Let’s complete your JG-Tech blueprint with the **À la carte module** — the adaptable toolset for one-off services, legacy requests, and specialty workflows. Think of it as the *sidebar menu* you pull from when clients ask for “just this one thing” that doesn’t quite fit Tier 1–3. 🧠🛠️📎

# **🟣 À la Carte Services Module**

## **🧩 Summary:**

This section holds flexible, standalone functions that can be bolted onto any tier — or used independently. Includes media conversions, specialty branding, archival, and retro-format handling.

### **🎥 MP4 Video Watermarking (Visible Overlay)**

bash

# Requires: ffmpeg

ffmpeg -i input.mp4 -i watermark.png \

-filter\_complex "overlay=W-w-10:H-h-10" \

output\_wm.mp4

* overlay=W-w-10:H-h-10 = bottom-right position
* Input watermark must be PNG with transparency
* Result: \_wm.mp4 file stored alongside original

### **🧠 MP4 Metadata Injection (Invisible Tagging)**

bash

ffmpeg -i input.mp4 \

-metadata title="JG-Tech – ClientName" \

-metadata author="Jim G." \

output\_embed.mp4

* Useful for silent tagging and branding
* Visible only in media info tools

### **📄 PDF Watermarking (Visible Text Stamp)**

python

from PyPDF2 import PdfReader, PdfWriter

from PyPDF2.generic import RectangleObject

def stamp\_pdf(input\_file, watermark\_text, output\_file):

reader = PdfReader(input\_file)

writer = PdfWriter()

for page in reader.pages:

page.merge\_text(

watermark\_text,

position=RectangleObject([400, 20, 550, 60]), # bottom corner

opacity=0.5,

font\_size=10

)

writer.add\_page(page)

with open(output\_file, "wb") as f:

writer.write(f)

### **🗂️ Legacy File Conversion (Sample Snippet)**

python

def convert\_txt\_to\_pdf(txt\_path, output\_path):

from fpdf import FPDF

pdf = FPDF()

pdf.add\_page()

pdf.set\_font("Arial", size=12)

with open(txt\_path, "r") as f:

for line in f:

pdf.cell(200, 10, txt=line.strip(), ln=True)

pdf.output(output\_path)

### **📦 One-Off File Sorter by Extension (Quick Utility)**

python

def sort\_files\_by\_extension(directory):

for file in os.listdir(directory):

ext = os.path.splitext(file)[1].lower().replace('.', '')

ext\_path = os.path.join(directory, ext)

os.makedirs(ext\_path, exist\_ok=True)

shutil.move(os.path.join(directory, file), os.path.join(ext\_path, file))

* Instantly creates subfolders like /jpeg, /mp4, /pdf
* Use post-upload or for archival cleanup

## **📝 Notes:**

* À la carte tools are modular — insert into any tier as needed
* Suffix each processed file (\_wm, \_embed, \_conv, \_tag) for clarity
* Ideal for isolated jobs, proofs of concept, or client-specific requests