An Interactive Job And Internship Platform For Technical Education Department

Submitted for partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

by

Shaik Safileen	-	**21BQ1A42G3**
P.V.S.S.Saikumar	-	**21BQ1A42E9**
Alisetti Saiharsha	-	**21BQ1A42J0**
Peeka Akash	-	**21BQ1A42J2**

Under the guidance of

Dr. S.L.V.V.D.Sarma

Associate Professor

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELLIGENCE & MACHINE LEARNING
VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTU Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified
NAMBUR (V), PEDAKAKANI (M), GUNTUR – 522 508
Tel no: 0863-2118036, url: www.vvitguntur.com

April 2025

CERTIFICATE

This is to certify that this System Report is the bonafide work of **Ms. Shaik Safileen, Mr. Puvvada Venkata Siva Satya Saikumar, Mr. Alisetti Saiharsha, Mr. Peeka Akash**, bearing Reg. No. **21BQ1A42G3, 21BQ1A42E9, 21BQ1A42J0, 21BQ1A42J2** respectively who had carried out the System entitled "**An interactive job and internship platform for technical education department**" under our supervision.

Project Guide:
(Dr.S.L.V.V.D.Sarma, Associate Professor)

Head Of the Department:
(Dr.K.Suresh babu, Professor)

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

We, Ms. Shaik Safileen, Mr. Puvvada Venkata Siva Satya Saikumar, Mr. Alisetti Saiharsha, Mr. Peeka Akash hereby declare that the System Report entitled **"An interactive job and internship platform for technical education department"** done by us under the guidance Of Dr.S.L.V.V.D.Sarma, Associate Professor, CSE -Artificial Intelligence & Machine Learning at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science Engineering - Artificial Intelligence & Machine Learning. The results embodied in this report have not been submitted to any other University for the award of any degree.

DATE :

PLACE : Nambur

SIGNATURE OF THE CANDIDATES

Shaik Safileen,

P.V.S.S.Saikuma,

Alisetti Saiharsha,

Peeka Akash

ACKNOWLEDGEMENT

We take this opportunity to express my deepest gratitude and appreciation to all those people who made this System work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and helped me towards the successful completion of this System work.

First and foremost, we express my deep gratitude to **Sri. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express my sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express my sincere gratitude to **Dr. K. Suresh Babu**, Professor & HOD, Computer Science Engineering – Artificial Intelligence & Machine Learning Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express my sincere gratefulness to our Guide **Dr.S.L.V.V.D.Sarma**, Associate Professor, CSE-Artificial Intelligence & Machine Learning for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this System.

We would like to express our sincere heartfelt thanks to our Project Coordinator **Ms.Deepika**, Assistant Professor, CSE-Artificial Intelligence & Machine Learning for her valuable advices, motivating suggestions, moral support, help and coordination among us in successful completion of this System.

We would like to take this opportunity to express my thanks to the **Teaching and Non- Teaching** Staff in the Department of Computer Science Engineering -Artificial Intelligence and Machine Learning, VVIT for their invaluable help and support.

Name (s) of Students

Shaik Safileen

P.V.S.S.Saikumar

Alisetti Saiharsha

Peeka Akash

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

ML	Machine Learning
MERN	MongoDB, Express, React JS, Node JS
KNN	K-Nearest Neighbors
AI	Artificial Intelligence
HTTP	Hyper Text Transfer Protocol

ABSTRACT

This System presents a comprehensive job portal system developed using the MERN stack, comprising modules for Admin, Jobseeker/Student, and Recruiter/Company. The Admin module allows the administrator to manage the portal effectively by viewing registered users, companies, and job postings, and logging out. The User module enables job seekers and students to register, log in, search for jobs based on their skills, receive recommendations, apply for jobs, participate in live or virtual exams, and track the status of their applications. The Recruiter module is designed for companies to register, log in, add job requirements, and create exams (MCQs) for applicants, and view user profiles and exam details. Recruiters can select or reject candidates based on exam performance and other criteria. The System aims to provide a seamless job application and recruitment experience, integrating key functionalities like job recommendations, online exams, and application status tracking, ensuring an efficient and user-friendly platform for both job seekers and recruiters.

Keywords: Admin, Jobseeker/Student, and Recruiter/Company

CHAPTER 1

INTRODUCTION

In today's competitive job market, both job seekers and recruiters face numerous challenges in navigating the hiring process efficiently. To address these challenges, this System presents a job portal system built using the MERN (Mongo DB, Express.js, React.js, Node.js) stack, designed to streamline the recruitment process for both job seekers and companies. The platform incorporates three key modules: Admin, Jobseeker/Student, and Recruiter/Company, each tailored to meet the specific needs of the respective users.

The Admin module allows for the effective management of the system, including oversight of user registrations, job postings, and recruiter activities. Job seekers, using the User module, can create profiles, search for jobs based on their skills, apply for positions, and track the status of their applications. Additionally, the platform offers job recommendations and supports live and virtual exams, enabling candidates to showcase their abilities through structured assessments.

On the recruiter side, companies can post job requirements, create and manage exams for applicants, and review candidate profiles and exam results. Recruiters are empowered to make informed hiring decisions based on candidate performance in exams and other relevant criteria.

By integrating advanced features like online exams, application status tracking, and personalized job recommendations, this System aims to enhance the job application and recruitment experience, offering a seamless, efficient, and user-friendly platform for all participants.

1.1 MOTIVATION:

The job and internship market has become highly competitive, with thousands of applicants vying for limited positions. AI and Machine Learning (AIML) have revolutionized the recruitment industry by enabling automated resume screening, skill matching, and candidate evaluation. These technologies help companies streamline their hiring process, ensuring that only the most relevant candidates proceed to the next stages.

The importance of AIML in recruitment lies in its ability to:

1. Reduce hiring time by automating resume screening.
2. Improve fairness by eliminating human bias in shortlisting.
3. Provide personalized job recommendations based on a candidate’s skills and experience.

1.2. PROBLEM STATEMENT:

The current recruitment process is often inefficient, time-consuming, and lacks transparency, resulting in poor job matching and delayed communication between job seekers and recruiters. There is a need for an integrated solution that provides personalized recommendations, online assessments, and real- time updates to enhance the efficiency and user experience of the recruitment process.

1.3. OBJECTIVE OF THE SYSTEM:

The objective of this System is to design and develop a job portal that facilitates seamless interaction between job seekers, recruiters, and admins. The portal aims to enhance the recruitment process by providing features such as personalized job recommendations, online assessments, and application tracking, thereby improving efficiency and user experience for all stakeholders

1.4. SCOPE OF THE SYSTEM:

1. Admin Scope:

User Management: Manage employers and job seekers by approving, suspending, or removing accounts.

Job Listing Oversight: Monitor job postings to ensure compliance with platform policies.

Exam Management: Create, update, and monitor virtual exams for candidate assessments.

Reports & Analytics: Generate reports on job postings, employer activity, and candidate engagement.

Platform Maintenance: Oversee website security, performance, and feature updates.

Complaint & Support Handling: Address user grievances and resolve disputes between employers and job seekers.

2. Employer Scope:

Profile Management: Create and manage a company profile, including branding and hiring details.

Job Posting: Post job vacancies with descriptions, requirements, and application deadlines.

Candidate Search: Search for potential candidates based on skills, experience, and other criteria.

Application Tracking: View and manage applications with real-time updates on candidate progress.

Virtual Exams: Conduct skill-based online assessments for screening candidates.

Communication: Directly contact shortlisted candidates via chat or email.

3. **Job Seeker (Student/User) Scope:**

Profile Creation: Build a professional profile with personal details, skills, resume, and work experience.
Job Search & Application: Search for jobs based on skills, industry, and location, and apply directly.
Skill-Based Job Recommendations: Get AI-driven job recommendations based on profile and skills.
Exam Participation: Take virtual assessments to enhance job prospects.
Application Tracking: Monitor job application statuses and employer responses.
Notifications & Updates: Receive alerts about new job postings, exams, and employer responses.

1.5. METHODOLOGY OVERVIEW:

The System follows a structured approach using **CRISP-DM (Cross Industry Standard Process for Data Mining)**, ensuring an efficient job and internship platform. The six key phases of CRISP-DM are applied as follows

1. **Data Collection:**

Purpose: The platform relies on structured and unstructured data collection for effective job matching.
Process:
Collect **resumes** from students applying for jobs and internships.
Gather **job descriptions** from recruiters/employers.
Retrieve **candidate profiles**, including skills, experience, and qualifications.
Store **historical hiring data** to refine the job recommendation system.
Collect **recruiter feedback** to enhance job matching accuracy.
Module Involved:
Resume Upload & Parsing Module
Job Posting Module

2. **Data Preprocessing**

Purpose: Prepare raw data for machine learning by converting unstructured resumes into structured data.
Process:
Natural Language Processing (NLP) techniques are used to extract key information (skills, education, work experience) from resumes.
Text Cleaning & Standardization: Remove irrelevant words, correct inconsistencies, and handle missing data.

Skill Extraction & Categorization: Standardize different skill representations (e.g., "Java Developer" vs. "Java Programmer").
Job Description Processing: Remove duplicate or outdated job postings to maintain data integrity.

Technologies Used:
NLP Models: Named Entity Recognition (NER), TF-IDF, and Word Embeddings (Word2Vec, BERT).
MongoDB: Used to store structured candidate and job data.
Module Involved:
Resume Processing & Skill Extraction Module
Job Description Processing Module

3. Modeling

Purpose: Build a **machine learning-based** recommendation system for accurate job matching.
Process:
Resume Scoring Algorithms:
Use **TF-IDF** (Term Frequency-Inverse Document Frequency) for keyword-based matching.
Implement **BERT (Bidirectional Encoder Representations from Transformers)** for context-aware resume matching.
Apply **Word2Vec** for similarity analysis between job descriptions and candidate skills.
Recommendation Engine:
Uses machine learning to suggest the best job matches based on candidate profiles.
Factors include skill relevance, experience, location preferences, and recruiter feedback.

Technologies Used:
Machine Learning Frameworks: Scikit-learn, TensorFlow, Hugging Face Transformers.
Database: MongoDB for storing trained models and candidate-job matching data.
Module Involved:
Job Matching Module
Resume Scoring & Ranking Module

4. Evaluation

Purpose: Assess model performance to ensure effective job matching.
Process:
Measure ML model accuracy using:
Precision: How many recommended candidates are relevant.

Recall: How many relevant candidates are recommended.

F1-score: A balance between precision and recall.

Feedback Loop: Recruiter feedback is used to fine-tune the model over time.

Technologies Used:

Evaluation metrics in **Scikit-learn**

Data visualization using **Matplotlib** and **Seaborn.**

Module Involved:

Model Evaluation Module

Feedback Integration Module

5. Deployment

Purpose: Make the platform available to users with real-time processing.

Process:

Backend: Deployed using **Node.js and Express.js** on cloud infrastructure.

Database: MongoDB Atlas for scalable cloud storage.

ML Model Deployment:

Hosted as an API on **AWS Lambda** for serverless execution.

Uses **AWS S3** for storing resumes and job-related files.

ML models deployed on **AWS SageMaker** for scalable inference.

Frontend: Built with **React.js**, hosted on **AWS Amplify**.

Authentication & Security:

User authentication using **JWT (JSON Web Tokens)**.

Role-based access control (Admin, Employer, Student).

Technologies Used:

AWS Services: Lambda, S3, SageMaker, Amplify, DynamoDB

CI/CD Pipelines: Implemented using GitHub Actions & AWS CodePipeline

Module Involved:

User Authentication Module

Cloud Deployment & API Integration Module

System Accuracy & Matching Performance

Initial Accuracy Benchmark:

TF-IDF Matching: ~75%

BERT-based Resume Scoring: ~85%

Hybrid TF-IDF + BERT Model: ~90%

Improvement Over Time:

Continuous learning with recruiter feedback improves accuracy.

Model adaptation to new job trends and skills enhances recommendations.

1.6. ORGANIZATION OF THE REPORT:

The report is structured into key chapters: Introduction (overview of AI in recruitment and System objectives), Background (importance of AI in HR tech and real-world applications), Methodology (CRISP-DM approach covering data collection, preprocessing, modeling, evaluation, and deployment), System Design & Implementation (MERN stack integration, AI models for resume scoring and job matching), Results & Evaluation (performance analysis and user feedback), and Conclusion & Future Scope (key findings, limitations, and enhancements like AI-driven career guidance).

CHAPTER 2
LITERATURE SURVEY

2.1 Related Work:

Several studies have highlighted the evolving landscape of job search mechanisms through the integration of technology, social media, and ERP systems in educational and employment sectors. Gasparėnienė et al. (2021) explored the growing significance of social media platforms such as LinkedIn in facilitating job searches. Their study emphasizes how these platforms provide broader reach, enhanced networking opportunities, and personalized job recommendations. However, they also note challenges such as data privacy concerns and the lack of feedback for candidates after unsuccessful applications.

Similarly, Hamdane et al. (2022) proposed a big data-based architecture tailored to bridge the gap between graduates and recruiters in Moroccan universities. By leveraging academic records and user preferences, the system uses analytics to recommend suitable job opportunities. The study acknowledges the efficiency of this approach but also outlines drawbacks such as high computational costs and the complexity of securing large data sets in real-time.

In the context of educational institutions, Kendle et al. (2022) presented an Enterprise Resource Planning (ERP) system designed to streamline the training and placement process. Their model allows students to upload resumes, track applications, and enables recruiters to post jobs and manage candidates efficiently. Nonetheless, the system faces issues in scaling, handling high volumes of data, and maintaining seamless interactions during peak times.

Further literature enriches this discussion. Joshchukwuere et al. (2023) provided a comprehensive review of social media's role in recruitment. They noted that platforms like Facebook, Twitter, and LinkedIn are increasingly used to profile candidates and accelerate the hiring process. Despite these benefits, the study warns of potential ethical concerns, including biased candidate screening and overdependence on social media presence.

Additionally, a report by the International Labour Organization (2020) delved into the potential of big data to anticipate and match future skill needs in the labor market. Using case studies from countries such as Austria and India, the report illustrates how AI and analytics can be employed to align job market demand with educational outputs. However, it also brings attention to challenges like data fragmentation, governance, and accessibility to quality data.

The World Bank (2022) extended this line of research by analyzing how job matching platforms have adopted AI and big data to enhance the efficiency of job recommendations. These platforms, both public and private, are increasingly relying on skill-based profiles to generate accurate job matches. Nevertheless, the report points out risks such as algorithmic bias, unequal access, and the need for more personalized solutions for niche job roles.

In the domain of ERP systems in higher education, a study titled *An Effect of ERP on Placement Management System* (2019) evaluated how ERP solutions help Training and Placement Officers (TPOs) manage student data, schedule interviews, and coordinate with recruiters. The study found improved efficiency in the placement process but also identified limitations related to system scalability and user training. Similarly, another study from 2020 explored ERP implementation in higher education institutions, emphasizing strategic project management to reduce costs and overcome technical barriers. However, the researchers observed that resistance to change and integration challenges with existing systems remained critical issues during deployment.

2.2. EXISTING SOLUTIONS AND LIMITATIONS:

Several AI-driven recruitment platforms exist, each offering unique solutions for job matching, resume screening, and candidate evaluation. Below is a comparison of some existing approaches and their limitations:

1. Traditional Job Portals (e.g., LinkedIn, Indeed, Naukri) Effectiveness:

Provide a large database of job listings and candidate profiles.

Limitations:

Relies on keyword matching, often leading to irrelevant recommendations.

Lacks AI-driven personalized job matching based on skills and experience.

2. AI-Powered Resume Screening (e.g., HireVue, Pymetrics, X0PA AI)

Effectiveness:

Uses Natural Language Processing (NLP) to analyze resumes.

Ranks candidates based on skills, experience, and job description matching.

Limitations:

May struggle with non-standard resume formats and design-heavy resumes.

Some AI models inherit biases from training data, leading to **unfair filtering**

2.3. Gap Analysis

Despite advancements in AI-driven recruitment, existing solutions still have **major shortcomings** that affect both recruiters and job seekers. **Job Quest**

aims to bridge these gaps by offering a more efficient and AI-driven hiring experience.

Identified Gaps in Existing Solutions:

1. Ineffective Resume Matching:

   Many platforms rely on keyword-based matching, which often leads to irrelevant recommendations.

   Job Quest Solution: Uses NLP and AI-driven skill analysis for precise resume-job matching.

2. Lack of Personalized Job Recommendations:

   Current job portals suggest jobs based on past searches, not a candidate's actual skills or career goals.

   Job Quest Solution: Uses context-aware ML models to recommend jobs based on skills, experience, and aspirations.

2.4. Relevance of the System

Job Quest builds on previous research in AI-driven recruitment, leveraging advancements in Natural Language Processing (NLP), Machine Learning (ML), and Bias-Reduction Algorithms. Key Inspirations & Research Foundations:

1. Datasets Used in AI Hiring Research:

   LinkedIn Job Posts Dataset: Analyzing job descriptions and required skills.

   Resume Parsing Datasets: Used to train NLP models for resume screening.

2. Industry-Inspired Features:

   Inspired by LinkedIn, Indeed, and AI-driven hiring tools, but improved with better AI- driven career guidance.

   AI-powered resume scoring, interview tips, and job tracking to enhance user experience

CHAPTER 3
SYSTEM ANALYSIS

This chapter defines the requirements, feasibility, and overall system overview of Job Quest, an AI- powered recruitment platform.

1.1 Requirement Analysis Functional Requirements:

The system must:

Match candidates to jobs based on skills, experience, and preferences. Score and rank resumes using NLP-based algorithms. Provide AI-driven job recommendations based on user profiles. Offer an interactive chatbot for answering job-related queries and application tracking. Allow recruiters to filter and shortlist candidates efficiently.Provide career insights such as skill gaps and upskilling recommendations.

Admin Module:

Admin can log in securely to the system.

Admin can view and manage the list of registered users. Admin can view and manage the list of registered companies. Admin can view and monitor active job postings.

Admin has the ability to remove inappropriate content (job postings or user profiles).

Admin can monitor the integrity of data across the system (e.g., ensure correct user-job relationships).

Admin can generate reports on the number of registered users, companies, and jobs.

User Module (Jobseeker/Student):

Users can register by providing personal and professional details.

Users can log in with valid credentials.

Users can create and edit their profiles (including adding qualifications, skills, and experience).

Users can search for jobs based on their skills, experience, and location preferences.

Users receive skill-based job recommendations.

Users can apply for job postings.

Users can participate in virtual exams (such as MCQs) as part of the recruitment process.

Users can view the status of their job applications.

Users can track their progress in real time (application status, interview, and offer status).

Recruiter Module (Company):

Recruiters can register and log in with a company profile.

Recruiters can post new job openings and update job details.

Recruiters can create online assessments, such as MCQs, for applicants.

Recruiters can view user profiles (candidates who have applied).

Recruiters can evaluate the results of virtual exams.

Recruiters can accept or reject candidates based on exam results and profiles.

Recruiters can communicate with applicants regarding interview schedules or offer letters.

Non-Functional Requirements:

1. **Security:**

   All user data (personal information, application history, etc.) must be encrypted and securely stored. Role-based access control to ensure only authorized users (admins, recruiters, job seekers) access specific modules. Secure authentication mechanisms (e.g., two-factor authentication) for users and recruiters.

2. **Performance:**

   The system should support concurrent access by multiple users without performance degradation.The job search functionality should return results within 2 seconds for an optimal user experience.

3. **Usability:**

   The system should have a user-friendly interface, allowing job seekers, recruiters, and admins to easily navigate the portal.

4. **Scalability:**

   The system should be scalable to handle a growing number of users, job postings, and companies as the platform expands.

5. **Reliability:**

   The system should ensure 99.9% uptime, minimizing downtime for users, recruiters, and admins.

6. **Data Integrity:**

   All data input and output should be validated to prevent inconsistencies and ensure the integrity of information across the system.

7. **Audit and Monitoring:**

   The system should maintain audit logs for all significant actions (such as job postings, user registration, etc.) for compliance and review purposes.

8. **Maintainability:**

   The system should be designed to allow easy updates and maintenance without disrupting services.

9. **Backup and Recovery:**

   The system should implement regular data backups and have a disaster recovery plan to restore data in case of a system failure.

10. **Compliance:**

    The system must comply with local and international data protection regulations (e.g., GDPR) to ensure privacy and security of personal data.

HARDWARE REQUIREMENTS:

- **Processor - I3/Intel Processor**
- RAM - 4GB (min)
- Hard Disk - 160GB

SOFTWARE SYSTEM CONFIGURATION:

- Operating System : Windows 7/8/10
- Serverside Script : Express js
- Programming Language : TypeScript
- IDE/Workbench : VS Code
- Database : Mongodb
- Clint Side : React js

3.2. Feasibility Study Technical Feasibility:

The system is built using MERN Stack (MongoDB, Express.js, React, Node.js) with AI models using Python, TensorFlow, and NLP libraries (BERT, Word2Vec).

Cloud-based deployment (AWS, Firebase) ensures high availability and performance.

Economic Feasibility:

Open-source tools and cloud-based services reduce costs. AI automation minimizes manual recruitment efforts, making it cost-effective for companies.

Operational Feasibility:

User-friendly interface with simple navigation for both job seekers and recruiters. Automated AI suggestions make job searching and hiring more efficient. Chatbot integration provides instant assistance, reducing user effort.

3.3. Proposed System Overview

Job Quest enhances traditional job portals by integrating AI-driven resume scoring, skill-based job recommendations, and automated hiring processes. Unlike existing platforms that rely on keyword matching, Job Quest utilizes NLP and Machine Learning to provide accurate, bias-free, and personalized job matches. It also offers career insights, real-time job tracking, and an AI chatbot, making the recruitment process more efficient for both job seekers and employers.

Advantages of Job Quest Over Traditional Job Portals

Bias-Free Hiring: AI-driven assessment ensures fair and unbiased candidate evaluations.

Higher Accuracy: Machine learning models improve job-candidate matching accuracy over time.

Time Efficiency: Automated processes reduce hiring timelines for employers and application effort for job seekers.

Personalized Experience: Tailored recommendations, insights, and guidance make job hunting more effective and user-friendly. By integrating AI into every aspect of the recruitment process, Job Quest transforms job searching and hiring into a seamless, efficient, and rewarding experience for all stakeholders

CHAPTER 4
SYSTEM DESIGN

Introduction of Input Design:

In an information system, input refers to the raw data that is processed to produce meaningful output. The process of input design is vital, as the quality of system input determines the quality of the system's output.

During input design, developers must carefully consider the various input devices that may be used, such as personal computers (PCs), Magnetic Ink Character Recognition (MICR), and Optical Mark Recognition (OMR). A well-designed input form or screen should effectively serve specific purposes like storing, recording, and retrieving information. It should also ensure the accurate and proper completion of data entry tasks.

Additionally, input forms should be easy to fill out and straightforward, focusing on user attention, consistency, and simplicity. These design goals are achieved through a sound understanding of basic design principles, such as identifying the inputs needed by the system and analyzing how end users respond to different elements on the forms and screens.

Objectives for Input Design:

The main objectives of input design are to ensure smooth and efficient data entry and input procedures. It aims to reduce the overall volume of input data by streamlining processes and formats. Another key objective is to design source documents or other methods to capture data effectively. This includes the development of input data records, data entry screens, and user interface screens. Additionally, input design should incorporate validation checks and implement strong input controls to ensure data accuracy, prevent errors, and enhance overall system reliability.

Output Design:

The design of system output is considered one of the most important tasks in system development. During the output design process, developers determine what types of output are needed by the end users and how best to control and present that output. This also includes the creation of prototype report layouts to visualize the final outputs. Output design plays a critical role in how information is delivered to users and used for decision-making.

Objectives of Output Design:

The objectives of output design are focused on ensuring that the output fulfills its intended purpose without producing unnecessary information. It should meet the requirements of the end users by providing the correct information in the right quantity and format. The output must be directed to the appropriate individuals and be accessible at the right time to support timely and effective decision-making. A well-designed output system enhances user satisfaction and contributes to the overall efficiency of the information system.

4.1. SYSTEM ARCHITECTURE:

The **Job Quest** system is designed using a client-server architecture that integrates AI-powered recruitment functionalities. It combines the MERN (MongoDB, Express.js, React.js, Node.js) stack for backend support with advanced AI models to enhance features such as resume scoring and personalized job recommendations. This architecture ensures high scalability, operational efficiency, and a seamless user experience for both job seekers and recruiters.

1. Frontend (React.js)

The frontend of the Job Quest system is developed using **React.js**, which provides an interactive and user-friendly interface tailored for both job seekers and recruiters. It incorporates dynamic rendering and effective state management to allow smooth and intuitive navigation throughout the platform. The frontend communicates with the backend services through RESTful APIs, ensuring efficient and secure data transfer. Designed with responsiveness in mind, the UI adapts seamlessly across various devices, enhancing accessibility and usability. Moreover, the system implements robust **authentication mechanisms** and **role-based access control**, ensuring that users only access features and data relevant to their specific roles within the platform.

Functionality:

Navigation Handling

The Home component features a navigation bar that includes Job Quest branding, offering a visually consistent entry experience. It facilitates smooth navigation across the platform by providing distinct access options tailored to different user roles.

Dynamic Component Rendering

Using React's useState hook, the component keeps track of the currently active view—such as SignIn, AdminSignIn, or CompanySignIn. The handleButtonClick function is responsible for dynamically switching between these components, allowing the interface to respond in real time to user selections without requiring a full page reload.

User Role-Based Access

The system ensures tailored access for each type of user. **Job Seekers** can sign in through the SignIn component and are given the ability to explore job listings via the ViewJobs component. **Admins** are directed to a dedicated admin login interface (AdminSignIn), while **Companies** can authenticate through the CompanySignIn component, each receiving

access to functionalities relevant to their roles. This structured access improves security and ensures users interact with the content and features most pertinent to them.

2. Backend (Node.js, Express.js)

The **Backend** of the Job Quest platform is built using **Node.js** and **Express.js**, which together handle API requests and efficiently serve data to the frontend. The backend also integrates advanced **AI models** for tasks like resume parsing and job matching, helping to streamline the recruitment process. It implements essential business logic for a variety of platform functions, including user authentication, job posting, and the processing of job applications. To ensure secure access and interactions, the backend manages session handling and incorporates security features such as **JWT authentication** and **OAuth integration**. Furthermore, the backend provides an **admin panel** that enables administrators to monitor and manage the system, ensuring smooth operations and effective oversight.

Validates User Input

The route begins by using the registerValidation middleware to validate the user input, ensuring that the data provided is correct, complete, and follows the expected format. This step is crucial to maintain data integrity and prevent any malformed or incorrect data from entering the system.

Checks for Duplicate Emails

Before proceeding with the creation of a new user, the route checks whether the provided email is already registered in the system. This step prevents the creation of multiple accounts with the same email, ensuring that each user has a unique identity on the platform.

Creates a New User

Once the email is verified and no duplicates are found, the route creates a new user in the **MongoDB** database. Essential user details, including their **full name**, **email**, and **password**, are securely stored for future authentication and access to the platform.

Handles Errors Gracefully

The route ensures proper error handling by returning appropriate **HTTP status codes** for different scenarios. If the registration is successful, a **201**-status code is returned. If there are issues such as invalid input or missing data, a **400** status code is sent. For server-related issues, a **500** status code is provided, ensuring that both users and developers are informed of the request's outcome in a clear and effective manner.

3. **Database (MongoDB):**

The **Database** for the Job Quest platform is powered by **MongoDB**, which stores critical data such as user profiles, job listings, applications, and AI-generated results. Leveraging a **NoSQL document structure**, MongoDB allows for efficient and flexible data storage and retrieval, enabling the platform to handle a variety of data types and large volumes of information. To further enhance performance, the database implements **indexing** and **query optimization**, ensuring faster access to key data such as job listings and user information. Additionally, MongoDB provides robust **backup and recovery mechanisms**, which are essential for maintaining data integrity and ensuring that all platform data remains secure and recoverable in case of system failures or other issues.

4. **AI Models (Python, NLP, ML)**

The AI models integrated into the platform utilize Python, Natural Language Processing (NLP), and Machine Learning (ML) techniques to perform a variety of functions aimed at enhancing the recruitment process. The models are responsible for resume parsing, extracting keywords and relevant information from resumes to facilitate better matching between candidates and job listings. The machine learning algorithms analyze both resumes and job descriptions to provide accurate job matching. Additionally, the AI models power a recommendation engine, offering personalized job suggestions to users based on their qualifications, interests, and past behavior.

5. **Cloud Deployment (AWS/Firebase)**

The Job Quest platform is deployed using **cloud services** to ensure scalability, high availability, and robust performance. The backend services are hosted on **AWS EC2**, providing a reliable infrastructure for handling user requests and managing resources. For storage, **AWS S3** is utilized to securely store files such as resumes and job postings. **Firebase** is employed to manage real-time notifications and synchronize the database across devices, ensuring users receive timely updates. To optimize performance, a **Content Delivery Network (CDN)** is implemented, ensuring fast loading times for the frontend. Additionally, the platform integrates with **CI/CD pipelines**, enabling automated deployments and seamless updates for continuous improvement and maintenance of the system.

4.2. Block Diagram:

A block diagram would visually represent the major components of the system, highlighting the flow of data from input to processing. The **input** components include user profiles, resumes, and job descriptions, which serve as the primary data sources for the system. Once the data is collected, the system moves to the **processing** stage, where advanced AI techniques are used. This includes **AI-based resume parsing**, which extracts key information from resumes, **job matching** to align candidates with suitable job opportunities, and **chatbot interaction** to assist users in real-time. The diagram would clearly show how these components interact and how data flows through the system, providing a clear and structured overview of the platform's functionality.

Actors/Entities:
The system involves several key actors interacting within its environment.
User: Job seekers who leverage the platform to discover and apply for relevant employment opportunities.
Recruiter/Company: Organizations that utilize the system to post job openings and manage their hiring processes.
Admin: System administrators, overseeing the platform and its data to ensure smooth operation and functionality.

Data Base(MongoDB): This central repository securely stores and manages all system data, including user details, job postings, exam records, and other necessary information.

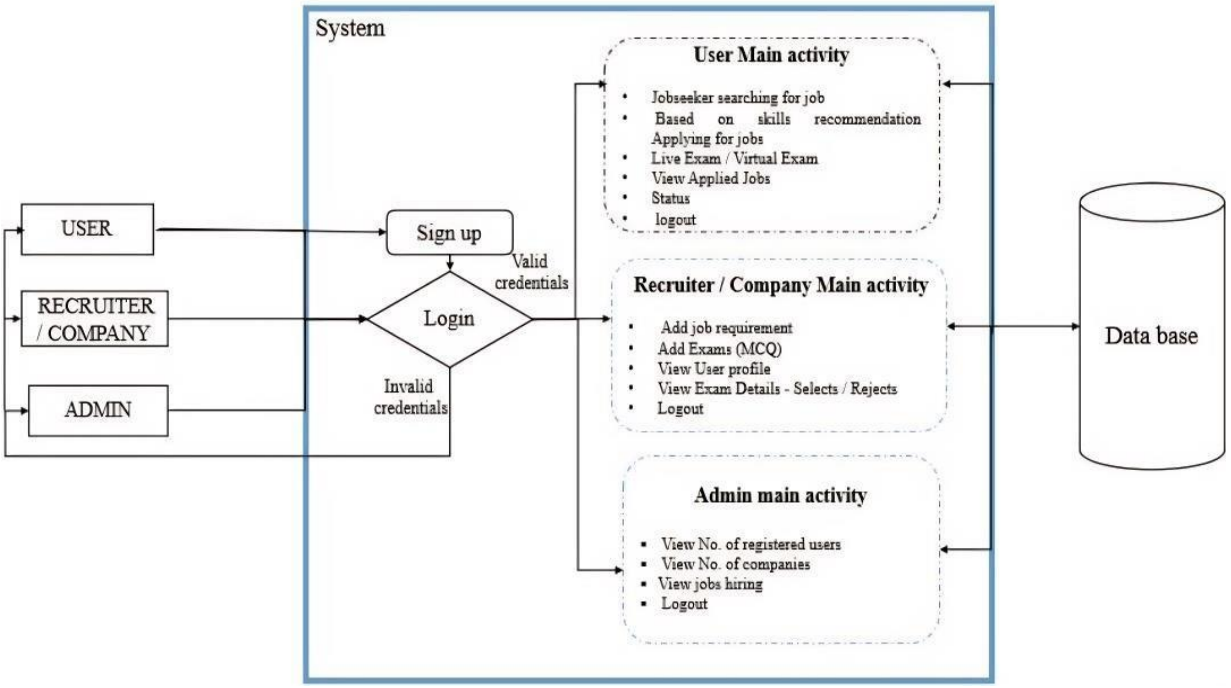

Figure 4.1: Block Diagram

4.3. Data Flow Diagrams (DFD):

Level 0 DFD (Context Diagram):

Job Quest is represented as a single, encompassing system designed to facilitate the job search and recruitment process. This system acts as a central hub, interacting with several key external entities.

Job Seekers (Users):

These individuals utilize Job Quest to construct and upload their professional resumes, conduct targeted searches for suitable job openings, and benefit from AI-driven personalized recommendations based on their skills and experience. They also manage their application processes and receive updates through the platform.

Recruiters:

Organizations and hiring managers leverage Job Quest to publish job listings, specifying required qualifications and responsibilities. They employ the system's filtering tools to efficiently sift through potential candidates, track the progress of applications, and streamline the hiring process.

Admin: They manage the system, including user accounts, system settings, and overall system maintenance.

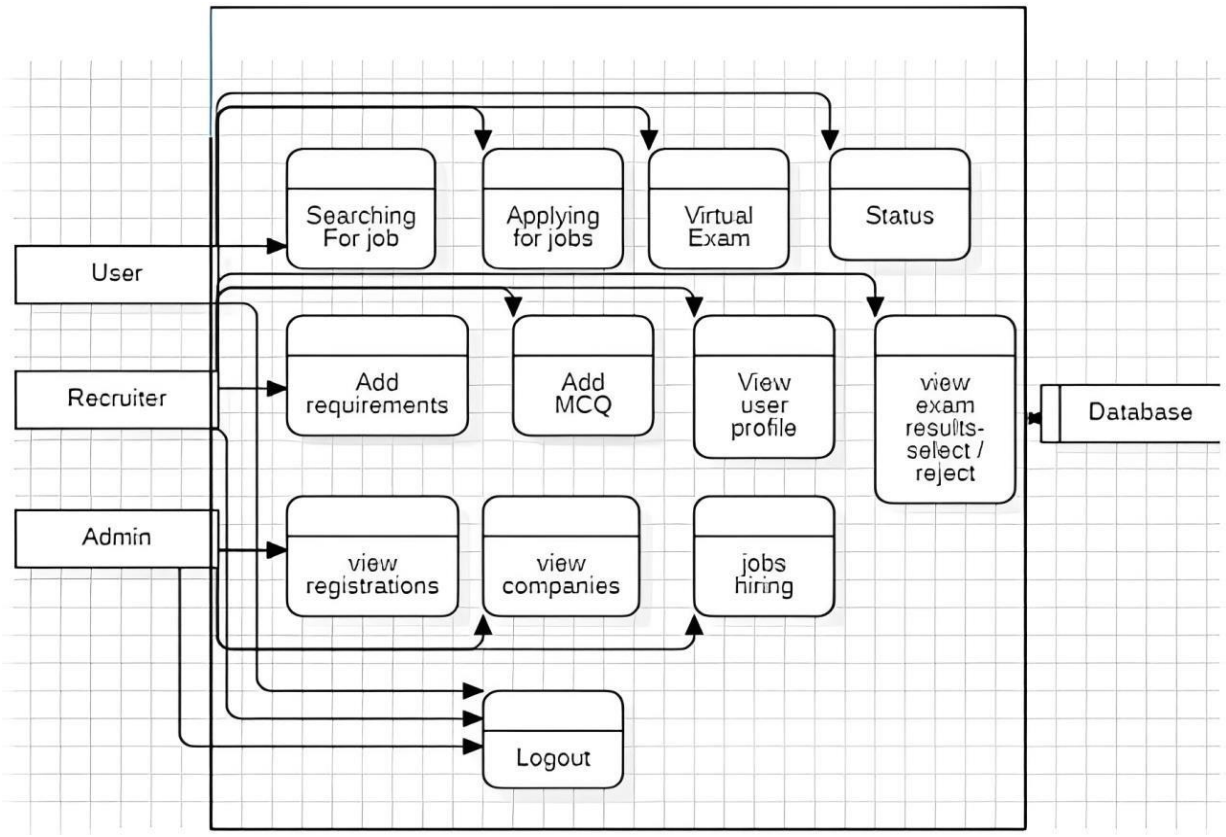

Figure 4.2: Level 0 DFD (Context Diagram)

Level 1 DFD

This diagram refines the Job Quest system by breaking it down into distinct sub-processes, revealing the internal workings of the platform.

Resume Processing Module

This module is responsible for analyzing and extracting key information from uploaded resumes. It identifies and categorizes skills, experience, and educational qualifications, subsequently ranking resumes based on their relevance to potential job openings. This allows for efficient candidate screening and initial assessment.

Job Matching Module

This core component of the system utilizes sophisticated algorithms to match candidate profiles with suitable job postings. It considers skills, experience, location preferences, and other relevant factors to provide personalized recommendations, facilitating a more effective job search for users.

Recruitment Analytics

This module provides recruiters with valuable insights into their hiring process. It tracks key metrics such as application rates, candidate response times, and interview outcomes, enabling data-driven decision-making and optimizing recruitment strategies. This module helps recruiters identify trends, improve efficiency, and ultimately hire the best candidates.

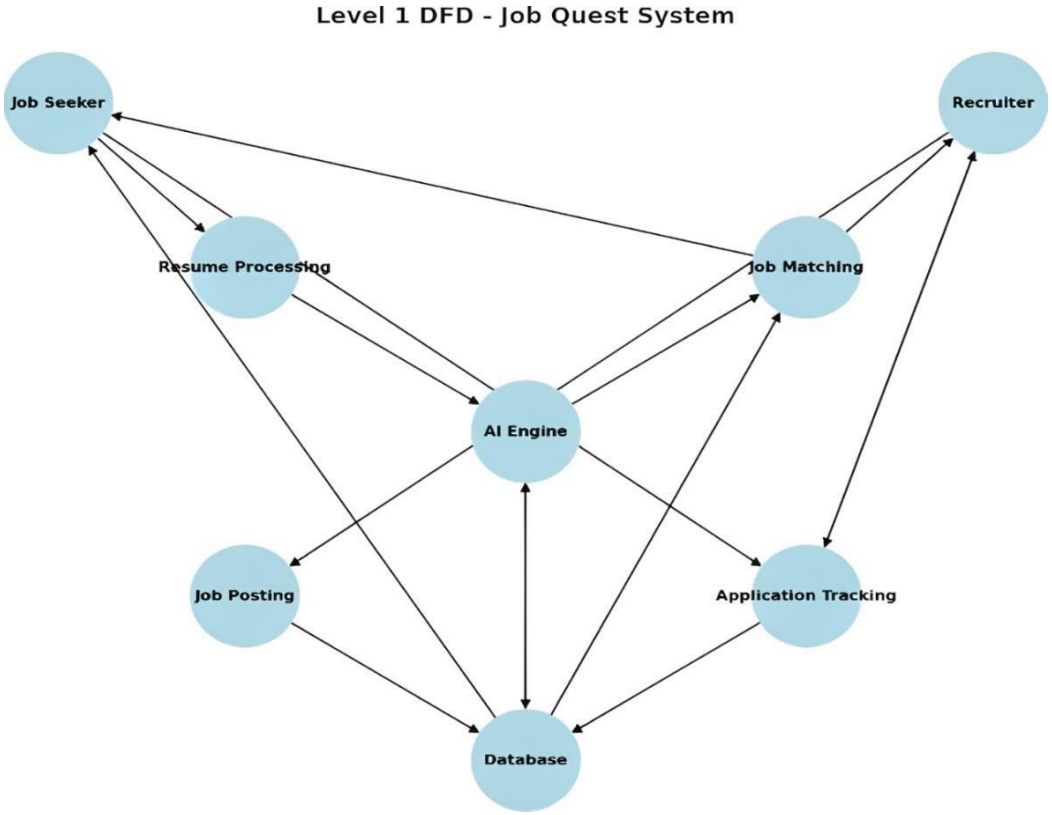

Figure 4.3: Level 1 DF

4.4. UML Diagrams

4.4.1 Use Case Diagram

Use case diagrams are used in software and systems engineering to visually represent the interactions between users (actors) and the system. They illustrate the functional requirements of the system from the user's perspective.

Components:

Actors: Represent external entities (users, other systems) that interact with the system. They are depicted as stick figures.

Admin: Manages the system.

User: Represents job seekers.

Recruiter (Company): Represents organizations hiring.

Use Cases: Represent specific goals or tasks that actors perform with the system. They are depicted as ovals.

Some of the usecases for this system are **Login, Signup, Logout, Jobsearch, Apply for Job, View User profiles.**

Relationships: Show the connections between actors and use cases, and between use cases themselves. They are depicted as lines.

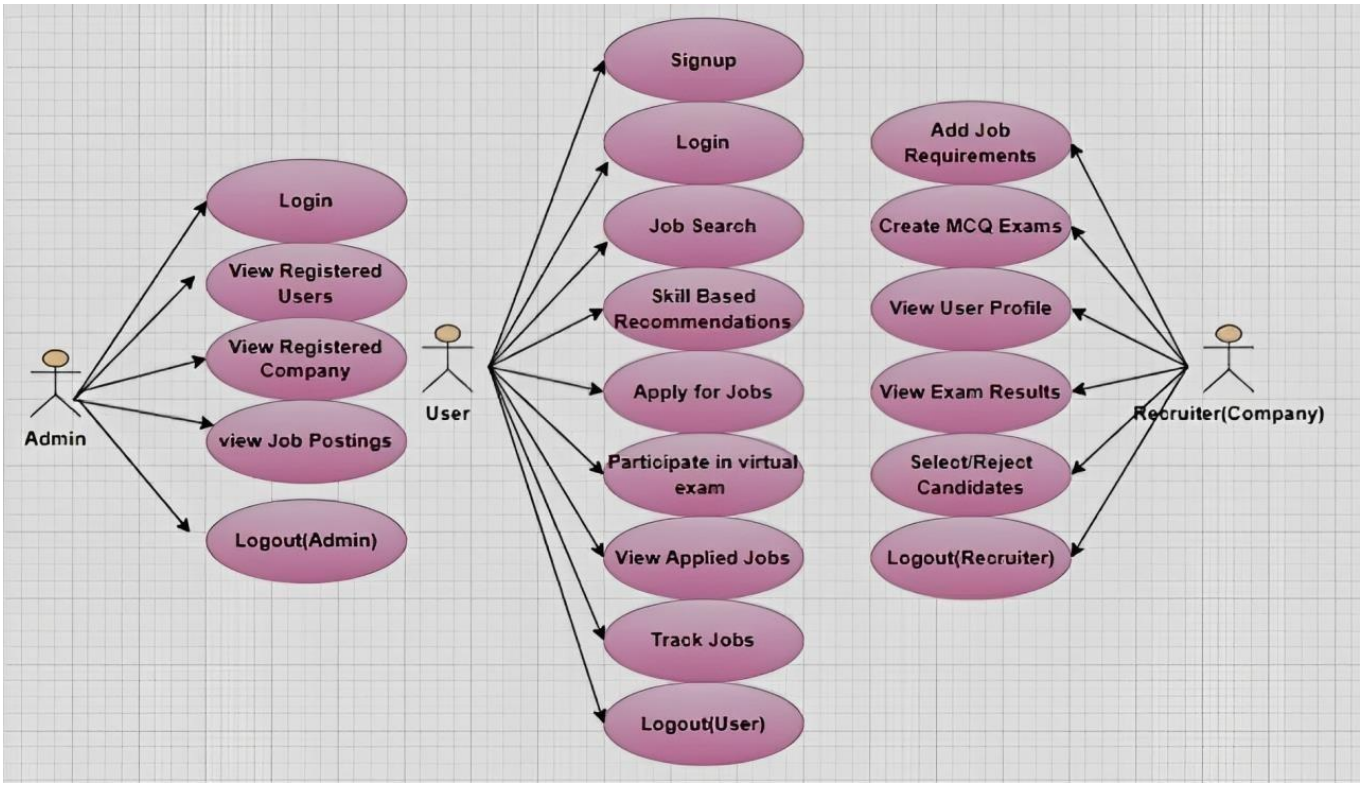

Figure 4.4: Use Case Diagram

4.4.2 Class Diagram

Class diagrams are a type of static structure diagram in UML (Unified Modeling Language) that describe the structure of a system by showing the system's classes, their attributes, operations (methods), and the relationships between objects.

Components:

Classes:

Represent blueprints for creating objects. They are depicted as rectangles divided into three sections: class name, attributes, and operations.

User: Represents job seekers.

Recruiter: Represents organizations hiring.

Admin: Represents system administrators.

Job: Represents job postings.

Candidate: Represents job applicants.

Assessment: Represents online assessments.

ApplicationStatus: Represents the status of a job application.

JobRequirement: Represents the requirements for a job.

ExamResult: Represents the results of an exam.

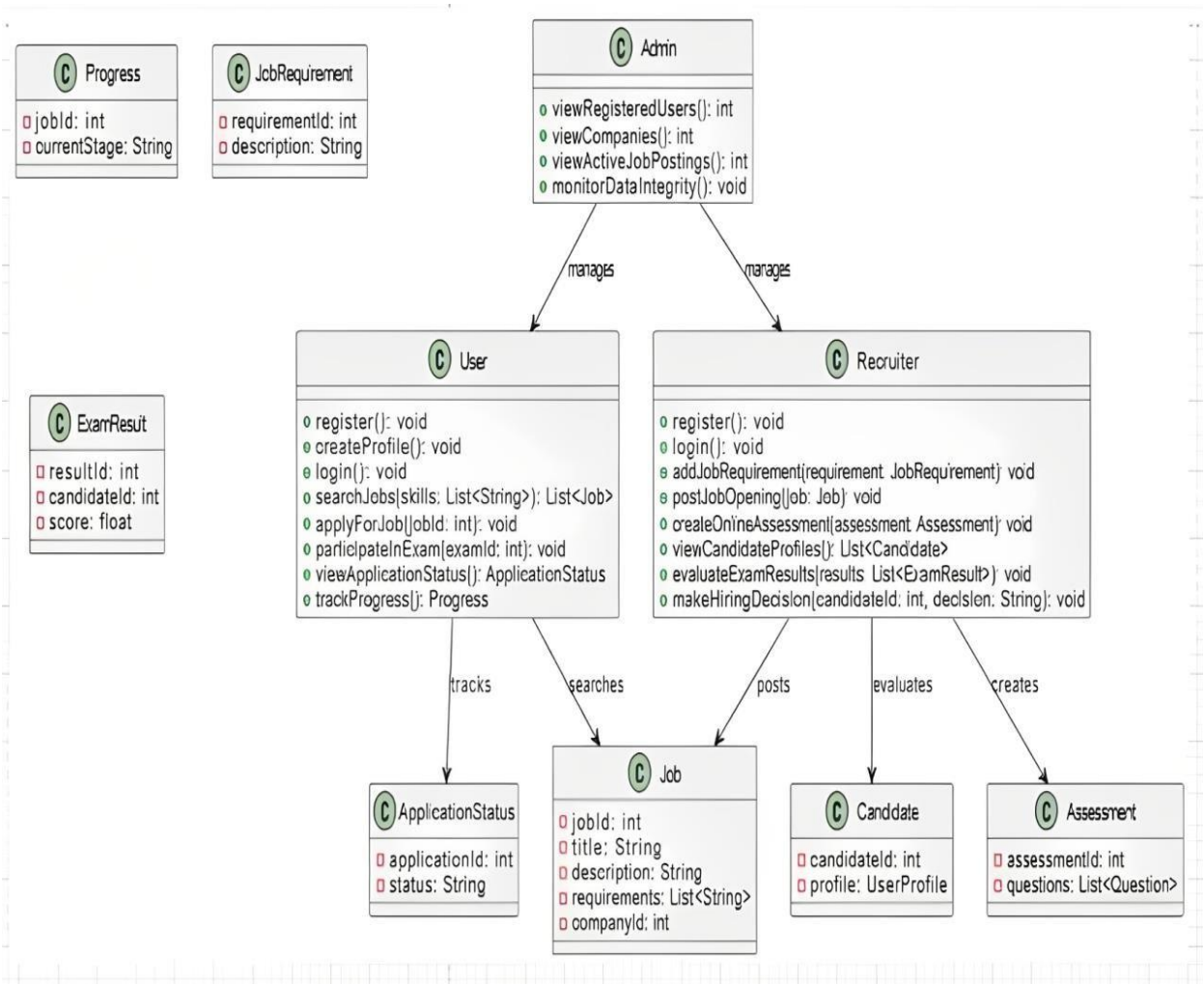

Figure 4.5: Class Diagram

4.4.3 Activity Diagram

Activity diagrams are a type of behavioural diagram in UML (Unified Modeling Language) that describe the flow of activities within a system. They show the sequence of actions, decisions, and parallel flows involved in a process.

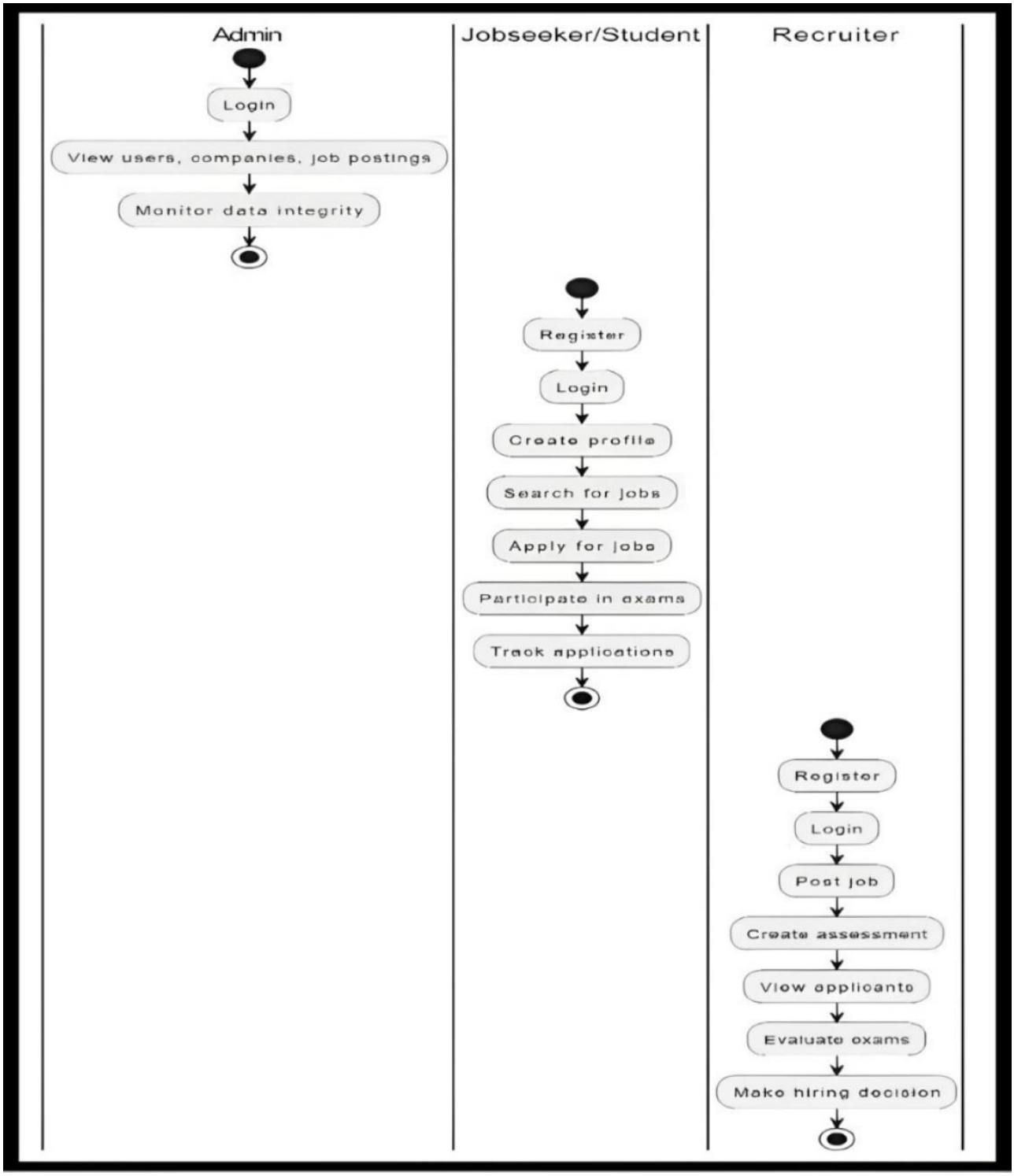

Figure 4.6: Activity Diagram

4.4.4 Sequence Diagram

The sequence diagram illustrates the step-by-step interaction between Job Seekers, Recruiters, the System, and the AI Engine in key workflows: **Job Seeker** uploads a resume and gets recommendations:

Logs in → Uploads resume → AI extracts skills & scores it → System suggests jobs.

Recruiter posts a job & shortlists candidates:

Logs in → Posts job → AI matches candidates → System ranks resumes → Recruiter shortlists.

Application process:

Job seeker applies → System updates database → Recruiter reviews → System notifies applicants.

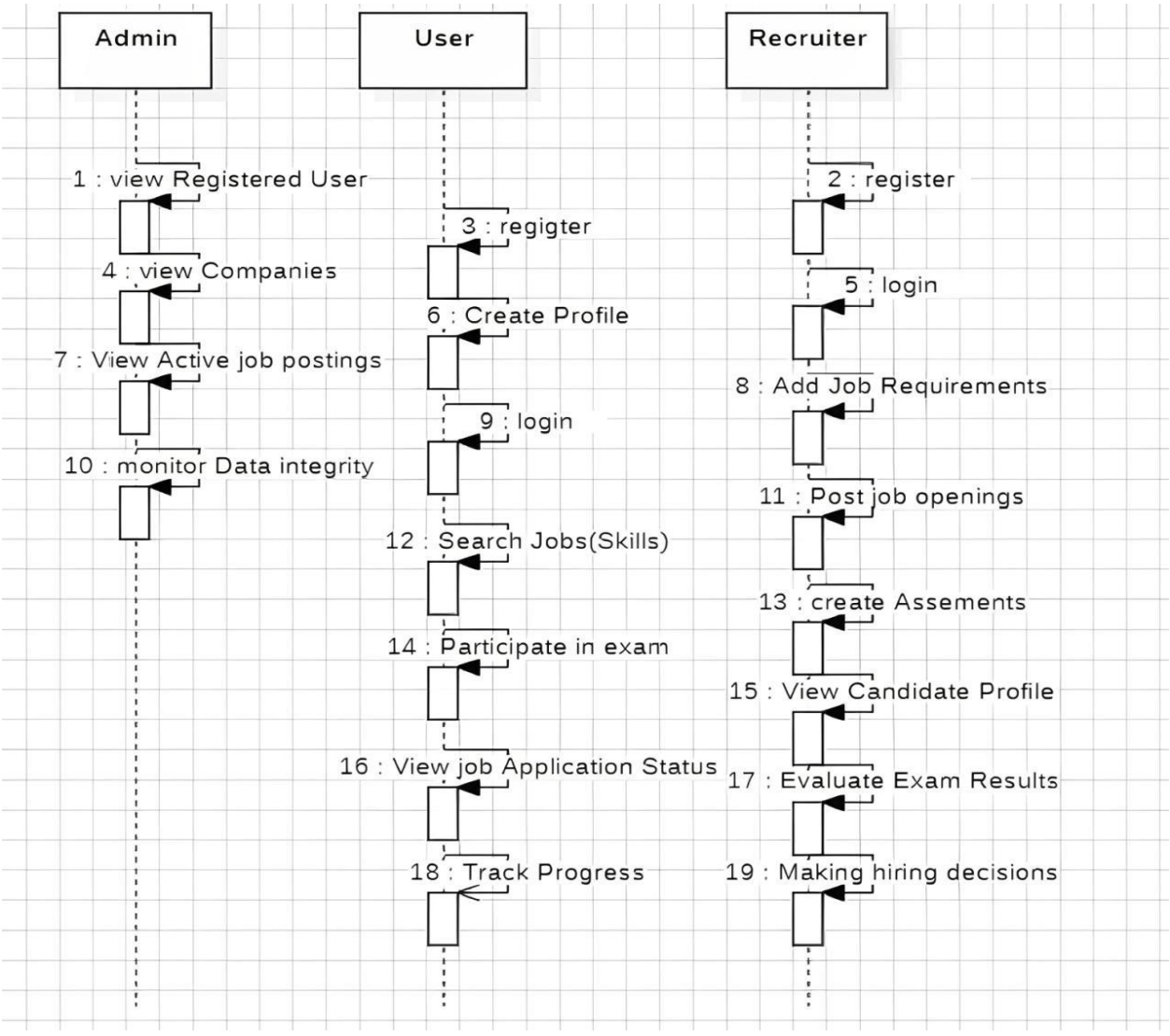

Figure 4.7: Sequency Diagram

4.5. DATABASE DESIGN:

The **database design** ensures efficient **data storage, retrieval, and management** for job seekers, recruiters, and AI-driven job matching.

4.5.1 Entity-Relationship (ER) Diagram:

The ER diagram includes key entities and their relationships:

Entities (Rectangles): Represent real-world objects or concepts that are relevant to the system. **Admin, User, Recruiter.**

Relationships (Diamonds): The diamonds represent relationships between entities: **Manage System, Applied Jobs**: This seems to be a relationship or a process that connects the User and Admin entities. **Posts Jobs, Registers:** This relationship connects the Recruiter and User entities.

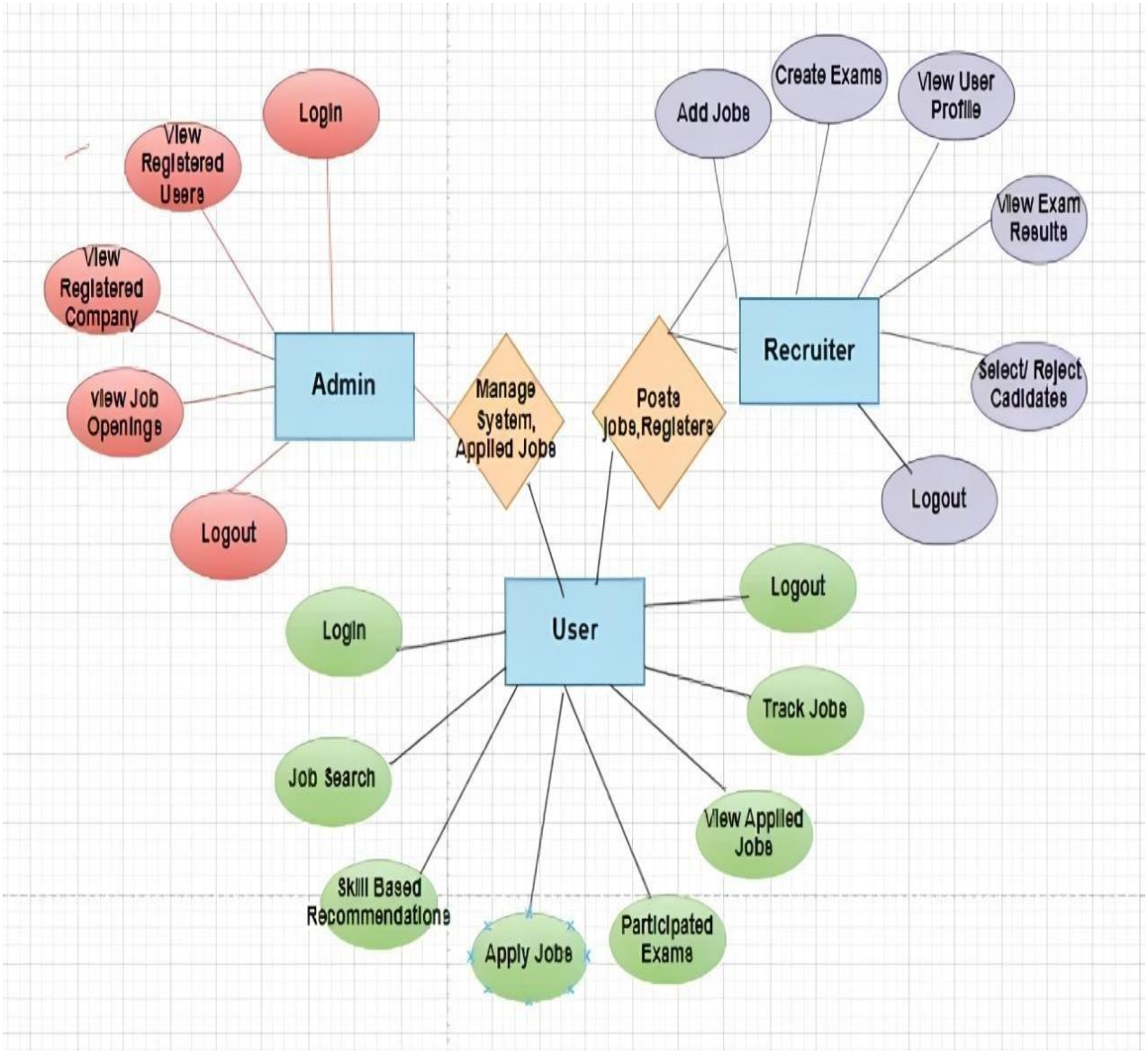

Figure 4.8: ER Diagram

CHAPTER 5
IMPLEMENTATION

5.1 PROGRAMMING LANGUAGES AND TECHNOLOGIES USED

Backend: Node.js (Express.js)

- Handles API requests and responses.
- Implements authentication, job matching, and resume scoring logic.

Frontend: React.js

- Provides an interactive UI for Job Seekers and Recruiters.
- Implements features like job searching, resume upload, and application tracking.

Database: MongoDB (Atlas)

- Stores user profiles, resumes, job postings, and applications.
- Enables efficient querying for job recommendations and candidate ranking.

AI Engine: Python (NLP, Machine Learning)

- Extracts skills and experience from resumes.
- Matches resumes with job descriptions using TF-IDF and Cosine Similarity.

Authentication: JWT (JSON Web Token)

- Ensures secure login and session management.

API Testing: Postman

- Used to test and debug APIs before deployment

5.2 DEVELOPMENT TOOLS AND ENVIRONMENT:

Code Editor and Integrated Development Environment (IDE) VS Code

(Visual Studio Code)

Purpose: Primary code editor for both backend and frontend development.

- Supports JavaScript, Node.js, React.js, and Python.
- Has extensions for debugging, linting, and Git integration.
- Allows seamless API testing with Thunder Client

API Testing and Debugging Tools

Postman

Purpose: Used for API testing, debugging, and documentation.

Why Used?

- Tests GET, POST, PUT, DELETE requests.
- Allows authentication testing (JWT Tokens).

Database Management Tools

MongoDB Atlas (Cloud Database)

Purpose: Stores all user data, job postings, resumes, and applications.

Why Used?

- NoSQL database – handles unstructured resume data efficiently.
- Cloud-based – accessible from any location.
- Scalable – handles large job postings and applications seamlessly.

5.3. MODULE WISE IMPLEMENTATION DETAILS

Admin Module:

The Admin module provides the administrator with a centralized interface to manage the job portal effectively. Admins can view the number of registered users, companies, and active job postings. They also have the authority to monitor and maintain data integrity across the system, ensuring smooth and efficient operations.

User Module (Jobseeker/Student):

The User module enables job seekers and students to register on the platform, create profiles, and log in to search for jobs that match their skills. The system provides skill-based job recommendations to enhance job search efficiency. Users can also apply for jobs, participate in virtual exams, view the status of their applications, and track their progress in real time.

Recruiter Module (Company):

The Recruiter module allows companies to register and log in to the portal. Recruiters can add job requirements, post new job openings, and create online assessments (MCQs) for applicants. They can also view candidate profiles, evaluate exam results, and make informed hiring decisions by selecting or rejecting applicants based on their performance and qualifications.

5.4 ALGORITHMS AND LOGIC USED

The k-Nearest Neighbors (kNN) algorithm is used to match jobs (jobs to candidates) based on skill similarity.

1. Convert skills into vectors (e.g., using one-hot encoding).
2. Compare skill vectors of users and jobs using distance metrics like cosine similarity.
3. Find the top-k closest matches:

Recommend jobs to users that match their skills.

Recommend candidates to recruiters based on job requirements.

CHAPTER 6
TESTING AND RESULTS

Feasibility Study

The feasibility of the System is analysed in this phase and business proposal is put forth with a very general plan for the System and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. Economic feasibility
2. Technical feasibility
3. Social feasibility

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

6.1 TESTING METHODOLOGIES

System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.1.3 Acceptance Testing

User Acceptance Testing is a critical phase of any System and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.1.4 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted. Invalid Input: identified classes of invalid input must be rejected. Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.2 TEST CASES:

Admin Module:

Invalid Admin Email Id:

Test Case:

Login with Invalid Admin Password

Precondition:

- Admin account exists in the system.
- The admin login page is accessible.

Test Steps:

1. Navigate to the admin login page.
2. Enter a valid admin username.
3. Enter an incorrect password.
4. Click on the Login button.

Expected Result:

- The system should display an error message like "Invalid username or password."
- The admin should not be logged in.
- The login page should remain visible

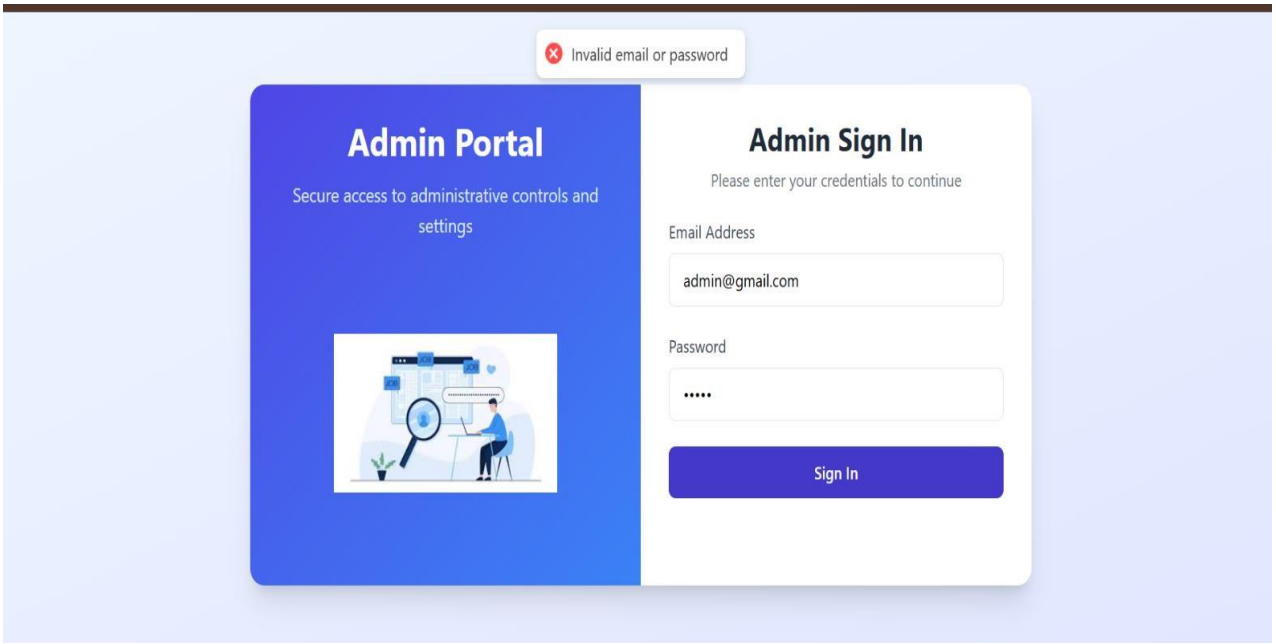

Figure 6.1: Invalid Admin Sign in

Invalid OTP:

Test Case:

Entering Incorrect OTP in Admin Login

Precondition:

- Admin has entered valid credentials and triggered OTP authentication.
- OTP has been sent to the registered contact (email/phone).

Test Steps:

1. Navigate to the admin login page.
2. Enter a valid admin username and password.
3. Click on the Login button.
4. On the OTP verification page, enter an incorrect OTP.
5. Click on the Verify button.

Expected Result:

- The system should display an error message like "An error during OTP verification."
- The admin should not be logged in.
- The system may allow limited attempts before blocking further attempts.

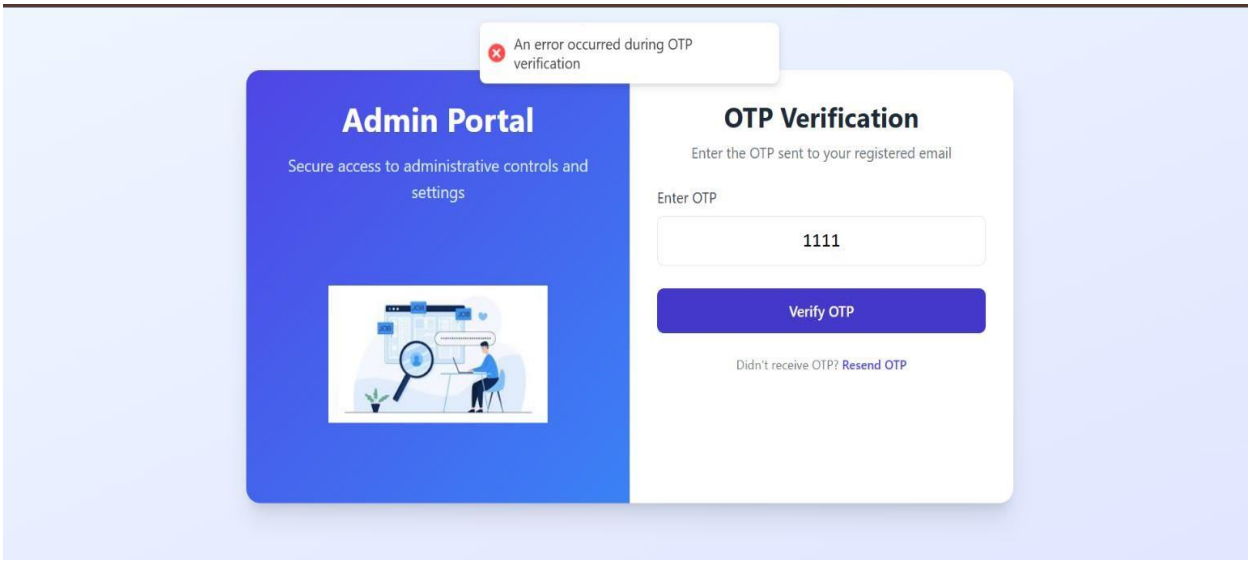

Figure 6.2 Testing with invalid OTP

Successful Admin Login:

Test Case:

Admin Login Successful

Precondition:

- Admin account exists in the system.
- Admin has valid credentials.
- The admin login page is accessible.

Test Steps:

1. Navigate to the admin login page.
2. Enter a valid admin username.
3. Enter the correct password.
4. Click on the Login button.
5. If OTP authentication is enabled:

○ Enter the correct OTP received via email/SMS.

○ Click on the Verify button.

Expected Result:

- The system should successfully authenticate the admin.
- The admin should be redirected to the dashboard/homepage.
- A success message like "Login successful." should be displayed.

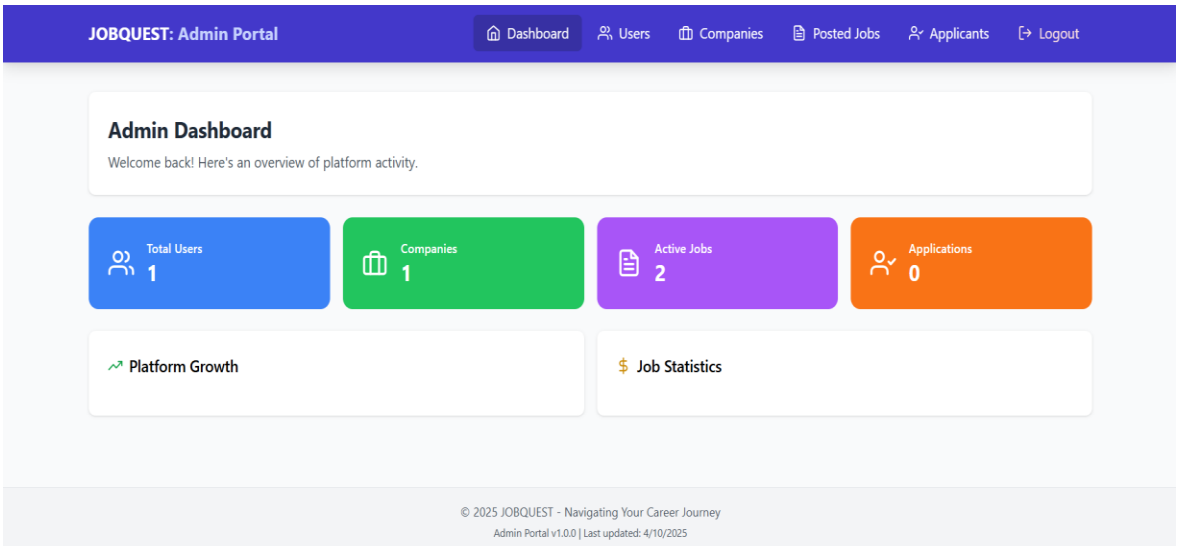

Figure 6.3: Admin Dashboard

View Registered Companies: Test Scenario:

View Registered Users

Precondition:

1. Navigate to the Admin dashboard.
2. Click on "View Registered Users.

Expected Result:

The system displays a list of registered users with relevant details (name, email, registration date).

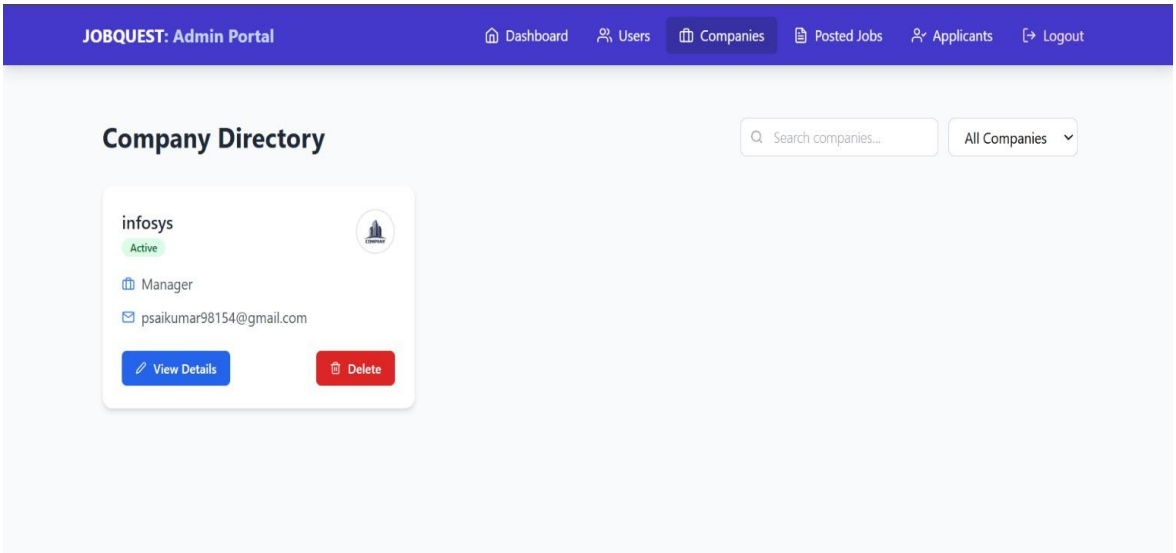

Figure 6.4: Registered Companies

View Active Job postings:

Test Scenario:

View Active Job Postings

Precondition:

Admin logged in

Test Steps:

1. Navigate to the Admin dashboard.
2. Click on "View Active Job Postings."

Expected Results:

The system displays a list of active job postings with details (job title, company, date posted).

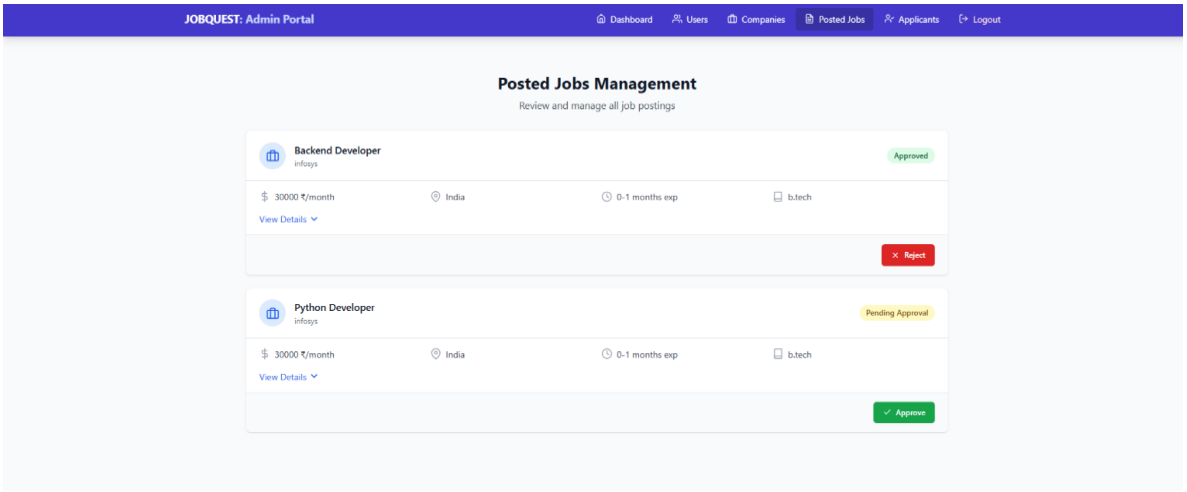

Figure 6.5: Active Job Postings

View Registered Users:

View Active Job postings:

Test Scenario:

View Active Registered Users

Precondition:

Admin logged in

Test Steps:

1. Navigate to the Admin dashboard.
2. Click on "View Users."

Expected Results:

The system displays a list of registered users

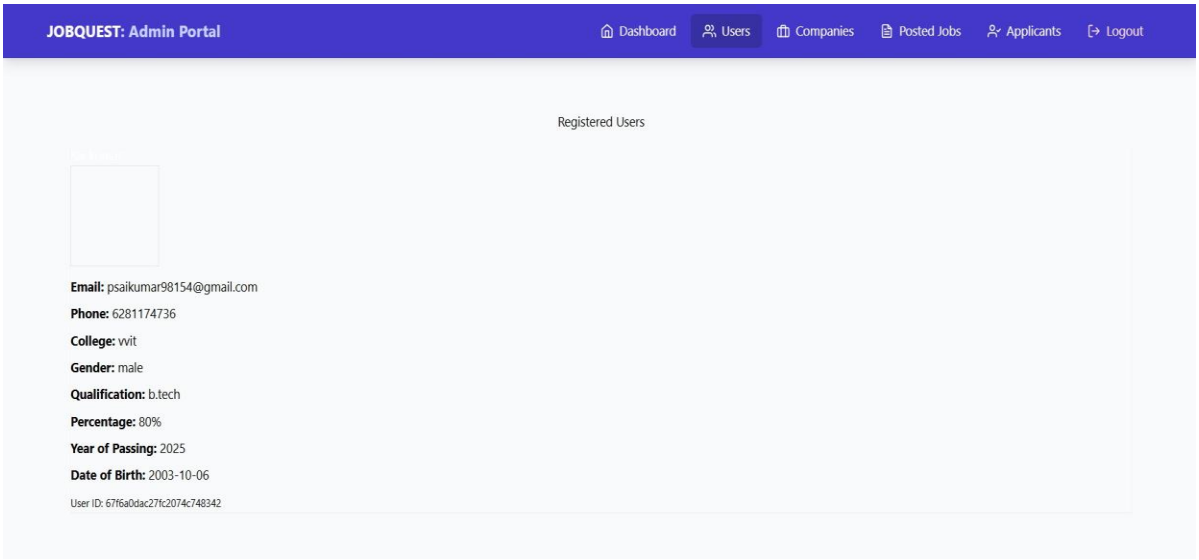

Figure 6.6: Registered Users

User Module:

User Registration:

Test Case:

User Registration

Precondition:

- The registration page is accessible.
- The user has a valid email and phone number.

Test Steps:

1. Navigate to the user registration page.
2. Enter the required details (e.g., name, email, password, etc.).
3. Click on the Sign Up button.

Expected Result:

- The system should create a new user account..

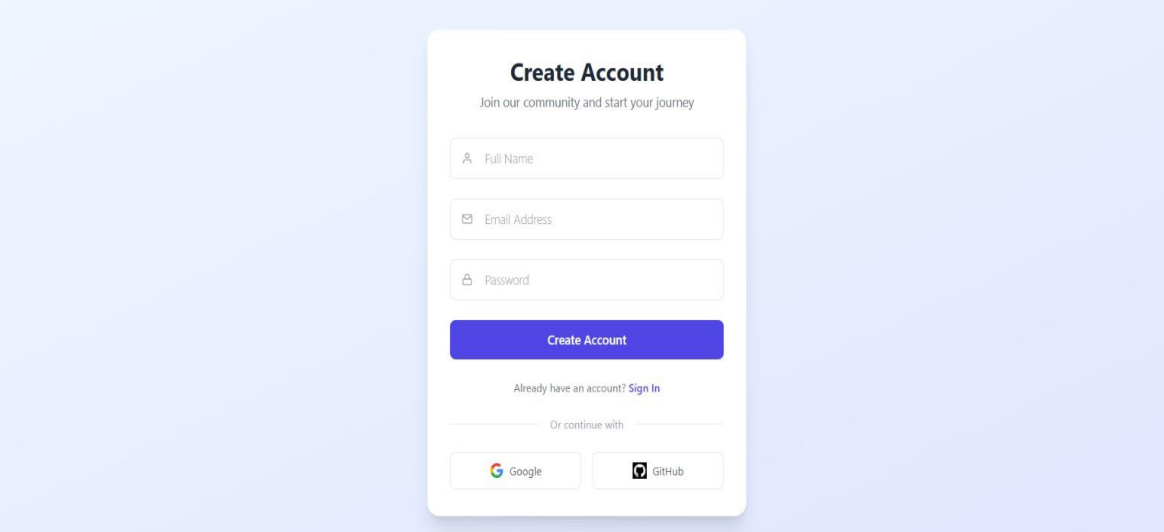

Figure 6.7: User Sign Up

Invalid user Email Id:

Test Case:

Login Using Invalid Email

Precondition:

- The login page is accessible.
- The email entered is not registered in the system.

Test Steps:

1. Navigate to the login page.
2. Enter an invalid or unregistered email.
3. Enter any password.
4. Click on the Login button.

Expected Result:

- The system should display an error message like "Invalid email".
- The user should not be logged in.

- The login page should remain visible.
- A "Sign Up" or "Forgot Password" link may be suggested.

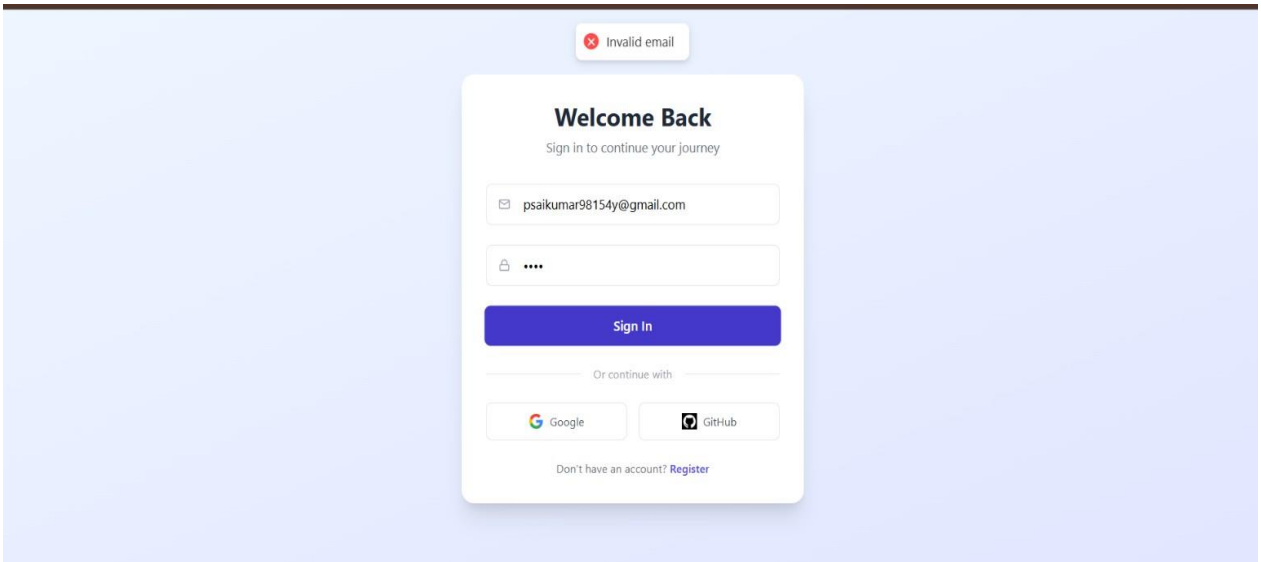

Figure 6.8: Testing with Invalid user Email Id

Invalid User Password:

Test Case:

Login Using Invalid User Password

Precondition:

- The user account exists in the system.
- The login page is accessible.

Test Steps:

1. Navigate to the login page.
2. Enter a valid registered email/username.
3. Enter an incorrect password.
4. Click on the Login button.

Expected Result:

- The system should display an error message like "Invalid password. Please try again."
- The user should not be logged in.
- The login page should remain visible.
- A "Forgot Password?" link may be displayed for password recovery.

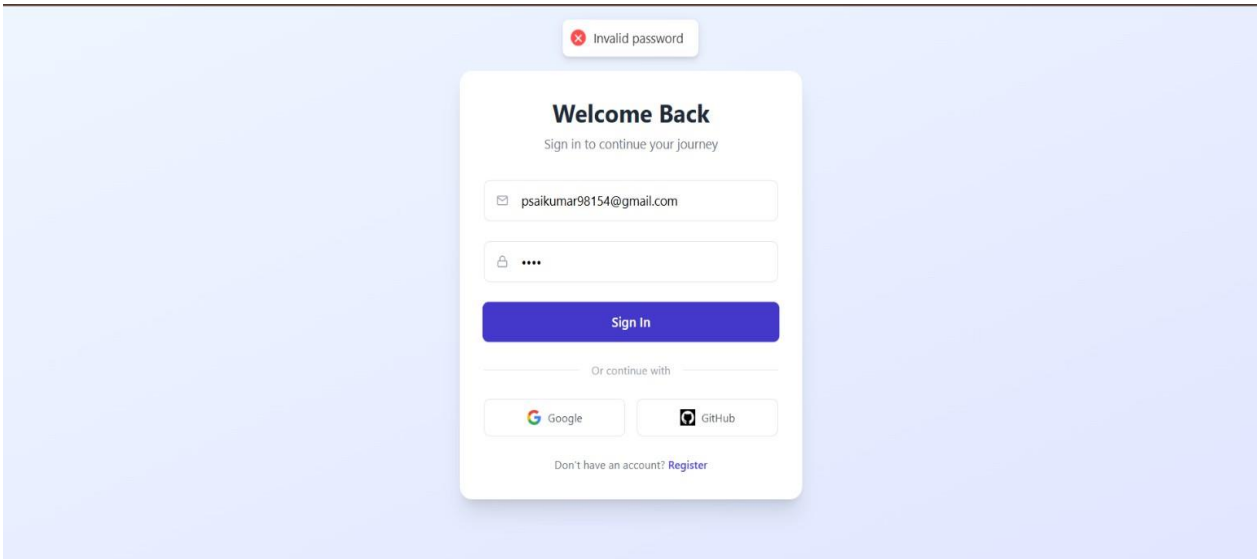

Figure 6.9: Testing with Invalid user Password

Invalid OTP:

Test Case:

Login Using Invalid OTP

Precondition:

- The user has entered valid credentials and reached the OTP verification step.
- A valid OTP has been sent to the registered email/phone.

Test Steps:

1. Navigate to the login page.
2. Enter a valid registered email/username and correct password.
3. Click on the Login button.
4. On the OTP verification page, enter an incorrect OTP.
5. Click on the Verify button.

Expected Result:

- The system should display an error message like "An error occurred during OTP verification."
- The user should not be logged in.
- The system may allow limited OTP attempts before blocking further attempts or requiring a new OTP request.

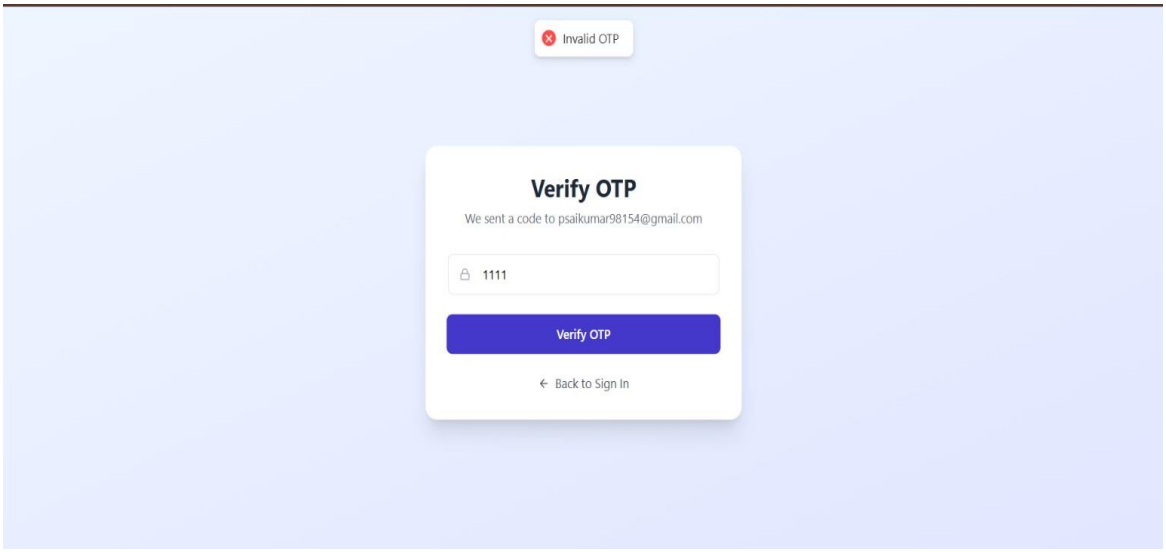

Figure 6.10: Testing with incorrect User OTP

Successful User Login:

Test Case:

User Login Successful

Precondition:

- The user account exists in the system.
- The login page is accessible.
- The user has valid credentials.

Test Steps:

1. Navigate to the login page.
2. Enter a valid registered email/username.
3. Enter the correct password.
4. Click on the Login button.

Expected Result:

- The system should successfully authenticate the user.
- The user should be redirected to the dashboard/homepage.
- A success message like "Login successful." should be displayed.

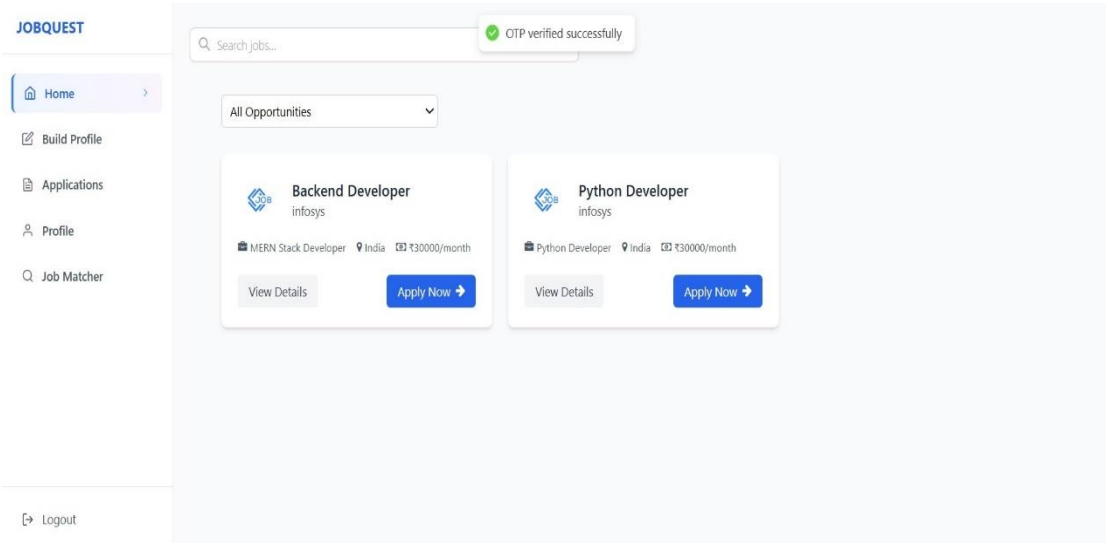

Figure 6.11: User Dashboard

Search for jobs:

Test Case:

User Searching for a Job

Precondition:

- The user is logged into the system.
- The job search functionality is available.

Test Steps:

1. Navigate to the Job Search page or section.
2. Enter a keyword (e.g., job title, company name, or skill).
3. Apply filters if available (e.g., location, experience level, salary range, job type).
4. Click on the Search button.
5. View the list of matching job postings.

Expected Result:

- The system should display a list of jobs that match the search criteria.
- The user should be able to see job details like title, company, location, salary, and application deadline.
- The user should have options to save, apply, or view more details for each job.

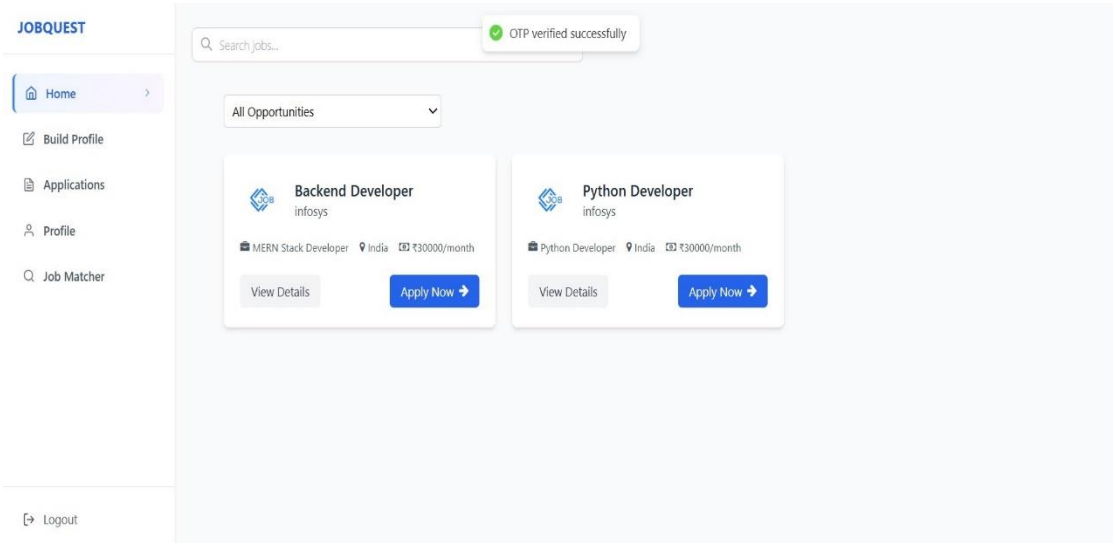

Figure 6.12: Job List

User Applying for Job:

Test Case:

User Applying for a Job

Precondition:

- The user is logged into the system.
- The job search functionality is available.
- The user has a complete profile with necessary details (resume, contact information, etc.).

Test Steps:

1. Navigate to the Job Search page.
2. Search for a job using keywords or filters.
3. Click on a job listing to view details.
4. Click on the Apply Now button.
5. If required, upload a resume or fill in additional application details.
6. Confirm the application by clicking Submit.

Expected Result:

- The system should display a confirmation message like "Application submitted successfully."
- The job should appear in the Applied Jobs section of the user's profile.
- The employer/recruiter should receive the application details.
- The user may receive a confirmation email or notification.

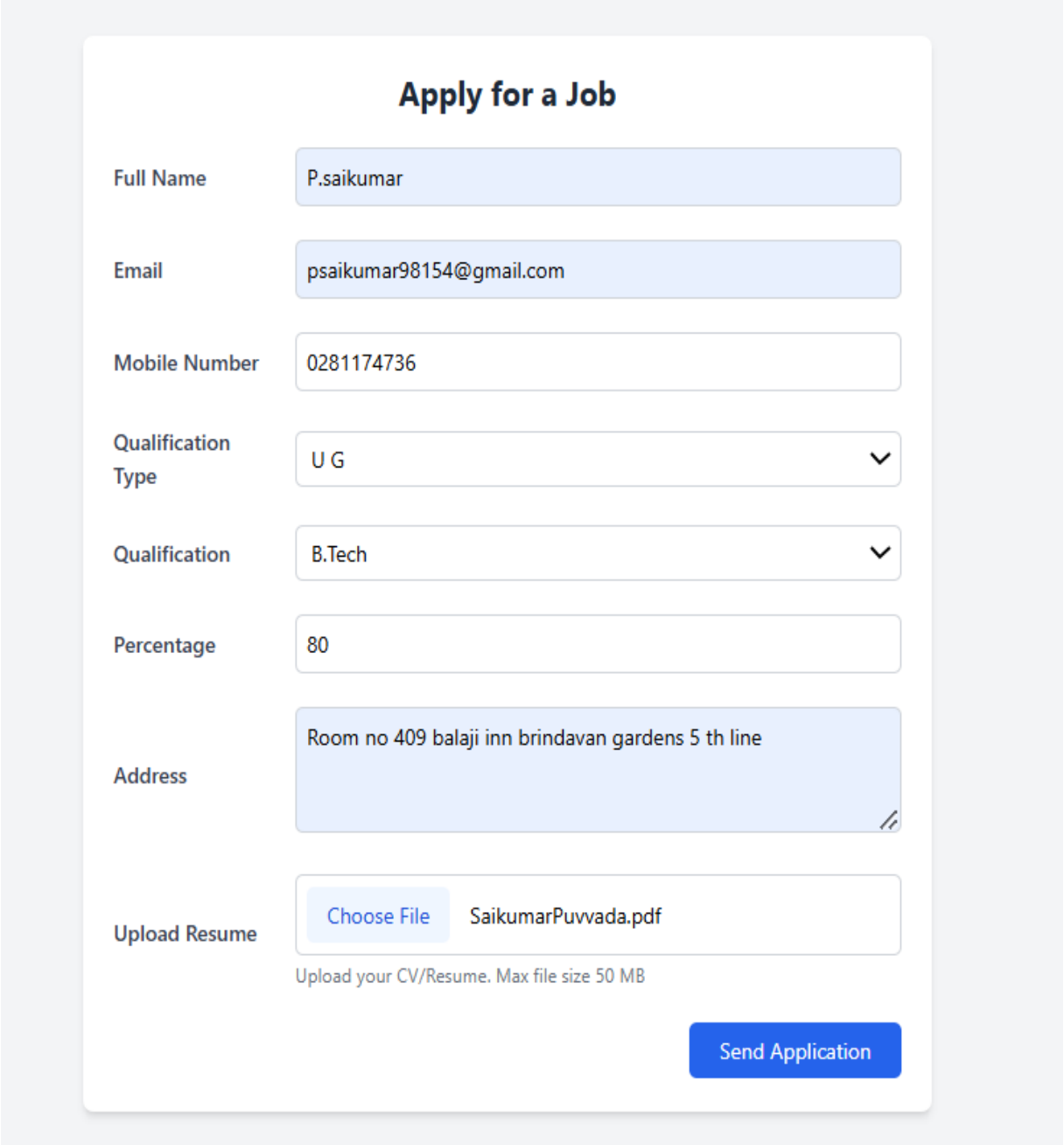

Figure 6.13: User Applying a Job

View Application Status:
Test Case:
View Application Status

Precondition:

- User (job seeker) is logged into the system.
- The "My Applications" or "Applied Jobs" section is accessible from the user's dashboard.

Test Steps:

1. Navigate to the " Applications" or section from the user dashboard.
2. View the list of jobs the user has applied for.
3. Select any job listing to view details.
4. Check the application status (e.g., Submitted, Under Review, Shortlisted, Rejected, Hired).

Expected Result:

- The system should display a list of all jobs the user has applied to.
- Application status should be clear and up-to-date.
- User should not be able to edit the application or status.

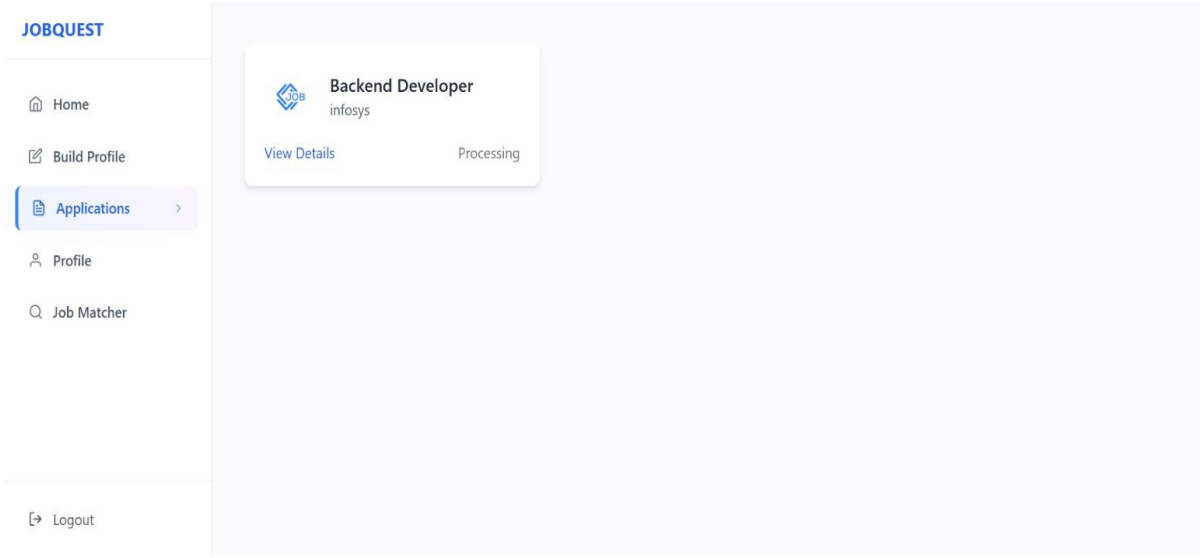

Figure 6.14: Application Status

Recruiter Module:

Recruiter Registration:

Test Case:

Recruiter Registration

Precondition:

- The recruiter registration page is accessible.
- The recruiter has a valid company email and necessary details.

Test Steps:

1. Navigate to the Recruiter Registration page.
2. Enter the required details:
   - Full Name
   - Company Name
   - Company Email
   - Phone Number
   - Password
3. Click on the Sign Up/Register button.

4. If email/phone verification is required:
   - Enter the OTP sent via email/SMS.
   - Click on the Verify button.
5. After successful verification, the system completes the registration process.

Expected Result:

- A new recruiter account should be created successfully.
- A success message like "Registration successful. Please log in." should be displayed.
- The recruiter may need admin approval before posting jobs.

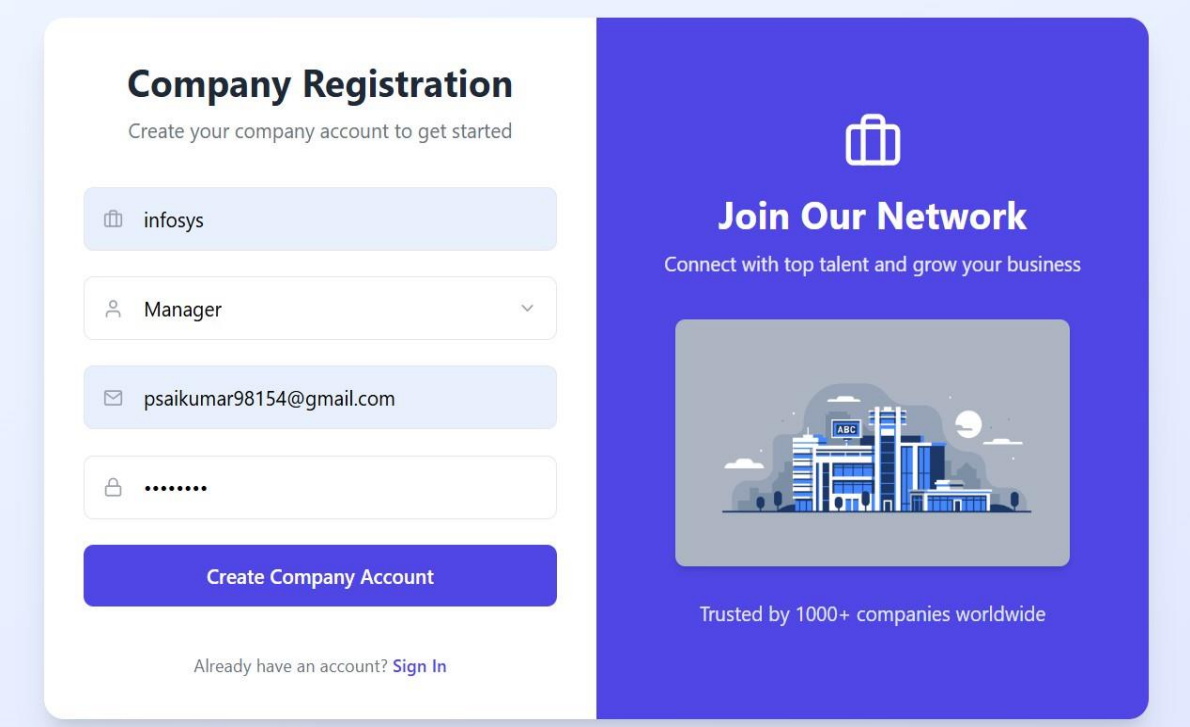

Figure 6.15: Recruiter Registration

Add a Job:

Test Case:

Add a Job

Precondition:

- Recruiter is logged into the system.
- The "Add Job" page/form is accessible.

Test Steps:

1. Navigate to the Add Job section from the recruiter dashboard.
2. Fill in the job details:
   - Job Title
   - Company Name
   - Location
   - Job Type (e.g., Full-Time, Internship)
   - Salary Range

- Job Description
- Required Skills
- Application Deadline

3. Click the Submit/Post Job button.

Expected Result:

- The system should validate all required fields.
- Upon successful validation, the job should be saved and listed under the recruiter's posted jobs.
- A success message like "Job posted successfully" should be displayed.
- The job should be visible to job seekers in the job listings.

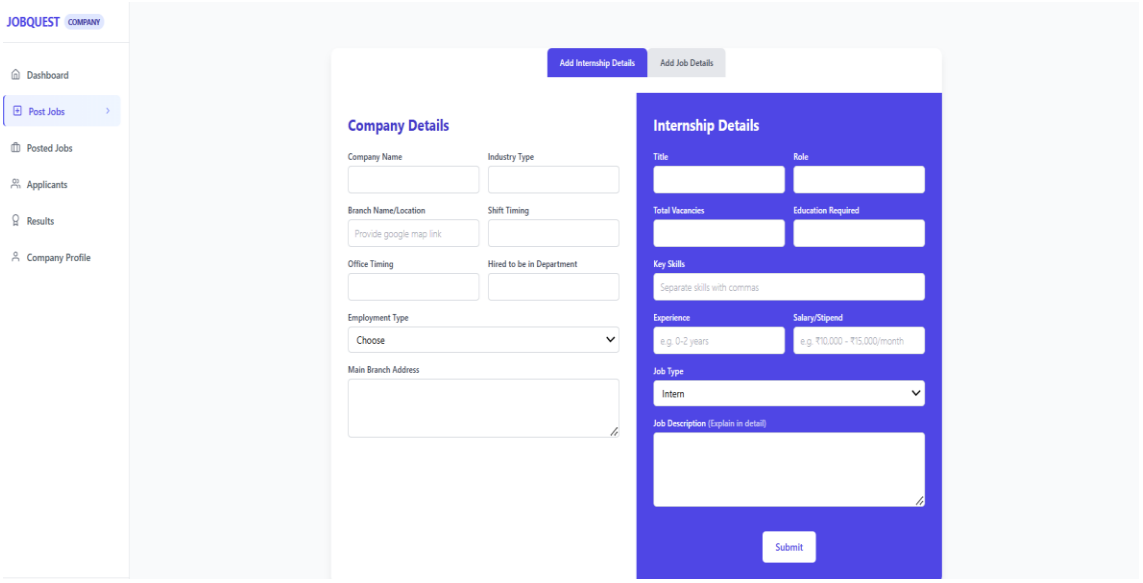

Figure 6.16: Creating a New Job

Online Assessment:

Test Case:

Online Assessment

Precondition:

- The user (candidate) is logged into the system.
- The user has applied for a job that requires an online assessment.
- The online assessment feature is enabled.

Test Steps:

1. Navigate to the Assessments section from the user dashboard.
2. Select the assigned test for the applied job.
3. Click on the Start Test button.
4. Answer each question within the time limit.
5. Click Submit after completing the test or wait for the timer to expire.

Expected Result:

- The system should record and save all responses.

- If the test is auto-graded, the candidate should see their score or a confirmation message like "Test submitted successfully.
- If the test requires manual evaluation, a message like "Your test will be reviewed soon." should be displayed.
- The recruiter should receive the test results in their dashboard.

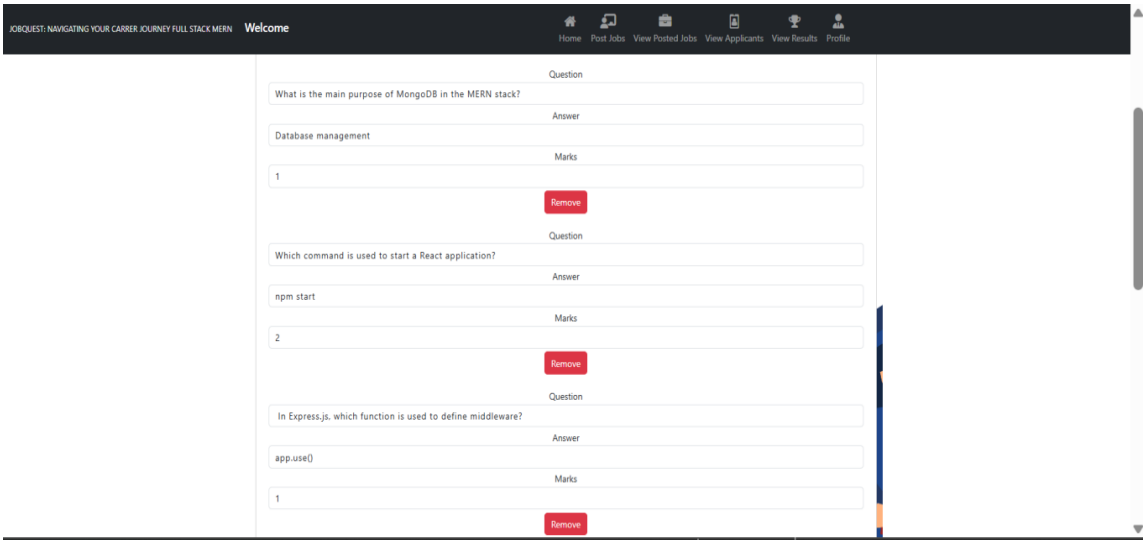

Figure 6.17: Creating Online Assessment

Evaluate Exam Results:

Test Case:

Evaluate Exam Results

Precondition:

- The online assessment has been completed by candidates.
- The system supports automatic or manual result evaluation.
- The recruiter or admin has access to the exam results.

Test Steps:

1. Navigate to the Admin/Recruiter Dashboard.
2. Select “View Results”.
3. Choose a specific candidate’s exam to evaluate.
4. If auto-evaluation is enabled:

○ The system should display scores based on pre-defined correct answers.

5. Click Save/Submit Results after evaluation.

Expected Result:

- The candidate’s score should be updated in the system.
- The recruiter/admin should be able to filter and download exam results if needed.

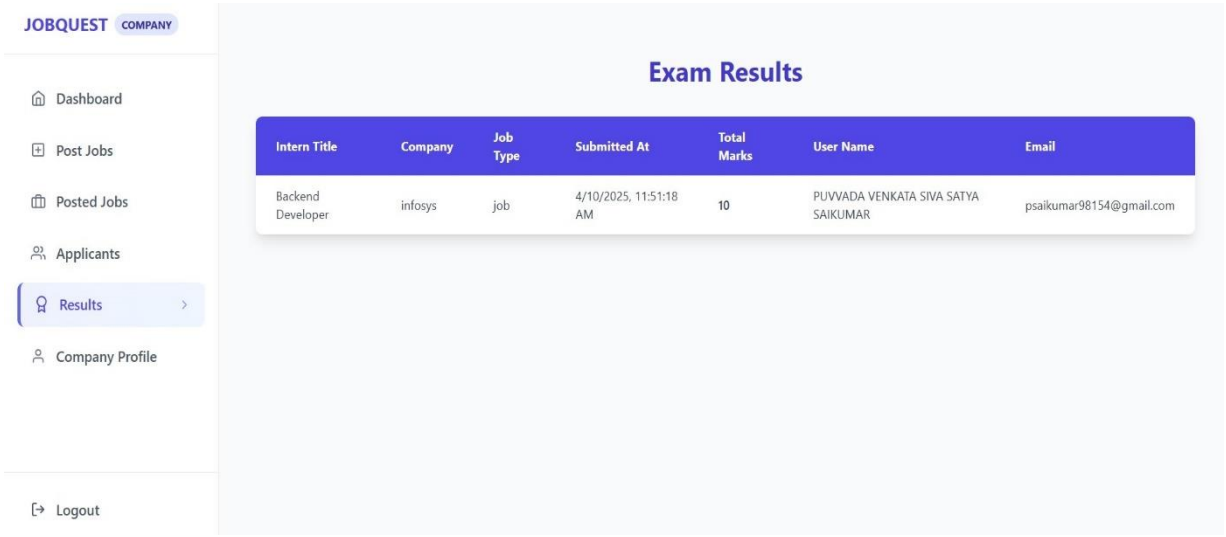

Figure 6.18: Exam Results

6.3 Performance Evaluation

Admin Module: The Admin module provides the administrator with a centralized interface to manage the job portal effectively. Admins can view the number of registered users, companies, and active job postings. They also have the authority to monitor and maintain data integrity across the system, ensuring smooth and efficient operations.

User Module (Jobseeker/Student): The User module enables job seekers and students to register on the platform, create profiles, and log in to search for jobs that match their skills. The system provides skill-based job recommendations to enhance job search efficiency. Users can also apply for jobs, participate in virtual exams, view the status of their applications, and track their progress in real time. **Recruiter Module (Company):** The Recruiter module allows companies to register and log in to the portal. Recruiters can add job requirements, post new job openings, and create online assessments (MCQs) for applicants. They can also view candidate profiles, evaluate exam results, and make informed hiring decisions by selecting or rejecting applicants based on their performance and qualifications.

6.4 Screenshots of Application Output:

Landing Page : When the Execution started this is the landing page.

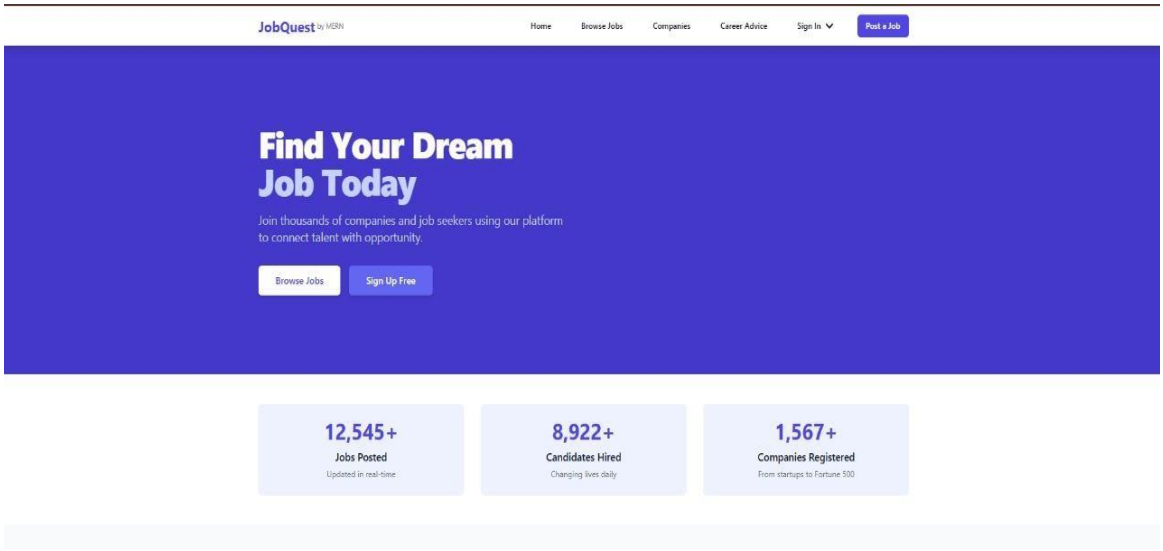

Figure 6.19: Landing Page

User Home Page : This is the user page after successful login. The job details will be displayed.

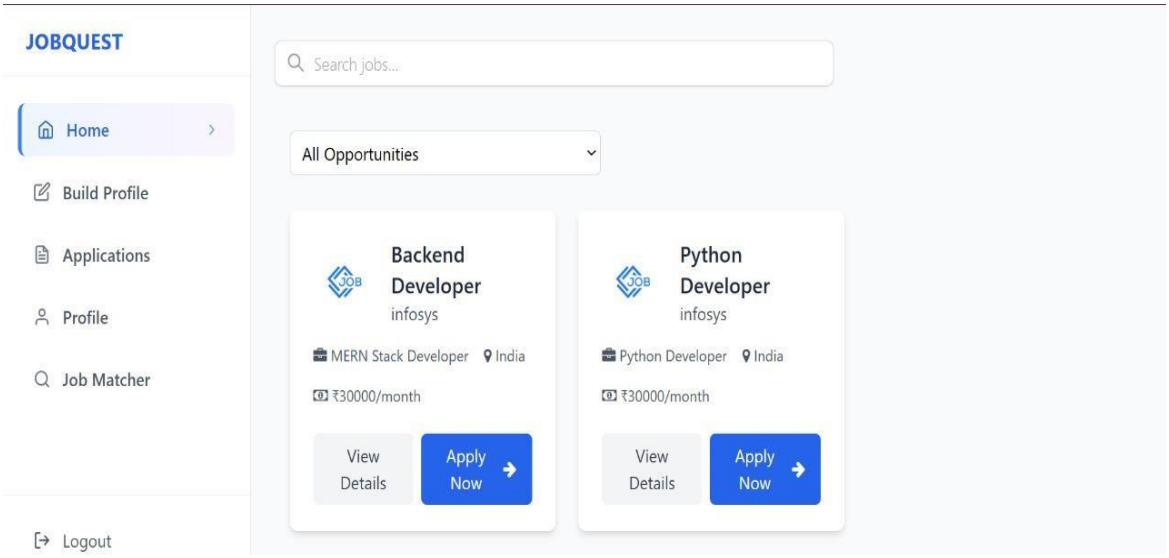

Figure 6.20: User Home Page

User Profile page: This is the user profile page here users can view their details.

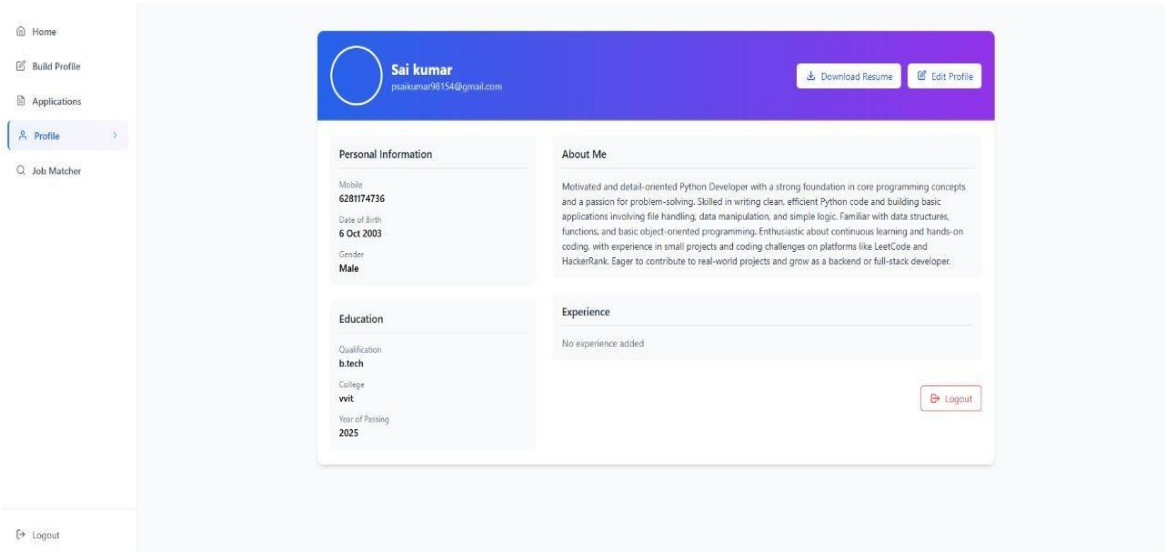

Figure 6.21: User Profile Page

User Job Recommendation page: In this page the users will get their personalized recommendation based on their preferences.

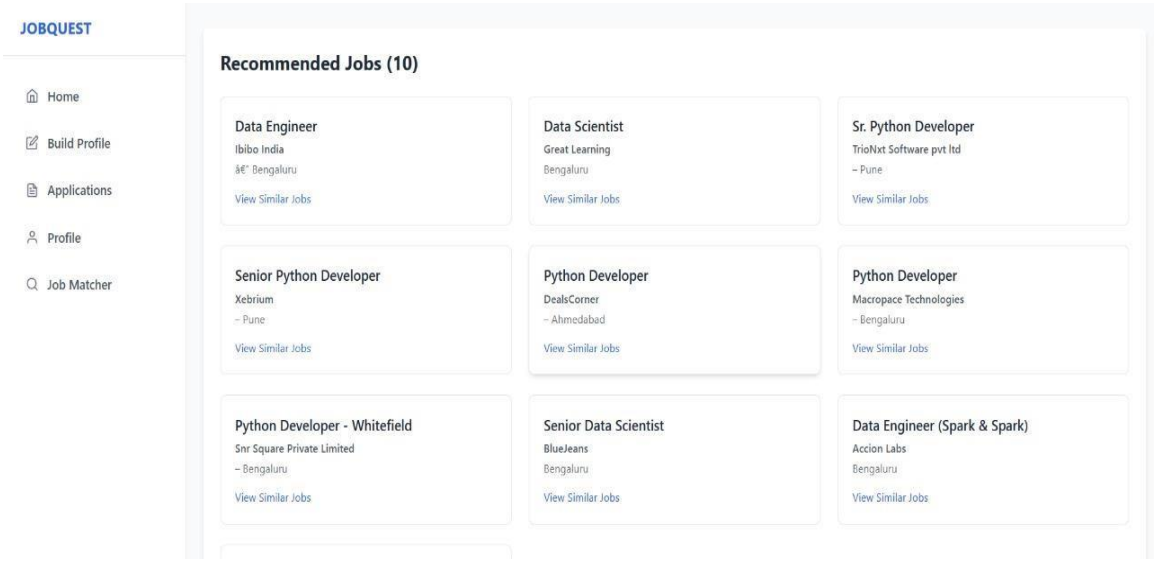

Figure 6.22: Job Recommendation

User Apply page : In this page user can apply for a job.

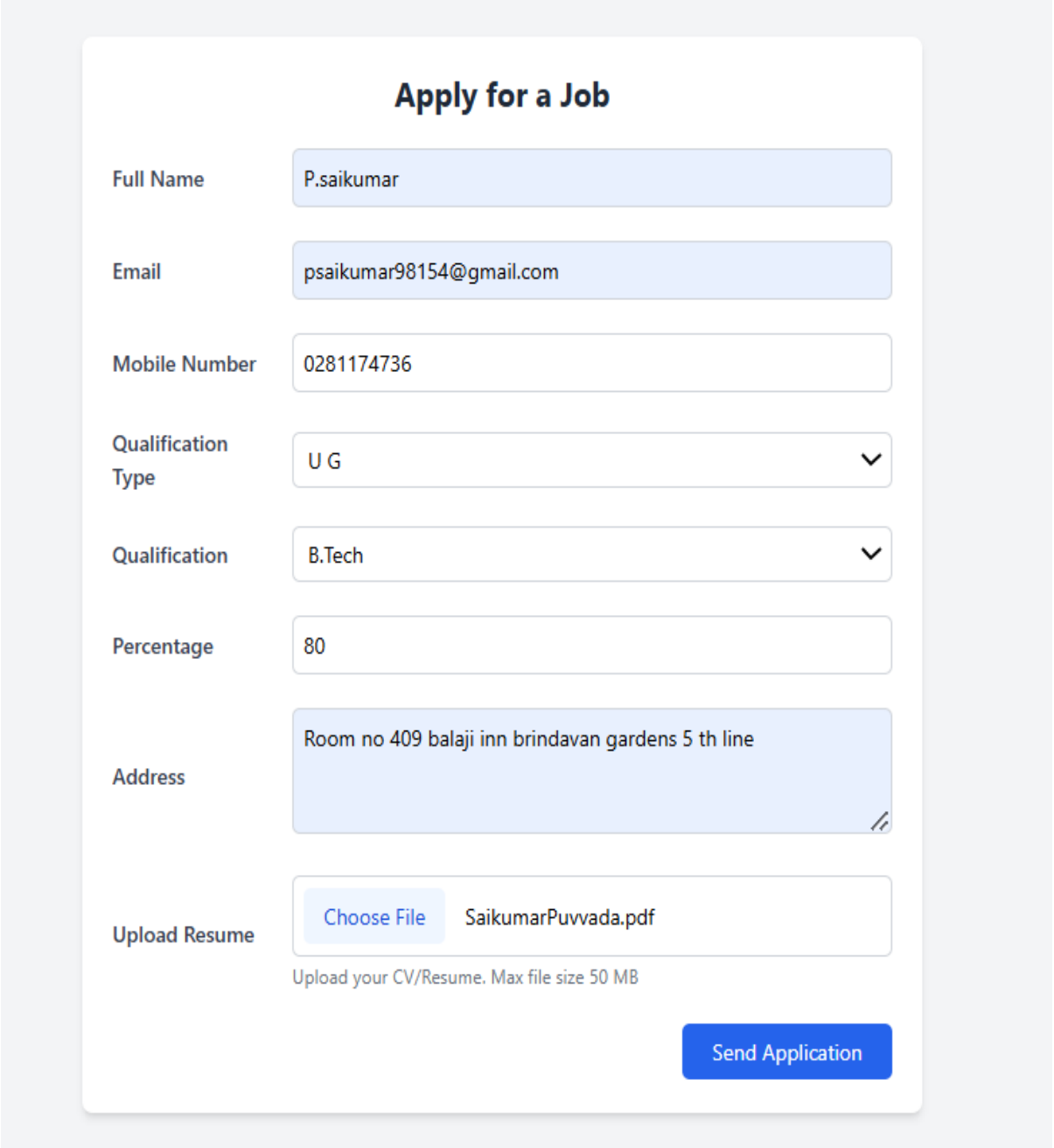

Figure 6.23: User Applying For a Job

Employer/Company Sign In page: This is the company Sign in page.

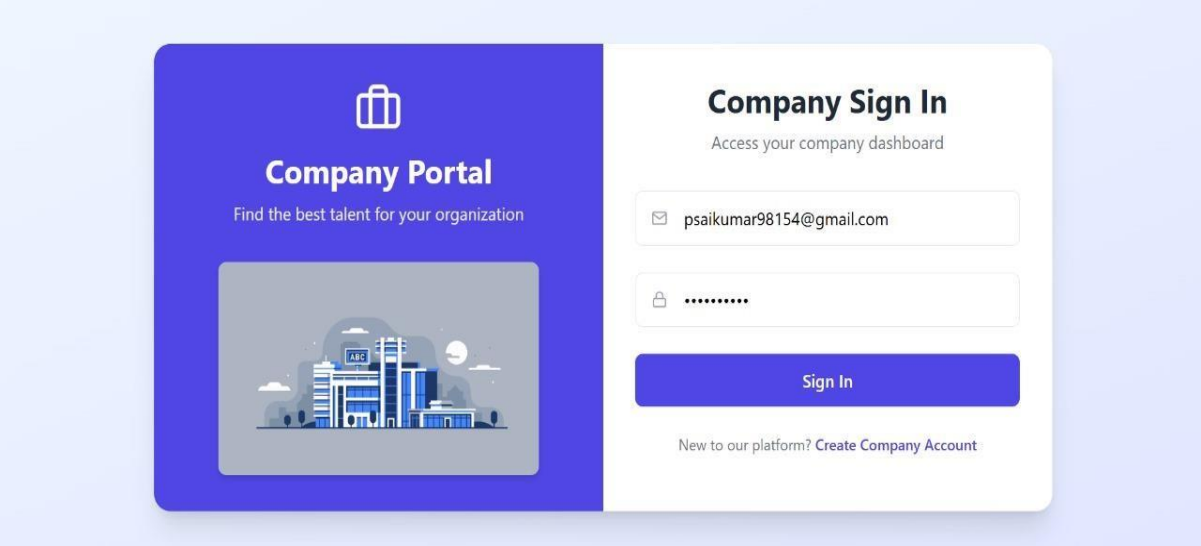

Figure 6.24: User Profile Page

Employer/Company Profile page: This is the company profile page.

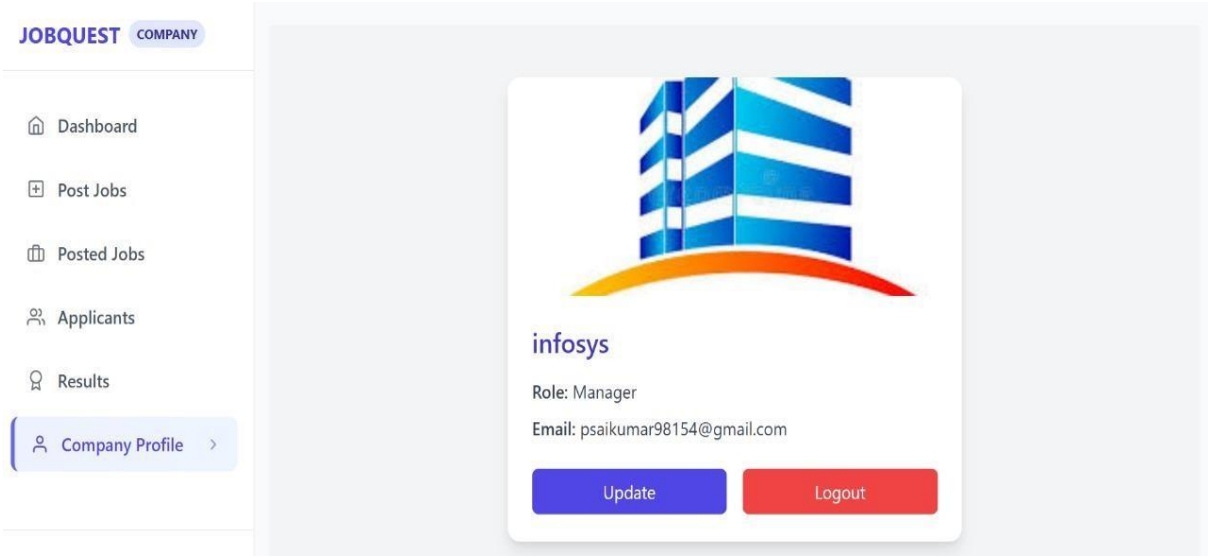

Figure 6.25:Company Profile Page

Employer/Company Job posting page: In this the employer will add the recruitement details of company details and role, location, salary etc.

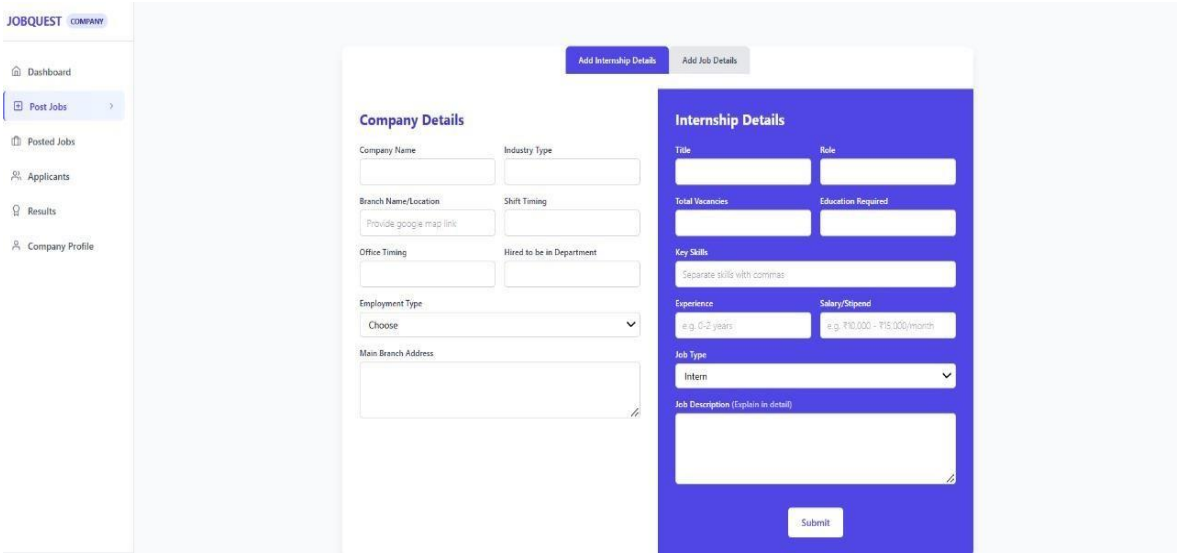

Figure 6.26:Job Posting Page

Employer/Company Posted Jobs page: In this page the employer/company can view the jobs posted by them and can update or delete them.

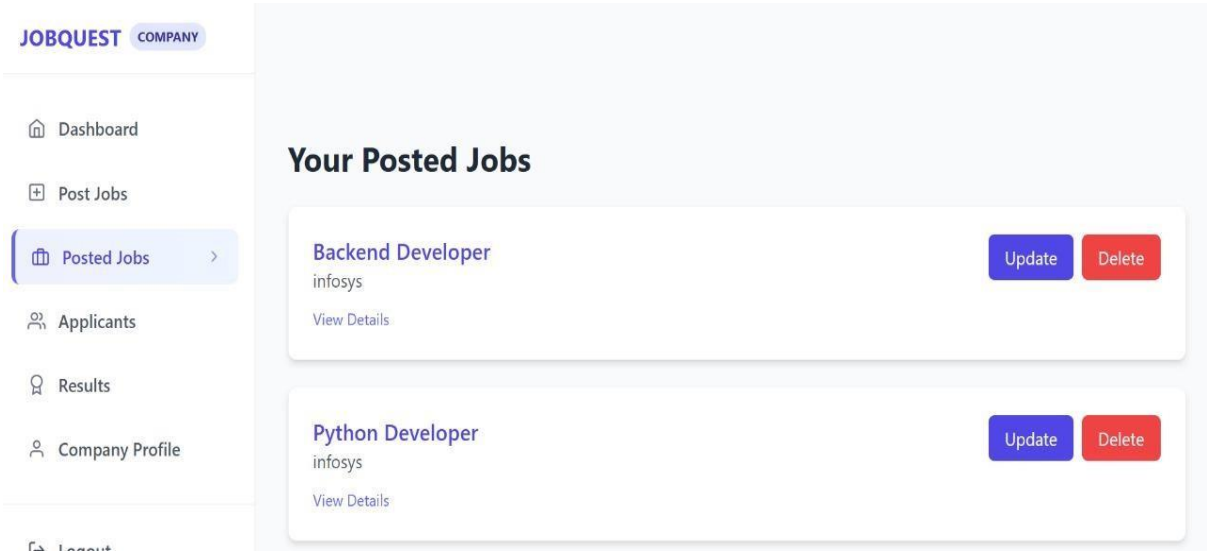

Figure 6.27: Posted Jobs

Employer/Company Dashboard page: In this the hiring details of jobs posted and active applications of company will be displayed.

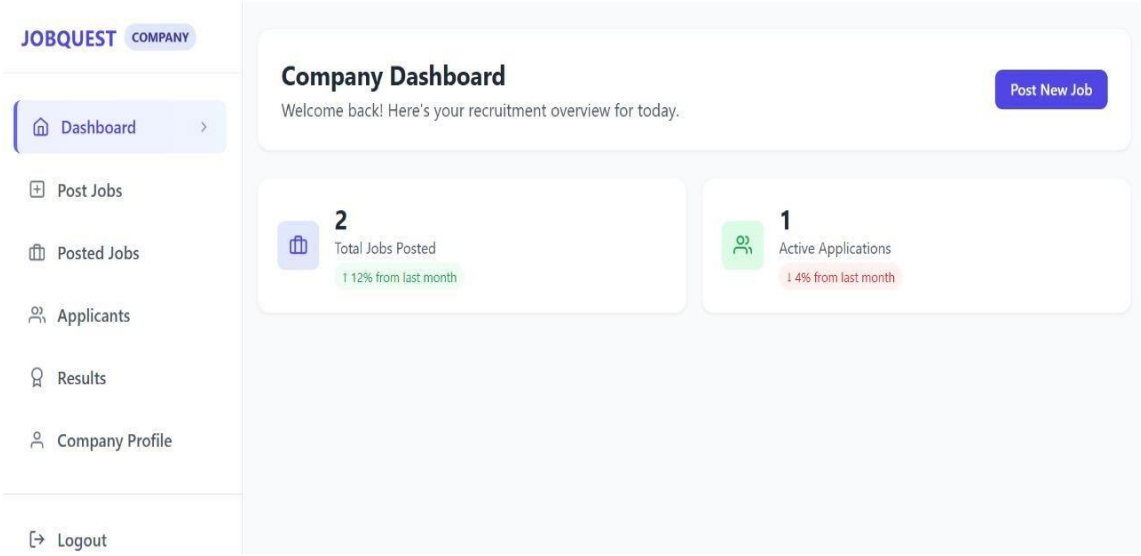

Figure 6.28:Company Dashboard

Employer/Company Application List page: In this page the Employer can view the applicants.

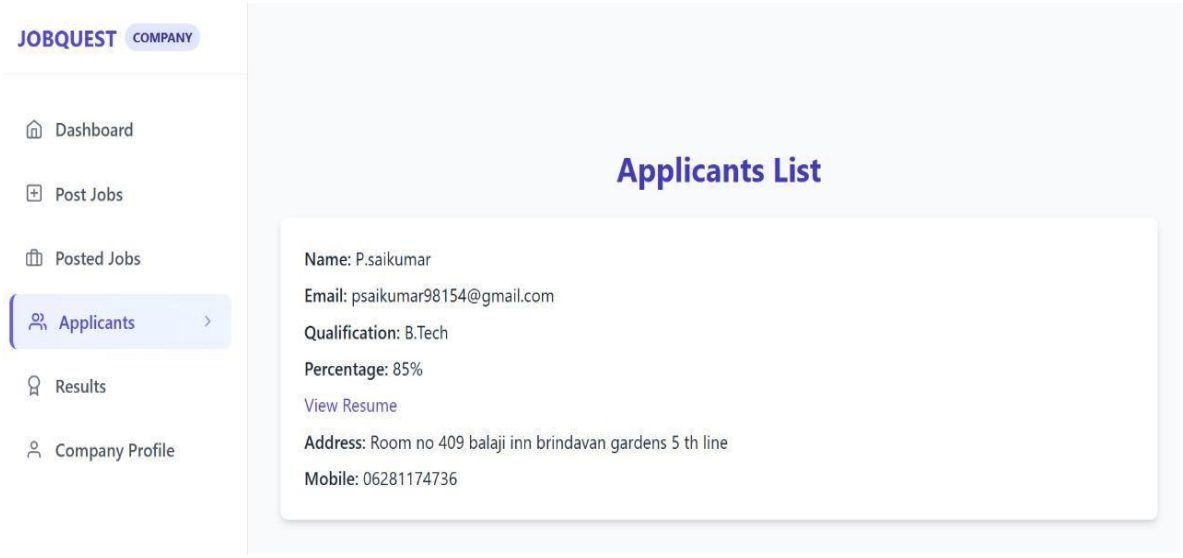

Figure 6.29:Company's Application List

Admin Dashboard page : This is the admin dashboard page.

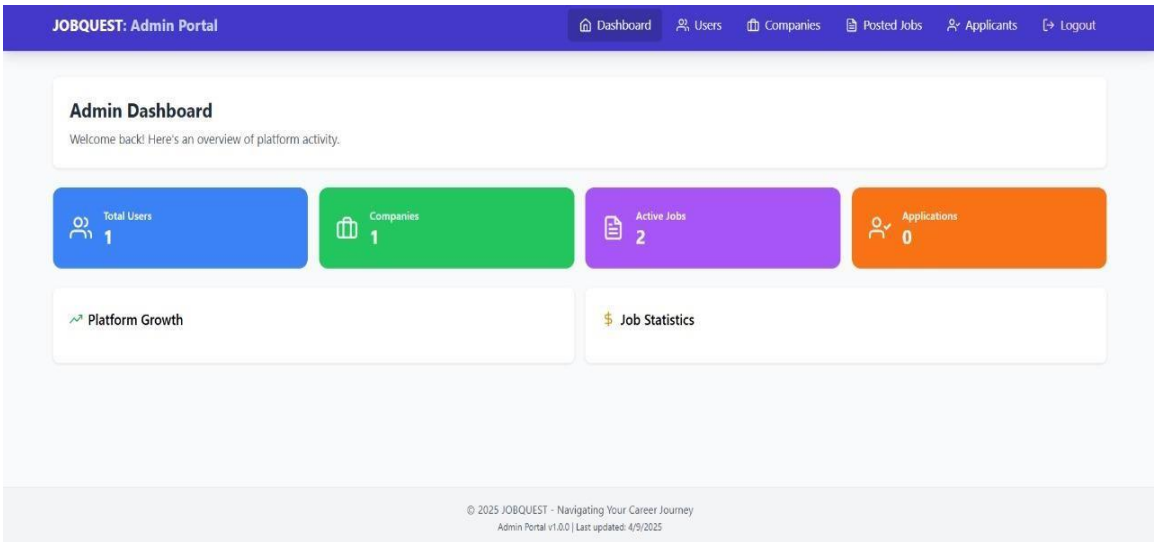

Figure 6.30:Admin Dashboard

Admin viewing Registered Users page: In this page admin can view the registered users.

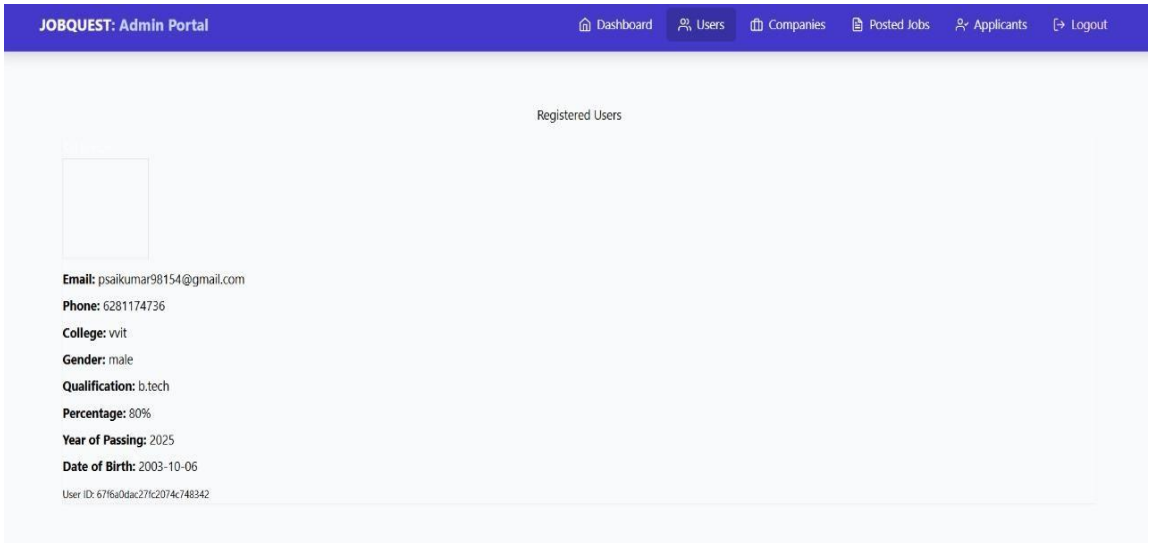

Figure 6.31: Admin Viewing Registered Users

Admin Managing Posted Jobs page:In this page the admin can view and manage the posted jobs.

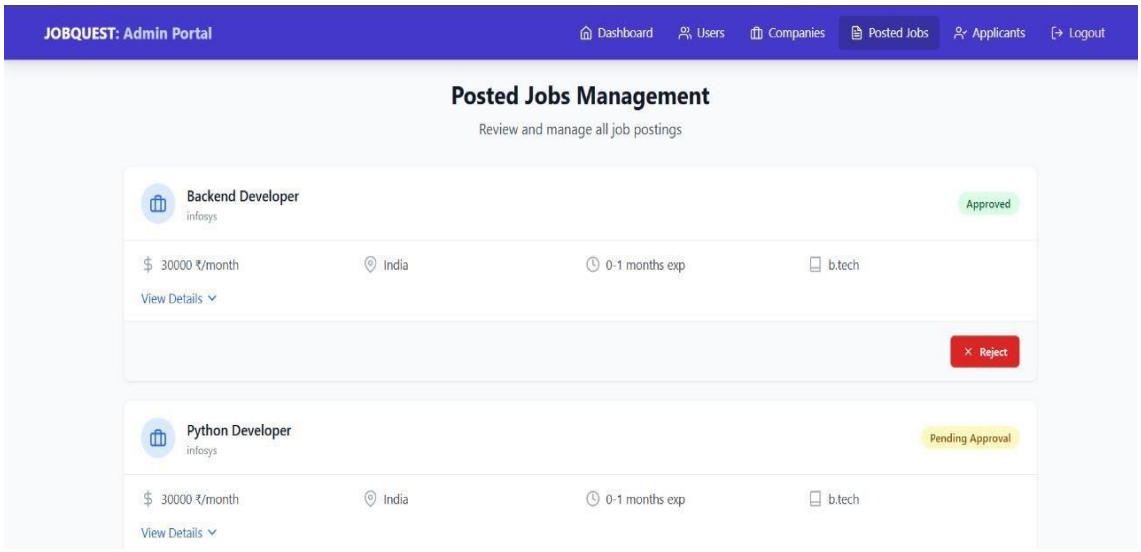

Figure 6.32:Admin Managing Posted Jobs page

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 SUMMARY OF FINDINGS

The development of Job Quest has demonstrated the potential of AI-driven recruitment solutions to enhance traditional job portals. By integrating resume scoring, skill-based job recommendations, and automated hiring workflows, Job Quest streamlines the job search and recruitment process for both job seekers and employers. Unlike conventional platforms that rely solely on keyword matching, Job Quest employs Natural Language Processing (NLP) and Machine Learning (ML) to deliver personalized, bias-free, and accurate job recommendations. The system also provides real-time job tracking, career insights, and an AI-powered chatbot, improving user engagement and efficiency.

7.2 KEY ACHIEVEMENTS AND CONTRIBUTION

The development and implementation of Job Quest have resulted in several key achievements:

- AI-Powered Job Matching: Enhanced accuracy in job recommendations using Machine Learning algorithms to analyze candidate profiles, job descriptions, and industry trends.
- Resume Scoring System: A data-driven approach to evaluating and improving resumes, providing job seekers with actionable feedback.
- Automated Hiring Processes: Streamlined candidate shortlisting, interview scheduling, and communication through AI-driven automation.
- Real-Time Job Tracking: Improved transparency for applicants, enabling them to track their job applications and receive instant updates.
- Career Insights & Personalized Guidance: Providing users with insights on trending skills, industry demands, and career growth opportunities.

7.3 CHALLENGES FACED

During the development of Job Quest, several challenges were encountered, including:

- Data Availability & Quality: Ensuring a diverse and high-quality dataset for training AI models to improve job-candidate matching.
- Bias Mitigation: Addressing potential biases in AI-driven recruitment to promote fairness and inclusivity.
- User Adoption & Experience: Designing an intuitive and user-friendly interface to encourage adoption by both job seekers and recruiters.

- Scalability & Performance: Optimizing the platform to handle large volumes of user interactions, job postings, and application tracking efficiently.
- Integration with Existing Systems: Ensuring seamless integration with HR software, LinkedIn profiles, and other job portals for better interoperability**.**

7.4 FUTURE SCOPE AND IMPROVEMENTS

Future enhancements to this job portal system can focus on incorporating advanced features and technologies to further improve the user experience and streamline the recruitment process. One of the key enhancements could be the integration of machine learning algorithms for more accurate job recommendations and candidate matching. By analyzing user profiles, job descriptions, and application history, the system could suggest job opportunities that align better with a candidate's skills and preferences, improving the relevance of recommendations. Additionally, the portal could implement a chatbot powered by AI to assist users with common queries, application guidance, and job search support, providing 24/7 assistance and reducing the need for human intervention.

Another potential enhancement is the addition of video interview functionality, allowing recruiters to conduct remote interviews directly through the platform. This would not only save time but also enable companies to assess candidates from different locations without requiring travel. The system could also offer integrated tools for recruiters to evaluate candidate performance through real-time coding challenges, collaborative whiteboarding sessions, and behavioral assessments, enriching the recruitment process.

CONCLUSION

In conclusion, this job portal system, built using the MERN stack, provides a comprehensive and efficient platform for both job seekers and recruiters. By streamlining essential functionalities such as user registration, job searching, job application tracking, and recruitment management, the system offers a seamless experience for all users. Job seekers can easily search for and apply to positions that match their skills, participate in exams, and track the progress of their applications, while recruiters can efficiently manage job postings, create exams, and evaluate candidates. The Admin module ensures that the system remains well-maintained and organized. With its user-friendly interface and integration of important features like job recommendations, live exams, and application status updates, this job portal simplifies the recruitment process and enhances job-seeking efforts, making it a valuable tool for both companies and individuals seeking employment.

CHAPTER 8

REFERANCES

[1] Author: Gasparėnienė, L., Matulienė, S., Žemaitis, E. (2025)
Title: *Opportunities of Job Search through Social Media Platforms and its Development in Lithuania*
Outcome: This study highlights the growing use of platforms like LinkedIn for job searches, emphasizing personalized recommendations and the shift from traditional methods to social media-centric approaches in Lithuania.
Disadvantages: Concerns over data privacy and limited feedback for job seekers.

[2] Author: Hamdane, A., Belhaj, N., El Hamdaoui, H., El Bekkali, M. (2024)
Title: *Big Data Based Architecture for Connecting Graduates and Recruiters in Morocco*
Outcome: Proposes a system leveraging advanced analytics to bridge the gap between university graduates and potential employers.
Disadvantages: High computational costs and data security concerns.

[3]Author: Kendle, A. S., Nagare, M. S., Patre, H. G., Zanwar, R. S. (2024)
Title: *Enterprise Resource Planning System for Training and Placement Cell in Educational Institutes*
Outcome: Introduces an ERP system for automating the placement process in educational institutes, facilitating tracking of job applications and recruiter-student interactions.
Disadvantages: Scalability issues and data handling challenges during high-demand periods.

[4] Author: Heggo, I. A., Abdelbaki, N. (2014)
Title: *Hybrid Information Filtering Engine for Personalized Job Recommender Systems*
Outcome: Presents a blend of content-based and collaborative filtering techniques to enhance job recommendations.
Disadvantages: Computational intensity of the hybrid approach.

[5] Author: Kaur, D., Kaur, R. (2024)
Title: *Does Electronic Word-of-Mouth Influence E-Recruitment Adoption?*
Outcome: Examines the role of eWOM in influencing the adoption of e-recruitment platforms, highlighting how positive reviews increase platform credibility.
Disadvantages: Managing fake or biased reviews remains a challenge in maintaining user trust.

[6] Author: Ghanouane, K., Benkaraache, T. (2024)
Title: *Big Data Education Landscape for Graduates in Morocco: Insights from 2022 Offerings*
Outcome: Analyzes education programs in Morocco focused on big data, emphasizing the need to balance technical skills like programming with functional skills like communication and change management.
Disadvantages: Potential gaps in aligning educational offerings with industry requirements.

[7] Author: Khlebarodava, H., Remeikiene, R. (2023)
Title: *Opportunities of E-recruitment through Social Media Platforms and its Development in Lithuania*
Outcome: Reviews the growing role of social media in job searches, exploring benefits like democratized access to job listings.
Disadvantages: Privacy concerns faced by users.

[8] Author: MassHire JobQuest (2023)
Title: *MassHire JobQuest System*
Outcome: A crucial tool for job seekers in Massachusetts, offering features like skill matching, job recommendations, and career planning.
Disadvantages: Limited information on system limitations; further research needed.

[9] Author: Deshmukh, P. B. et al. (2022)
Title: *Placement Portal using MERN Technology*
Outcome: Represents an ERP system for Training and Placement Cells in educational institutes, aiming to automate the entire placement process.
Disadvantages: Challenges in system scalability and user adaptability.

[10] Author: Heggo, I. A., Abdelbaki, N. (2022)
Title: *Hybrid Information Filtering Engine for Personalized Job Recommender System*
Outcome: Provides a solution for personalized job recommendations using both content-based and collaborative filtering.
Disadvantages: Faces computational intensity challenges.

[11] Author: Wang, P., Hu, S., Wen, B., Lu, Z. (2022)
Title: "There is a Job Prepared for Me Here": Understanding How Short Video and Live-streaming Platforms Empower Ageing Job Seekers in China
Outcome: This study explores how short video and live-streaming platforms are utilized by ageing job seekers in China to overcome employment challenges.
Disadvantages: Potential issues include digital literacy barriers and the risk of age-related biases persisting in online platforms.

[12] Author: Agarwal, P. (2022)
Title: Big Data Integration with ERP Systems for Innovation and Efficiency
Outcome: The research discusses how integrating big data with ERP systems can drive innovation and enhance operational efficiency in organizations.
Disadvantages: Challenges include data integration complexities and the need for substantial infrastructure investments

[13] Author: Aral, S., et al. (2022)
Title: Looking for a New Job? Ask Someone You Barely Know Outcome: This study highlights the effectiveness of weak social ties over strong ones in job searches, emphasizing the role of acquaintances in providing new job information.
Disadvantages: Reliance on weak ties may lead to less personalized support and guidance during the job search process.

[14] Author: Jones, A. (2021)
Title: "LinkedIn Catfish" Aliyah Jones Created a Fake Profile to Expose Racial Inequity in Hiring
Outcome: An experiment revealing racial disparities in hiring practices, where a fictitious profile received significantly more interview invitations than the real applicant.
Disadvantages: Highlights systemic biases in recruitment processes that need to be addressed.

[15] Author: Psicosmart (2021)
Title: Leveraging Data Analytics in HR ERP Systems
Outcome: A case study showing that integrating data analytics into HR ERP systems can reduce recruitment costs and improve employee retention.
Disadvantages: Implementation may require significant changes in organizational processes and staff training.

[16] Author: Third Stage Consulting (2020)
Title: The Impact of Big Data on ERP Systems
Outcome: Discusses how big data enhances ERP systems by providing deeper insights and improving decision-making processes.
Disadvantages: Potential issues include data overload and the need for advanced analytics capabilities

[17] Author: Outstaff Your Team (2020)
Title: Big Data Recruitment: How to Automate the HR Sector
Outcome: Explores the benefits of using big data in automating HR recruitment processes, leading to more efficient hiring.
Disadvantages: High implementation costs and the necessity for specialized technical expertise.

[18] Author: Top10ERP (20218)
Title: AI in ERP: The Next Wave of Intelligent ERP Systems for 2025
Outcome: Examines the integration of AI into ERP systems, resulting in more intelligent and predictive enterprise solutions.
Disadvantages: Challenges include ensuring data quality and managing the complexity of AI algorithms.

[19] Author: Vorecol (2018)
Title: How Does Real-time Data from ERP Systems Transform HR Decision-making
Outcome: Analyzes how real-time data from ERP systems can enhance HR decision-making and improve organizational responsiveness.
Disadvantages: Potential issues with data accuracy and the need for continuous system monitoring.

[20] Author: NetSuite (2018)
Title: 60 Critical ERP Statistics: Market Trends, Data and Analysis
Outcome: Provides comprehensive statistics on ERP implementation trends, highlighting success factors and common challenges.
Disadvantages: Identifies issues such as insufficient testing and lack of stakeholder communication during ERP implementations.