

Exercice Technique – Application d’Extraction Automatique de CV

Full Python – Sans IA

Objectif

Développer une mini-application **full Python**, capable de :

1. **Uploader un CV (PDF ou DOCX)** depuis un frontend Python (Streamlit)
2. **Envoyer le fichier au backend (FastAPI)**
3. **Extraire automatiquement** (sans IA) :
 - nom
 - prénom
 - email
 - téléphone
 - diplôme principal
4. Afficher le résultat côté frontend
5. Permettre la modification des données
6. Télécharger un JSON final

Structure exigée du projet

```
cv-extractor/  
  backend/  
    main.py  
    requirements.txt  
    services/  
      pdf_parser.py  
      docx_parser.py  
      extractor.py  
    models/  
      cv_result.py  
  frontend/  
    app.py  
    requirements.txt  
  docker/  
    Dockerfile.backend  
    Dockerfile.frontend  
    docker-compose.yml  
  README.md
```

Partie A – Backend en FastAPI

Stack imposée

- FastAPI
- Uvicorn
- PyPDF2 ou PDFPlumber
- python-docx
- regex (`re`)
- Pydantic

Endpoints

POST /api/v1/upload-cv

Entrée : fichier PDF/DOCX

Sortie JSON :

```
{
  "first_name": "Alain",
  "last_name": "Bernard",
  "email": "xxxx@gmail.com",
  "phone": "+336xxxxxxx",
  "degree": "Master Informatique"
}
```

Partie B – Pipeline d'extraction SANS IA

Le backend doit :

1 Extraire le texte brut

- PDF → PDFPlumber ou PyPDF2
- DOCX → python-docx
- Nettoyage :
 - supprimer les doublons d'espaces
 - normaliser accents
 - convertir en minuscule pour le matching

2 Extraire les informations via règles & regex

- ✓ Email
- ✓ Téléphone (formats FR/Intl)
- ✓ Nom & prénom (optionnel)
- ✓ Diplôme principal

Partie C – Frontend Streamlit

Pages obligatoires

Page 1 : Upload

- Upload PDF / DOCX
- Bouton "Analyser le CV"
- Appel POST → backend
- Loader via st.spinner

Page 2 : Résultat

Affichage dans champs modifiables :

- Nom
- Prénom
- Email
- Téléphone
- Diplôme

Bouton “Télécharger en JSON”

génère un fichier JSON téléchargeable

Partie D – Docker (full Python)

docker-compose.yml

Doit lancer :

- backend : FastAPI (port 8000)
- frontend : Streamlit (port 8501)

Partie E – Qualité exigée

Backend

- Gestion d'erreurs (HTTPException)
- Validation des fichiers
- Code structuré : services, models, utils
- Tests unitaires :
 - extraction email
 - extraction téléphone
 - extraction diplôme

Frontend

- Code clair, structuré
- UX simple et propre

README.md

Contient :

- installation
- lancement local
- lancement docker
- exemples d'API