

You are a senior Python architect building a desktop, plugin-based document intelligence application named:

DOCTURA DESKTOP (DocTura)

=====

PRODUCT GOAL

=====

Build a generic, offline-first, enterprise-grade document intelligence system that converts semi-structured and structured documents into high-quality Excel, CSV, Word, and PDF outputs, with deterministic validation, audit logging, and optional AI summarisation.

DocTura MUST handle:

- layout-shifting PDFs
- multi-page tables
- header-delimited rosters
- score-domain-constrained statistical reports
- round-trip workflows (PDF → Excel/Word → PDF)

The system design is permanently informed by a known failure case where a PDF-based statistical report truncated data due to page-boundary assumptions. Therefore:

- Page boundaries MUST NOT determine logical table boundaries.
- Domain rules MUST override visual layout.

=====

DEFAULT BEHAVIOURS

=====

- Extraction Mode: HYBRID
(page-preserved sheets + logical tables)
- Title Handling: METADATA SHEET ONLY
- Naming: SMART NAMING ENABLED
- Validation: DETERMINISTIC AND MANDATORY
- Core functionality works OFFLINE
- Theme support: CORPORATE THEME and INDIGENOUS THEME

=====

THEME DEFINITIONS (MUST IMPLEMENT)

=====

CORPORATE THEME

- Primary: #0B1F3B (Deep Navy Blue)
- Secondary: #4A5568 (Slate Grey)
- Accent: #C9A227 (Gold)
- Background: #F7F9FC (Off-White)
- Surface: #FFFFFF
- Text Primary: #1A202C
- Text Secondary: #6B7280
- Success: #1F7A1F
- Warning: #D97706
- Error: #9B1C1C

Usage:

- Primary buttons: Navy Blue
- Accent actions (Convert/Export): Gold
- Headers: Navy Blue
- Tables: White background, subtle grey gridlines

INDIGENOUS THEME

- Primary: #5A3E2B (Earth Brown)
- Secondary: #1E5631 (Forest Green)
- Accent: #C05621 (Burnt Orange)
- Highlight: #D69E2E (Ochre Yellow)
- Background: #FAF3E0 (Warm Sand)
- Surface: #FFF8ED
- Text Primary: #2D1B12
- Text Secondary: #6B705C
- Success: #2F855A
- Warning: #B7791F
- Error: #9C4221

Usage:

- Primary buttons: Forest Green
- Accent actions: Burnt Orange
- Headers: Earth Brown
- Tables: Warm Sand background, soft borders

Theme selection affects UI and export styling only, never data.

=====

INPUTS

=====

- PDF (native and scanned)
- DOCX
- PNG/JPG/TIFF
- WAV/MP3/M4A

=====

OUTPUTS

=====

- XLSX (always)
- Optional CSV
- Optional DOCX
- Optional PDF

=====

EXTRACTION & SEGMENTATION

=====

Implement HYBRID extraction:

- 1) Page-preserved representation (Page_01, Page_02, ...)
- 2) Logical table reconstruction using:

- Score-domain routing (for statistical distributions)
- Header-repetition routing (for rosters)

Examples:

- Scaled Objective: 0–19
- Scaled Essay: 15–40
- Header repetition defines table start/end for staff lists

Ignore page breaks during logical segmentation.

=====

EXCEL OUTPUT OPTIONS

=====

User must choose:

- Each table → separate worksheet

OR

- All tables → single worksheet

If single worksheet:

- Vertical stacking (down rows)

OR

- Horizontal placement (across columns with borders)

=====

WORD OUTPUT OPTIONS

=====

- Portrait or Landscape
- Optional page break per table
- Section titles as headings
- Tables rendered using Word table API

=====

REVERSE PDF GENERATION

=====

Support:

- Excel → PDF

- Word → PDF

Structural rendering only (deterministic).

Respect orientation, page breaks, and table layout.

=====

VALIDATION

=====

Deterministic rules only:

- Percent totals = 100.00 ± tolerance
- No duplicate rows
- Score ranges enforced
- Monotonic cumulative frequency
- Header-before-data enforcement for rosters
- No orphan rows

=====

PLUGIN ARCHITECTURE

=====

Implement plugins with interface:

- detect
- extract
- route
- validate
- summarize (optional)

Built-in plugins:

- 1) waec_marksdist_plugin
 - score-domain routing
- 2) international_staff_list_plugin
 - header-based segmentation

=====

REPOSITORY STRUCTURE

=====

docutura/
 app/
 core/
 plugins/
 enterprise/
 tests/
 README.md
 pyproject.toml

=====

GUI

=====

Use PySide6 (Qt).

Provide:

- File selection / drag-drop
- Output format selection
- Excel layout selection
- Word orientation selection
- Theme selection
- Progress bar and live logs
- Output folder shortcut

=====

DELIVERABLE

=====

Generate the full repository so that:

pip install -e .

python -m docutura.app.main

runs the desktop application successfully.