A
REPORT
ON

JOB-PORTAL

SUBMITTED BY

BHAVIN A. MAKWANA

ARYAN MEHTA

B. Tech CE, SEM VI

SUBJECT : **SYSTEM DESIGN PRACTICES**

GUIDED BY

Prof. JIGAR M. PANDYA

Computer Engineering Department

Dharamsinh Desai University

2023-2024

DHARAMSINH DESAI UNIVERSITY

COLLEGE ROAD, NADIAD-387001

DDU

DHARAMSINH DESAI UNIVERSITY

Certificate

This is to certify that,

Mr. BHAVIN A. MAKWANA has completed **SYSTEM DESIGN PRACTICES** Project work for **Semester VI** having title **JOB-PORTAL,** in a group of **TWO** persons under the guidance of the Faculty Guide **Prof. JIGAR M. PANDYA**

Prof. JIGAR M. PANDYA
Assistant Professor,
Computer Engineering Department
Dharamsinh Desai University,
Nadiad

Dr. C.K. BHENSDADIA
Head of Department
Computer engineering,
Dharamsinh Desai University,
Nadiad

2023-2024
DHARAMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD-387001

DDU

DHARAMSINH DESAI UNIVERSITY

Certificate

This is to certify that,
Mr. ARYAN MEHTA has completed **SYSTEM DESIGN PRACTICES** Project work for **Semester VI** having title **JOB-PORTAL,** in a group of **TWO** persons under the guidance of the Faculty Guide **Prof. JIGAR M. PANDYA**

Prof. JIGAR M. PANDYA
Assistant Professor,
Computer Engineering Department
Dharamsinh Desai University,
Nadiad

Dr. C.K. BHENSDADIA
Head of Department
Computer engineering,
Dharamsinh Desai University,
Nadiad

2023-2024
DHARAMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD-387001

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards our project guide Prof. “JIGAR M. PANDYA” for giving her continuous valuable guidance, comments and suggestions throughout the project. Last but not the least, the successful completion of our project would not have been possible without the dedicated support from all our faculty members and group-member.

INDEX

ABSTRACT

The Job-Portal is an innovative web-based application designed to facilitate the recruitment process for both job seekers and employers with a interactive and user friendly interface. Website with two user panels for job-seekers and job employer. The tedious task to get a job or recruit for a job-position will be simple in much easier and few steps. Will provide wide options to a job-seeker and a vast society of jobbers for a employer to select.

1. INTRODUCTION

1.1 PROBLEM SUMMARY

1.1.1 Problem Identification

In previous we found problems of manually or on physical basis recruitment process for a job. Tedious job to find a job for job seeker and also for a employer to recruit.

1.1.2 Problem Solution

In solution to this we developed a website which provides services for both job-seekers and employers as well. A wide community of job-seekers can be reached out by employer through this app and application for a job will be on one touch for a job seeker.

2. Software Requirement and Specification

R 1. LOGIN AS JOB-SEEKER

DESCRIPTION : Here, the JOB-SEEKER can create an account on our website as a user. And can also add the basic details of themselves.

R 1.1 : CREATE ACCOUNT

INPUT : Account details (role, Name, Email, Phone-Number, Password) will be provided

OUTPUT : Details of particular user will be added and registered.

R 1.2 : LOGIN TO ACCOUNT

INPUT : Account details (Login as role, Email, Password) will be provided.

OUTPUT : Details of particular user will be logged in after valid credentials.

R 1.3 : JOB DETAILS

INPUT : Click on “JOB DETAILS” button.

OUTPUT : User will be able to display details of jobs like title, category, country, city, location, description, job posted date, salary and a button to apply for that job

R 1.4 : APPLICATION FORM

INPUT : Click on “Apply Now” button.

OUTPUT : User will be able apply for job by entering details likename, email, phone number, address, cover letter, uploading resume and click on send application button

R 1.5 : MY APPLICATION PAGE

INPUT : Click on “MY APPLICATION” page.

OUTPUT : User will be redirected my application page displaying Name, Email, Phone number, Address, Cover Letter details and a delete button.

R 1.5.1 : DELETE APPLICATION

INPUT : Click on “DELETE APPLICATION” button.

OUTPUT : User application will be deleted

R 2. LOGIN AS EMPLOYER

DESCRIPTION : Here functionalities like all jobs published, applicants list, new job functionality, jobs by specific employer published list will be available.

R 2.1 : ALL JOBS

INPUT : Click on “ALL JOBS” button.
OUTPUT: All jobs will be displayed.

R 2.1.1 : JOB DETAILS

INPUT : Click on “JOB DETAILS” button.
OUTPUT: Specific job detail will be displayed.

R 2.2 : APPLICANTS LIST

INPUT : Click on “APPLICANTS APPLICATIONS” button.
OUTPUT: All application from job seekers will be displayed.

R 2.3 : POST NEW JOB

INPUT : Click on “POST NEW JOB” button.
OUTPUT : A employer will be able to add new job by giving details like job title, selecting category, country, city, location, salary type, job description.

R 2.4 : VIEW JOBS

INPUT : Click on “VIEW ALL JOBS” button.
OUTPUT: List of posted jobs will be displayed with details and an edit and delete option.

R 3. Other Nonfunctional Requirements

R 3.1 performance requirements:

N.1. The system should have a response time of less than 2 second for all user Requests.

N.2. Application can process their resume 98% of the time without failure.

R 3.2 safety requirements:

N.3. Database should be backed up every hour.

N.4. Under failure, the system should be able to come back to normal operation within an hour.

R 3.3 security requirements

N.5. All external communication between the data's server and client must be encrypted.

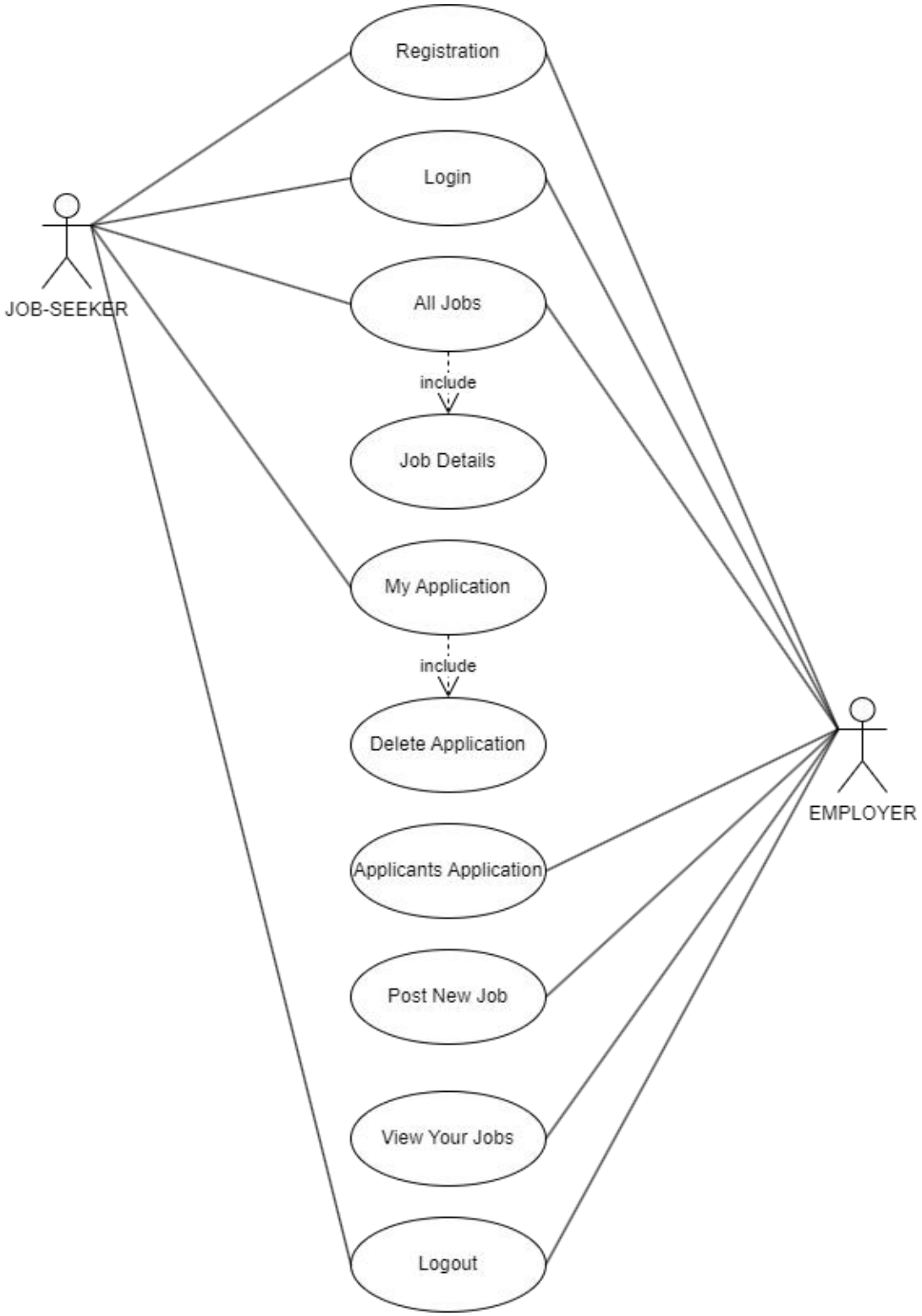
N.6. All data must be stored, protected or protectively marked.

N.7. Payment process should use HTTP over secure to secure the payment transaction.

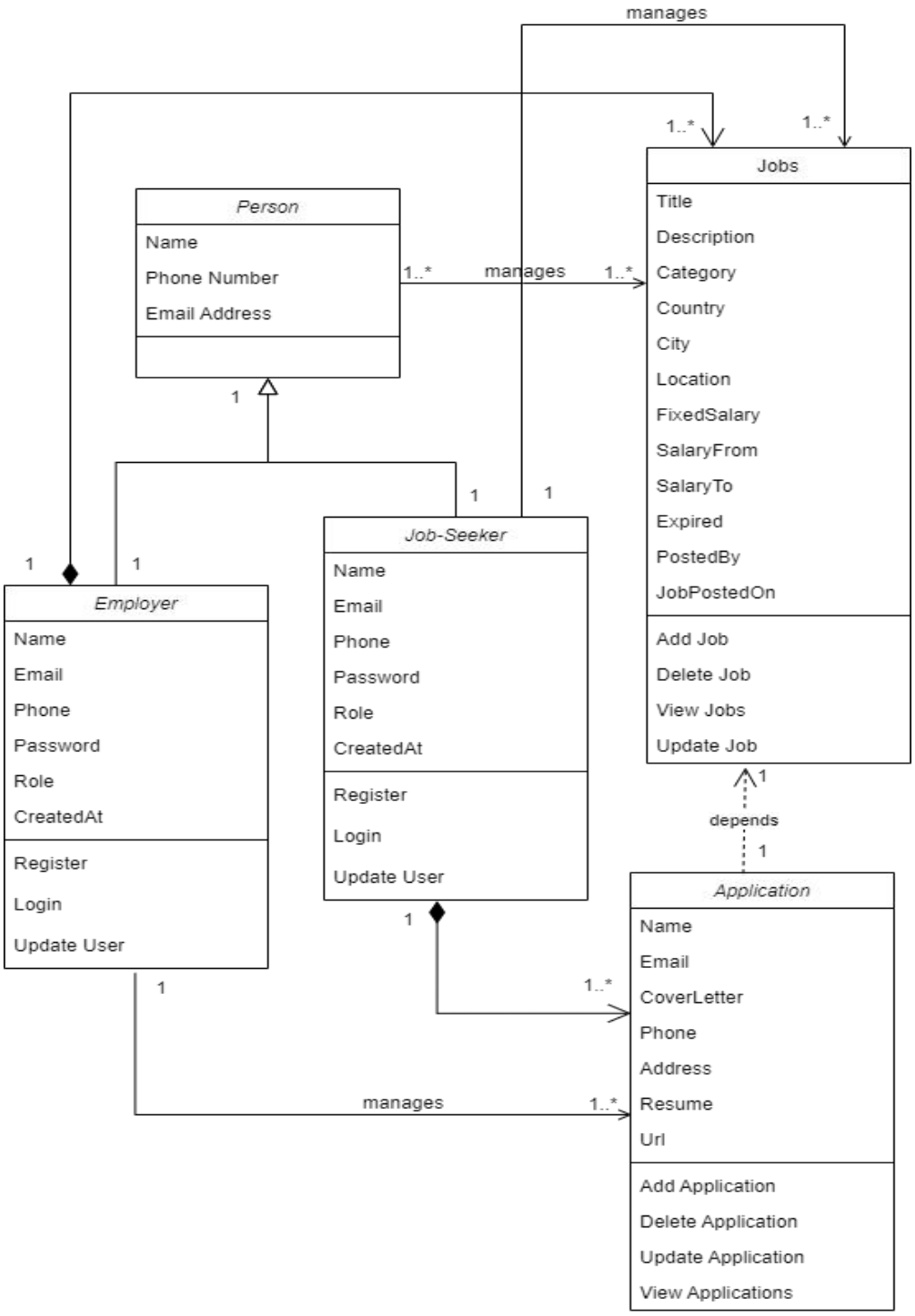
R 3.5 Design Requirements

N.8 The system should have a user-friendly interface that is easy to navigate and understand.

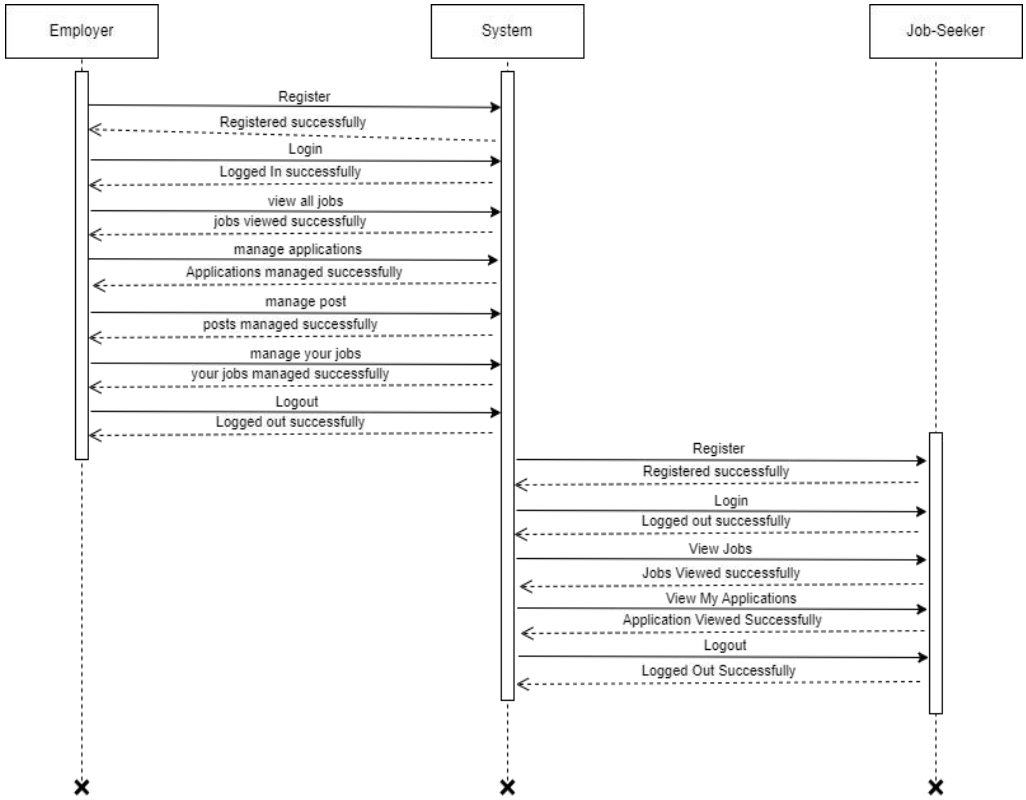
3.1 Use-Case Diagram



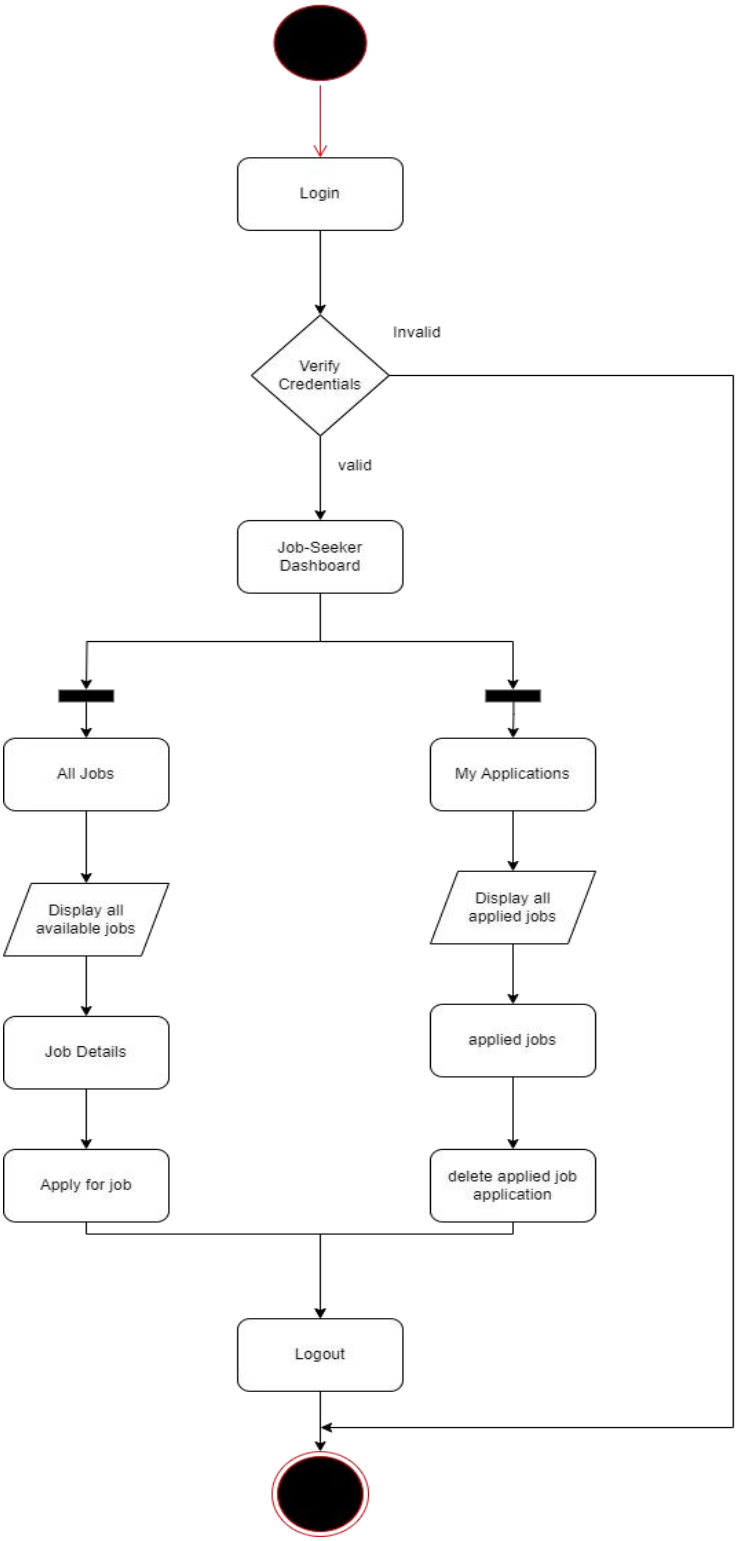
3.2 Class Diagram



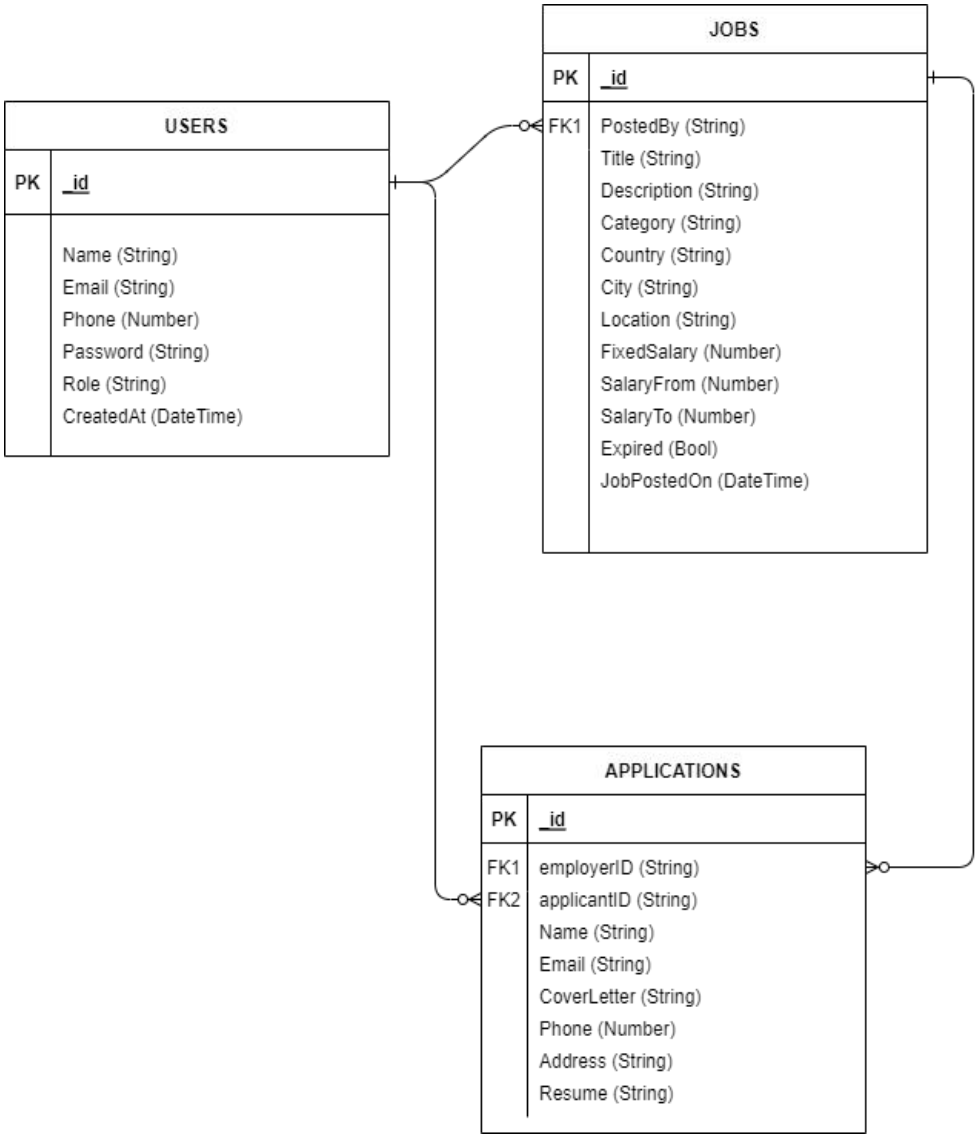
3.3 Sequence Diagram



3.4 Activity Diagram



3.5 ER Diagram



4. Data Dictionary

Jobs Table

Field Name	Data Type	Field Length	Constraints	Description
Id	Long	100	Primary key	Job id
Title	String	30	-	Title for job
Description	String	500		Description Of Job
Category	String	-		Category Of Job
Country	String	-		Country of job location
City	String	-		City of Job Location
Location	String	20		Address Of Job
FixedSalary	Number	9		Fixed Salary of Job
SalaryFrom	Number	9		Minimum range of salary
SalaryTo	Number	9		Maximum range of salary
Expired	Bool	-		Job Expiry
PostedBy	Long	100	Foreign Key	Id of Employer who posted job
JobPostedOn	DateTime	-	-	Date of Job Posted

User Table

Field Name	Data Type	Field Length	Constraints	Description
Id	Long	100	Primary Key	User Id
Name	String	30	-	Name of User
Email	String	-	-	Email Of User
Phone	Number	-	-	Number Of User
Password	String	32	-	Password Of User
Role	String	-	-	Role Of User(Employer/Job-Seeker)
CreatedAt	DateTime	-	-	Date of User Account Creation

Application Table

Field Name	Data Type	Field Length	Constraints	Description
Id	Long	100	Primary Key	ApplicationID
Name	String	30		Name of Applicant
Email	String	-		Email Of Applicant
CoverLetter	String	-		Description provided by Job-Seeker
Phone	Number	-		Number Of Job-Seeker
Address	String	-		Address Of Job-Seeker
Resume	-	-		Resume Of Job-Seeker
ApplicationID	Long	-	-	Id Of Application
EmployerID	Long	-	Foreign Key	Id Of Employer

4. Implementation

We have made the use of MERN Technology and implemented using the using react language.

In our job portal project developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack, the React implementation plays a crucial role in delivering a dynamic and responsive user interface. React, being a JavaScript library for building user interfaces, provides us with a component-based architecture that facilitates the creation of reusable UI components.

Language Used

React -

One of the key aspects of our React implementation is the creation of components that represent various parts of the job portal application, such as job listings, user profiles, and application forms. These components are designed to be modular and composable, allowing for easy integration and maintenance.

Furthermore, we leverage React's state management capabilities to manage the application's data flow efficiently. Stateful components maintain the application's state and handle user interactions, while stateless functional components are used for presentational purposes, keeping the UI clean and manageable.

Framework Used

Express.js -

The implementation of Express.js in our job portal project involves defining routes and controllers to handle various CRUD (Create, Read, Update, Delete) operations related to job listings, user profiles, authentication, and other functionalities. Express routes are designed to map specific HTTP requests to corresponding controller functions, allowing for organized and maintainable code structure.

Furthermore, Express middleware is utilized to enhance the functionality and security of our job portal application. Middleware functions are employed for tasks such as parsing incoming request bodies, validating user input, authenticating users, and handling errors gracefully. This middleware-based approach enables us to implement cross-cutting concerns effectively and ensures consistent behavior across different parts of the application.

Modules Used

Node Modules -

One of the primary functions of Node.js in our job portal project is handling HTTP requests from the client-side React application. These requests are routed through Express.js endpoints, which are responsible for processing incoming data, performing necessary operations such as CRUD (Create, Read, Update, Delete) operations on the MongoDB database, and sending back appropriate responses to the client.

Database Used

MongoDB

The implementation of MongoDB in our job portal project involves designing a schema-less database structure that accommodates the dynamic nature of job-related data. MongoDB's document-oriented model allows us to store data in a JSON-like format, which closely aligns with the structure of data objects used in our React.js frontend components. This schema flexibility enables rapid development and iteration, as the database can easily adapt to changing requirements without requiring costly schema migrations.

Tokenizer Used

JSONWebTokens -

The implementation of JSON Web Tokens in our job portal project involves several key aspects. Firstly, when a user successfully logs in or registers on the platform, a JWT is generated on the server-side using a secret key known only to the server. This token contains encoded information about the user's identity and any relevant permissions or roles associated with their account.

Once generated, the JWT is then securely transmitted to the client-side React application and stored either in browser cookies or local storage for subsequent requests. This token serves as a bearer token, allowing the user to access protected routes and resources within the application without needing to re-authenticate with every request.

Employer Module:

Authentication: This module manages employer authentication, allowing employers to register new accounts, log in securely, and manage their authentication credentials.

Profile Management: Employers can create and manage their profiles, including company details, contact information, and profile picture.

Job Posting: This functionality enables employers to post new job listings, providing details such as job title, description, requirements, location, and application deadlines.

Job Management: Employers can view, edit, and delete their existing job listings, as well as track the status of applications received for each job.

Candidate Screening: Employers may have access to tools for screening and evaluating job applications by a list of job-applicants.

Job-seeker Module:

Authentication: Similar to the employer module, this module handles jobseeker authentication, enabling jobseekers to create accounts, log in securely, and manage their authentication credentials.

Profile Management: Jobseekers can create and maintain their profiles, including personal details, work experience, education, skills, resume upload, and profile picture.

Job Search: This functionality allows jobseekers to search for job listings based on various criteria such as job title, location, industry manually.

Application Submission: Jobseekers can apply for job listings by submitting their resumes, cover letters, and other required documents. They may also track the status of their applications.

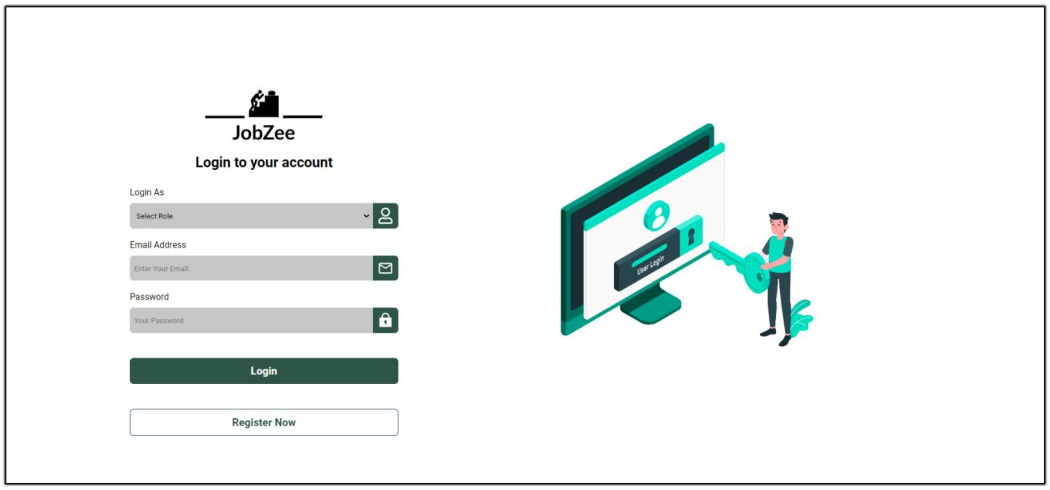
These modules provide the core functionality required for employers and jobseekers to interact within the job portal application, streamlining the recruitment process and enhancing the overall user experience. Additional modules or features can be integrated as needed to further enhance the functionality and usability of the application.

6. Testing

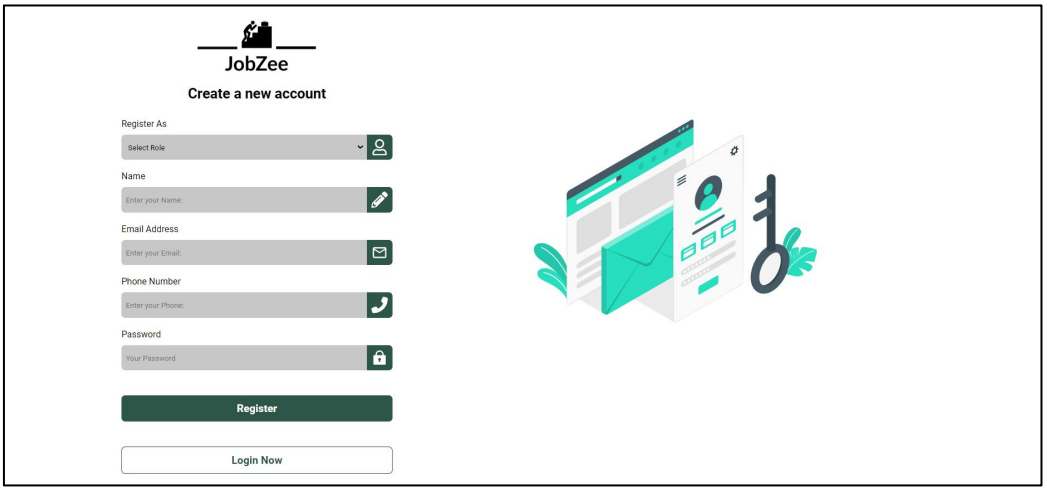
Sequence	Input	Output	Expected Output	Pass/Fail
1	Wrong Login Credentials	Error For Not Match	Error For Not Matched	Pass
2	Correct Login Credentials	Login Successful	Login Successful	Pass
3	Wrong Registration Value	Error For Wrong Entered Value	Error For Wrong Entered Value	Pass
4	Correct Entered Registration Value	Registered Successfully	Registered Successfully	Pass
5	Resource Request	Resource invalid or not found	Resource invalid or not found	Pass
6	Resource Request	Resource valid and proceed	Resource valid and proceed	Pass
7	Invalid Webtoken	Webtoken is invalid	Webtoken is invalid	Pass
8	Expired Token	Webtoken is Expired	Webtoken is Expired	Pass

7. Screenshots

Login Page

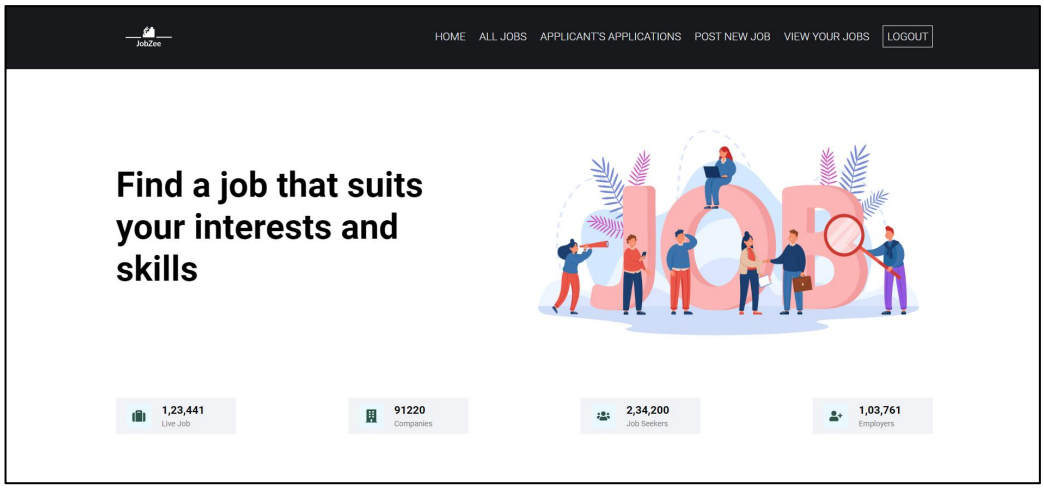


Registration Page

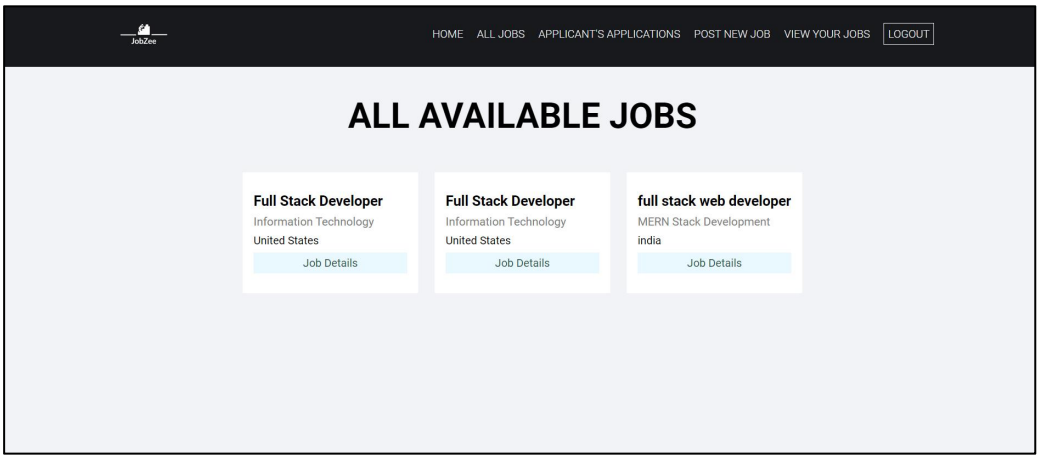


Employer

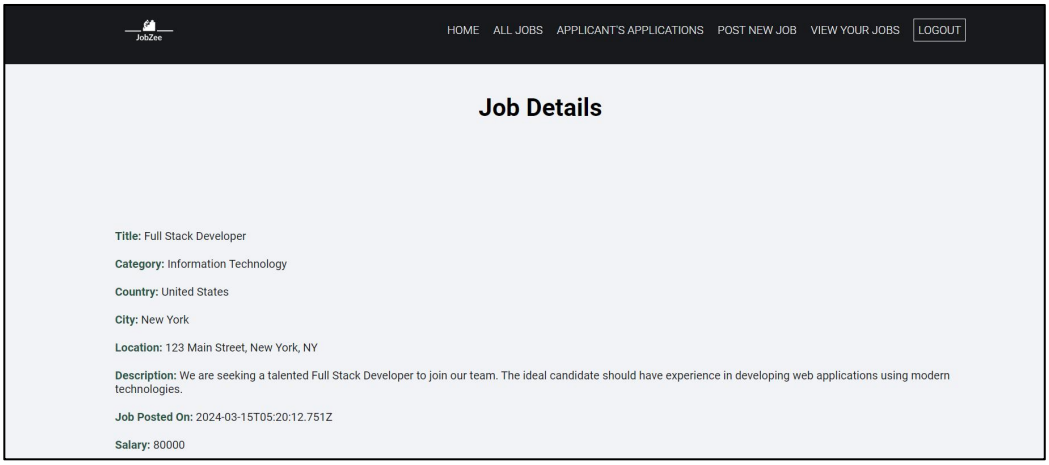
Employer Dashboard



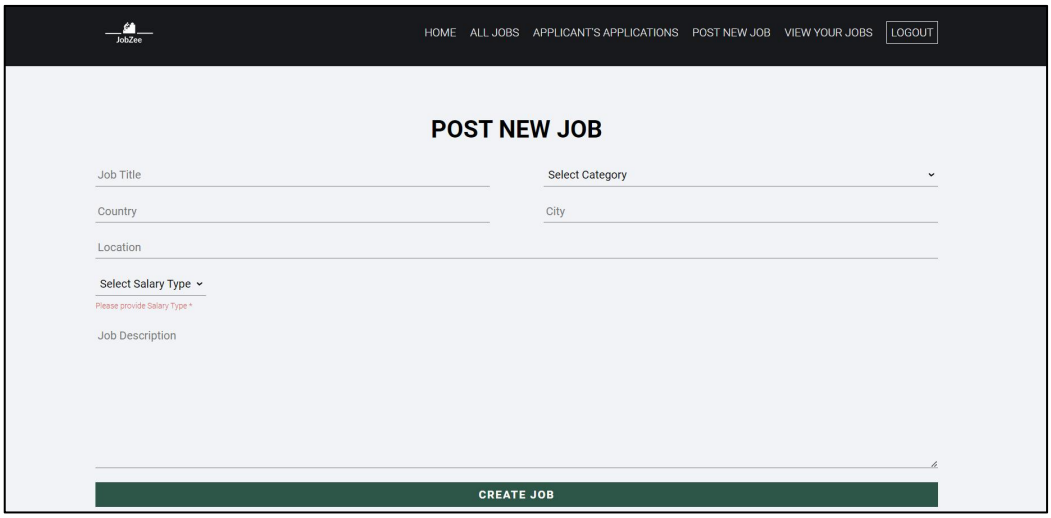
All Jobs Page



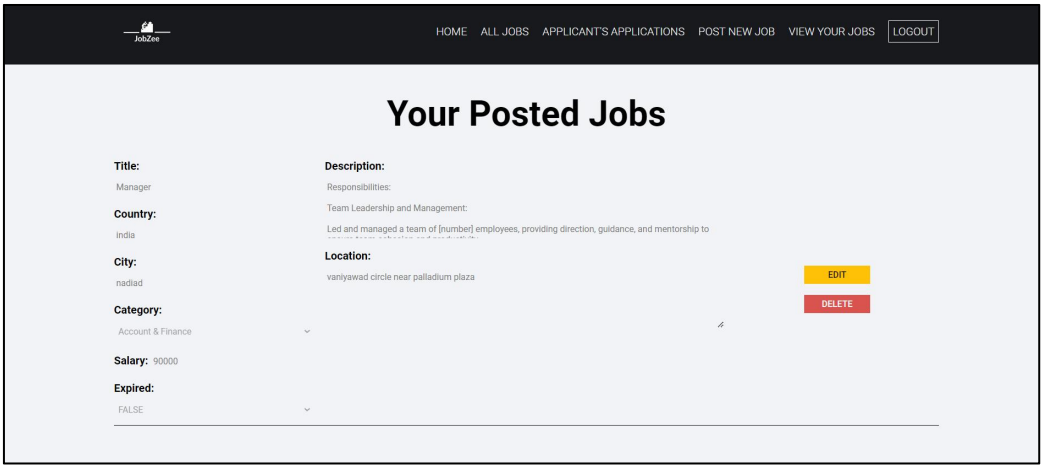
All Jobs Detail Page



Post New Job Page

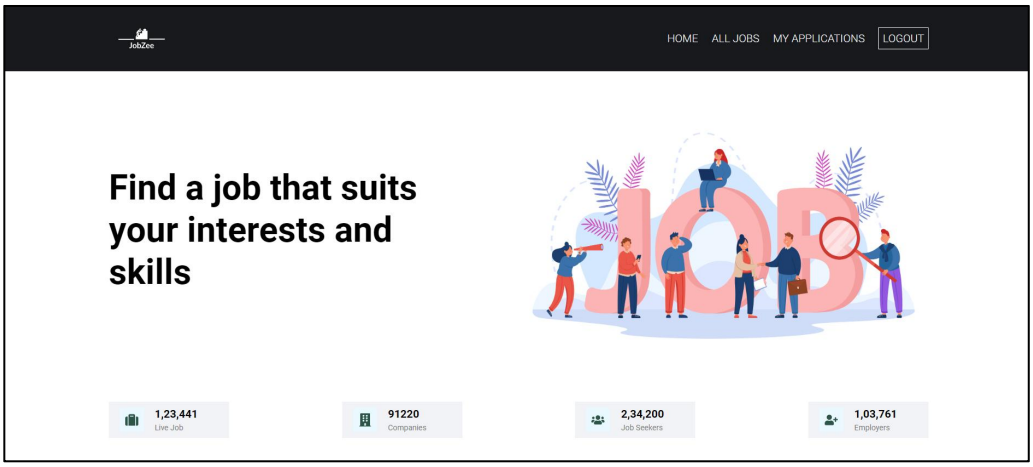


All Job Page

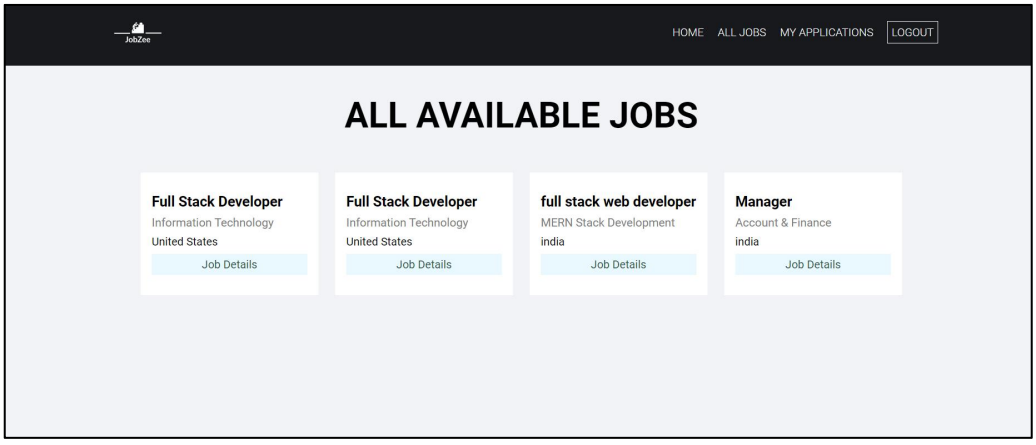

JobZee
HOME
ALL JOBS
APPLICANT'S APPLICATIONS
POST NEW JOB
VIEW YOUR JOBS
LOGOUT
Your Posted Jobs
Title:
Manager
Country:
india
City:
nadiad
Category:
Account & Finance
Salary: 90000
Expired:
FALSE
Description:
Responsibilities:
Team Leadership and Management:
Led and managed a team of [number] employees, providing direction, guidance, and mentorship to
Location:
vaniyawad circle near palladium plaza
EDIT
DELETE

Job-Seeker

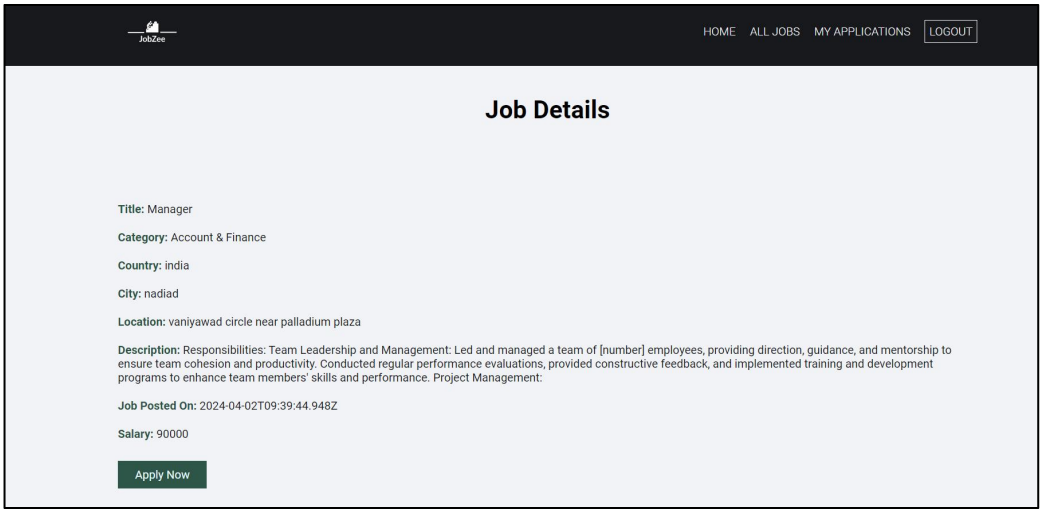
Home Page



Jobs Available Page



Job Detail Page



8. Conclusion

In conclusion, the MERN (MongoDB, Express.js, React.js, Node.js) job portal project represents a significant endeavor in modern web development, aimed at providing a comprehensive and user-centric platform for job seekers and employers alike. Throughout the development process, we have leveraged the strengths of each technology within the MERN stack to create a robust, scalable, and feature-rich application that addresses the needs and challenges of the job market.

By utilizing MongoDB as our database management system, we have benefited from its flexible document-oriented model, enabling us to store and manage job listings, user profiles, and other application data efficiently. Express.js has served as the foundation of our server-side logic, providing a lightweight and powerful framework for handling HTTP requests, authentication, and routing.

React.js has empowered us to build dynamic and responsive user interfaces, with its component-based architecture enabling the creation of reusable UI components and seamless integration with our backend services. Meanwhile, Node.js has facilitated real-time interactions, scalability, and asynchronous processing, ensuring high performance and responsiveness throughout the application.

9. Limitation and future scopes

Limitations:

1. Scalability: Depending on the infrastructure and resources available, scalability might be a limitation, especially as the user base grows. Managing a large number of concurrent users and handling increased data volume could pose challenges.

2. Performance: As the application grows in complexity and data volume, maintaining optimal performance could become an issue. Ensuring efficient query execution, minimizing latency, and optimizing resource utilization will be essential.

3. Security: Despite implementing authentication and authorization measures, security threats such as data breaches, SQL injection, cross-site scripting (XSS), and other vulnerabilities remain a concern. Continuous monitoring, vulnerability assessments, and regular security updates are necessary.

4. Compatibility: Ensuring compatibility across different devices, browsers, and screen sizes might be challenging, especially with the wide variety of configurations and user preferences.

5. User Engagement: Sustaining user engagement and retention over time could be a challenge, especially in a competitive market. Continuous improvements, personalized experiences, and effective marketing strategies will be needed to keep users actively using the platform.

Future Scope:

1. AI Integration: Implementing artificial intelligence (AI) and machine learning (ML) algorithms for job matching, candidate screening, and personalized recommendations could enhance the platform's effectiveness and user experience.

2. Advanced Analytics : Incorporating advanced analytics and reporting features could provide insights into job market trends, user behavior, and performance metrics, helping employers and job seekers make informed decisions.

3. Mobile Application: Developing native mobile applications for iOS and Android platforms or adopting a progressive web app (PWA) approach to enhance accessibility and reach a wider audience.

4. Social Integration: Integrating social media platforms for seamless login, sharing job opportunities, and networking could enhance user engagement and expand the platform's reach.

5. Global Expansion: Expanding the platform to cater to international job markets and multilingual support could unlock new opportunities and attract a diverse user base.

By addressing these limitations and exploring these future scope areas, the job portal project can evolve into a comprehensive and innovative platform that meets the evolving needs of job seekers and employers in the digital age.