Water LLM Engine - Function Logic (Layman Version)

Water LLM Engine - Full Function Reference (Layman Level)

get_integration_config(location)

INPUT REQUIRED:

Location name (e.g., 'London')

ACTION TAKEN:

Looks up a configuration database to fetch connection settings like API URLs, SCADA addresses, and weather services for that location.

LOGIC IN SIMPLE TERMS:

Matches the location in the database and returns all required settings for system integration in that city. Think of it as opening the city's operations control sheet.

actuate_asset(command, location)

INPUT REQUIRED:

A command (e.g., 'open_valve') and a location

ACTION TAKEN:

Sends this command to multiple types of control systems: SCADA, MQTT, OPC-UA, or PLC.

LOGIC IN SIMPLE TERMS:

It checks which systems are available for that city and tries to send the command to all of them. It then returns which ones succeeded or failed. Like pressing all available buttons to ensure action is taken.

fetch_weather_data(location)

INPUT REQUIRED:

Location name

ACTION TAKEN:

Contacts the weather service for that city and requests rainfall forecast data.

Water LLM Engine - Function Logic (Layman Version)

LOGIC IN SIMPLE TERMS:

Uses the weather API link configured for that city to get live or forecasted rainfall values. Helps anticipate if overflow might happen soon.

fetch_sensor_data(location)

INPUT REQUIRED:

Location name

ACTION TAKEN:

Calls the live sensors for the city and fetches data like tank fill percentage and water inflow rate.

LOGIC IN SIMPLE TERMS:

Reads real-world measurements from sensors deployed on tanks and pipes. These values are then used in decision-making functions.

call_gpt(prompt, temperature)

INPUT REQUIRED:

Prompt (a question or scenario), and an optional temperature setting for creativity

ACTION TAKEN:

Sends the prompt to GPT-4 (the AI engine) and receives a reply.

LOGIC IN SIMPLE TERMS:

Used to generate smart recommendations, reports, and regulatory advice based on live data or configured scenarios.

calculate_overflow_risk(rain_mm, tank_fill_percent)

INPUT REQUIRED:

Rainfall in mm and how full the tank is (%)

ACTION TAKEN:

Assigns a risk level (LOW, MEDIUM, or HIGH) based on thresholds.

Water LLM Engine - Function Logic (Layman Version)

LOGIC IN SIMPLE TERMS:

Simple if-else logic: more than 20mm rain + tank over 90% = HIGH risk. Used to decide if action needs to be taken.

predict_overflow(rainfall_mm, tank_fill_percent)

INPUT REQUIRED:

Rainfall in mm and tank fill level

ACTION TAKEN:

Checks if conditions are severe enough to expect overflow.

LOGIC IN SIMPLE TERMS:

Returns TRUE if rainfall > 80mm or tank is > 90% full meaning overflow is likely.

dynamic_control_advice(tank_fill_percent)

INPUT REQUIRED:

How full the tank is in percent

ACTION TAKEN:

Suggests what to do operationally, like starting a pump or doing nothing.

LOGIC IN SIMPLE TERMS:

Uses thresholds to return advice: Over 90% = use backup systems. Between 7590% = increase pump. Below 75% = no action.

detect_anomalies(sensor_data)

INPUT REQUIRED:

Live sensor readings

ACTION TAKEN:

Scans for incorrect or suspicious values in the data.

Water LLM Engine - Function Logic (Layman Version)

LOGIC IN SIMPLE TERMS:

Flags tank fill over 100%, or if water is flowing backwards (negative inflow). Helps catch sensor or system faults.