WorkSphere

Submitted by,

Surati Abhishek Jayeshkumar (202404104610008),

Rawal Vagendra Prakashbhai (202404104610092),

Bhatporiya Visarg Manishbhai (202404104610111),

Bhatt Vaibhav Vijaykumar (202404104610113)

Guided by,

Tanvi K Patel

for partial fulfillment of the requirements

for the Degree of Master of Computer Application,

Shrimad Rajchandra Institute of Management and Computer Application,

Uka Tarsadia University,

August, 2025.

DECLARATION

We hereby declare that the project titled “WorkSphere” is fully implemented by us. It is neither paid nor copied. Even though, later on, in case of any infringement found for this project work, we are solely responsible for the same and understand that as per UGC norms, the University can revoke the degree conferred to us.

Enrollment No.	Student Name	Signature
202404104610008	Surati Abhishek Jayeshkumar	
202404104610092	Rawal Vagendra Prakashbhai	
202404104610111	Bhatporiya Visarg Manishbhai	
202404104610113	Bhatt Vaibhav Vijaykumar	

As a guide, I assure you that there is no plagiarism found in the submitted document.

Guide Name: Tanvi K Patel

Signature:

Date:

Place:

Index

1. Introduction of System:

1.1 Introduction of System:

The **WorkSphere** is designed to connect job seekers with potential employers, streamlining the recruitment process. The system provides a platform where companies can post job openings, and applicants can apply for jobs, manage their profiles, and track application progress.

1.2 Definition:

The system is defined as a web-based application designed to facilitate job recruitment. It supports two main user types: job seekers and employers. Job seekers can register, create and update their profiles, upload resumes, search for job openings, and apply for positions. Employers can register, post job advertisements, and manage applications received from potential candidates. An administrator has the responsibility of overseeing the overall system, ensuring that the platform runs smoothly.

1.3 Scope:

- **Advanced Matching Algorithms and AI:**
    - Automated candidate ranking, intelligent job matching, and machine learning–based recommendations are planned for future versions.
- **Integration with External Systems:**
    - No integration with third-party HR, payroll, or applicant tracking systems is included.

- **Mobile Application Development:**
  - The project is limited to a web-based interface; native mobile apps (iOS/Android) will be developed in subsequent phases.
- **User Dashboard Customization:**
  - Customizable dashboards or personalized widgets for both job seekers and employers are not provided in the current version.
- **Social Media Integration:**
  - Features such as social login, sharing job postings on social platforms, or importing data from LinkedIn profiles are excluded.
- **Payment Gateway and Monetization:**
  - Payment processing for premium features, subscription models, or advertisement management is not part of the initial release.
- **Extended Security Features:**
  - Beyond the standard authentication and authorization, advanced security measures such as biometric logins are planned for later releases.
- **Internationalization and Localization:**
  - Multilingual support and regional adaptations (e.g., local currency, time zones) are not implemented in this version.

1.4 Tools & Technologies:

Frontend	React, Tailwind CSS
Backend	Node.js, Express.js
Database	MongoDB (using Mongoose for ORM)
Authentication	JSON Web Tokens (JWT) for secure access

2. System Analysis

2.1 Identification of Need

In today's job market, both companies and job seekers face several challenges:

- **Scattered Opportunities**
  Job listings are spread across multiple sites and paper advertisements, making it hard for candidates to find all relevant openings in one place.
- **Lengthy Hiring Cycles**
  Employers often rely on manual sorting of resumes, leading to delays of several weeks before initial contact.
- **Poor Tracking & Communication**
  Applicants lose track of where they applied, and companies struggle to keep candidates informed of their application status.
- **Skill–Job Mismatch**
  Without a good matching system, candidates see irrelevant listings, and employers receive unsuitable applications.

WorkSphere Job Portal addresses these needs by providing:

- A **centralized** platform where all job posts are easy to browse.
- **Automated matching** of candidate skills to job requirements.
- **Real-time status updates** so applicants and recruiters always know what's happening.
- **Reports and dashboards** that help hiring managers measure time-to-hire and application quality.

2.2 Functionalities

2.2.1 Functional Requirements

ID	Requirement
FR1	**User Registration & Login –** Candidates and employers can sign up with email/password or social accounts and log in securely.
FR2	**Profile Management –** Users can create and edit their profiles (resume, skills, experience for candidates; company details for employers).
FR3	**Job Posting & Management –** Employers can post new jobs, set application deadlines, and view/manage active listings.
FR4	**Job Search & Filters –** Candidates can search by role, location, salary range, and keywords.
FR5	**Application Submission –** Candidates apply to jobs with a single click; resumes and cover letters are submitted automatically.
FR6	**Kanban-style Application Tracking –** Visual board for candidates and recruiters showing stages: Applied, Under Review, Shortlisted, Interview Scheduled, Offered, Rejected.
FR7	**Notification System –** Email, push, and in-app notifications for application updates, interview invites, and recruiter messages.
FR8	**Admin Tools –** Manage users/companies, view **activity logs**, monitor chats, generate **PDF/Excel reports**, and handle escalations.
FR9	**Real-Time Chat –** Secure chat between candidates and recruiters with message history.
FR10	**Analytics Dashboard –** Employers and admins view charts showing job performance, application trends, and hiring funnel.

2.2.1Non-Functional Requirements

ID	Requirement
NFR1	**Performance** – All pages and dashboards must load in under 2 seconds; real-time chat must have <200ms message delivery latency.
NFR2	**Reliability** – Maintain 99.9% uptime with auto-recovery for backend services.
NFR3	**Security** – Industry-standard encryption (bcrypt, TLS 1.3), secure APIs, and protection against XSS, CSRF, and SQL/NoSQL injection.
NFR4	**Scalability** – Handle up to 10,000 concurrent users with the ability to scale AI and chat services independently.
NFR5	**Maintainability** – Modular architecture with microservice-ready backend; AI, chat, and analytics modules deployable independently.
NFR6	**Data Privacy** – Compliance with **GDPR, CCPA**, and secure handling of AI-generated content and chat logs.
NFR7	**Usability** – Mobile-responsive design; allow new users to complete job search and application within 3 minutes.

2.3 Software Development Process

Data Model Description

Our job portal project, WorkSphere, utilizes a **document-oriented NoSQL data model** implemented with **MongoDB**. To manage data integrity and enforce structure, we use **Mongoose** as an Object Document Mapper (ODM). The design of our data model was carried out in a sequential, disciplined manner following a process similar to the **Waterfall Model**, ensuring that every phase of the design was thoroughly analyzed, documented, and verified.

Waterfall Model: Overview

The Waterfall Model is a linear and sequential approach to software development. It divides the process into distinct phases, where each phase must be completed before the next one begins. The main phases are:

1. **Requirements Analysis**
2. **System Design (Conceptual, Logical, and Physical Design)**
3. **Implementation (Coding)**
4. **Testing & Verification**
5. **Deployment**
6. **Maintenance**

Each phase produces specific deliverables and documentation, and the output of one phase serves as the input for the next.

1. Requirements Analysis

- **Objective:** Identify both functional and non-functional requirements.
- **Activities:**
  - Conducted interviews and surveys with potential users (job seekers and employers).
  - Documented requirements like user registration, job posting, application tracking, etc.
- **Outcome:** A comprehensive requirements specification that defined what the system should achieve.

2. System Design

- **Conceptual Design:**
  - Created an initial Entity-Relationship (ER) diagram to represent key entities such as Users, Companies, Jobs, and Applications.
- **Logical Design:**
  - Mapped entities to MongoDB collections, defined fields and relationships.
  - Selected MongoDB for its flexibility and scalability and used Mongoose for schema enforcement.
- **Physical Design:**
  - Finalized the design documents and created detailed schema definitions.
- **Outcome:** A clear blueprint for how the system will be structured, both in terms of software architecture (frontend, backend, database) and data modeling.

3. Implementation

- **Objective:** Translate the design into a working system.
- **Activities:**
  - Developed the frontend using React and Tailwind CSS.
  - Built the backend using Node.js and Express.js.
  - Implemented the data model in MongoDB using Mongoose.

- **Outcome:** A fully functioning job portal that supports features like registration, job search, posting, and application management.

4. Testing & Verification

- **Objective:** Ensure the system meets all requirements and works as intended.
- **Activities:**
  - Performed unit tests, integration tests, and system testing.
  - Verified each module (e.g., user authentication, job posting) individually before integrating them.
- **Outcome:** A robust system with confirmed functionality, ready for deployment.

5. Deployment

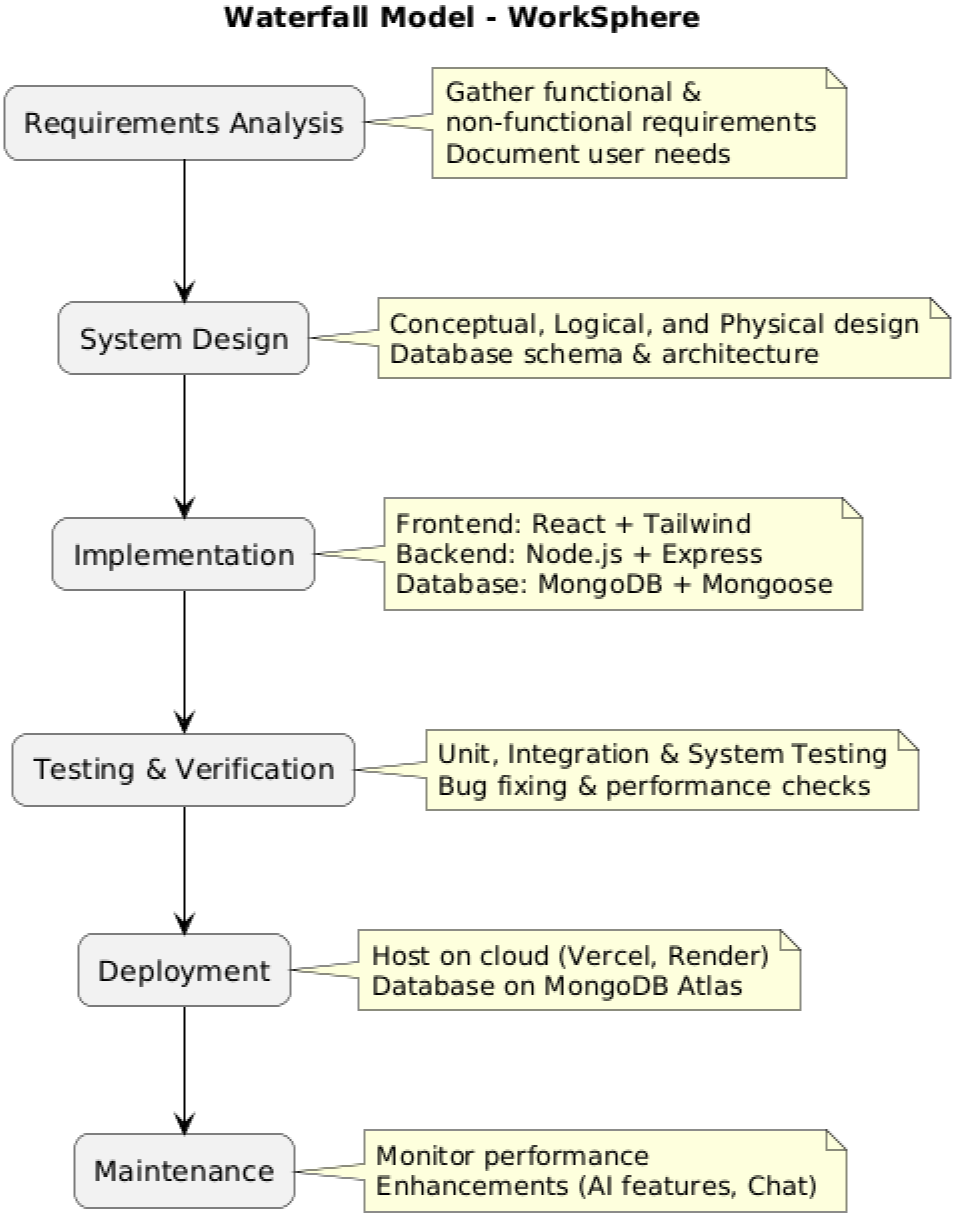
- **Objective:** Make the system available for end users.
- **Activities:**
  - Deployed the application on a cloud-based server or prepared it for local testing.
  - Set up continuous integration with Git and GitHub for version control.
- **Outcome:** The system is live and accessible to job seekers, employers, and administrators.

6. Maintenance

- **Objective:** Keep the system updated and functional over time.
- **Activities:**
  - Monitor system performance and address issues as they arise.
  - Plan for future enhancements such as additional features (interview scheduling, notifications).
- **Outcome:** A system that remains effective and scalable as user demands evolve.

Why the Waterfall Model Was Chosen

- **Structured Process:** The clearly defined sequential phases ensured that each step was thoroughly planned and documented before moving on.
- **Fixed Scope:** The project requirements were well-defined early on, making a linear approach ideal.
- **Comprehensive Documentation:** The Waterfall model emphasizes detailed documentation, which aligns well with the LDP guidelines.
- **Ease of Management:** With a fixed timeline and deliverables for each phase, the project team and guides could monitor progress effectively.


Waterfall Model - WorkSphere
Requirements Analysis
Gather functional & non-functional requirements Document user needs
System Design
Conceptual, Logical, and Physical design Database schema & architecture
Implementation
Frontend: React + Tailwind Backend: Node.js + Express Database: MongoDB + Mongoose
Testing & Verification
Unit, Integration & System Testing Bug fixing & performance checks
Deployment
Host on cloud (Vercel, Render) Database on MongoDB Atlas
Maintenance
Monitor performance Enhancements (AI features, Chat)

2.3.1 Waterfall Model Diagram

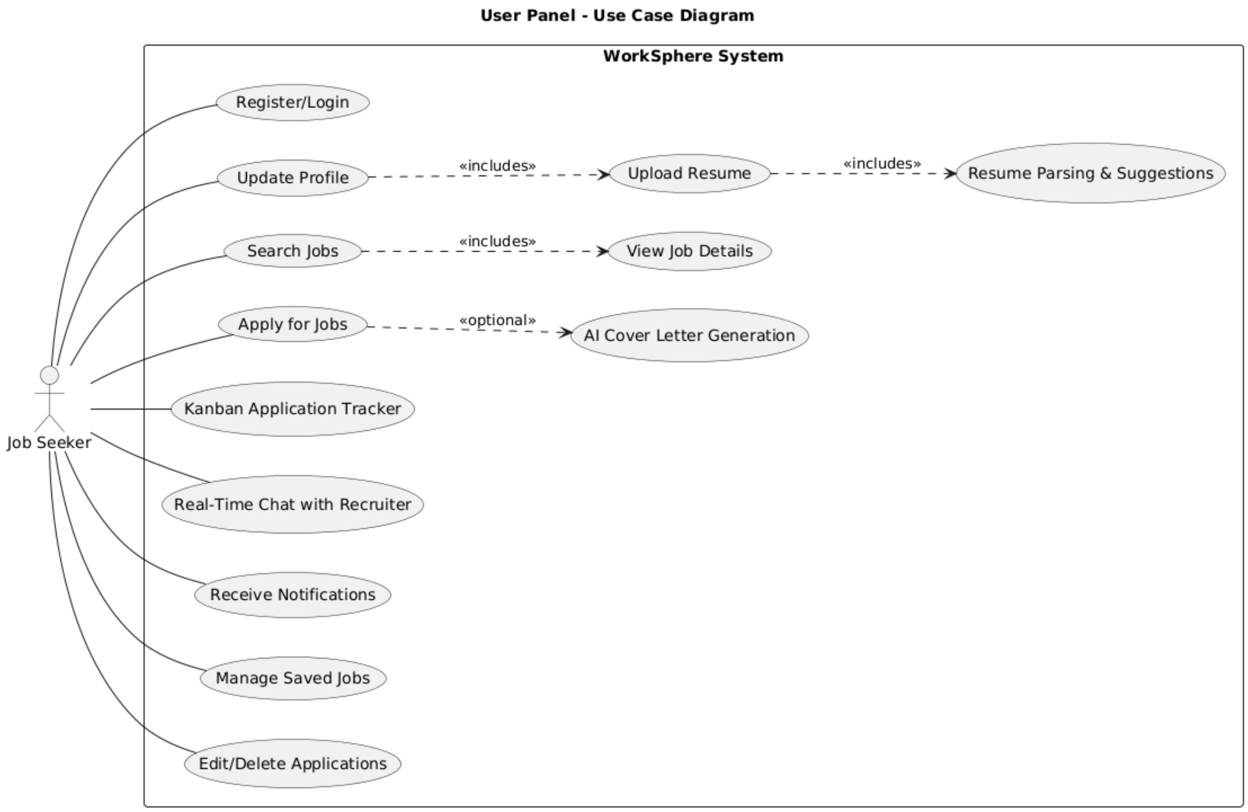
Explanation of the Diagram:

- **Requirements Analysis:** Gathering all functional and non-functional requirements.
- **System Design:** Divided into conceptual, logical, and physical design; creating a blueprint for the system.
- **Implementation:** Coding the system components (frontend, backend, database).

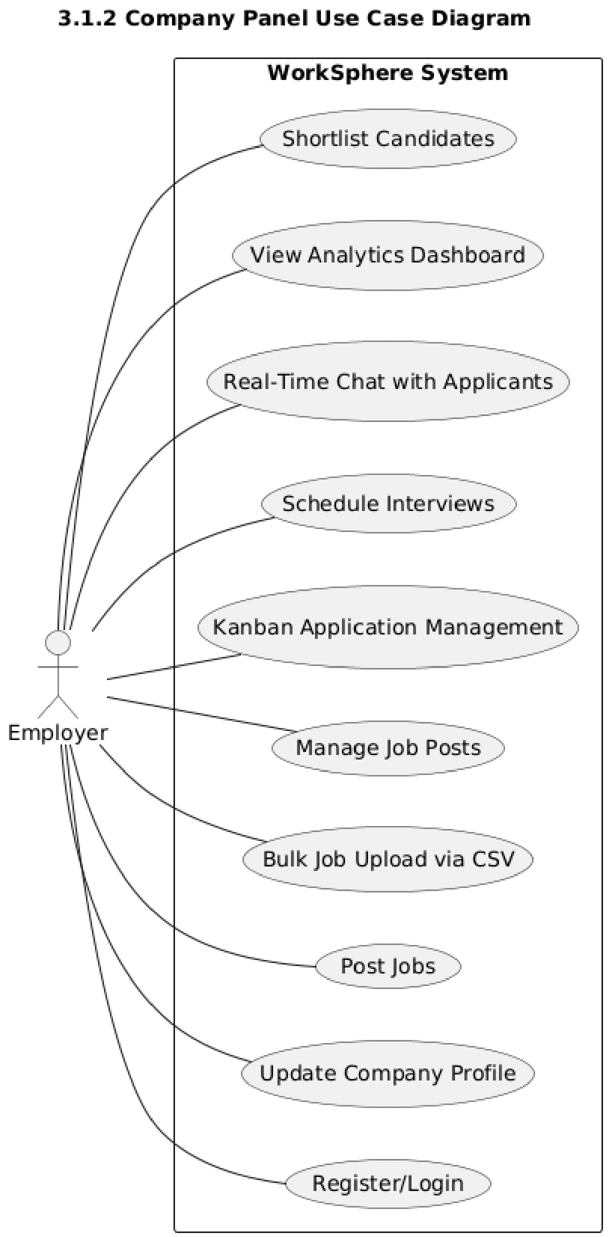
- **Testing & Verification:** Systematically verifying that the system meets the specified requirements.
- **Deployment:** Releasing the system to the live environment.
- **Maintenance:** Ongoing updates and enhancements based on user feedback and evolving needs.

3. Conceptual Diagram

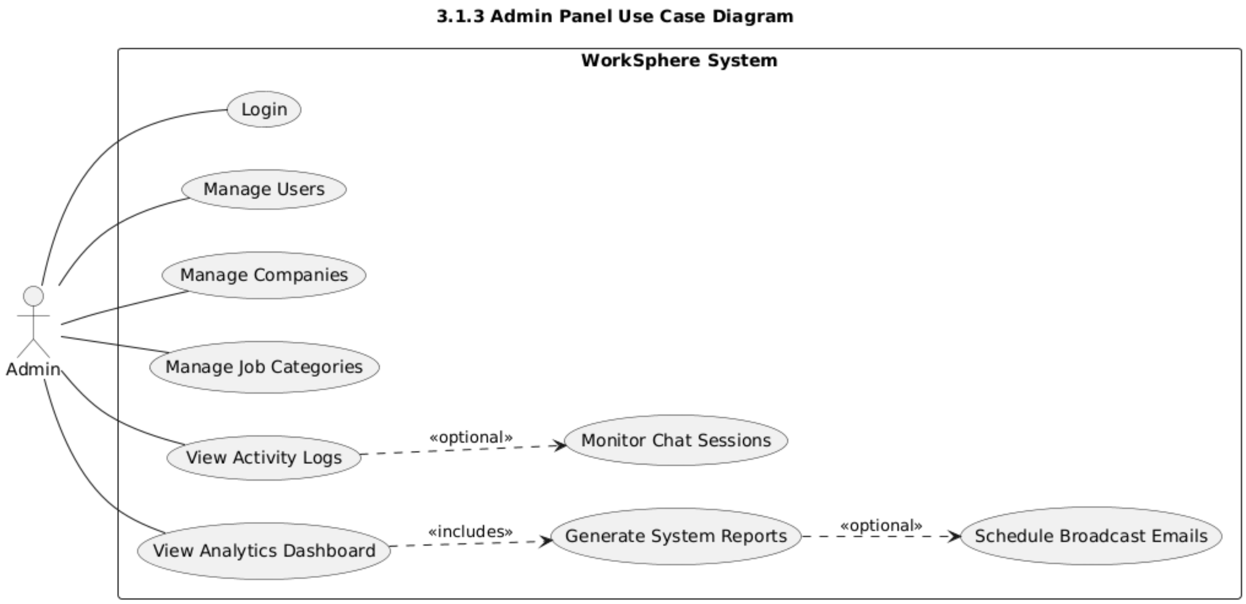
3.1 Use Case Diagram



3.1.1 User Panel Use Case Diagram

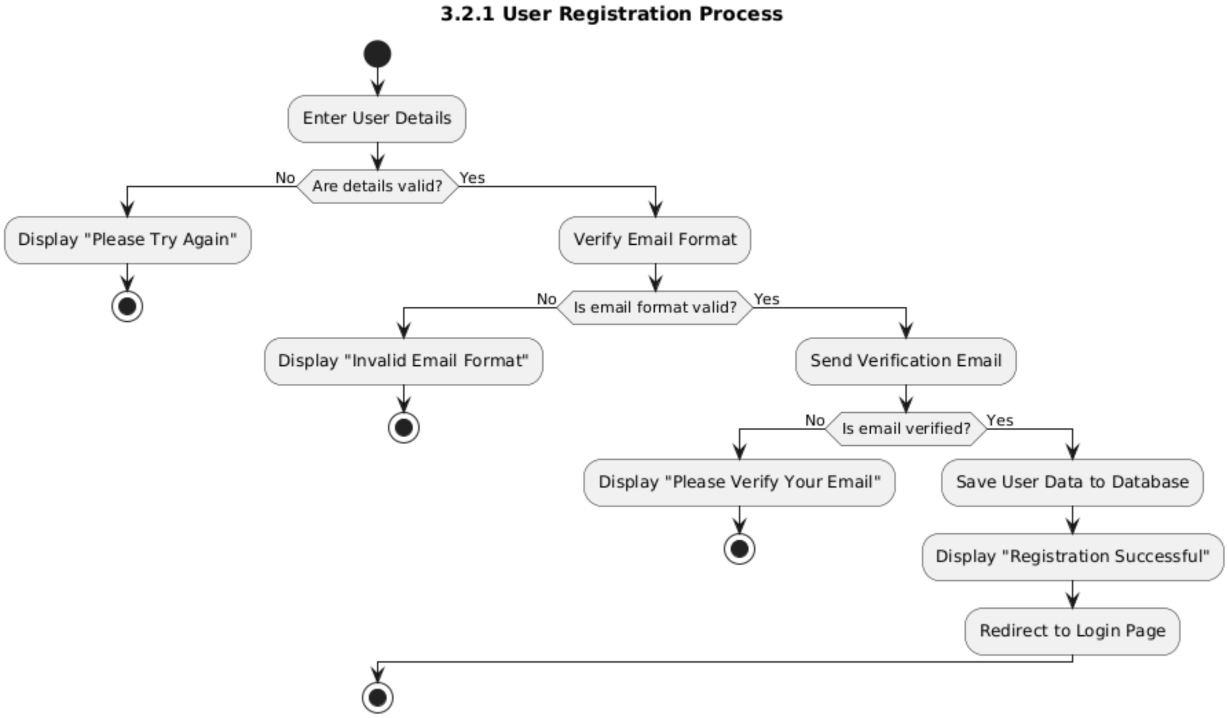


3.1.2 Company Panel Use Case Diagram

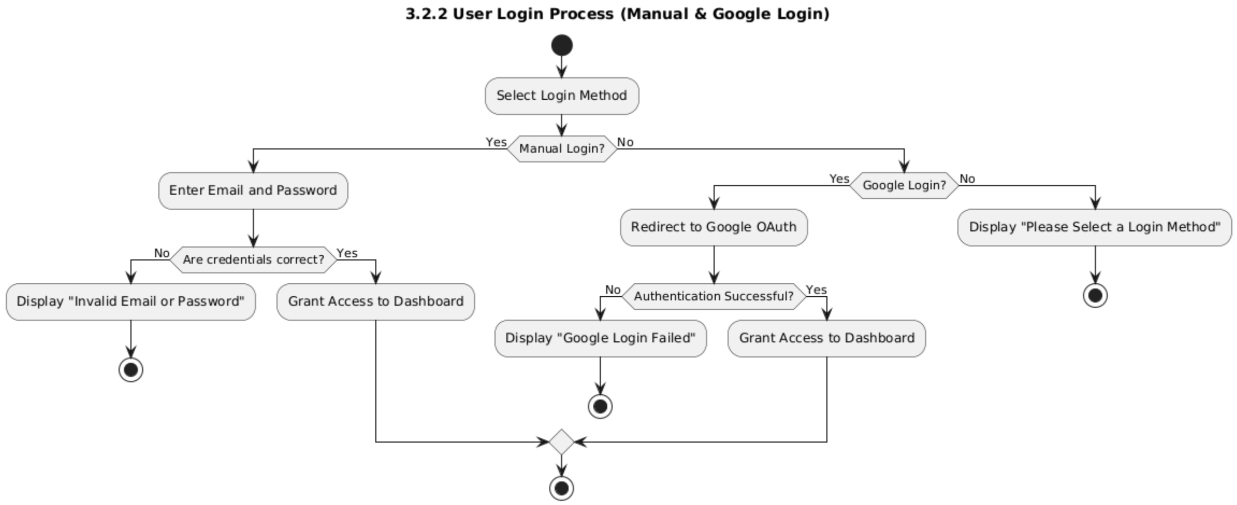


3.1.3 Admin Panel Use Case Diagram

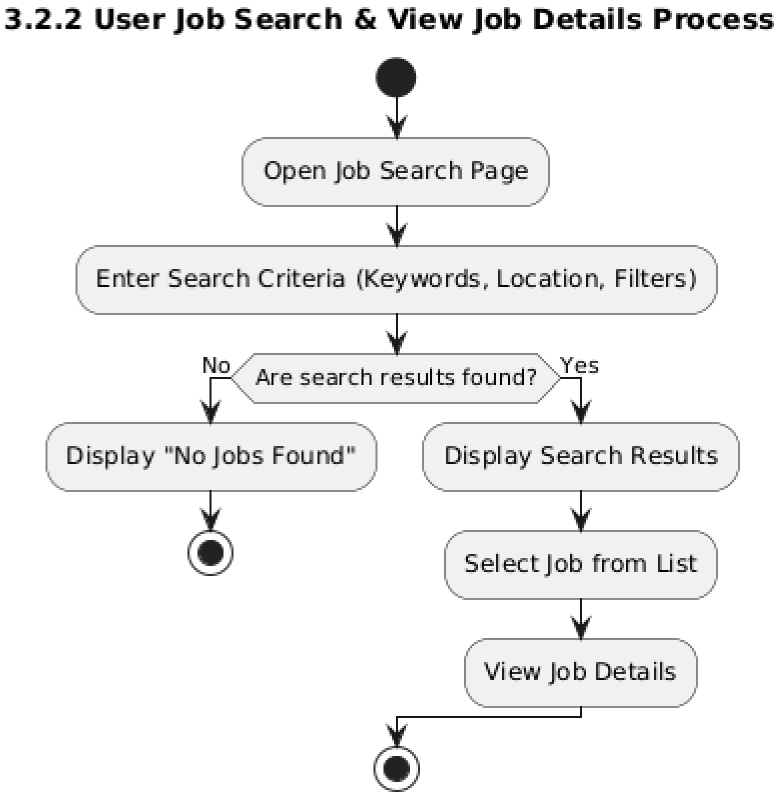
3.2 Activity Diagram


3.2.1 User Registration Process
Enter User Details
Are details valid?
No
Yes
Display "Please Try Again"
Verify Email Format
Is email format valid?
No
Yes
Display "Invalid Email Format"
Send Verification Email
Is email verified?
No
Yes
Display "Please Verify Your Email"
Save User Data to Database
Display "Registration Successful"
Redirect to Login Page

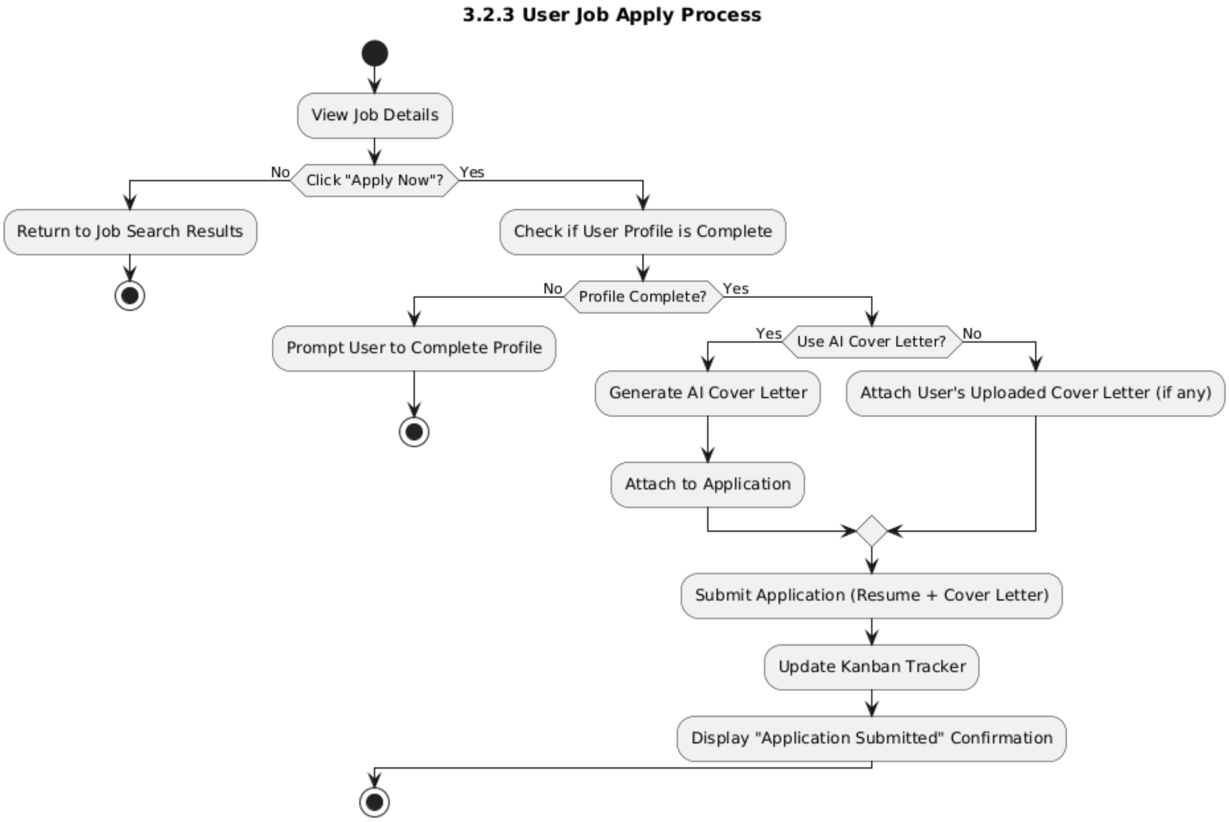
3.2.1. User Registration Process


3.2.2 User Login Process (Manual & Google Login)
Select Login Method
Manual Login?
Yes
No
Enter Email and Password
Are credentials correct?
No
Yes
Display "Invalid Email or Password"
Grant Access to Dashboard
Google Login?
Yes
No
Redirect to Google OAuth
Display "Please Select a Login Method"
Authentication Successful?
No
Yes
Display "Google Login Failed"
Grant Access to Dashboard

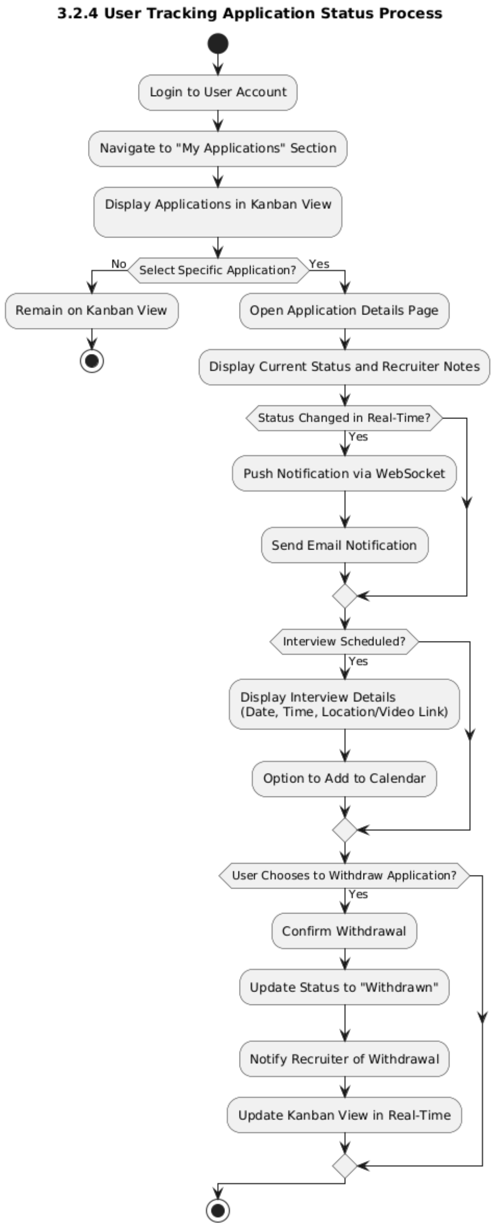
3.2.1 User Login Process


3.2.2 User Job Search & View Job Details Process
Open Job Search Page
Enter Search Criteria (Keywords, Location, Filters)
No
Are search results found?
Yes
Display "No Jobs Found"
Display Search Results
Select Job from List
View Job Details

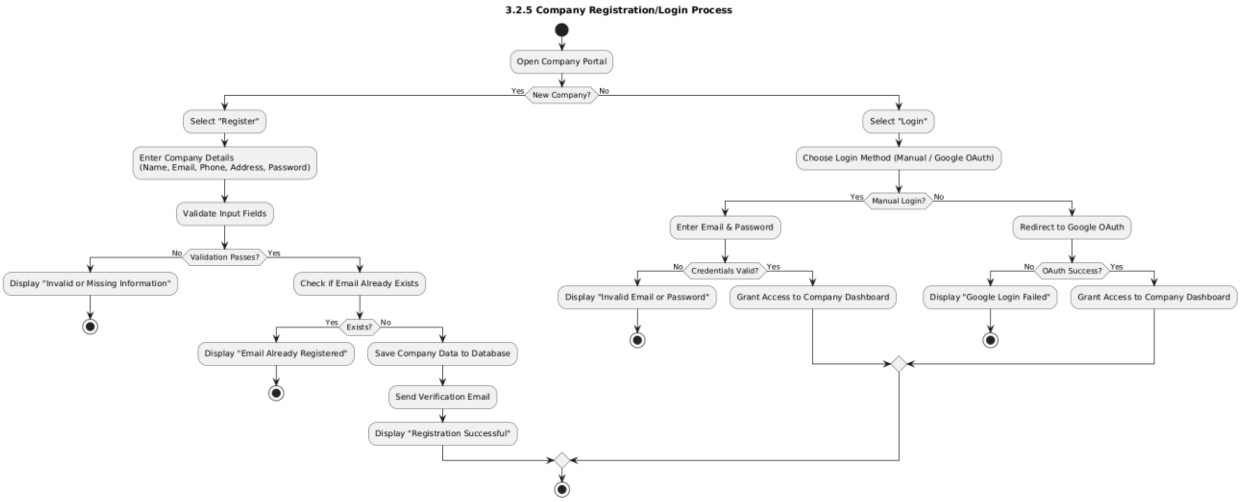
3.2.2 User Job Search & View Job Details Process


3.2.3 User Job Apply Process
View Job Details
Click "Apply Now"?
No
Yes
Return to Job Search Results
Check if User Profile is Complete
Profile Complete?
No
Yes
Prompt User to Complete Profile
Use AI Cover Letter?
Yes
No
Generate AI Cover Letter
Attach User's Uploaded Cover Letter (if any)
Attach to Application
Submit Application (Resume + Cover Letter)
Update Kanban Tracker
Display "Application Submitted" Confirmation

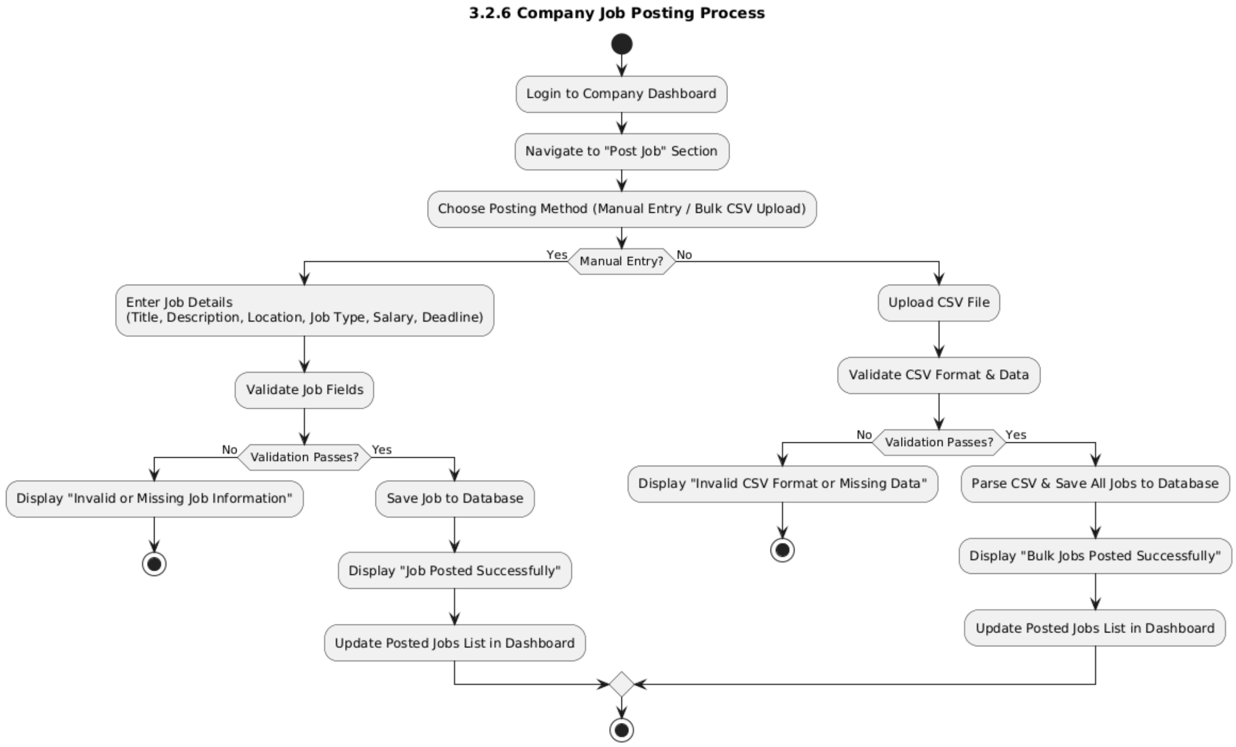
3.2.3 User Job Apply Process


3.2.4 User Tracking Application Status Process
Login to User Account
Navigate to "My Applications" Section
Display Applications in Kanban View
Select Specific Application?
No
Yes
Remain on Kanban View
Open Application Details Page
Display Current Status and Recruiter Notes
Status Changed in Real-Time?
Yes
Push Notification via WebSocket
Send Email Notification
Interview Scheduled?
Yes
Display Interview Details
(Date, Time, Location/Video Link)
Option to Add to Calendar
User Chooses to Withdraw Application?
Yes
Confirm Withdrawal
Update Status to "Withdrawn"
Notify Recruiter of Withdrawal
Update Kanban View in Real-Time

3.2.4 User Tracking Application Status Process

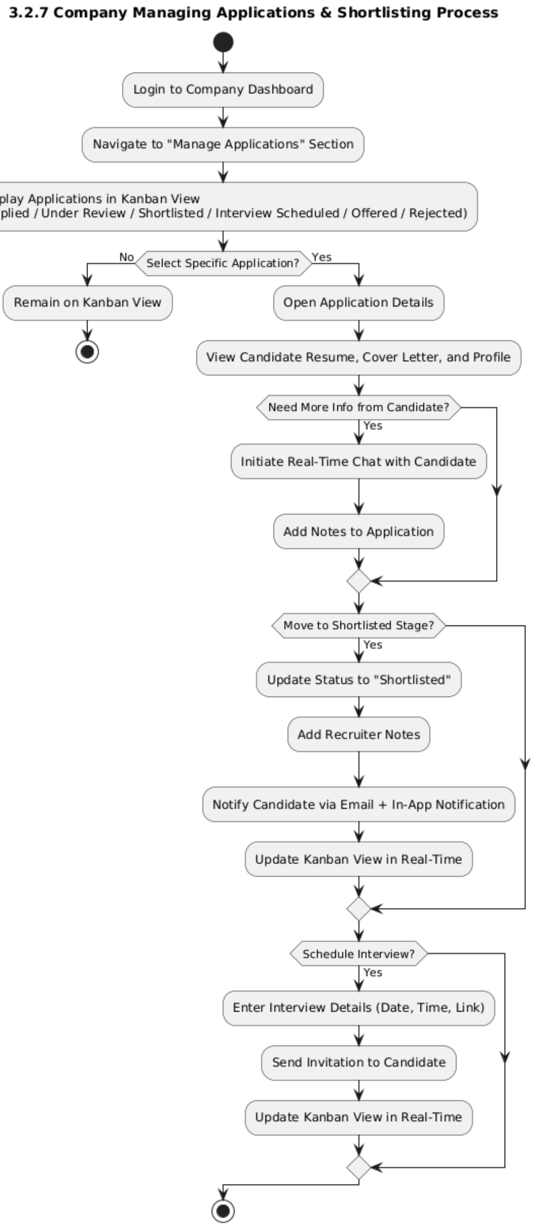

3.2.5 Company Registration/Login Process
Open Company Portal
New Company?
Yes
No
Select "Register"
Enter Company Details
(Name, Email, Phone, Address, Password)
Validate Input Fields
Validation Passes?
No
Yes
Display "Invalid or Missing Information"
Check if Email Already Exists
Exists?
Yes
No
Display "Email Already Registered"
Save Company Data to Database
Send Verification Email
Display "Registration Successful"
Select "Login"
Choose Login Method (Manual / Google OAuth)
Manual Login?
Yes
No
Enter Email & Password
Redirect to Google OAuth
Credentials Valid?
No
Yes
Display "Invalid Email or Password"
Grant Access to Company Dashboard
OAuth Success?
No
Yes
Display "Google Login Failed"
Grant Access to Company Dashboard

3.2.5 Company Registration/Login Process

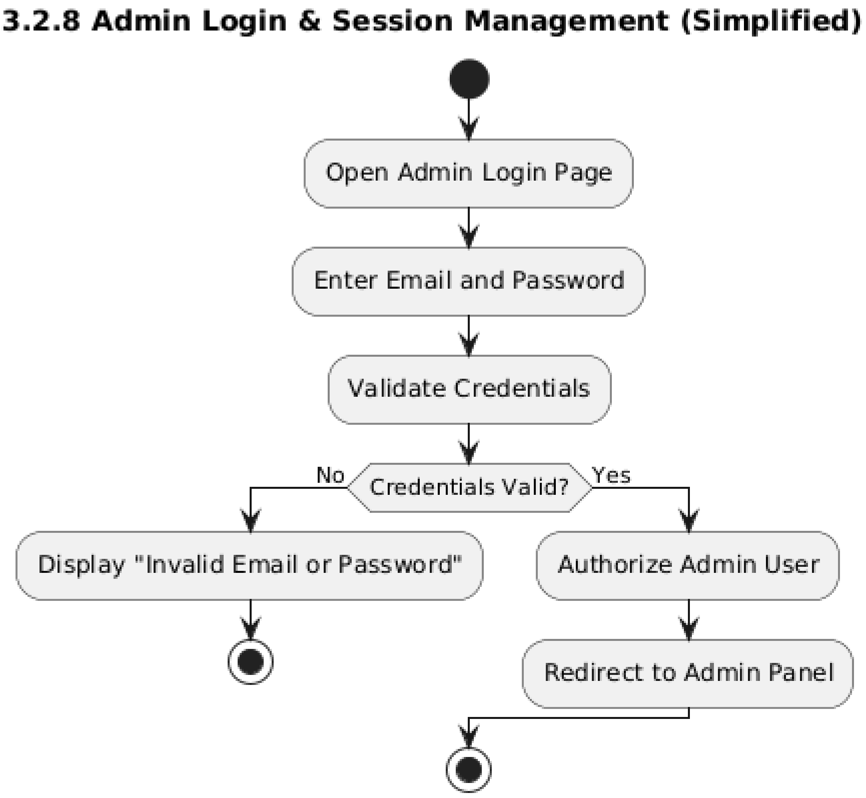


3.2.6 Company Job Posting Process
Login to Company Dashboard
Navigate to "Post Job" Section
Choose Posting Method (Manual Entry / Bulk CSV Upload)
Manual Entry?
Yes
No
Enter Job Details
(Title, Description, Location, Job Type, Salary, Deadline)
Validate Job Fields
Validation Passes?
No
Yes
Display "Invalid or Missing Job Information"
Save Job to Database
Display "Job Posted Successfully"
Update Posted Jobs List in Dashboard
Upload CSV File
Validate CSV Format & Data
Validation Passes?
No
Yes
Display "Invalid CSV Format or Missing Data"
Parse CSV & Save All Jobs to Database
Display "Bulk Jobs Posted Successfully"
Update Posted Jobs List in Dashboard

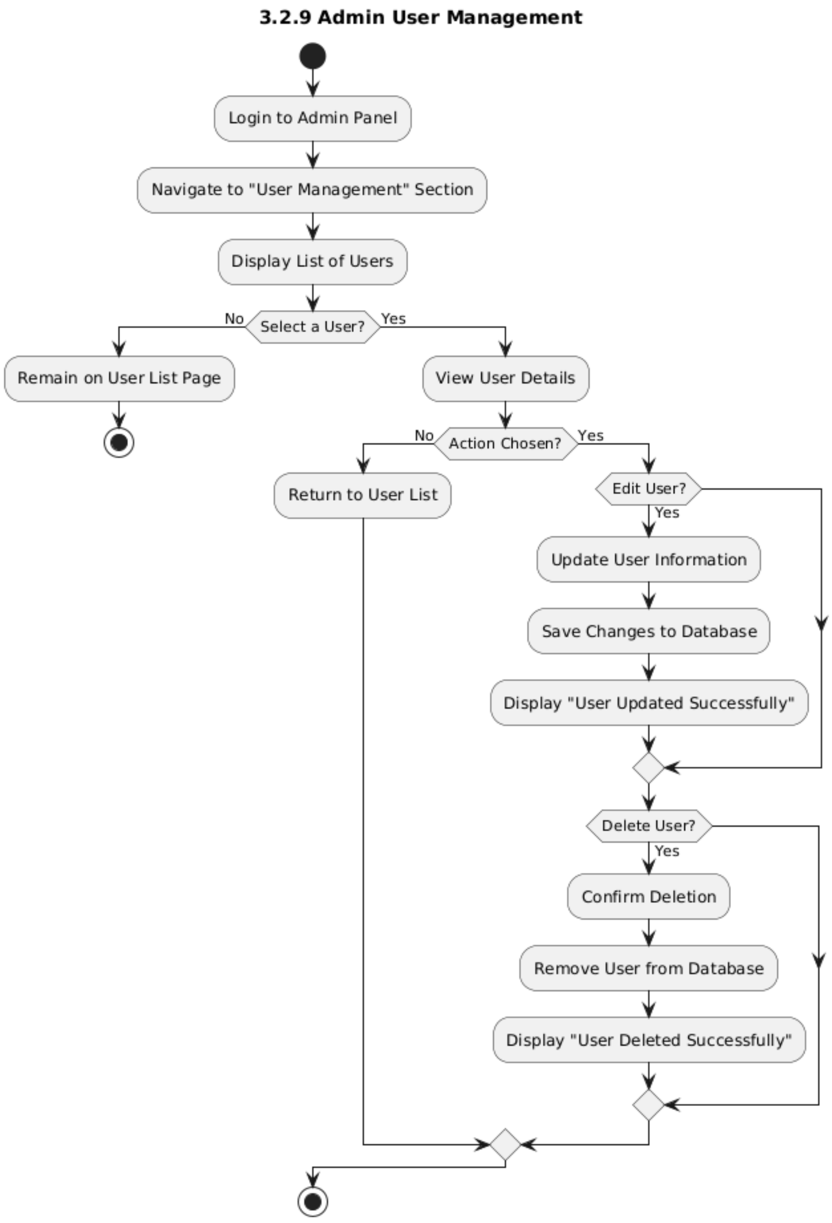
3.2.6 Company Job Posting Process


3.2.7 Company Managing Applications & Shortlisting Process
Login to Company Dashboard
Navigate to "Manage Applications" Section
Display Applications in Kanban View
(Applied / Under Review / Shortlisted / Interview Scheduled / Offered / Rejected)
Select Specific Application?
No
Yes
Remain on Kanban View
Open Application Details
View Candidate Resume, Cover Letter, and Profile
Need More Info from Candidate?
Yes
Initiate Real-Time Chat with Candidate
Add Notes to Application
Move to Shortlisted Stage?
Yes
Update Status to "Shortlisted"
Add Recruiter Notes
Notify Candidate via Email + In-App Notification
Update Kanban View in Real-Time
Schedule Interview?
Yes
Enter Interview Details (Date, Time, Link)
Send Invitation to Candidate
Update Kanban View in Real-Time

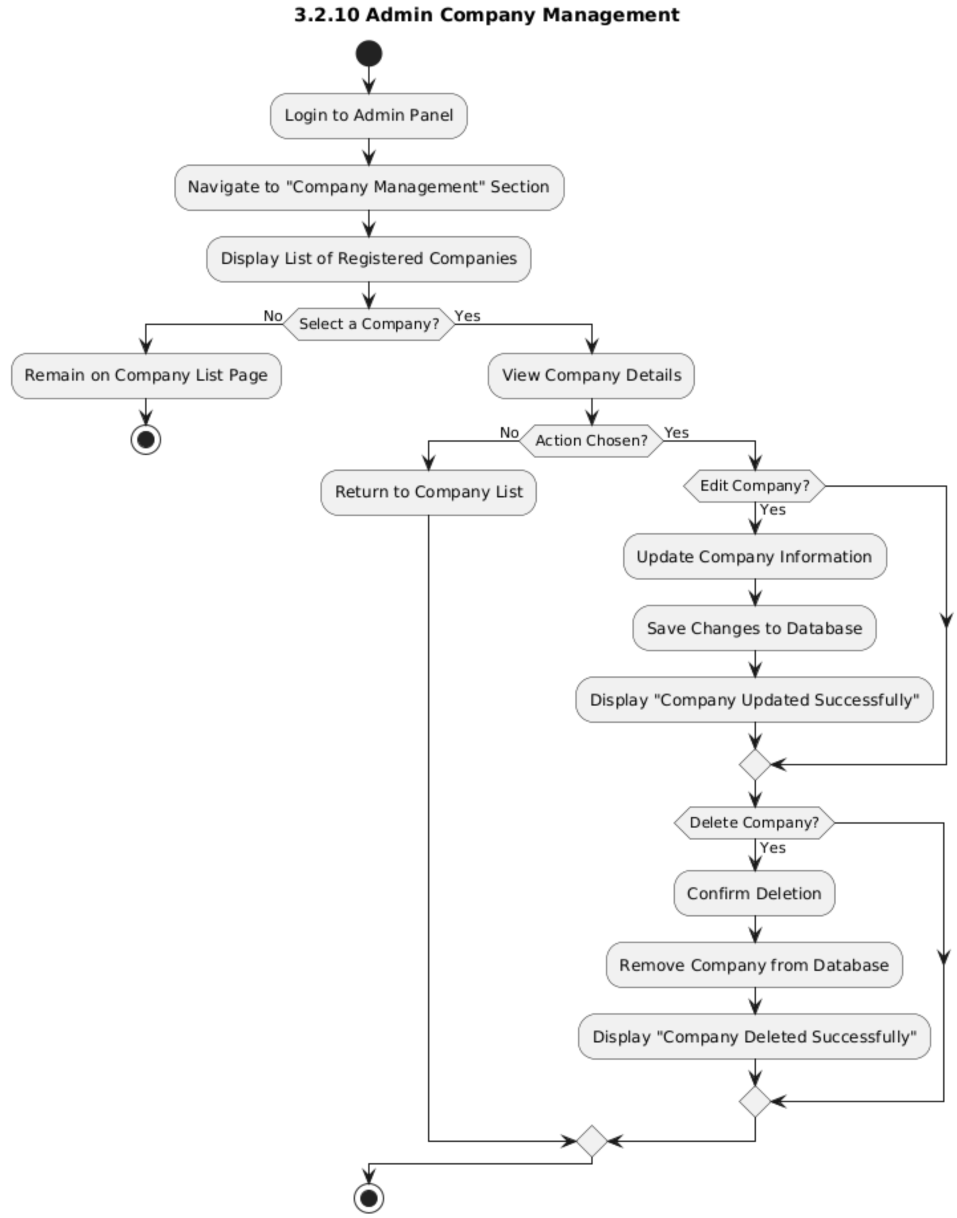
3.2.7 Company Managing Applications & Shortlisting Process



3.2.8 Admin Login & Session Management



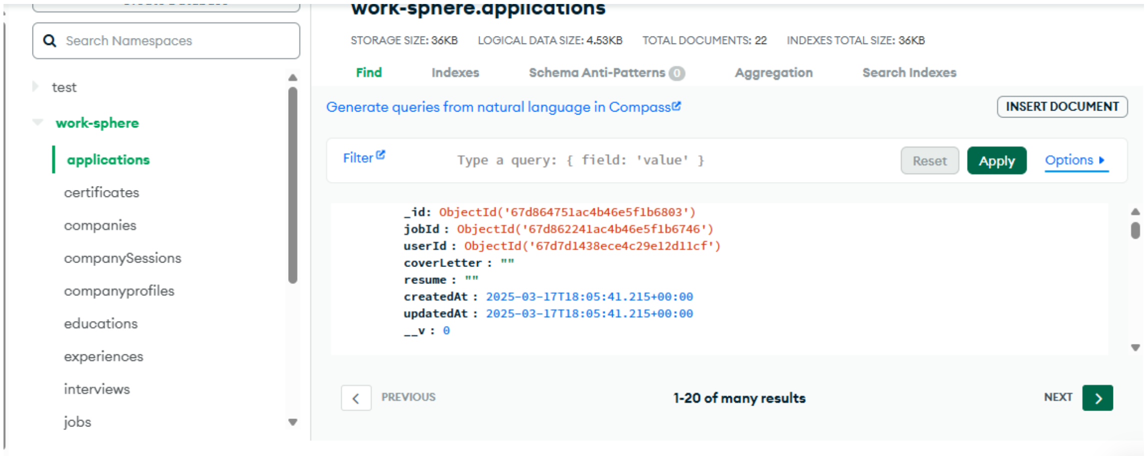
3.2.9 Admin User Management


3.2.10 Admin Company Management
Login to Admin Panel
Navigate to "Company Management" Section
Display List of Registered Companies
Select a Company?
No
Yes
Remain on Company List Page
View Company Details
Action Chosen?
No
Yes
Return to Company List
Edit Company?
Yes
Update Company Information
Save Changes to Database
Display "Company Updated Successfully"
Delete Company?
Yes
Confirm Deletion
Remove Company from Database
Display "Company Deleted Successfully"

3.2.10 Admin Company Management

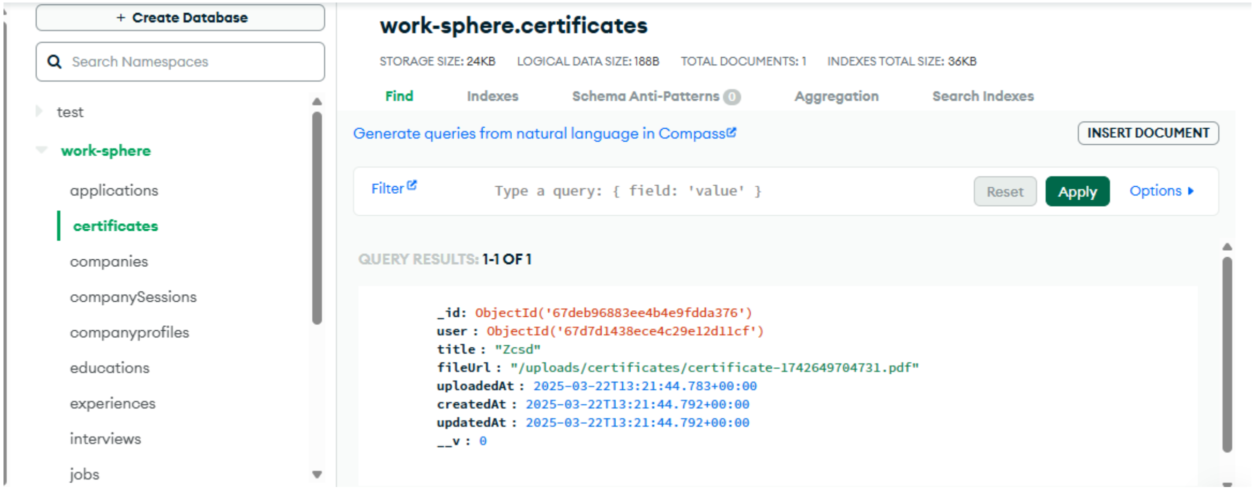
4. System Design

3.1 Database Design



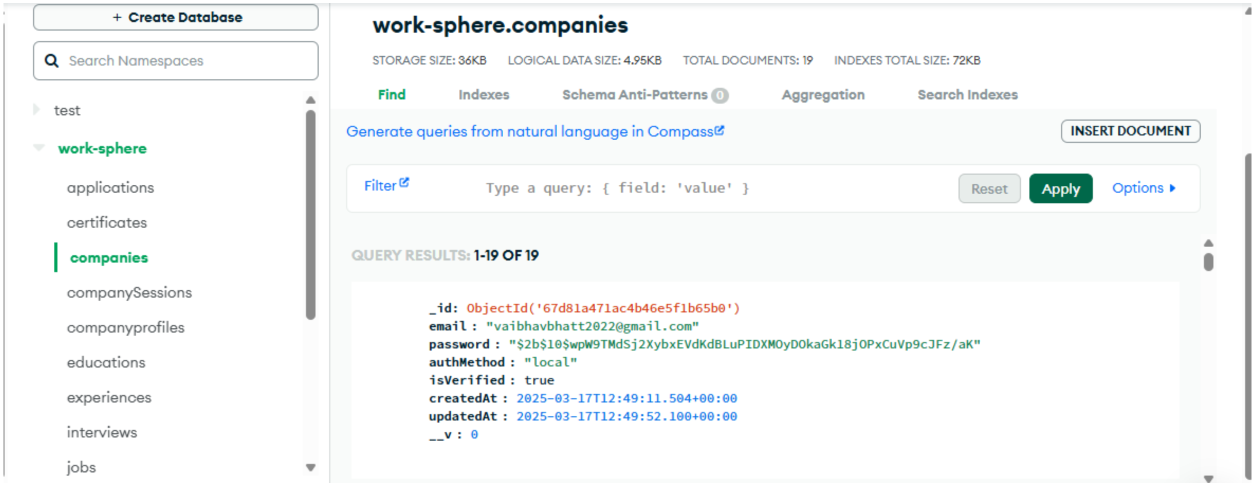
```
_id: ObjectId('67d864751ac4b46e5f1b6803')
jobId : ObjectId('67d862241ac4b46e5f1b6746')
userId : ObjectId('67d7d1438ece4c29e12d11cf')
coverLetter : ""
resume : ""
createdAt : 2025-03-17T18:05:41.215+00:00
updatedAt : 2025-03-17T18:05:41.215+00:00
__v : 0
```

4.1.1 Applications Document



```
_id: ObjectId('67deb96883ee4b4e9fdda376')
user : ObjectId('67d7d1438ece4c29e12d11cf')
title : "Zcsd"
fileUrl : "/uploads/certificates/certificate-1742649704731.pdf"
uploadedAt : 2025-03-22T13:21:44.783+00:00
createdAt : 2025-03-22T13:21:44.792+00:00
updatedAt : 2025-03-22T13:21:44.792+00:00
__v : 0
```

4.1.2 Certificate Document



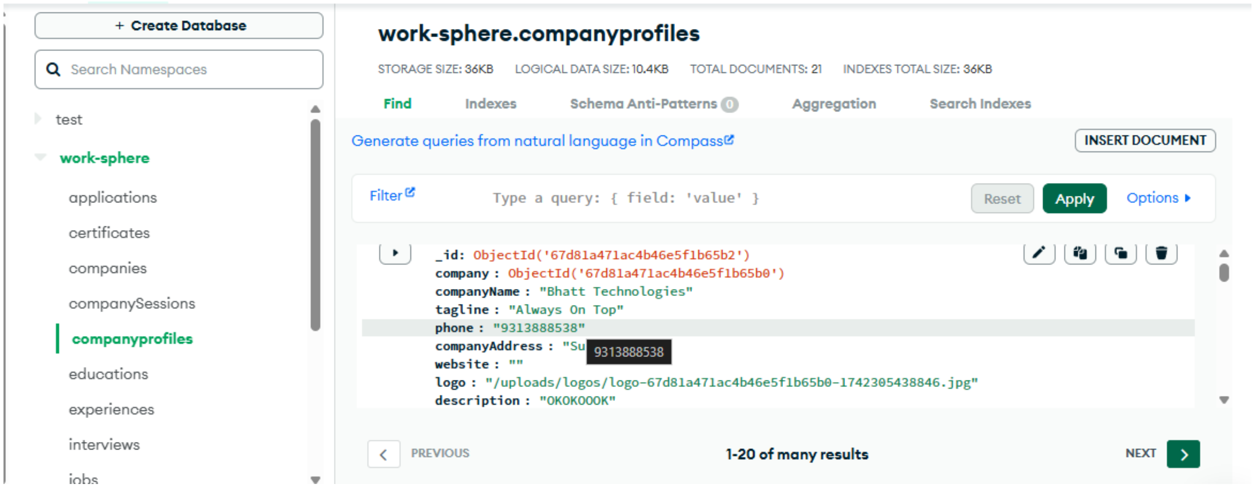
work-sphere.companies

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 4.95KB TOTAL DOCUMENTS: 19 INDEXES TOTAL SIZE: 72KB

QUERY RESULTS: 1-19 OF 19

```
_id: ObjectId('67d81a471ac4b46e5f1b65b0')
email : "vaibhavbhatt2022@gmail.com"
password : "$2b$10$wpW9TMdSj2XybxEVdKdBLuPIDXMOyDOkaGk18jOPxCuVp9cJFz/aK"
authMethod : "local"
isVerified : true
createdAt : 2025-03-17T12:49:11.504+00:00
updatedAt : 2025-03-17T12:49:52.100+00:00
__v : 0
```

4.1.3 Companies Document



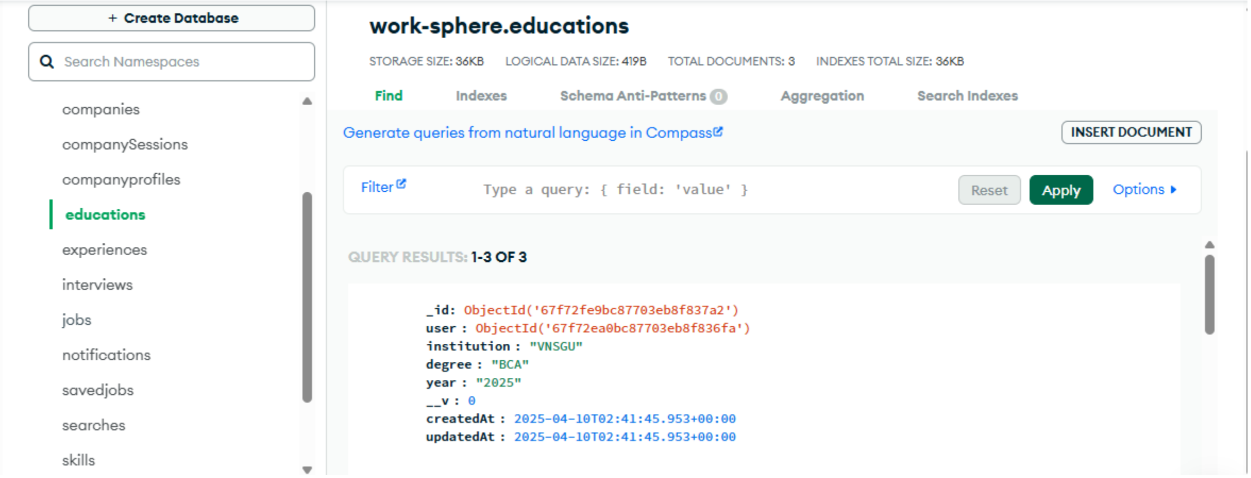
work-sphere.companyprofiles

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 10.4KB TOTAL DOCUMENTS: 21 INDEXES TOTAL SIZE: 36KB

```
_id: ObjectId('67d81a471ac4b46e5f1b65b2')
company : ObjectId('67d81a471ac4b46e5f1b65b0')
companyName : "Bhatt Technologies"
tagline : "Always On Top"
phone : "9313888538"
companyAddress : "Su 9313888538
website : ""
logo : "/uploads/logos/logo-67d81a471ac4b46e5f1b65b0-1742305438846.jpg"
description : "OKOKOOOK"
```

1-20 of many results

4.1.4 Company Profile Document



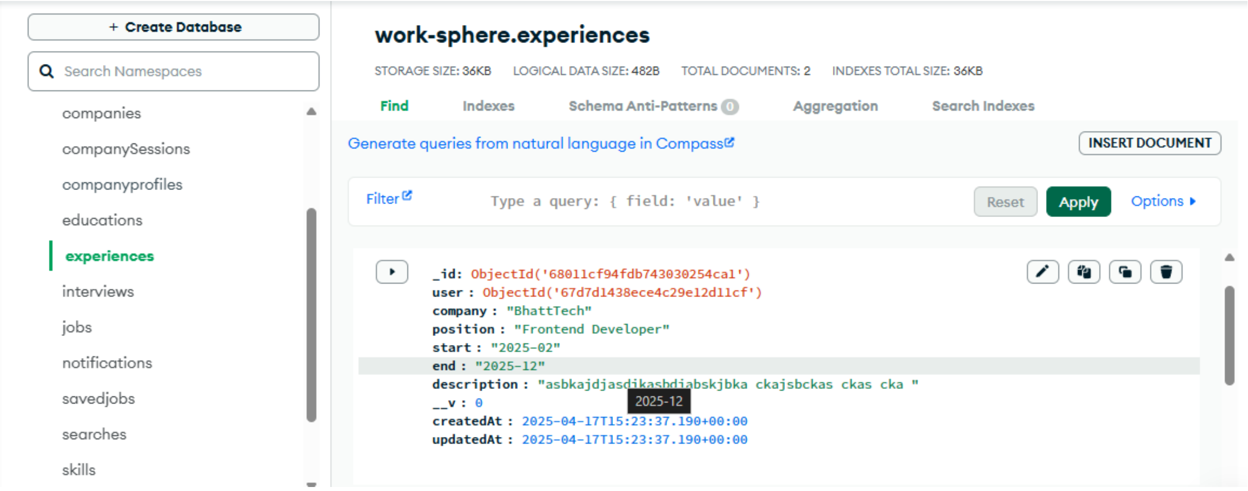
work-sphere.educations

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 419B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('67f72fe9bc87703eb8f837a2')
user : ObjectId('67f72ea0bc87703eb8f836fa')
institution : "VNSGU"
degree : "BCA"
year : "2025"
__v : 0
createdAt : 2025-04-10T02:41:45.953+00:00
updatedAt : 2025-04-10T02:41:45.953+00:00
```

4.1.5 Education Document

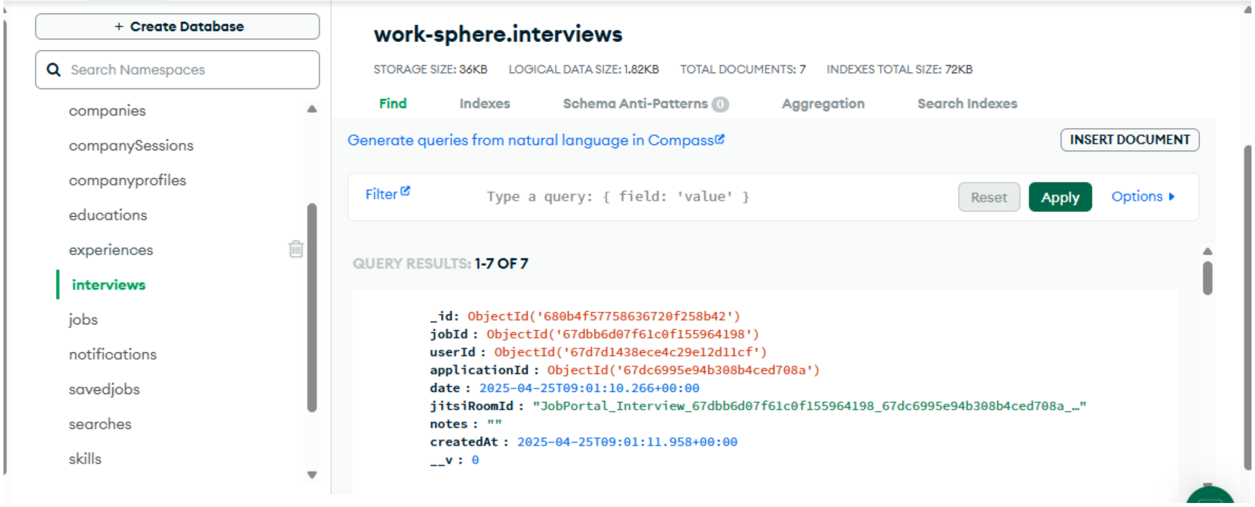


work-sphere.experiences

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 482B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

```
_id: ObjectId('68011cf94fdb743030254ca1')
user : ObjectId('67d7d1438ece4c29e12d11cf')
company : "BhattTech"
position : "Frontend Developer"
start : "2025-02"
end : "2025-12"
description : "asbkajdjasdikasbdiabskjbka ckajsbckas ckas cka "
__v : 0
createdAt : 2025-04-17T15:23:37.190+00:00
updatedAt : 2025-04-17T15:23:37.190+00:00
```

4.1.6 Experiences Document



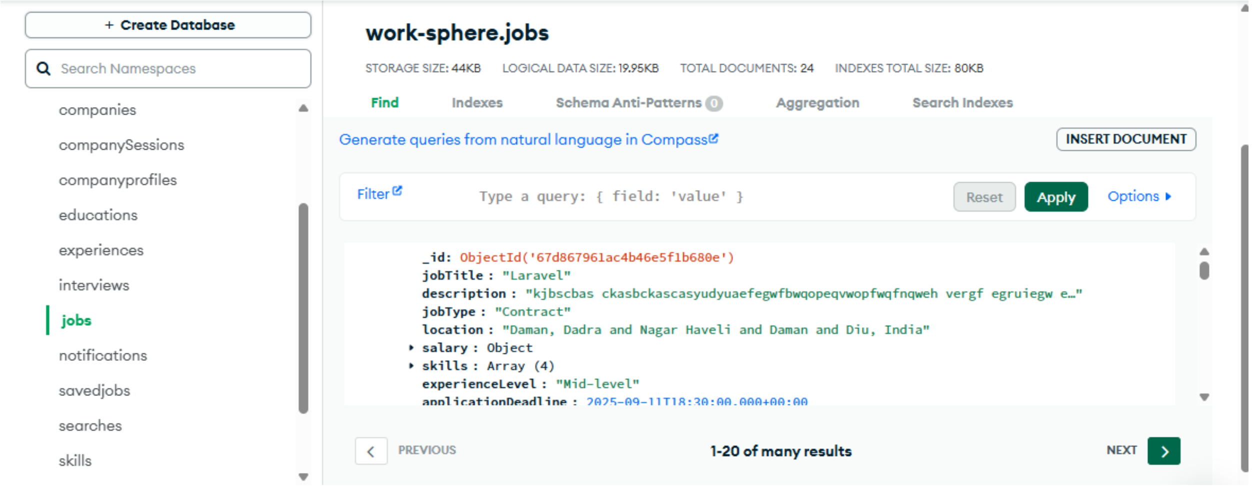
work-sphere.interviews

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 1.82KB TOTAL DOCUMENTS: 7 INDEXES TOTAL SIZE: 72KB

QUERY RESULTS: 1-7 OF 7

```
_id: ObjectId('680b4f57758636720f258b42')
jobId : ObjectId('67dbb6d07f61c0f155964198')
userId : ObjectId('67d7d1438ece4c29e12d11cf')
applicationId : ObjectId('67dc6995e94b308b4ced708a')
date : 2025-04-25T09:01:10.266+00:00
jitsiRoomId : "JobPortal_Interview_67dbb6d07f61c0f155964198_67dc6995e94b308b4ced708a_…"
notes : ""
createdAt : 2025-04-25T09:01:11.958+00:00
__v : 0
```

4.1.7 Interview Document



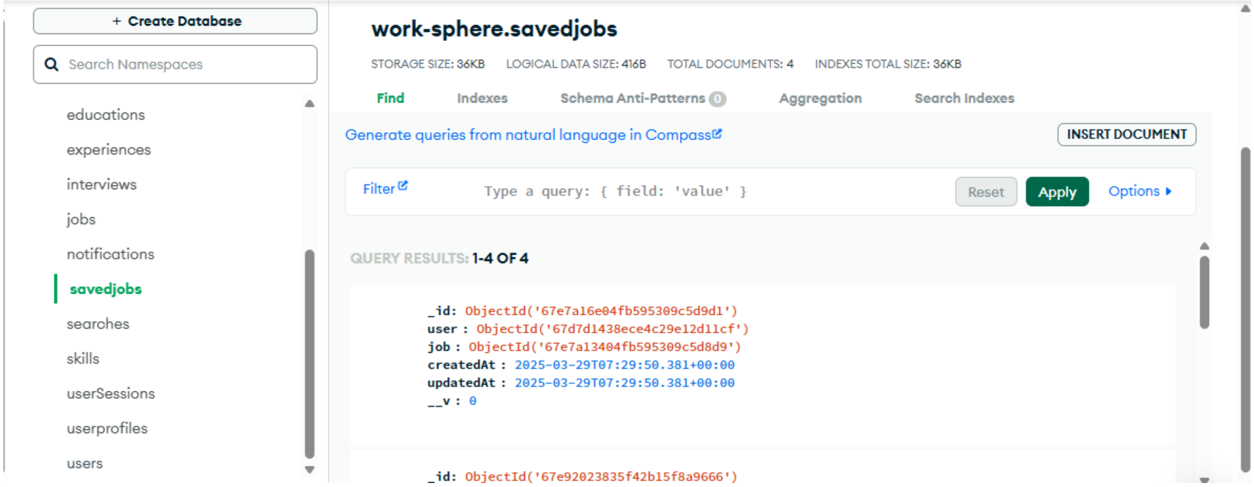
work-sphere.jobs

STORAGE SIZE: 44KB LOGICAL DATA SIZE: 19.95KB TOTAL DOCUMENTS: 24 INDEXES TOTAL SIZE: 80KB

```
  _id: ObjectId('67d867961ac4b46e5f1b680e')
  jobTitle : "Laravel"
  description : "kjbscbas ckasbckascasyudyuaefegwfbwqopeqvwopfwqfnqweh vergf egruiegw e…"
  jobType : "Contract"
  location : "Daman, Dadra and Nagar Haveli and Daman and Diu, India"
▸ salary : Object
▸ skills : Array (4)
  experienceLevel : "Mid-level"
  applicationDeadline : 2025-09-11T18:30:00.000+00:00
```

1-20 of many results

4.1.8 Jobs Document



work-sphere.savedjobs

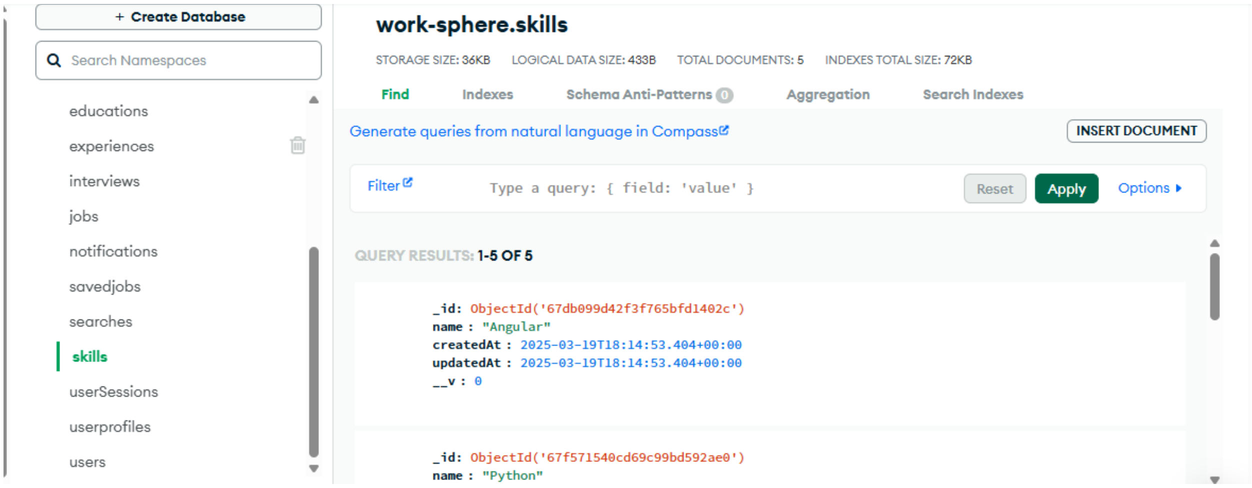
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 416B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-4 OF 4

```
_id: ObjectId('67e7a16e04fb595309c5d9d1')
user : ObjectId('67d7d1438ece4c29e12d11cf')
job : ObjectId('67e7a13404fb595309c5d8d9')
createdAt : 2025-03-29T07:29:50.381+00:00
updatedAt : 2025-03-29T07:29:50.381+00:00
__v : 0

_id: ObjectId('67e92023835f42b15f8a9666')
```

4.1.9 Saved Jobs Document



work-sphere.skills

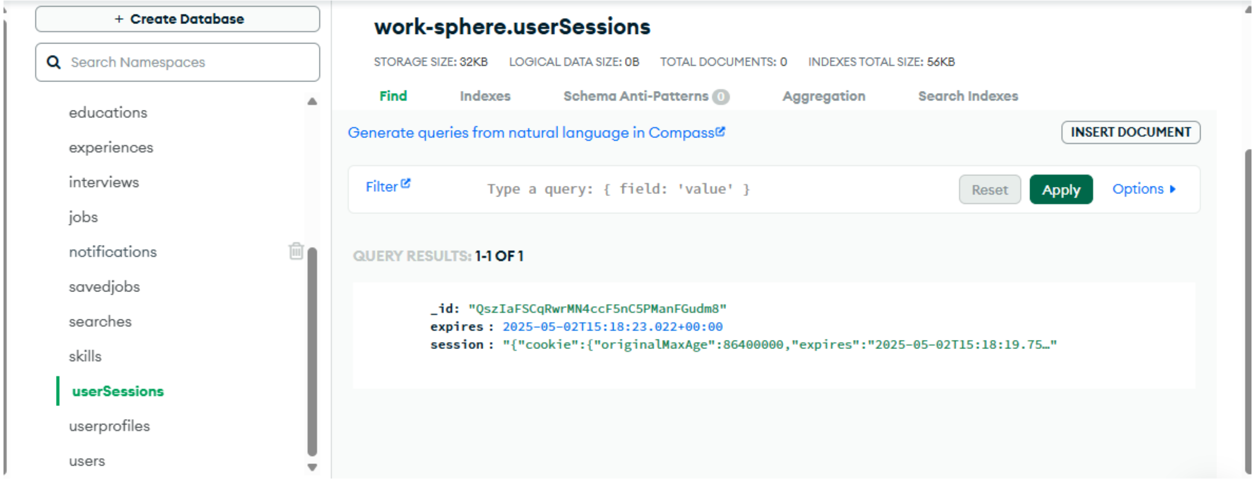
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 433B TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 72KB

QUERY RESULTS: 1-5 OF 5

```
_id: ObjectId('67db099d42f3f765bfd1402c')
name : "Angular"
createdAt : 2025-03-19T18:14:53.404+00:00
updatedAt : 2025-03-19T18:14:53.404+00:00
__v : 0

_id: ObjectId('67f571540cd69c99bd592ae0')
name : "Python"
```

4.1.10 Skills Document



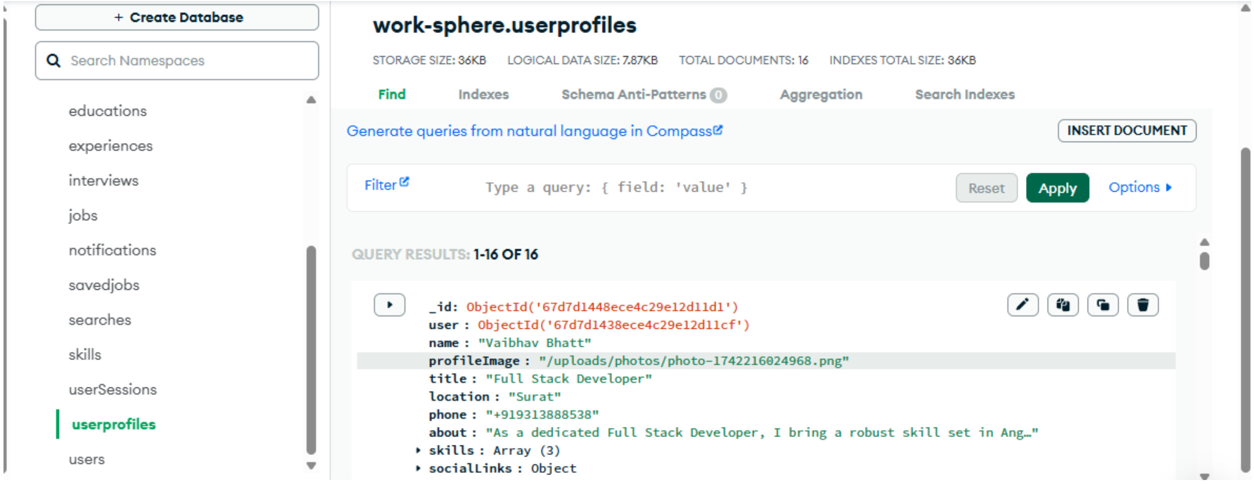
work-sphere.userSessions

STORAGE SIZE: 32KB LOGICAL DATA SIZE: 0B TOTAL DOCUMENTS: 0 INDEXES TOTAL SIZE: 56KB

QUERY RESULTS: 1-1 OF 1

```
_id: "QszIaFSCqRwrMN4ccF5nC5PManFGudm8"
expires : 2025-05-02T15:18:23.022+00:00
session : "{"cookie":{"originalMaxAge":86400000,"expires":"2025-05-02T15:18:19.75…"
```

4.1.11 User Session Document



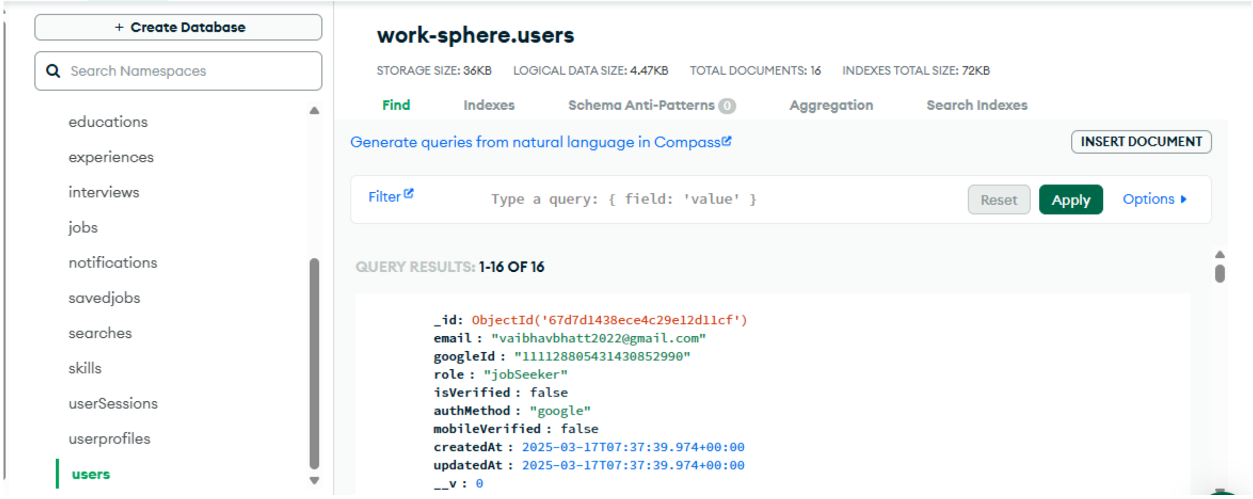
work-sphere.userprofiles

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 7.87KB TOTAL DOCUMENTS: 16 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-16 OF 16

```
_id: ObjectId('67d7d1448ece4c29e12d11d1')
user : ObjectId('67d7d1438ece4c29e12d11cf')
name : "Vaibhav Bhatt"
profileImage : "/uploads/photos/photo-1742216024968.png"
title : "Full Stack Developer"
location : "Surat"
phone : "+919313888538"
about : "As a dedicated Full Stack Developer, I bring a robust skill set in Ang…"
▸ skills : Array (3)
▸ socialLinks : Object
```

4.1.12 User Profile Document



work-sphere.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 4.47KB TOTAL DOCUMENTS: 16 INDEXES TOTAL SIZE: 72KB

QUERY RESULTS: 1-16 OF 16

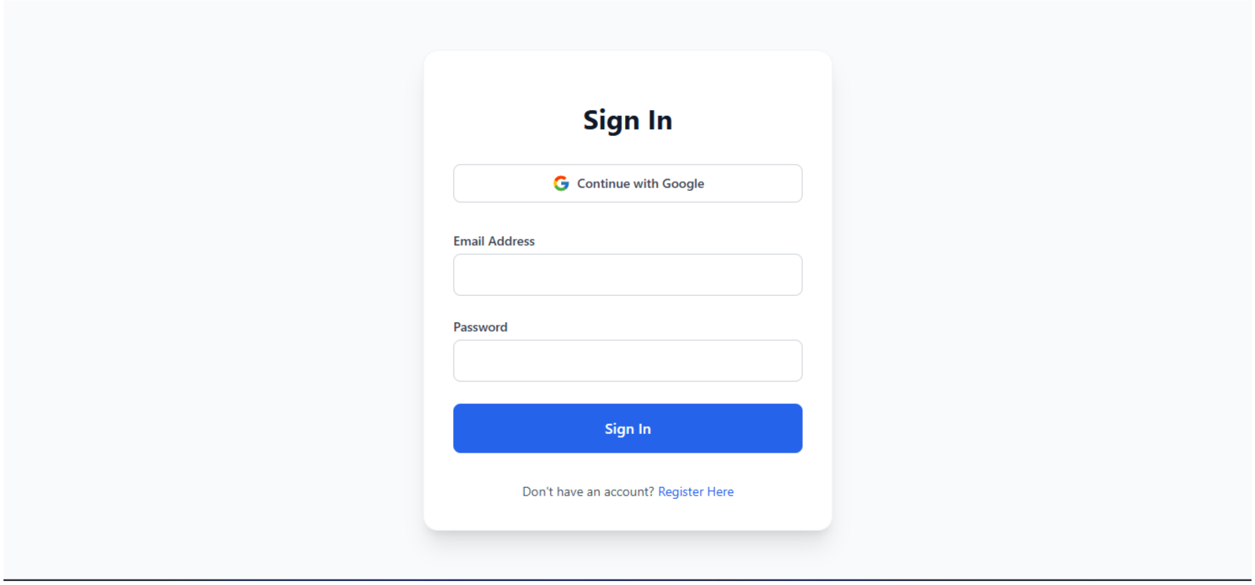
```
_id: ObjectId('67d7d1438ece4c29e12d11cf')
email : "vaibhavbhatt2022@gmail.com"
googleId : "111128805431430852990"
role : "jobSeeker"
isVerified : false
authMethod : "google"
mobileVerified : false
createdAt : 2025-03-17T07:37:39.974+00:00
updatedAt : 2025-03-17T07:37:39.974+00:00
__v : 0
```

4.1.13 User Document

4.2 User Interface Design

The user interface is divided into three main sections:

4.2.1 Job Seeker Interface:



Sign In

Continue with Google

Email Address

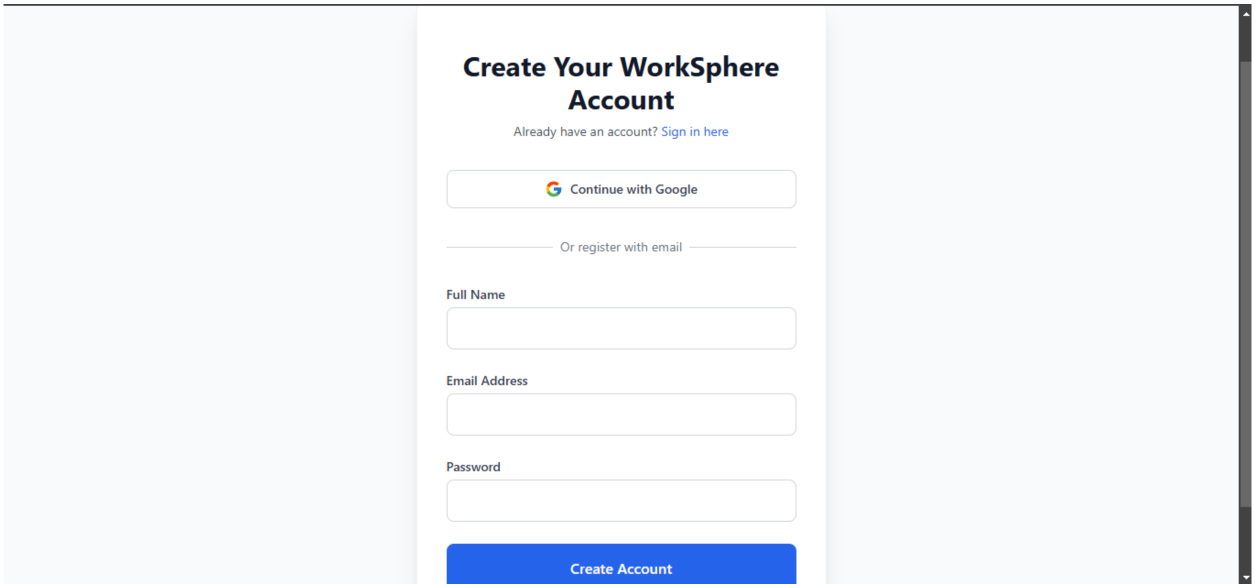
Password

Sign In

Don't have an account? Register Here

4.2.1.1 Job Seeker Login Page

Description	Job Seeker Sign in Page with Email and Password Field
Data From	User Collection
Data To	
Critical Validations	Email field contain only specific format and required Password field is also required



Create Your WorkSphere Account

Already have an account? Sign in here

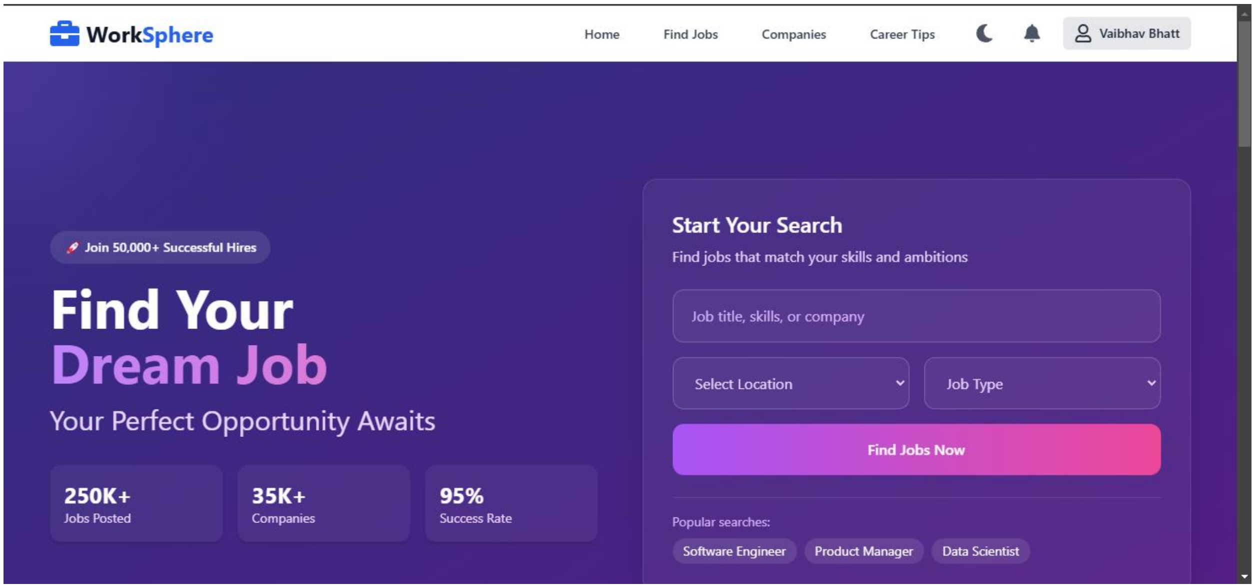
Full Name

Email Address

Password

4.2.1.2 Job Seeker Register Page

Description	Job Seeker Register Page with Full Name, Email And Password Field
Data From	
Data To	User Collection
Critical Validations	Email field contain only specific format and required Password field is required Password Field Must Contain At least 6 character that include numbers, alphabets and special characters



Join 50,000+ Successful Hires

Find Your Dream Job

Your Perfect Opportunity Awaits

250K+ Jobs Posted

35K+ Companies

95% Success Rate

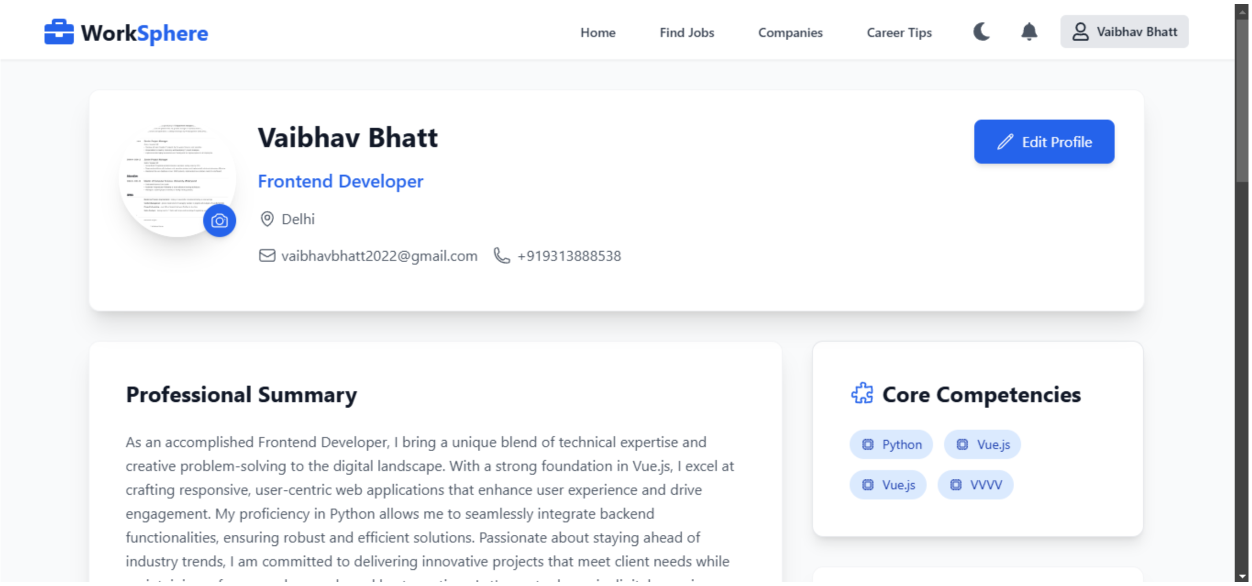
Start Your Search

Find jobs that match your skills and ambitions

Popular searches:

4.2.1.3 Job Seeker Home Page

Description	Job Seeker Sign Home Page That Displays All the Featured Companies and Jobs
Data From	Jobs Collection and Company Collection
Data To	
Critical Validations	



Vaibhav Bhatt

Frontend Developer

Delhi

vaibhavbhatt2022@gmail.com +919313888538

Professional Summary

As an accomplished Frontend Developer, I bring a unique blend of technical expertise and creative problem-solving to the digital landscape. With a strong foundation in Vue.js, I excel at crafting responsive, user-centric web applications that enhance user experience and drive engagement. My proficiency in Python allows me to seamlessly integrate backend functionalities, ensuring robust and efficient solutions. Passionate about staying ahead of industry trends, I am committed to delivering innovative projects that meet client needs while

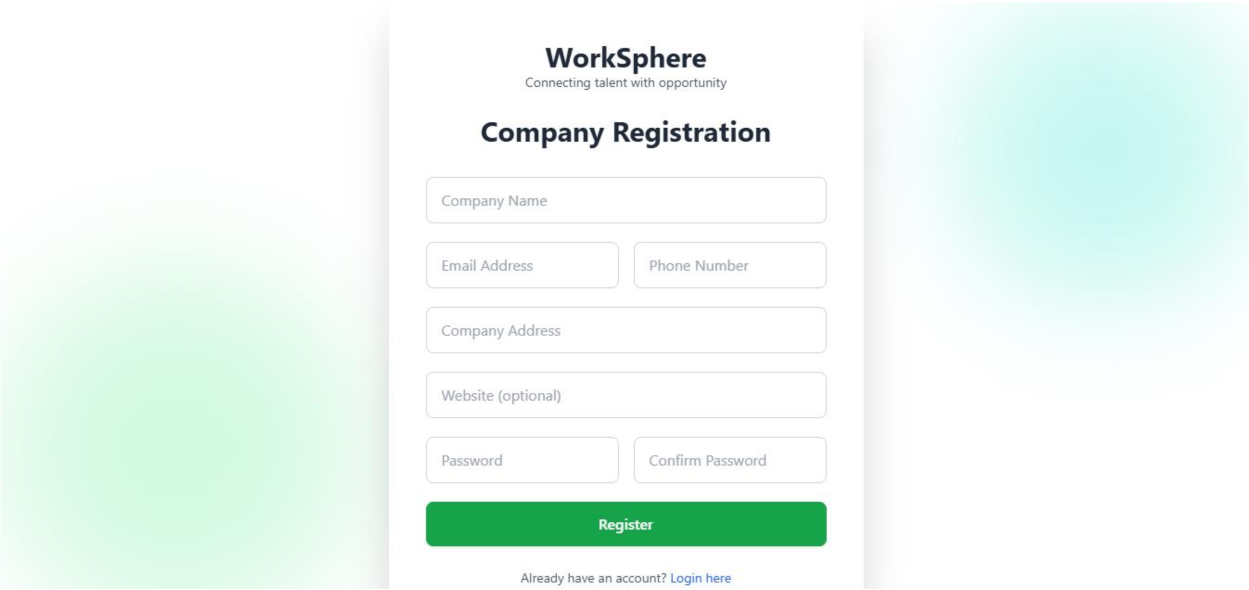
Core Competencies

Python Vue.js Vue.js VVVV

4.2.1.4 Job Seeker Profile Page

Description	Job Seeker Profile Page That Contains All Required Components That Contains All User Details That Is Used for Applying in Job
Data From	User And User Profile Collection
Data To	User Profile Collection
Critical Validations	

3.2.2 Employer Interface:



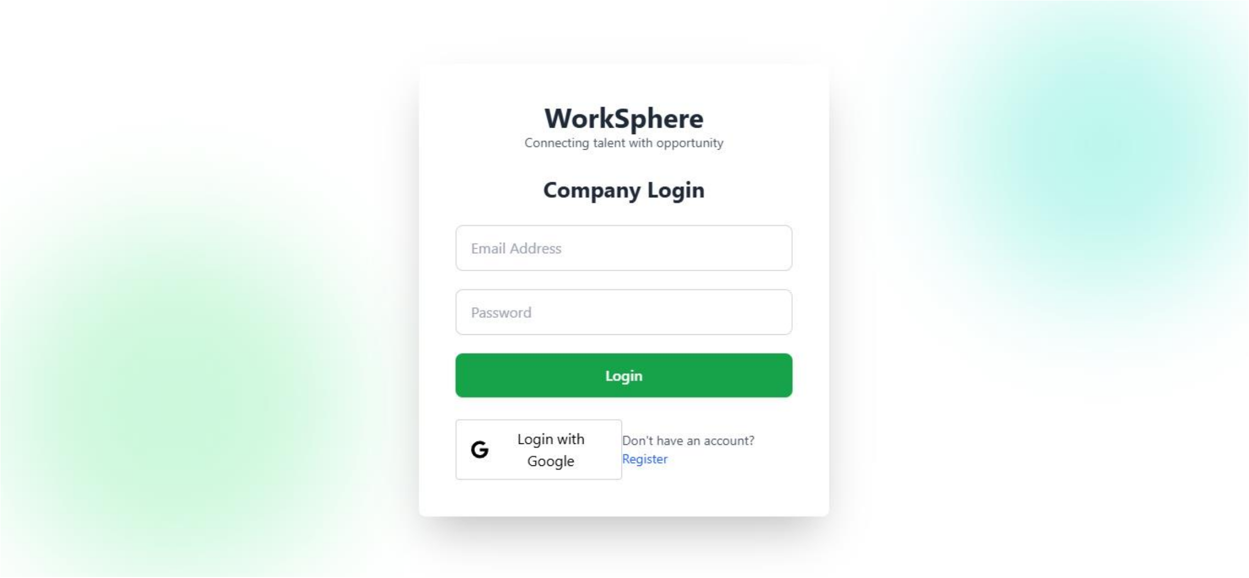
WorkSphere

Connecting talent with opportunity

Company Registration

4.2.2.1 Company Registration Page

Description	Employer/Company Registration Page
Data From	
Data To	Company Collection
Critical Validations	Company Name Is Required Email Must Contain Specific Format Phone Number Is Required Company Address Is Required Password Must Match Specific criteria Password And Confirm Password Should Match



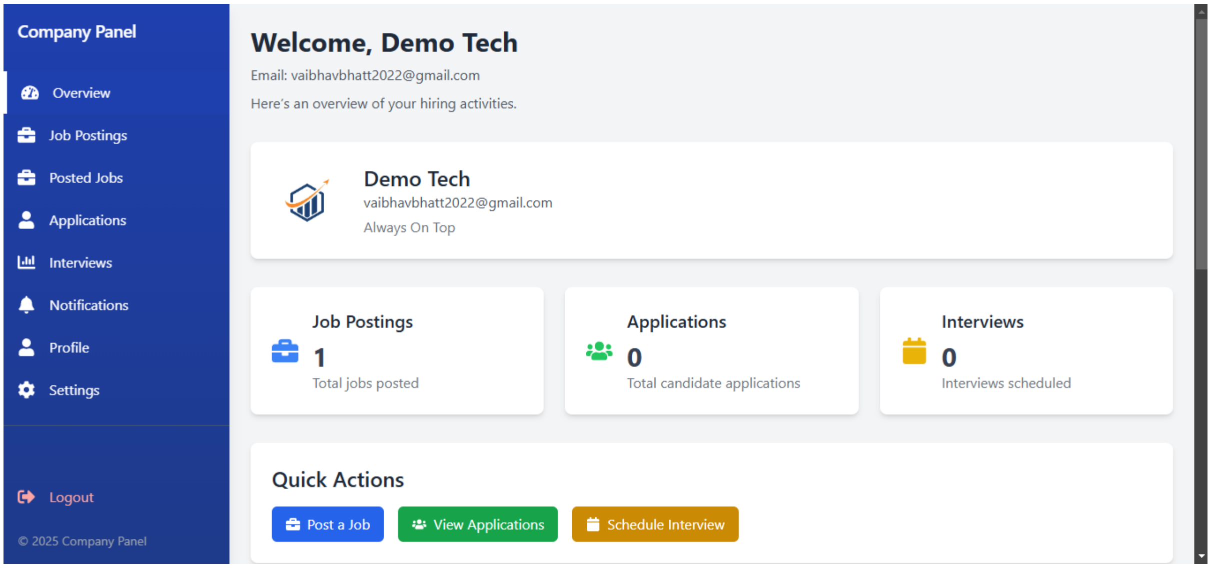
WorkSphere

Connecting talent with opportunity

Company Login

4.2.2.2 Company Login Page

Description	Employer/Company Login Page
Data From	Company Collection
Data To	
Critical Validations	Company Name Is Required Email Must Contain Specific Format Phone Number Is Required Company Address Is Required Password Must Match Specific criteria Password And Confirm Password Should Match



Welcome, Demo Tech

Email: vaibhavbhatt2022@gmail.com

Here's an overview of your hiring activities.

Demo Tech
vaibhavbhatt2022@gmail.com
Always On Top

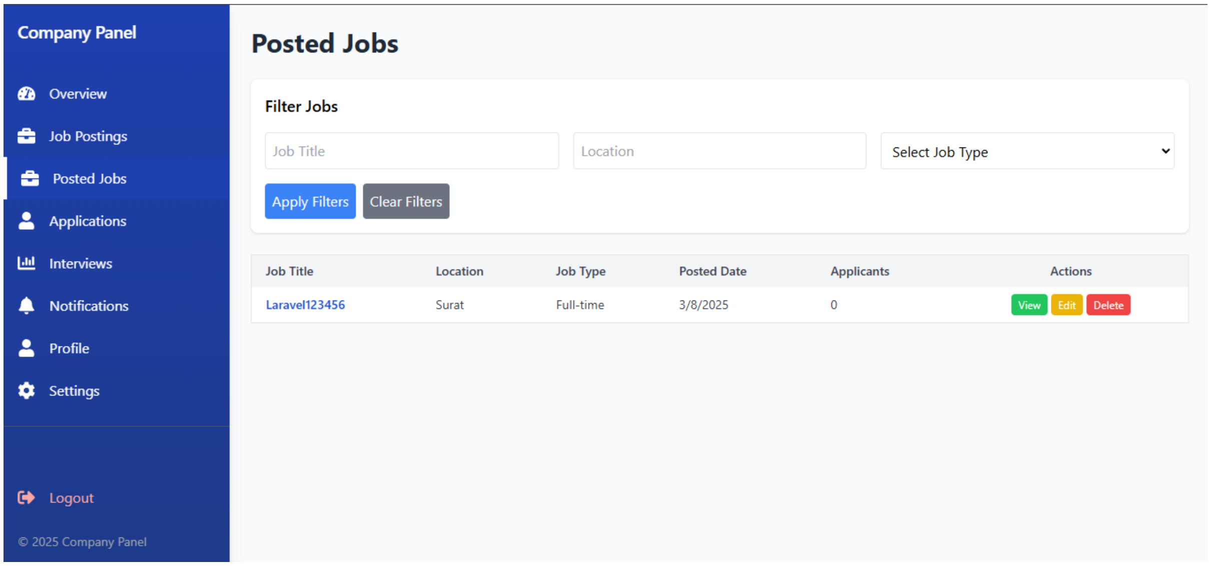
Job Postings
1
Total jobs posted

Applications
0
Total candidate applications

Interviews
0
Interviews scheduled

4.2.2.3 Company Dashboard Page

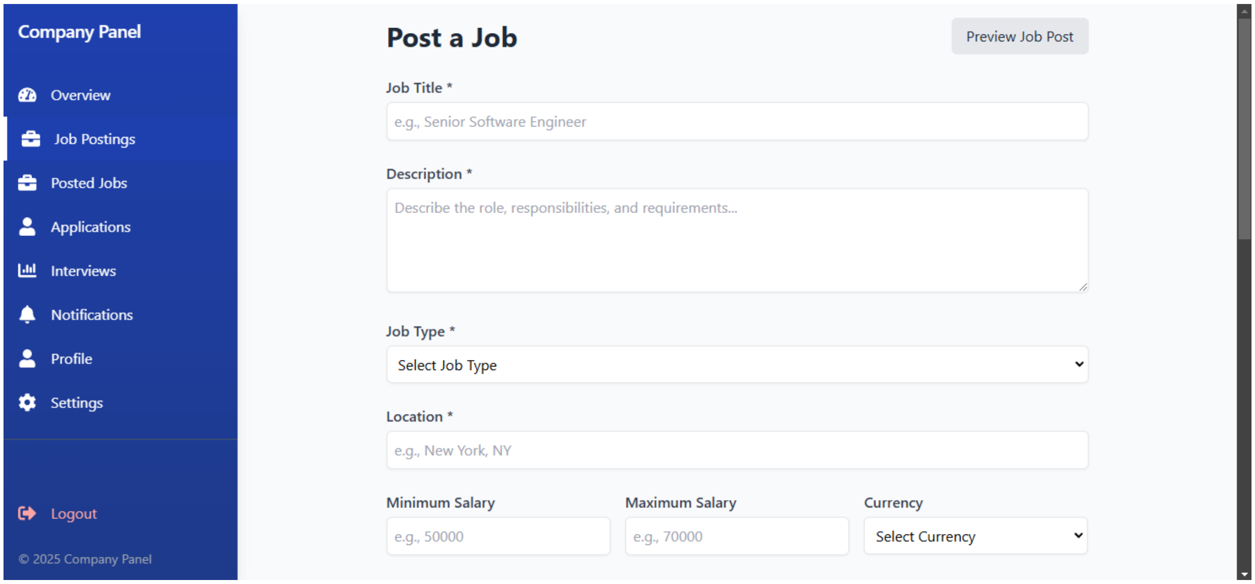
Description	Employer/Company Dashboard Page Display All the Main Statistical Data of jobs posting, applications, interviews etc.
Data From	Company Collection, Company Profile Collection, Jobs Collection
Data To	
Critical Validations	



Job Title	Location	Job Type	Posted Date	Applicants	Actions
Laravel123456	Surat	Full-time	3/8/2025	0	View Edit Delete

4.2.2.4 Company Posted Jobs Page

Description	Employer/Company Posted Jobs Page that displays all the jobs postings that posted by company with view, edit and delete features
Data From	Jobs Collection
Data To	Jobs Collection
Critical Validations	



Post a Job

Job Title *

e.g., Senior Software Engineer

Description *

Describe the role, responsibilities, and requirements...

Job Type *

Select Job Type

Location *

e.g., New York, NY

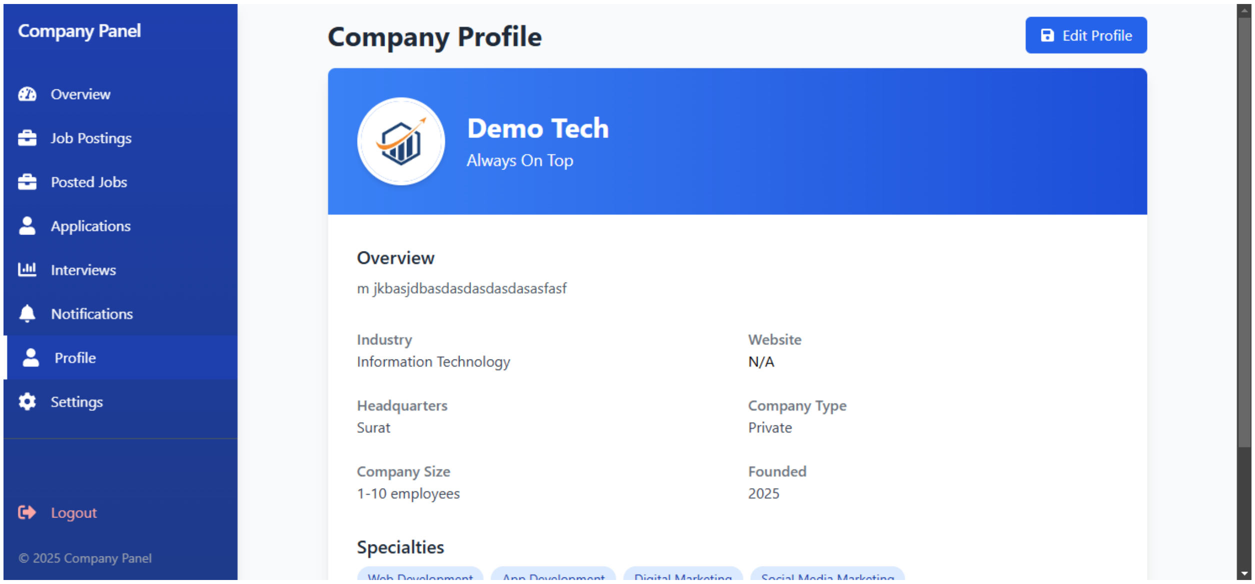
Minimum Salary: e.g., 50000

Maximum Salary: e.g., 70000

Currency: Select Currency

4.2.2.5 Company Job Posting Page

Description	Employer/Company Job Posting Page using this page company can add new job postings
Data From	
Data To	Jobs Collection
Critical Validations	Job Title Must Match Specific Characters Criteria Description Must Contain at least 20 characters Job type is Required Location Is Required Also Have to Specify Minimum and Maximum Salary and Option to Choose Currency



Company Profile

Demo Tech

Always On Top

Overview

m jkbasjdbasdasdasdasdasafsaf

Industry
Information Technology

Website
N/A

Headquarters
Surat

Company Type
Private

Company Size
1-10 employees

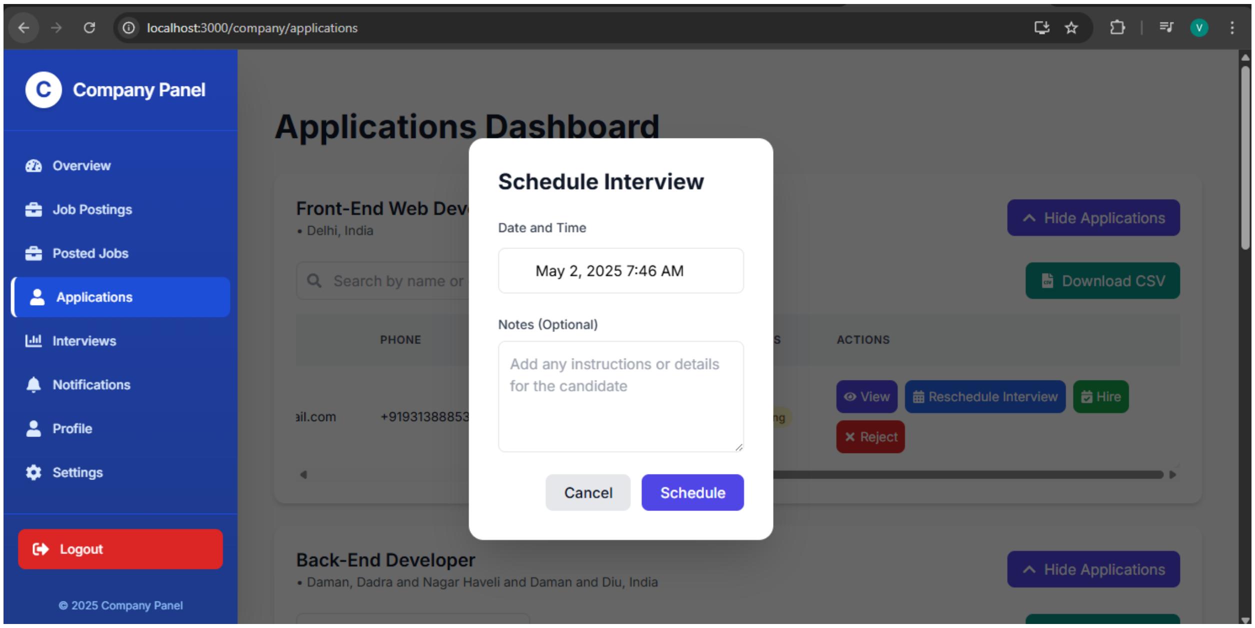
Founded
2025

Specialties

Web Development | App Development | Digital Marketing | Social Media Marketing

4.2.2.2 Company Profile Page

Description	Employer/Company Profile Page Company Can Edit Their Personal Details
Data From	Company Profile Collection
Data To	Company Profile Collection
Critical Validations	Major all fields are required



Applications Dashboard

Front-End Web Dev

• Delhi, India

ail.com +91931388853

Back-End Developer

• Daman, Dadra and Nagar Haveli and Daman and Diu, India

Schedule Interview

Date and Time

May 2, 2025 7:46 AM

Notes (Optional)

Add any instructions or details for the candidate

4.2.2.3 Company Interview Schedule Model

Description	Company Can Easily Schedule Interview With Applicant And System Autometically Send Main With Interview Details And Joining Link
Data From	
Data To	Interview Collection
Critical Validations	

5. Testing

5.1. Test cases

Test Scenario	Test Case ID	Test Case Name	Test Steps	Test Data	Expected Outcome	Actual Outcome	Result
User Registration	TC01	Valid registration	1. Go to "Sign Up" page 2. Enter valid Name, Email, Password 3. Click "Register"	Name="Suresh", Email="davogak340@firain.com", Password="Demo@123"	"Registration successful" message; redirected to login page	Registration Successful message; And Redirect to login page	pass
User Registration	TC02	Duplicate email registration	1. Go to "Sign Up" page 2. Enter Name, existing Email 3. Click "Register"	Name="Suresh", Email="davogak340@firain.com", Password="Demo@123"	Error "Email already registered"; stay on Sign Up page	User Already Exists Message;	fail
User Login	TC03	Successful login	1. Go to "Login" page 2. Enter valid Email, Password 3. Click "Login"	Email="davogak340@firain.com", Password="Demo@123"	User is logged in; lands on their dashboard	Login Successful And redirect to home page	pass
User Login	TC04	Login with wrong password	1. Go to "Login" page	Email="alice@example.	Error "Invalid email or	Incorrect Passwor	fail

			2. Enter valid Email, wrong Password 3. Click “Login”	com”, Password=“wrongpass”	password”; stay on Login page	d Message	
Job Posting (Employer)	TC05	Create new job post	1. Employer logs in 2. Navigate to “Post Job” 3. Fill Title, Description, Location, Salary, Deadline 4. Click “Submit”	Title=“Backend Dev”, Desc=“Node.js role”, Location=“Remote”, Salary=“80k”	“Job posted successfully”; job appears in “Manage Jobs”	Job posted Successful massage and redirect to posted jobs page	pass
Job Search (Candidate)	TC06	Search with filters	1. Candidate logs in 2. Go to “Search Jobs” 3. Enter Keyword, select Location, Salary range 4. Click “Search”	Keyword=“Dev”, Loc=“Remote”, SalaryMin=60000	Only jobs matching filters are displayed		
Job Application (Candidate)	TC07	Apply to job	1. Candidate performs a search 2. Click “Apply”	-	“Application submitted” message; entry visible under		

			on a job listing 3. Confirm on the dialog popup		“My Applications”		

6. References:

Books

- **Chodorow, Kristina.** *MongoDB: The Definitive Guide – Powerful and Scalable Data Storage.* O'Reilly Media, 2013.
  (A comprehensive guide covering MongoDB, data modeling, and best practices.)
- **Vasan Subramanian.** *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node.*
  Apress, 2019.
  (This book provides practical insights into building full-stack applications using the MERN stack.)
- **Hahn, Evan.** *Express in Action: Writing, building, and testing Node.js applications.* Manning Publications, 2016.
  (A detailed guide to building robust server-side applications with Express.js.)
- **Stefanov, Stoyan.** *React Up & Running: Building Web Applications.* O'Reilly Media, 2016.
  (An accessible introduction to React and building interactive web applications.)

Web

- **Indeed** – https://www.indeed.com
- **LinkedIn** – https://www.linkedin.com/jobs
- **Naukri.com** – https://www.naukri.com
- **Monster India** – https://www.monsterindia.com
- **CareerBuilder** – https://www.careerbuilder.com

7.Glossary

- **API (Application Programming Interface)**: A set of rules and protocols that allows the frontend and backend to communicate, enabling data exchange.
- **Applicant**: A user who applies for a job listing on the platform.
- **Authentication**: The process of verifying user credentials (email, password, or OAuth) to grant access to the system.
- **Authorization**: The process of determining what actions or resources a user can access based on their role (e.g., Applicant, Employer).
- **Backend**: The server-side part of the application, built using **Node.js**, **Express.js**, and **MongoDB**, which handles data processing, authentication, and API management.
- **Bearer Token**: A security token used in authentication headers to authorize API requests.
- **Benefits**: The perks or compensations provided by a company for a job position.
- **CORS (Cross-Origin Resource Sharing)**: A security feature that controls how frontend applications interact with backend APIs hosted on different domains.
- **Company Profile**: A company's information, including name, industry, website, description, and **uploaded logo**.
- **Company Authentication**: The process where a company logs in to manage job postings and view applicants.
- **CRUD Operations**: The four fundamental operations for data management—Create, Read, Update, and Delete.
- **Company Dashboard**: The admin panel where companies manage their job listings, applications, and profile.
- **Company Logo**: An uploaded image representing a company, displayed on job listings and company profiles.
- **Database (MongoDB)**: The NoSQL database used to store job postings, user profiles, applications, and company information.

- **Document Schema**: A predefined structure in **Mongoose** for storing data in MongoDB collections.
- **Employer**: A company or organization that posts job listings on the portal.
- **Encryption**: The process of securing sensitive data, such as passwords, using hashing algorithms like **bcrypt.js**.
- **Express.js**: A lightweight **Node.js** framework used for building the backend API.
- **Experience Level**: The level of experience required for a job, such as **Entry-Level**, **Mid-Level**, **Senior**, or **Executive**.
- **Frontend**: The client-side part of the application, developed using **React.js**, which interacts with the backend API.
- **Fetch API**: A method used in JavaScript to request data from the backend API.
- **Form Validation**: A technique used to ensure that user inputs (e.g., job application, registration) meet the required format before submission.
- **Helmet.js**: A security middleware in **Express.js** that helps protect the backend from common web vulnerabilities.
- **HTTP Requests**: Methods like **GET, POST, PUT, DELETE** used by the frontend to communicate with the backend.
- **Job Application**: The process where applicants submit their details and resumes for a job listing.
- **Job Categories**: The classification of jobs based on industry or specialization.
- **Job Controller**: A backend file responsible for handling job-related API requests.
- **Job Details Page**: A frontend page that displays full job information, including title, description, salary, skills, and **company logo**.
- **Job Listing**: A job post that includes details like **title, location, salary, requirements, and job type**.
- **Job Type**: The category of employment, such as **Full-time, Part-time, Contract, Internship, Temporary**.
- **Middleware**: Functions that process API requests before they reach controllers, such as authentication and error handling.

- **MongoDB Atlas**: A cloud-hosted MongoDB database used for storing and managing job-related data.
- **Mongoose.js**: An ODM (Object Data Modeling) library for **MongoDB**, used to define schemas and interact with the database.
- **Node.js**: A runtime environment that allows JavaScript to be used for backend development.
- **Navigation (React Router)**: A method for handling page routing in the frontend, allowing users to navigate between different views.
- **Passport.js**: A middleware used for implementing authentication strategies (e.g., JWT, Google OAuth) in the backend.
- **Protected Route**: A frontend route that is only accessible to authenticated users.
- **Profile Section**: A user profile page that includes personal details, resumes, and applied jobs.
- **React.js**: A JavaScript library for building dynamic and interactive user interfaces.
- **RESTful API**: A backend API that follows REST (Representational State Transfer) principles for handling data.
- **Rate Limiting**: A security feature that restricts the number of API requests to prevent abuse.
- **Redux**: A state management tool in React that stores and manages application-wide data.
- **Salary Range**: The minimum and maximum salary offered for a job.
- **Session Management**: The process of maintaining user authentication across multiple requests.
- **Server.js**: The main backend entry point that initializes the **Express.js** server and connects to **MongoDB**.
- **Skills Section**: A field that lists the skills required for a job.
- **State Management**: The way React manages application states, including job data and user authentication.
- **Tailwind CSS**: A utility-first CSS framework used to style the frontend.

- **Token-Based Authentication**: A method where users authenticate using **JWT (JSON Web Token)**.
- **User Authentication**: The process where applicants log in to apply for jobs and track applications.
- **User Middleware**: A backend function that ensures users are authenticated before accessing protected resources.
- **Uploaded Files**: Includes **resumes, company logos, and job-related documents** stored on the server.
- **Validation (Joi.js)**: A library used in **Express.js** to validate user inputs, ensuring data integrity.
- **View Jobs Page**: A frontend page that displays all available jobs in a professional **grid layout**.
- **WebSocket (Feature)**: A technology that allows real-time communication (e.g., live job notifications) between the server and frontend.
- **Work Experience**: The past employment history listed in **applicant resumes**.