



AI RESUME & COVER LETTER GENERATOR

Presented By: **SWAPNIL KESHARWANI**

Institution: **UNITED COLLEGE of ENGINEERING & RESEARCH, NAINI, PRAYAGRAJ**

University Roll No.: **2300101530170**



INTRODUCTION

- Resumes and cover letters are essential tools in job applications.
- Manual creation is time-consuming and often lacks personalization.
- AI offers a smart solution by generating role-specific and skill-based content quickly.



OBJECTIVE

- To develop a web-based tool that uses AI to automatically generate professional resumes and cover letters.
- Allow users to input details like skills, job role, and experience.

TECHNOLOGY USED

Component	Technology / Library	Purpose
🧠 Language Model	transformers >= 4.28	Generate text using pre-trained models (e.g., GPT-2, Falcon)
🔧 Deep Learning	torch >= 1.9	Backend computation for LLM inference
🧱 Tokenizer	tokenizers >= 0.13.3, sentencepiece >= 0.1.95	Efficient tokenization for model input/output
📦 Serialization	protobuf >= 3.20.0	Handles serialized model data (required by some transformer models)
🌐 Web UI	Gradio	Easy-to-use frontend for users to interact with the AI
📁 (Optional) PDF Export	reportlab / weasyprint	Convert generated text into downloadable resumes/letters



FEATURES

- Role-specific resume generation.
- Professional formatting.
- Cover letter customization.
- Export as PDF/DOCX.



WORKING/METHODOLOGY

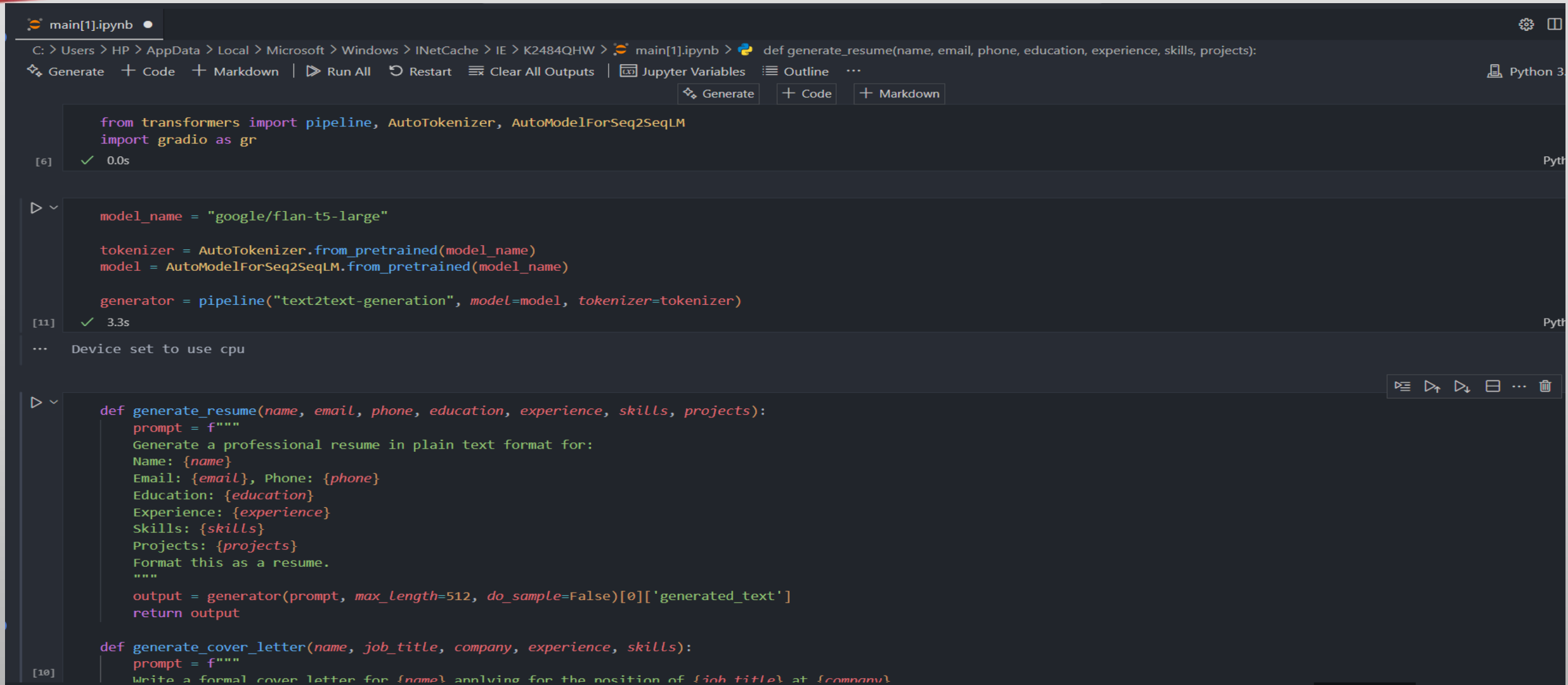
- User selects job role.
- Inputs skills, experience, goals.
- AI processes prompt and generates text.
- Preview & download options provided.



SCOPE OF THE PROJECT

- Web app that collects user input and generates content.
- Uses LLMs for personalized results.
- Output can be previewed and downloaded in PDF or DOCX.

UI SCREENSHOTS/DEMO



The screenshot displays a Jupyter Notebook interface with a dark theme. The top bar shows the file name 'main[1].ipynb' and a path: 'C: > Users > HP > AppData > Local > Microsoft > Windows > INetCache > IE > K2484QHW > main[1].ipynb'. Below the path, there are buttons for 'Generate', 'Code', 'Markdown', 'Run All', 'Restart', 'Clear All Outputs', 'Jupyter Variables', 'Outline', and a menu icon. The notebook contains three code cells. The first cell (index 6) imports 'pipeline', 'AutoTokenizer', and 'AutoModelForSeq2SeqLM' from 'transformers', and 'gradio' as 'gr'. It runs successfully in 0.0s. The second cell (index 11) initializes the model with 'google/flan-t5-large', creates a tokenizer, and sets up a text-to-text generation pipeline. It runs successfully in 3.3s. The third cell (index 10) defines two functions: 'generate_resume' and 'generate_cover_letter'. The 'generate_resume' function takes parameters for name, email, phone, education, experience, skills, and projects, constructs a prompt, and uses the pipeline to generate a resume. The 'generate_cover_letter' function takes parameters for name, job_title, company, and experience, and constructs a prompt for a cover letter. The notebook interface includes a left sidebar with a file explorer, a top right corner with settings and window icons, and a bottom right corner with a toolbar for cell navigation.

```
main[1].ipynb

C: > Users > HP > AppData > Local > Microsoft > Windows > INetCache > IE > K2484QHW > main[1].ipynb

Generate + Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ... Python 3

[6] ✓ 0.0s

model_name = "google/flan-t5-large"

tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

generator = pipeline("text2text-generation", model=model, tokenizer=tokenizer)

[11] ✓ 3.3s

... Device set to use cpu

[10] def generate_resume(name, email, phone, education, experience, skills, projects):
    prompt = f"""
    Generate a professional resume in plain text format for:
    Name: {name}
    Email: {email}, Phone: {phone}
    Education: {education}
    Experience: {experience}
    Skills: {skills}
    Projects: {projects}
    Format this as a resume.
    """
    output = generator(prompt, max_length=512, do_sample=False)[0]['generated_text']
    return output

def generate_cover_letter(name, job_title, company, experience, skills):
    prompt = f"""
    Write a formal cover letter for {name} applying for the position of {job_title} at {company}
    """
```


FileEditSelectionViewGoRunTerminalHelp

Search

Python 3.12.5

main[1].ipynb

C: > Users > HP > AppData > Local > Microsoft > Windows > INetCache > IE > K2484QHW > main[1].ipynb > def generate_resume(name, email, phone, education, experience, skills, projects):

GenerateCodeMarkdownRun AllRestartClear All OutputsJupyter VariablesOutline

```
def generate_resume(name, email, phone, education, experience, skills, projects):
    prompt = f"""
    Generate a professional resume in plain text format for:
    Name: {name}
    Email: {email}, Phone: {phone}
    Education: {education}
    Experience: {experience}
    Skills: {skills}
    Projects: {projects}
    Format this as a resume.
    """

    output = generator(prompt, max_length=512, do_sample=False)[0]['generated_text']
    return output

def generate_cover_letter(name, job_title, company, experience, skills):
    prompt = f"""
    Write a formal cover letter for {name} applying for the position of {job_title} at {company}.
    Highlight relevant experience: {experience}.
    Mention important skills: {skills}.
    """

    output = generator(prompt, max_length=512, do_sample=False)[0]['generated_text']
    return output
```

[10] ✓

Python

Both `max_new_tokens` (=256) and `max_length` (=512) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/glossary#max_new_tokens)

Both `max_new_tokens` (=256) and `max_length` (=512) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/glossary#max_new_tokens)

Both `max_new_tokens` (=256) and `max_length` (=512) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/glossary#max_new_tokens)

Both `max_new_tokens` (=256) and `max_length` (=512) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation for more information. (https://huggingface.co/docs/transformers/main/en/glossary#max_new_tokens)

def generate_documents(name, email, phone, education, experience, skills, projects, job_title, company):

resume = generate_resume(name, email, phone, education, experience, skills, projects)

cover_letter = generate_cover_letter(name, job_title, company, experience, skills)

[8]

0 0 Connect

Spaces: 4 CRLF Signed out Cell 3 of 5 Go Live

FileEditSelectionViewGoRunTerminalHelp

Search

main[1].ipynb

C: > Users > HP > AppData > Local > Microsoft > Windows > INetCache > IE > K2484QHW > main[1].ipynb > def generate_resume(name, email, phone, education, experience, skills, projects):

GenerateCodeMarkdownRun AllRestartClear All OutputsJupyter VariablesOutlinePython 3.12.5

```
def generate_documents(name, email, phone, education, experience, skills, projects, job_title, company):
    resume = generate_resume(name, email, phone, education, experience, skills, projects)
    cover_letter = generate_cover_letter(name, job_title, company, experience, skills)
    return resume, cover_letter

iface = gr.Interface(fn=generate_documents,
    inputs=[
        gr.Textbox(Label="Name"),
        gr.Textbox(Label="Email"),
        gr.Textbox(Label="Phone"),
        gr.Textbox(Label="Education"),
        gr.Textbox(Label="Work Experience"),
        gr.Textbox(Label="Skills"),
        gr.Textbox(Label="Projects"),
        gr.Textbox(Label="Job Title (for Cover Letter)"),
        gr.Textbox(Label="Company (for Cover Letter)"),
    ],
    outputs=[
        gr.Textbox(Label="Generated Resume"),
        gr.Textbox(Label="Generated Cover Letter"),
    ],
    title="AI Resume & Cover Letter Generator",
    description="Powered by Hugging Face Transformers and FLAN-T5"
)

iface.launch()
```

[8] ✓ 1.3s

Python

... * Running on local URL: <http://127.0.0.1:7860>

... * To create a public link, set `share=True` in `launch()`.

Spaces: 4CRLFSigned outCell 3 of 5Go Live

AI Resume & Cover Letter Genera

127.0.0.1:7860

🔍 ☆ ⚙️ 👤 ⋮ 🌐

AI Resume & Cover Letter Generator

Powered by Hugging Face Transformers and FLAN-T5

Name

Email

Phone

Education

Work Experience

Skills

Projects

Job Title (for Cover Letter)

Company (for Cover Letter)

Generated Resume

Generated Cover Letter

Flag

Clear

Submit

Use via API 🚀 · Built with Gradio 🍷 · Settings ⚙️



PROBLEM STATEMENT

- Job seekers struggle with writing effective resumes and cover letters.
- Existing tools are either too generic or complicated.
- Lack of AI-driven personalization limits impact.



CHALLENGES FACED

- Prompt accuracy and output formatting.
- Maintaining context and grammar.
- Handling diverse user input.



SOLUTIONS TO CHALLENGES

- Improved prompt engineering.
- Pre-defined templates and formatting logic.
- Extensive testing for various roles and profiles.



TESTING & RESULTS

- Tested with different user profiles.
- Output quality validated with feedback.
- High user satisfaction and usability.



CONCLUSION

- AI-driven resume & cover letter tool saves time and improves quality.
- Useful for students, professionals, job seekers.




FUTURE SCOPE

- Multilingual support.
- LinkedIn and ATS integration.
- Voice-based input and smart recommendations.



REFERENCES & BIBLIOGRAPHY

- Hugging Face Transformers.
- Gradio / Streamlit Documentation.
- Research on LLMs and resume optimization.



THANK YOU !!!