**22AIE311 –SOFTWARE ENGINEERING**

**AI-Based Resume Analyzer**

**GROUP -17**

|  |  |
| --- | --- |
| NAME | ROLL NO |
| Nikitha | AM.EN.U4AIE22127 |
| Sonam | AM.EN.U4AIE22157 |
| Manasa | AM.EN.U4AIE22160 |

**ABSTRACT :**

In today’s fast-paced job market, both recruiters and job seekers face challenges in resume evaluation and optimization. This project introduces an AI-Based Resume Analyzer, a smart system that leverages **Natural Language Processing (NLP) and Machine Learning (ML)** to automate resume screening and provide valuable insights for both recruiters and applicants. The system extracts key information such as **skills, experience, education, and certifications**, then matches them against job descriptions to rank candidates based on relevance.

For recruiters, the tool utilizes **Named Entity Recognition (NER)**, keyword extraction, and classification algorithms to categorize resumes efficiently, generate visual analytics, and provide a resume scoring mechanism for better decisionmaking. For applicants, it offers personalized recommendations, including missing skills, suggested courses, predicted job roles, and resume improvement tips, helping them enhance their resumes and increase their chances of landing their desired jobs. With the integration of deep learning models such as BERT and spaCy, the system ensures fair, unbiased evaluations while improving recruitment efficiency.

This project highlights the practical application of **AI in HR technology**, offering a scalable, data-driven solution that enhances hiring processes, helps applicants refine their resumes through **self-assessment**, and ultimately bridges the gap between job seekers and employers.

Keywords : automate resume screening ,visual analytics ,unbiased evaluations, bridges .

**1. Define the Scope and Requirements**

* **Project Goal:** Automate resume screening and give recommendations.
* **Key Features:**
  + Extract skills, education, and experience.
  + Rank resumes by job fit.
  + Visual analytics and resume scoring.
  + Recommendations for job seekers.

**📊 2. Set Up the Development Environment**

* **Tools:**
  + IDE: *VS Code* or *PyCharm*
  + Version Control: *Git* + *GitHub/GitLab*
  + Project Management: *Trello*, *Jira*, or *Asana*

**📡 3. Data Collection and Preprocessing**

* **Resume Data:** Collect sample resumes (PDF, Word).
* **Job Descriptions:** Scrape or collect real job postings.
* **Data Cleaning:** Remove unnecessary text, fix formatting.
* **Libraries:** *pandas*, *numpy*, *nltk*, *re* (for regex).

**🧩 4. Backend Development**

* **Language:** *Python*
* **Framework:** *Flask* or *Django* for building the API.
* **Database:**
  + Relational: *MySQL* or *PostgreSQL*
  + NoSQL: *MongoDB*
* **Resume Parsing:**
  + *spaCy* for NER (Named Entity Recognition).
  + *PyPDF2* or *pdfplumber* for reading PDFs.
  + *BERT* model for deep learning analysis.

**🎨 5. Frontend Development**

* **Frameworks:** *React*, *Angular*, or *Vue.js*
* **Design:** *Bootstrap* or *Tailwind CSS* for responsive UI.
* **Features:**
  + User Dashboard for recruiters and job seekers.
  + Resume upload form.
  + Visualization of analytics (use *Chart.js* or *D3.js*).

**🧮 6. Machine Learning and NLP**

* **Skills Extraction:** *spaCy* and *BERT*
* **Resume Scoring:** Use a *classification algorithm* (like Logistic Regression, Random Forest, or a simple deep learning model).
* **Job Matching:** *Cosine Similarity* to measure the match between resumes and job descriptions.

**📡 7. Cloud Deployment**

* **Platform:** *AWS*, *Google Cloud*, or *Azure*
* **CI/CD:** Set up with *GitHub Actions* or *Jenkins*
* **Containerization:** Use *Docker* for easy deployment.

**🛠️ 8. Testing**

* **Unit Testing:** *pytest* for backend, *Jest* for frontend.
* **Integration Testing:** Test API connections.
* **User Acceptance Testing (UAT):** Gather feedback from recruiters and job seekers.

**🔒 9. Security**

* **Data Protection:** Encrypt resumes (use *bcrypt* or *AES*).
* **Authentication:** Implement *JWT* (JSON Web Tokens) for secure login.

**📢10. Documentation and Deployment**

* **Documentation:** Use *Swagger* for API docs.
* **Deployment:** Use *Heroku* for initial deployment or *AWS EC2* for scaling.
* **User Guide:** Write a clear guide on how to use the system.