Microsoft Elevate

CAPSTONE PROJECT

AI Resume & Career Kit Generator Using Generative AI

PRESENTED BY

STUDENT NAME: PREM KUMAR SINGH

COLLEGE NAME: TKR COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT: INFORMATION TECHNOLOGY

EMAIL ID: PREMKUMARSINGH355@GMAIL.COM

Microsoft Elevate

OUTLINE:

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

Microsoft Elevate

PROBLEM STATEMENT:

Creating professional resumes and cover letters is a time-consuming and skill-dependent task.

Students and fresh graduates often face the following challenges:

- Lack of experience in professional resume writing
- Limited understanding of ATS (Applicant Tracking System) optimization
- Difficulty tailoring resumes to specific job descriptions
- Inability to identify missing keywords required by recruiters

There is a need for an AI-driven system that can automatically generate professional, ATS-optimized resumes, analyze job descriptions, and provide improvement suggestions.

Microsoft Elevate

PROPOSED SOLUTION:

The proposed system is an **AI-powered Resume & Career Kit Generator** built using Streamlit and Groq LLaMA 3.3.

- **Workflow**
  - User inputs: Personal Details, Technical Details and job details
  - Prompt is structured using prompt engineering.
  - LLM generates:
    - Professional Summary
    - Resume
    - Cover Letter
  - Resume Evaluation (Rule-Based NLP):
  - ATS Analyzer:
    - Compares resume vs job description
    - Detects missing keywords
    - Provides match score

The application uses a Large Language Model (LLM) to generate context-aware, role-specific resume content.

SYSTEM APPROACH:

- **Frontend**
  - Streamlit (Python-based UI framework)
  - Custom CSS (Dark/Light theme)
  - Responsive multi-tab interface
- **Backend**
  - Groq LLaMA 3.3 70B model
  - Prompt engineering for structured output
  - Python-based logic modules (utils.py)
- **Libraries Used**
  - streamlit
  - dotenv
  - fpdf2 (PDF generation)
  - Groq SDK

ALGORITHM & DEPLOYMENT:

Algorithm Used

1 Transformer-Based Large Language Model (LLaMA 3.3)

- Based on self-attention mechanism (Vaswani et al., 2017)
- Pretrained on large-scale text corpora
- Performs Natural Language Generation (NLG)

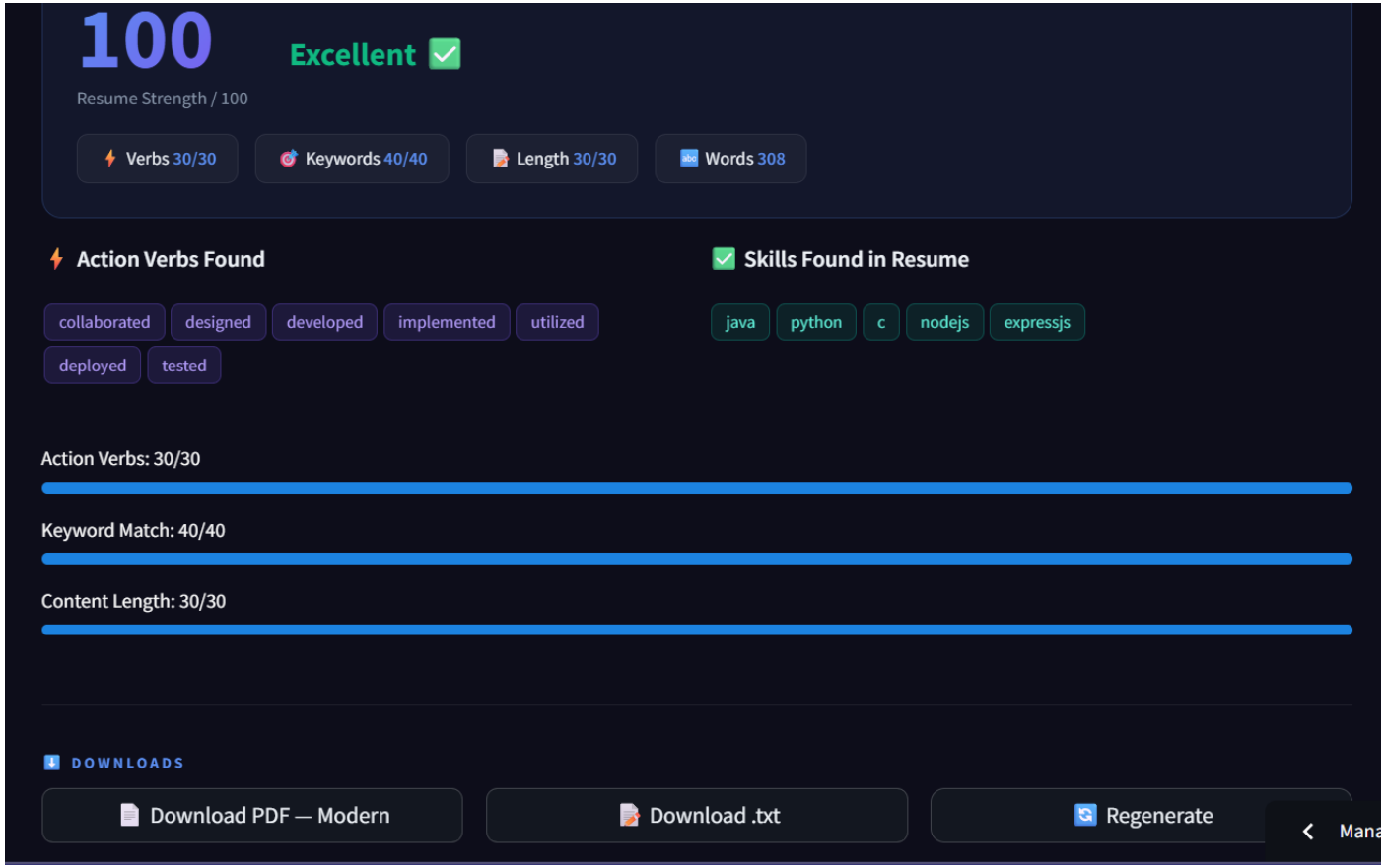
Deployment

- Developed using Streamlit
- Deployed on Streamlit Cloud
- API secured using environment variables
- Runs entirely on web interface

GitHub Link: https://github.com/Prem-080/AI-Resume-Builder
Deployment Link: https://ai-resume-build.streamlit.app/

RESULT:

The system successfully generated:
ATS-optimized resume
Personalized cover letter
LinkedIn bio
Resume strength score: 100/100
Keyword match: 40/40
The interface provides real-time visual feedback and downloadable documents.



Microsoft Elevate

CONCLUSION:

- The AI Resume & Career Generator demonstrates how Generative AI can be applied to solve real-world career preparation problems.
- Key achievements:
- Automated resume writing
- Improved ATS compatibility
- Transparent scoring mechanism
- Job description alignment
- Professional UI experience
- The system reduces resume preparation time significantly while improving quality and customization.

Microsoft Elevate

FUTURE SCOPE:

- Fine-tuning LLaMA on HR-approved resume datasets
- Replace rule-based scoring with BERT-based semantic similarity
- Multi-language resume generation
- User authentication & resume storage
- Integration with LinkedIn API
- AI-powered interview question generator

REFERENCES:

- Vaswani et al., 2017 – *Attention Is All You Need*
- Brown et al., 2020 – *Language Models are Few-Shot Learners*
- Meta AI – LLaMA 3 Documentation
- Groq API Documentation
- Streamlit Official Documentation

Microsoft Elevate

Thank You