# 

# AI AGENT DEVELOPMENT INTERNSHIP PROJECT

## LeadMate – CRM AI Agent

Submitted by: Hevanth Kumar

Role: AI Agent Developer (Part-Time)

Submission Date: 2nd July 2025

Institution / Organization: [Your Organization Name Here]

# CERTIFICATE OF COMPLETION

This is to certify that

Hevanth Kumar

has successfully completed the AI Agent Internship Project titled

“LeadMate – CRM AI Agent”

under the guidance of [Mentor Name/Organization] during the Summer 2025 session.

Date: 2nd July 2025

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AI Agent Development Assignment – Internship Project

🚀 LeadMate – CRM AI Agent

Internship Project Submission by Hevanth Kumar

Project Title: LeadMate – AI Assistant for CRM Follow-Ups

Submitted by: Hevanth Kumar

Internship Role: Artificial Intelligence – AI Agent Developer

Assignment: Build a Simple AI Agent using 4-Layer Prompt Architecture

Submission Date: 2nd July 2025, 10:00 PM

# 💻 Mockup Preview – LeadMate UI (Concept Design)

----------------------------------------  
| LEADMATE CRM AI DASHBOARD |  
----------------------------------------  
| 🔍 Search Leads [\_\_\_\_\_\_\_\_\_\_] |  
| 🗓️ Today's Top 3 Leads: |  
|--------------------------------------|  
| ✅ Priya Sharma |  
| ➤ Last Action: Visited pricing |  
| ➤ Suggested Action: Call today |  
|--------------------------------------|  
| ⚠️ Ramesh Nair |  
| ➤ Last Action: Opened email |  
| ➤ Suggested Action: Send email |  
|--------------------------------------|  
| ❗ Anjali Rao |  
| ➤ Last Action: Demo booked |  
| ➤ Suggested Action: Follow-up |  
----------------------------------------  
| 📩 Schedule Follow-up | 🔄 Refresh |  
----------------------------------------

\*This simple dashboard sketch shows how the AI Agent can help sales reps quickly identify which leads to follow up on, based on recent interactions.\*

# SECTION 1: BASIC DETAILS

- Name: Hevanth Kumar

- AI Agent Title / Use Case: LeadMate – An AI Agent to help sales reps prioritize leads and manage follow-ups effectively using CRM data

# SECTION 2: PROBLEM FRAMING

2.1. What problem does your AI Agent solve?  
In most growing businesses, salespeople are juggling too many leads. Important ones slip through the cracks, and follow-ups are often missed. My agent solves this by reminding the user who to follow up with, when, and why.

2.2. Why is this agent useful?  
It saves time and prevents missed opportunities. With smart suggestions based on past data, it helps salespeople focus their energy on the right people at the right time.

2.3. Who is the target user?  
Sales reps or customer success teams working in fast-paced environments, especially in startups and small businesses.

2.4. What not to include?  
It won’t handle messaging or campaigns. It’s focused strictly on suggesting follow-up actions using CRM data.

# SECTION 3: 4-LAYER PROMPT DESIGN

## 🔹 3.1 INPUT UNDERSTANDING

Prompt:  
"What are my top leads for today?"

What is this prompt responsible for?  
It helps the AI know the user's main intent: finding out who to follow up with.

Example Output:  
Here are your top 3 leads for today:  
- Priya Sharma: Opened pricing page. ➤ Call today.  
- Ramesh Nair: Opened last email. ➤ Send follow-up email.  
- Anjali Rao: Booked demo but no response. ➤ Check-in.

## 🔹 3.2 STATE TRACKER

Prompt:  
"Keep track of last contact date, lead score, and last action for each lead."

How does this help the agent “remember”?  
This allows the agent to build context and avoid repeating the same suggestions.

How memory is simulated:  
Used internal variables like `last\_contacted`, `score`, `action\_needed`.

## 🔹 3.3 TASK PLANNER

Prompt:  
"Evaluate lead score, engagement activity, and time since last contact. Then prioritize top 3 and suggest next steps."

Steps it performs:  
- Scores each lead  
- Sorts by urgency and engagement  
- Determines action (call, email, follow-up)

How complexity is handled:  
Used rule-based logic for scoring + branching decisions depending on engagement level.

## 🔹 3.4 OUTPUT GENERATOR

Prompt:  
"Present the top leads in a friendly, clear format with action recommendations. Use bullet points, emojis for urgency, and avoid long paragraphs."

What output format did I aim for?  
Concise, direct, and human-readable messages.

Special behaviors:  
Tone-controlled suggestions and fallback prompts for unclear queries.

# SECTION 4: CHATGPT EXPLORATION LOG

| Attempt | Prompt Variant | What Happened | What I Changed | Why I Changed It |  
|--------|----------------|---------------|----------------|------------------|  
| 1 | "Top 3 leads today" | Response too generic | Added rules for scoring | To make it smarter |  
| 2 | "Show leads needing action" | Mixed up urgency | Added engagement recency | For accurate priority |  
| 3 | "Summarize leads" | Output too long | Simplified response format | Better user experience |

# SECTION 5: OUTPUT TESTS

Test 1 – Clear Input  
Input: "Top 3 leads today"  
Output:  
1. Priya – Opened pricing page. ➤ Call now.  
2. Ramesh – Clicked email, no reply. ➤ Send email.  
3. Anjali – Demo booked. ➤ Check-in.

Test 2 – Vague Input  
Input: "Help with leads?"  
Output:  
"Would you like a summary of who to follow up with or lead status updates?"

Test 3 – No Input  
Input: ""  
Output:  
"I didn’t get that. You can ask things like: ‘Top leads today’ or ‘Who to follow up with?’"

# SECTION 6: REFLECTION

6.1. Hardest Part  
Getting the agent to sound smart without making it too complicated. Balancing simplicity and usefulness was tricky.

6.2. Most Enjoyable Part  
Seeing the prompts improve with each iteration and watching the agent give practical suggestions.

6.3. What I'd Improve with More Time  
I would connect it with actual CRM sample data and test it with real-time inputs.

6.4. What I Learned  
Prompt design is not just writing – it’s like thinking in code. Structure and clarity are everything.

6.5. When I Felt Stuck  
I asked ChatGPT what was wrong, reviewed each layer, and iterated. Small tweaks often solved big issues.

# SECTION 7: HACK VALUE (Optional)

- Simulated memory tracking using role and system messages  
- Built fallback responses for vague inputs  
- Used score-based logic to simulate real-world prioritization

---  
End of Project Submission