

# CertPrep AI

## Multi-Agent Certification Preparation System

Technical Documentation · Microsoft Agents League 2026 · Battle #2

■ Date	February 2026
■ Track	Battle #2 – Reasoning Agents with Microsoft Foundry
■ Architecture	Multi-agent pipeline (7 agents + 17 guardrails)
■■ Stack	Python 3.11 · Streamlit · Plotly · Pydantic · ReportLab

## 1. Executive Summary

CertPrep AI is a **multi-agent, pipeline-based** certification preparation system designed to help learners pass the **Microsoft AI-102 (Azure AI Engineer Associate)** exam on their first attempt. The system guides a student from initial intake through structured domain profiling, personalised study planning, daily learning content curation, mid-journey readiness assessment, automated knowledge quizzes, and a final exam booking decision.

The application is implemented as a **Streamlit web app** with 7 interactive tabs, backed by 7 specialised Python agent classes and a 17-rule guardrails pipeline. All agent transitions are validated, all outputs are visible to the user, and the system supports both a **mock (offline) mode** and a **live Azure OpenAI mode**.

## 2. System Architecture — Agent Pipeline

The pipeline is organised into four sequential blocks, each encapsulating one or more agents. A dedicated guardrails layer sits between every block, providing input validation, output bounds checking, and content safety inspection.

Block	Agent(s)	Role	Output
Block 0 Input	Streamlit UI form	Collect student intake form data	RawStudentInput dataclass
Block 1 Intake & Profiling	LearnerIntakeAgent LearnerProfilingAgent	Interview student → infer experience level, domain knowledge scores, risk domains	LearnerProfile (Pydantic model)
Block 1.1 Learning Path	LearningPathCuratorAgent StudyPlanAgent	Map domains to MS Learn modules → Build week-by-week Gantt study schedule	LearningPath + StudyPlan
Block 1.2 Progress Tracking	ProgressAgent Engagement Agent (email)	Self-check-in → readiness scoring → smart nudges → weekly email report	ReadinessAssessment + HTML email
Block 2 Assessment	AssessmentAgent	Generate domain-weighted quiz (up to 30 Qs) → score → provide feedback	Assessment + AssessmentResult
Block 3 Certification	CertificationRecommendationAge nt	GO / NO-GO exam decision → exam logistics → next certification path	CertRecommendation

## 3. Agent-by-Agent Breakdown

Agent 1: LearnerIntakeAgent		Block: Block 1 · File: src/cert_prep/intake_agent.py
<b>Trigger</b>	User submits intake form	
<b>Input</b>	Student form values (name, background, hours, weeks, certs, concerns)	
<b>Output</b>	RawStudentInput dataclass	
<b>Reasoning</b>	Collects structured answers to 7 standardised questions. In CLI mode it uses Rich prompts; in Streamlit mode the form replaces the CLI. No AI inference at this stage — pure data collection.	
<b>Guardrails</b>	G-01 (empty name/target), G-02 (hours range), G-03 (weeks range), G-04 (cert code check), G-05 (PII notice)	

Agent 2: LearnerProfilingAgent		Block: Block 1 · File: src/cert_prep/intake_agent.py / mock_profiler.py
<b>Trigger</b>	After LearnerIntakeAgent returns RawStudentInput	
<b>Input</b>	RawStudentInput	
<b>Output</b>	LearnerProfile (6 DomainProfile objects + metadata)	
<b>Reasoning</b>	In Mock mode: rule-based inference maps keywords in background_text to domain knowledge levels (UNKNOWN / WEAK / MODERATE / STRONG) and sets confidence scores. In Live mode: sends a JSON-schema-anchored system prompt to Azure OpenAI gpt-4o, parses the response directly into the Pydantic LearnerProfile model.	
<b>Guardrails</b>	G-06 (6 domains present), G-07 (confidence ∈ [0,1]), G-08 (valid domain IDs)	

Agent 3: LearningPathCuratorAgent		Block: Block 1.1 · File: src/cert_prep/learning_path_curator.py
<b>Trigger</b>	After profile generation (autorun on submit)	
<b>Input</b>	LearnerProfile	
<b>Output</b>	LearningPath (~20–30 MS Learn modules, domain-grouped)	
<b>Reasoning</b>	Iterates over sorted domain profiles (risk → normal → skip). For each domain pulls modules from a curated offline catalogue of 30+ MS Learn entries. Applies priority boosting for risk domains (supplemental → core). Skips beginner-level modules for learners with moderate/strong knowledge. Respects a 2x budget cap to prevent overwhelming the learner.	
<b>Guardrails</b>	G-17 (URL trusted domain check — all links must be learn.microsoft.com etc.)	

Agent 4: StudyPlanAgent		Block: Block 1.1 · File: src/cert_prep/study_plan_agent.py
Trigger	After profile generation (autorun on submit)	
Input	LearnerProfile + existing_certs list	
Output	StudyPlan (Gantt tasks, prerequisite info, hours breakdown)	
Reasoning	Checks _CERT_PREREQ_MAP to identify missing, held, and helpful certifications. Uses the Largest Remainder Method at day granularity (7 days/week) to allocate study hours across active domains proportional to exam weight × knowledge deficit. Front-loads risk domains; schedules skip-eligible domains minimally at the end. Last week is reserved as a review + practice exam block.	
Guardrails	G-09 (start_week ≤ end_week), G-10 (allocated hours ≤ budget + 10%)	
Agent 5: ProgressAgent		Block: Block 1.2 · File: src/cert_prep/progress_agent.py
Trigger	User submits My Progress check-in form	
Input	LearnerProfile + ProgressSnapshot (hours, weeks, domain self-ratings, practice exam)	
Output	ReadinessAssessment (score, verdict, nudges, GO/NO-GO, domain status)	
Reasoning	Composite readiness formula: readiness = 0.55 × weighted_domain_score + 0.25 × hours_progress_pct + 0.20 × practice_factor. Verdict thresholds: ≥75% = EXAM_READY, ≥60% = NEARLY_READY, ≥45% = NEEDS_WORK, <45% = NOT_READY. GO/NO-GO: GO if ≥75% + 0 critical domains; CONDITIONAL GO if ≥65% + ≤1 critical. Generates up to 6 smart nudges categorised (DANGER/WARNING/INFO/SUCCESS).	
Guardrails	G-11 (hours ≥ 0), G-12 (ratings ∈ [1,5]), G-13 (practice score ∈ [0,100])	
Agent 6: AssessmentAgent		Block: Block 2 · File: src/cert_prep/assessment_agent.py
Trigger	User clicks 'Generate New Quiz' (human-in-the-loop gate)	
Input	LearnerProfile + requested question count	
Output	Assessment (questions) → AssessmentResult (score, domain breakdown, feedback)	
Reasoning	Samples questions from a 30-question bank (5 per domain, 3 difficulty levels) proportional to exam domain weights. Excludes skipped domains. Fixes rounding to guarantee exactly N questions. Scoring: correct answers / total × 100. Per-domain scores enable targeted feedback. PASS threshold: 60%. Result feeds CertificationRecommendationAgent.	
Guardrails	G-14 (≥5 questions), G-15 (no duplicate IDs)	

<b>Agent 7: CertificationRecommendationAgent</b>		<b>Block:</b> Block 3 · <b>File:</b> src/cert_prep/cert_recommendation_agent.py
<b>Trigger</b>	After AssessmentResult or ReadinessAssessment is available	
<b>Input</b>	LearnerProfile + AssessmentResult (or ReadinessAssessment)	
<b>Output</b>	CertRecommendation (GO/NO-GO, ExamInfo, next-cert suggestions, booking checklist)	
<b>Reasoning</b>	GO threshold: assessment score $\geq 70\%$ (HIGH confidence $\geq 85\%$ ). If GO: returns ExamInfo (passing score, duration, cost, Pearson VUE link, free practice URL) + 7-item booking checklist + 2–3 next-cert suggestions. If NO-GO: generates a targeted remediation plan listing weak domains and recommends a 2–3 week focused study cycle before retaking. If looping from a failed assessment, the pipeline returns to Block 1.1 (LearningPathCuratorAgent).	
<b>Guardrails</b>	G-17 (all recommendation URLs verified against trusted domain list)	

## 4. Guardrails Layer — Responsible AI

Every agent transition passes through the **GuardrailsPipeline**, which checks 17 rules across 5 categories. Violations are classified as BLOCK (hard-stop), WARN (advisory), or INFO (transparent logging). No hallucinated or internally inconsistent data is passed between agents.

Code	Level	Category	Rule Description
G-01	BLOCK/WARN	Input	Non-empty: student name, exam target, background description
G-02	WARN	Input	Hours per week in sensible range [1–80]
G-03	BLOCK/WARN	Input	Weeks available $\geq 1, \leq 52$ (warn if $>52$ )
G-04	WARN	Input	Exam code recognised in certification catalogue
G-05	INFO	Input	PII notice: student name stored in session only, not sent externally
G-06	WARN	Profile	All 6 AI-102 domains present in profiling output
G-07	BLOCK	Profile	Confidence scores within [0.0, 1.0] range
G-08	WARN	Profile	Risk domain IDs must be valid AI-102 domain identifiers
G-09	BLOCK	Plan	No study task may have start_week > end_week
G-10	WARN	Plan	Total allocated hours must not exceed budget by more than 10%
G-11	BLOCK	Progress	Hours spent must be $\geq 0$
G-12	BLOCK	Progress	Domain self-ratings must be in [1, 5] range
G-13	BLOCK	Progress	Practice exam score must be in [0, 100] when provided
G-14	WARN	Quiz	Assessment must contain at least 5 questions for reliability

G-15	BLOCK	Quiz	No duplicate question IDs in a generated assessment
G-16	BLOCK	Content	Heuristic harmful/profanity content check on all free-text outputs
G-17	WARN	Content	All URLs must originate from trusted domains (learn.microsoft.com etc.)

## 5. Key Data Models

Model / Dataclass	Module	Purpose	Key Fields
RawStudentInput	models.py	Raw form data before profiling	student_name, exam_target, background_text, hours_per_week, weeks_available
LearnerProfile (Pydantic)	models.py	Structured learner profile	student_name, experience_level, learning_style, domain_profiles, risk_domains, analogy_map
DomainProfile (Pydantic)	models.py	Per-domain knowledge assessment	domain_id, knowledge_level, confidence_score, skip_recommended, notes
LearningPath	learning_path_curator.py	MS Learn module mapping	curated_paths (dict), all_modules, total_hours_est, skipped_domains
LearningModule	learning_path_curator.py	Single MS Learn module	title, url, domain_id, duration_min, difficulty, priority
StudyPlan	study_plan_agent.py	Gantt study schedule	tasks, prereq_info, review_start_week, total_weeks
StudyTask	study_plan_agent.py	One domain's study block	domain_id, start_week, end_week, total_hours, priority, confidence_pct
ProgressSnapshot	progress_agent.py	Mid-journey self-report	total_hours_spent, weeks_elapsed, domain_progress, done_practice_exam, practice_score_pct
ReadinessAssessment	progress_agent.py	Readiness scoring output	readiness_pct, verdict, nudges, exam_go_nogo, domain_status, recommended_focus
Assessment	assessment_agent.py	Generated quiz instance	questions (list[QuizQuestion]), total_marks, pass_mark_pct
AssessmentResult	assessment_agent.py	Scored quiz output	score_pct, passed, domain_scores, feedback, verdict, recommendation

Model / Dataclass	Module	Purpose	Key Fields
CertRecommendation	cert_recommendation_agent.py	Exam booking decision	go_for_exam, exam_info, next_cert_suggestions, remediation_plan, booking_checklist
GuardrailResult	guardrails.py	Guardrail check outcome	passed, violations (list[GuardrailViolation]), blocked, warnings, infos

## 6. Reasoning Patterns Applied

### Planner–Executor

StudyPlanAgent (planner) decomposes the total study budget into per-domain week-blocks using exam weight + knowledge deficit, then 'executes' by emitting concrete StudyTask records with explicit start/end weeks.

### Critic / Verifier

The GuardrailsPipeline acts as an automated critic after every agent. It verifies that outputs meet structural constraints (bounds, completeness) before the next agent receives them, preventing cascading errors.

### Self-Reflection & Iteration

The Progress→Assessment→Recommendation loop implements iterative self-reflection: if the learner fails the quiz (<60%), the pipeline loops back to LearningPathCuratorAgent for a new curated study cycle rather than terminating.

### Role-Based Specialisation

Each agent has a single, well-defined responsibility: Intake (data collection) → Profiling (inference) → Curation (content mapping) → Planning (scheduling) → Progress (tracking) → Assessment (evaluation) → Recommendation (decision). No agent performs more than one role.

### Human-in-the-Loop

The assessment is not triggered automatically — the student must explicitly confirm they are ready by clicking 'Generate New Quiz'. Similarly the progress check-in is voluntary, preserving learner agency.

## 7. Streamlit UI — 7-Tab Structure

### Tab 1: ■■ Domain Map

Radar chart + bar chart of domain confidence scores. Auto-generated Radar Insights and Bar Chart Insights callouts highlight strongest/weakest domains, below-threshold areas, and skip candidates.

### Tab 2: ■ Study Setup

Prerequisites section (missing/held/helpful certs), Prerequisite notes, interactive Plotly Gantt chart, hours breakdown dataframe, and condensed profile summary cards.

### Tab 3: ■ Learning Path

Domain-by-domain expandable lists of curated MS Learn modules. Each module shows title (linked), type, difficulty, duration, and priority. Summary KPIs: modules curated, estimated hours, budget utilisation.

### Tab 4: ■ Recommendations

Personalisation recommendation from profiling, readiness outlook progress bar, CertificationRecommendationAgent output (GO/NO-GO card, exam logistics, booking checklist, next-cert suggestions), and full agent pipeline status tracker.

### Tab 5: ■ My Progress

Self-assessment form (hours, weeks, domain sliders, practice exam). On submit: Plotly gauge + GO/NO-GO card + colour-coded nudge alerts + domain status table (actual vs expected) + focus recommendation + email section.

### Tab 6: ■ Knowledge Check

On-demand quiz (5–30 Qs). Question bank covers all 6 domains with 3 difficulty levels. Results include score, per-domain breakdown, per-question feedback with explanations, and a 'Retake Quiz' option.

### Tab 7: ■ Raw JSON

Raw RawStudentInput and LearnerProfile JSON for debugging/inspection. Download button exports the full profile as a JSON file.

## 8. Repository File Structure

File / Directory	Purpose
streamlit_app.py	Main Streamlit UI (1600+ lines). Entry point for `streamlit run`.
demo_intake.py	CLI demonstration of the LearnerIntakeAgent + LearnerProfilingAgent.
requirements.txt	Python dependencies (openai, streamlit, plotly, pydantic, reportlab, ...)
src/cert_prep/models.py	Core data models: enums, AI102_DOMAINS, RawStudentInput, LearnerProfile, DomainProfile.
src/cert_prep/intake_agent.py	LearnerIntakeAgent (CLI) + LearnerProfilingAgent (Azure OpenAI).
src/cert_prep/mock_profiler.py	Rule-based mock profiler — no Azure credentials needed.
src/cert_prep/study_plan_agent.py	StudyPlanAgent: prerequisite lookup + Largest Remainder allocation.
src/cert_prep/learning_path_curator.py	LearningPathCuratorAgent: 30+ MS Learn module catalogue.
src/cert_prep/progress_agent.py	ProgressAgent: composite readiness scoring + SMTP email dispatch.
src/cert_prep/assessment_agent.py	AssessmentAgent: 30-question bank + quiz scoring + per-domain feedback.

File / Directory	Purpose
src/cert_prep/cert_recommendation_agent.py	CertificationRecommendationAgent: GO/NO-GO + next-cert path.
src/cert_prep/guardrails.py	GuardrailsPipeline: 17 validation rules across 5 categories.
src/cert_prep/agent_trace.py	AgentTrace: step-by-step reasoning trace for debugging/transparency.
src/cert_prep/config.py	AzureOpenAIConfig: reads from .env file.
Notes/generate_docs.py	This script — generates technical_documentation.pdf and judge_playbook.pdf.
docs/technical_documentation.pdf	← generated by this script
docs/judge_playbook.pdf	← generated by this script

## 9. Competition Evaluation Criteria Coverage

Criterion	Weight	How CertPrep AI addresses it
Accuracy & Relevance	25%	Directly implements the challenge scenario (AI-102 prep). Profiling accurately maps student backgrounds to domain knowledge. Rule-based inference validated against 3 diverse personas. All agent outputs structured and bounded.
Reasoning & Multi-step Thinking	25%	7-agent pipeline with explicit block sequencing. Planner–Executor, Critic/Verifier, and Self-Reflection patterns applied. Domain allocation uses mathematical reasoning (Largest Remainder Method). Readiness formula is explainable and multi-factor.
Creativity & Originality	15%	Returning-user journey with mid-journey progress tracking is novel vs the reference architecture. Animated colour-coded KPI cards, Plotly Gantt, radar+bar dual-chart domain assessment. SMTP weekly summary email with HTML KPI cards. GO/NO-GO exam decision with booking checklist.
User Experience & Presentation	15%	Fully interactive Streamlit web app with 7 tabs. Colour-coded guardrail notices, smart nudge alerts, visual readiness gauge. All outputs immediately visible — no hidden intermediate states. Export to JSON, HTML email, PDF documentation.
Reliability & Safety	20%	17-rule GuardrailsPipeline blocks bad data at every transition. URL trust checking (G-17) prevents hallucinated or poisoned links. PII notice in G-05. Mock mode requires no live AI credentials. Graceful fallback: Live Azure OpenAI failure falls back to mock profiler.