

Data Collection and Preprocessing Phase

Date	12 July 2024
Team ID	SWTID1720161415
Project Title	JobSwift : Accelerating Careers with AI Powered Applications
Maximum Marks	6 Marks

Preprocessing Template

Section	Description
User Input Collection	The app collects various user inputs such as name, experience, skills, projects, education, awards, LinkedIn profile, GitHub profile, company name, and job title through Streamlit's interactive widgets
Validation	Before generating the resume, cover letter, or interview questions, the app checks if the required fields are filled out. This ensures that the prompts sent to the PaLM API are complete and coherent
Prompt Construction:	The collected inputs are used to construct prompts for the PaLM API. This involves formatting the inputs into a structured text that the PaLM model can understand and generate appropriate responses for.
API Interaction:	The constructed prompts are then sent to the PaLM API, which processes the text and generates the desired outputs (e.g., resumes, cover letters, interview questions).

Data Preprocessing Code Screenshots

Collecting User Inputs	<pre> name = st.text_input("Enter Your Name") experience = st.text_input("Enter Your Experience in Years") skills = st.text_area("Enter Your Skills") projects = st.text_area("Enter Your Projects") education = st.text_area("Enter Your Education") awards = st.text_area("Enter Your Awards and Recognition") linkedin=st.text_input("Enter LinkedIn profile(Optional)") github=st.text_input("Enter Github profile(Optional)") </pre>
-------------------------------	---

	<pre>def generate_cover_letter(company_name, job_title): company_name = st.text_input("Enter Company Name") job_title = st.text_input("Enter Job Title") skills = st.text_area("Enter your skills")</pre>
Validating User Inputs:	<pre>if st.button("Generate Resume"): if name and experience and skills and projects and education and awards: st.write(generate_resume(name, experience, skills, projects, education, awards, linkedin, github)) else: st.error("Please fill all the fields") if st.button("Generate Interview Questions"): if skills: st.write(generate_interview_questions(skills)) else: st.error("Please fill all the fields")</pre>
Constructing Prompts for the API	<pre>def generate_resume(name, experience, skills, projects, education, awards, linkedin=None, github=None): prompt = f"My name is {name}." prompt += f"Experience:\n{experience}\n\n" prompt += "\n\nCareer Objective:\nProvide a brief career objective based on the inputs.\n\n" prompt += f"Skills:\n{skills}\n\n" prompt += "Projects:\nProvide detailed descriptions of the following projects:\n" for project in projects: prompt += f"- {project}\n" prompt += f"\nEducation:\n{education}\n\nAwards and Recognition:\n{awards}" if linkedin or github: prompt += "\n\nLinks:\n" if linkedin: prompt += f"LinkedIn: {linkedin}\n" if github: prompt += f"GitHub: {github}\n" response = palm.generate_text(model=model_name, prompt=prompt, temperature=0) return response.result</pre>
API Interaction:	<pre>def generate_cover_letter(company_name, job_title): prompt = f"I am interested in the {job_title} position at {company_name}." response = palm.generate_text(model=model_name, prompt=prompt, temperature=0) return response.result</pre>