();

if (response.ok) {
    alert(" ✅ Signup Successful!");
    window.location.href = "login.html"; // Redirect to login page
} else {
    alert(" ❌ Error: " + result.error);
}
} catch (error) {
    console.error(" ❌ Error:", error);
    alert(" ❌ Something went wrong!");
}
}

});

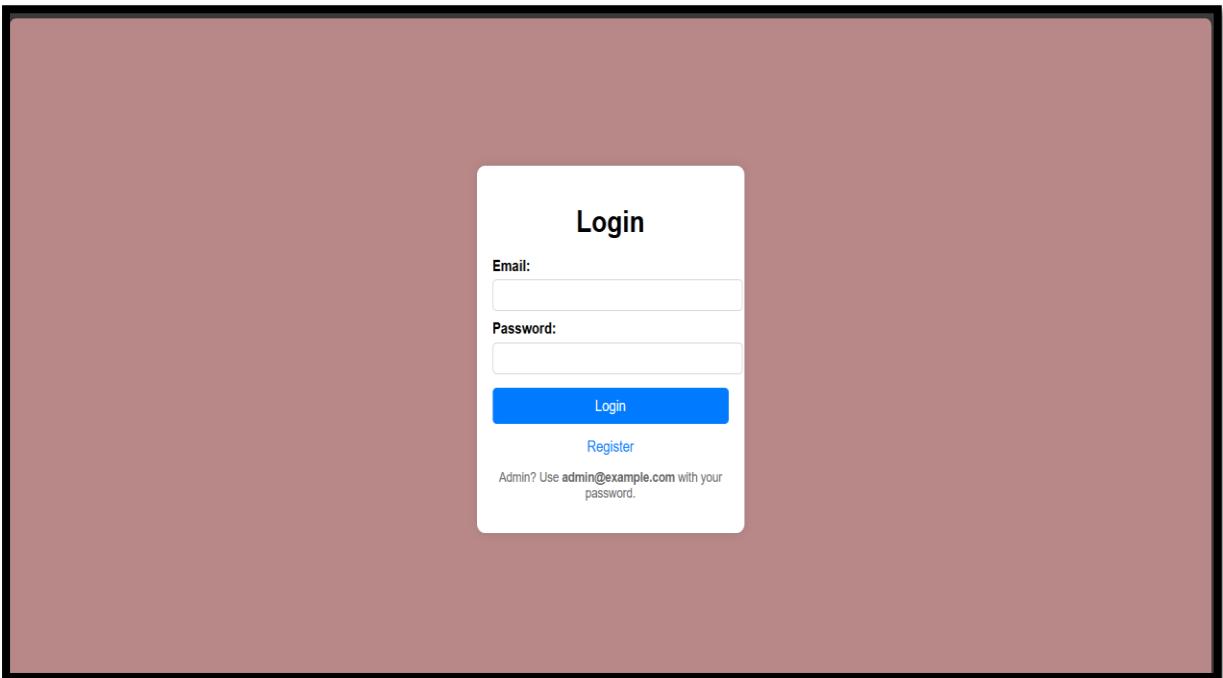
</script>

</body>

</html>

```

O/P:



### Sign up.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>About Us - Job Portal</title>

<link rel="stylesheet" href="about page.css">

</head>

<body>

<header>

<div class="logo">

<h1>Job Portal</h1>

</div>
```

```

<nav>
    <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="job.html">Jobs</a></li>
        <li><a href="about page.html">About Us</a></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
</nav>

</header>

<section class="about">
    <h2>About Job Portal</h2>    <p>Job Portal is a place where job seekers and employers connect. We aim to provide the best career opportunities for job seekers and help businesses find the right talent for their teams.</p>
</section>

<script src="about page.js"></script>

<footer>
    <p>&copy; 2025 Job Portal. All rights reserved.</p>
</footer>

</body>
</html>

/* General Styles */

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

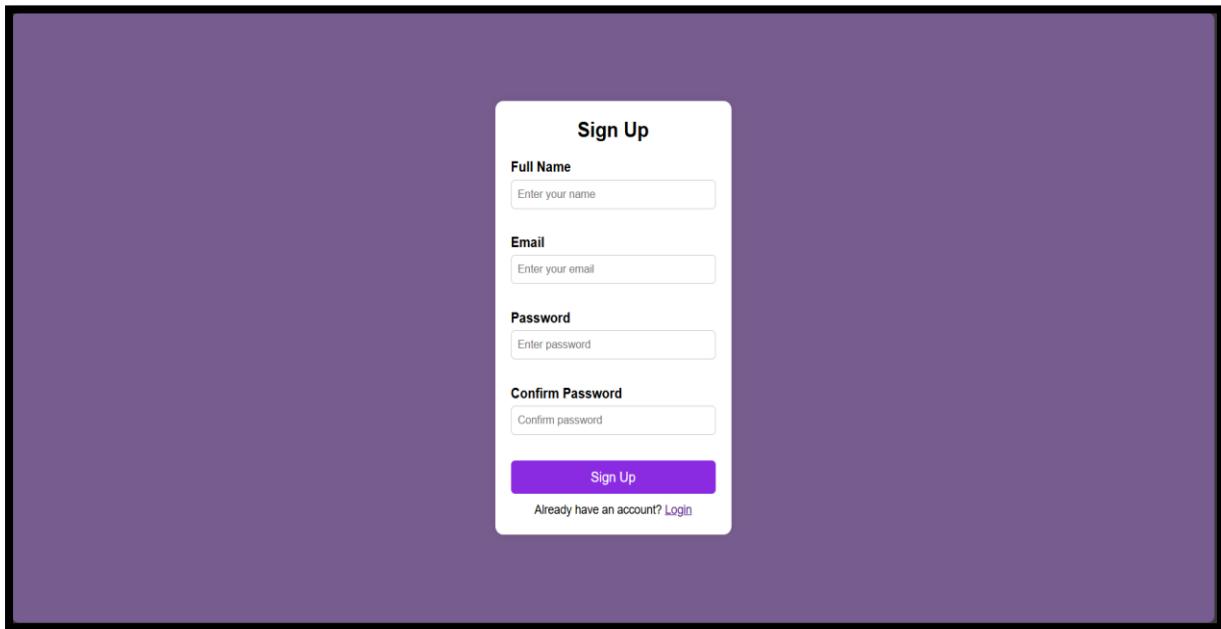
```

```
color: #333;  
}  
  
/* Header */  
  
header {  
background: #007bff;  
color: white;  
padding: 15px 20px;  
display: flex;  
justify-content: space-between;  
align-items: center;  
}  
  
.logo h1 {  
margin: 0;  
}  
  
nav ul {  
list-style: none;  
margin: 0;  
padding: 0;  
display: flex;  
}  
  
nav ul li {  
margin: 0 15px;  
}  
  
nav ul li a {  
color: white;
```

```
text-decoration: none;  
font-size: 16px;  
}  
  
nav ul li a:hover {  
text-decoration: underline;  
}  
  
/* About Section */  
  
.about {  
text-align: center;  
padding: 50px 20px;  
}  
  
.about h2 {  
font-size: 28px;  
margin-bottom: 15px;  
}  
  
.about p {  
font-size: 18px;  
max-width: 600px;  
margin: 0 auto;  
}  
  
/* Footer */  
  
footer {  
background: #007bff;  
color: white;  
text-align: center;
```

```
padding: 10px;  
position: absolute;  
bottom: 0;  
width: 100%;  
}  
  
document.addEventListener("DOMContentLoaded", function () {  
    console.log("About Us page loaded successfully");  
    // Smooth scrolling for navigation links  
    document.querySelectorAll("nav ul li a").forEach(link => {  
        link.addEventListener("click", function (event) {  
            if (this.getAttribute("href").startsWith("#")) {  
                event.preventDefault();  
                const targetId = this.getAttribute("href").substring(1);  
                const targetElement = document.getElementById(targetId);  
                if (targetElement) {  
                    window.scrollTo({  
                        top: targetElement.offsetTop - 50,  
                        behavior: "smooth"  
                    });  
                }  
            }  
        });  
    });  
});
```

O/P:



## Job Application.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Apply for Job</title>

    <link rel="stylesheet" href="Apply.css">

</head>

<body>

    <header>

        <h1>Apply for Job</h1>

        <nav>
```

```

<ul>

    <li><a href="index.html">Home</a></li>

    <li><a href="job.html">Jobs</a></li>

    <li><a href="about page.html">About us</a></li>

    <li><a href="contact.html">contact </a></li>

    <li><a href="login.html">Login</a></li>

    <li><a href="register.html">Register</a></li>

</ul>

</nav>

</header>

<section class="apply-container">

    <form id="applyForm" class="job-form">

        <h2>Job Application</h2>

        <label for="fullName">Full Name:</label>

        <input type="text" id="fullName" name="fullName" required>

        <label for="email">Email Address:</label>

        <input type="email" id="email" name="email" required>

        <label for="phone">Phone Number:</label>

        <input type="tel" id="phone" name="phone" required>

        <label for="coverLetter">Cover Letter:</label>

        <textarea id="coverLetter" name="coverLetter" rows="5" required></textarea>

        <label for="resume">Upload Resume (PDF):</label>

        <input type="file" id="resume" name="resume" accept=".pdf" required>

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

    </form>

```

```
<script src="Apply.js"></script>

<div id="message"></div>

</section>

<script src="apply.js"></script>

</body>

</html>

/* Apply for Job Page Styles */

body {

    font-family: 'Segoe UI', sans-serif;

    background-color: #f5f6fa;

    margin: 0;

    padding: 0;

}

header {

    background-color: #007bff;

    color: white;

    padding: 15px 30px;

    display: flex;

    justify-content: space-between;

    align-items: center;

}

header h1 {

    margin: 0;

}

nav a {
```

```
color: white;  
text-decoration: none;  
margin-left: 20px;  
font-weight: bold;  
}  
  
nav a:hover {  
text-decoration: underline;  
}  
  
.apply-container {  
max-width: 600px;  
margin: 40px auto;  
padding: 30px;  
background-color: white;  
border-radius: 10px;  
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}  
  
.job-form {  
display: flex;  
flex-direction: column;  
}  
  
.job-form h2 {  
color: #007bff;  
margin-bottom: 15px;  
}  
  
.job-form label {
```

```
margin-top: 10px;  
font-weight: bold;  
}  
  
.job-form input,  
.job-form textarea {  
width: 100%;  
padding: 10px;  
margin-top: 5px;  
border: 1px solid #ccc;  
border-radius: 5px;  
}  
  
.job-form input[type="file"] {  
padding: 5px;  
}  
  
.job-form button {  
margin-top: 20px;  
padding: 12px;  
background-color: #28a745;  
color: white;  
border: none;  
border-radius: 5px;  
font-size: 16px;  
cursor: pointer;  
}  
  
.job-form button:hover {
```

```

background-color: #218838;

}

/* Message Box */

#message {
    margin-top: 20px;
    font-size: 16px;
    text-align: center;
}

document.addEventListener("DOMContentLoaded", function () {

    const applyForm = document.getElementById("applyForm");
    const messageDiv = document.getElementById("message");

    applyForm.addEventListener("submit", function (event) {
        event.preventDefault(); // Prevent actual form submission

        // Collect form data (optional, for validation or backend use)

        const fullName = document.getElementById("fullName").value;
        const email = document.getElementById("email").value;
        const phone = document.getElementById("phone").value;
        const coverLetter = document.getElementById("coverLetter").value;
        const resume = document.getElementById("resume").files[0];

        // Show success message on the current page

        messageDiv.innerHTML = `<p style="color: green; font-weight: bold;">Thank you, ${fullName}! Your application has been submitted successfully.</p>`;

        // Optional: redirect to a success page after 2 seconds

        setTimeout(function () {
            window.location.href="file:///C:/Users/Anith/OneDrive/Desktop/job%20portal/succes.html"; // Create this page!
        }, 2000);
    });
});

```

```

    }, 2000);

});

document.getElementById("applyForm").addEventListener("submit", function(event) {
    event.preventDefault(); // Prevents the form from actually submitting to a backend

    // Perform any necessary validation or form processing here

    // Redirect to the success page
    window.location.href = "successapply.html";
});

```

O/P:

The image shows a job application form titled "Job Application". The form consists of several input fields and a file upload section. The fields are labeled: "Full Name:", "Email Address:", "Phone Number:", and "Cover Letter:". The "Cover Letter" field is highlighted with a red border and contains the placeholder text "Please fill out this field.". Below these fields is a section for uploading a resume in PDF format, with a "Choose File" button and a message "No file chosen". At the bottom of the form is a large green "Submit Application" button.

## Contact.html

```

<!DOCTYPE html>

<html lang="en">
<head>

```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Contact - Job Portal</title>
<link rel="stylesheet" href="contact.css">
</head>
<body>
<header>
<div class="logo">
<h1>Job Portal</h1>
</div>
<nav>
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="job.html">Jobs</a></li>
<li><a href="about page.html">About Us</a></li>
<li><a href="#">Contact</a></li>
</ul>
</nav>
</header>

<section class="contact">
<h2>Contact Us</h2>
<div id="notification" class="notification" style="display: none;"></div>
<form>
<label for="name">Your Name:</label>
```

```
<input type="text" id="name" name="name" required>

<label for="email">Your Email:</label>

<input type="email" id="email" name="email" required>

<label for="message">Message:</label>

<textarea id="message" name="message" rows="5" required></textarea>

<button type="submit" class="btn">Send Message</button>

</form>

</section>

<script src="contact.js"></script>

<footer>

<p>&copy; 2025 Job Portal. All rights reserved.</p>

</footer>

</body>

</html>

/* Reset default margins and paddings */

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

/* General body settings */
```

```
body {  
    font-family: 'Arial', sans-serif;  
    line-height: 1.6;  
    background-color: #f4f4f4;  
    color: #333;  
}
```

```
/* Header */
```

```
header {  
    background: #333;  
    color: white;  
    padding: 15px 0;  
}
```

```
header .logo {
```

```
    text-align: center;  
    font-size: 24px;  
    font-weight: bold;  
}
```

```
header nav ul {
```

```
    display: flex;  
    justify-content: center;  
    list-style: none;  
}
```

```
header nav ul li {
```

```
    margin: 0 15px;
```

```
}

header nav ul li a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}

header nav ul li a:hover {
    text-decoration: underline;
}

/* Hero Section */

.hero {
    background: url('https://via.placeholder.com/1920x800') no-repeat center center/cover;
    color: white;
    text-align: center;
    padding: 100px 20px;
}

.hero-content h1 {
    font-size: 3rem; }

.hero-content p {
    font-size: 1.2rem;
    margin: 20px 0;
}

.hero .btn {
    background-color: #007BFF;
    padding: 12px 25px;
```

```
color: white;  
text-decoration: none;  
font-weight: bold;  
border-radius: 5px;  
transition: background-color 0.3s ease;  
}  
  
.hero .btn:hover {  
background-color: #0056b3;  
}  
  
/* Job Listings */  
  
.job-listings {  
padding: 40px 20px;  
}  
  
.job {  
background: white;  
margin-bottom: 20px;  
padding: 20px;  
border-radius: 5px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
.job h3 {  
font-size: 1.5rem;  
margin-bottom: 10px;  
}  
  
.job p {  
margin-bottom: 10px;
```

```
}
```

```
.job .btn {
```

```
background-color: #28a745;
```

```
padding: 8px 20px;
```

```
color: white;
```

```
text-decoration: none;
```

```
border-radius: 5px;
```

```
font-weight: bold;
```

```
}
```

```
.job .btn:hover {
```

```
background-color: #218838;
```

```
}
```

```
/* Footer */
```

```
footer {
```

```
text-align: center;
```

```
padding: 10px;
```

```
background-color: #333;
```

```
color: white;
```

```
}
```

```
/* Forms */
```

```
form label {
```

```
display: block;
```

```
margin: 10px 0 5px;
```

```
}
```

```
form input,
```

```
form textarea {  
    width: 100%;  
    padding: 12px;  
    margin-bottom: 10px;  
    border: 1px solid #ddd;  
    border-radius: 5px;  
}  
  
form button {  
    background-color: #007BFF;  
    padding: 12px 25px;  
    color: white;  
    border: none;  
    border-radius: 5px;  
    font-weight: bold;  
}  
  
form button:hover {  
    background-color: #0056b3;  
}  
  
document.addEventListener("DOMContentLoaded", () => {  
    const form = document.querySelector("form");  
    form.addEventListener("submit", (e) => {  
        e.preventDefault(); // Stop default form submission  
        const name = document.getElementById("name").value.trim();  
        const email = document.getElementById("email").value.trim();  
        const message = document.getElementById("message").value.trim();  
    })  
})
```

```

// Simple field validation

if (!name || !email || !message) {
    alert("Please fill in all the fields.");
    return;
}

// Email format check

const emailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;

if (!email.match(emailPattern)) {
    alert("Please enter a valid email address.");
    return;
}

// Simulate form submission

alert("Thank you, " + name + "! Your message has been sent.");

// Optionally clear form

form.reset();

});

});

document.addEventListener("DOMContentLoaded", function () {

const form = document.getElementById("contactForm");

const notification = document.getElementById("notification");

form.addEventListener("submit", function (e) {
    e.preventDefault(); // Prevent actual form submission

    // Simulate a successful form submission

    notification.textContent = "Message sent successfully!";
}

```

```

notification.style.display = "block";
notification.style.backgroundColor = "#4CAF50";
notification.style.color = "#fff";
notification.style.padding = "10px";
notification.style.marginBottom = "15px";
notification.style.borderRadius = "5px";
notification.style.textAlign = "center"
form.reset();

// Hide notification after 3 seconds
setTimeout(() => {
  notification.style.display = "none";
}, 3000);
});
});

```

O/P:

The screenshot shows a web page titled "Job Portal" with a dark header bar containing navigation links: Home, Jobs, About Us, and Contact. Below the header is a section titled "Contact Us". The form fields include "Your Name:" (input field), "Your Email:" (input field), and "Message:" (text area). A blue button labeled "Send Message" is located at the bottom left of the form area. At the very bottom of the page is a dark footer bar with the copyright notice: "© 2025 Job Portal. All rights reserved."

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Jobs - Job Portal</title>

<link rel="stylesheet" href="job.css">

</head>

<body>

<header>

<div class="logo">

<h1>Job Portal</h1>

</div>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="#">Jobs</a></li>

<li><a href="about page.html">About Us</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

</header>

<section class="job-listings">

<h2>Available Job Listings</h2>
```

```
<div class="job">  
  <h3>Software Developer</h3>  
  <p>Company: XYZ Ltd</p>  
  <p>Location: New York</p>  
  <a href="#" class="btn">Apply Now</a>  
</div>  
  
<div class="job">  
  <h3>Full Stack Developer</h3>  
  <p>Company: StartupX </p>  
  <p>Location: Mumbai, India</p>  
  <a href="#" class="btn">Apply Now</a>  
</div>  
  
<div class="job">  
  <h3>Data Scientist</h3>  
  <p>Company: ABC Corp</p>  
  <p>Location: San Francisco</p>  
  <a href="#" class="btn">Apply Now</a>  
</div>  
  
<div class="job">  
  <h3>UI/UX Designer</h3>  
  <p>Company: Innovate Tech</p>  
  <p>Location: Chicago</p>  
  <a href="#" class="btn">Apply Now</a>  
</div>  
  
<div class="job">
```

```
<h3>Frontend Developer </h3>
<p>Company:CodeCrafters Inc. </p>
<p>Location:Bangalore, India </p>
<a href="#" class="btn">Apply Now</a>
</div>

<div class="job">
<h3>Digital Marketing Executive</h3>
<p>Company:MarketWise</p>
<p>Location:Jaipur, India</p>
<a href="#" class="btn">Apply Now</a>
</div>

<div class="job">
<h3> Cloud Engineer</h3>
<p>Company:CloudBridge</p>
<p>Location:Gurgaon, India</p>
<a href="#" class="btn">Apply Now</a>
</div>

<div class="job">
<h3>Backend Developer </h3>
<p>Company:DevSolutions </p>
<p>Location:Pune, India</p>
<a href="#" class="btn">Apply Now</a>
</div>

</section>
<footer>
```

```
<p>&copy; 2025 Job Portal. All rights reserved.</p>

</footer>

<script src="job.js"></script>

</body>

</html>

/* General Styles */

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-color: #f4f4f4;

}

/* Header */

header {

    background-color: #0073e6;

    color: white;

    padding: 15px;

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.logo h1 {

    margin: 0;

}

nav ul {
```

```
list-style: none;  
padding: 0;  
}  
  
nav ul li {  
display: inline;  
margin-right: 20px;  
}  
  
nav ul li a {  
color: white;  
text-decoration: none;  
font-weight: bold;  
}  
  
nav ul li a:hover {  
text-decoration: underline;  
}  
  
/* Job Listings */  
  
.job-listings {  
width: 80%;  
margin: 20px auto;  
background: white;  
padding: 20px;  
border-radius: 8px;  
box-shadow: 0px 10px rgba(0, 0, 0, 0.1);  
}  
  
.job {
```

```
background: #fff;  
padding: 15px;  
margin: 15px 0;  
border-left: 5px solid #0073e6;  
border-radius: 5px;  
box-shadow: 2px 10px rgba(0, 0, 0, 0.1);  
}  
  
.job h3 {  
color: #333;  
}  
  
.btn {  
display: inline-block;  
padding: 8px 12px;  
background: #0073e6;  
color: white;  
text-decoration: none;  
border-radius: 5px;  
font-weight: bold;  
}  
  
.btn:hover {  
background: #005bb5;  
}  
  
/* Footer */  
  
footer {  
text-align: center;
```

```

padding: 10px;
background: #0073e6;
color: white;
margin-top: 20px;
}

document.addEventListener("DOMContentLoaded", () => {

    // Select all Apply buttons inside job cards
    const applyButtons = document.querySelectorAll(".job-card button");
    applyButtons.forEach((button) => {
        button.addEventListener("click", () => {
            // Redirect to the Apply page
            window.location.href = "apply.html"; // Adjust path if needed
        });
    });

    // Optional: Search functionality
    const searchBox = document.querySelector(".search-box");
    const searchButton = document.querySelector(".hero button");
    searchButton.addEventListener("click", () => {
        const query = searchBox.value.trim().toLowerCase();
        const jobs = document.querySelectorAll(".job-card");
        let found = false;
        jobs.forEach((job) => {
            const title = job.querySelector("h3").textContent.toLowerCase();
            const location = job.querySelector("p:nth-child(3)").textContent.toLowerCase();
            if (title.includes(query) || location.includes(query)) {

```

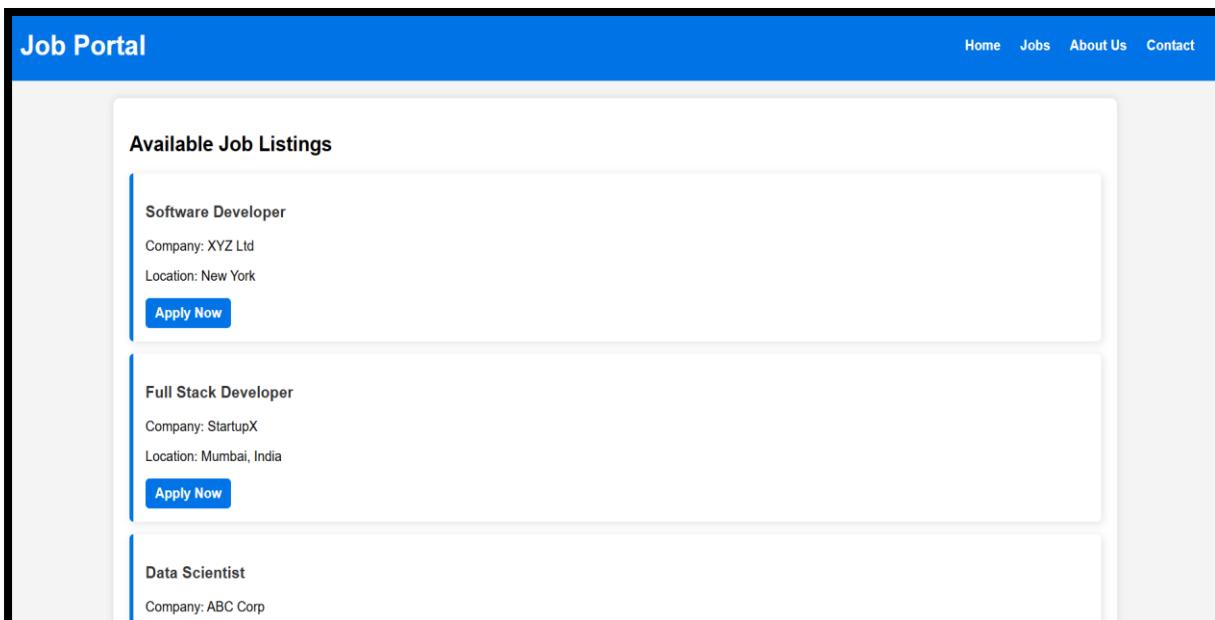
```

        job.style.display = "block";
        found = true;
    } else {
        job.style.display = "none";
    }
});

if (!found) {
    alert("No jobs found for: " + query);
}
});

O/P:

```



<!DOCTYPE html>

<html lang="en">

```
<link rel="stylesheet" href="loginsucces.css">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Success</title>
</head>

<body>
    <h2>Login Successful!</h2>
    <p>Welcome to your dashboard.</p>
    <a href="index.html">Go to Home</a>
    <script src="loginsucces.js"></script>
</body>

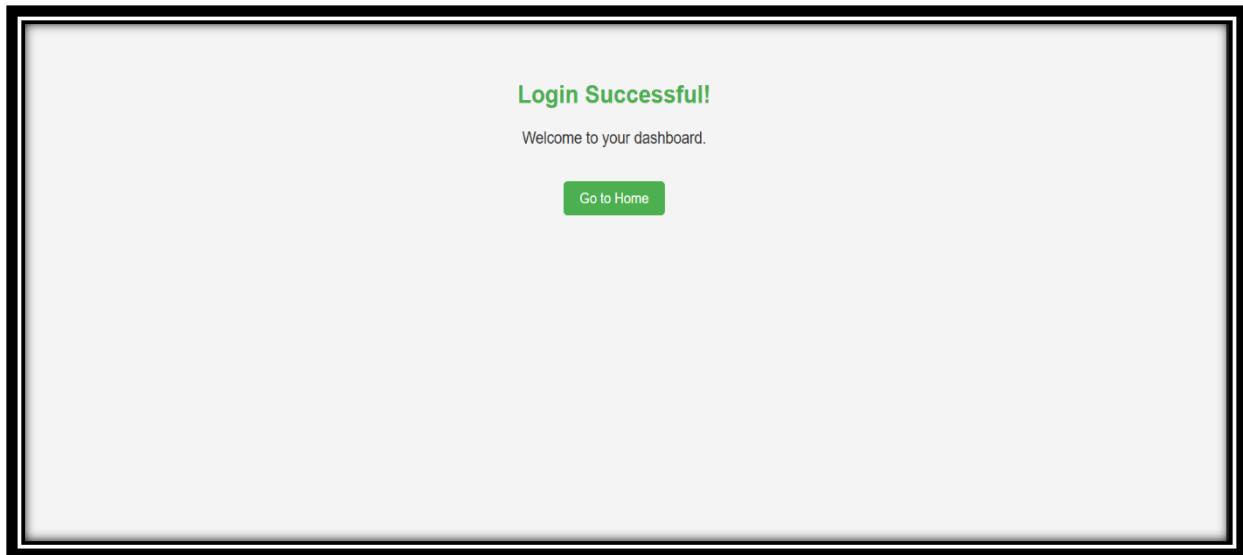
<script src="loginsucces.js"></script>
</html>

/* styles.css */

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    text-align: center;
    padding: 50px;
}

h2 {
    color: #4CAF50;
    font-size: 28px;
}
```

```
p {  
    font-size: 18px;  
    color: #333;  
}  
  
a {  
    display: inline-block;  
    margin-top: 20px;  
    padding: 10px 20px;  
    background-color: #4CAF50;  
    color: white;  
    text-decoration: none;  
    border-radius: 5px;  
    font-size: 16px;  
}  
  
a:hover {  
    background-color: #45a049;  
}  
  
O/P:
```



```
<!DOCTYPE html>

<html lang="en">
    <link rel="stylesheet" href="rs.css">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Registration Successful</title>
        <link rel="stylesheet" href="registersucces.css">
    </head>
    <body>
        <h2>Registration Successful!</h2>
        <p>Your account has been created successfully.</p>
        <a href="stylesheet">Go to Login</a>
    </body>
```

```
<script src="registersucces.js"></script>  
</html>
```

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
    text-align: center;  
    padding: 50px;  
}
```

```
h2 {  
    color: #4CAF50;  
    font-size: 28px;  
}
```

```
p {  
    font-size: 18px;  
    color: #333;  
}
```

```
a {  
    display: inline-block;  
    margin-top: 20px;  
    padding: 10px 20px;  
    background-color: #4CAF50;  
    color: white;  
    text-decoration: none;  
    border-radius: 5px;  
    font-size: 16px;
```

```
}

a:hover {

background-color: #45a049;

}

function register() {

let username = document.getElementById("username").value;

let email = document.getElementById("email").value;

let password = document.getElementById("password").value;

if (username && email && password) {

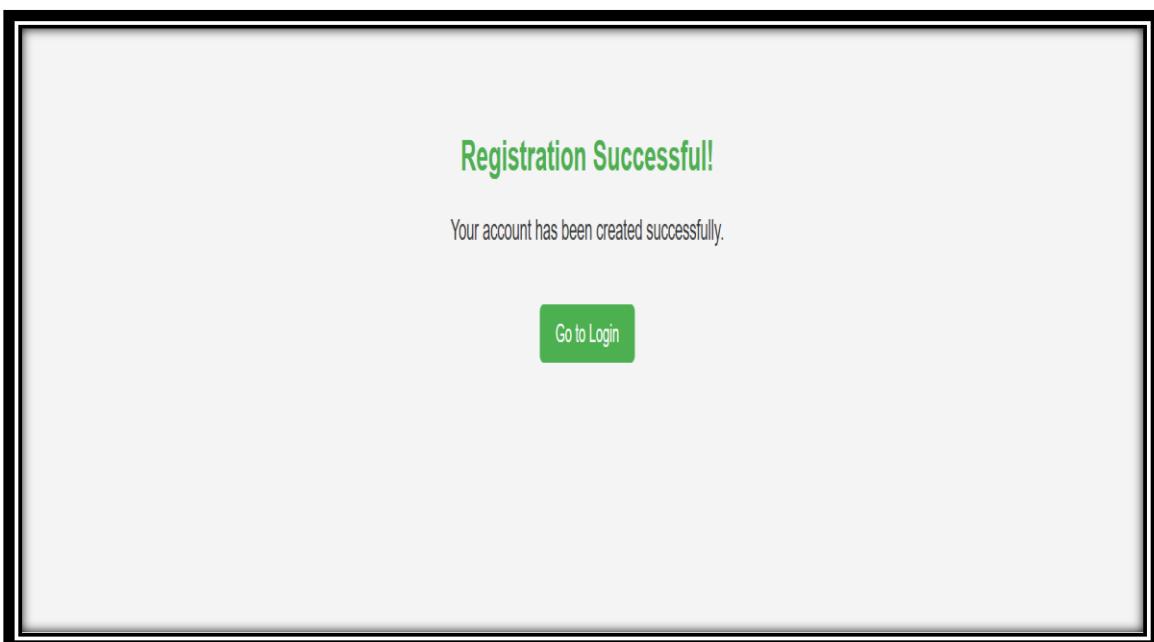
alert("Registration Successful!");

window.location.href = "register-success.html"; // Redirect to success page

} else {

alert("Please fill all fields.");
}
```

O/P:



```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Application Submitted</title>

<link rel="stylesheet" href="Apply.css">

<style>

.success-container {

    text-align: center;

    margin-top: 50px;

} .success-container h1 {

    color: green;

}.success-container p {

    font-size: 18px;

    margin: 10px 0;

}.success-container a {

    display: inline-block;

    padding: 10px 20px;

    margin-top: 20px;

    background-color: #007bff;

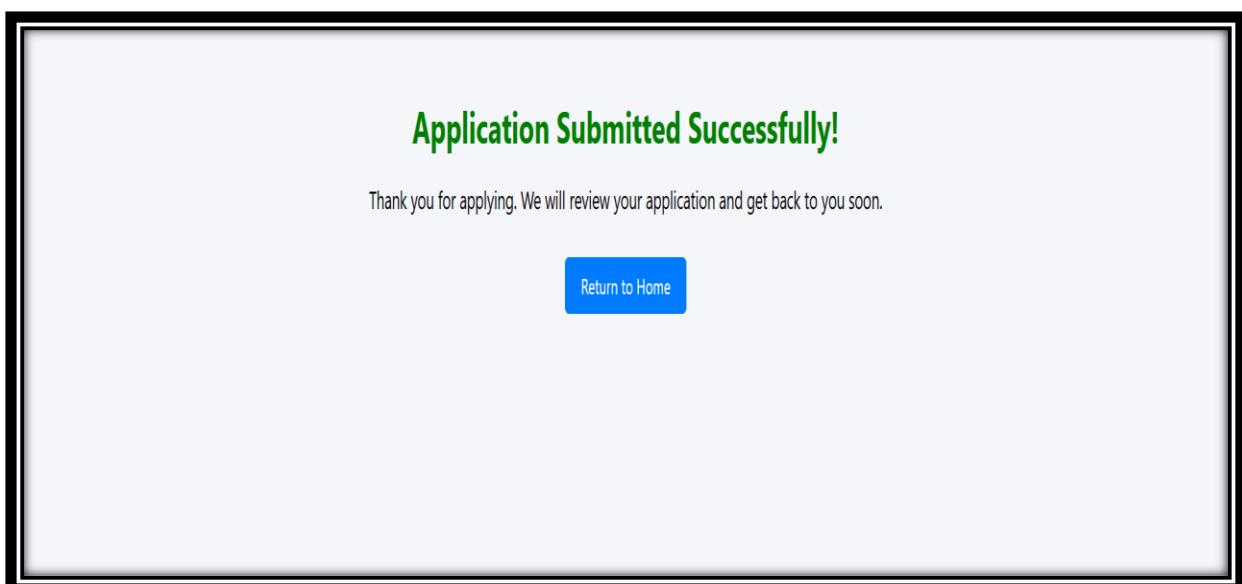
    color: white;

    text-decoration: none;

    border-radius: 5px;
```

```
        } .success-container a:hover {  
            background-color: #0056b3;  
        }  
  
</style>  
  
</head>  
  
<body>  
  
<div class="success-container">  
  
    <h1>Application Submitted Successfully!</h1>  
  
    <p>Thank you for applying. We will review your application and get back to you soon.</p>  
  
    <a href="index.html">Return to Home</a>  
  
</div>  
  
</body>  
  
</html>
```

O/P:



```
// Load environment variables
```

```
require('dotenv').config();

const express = require('express');

const mongoose = require('mongoose');

const bcrypt = require('bcrypt');

const path = require('path');

const cors = require('cors');

const bodyParser = require('body-parser');

const User = require('./models/User');

const app = express();

const PORT = process.env.PORT || 3000;

// Middleware

app.use(cors());

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: false

}));app.use(express.static(path.join(__dirname, 'public')));

// MongoDB Connection

const uri = process.env.MONGO_URI;

if (!uri) {

  console.error('MongoDB URI not defined. Check your .env file.');

  process.exit(1);

}

mongoose

.connect(uri)

.then(() => console.log('Connected to MongoDB'))

.catch(err => {

  console.error('Error connecting to MongoDB:', err);

  process.exit(1);

});
```

```

// Simulated session (for demonstration)

let loggedInUserId = null;

// Routes

app.get('/', (req, res) => {

    res.send('<h1>Welcome</h1><a href="/register.html">Register</a> | <a href="/login.html">Login</a>');

});

// Register

app.post('/register', async (req, res) => {

    const { username, email, password } = req.body;

    if (!username || !email || !password) {

        return res.status(400).send('All fields are required.');

    }

    const hashedPassword = await bcrypt.hash(password, 10);

    const newUser = new User({ username, email, password: hashedPassword });

    await newUser.save();

    res.send('Registration successful. Please <a href="/login.html">login</a>.');

} catch (err) {

    console.error('Registration error:', err);

    res.status(500).send('Internal Server Error');

}

});

// Login

app.post('/login', async (req, res) => {

    const { email, password } = req.body;

    if (!email || !password) {

        return res.status(400).send('All fields are required.');

    }

}

```

```

try {
    const user = await User.findOne({ email });

    if (!user || !(await bcrypt.compare(password, user.password))) {
        return res.status(401).send('Invalid email or password.');
    }

    loggedInUserId = user._id;
    res.json({ message: 'Login successful', userId: loggedInUserId });
}

} catch (err) {
    console.error('Login error:', err);
    res.status(500).send('Internal Server Error');
}

});

// Start server

app.listen(PORT, () => {
    console.log(`✓ Server running at http://localhost:${PORT}`);
});

```

O/P:

The screenshot shows the MongoDB Compass interface with the 'users' collection selected. There are three documents listed:

- User 1:** \_id: ObjectId('67ecd6ad412545492dece6b'), username: "pravalika", email: "22nn1a428@gmail.com", password: "\$2b\$10\$QJ2wSsAtQEgFmDxyn.jeLebATwcXK01MgxQDr4DpehOFdaU.MDlQq", role: "user", \_\_v: 0
- User 2:** \_id: ObjectId('67ecd6ad412545492dece6f'), username: "chennaesettyanitha579@gmail.com", email: "chennaesettyanitha579@gmail.com", password: "\$2b\$10\$NavJXggd67ARTbJoo6WIK0x4N73P3Hy0vnmuEs5nVZTDPBKBaHfh0a", role: "user", \_\_v: 0
- User 3:** \_id: ObjectId('67ecd91d6983be1606feff53'), username: "chennaesetty\_mahesh", email: "chennaesetty\_mahesh24@gmail.com", password: "\$2b\$10\$evXeXtpgyJjnVZumeG542u/3SllVoo/GYJaU8fK0TwfvqTN2FPXX2", role: "user", \_\_v: 0