

A
PROJECT REPORT
ON
Job Portal Website

SUBMITTED TO
SHIVAJI UNIVERSITY, KOLHAPUR
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF DEGREE
BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND
ENGINEERING

SUBMITTED BY

MR. Subahan Riyaj Mulla	22UCS093
MR. Shreyansh Nitin Patil	22UCS102
MR. Ashutosh Sanjay Langare	22UCS075
MR. Kunal Sunil Magdum	22UCS079

UNDER THE GUIDANCE OF
Prof. Y.M.Kamble



DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING
DKTE SOCIETY'S TEXTILE AND ENGINEERING
INSTITUTE, ICHALKARANJI
(AN EMPOWERED AUTONOMOUS INSTITUTE)
2024-2025

D.K.T.E. SOCIETY'S

**TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN EMPOWERED AUTONOMOUS INSTITUTE)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that, project work entitled

Job Portal Website

is a Bonafide record of project work carried out in this college by

MR. Subahan Riyaj Mulla	22UCS093
MR. Shreyansh Nitin Patil	22UCS102
MR. Ashutosh Sanjay Langare	22UCS075
MR. Kunal Sunil Magdum	22UCS079

is in the partial fulfillment of award of degree Bachelor of Technology in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2024-2025.

Prof. Y.M.Kamble
(PROJECT GUIDE)

PROF.(DR.) D.V.KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) L.S.ADMUTHE
(DIRECTOR)

EXAMINER: _____

DECLARATION

We hereby declare that, the project work report entitled “<<project title>>” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.Tech.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

Title	Name	PRN	Signature
MR.	Subahan Riyaj Mulla	22UCS093	
MR.	Shreyansh Nitin Patil	22UCS102	
MR.	Ashutosh Sanjay Langare	22UCS075	
MR.	Kunal Sunil Magdum	22UCS079	

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to <<guide name>> for his valuable guidance, support, and encouragement in the completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V. Kodavade for his cooperation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help, and assistance which they extended as and when required.

Thank you,

MR.	Subahan Riyaj Mulla	22UCS093
MR.	Shreyansh Nitin Patil	22UCS102
MR.	Ashutosh Sanjay Langare	22UCS075
MR.	Kunal Sunil Magdum	22UCS079

ABSTRACT

This project presents the development of a dynamic and user-friendly **Job Portal Website** built using the **MERN (MongoDB, Express.js, React.js, Node.js)** technology stack. The portal serves as a bridge between job seekers and employers, providing a platform where users can register, search and apply for jobs, while recruiters can post job openings, manage applications, and find suitable candidates efficiently.

The system is designed with a focus on performance, scalability, and user experience. **MongoDB** ensures flexible and efficient data storage, while **Express.js** and **Node.js** form a robust backend API. **React.js** powers the frontend, providing an interactive and responsive user interface. The portal includes essential features such as user authentication, job posting and searching, application tracking, and role-based access for job seekers and recruiters.

The project not only demonstrates the practical application of the MERN stack but also highlights the importance of full-stack development in creating modern web solutions. This portal aims to simplify the recruitment process and contribute to the growing digital employment ecosystem.

INDEX

1. Introduction	7
a. Problem definition	7
b. Aim and objective of the project	7
c. Scope and limitation of the project	7
d. Timeline of the project	8
e. Project Management Plan	8
2. Background study and literature overview	9
a. Literature overview	9
b. Critical appraisal of other people's work	9
c. Investigation of current project and related work	9
3. Requirement analysis	10
a. Requirement Gathering	10
b. Requirement Specification	10
c. Use case Diagram	11
4. System design	12
a. System architecture	12
b. Architecture layers	12
c. Algorithmic description of each module	13
d. Data flow diagram	13
5. Implementation	14
a. Environmental Setting for Running the Project	14
b. Detailed Description of Methods	15
c. Implementation Details	16
6. Integration and Testing	17
a. Description of the Integration Modules	17
b. Testing	17
7. Performance Analysis	18
8. Future Scope	18
9. Applications	18
10. References	18

Introduction

In the modern era, digital platforms have revolutionized the way people find and apply for jobs. Traditional job-hunting methods are time-consuming and inefficient, prompting the need for a more streamlined and accessible system. This project aims to provide a comprehensive Job Portal Website using the MERN stack, designed to connect job seekers with recruiters effectively and efficiently. It integrates modern web technologies to create a seamless user experience for all stakeholders involved in the recruitment process.

a. Problem Definition

Despite the presence of several job platforms, many lack user-friendly design, real-time communication, and efficient filtering systems. Additionally, smaller companies often struggle to reach suitable candidates due to high platform fees or lack of visibility. Job seekers, especially freshers, face challenges like irrelevant job listings, poor application tracking, and lack of communication from employers. This project addresses these issues by developing a job portal that focuses on simplicity, functionality, and accessibility for both job seekers and employers.

b. Aim and Objective of the Project

Aim:

To develop a responsive and functional Job Portal Website using the MERN stack that bridges the gap between job seekers and employers.

Objectives:

- Implement user registration and login functionalities for both recruiters and applicants.
- Allow recruiters to post job listings and manage applications.
- Provide job seekers with a personalized dashboard to search, filter, and apply for jobs.
- Integrate real-time features like notifications or application tracking.
- Ensure the platform is scalable, secure, and mobile-friendly.

c. Scope and Limitation of the Project

Scope:

- The project supports two primary users: Job Seekers and Recruiters.
- Functional modules include user authentication, job posting, job

searching, application tracking, and role-based access.

- It is designed using the MERN stack, ensuring a modern and maintainable codebase.

Limitations:

- The platform does not include video interviews or resume parsing capabilities.
- Third-party integrations (e.g., LinkedIn API, resume builders) are not implemented.
- Email/SMS notifications may be limited to basic alerts due to time and resource constraints.

d. Timeline of the Project

Phase	Duration	Task
Requirement Analysis	Week 1	Understanding features and defining goals
UI/UX Design	Week 2	Wireframing and prototyping
Backend Development	Weeks 3 – 4	Setting up Node.js, Express, and MongoDB
Frontend Development	Weeks 5 – 6	React integration and UI development
Integration & Testing	Week 7	Full-stack integration and bug fixing
Deployment	Week 8	Hosting on a platform (e.g., Render, Vercel)
Final Report Preparation	Week 9	Documentation and report writing

e. Project Management Plan

The project followed the **Agile methodology**, with weekly sprints and continuous feedback loops. Each sprint focused on a specific module or feature set, followed by testing and review. Tools like Trello or Notion were used to track progress, while Git was used for version control. Tasks were divided based on frontend and backend development responsibilities, ensuring parallel progress.

Background Study and Literature Overview

a. Literature Overview

Several research studies and technical papers highlight the design and functionality of job portals and recruitment systems:

- **"Online Job Portal System using MERN Stack" (2022)** discusses how MERN technologies are ideal for modern web applications due to their modular structure, rapid development capabilities, and scalability.
- **"E-Recruitment System: Design and Development" (IJERT, 2021)** emphasized the importance of digital recruitment tools in reducing hiring costs and time-to-hire.
 - **"Comparative Study of Job Portals" (IRJET, 2020)** reviewed portals like Naukri, Indeed, and Monster, concluding that while they offer a wide reach, they often lack customization and user-friendliness.

b. Critical Appraisal of Other People's Work

Popular Platforms Analyzed:

1. **Naukri.com** – One of India's largest job portals, but cluttered UI and overwhelming ad content often lead to a poor user experience.
2. **LinkedIn** – Powerful for networking, but often inaccessible to freshers or non-professional users due to its professional focus and complexity.
3. **Indeed** – Offers global reach but lacks strong recruiter-side controls and tailored applicant tracking features.

Limitations Identified:

- Poor user interface and responsiveness on smaller devices.
- Limited job filtering based on skills, experience, and preferences.
- Lack of real-time communication or notification features.
- High dependency on premium features for better visibility.

c. Investigation of Current Project and Related Work

The Job Portal Website created under this project draws inspiration from existing systems while aiming to improve functionality, accessibility, and user engagement.

Key Differentiators in This Project:

- **Built using MERN stack** for speed, scalability, and maintainability.
- **Custom role-based dashboards** for recruiters and job seekers.
- **Efficient filtering/search system** for jobs based on location, skillset, and experience.
- **Secure authentication system** using JWT for protected routes.
- **Real-time updates and application tracking** to improve interaction.

Requirement Analysis

a. Requirement Gathering

Requirement gathering was carried out through informal surveys, existing platform studies, and observation of real-world recruitment issues. The primary users identified are:

- **Job Seekers** – looking to register, search, and apply for jobs.
- **Recruiters** – aiming to post job vacancies and manage applications.
- **Admin** (optional) – for overseeing platform operations and user activities.

Methods used for gathering requirements:

- Observation of platforms like Naukri, Indeed, and LinkedIn.
- Feedback from potential job seekers (students, freshers).
- Input from recruiters (faculty and HR references).

b. Requirement Specification

The requirements are categorized into **functional** and **non-functional**:

Functional Requirements

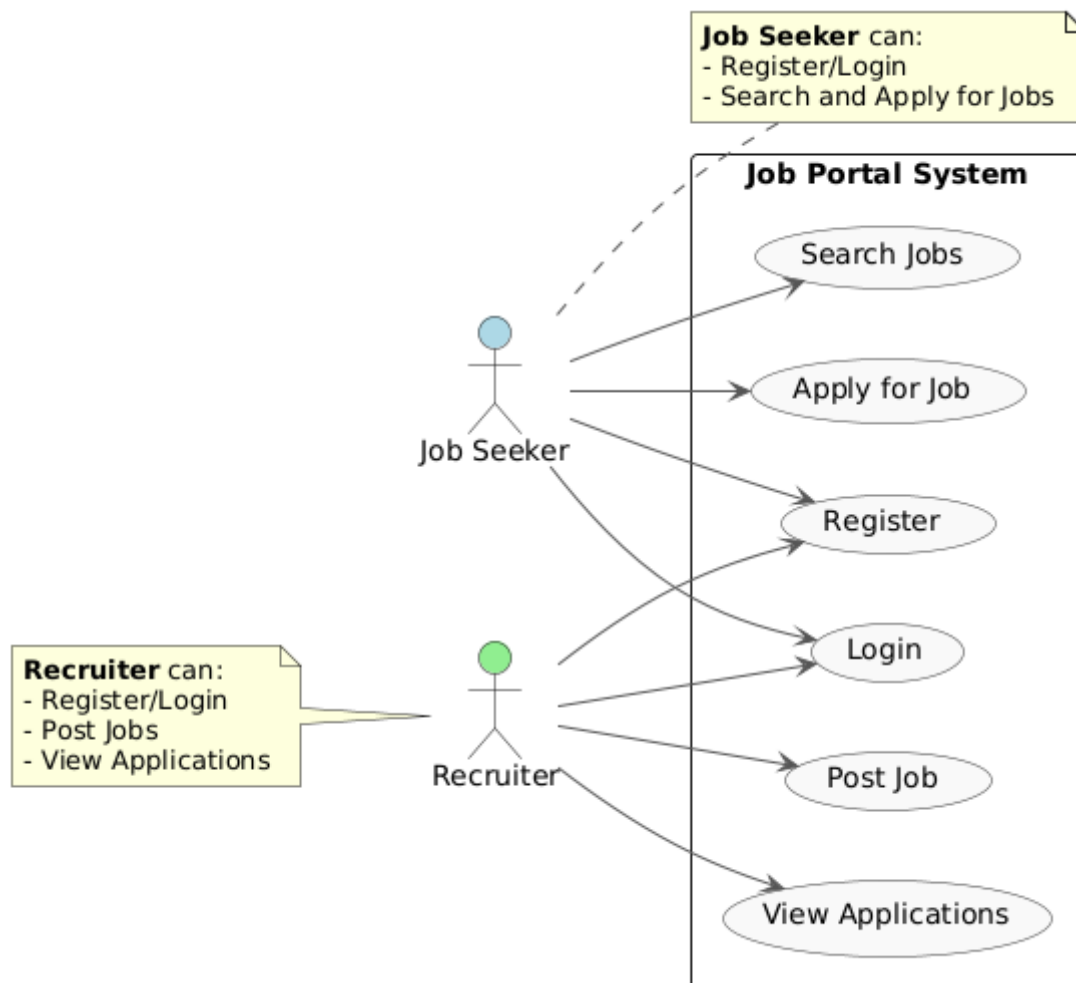
- **User Registration and Login** (with role-based access).
- **Job Posting and Editing** (for recruiters).
- **Job Search and Apply** (for job seekers).
- **Application Status View** (applied, shortlisted, rejected).
- **Dashboard View** (customized for recruiters and seekers).
- **Job Filters** (based on location, title, type, etc.).
- **Authentication and Authorization** (using JWT).

Non-Functional Requirements

- **Responsiveness** – Must support mobile, tablet, and desktop.

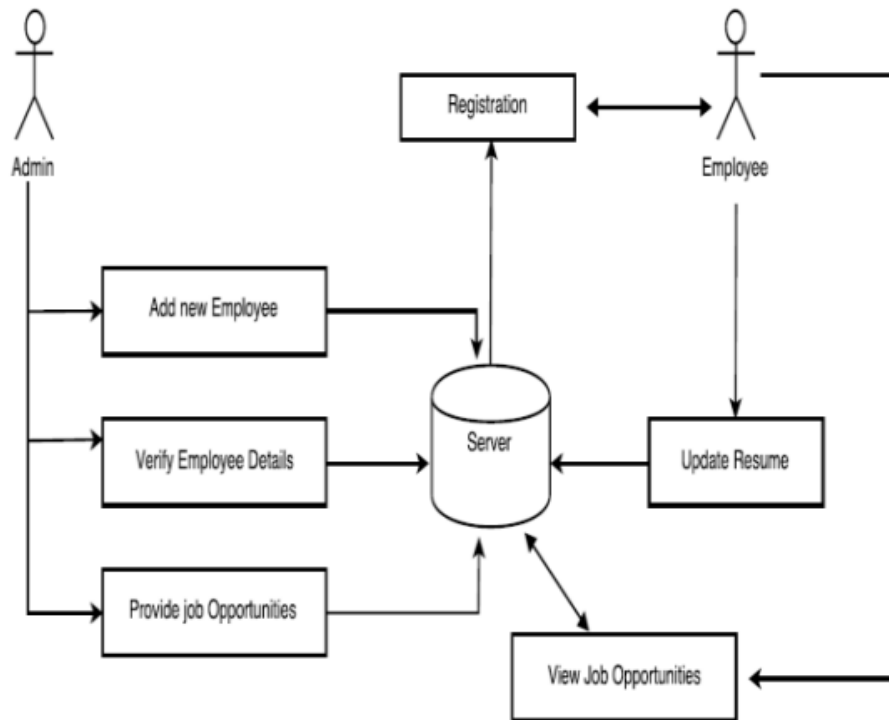
- **Performance** – Fast loading and efficient rendering using React.
- **Security** – Data protection through hashed passwords and secure sessions.
- **Scalability** – Designed to support an increasing number of users/jobs.
- **Maintainability** – Clean architecture using reusable components and modular backend.

c. Use Case Diagram

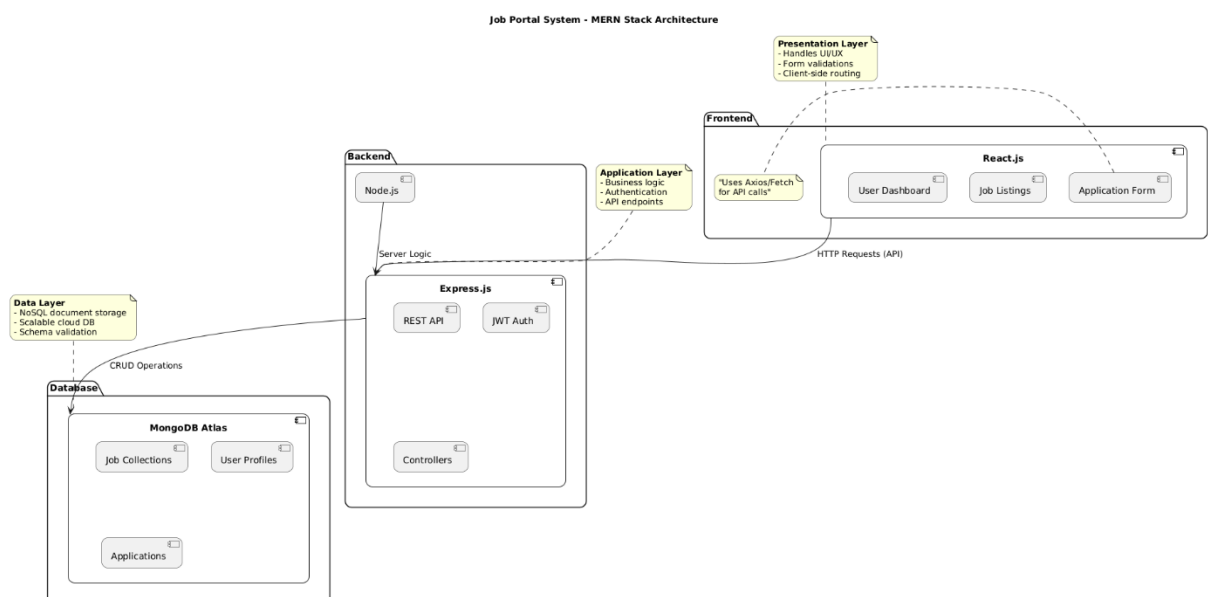


System design

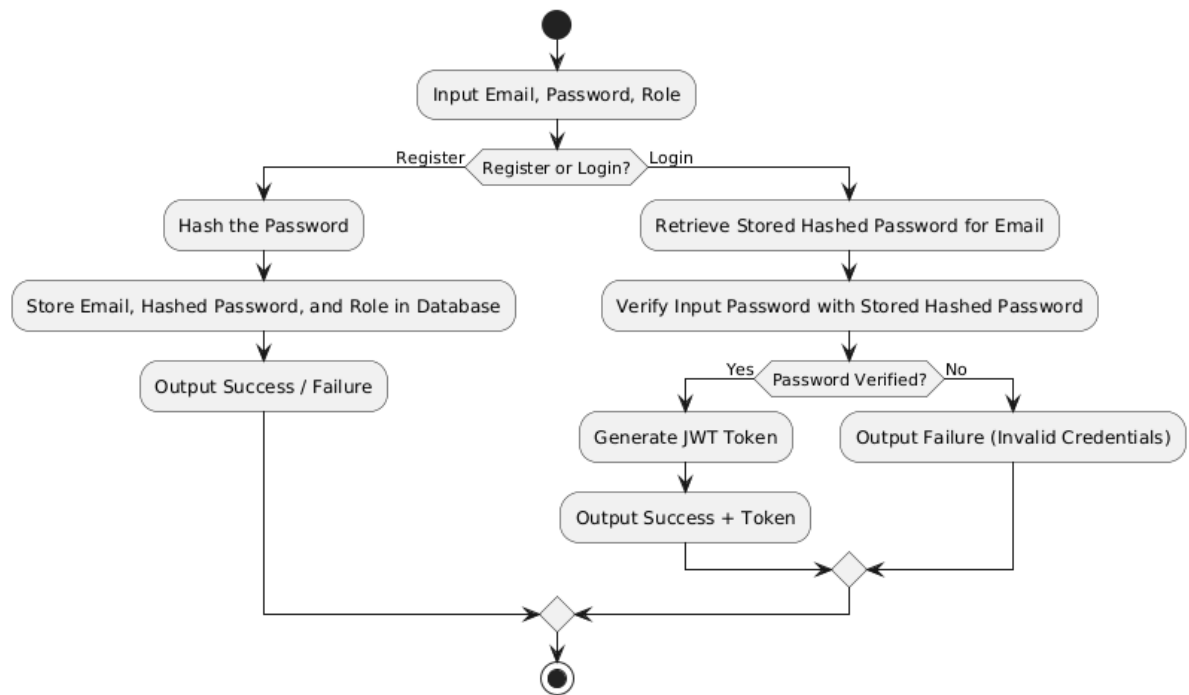
a. System architecture



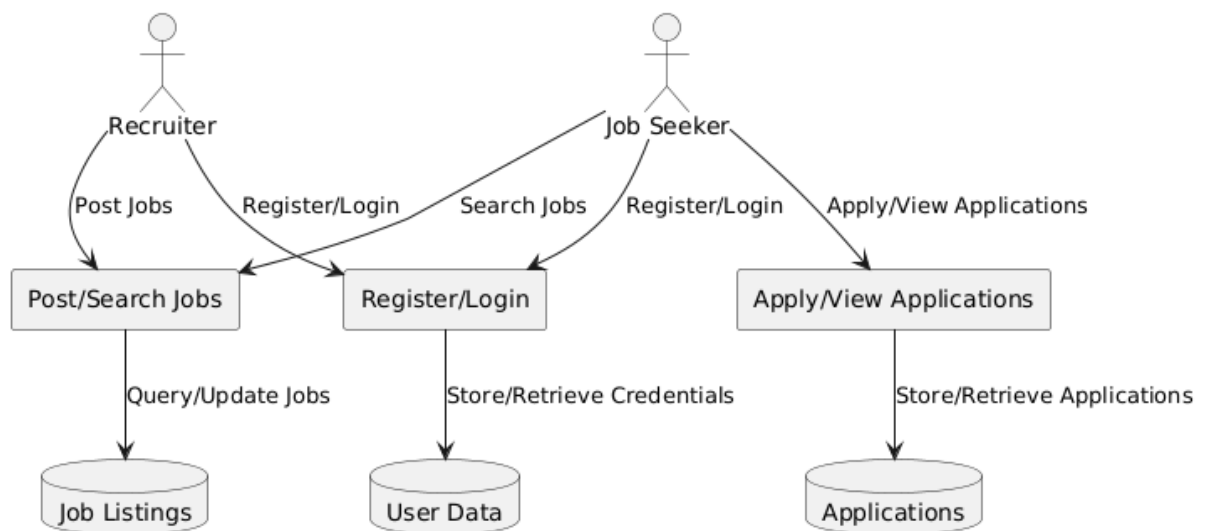
b. Architecture layers



c. Algorithmic description of each module



d. Data flow diagram



Implementation

a. Environmental Setting for Running the Project

To develop and run the Job Portal Website using the MERN stack, the following environment setup was used:

Software Requirements:

- **Operating System:** Windows 10 / Ubuntu 20.04
- **Code Editor:** Visual Studio Code
- **Browser:** Google Chrome
- **Version Control:** Git and GitHub
- **Package Managers:** Node Package Manager (NPM), Yarn (optional)

Technology Stack:

- **Frontend:** React.js (v18+)
- **Backend:** Node.js (v18+) with Express.js
- **Database:** MongoDB Atlas (cloud-hosted)
- **Authentication:** JSON Web Tokens (JWT), bcrypt for password hashing
- **Deployment Tools:** Render/Heroku (backend), Vercel/Netlify (frontend)

b. Detailed Description of Methods

The project is divided into three main layers, each serving a specific purpose:

1. Frontend Methods (React.js):

- Components are created for:

- Job Seeker and Recruiter Dashboards
 - Job Listings
 - Registration/Login Forms
- Axios is used to make HTTP requests to backend APIs.
- Routing is handled using React Router.
- State management with React's useState, useEffect and context APIs.

2. Backend Methods (Node.js + Express):

- RESTful APIs are implemented to handle:
 - User authentication (register/login)
 - Job posting, retrieval, and filtering
 - Job application handling
- Routes are structured under /api/users, /api/jobs, /api/applications.
- Middleware is used for authentication (JWT token verification).
- MongoDB Mongoose is used to define schemas for Users, Jobs, and Applications.

3. Database (MongoDB):

- Collections used:
 - **Users:** Contains Job Seekers and Recruiters
 - **Jobs:** Contains job postings with title, description, and recruiter ID
 - **Applications:** Links between users and job IDs

c. Implementation Details

User Authentication:

- On registration, passwords are hashed using bcrypt and stored securely.
- On login, JWT tokens are generated and sent to the client.
- Protected routes on backend verify token before granting access.

Job Posting and Searching:

- Recruiters can create job posts using a form.
- Jobs are fetched using GET requests and rendered in a list.
- Filters such as location, role, or company can be applied.

Job Application:

- Job Seekers can click "Apply" on a job, which creates an entry in the Applications collection.
- Duplicate applications are prevented using Mongoose constraints.

Admin Access (Optional):

- Admin accounts can view all users and jobs.
- This is protected using an isAdmin flag and middleware.

Deployment:

- The frontend is hosted on **Vercel**, linked to the GitHub repository.
- The backend is deployed on **Render**, connected with environment variables.
- **MongoDB Atlas** is used for cloud database access.

Integration and Testing

a. Description of the Integration Modules

The project integrates three key layers: **frontend (React.js)**, **backend (Node.js + Express)**, and **database (MongoDB Atlas)**.

- **Frontend–Backend Integration:** Axios is used for API calls like login, job posting, and applying to jobs.
- **Backend–Database Integration:** Mongoose is used to define schemas and connect to MongoDB.
- **Authentication Integration:** JWT is used for secure login and route protection, stored in localStorage and verified by middleware.

b. Testing

Several types of testing were performed:

- **Manual Testing:** All forms and workflows were tested with valid and invalid inputs.
- **API Testing:** Postman was used to test all backend endpoints.
- **Integration Testing:** Ensured seamless interaction between frontend, backend, and database.
- **Tools Used:** Postman, MongoDB Compass, browser dev tools, and console logs for debugging

Performance Analysis

The system performs efficiently under normal usage. API responses are quick, and MongoDB handles data operations smoothly. Basic load testing showed the site remains responsive with multiple simultaneous users. Optimized database queries and minimal frontend re-renders improved overall performance.

Future Scope

- Add AI-based job recommendations
 - Enable resume parsing and matching
 - Add admin dashboard for advanced analytics
 - Implement email/SMS job alerts
 - Mobile app version for wider reach
-

Applications

- Helps job seekers find relevant opportunities quickly
 - Enables recruiters to post and manage job listings
 - Can be used by colleges for campus placements
 - Useful for startups and small businesses hiring remotely
-

References

1. MongoDB Documentation, "Mongoose ODM," mongoosejs.com, 2023.
2. React Documentation, "React - A JavaScript library," reactjs.org, 2023.
3. Express Documentation, "Express.js Web Framework," expressjs.com, 2023.
4. JWT.io, "JSON Web Token Introduction," jwt.io, 2023.