A
Project Report On
"Job Management System "

Prepared by

Khyati Thakkar(23DCS132)

Under the guidance of

Prof. Dipak Ramoliya

A Report Submitted to

Charotar University of Science and Technology

For Partial Fulfillment of the Requirements for the

5th Semester Summer Internship-I (CSE306)

Submitted at

Department of Computer Science & Engineering

Devang Patel Institute of Advance Technology and Research

At: Changa, Dist: Anand – 388421

July 2025

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**Job-Management**" is a bonafide work carried out by **Khyati Thakkar (23DCS132)** under the guidance and supervision of **Prof. Dipak Ramoliya** for the subject **CSE306 Summer Internship-I** of 5th Semester of Bachelor of Technology in **Department of Computer Science & Engineering, DEPSTAR** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Dipak Ramoliya
Assistant Professor,
Department of Computer Science & Engineering
DEPSTAR, CHARUSAT, Changa, Gujarat.

Ms. Mansi Gami
Web Developer
Sparks To Ideas
Ahmedabad

Dr. Amit Nayak
Associate Professor
Head of Department,
Department of Computer Science & Engineering,
DEPSTAR,
CHARUSAT, Changa, Gujarat.

INTERNSHIP COMPLETION CERTIFICATE

Subject:- MERN Stack Internship

This is to certify that **Ms. Khyati Thakkar** has successfully completed our **MERN Stack Internship** Program in **Sparks To Ideas** session Starting from 12th May 2025 to 12th June 2025. She completed her internship in MERN Stack technology. We wish her all the best wishes for her bright future.

Sincerely,
Ajay Chauhan
Sparks To Ideas
Co - Founder
Sparks To Ideas

www.sparkstoideas.com
info@sparkstoideas.com

406 Akshat Tower near Pakwan hotel
opposite rajpath club SG highway
Ahmadabad

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to **Sparks To Ideas**, Ahmedabad, for providing me with the opportunity to undertake my summer internship in the field of **MERN Stack**. This internship has been an invaluable experience that has allowed me to apply my academic knowledge in a professional setting and enhance my practical skills.

I am sincerely thankful to my internal guide, **Prof. Dipak Ramoliya**, for his constant encouragement, valuable suggestions, and inspiring support throughout the duration of my internship. His expert guidance helped me stay focused and motivated, and his insights contributed significantly to the successful completion of this project.

I extend my special thanks to my external guide, **Mansi Gami**, for her immense support, practical guidance, and professional advice during my time at the organization. H er mentorship helped me understand industry standards and expectations, as well as the importance of thinking, consistency.

I would also like to thank the entire team at **Sparks To Ideas** for their friendly working environment and for sharing their knowledge and experience generously. Working with such a dedicated team helped me develop confidence and collaboration skills.

Furthermore, I am grateful to the **Department of Computer Science & Engineering, Charotar University of Science and Technology (CHARUSAT)** for providing me with this wonderful learning opportunity and for encouraging students to gain industry exposure through internships.

Lastly, I would like to thank my family and friends for their constant support and motivation throughout this jou

ABSTRACT

This report outlines the development and deployment of a feature-rich Job Management System , created as part of a summer internship project. The **Job Management System** is a web-based platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) to streamline the recruitment process between companies and job seekers. The system provides two dedicated panels—one for companies and another for job seekers—with secure authentication and intuitive interfaces.

In the **Company Panel**, organizations can register, log in, and post detailed job openings including title, description, location, salary, and requirements. Companies can manage their posted jobs by editing or deleting listings as needed.

The **Seeker Panel** allows job seekers to create profiles, browse available job opportunities, filter jobs based on criteria such as location and salary, and submit applications directly through the platform. The system maintains user data securely and enables seekers to track the status of their applications.

This project aims to automate and simplify traditional job application workflows, reduce administrative effort for companies, and provide an accessible, user-friendly experience for applicants. By leveraging modern web technologies, the system delivers responsive interfaces and efficient data handling, contributing to faster and more transparent hiring processes.

TABLE OF CONTENTS

DESCRIPTION OF COMPANY

Sparks to Ideas is an IT company based in Ahmedabad, Gujarat. It offers services like website development, mobile app creation, software solutions, and digital marketing. The company works with modern technologies such as the MERN stack, Flutter, and cloud platforms. It has different teams for designing, development, testing, and client support. Sparks to Ideas focuses on creating user-friendly and customized solutions for its clients, who come from different industries across India and abroad. The work environment is supportive and encourages learning, especially for students and freshers.

During my summer internship, I worked in the Web Development team on a real project. I got hands-on experience using tools and technologies like React, Node.js, and MongoDB. I also learned about working in a team, attending daily meetings, and using project tools to manage tasks. The company gave me a great opportunity to improve my skills and understand how real-world software projects are handled. Overall, it was a very helpful and learning-focused experience

1. PROJECT DEFINITION

1.1 INTRODUCTION

The Job Management System is an online web application designed to connect companies with job seekers efficiently through a centralized platform. Traditional recruitment processes often involve manual posting of job openings and collecting applications via email or paper forms, which can be time-consuming and prone to miscommunication. This project addresses these challenges by creating an automated system that simplifies job posting, searching, and application management.

1.2 OBJECTIVE OF THE PROJECT

- To develop a centralized web-based platform that connects companies and job seekers
- To automate the process of job posting, searching, and applying for positions.
- To provide secure user authentication for both companies and seekers using JWT (JSON Web Tokens).
- To enable companies to manage job listings easily, including creating, editing, and deleting postings.
- To implement filtering and search functionality so seekers can quickly find relevant jobs based on criteria such as location, salary, and skills.

1.3 SCOPE OF THE PROJECT

- Companies can perform CRUD operations on their job postings through a secure dashboard, including creating new job listings, editing details, and removing outdated positions.
- Job seekers can browse all available jobs, apply filters such as location, salary range, and required skills, and submit applications directly through the platform.
- The system uses JWT authentication to ensure secure login sessions for both companies and seekers, protecting user data and access privileges.

- React.js provides a responsive and dynamic interface, allowing users to navigate the platform seamlessly on different devices.

- Applications are stored and tracked in MongoDB, enabling seekers to monitor the status of their submissions and companies to review applicant details efficiently.

1.4 PROBLEM STATEMENT

Traditional job recruitment processes are often manual, time-consuming, and inefficient. Companies typically advertise job openings through separate channels such as newspapers, email chains, or fragmented online postings, making it difficult to manage and update listings. At the same time, job seekers must search multiple sources to find relevant opportunities, leading to confusion and missed chances.

2. DESCRIPTION

2.1 INTRODUCTION

- The project titled "JobConnect – Job Posting & Application Management Platform" is a full-stack web application developed using the MERN Stack (MongoDB, Express.js, React.js, Node.js).
- It was built as part of a MERN stack project to demonstrate practical development skills by designing a scalable, secure, and user-friendly job management system.
- The system includes two role-based panels: Company Panel (for posting and managing job vacancies) and Seeker Panel (for browsing jobs, filtering listings, and submitting applications).
- Key features include secure JWT authentication, role-based access control, advanced job filtering, intuitive dashboards for managing postings and applications, and a responsive interface optimized for different devices to deliver a smooth user experience.

2.2 WORKFLOW AND TOOLS USED

2.2.1 WORKFLOW

- Designed responsive UI components using React.js and Material UI for the user and company dashboards.
- Built RESTful APIs using Express.js and Node.js to manage booking data, user profiles, driver info, and ride interactions.
- Used MongoDB to handle data related to users, companies, job postings, and job applications.
- Implemented secure login authentication with role-based authorization to restrict access to company and seeker dashboards.
- Deployed both frontend and backend on live servers to enable public access and usability testing of the platform.

2.2.2 TOOLS USED

- React.js – For building the user interface and managing dashboard interactions.
- Node.js & Express.js – Backend runtime and framework to handle server-side logic and routes.
- MongoDB – NoSQL database used to store booking details, user/company data, and application statuses.
- Postman – For testing APIs during backend development.
- Visual Studio Code – Primary development environment.
- Git & GitHub – Version control and collaborative development.

2.3 COMPONENTS IMPLEMENTED

- Seeker Panel – Allows job seekers to create profiles, browse job listings, apply filters (e.g., location, skills, salary), and submit applications for selected jobs.
- Company Panel – Enables companies to post new jobs, manage existing job listings, and review applications received from seekers.
- Authentication System – Secure, role-based login mechanism that ensures restricted and separate access for seekers and company users.
- Job Posting Module – Lets companies add, update, and delete detailed job listings with title, description, location, requirements, and salary range.
- Job Filtering System – Provides seekers with dynamic filtering and search options to quickly find relevant jobs based on their preferences.
- Application Management – Maintains application records, allowing seekers to track applied jobs and companies to view applicant details.

2.4 DEVELOPMENT STRATEGY

- Adopted a modular architecture to organize code by role (user, company) and functionality (applying, authentication, post job).
- Followed full-stack principles using MERN with TypeScript to improve maintainability, scalability, and type safety across the project.
- Emphasized real-time interaction using sockets for ride status updates and interactive UX features.

- Ensured clean API structuring and secure request handling with middleware-based authentication.
- Applied best practices for state management, code reusability, and responsive design, making the system future-ready for features like wallet integration, ride cancellation, and notification services.

3. SOFTWARE AND HARDWARE REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

- **Visual Studio Code (VS Code):**
  Primary Integrated Development Environment (IDE) used for writing, editing, debugging, and managing frontend and backend code efficiently.
- **React.js (with TypeScript):**
  Used for building dynamic and responsive UI components for User, company dashboards with a modular design approach.
- **Node.js & Express.js:**
  Server-side environment and framework used to create RESTful APIs, manage routing, handle business logic, and connect with the database.
- **MongoDB (with Mongoose):**
  NoSQL database used to store and manage data related to users, company , post, and applying history. Integrated using Mongoose ODM for schema modeling.
- **Leaflet.js:**
  JavaScript library used for interactive map integration, allowing users to select pickup and drop-off points visually.
- **Socket.IO:**
  Used to implement real-time communication features such as ride status updates between users and drivers.
- **Tailwind CSS / Material UI:**
  For consistent and responsive design across all device sizes, providing a modern and intuitive user interface.
- **Postman:**
  Utilized for API endpoint testing and debugging backend routes related to authentication, bookings, and fare estimation.
- **Git & GitHub:**
  For source code version control, collaborative development, and deployment management.

3.1.1 CLOUD ACCOUNT

- MongoDB Atlas (or local MongoDB installation): For hosting the job's application and user management database securely in the cloud.

3.2 HARDWARE REQUIREMENTS

3.2.1 LAPTOP/DESKTOP COMPUTER

- Minimum RAM: 4 GB (8 GB recommended for smoother development experience).
- Processor: Intel i3 or higher.
- Storage: At least 2 GB free space for codebase, node modules, and project files.

3.2.2 NETWORK REQUIREMENTS:

Stable internet connection with at least 10 Mbps bandwidth; secure firewall/router setup for safe data transmission.

4. MAJOR FUNCTIONALITIES

4.1 VERSION CONTROL

- Utilized Git for tracking source code changes and managing development progress.
- Hosted the project on GitHub for collaborative development and version control.

4.2 FRONTEND FUNCTIONALITY

- Built using React.js to create responsive, modular, and interactive user interfaces for Seeker and Company dashboards.
- Key components implemented:
    - Home Page / Landing Page
    - Job Listings Page with Search and Filter Options (Seeker Panel)
    - Job Application Form
    - Seeker Dashboard (Applied Jobs, Profile)
    - Company Dashboard (Posted Jobs Management)
    - Login/Register Pages for both seekers and companies
- Used React Hooks (useState, useEffect, useContext) for managing state, triggering updates, and ensuring a smooth user experience.
- Material UI and Tailwind CSS were used to ensure consistency, responsiveness, and visual appeal across devices.

4.3 BACKEND FUNCTIONALITY

- Developed using Node.js and Express.js, exposing RESTful APIs to handle user authentication, job postings, job searching, and application management.
- Middleware functionalities included:
    - Request validation to prevent invalid or malformed inputs.
    - Global error handling for consistent API responses.
    - Authentication & Role-Based Authorization to secure endpoints and control access.

- Request validation to prevent invalid or malformed inputs when creating accounts, posting jobs, or submitting applications

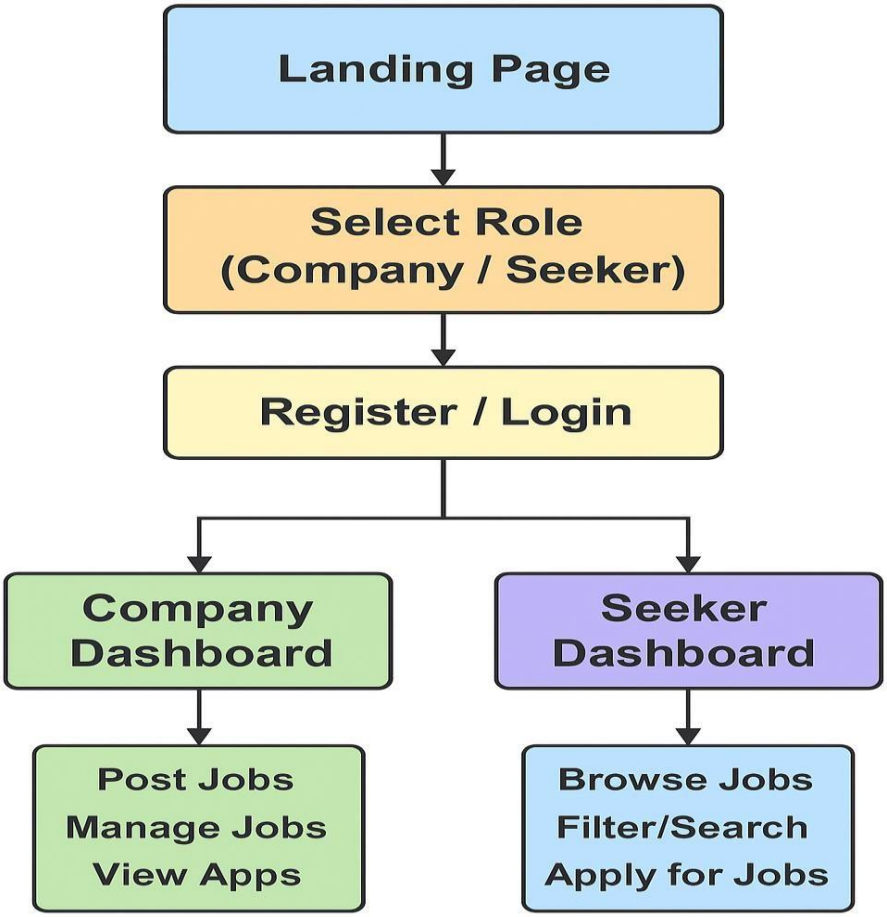
4.4 DATABASE OPERATIONS

- Used MongoDB (with Mongoose) for managing data related to users, companies, job postings, and job applications.
- Designed optimized schemas for:
  - User data (name, email, password, role, profile details)
  - Company profiles (company name, contact information, posted jobs)
  - Job postings (title, description, location, salary, requirements)
  - Applications (job reference, applicant reference, application status)
- Implemented full CRUD operations via backend APIs to support user registration, job creation and management, application submission, and data.

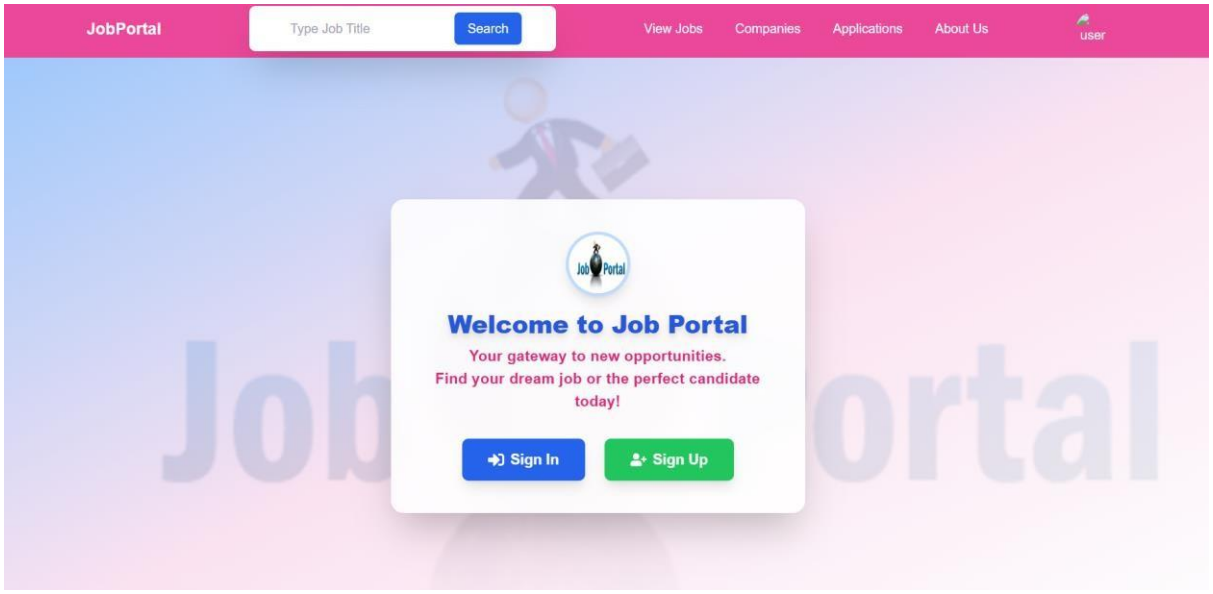
4.5 AUTHENTICATION AND AUTHORIZATION

- Implemented a **role-based login system** where Users, companies have distinct access and permissions.
- Used **JSON Web Tokens (JWT)** for secure, token-based session management and protected route access.
- Only authenticated users can:
  - **Job seeker (Users)**
  - **Post job (companies)**
- Routes and dashboards are protected to ensure only authorized roles can access specific functionalities.

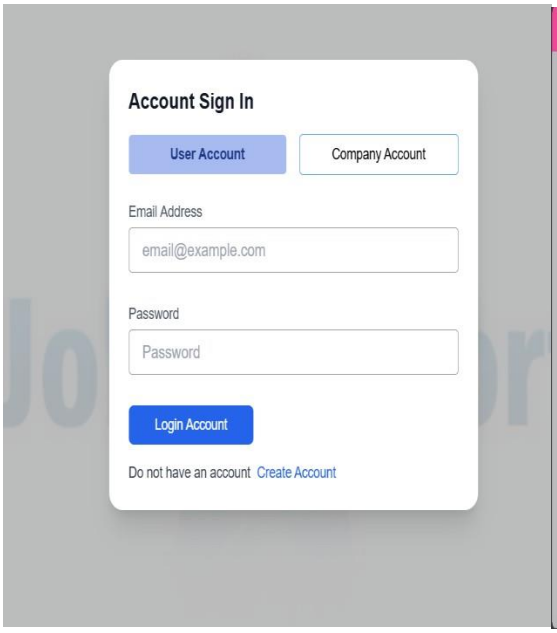
5. FLOWCHART


Landing Page
Select Role
(Company / Seeker)
Register / Login
Company
Dashboard
Seeker
Dashboard
Post Jobs
Manage Jobs
View Apps
Browse Jobs
Filter/Search
Apply for Jobs

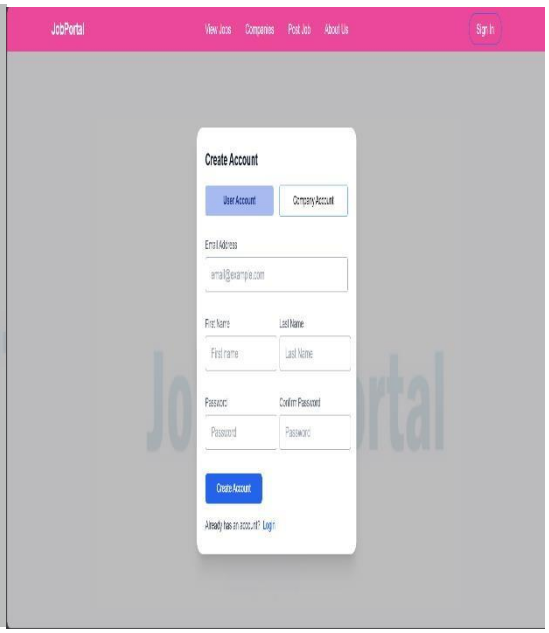
6. SCREENSHOTS OF PROJECT OUTPUT



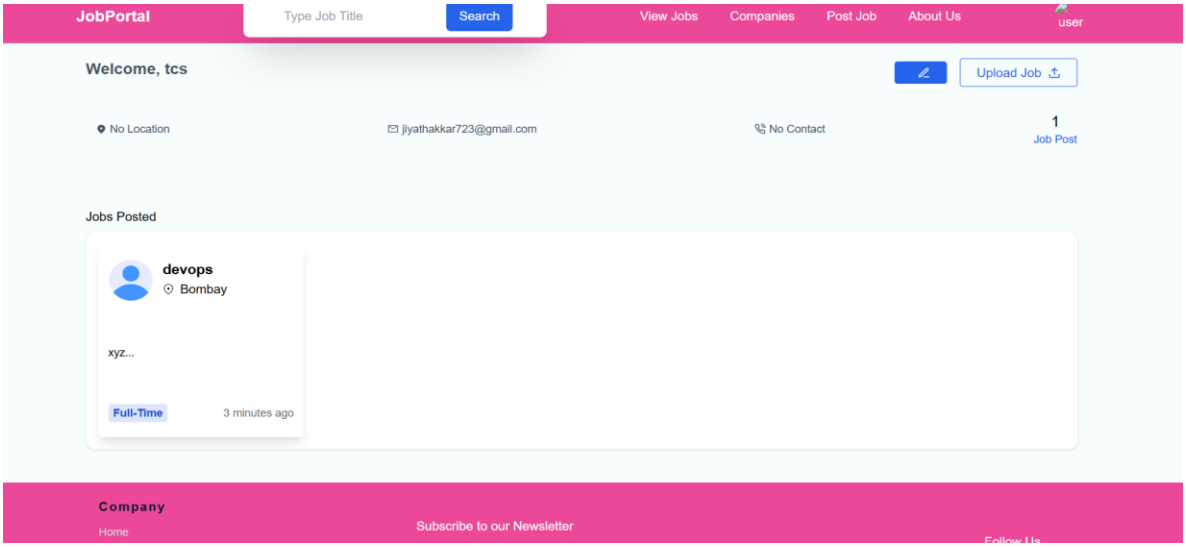
1.1 Home Page



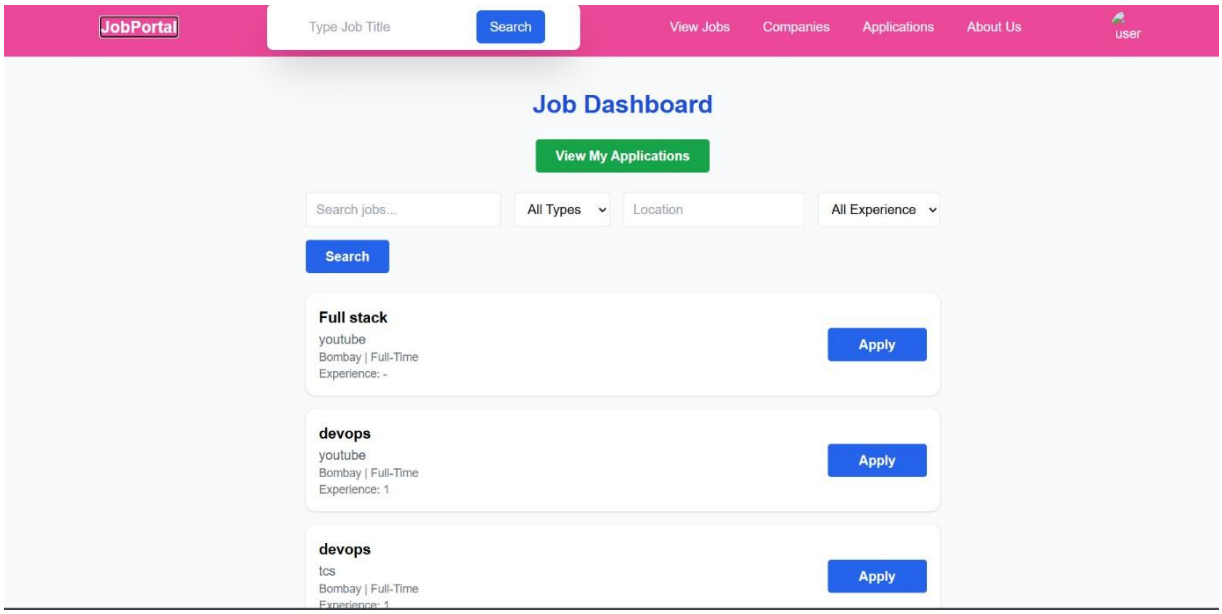
1.2 Login Page



1.3 Register Page

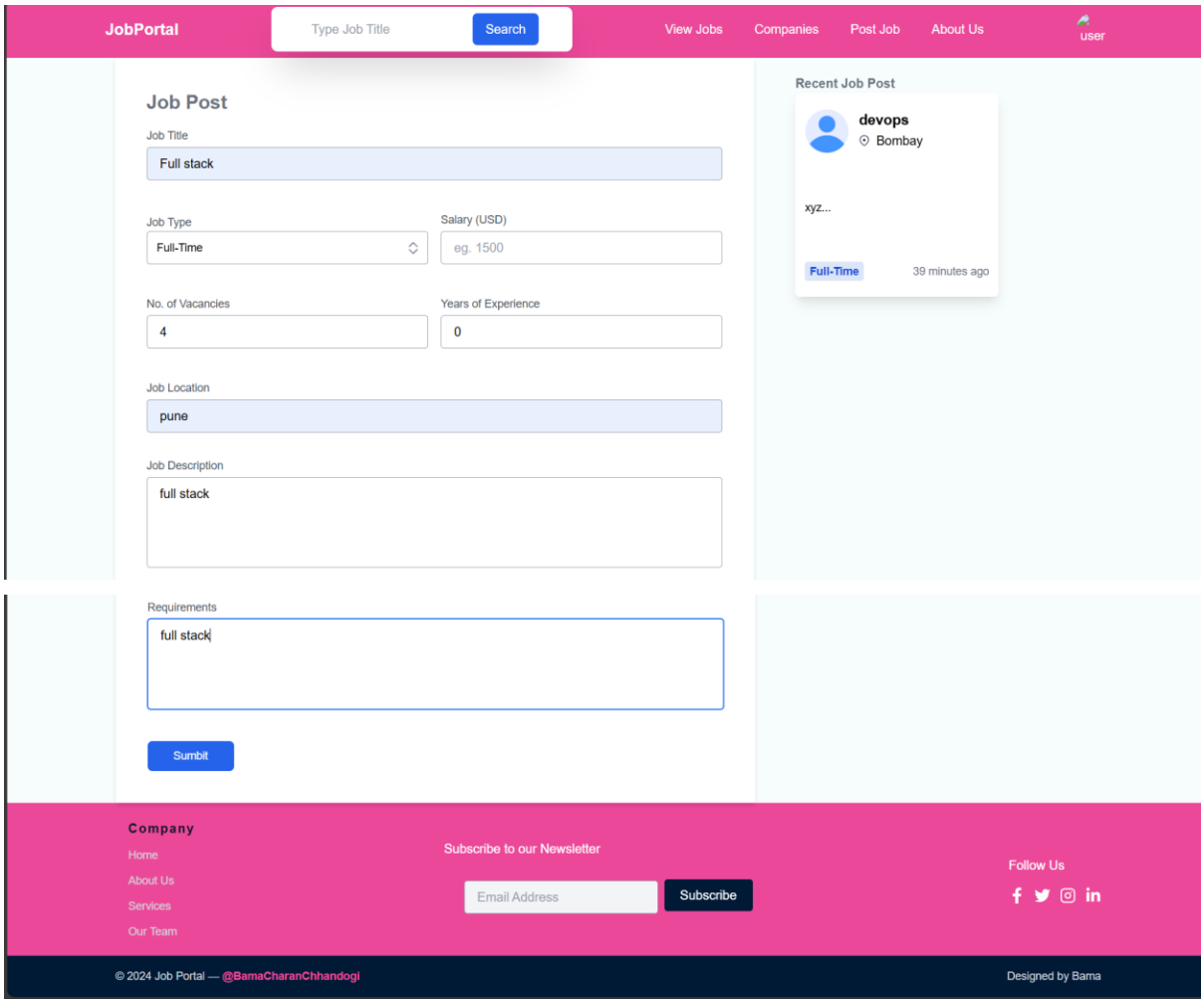


1.4 Company Dashboard

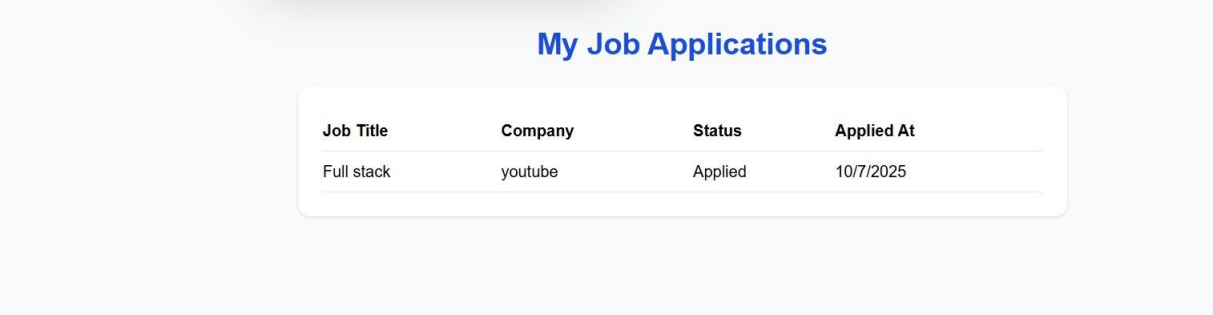


1.5 User Dashboard

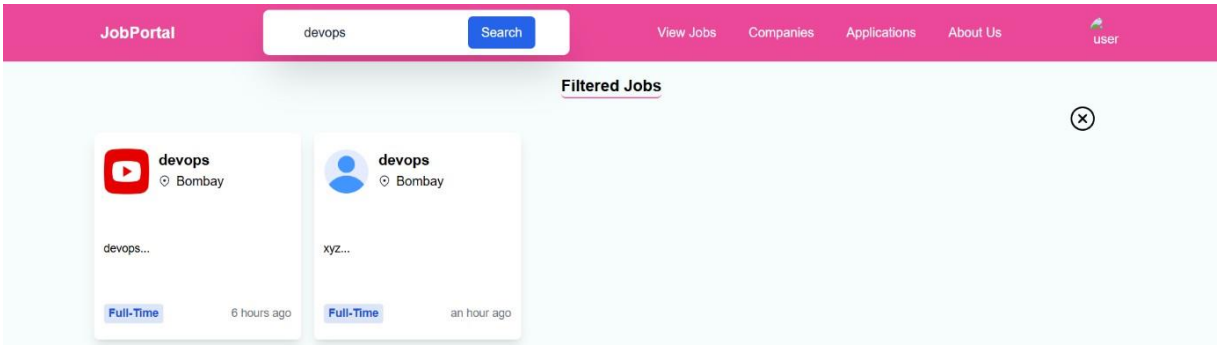
1.6 View Companies



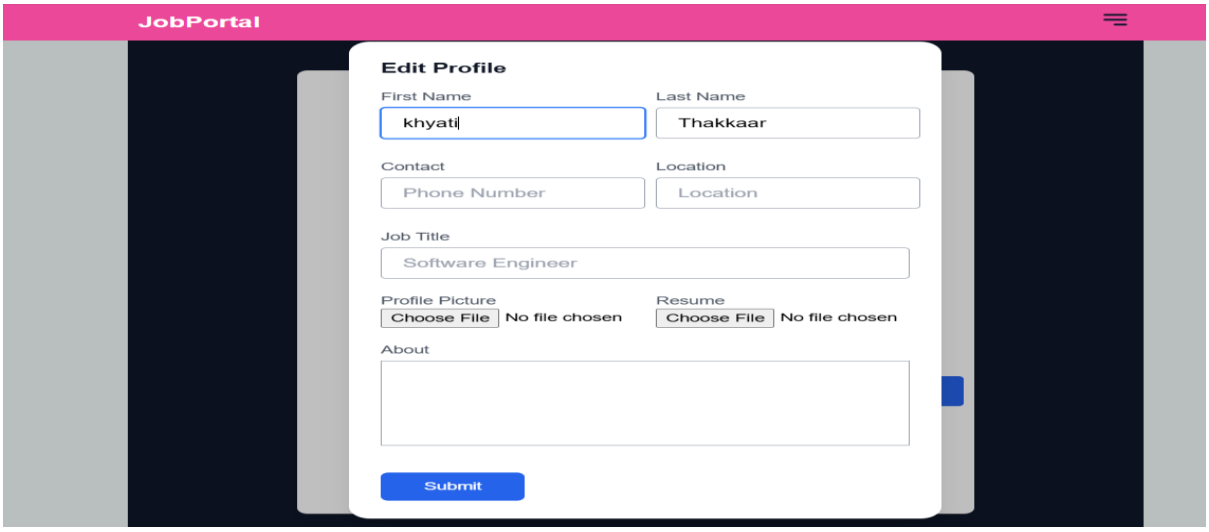
1.7 Post Job



1.8 My Job Application



1.9 Filtered Jobs



2.0 Edit Profile

7. LIMITATIONS OF PROJECT

7.1 No Real-Time Application Status Notifications

The system does not send automatic email or SMS notifications to seekers or companies when a job application is submitted, accepted, or rejected. All updates must be viewed manually in the dashboard.

7.2 No Resume File Uploads

Job seekers cannot upload resumes or supporting documents. All applications are limited to form-based submissions only.

7.3 No Payment or Subscription Integration

The platform does not include payment gateway integration for paid job postings, premium subscriptions, or featured listings.

7.4 Limited Search & Filter Options

Although the system supports basic filtering (e.g., by job title, location, salary), it lacks advanced search features such as keyword relevance ranking, category browsing, or saved searches.

7.5 Manual Application Management

Companies must manually review, accept, or reject applications. There is no automated screening or AI-powered shortlisting.

7.6 No Admin Panel

Currently, there is no separate admin dashboard for overseeing user accounts, moderating job postings, or generating system-wide reports.

7.7 Limited Analytics & Reporting

The dashboards provide only basic data on job postings and applications. Detailed analytics such as trends, user engagement metrics, or graphical reports are not implemented.

7.8 No Mobile App Support

The platform is web-based only. There is no dedicated Android or iOS mobile application, which limits accessibility for users preferring mobile apps.

8. OUTCOME

8.1 GAINED PRACTICAL EXPERIENCE IN FULL STACK DEVELOPMENT

- Developed a complete **Job Management System** using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**.
- Understood the end-to-end data flow between different user roles (**Company and Seeker**) from frontend interactions to backend logic and database operations.
- Implemented **secure role-based access** and built dynamic dashboards tailored to each user role, ensuring modularity, clarity, and scalability.

8.2 LEARNED CORE WEB DEVELOPMENT TOOLS AND TECHNOLOGIES

- Acquired practical knowledge of key web technologies like **React.js** (for interactive UI), **Express.js** (for backend logic), and **MongoDB** (for managing NoSQL databases).
- Used tools like **Postman** (for API testing), **Git & GitHub** (for version control and team collaboration), and **VS Code** (for streamlined development).
- Gained experience in deploying both frontend and backend using platforms like **Render** or **Vercel**.

8.3 BUILT A FUNCTIONAL JOB MANAGEMENT PLATFORM

- Created a working job portal where **companies can post and manage job openings**, and **seekers can search and apply for jobs**.
- Developed key modules such as **job posting**, **job filtering**, **application submission**, and **dashboard-based management** for both user types.
- Ensured a **responsive and accessible design**, delivering a smooth and modern user experience across devices.

8.4 IMPROVED CODING, DEBUGGING AND API INTEGRATION SKILLS

- Strengthened problem-solving and debugging skills by resolving **frontend-backend integration issues**, particularly in authentication and data flow.
- Learned to build and test **RESTful APIs**, handle **middleware** for validation and security, and implement **JWT-based authentication** for protected routes.
- Enhanced **state management** using React Hooks (e.g., useState, useEffect) to maintain dynamic and consistent UI behavior.

8.5 ENHANCED PROJECT MANAGEMENT AND PRESENTATION SKILLS

- Followed a structured approach to development by organizing frontend and backend codebases using clean code principles and reusable components.
- Gained confidence in **explaining system architecture**, functionalities, and technical challenges during presentations and reviews.
- Practiced documenting features, managing development progress, and showcasing the system in a clear, professional manner.

9. FUTURE ENHANCEMENTS

9.1 Resume Upload Functionality

Allow job seekers to upload their resumes and supporting documents when applying for jobs, enabling companies to review detailed applicant profiles.

9.2 Notification System (Email/SMS)

Implement a notification system using services like Twilio or Email JS to send confirmation emails and application status updates to seekers and companies automatically.

9.3 Payment Gateway Integration for Premium Features

Integrate secure payment gateways such as Razorpay or Stripe to enable companies to purchase premium listings, highlight job posts, or subscribe to additional services.

9.4 Advanced Search and Recommendation Engine

Develop intelligent job recommendations and advanced search filters that match seekers with relevant jobs based on their skills, experience, and interests.

9.5 Mobile Application Development

Create dedicated Android and iOS apps using React Native or Flutter to provide a more accessible and user-friendly experience for mobile users.

10. REFERENCES

The following references were used during the project event management system:

10.1 WEB REFERENCES

- [1] React.js Official Documentation – For building dynamic and responsive frontend components.
- [2] Node.js Official Documentation – For developing the backend server and RESTful APIs.
- [3] Express.js Official Documentation – For handling backend routing and middleware integration.
- [4] MongoDB Official Documentation – For managing NoSQL databases and Mongoose schemas.
- [5] Postman – For testing API endpoints and backend functionality.
- [6] GitHub – For version control, code management, and collaborative development.