**ResumeSync – Version 1 Review Document**

**Student:** Shamim Shakil

**Course:** Python Application Development

**Instructor:** Chris Harding

**Date:** October 2025

**1. Project Overview**

**ResumeSync** is an AI-powered résumé–job matching tool that helps job seekers evaluate how well their résumés align with job postings.

Version 1 delivers a **fully functional command-line prototype** with the following features:

* Job search and scraping from **LinkedIn** via *JobSpy*
* Resume upload and PDF text extraction via *PyPDF2*
* AI-based résumé–JD comparison using *OpenAI API*
* Output of strengths, weaknesses, missing skills, and an optional match score

**2. Implemented Components**

* **Core CLI workflow:** job → select → upload → analyze → result (implemented)
* **JobSpy integration:** fetches 5 LinkedIn postings per query (implemented)
* **OpenAI analysis:** compares resume text and JD keywords (implemented)
* **PDF parser:** extracts text (cleaned automatically)
* **Error handling & prompt tuning:** partially done
* **Streamlit GUI:** Will be implemented on V2

**3. Demonstration Summary**

**Current CLI flow**

1. User enters a role (e.g., “UX Designer”) + location (“United States”).
2. Top 5 LinkedIn jobs appear with titles + companies.
3. User picks a job number to view details.
4. System fetches full JD and offers to upload a resume (PDF path).
5. CLI shows results such as:

**4. Issues & Solutions**

| **Issue** | **Resolution / Plan** |
| --- | --- |
| JobSpy not scraping Indeed/Glassdoor/ZipRecruiter (login-restricted) | Aborted – LinkedIn only |
| Private or login-gated LinkedIn URLs | Handled by filtering for public URLs only |
| OpenAI API cost + error handling | Added basic retry logic and prompt simplification |
| Resume preview in CLI too cluttered | Removed preview; auto-parses in background |
| GUI delay (Streamlit) | Deferred to Version 2 milestone |

**5. Outstanding Work & Help Needed**

* **Prompt tuning** for more balanced skill matching
* **Improved output formatting** for human-readable summaries
* **Error logging & rate-limit handling** for OpenAI calls

**6.Milestones for Version 2 (Next 3–4 Weeks)**

| **Milestone** | **Description** |
| --- | --- |
| **M1 – Streamlit UI Prototype** | Build drag-and-drop interface (simple form + text area) |
| **M2 – Enhanced OpenAI Prompt** | Add contextual scoring and clearer skill categorization |
| **M3 – Export PDF Reports** | Enable “Download Analysis as PDF” |
| **M4 – History & Session Memory** | Store last 5 analyses locally for quick comparison |
| **M5 – Testing & Documentation** | Write final README, usage guide, and demo video |

**7. Self-Reflection**

* **Progress Satisfaction:** 8 / 10 – Core pipeline works end-to-end.
* **Feasibility:** Project can be finished within 4 weeks with focused GUI work.
* **Easier than expected:** JobSpy integration and CLI structure.
* **Harder than expected:** Handling private URLs. (Indeed, Glassdoor, Zip recruiter)
* **Light-bulb moment:** Realizing that a simple CLI MVP with clean AI logic is more impactful than a half-working GUI.

**8. Summary**

Version 1 successfully delivers a usable CLI prototype for resume job analysis using AI.

It lays the foundation for Version 2, which will transform the CLI logic into a Streamlit interface with visual dashboards and exportable results.