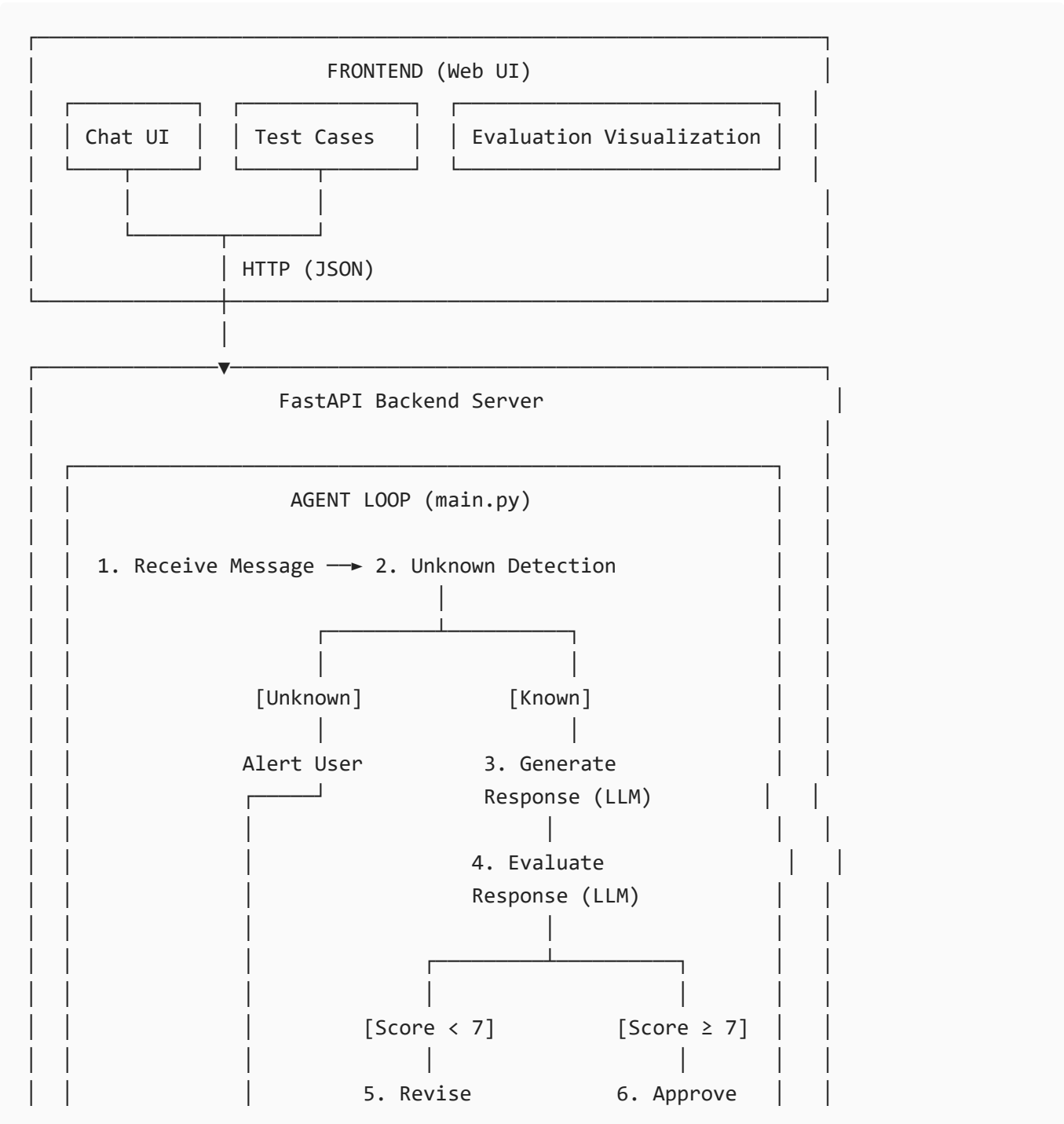


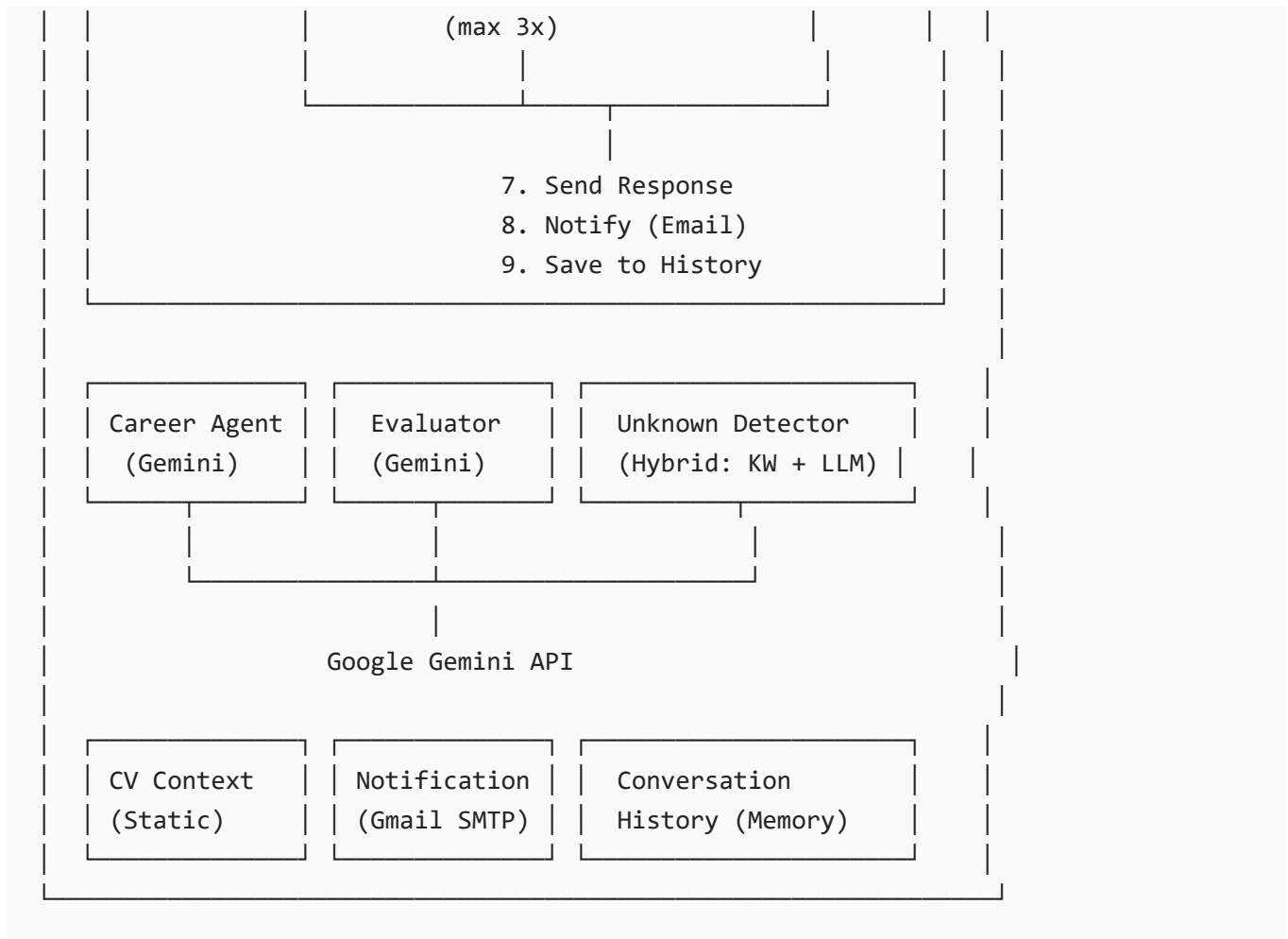
Career Assistant AI Agent - Architecture Documentation

System Overview

The Career Assistant AI Agent is an intelligent system that communicates with potential employers on behalf of **Burak Yalçın**. It uses Google's Gemini AI to generate professional responses, evaluate their quality, and detect questions outside its knowledge scope.

Architecture Diagram





Component Details

1. Career Agent (`career_agent.py`)

- **Type:** Primary Response Agent
- **LLM:** Google Gemini 2.0 Flash
- **Input:** Employer message + CV context + optional revision feedback
- **Output:** Professional response text
- **Prompt Design:** System prompt includes full CV as context with strict response guidelines

2. Response Evaluator (`evaluator_agent.py`)

- **Type:** Self-Critic / Judge Agent
- **Approach:** LLM-as-a-Judge
- **Scoring Criteria** (each 1-10):
 - Professional Tone
 - Clarity
 - Completeness
 - Safety (no hallucinations)

- Relevance
- **Threshold:** Overall score ≥ 7.0 to pass
- **Max Revisions:** 3 attempts before accepting

3. Unknown Question Detector (`unknown_detector.py`)

- **Type:** Safety / Confidence Assessment Tool
- **Approach:** Hybrid (Keyword Matching + LLM Confidence Scoring)
- **Categories:** Salary Negotiation, Legal, Out-of-Domain, Ambiguous Offer
- **Confidence Threshold:** 0.6 (below triggers alert)
- **Action on Detection:** Logs event + sends email alert

4. Email Notification (`notification.py`)

- **Type:** Communication Tool
- **Protocol:** Gmail SMTP with TLS
- **Triggers:**
 - New employer message received
 - Response approved and sent
 - Unknown question detected (human intervention needed)
- **Format:** HTML email with color-coded alerts

5. Conversation History (Memory)

- **Storage:** In-memory list (`conversation_history`)
- **Tracked Data:** Message, response, scores, timestamps, revision count
- **Endpoint:** `GET /api/history`

API Endpoints

Method	Endpoint	Description
POST	<code>/api/message</code>	Process employer message through agent loop
GET	<code>/api/history</code>	Get conversation history
GET	<code>/api/logs</code>	Get agent operation logs
GET	<code>/api/health</code>	Health check
GET	<code>/</code>	Serve frontend

Agent Loop Flow

1. **Receive** employer message via POST `/api/message`
2. **Notify** about new message (email)
3. **Detect** unknown questions (hybrid keyword + LLM)
4. **Alert** if unknown (email + log)
5. **Generate** response (Career Agent with CV context)
6. **Evaluate** response quality (Evaluator Agent)
7. **Revise** if score < 7.0 (up to 3 attempts)
8. **Approve** and notify (email)
9. **Store** in conversation history
10. **Return** complete response with metadata

Technology Stack

Component	Technology
Backend	Python 3.11 + FastAPI
LLM	Google Gemini 2.0 Flash
Notification	Gmail SMTP
Frontend	HTML/CSS/JS (Vanilla)
Styling	Custom CSS (Dark theme + Glassmorphism)
Data Models	Pydantic v2