# Smart Water System

**Phase 1: Project Definition and design thinking:**

**Project Definition:**The project involves implementing IoT sensors to monitor water consumption in public places such as parks and gardens. The objective is to promote water conservation by making real-time water consumption data publicly available. This project includes defining objectives, designing the IoT sensor system, developing the data-sharing platform, and integrating them using IoT technology and Python.

**Real-time Water Consumption Monitoring:**Implement a system to track and analyze water usage in real-time.Provide users with access to their water consumption data for better awareness.

**1)Public Awareness:**Raise awareness about the importance of water conservation through education and outreach programs. Encourage responsible water use through public campaigns and information dissemination.

**2)Water Conservation:**Promote water-saving practices in households, businesses, and industries.

Develop guidelines for efficient water usage and conservation measures.

**3)Sustainable Resource Management:**Assess and monitor the condition of local water sources and ecosystems.Develop strategies for sustainable water resource management and protection.

**4)Technology Integration**:Explore the integration of innovative technologies like IoT sensors and data analytics for efficient water management.Use technology to detect leaks and reduce water wastage.

**5)Community Engagement:**Involve local communities in water conservation efforts.

Establish partnerships with community organizations and stakeholders.

**6)Policy and Regulation**:Advocate for water conservation policies and regulations at local and regional levels.

Ensure compliance with water-Real-time Water Consumption Monitoring:

**Design and Deployment Plan for IoT Water Consumption Sensors in Public Places:**

**1)Objective:** Monitor water consumption in public places to promote water conservation and efficiency.

**2)Sensor Selectio:** Choose water flow sensors to measure consumption accurately. Include temperature and humidity sensors to account for environmental factors.

**3)Data Transmission:** Utilize wireless communication (e.g., Wi-Fi, LoRa, or cellular) for real-time data transfer.

4)**Power Supply:** Consider energy-efficient options like battery-powered sensors with long life.

5)**Sensor Placement:** Install sensors at water sources (faucets, showers, etc.) in public areas.

## Real-Time Transit Information Platform:

**1. Home Screen:**A clean interface with a prominent "Find Parking" button.

A map displaying the user's current location.

Icons indicating nearby parking lots or garages.

**2. Search and Filter:**Users can search for parking based on location or address.

Filters for parking type (e.g., street parking, lots, garages) and price range.

**3. Real-Time Updates:**Each parking location card shows real-time availability (e.g., number of available spots).

Colors (e.g., green, yellow, red) indicate availability status.

Update frequency mentioned (e.g., "Updated 5 mins ago").

**4. Location Details:**Tapping a parking location card provides more details:

Address, distance from current location.

Hourly/daily rates.

Navigation option (e.g., "Get Direction").

**5. User Profile:**User profile section with options to set preferences (e.g., preferred payment method, vehicle details).

**Integration Approach:**

**1)Communication Protocol:** Choose an appropriate communication protocol (e.g., MQTT, HTTP, CoAP) that suits the IoT sensors and the data-sharing platform.

2)**Connectivity Options:** Decide on the connectivity options, such as Wi-Fi, cellular, LoRaWAN, or Bluetooth, depending on the location and requirements of the sensors.

3) **Data Format:** Define a standardized data format (e.g., JSON, XML) for the sensor data to ensure compatibility with the platform.

4) **Data Encryption:** Implement data encryption and security measures to protect data during transmission.

5)**Data Frequency:** Determine how often sensors will send data (real-time, periodic, event-triggered) based on the application's needs and sensor capabilities.