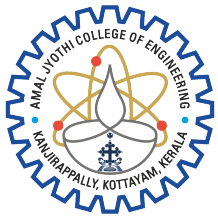

AMAL JYOTHI
COLLEGE OF ENGINEERING
AUTONOMOUS
KANJIRAPPALLY

AQUA MONITOR

23INMCA310 - Mini Project 1

Scrum Master
Ms. Meera Rose Mathew
Assistant Professor
Department of Computer Applications

THOMSON SHIJU
AJC23MCA-I060
INTMCA2023-28 S6
Roll No. 58

DEPARTMENT OF
COMPUTER APPLICATIONS

ABSTRACT

AquaMonitor is an IoT-based smart water-level monitoring and motor control system designed to automate household and industrial water management. The system continuously measures the water level in storage tanks using wire connections connected to an ESP32 microcontroller. The collected data is transmitted to a cloud database, enabling real-time monitoring through a web or mobile application.

The platform also provides automated motor control, where the motor is turned ON or OFF based on predefined threshold levels, reducing manual effort and preventing water wastage. Users can remotely monitor tank status, receive alerts, and manually operate the motor from any location through the cloud-enabled interface.

AquaMonitor enhances convenience, optimizes water usage, and provides a reliable smart home solution using IoT technology. The system integrates real-time data, cloud communication, a user-friendly application interface, and seamless motor automation to create an efficient and scalable water management solution.

- WEB APPLICATION PAGES
  - LOGIN PAGE
    - User login
  - DASHBOARD
    - Water level
    - Motor ON/OFF
    - Status indicators
  - DEVICE STATUS PAGE
    - ESP32 status
    - Cloud sync status
  - AUTOMATION SETTINGS PAGE
    - Min/Max water level
    - Manual/Auto mode
    - Schedule timings
  - DATA ANALYTICS PAGE
    - Daily/weekly/monthly water usage
  - USER PROFILE PAGE
    - Name, email, phone
    - Change password
    - App theme (light/dark mode)

- ADMIN PANEL
  - Admin login
  - Manage users
  - Sending notifications

- The frontend is built using **React Native** to create a simple, responsive web interface. It displays real-time tank levels, motor status, and allows users to control the motor through interactive buttons and visuals.

- The backend uses **MQTT Broker to handle the** communication between the web app, cloud database, and ESP32. It processes sensor data, updates the cloud, and controls the motor. **Firebase** is used for real-time data storage and synchronization.

- The system uses **MQTT (Message Queuing Telemetry Transport)** as the cloud communication protocol. MQTT is a lightweight, fast, and reliable messaging service designed for IoT devices. Through an MQTT broker, the ESP32 sends real-time water level data and receives motor ON/OFF commands. This allows users to monitor and control the system from anywhere in the world with low latency and secure communication.

IoT Devices

1. Esp32
2. Relay
3. Water Pump
4. Power Supply
5. Jumper Wires
6. Breadboard
7. Resistors(Optional)