PROJECT TITLE : SMART WATER SYSTEM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTRODUCTION:

Smart Water Systems, driven by the Internet of Things (IoT), are revolutionizing the way we manage water resources. These systems combine sensors, data analytics, and automation to enhance water management, especially in the face of growing challenges such as water scarcity and climate change.

CONTENT FOR PROJECT PHASE 2:

Is this section you need to put your design into innovation to solve the problem. Create a doc around it, and share the same for assessment

Objective Innovative Idea for Smart Water System in IoT:

"Aquatic Ecosystem Guardian"

Problem: Aquatic ecosystems are threatened by pollution and climate change.

Innovation: Create an IoT system with sensors monitoring water quality, temperature, and pollutants. Predictive analytics trigger alerts for unusual changes, enabling rapid responses. GPS and flow data pinpoint pollution sources. Data accessibility educates the public and informs policy. Ecosystem restoration efforts are data-driven, and community involvement fosters stewardship.

The "Aquatic Ecosystem Guardian" combines IoT technology and environmental conservation to proactively protect vital aquatic ecosystems and encourage sustainable water practices.

IoT Sensor Design Idea:

"Smart Water Saver Faucet Sensor" Objective:

Develop a faucet attachment with IoT sensors for residential and commercial use, focused on water conservation and efficiency.

Design Details:

- ➢Monitor water flow rates in real-time for precise consumption tracking.
- ➢Detect and report even minor leaks promptly.
- ➢Enable touchless operation through gesture control for convenience and hygiene.
- ➢Connect to a mobile app providing insights, leak alerts, and water-saving tips.
- ➢Collect data on water usage patterns, temperature, and flow rates for informed decision-making.
- ➢Allow users to set and maintain preferred water temperatures.
- ➢Integrate with voice assistants for voice-activated control.
- ➢Include an automatic shutoff feature to prevent accidental waste.

- ➢The "Smart Water Saver Faucet Sensor" offers a comprehensive solution to promote responsible water usage while providing user convenience and data-driven insights.

Innovative Real-Time Transit Information Platform:

"Transit Tracker 360"

"Transit Tracker 360" is a dynamic IoT-based platform aiming to transform public transportation. It integrates real-time data from various transit modes, offers personalized routes, and updates passengers on vehicle locations and delays. Users can make informed decisions based on cost, time, and environmental impact. Crowdsourcing features help passengers report on crowd levels, and the platform encourages ecofriendly choices by tracking carbon footprints. Users can book tickets, pay fares, and access AR-based indoor navigation within transit hubs. This comprehensive platform seeks to enhance the public transportation experience, reduce congestion, and promote sustainability for smarter, more efficient transit systems.

Innovative Integration Approach for Smart Water System in IoT:

"Water-Energy Nexus Optimization"

The “Water-Energy Nexus Optimization” approach aims to create a symbiotic relationship between smart water systems and energy management, fostering resource efficiency, sustainability, and cost reduction.

Integration Features

- ❖ Smart Grid Connectivity: By linking smart water infrastructure with the electrical grid, this approach enables real-time coordination. During periods of peak energy demand, water pumping can be strategically reduced to lower electricity costs, without compromising water availability.

- ❖ Renewable Energy Integration: Incorporating renewable energy sources like solar panels and wind turbines into water treatment facilities empowers them to generate their power. This reduces reliance on fossil fuels, curbing both energy costs and environmental impact.

- ❖ Integration Features: Real-time GPS tracking, passenger count sensors, traffic data, weather conditions, mobile apps for commuters, payment systems, and data analytics for route optimization and real-time updates.

- Energy-Efficient Components: The adoption of energy efficient pumps and motors in water distribution networks minimizes electricity consumption while maintaining reliable water supply.

- Energy Recovery: Energy recovery systems within wastewater treatment plants capture and repurpose energy from the treatment process, further reducing energy dependence.

- ❖ Real-time Analytics: IoT sensors and data analytics optimize water pumping schedules based on fluctuating energy prices, water demand patterns, and network efficiency.

- ❖ Predictive Maintenance: Implementing predictive maintenance algorithms minimizes downtime and energy wastage in water infrastructure.

- ❖ Data Sharing: The sharing of real-time data between water and energy utilities enhances coordination, ensuring efficient resource management.

- ❖ The “Water-Energy Nexus Optimization” integration approach fosters a mutually beneficial relationship between water and energy systems, promoting sustainability, reducing operational costs, and enhancing resource utilization across both sectors.

Absolutely, presenting your innovative design for assessment in the context of smart water systems is essential for garnering feedback and support. Here’s a suggested structure for your document:

Title: Innovative Design for Smart Water System Enhancement

Creating a complete Python code for a smart water system in IoT is a complex task, but I can provide you with a high-level structure and some code snippets to get you started. This example assumes you are using a Raspberry Pi with sensors to collect water data and a cloud platform for data processing. You may need to adapt this code to your specific hardware and platform.

```
# Import necessary libraries import time import RPi.GPIO
as GPIO import Adafruit_DHT from datetime import
datetime from azure.iot.device import IoTHubDeviceClient,
Message

# Define GPIO pins and sensor type (DHT22 in this example)
DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN = 4
WATER_SENSOR_PIN = 17

# Azure IoT Hub configuration
CONNECTION_STRING = "<Your Azure IoT Hub Connection String>"
```

```
# Initialize Azure IoT Hub client client
=
IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)

# Function to read temperature and humidity from DHT22 sensor def
read_dht_sensor():
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR,
DHT_PIN)     return humidity, temperature

# Function to detect water level def
detect_water_level():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(WATER_SENSOR_PIN, GPIO.IN)
return GPIO.input(WATER_SENSOR_PIN)
 try:     while
True:
        # Read temperature and humidity
humidity, temperature = read_dht_sensor()

        #        Detect        water        level
water_level     =     detect_water_level()
```

```
# Create a JSON message with sensor
data        message = Message(

'{{"timestamp":"{}","temperature":{},"humidity":{},"water_level":{}}}'.format(
                datetime.now().isoformat(),                temperature,
humidity,                water_level,
            )
        )

        # Send message to Azure IoT Hub
client.send_message(message)        print("Message
sent to Azure IoT Hub")

        time.sleep(60)  # Send data every 60 seconds

except KeyboardInterrupt:
    print("Program terminated by user")
    GPIO.cleanup()
```

In this example, we are using the DHT22 sensor to measure temperature and humidity, and a water level sensor connected to a GPIO pin. Data is sent to an Azure IoT Hub using the Azure IoT SDK.

Make sure to replace <Your Azure IoT Hub Connection String> with your actual Azure IoT Hub connection string.