# 📄 CoverCraft Pro – AI-Powered Resume & JD Analyzer

## 🎯 Project Objectives

A user uploads:

* - A resume (PDF or DOCX)
* - A job description (JD) (PDF, DOCX, or pasted text)

Your backend will:

1. Summarize both documents

2. Extract skills, tools, roles from both

3. Score similarity/match between resume and JD

4. Generate a tailored cover letter

5. Suggest improved resume bullet points

## 🧱 Backend Features & Stack

* - PDF & DOCX Parsing: PyMuPDF, python-docx, pdfminer.six
* - Summarization: OpenAI GPT-4 or Claude 3 via API
* - NER: spaCy + custom rules or HuggingFace transformers
* - Embedding for Match Score: SentenceTransformers (SBERT) + cosine\_similarity
* - Bullet Rewrite & Cover Letter Gen: GPT-4, Prompt Engineering
* - Backend Framework: FastAPI (Python) or Express.js (Node)
* - File Storage: Local disk for MVP, or AWS S3 for production
* - Optional: ChromaDB or Pinecone for vector memory (future)

## 🛠️ Phase-by-Phase Roadmap

Phase 1 – Setup & Parsing

* - Set up backend (FastAPI)
* - Add endpoint: /upload
* - Parse PDF/DOCX, clean extracted text

Phase 2 – Summarization (Gen AI)

* - Summarize resume and JD using GPT-4 or Claude
* - Use prompt-based summarization

Phase 3 – Skills & Keywords Extraction (NLP)

* - Use spaCy or HuggingFace for NER
* - Extract hard/soft skills, tools

Phase 4 – Match Score Calculation

* - Use SentenceTransformers + cosine similarity
* - Return score with explanation

Phase 5 – Cover Letter Generation (Gen AI)

* - Prompt GPT with resume + JD summary
* - Generate a professional 3-paragraph letter

Phase 6 – Resume Bullet Improvement

* - Extract bullet points
* - Prompt GPT to rewrite each bullet

Phase 7 – Connect with Frontend

* - API endpoints: /upload, /summary, /skills, /match-score, /cover-letter, /bullet-rewrite

Phase 8 – Polish & Production Prep

* - Add error handling, token limits, deploy backend

## 📚 What You Need to Learn

NLP & AI:

* - PDF Parsing: PyMuPDF, pdfminer, python-docx
* - Summarization & Generation: OpenAI Docs
* - SentenceTransformers: sbert.net
* - spaCy NER: spacy.io
* - HuggingFace Transformers (optional)

Backend:

* - FastAPI Basics: fastapi.tiangolo.com
* - File Uploads & Async Handling
* - Rate Limiting: httpx, tenacity, asyncio
* - Cosine Similarity: scikit-learn, numpy

## 🎁 Bonus Feature Ideas

* - User login + saved resumes/JDs
* - Export cover letter as PDF
* - Tone selector for cover letter (casual/formal)
* - Analytics dashboard (match scores over time)