CAREERCONNECT: A ROLE BASED JOB PORTAL FOR RECRUITERS TO FIND CANDIDATES BY SKILLS AND JOB DESCRIPTION

A MINI-PROJECT REPORT

Submitted by

PADMAPRIYA S -2116220701193

in partial fulfilment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

AUTONOMOUS, CHENNAI

NOV/DEC,2024

BONAFIDE CERTIFICATE

Certified that this mini project "**CAREERCONNECT : ROLE BASED JOB PORTAL FOR RECRUITERS TO FIND CANDIDATES BY SKILLS AND JOB DESCRIPTION**" is the bonafide work of "**PADMAPRIYA S(2116220701193)"** who carried out the project work under my supervision.

SIGNATURE

Mrs. JANANEE V,

Assistant Professor,

Computer Science & Engineering

Rajalakshmi Engineering College

Thandalam, Chennai -602105.

Submitted for the End semester practical examination to be held on

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I express my sincere thanks to my beloved and honourable chairman **MR.S.MEGANATHAN** and the chairperson **DR.M.THANGAM MEGANATHAN** for their timely support and encouragement.

I am greatly indebted to my respected and honourable principal

Dr. S.N.MURUGESAN for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by my head of the department **Dr. P. KUMAR**, and my Academic Head

Dr. R.SABITHA, for being ever supporting force during my project work.

I also extend my sincere and hearty thanks to my internal guide

Mrs. JANANEE V for her valuable guidance and motivation during the completion of this project.

My sincere thanks to my family members, friends and other staff members of Computer Science and Engineering.

PADMAPRIYA S 2116220701193

ABSTRACT

A Role-Based Job Portal for Recruiters to Find Candidates by Skills and Job Description is a web-based platform designed to streamline the job search and recruitment process. Built using the MERN stack (MongoDB, Express.js, React.js, and Node.js), his platform enables recruiters to post detailed job listings, specifying job descriptions, skill requirements, salary ranges, and other relevant information. Students, on the other hand, can browse these listings, apply filters such as salary, job type, and required skills, and submit applications by providing their name, contact details, and a resume link.

The system offers a role-based functionality where recruiters can efficiently manage job postings and search for qualified candidates, while students can explore job opportunities that align with their skills and career goals. By integrating advanced filtering and search capabilities, CareerConnect ensures that students can easily find the most relevant job openings, and recruiters can target candidates with the required qualifications.

Secure login and registration features ensure that both recruiters and students can access the platform safely. For recruiters, the system simplifies job posting and candidate discovery, while students benefit from a streamlined job search experience. Overall, CareerConnect provides an efficient, user-friendly solution for both recruiters and job seekers, making the hiring process faster and more effective by focusing on skill-based matching and clear job descriptions.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

A Role-Based Job Portal for Recruiters to Find Candidates by Skills and Job Description is an innovative platform developed to facilitate the interaction between recruiters and students in the job market. This portal allows recruiters to post job opportunities, specifying detailed job descriptions, skill requirements, salary ranges, and other essential criteria. On the other side, students can explore these job listings, search for roles that match their skills, and apply directly by submitting their details and resumes. The integration of advanced filtering and search options ensures a more targeted and efficient recruitment process, ultimately benefiting both recruiters and job seekers by making the job search experience more streamlined and effective.

1.2 SCOPE OF THE WORK

The scope of CareerConnect encompasses the development and implementation of a robust job portal that serves both recruiters and students. The platform provides user roles for recruiters and students, each with distinct functionalities such as job posting management for recruiters and resume submission for students.

CareerConnect ensures that the recruitment process is made more efficient through advanced search and filtering features, which allow students to narrow down opportunities based on specific parameters such as salary range, job role, and work schedule. The platform also focuses on creating a seamless user experience, from account creation to job application submission, with secure login and registration processes.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The primary aim of CareerConnect is to create a user-friendly and efficient job portal that bridges the gap between recruiters seeking qualified candidates and students searching for relevant job opportunities.

The specific objectives of the project are:

1. To develop a role-based job portal that provides tailored functionalities for both recruiters and students.
2. To enable recruiters to post jobs with comprehensive descriptions, skill requirements, and salary details.
3. To provide students with advanced search and filtering options to find jobs that match their qualifications and preferences.
4. To create a secure and scalable platform that supports a seamless user experience, from account creation to job application.
5. To enhance the efficiency of the recruitment process by allowing recruiters to easily manage and target candidates based on required skills.
6. To ensure that students can apply to relevant job postings by submitting resumes and personal details through the platform.

CHAPTER 2

SYSTEM SPECIFICATIONS

Hardware Specifications

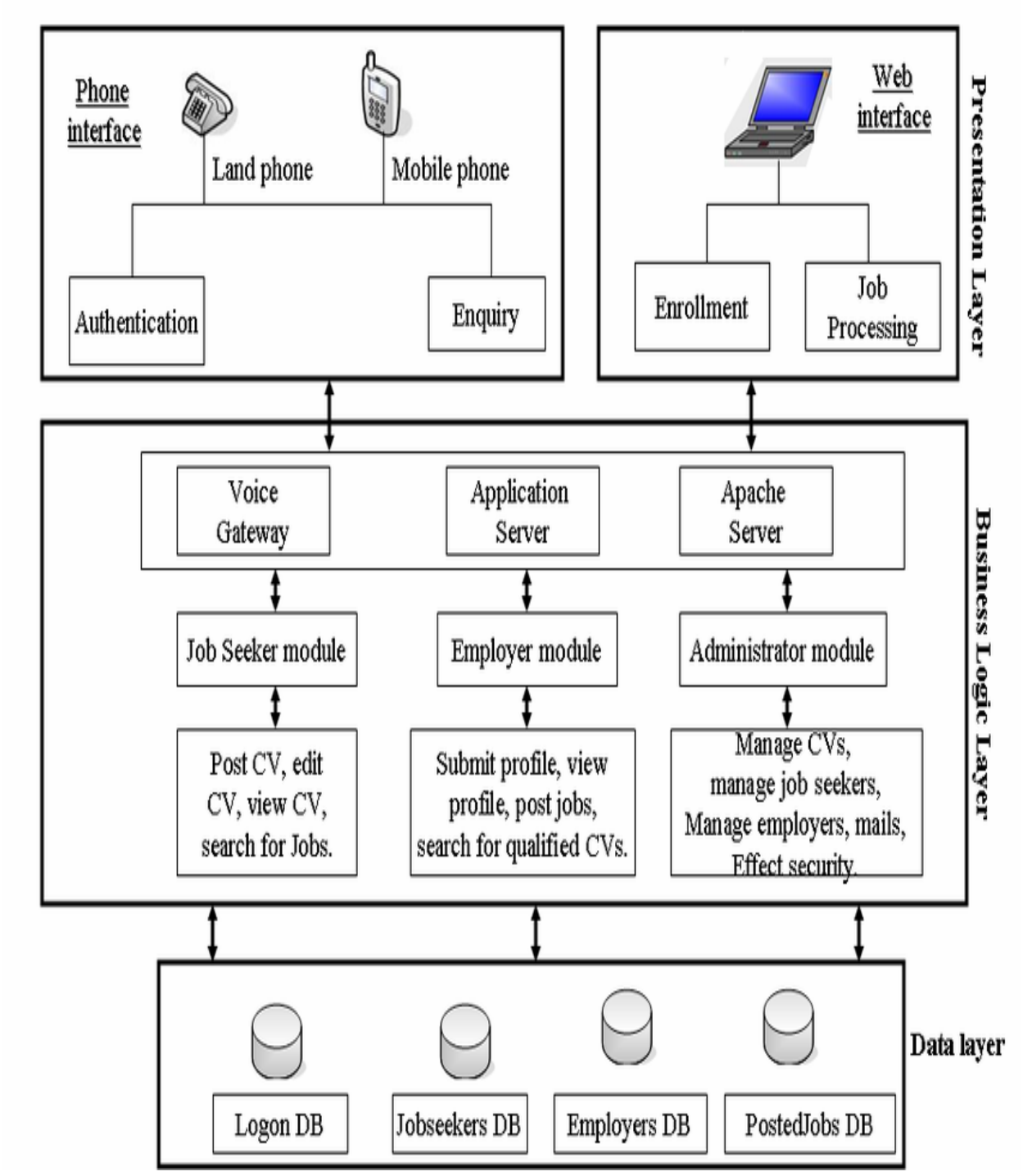
- **Processor:** Intel Core i5 or equivalent
- **RAM:** 8 GB or higher
- **Operating System:** Windows, macOS, or Linux
- **Storage :** Minimum 10 GB of free disk space for project files, databases, and software installations

Software Specifications

- **Frontend Framework:** React.js (for building interactive user interfaces)
- **Backend Framework:** Node.js with Express.js (for managing server-side operations)
- **Database:** MongoDB (for efficient data storage and retrieval)
- **Development Environment:** Visual Studio Code
- **Package Manager:** NPM (Node Package Manager) to manage dependencies
- **Browser:** Any modern browser (Google Chrome, Firefox, etc.)
- **Operating System:** Windows, macOS, or Linux

CHAPTER 3

3.ARCHITECTURE DIAGRAM



CHAPTER 4

MODULE DESCRIPTION

4.1. User Authentication Module

This module handles the login and registration processes for both recruiters and students. It verifies the credentials provided by users during login and allows them to sign up with role-based access (Recruiter or Student).

4.2. Recruiter Module

This module allows recruiters to post jobs, manage job listings, and view candidate applications. Recruiters can specify the job description, required skills, salary, and other details when creating a job posting.

4.3. Student Module

This module enables students to search for job opportunities, filter jobs based on their preferences, and apply by submitting their resume and other required information.

4.4. Job Search and Filter Module

This module is dedicated to enabling students to search and filter job postings based on specific criteria such as salary, job type, and required skills. It enhances user experience by providing targeted job recommendations.

4.5. Job Posting Module

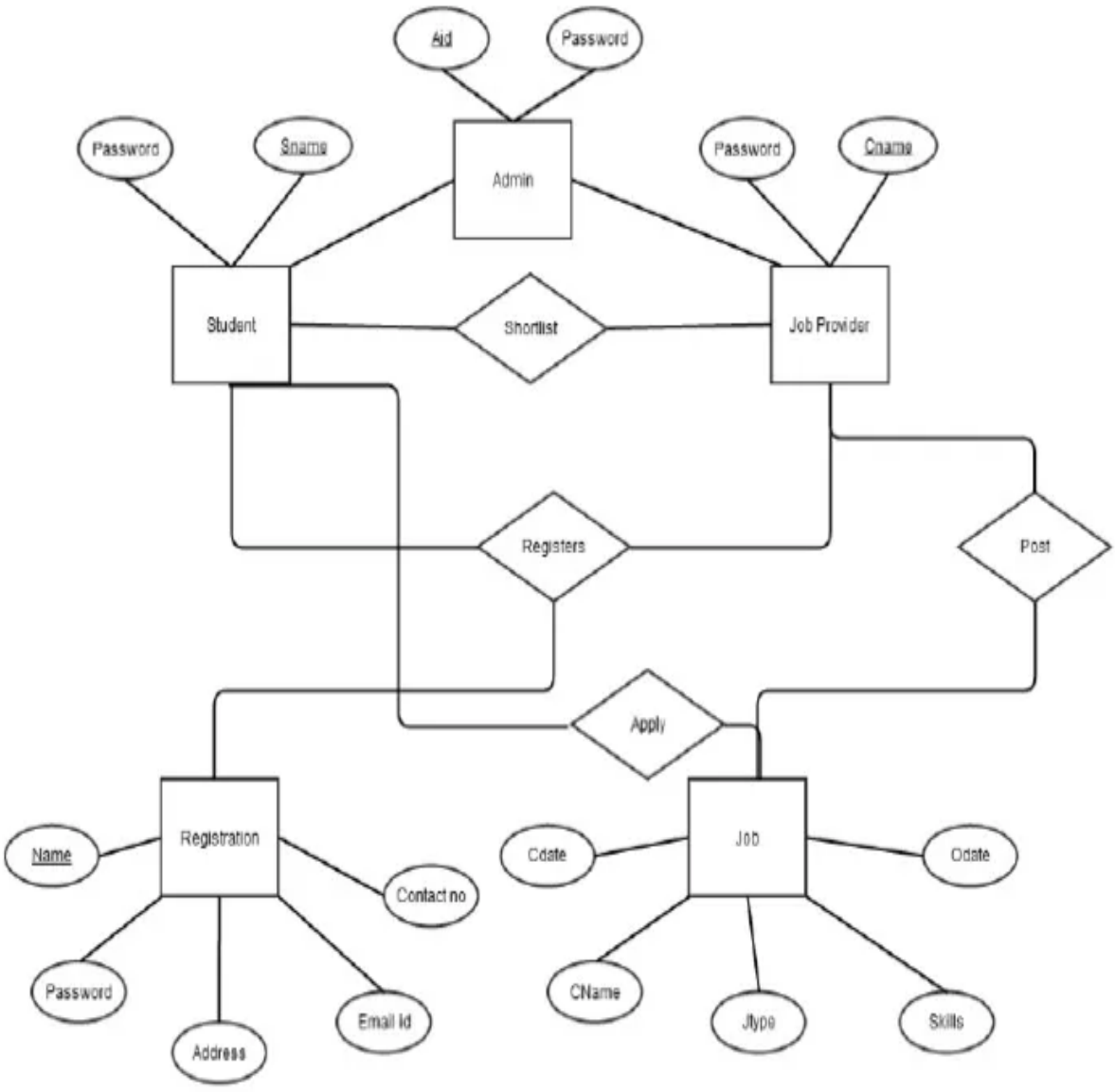
This module is part of the recruiter's interface, where they can post new job openings. It allows recruiters to provide job details and manage their listings.

CHAPTER 5

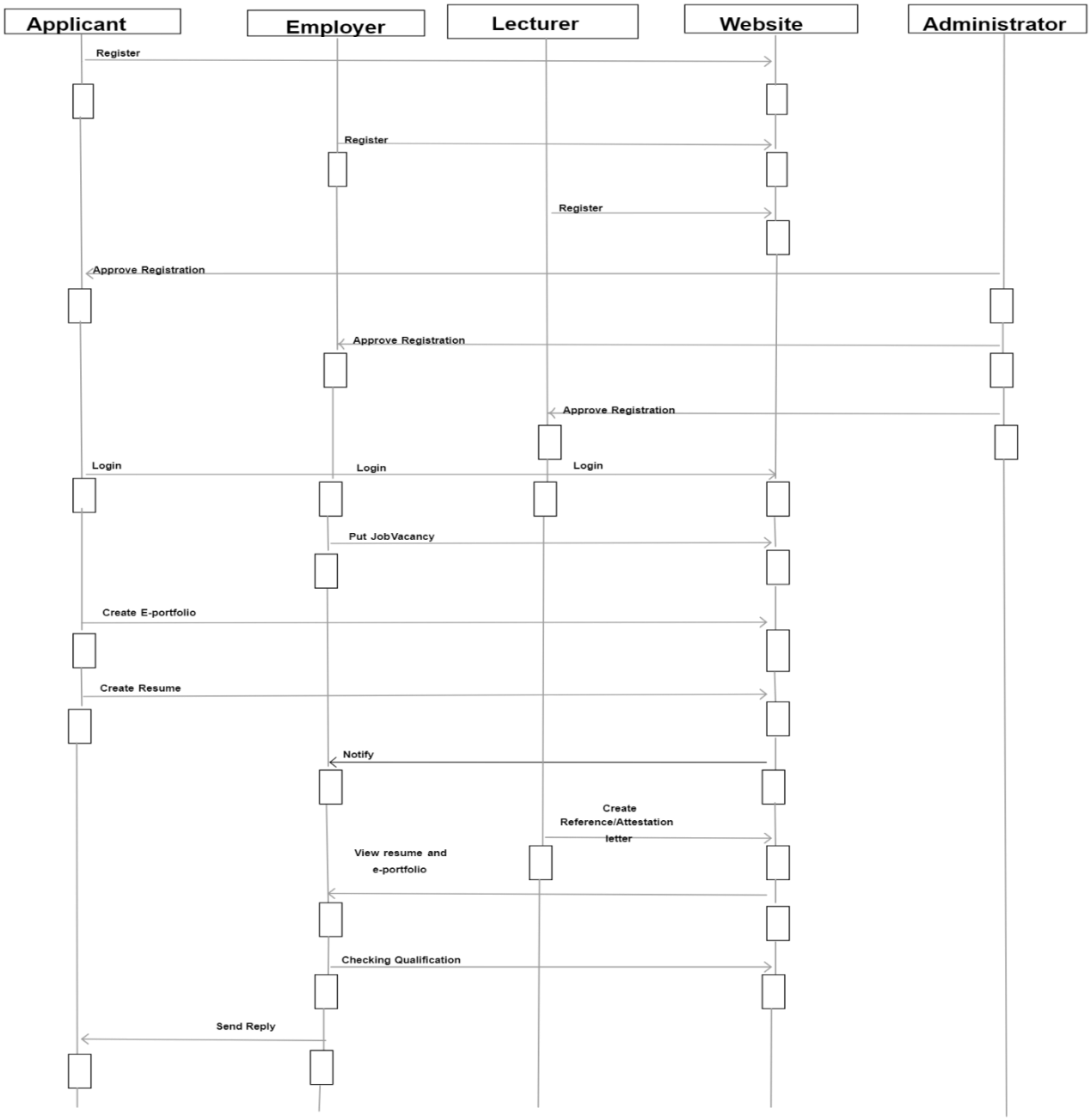
SYSTEM DESIGN

5.1 USE CASE DIAGRAM

5.2 ER DIAGRAM



5.3. SEQUENCE DIAGRAM



CHAPTER 6

SAMPLE CODING

User Registration

```
import React, { useState } from 'react';
import axios from 'axios';
const Register = () => {
  const [fullname, setFullname] = useState('');
  const [email, setEmail] = useState('');
  const [phoneNumber, setPhoneNumber] = useState('');
  const [password, setPassword] = useState('');
  const [role, setRole] = useState('student');
  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const response = await axios.post('/api/register', {
        fullname,
        email,
        phoneNumber,
        password,
        role,
      });
      console.log(response.data);
```

```
    } catch (error) {
      console.error('Error registering user:', error);
    }
  };
  return (
    <form onSubmit={handleSubmit}>
      <input type="text" value={fullname} onChange={(e) => setFullname(e.target.value)} placeholder="Full Name" required />
      <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} placeholder="Email" required />
      <input type="text" value={phoneNumber} onChange={(e) => setPhoneNumber(e.target.value)} placeholder="Phone Number" required />
      <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} placeholder="Password" required />
      <select value={role} onChange={(e) => setRole(e.target.value)}>
        <option value="student">Student</option>
        <option value="recruiter">Recruiter</option>
      </select>
      <button type="submit">Register</button>
    </form>
  );
};
export default Register;
```

Job Posting

```
const express = require('express');

const router = express.Router();

const Job = require('../models/Job');

router.post('/api/jobs', async (req, res) => {

 const { jobTitle, jobDescription, salary, skillsRequired, postedBy } = req.body;

 const newJob = new Job({

  jobTitle,

  jobDescription,

  salary,

  skillsRequired,

  postedBy,

 });

 try {

  const savedJob = await newJob.save();

  res.status(201).json(savedJob);

 } catch (error) {

  console.error('Error posting job:', error);

  res.status(500).json({ message: 'Server error' });

 }
```

```
});
module.exports = router;
```

Job Search Functionality

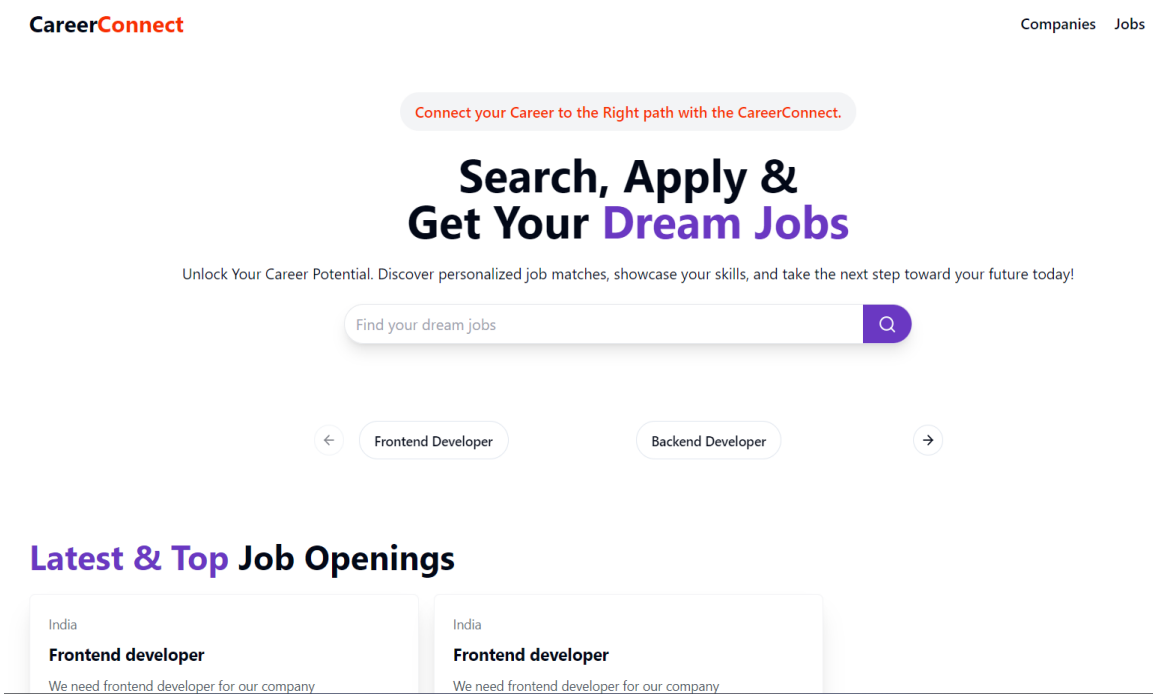
```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
const JobSearch = () => {
  const [searchQuery, setSearchQuery] = useState('');
  const [jobs, setJobs] = useState([]);
  const handleSearch = async () => {
    try {
      const response = await axios.get(`/api/jobs?query=${searchQuery}`);
      setJobs(response.data);
    } catch (error) {
      console.error('Error fetching jobs:', error);
    }
  };
  useEffect(() => {
    handleSearch(); //fetch initial job listing
  }, []);
  return (
    <div>
```

```
      <input type="text" value={searchQuery} onChange={(e) => setSearchQuery(e.target.value)} placeholder="Search jobs..." />
      <button onClick={handleSearch}>Search</button>
      <ul>
        {jobs.map((job) => (
          <li key={job._id}>{job.jobTitle} - {job.salary}</li>
        ))}
      </ul>
    </div>
  );
};
export default JobSearch;
```

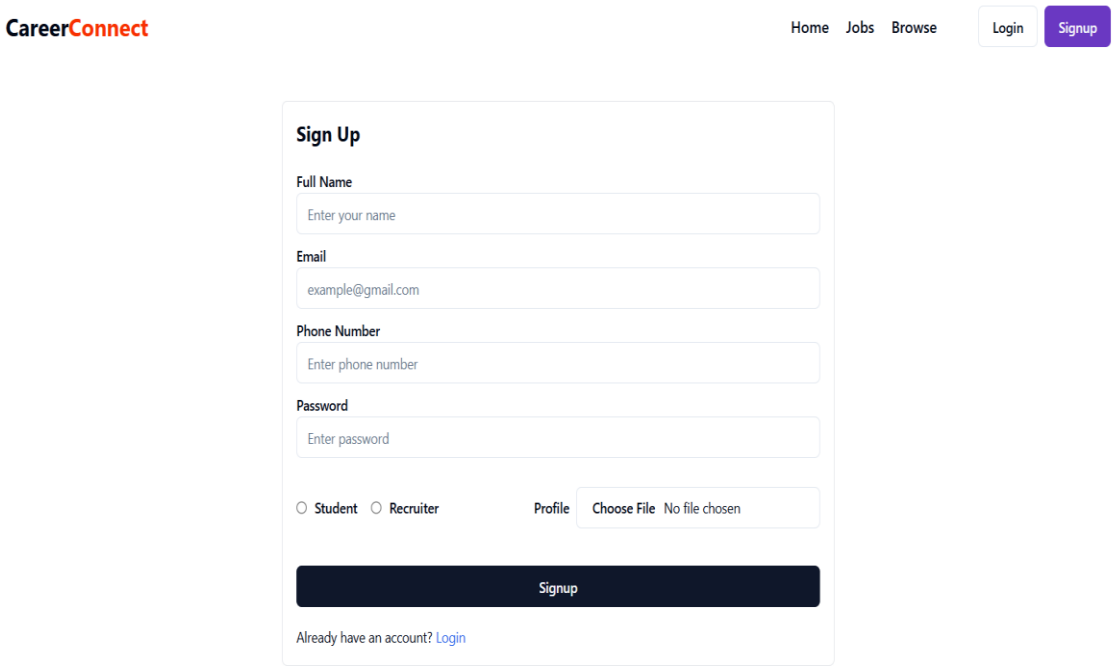
CHAPTER 7

SCREEN SHOTS

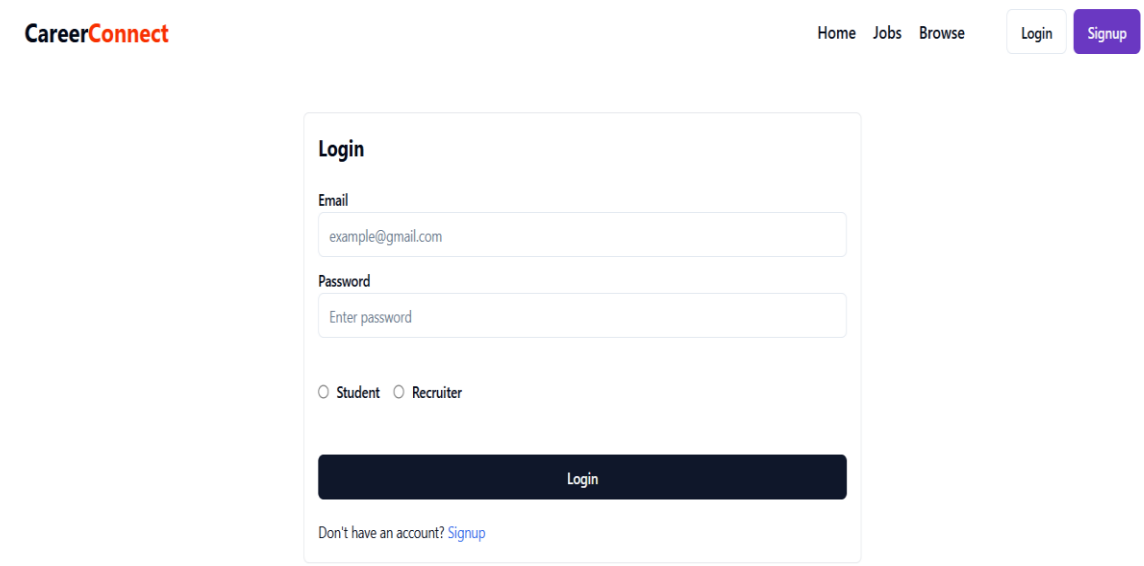
Home page



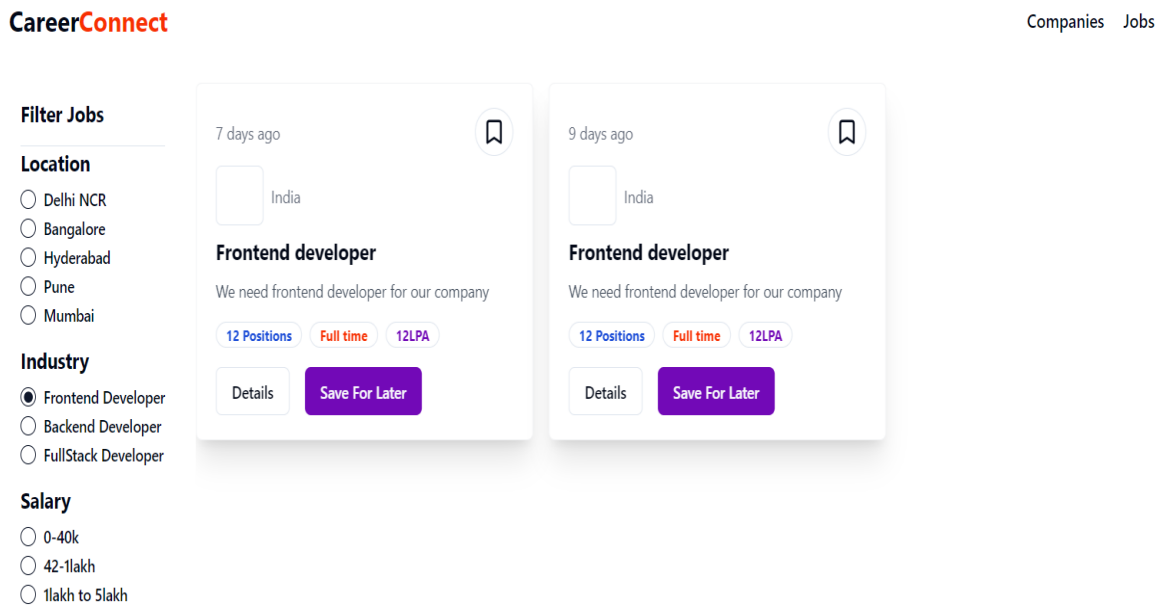
Signup Page

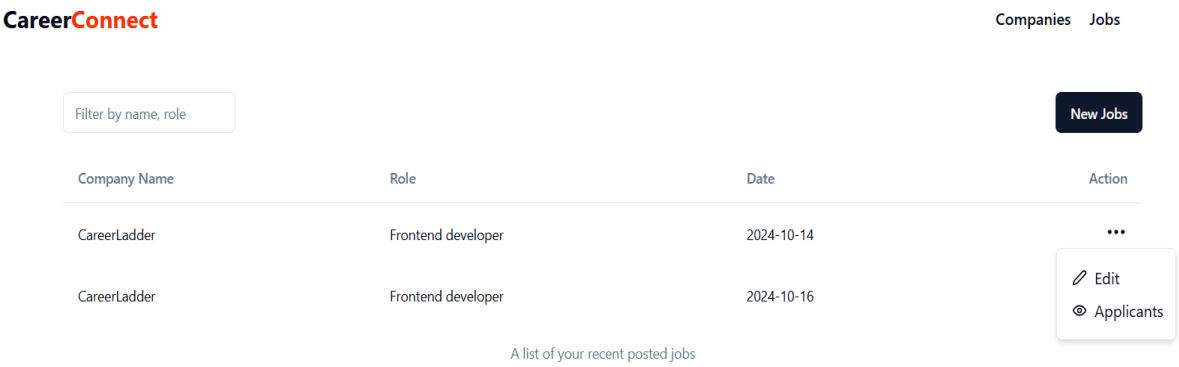


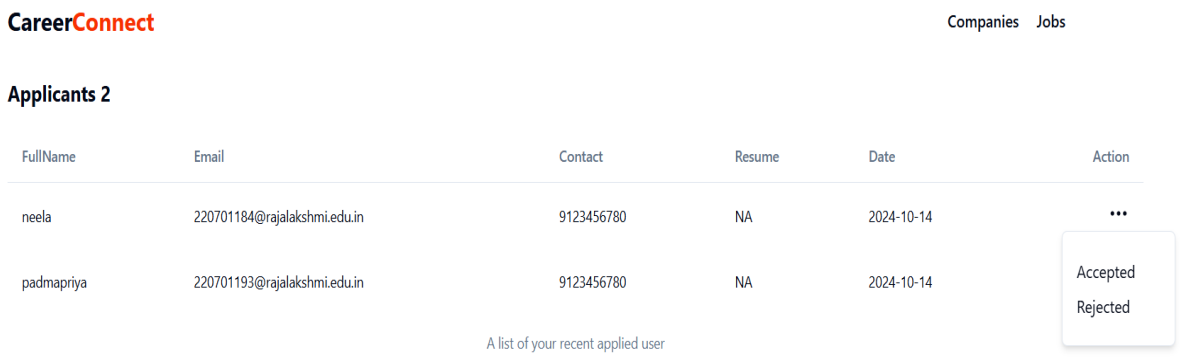
Login Page

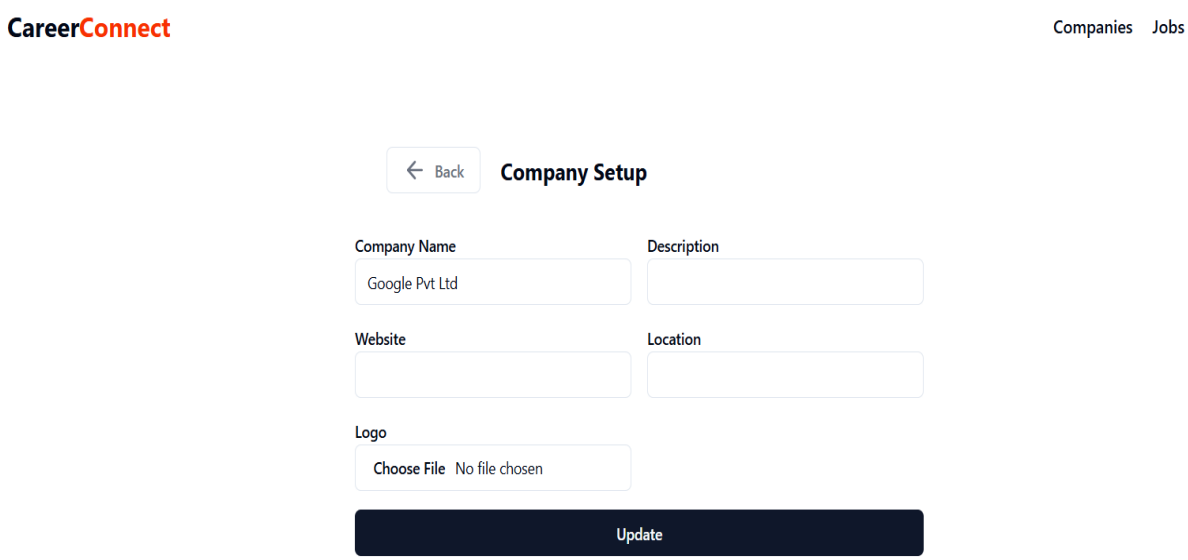


Job Search Page




CareerConnect
Companies
Jobs
Filter by name, role
New Jobs
Company Name
Role
Date
Action
CareerLadder
Frontend developer
2024-10-14
CareerLadder
Frontend developer
2024-10-16
Edit
Applicants
A list of your recent posted jobs


CareerConnect
Companies
Jobs
Applicants 2
FullName
Email
Contact
Resume
Date
Action
neela
220701184@rajalakshmi.edu.in
9123456780
NA
2024-10-14
padmapriya
220701193@rajalakshmi.edu.in
9123456780
NA
2024-10-14
Accepted
Rejected
A list of your recent applied user


CareerConnect
Companies
Jobs
Back
Company Setup
Company Name
Google Pvt Ltd
Description
Website
Location
Logo
Choose File
No file chosen
Update

CareerConnect

Companies Jobs

Your Company Name

What would you like to give your company name? you can change this later.

Company Name

google

Cancel Continue

CareerConnect

Companies Jobs

Title

Description

Requirements

Salary

Location

Job Type

Experience Level

No of Postion

0

Post New Job

*Please register a company first, before posting a jobs

CHAPTER 8

CONCLUSION

In conclusion, CareerConnect serves as a comprehensive role-based job portal designed to bridge the gap between recruiters and candidates. By providing a platform where recruiters can post job openings with detailed descriptions and skill requirements, while students can apply for positions tailored to their qualifications, CareerConnect effectively addresses the needs of both parties in the job market.

The project was developed using a robust tech stack, including React.js for the frontend, Node.js and Express for the backend, and MongoDB for data storage. This combination allowed for the creation of a responsive and user-friendly interface, as well as a reliable backend that efficiently handles user requests and job data management.

Throughout the development process, emphasis was placed on usability and accessibility, ensuring that users can navigate the platform with ease. The integration of essential functionalities such as job filtering by salary, role, and time enhances the user experience and increases the likelihood of successful job placements.

Looking ahead, future enhancements could include the integration of machine learning algorithms to improve job matching based on candidate profiles, as well as additional features such as notifications for job postings and application statuses. These improvements would further enrich the platform and provide users with a more personalized experience.

REFERENCES

1.REACT: https://legacy.reactjs.org/

2.MONGODB: https://www.mongodb.com/resources/languages/javascript

3.shadcn-ui: https://ui.shadcn.com/

4.HTML: https://www.w3schools.com/html/

5.CSS: https://www.w3schools.com/css/

6.JAVASCRIPT: https://www.w3schools.com/js/