Agent Engineer Take-Home Assessment

Time Budget: 2-3 days

Language: All agent conversations must be in Arabic

You will submit:

1. `DESIGN.md` - Your thinking before implementation
2. Working code in a GitHub repo
3. Architecture diagram showing interactions between each “Journey”
4. 5-10 minute recording explaining your approach

What You're Building

You will build an **Arabic-language restaurant ordering agent** that handles delivery orders through a multi-agent architecture. Think of it as the AI backend for a restaurant's WhatsApp ordering system or call center.

The Scenario

A customer messages the restaurant. Your system must:

1. Greet them and confirm they want to place an order (not file a complaint)
2. Collect their delivery location and validate coverage
3. Take their order from a 100+ item menu
4. Confirm the order with total (including delivery fee)

What We're Testing

This assessment evaluates your intuition on:

Area	Questions We Want You to
Handoff Design	What context moves betwe the transition?
Message Architecture	When do you use system vs
Context Limits	100 menu items is a lot of to
Arabic Capabilities	Which model handles Arabi
Tradeoff Reasoning	There's no single right answ

Part 1: Design Document

Before writing any code, create a `DESIGN.md` that addresses:

1.1 Architecture Overview

- Diagram showing your agent flow (Greeting → Location → Order → Checkout)
- What triggers each handoff
- What data/context transfers between agents

1.2 Context Management Strategy

Answer these questions:

1. **Message Roles:** For each instruction type (agent procedures, menu data, conversation history), will you use `system`, `user`, or `assistant` role? Why?
2. **Context Budget:** With 100+ menu items, how do you prevent context overflow?
    - Estimate tokens for: system instructions, menu, conversation history
    - What's your strategy when approaching limits?
    - Keep in mind you don't need to reach the limit for the TTFT to become unusable

3. **Handoff Context:** For each transition:
    - What MUST transfer?
    - What SHOULD transfer?
    - What should NOT transfer?
4. **History Handling:** When Agent B receives handoff from Agent A:
    - Does B see A's full conversation?
    - Does B see A's system prompt?
    - How do you summarize/truncate?
    - Why do you do this? What are the cons of transferring the entire chat between agents?

1.3 LLM Selection

Justify your model choice:

- Arabic language capabilities
- Function calling reliability
- Latency for interactive use
- Cost with large menu in context

1.4 Edge Cases

How does your design handle:

- User changes mind mid-order ("actually, pickup instead")
- Item not on menu
- Modify previous order item

- Uncovered delivery location

Part 2: Implementation

2.1 Agent Structure

Agent	Purpose	Entry Condition
Greeting	Welcome, determine intent	Start
Location	Collect & validate address	Intent = order
Order	Take order items	Location valid
Checkout	Summarize & close	Order complete

2.2 Suggested Tools

```python
# Location Agent def check_delivery_district(district: str) -> dict: """Returns: {covered: bool, delivery_fee: float, estimated_time: str}""" # Order Agent def search_menu(query: str) -> list[dict] def get_item_details(item_id: str) -> dict def add_to_order(item_id: str, quantity: int, notes: str = "") -> dict def get_current_order() -> dict # Checkout Agent def calculate_total() -> dict def confirm_order() -> dict
```

2.3 Menu Requirements

100+ items across categories:

- Main dishes (30+)
- Appetizers (20+)
- Beverages (20+)
- Desserts (15+)
- Sides (15+)

Each item: `id`, `name_ar`, `name_en`, `price`, `category`, `description_ar`

2.4 Logging Requirements

Log to console or file:

- Agent transitions with timestamp

- Tool calls with parameters and response
- Context size (token estimate) at each handoff
- Any context truncation

```
[14:23:01] HANDOFF: greeting → locati
on [14:23:01] CONTEXT: Transferred 3
messages (est. 245 tokens) [14:23:01]
MEMORY: {customer_name: "أحمد", inten
t: "delivery_order"} [14:23:15] TOOL:
check_delivery_district({district: "ا
لنرجس"}) [14:23:15] TOOL_RESULT: {cov
ered: true, delivery_fee: 15}
```

2.5 Technical Requirements

- Any agent framework (OpenAI Agents SDK, LangGraph, Livekit, Pipecat, custom)
- Runnable locally with clear setup instructions
- Include `requirements.txt`
- Provide `.env.example`

Part 3: Video Explanation

Record a YouTube video (unlisted fine) covering:

1. **Architecture Walkthrough** - Agent flow, handoff triggers
2. **Context Management** - Message structure, menu handling, show handoff logs
3. **Tradeoffs** - Alternatives considered, what you'd change, voice considerations
4. **Live Demo** - Complete order flow with visible logs

Evaluation Focus

- **Handoff Design** - Clean boundaries, appropriate context transfer
- **Context Management** - Token budgeting for both goal understanding and TTFT, message role choices, menu handling
- **Arabic Quality** - Natural conversation, appropriate formality
- **Code Quality** - Readable, modular
- **Reasoning** - Tradeoff articulation in design doc and video

Hints

- Menu is intentionally large. Don't put all 100 items in system prompt.
- Think: when does agent need full menu vs search results?

- More context isn't always better
- We care about WHY you chose your framework, not which one
- Feel free to use the free tier on https://openrouter.ai/

Bonus Points (Optional)

Address any of the following to demonstrate production-readiness:

- **Deployment Strategy:** Where and how would you deploy this? regional considerations for Saudi latency and Saudi data residency requirements
- **Model Routing:** Would you use different LLMs for different agents? Could Greeting use a smaller/faster model? Justify your choices.
- **Arabic Nuances:** How do you handle diacritics for TTS, dialect mixing (MSA system + عامية user), and code-switching ("أبي large meal")?
- **Human Escalation:** At what point does the system hand off to a human agent? How do you preserve context for that handoff?
- **Cost Modeling:** Estimate the cost per completed order (LLM tokens + infrastructure). Where would you optimize first?