FULL PROJECT BLUEPRINT

Intelligent Document Understanding (IDU) System

From ZERO → PRODUCTION

I'll break this into 12 phases.
You follow them in order.
By the end, you have a real AI product, not a toy.

PHASE 0 — Project Definition (DO NOT SKIP)

Goal	Convert unstructured documents (PDF / image) into structured data using AI.
Input	Invoice / form image or PDF
Output	Clean JSON

PHASE 1 — Folder Structure (Professional)

```
document-ai/

data/
   raw/
   processed/
   annotations/

cv/
   text_detection/
   ocr/
   preprocessing/

nlp/
   classification/
   ner/
   postprocessing/

api/
   main.py
   schemas.py

app/
   streamlit_app.py

models/
notebooks/
```

```
Dockerfile
requirements.txt
README.md
```

This alone already screams ”engineer”.

PHASE 2 — Data Collection

- **Minimum Required:** 300–500 invoices/forms
- **Sources:**
  - FUNSD dataset
  - RVL-CDIP
  - Manually download invoices (realistic)
- **Store:**
  - image.jpg
  - label.json

PHASE 3 — Image Preprocessing (CV Core)

Why?

OCR performance depends on image quality.

Techniques

- Grayscale
- Adaptive thresholding
- Noise removal
- Deskewing
- Resize (DPI normalization)

Tools

- OpenCV
- NumPy

Output

Clean images → data/processed/

PHASE 4 — Text Detection (Computer Vision)

Model

CRAFT (Character Region Awareness for Text)

What it does

Detects text bounding boxes

Output

```
[
  {"bbox": [x1,y1,x2,y2], "confidence": 0.98}
]
```

Tech

- PyTorch
- Pretrained CRAFT model

PHASE 5 — OCR (Deep Learning)

Model Options

- **BEST:** CRNN
- **FASTER:** Tesseract + post-processing

CRNN Pipeline

1. Crop text regions
2. CNN → feature extraction
3. BiLSTM → sequence modeling
4. CTC loss → decoding

Output

```
[
  {"text": "Invoice Number", "bbox": [...]}
]
```

PHASE 6 — Document Classification (AI Brain)

Why?

Invoices ≠ receipts ≠ contracts.

Model

BERT fine-tuned

Classes

- Invoice
- Receipt
- Form
- ID document

Input

Concatenated OCR text

Output

```
{"document_type": "Invoice"}
```

PHASE 7 — Named Entity Recognition (IMPORTANT)

Goal

Extract:

- Invoice number
- Date
- Total
- VAT
- Currency
- Vendor name

Model

BERT / LayoutLM (BONUS)

Labels

```
B-INVOICE_NO
B-DATE
B-TOTAL
B-CURRENCY
```

Output

```
{
  "invoice_no": "INV-2025-01",
  "total": 1240.50,
  "currency": "TND"
}
```

PHASE 8 — Post-Processing Logic (AI + Rules)

Why?

AI ≠ perfect.

Examples

- Regex validation
- Currency normalization
- Date format correction
- Confidence filtering

> **This shows engineering maturity.**

PHASE 9 — Evaluation & Metrics

CV Metrics

- OCR CER / WER
- Detection precision/recall

NLP Metrics

- F1 score (NER)
- Accuracy (classification)

Add:

- Error examples (screenshots)

PHASE 10 — API (REAL DEPLOYMENT)

Framework

FastAPI

Endpoints

```
POST /analyze-document
```

Flow

Upload → preprocess → detect → OCR → NLP → JSON

Output

```
{
  "doc_type": "Invoice",
  "entities": {...}
}
```

PHASE 11 — Frontend (Minimal but Clean)

Tool

Streamlit

Features

- Upload file
- Preview document
- Show extracted data
- Download JSON

PHASE 12 — Docker & Deployment

Docker

```
docker build -t document-ai .
docker run -p 8000:8000 document-ai
```

Deploy On

- AWS EC2
- Railway
- Render

Bonus

- Swagger docs
- Logging

PHASE 13 — README (VERY IMPORTANT)

Must Include

- Architecture diagram
- Sample inputs/outputs
- Metrics
- Model explanations
- Ethical considerations