TRIBHUVAN UNIVERSITY

FACULTY OF HUMANITIES AND SOCIAL SCIENCE

Student's Job Portal

A PROJECT REPORT

Submitted to:

Department of Computer Application

Mechi Multiple Campus

Bhadrapur, Jhapa

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

Submitted By:

Anupama Niroula (Reg no.: 6-2-2-42-2020)

Om Prakash Ganesh (Reg no.: 6-2-2-74-2020)

BCA 6th Semester

Under the Supervision of

Mr. Sunil Sharma

TRIBHUVAN UNIVERSITY

FACULTY OF HUMANITIES AND SOCIAL SCIENCE

Mechi Multiple Campus

Bachelor in Computer Applications (BCA)

SUPERVISOR'S RECOMMENDATION

We hereby recommend that this project prepared under my supervision entitled "**Student's Job Portal**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

.....

Project Supervisor

BCA Department

Mechi Multiple Campus

TRIBHUVAN UNIVERSITY

FACULTY OF HUMANITIES AND SOCIAL SCIENCE

Mechi Multiple Campus

Bachelor in Computer Applications (BCA)

SUPERVISOR'S RECOMMENDATION

We hereby recommend that this project prepared under my supervision entitled "**Student's Job Portal**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

.....

Project Supervisor

BCA Department

Mechi Multiple Campus

LETTER OF APPROVAL

This is to certify that this project prepared, which is entitled "**Student's Job Portal**" in the partial fulfillment of requirement for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Sunil Sharma	Raju Paudel
Supervisor	Deputy Director
.....	.....
Internal Examiner	External Examiner

ACKNOWLEDGEMENT

We offer our sincerest gratitude to my supervisor, Mr. Sunil Sharma, who has supported us throughout our report with his patience and knowledge allowing us to work in our own way. One simply could not wish for a better or friendlier supervisor.

We would like to extend our gratitude to director Mr. Raju Paudel for providing us with the all the facilities that was required.

We are also thankful for other faculty and staff member of our department for kind cooperation and help. Many people have helped us during the completion of our report, and we would like to take this opportunity to show our gratitude and respect to all those who made this thesis possible and an unforgettable experience for us.

Lastly, we would like to thank our entire family members and friends for their continuous moral support. We would like to extend our regards to all who have helped our directly or indirectly in accomplishing the study.

Anupama Niroula

Om Prakash Ganesh

ABSTRACT

Student's Job Portal is a user-friendly platform aimed at simplifying the job search and recruitment process for students. Our intuitive interface allows students to effortlessly search for internships and entry-level positions, explore detailed job descriptions, and securely submit applications. Employers can easily post opportunities tailored to student needs and manage applications seamlessly. This system streamlines the recruitment journey, ensuring a smooth experience for both students and employers. Designed with React.js, Material UI and Tailwind CSS for a user-friendly frontend and powered by Node.js, Express.js, and Mongo DB for efficient data management, Student Job Portal connects students with exciting career opportunities quickly and effectively.

Keywords: streamlines, Material UI, Tailwind CSS, Node.js, Express.js, MongoDB

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ERD: Entity Relation Diagram

DB: Database

UI: User Interface

CSS: Cascading Style sheet

DFD: Dataflow Diagram

MERN: Monogo DB Express.js React.js Node.js

Chapter 1

INTRODUCTION

1.1 Introduction

The importance of a Student’s job portal can be understood by considering the significant role of employment in students' lives. Jobs are essential for gaining work experience, funding education, and personal development. In this context, a job portal becomes crucial as it facilitates the availability, accessibility, and management of job opportunities, catering to diverse career aspirations and professional needs of students.

A job portal is vital as it connects students with employment opportunities that align with their skills, qualifications, and career goals. The purpose of this project proposal is to develop a comprehensive job portal that will modernize the process of job searching and recruitment for both students and employers. This job portal provides an online platform where employers can post job openings, and students can search, apply, and manage their job applications. The primary objective of this project is to create an efficient and user-friendly system that improves the overall experience of searching and allocating jobs.

1.2 Problem statement

The traditional methods of job searching and recruitment are often inefficient, relying on scattered job boards, physical classifieds, and word-of-mouth referrals. Job seekers face challenges navigating multiple platforms with inconsistent job listings and outdated information. Meanwhile, employers struggle with managing different job postings, leading to delays in connecting qualified individuals.

The Student’s Job Portal aims to modernize this outdated approach by providing a centralized digital platform. By leveraging technology, the Job Portal bridges the gap between job seekers and employers, facilitating quicker and more effective matches. With a user-friendly interface and robust functionalities, the Job Portal simplifies the complexities of job searching and recruitment. It promotes fair access to employment opportunities, resolves challenges associated with outdated methods, and contributes to a more responsive job market.

1.3 Objectives

- To bridge the gap between students seeking employment opportunities and employers looking to hire young talent.
- To solve the ambiguity in both job recruitment and hiring process.

1.4 Scope and Limitations

The proposed job portal platform aims to bridge the gap between job seekers and employers, providing a seamless and efficient way for users to find job opportunities and manage applications. The platform caters to both job seekers and employers, ensuring a comprehensive and user-friendly experience for all participants in the job market.

1.4.1 Scope

Job Seeker Scope

- Registration and profile management.
- Job searching and filtering.
- View detailed Job's listings.

Employer Scope

- Jobs Listings and management.
- Registration and profile management.
- Handling job inquiries and applications.

1.4.2 Limitations

- Requires web-enabled devices.
- May not be sufficient for all geographical areas.
- Does not handle legal contracts or financial arrangements.
- Require ongoing development and maintenance to address user feedback, add new features, enhance security, and adapt to changing technology or market trends.

1.5 Report and Organization

The report on the Student's Job Portal project is organized into five (5) sections to provide a comprehensive overview of the project, its development process, and the key findings as:

1. **Introduction**: This chapter introduces the concept of the Student's Job Portal and its importance in addressing the challenges of traditional job recruitment processes.

2. **Background Study and Literature Review**: This chapter presents the background and problem statement, highlighting the need for a centralized and user-friendly platform conducting a comprehensive review of existing academic research, articles, publications, and relevant sources that provide insights into the job recruitment industry, related technologies, and best practices.

3. **System Analysis and Design**: This chapter presents an overview of the system design, including the architectural components and data flow and provides details on the database structure, user interface design, entity relationship diagram, data flow diagram, database schema and integration of key functionalities.

4. **Implementation and Testing**: This chapter focuses on the actual development process and the thorough testing methodologies employed to ensure the system's functionality, reliability, and performance including the implementation of system features and the comprehensive testing conducted to validate its effectiveness.

5. **Conclusion and Future Recommendation**: This chapter will summarize the key findings, achievements, and lessons learned from the project and highlight the potential impact and benefits of the Student's Job Portal. It will outline potential future enhancements and recommendations for further development.

Chapter 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Finding a suitable job can be a challenging task for students, often hindered by limited opportunities, lack of transparency, and the absence of effective mediums to connect employers and job seekers. Traditional methods such as classified advertisements, word-of-mouth, and physically visiting companies are often time-consuming and inefficient.

In Nepal, student's job portals are in a developmental phase, showing significant promise yet facing notable challenges. The increase in internet penetration and digital literacy is encouraging more students to seek job opportunities online. Additionally, the growing startup ecosystem in Nepal is creating new job prospects, particularly in technology, marketing, and business development sectors. Government initiatives and efforts by non-profit organizations aim to enhance employability and career readiness among students, further supporting this positive trend.

To address these issues, student job portals leverage modern technologies to create comprehensive solutions that simplify the job search process. By centralizing information and offering advanced search functionalities, these portals aim to transform the way students find and apply for jobs, making the process more efficient, transparent, and convenient for all parties involved.

2.2 Literature review

Here are some of the online job portals that have been serving locally:

1. **MeroJob** [1]: MeroJob is one of the leading job portals in Nepal, offering a section for internships and entry-level positions suitable for students and fresh graduates.
2. **RamroJob** [2]: RamroJob features job postings from diverse industries and includes sections for internships and trainee positions, focusing on entry-level roles suitable for students.

3. **JobsNepal** [3]: JobsNepal offers a wide range of job opportunities including internships and entry-level jobs, catering to both experienced professionals and fresh graduates.
4. **Kantipur Job** [4]: Kantipur Job is associated with Kantipur Publications; this portal provides job listings across various sectors in Nepal, including opportunities suitable for students and recent graduates.
5. **Merorojgar** [5]**:** Merorojgari is a job portal in Nepal that lists job vacancies across various sectors including banking, IT, engineering, healthcare, and more. It also offers job alerts, resumes creation tools, and allows employers to post job openings or search resumes.

While these portals provide valuable resources and opportunities, the focus on mid-level and senior positions can limit the availability of suitable jobs for students and fresh graduates. This highlights the need for more dedicated sections or platforms that specifically cater to the unique needs of the student demographic.

In conclusion, while there are prevalent student job portals in Nepal, there remains a significant opportunity to enhance these platforms. By increasing the focus on internships, trainee positions, and entry-level roles, these portals can better serve the student population. This will not only aid in their career development but also ensure a smoother transition from education to the workforce. Continued innovation and targeted efforts to expand the range of student-specific job opportunities will be key to maximizing the benefits of these platforms for both job seekers and employers.

Chapter 3

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

The requirements of the system are clearly specified, understood, and are known straight. So, it will be feasible to use waterfall model in development of student's job portal. The tools and technologies that are used in the system development are well understood and it is not dynamic. There are plenty of resources and guides are available with required expertise to support the development of the system.

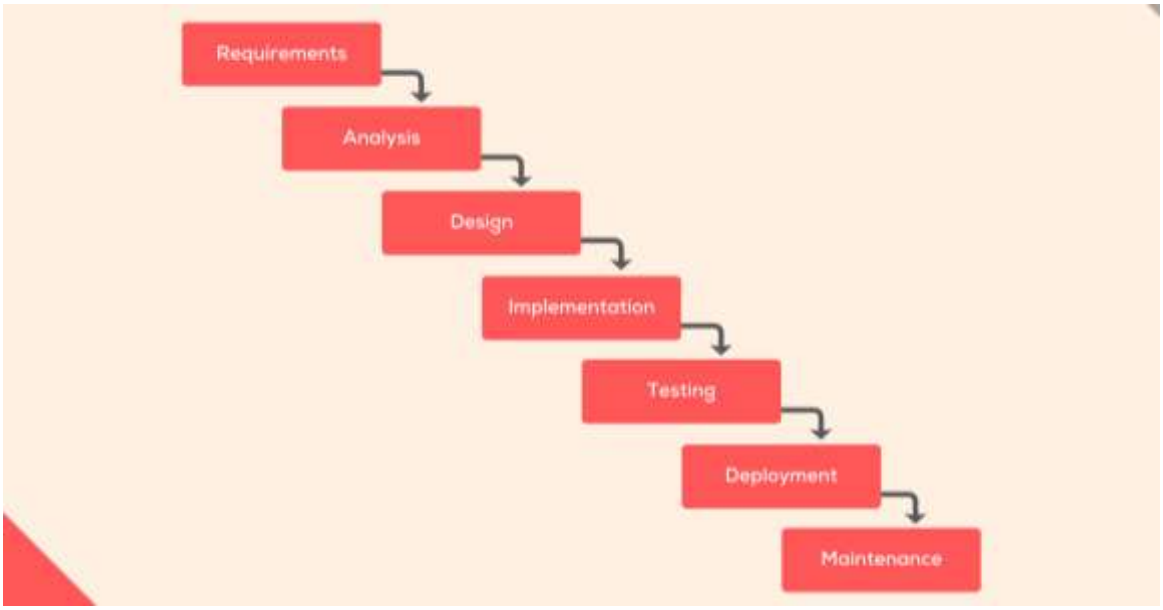


Figure 3.1.1 Waterfall model for Student's job portal

3.1.1 Requirement Identification

Functional Requirements

- User Registration and authentication- The system will provide a user registration process and secure authentication mechanisms to ensure that only authorized users can access and utilize the platform.
- Job Listing- Employers will be directly allowed to make job listings.
- Job Application- Job seekers will be able to apply for job as per preference.
- Application Management - Both job seeker and Employer can manage their applications.

- Specified Type- The system will provide jobs based on type of jobs such as part-time jobs, internships, on-campus job, entry-level positions etc.

Usecase Diagram

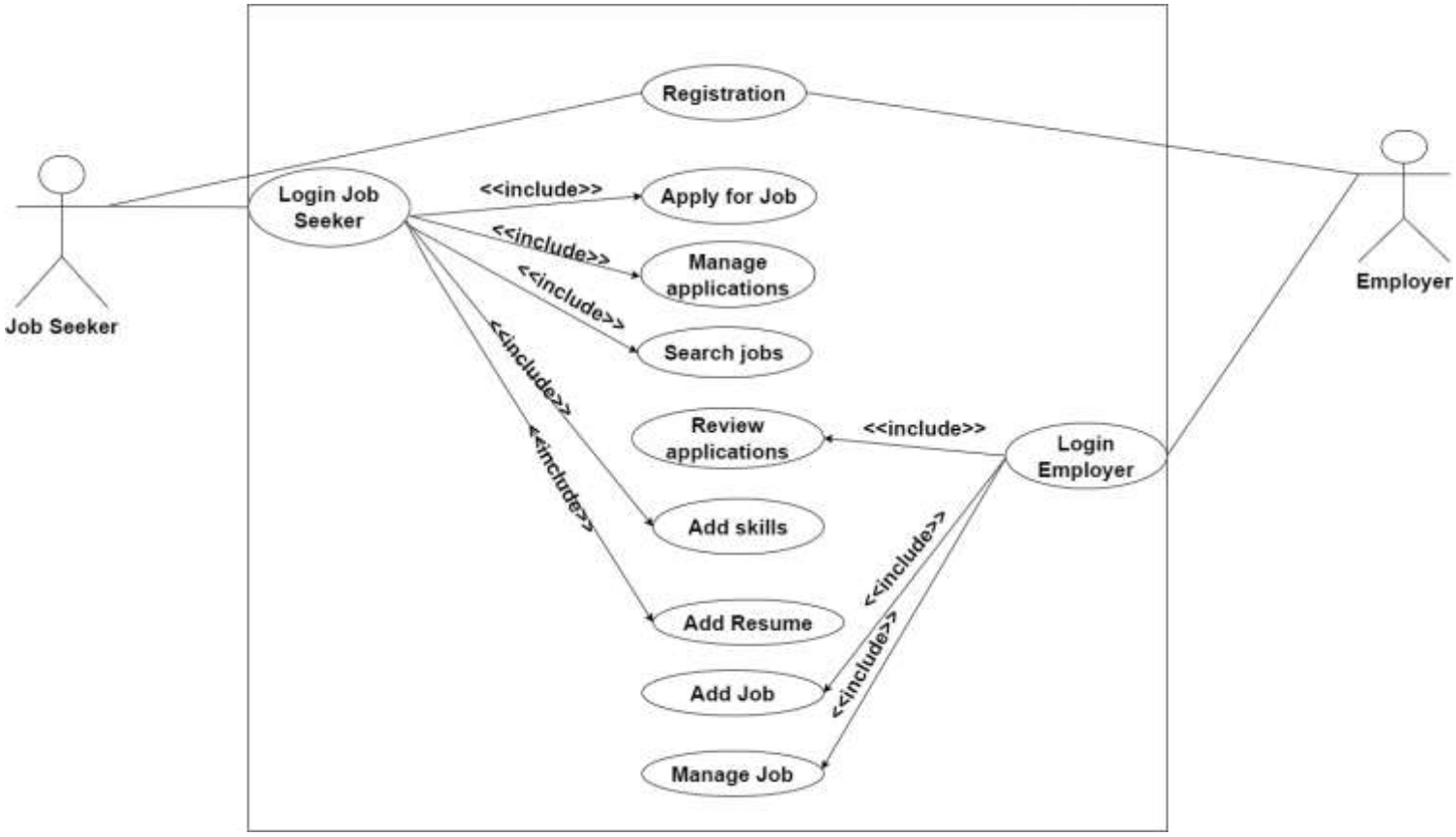


Figure 3.1.1.1 Usecase for Student's job portal

Non Functional Requirements

- Performance: The system must handle a large number of concurrent users without significant slowdowns.
- Security: User passwords must be securely stored using encryption techniques.
- Scalability: The system must be designed to handle increasing user and job data without compromising performance.
- Usability: The user interface must be intuitive, user-friendly, and accessible.
- Reliability: The system must be highly available and minimize slowdown to ensure uninterrupted access for users.
- Availability: The system will be available 24/7/365.

3.1.2 Feasibility Analysis

This project is developed in Windows 11 Operating System using VS Code as code editor and Brave for viewing and debugging the project. Therefore, considering the feasibility of the software in the above environment, the development of system is successfully carried out.

Technical Feasibility

The student job portal has utilized React.js, Material UI and Tailwind CSS for the frontend development, allowing for structured web pages, appealing styles, and interactive features. Node.js and Express.js is used on the backend to handle processing and logic, while Mongo DB stores and retrieves data efficiently. This combination of technologies has enabled the creation of a user-friendly student's job portal, providing a seamless experience for users to browse, apply, and interact with available jobs.

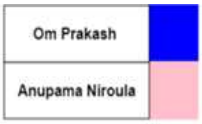
Operational Feasibility

This system addresses the operational needs of users effectively. It solves the problems faced by both job seeker and employer in job allotment process. The system provides a user-friendly interface which enhances the user experience, making it easy for users to browse, book, and interact with available jobs.

Economic Feasibility

Since this system utilizes open-source technologies and free development tools like React.js, Material UI and Tailwind CSS, Node.js, Express.js, and Mongo DB, there is no need for expensive licenses or software. So, it is economically feasible for development.

Gantt chart



Task Name	Duration	Start Date	End Date	July	August	September	Status
Requirement Gathering	5 Days	2024/07/01	2024/07/10				Completed
Analysis	7 Days	2024/07/11	2024/07/17				Completed
System Design	10 Days	2024/07/18	2024/07/28				Completed
Coding	45 Days	2024/07/29	2024/09/11				Ongoing
Testing	15 Days	2024/09/01	2024/09/15				Ongoing
Deployment and Maintenance	7 Days	2024/09/16	2024/09/22				Ongoing
Documentation	65 Days	2024/07/20	2024/09/15				Ongoing

Figure 3.1.2.1 Gantt chart for Student's job portal

System Flowchart

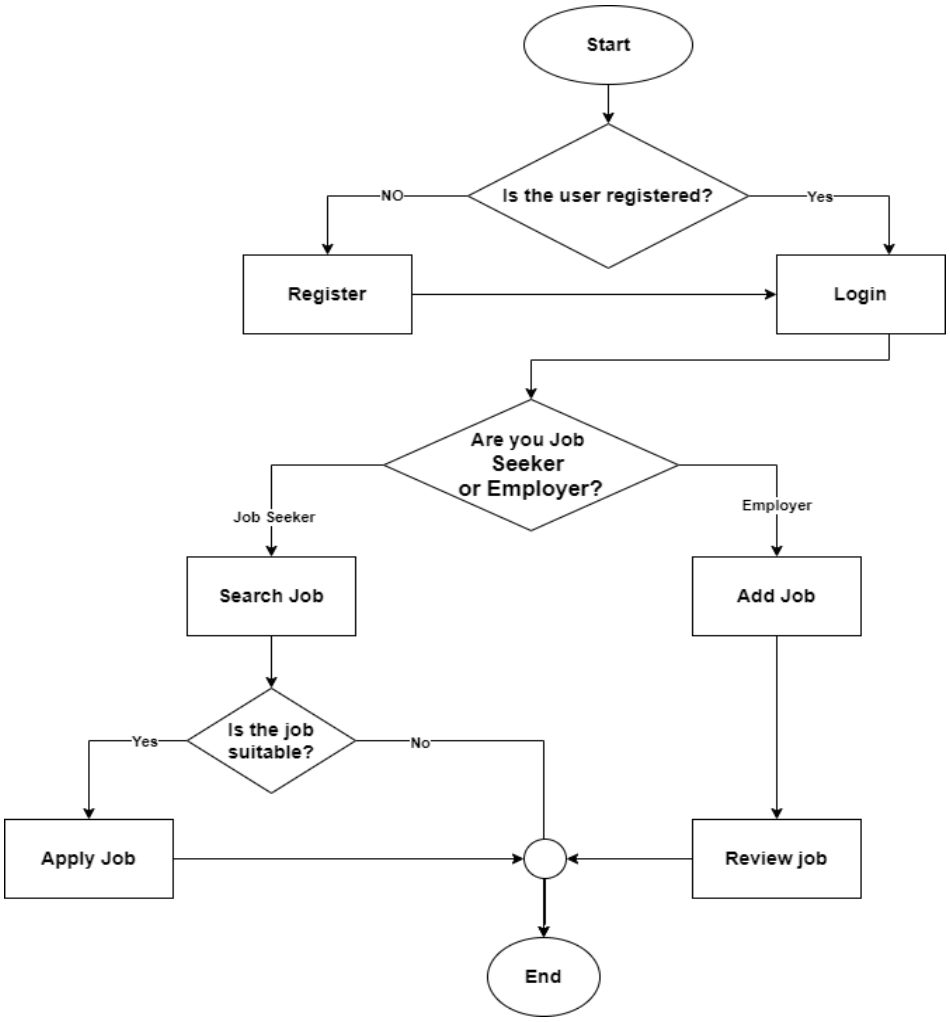


Figure 3.1.2.2 Flow chart for Student's job portal

3.1.3 Data Modeling (ER Diagram)

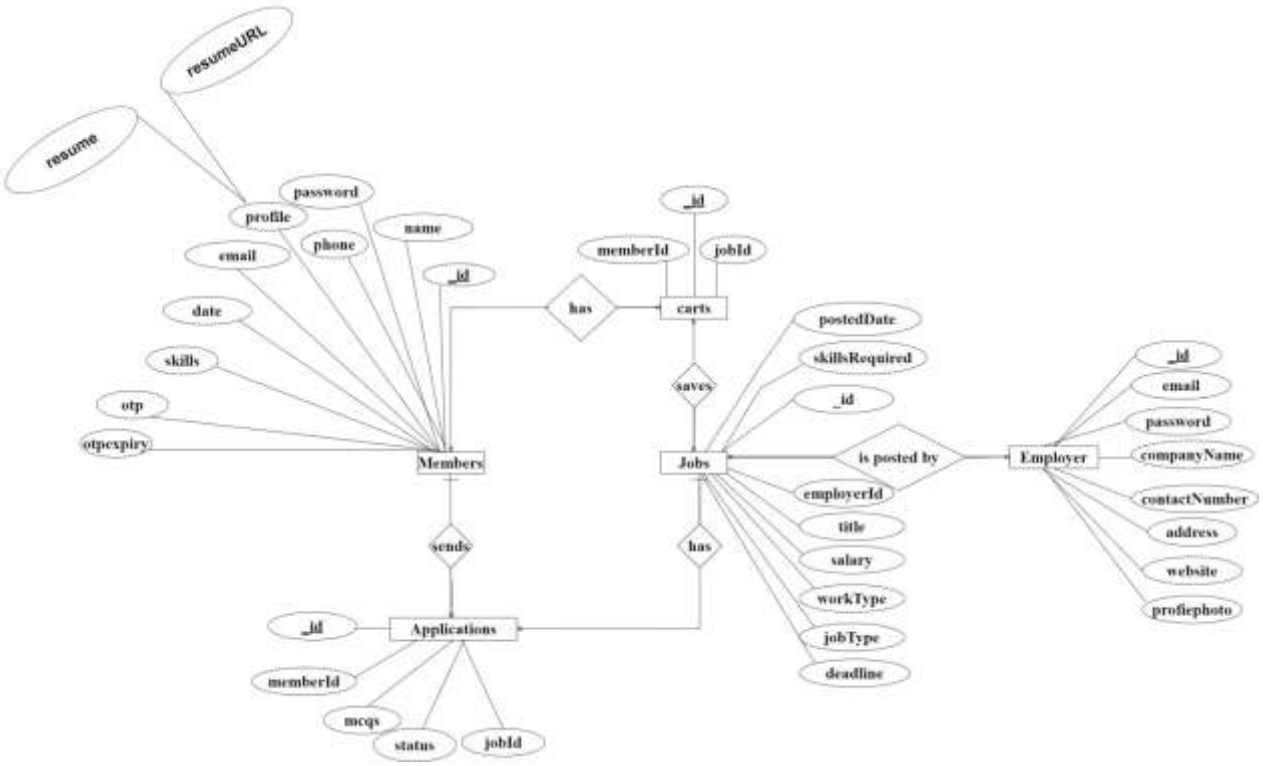


Figure 3.1.3.1 ERD for Student's job portal

3.1.4 Process Modeling (DFD)

In this diagram, we have three main external entities: Job Seeker, Student's Job Portal, and Employer.

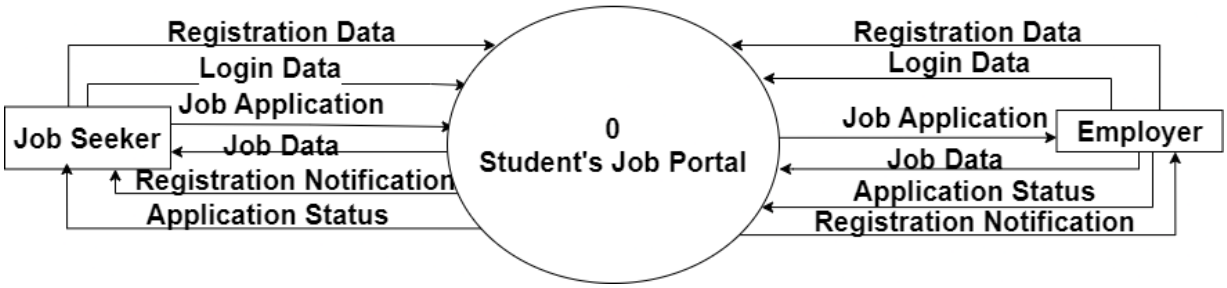


Figure 3.1.4.1 Context Diagram for Student's job portal

The above figure shows the basic functioning of the system where Employer adds the job details into the system as an input and Job seeker sends application for the job added by the Employer. Employer receives the application and accepts or rejects it where job seeker gets the response of approval or rejection as the output from the system.

This context diagram provides a high-level representation of the Student's job portal (Student's Job Portal) and the interactions between Job seekers and Employers. It helps to understand the scope and external entities involved in the system without involving into the internal details of the system's processes and data flows.

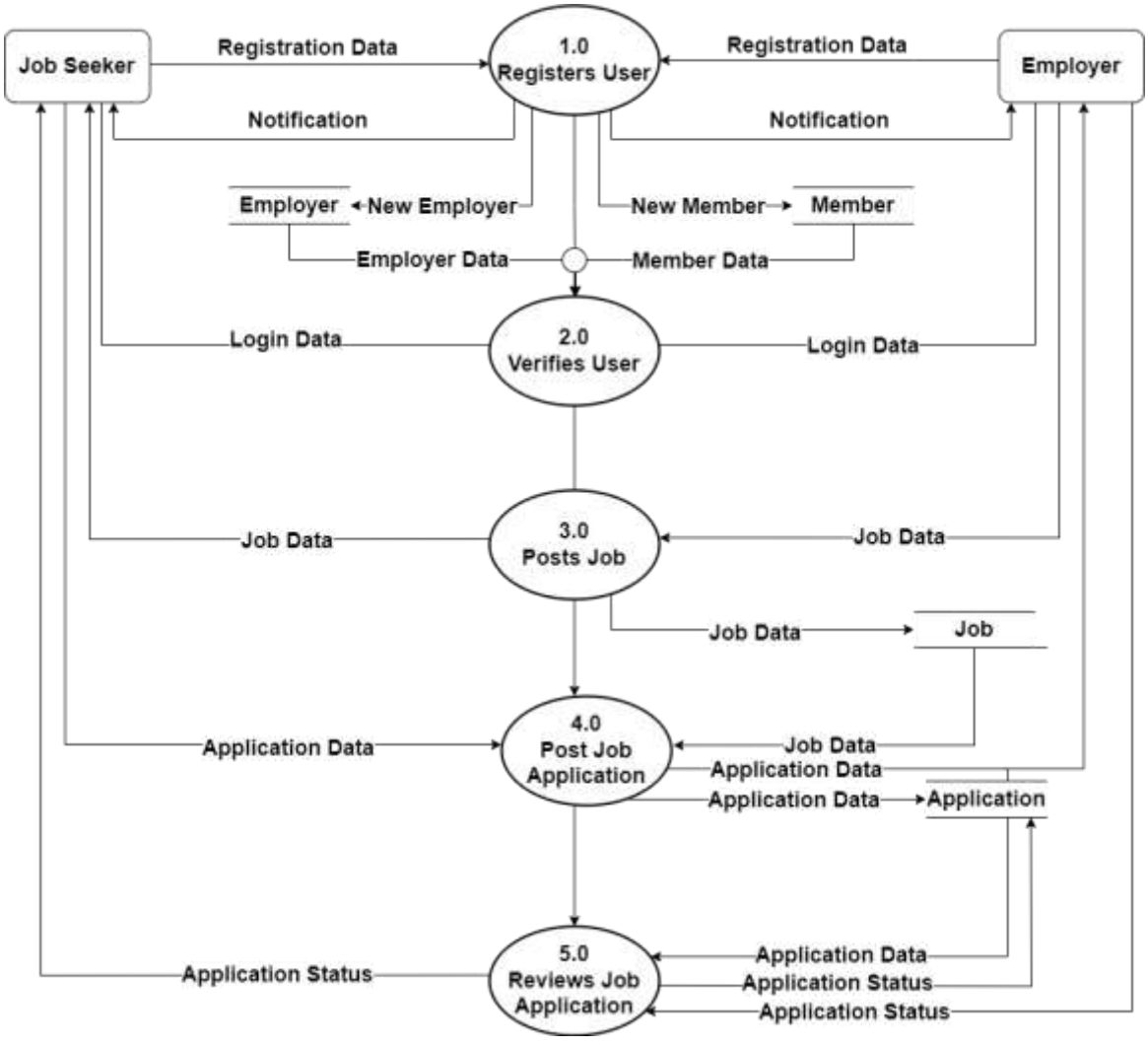


Figure 3.1.4.2 Level 1 DFD for Student's job portal

3.2 System Design

3.2.1 Architectural Design

Architectural design represents the structure of data and program components that are required to build a computer based system, it is not an operational but it is a representation of system. For this system, three tier architecture is used which includes user interface/client tier, web server/web tier and database/data tier. In architectural design, basic structure of the system is shown.

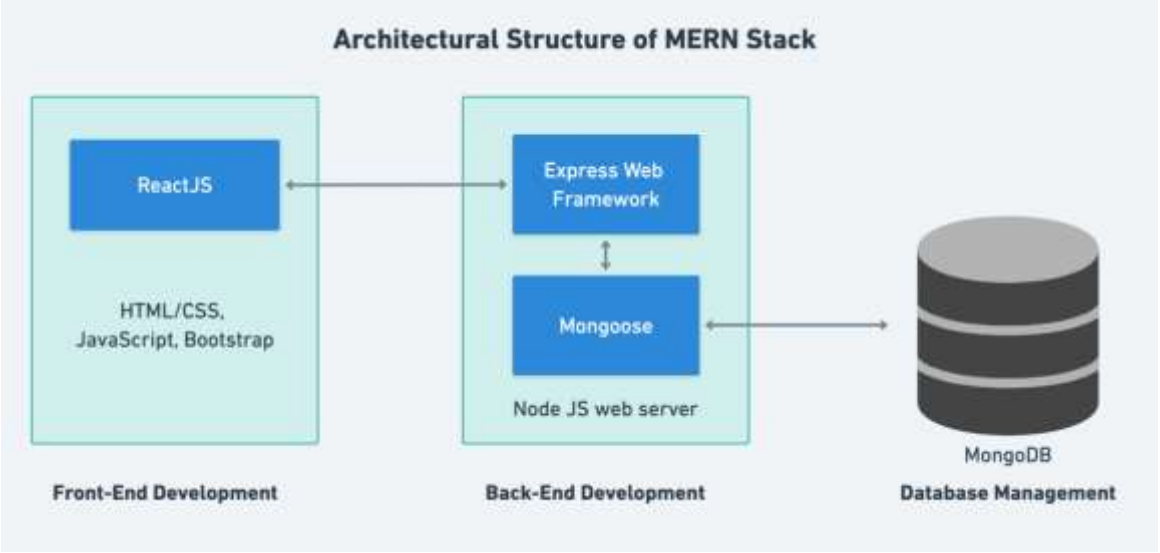


Figure 3.2.1.1 Architecture diagram of Student's Job Portal

3.2.2 Database Schema Design

Figure 3.2.2.1 Schema Diagram for Student's job portal

3.2.3 Interface Design (UI/UX)

The proposed Student's Job Portal is designed to streamline the job search and hiring process for both students and employers. To begin, students and employers create their profiles on the platform. Students can sign up using their student email and fill out profiles with personal details, education and skills. They can also upload their resumes. Employers are able to post job openings and internships tailored for students.

Employers are able to post detailed job descriptions, including job title, responsibilities, qualifications, location, salary range, and application deadlines. They can manage these postings through a dedicated dashboard where they can review applications. This simplifies the job management process and makes it easier to find qualified candidates quickly.

Students search for jobs using various filters such as job type, location, and industry. They can view detailed job descriptions and apply directly through the portal, submitting their resume. This platform tracks the status of applications and provides updates to students, keeping them informed about their progress.

This portal utilizes algorithms to analyze student profiles and job listings, providing personalized job recommendations. This algorithm-driven job matching will help students find opportunities that best fit their skills, experience, and career goals.

3.3 Algorithm Details

Content Based Filtering Algorithm

In the context of job portals, content based filtering algorithms play a vital role in matching job seekers with relevant job listings based on their skills.

Working of Algorithm:

1. Skills extraction: Extract skills from job descriptions and user profiles.
2. Similarity calculation: Compute the similarity between the user's profile and job descriptions.
3. Job Recommendation: Recommend jobs with the higher similarity scores to the user's profile.

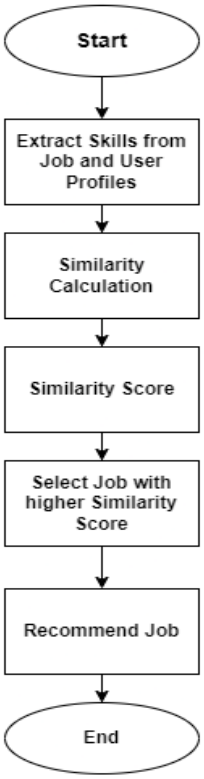


Figure 3.3.1 Flowchart for Content based filtering algorithm

Source Code

```
router.get('/recommendJobsBySkill', CheckLogin, async (req, res) => {
  const memberId = req.user._id;

  try {
    // Fetch user's skills
    const userDocument = await Member.findById(memberId);
    if (!userDocument || !userDocument.skills || userDocument.skills.length === 0) {
```

```
            return res.status(404).json({ success: false, message: 'No skills found for the user'
});
        }

        // Extract skills array from the user's document
        const skillsArray = userDocument.skills;

        // Find jobs where any skill in skillsArray is in skillsRequired and populate
employerId
        const jobs = await Job.find({
            skillsRequired: { $in: skillsArray }
        }).populate('employerId');

        if (jobs.length === 0) {
            return res.status(404).json({ success: false, message: 'No jobs found for the user\'s
skills' });
        }

        // Enhance job details with the number of matching skills
        const jobsWithMatchCount = jobs.map(job => {
            const matchingSkills = job.skillsRequired.filter(skill =>
skillsArray.includes(skill));
            return {
                ...job.toObject(),
                matchingSkillsCount: matchingSkills.length,
                matchingSkills: matchingSkills
            };
        });

        // Sort jobs by matching skills count in descending order
        jobsWithMatchCount.sort((a, b) => b.matchingSkillsCount -
a.matchingSkillsCount);

        res.json({ success: true, jobs: jobsWithMatchCount });
    } catch (error) {
        console.error(`Error recommending jobs by skill: ${error.message}`);
        res.status(500).json({ success: false, message: 'Internal server error', error:
error.message });
    }
});
```

Chapter 4

IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

The above tools are use for the accomplishment of the project:

1. **Front-end Tools**

   React JS: React JS was selected due to its component-based architecture, which facilitates the creation of reusable UI components, promoting efficient development and maintenance of the user interface.

   MUI (Material-UI): MUI was chosen to implement Google's Material Design principles efficiently. It provides a comprehensive set of pre-designed components, ensuring a consistent and professional appearance across the application.

2. **Back-end Tools**

   Node.js: Node.js was utilized because it allows for server-side execution of JavaScript, enabling the development of a full-stack JavaScript application. Its non-blocking I/O model makes it suitable for building scalable network applications.

   Express.js: Express.js was selected for its simplicity and flexibility in building web applications. It offers a robust set of features for web and mobile applications, including routing, middleware integration, and HTTP utility methods.

3. **Database**

   MongoDB: MongoDB was chosen due to its flexibility and scalability. As a NoSQL database, it allows for the storage of JSON-like documents, making it easy to handle and query data. Its schema-less design is advantageous for evolving data requirements.

4. **Documentation Tools**

   Microsoft Word: Microsoft Word is used for writing and editing the documentation because of its powerful formatting and editing capabilities. It ensures that documentation is well-structured and professional.

   Draw.io: Draw.io was selected to generate diagrams for system analysis and design. It offers a user-friendly interface with drag-and-drop functionality, allowing for quick and efficient creation of various diagrams, thereby saving time and effort.

VS Code (Visual Studio Code): VS Code is used for writing code due to its rich feature set, including syntax highlighting, code completion, debugging, and integrated Git support. It enhances developer productivity and code quality.

5. **Version Control**

   Git: Git was chosen as the version control system for this project. It allows for tracking changes in the source code, facilitating collaboration among multiple developers. Git's branching and merging capabilities enable efficient workflow management and ensure code integrity throughout the development lifecycle.

4.2 Testing

4.2.1 Testing cases for unit testing

Table 4.2.1.1 Test case for User Registration

Test case	Scenarios	Input	Expected output	Actual output	Remarks
1	Register user with Filling incorrect Details	Email: anupamaniroula@gmail.com	Registration Unsuccessful	Student's email required with .edu.np.	**FAIL**
2	Register user with Filling correct details	Email: anupama.777402@memc.tu.edu.np Phone Number : 9804560231 Password: 87654321 Confirm Password: 87654321	Registration is successful	Registration success	**PASS**

Table 4.2.2.1 Test case for User Login

Test case	Scenarios	Input	Expected output	Actual output	Remarks
1	Sign in user with filling the incorrect details	Email Address: anupama.777402@gmail.com Enter password: 87654321	Signing unsuccessful	Invalid email and password	**FAIL**
2	Sign in user with filling correct details	Email Address: anupama.777402@memc.tu.edu.np Enter password: 87654321	Signing is successful	Login success	**PASS**

Table 4.2.3.1 Test case for Job Addition

Test case	Scenario	Input	Expected Output	Actual output	Remarks
1	Adding Job details with invalid value in field.	Title: Select Location: Jhapa Salary: 5000 JobType: "On-Site"	Please Enter the valid title of job.	Please Enter the valid title of ob.	**FAIL**
2	Adding Job details with valid value in fields	Title: React Intern Location: Jhapa Salary: 5000 JobType: "On-Site"	Successful job insertion to the system.	Successful job insertion.	**PASS**

4.2.2 Test case for System testing

Table 4.2.2.2 System testing for Student's job portal

Test Case ID	Test Case Title	Objective	Status	Comments
TC-01	User Registration	Verify user can register for an account.	Pass	All functionalities working
TC-02	User Login	Verify registered users can log in.	Pass	No issues encountered
TC-03	Job Search Functionality	Verify users can search jobs	Pass	Accurate results displayed
TC-04	Apply for job	Verify users can apply for a job	Pass	Application status visible
TC-05	Employer Job posting	Verify employee can post a job	Pass	Job visible to users
TC-06	Password recovery	Verify password recovery process	Pass	Recovery mail received

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Lesson Learnt/ Outcomes

Developing a Student's job portal taught us the importance of understanding user needs, managing data effectively, creating a seamless user experience, ensuring scalability and performance, and exploring business opportunities. By incorporating these lessons, we, the project team, have built a user-friendly and trustworthy system that meets the requirements of job seekers and employers, while continuously improving and adapting to their needs.

5.2 Conclusion

In conclusion, the Student's job portal effectively bridges the gap between students seeking employment and employers looking to hire young talent. By addressing ambiguities in the recruitment and hiring processes, the platform offers a seamless and user-friendly experience. Through advanced technologies and modern development practices, we have created a reliable and scalable system tailored to the needs of both job seekers and employers.

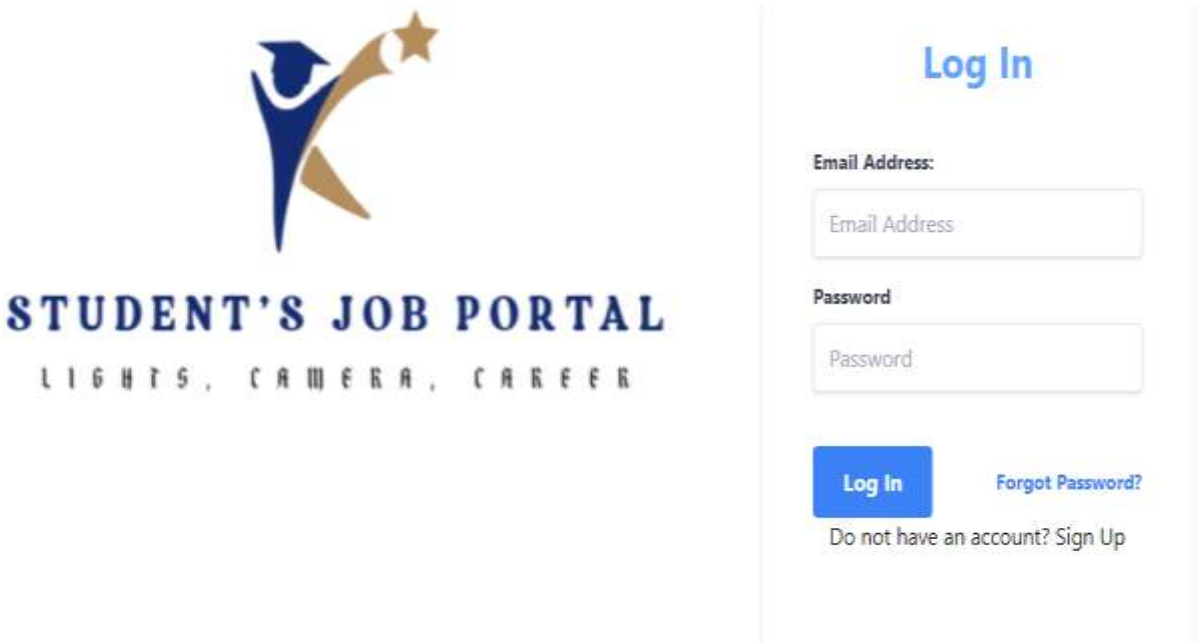
5.3 Future Recommendations

While the Student's job portal has been successfully developed and implemented, there are several future recommendations that can further enhance its functionality and user experience:

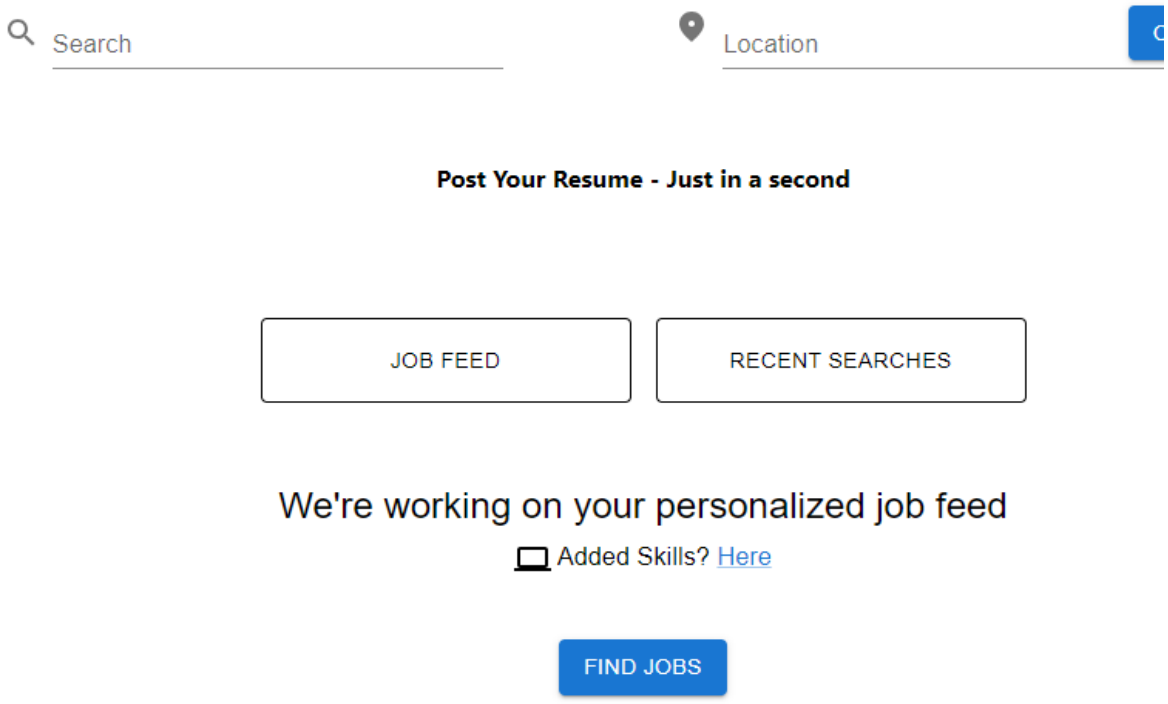
1. Mobile Application
2. Extended resume development interface
3. User Reviews and Ratings
4. Expansion to various Geographical Regions
5. Enhanced Search with Maps

APPENDICES

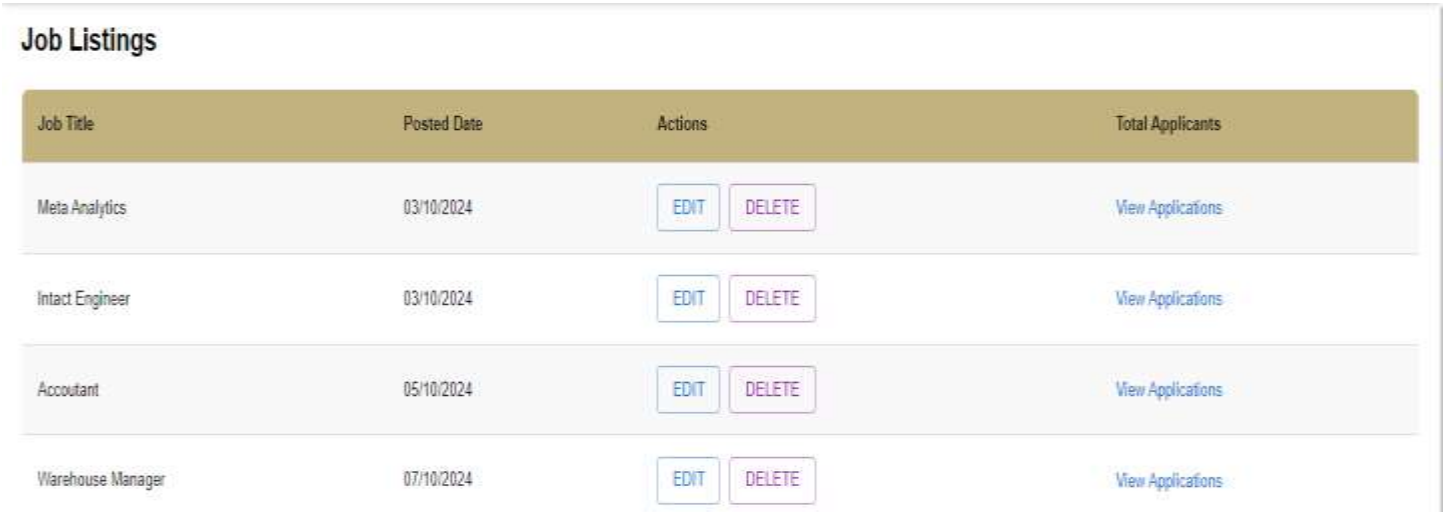
Login Page



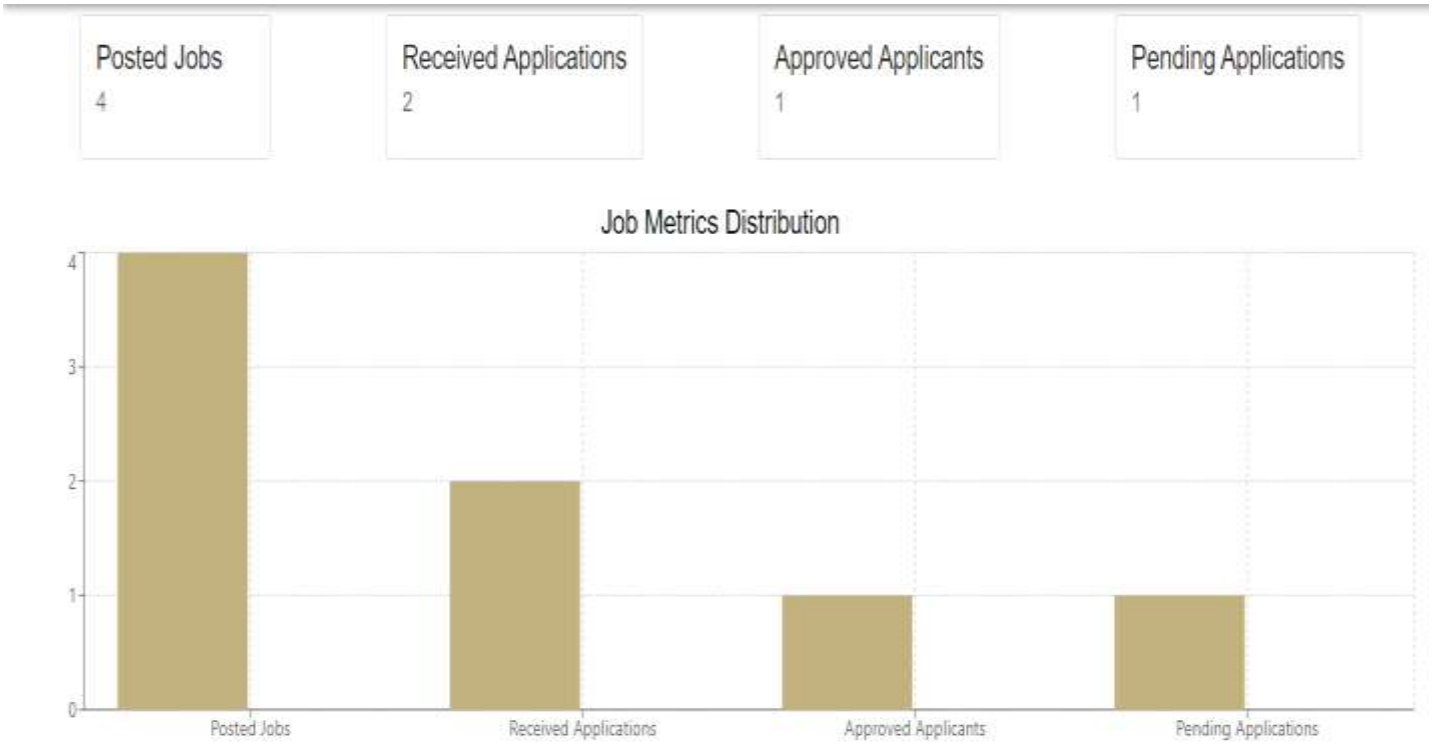
Dashboard



Job Listing



Job Metrics Distribution



REFERENCES

[1] "Jobs in Nepal | Job Vaccancies in Nepal," Mero Job, 2009. [Online]. Available: https://www.merojob.com. [Accessed 26 06 2024].

[2] "JOBS IN NEPAL," Creative Job Pvt. Ltd, [Online]. Available: https://www.ramrojob.com/. [Accessed 26 06 2024].

[3] "Find Jobs, Vacancies, Careers Online," Jobs Nepal, [Online]. Available: https://www.jobsnepal.com/. [Accessed 26 06 2024].

[4] "THE BEST WAY TO FIND A JOB.," Kantipur Management Private Limited, 2012. [Online]. Available: https://www.kantipurjob.com/. [Accessed 26 06 2024].

[5] "Mero Rojgari," [Online]. Available: https://www.merorojgari.com. [Accessed 30 09 2024].