

# **RESUME UPLOADING PLATFORM**

## **A MINI PROJECT REPORT**

Submitted by

**MANOJ R L - 220701161**

**In partial fulfilment for the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE**



**RAJALAKSHMI**  
**ENGINEERING COLLEGE**  
An AUTONOMOUS Institution  
Affiliated to ANNA UNIVERSITY, Chennai

**RAJALAKSHMI ENGINEERING COLLEGE**

**AUTONOMOUS, CHENNAI**

**NOV/DEC, 2024**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**Resume uploading platform**” is the bonafide work of “**MANOJ R L(220701161)**” who carried out the project work under my supervision.

### **SIGNATURE**

**Mrs. V. JANANEE**

**Assistant Professor,**

**Computer Science and Engineering,**

**Rajalakshmi Engineering college,**

**Thandalam, Chennai - 602 105**

Submitted for the Practical Examination held on \_\_\_\_\_

**EXTERNAL EXAMINER**

**INTERNAL EXAMINER**

## ACKNOWLEDGEMENT

I express my sincere thanks to my beloved and honourable chairmanm **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.

CHAPTER NO.	TABLE OF CONTENTS	TITLE PAGE
	ABSTRACT	5
1	INTRODUCTION	6
1.2	SCOPE OF THE WORK	6
1.3	AIM AND OBJECTIVES OF THE PROJECT	6
2	SYSTEM SPECIFICATIONS	7
2.1	HARDWARE SPECIFICATIONS	7
2.2	SOFTWARE SPECIFICATIONS	8
3	ARCHITECTURE DIAGRAM	9
4	MODULE DESCRIPTION	10
5	SYSTEM DESIGN	12
5.1	USE CASE DIAGRAM	12
5.2	ER DIAGRAM	13
5.3	DATA FLOW DIAGRAM	14
5.4	ACTIVITY DIAGRAM	15
6	SAMPLE CODING	16
7	SCREEN SHOTS	22
8	CONCLUSION	24
9	REFERENCES	25

## ABSTRACT

The **Resume Uploading Platform** is a comprehensive web-based solution aimed at simplifying the recruitment process for both job seekers and employers. It serves as a modern and intuitive tool designed to improve the way job seekers apply for positions by offering a streamlined experience for uploading, managing, and submitting resumes. On the other hand, the platform provides employers with an efficient system to filter, access, and evaluate a large pool of potential candidates, reducing time-to-hire and enhancing the overall recruitment process.

This project stands to significantly improve the hiring landscape by digitizing and centralizing the most critical parts of the job search process. By offering innovative tools like real-time application tracking, smart job matching, and streamlined resume management, the platform empowers users to take control of their career journey and shortens the time to hire for companies.

The **Resume Uploading Platform** is designed to scale with future needs, with the potential for integration with third-party HR systems, advanced analytics for job trends, and expanded employer features like candidate assessments and interview scheduling. In a rapidly evolving job market, this platform positions itself as a critical solution for efficient, digital-first hiring practices that keep both job seekers and employers ahead of the curve.

# CHAPTER 1

## INTRODUCTION

### 1.1 INTRODUCTION

The **Resume Uploading and Job Application Platform** is an advanced and user-friendly system designed to streamline the job search and recruitment process for both job seekers and employers. In today's competitive job market, effective job applications and efficient candidate selection are crucial for career success and organizational growth. The platform addresses these needs by offering seamless resume management, quick job search capabilities, and intuitive tools for applying to positions.

### 1.2 SCOPE OF THE WORK

The **Resume Uploading and Job Application Platform** encompasses a wide range of features and functionalities aimed at simplifying and optimizing the job search and recruitment process. The platform provides job seekers with the ability to easily upload, manage, and customize resumes across various job applications, while also offering employers tools to efficiently review and shortlist candidates based on their qualifications. The platform integrates advanced technologies such as resume parsing, machine learning-based job matching, and data-driven analytics to offer personalized job recommendations to candidates and streamline the hiring process for employers. Additional features include secure data handling, mobile compatibility, notifications for job application updates, and integration with professional networks. By offering a holistic and automated system, the platform significantly enhances the experience for both job seekers and recruiters, making the entire recruitment journey faster, more effective, and less prone to errors.

### 1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of the **Resume Uploading and Job Application Platform** is to create an efficient, user-centric system that simplifies resume management and job applications while enabling employers to find qualified candidates quickly. The platform strives to bridge the gap between job seekers and employers by leveraging advanced technologies to make the recruitment process more accessible, organized, and effective for all parties involved.

## **CHAPTER 2**

### **SYSTEM SPECIFICATIONS**

The resume uploading platform is a comprehensive web application designed to connect job seekers with recruiters. It allows users to upload, manage, and share their resumes while helping recruiters find suitable candidates efficiently. The platform leverages modern technologies to provide a secure, scalable, and user-friendly experience for both job seekers and recruiters.

## **2.1 SOFTWARE SPECIFICATIONS**

### **1. System Architecture**

Architecture Type: Three-tier architecture (frontend, backend, database).

Frontend: Built using React.js (or an equivalent SPA framework like Vue.js or Angular) for responsive and interactive user interfaces.

Backend: RESTful API or GraphQL, developed using Node.js with Express.js (or Django in Python) for managing application logic.

Database: PostgreSQL for relational data storage, with optional MongoDB for document-based storage, and AWS S3 for file storage.

### **2. Frontend Specification**

Framework: React.js

APIs: Axios for API requests to interact with the backend.

### **3. Backend Specification**

Language: JavaScript (Node.js) or MongoDB

Framework: Angular.js (for Node)

File Handling: Multer for file uploads (Node.js)

Email Service: Nodemailer or a third-party service like SendGrid

### **4. Database Schema**

Relational Database: MongoDB tables for structured data

Users Table: Fields for UserID, Email, Password (hashed), Role (Job Seeker/Recruiter), Profile Data.

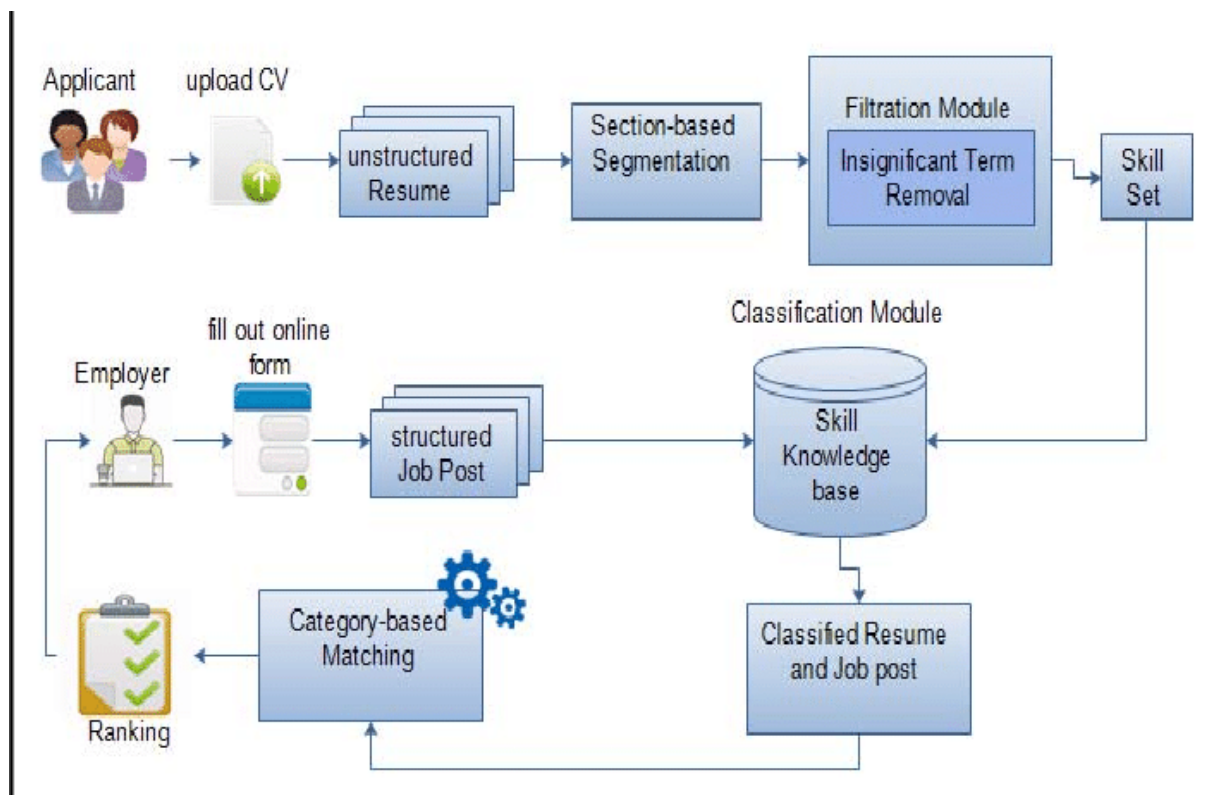
Resumes Table: Fields for ResumeID, UserID, FilePath, ParsedData, DateUploaded.

NoSQL Database (optional): MongoDB collections for unstructured or JSON-like data.



## CHAPTER 3

### ARCHITECTURE DIAGRAM



## CHAPTER 4

### MODULE DESCRIPTION

#### 4.1. User Registration and Login Module

This module is responsible for managing user access to the platform, ensuring that both job seekers and employers have secure, personalized accounts. The primary functions include:

- **User Registration:** New users (both job seekers and employers) can register on the platform by providing necessary information such as email, password, and basic details. Job seekers may be required to provide additional information related to their professional background.
- **User Authentication:** Secure login mechanism using email/username and password. It also supports password recovery and reset functionalities.
- **OAuth Integration:** Users can register and log in via third-party services such as Google, LinkedIn, or Facebook for ease of access.
- **User Profiles:** After registration, users can create and edit their profiles. Job seekers can upload resumes, provide details about their education, work experience, and skills, while employers can build company profiles.

#### 4.2. User Resume Uploading Module

This module enables job seekers to upload and manage their resumes efficiently. The key functionalities include:

- **Resume Uploading:** Job seekers can upload multiple resumes in different formats (PDF, DOCX, etc.) and label them based on job preferences.
- **Resume Parsing:** The system automatically extracts relevant data (name, contact information, work experience, etc.) from uploaded resumes to streamline profile creation.
- **Resume Customization:** Users can edit and customize their resumes directly within the platform, allowing for adjustments tailored to specific job applications.
- **Resume Versioning:** Users can maintain multiple versions of their resumes, selecting different resumes for various job applications.

### 4.3 Testing Status Module

This module is designed to provide users with detailed information about the status of their job applications and interview scheduling. Its features include:

- **Application Status Tracking:** Users can track the progress of their job applications, whether it's under review, shortlisted, or rejected.
- **Interview Scheduling:** Notifications and status updates are provided if the application proceeds to the interview stage, including details on interview date, time, and format (in-person, virtual).
- **Feedback Integration:** Employers can provide feedback on the job seeker's application or interview, which is displayed to the user in this module.

### 4.4 Admin Module

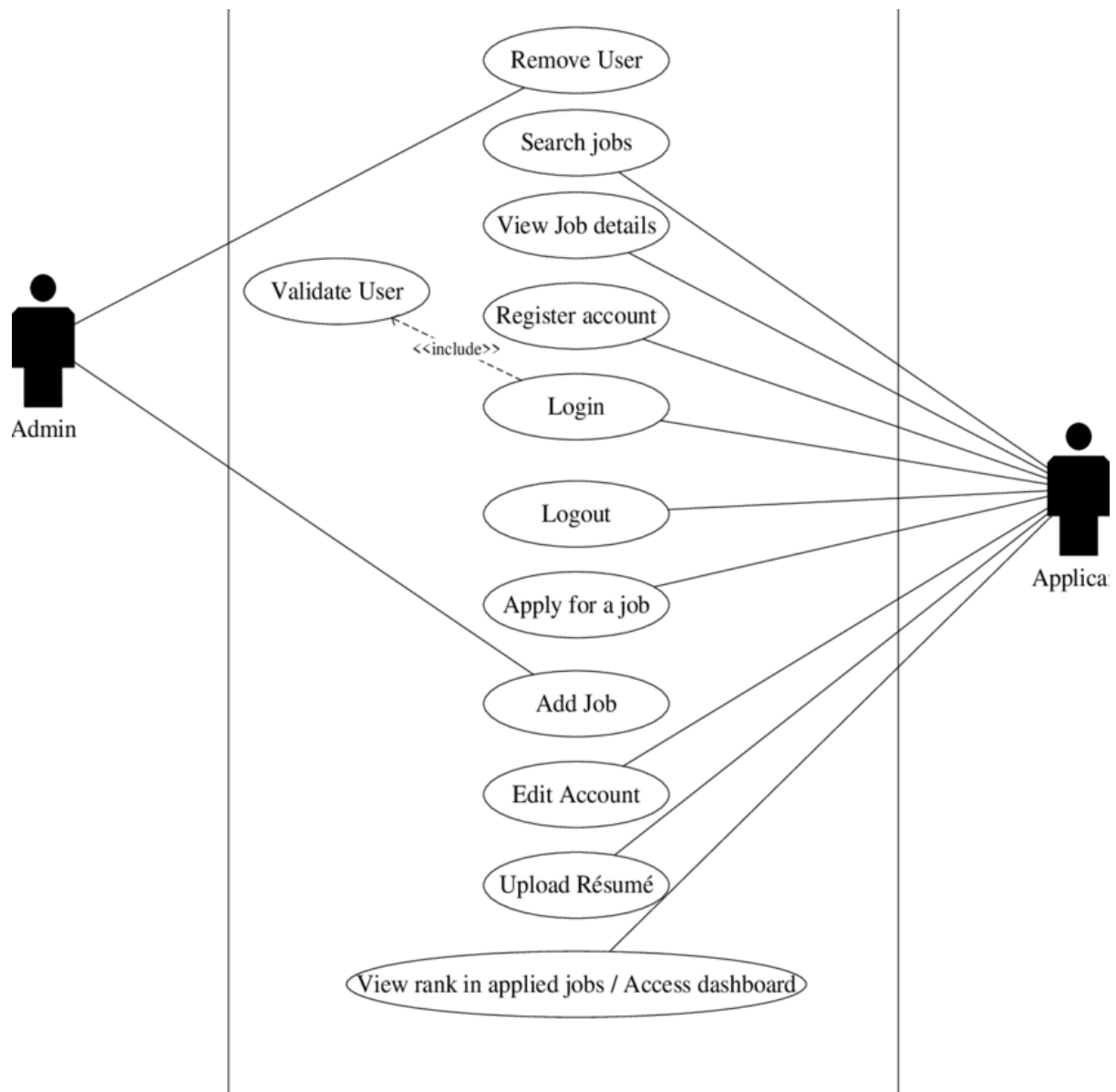
The admin module is designed for system administrators to oversee and manage the entire platform's functionality. The key components of this module include:

- **User Management:** Admins can view and manage all registered users, both job seekers and employers. This includes account activation, suspension, or deletion.
- **Job Posting Management:** Admins can monitor and manage job postings created by employers, ensuring they meet platform standards and compliance guidelines.
- **Analytics and Reporting:** Admins can generate reports on platform activity, including the number of active users, job postings, and application submissions.
- **Content Moderation:** Admins can review and moderate user-generated content such as resumes, job descriptions, and feedback to ensure compliance with terms of service and content policies.
- **System Monitoring and Maintenance:** Admins ensure the smooth running of the platform by monitoring system performance, resolving technical issues, and implementing updates as needed.

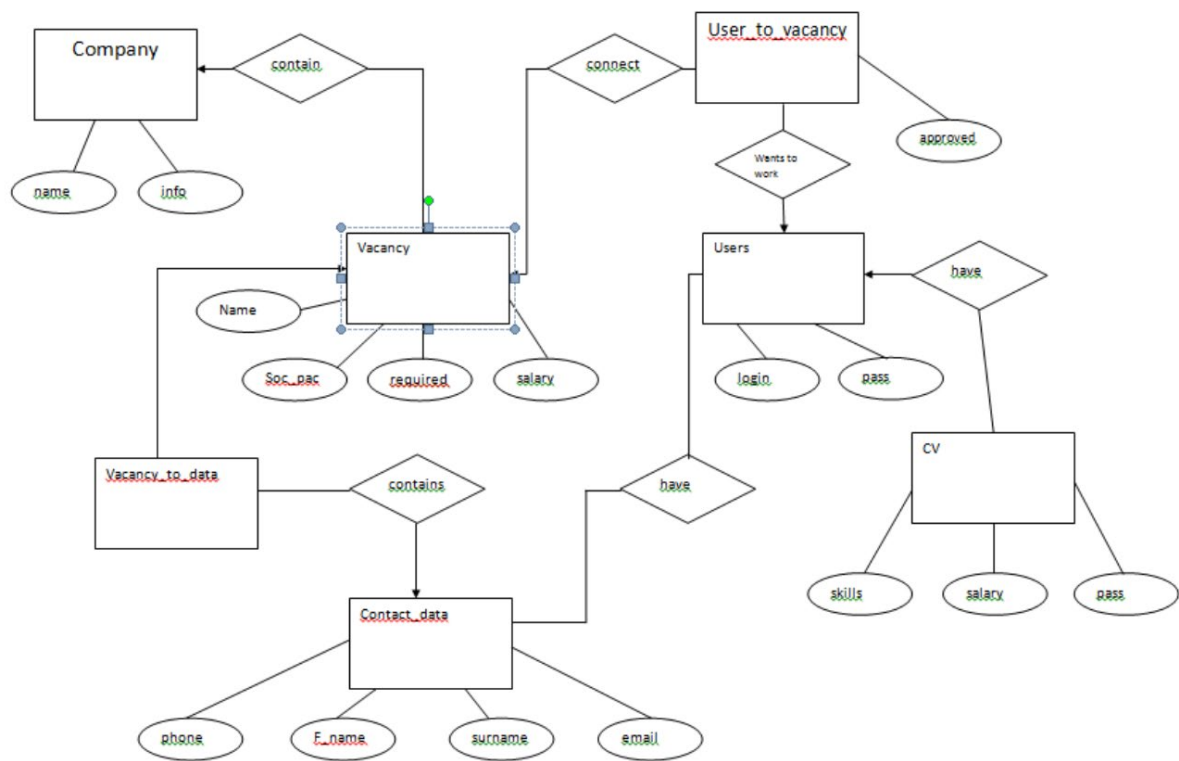
## CHAPTER 5

### SYSTEM DESIGN

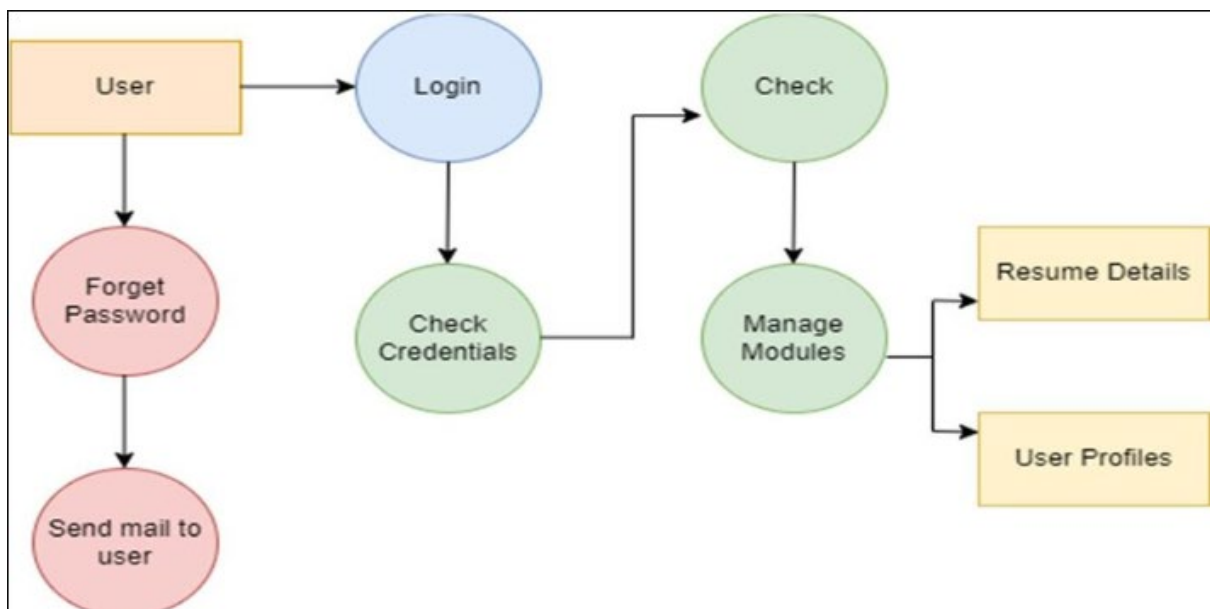
#### 5.1 USE CASE DIAGRAM



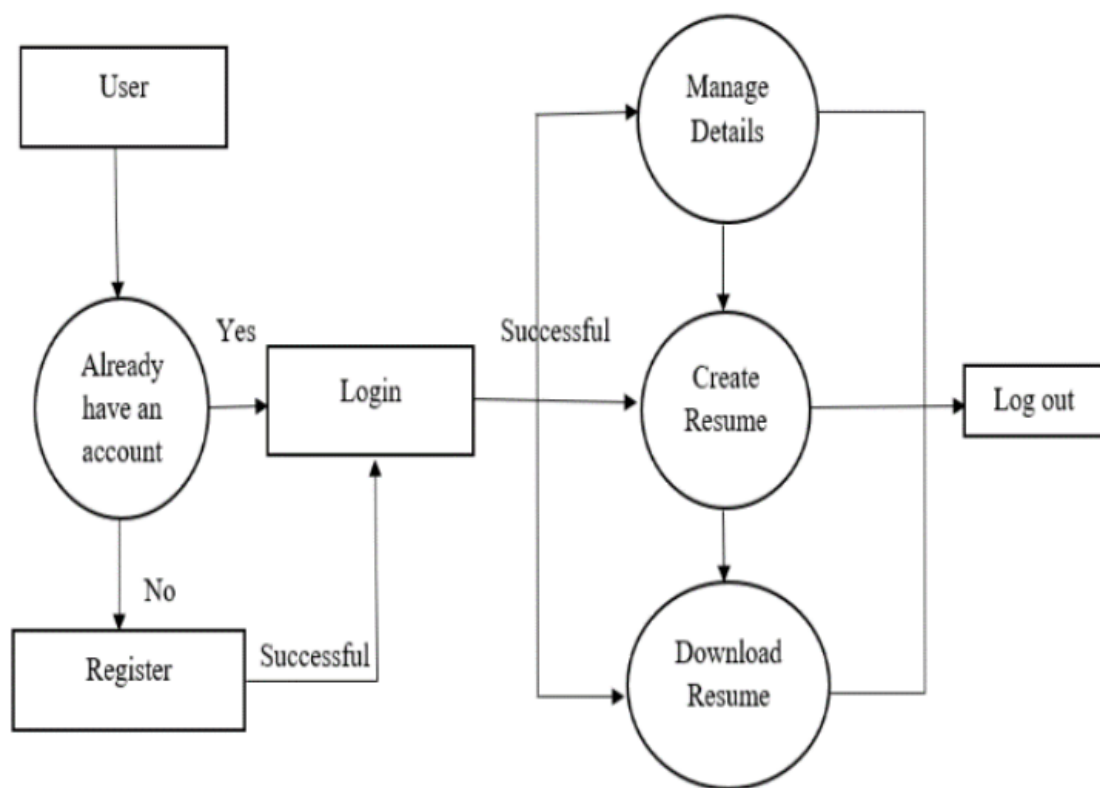
## 5.2 ER DIAGRAM



### 5.3 DATA FLOW DIAGRAM



## 5.4 ACTIVITY DIAGRAM



## SAMPLE CODE

```
import React, { useState } from 'react';

import './form.css'; // Import the CSS file for styling

const FormPage = () => {

  const [formData, setFormData] = useState({name: email: "",dob: "",state: "",
                                             gender: "",jobLocation: [],      avatar: null,
                                             });

  // Handle input changes

  const handleChange = (e) => {

    const { name, value } = e.target;

    setFormData((prevState) => ({...prevState, [name]: value,}));

    // Handle file change

    const handleFileChange = (e) => {

      setFormData((prevState) => ({
        ...prevState,
        avatar: e.target.files[0],
      }));

    };

    // Handle job location checkbox changes

    const handleCheckboxChange = (e) => {

      const { value, checked } = e.target;

      setFormData((prevState) => {if (checked) {

        return {

          ...prevState,
```



```

    jobLocation: [...prevState.jobLocation, value],
    };
  } else {
    return {
      ...prevState,
      jobLocation:
        prevState.jobLocation.filter((location) => location !==
        value),
    };
  }
});
};

```

*// Submit form data to backend*

```

const handleSubmit = async (e) => {
  e.preventDefault();
  const data = new FormData();
  data.append('name', formData.name);
  data.append('email', formData.email);
  data.append('dob', formData.dob);
  data.append('state', formData.state);
  data.append('gender', formData.gender);
  formData.jobLocation.forEach((location) => data.append('jobLocation',
  location));
  data.append('avatar', formData.avatar);

```

```
try {constresponse = await fetch('http://localhost:5001/api/candidates/add',  
{method: 'POST', body: data, });
```

```
if (response.ok) {alert('Candidate added successfully');}
```

```
else {
```

```
    alert('Failed to upload resume');
```

```
    }
```

```
}
```

```
catch (error) {console.error('Error uploading data:', error);
```

```
    alert('An error occurred while uploading');
```

```
}
```

```
};
```

```
return (
```

```
<div className="form-page">
```

```
<nav className="navbar">
```

```
  <div className="container">
```

```
    <h1 className="logo">ResumeRise - Resume Uploader Platform</h1>
```

```
    <ul className="nav-links">
```

```
      <li><a href="#about">About Us</a></li>
```

```
      <li><a href="#form">Upload Resume</a></li>
```

```
      <li><a href="#" className="sign-in-button">Sign In</a></li>
```

```
    </ul>
```

```
  </div>
```

```
</nav>
```

```
<header className="hero">
```

```
  <div className="hero-content">
```

```
<center><h2>Update Your Resume with Ease</h2></center>

<strong><center><p>Upload your resume and make sure it passes ATS
checks!</p></center></strong>

<a href="#form" className="cta-button">Upload Your Resume</a>

    </div>

</header>

<form className="form-container" onSubmit={handleSubmit}>

  <h2>Upload Resume</h2>

  <div className="form-group">

    <label>Name: </label>

    <input type="text" name="name" onChange={handleChange} required />

  </div>

  <div className="form-group">

    <label>Email: </label>

    <input type="email" name="email" onChange={handleChange} required />

  </div>

  <div className="form-group">

    <label>Date of Birth: </label>

    <input type="date" name="dob" onChange={handleChange} required />

  </div>

  <div className="form-group">

    <label>State: </label>

    <select name="state" onChange={handleChange} required>

      <option value="">Select State</option>

      <option value="Tamil Nadu">Tamil Nadu</option>

      <option value="Kerala">Kerala</option>

    </select>

  </div>

</form>
```

```
<option value="Madhyapradesh">Madhyapradesh</option>
</select>
</div>
<div className="form-group">
<label>Gender: </label>
<input type="radio" name="gender" value="male" onChange={handleChange}
required /> Male
<input type="radio" name="gender" value="female"
onChange={handleChange} required /> Female
</div>
<div className="form-group">
<label>Preferred Job Location: </label>
<div className="checkbox-group">
<label><input type="checkbox" name="jobLocation" value="Mumbai"
onChange={handleCheckboxChange} /> Mumbai</label>
<label><input type="checkbox" name="jobLocation" value="Bangalore"
onChange={handleCheckboxChange} /> Bangalore</label>
<label><input type="checkbox" name="jobLocation" value="Chennai"
onChange={handleCheckboxChange} /> Chennai</label>
<label><input type="checkbox" name="jobLocation" value="Kolkata"
onChange={handleCheckboxChange} /> Kolkata</label>
</div>
</div>
<div className="form-group">
<label>Upload Resume: </label>
<input type="file" name="avatar" onChange={handleFileChange} required />
</div>
<button className="submit-button" type="submit">Upload Resume</button>
```

```
</form>
```

```
<section id="about" className="about-section">
```

```
<div className="container">
```

```
<h2><strong>About Us</strong></h2>
```

```
<p>At our core, we are committed to empowering job seekers to showcase the best version of themselves to potential employers. In today's competitive job market, a standout resume is crucial, and our resume uploading service is designed to help you achieve just that. We understand that many companies use Applicant Tracking Systems (ATS) to filter resumes, which is why we meticulously ensure that your resume meets all ATS requirements. This means optimizing formatting, keywords, and structure, so that your qualifications shine through and capture the attention of hiring managers.</p>
```

```
Our platform not only assists you in crafting a professional resume but also guides you through the nuances of what employers are looking for. By utilizing our service, you can confidently submit a resume that highlights your skills and experience while adhering to industry standards. We believe that everyone deserves the opportunity to land their dream job, and we are here to support you every step of the way, making your job search journey as smooth and successful as possible. Let us help you make a lasting impression and take the next step toward your career goals.</p>
```

```
</div>
```

```
</section>
```

```
</div>
```

```
);
```

```
};
```

```
export default FormPage;
```

# CHAPTER 7

## SCREENSHOTS

### 7.1 uploading page

ResumeRise - Resume Uploader Platform

About Us Upload Resume Sign In

### Update Your Resume with Ease

Upload your resume and make sure it passes ATS checks!

Upload Your Resume

#### Upload Resume

Name:

Email:

Date of Birth:

State:

Gender: ☐ Male ☐ Female

Preferred Job Location: ☐ Mumbai ☐ Bangalore ☐ Chennai ☐ Kolkata

Upload Resume:  No file chosen

### 7.2 Connectivity page

```
npm start
Compiled with warnings.

[eslint]
src/components/form.js
  Line 88:29:  The href attribute requires a valid value to be accessible. Provide a valid, navigable address as the href value. If you cannot provide a valid href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/jsx-eslint/eslint-plugin-jsx-a11y/blob/HEAD/docs/rules/anchor-is-valid.md jsx-a11y/anchor-is-valid

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

WARNING in [eslint]
src/components/form.js
  Line 88:29:  The href attribute requires a valid value to be accessible. Provide a valid, navigable address as the href value. If you cannot provide a valid href, but still need the element to resemble a link, use a button and change it with appropriate styles. Learn more: https://github.com/jsx-eslint/eslint-plugin-jsx-a11y/blob/HEAD/docs/rules/anchor-is-valid.md jsx-a11y/anchor-is-valid

webpack compiled with 1 warning

Last login: Thu Oct 24 08:55:49 on ttys001
> cd resume-uploader2
> cd backend
> npm start

> backend@1.0.0 start
> node server.js

Server running on port 5001
MongoDB Connected...
[ ]
```

## 7.3 Database page

The screenshot displays the MongoDB Atlas web interface. The top navigation bar includes the Atlas logo, a user profile for 'MADHAN...', and links for 'Access Manager' and 'Billing'. Below this, a secondary navigation bar shows 'Resume Uploader', 'Data Services', and 'Charts'. The left sidebar contains a menu with 'Overview', 'DATABASE', 'Clusters', 'SERVICES', 'Atlas Search', 'Stream Processing', 'Triggers', 'Migration', 'Data Federation', 'SECURITY', 'Quickstart', 'Backup', 'Database Access', 'Network Access', 'Advanced', and 'Goto'. The main content area is divided into tabs: 'Overview', 'Real Time', 'Metrics', 'Collections', 'Atlas Search', 'Performance Advisor', 'Online Archive', and 'Cmd Line Tools'. The 'Collections' tab is active, showing a list of databases and collections. The 'test' database is expanded, revealing the 'candidates' collection. The 'test.candidates' collection page shows statistics: 'STORAGE SIZE: 36KB', 'LOGICAL DATA SIZE: 948B', 'TOTAL DOCUMENTS: 6', and 'INDEXES TOTAL SIZE: 36KB'. Below the statistics are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. The 'Find' tab is selected, displaying a query editor with the filter 'Type a query: { field: 'value' }'. Below the query editor, two document entries are shown in a list view. Each entry displays its unique identifier and a set of key-value pairs representing the document's data.

cloud.mongodb.com/v2/67010374ac51a031d084bf64#/metrics/replicaSet/670104e76179c7c67bda97/explorer/test/candidates/find

6702bc41151bd5f2e00786de

Atlas MADHAN... Access Manager Billing All Clusters Get Help MADHAN

Resume Uploader Data Services Charts

Overview DATABASE Clusters SERVICES Atlas Search Stream Processing Triggers Migration Data Federation SECURITY Quickstart Backup Database Access Network Access Advanced Goto

Overview Real Time Metrics Collections Atlas Search Performance Advisor Online Archive Cmd Line Tools

DATABASES: 2 COLLECTIONS: 7

+ Create Database

Search Namespaces

sample\_mflix

test

candidates

test.candidates

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 948B TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

INSERT DOCUMENT

Filter Type a query: { field: 'value' } Reset Apply Options

Document 1:

```
{
  "_id": ObjectId("6702bc41151bd5f2e00786de"),
  "name": "Ganesh",
  "email": "ganesh@gmail.com",
  "dob": 1972-01-19T00:00:00.000+00:00,
  "state": "West Bengal",
  "gender": "Male",
  "joblocation": Array (1),
  "avatar": "uploads/1728232513459-worst case- rds.pdf",
  "__v": 0
}
```

Document 2:

```
{
  "_id": ObjectId("67038cb03ab19d6f1fabe1d4"),
  "name": "Manoj",
  "email": "manoj@gmail.com",
  "dob": 2004-09-09T00:00:00.000+00:00,
  "state": "Bihar",
  "gender": "Male",
  "joblocation": Array (1),
  "avatar": "uploads/1728285872709-unit 4 ce.pdf",
  "__v": 0
}
```

## 7.Conclusion

The resume uploading platform aims to bridge the gap between job seekers and recruiters by providing a streamlined, secure, and user-friendly experience. Through a well-defined three-tier architecture, the platform integrates frontend, backend, and database solutions that work together seamlessly, offering essential features like resume upload, parsing, and keyword-based matching. Built on a modern tech stack with React.js for the frontend, Node.js for the backend, and PostgreSQL for relational data management, the platform leverages the power of cloud infrastructure for scalability and reliability.

For job seekers, the platform allows effortless resume management, profile creation, and personalized job recommendations, enhancing their job search experience. Recruiters benefit from advanced search and filtering tools that help identify ideal candidates quickly and effectively. Meanwhile, an admin panel provides comprehensive oversight and moderation capabilities, ensuring that user data is well-protected and compliant with data privacy regulations, such as GDPR.

The system's performance and security are prioritized through JWT-based authentication, encrypted data storage, and secure API communication. Additionally, Docker-based containerization and a CI/CD pipeline allow for efficient deployment and maintenance, ensuring continuous updates and system stability.

Overall, the resume uploading platform represents a scalable and robust solution for connecting job seekers and recruiters. With its feature-rich interface, optimized performance, and commitment to security, this platform not only meets the immediate needs of users but also provides a strong foundation for future enhancements and expansion in the recruitment ecosystem.



## **CHAPTER 9**

### **REFERENCES**

1. LinkedIn Jobs (2005): LinkedIn launched its Jobs section to connect professionals with job opportunities based on their profiles, skills, and experience. It includes resume parsing, profile building, and keyword-based job matching.
2. Indeed Resume (2011): Indeed introduced a resume feature allowing job seekers to upload resumes and recruiters to search through candidate profiles using advanced filters and keyword matching, simplifying the hiring process.
3. Glassdoor Job Search & Resume Upload (2015): Glassdoor added resume uploading and job application tracking features to its platform, enabling candidates to manage job applications and receive personalized job recommendations.
4. ZipRecruiter (2010): ZipRecruiter created a platform for job seekers to upload resumes and apply to multiple job postings, while recruiters can access an extensive database of candidates, complete with advanced search functionalities.
5. Jobscan (2013): A tool specifically designed to help candidates optimize their resumes by comparing them against job descriptions for better keyword matching and ATS (Applicant Tracking System) compatibility.