CAPSTONE PROJECT

Al Resume & Portfolio Builder

Presented By:
Shubham Aher - VIIT Pune - CSE(IOTCSBT)



OUTLINE

- Problem Statement
- System Development
 Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References



Problem Statement

- Core Problem: Many students struggle to effectively present their skills, projects, and experiences in an attractive and professional format suitable for job or internship applications.
- Challenge 1: Generic, one-size-fits-all resume templates often fail to highlight a student's individual strengths and unique accomplishments.
- Challenge 2: Crafting compelling and tailored resumes, cover letters, and portfolio summaries requires significant time and strong writing skills, which can be a barrier for many.
- **The Need:** There is a clear need for a smart, automated solution that can generate personalized and professional application documents based on individual student data, ultimately enhancing their employment opportunities.



System Approach

Frontend: Streamlit

Used to create a simple, interactive, and user-friendly web interface for data input and displaying results.

Backend: Python

 The core logic is written in Python to handle user data, process requests, and integrate with other services.

Al Model Engine: Cerebras API (Llama Maverick Model)

 Leverages the llama-4-maverick-17b-128e-instruct model to generate high-quality, human-like text for resumes, cover letters, and portfolio summaries.

PDF Generation: FPDF Library

 Enables the conversion of the Al-generated text content into universally shareable and printable PDF documents.

• Environment Management: Python-dotenv

 Used for securely managing the API key and other environment variables for both local development and cloud deployment.



Algorithm & Deployment

- User Data Collection: The application starts by collecting the user's personal details, education, skills, experience, and project information through a clean Streamlit form.
- Input Validation: It performs a check to ensure that all required fields are filled before proceeding, preventing errors and unnecessary API calls.
- **Prompt Engineering:** Based on the user's selected document type (Resume, Cover Letter, or Portfolio), a specific, detailed prompt is constructed. This tailored prompt guides the AI to generate the most relevant content.
- Al Content Generation: The constructed prompt is sent to the Cerebras API. The Llama model processes the input and generates a professional, well-structured document.
- **Display and Download:** The generated content is displayed on the screen for the user to review. The user can then click a button to convert the text into a PDF, which is created using the FPDF library and made available for download.
- **Deployment:** The application is deployed on Streamlit Cloud, making it publicly accessible to users through a web browser without requiring any local installation.



Result

```
def construct_resume_prompt(user_data):
    """Construct a prompt for resume generation"""
    prompt = f"""Create a professional resume for the following candidate:

Name: {user_data['name']}
Email: {user_data['email']}
Phone: {user_data['phone']}

Career Objective:
{user_data.get('objective', 'Seeking a challenging position to utilize my skills and contribute to or

Education:
{user_data['education']}

Skills:
{user_data['skills']}

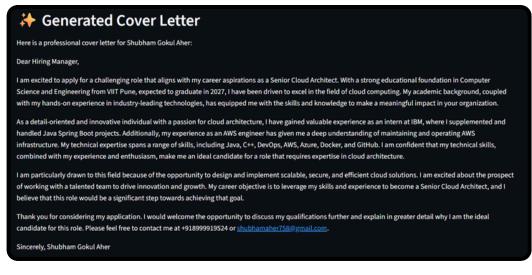
Work Experience:
{user_data['experience']}

Projects:
{user_data.get('projects', 'N/A')}
```

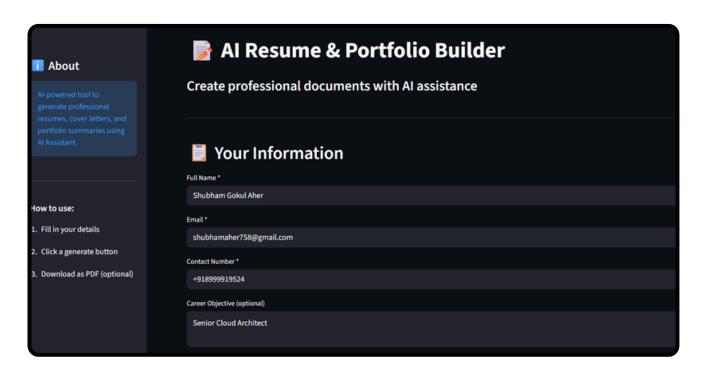
Code



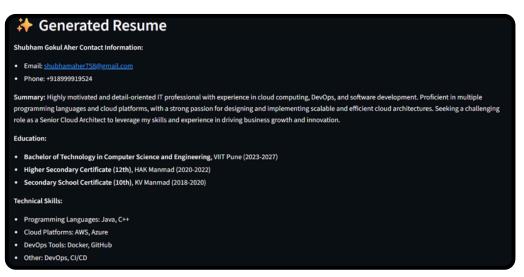
Portfolio



Cover letter



Homepage



Resume



GITHUB AND DEPLOYMNET LINK

• Github Link :

https://github.com/shubhamaher8/Al-Resume-And-Portfolio-Builder

• Deployment link: https://ai-resume-and-portfolio-build.streamlit.app



Conclusion

- The Al Resume & Portfolio Builder successfully addresses the challenge students face in creating professional application documents.
- By integrating a powerful generative AI with a simple web interface, the project provides an automated and effective solution for generating tailored resumes, cover letters, and portfolios.
- The system empowers students to better showcase their skills and experiences, directly contributing to improving their job and internship prospects with high-quality, personalized materials.



Future scope

- Implement Rich Text Editor: Allow users to manually edit and format the Al-generated content directly in the application before downloading the PDF.
- Add Customizable Templates: Introduce multiple professional templates (e.g., modern, classic, creative) that users can choose from to format their final documents.
- Incorporate Keyword Optimization: Add a feature to suggest relevant keywords for a specific job description to make resumes more ATS-friendly.
- **User Accounts and History**: Enable users to create accounts to save their information and access previously generated documents.



References

Title: Al-Based Resume Builder

Source: International Journal of Innovative Research in Technology (IJIRT)

Link: https://ijirt.org/Article?manuscript=170083

Relevance: This paper directly discusses the development of an Al-based resume builder,

making it highly relevant to your project's core concept.

Title: Streamlit Documentation

Source: Streamlit.io

Link: https://streamlit.io/

Relevance: As a primary technology in your project's frontend, citing the official

documentation is standard practice.



THANK YOU

