Phase 4: Development Part 2

BY:

T.GOWTHAM

S.SIVA

A.PERIYASAMY

M.RAJESHKUMAR

Objectives:

The objectives of smart water quality monitoring system are:

- To measure perilous quality metrics like physical, chemical and microbial properties.
- To find the deviations in measured metrics and give timely warning in recognition threats or hazards.
- To provide real-time analysis of the sensor data and recommend appropriate corrective measures.

Promoting Water Conservation:

User Engagement:

Develop features to engage users in water conservation efforts. For example, provide real-time usage statistics and historical trends.

Alerts and Notifications:

Implement notifications to alert users when water consumption is unusually high, encouraging them to investigate and reduce water usage.

Gamification:

Create a gamified system where users can earn rewards or compete with others to save water.

Tips and Recommendations:

Offer water-saving tips and recommendations based on usage patterns.

Community and Social Sharing:

Allow users to share their water-saving achievements on social media or within a community on the platform.

Deployment and Maintenance:

- Deploy the platform on the Raspberry Pi and ensure it runs continuously.
- Monitor the system for any issues or anomalies and set up alerting mechanisms.
- Regularly update both the Raspberry Pi OS and the software components to ensure security and stability.
- Backup data to prevent data loss.
- Document the setup and maintenance processes for future reference.
- Domestic water is intended for human consumption for drinking and cooking purposes.
- water quality measurement involves manual collection of water at various locations, storing the samples
- approaches are efficient due to the unavailability of real time water quality information delayed detection of contaminants and not cost effective solution.

Smart water quality approaches have been considered for lake and sea water applications.

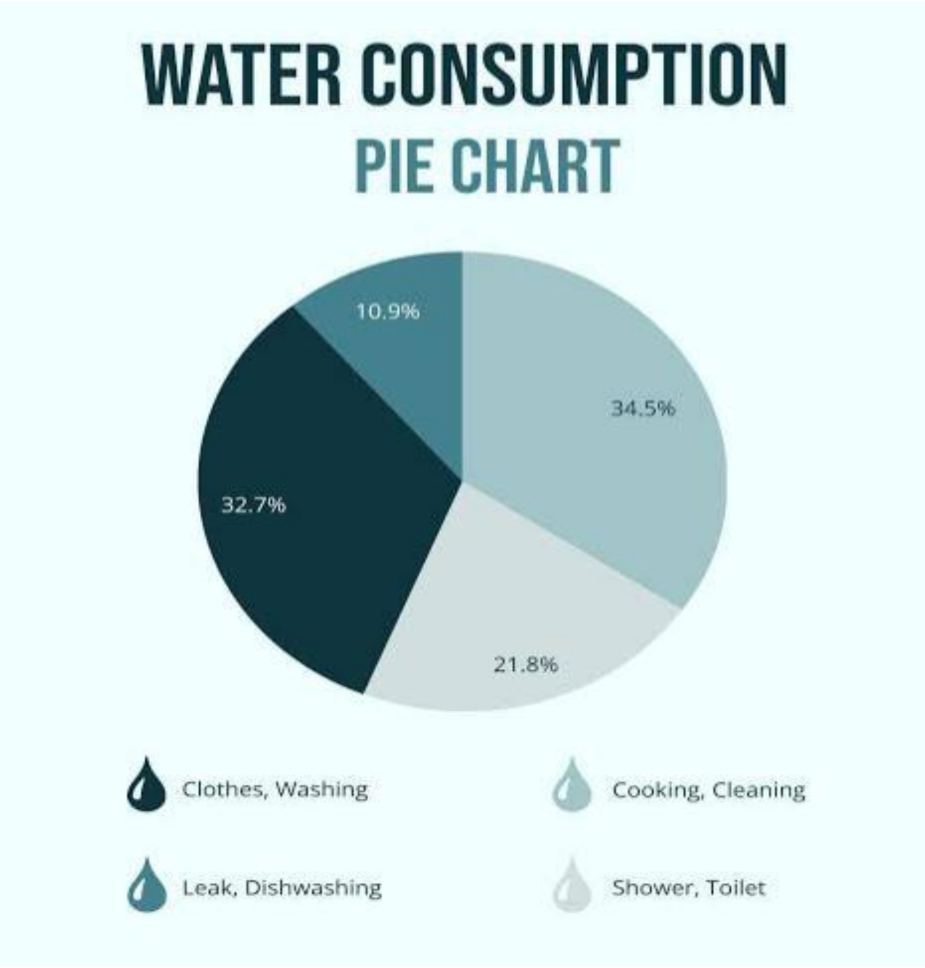
Application:

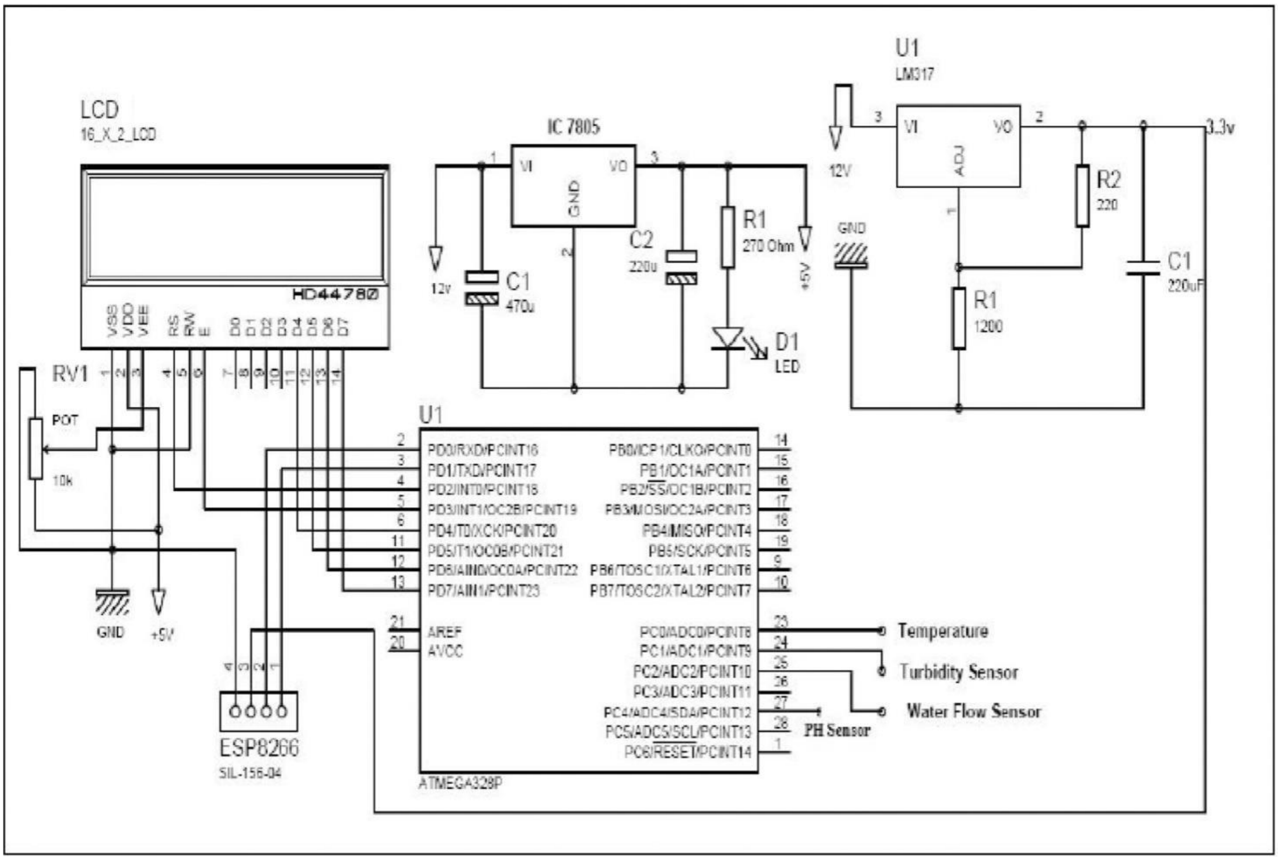
- distributed wireless sensor networks are required to monitor the parameters
- The data'sare monitored to a centralized controller they wereusing wireless communication.
- In order to collect data on pH, turbidity, level of water, temperature, and humidity of the surrounding atmosphere, the water quality monitoring system employs IoTsensors
- It Proposes a cost effective and efficient IoT based smart water quality monitoring system which monitors the quality parameters uninterruptedly.
- Declining quality of water is detrimental to the health, environment and economy.

- Water quality monitoring is demarcated as the assortment of data at set or desired places and at periodic intervals

Schematic analysis

The schematic diagram of the proposed work is as shown in fig





- **Improve water quality** and prevent contamination by chemical waste and natural pollution such as acidification. In order to improve and maintain the quality of water, companies use sensors and IoT technology for real-time monitoring and control.
- Enhance the efficiency of water systems such as water collectors, treatment plants, distribution mains and wastewater recycling centers. Using IoT and data solutions for asset management, companies can keep important measurements such as water pressure, temperature, flow, etc. at hand, integrate predictive maintenance and avoid breakage and downtime of equipment.
- **Practice consumption monitoring** via IoT-based water management systems. It helps to optimize and keep under control the usage of water resources at different levels — households, communities

Benefits of using IoT for water management:

- Transparency
- Immediate response
- Automation
- optimized use of human resources
- Sustainability
- Optimized cost

Running on Raspberry Pi

Important points from the provided passage on installing and running Node-RED on a Raspberry Pi:

Prerequisites:

✓Ensure you are using Raspberry Pi OS, specifically Bullseye for the best compatibility.

Installation and Upgrade:

✓A script is provided to install Node.js, npm, and Node-RED on the Raspberry Pi. It can also be used to upgrade Node-RED when new releases are available.

Script Actions:

✓Removes existing Node-RED if present.

✓Checks Node.js version and upgrades it to at least v14 if necessary.

✓Installs the latest Node-RED version using npm.

✓Optionally installs Pi-specific nodes.

✓Sets up Node-RED to run as a service with start, stop, restart, and logcommands.

Running Locally:

✓Use the `node-red` command to run Node-RED in a terminal. Use `node-red-pi` with the `max-old-space-size` argument to free up memory on Raspberry Pi.

Running as a Service:

✓The installation script configures Node-RED to run as a service. Use commands like `node-red-start`, `node-red-stop`, and `node-red-restart` tomanage the service. It runs in the background.

Autostart on Boot:

✓To enable Node-RED to run on boot, use `sudo systemctl enable nodered.service`. To disable it, run `sudo systemctl disable nodered.service`.

Opening the Editor:

✓Access the Node-RED editor in a browser. Use `http://localhost:1880` when on the Pi desktop, or `<hostname>:1880` from another machine, replacing `<hostname>` with the Pi's IP address (you can find the IP with `hostname -I` on the Pi).

Data-sharing platform:

Building an Integrated IoT Solution with ThingSpeak and Firebase

Step 1: Set Up ThingSpeak for IoT

✓Create a ThingSpeak account.

✓Create a new ThingSpeak channel to receive IoT data.

✓Define fields in your channel to store specific sensor data (e.g.,temperature, humidity).

✓Generate API keys for your ThingSpeak channel, allowing the IoT device to send data.

Step 2: Connect Your IoT Device

✓Configure your IoT device to collect sensor data.

✓Implement the code on your device to send data to ThingSpeak using thegenerated API keys.

Step 3: Data Visualization and Analysis

✓Use ThingSpeak's built-in features to visualize and analyze the data.

✓Create charts and graphs to monitor trends.

✓Set up email or SMS alerts based on data thresholds.

Step 4: Integration with Firebase

✓Create a Firebase project and initialize Firebase in your application.

✓Set up Firebase Realtime Database to store IoT data.

✓Configure security rules to control data access.

Step 5: Real-time Data Transfer

✓Modify your IoT device code to simultaneously send data to ThingSpeak and Firebase.

✓Use Firebase SDK to push IoT data to the Realtime Database.

Step 6: Data Retrieval

✓Implement code in your application to retrieve IoT data from FirebaseRealtime Database.

✓Update data in real-time as your IoT device sends new information.

Step 7: Data Analytics and Decision-making

✓Use Firebase's data storage and retrieval capabilities to conduct furtheranalysis.

✓Implement algorithms or business logic to derive insights from IoT data.

✓Make informed decisions and trigger actions based on the real-time datastored in Firebase.

Step 8: Scaling and Optimization

✓Ensure your system can scale as more IoT devices are added.

✓Optimize the integration to minimize latency and enhance dataprocessing.

Technologies like an(WebPage,Application and etc) to create a platform that displays real-time water consumption data

Platform Components:

IoT Sensors:

Deploy IoT sensors to monitor water consumption at various points, such as residential buildings, commercial properties, or water distribution systems.

Sensors should be capable of measuring and transmitting real-time water consumption data.

Data Transmission:

Sensors transmit data securely to a central server using protocols like MQTT, HTTP, or other suitable IoT communication methods.

Implement secure data encryption and authentication mechanisms.

Central Data Storage:

Set up a centralized database or data storage system to collect and store incoming data from sensors.

Data Processing and Analytics:

Implement data processing algorithms to calculate real-time water consumption and identify trends and anomalies.

Use analytics tools to gain insights into water usage patterns.

Web Application:

Develop a user-friendly web application to display water consumption data and insights.

Enable users to access the platform from desktop and mobile devices.

Ensure responsive design and cross-browser compatibility.

User Authentication and Management:

Implement user authentication to ensure data privacy and security.

Allow users to create accounts and manage their profiles.

Dashboard:

Create interactive dashboards displaying real-time and historical water consumption data.

Include charts, graphs, and visualizations to make data easily comprehensible.

Provide options for customizing dashboard views.

Alerts and Notifications:

Implement alerting mechanisms to notify users about water conservation opportunities or unusual consumption patterns.

Allow users to set up custom alerts based on their preferences.

Gamification Features:

Integrate gamification elements to encourage users to conserve water.

Reward users for achieving water-saving milestones or participating in conservation challenges.

Educational Content:

Offer educational resources and tips on water conservation.

Include articles, videos, and infographics to raise awareness and inform users.

Community Engagement:

Foster a sense of community by allowing users to share their conservation efforts and achievements.

Enable users to interact through forums, comments, and social features.

Mobile App:

Develop a companion mobile app for on-the-go access.

Include features like barcode scanning for water-efficient products and integration with smartphone sensors.

Data Export and Sharing:

Allow users to export their data for personal analysis or sharing with local authorities or water-saving organizations.

APIs and Integration:

Provide APIs for integration with third-party devices and services.

Integrate weather data to provide weather-based recommendations forirrigation and water usage.

Reporting and Feedback:

Collect user feedback and suggestions for improving the platform.

Generate reports on water conservation impact and share them with usersand stakeholders.

Security and Compliance:

Implement robust security measures to protect user data. Comply with data protection regulations and privacy laws.

Scalability and Performance:

Design the platform to handle a growing number of sensors and users.

Ensure high availability and performance.

User Engagement Strategies:

Implement a rewards system to incentivize water conservation.

Launch awareness campaigns and challenges to engage users.

Collaborate with water utilities, local governments, and NGOs to promotethe platform.

Provide real-time feedback on users' water usage and progress.Encourage users to share their achievements on social media.

Offer discounts on water-efficient products as rewards.

Host community events and webinars on water conservation topics.

Conclusion:

In The Presumption of Water Consumption,Remember to consider privacy and security during every step, especially when dealing with IoT and user data. Additionally, you might need to comply with local regulations regarding data collection and privacy.