**📘 Job Portal Software - Full Documentation**

**1. Overview**

The **Job Portal Software** is a web application designed to connect **job seekers** with **employers**. It allows job seekers to search and apply for jobs, and employers to post vacancies and hire candidates.

**2. Stakeholders and User Roles**

|  |  |
| --- | --- |
| Role | Description |
| Job Seeker | A user looking for job opportunities and applies for jobs. |
| Employer | A company or individual posting jobs and hiring candidates. |
| Admin | Manages users, jobs, and overall platform health. |

**3. Features Overview**

**For Job Seekers:**

* Create and manage profile (resume, skills, experience)
* Browse/search/filter job listings
* Apply for jobs
* Track application status
* Receive notifications about jobs and application updates

**For Employers:**

* Create company profile
* Post new job listings
* View and filter applicants
* Contact shortlisted candidates
* Manage job postings (edit, deactivate)

**For Admin:**

* User management (approve/reject registrations)
* Job approval and moderation
* Manage site settings and analytics

**4. User Journey**

**Job Seeker Flow:**

1. Sign up/login.
2. Complete profile with resume and skills.
3. Search jobs by category, location, salary, etc.
4. Apply for selected jobs.
5. Track application status and communicate with employers.

**Employer Flow:**

1. Sign up/login.
2. Create company profile.
3. Post job listings with descriptions, requirements, salary, etc.
4. Review applicants.
5. Shortlist and contact candidates.
6. Close job listings when filled.

**5. How It Works (System Workflow)**

1. **Registration**  
   Users register as Job Seekers or Employers.
2. **Profile Setup**  
   Job Seekers upload resumes; Employers set up company pages.
3. **Job Posting**  
   Employers submit jobs for review or auto-approval.
4. **Job Searching**  
   Job Seekers browse and filter jobs.
5. **Application**  
   Job Seekers apply with CV and cover letter.
6. **Application Management**  
   Employers review and communicate with candidates.
7. **Notifications**  
   Both parties receive updates on applications and job statuses.
8. **API Endpoints**

**Authentication**

* POST /api/auth/register — Register user (job seeker or employer)
* POST /api/auth/login — Login
* POST /api/auth/logout — Logout

**Job Seeker APIs**

* GET /api/jobs — List jobs (with search & filters)
* GET /api/jobs/:id — Job details
* POST /api/applications — Apply for a job
* GET /api/applications — Get user's applications
* PUT /api/applications/:id — Update application (withdraw, etc.)

**Employer APIs**

* POST /api/jobs — Create job posting
* GET /api/jobs — List employer's jobs
* GET /api/jobs/:id/applicants — List applicants for a job
* PUT /api/jobs/:id — Edit job
* DELETE /api/jobs/:id — Delete/close job

**Admin APIs**

* GET /api/users — List all users
* PUT /api/users/:id/approve — Approve user accounts
* DELETE /api/users/:id — Delete user
* GET /api/jobs/pending — List jobs pending approval
* PUT /api/jobs/:id/approve — Approve job posting

**7. Database Schema (Simplified)**

**Users Table**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | UUID | Primary key |
| email | String | User email |
| password\_hash | String | Hashed password |
| role | Enum | 'job\_seeker' or 'employer' or 'admin' |
| status | Enum | 'pending', 'active', 'suspended' |
| created\_at | Timestamp | Account creation time |

**Profiles Table (Job Seekers)**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| user\_id | UUID (FK) | References Users table |
| full\_name | String | Full name |
| resume\_url | String | Link to resume file |
| skills | Text Array | List of skills |
| experience | Text | Work experience details |

**Companies Table (Employers)**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| user\_id | UUID (FK) | References Users table |
| company\_name | String | Company name |
| website\_url | String | Company website |
| description | Text | Company description |

**Jobs Table**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | UUID | Primary key |
| employer\_id | UUID (FK) | References Companies |
| title | String | Job title |
| description | Text | Job description |
| location | String | Job location |
| salary\_range | String | Salary range |
| status | Enum | 'open', 'closed', 'pending\_approval' |
| created\_at | Timestamp | Date posted |

**Applications Table**

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| id | UUID | Primary key |
| job\_id | UUID (FK) | References Jobs |
| job\_seeker\_id | UUID (FK) | References Users |
| cover\_letter | Text | Candidate cover letter |
| status | Enum | 'applied', 'shortlisted', 'rejected', 'hired' |
| applied\_at | Timestamp | Application submission time |

**8. Authentication & Authorization**

* Use JWT or OAuth for secure authentication.
* Role-based access control:
  + Job seekers can only apply and manage their applications.
  + Employers can manage their own jobs and applicants.
  + Admins have full access for moderation.

**9. Frontend Behavior**

* **Job Seekers**:
  + Responsive job search with filters (location, salary, keywords).
  + Profile editor with resume upload.
  + Application status dashboard.
* **Employers**:
  + Job posting form with validations.
  + Applicant list with status filters.
  + Notifications for new applications.
* **Admin**:
  + Dashboard for pending approvals.
  + User and job management panels.

**10. Notifications**

* Email/SMS alerts for:
  + New job postings.
  + Application status updates.
  + Employer contact messages.

**11. Error Handling & Validation**

* Client and server-side input validation.
* User-friendly error messages.
* Handling expired tokens, unauthorized access, and API failures gracefully.

**12. Deployment & Hosting**

* Backend API and frontend can be hosted on cloud services like AWS, Azure, or DigitalOcean.
* Use Docker for containerization.
* Implement CI/CD pipelines for automated testing and deployment.

**13. Compliance & Security**

* Secure user data with encryption and HTTPS.
* GDPR compliance for user data in EU.
* Regular security audits and updates.

**14. Summary**

|  |  |
| --- | --- |
| Who Applies for Jobs? | Job Seekers (registered users looking for work) |
| Who Hires? | Employers (registered companies posting jobs) |
| How It Works? | Job seekers apply via UI → Employers review → Hiring decision → Notifications sent |

**Table of Contents for Job Portal Software documentation:**

**📑 Table of Contents — Job Portal Software Documentation**

1. **Overview**
2. **Stakeholders and User Roles**
3. **Features Overview**
4. **User Journey**
   * 4.1 Job Seeker Flow
   * 4.2 Employer Flow
5. **How It Works (System Workflow)**
6. **API Endpoints**
   * 6.1 Authentication APIs
   * 6.2 Job Seeker APIs
   * 6.3 Employer APIs
   * 6.4 Admin APIs
7. **Database Schema (Simplified)**
   * 7.1 Users Table
   * 7.2 Profiles Table (Job Seekers)
   * 7.3 Companies Table (Employers)
   * 7.4 Jobs Table
   * 7.5 Applications Table
8. **Authentication & Authorization**
9. **Frontend Behavior**
10. **Notifications**
11. **Error Handling & Validation**
12. **Deployment & Hosting**
13. **Compliance & Security**
14. **Summary**

Would you like me to help write or expand any specific section next?