ChatBot Implementation & Deployment

---

1. Meta (WhatsApp) Developer Setup

To enable messaging, follow these steps to configure the Meta Cloud API:

- **Create App:** Visit Meta for Developers, create a "Business" app, and add the **WhatsApp** product.
- **Phone Number:** Link a business phone number to your WhatsApp Business Account.
- **Webhook Configuration:**
  - **Callback URL:** Enter the **Production URL** from your n8n **Meta Webhook Trigger** node.
  - **Verify Token:** Set this to verify_tracker.
  - **Fields:** Subscribe to messages.
- **WhatsApp Template:** Create a message template named liquidation_alert in the Meta Business Suite for the reminder system.
- **Access Token:** Generate a **Permanent System User Access Token** with whatsapp_business_messaging permissions.

---

2. Database Schema (Neon PostgreSQL)

Execute these SQL commands in your Neon console to create the **8 required tables** with full constraints:

SQL

```
-- 1. User Profiles
CREATE TABLE users (
    phone_number VARCHAR(15) PRIMARY KEY,
    name VARCHAR(100),
    headquarter VARCHAR(100),
    area VARCHAR(100),
    zone VARCHAR(100)
);
```

```sql
-- 2. Authorized Access Control
CREATE TABLE "TA_PhoneNumber" (
    phone_number VARCHAR(15) PRIMARY KEY
);


-- 3. Session Management
CREATE TABLE conversation_states (
    phone_number TEXT PRIMARY KEY,
    current_step TEXT NOT NULL,
    data JSONB NOT NULL,
    last_updated TIMESTAMP DEFAULT now()
);


-- 4. Main Inventory Storage
CREATE TABLE liquidation_records (
    phone_number TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT now(),
    employee_name TEXT,
    hq TEXT,
    zone TEXT,
    area TEXT,
    products JSONB,
    record_date DATE DEFAULT CURRENT_DATE,
    CONSTRAINT liquidation_records_pkey PRIMARY KEY (phone_number, record_date)
);


-- 5. Updated Records History
CREATE TABLE updated_record (
    phone_number TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT now(),
    employee_name TEXT,
    hq TEXT,
    zone TEXT,
    area TEXT,
    products JSONB,
    record_date DATE DEFAULT CURRENT_DATE,
    CONSTRAINT liquidation_records_backup_pkey PRIMARY KEY (phone_number, record_date)
);

-- 6. Chat Logs (Transactional History)
CREATE TABLE chat_logs (
    phone_number TEXT,
    employee_name TEXT,
    hq TEXT,
    zone TEXT,
    area TEXT,
    products JSONB,
    created_at TIMESTAMP,
    record_date DATE
);


-- 7. Message Deduplication
CREATE TABLE processed_messages (
    message_id TEXT PRIMARY KEY,
    phone_number TEXT NOT NULL,
    processed_at TIMESTAMP DEFAULT now(),
    created_at TIMESTAMP DEFAULT now()
);


-- 8. WhatsApp Message Logs
CREATE TABLE whatsapp_message_logs (
    id SERIAL PRIMARY KEY,
    phone_number TEXT,
    message_type TEXT,
    status TEXT,
    message_id TEXT UNIQUE,
    created_at TIMESTAMP DEFAULT now()
);
```

---

3. Logical Flow

Main Bot Flow

1. **Trigger:** User sends a message via WhatsApp.
2. **Validation:** System checks for duplicate messages and authorizes the sender's phone number against the database.
3. **State Loading:** System retrieves the user's progress. If the session is older than 10 minutes, it is reset.
4. **Information Gathering:**
   - **Profile:** Collects Name, HQ, Zone, and Area (skipped for returning users).
   - **Context:** User selects the record date (Today, Yesterday, or Custom).
   - **Inventory:** User selects Product Family -> Product Name -> SKU.

5. **Stock Entry:** User enters Opening Stock and Liquidation Quantity; the system validates that liquidation does not exceed opening stock.
6. **Review & Submit:** User reviews entries, makes any necessary edits, and confirms.
7. **Finalization:** Data is saved, the session is cleared, and a summary is sent via WhatsApp.

Daily Reminder Flow

1. **Trigger:** Scheduled Cron runs daily at 4:30 PM.
2. **Data Check:** Queries for registered users who have not submitted a record for the current date.
3. **Action:** Sends a "Free Text" reminder if active within 24 hours, otherwise sends a "Template" message.

---

4. Deployment Guide

A. Hosting Options

Option A: n8n Cloud (Managed Service)

This is a **Ready-to-Go** managed service with zero server maintenance.

- **Setup:** Sign up at n8n.io and select a plan based on your expected message volume.
- **Instance:** n8n provides a dedicated managed URL (e.g., yourcompany.n8n.cloud).
- **Workflow Setup:** Create a new workflow, select **Import from File**, and upload your .json files.
- **Security:** Enable MFA (Multi-Factor Authentication) to protect database credentials.

Option B: Amazon Lightsail (Self-Hosted/Docker Compose)

- **Hardware Requirements:** Use a server with **2 vCPUs, 2 GB RAM, and 60 GB SSD**.
- **Instance Setup:** Launch an **Ubuntu 22.04 LTS** instance and attach a **Static IP**.
- **Docker Installation:** Connect via SSH and install Docker and Docker Compose.
  ```
  sudo apt update && sudo apt upgrade -y
  sudo apt install docker.io docker-compose -y
  sudo systemctl enable --now docker
  ```

- **Docker Compose Setup:** Create a directory for n8n and configure the docker-compose.yml file.

Bash

```
mkdir ~/n8n && cd ~/n8n
nano docker-compose.yml
```

Paste the following content (replace your-domain.com with your actual domain):

YAML

```
version: '3.8'
services:
  n8n:
    image: n8nio/n8n:latest
    restart: always
    ports:
      - "5678:5678"
    environment:
      - N8N_HOST=your-domain.com
      - N8N_PORT=5678
      - N8N_PROTOCOL=https
      - NODE_ENV=production
      - WEBHOOK_URL=https://your-domain.com/
    volumes:
      - ./n8n_data:/home/node/.n8n
```

Start the container: sudo docker-compose up -d.

- **Networking & SSL:** Open ports **80, 443, and 5678** in the firewall. Use **Nginx** and **Certbot** for the mandatory HTTPS reverse proxy.

B. Database Hosting (Neon PostgreSQL)

- **Provisioning:** Log into Neon.tech and create a new project.
- **Schema Execution:** Open the Neon SQL Editor and execute the 8-table SQL script provided in this doc.
- **Connection:** Copy connection parameters and set **SSL** to **Require** in the n8n credential settings.

---

5. Final Go-Live Steps

1. **Credentials Configuration:** Set up **Postgres** (Neon) and **WhatsApp API** (Meta) credentials in n8n.
2. **Import Workflow:** Open n8n, create a new workflow, select **"Import from File"**, and upload the JSON code for both workflows.

3. **Update API Endpoints:** Ensure all **HTTP Request** nodes use your unique Phone Number ID (e.g., 925047117349142) in the URL: https://graph.facebook.com/v22.0/[YOUR_PHONE_ID]/messages.
4. **Meta Webhook Update:** Copy the **Production Webhook URL** from n8n to Meta's settings using verify_tracker as the token.
5. **Activate:** Toggle both workflows to **Active**.