

## Accelerated 3-Week (15-Day) Project Plan - Smart Job Match AI

Time Commitment: 6 hours/day

### Day 1-2: Resume & Job Parser

- Learn file parsing (PDF, DOCX, CSV)
- Use spaCy or regex to extract skill terms

Deliverable: resume\_parser.py, job\_parser.py

### Day 3-4: Skill Dictionary & Tagging

- Build a skill taxonomy (AI, NLP, tools)
- Implement tagging logic

Deliverable: skills\_list.csv, skill tagging module

### Day 5-6: Sentence Embeddings & Similarity

- Use sentence-transformers (SBERT)
- Compute cosine similarity between resume and jobs

Deliverable: embedding\_matcher.py

### Day 7: Ranking Engine

- Rank job matches using score + skill match %

Deliverable: match\_engine.py, match\_results.csv

### Day 8-9: Gap Analyzer & Recommendations

- Identify missing skills

- Map to learning links (Coursera, GitHub)

Deliverable: skill\_gap.py, learning\_plan.md

#### Day 10: Prompt Engineering Module (Optional)

- Use OpenAI API for career summary generation

Deliverable: career\_advisor.py, sample prompts

#### Day 11: Report Generator

- Output markdown, JSON, and CSV reports

Deliverable: generate\_report.py

#### Day 12: CLI Integration

- Build main.py with argparse or click
- Allow full pipeline to run via terminal

Deliverable: main.py

#### Day 13: Code Refactoring & Testing

- Modularize code
- Add unit tests with pytest

Deliverable: tests/, modular repo structure

#### Day 14-15: Final Polish & GitHub Push

- Write README, include usage examples
- Optional: record CLI demo

Deliverable: Final GitHub repo with sample inputs/outputs