

AI Agent Development Assignment  
Submitted by: Manthan Kadam

# Assignment Title: AI Agent for Hidden Energy-Waste Detection in Workplaces

SECTION 1: BASIC DETAILS

...

SECTION 2: PROBLEM FRAMING

...

SECTION 3: 4-LAYER PROMPT DESIGN

...

SECTION 4: CHATGPT EXPLORATION LOG

...

SECTION 5: OUTPUT TESTS

...

SECTION 6: REFLECTION

...

SECTION 7: HACK VALUE

...

## 1. Basic details

Name: Manthan Kadam

AI Agent Title: AI Agent for Hidden Energy-Waste Detection in Workplaces

## 2. PROBLEM FRAMING

### 2.1 What problem does this AI Agent solve?

Most offices, hostels, and workplaces waste a huge amount of energy without realizing it.

There is no simple tool that analyzes everyday usage and points out hidden energy waste.

This agent identifies exactly where electricity is being wasted and how to fix it.

### 2.2 Why is this agent useful?

It helps users reduce electricity bills, improve efficiency, and understand their energy patterns.

Even without smart meters, it gives actionable insights based only on text inputs.

### 2.3 Who is the target user?

Office managers, hostel owners, small businesses, home users, and any workplace looking

to reduce electricity waste without expensive audits.

### 2.4 What is not included?

The agent does not control devices, does not access smart-meter data, and does not perform real-time monitoring. It only analyzes the information provided by the user.

## 3.1. 4-LAYER PROMPT DESIGN

You are the “Input Understanding Module” of an Energy-Waste Detection AI Agent.

Your job is ONLY to identify what type of information the user is providing.

Classify the input into one of the following categories:

1. Appliance List
2. Usage Pattern
3. Location Type (office, hostel, home, shop)
4. Energy Issue Description
5. General Questions
6. Invalid Input

Output a JSON with:

```
{  
  "input_type": "...",
```

```
"details_extracted": "...",  
"missing_information": "..."  
}
```

Do not give solutions. Do not analyze energy waste.  
Your job is ONLY to understand and categorize the input.

## 3.2 STATE TRACKER

You are the “State Tracker Module” of the Energy-Waste Detection AI Agent.

Your job:

- Store the appliances mentioned by the user.
- Store the usage hours.
- Store the location type.
- Add new information to the existing state.
- Identify missing data required for energy audit.

State Format:

```
{  
  "location_type": "",  
  "appliances": {},  
  "usage_pattern": {},  
  "missing_info": []  
}
```

Rules:

- Do NOT perform analysis.
- Do NOT give solutions.
- Only update and maintain the internal state.
- If the user input is invalid, request specific missing information.

## 3. 3. TASK PLANNER

You are the “Task Planner Module” of the Energy-Waste Detection AI Agent.

Your job is to decide WHAT steps should be executed before giving the final answer.

Given the latest input + current state, follow this internal plan:

1. Check if state has required details:
  - location type
  - appliance list
  - usage pattern
2. If something is missing → request it.
3. If information is complete:
  - Identify high-energy appliances.
  - Identify time-waste patterns.
  - Compare usage patterns with standard optimal values.
  - Calculate possible waste areas.
  - Prioritize issues based on severity.
  - Generate improvement steps.
4. Send a structured plan to the Output Generator.

Do NOT generate solutions or text for the user.

ONLY prepare the reasoning steps as structured JSON.

## 3.4. OUTPUT GENERATOR

You are the “Output Generator Module” of the Energy-Waste Detection AI Agent.

Your job is to take the Task Planner’s structured reasoning and convert it into:

- Simple, clear explanations
- Bullet points
- Actionable improvement steps
- Estimated savings (if possible)
- Priority list

Formatting Rules:

- Use short sentences
- Use clean bullet points
- Avoid technical jargon
- Highlight the top 3 major issues
- End with a simple 1–2 line summary

Tone:

Helpful, friendly, and practical. No blame, no negativity.

## 4. CHATGPT EXPLORATION LOG

### Attempt 1 — Initial Prompt (Too Broad)

Make an AI that detects energy waste in offices.

### Attempt 2 — Added Layer Separation (Better But Incomplete)

I want 4 different prompts: one for Input Understanding, one for State Tracking, one for Task Planning, and one for Output.

### Attempt 3 — Final Professional Version (Successful)

You are the “Output Generator Module” of the Energy-Waste Detection AI Agent.

Your job is to take the Task Planner’s structured reasoning and convert it into:

- Simple, clear explanations
- Bullet points
- Actionable improvement steps
- Estimated savings (if possible)
- Priority list

Formatting Rules:

- Use short sentences
- Use clean bullet points
- Avoid technical jargon
- Highlight the top 3 major issues
- End with a simple 1–2 line summary

Tone:

Helpful, friendly, and practical. No blame, no negativity.

## 5. OUTPUT TESTS

### Test 1 —

Normal Input

My office has 5 computers, 2 ACs, 1 fridge.  
AC runs from 9am to 6pm. Computers stay ON even during lunch break.

Agent Output:

Your biggest energy waste comes from continuous AC use + computers running idle.

### Test 2 —

Vague Input

Energy is getting wasted in my office. Please help.

Agent Output:

I can help you detect the exact energy waste, but I need more details.

Please tell me:

1. How many appliances do you use?
2. What are their usual operating hours?

3. What type of workplace is this (office, hostel, shop, etc.)?

Once I have this info, I will scan your usage and find hidden waste areas.

I can help you detect the exact energy waste, but I need more details.

Please tell me:

1. How many appliances do you use?

2. What are their usual operating hours?

3. What type of workplace is this (office, hostel, shop, etc.)?

Once I have this info, I will scan your usage and find hidden waste areas.

## Test 3 — Invalid Input / Empty Input

Input:

("")

Agent Output:

I couldn't understand your input.

To analyze energy waste, please provide at least one of the following:

- Appliance list
- Usage hours
- Location type
- Any specific energy issue you've noticed

Example:

"My hostel has 20 lights, 1 water heater, lights stay on all night."

## 7. REFLECTION

### **6.1 What was the hardest part of this assignment?**

The hardest part was realizing that building an AI Agent is not the same as writing a single chatbot prompt.

Understanding the separation between Input Understanding, State Tracking, Task Planning, and Output Generation was initially confusing.

I struggled to convert my ideas into clean, modular components.

Once I broke the problem into small layers, the entire architecture finally started making sense.

### **6.2 What part did you enjoy the most?**



I enjoyed seeing how the agent became smarter after every iteration.  
Especially when the prompts became clearer and the output turned into something usable and realistic.

It felt good to watch the agent classify inputs, store state, and then generate actionable energy-saving insights.

The feeling of “I actually built an intelligent system” was exciting.

### **6.3 If given more time, what would you improve or add?**

I would add more specialized detection logic for factories, hospitals, and hostels.  
I also want to include a cost-saving calculator and maybe a small database of “optimal appliance usage benchmarks.”

Another improvement would be a simple UI where users can upload energy data instead of typing everything manually.

### **6.4 What did you learn about ChatGPT or prompt design?**

I learned that ChatGPT follows instructions very precisely — but only if the prompts are structured clearly.

Layering the prompts helped me understand how real AI agents think internally.  
I also understood the importance of iteration: first version is always weak, but improving it step-by-step gives much smarter results.

Good prompt engineering is basically good system design.

### **6.5 Did you ever feel stuck? How did you handle it?**

Yes, especially at the beginning when I didn’t know how to design each module.  
I solved it by breaking the problem down: first the use-case, then the agent layers, then the tests.

Whenever I felt lost, I asked ChatGPT small, direct questions instead of big ones.  
That made the problem much easier to handle and helped me complete the assignment with clarity.

# 1. HACK VALUE

Although this was a simple agent assignment, I added a few elements that go beyond the basic requirements:

- 1. Added a structured JSON format for both Input Understanding and Task Planning.  
This makes the agent easier to integrate into real systems in the future.
- 2. Designed the State Tracker to identify “missing information,” not just store inputs.  
This improves the agent's ability to guide the user instead of getting confused.
- 3. Introduced priority-based energy waste detection (Top 3 major issues).  
This makes the output more actionable and practical for workplaces.

These small enhancements give the agent real-world usability and show how it could grow into a full SaaS product.

## Rubric for Evaluation (Work Simulation Style)

Dimension	Weight	What Good Looks Like	Red Flags
Problem Framing	20%	Clear, original use-case: Hidden Energy-Waste Detection in workplaces. Well-scoped agent behavior. Shows user-centric thinking.	Generic chatbot or vague problem. Unclear target users or environment.

Prompt Architecture Quality	25%	Each of the 4 prompts reflects distinct responsibilities (Input Understanding, State Tracker, Task Planner, Output Generator). Modular, testable, concise.	Overlapping prompts, verbose, or non-modular. Responsibilities unclear.
Exploration Quality (AI Collaboration)	20%	Documented iterations: 3 attempts showing debugging, improvement, and refinement. Effective use of ChatGPT as collaborator.	One-shot prompt. No iteration. No debugging or alternative exploration.
Output Clarity & Functional Coverage	15%	Agent outputs actionable energy-saving advice, bullet points, priority-based. Works across normal, vague, and invalid inputs.	Output trivial, unclear, repetitive, or structurally broken.
Documentation of Process	15%	Sections 1–7 complete. Reflection and reasoning are honest. Journey clearly documented.	Only final output shown. No reflection. No reasoning trail.
Initiative / Hack Value	5%	Added missing info detection, JSON structured reasoning, priority-based improvements. Shows potential for real-world SaaS implementation.	Minimum effort. No extra features or initiative shown.

## Beginner-Friendly WBS: AI Agent for Hidden Energy-Waste Detection in Workplaces

**Total Estimated Time:** 8–10 hours

**Success Formula:** Curiosity + Clear Thinking + Documenting the Journey

---

## **PHASE 0: Orientation – “What am I doing?” (60–75 min)**

Task	Time
0.1 Read assignment brief, structure, and expected outputs	10 min
0.2 Watch/read basic explainer on prompts and AI response logic	15–20 min
0.3 Review linked docs (architecture, prompts, examples) and take notes	30–45 min
0.4 Summarize: “What are the 4 steps in AI Agent design?” in your own words	10 min

---

## **PHASE 1: Choose Use Case + Imagine the Agent (90–120 min)**

Task	Time
1.1 Brainstorm 2–3 situations where AI could help (studies, planning, energy efficiency, etc.)	20 min

1.2 Choose one: Hidden Energy-Waste Detection. Write short note: user + problem solved	20 min
1.3 Create example user inputs (e.g., “My office has 5 computers, lights stay on all night”)	15 min
1.4 Think about internal agent logic (no code, just reasoning steps)	20 min
1.5 Define what the agent should NOT do (e.g., control devices, access smart meters)	10 min

---

## PHASE 2: Design the 4-Layer Prompts (2–2.5 hours)

Task	Time
2.1 Input Understanding – Classify user inputs and extract details	20–30 min
2.2 State Tracker – Store inputs, simulate memory, track missing info	20–30 min
2.3 Task Planner – Break request into structured reasoning steps	30–40 min
2.4 Output Generator – Define final reply format, tone, actionable steps	30–40 min

---

## PHASE 3: ChatGPT Exploration & Debugging (2–2.5 hours)

Task	Time
3.1 Run each prompt layer in ChatGPT, copy outputs	20–30 min
3.2 Note issues and ask ChatGPT to improve prompts	30–45 min
3.3 Iterate 2–3 versions for each layer	45–60 min
3.4 Save all prompts + outputs in a table/doc	30 min

---

## PHASE 4: Write Final Submission Document (2–2.5 hours)

Task	Time
4.1 Sections 1–2: Title, Problem Framing, Use Case, Agent Scope	20–30 min
4.2 Section 3: Final version of 4-layer prompts with comments	30–40 min
4.3 Section 4: Exploration Log (attempts, changes, rationale)	30–40 min

4.4 Section 5: Output Tests (Normal, Vague, Invalid Inputs)	20 min
4.5 Section 6: Reflection – Learnings, struggles, improvements	30 min
4.6 Section 7: Hack Value / Bonus Features	10–15 min

---

**Total Estimated Time:** ~9 hours

*(May vary depending on depth of thought, iterations, and reflection.)*

**Tips for Success:**

- Keep a scratchpad for prompt ideas and trial outputs.
- Treat ChatGPT like a teammate, not a magic box.
- Focus on documenting iterations, not speed.
- Use analogies: each prompt layer is like a department in a company.

