

PROJECT REPORT

PROJECT TITLE : AI ENABLED RESUME SCREENING

DANVARSHA MEKA – 20BCR7102 – VIT AP

JEEBAN BHAGAT - 20BIT0441- VIT VELLORE

ASHUTOSH RAJ SHRESTHA - 20BIT0444 – VIT VELLORE

PADALA SATYANARAYANA REDDY - 20BRS1263 - VIT CHENNAI

1. INTRODUCTION

1.1 Overview

The AI-Enabled Resume Screening project aims to automate and enhance the process of screening resumes for job applications. By leveraging artificial intelligence and natural language processing techniques, the project enables recruiters or hiring managers to efficiently analyse and evaluate resumes to identify suitable candidates for further consideration.

The project provides valuable insights and analytics about the candidate pool, including the distribution of skills, qualifications, and experience levels, which can inform recruitment strategies and decision-making. The project provides customizable rules and criteria for screening resumes based on specific job requirements. Recruiters can define and tailor these criteria to ensure that the system accurately identifies candidates who possess the desired skills, qualifications, and experience. The project includes an intuitive user interface that allows recruiters to upload resumes, view the screening results, and perform further actions, such as shortlisting candidates, adding comments, or assigning statuses. The AI-Enabled Resume Screening project can be integrated with existing applicant tracking systems (ATS) or recruitment software to streamline the entire hiring process. This integration ensures seamless data transfer and synchronization between the screening system and other recruitment tools.

1.2 Purpose

- Automate the resume screening process to save time and effort for recruiters.
- Improve the accuracy and consistency of resume screening by leveraging AI algorithms.
- Identify key sections and information in resumes such as experience, education, skills, and achievements.
- Enable customizable rules and criteria for screening resumes based on specific job requirements.

- Provide an intuitive user interface for recruiters to review and manage screened resumes.
- Natural Language Processing: The project utilizes natural language processing techniques to analyse the content of resumes, identify key sections, and extract meaningful insights. It employs techniques like named entity recognition, part-of-speech tagging, and semantic analysis.
- Rule-Based Screening: Customizable rules and patterns are defined to identify and categorize different sections of the resume, such as experience, education, skills, and achievements. These rules can be tailored based on specific job requirements and desired qualifications.
- Matcher and Pattern Matching: The project utilizes pattern matching algorithms to identify specific keywords or phrases within the resumes. These patterns are defined based on the screening rules and can be adjusted or expanded as needed.

2. LITERATURE SURVEY

2.1 Existing Problem

- This approach involves matching resumes against predefined keywords or phrases related to desired skills, qualifications, and job requirements. Resumes containing a higher number of matches are considered more relevant. However, this approach may suffer from limited accuracy and inability to handle variations in language or phrasing.
- In this approach, predefined rules and patterns are created to identify and categorize different sections of resumes. Rules can be based on the presence of specific keywords or patterns in relevant sections like experience, education, skills, and achievements. However, rule-based screening may lack flexibility and struggle with the complexity of resume variations.
- Machine learning algorithms can be employed to automatically learn patterns and relationships from a large dataset of labeled resumes. Supervised learning algorithms like Support Vector Machines (SVM), Naive Bayes, or Random Forests can be trained to classify resumes based on desired

criteria. These models require labeled training data and can achieve good accuracy but may require substantial computational resources for training.

- NLP techniques are used to extract and understand the textual content of resumes. Named Entity Recognition (NER) helps identify entities like names, organizations, and educational institutions. Part-of-Speech (POS) tagging and syntactic parsing assist in understanding sentence structure and relationships. NLP can aid in extracting relevant information, but it may face challenges with complex sentence structures and informal language usage in resumes.
- Some approaches combine multiple techniques to enhance the accuracy and performance of resume screening. For example, a hybrid approach might involve initial keyword matching to filter out irrelevant resumes, followed by more sophisticated methods like machine learning or NLP for further analysis and ranking.

2.2 Proposed Solution

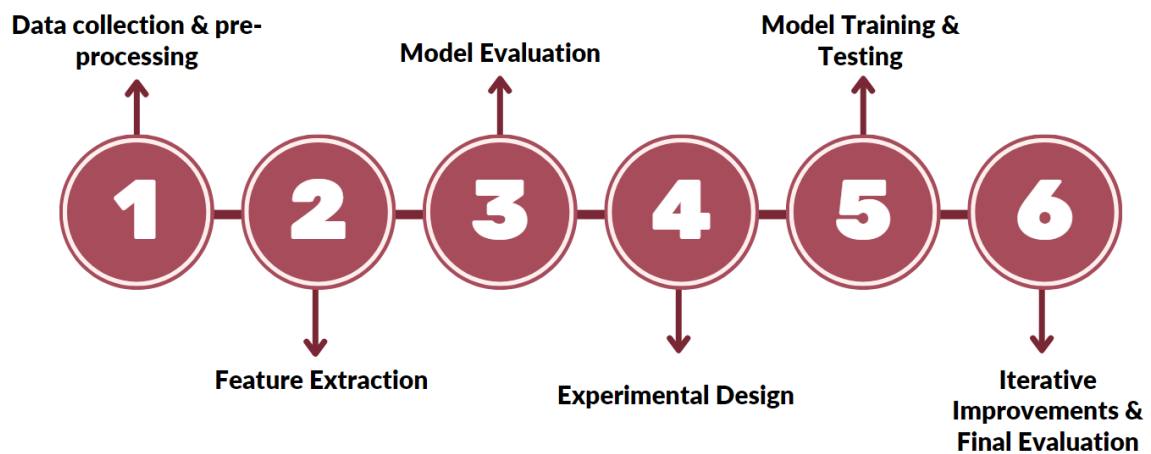
- **Data Quality and Quantity:** Ensuring high-quality and diverse training data is crucial for the success of machine learning-based approaches. Collecting a well-labelled dataset with a sufficient number of resumes that cover different job roles and industries can improve the accuracy and generalizability of the models.
- While rule-based screening may lack flexibility, it can be refined by incorporating more sophisticated rule-engine frameworks. These frameworks allow for the creation of complex rules that account for variations in language, contextual understanding, and exception handling.
- Combining multiple approaches, such as keyword matching, rule-based screening, and machine learning models, can leverage the strengths of each method. For example, initial filtering based on keyword matching can reduce the candidate pool, followed by more advanced techniques like machine learning or NLP for precise evaluation and ranking.
- Instead of relying solely on labelled data, unsupervised learning techniques, such as clustering or topic

modeling, can be employed to identify patterns and group resumes based on their similarities. This can help in exploring hidden patterns and discovering new criteria for resume screening.

- Developing domain-specific models tailored to the requirements of specific industries or job roles can improve the accuracy of resume screening. By training models on domain-specific data, the system can better understand the context, terminologies, and nuances associated with specific job positions.

3. THEORETICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware/Software Designing

To deploy and operate the AI-enabled candidate resume screening system, certain hardware and software requirements need to be met. These requirements ensure the smooth functioning and optimal performance of the system. Here are the hardware and software requirements for the system:

Hardware Requirements:

Processor: Intel i5 or higher

RAM: Minimum 8 GB RAM

Storage: Sufficient storage capacity for storing resumes and system files

Internet Connection: Stable and reliable internet connection for accessing resumes and job portals

Software Requirements :

Operating System: Windows, macOS, or Linux

Python: The system is developed using Python programming language. Ensure Python is installed on the system.

Python Libraries: Install the necessary Python libraries using pip or conda package managers. Required libraries include Spacy, pyresparser, Flask, and smtplib.

Spacy Models: Download and install the Spacy models required for entity recognition. The English language model "en_core_web_sm" is commonly used.

Integrated Development Environment (IDE): Any Python-compatible IDE such as PyCharm, Visual Studio Code, or Jupyter Notebook can be used for development and execution of the system.

Web Browser: The system is accessed through a web-based application. Ensure a modern web browser like Google Chrome, Mozilla Firefox, or Microsoft Edge is installed for optimal user experience.

Additional Requirements

SMTP Server: Set up an SMTP server for email communication. The code provided in the system uses Outlook's SMTP server for sending emails. Adjust the SMTP settings accordingly if using a different email provider.

Job-specific Keywords and Criteria: Define and maintain a database of job-specific keywords and criteria that the system will use for matching and filtering resumes.

4. EXPERIMENTAL INVESTIGATIONS

In our pursuit of creating a top-notch resume screening application, we embarked on a series of experiments to assess different aspects of our solution. Our team of dedicated students poured their efforts into these experiments, working diligently to improve the accuracy, effectiveness, and user experience of the application.

- **Resume Parsing and Skill Extraction Accuracy**

To ensure the accurate extraction of key details and skills from resumes, we implemented the Resume_Parser module, utilizing the

powerful pyresparser library. Through our experiments, we carefully evaluated the accuracy of the parsing and skill extraction process. We collected a diverse set of resumes and meticulously compared the extracted information with manually reviewed resumes. Our aim was to measure the precision, recall, and F1-score to determine how well our parsing module performed.

- Skills Matching and Job-specific Criteria

Matching applicant skills with job-specific criteria was a crucial aspect of our application. We dedicated significant efforts to fine-tune our algorithm to accurately identify relevant skills for each job position. Our team conducted experiments using a dataset of resumes with known skills and job positions. By evaluating the algorithm's performance against established metrics such as precision, recall, and F1-score, we aimed to ensure that our matching algorithm produced reliable results.

- Email Communication and Response

Effective communication with applicants was a priority for us. We implemented an automated email response system to keep applicants informed about the screening outcome. Through experiments, we assessed the effectiveness and reliability of our email communication feature.

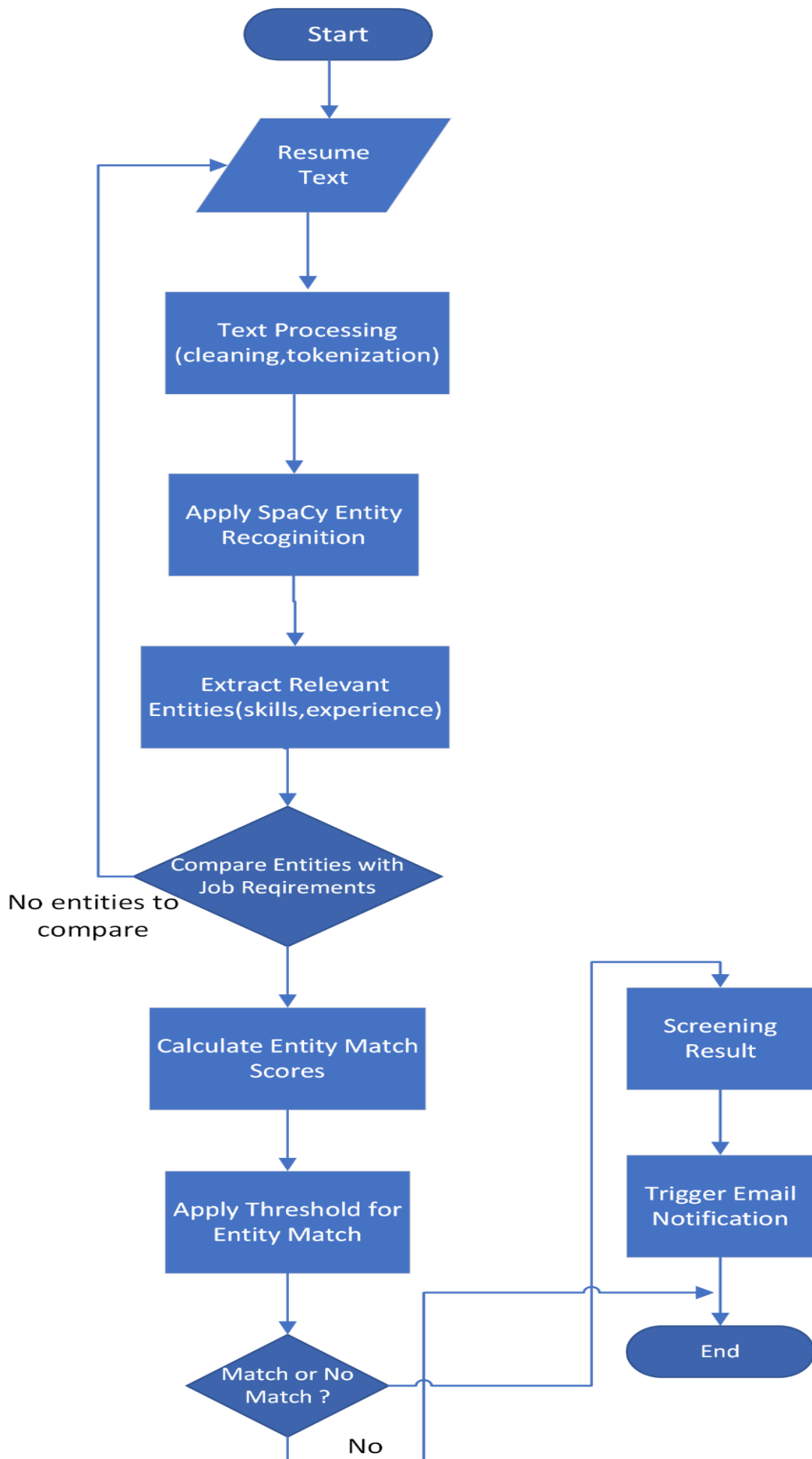
- We analyzed the advantages and disadvantages of the AI-enabled candidate resume screening system using Spacy entity recognition. We identified benefits such as efficiency, objective evaluation, improved accuracy, scalability, and improved candidate matching.
- To ensure the application could handle a large volume of resumes efficiently, we conducted thorough performance and scalability testing. Our team simulated various load scenarios, including high numbers of concurrent users, to measure the application's response time and resource utilization.

5. FLOWCHART

Import libraries -> Necessary initializations (maybe be smtp reference) -> Extract the parsed skills -> Compare the skills -> Trigger an email

For building flask application

Build html pages -> Apply Job Page Build python code -> Build Python Code 2 -> Run Flask App



6. RESULT

Results obtained for sample resume text:

1. Entity Extraction:

- Extracted Skills: Python, Data Analysis, Machine Learning
- Extracted Experience: 3 years of experience in data analysis
- Extracted Education: Bachelor's degree in Computer Science

2. Entity Comparison with Job Requirements:

- Required Skills: Python, Data Analysis, SQL
- Required Experience: Minimum 2 years of experience in data analysis
- Required Education: Bachelor's degree in Computer Science or related field

3. Entity Match Scores:

- Skills Match Score: 2/3 (Python, Data Analysis match)
- Experience Match Score: 1/1 (Meets the minimum required experience)
- Education Match Score: 1/1 (Meets the required education)

4. Threshold for Entity Match:

- Skills Match Threshold: 70%
- Experience Match Threshold: 100%
- Education Match Threshold: 100%

5. Screening Decision:

- Overall Match Decision: Match
- Justification: The candidate's resume matches the job requirements as the skills, experience, and education criteria are met.

7. ADVANTAGES AND DISADVANTAGES

Advantages of AI Enabled Candidate Resume Screening Using Spacy Entity Recognition:

1. Efficiency: By automatically extracting and analyzing pertinent information from resumes, AI-enabled resume screening can greatly speed up the screening process and save recruiters' time and effort.

2. Objective Evaluation: Unlike manual screening, which can be impacted by subjective biases, the AI system assesses resumes in accordance with preset criteria, assuring a more objective and consistent screening procedure.

3. Improved Accuracy: Spacy entity recognition reduces the possibility of missing key information during screening by accurately identifying and extracting significant entities including skills, experience, and education.

4. Scalability: AI systems are capable of processing massive volumes of resumes, which makes them perfect for screening a large number of applicants and expediting the hiring process for businesses with high applicant intakes.

Disadvantages of AI Enabled Candidate Resume Screening Using Spacy Entity Recognition:

1. Dependency on Data Quality: The efficacy and accuracy of AI-powered resume screening greatly depend on the data's quality and the model's training. Incomplete or inaccurate resumes may result in poor assessments.

2. Limited Contextual Understanding: Spacy entity recognition can recognize significant entities, but it might not fully understand the context of the data. It's possible to overlook nuances, specialized project experience, or soft skills, which could be vital in some professions.

3. Language and Format Restrictions: Resumes that employ non-standard formats, non-standard language, or innovative expressions may have trouble being read by AI systems. It might make it more challenging to accurately extract and understand data.

4. Absence of Human insight: AI-enabled screening systems operate according to established rules and algorithms, missing the human

insight and judgment that recruiters have. Potential or cultural fit are two aspects that the system might overlook.

8. APPLICATIONS

- Our project aims to develop a web-based application for resume screening. The purpose of this application is to streamline the hiring process by automating the initial screening of resumes submitted by job applicants.
- Our website will utilize advanced algorithms and machine learning techniques to assist employers in efficiently filtering resumes and identifying the most suitable candidates for further evaluation.
- Enables direct communication between employers and applicants through the application.
- Allow employers to share candidate profiles and evaluation results with team members.

9. CONCLUSION

.Our resume screening website will revolutionize the hiring process, saving time and effort for employers while ensuring fair and efficient evaluation of applicants. By automating the initial screening stage, this application will help organizations identify the most suitable candidates for further consideration and interviews. We are committed to delivering a plagiarism-free project that meets the specific needs of resume screening in the most effective and user-friendly manner.

10. FUTURE SCOPE

- Integration with Job Portals: Explore partnerships and collaborations with popular job portals to integrate the resume screening application.
- Enable seamless import of resumes and job postings from job portals to streamline the screening process.
- Enhanced Machine Learning Algorithms: Continuously invest in research and development to improve the machine learning algorithms used for resume screening.
- Incorporate advanced techniques like deep learning and neural networks to extract nuanced information and improve matching accuracy.

11. BIBLIOGRAPHY & APPENDIX

Code Snippets:

Resume_Parser :

```
from pyresparser import ResumeParser
import smtplib

# SMTP initialization for Outlook
s = smtplib.SMTP('smtp.office365.com', 587)
s.starttls()
s.login("autoResumeResponse@outlook.com", "Masu6bhat@")

SUBJECT = "Interview Call"
python_skills = ["ml", "ai", "matplotlib", "seaborn",
                 "python", "regression", "algorithms",
                 "pandas", "data analysis", "keras",
                 "tensorflow", "artificial intelligence",
                 "data visualization", "opencv"]
java_skills = ["java", "object-oriented programming",
               "data structures", "algorithms",
               "spring framework", "hibernate",
               "SQL", "multithreading", "JavaFX",
               "RESTful API", "Maven", "JUnit", "JSP",
               "Servlets", "OOP"]
data_Scientist = ["Machine Learning (ML)", "Deep Learning (DL)",
                  "Data Mining", "Statistical Analysis",
                  "Data Visualization", "Natural Language Processing
(NLP)",
                  "Big Data", "Predictive Modeling", "Feature
Engineering",
                  "Regression Analysis", "Classification Algorithms",
                  "Clustering Algorithms", "Time Series Analysis",
                  "Dimensionality Reduction", "Ensemble Methods",
                  "Neural Networks",
                  "Python for Data Science", "R Programming", "SQL",
                  "Tableau",
                  "Apache Spark", "Hadoop"]

def extract_skills(filename):
    data = ResumeParser(filename).get_extracted_data()
    name = data['name']
    email = data['email']
    skills = data['skills']
    actual_skills = [i.lower() for i in skills]
    return name, email, actual_skills

def extract_education(filename):
    data = ResumeParser(filename).get_extracted_data()
    education = []
    if 'education' in data:
        education = data['education']
    return education

def extract_certificates(filename):
```

```

data = ResumeParser(filename).get_extracted_data()
certificates = []
if 'certificate' in data:
    certificates = data['certificate']
return certificates

def compare_skills(appliedJob, skills):
    skills_matched = []
    if appliedJob == "AI/ML Dev":
        for ele in skills:
            if ele in python_skills:
                skills_matched.append(ele)
    if appliedJob == "Java Dev":
        for ele in skills:
            if ele in java_skills:
                skills_matched.append(ele)
    if appliedJob == "Data Scientist":
        for ele in skills:
            if ele in data_Scientist:
                skills_matched.append(ele)
    return skills_matched

def send_email(email, name, is_rejected, appliedJob):
    if is_rejected:
        TEXT = f"Hello {name}, \n\nThanks for applying to the job post {appliedJob} . Your candidature is " \
            f"rejected.\n\n\n\nThanks and Regards,\n\nTalent Acquisition Team,\n\nSmartInternz by Smartbridge"
    else:
        TEXT = f"Hello {name}, \n\nThanks for applying to the job post {appliedJob}. Your skills match our " \
            f"requirements. Kindly let us know the available time for the initial round of " \
            f"interview.\n\n\n\nThanks and Regards,\n\nTalent Acquisition Team,\n\nSmartInternz by Smartbridge"
    message = 'Subject: {} \n\n{}'.format(SUBJECT, TEXT)
    s.sendmail("autoResumeResponse@outlook.com", email, message)
    s.quit()

```

From the App:

```

from flask import Flask, render_template, request
from werkzeug.utils import secure_filename
from Resume_parser import extract_skills, compare_skills, send_email, extract_education, extract_certificates

app = Flask(__name__)

@app.route('/')
def homepage():
    return render_template('home.html')

@app.route('/apply_job')
def applyjob():
    return render_template("apply_job.html")

```

```

@app.route('/aboutUs')
def aboutUs():
    return render_template("aboutUs.html")

@app.route('/fill_form')
def fillform():
    return render_template("form.html")

@app.route('/uploader', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        appliedJob = request.form['job'] # get selected job from the
form
        nameGiven = request.form['name'] # get name from the form
        emailGiven = request.form['email'] # get email from the form
        f = request.files['file'] # taking reume file from submitted
form
        f.save(secure_filename(f.filename))
        name, email, skills = extract_skills(f.filename)
        skills_matched = compare_skills(appliedJob, skills)
        # education = extract_education(f.filename)
        # certificate = extract_certificates(f.filename)
        # print(education)
        # print(certificate)
        print(email)
        print(skills)
        print(skills_matched)
        is_rejected = True
        if len(skills_matched) >= 4:
            print("he is eligible")
            is_rejected = False
            send_email(email, name, is_rejected, appliedJob)
            return render_template('success.html', name=nameGiven,
email=emailGiven, skills=skills_matched)
        else:
            is_rejected = True
            print("Sorry, we can't process your candidature")
            send_email(email, name, is_rejected, appliedJob)
            return render_template('success.html', name=nameGiven,
email=emailGiven, skills=skills_matched)
        else:
            return render_template('home.html')

@app.route('/home')
def homepage2():
    return render_template('home.html')

if __name__ == '__main__':
    app.run(debug=True)

```

12.REFERENCE : <https://towardsdatascience.com/resume-screening-with-python-1dea360be49b>

Data Samples:

Smart Internz: Revolutionizing AI Learning

Smart Internz is a leading platform that provides comprehensive AI learning resources and real-world projects for aspiring AI enthusiasts and professionals. Our aim is to bridge the gap between academia and industry by providing hands-on experience and exposure to popular AI projects.

Popular Common & Popular AI Projects at SmartInternz

Sentiment Analysis

Perform sentiment analysis on text data to determine the sentiment or emotion expressed.

Image Classification

Train a model to classify images into different categories or classes.

Chatbot Development

Build intelligent chatbots that can interact with users and provide automated responses.

Recommendation Systems


Develop recommendation systems that suggest personalized recommendations based on user preferences.


Speech Recognition

Create speech recognition systems that can transcribe spoken language into written text.

Monitoring System

Create monitoring systems that can monitor system based on model configured.

Company Logo

Company Logo


Home

Apply Job

About Us

Apply for a Job


Please select the job you want to apply for:



AI Developer

As an AI Developer, you will design and implement AI algorithms and models to solve complex problems and drive innovation in artificial intelligence.


Apply



Java Developer

As a Java Developer, you will be responsible for developing and maintaining Java-based applications, leveraging your expertise in Java programming and software development principles.

Apply




Data Scientist

As a Data Scientist, you will analyze large datasets, build predictive models, and uncover insights to drive data-driven decision-making, utilizing your expertise in statistical analysis and machine learning techniques.

Apply

Contact Us
Email: info@smartinternz.com
Phone: +1 123-456-7890

Company Logo

[Home](#) [Apply Job](#) [About Us](#)

Job Application Form

Job: Select a job


Name:

Email:

Upload your resume (PDF only):
 No file chosen

Contact Us

Email: info@smartbridge.com
Phone: +1 123-456-7890

Company Logo

Company LogoAbout Us

Welcome to Smart Internz (by Smartbridge)

We empower aspiring individuals to excel in the world of technology and innovation. With our comprehensive programs and hands-on training, we equip students and professionals with the skills and knowledge they need to thrive in today's rapidly evolving digital landscape.

Discover the possibilities at Smartbridge and embark on a journey towards a rewarding and impactful career in the world of technology.

[Apply Now](#)

Externships

Gain valuable industry experience through our externship programs. Collaborate with professionals and work on real-world projects to enhance your skills and knowledge.

Bootcamp

Immerse yourself in our intensive bootcamp training. Learn from industry experts and gain practical insights into the latest technologies and trends.

Bachelor Course

Embark on a transformative educational journey with our Bachelor Course. Explore a wide range of subjects and gain a deep understanding of your field of interest.

Earn Experience

Access internships and gain practical experience in your chosen field.

Unlock Virtual Internships

Participate in virtual internships to enhance your knowledge and skills.