

Great — here is a **simple, clean, and realistic MVP** you can build and launch in **7 days**, using only free-tier hosting.

This MVP contains **4 API endpoints**, requires **almost zero RAM**, and provides **real value for developers & companies**.

---



# YOUR 7-DAY MVP: Document Utility API SaaS

Below is the full plan: endpoints, responses, tech stack, deployment, and pricing.  
You can literally **copy-build** this as your first SaaS.

---



## MVP WILL HAVE 4 ENDPOINTS

These 4 are enough for a real product, paying customers, and market validation.

---



### /extract-metadata

Extracts metadata from PDF or DOCX.

#### Input

File upload.

#### Output (JSON)

```
{
  "file_type": "pdf",
  "title": "Annual Report",
  "author": "Ali Khan",
  "page_count": 12,
  "word_count": 3945,
  "headings": ["Executive Summary", "Overview", "Conclusion"],
  "links": ["https://example.com"]
}
```

#### Real Use Cases

- HR portals extracting candidate info
- Research tools
- Resume analysis

- Document automation
- 

## **/validate-file**

Validates PDF, DOCX, CSV before storage.

### **Output Example**

```
{
  "valid": false,
  "errors": [
    "CSV missing header",
    "Unexpected number of columns: expected 5, got 3"
  ]
}
```

### **Checks you will implement:**

- File not corrupted
- Not empty
- PDF pages > 0
- DOCX contains readable text
- CSV header exists
- No harmful extensions

### **Use Cases**

- Upload portals
  - Government forms
  - College assignment upload
  - Business software
- 

## **/clean-text**

Cleans & normalizes user text.

### **Input**

Raw text (string).

## Output

```
{  
  "cleaned_text": "Hello World!",  
  "length_before": 23,  
  "length_after": 12  
}
```

## Transformations you apply

- Remove double spaces
- Remove weird Unicode (💎, Ã, etc.)
- Normalize encoding
- Convert “smart quotes” → quotes
- Remove HTML tags
- Optionally remove emojis

## Use Cases

- Chat apps
  - CMS text cleaning
  - Search engines
  - AI preprocessing
- 

## /standardize-filename

Standardizes ugly file names.

## Input

"cv final new updated(2).docx"

## Output

"CV\_2025\_Doc\_2.docx"

## Rules

- Remove special chars
- Convert spaces → underscores
- Capitalize

- Add date/version if needed

## Use Cases

- HR systems
  - CMS
  - Cloud storage
  - Project management tools
- 



## WHY THIS MVP IS PERFECT

- No heavy libraries
  - Everything done **in memory**
  - Works on **free hosting**
  - No background jobs
  - VERY LITTLE RAM consumption
  - No Office conversion engine
- 



## TECH STACK (simple + stable)

### Backend

- FastAPI **or** Django REST (FastAPI recommended for SaaS)
- Python 3.10+

### Libraries

- `pdfplumber` – extract text/metadata
- `PyPDF2` – basic PDF parsing
- `python-docx` – docx parsing
- `python-magic` – detect file type
- `regex` – text cleanup

All low-memory.

## Database

- Optional in MVP (use SQLite / JSON file)
  - Later → Postgres
- 



## FREE TIER DEPLOYMENT PLAN

### Best free options:

Platform	Why good
<b>Railway Free</b>	Perfect for small APIs
<b>Render Free</b>	Easy deploy, custom domain
<b>Fly.io</b>	Lightweight, fast
<b>Cloudflare Workers</b>	Free, global, super stable
<b>Your API fits in any of these.</b>	

---



## PRICING (copy this)

### Free

- 100 requests/month
- All 4 endpoints
- 5MB file size limit

### Basic — \$5/month

- 10,000 requests
- 20MB file limit

### Pro — \$19/month

- 100,000 requests
- Priority queue

### Enterprise — \$99+/month

- Dedicated resources
- Custom limits
- SLA

This pricing is standard for developer tools.

---



## LANDING PAGE (simple copy)

**Your tagline:**

**Document Utility APIs for Developers – Extract. Validate. Clean. Standardize.**

### Features

- Ultra-fast
  - Developer friendly
  - Free plan
  - REST API
  - Works with PDF, DOCX, CSV
- 



## 7-DAY BUILD SCHEDULE

### Day 1

- Create FastAPI project
- Setup routes

### Day 2

- Build /extract-metadata

### Day 3

- Build /validate-file

### Day 4

- Build /clean-text

### Day 5

- Build /standardize-filename

### Day 6

- Create landing page (Next.js or simple HTML)

## Day 7

- Deploy
  - Publish on GitHub
  - Post on Reddit + Product Hunt
- 



## READY FOR NEXT STEP?

If you want, I will prepare:

- ✓ **Sample FastAPI code for each endpoint**
- ✓ **Folder structure**
- ✓ **API documentation (Swagger style)**
- ✓ **Database plan (if adding billing later)**
- ✓ **Branding name suggestions**
- ✓ **SaaS UI dashboard (Next.js template)**

Just tell me:

**Do you want the code template or the SaaS name ideas next?**