**Summarized version for the the cv   
Car Clinic – Smart Repair Advisor**  
*AI-powered Recommendation & Chatbot System | Python, FastAPI, Prefect, llama-cpp, Sentence-BERT*

Developed a modular, offline-capable AI pipeline that scrapes Reddit automotive data, cleans it with LLMs, generates semantic tags and embeddings, and recommends the most relevant repair branch based on problem similarity and location. Final output powers an emergency chatbot for real-time assistance to users and mechanics. Integrated with Prefect for orchestration and GitHub Actions for CI/CD automation.

Questions to answer for a better project description:  
**🔍 Info I Need from You:**

1. **Scale**:  
    • How many Reddit posts/comments does your pipeline process daily (or in total)?  
    • How many branches or expertise categories are supported?
2. **LLM Accuracy**:  
    • Have you benchmarked the LLM cleaning accuracy or tag quality (e.g., % valid outputs, hallucination rate)?
3. **Recommendation System**:  
    • Any numbers for match accuracy (e.g., top-1 accuracy or average cosine similarity of top matches)?  
    • Any anecdotal validation with mechanics or internal testers?
4. **Performance**:  
    • How fast is the pipeline (e.g., minutes per 100 posts cleaned)?  
    • Is it fully autonomous (zero manual intervention daily)?
5. **Tech stack**:  
    • Is it deployed anywhere currently (e.g., on a private server, company dev environment)?  
    • Is it running via GitHub Actions or triggered locally?

Once I get your answers, I’ll create a **short, CV-ready bullet point** that includes:

* Clear impact
* Scalable architecture
* LLM/ML relevance
* Metrics + keywords for ATS visibility

Initial version 2:  
**📄 How to Write it in Your CV:**

**Option 1 – Concise (For bullet lists)**

**Smart Repair Advisor – Car Clinic (LLM-Powered Recommendation System)**  
*Python, FastAPI, Prefect, PRAW, llama-cpp, Sentence-BERT, GitHub Actions*

* Developed an end-to-end AI system that extracts, cleans, and processes car repair issues from Reddit using offline LLMs.
* Built semantic tagging and embedding pipelines to match problems with branch expertise and recommend optimal repair centers.
* Designed a real-time **Emergency Repair Assistant chatbot** for customers and mechanics using RAG + recommender logic.
* Orchestrated the modular data pipeline using Prefect and automated CI/CD with GitHub Actions.
* Delivered complete documentation, unit-tested modules, and deployment-ready FastAPI interfaces.

Initial version 3:  
**Option 2 – Extended (For project section)**

**Smart Repair Advisor – Car Clinic | AI-Powered Automotive Assistant**  
*Technologies: Python, FastAPI, Prefect, Sentence-BERT, llama-cpp, GitHub Actions, Docker*  
A modular AI system built to autonomously extract and understand car-related issues from Reddit and recommend optimal Car Clinic repair branches based on semantic matching, tags, and proximity.

* Scraped and cleaned Reddit data using offline LLMs to structure (problem → solution) pairs.
* Applied data augmentation, keyword tagging, and embedding generation for high recall and understanding.
* Created a hybrid recommender system combining cosine similarity + tag overlap to rank nearby branches.
* Designed and deployed a real-time **Emergency LLM Chatbot** to assist users and mechanics with instant guidance.
* Fully automated the pipeline with Prefect and GitHub Actions and exposed services via a FastAPI backend.