DYNAMIC CV ANALYZER FOR HR

NAME:ARSHPREET SINGH

UID:24BCS120040

NAME: VISHAL CHAUDHARY

UID :24BCS11093

SECTION:24BCS_KRG-711B

SUBJECT : FULLSTACK DEVELOPMENT I

SUBJECT CODE: 24CSP-210

FACULITY : Er.JUNED ALAM

Dynamic CV Analyzer for HRs

1. Introduction

The **Dynamic CV Analyzer for HRs** is a full-stack web application designed to help recruiters and HR professionals efficiently analyze resumes and match them with job descriptions. The system uses NLP techniques to extract skills, experience, and keywords from resumes and computes a compatibility score against job requirements.

---

2. Problem Statement

Recruiters often spend excessive time manually screening resumes. Traditional keyword-based filtering is inefficient and inaccurate. There is a need for an automated, intelligent system that can: - Parse resumes - Extract relevant skills and experience - Match resumes with job descriptions - Rank candidates objectively

---

3. Objectives

- Automate resume screening using NLP
- Improve accuracy of candidate-job matching
- Reduce manual effort for HRs
- Provide a clean dashboard for managing resumes and job postings

---

4. Scope of the Project

In Scope

- Resume upload and parsing
- Job description creation
- Skill extraction and matching
- Match score generation
- HR dashboard

Out of Scope

- Real-time interviews
- Video resume analysis
- Third-party ATS integrations

5. Functional Requirements

- User authentication (HR/Admin)
- Upload resumes in PDF/DOCX format
- Create and manage job descriptions
- Extract skills, education, experience from resumes
- Match resumes with jobs and generate compatibility score
- View ranked list of candidates

---

6. Non-Functional Requirements

- Secure file handling
- Scalability for large resume datasets
- Response time < 3 seconds for analysis
- Maintainable and modular codebase

---

7. Technology Stack

Frontend

- React.js
- HTML, CSS, JavaScript

Backend

- Spring Boot (Java)
- OpenNLP (for NLP processing)
- MySQL (Database)
- REST APIs

---

8. System Architecture (High-Level Design)

The system follows a **client-server architecture**: - Frontend handles UI and user interaction - Backend exposes REST APIs - NLP engine processes resumes - Database stores users, resumes, jobs, and scores

Flow: User → Frontend → Backend APIs → NLP Engine → Database → Frontend

---

9. Low-Level Design

Database Tables

- User (id, name, email, role, password)
- Resume (id, user_id, file_path, parsed_text)
- Job (id, title, description, skills)
- MatchResult (resume_id, job_id, score)

NLP Processing

- Tokenization
- Skill keyword extraction
- Experience and education identification
- Similarity scoring using keyword overlap

---

10. Module Division (Total 8 Modules)

Frontend Modules (3)

1. **Authentication Module**
    - Login / Logout
    - Role-based access
2. **HR Dashboard Module**
    - Job creation
    - Resume listing
    - Match score view
3. **Resume Upload & Results UI**
    - Resume upload form
    - Candidate ranking display

Backend Modules (5)

4. **User Management Module**
    - Authentication APIs
    - Role handling
5. **Resume Management Module**
    - File upload
    - Resume storage
    - Text extraction
6. **Job Management Module**
    - Job CRUD operations

    - Skill requirement storage
7. **NLP & Skill Extraction Module**
    - Resume parsing
    - Skill and keyword extraction using OpenNLP
8. **Matching & Scoring Module**
    - Resume-job comparison
    - Compatibility score calculation
    - Candidate ranking

---

11. Security Considerations

- Password hashing
- Secure file uploads
- Role-based API access

---

12. Future Enhancements

- AI-based semantic matching
- Resume recommendation engine
- Export reports
- Multi-language resume support

---

13. Conclusion

The Dynamic CV Analyzer for HRs is a practical, industry-relevant project that demonstrates full-stack development, backend logic, and NLP integration. It effectively solves a real-world recruitment problem and is suitable for academic as well as resume purposes.