Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT REPORT ON

Job Portal

Using Content-Based and Decision Tree Algorithm

A report submitted to

Department of Computer Application

Asian College of Higher Studies

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Suman Maharjan [210618]

Under the Supervision of

Govinda Gautam

Tribhuvan University

Faculty of Humanities and Social Sciences

Asian College of Higher Studies (ACHS)

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by Suman Maharjan entitled "**Job Portal"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the mid evaluation.

SIGNATURE

Govinda Gautam

SUPERVISIOR

Asian College of Higher Studies (ACHS)

Ekantakuna, Jawalakhel

ABSTRACT

The Job Portal System is a secure, user-friendly online platform that connects job seekers and employers by enabling seamless registration, profile creation, resume uploads, job searching, application tracking, and job posting management. It employs reliable CRUD operations for handling dynamic data such as user profiles, job listings, and applications, while using JSON Web Tokens (JWT) to ensure secure authentication and authorization for verified users. The system incorporates features like skill tagging and profile validation to maintain authenticity in recruitment, and plans to include a recommendation engine that matches candidates with suitable jobs based on their skills and preferences. Overall, it provides a streamlined recruitment workflow, strong data security, and a foundation for future enhancements through smart algorithms and data analysis.

Keywords: *Job Portal, Recruitment System, CRUD, JWT Authentication, Resume Upload, Job Matching, Employer Dashboard.*

ACKNOWLEDGEMENT

This project would not have been possible without the valuable guidance, support, and encouragement received from many individuals and institutions. The sincere gratitude goes to **Asian College of Higher Studies (ACHS)** and its faculty members for their constant support, supervision, and for providing the necessary resources and information required for the successful completion of this project.

Special appreciation is extended to the project supervisor, **Mr. Govinda Gautam**, whose expert guidance, insightful suggestions, and kind cooperation greatly contributed to the completion of this work.

Grateful acknowledgment is also made to the seniors, colleagues, and friends whose advice, feedback, and assistance proved to be invaluable during the various stages of the project.

Their collective contributions and encouragement are deeply appreciated.

Contents

LIST OF ABBREVIATIONS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheet
DB	Database
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
JWT	Json Web Tokens
TS	TypeScript
UI	User Interface
UX	User Experience

Table of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

In today's competitive job market, where many individuals vie for limited positions, finding the right job can be quite challenging. That's where job portals come in—they act as helpful platforms that connect job seekers with opportunities that align with their skills and qualifications. These websites streamline the job search process by gathering a wide range of job listings in one place, saving users time and effort.

Job portals also allow individuals to explore opportunities across various fields and locations, making it easier to discover the perfect job. For employers, these platforms provide an efficient way to find talented candidates who match their job requirements.

"JobNepal" is a job portal website designed to help job seekers search and apply for openings based on their qualifications and interests. It offers a seamless system to identify opportunities that closely match users' skills and preferences. Job Nepal not only benefits job seekers but also provides a simplified interface for employers to post openings and connect with potential hires.

According to recent research, 78% of experienced job seekers use the internet to find employment opportunities, compared to 64% of recent graduates. More experienced candidates also tend to register on multiple job portals and rely less on social networks like friends, family, and classmates [1]. The use of job portals continues to grow, with even currently employed individuals using these platforms to seek better opportunities.

Job portals collect job information from various sources, including company websites and direct submissions from employers. This makes it easier for users to find relevant opportunities and for companies to connect with qualified candidates. This collaborative system is transforming how job searches and recruitment are conducted making the entire process more efficient and accessible for everyone involved.

1.2 Problem Statement

Finding jobs through newspapers is often a hectic and time-consuming process, requiring individuals to spend a lot of effort and wait for suitable vacancies. While online job platforms offer more convenience, many of them display outdated listings, making it difficult for users to find current and active opportunities. Additionally, the presence of fake employers and misleading job advertisements creates serious trust and safety concerns for job seekers. Most existing job portals also lack intelligent matching systems, which results in poor job recommendations that do not align well with a user's skills or preferences.

1.3 Objectives

The objectives of the system are:

- To develop a web-based application that efficiently connects job seekers with job openings by smartly matching their profiles.
- To recommend the jobs to job seekers that align with their skills
- To make the job application process easier by allowing job seekers to apply directly through the portal.

1.4 Scope and Limitation

1.4.1 Scope

- Develop a web-based platform for both job keepers and Jobseekers.
- Implement secure authentication to ensure user privacy and protect sensitive information.
- Enable full CRUD operations (Create, Read, Update, Delete) for job postings, allowing employes to manage their job listing.
- Provide profile management features where users can create, edit and maintain their personal and profession details.
- Facilitate role-based access control to separate Job keeper and Jobseeker functionalities.

1.4.2 Limitations

- The application is currently optimized for desktop.
- Automated email notifications are not yet implemented.
- The admin panel for managing users, jobs, and site setting has not been developed yet.
- Recommendation algorithms or intelligent job matching systems are not implemented yet.

1.5 Development Methodology

To build the job portal website, we have adopted the Waterfall model. This approach divides the project into clear, sequential phases: requirement gathering, design, development, testing, deployment, and maintenance. Each phase is completed before the next begins, ensuring a well-structured workflow and defined deliverables at every stage.

The Waterfall model is well-suited for our job portal because the core requirements such as user registration, job posting, resume uploads, and job recommendations are clearly defined from the start. This systematic and disciplined method allows us to thoroughly plan each phase, document progress accurately, and minimize changes during development. By following this model, we aim to deliver a stable, fully functional job portal that meets user needs with minimal risk of mid-project disruption.

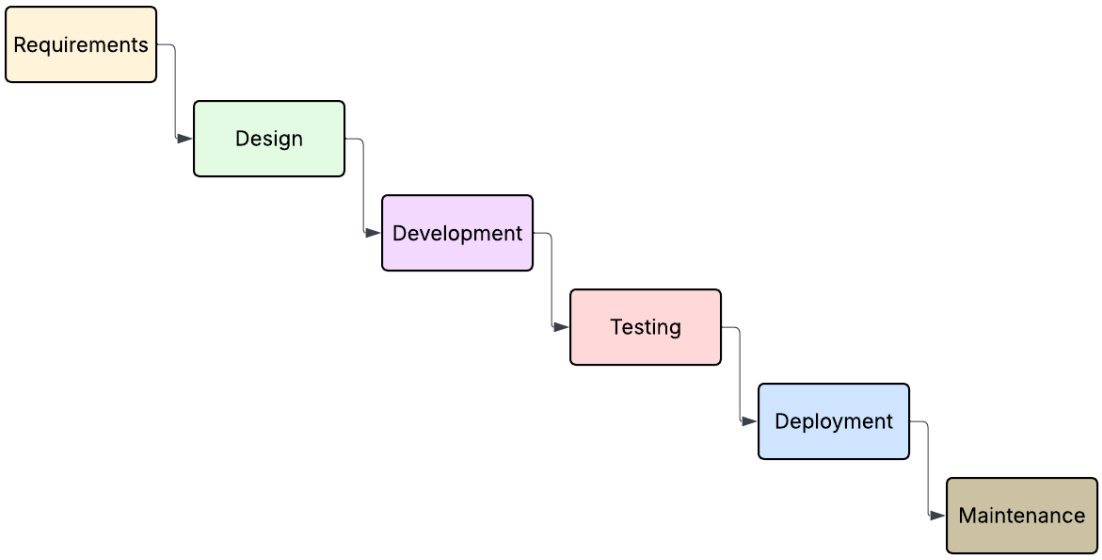


Figure 1.1. 1: Waterfall Model

In the development of the Job Portal system, the first step is **requirement definition**, where all the needs and expectations from both Jobkeepers and Jobseekers are gathered. This includes features like user registration, job posting, application management, and profile handling. Next, in the system and software design phase, a clear and simple design is created to outline

the overall architecture and user interface of the portal, providing a visual and functional blueprint of how the system will operate.

Following the design, the implementation and unit testing phase begins, where developers write the actual code according to the specifications, and individual modules such as job posting, profile management, and authentication are tested for correctness. Afterward, in the **integration and system testing** phase, all modules are combined and tested as a complete system. This is when the portal is shared with stakeholders and potential users for thorough testing to ensure it meets all the initial requirements.

Finally, during the maintenance phase, the Job Portal is actively used by customers. Users report any bugs, missing features, or usability issues. The development team addresses these problems with updates and patches, ensuring the portal remains functional and effective according to user feedback until the system fully satisfies all parties involved.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Job portals have significantly changed the recruitment process by offering a centralized desktop platform where employers and job seekers can efficiently connect and manage applications. Since their inception in the early 2000s, desktop job portals have evolved from basic job boards into sophisticated systems with comprehensive candidate profiles, detailed job descriptions, and application tracking features [2]. The desktop environment allows for richer interfaces and more complex functionalities, supporting in-depth filtering, resume uploads, and employer dashboards that enhance the hiring experience.

Security remains a top priority for desktop job portals, especially in protecting sensitive user data and maintaining secure authentication mechanisms. The implementation of JSON Web Tokens (JWT) provides a robust solution for verifying user sessions while minimizing risks related to sensitive information exposure [3]. This security measure ensures that both job seekers and recruiters can confidently interact with the platform.

Moreover, integrating AI and data analytics into desktop portals is shaping the future of recruitment technology. By collecting structured user and job data, desktop portals can offer personalized job recommendations, improve candidate-job matching algorithms, and optimize the recruitment workflow. While large platforms often have the resources to implement advanced analytics, designing the system with scalability in mind allows smaller portals to gradually adopt these innovations [4].

2.2 Literature Review

Job portals have become an integral part of modern recruitment by providing a centralized online platform where employers and job seekers can efficiently connect, share information, and manage job applications. Over time, job portals have evolved from basic online bulletin boards to sophisticated web-based systems featuring detailed job listings, candidate profiles, and application tracking systems that streamline the entire hiring process [5]. A well-designed job portal can reduce recruitment costs, shorten the hiring cycle, and expand employers' access to a diverse talent pool across regions. For job seekers, such platforms offer convenience and transparency, allowing them to search for relevant jobs, upload resumes, and track their application status in real time. Compared to traditional hiring methods, which relied heavily on physical postings, newspaper ads, and in-person applications, job portals provide faster communication, automated notifications, and broader reach. Despite these advantages, challenges remain, including ensuring data security, protecting sensitive personal information, and providing a user-friendly interface that accommodates users with varying levels of technical skills. Many modern job portals now include desktop-focused interfaces to support advanced admin panels for managing job postings, user accounts, and system analytics [6]. Meanwhile, the rise of AI-powered features and recommendation algorithms helps improve candidate-job matching accuracy and offers personalized suggestions, enhancing user satisfaction and hiring outcomes. However, smaller or developing platforms often face technical and financial barriers in adopting such innovations or integrating mobile support. Looking ahead, trends such as remote hiring, integration with professional social networks, and deeper analytics capabilities continue to shape the evolution of online recruitment. Therefore, creating a secure, scalable, and flexible job portal remains crucial for bridging gaps in the hiring process and improving the overall effectiveness and accessibility of online recruitment for diverse user groups.

Chapter 3: System Analysis and Design

3.1 System Analysis

The **System Analysis** phase serves as a crucial foundation in the development of the Job Portal System, ensuring that all requirements are clearly defined and thoroughly understood before moving forward. In alignment with the Waterfall development model, this phase began with identifying and gathering detailed **functional and non-functional requirements** through research and stakeholder input. A comprehensive **feasibility study** was then conducted to evaluate the technical, operational, and economic viability of the project within the available resources and timeline. Furthermore, the system was modeled using various diagrams including object models, state diagrams, sequence diagrams, and activity diagrams to visualize the structure, behavior, and flow of processes within the application.

3.1.1 Requirement Analysis

The system is designer with specific software requirement, both functional and non-functional, that need to be integrated. The following categories outline these essential requirements.

i. Functional Requirement

Functional requirements define the specific behaviors and features that the Job Portal system must provide to meet user needs. These requirements ensure that the system performs essential tasks accurately and efficiently for both JobKeepers (employers) and JobSeekers (job applicants)

For Jobkeeper:

- Register and Login securely.
- Post a new job vacancy.
- View, edit and delete posted jobs.
- View JobSeekers Who Apply for Jobs
- Edit and manage company profile.

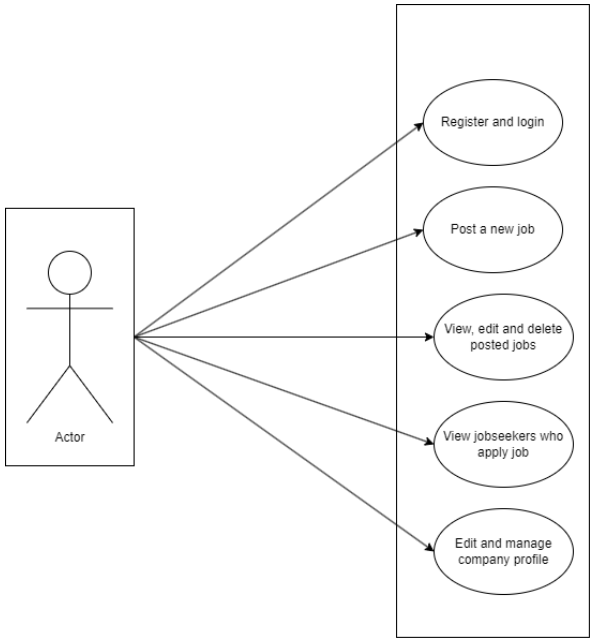


Figure 3.1. 1: Use Case of Job Keeper

For Jobseeker:

- Register and Login.
- Edit personal profile and CV information.
- Explore the job vacancy.
- Browse and view available job postings.

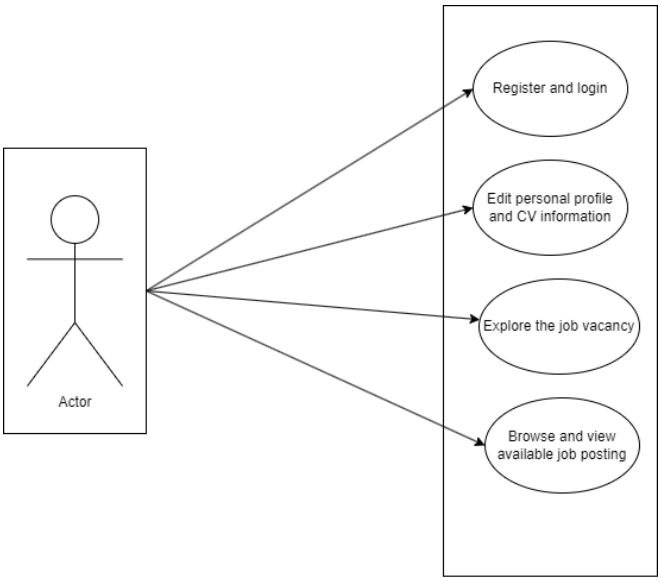


Figure 3.1. 2: Use Case for Job Seeker

For Admin

- Admin registration.
- Manage user accounts.
- Manage posted jobs.
- Monitor system activity.

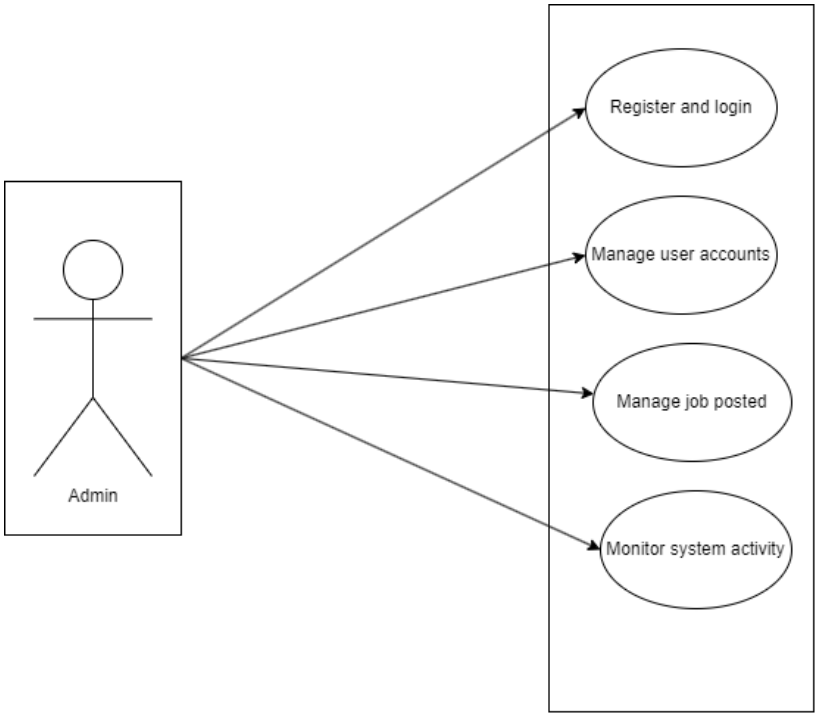


Figure 3.1. 3: Use Case for Admin

ii. Non-Functional Requirements

The job portal system must fulfill several non-functional requirements to ensure it delivers a smooth, secure, and reliable experience for all users.

- Security: All sensitive user and company data are protected with secure authentication, including JWT for session management.
- Performance: The system must handle multiple job postings and user interactions with minimal delay.
- Scalability: Designed to support growth in the number of users and job listings.

- Usability: Simple and intuitive interface for both job seekers and employers.
- Availability: System must be accessible 24/7 expect for minimal maintenance downtime.
- Maintainability: Easy to update modules for adding future features like AI based recommendations.

3.1.2 Feasibility Analysis

A feasibility analysis for the Job Portal project determines whether building and deploying this system is practical and achievable within the available resources, skills, and timeline.

i. Technical Feasibility:

By the technical feasibility of the project is moderate. The system requirements can be classified as follows:

- Hardware Requirements:
  a. 4GB RAM
  b. 128GB HDD/SSD
  c. Basic I/O devices
  d. Intel i3/ Ryzen r3
- Software Requirements:
  a. A Web Browser
  b. Platform: Windows 10 and above.
  c. Frontend: Angular, TypeScript, CSS
  d. Backend: Node.js
  e. Database: MongoDB

ii. Operational Feasibility:

The system's structure is simple for users to understand. Employers and job seekers can easily perform tasks like posting jobs or editing profiles.

iii. Economic Feasibility:

Developed using open-source technologies, minimizing licensing costs. The local server setup ensures cost-effective development.

iv. Schedule Feasibility:

Major modules such as authentication, job CRUD, and profile management are completed. The remaining modules, including JobSeeker browsing, admin panel, and recommendation algorithms, are scheduled to be finished within 1 month.

3.1.3 Object Modelling using Class and object Diagrams

The object modelling defines core entities like **UserProfile**, **Job**, and **AdminPanel**, showing their attributes (e.g., CV, location, DOB) and operations (e.g., postJob(), editJob(), confirmJob()). Class diagrams specify the system structure, while object diagrams illustrate real instances and relationships between users, jobs, and admin controls, ensuring a clear, consistent backend design.

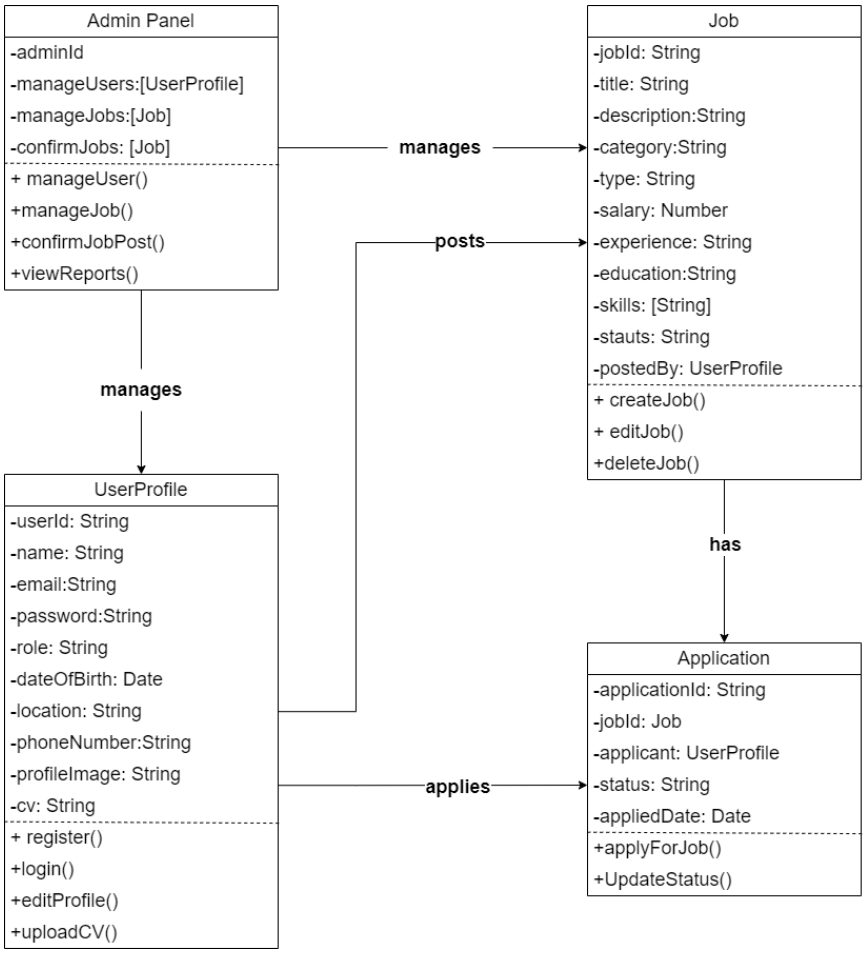


Figure 3.1. 4: Class and object Diagram of Job Portal

3.1.4 Dynamic Modelling using State and Sequence Diagrams

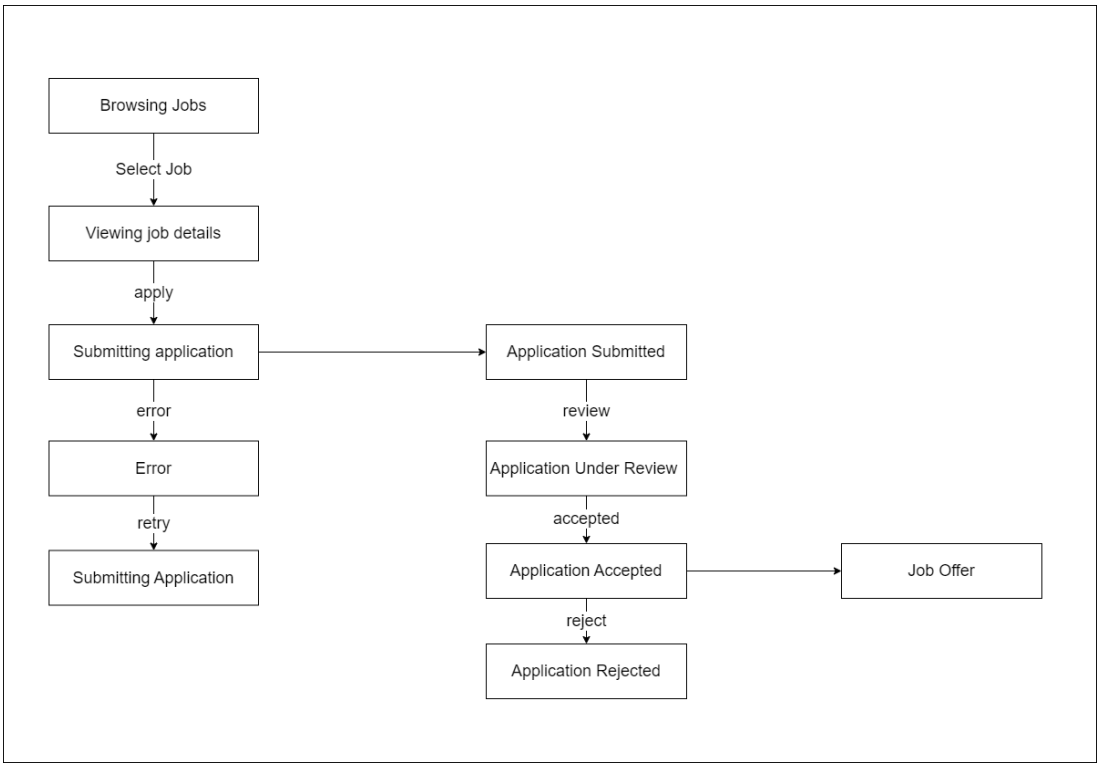


Figure 3.1. 5: State Diagram of Job Portal

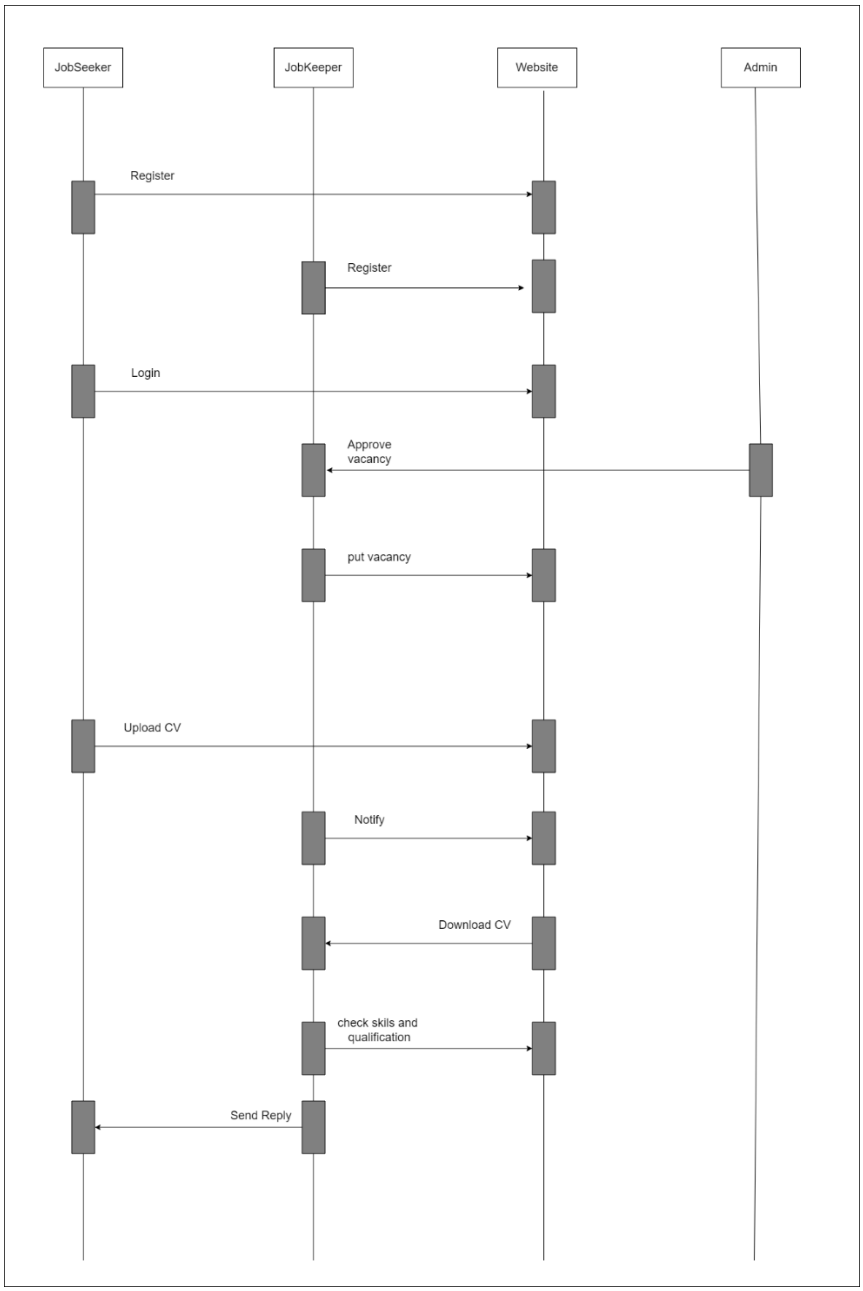


Figure 3.1. 6: Sequence Diagram of Job Portal

3.1.5 Process Modelling using Activity Diagrams

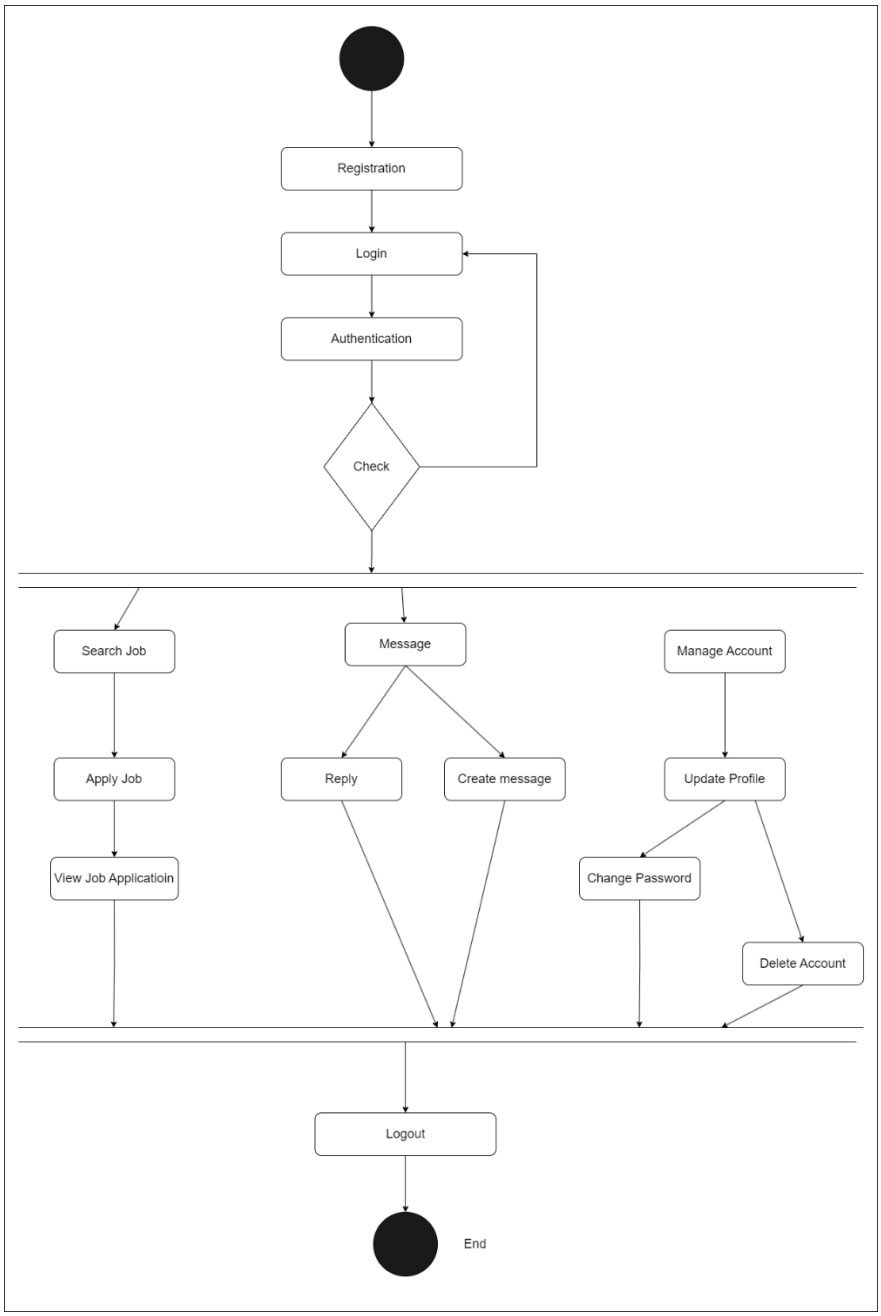


Figure 3.1. 7: Activity Diagram for User

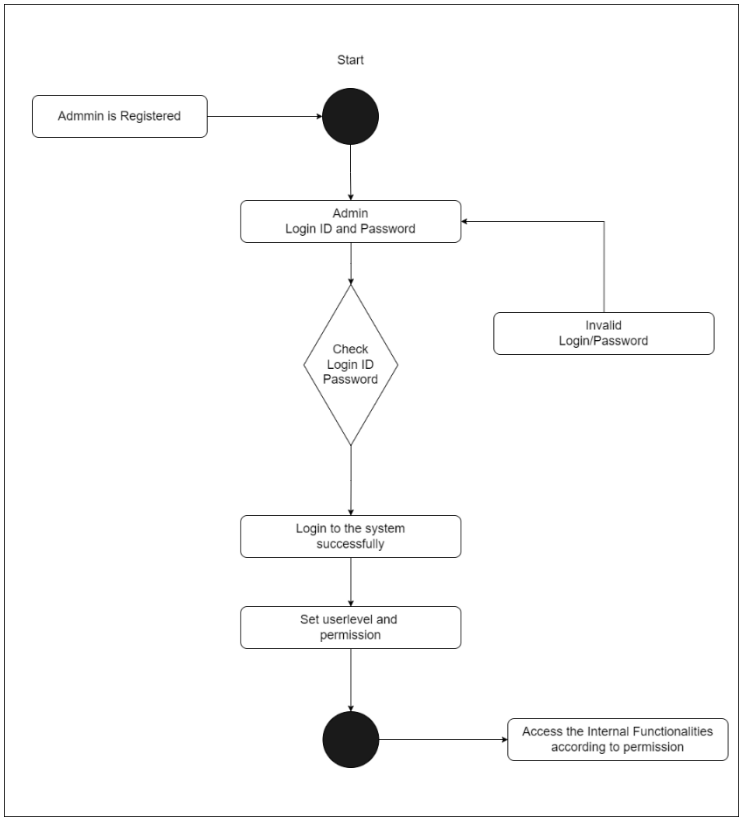


Figure 3.1. 8: Activity Diagram for Admin

3.2 System design

System design for a Job Portal defines the detailed architecture, components, and data flow needed to turn user and business requirements into a working platform. It outlines how modules like user management, job posting, application tracking, and admin controls interact. The goal is to ensure the portal is efficient, scalable, and easy to maintain while meeting all functional and non-functional requirements.

3.2.1 Component Diagrams

The component diagram illustrates the main parts of the Job Portal system and how they interact. Key components include **User**, **Job Post**, **Application**, **Authentication**, **Admin**, and **Database/System**. Connectors show relationships such as **User ID** linking User to Application and Job Post, and **Job ID** connecting Job Post to Application. The **admin** component connects to User, Job Post, and Application components, representing control over user management,

job moderation, and application processing. These connectors represent the flow of data and interactions between components, giving a clear visual overview of the system's architecture.

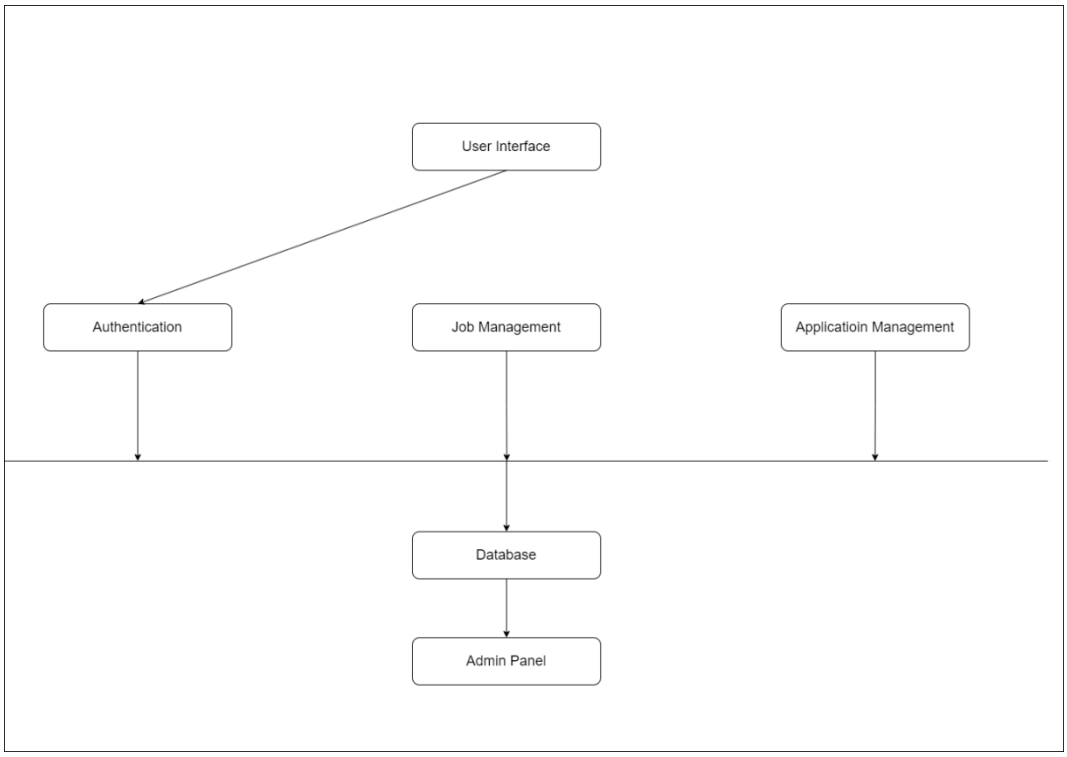


Figure 3.2. 1: Component Diagram for Job Portal

3.2.2 Deployment Diagrams

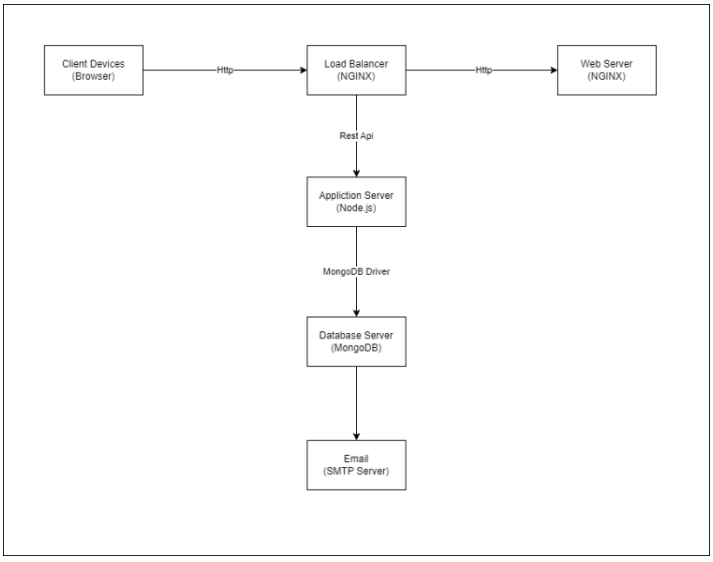


Figure 3.2. 2: Deployment Diagram

3.3 Algorithm Details

- **Content based algorithm**

A content-based recommendation system compares the features of job listings (e.g. required skills, job title, location, experience level) with the features of job seekers (e.g. skills, education, experience, preferences). It calculates a similarity score **S** between the job seeker's profile vector **u** each job listing vector **j.** One common way to compute this similarity is using the cosine similarity formula:

$$S(u,j) = /u /// j /u \cdot j$$

Where u.j is the dot product of the two fracture vectors, and ||u|| and ||j|| are their magnitudes. The system ranks jobs based on the highest similarity scores, recommending those with the most overlapping skills and matching features to the job seeker. This approach help job seekers find jobs that closely fit their skills and preferences.

- **Decision Tree Algorithm**

The Decision Tree algorithm is used here to predict outcomes like expected salary based on user profile inputs such as skills, qualifications, and years of experience. The tree splits the dataset based on features to minimize prediction error. A common criterion for splitting is to minimize the Mean Squared Error (MSE), defined as

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where:

- Y_i is the actual salary for data point *i*.
- $y^_i$ is the predicted salary for data point *i*.
- N is the number of data points in the node.

At each split, the algorithm selects the feature and split point that result in the greatest reduction in MSE, calculated as

$$\Delta MSE = MSEparent - (NNLMSEL + NNRMSER)$$

Where:

- MSE_{parent} is the MSE before the split.
- MSE_L and $MSE_{R\ are}$ the MSEs of the left and right child nodes.
- $N_{L\ and}$ $N_{R\ are}$ the numbers of samples in the left and right child nodes.

Chapter 4: Implementation and Testing

4.1 Implementation

This project follows a well-organized implementation process for the Job Portal system, adopting the waterfall methodology due to its linear and methodical approach. The process progresses through sequential phases: planning, analysis, design, implementation, and testing.

- In the **planning** phase, the project objectives, scope, and key requirements are established.
- During the **analysis** phase, detailed user needs, existing recruitment processes, and system constraints are examined.
- The **design** phase focuses on creating system architecture, database schema, and user interface layouts to support job posting, searching, and application functionalities.
- The **implementation** phase covers the actual development of frontend interfaces, backend APIs, and database integration to enable seamless interaction between job seekers, employers, and administrators.
- Finally, the **testing** phase involves rigorous validation to ensure functionality, security, and usability standards are met before deployment and release.

4.1.1 Tools Used

The following are the tools used for the development of this project.

1. Angular: Angular is a popular open-source front-end web application framework developed by Google. It enables developers to build dynamic, single-page applications (SPAs) using TypeScript. Angular provides a component-based architecture, two-way data binding, dependency injection, and built-in tools for routing and form handling, making it ideal for creating responsive and scalable user interfaces in modern web applications like job portals.
2. TypeScript: TypeScript is a strongly typed, statically compiled superset of JavaScript developed by Microsoft. It adds optional static typing, classes, and interfaces, enabling developers to write more robust and maintainable code. TypeScript compiles down to plain JavaScript, making it compatible with all browsers and environments. It is commonly used

in modern web development frameworks like Angular to enhance code quality and tooling support.

3. NODE.JS: Node.js is an open-source, cross-platform JavaScript runtime built on Chrome's V8 engine that allows server-side execution of JavaScript. It enables developers to build fast, scalable backend services using event-driven, non-blocking I/O architecture. Node.js is widely used to create RESTful APIs and real-time applications, making it a popular choice for backend development in job portal systems.
4. MongoDB: MongoDB is a popular open-source NoSQL database that stores data in flexible, JSON-like documents called BSON. It allows for scalable, high-performance storage and retrieval, making it ideal for handling large volumes of unstructured or semi-structured data. MongoDB supports dynamic schemas, which is useful for evolving data models in applications like job portals where user and job data can vary.

4.1.2 Implementation details of Modules

- Profile Creation Module

When a user creates an account for the first time, they are required to enter name, email, password, confirm password and select the role. This basic information is stored securely to create the user profile. The system ensures password confirmation to avoid mistakes and uses hashing to protect user passwords.

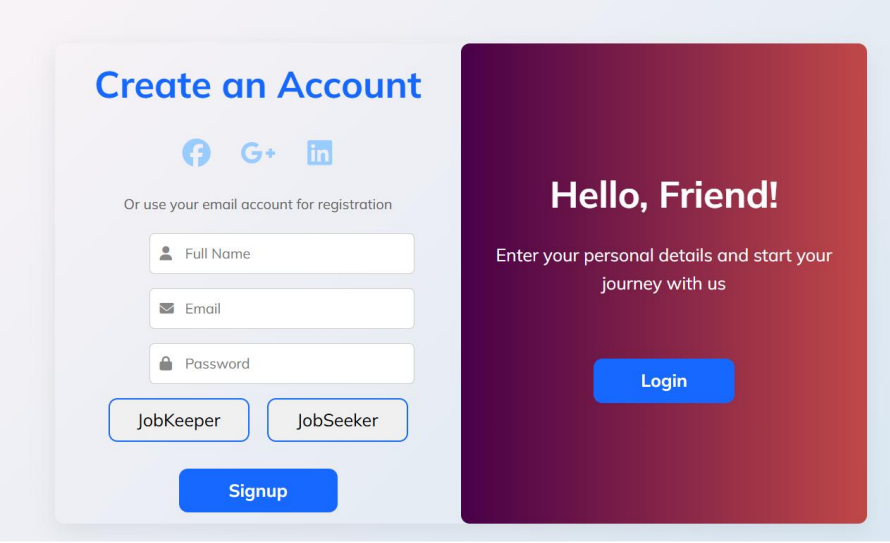


Figure 4. 1: Signup and Login page

- Jobkeeper page module

This module serves as the primary interface for employers (Jobkeepers) to manage their job listings and recruitment activities. It allows jobkeepers to post new job openings, view and update existing job posts, and monitor applications from candidates. The module includes features such as job status management (active, paused, closed), editing job details, and tracking applicant progress.

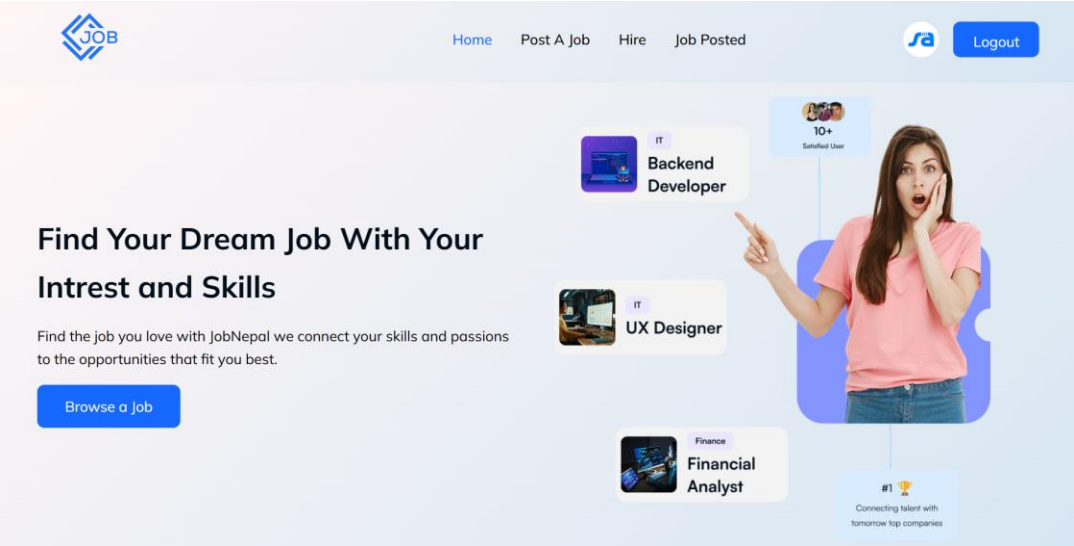


Figure 4. 2: Interface for Jobkeeper

- Post job page module

This module provides employers with a user-friendly interface to create and submit new job listings. It includes form fields for entering job title, description, required skills, experience level, salary range, location, and job type (full-time, part-time, etc.). The module validates inputs to ensure completeness and correctness before submission.

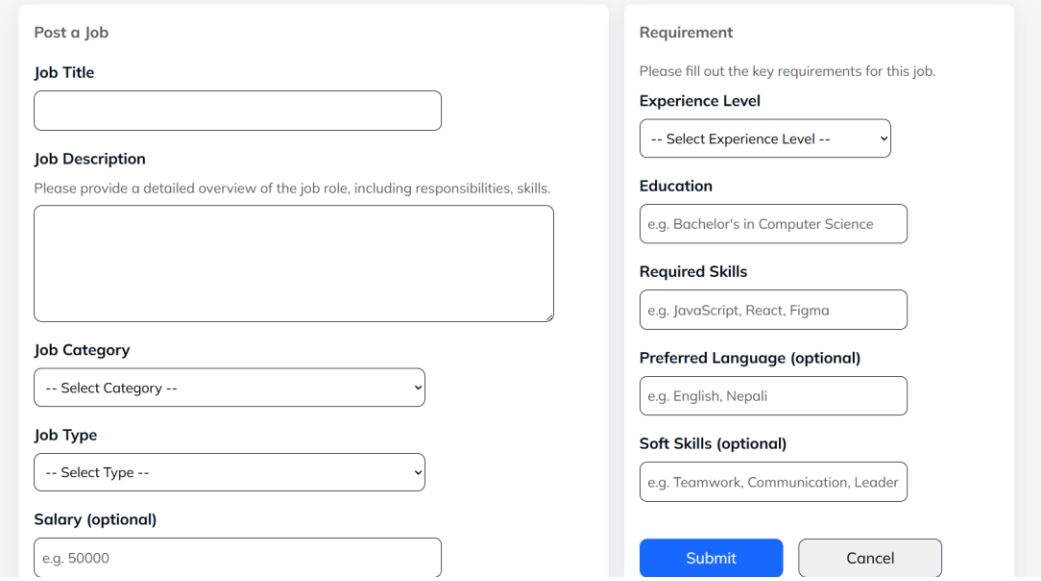


Figure 4. 3: Post a Job page module

- Profile page module

This module allows users both job seekers and job keeper to view and manage their personal information and account settings. For job seekers, it includes details like name, contact info, education, work experience, skills, and uploaded resumes or portfolios. Employers can update company details and contact information.

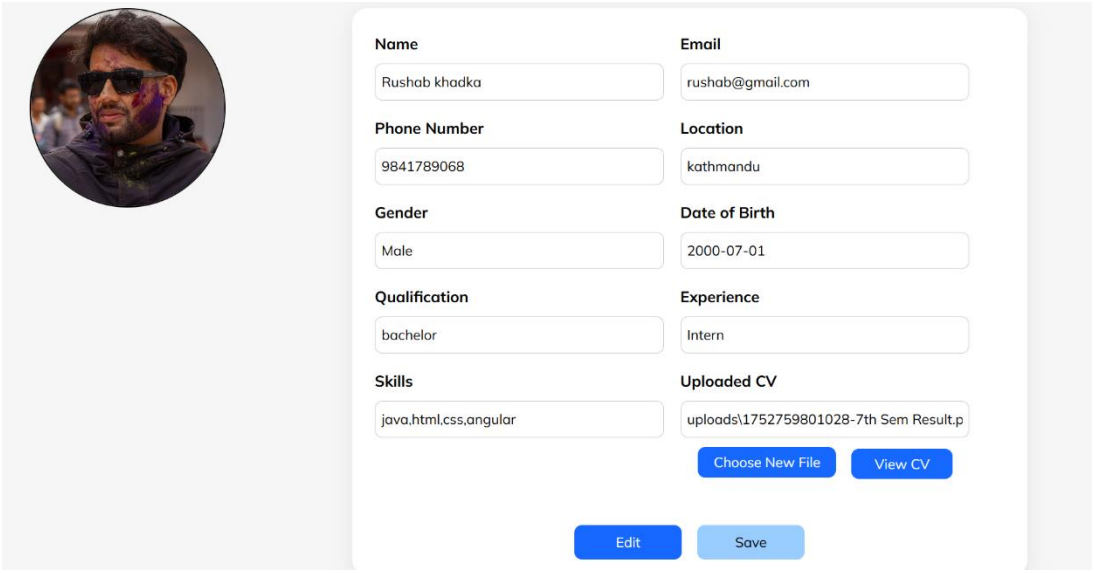


Figure 4. 4: Profile Page

4.2 Testing

4.2.1 Test case for Unit Testing

Table 4. 1: Test Table for Signup and Login

Objectives	To test user registration and login
Test1	
Action	User credential was inserted and login was initiated
Expected Result	Could not login without registration
Actual Result	Could not login without registration
Test 2	
Action	Registration was initiated by filling registration form
Expected Result	Signup Successful
Actual Result	Signup Successful
Test 3	
Action	User credentials were inserted for login
Expected Result	User can now create their profile
Action	User can now create their profile
Conclusion	Test Successful

Table 4. 2: Test Table for profile create

Objectives	To test user profile
Test1	
Action	Fill up the all-user profile information while creating profile
Expected Result	Form inserted and now you can see the landing page and recommended jobs as per your skills
Actual Result	You can see the recommended jobs as per your skills
Test 2	
Action	Trying to submit form without filling all the fields
Expected Result	Ask to fill the particular field
Actual Result	Test Successful
Conclusion	Test Successful

Table 4. 3: Test Table for Job Posting

Objectives	To test job posting on jobkeeper end
Test1	
Action	Fill up the all-job information on job posting form and submitted
Expected Result	Form inserted and you can see your posted job
Actual Result	You can see your posted job
Conclusion	Test Successful

Chapter 5: Conclusion and Future Recommendation

5.1.1 Conclusion

Job portals are important platforms that help job seekers find the right job opportunities and connect with potential employers. With this project, job seekers can easily create accounts and log into the application. They can then explore and apply for jobs that match their skills. The system uses clever algorithms to suggest jobs based on skills. This makes it easier to find suitable jobs. Job keeper also benefits by using the system to find qualified candidates for their job openings. Job seekers can add, update and remove their resumes, and job keeper can post job listing, review resumes and communicate with potential hires. Once employers have filled a position, they can remove job listing from the portal

5.1.2 Future Recommendations

The Future scope includes the integration of chatting application where job seeker can directly contact to the job keeper through the portal. This system can be made more user friendly with better responsiveness.

References

[1] A. Smith, "pewresearch," 19 November 2015. [Online]. Available: https://www.pewresearch.org/internet/2015/11/19/searching-for-work-in-the-digital-era/.

[2] L. Ellis, "wsj," 10 May 2024. [Online]. Available: https://www.wsj.com/lifestyle/careers/ai-job-application-685f29f7?.

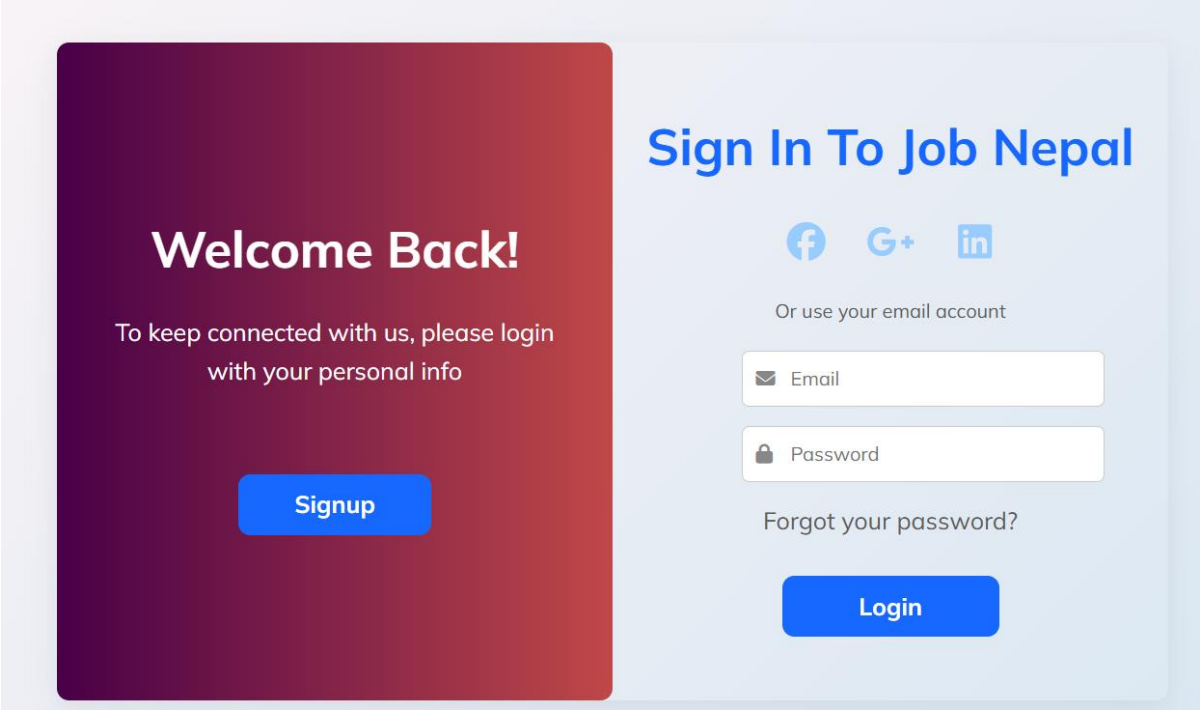
[3] E. Technology, "medium," 28 July 2023. [Online]. Available: https://medium.com/%40extio/understanding-json-web-tokens-jwt-a-secure-approach-to-web-authentication-f551e8d66deb.

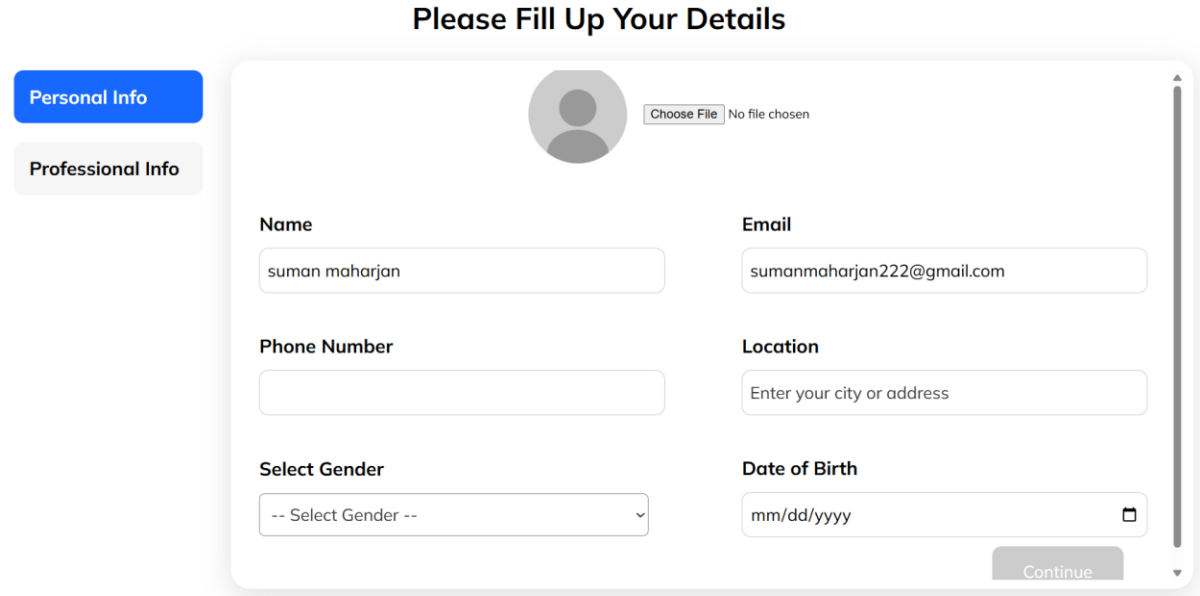
[4] N. Ghai, "economictimes," 12 June 2025. [Online]. Available: https://economictimes.indiatimes.com/jobs/hr-policies-trends/indian-recruiters-pivot-to-quality-hiring-as-ai-tools-take-hold/articleshow/121794666.cms?.

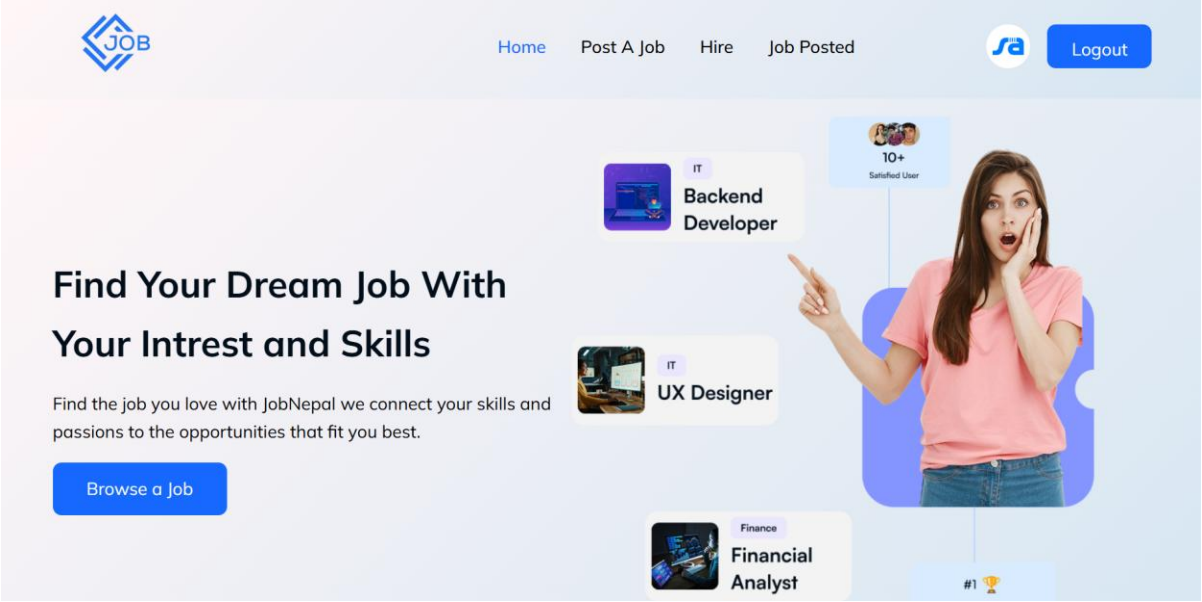
[5] M. Johnson, "Advances in Online Recruitment Platforms," *International Journal of Human Resources Technology,* vol. 15, no. 2, p. 112–128, 2020.

[6] S. &. K. A. Lee, "Desktop Interfaces in Job Portals," *Enhancing User Experience. Journal of Digital Employment Trends,* vol. 9, no. 4, p. 203–215, 2021.

Appendices


Welcome Back!
To keep connected with us, please login with your personal info
Signup
Sign In To Job Nepal
Or use your email account
Email
Password
Forgot your password?
Login


Please Fill Up Your Details
Personal Info
Professional Info
Choose File
No file chosen
Name
suman maharjan
Email
sumanmaharjan222@gmail.com
Phone Number
Location
Enter your city or address
Select Gender
-- Select Gender --
Date of Birth
mm/dd/yyyy
Continue



Name
Rushab khadka

Email
rushab@gmail.com

Phone Number
9841789068

Location
kathmandu

Gender
Male

Date of Birth
2000-07-01

Qualification
bachelor

Experience
Intern

Skills
java,html,css,angular

Uploaded CV
uploads\1752759801028-7th Sem Result.p

Choose New File | View CV

Edit | Save

Post a Job

Job Title

Job Description

Please provide a detailed overview of the job role, including responsibilities, skills.

Job Category

-- Select Category --

Job Type

-- Select Type --

Requirement

Please fill out the key requirements for this job.

Experience Level

-- Select Experience Level --

Education

e.g. Bachelor's in Computer Science

Required Skills

e.g. JavaScript, React, Figma

Preferred Language (optional)

e.g. English, Nepali

Soft Skills (optional)

e.g. Teamwork, Communication, Le

Posted Jobs

Job Id	Title	Category	Type	Salary	Experience	Education	Skills	Status	Actions
283CE5	Frontend Developer	Programming & Tech	Freelance	3000	Entry Level	Bachelor in Computer Background	Html, Css, JavaScript, Angular	pending	