**SOURCE CODE:**

import os

import nltk

import pandas as pd from nltk.corpus import stopwords from nltk.tokenize import word\_tokenize

# Configure paths

folder\_path = "D:/Saketh/Python Programming/sampleprograms/NLP/NLPLab"

files = [f for f in os.listdir(folder\_path) if f.endswith('.txt')]

# Download necessary NLTK resources nltk.download('punkt', quiet=True) nltk.download('stopwords', quiet=True)

stop\_words = set(stopwords.words('english'))

keywords = ["python", "java", "javascript", "developer", "engineer"] # Initialize lists to store shortlisted and non-shortlisted data shortlisted\_data = []

non\_shortlisted\_data = []

# Function to extract resume information def extract\_info(lines):

info = {

"Name": "",

"Position Applied": "",

"Skills": "",

"Experience": "",

"Contact": ""

}

for line in lines:

if line.startswith('Name: '):

info["Name"] = line.replace("Name: ", "").strip() elif line.startswith('Position Applied: '):

info["Position Applied"] = line.replace("Position Applied: ", "").strip() elif line.startswith('Skills: '):

info["Skills"] = line.replace("Skills: ", "").strip() elif line.startswith('Experience: '):

info["Experience"] = line.replace("Experience: ", "").strip() elif line.startswith('Contact: '): info["Contact"] = line.replace("Contact: ", "").strip()

# Convert skills string into a list

skills = info["Skills"].split(',') if info["Skills"] else [] info["Skills"] = [skill.strip() for skill in skills]

return info # Iterate over all files for file in files:

file\_path = os.path.join(folder\_path, file) with open(file\_path, 'r') as f:

content = f.read()

lines = content.split('\n') resume\_info = extract\_info(lines)

resume\_content = ' '.join([line for line in lines if not line.startswith("Name: ") and not line.startswith("Contact: ")]) tokens = word\_tokenize(resume\_content.lower())

filtered\_tokens = [word for word in tokens if word not in stop\_words] matches = [word for word in keywords if word in filtered\_tokens]

# Check if the resume is shortlisted if len(matches) >= 2:

resume\_info["Shortlisted"] = "Yes" shortlisted\_data.append(resume\_info) else:

resume\_info["Shortlisted"] = "No"

non\_shortlisted\_data.append(resume\_info)

df\_shortlisted = pd.DataFrame(shortlisted\_data)

df\_non\_shortlisted = pd.DataFrame(non\_shortlisted\_data)

file = "shortlisted\_candidates.xlsx" with pd.ExcelWriter(output\_file, engine='xlsxwriter') as writer:

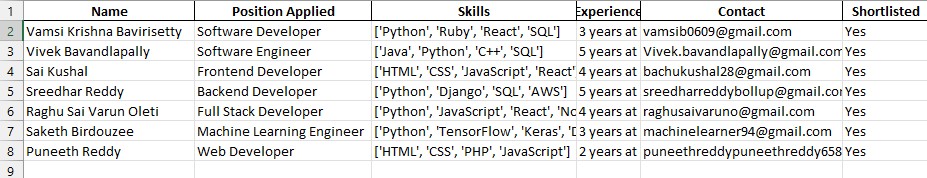
df\_shortlisted.to\_excel(writer, sheet\_name="Shortlisted", index=False) df\_non\_shortlisted.to\_excel(writer, sheet\_name="Non-Shortlisted", index=False)

print(f"Data has been saved to {output\_file} with two sheets: 'Shortlisted' and 'Non-Shortlisted'")

**OUTPUT:**

Data has been saved to shortlisted\_candidates.xlsx with two sheets: 'Shortlisted' and 'Non-Shortlisted'

**SHEET 1: (Shortlisted)**



**SHEET 2: (Non-Shortlisted)**

