**JobFit – Development Diary**

**Day 2: Text Extraction from Uploaded Files**  
**Date:** 22 July 2025

**1. Objective**

Implement text extraction functionality to read resume and job description content from uploaded files in .pdf, .docx, and .txt formats. This is a crucial step before implementing matching logic.

**2. Libraries Installed**

* PyPDF2: for PDF text extraction
* docx2txt: for DOCX text extraction

bash

CopyEdit

pip install PyPDF2 docx2txt

**3. Code Implementation**

**Text Extraction Functions**

python

CopyEdit

import os

import docx2txt

from PyPDF2 import PdfReader

def extract\_text\_from\_docx(path):

return docx2txt.process(path)

def extract\_text\_from\_pdf(path):

reader = PdfReader(path)

text = ""

for page in reader.pages:

page\_text = page.extract\_text()

if page\_text:

text += page\_text

return text

def extract\_text\_from\_txt(path):

with open(path, "r", encoding="utf-8") as f:

return f.read()

def extract\_text(path):

ext = os.path.splitext(path)[1].lower()

if ext == ".docx":

return extract\_text\_from\_docx(path)

elif ext == ".pdf":

return extract\_text\_from\_pdf(path)

elif ext == ".txt":

return extract\_text\_from\_txt(path)

else:

return ""

**4. Integration in Flask Route**

Text extraction performed after file upload, with output preview in console:

python

CopyEdit

@app.route("/", methods=["GET", "POST"])

def home():

if request.method == "POST":

resume = request.files.get("resume")

job = request.files.get("job")

if resume:

resume\_path = os.path.join(UPLOAD\_FOLDER, resume.filename)

resume.save(resume\_path)

resume\_text = extract\_text(resume\_path)

print("Extracted Resume Text (first 500 chars):")

print(resume\_text[:500])

if job:

job\_path = os.path.join(UPLOAD\_FOLDER, job.filename)

job.save(job\_path)

job\_text = extract\_text(job\_path)

print("Extracted Job Description Text (first 500 chars):")

print(job\_text[:500])

return render\_template("index.html")

**5. Testing**

* Uploaded files of each supported format.
* Verified text output correctness and absence of errors.
* Confirmed smooth integration with file upload system.

**6. Current Project Structure**

bash

CopyEdit

jobfit/

│

├── app.py

├── templates/

│ └── index.html

├── static/

│ ├── css/style.css

│ └── js/script.js

└── upload/

**7. Status Summary**

* ✅ File upload and saving working
* ✅ Text extraction from PDF, DOCX, TXT working
* ✅ Console output verified

**8. Next Steps**

* Implement technology keyword matching
* Design keyword list with synonyms and variants
* Create results page to display match score and missing skills