Restaurant Automation Bot – Full Technical Documentation

1. Project Overview

The **Restaurant Automation Bot** is an AI-powered, multi-agent automation system built using **n8n**, **Telegram**, **LLMs**, and **Google Sheets**. It is designed to fully automate restaurant customer interactions, order management, and staff coordination through conversational AI.

The system acts as a virtual restaurant assistant that can:

- Answer customer questions
- Take and manage orders
- Generate and track order IDs
- Communicate orders to staff
- Update order statuses in real time

The project is production-oriented and designed to be easily customized for different restaurants.

---

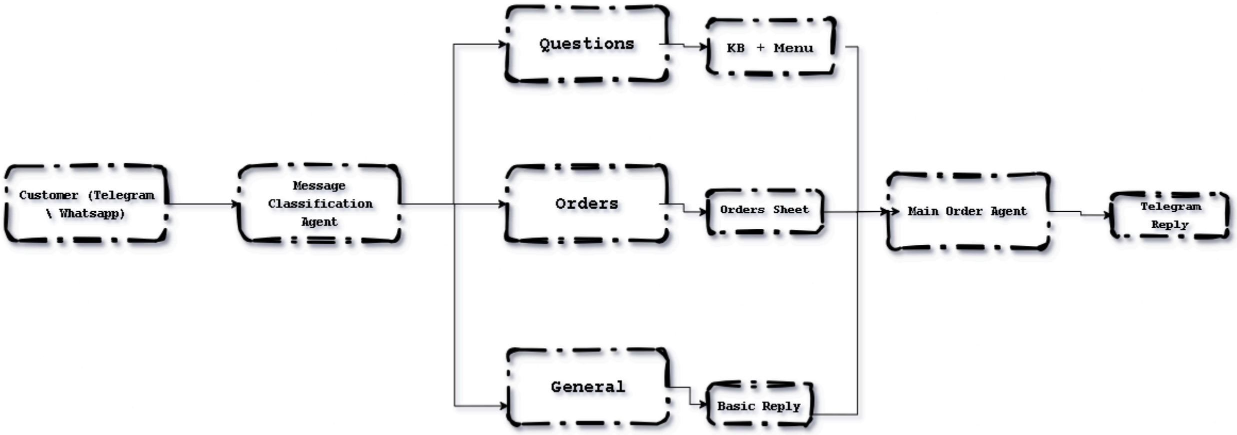
2. Core Objectives

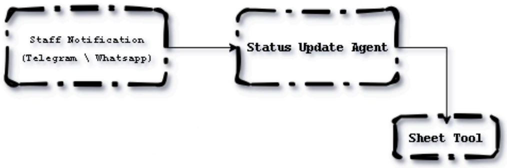
- Reduce manual workload on restaurant staff
- Provide 24/7 automated customer support
- Centralize order management in a simple backend (Google Sheets)
- Enable real-time communication between customers and staff
- Support Arabic (Egyptian dialect) and English

---

3. System Architecture Overview

- The following schematic illustrates the logical workflow of the Restaurant Automation Bot.



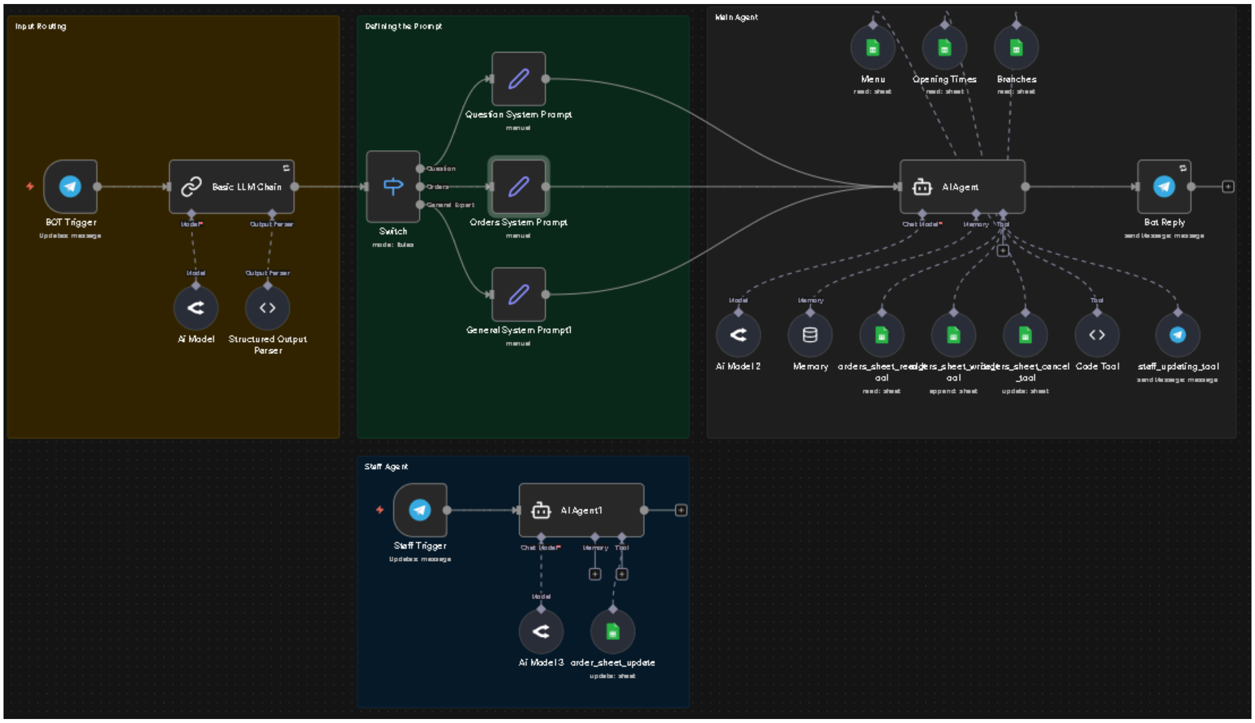


The system is composed of **four main layer**:

1. **Input Layer (Telegram Bots)**
2. **AI Routing & Decision Layer**
3. **Specialized AI Agents**
4. **Backend & External Tools (Google Sheets, Staff Bot)**

All logic is orchestrated using **n8n workflows**.

- The following diagram provides a high-level overview of the complete n8n workflow, illustrating how customer messages are routed through AI agents, backend services, and staff automation processes.



4. Input Layer

4.1 Customer Bot (Telegram)

- Entry point for customer messages
- Receives all text messages from users
- Sends AI-generated replies back to customers

4.2 Staff Bot (Telegram)

- Used internally by restaurant staff
- Receives short operational messages such as:

    - "order 123 is ready"
    - "order 456 delivered"
- Triggers order status updates automatically

---

5. AI Routing & Classification

5.1 Message Classification Agent

A **LangChain LLM Chain** classifies each incoming customer message into one of three categories:

- **Questions Expert** – Menu, branches, opening times, general inquiries
- **Orders Expert** – New orders, cancellations, order tracking
- **Generalist** – Greetings and casual messages

The output is a structured JSON response:

```
{
  "employee": "Orders Expert"
}
```

5.2 Switch Node Routing

Based on the classification result, the message is routed to the appropriate system prompt and AI agent.

---

6. Specialized AI Agents

6.1 Orders Expert Agent

Responsible for full order lifecycle management.

Capabilities:

- Collect customer details (Name, Phone, Address)
- Assist with menu selection
- Calculate total price
- Add delivery fee (10%)
- Generate unique 6-digit Order ID

- Store orders in Google Sheets
- Cancel orders using Order ID
- Retrieve order details by ID
- Notify staff after order confirmation

Key Rules:

- All required data must be collected before order submission
- Order details must be confirmed explicitly
- Only one Order ID is generated per order

6.2 Questions Expert Agent

Handles customer inquiries using the restaurant knowledge base.

Capabilities:

- Answer menu-related questions
- Provide prices
- Share opening times and branch locations
- Respond in the same language used by the customer

6.3 Generalist Agent

Handles general messages such as greetings or unclear inputs.

---

7. Language & Localization Handling

- Automatically detects the user’s language
- Responds in:
  - English
  - Arabic (Egyptian Dialect)
- Prices displayed in:
  - USD for English
  - EGP (جنيه) for Arabic

---

8. Memory Management

A **Memory Buffer Window** is used to:

- Maintain short conversation context
- Support multi-step order collection
- Improve conversational continuity

Memory is scoped per Telegram chat ID.

---

9. Backend & Data Storage (Google Sheets)

Google Sheets acts as the primary backend database.

9.1 Orders Sheet

Columns:

- Time
- Order
- Price
- Order ID
- Customer Name
- Phone Number
- Address
- Status

Possible Status Values:

- In Kitchen
- Being Served
- Delivered
- Cancelled

9.2 Menu Sheet

Stores menu items and prices.
Used by AI agents to dynamically calculate orders.

9.3 Opening Times Sheet

Contains restaurant working hours.

9.4 Branches Sheet

Contains restaurant branch locations and details.

---

10. Order ID Generation

- Generated using a JavaScript-based code tool
- Format: Random 6-digit number
- Stored in the sheet and reused consistently in messages

---

11. Staff Workflow Automation

11.1 Staff Trigger

Staff messages are monitored through a separate Telegram bot.

11.2 Status Update Agent

An AI agent extracts:

- Order ID
- Order status

And updates only the **Status** column in the Orders Sheet.

Allowed statuses:

- Being Served
- Delivered
- Cancelled

If the message is unclear, no action is taken.

---

12. Error Handling & Safeguards

- No order is cancelled without a valid Order ID
- No status update without explicit Order ID
- If an order is not found, the user is informed
- AI is instructed not to guess missing information

---

13. Customization Options

The system can be customized for:

- Different restaurants
- Different currencies
- Different languages
- WhatsApp, Web Chat, or Website integration
- Alternative databases (Airtable, Supabase, PostgreSQL)

---

14. Deployment Requirements

- n8n (Cloud or Self-hosted)
- Telegram Bot Tokens
- Google Sheets API access
- OpenRouter API key (LLM models)

---

15. Limitations & Notes

- Order ID collision is statistically rare but possible
- Google Sheets API limits apply

- Designed for small to medium restaurant operations

---

16. Testing & Example Conversations

This section demonstrates real end-to-end behavior of the system through actual Telegram chat interactions. The screenshots below validate that the workflow, AI routing, and backend automation operate correctly in real usage scenarios.

16.1 Greeting & Menu Showcase

This test case shows the initial interaction between the customer and the bot.

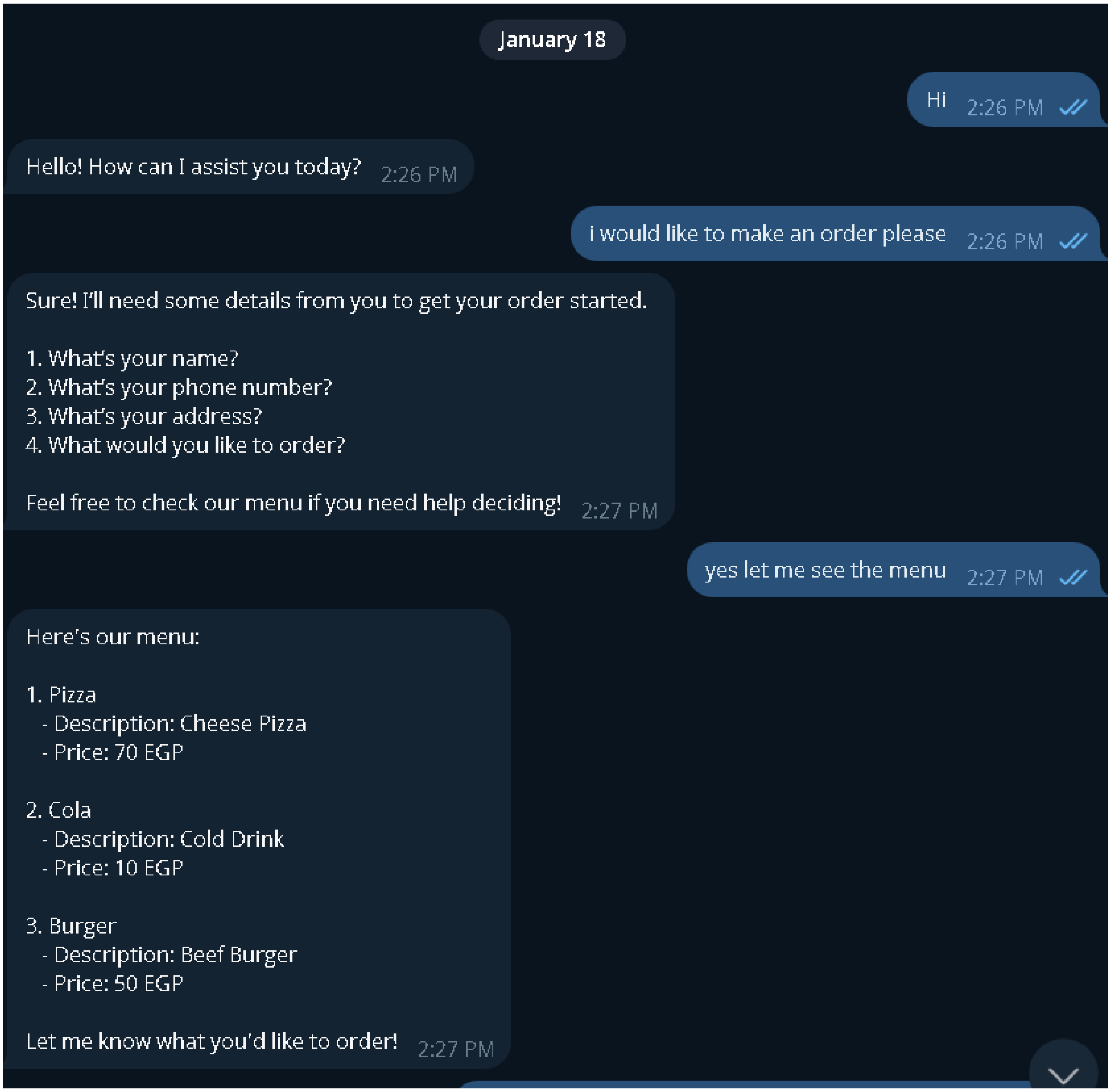
Scenario covered:

- User greeting (e.g., Hi / Hello)
- Automatic routing to the Generalist or Questions Expert
- Menu retrieval and presentation
- Language detection and localized response

Expected behavior:

- Friendly greeting
- Clear menu listing
- Prices shown in the appropriate currency

Screenshot:



January 18

Hi 2:26 PM

Hello! How can I assist you today? 2:26 PM

i would like to make an order please 2:26 PM

Sure! I'll need some details from you to get your order started.

1. What's your name?
2. What's your phone number?
3. What's your address?
4. What would you like to order?

Feel free to check our menu if you need help deciding! 2:27 PM

yes let me see the menu 2:27 PM

Here's our menu:

1. Pizza
   - Description: Cheese Pizza
   - Price: 70 EGP

2. Cola
   - Description: Cold Drink
   - Price: 10 EGP

3. Burger
   - Description: Beef Burger
   - Price: 50 EGP

Let me know what you'd like to order! 2:27 PM

16.2 Order Taking & Price Calculation

This test case demonstrates the order creation flow.

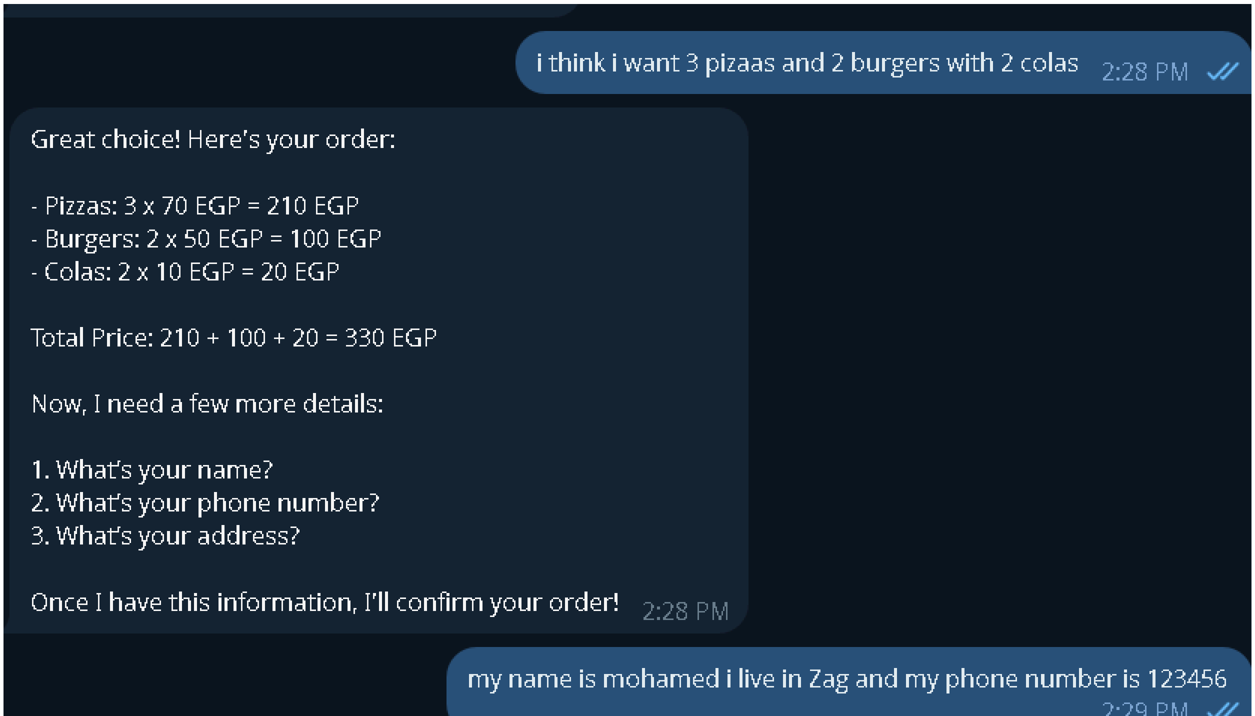
Scenario covered:

- User selects menu items
- Orders Expert agent is activated
- Item prices are retrieved from the Menu sheet
- Total price is calculated
- Delivery fee (10%) is added
- Bot requests required customer information

Expected behavior:

- **Clear item breakdown**
- **Accurate total calculation**
- **Explicit request for customer details (Name, Phone, Address)**

Screenshot:



i think i want 3 pizaas and 2 burgers with 2 colas 2:28 PM

Great choice! Here's your order:

- Pizzas: 3 x 70 EGP = 210 EGP
- Burgers: 2 x 50 EGP = 100 EGP
- Colas: 2 x 10 EGP = 20 EGP

Total Price: 210 + 100 + 20 = 330 EGP

Now, I need a few more details:

1. What's your name?
2. What's your phone number?
3. What's your address?

Once I have this information, I'll confirm your order! 2:28 PM

my name is mohamed i live in Zag and my phone number is 123456 2:29 PM

16.3 Final Order Review & Confirmation

This test case shows the final confirmation stage before order submission.

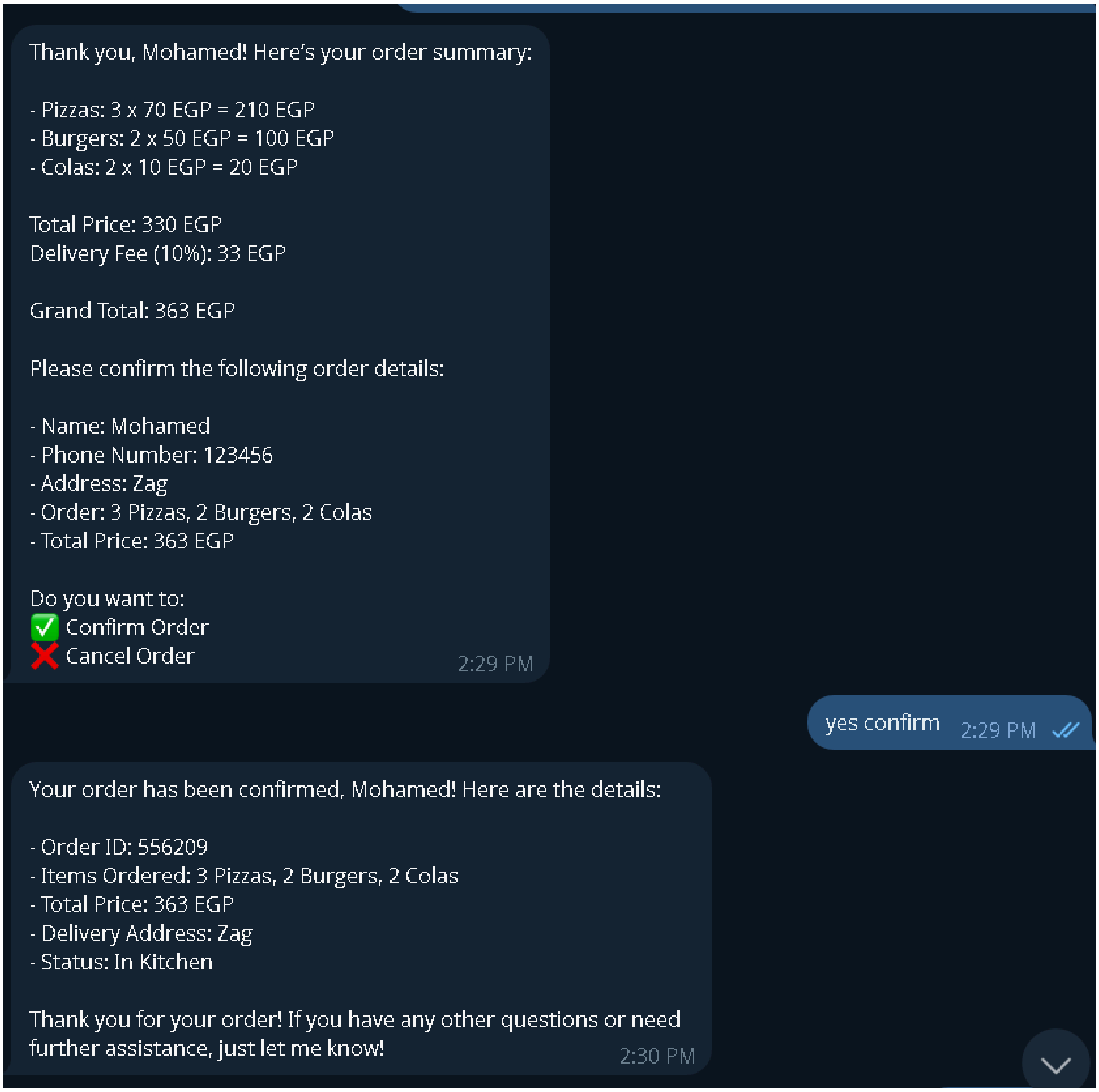
Scenario covered:

- Full order summary
- Generated unique Order ID
- Final price including delivery
- Confirmation and cancellation options
- Order persistence in Google Sheets
- Staff notification trigger

Expected behavior:

- Clear and complete order summary
- Visible Order ID
- Explicit confirmation options

Screenshot:



Thank you, Mohamed! Here's your order summary:

- Pizzas: 3 x 70 EGP = 210 EGP
- Burgers: 2 x 50 EGP = 100 EGP
- Colas: 2 x 10 EGP = 20 EGP

Total Price: 330 EGP
Delivery Fee (10%): 33 EGP

Grand Total: 363 EGP

Please confirm the following order details:

- Name: Mohamed
- Phone Number: 123456
- Address: Zag
- Order: 3 Pizzas, 2 Burgers, 2 Colas
- Total Price: 363 EGP

Do you want to:
✅ Confirm Order
❌ Cancel Order
2:29 PM

yes confirm 2:29 PM

Your order has been confirmed, Mohamed! Here are the details:

- Order ID: 556209
- Items Ordered: 3 Pizzas, 2 Burgers, 2 Colas
- Total Price: 363 EGP
- Delivery Address: Zag
- Status: In Kitchen

Thank you for your order! If you have any other questions or need further assistance, just let me know!
2:30 PM

17.Project Status

- Architecture: ✅ Complete
- Core Features: ✅ Implemented
- Production Ready: ✅ Yes
- Freelance Ready: ✅ Yes

---

18. Conclusion

This project demonstrates a **production-grade AI automation system** for restaurants, combining conversational AI, workflow automation, and real-world business logic into a scalable and customizable solution.

It is suitable for:

- Freelance delivery
- Portfolio presentation
- Client customization
- SaaS-style extensions