Tittle: Ai-Based Resume Screening

Team Members:

Param Joshi 20BIT0326

Ashish Absolom 20BIT0201

Divyanshu Kumar 20BIT0418

Chhavi Jain 20BKT0144

1 INTRODUCTION

1.1 Overview

This report presents a project focused on developing a website for resume screening using AI-based technology. The aim is to create a system that can efficiently analyze resumes in PDF or Word format and match them with specific keywords relevant to job requirements. By generating suitability scores for candidates, the system provides valuable feedback on their chances of success.

1.2 Purpose

The purpose of this project is to streamline the resume screening process, allowing recruiters to save time and effort. By automating the initial screening phase, the system aims to enhance the efficiency and effectiveness of candidate selection, resulting in improved hiring outcomes.

2 LITERATURE SURVEY

2.1 Existing problem

Traditional resume screening methods are often time-consuming and susceptible to human bias. Recruiters face challenges in sifting through numerous resumes to identify the most suitable candidates. This can lead to inefficiencies, delayed response times, and the potential oversight of qualified individuals.

2.2 Proposed solution

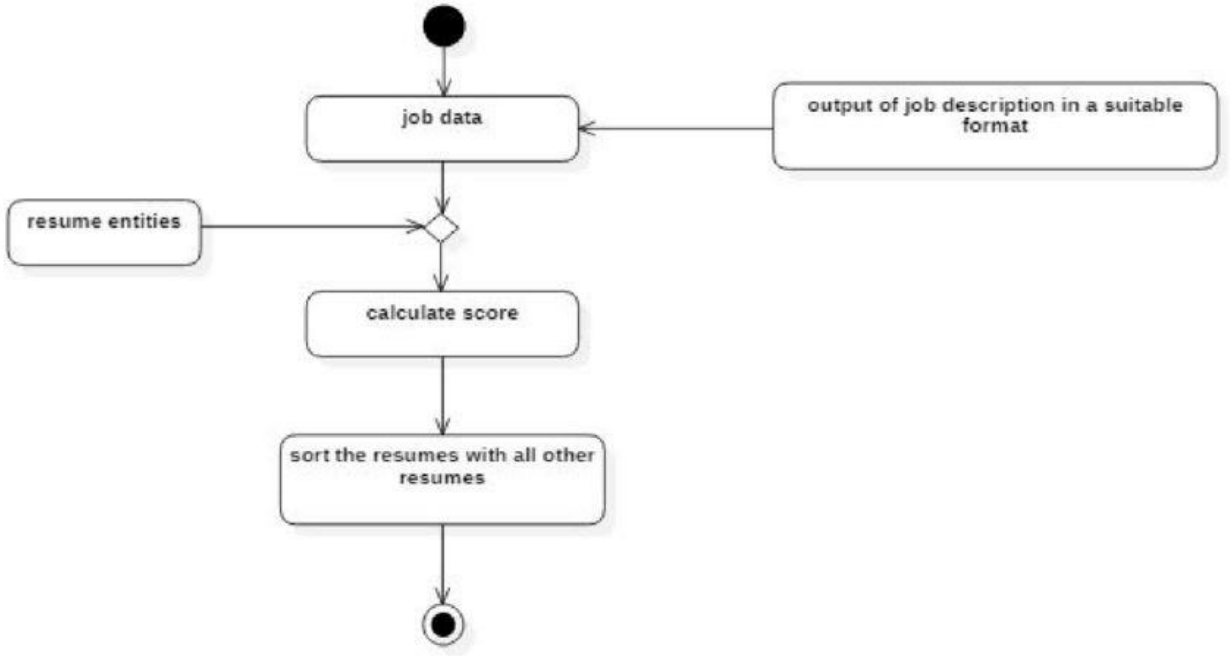
To address these challenges, the proposed solution involves an AI-based resume screening website. The system utilizes advanced techniques such as natural language processing (NLP) and keyword matching

algorithms to automate the analysis of resumes. By doing so, it aims to optimize the initial screening phase, allowing recruiters to focus on the most qualified candidates.

3 THEORETICAL ANALYSIS

3.1 Block diagram

A diagrammatic representation of the project's architecture and flow is provided below.



3.2 Hardware / Software designing

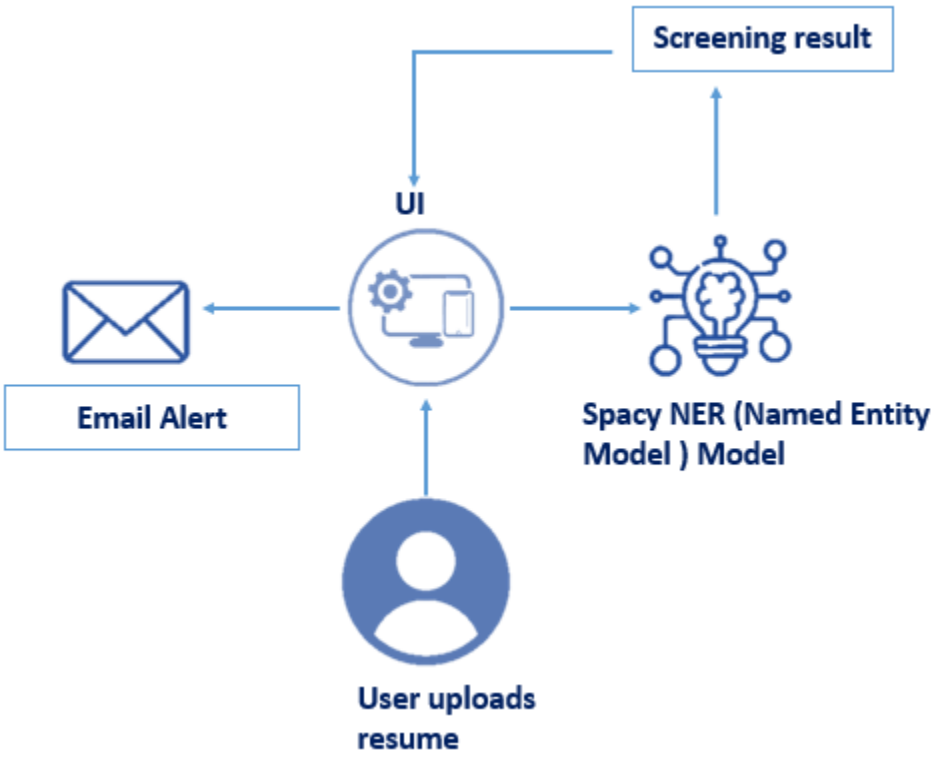
The project requires specific hardware and software components, including a server or hosting environment for deploying the website. Software-wise, the development incorporates HTML, CSS, Python, the Flask framework, and NLP libraries like pyresparser and SpaCy for resume parsing and keyword matching.

4 EXPERIMENTAL INVESTIGATIONS

Throughout the project development, extensive experiments and investigations were conducted to refine and validate the solution. These included building and training NLP models, testing the accuracy of the keyword matching algorithm, and integrating email services for candidate notifications.

5 FLOWCHART

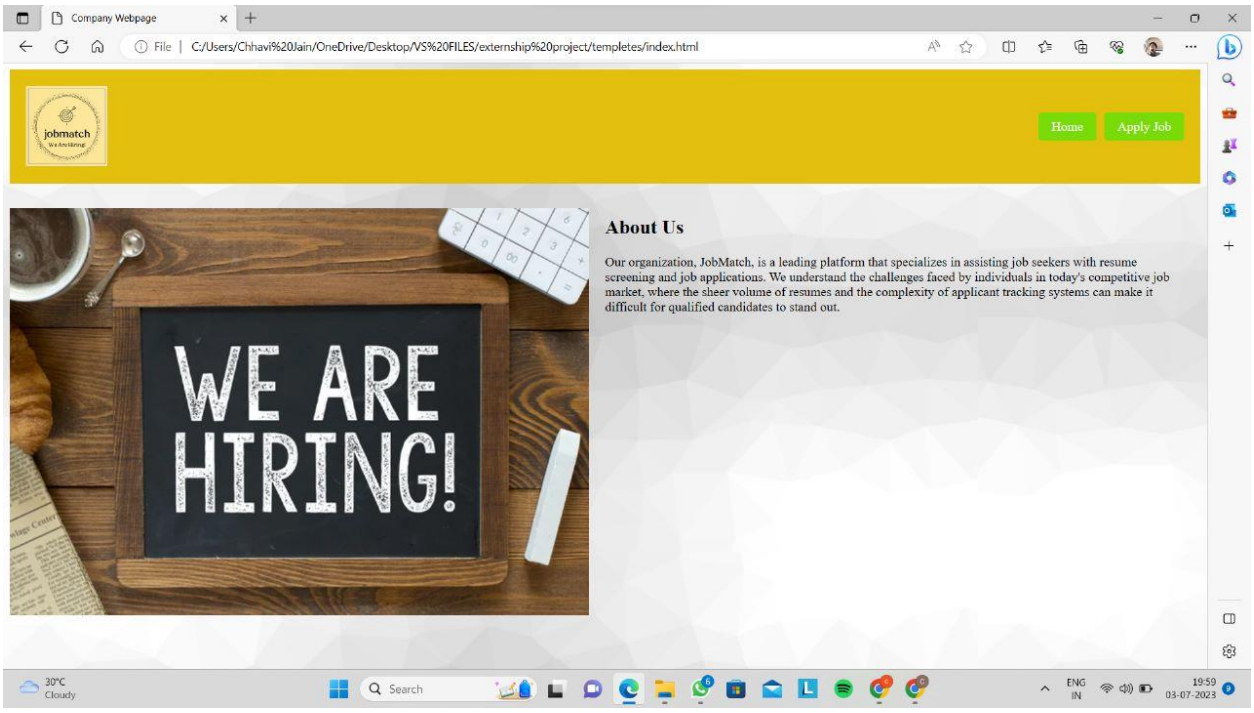
The control flow of the solution is illustrated below.



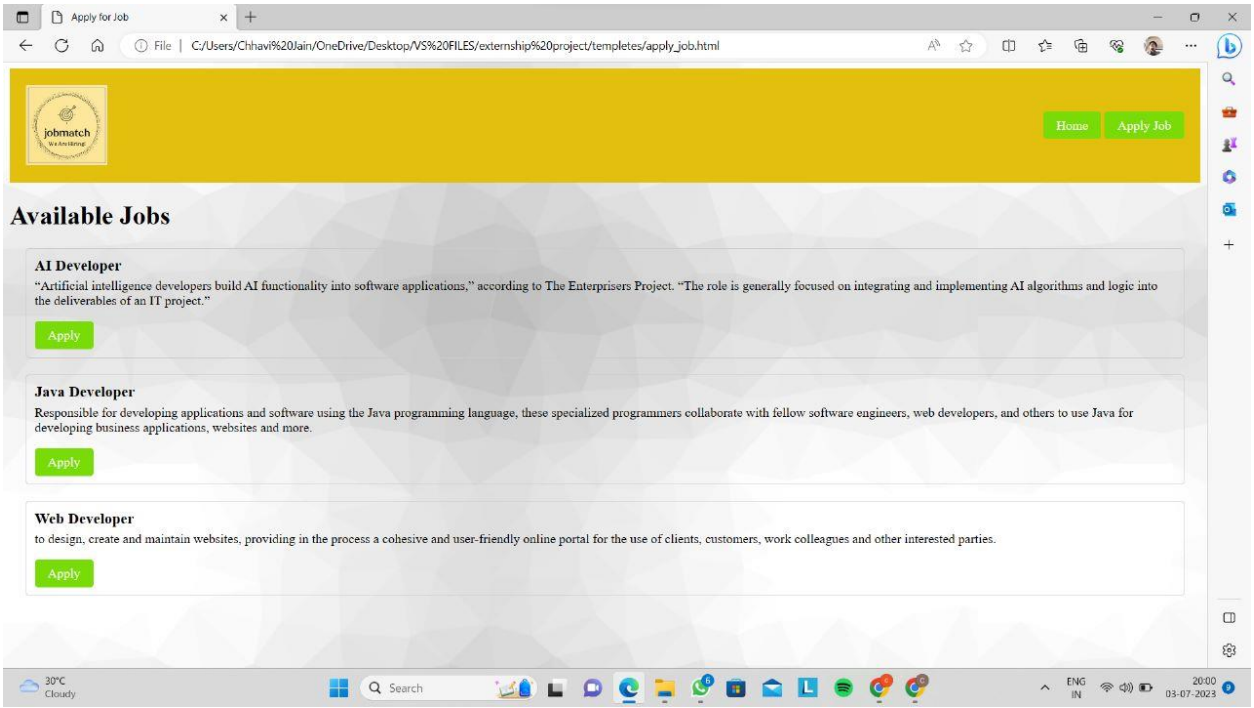
6 RESULT

The final result of the project is an operational resume screening website. The system effectively scans resumes in PDF or Word format, matches keywords to job requirements, generates suitability scores, and notifies candidates via email. Screenshots displaying the website's interface are shown below.

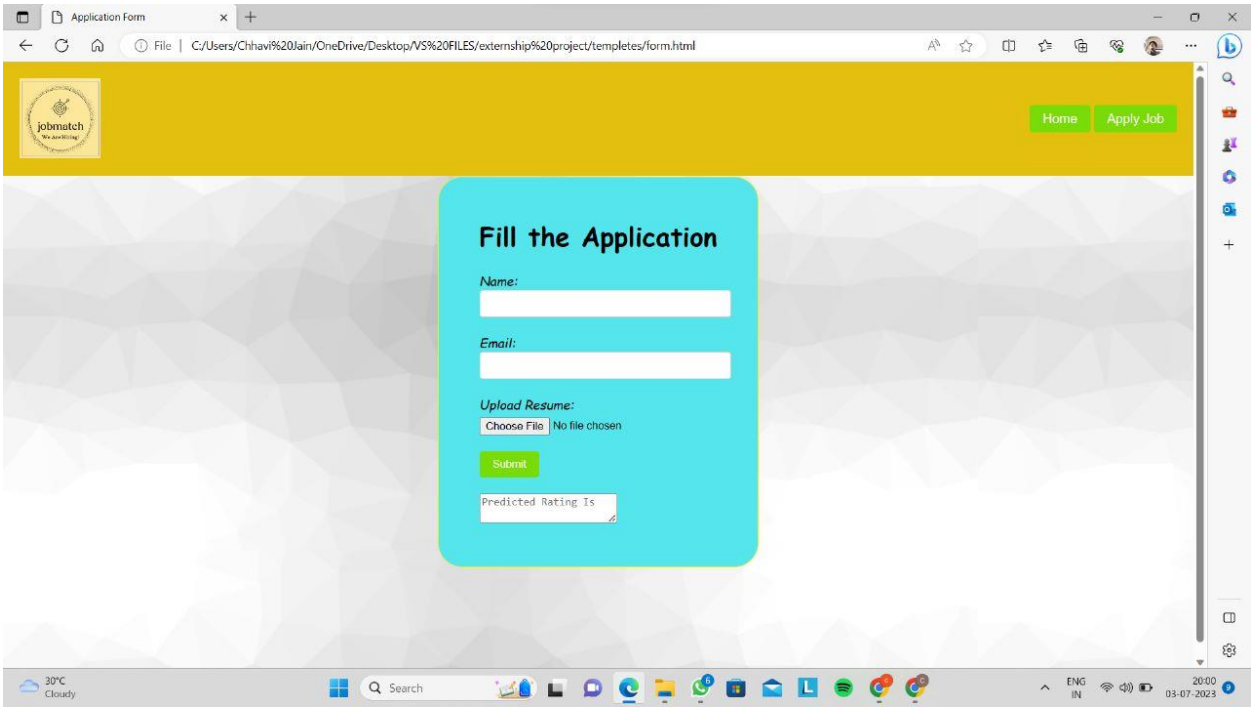
Home page:



Job application page:



Form:



7 ADVANTAGES & DISADVANTAGES

The proposed solution offers several advantages, including:

- Time and effort savings for recruiters through automated resume screening
- Reduced human bias in the initial screening process with objective keyword matching
- Improved efficiency in identifying qualified candidates for job positions

However, there are a few limitations and potential disadvantages to consider:

- The accuracy of the keyword matching algorithm and NLP models may be limited
- Parsing complex resume formats or unconventional layouts may pose challenges
- Soft skills and non-keyword-based qualifications may not be adequately assessed

8 APPLICATIONS

The proposed solution has applications in various domains, including:

- Recruitment agencies

- HR departments of organizations
- Online job portals and career websites

9 CONCLUSION

In conclusion, the project successfully developed an AI-based resume screening website that automates the initial screening phase of the hiring process. By efficiently matching keywords to job requirements, the system enhances the effectiveness of candidate selection and improves the overall quality of hiring outcomes.

10 FUTURE SCOPE

Future enhancements could include fine-tuning the keyword matching algorithm for improved accuracy, incorporating advanced NLP techniques for extracting additional information from resumes, integrating machine learning models to assess soft skills, and expanding compatibility to support a wider range of resume formats.

11 BIBLIOGRAPHY

A list of references, including previous works, websites, books, or research papers consulted during the project, is provided below.

1. Resumes and Cover Letters: The Balance Careers. Available at: https://www.thebalancecareers.com/resumes-4161919
2. Text Mining with R: A Tidy Approach by Julia Silge and David Robinson. O'Reilly Media, 2017.
3. Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper. O'Reilly Media, 2009.
4. "Automatic Extraction of Keywords from Resumes Using NLP Techniques" by Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Proceedings of the 6th International Conference on Data Mining (DMIN), 2010.
5. "Resume Mining: A Survey" by Smita S. Kharde, and Vijaya R. Wadhai. International Journal of Engineering and Innovative Technology (IJEIT), Volume 4, Issue 6, 2015.
6. "Efficient Parsing of Resume Documents" by Kishore Jonnalagedda, Jeff Sweeney, and Keith Curtis. Proceedings of the 18th ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2018.
7. "An Effective Resume Screening Method based on Syntactic and Semantic Analysis" by Tarek M. Mahmoud, Norisma Idris, and Shamsul Sahibuddin. International Journal of Electrical and Computer Engineering (IJECE), Volume 9, Issue 4, 2019.
8. spaCy Available at: https://spacy.io/
9. NLTK Available at: https://www.nltk.org/
10. PyResParser GitHub Repository Available at: https://github.com/OmkarPathak/pyresparser

APPENDIX

A. Source Code

homepage.html:

```
<!DOCTYPE html>
<html>

<head>
  <title>Company Webpage</title>
  <style>
    .header {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 20px;
      background-color: #e3bf0e;
    }

    .logo {
      height: 100px;
      width: 100px;
    }

    .nav-buttons {
      display: flex;
      gap: 10px;
    }

    .nav-buttons a {
      padding: 8px 16px;
      text-decoration: none;
      background-color: #79dc09;
      color: #fff;
      border-radius: 4px;
    }

    .content-container {
      display: flex;
    }

    .image-container {
      flex: 1;
      text-align: right;
    }

    .image-container img {
      max-width: 100%;
```

```
      height: 500px;
      padding-top: 30px;
    }

    .content {
      flex: 1;
      padding: 20px;
    }
  </style>
</head>

<body background="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\bg1.jpg" background-size="cover">
  <header class="header">
    <div class="logo">
      <img src="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\Logo.png" width="100px"
        alt="Company Logo">
    </div>
    <nav class="nav-buttons">
      <a href="index.html">Home</a>
      <a href="apply_job.html">Apply Job</a>
    </nav>
  </header>
  <div class="content-container">
    <div class="image-container">
      <img src="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\hiring.jpg" alt="Image Description">
    </div>
    <div class="content">
      <h2>About Us</h2>
      <p>Our organization, JobMatch, is a leading platform that specializes in assisting job seekers with resume
        screening and job applications. We understand the challenges faced by individuals in today's competitive job
        market, where the sheer volume of resumes and the complexity of applicant tracking systems can make it difficult
        for qualified candidates to stand out.</p>
    </div>
  </div>
</body>

</html>
```

form.html:

```
<!DOCTYPE html>
<html>

<head>
  <title>Application Form</title>
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      font-family: Arial, sans-serif;
    }

    .header {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 20px;
      background-color: #e3bf0e;
      position: fixed;
      top: 0;
      left: 0;
      right: 0;
    }

    .logo {
      height: 100px;
      width: 100px;
    }

    .nav-buttons a {
      padding: 8px 16px;
      text-decoration: none;
      background-color: #79dc09;
      color: #fff;
      border-radius: 4px;
    }

    .form-container {
      max-width: 500px;
      padding: 50px;
      border: 1px solid #e8f406;
```

```
      border-radius: 30px;
      background-color: hsl(183, 81%, 63%);
    }

    .form-container h1 {
      margin-top: 0;
      font-family: cursive;
    }

    .form-group {
      margin-bottom: 20px;
    }

    .form-group label {
      display: block;
      font-style: italic;
      font-family: cursive;
    }

    .form-group input[type="text"],
    .form-group input[type="email"] {
      width: 100%;
      padding: 8px;
      border: 1px solid #ccc;
      border-radius: 4px;
    }

    .form-group input[type="file"] {
      margin-top: 5px;
    }

    .form-group input[type="submit"] {
      padding: 8px 16px;
      background-color: #79dc09;
      color: #fff;
      border: none;
      border-radius: 4px;
      cursor: pointer;
    }
  </style>
</head>

<body background="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\bg1.jpg" background-
size="cover">
  <header class="header">
    <div class="logo">
      <img src="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\Logo.png" width="100px"
        alt="Company Logo">
    </div>
    <nav class="nav-buttons">
      <a href="index.html">Home</a>
      <a href="apply_job.html">Apply Job</a>
    </nav>
  </header>

  <div class="form-container">
```

```
    <h1>Fill the Application</h1>
    <form>
      <div class="form-group">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
      </div>
      <div class="form-group">
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
      </div>
      <div class="form-group">
        <label for="resume">Upload Resume:</label>
        <input type="file" id="resume" name="resume" accept=".pdf" required>
      </div>
      <div class="form-group">
        <input type="submit" value="Submit">
      </div>
      <textarea placeholder="Predicted Rating Is" name="prediction"></textarea>
    </form>
  </div>

</body>

</html>
```

apply_job.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Apply for Job</title>
  <style>
    .header {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 20px;
      background-color: #e3bf0e;
    }
    .logo {
      height: 100px;
      width: 100px;
    }
    .nav-buttons a {
      padding: 8px 16px;
      text-decoration: none;
      background-color: #79dc09;
      color: #fff;
      border-radius: 4px;
    }
    .job-list {
      margin: 20px;
    }
    .job {
      margin-bottom: 20px;
      padding: 10px;
      border: 1px solid #ccc;
      border-radius: 4px;
    }
    .job h3 {
      margin: 0;
    }
    .job p {
      margin-top: 5px;
    }
    .apply-button {
      display: inline-block;
      padding: 8px 16px;
      background-color: #79dc09;
      color: #fff;
      border: none;
```

```
      border-radius: 4px;
      text-decoration: none;
    }
  </style>
</head>
<body background="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\bg1.jpg" background-size="cover">
  <header class="header">
    <div class="logo">
      <img src="C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\Logo.png" width="100px" alt="Company Logo">
    </div>
    <nav class="nav-buttons">
       <a href="index.html">Home</a>
       <a href="apply_job.html">Apply Job</a>
      </nav>
  </header>

  <h1>Available Jobs</h1>
  <div class="job-list">
    <div class="job">
      <h3>AI Developer</h3>
      <p>“Artificial intelligence developers build AI functionality into software applications,” according to The Enterprisers Project. “The role is generally focused on integrating and implementing AI algorithms and logic into the deliverables of an IT project.”</p>
      <a href="form.html" class="apply-button">Apply</a>
    </div>
    <div class="job">
      <h3>Java Developer</h3>
      <p> Responsible for developing applications and software using the Java programming language, these specialized programmers collaborate with fellow software engineers, web developers, and others to use Java for developing business applications, websites and more.</p>
      <a href="form.html" class="apply-button">Apply</a>
    </div>
    <div class="job">
      <h3>Web Developer </h3>
      <p>to design, create and maintain websites, providing in the process a cohesive and user-friendly online portal for the use of clients, customers, work colleagues and other interested parties.</p>
      <a href="form.html" class="apply-button">Apply</a>
    </div>
  </div>


</body>
</html>
```

app.py:

```
from flask import Flask, render_template, request
from werkzeug.utils import secure_filename
import smtplib
from pyresparser import ResumeParser

# creates SMTP session

s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()

SUBJECT = "Interview Call"

python_skills = ["ml", "ai", "matplotlib", "seabon",
                 "python", "reression", "algorithms"
                 "Pandas", "data analysis", "keras",
                 "tensorflow", "artificial intelligence",
                 "data visualization" "openov"]
java_skills = []

app = Flask(__name__)

@app.route('/')
def homepage():
    return render_template('index. html')
@app.route('/apply_job')
def applyjob():
    return render_template("apply_job.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':
        f = request.files['file']
        f.save(secure_filename(f.filename))
        data = ResumeParser(f.filename).get_extracted_data()
        name = data['name']
        email = data['email']
        skills = data["skills"]
        actual_skills = [i.lower() for i in skills ]
        # using list comprehension
        # # checking if string contains list element
        Skills_matched = [ele for ele in actual_skills if(ele in python_skills)]
        if(len(Skills_matched) >= 4 ):
```

```
            print("he is eligible")
            s.login("jobmatch28@gmail.com", "jobmatch1234")
            TEXT = "Hello "+name + ",\n\n"+ """Thanks for applying to the
            job post AI/ML; Developer, Your skils matches our requirement.
            Kindly Let us know the available time for initial round of interview.
            \n\n Thanks and Regards, \n \n Talent acquistition Team, \n Jobmatch"""
            message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
            s.sendmail("jobmatch28@gmail.com", email, message)
            s.quit()
            return render_template('form.html',prediction =
                                """Thanks for applying you will be mailed about
                                your candidature""")
        else:
            print("sorry we cant process your candidature")
            s.login("jobmatch28@gmail.com","jobmatch1234")
            TEXT = "Hello "+name + ", \n\n"+ """Thanks for applying to the job post AI/ML
            Developer , Your candidature is rejected.
            \n\n\n\n Thanks and Regards, \n\n Talent acquistition Team, \n\n Jobmatch"""
            message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
            s.sendmail("jobmatch28@gmail.com", email, message)
            s.quit()
            return render_template('form.html',prediction =
                                """Thanks for applying you will be
                                mailed about your candidature""")

    else:
        return render_template ('index.html')

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

resume_parser.py:

```
import nltk
nltk.download('stopwords')

# Library to have connection to your gmail
import smtplib
# library to extract the entities form your resume
from pyresparser import ResumeParser


import os
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
from google.oauth2 import credentials
from google_auth_oauthlib.flow import InstalledAppFlow
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError


creds = None
SCOPES = ['https://www.googleapis.com/auth/gmail.send']

# The file token.json stores the user's access and refresh tokens, and is
# created automatically when the authorization flow completes for the first
# time.
if os.path.exists('token.json'):
    creds = credentials.Credentials.from_authorized_user_file('token.json', SCOPES)

# If there are no (valid) credentials available, let the user log in.
if not creds or not creds.valid:
    if creds and creds.expired and creds.refresh_token:
        creds.refresh(Request())
    else:
        SCOPES = ['https://www.googleapis.com/auth/gmail.send']
        flow = InstalledAppFlow.from_client_secrets_file(r'C:\Users\Chhavi Jain\OneDrive\Desktop\VS
FILES\externship project\client_secret_668185049822-
brj2cgur4dnrbvjuuk5c9grre7av8qf8.apps.googleusercontent.com.json', SCOPES)
        creds = flow.run_local_server(port=0)
    # Save the credentials for the next run
    with open('token.json', 'w') as token:
        token.write(creds.to_json())



#creates SMTP session
```

```
s = smtplib.SMTP('smtp.gmail.com', 587)
# start TLS for security
s.starttls ()

# Authentication
#s.login("jobmatch28@gmail.com","jobmatch1234")

# give a subject
SUBJECT = "Interview Call"
# skills requirement for Ai developer
python_skills = ["mL","ai","python","matplotlib","reression","algorithms","seabon","pandas","data
analysis","keras","tensorflow", "artificial intelligence","data visualization","opencv"]
# skills requirement for Java developer
java_skills = []



# extract the skills from resume
data = ResumeParser(r'C:\Users\Chhavi Jain\OneDrive\Desktop\VS FILES\externship project\sample
resume2.pdf').get_extracted_data()
print (data)
# grab the name
name = data ['name']
# grab the Email
email = data['email']
# grab the Skills
skills = data["skills"]
# lowercase the skills
actual_skills = [i.lower() for i in skills ]



# using list comprehension
# checking if string contains list element
Skills_matched = [ele for ele in actual_skills
            if(ele in python_skills)]



# check the number of skills matched
if (len(Skills_matched) >= 4 ):

    print("he is eligible")
    # create a text that is to sent in an email
    TEXT = "Hello "+name + ",\n\n"+"Thanks for applying to the job post AI/ML Developer ."+"Your skils matches
our requirement. Kindly let us "+"know the available time for initial round of interview. "+"\n\n\n\n Thanks and
Regards, "+"\n\n Talent acquisition Team, \n\n Jobmatch"
    # send mail
    message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
    # send the mail
    s.sendmail("jobmatch28@gmail.com",email, message)
    # quit the session
    s.quit()
else:
    print("sorry we cant process your candidature")
```

```
    TEXT = "Hello "+name + ",\n\n"+"Thanks for applying to the job post AI/ML Developer, "+"Your candidature is rejected. "+"\n\n\n\n Thanks and Regards, "+"\n\n Talent acquisition Team, \n\n Jobmatch"
    message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)
    s.sendmail("jobmatch28@gmail.com", email, message)
    s.quit()
```