Tribhuvan University

Faculty of Humanities and Social Sciences

JOBHUB: JOB PORTAL PLATFORM

FINAL REPORT

Submitted to Department of Computer Application

Chitwan College of Technology

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

Submitted by

Nov, 2024

ABSTRACT

The objective of JobHub is to create a modern job portal designed to streamline the recruitment process by providing a seamless, user-friendly experience for both job seekers and employers. JobHub offers advanced features such as personalized job recommendations, robust search filters, and real-time notifications, enabling users to find the most relevant opportunities effortlessly. Employers benefit from intuitive job posting interfaces, applicant tracking, and actionable analytics. A standout feature of JobHub is the integration of machine learning algorithms that enhance job matching by tailoring recommendations based on user profiles and preferences, improving the likelihood of successful placements. The platform is built with scalability and security in mind, utilizing cutting-edge tools like React, MongoDB, and Node.js. Deployed with Tailwind CSS for a responsive design, JobHub aims not only to connect talent with opportunity efficiently but also to foster long-term growth through its scalable, data-driven architecture.

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to everyone who supported and contributed to the development of JobHub. First, we are incredibly thankful to Chitwan College of Technology for providing us with the opportunity to pursue this project and refine our skills. Our heartfelt thanks go to our respected mentor, Pravesh Koirala, for their unwavering guidance, insightful feedback, and constant encouragement throughout the project. We are also grateful to all the instructors and faculty members who offered their expertise and valuable insights along the way.

A special thanks to the entire development team for their dedication, collaboration, and relentless hard work in bringing JobHub to life. Additionally, we would like to thank our friends and families for their continuous encouragement and belief in the vision behind JobHub. Their unwavering support was instrumental in helping us complete this project within the given timeframe. Lastly, we extend our sincere appreciation to everyone who contributed, directly or indirectly, to the success of this project.

TABLE OF CONTENTS

LISTS OF ABBREVIATIONS

Admin - Administrator

API - Application Programming Interface

DB - Database

ML - Machine Learning

MERN - MongoDB, Express.js, React, Node.js

UI/UX - User Interface/User Experience

UAT - User Acceptance Testing

JWT - JSON Web Token

MFA - Multi-Factor Authentication

SEO - Search Engine Optimization

REST - Representational State Transfer

CRUD - Create, Read, Update, Delete

DBMS - Database Management System

LISTS OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction

In today's fast-paced digital world, connecting job seekers with the right employers has never been more critical. Introducing **JobHub,** an innovative job portal designed to streamline the recruitment process, making it more efficient, user-friendly, and effective for both job seekers and employers.

JobHub aims to bridge the gap between talent and opportunity by providing a comprehensive platform that caters to diverse industries and job roles. With an intuitive interface, advanced search capabilities, and personalized recommendations, **JobHub** ensures that users can find their ideal job or candidate with ease. Whether you're looking to advance your career or fill a crucial position within your organization, **JobHub** is the ultimate solution for all your employment needs.

1.2 Problem Statement

Despite the abundance of job portals available, many fail to effectively match job seekers with suitable opportunities and employers with ideal candidates. The current platforms often lack personalized recommendations, user-friendly interfaces, and advanced search functionalities. This results in a time-consuming and inefficient recruitment process. **JobHub** aims to address these issues by providing a seamless, efficient, and intuitive solution that enhances the job search and hiring experience for both parties.

- **Inefficient Matching Algorithms**
  Many job portals fail to provide accurate and relevant job or candidate matches, leading to frustration for users who must sift through unsuitable listings.
- **Limited Personalization**
  Existing platforms often lack personalized recommendations, making it difficult for users to find opportunities or candidates that align with their specific skills, preferences, and requirements.

- **Complex User Interfaces**
  The user interfaces of many job portals are not intuitive causing users to spend expensive time navigating the platform instead of focusing on their job search or recruitment.

- **Inadequate Filtering Options**
  Job seekers and employers struggle with inadequate filtering options, which hampers their ability to narrow down searches to most relevant opportunities or candidates.

1.3 Objectives

JobHub aims to address key challenges in job searching and recruitment by offering an advanced platform which is designed to enhance community engagement and streamline user interactions through the following features:

- **To Enhance User Experience**
  Improve **JobHub's** usability and navigation for a seamless experience, featuring a userfriendly interface, simplified job posting, and personalized recommendations based on user preferences.

- **To Optimize Job Matching**
  Utilize advanced algorithms and machine learning to enhance job matching accuracy, ensuring job recommendations align closely with candidate skills, experience, and preferences.

- **To Ensure Data Security**
  Implement stringent security measures, including encryption protocols and secure payment gateways, to safeguard user data and maintain confidentiality throughout the recruitment process.

- **To Provide Analytics and Insights**

Equip employers and job seekers with actionable insights through analytics tools, enabling informed decisions based on application metrics, job market trends, and candidate profiles.

1.4 Scope and Limitation

Scope:

JobHub aims to provide a comprehensive and user-friendly job portal designed to streamline the recruitment process for both job seekers and employers. Key features include advanced job matching algorithms, personalized recommendations, real-time notifications, and intuitive job posting and application management. The platform is suitable for a wide range of industries, catering to entry-level to executive-level job roles. With a focus on data security and seamless user experience, JobHub is built to handle large-scale job searches and application processes efficiently.

Limitations:

JobHub is currently focused on general job searches and may not fully cater to highly specialized or niche job markets such as academic or freelance opportunities. Initially, the platform will emphasize job listings, applications, and basic employer-candidate interactions. Future plan include advanced features such as AI-driven career counseling and deeper employer analytics. While JobHub offers a robust set of features for job matching, certain advanced machine learning features may take time to reach peak efficiency, as they will improve with more user data over time.

1.5 Development Methodology

The development of JobHub follows an Iterative Waterfall Model, allowing for structured progress through distinct phases while incorporating iterative feedback and adaptability to user needs. Key stages of development include:

- **Planning:** Identifying the specific needs of job seekers and employers, defining platform requirements, and outlining key functionalities such as job search, application management, and user profile management.

- **Design and Prototyping:** Creating wireframes and prototypes with a focus on user-centered design principles. This ensures the development of an intuitive, accessible, and visually appealing user interface that enhances the overall user experience.
- **Development:** Implementing the platform using the MERN stack (MongoDB, Express.js, React, Node.js) to ensure seamless integration between the frontend and back-end systems. Tailwind CSS is utilized to create a responsive and modern UI that caters to various devices and screen sizes.
- **Testing:** Carrying out extensive testing, including User Acceptance Testing (UAT), to validate that the platform meets functional and usability requirements. Feedback loops are incorporated to refine features and address any issues that may arise during testing.
- **Deployment:** Utilizing cloud-based deployment solutions for efficient and continuous integration, ensuring smooth deployment of the platform. The process also includes a plan for regular updates and improvements based on user feedback and evolving requirements.

1.6 Report Organization

This report consists of six chapters that cover the design and development of JobHub.

Chapter 1: Introduction – Provides a comprehensive overview of the project, including its objectives, problem statement, and the methodology followed during the development of JobHub.

Chapter 2: System Analysis – Analyzes the requirements of job seekers and employers, discusses the feasibility of the platform, and outlines the overall architecture of the JobHub system.

Chapter 3: System Design – Details the design process, key user interface decisions, wireframing, and the technology stack (MERN, Tailwind CSS) used for developing the platform.

Chapter 4: System Implementation – Explains the actual development process, including feature implementation, backend development using Node.js and Express, frontend development with React, and database management with MongoDB.

Chapter 5: Testing and Results – Describes the testing procedures, including User Acceptance Testing (UAT), performance testing, and the feedback collected from users. The chapter also discusses the results and adjustments made based on these tests.

Chapter 6: Conclusion and Future Work – Summarizes the project, evaluates its success in meeting the objectives, and outlines potential future enhancements and features to improve JobHub further.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

In order to understand the current social platform, we conducted an analysis of an existing platform, **LinkedIn, Indeed, Glassdoor.** Our analysis focused on the user interface, functionality, and features of the platform. Through our analysis, we identified several areas where the existing system could be improved. These include:

- **Enhanced Personalization:** Existing platforms like LinkedIn, Indeed, and Glassdoor could improve by offering more personalized job recommendations and content tailored to the specific. This could involve refining algorithms to better understand user preferences based on their interaction history and providing more relevant job.
- **Improved User Interface and Navigation:** There's a potentials for platforms such as LinkedIn, Indeed, and Glassdoor to enhance user experience by optimizing their interfaces for easier navigation and accessibility. This could include simplifying the job search process, improving filter options, and streamlining the application process to make it more intuitive and user-friendly.

2.2 Literature Review

We reviewed several existing freelancing platforms to understand their strengths and weaknesses in order to better position JobHub in the competitive recruitment industry. Platforms like Upwork and Freelancer were analyzed for their job matching features, user interface, and overall usability.

LinkedIn, traditionally a professional networking site, has ventured into freelancing opportunities through its job listing features. While its integration with professional profiles enhances credibility, the platform's job search functionality often lacks specificity, leading to irrelevant job suggestions for freelancers [1].

Indeed, primarily known for traditional job listings, has introduced freelance opportunities but still falls short in its ability to cater to the unique needs of freelancers. The platform's focus on full-time positions and limited filtering options for freelance work can create inefficiencies for users seeking flexible job arrangements [2].

Additionally, we looked into Glassdoor, which emphasizes employer reviews and salary transparency in the freelance domain. While this feature adds value by informing freelancers about potential clients, the platform can sometimes be challenging to navigate, particularly when searching for freelance opportunities versus traditional job listings, creating a fragmented user experience [3].

We also examined platforms like Upwork stands out for its diverse range of job categories and robust project management tools, allowing freelancers to showcase their skills effectively. However, some users report that the bidding process can be overwhelming, with many low-budget projects that may not align with their expertise, leading to dissatisfaction [4].

Moreover, Freelancer offers a similar bidding system but lacks the same level of project quality control as Upwork. While it provides a large user base and various job opportunities, users often find that the lack of strict vetting for projects can result in lower-paying jobs and unreliable clients [5].

Additionally, we explored Fiverr, which allows freelancers to offer specific services at fixed prices. This platform's user-friendly interface and clear pricing model appeal to both freelancers and clients. However, the limited scope of services may deter professionals with specialized skills from engaging, indicating a potential gap that JobHub can fill by offering a broader range of project categories [6].

Finally, we also examined platforms like PeoplePerHour, which focuses on connecting freelancers with small businesses. While it emphasizes local job opportunities, its smaller user base compared to larger platforms can restrict the availability of highquality projects, suggesting an opportunity for JobHub to attract a more extensive network of freelancers and employers [7].

CHAPTER 3 :SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

System analysis is an essential part of understanding the needs and structure of JobHub. By analyzing both functional and non-functional requirements, we can ensure that the platform delivers the expected user experience.

3.1.1 Requirement Analysis

i. **Functional requirements (illustrate using Use-case diagram/Use-case description)**

- **User Registration and Login:** Users (job seekers and employers) can register and log in using email.
- **Profile Customization:** Users can create and edit profiles, including personal info, resumes, and job preferences.
- **Job Posting and Application Management:** Employers can post job listings, and job seekers can search, filter, and apply for jobs.
- **Personalized Job Recommendations:** Job seekers receive tailored job suggestions based on their profiles and interactions.
- **Job Application Tracking:** Job seekers can track their application statuses, while employers can manage incoming applications.
- **Analytics for Employers:** Employers can access analytics on job postings, such as application numbers and demographics.

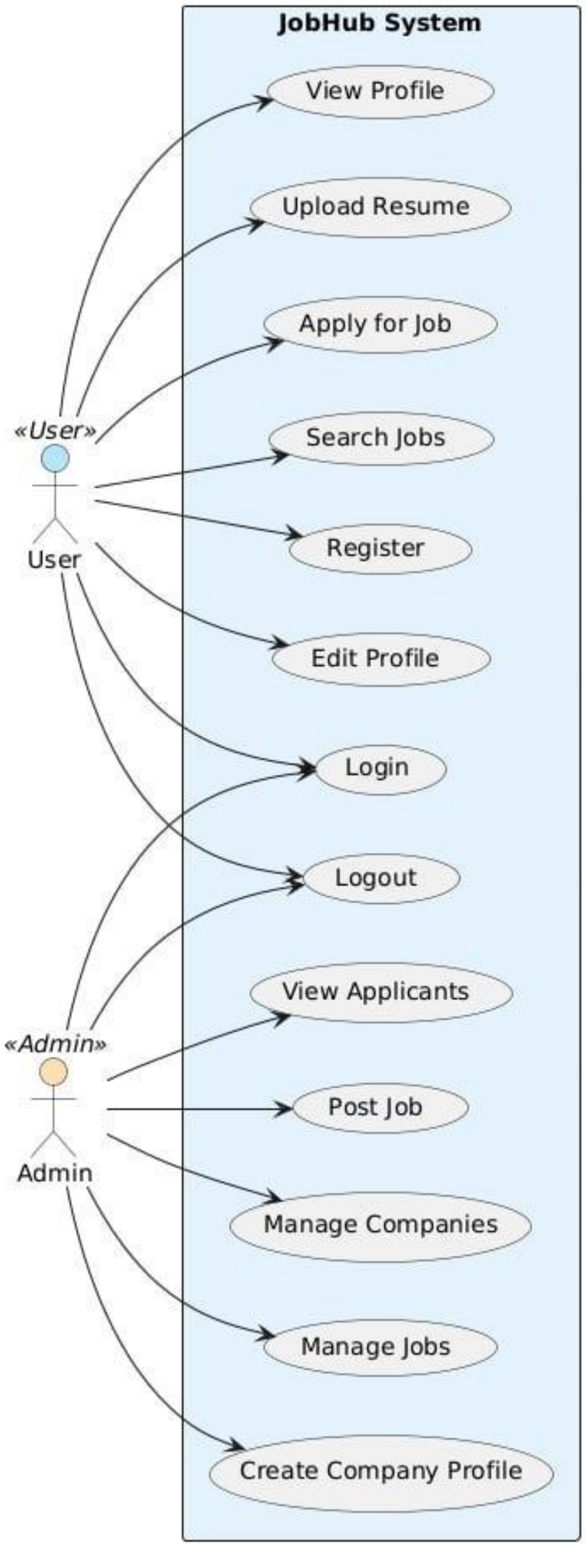


Figure 3-1: Use Case Diagram

ii. **Non-functional requirements**

The non-functional requirements of JobHub ensure that the system performs well and provides a reliable and secure experience for users.

- **Performance**: The platform should load within 2 seconds under normal conditions and handle up to 1,000 concurrent users without significant degradation in performance.
- **Scalability**: The system must be scalable to accommodate future growth in user base and data load, with the ability to expand both front-end and back-end resources as necessary.
- **Security**: JobHub must ensure data security through encryption of sensitive information (such as resumes and application details) and secure user authentication (OAuth for login, MFA for additional security).
- **Usability**: The platform must provide an intuitive and accessible user interface that requires minimal training for both job seekers and employers.
- **Reliability**: The system should have an uptime of 99.9% and provide robust data backup and recovery options.
- **Maintainability:** The system should be built in a modular way so that updates and maintenance can be done with minimal impact on users.

3.1.2 Feasibility Analysis (Technical, Operational, Economic)

JobHub was evaluated based on the following criteria:

Technical Feasibility: **JobHub**, a job portal built on the MERN stack (MongoDB, Express.js, React, Node.js) with Tailwind CSS, offers strong technical feasibility. This tech stack combination not only supports seamless data flow and real-time updates but also enables rapid development and scalability, making **JobHub** well-suited for handling extensive user interactions and data processing typical of a job portal.

- **Operational Feasibility**: **JobHub's** operational feasibility is anchored in its comprehensive approach to user registration, job listing management, and community engagement features. **JobHub** prioritizes seamless user experiences through intuitive interfaces and efficient backend systems.
- **Economic Feasibility**: **JobHub**, a job portal built on the MERN stack and Tailwind CSS, centres on its cost-effectiveness and potential for financial viability. By leveraging open-source technologies like MongoDB, Express.js, React, and Node.js, alongside Tailwind CSS for efficient frontend design, **JobHub** minimizes upfront development costs while maximizing scalability and performance.

3.1.3 Object Modelling: Object & Class Diagram

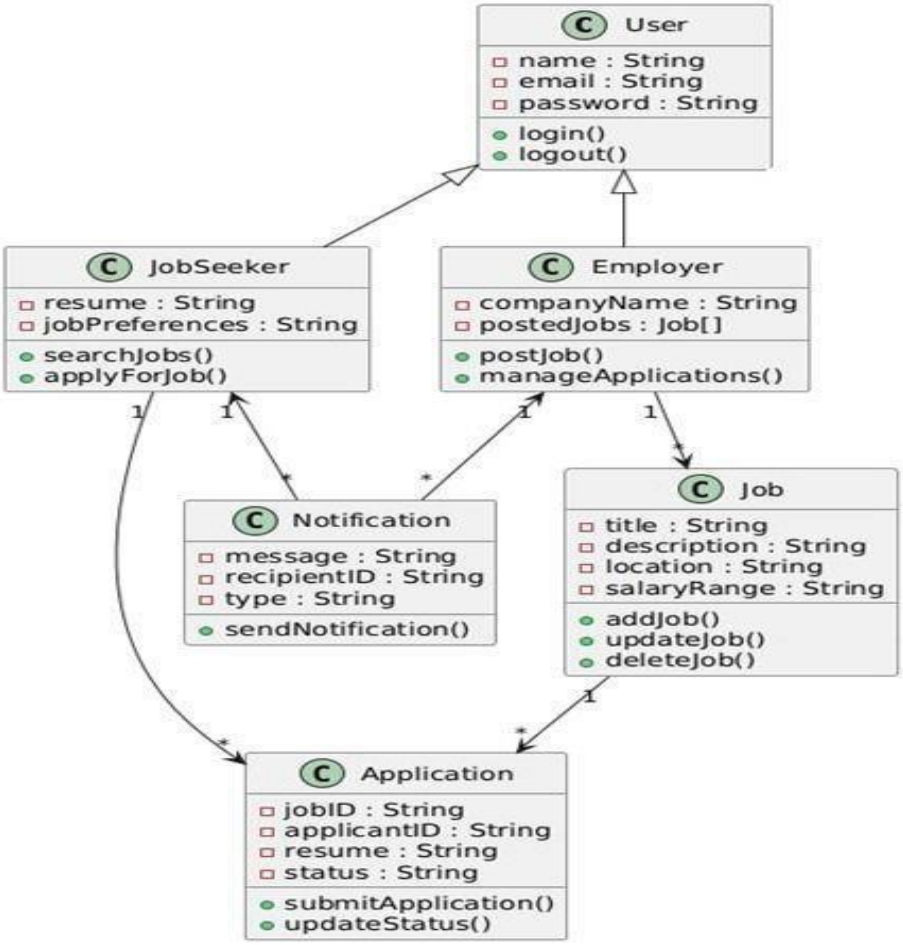


Figure 3-2: Object & Class Diagram

3.1.4 Dynamic Modelling: State & Sequence Diagram

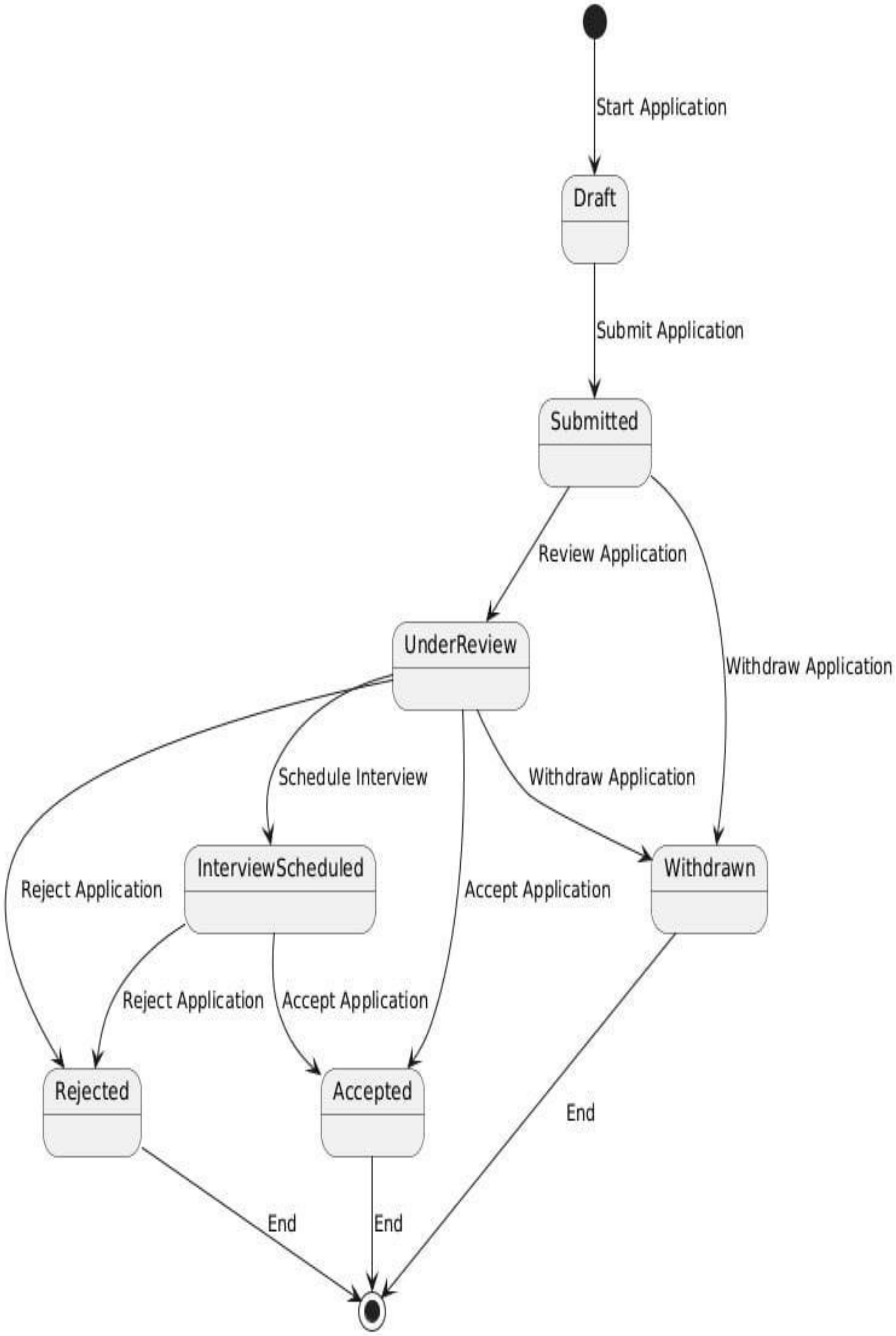


Figure 3-3: State Diagram

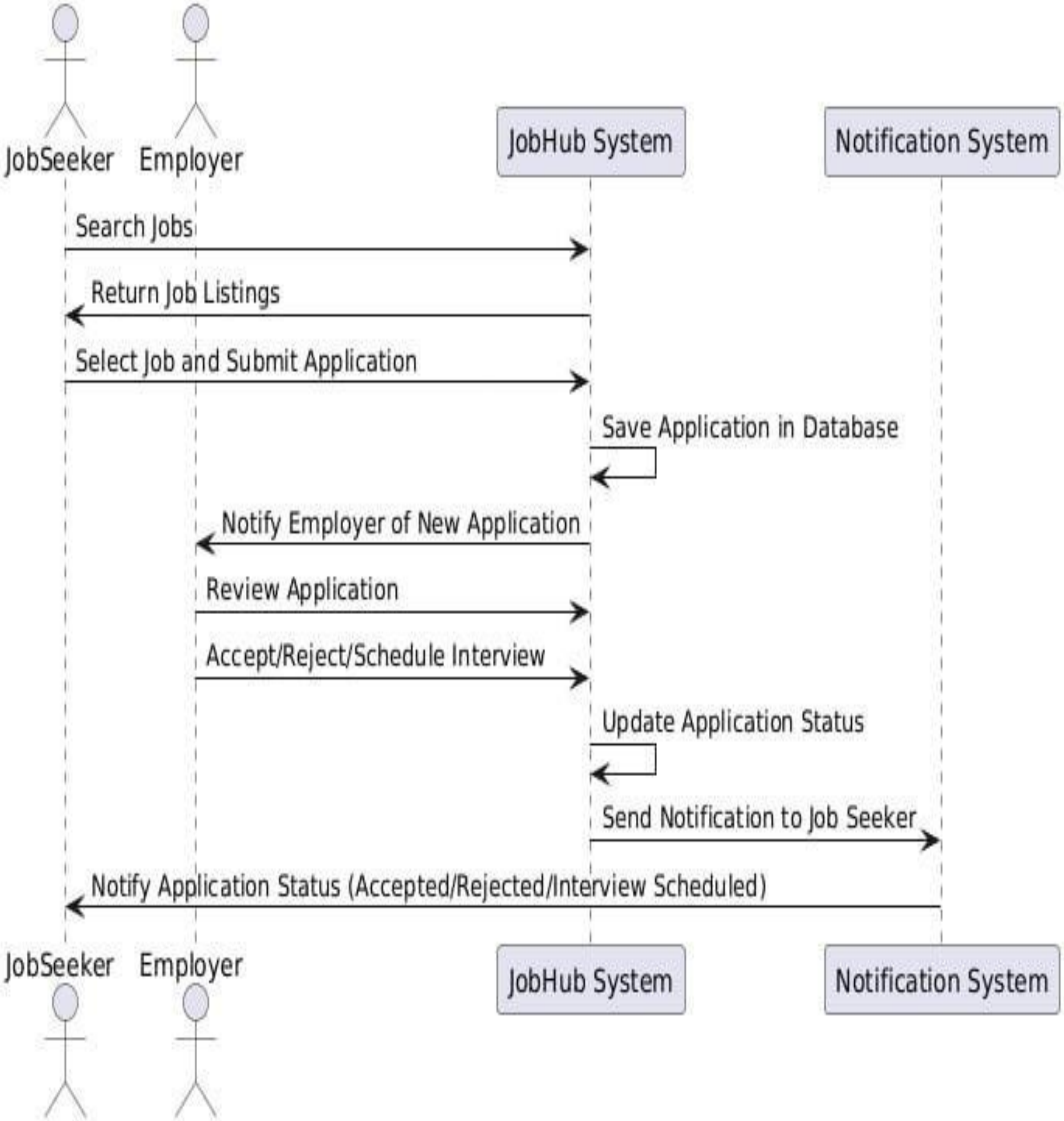

JobSeeker
Employer
JobHub System
Notification System
Search Jobs
Return Job Listings
Select Job and Submit Application
Save Application in Database
Notify Employer of New Application
Review Application
Accept/Reject/Schedule Interview
Update Application Status
Send Notification to Job Seeker
Notify Application Status (Accepted/Rejected/Interview Scheduled)
JobSeeker
Employer
JobHub System
Notification System

Figure 3-4: Sequence Diagram

3.1.5 Process Modelling: Activity Diagram

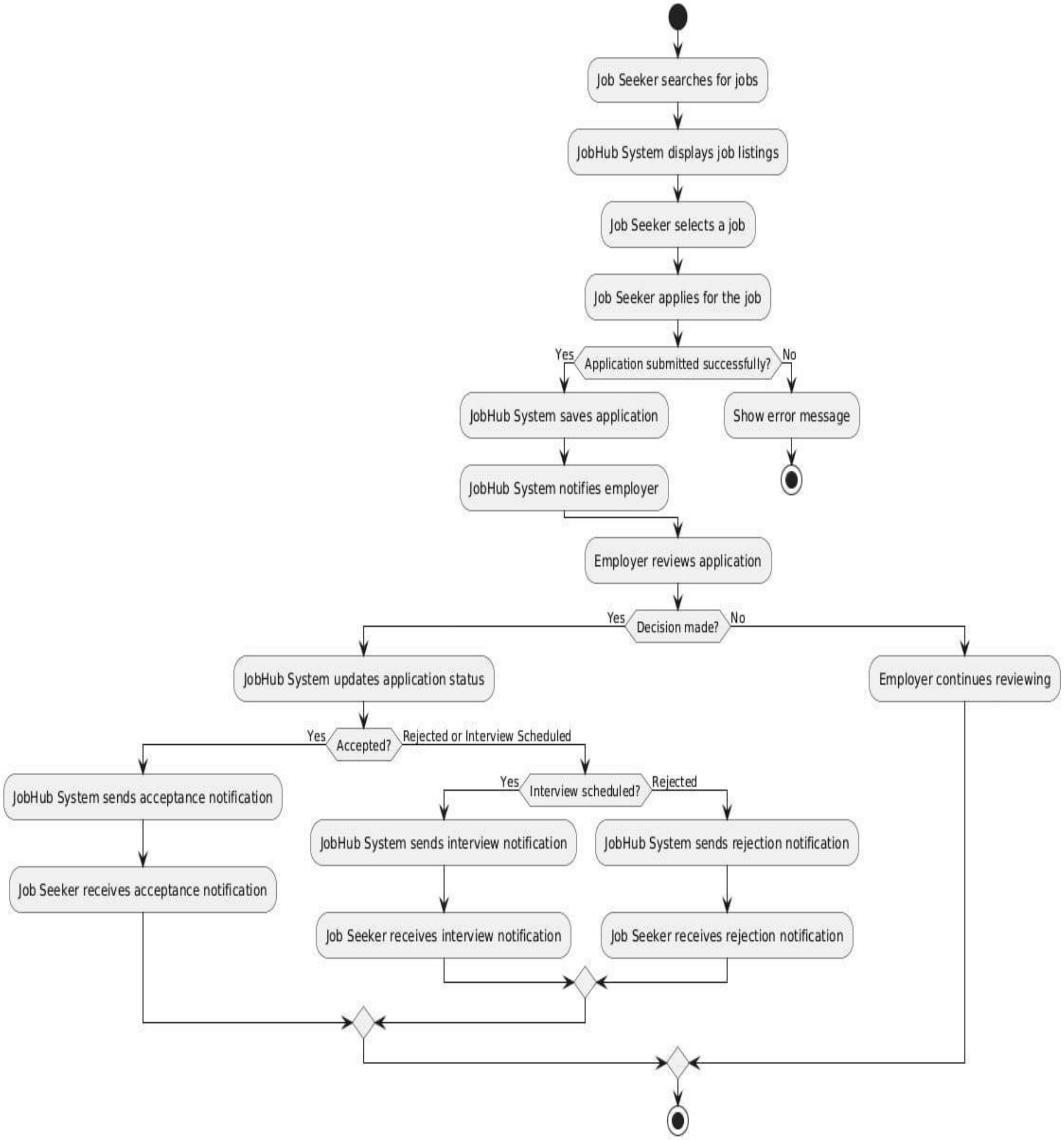

Job Seeker searches for jobs
JobHub System displays job listings
Job Seeker selects a job
Job Seeker applies for the job
Yes
Application submitted successfully?
No
JobHub System saves application
Show error message
JobHub System notifies employer
Employer reviews application
Yes
Decision made?
No
JobHub System updates application status
Employer continues reviewing
Yes
Accepted?
Rejected or Interview Scheduled
JobHub System sends acceptance notification
Yes
Interview scheduled?
Rejected
JobHub System sends interview notification
JobHub System sends rejection notification
Job Seeker receives acceptance notification
Job Seeker receives interview notification
Job Seeker receives rejection notification

Figure 3-5: Activity Diagram

3.2 System Design

The system design of JobHub outlines the architecture of the platform, ensuring efficient interaction between the frontend, backend, and database components. JobHub is built on the MERN stack (MongoDB, Express.js, React, Node.js) with Tailwind CSS for styling. Each component is structured to ensure modularity, scalability, and seamless communication across the system.

The platform includes the following key layers:

- **Frontend (React.js + Tailwind CSS):** Responsible for rendering the user interface, allowing job seekers and employers to interact with the platform in real-time. It communicates with the backend via REST APIs for data fetching and updating.
- **Backend (Express.js + Node.js):** Handles business logic, authentication, and API requests. This layer processes all user interactions, such as job applications, searches, and profile management.
- **Database (MongoDB):** Stores data persistently, including user profiles, job listings, applications, and notifications. MongoDB's flexible schema design allows for easy scaling as new features are added.

Authentication and Security: User sessions are managed using JWT tokens.

3.2.1 Refinement of Classes and Object

As the system design progresses, core classes and objects are refined to meet all functional requirements of JobHub. These classes include methods and attributes that represent the core entities such as users, jobs, and applications. Below are the key classes refined during this phase:

The User class represents job seekers and employers on the platform. It manages user registration, login, profile updates, and application tracking. The Job class represents job listings created by employers. It includes job details and the logic for creating and updating job postings. The Application class represents job applications submitted by users. The Search class handles job searches and applies filters based on user input.

3.2.2 Component Diagram

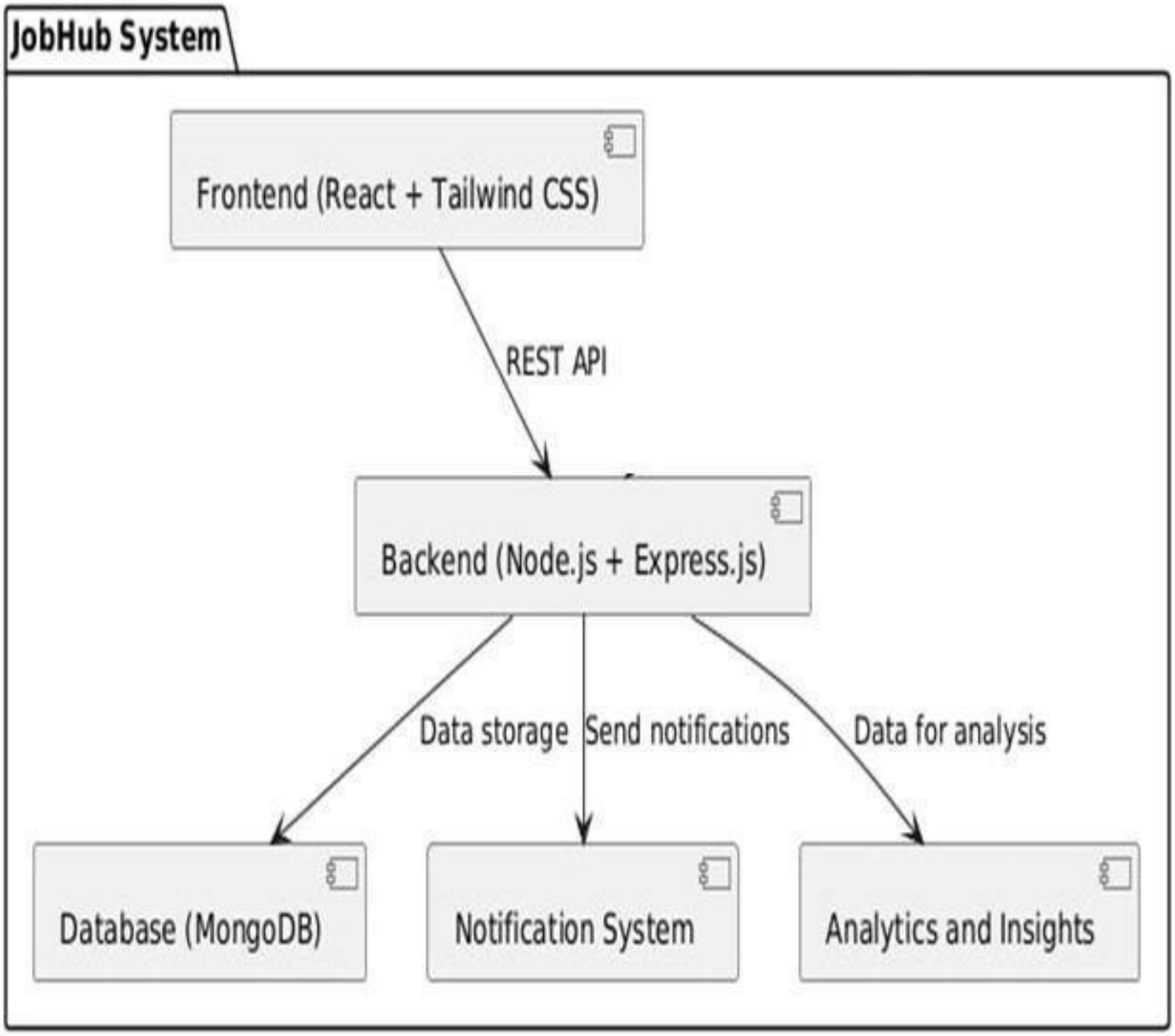


Figure 3-6: Component Diagram

3.2.3 Deployment Diagram

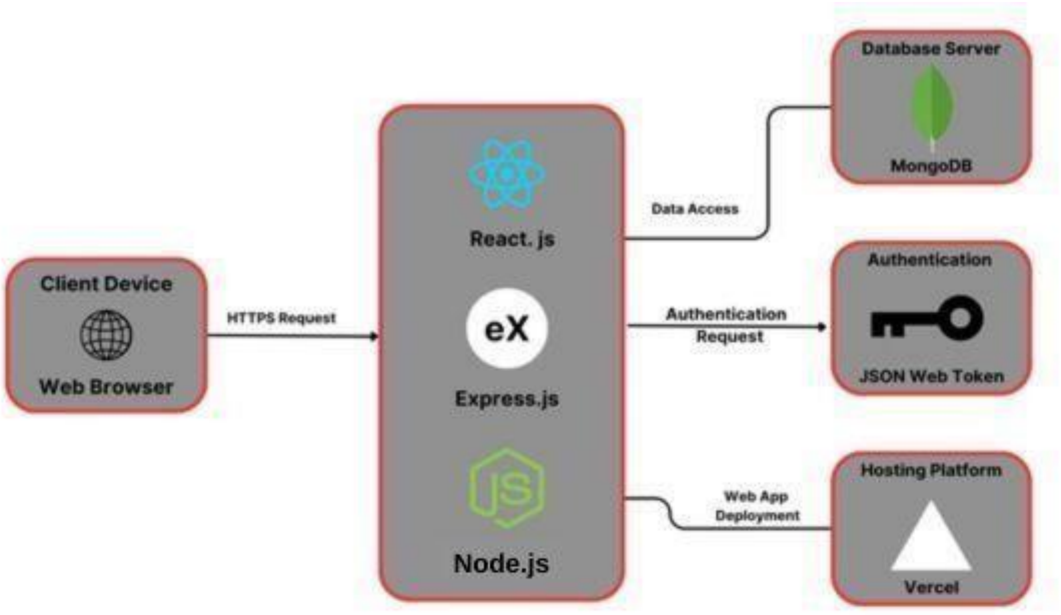


Figure 3-7: Deployment Diagram

3.3 Algorithm Details

The key algorithms powering JobHub include:

- **Searching and Filtering Algorithm:** Allows users to efficiently search and narrow down job listings based on specific criteria.

Searching Algorithm:

```
// Get all jobs (for students)
export const getAllJobs = async (req, res) => {    try {       const keyword =
req.query.keyword || ""; // Get search keyword from query parameters (if any)
const query = {
    $or: [
      { title: { $regex:   keyword, $options: "i" } }, // Search in job title
    { description: { $regex: keyword, $options: "i" } }, // Search in job
description
   ],
  };

  // Fetch jobs based on the search query, populate company details, and sort
by creation date
  const jobs = await Job.find(query)
   .populate({
path: "company",
   })
   .sort({ createdAt: -1 });

  if (!jobs) {
   return res.status(404).json({
message: "Jobs not found.",
success: false,
   });
  }
  return res.status(200).json({
jobs,       success: true,
  });
 } catch (error) {      console.log(error); // Log the
error for debugging
 }
```

```
            };
```

/Filtering Algorithm:

```
//Define filter data for jobs
const fitlerData = [
  {
    fitlerType: "Location", // Type of filter     array: ["Kathmandu", "Hetuda", "Butwal",
"Bharatpur", "Pokhara"], // Options for Location
  },
{
    fitlerType: "Industry", // Type of filter      array: ["Frontend Developer", "Backend
Developer", "FullStack Developer"], // Options for Industry
  },
];

const FilterCard = () => {
  const [selectedValue, setSelectedValue] = useState(""); // State to hold selected value
const dispatch = useDispatch(); // Get dispatch function

  // Handler for changing selected value   const
changeHandler = (value) => {
    setSelectedValue(value); // Update state with selected value
  };
```

```
 // Effect to dispatch the selected query whenever it changes  useEffect(() =>
{    dispatch(setSearchedQuery(selectedValue)); // Dispatch the searched
query
  }, [selectedValue]);
```

- **Recommendation Algorithm:** Provides personalized job recommendations to users based on their profiles, preferences, and past interactions.

```
import { Job } from "../models/job.model.js"; import
{ User } from "../models/user.model.js";

// Function to clean and normalize skills function cleanSkills(skills) {      return
skills.map(skill =>         skill.toLowerCase().trim().replace(/['"]+/g, '') // Convert to
lowercase, trim spaces,
remove quotes
  );
}

// Function to calculate content-based recommendation using cosine similarity
function cosineSimilarity(jobSkills, userSkills) {   const normalizedJobSkills
= cleanSkills(jobSkills);   const normalizedUserSkills =
cleanSkills(userSkills);

  const      matchCount      =      normalizedJobSkills.filter((skill)      =>
normalizedUserSkills.includes(skill)
```

```
  ).length;

  const totalSkills = new Set([...normalizedJobSkills, ...normalizedUserSkills]).size;

  const similarityScore = (matchCount / totalSkills) * 100;  return

 similarityScore;

}

export const getRecommendations = async (req, res) => {   try
{
   const userId = req.params.userId;
const user = await User.findById(userId);
const jobs = await Job.find();

   if (!user.profile.skills || user.profile.skills.length === 0) {         return
res.status(400).json({ message: "User has no skills listed" });
   }

   const recommendations = jobs.map((job) => {
     // Ensure the job has requirements
     if  (!job.requirements  ||  job.requirements.length  ===  0)  {
return null; // Skip jobs with no requirements
     }
     // Calculate similarity score       const
similarityScore = cosineSimilarity(
job.requirements,         user.profile.skills
     );
     return { ...job.toObject(), similarityScore }; // Use 'similarityScore' for clarity
   }).filter(job  =>  job  !==  null);  //  Remove  null  values

 const sortedRecommendations = recommendations

     .filter((job) => job.similarityScore > 0) // Only show relevant jobs
     .sort((a, b) => b.similarityScore - a.similarityScore); // Sort by similarityScore

   res.json(sortedRecommendations);
  } catch (error) {     res.status(500).json({ message:
"Server error", error });   }
};
```

Form Validation Details

The key form validation powering JobHub include:

- **JobPost Form Validation:** Validates job posting inputs to ensure completeness and accuracy, considering the user's profile, relevant preferences, and any job-specific requirements.

JobPost Form Validation Code :-

```
const validateInput = () => {     const { title, description, requirements, salary, location, jobType,
experience } = input;     const invalidCharsRegex = /[0-9@!:\-_$%^&*()+=<>?{}[\]~`]/; //
Regex for invalid
characters

   // Validate title     if (!title ||
invalidCharsRegex.test(title)) {
toast.error("Title must be non-empty and
contain no numbers or special characters.");
return false;
   }

   // Validate description     if (!description || invalidCharsRegex.test(description)) {
toast.error("Description must be non-empty and contain no numbers or special characters.");
return false;
   }

   // Validate requirements     if (!requirements || invalidCharsRegex.test(requirements)) {
toast.error("Requirements must be non-empty and contain no numbers or special characters.");
return false;
   }
   // Validate salary     if (!salary) {
toast.error("Salary is required.");
return false;
   }

   // Validate location     if (!location ||
invalidCharsRegex.test(location)) {
toast.error("Location must be non-empty and
contain no numbers or special characters.");

     return false;
   }

   // Validate jobType     if (!jobType || invalidCharsRegex.test(jobType)) {
toast.error("Job Type must be non-empty and contain no numbers or special characters.");
     return false;
   }

   // Validate experience     if (!experience || invalidCharsRegex.test(experience)) {
toast.error("Experience must be non-empty and contain no numbers or special characters.");
     return false;
   }

   return true; // Return true if all validations pass
```

```
};
```

- **Company Steup Form Validation:** Ensures correct and thorough data entry for setting up a company profile, guiding users to input relevant details based on industry standards and required company information.

Company Setup Form Validation Code :- const

```
validateInput = () => {     const { name, description, website,
location, file } = input;     // Regular expression for invalid
characters (numbers and special characters)     const
invalidCharsRegex = /[0-9@!:\-_$%^&*()+=<>?{}[\]~`]/;

  if (!name) {      toast.error("Please provide a
company name.");      return false;
  }

  if (invalidCharsRegex.test(name)) {       toast.error("Company name should not
contain numbers or special characters.");
    return false;
  }

  if (!description) {      toast.error("A
description is required.");      return false;
  }
```

```
   if (invalidCharsRegex.test(description)) {       toast.error("Description should
not contain numbers or special characters.");       return false;
   }


   if (!website) {
    toast.error("Don't forget to include the website.");
return false;
   }


   if (!location) {
toast.error("Location is necessary.");
return false;
   }


   if (file && !["image/jpeg", "image/png", "image/gif"].includes(file.type)) {
toast.error("The file type is not valid. Please upload an image in JPEG, PNG, or GIF
format.");       return false;
   }


   return true;
 };
```

- **Signup Form Validation:** Validates user signup data to ensure it meets all required criteria, verifying fields such as email, password strength, and other essential profile information for a secure and complete registration process.

Signup Form Validation Code :-

```
const  validate  =  ()  =>  {
let tempErrors = {};
  const    emailRegex    =    /^[^\s@]+@[^\s@]+\.[^\s@]{2,}$/i;
const phoneRegex = /^[0-9]{10}$/;
  const passwordRegex = /^(?=.[A-Z])(?=.[a-z])(?=.*\d)[A-Za-z\d]{8,}$/;

  if (!input.fullname) tempErrors.fullname = "Full name is
required.";    if (!input.email || !emailRegex.test(input.email))
tempErrors.email = "Please enter a valid email.";
  if (!input.phoneNumber || !phoneRegex.test(input.phoneNumber))
tempErrors.phoneNumber = "Please enter a valid 10-digit phone number.";    if
(!input.password || !passwordRegex.test(input.password))
tempErrors.password =
    "Password must be at least 8 characters long and include at least one uppercase letter,
one lowercase letter, and one number.";

  setErrors(tempErrors);
  return Object.keys(tempErrors).length === 0;
 };
```

CHAPTER 4 :IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

The successful development of JobHub involved a strategic selection of modern tools and technologies to create a smooth, scalable, and user-friendly experience. Below is a summary of the tools employed:

- **Frontend: React.js**

  We utilized React.js for the frontend, benefiting from its server-side rendering capabilities that provide quick page loads and enhance SEO. Its hybrid architecture enables us to efficiently develop both static and dynamic pages.

- **Backend: Node.js & Express.js**

  For the backend, we chose Node.js due to its effectiveness in managing asynchronous tasks, which is essential for real-time communication. The use of Express.js complements Node.js in creating a robust backend infrastructure.

- **Database: MongoDB**

  MongoDB acts as the database for JobHub, where we store critical information such as user profiles, job postings, and messages. Its flexibility and scalability make it well-suited for managing unstructured data, which is crucial for our platform.

- **UI: Shadcn UI & Tailwind CSS**

  For designing the user interface, we incorporated Shadcn UI for its ready-to use components and Tailwind CSS for fast styling. This combination allows JobHub to maintain a sleek, modern, and responsive aesthetic.

- **Deployment: Vercel**

  We employed Vercel for deploying JobHub. Its seamless integration with Next.js facilitates smooth deployment and automatic scaling, ensuring the platform can efficiently accommodate varying traffic levels.

4.1.2 Implementation Details of Modules

Each module in JobHub was carefully designed and implemented to fulfill specific functionalities. Here's an overview of some key modules:

- **User Registration & Login Module**

  This module facilitates secure user registration and login using email. Once a user registers or logs in, their session is managed using JWT (JSON Web Tokens), ensuring continuous authentication during their interaction with the portal. User data, including authentication details, are securely stored in MongoDB, providing persistent session handling and seamless login/logout experiences.

- **Profile Management Module**

  The profile management module allows users to create and manage their profiles, including adding resumes, setting job preferences, and customizing other professional details like skills and experience. It interacts with MongoDB to store user profile data and supports real-time updates through the React frontend, which ensures a smooth experience for users when updating their profiles. The system enables users to upload resumes and cover letters, which are stored securely in the database for future job applications.

- **Job Posting & Management Module**

  Employers can use this module to post jobs easily by filling out details such as job title, description, requirements, salary range, and company information. The module interacts with MongoDB to store job postings, and a backend API built with Express.js allows employees to edit, update, or delete job listings as needed. This module also features a dashboard for employers to manage job postings and track applicant responses in real time. The backend logic ensures that jobs are displayed in relevant categories and can be filtered by job seekers.

- **Job Search & Filtering Module**

  The search module enables users to efficiently find jobs that match their preferences, such as location, job type, experience level, and salary range. This module leverages a Search Algorithm that indexes job listings and provides highly relevant results based on keywords and filtering criteria set by the user. The module interacts with MongoDB to fetch job listings and filters the results in real-time, ensuring that users can quickly narrow down their job searches to the most relevant opportunities.

- **Job Application & Tracking Module**

  Job seekers can use this module to apply for jobs directly through the portal by submitting resumes, cover letters, and other relevant documents. This module securely stores application data in MongoDB and allows employers to view applicant profiles and track the status of each job application in real time. The Node.js backend ensures that both job seekers and employers are notified of application updates through email or real-time notifications.

- **Analytics & Insights Module**

  Employers have access to this module to analyze the performance of their job listings, track application statistics, and review hiring trends. The module collects and processes data from user interactions and application metrics, generating insightful reports. MongoDB stores this data, and Node.js services aggregate it, allowing employers to make data-driven decisions for optimizing their recruitment efforts. Additionally, job seekers can use analytics to track their job application success rates.

- **Recommendations & Matching Algorithm Module**

  This module enhances the job matching experience by providing personalized job recommendations based on the candidate's skills, experience, and preferences. It leverages a Recommendation Algorithm that analyzes user profiles, resumes, and job listings to find optimal matches. The algorithm constantly updates its suggestions based on user interactions and application history. Data is stored in MongoDB, and the Node.js backend processes matching logic, ensuring candidates are presented with jobs closely aligned with their qualifications.

4.2 Testing

Table 4.2-1: Sign up

SN	Desc.	Expected Result	Result	Remarks
01.	Sign Up	User information should be saved.	User Information is saved in the database.	Passed
02.	Empty Fields	Dialogue Box should be shown saying, “Please fill the form”	Dialogue box is shown saying, “please fill the from”	Passed

Table 4.2-2 Login

SN	Desc.	Expected Result	Result	Remarks
01.	Login	User should be redirected to dashboard after authentication.	User is redirected to dashboard after authentication.	Passed
02.	Login via Email	Redirected to email login	Login successfully	passed
03.	Empty fields	Dialogue box should be shown saying, “Please fill the form”	Dialogue box is shown saying, “Please fill the form”	Passed

Table 4.2-3: Jobs

SN	Desc.	Expected Results	Result	Remarks
01.	Creating Job	Job should be created	Job Created	passed
02.	Edit on post	Updated job should be posted	Job posted	passed

Table 4.2-4: Company

SN	Desc.	Expected Results	Results	Remarks

01.	Creating Company	Company should be created	Company Created	Passed
02.	Edit on company	Updated Company should be posted	Company posted	Passed

Table 4.2-5: Logout

SN	Desc.	Expected Results	Result	Remarks
01.	Logout	User should be logged out	User Logged out	Passed

4.2.1 Test Cases For Unit Testing

Table 4.2.1-1: Unit Testing

SN	Desc.	Expected Results	Results	Remarks
01.	Overall system testing	All the functionalities of the system should work properly after integration.	All the functionalities of the system are working properly after integration.	Passed

CHAPTER 5 : CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Conclusion

In conclusion, JobHub has made significant progress in addressing the challenges outlined in the proposal. The methodology, system design, and core features have been well thought-out and are engagement remain, the project is moving forward effectively.

The next phases will focus on completing development, refining algorithms, and ensuring the platform meets the needs of its users. With a clear roadmap, robust feedback processes, and ongoing development, JobHub is positioned to become a valuable tool in the recruitment industry.

5.2 Lesson Learnt/Outcome

At the end of the development process, the expected outcomes are:

- **Enhanced User Experience:** JobHub will offer a seamless, intuitive user interface, making it easy for users to search, apply, and engage with job postings.
- **Optimized Job Matching:** The platform's machine learning-powered algorithms will ensure job seekers receive relevant recommendations tailored to their skills and preferences.

- **Data security:** The platform will feature robust security measures, safeguarding user data and ensuring compliance with data protection laws.
- **Actionable Insights:** Employers and job seekers alike will benefit from analytics tools that provide insights into job market trends and candidate performance, enabling informed decisions.

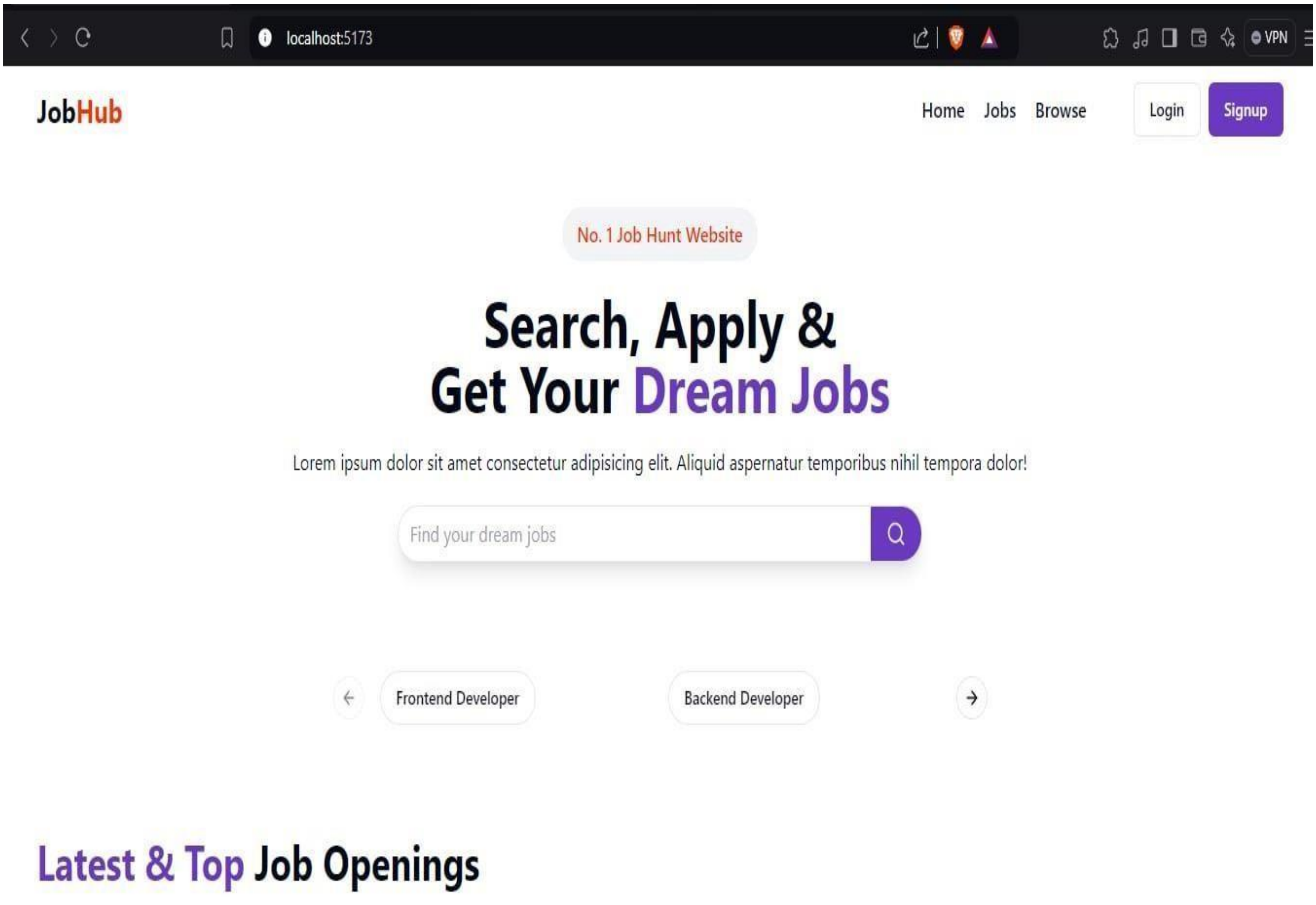
5.3 Future Recommendations

While JobHub has successfully met its initial objectives, there are several areas where the platform could expand and improve in the future:

- **AI-Driven Job Matching:** Although current matching algorithms perform well, integrating more advanced AI and machine learning techniques could provide even better job recommendations and career insights for users, further enhancing user satisfaction and engagement.
- **Mobile Application Development:** Expanding JobHub beyond the web-based platform by developing native mobile applications for IOS and Android would increase accessibility and user engagement. Mobile apps could leverage devicespecific features like geo-location for enhanced job search capabilities.
- **Employer-Candidate Communication:** Building out more advanced communication tools, such as live chat or video interviews, could streamline the hiring process and foster stronger connections between job seekers and employers directly through the platform.
- **Internationalization and Localization:** As JobHub grows, expanding its reach to global markets with multilingual support and region-specific job listings will broaden its user base, making it a truly global job portal.

By continuously improving and expanding its feature set, JobHub can stay at the forefront of the job portal industry and cater to the evolving needs of its users.

APPENDICES

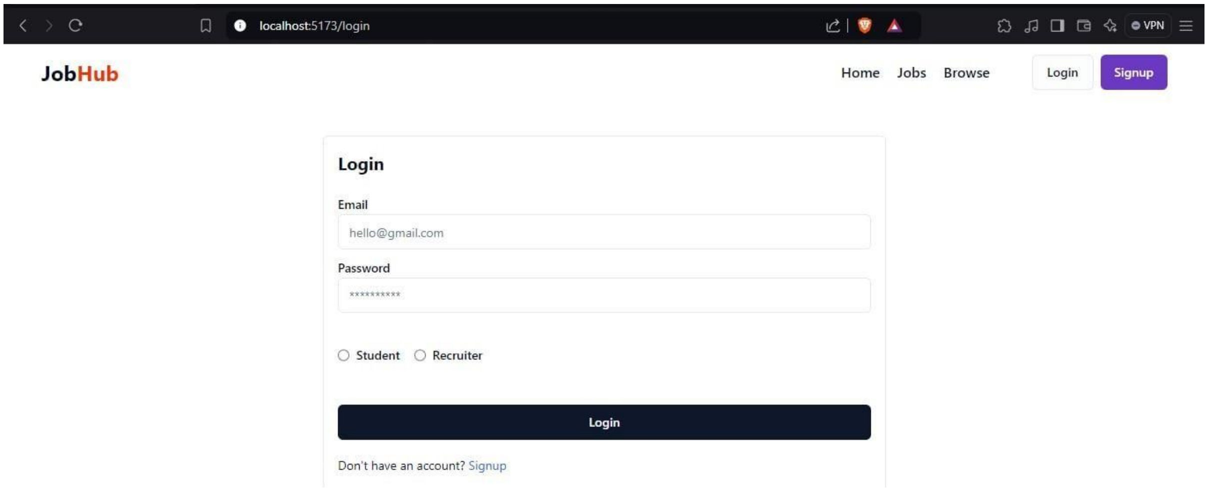


Search, Apply & Get Your Dream Jobs

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquid aspernatur temporibus nihil tempora dolor!

Latest & Top Job Openings

Figure 5-1: Hero Section



Login

Email

hello@gmail.com

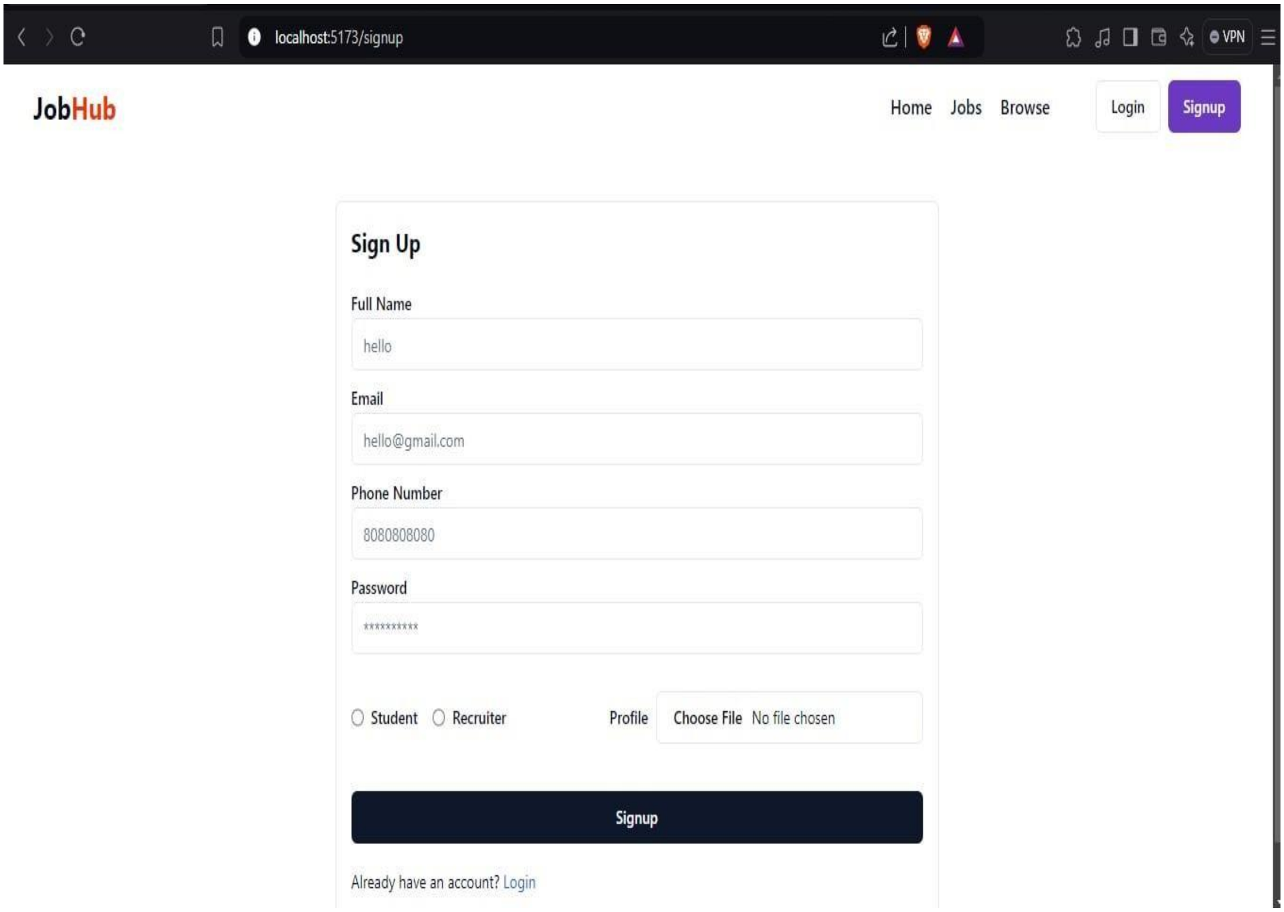
Password

****

○ Student ○ Recruiter

Don't have an account? Signup

Figure 5-2: Login Page



Sign Up

Full Name

hello

Email

hello@gmail.com

Phone Number

8080808080

Password

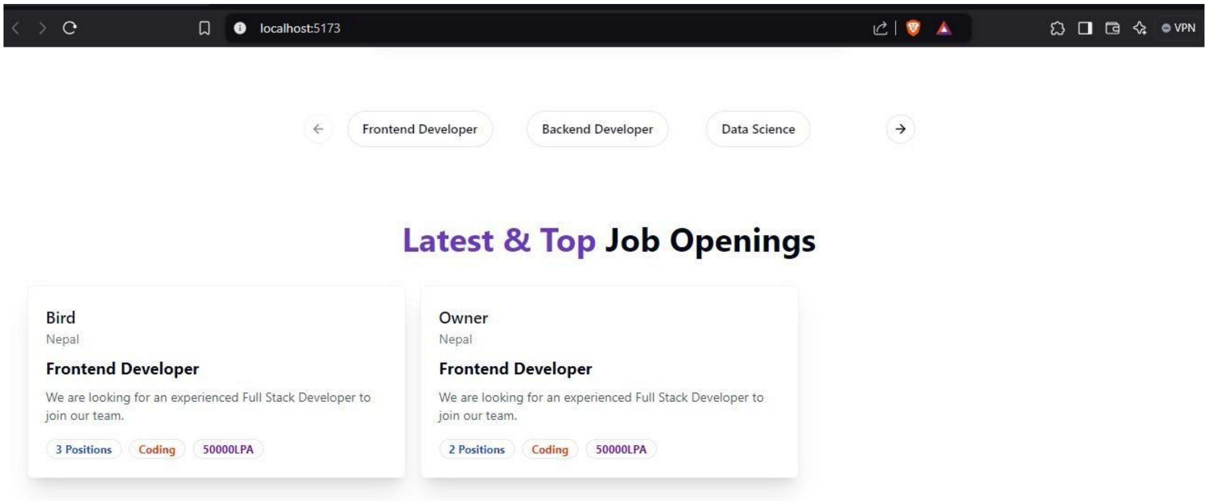
****

○ Student ○ Recruiter Profile Choose File No file chosen

Signup

Already have an account? Login

Figure 5-3: Signup Page



Latest & Top Job Openings

Bird
Nepal
Frontend Developer
We are looking for an experienced Full Stack Developer to join our team.
3 Positions | Coding | 50000LPA

Owner
Nepal
Frontend Developer
We are looking for an experienced Full Stack Developer to join our team.
2 Positions | Coding | 50000LPA

Figure 5-4: Latest Job Section

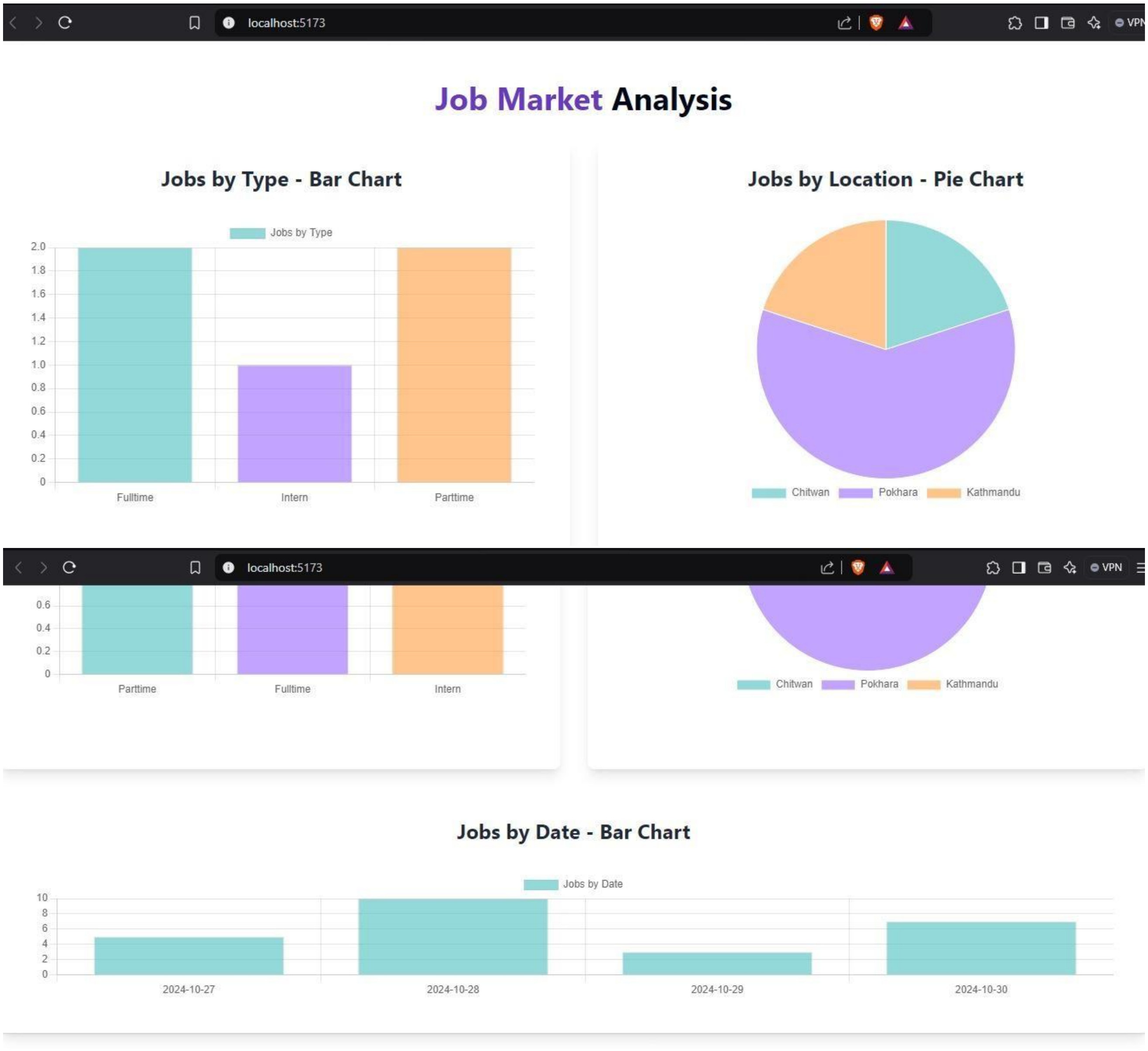
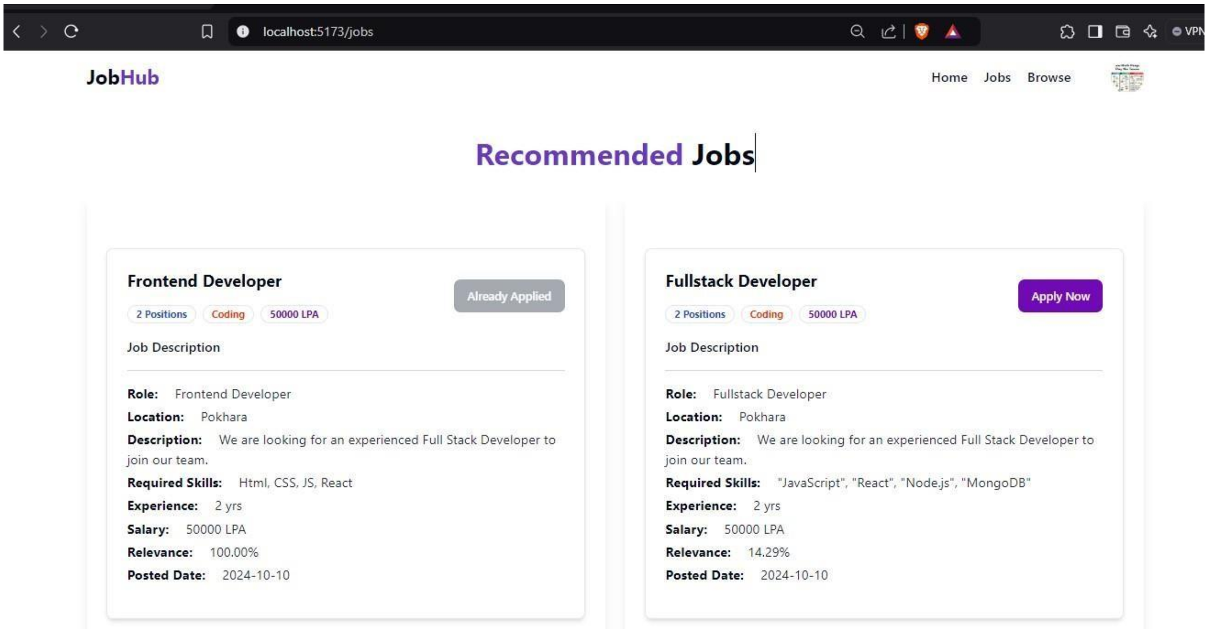


Figure 5-5: Job Market Analysis



Recommended Jobs

Frontend Developer — Already Applied

2 Positions · Coding · 50000 LPA

Job Description

Role: Frontend Developer
Location: Pokhara
Description: We are looking for an experienced Full Stack Developer to join our team.
Required Skills: Html, CSS, JS, React
Experience: 2 yrs
Salary: 50000 LPA
Relevance: 100.00%
Posted Date: 2024-10-10

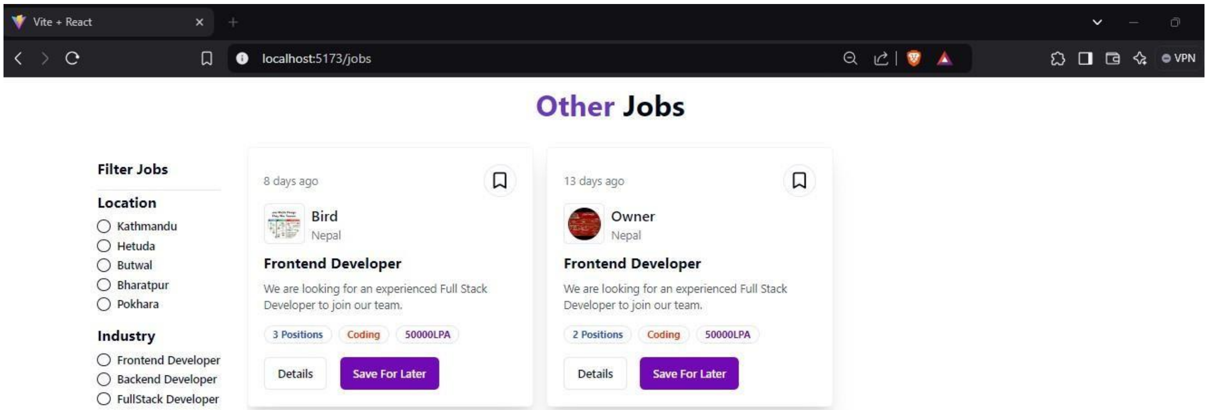
Fullstack Developer — Apply Now

2 Positions · Coding · 50000 LPA

Job Description

Role: Fullstack Developer
Location: Pokhara
Description: We are looking for an experienced Full Stack Developer to join our team.
Required Skills: "JavaScript", "React", "Node.js", "MongoDB"
Experience: 2 yrs
Salary: 50000 LPA
Relevance: 14.29%
Posted Date: 2024-10-10

Figure 5-6: Recommendation Section



Other Jobs

Filter Jobs

Location
○ Kathmandu
○ Hetuda
○ Butwal
○ Bharatpur
○ Pokhara

Industry
○ Frontend Developer
○ Backend Developer
○ FullStack Developer

8 days ago

Bird
Nepal

Frontend Developer

We are looking for an experienced Full Stack Developer to join our team.

3 Positions · Coding · 50000LPA

Details · Save For Later

13 days ago

Owner
Nepal

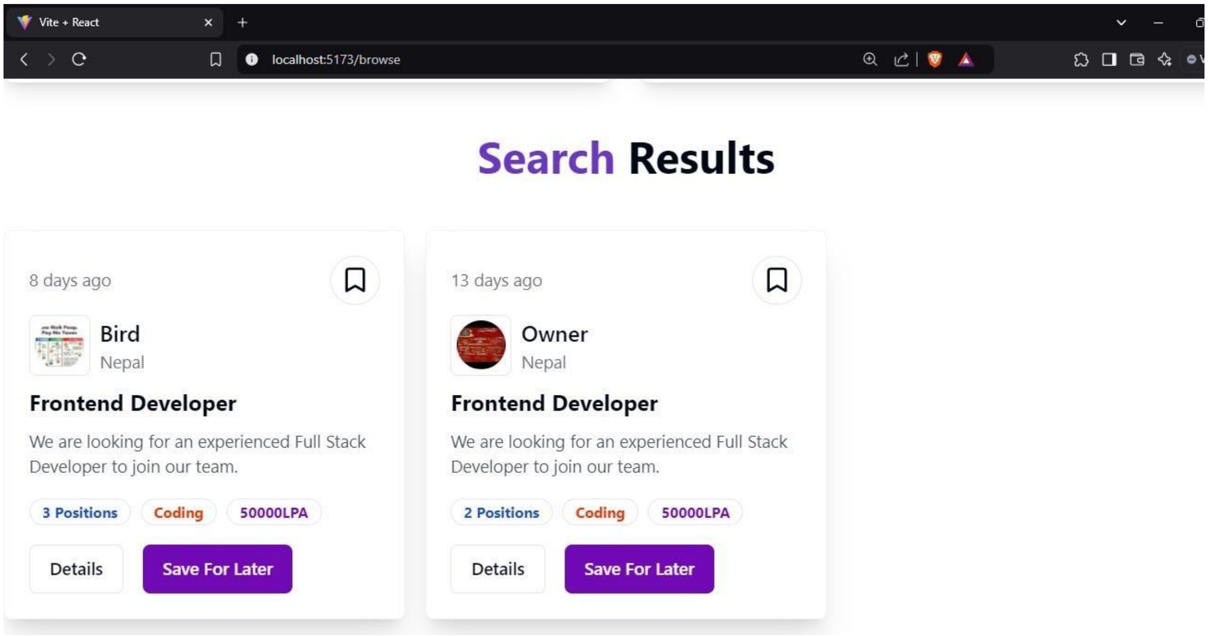
Frontend Developer

We are looking for an experienced Full Stack Developer to join our team.

2 Positions · Coding · 50000LPA

Details · Save For Later

Figure 5-7: Filter Job Section



Search Results

8 days ago

Bird
Nepal

Frontend Developer

We are looking for an experienced Full Stack Developer to join our team.

3 Positions | Coding | 50000LPA

13 days ago

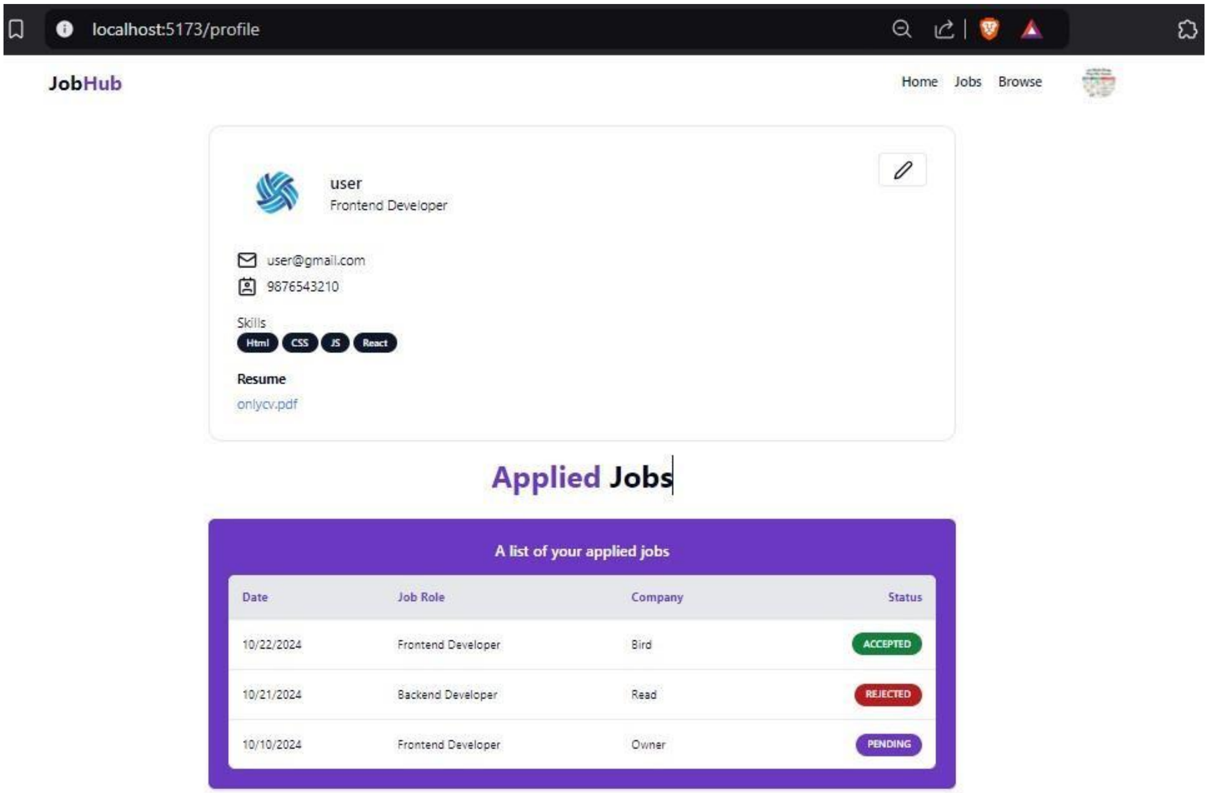
Owner
Nepal

Frontend Developer

We are looking for an experienced Full Stack Developer to join our team.

2 Positions | Coding | 50000LPA

Figure 5-8: Search Result Section



user
Frontend Developer

user@gmail.com
9876543210

Skills
Html CSS JS React

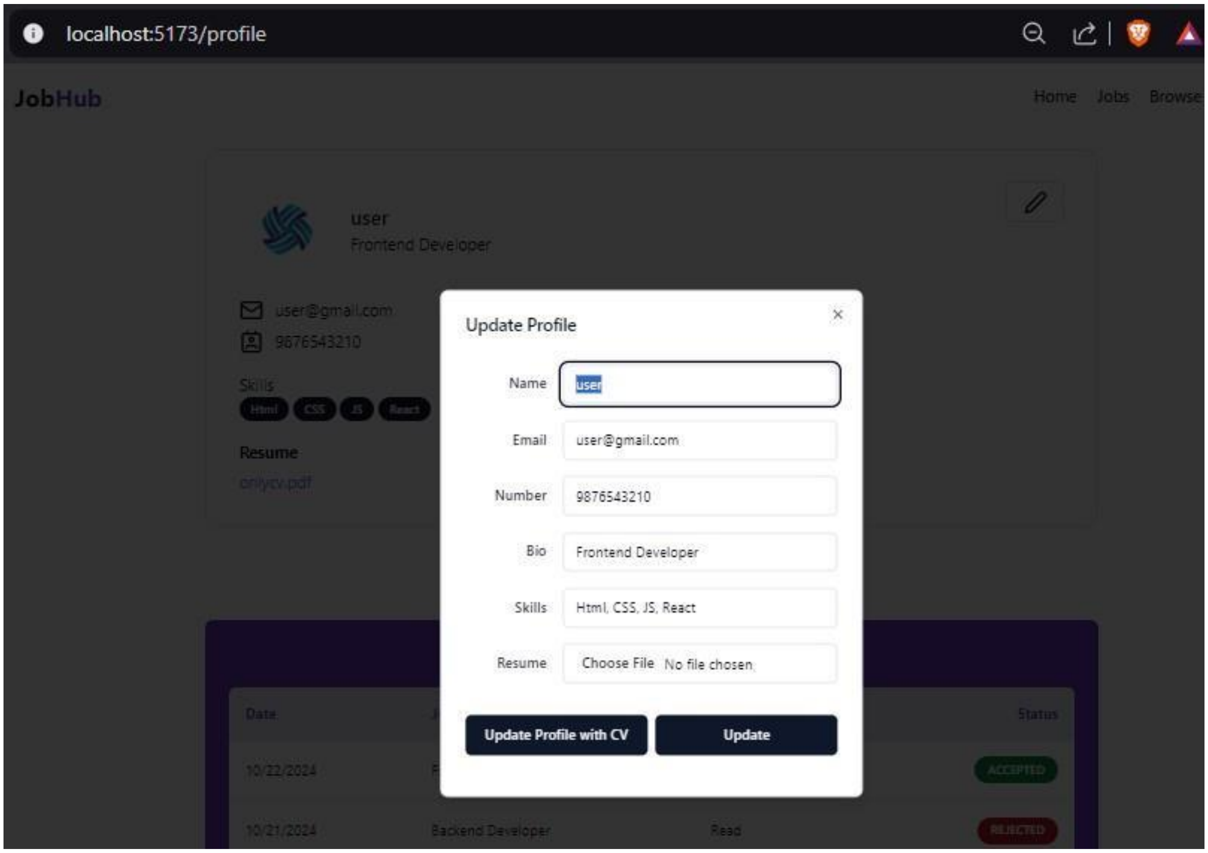
Resume
onlycv.pdf

Applied Jobs

A list of your applied jobs

Date	Job Role	Company	Status
10/22/2024	Frontend Developer	Bird	ACCEPTED
10/21/2024	Backend Developer	Read	REJECTED
10/10/2024	Frontend Developer	Owner	PENDING

Figure 5-9: User Profile Section



user
Frontend Developer

user@gmail.com
9876543210

Skills
Html CSS JS React

Resume
onlycv.pdf

Name: user

Email: user@gmail.com

Number: 9876543210

Bio: Frontend Developer

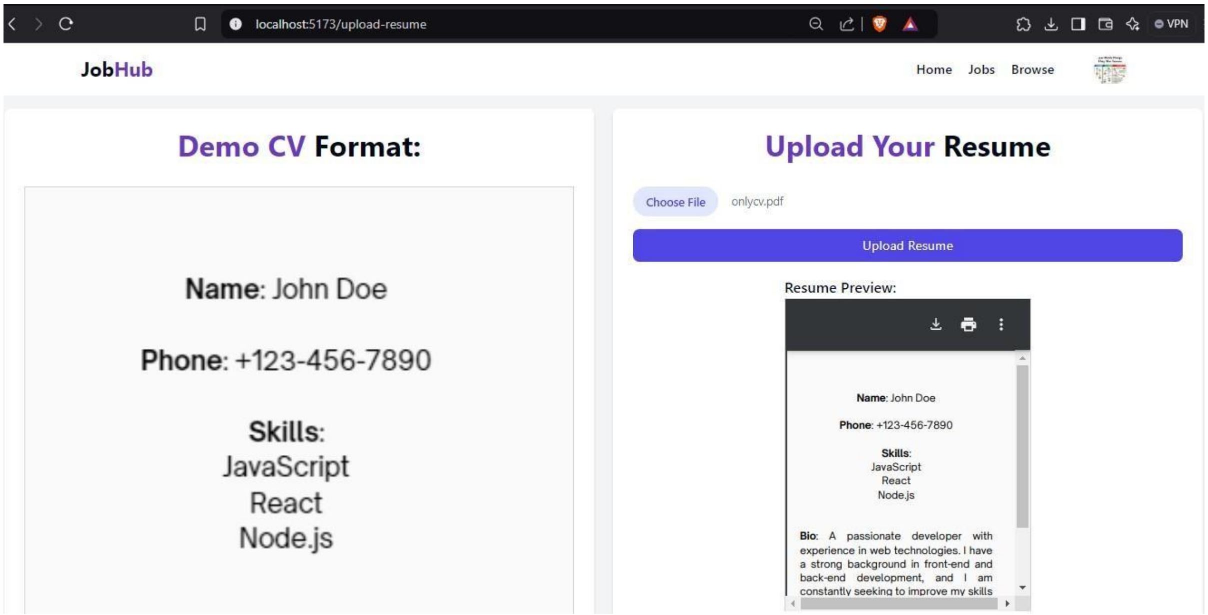
Skills: Html, CSS, JS, React

Resume: Choose File No file chosen

10/22/2024 … ACCEPTED

10/21/2024 Backend Developer Read REJECTED

Figure 5-10: Update User Profile Section



Demo CV Format:

Name: John Doe

Phone: +123-456-7890

Skills:
JavaScript
React
Node.js

Upload Your Resume

Choose File onlycv.pdf

Resume Preview:

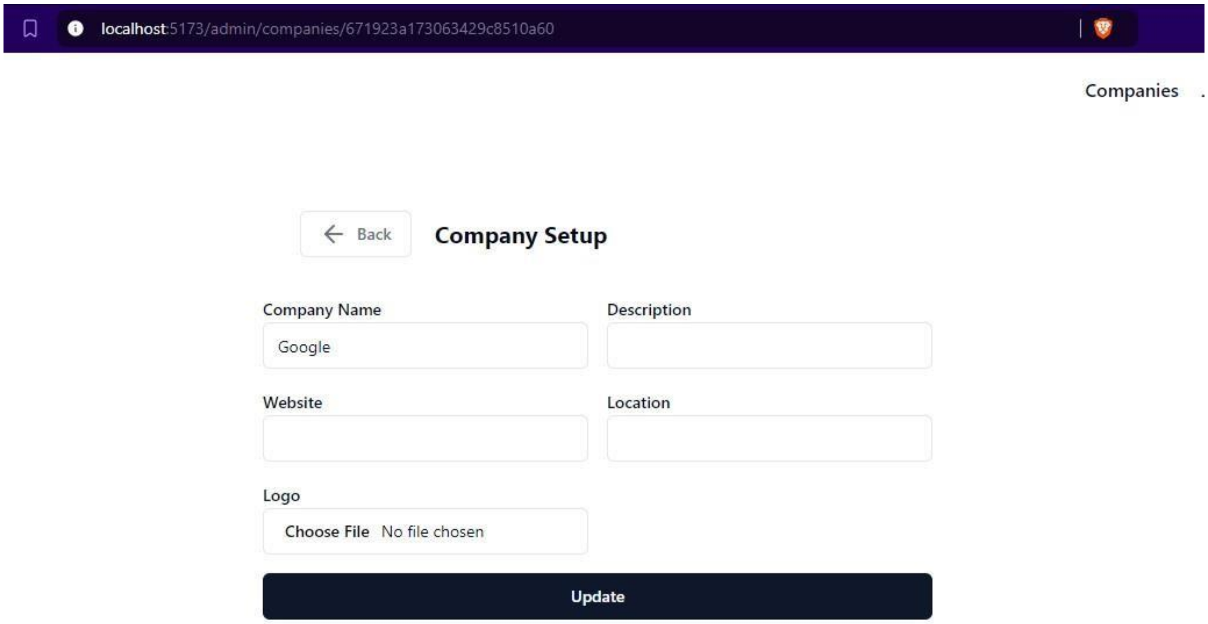
Name: John Doe

Phone: +123-456-7890

Skills:
JavaScript
React
Node.js

Bio: A passionate developer with experience in web technologies. I have a strong background in front-end and back-end development, and I am constantly seeking to improve my skills

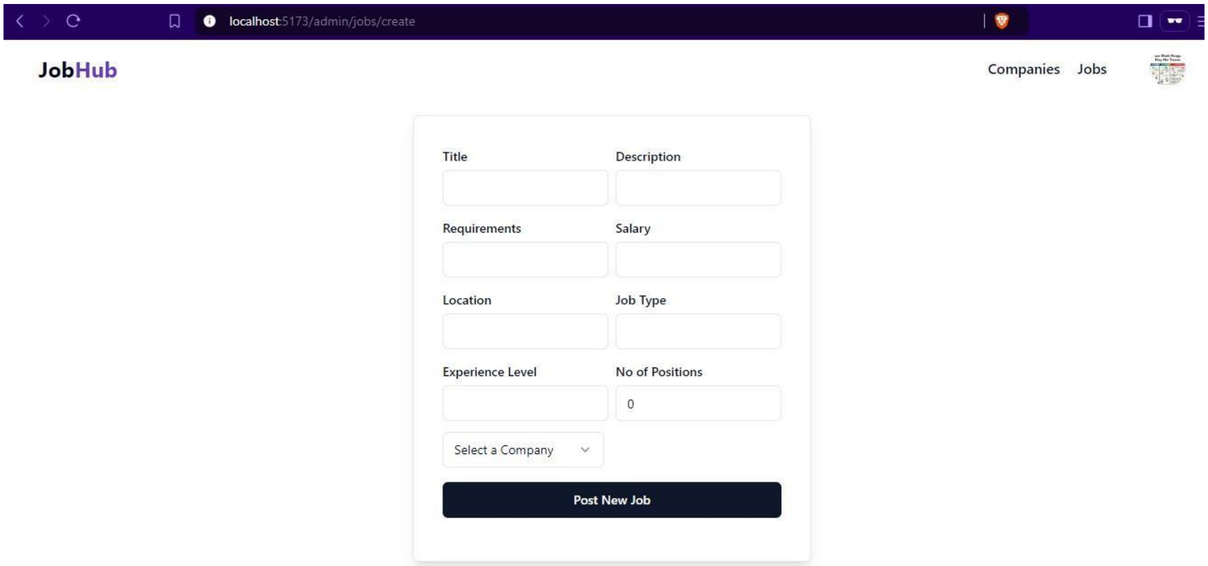
Figure 5-11: Resume Upload Section



Company Name: Google

No file chosen

Figure 5-12: Company Create Section



No of Positions: 0

Figure 5-13: Job Create Section

REFERENCES

[1] *Develop skills that move your career forward* (no date) *LinkedIn Learning with Lynda: Online Training Courses for Creative, Technology, Business skills*. Available at:

https://www.linkedin.com/learning/ (Accessed: 10 July 2024).

[2] *Indeed careers*. Available at: https://www.indeed.jobs/ (Accessed: 10 July 2024).

[3] *Glassdoor | Job Search & Career Community*. Available at: https://www.glassdoor.com/index.htm (Accessed: 10 July 2024).

[4] Upwork (No date). Find Freelance Jobs and Projects. Available at: https://www.upwork.com/ (Accessed: 10 July 2024).

[5] Freelancer (No date). Hire Freelancers & Find Freelance Jobs. Available at: https://www.freelancer.com/ (Accessed: 10 July 2024).

[6] Fiverr (No date). Find Services Starting at $5. Available at: https://www.fiverr.com/ (Accessed: 10 July 2024).

[7] PeoplePerHour (No date). Hire Freelancers for Projects. Available at: https://www.peopleperhour.com/ (Accessed: 10 July 2024).