# Extendable Tools (Orchestrator)

To support future requirements, the orchestrator system is designed to be easily extendable. Developers can add new tools (functions) with minimal effort, allowing the AI orchestrator to automatically detect and use them based on user input.

## 1. Create a New Function

- The function must accept a single parameter: `user\_input`.  
- Write the tool’s logic inside the function to generate the required response.

def calculate\_profit(user\_input: str) -> str:  
 """Calculates profit based on cost and selling price in the input."""  
 return "Profit is ₹30"

## 2. Add a Description in system\_prompt

- In the orchestrator’s `system\_prompt`, list the function with a short description.  
- This helps the LLM (language model) understand when to use the tool.

system\_prompt = """  
Functions:  
- check\_in\_document: For company or document-related questions.  
- check\_in\_db: For product prices, rates, or system data.  
- suggest\_clothing\_combination: For outfit or clothing suggestions.  
- get\_current\_date: To get the current date.  
- calculate\_profit: For profit calculations.  
- your\_new\_function\_name: Describe the purpose here.  
"""

## 3. Register the Tool in tool\_labels Dictionary

- Add a key-value pair to the `tool\_labels` dictionary.  
- The key is the function name.  
- The value is the label that will appear in the user-facing response.

tool\_labels = {  
 "check\_in\_document": "Company Info",  
 "check\_in\_db": "Product Rates",  
 "suggest\_clothing\_combination": "Clothing Suggestions",  
 "get\_current\_date": "Current Date",  
 "calculate\_profit": "Profit Calculation",  
 "your\_new\_function\_name": "Your Tool Label",  
}

## 4. That’s It!

- The orchestrator will now automatically detect and use the tool when relevant.  
- No additional changes are needed in the main orchestration logic.

## Example: Adding a Currency Converter Tool

### Step 1: Function

def convert\_currency(user\_input: str) -> str:  
 """Converts currency based on user input."""  
 return "₹100 = $1.20"

### Step 2: Add to system\_prompt

- convert\_currency: For converting between currencies like INR to USD or EUR.

### Step 3: Add to tool\_labels

tool\_labels = {  
 ...  
 "convert\_currency": "Currency Converter",  
}

## Summary Table

|  |  |
| --- | --- |
| Step | Description |
| 1 | Create a function with `user\_input` as the parameter. |
| 2 | Add a description to `system\_prompt`. |
| 3 | Register the tool in `tool\_labels`. |
| 4 | Done! The orchestrator can now use the new tool automatically. |