Running the Intelligent Document Insight Engine

The project is shipped as a self-contained Docker solution; you **don't need to install any Python libraries or set up a local runtime**. Follow the steps below to build, run, and interact with both Round 1A (outline extraction) and Round 1B (persona-aware ranking).

1. Prerequisites

Component	Required version	Notes
Docker Engine / Desktop	20.10 +	Linux, macOS, or Windows 10/11 with WSL 2
CPU & RAM	≥ 4 vCPU & 8 GB	Model weights load into memory during Round 1B
Disk space	≈ 1 GB	Image size after build is ≈ 700 MB

> **Windows users:** enable WSL 2 and file-sharing for the drive where your `input/` and `output/` folders reside [1][2].

2. Obtain the code package

Download `connecting-dots.zip` from the previous assistant message.

Unzip to a working directory, e.g. `C:\doc-engine` or `~/doc-engine`.

Inside you will see:

```
app/
Dockerfile
docker-compose.yml
README.md
input/
output/
```

3. Build the Docker image

Open a terminal in the project root and run:

```
docker build -t connecting-dots .
```

First build takes 3 min on a typical laptop; subsequent builds are cached.

4. Prepare test PDFs

Copy one or more PDF files (≤ 50 pages each) into the `input/` folder.

Ensure filenames contain no spaces; the container scans `*.pdf`.

5. Run Round 1A – outline extraction

```
docker run -- rm ^
  -v $(pwd)/input:/app/input ^
  -v $(pwd)/output:/app/output ^
  connecting-dots --mode 1A
```

On Windows PowerShell replace `$(pwd)` *with* `${PWD}`.

Results: for every `document.pdf` you will find `document_outline.json` in `output/` containing the hierarchical heading list.

6. Run Round 1B – persona-aware ranking

```
docker run -- rm ^
  -v $(pwd)/input:/app/input ^
  -v $(pwd)/output:/app/output ^
  connecting-dots --mode 1B ^
  --persona  "Investment Analyst"   ^
  --job  "Analyze revenue trends"
```

The container loads all PDFs, embeds sections, and writes `persona_ranking.json` to `output/` within 60 s.

7. Combined workflow

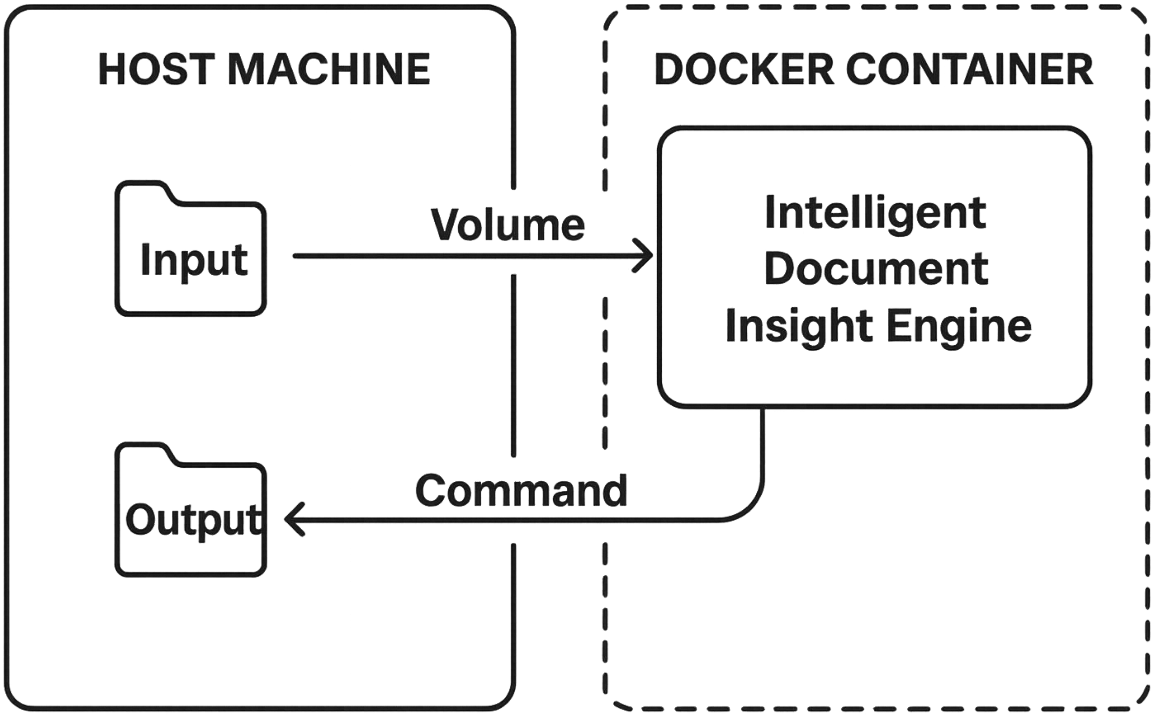
If you want both rounds in one shot:

```
docker run -- rm ^
  -v $(pwd)/input:/app/input ^
  -v $(pwd)/output:/app/output ^
  connecting-dots --mode both ^
  --persona  "Product Manager"   ^
  --job  "Find UX best-practice sections"
```

The script first produces outline JSONs then immediately performs ranking using the extracted sections.

8. Understanding volume mounts

The `-v hostDir:containerDir` flags bind local folders so the engine can read PDFs and write results without rebuilding the image.



Volume mounts when running the Intelligent Document Insight Engine Docker container.

9. Troubleshooting

Symptom	Likely cause	Fix
“no such file or directory” when mounting	Windows path not shared with Docker	Check Docker Desktop → Settings → Resources → File Sharing
Processing takes > 60 s	Huge PDFs or > 10 files	Split PDFs or process batches
“SentenceTransformer import” error	Host lacks AVX and Docker falls back to older CPU mode	Set environment variable `FAST_EMBEDDINGS=1` to use TF IDF fallback

10. Optional: interactive FastAPI UI

The repo also contains a minimal FastAPI interface (`web_app.py`). To explore:

```
docker compose --profile web up --build
```

Then browse to `http://localhost:8000`. The UI lets you upload PDFs and trigger ranking from the browser.

11. Clean-up

Stop and remove all containers:

```
docker compose down
```

Remove the image if no longer needed:

```
docker rmi connecting-dots
```

You’re all set

With Docker handling dependencies and isolation, running the Intelligent Document Insight Engine is reduced to **three commands**: *build* → *mount* → *run*. Enjoy extracting smart outlines and persona-tailored insights from your PDFs!

⁂