

Handwritten Assignment Studio - Master Prompt

Build a production-ready full-stack web application called:

"Handwritten Assignment Studio"

The goal:

Users paste academic text and generate ultra-realistic handwritten-style PDFs with advanced formatting controls.

Tech Stack:

Backend:

- Python (Flask)
- ReportLab for PDF generation
- Pillow for realistic paper effects (optional)
- Flask-Login (for optional authentication)
- Gunicorn-ready

Frontend:

- HTML5
- Modern CSS
- Vanilla JavaScript
- Responsive design
- Live preview system

Project Structure:

handwritten-studio/

■

■■■ app.py

■■■ requirements.txt

■■■ README.md

■■■ /static

■ ■■■ style.css

■ ■■■ script.js

■ ■■■ preview.css

■

■■■ /templates

■ ■■■ index.html

■ ■■■ login.html

■

■■■ /fonts

■ ■■■ DancingScript-Regular.ttf

■ ■■■ Pacifico-Regular.ttf

■ ■■■ ComicNeue-Regular.ttf

■

■■■ /generated

CORE FEATURES

1) TEXT INPUT SYSTEM

- Large textarea
- Word counter
- Character counter
- Auto detect headings
- Option to manually mark sections using tags

2) ADVANCED FORMATTING ENGINE

- Detect HOME ASSIGNMENT → center + black + larger font
- Detect Name, ID → left aligned black
- Detect Date → right aligned black
- Detect numbered lines (1., 2., 3.) → Question formatting
- Remaining lines → Answer formatting (dark blue #0A1F5C)

3) USER CONTROLS PANEL

- Font selector
- Font size slider
- Line spacing slider
- Ink color picker
- Margin slider
- Page style selector
- Page size selector
- Realism toggles (spacing variation, jitter, ink variation)

4) LIVE PREVIEW SYSTEM

- Real-time preview before PDF generation

5) PDF GENERATION ENGINE

- Use ReportLab BaseDocTemplate
- Small margins
- Multi-page auto handling
- Optional notebook margin lines
- Optional ruled lines

Functions:

- generate_pdf(text, settings)
- format_line(line, settings)
- draw_page_background(canvas, settings)

6) EXPORT OPTIONS

- Download as PDF
- Download as JPG
- Download as PNG

7) SESSION HANDLING

- Store user settings
- Remember last used font
- Temporary file cleanup

8) SECURITY

- Sanitize input
- Prevent script injection
- Restrict font access

9) OPTIONAL USER ACCOUNT SYSTEM

- Register/Login
- Save history

- Premium font access

10) UI DESIGN REQUIREMENTS

- Clean modern UI
- Two-column layout (Controls + Preview)
- Mobile responsive

11) DEPLOYMENT READY

- Procfile
- Unicorn support
- Environment variables support

12) REQUIREMENTS.TXT

Flask

reportlab

Pillow

flask-login

unicorn

13) README

- Installation instructions
- How to run
- How to add fonts
- Deployment steps

Generate full working production-quality code for:

- Backend (app.py)
- Frontend templates
- CSS
- JavaScript
- requirements.txt
- README.md