**SMART WATER MANAGEMENT**

**PHASE 3- DEVELOPMENT PART 1**

**TEAM MEMBERS:**

**K.EZHILARASI**

**S.AFREEN THABASSUM**

**B.G.EZHIL APARNA**

**R.ISHWARYA**

**TEAM ID:**

**444**

**PROBLEM STATEMENT:**

Start building the IOT Smart water management

**INTRODUCTION:**

Smart water management in IoT (Internet of Things) is a cutting-edge approach to efficiently monitor, control, and optimize the use of water resources. It leverages IoT technologies to collect real-time data from various sensors and devices, allowing for intelligent decision-making and resource conservation. This innovative approach addresses critical water-related challenges, such as water scarcity, leak detection, water quality monitoring, and sustainable consumption. By seamlessly connecting sensors, actuators, and data analytics, smart water management in IoT offers a comprehensive solution to ensure the sustainable and responsible use of this precious resource.

**DEPLOY IOT SENSORS IN SMART WATER MANAGEMENT :**

* **Identify Objectives :** Determine your specific goals, whether it's monitoring water quality, detecting leaks, optimizing irrigation, or managing water resources efficiently.

* **Select Appropriate Sensors :** Choose IoT sensors that match your objectives. For instance, water quality sensors, flow meters, pressure sensors, or leak detectors. Ensure they are compatible with your IoT platform.
* **IoT Platform Selection:**  Select an IoT platform or cloud service that can handle data from your sensors. Popular options include AWS IoT, Azure IoT, or specialized platforms like Particle
* **Connectivity:** Ensure your sensors have the necessary connectivity, like Wi-Fi, LoRa, or cellular, to transmit data to the IoT platform
* **Sensor Placement:** Strategically place sensors at relevant points in your water system. For instance, near water sources, distribution pipelines, or water treatment facilities.
* **Data Transmission**: Set up sensors to collect data and transmit it to the IoT platform in real-time. Configure data transmission intervals, if needed.

* **Data Analysis:** Use data analytics tools to gain insights and make informed decisions. Machine learning can help predict water usage patterns or detect anomalies.

* **Alerting and Response:** The IOT sensors are placed at the system point to detect water level, check water quality, leakage of chemicals, etc.it automatically sends notifications to the concerned authorities by sending data through the cloud system and solves the issue as soon as possible.
* **Maintenance and Monitoring:** using ultrasonic , vibration, and pressure sensors to help monitor dam function with pressure sensors.
* **community Awareness:** water conservation is one of the prime concerns in the current scenario where environmental conditions are deteriorating at an alarming rate.
* **Data Storage and Historical Analysis:**

Internet of Things (IoT) can aid drinking water supply management. It uses literature review to highlight strengths and weaknesses of the technologies, as well as a multicriteria analysis (MCA) to analyse factors impacting implementation in select G20 countries: South Africa, South Korea, Japan, India, and Indonesia

**IOT SENSORS TO SEND SMART WATER LEVEL**

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  | Import pyttsx3  import speech\_recognition as sr | |
|  |  | import datetime | |
|  |  | import wikipedia | |
|  |  | import webbrowser | |
|  |  | import os | |
|  |  | import smtplib | |
|  |  |  | |
|  |  |  | |
|  |  | engine = pyttsx3.init('sapi5') | |
|  |  | voices = engine.getProperty('voices') | |
|  |  | #print(voices[0].id) | |
|  |  | engine.setProperty('voices',voices[0].id) | |
|  |  |  | |
|  |  | def speak(audio): | |
|  |  | engine.say(audio) | |
|  |  | engine.runAndWait() | |
|  |  |  | |
|  |  |  | |
|  |  | def WishMe(): | |
|  |  | hour = int(datetime.datetime.now().hour) | |
|  |  | if hour>=0 and hour<12: | |
|  |  | speak("Good Morning! Sir") | |
|  |  |  | |
|  |  | elif hour>=12 and hour<18: | |
|  |  | speak("Good Afternoon! Sir") | |
|  |  |  | |
|  |  | else: | |
|  |  | speak("Good Evening! Sir") | |
|  |  |  | |
|  |  | speak("I am JARVIS. Please tell me, How may I help you ") | |
|  |  |  | |
|  |  | def takeCommand(): | |
|  |  | #It takes microphone input from the user and returns the string output. | |
|  |  |  | |
|  |  |  | |
|  |  | r = sr.Recognizer() | |
|  |  | with sr.Microphone() as source: | |
|  |  | print("Listerning...") | |
|  |  | r.pause\_threshold = 1 | |
|  |  | audio = r.listen(source) | |
|  |  |  | |
|  |  | try: | |
|  |  | print("Recognizing...") | |
|  |  | query = r.recognize\_google(audio,language='en-in') | |
|  |  | print(f"User said: {query}\n") | |
|  |  |  | |
|  |  | except Exception as e: | |
|  |  | print(e) | |
|  |  |  | |
|  |  | print("Say that again please...") | |
|  |  | return "None" | |
|  |  | return query | |
|  |  |  | |
|  |  | def sendEmail(to, content): | |
|  |  | server = smtplib.SMTP('smtp.gmail.com', 587) | |
|  |  | server.ehlo() | |
|  |  | server.starttls() | |
|  |  | server.login('youremail@gmail.com', 'your-password-here') | |
|  |  | server.sendmail('youremail@gmail.com', to, content) | |
|  |  | server.close() | |
|  |  |  | |
|  |  | if \_\_name\_\_=="\_\_main\_\_": | |
|  |  | WishMe() | |
|  |  | while True: | |
|  |  | #if 1: | |
|  |  | query = takeCommand().lower() | |
|  |  |  | |
|  |  | # Logic for executing tasks nased on query | |
|  |  | if 'wikipedia' in query: | |
|  |  | speak('Searching Wikipedia...') | |
|  |  | query = query.replace("wikipedia","") | |
|  |  | results = wikipedia.summary(query, sentences=2) | |
|  |  | speak("According to Wikipedia") | |
|  |  | print(results) | |
|  |  | speak(results) | |
|  |  |  | |
|  |  | elif 'open youtube' in query: | |
|  |  | speak('Opening Youtube, sir...') | |
|  |  | webbrowser.open("youtube.com") | |
|  |  |  | |
|  |  | elif 'open google' in query: | |
|  |  | speak('Opening googel, sir...') | |
|  |  | webbrowser.open("google.com") | |
|  |  |  | |
|  |  | elif 'open water system' in query: | |
|  |  | speak('Opening Water Management System...') | |
|  |  | webbrowser.open("https://console.firebase.google.com/u/0/project/water-  management-system-8ab48/settings/serviceaccounts/databasesecrets") | |
|  |  |  | |
|  |  | elif 'play music' in query: | |
|  |  | speak('Playing the music, sir...') | |
|  |  | music\_dir = 'C:\\Users\\HP\\Music' | |
|  |  | songs = os.listdir(music\_dir) | |
|  |  | print(songs) | |
|  |  | os.startfile(os.path.join(music\_dir, songs[0])) | |
|  |  |  | |
|  |  | elif 'the time' in query: | |
|  |  | strTime = datetime.datetime.now().strftime("%H:%M:%S") | |
|  |  | speak(f"Sir, the time is {strTime}") | |
|  |  |  | |
|  |  |  | |
|  |  | elif 'open code' in query: | |
|  |  | codePath = "C:\\Users\\HP\\AppData\\Local\\Programs\\Microsoft VS Code\\Code  .exe" | |
|  |  | os.startfile(codePath) | |
|  |  |  | |
|  |  | elif 'Thank you Jarvis!' in query: | |
|  |  | speak('Welcome, sir...') | |
|  |  |  | |
|  |  | elif 'how are you' in query: | |
|  |  | speak("I am fine, Thank you") | |
|  |  | speak("How are you, Sir") | |
|  |  |  | |
|  |  | elif 'fine' in query or "good" in query: | |
|  |  | speak("It's good to know that your fine") | |
|  |  |  | |
|  |  | elif 'email to flash' in query: | |
|  |  | try: | |
|  |  | speak("What should I say?") | |
|  |  | content = takeCommand() | |
|  |  | to = "" | |
|  |  | #sendEmail(to, content) | |
|  |  | speak("Email has been sent!") | |
|  |  | except Exception as e: | |
|  |  | print(e) | |
|  |  | speak("Sorry my friend madmax, I am not able to send this email") | |
|  |  |  | |
|  |  | elif 'exit' in query: | |
|  |  | speak("Thanks for giving me your time") | |
|  | |  | |

|  |  |  |
| --- | --- | --- |
|  |  |  |