

Build an AI Document Processing System with AI

Execution Path

A structured route from idea to working output - the steps, outcomes, workflows, resources, and tools, in order.

Category: AI Systems Steps: 7 Phases: 4

Exported: 2026-07-21 (UTC)

Source: <https://newprompt.net/blueprints/build-ai-document-processing-system-with-ai/execution-map>

PATH SUMMARY

7 steps across 4 phases.

Use this document as a portable execution guide. Each step lists the expected outcome and the workflow, resource, and tool used in NewPrompt.

PHASE 1 · Define & Scope

Clarify what you're building and for whom.

Step 1 · Define the document processing use case

Frame what the system is for before building it - which documents it ingests, which fields matter, and what 'correct' output looks like - so every later stage serves a defined job instead of guessing.

OUTCOME

A clear use case: the documents in scope and the fields the system must produce.

USED IN THIS STEP

Workflow AI Product Requirements Workflow

Resource Product Requirements Assistant

Tool System Prompt Generator

PHASE 2 · Design

Design the right solution before building.

Step 2 · Design the document intake pipeline

Map how documents enter and move through processing - the ingestion, the stages each document passes, and where things can go wrong - as a designed pipeline rather than one opaque script.

OUTCOME

An intake and processing pipeline designed stage by stage.

USED IN THIS STEP

Workflow AI Data Pipeline & ETL Workflow

Resource Data Analyst Role Prompt

Tool Role Prompt Generator

Step 5 · Pin the structured output contract

Lock the shape of the output - a strict schema with validation rules - so downstream systems consume parseable, predictable data every time instead of whatever the model happens to return.

OUTCOME

A strict, parseable output contract the system produces every run.

USED IN THIS STEP

Workflow AI Reliable JSON Output Workflow

Resource Turn a JSON Schema into a Prompt

Tool JSON Output Prompt Builder

PHASE 3 · Build & Refine

Build, test, secure, and make it production-ready.

Step 3 · Extract the fields from documents

Build the extraction that pulls the defined fields out of unstructured documents - handling missing values and ambiguity explicitly, so the output is the right fields, not a confident guess.

OUTCOME

Defined fields extracted reliably from unstructured input.

USED IN THIS STEP

Workflow AI Data Extraction Workflow

Resource Extract Data From Text with AI

Tool Extraction Prompt Generator

Step 4 · Classify document types and routing

Recognize what each document is - invoice, contract, form, email - from a fixed set, with edge-case rules, so each document is routed to the right processing path instead of one generic flow.

OUTCOME

Documents classified by type and routed correctly.

USED IN THIS STEP

Workflow	AI Classification Workflow
Resource	Text Classification Prompt - the Anatomy
Tool	Data Classification Prompt

Step 6 · Evaluate extraction accuracy and failures

Build the test set the system has to pass - documents with known correct outputs, including the hard and malformed cases - so accuracy and failure modes are measured before the system runs on real volume unattended.

OUTCOME

Extraction accuracy tested against real and failure-case documents.

USED IN THIS STEP

Workflow	AI Agent Evaluation Workflow
Resource	Agent Test Scenario Prompt
Tool	Test Case Prompt Generator

PHASE 4 · Ship & Validate

Ship with confidence and validate results.

Step 7 · Prepare deployment and monitoring

Plan the move to production - readiness checks, and monitoring on extraction quality and document volume - so silent drift in accuracy is caught early instead of after it has corrupted a downstream system.

OUTCOME

A deploy-and-monitor plan that watches extraction quality in production.

USED IN THIS STEP

Workflow	AI Deployment & Release Workflow
Resource	Production Readiness Review Prompt
Tool	Markdown Output Builder

This PDF is a static export of the NewPrompt execution path. It does not include workspace progress, personal notes, saved outputs, or project state.