📄 Working Code

```python
import fitz  # PyMuPDF
import re
import spacy
import pandas as pd
from rapidfuzz import fuzz, process

# Load spaCy model
nlp = spacy.load("en_core_web_sm")

# PDF path
pdf_path = "/dbfs/FileStore/pdfs/resume.pdf"
doc = fitz.open(pdf_path)
pages = [page.get_text() for page in doc]
full_text = "\n".join(pages)

# Known values
known_skills = ['Azure Databricks', 'Delta Lake', 'Azure Data Factory', 'Power BI',
'SQL', 'Python', 'Spark', 'Palantir Foundry', 'S3', 'Redshift', 'Snowflake',
'Informatica', 'Data Modeling', 'Tableau', 'Pandas', 'NumPy']
roles_list = ['Data Engineer', 'Data Architect', 'Tech Lead', 'Consultant', 'Analyst',
'Solution Architect']
known_companies = ["BP", "Palantir", "Toyota", "CBRE", "Infosys", "TCS", "Heidelberg",
"Capgemini", "Deloitte", "Microsoft", "Google", "Wipro", "Virtusa", "AWS", "Azure",
"Accenture", "AT&T", "Nationwide Insurance", "Allied Insurance", "Scottsdale
Insurance", "State Farm", "Loblaw", "Amazon"]

# Extract contact info
email = re.search(r"[\w\.-]+@[\w\.-]+", full_text)
phone = re.search(r"\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}", full_text)
linkedin = re.search(r"https?://(www\.)?linkedin\.com/in/[a-zA-Z0-9\-]+", full_text)
name = "Not found"
for line in pages[0].splitlines():
    if "| Data Engineer" in line:
        name = line.split("|")[0].strip()
        break

# Extract skills, roles, durations
skills_found = [s for s in known_skills if s.lower() in full_text.lower()]
roles_found = [r for r in roles_list if r.lower() in full_text.lower()]
durations = re.findall(r"(\d+(\.\d+)?\s+years?)|(\d{4}\s*[--]\s*\d{4})|([A-Za-
z]{3,9}\s+\d{4}\s*[--]\s*[A-Za-z]{3,9}\s+\d{4})", full_text)
flattened_durations = list(set(["".join(filter(None, grp)).strip() for grp in
durations]))

# Extract companies using fuzzy match + NER
```

```
fuzzy_matches = process.extract(full_text, known_companies, scorer=fuzz.partial_ratio,
limit=20)
fuzzy_companies = [name for name, score, _ in fuzzy_matches if score >= 85]
doc_nlp = nlp(full_text)
ner_orgs = [ent.text.strip() for ent in doc_nlp.ents if ent.label_ == "ORG"]
companies = list(set(fuzzy_companies + ner_orgs))

# Build dictionary
parsed_resume = {
    "name": name,
    "email": email.group() if email else "Not found",
    "linkedin": linkedin.group() if linkedin else "Not found",
    "phone": phone.group() if phone else "Not found",
    "skills": skills_found,
    "roles": roles_found,
    "durations": flattened_durations,
    "companies": companies
}

# Save to CSV
csv_row = {k: ", ".join(v) if isinstance(v, list) else v for k, v in
parsed_resume.items()}
df = pd.DataFrame([csv_row])
df.to_csv("/dbfs/FileStore/parsed_resumes/parsed_resume_clean.csv", index=False)
print("✅ Resume successfully parsed and saved to CSV.")
```

Sample Output:

```
Name     : DURGA PRASAD PATSA
Email    : durgaXXXXXXX@gmail.com
LinkedIn : https://linkedin.com/in/durga-prasad-patsa-1a0335bb
Phone    : 309-XXX-XXXX

Skills:
 - Azure Databricks
 - Delta Lake
 - Power BI
 ...

Companies:
 - BP
 - AT&T
 - Microsoft
 - Nationwide Insurance
```