the user to import a resume file (in plain text format or .docx format), extracts the key information, and generates a customized cover letter based on that resume. The script uses the tkinter library for the GUI and python-docx for handling .docx files

In [2]:
```
import tkinter as tk
from tkinter import filedialog, messagebox
from docx import Document

def extract_resume(file_path):
    """Extract text from a plain text or Word document."""
    try:
        if file_path.endswith(".docx"):
            doc = Document(file_path)
            return "\n".join([para.text for para in doc.paragraphs if para.text.strip()])
        elif file_path.endswith(".txt"):
            with open(file_path, 'r', encoding='utf-8') as file:
                return file.read()
        else:
            raise ValueError("Unsupported file format. Use .txt or .docx files.")
    except Exception as e:
        messagebox.showerror("File Error", f"Error reading file: {e}")
        return ""

def parse_resume(resume_text):
    """Dummy parser to extract name, skills, and job role from resume text."""
    # Simplistic parsing logic (improve with NLP for better results)
    lines = resume_text.split("\n")
    name = lines[0] if lines else "Your Name"
    skills = "Python, teamwork, problem-solving"  # Replace with actual parsing logic
    return name, skills

def import_resume():
    """Import a resume and populate fields based on its content."""
    file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt"), ("Word Documents", "*.docx")])
    if not file_path:
        return  # User canceled the file dialog

    resume_text = extract_resume(file_path)
    if resume_text:
        name, skills = parse_resume(resume_text)
        name_entry.delete(0, tk.END)
        name_entry.insert(0, name)
        skills_entry.delete(0, tk.END)
```

```
        skills_entry.insert(        0, skills)

def generate_cover_letter():
    """Generates a simple cover letter based on input fields."""
    name = name_entry.get()
    job_title = job_title_entry.get()
    company = company_entry.get()
    skills = skills_entry.get()

    if not all([name, job_title, company, skills]):
        messagebox.showwarning("Missing Information", "Please fill out all fields!")
        return

    cover_letter = (
        f"Dear Hiring Manager,\n\n"
        f"I am excited to apply for the {job_title} position at {company}. "
        f"As a professional with a passion for excellence and expertise in {skills}, "
        f"I am eager to contribute my skills and enthusiasm to your team.\n\n"
        f"I would welcome the opportunity to discuss how my background aligns with your needs. "
        f"Thank you for considering my application.\n\n"
        f"Best regards,\n{name}"
    )
    result_text.delete(1.0, tk.END)
    result_text.insert(tk.END, cover_letter)

# GUI Setup
root = tk.Tk()
root.title("Automated Cover Letter Maker")

# Input Fields
tk.Label(root, text="Your Name:").grid(row=0, column=0, padx=10, pady=5, sticky=tk.W)
name_entry = tk.Entry(root, width=40)
name_entry.grid(row=0, column=1, padx=10, pady=5)

tk.Label(root, text="Job Title:").grid(row=1, column=0, padx=10, pady=5, sticky=tk.W)
job_title_entry = tk.Entry(root, width=40)
job_title_entry.grid(row=1, column=1, padx=10, pady=5)

tk.Label(root, text="Company Name:").grid(row=2, column=0, padx=10, pady=5, sticky=tk.W)
company_entry = tk.Entry(root, width=40)
company_entry.grid(row=2, column=1, padx=10, pady=5)

tk.Label(root, text="Skills (comma-separated):").grid(row=3, column=0, padx=10, pady=5, sticky=tk.W)
skills_entry = tk.Entry(root, width=40)
```

```
skills_entry.grid(row=3, column=1, padx=10, pady=5)

# Buttons
generate_button = tk.Button(root, text="Generate Cover Letter", command=generate_cover_letter)
generate_button.grid(row=4, column=0, padx=10, pady=10, sticky=tk.W)

import_button = tk.Button(root, text="Import Resume", command=import_resume)
import_button.grid(row=4, column=1, padx=10, pady=10, sticky=tk.E)

# Output Area
tk.Label(root, text="Generated Cover Letter:").grid(row=5, column=0, padx=10, pady=5, sticky=tk.W)
result_text = tk.Text(root, width=60, height=15)
result_text.grid(row=6, column=0, columnspan=2, padx=10, pady=5)

# Start the application
root.mainloop()
```

How It Works:

Resume Import:

The "Import Resume" button allows the user to upload a resume file in .txt or .docx format. The script extracts the content and fills the Name and Skills fields based on the resume. Cover Letter Generation:

The "Generate Cover Letter" button creates a cover letter using the input fields. Output Area:

The cover letter appears in the text area at the bottom of the GUI.

In [ ]: