SMART WATER FOUNTAINS

Objectives:

Water Efficiency: Ensure water fountains are operational and minimize water wastage.

Public Awareness: Raise awareness about water conservation through real-time data and a mobile app.

Data Collection: Collect data from IoT sensors for analysis and reporting.

User-Friendly Interface: Develop a user-friendly mobile app to display fountain status.

IoT Sensor Setup:

A network of IoT sensors is deployed to monitor the water fountains. Each sensor includes:

Water Flow Sensor: Measures water flow in real-time.

Ultrasonic Sensor: Detects user presence.

Microcontroller (e.g., Arduino): Collects and processes sensor data.

Wi-Fi Module: Transmits data to the central server.

Power Supply (e.g., battery or solar panel).

These sensors are strategically placed near water fountains and connected to a central server, preferably using MQTT or a RESTful API.

Raspberry Pi Integration:

A Raspberry Pi is employed as the central server to collect and process data from the IoT sensors. The Raspberry Pi:

Receives data from the sensors.

Stores data in a database (e.g., MySQL or MongoDB).

Runs a web server to provide data to the mobile app.

Communicates with the mobile app using WebSocket for real-time updates.

Mobile App Development:

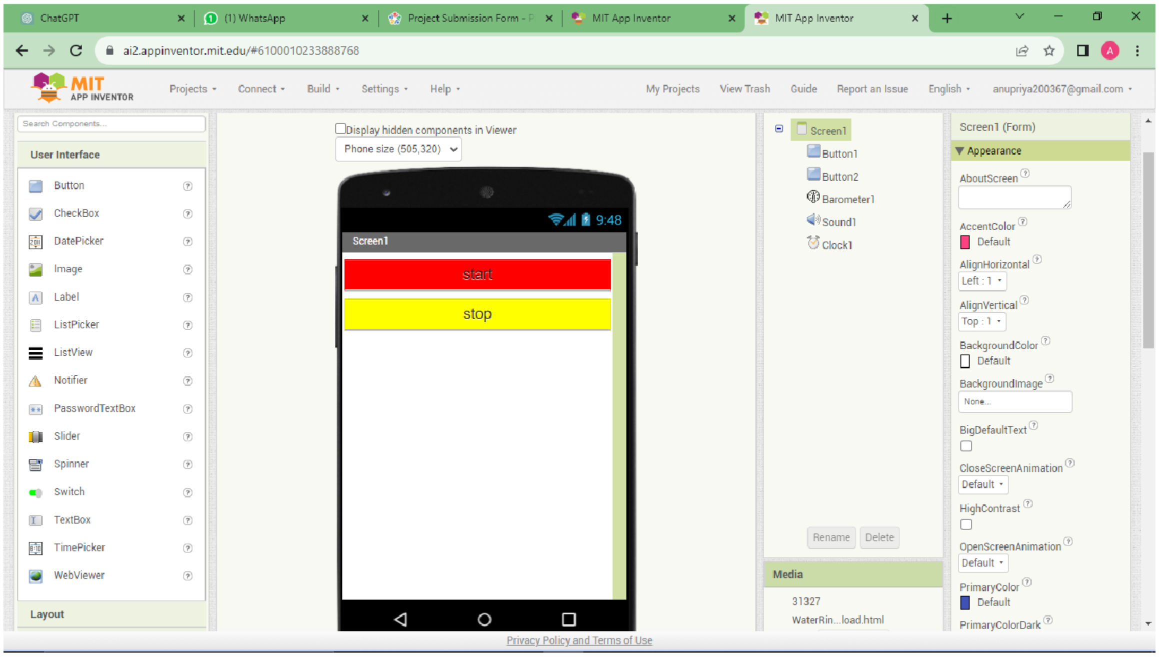
The mobile app is a crucial component for user interaction. It is developed for both Android and iOS platforms. Key features include:

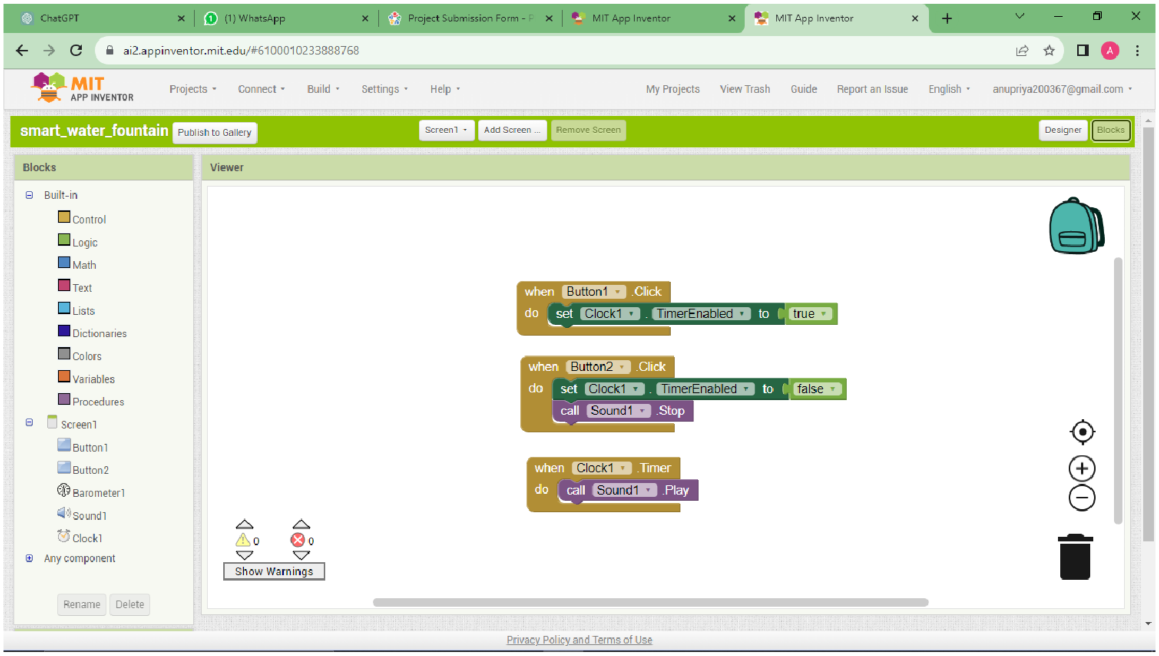
Fountain Map: Displays the locations of nearby fountains.

Fountain Status: Shows real-time fountain status (working or non-working).

Historical Data: Offers water usage statistics and trends.

User Accounts: Allows users to create accounts and personalize their experience.

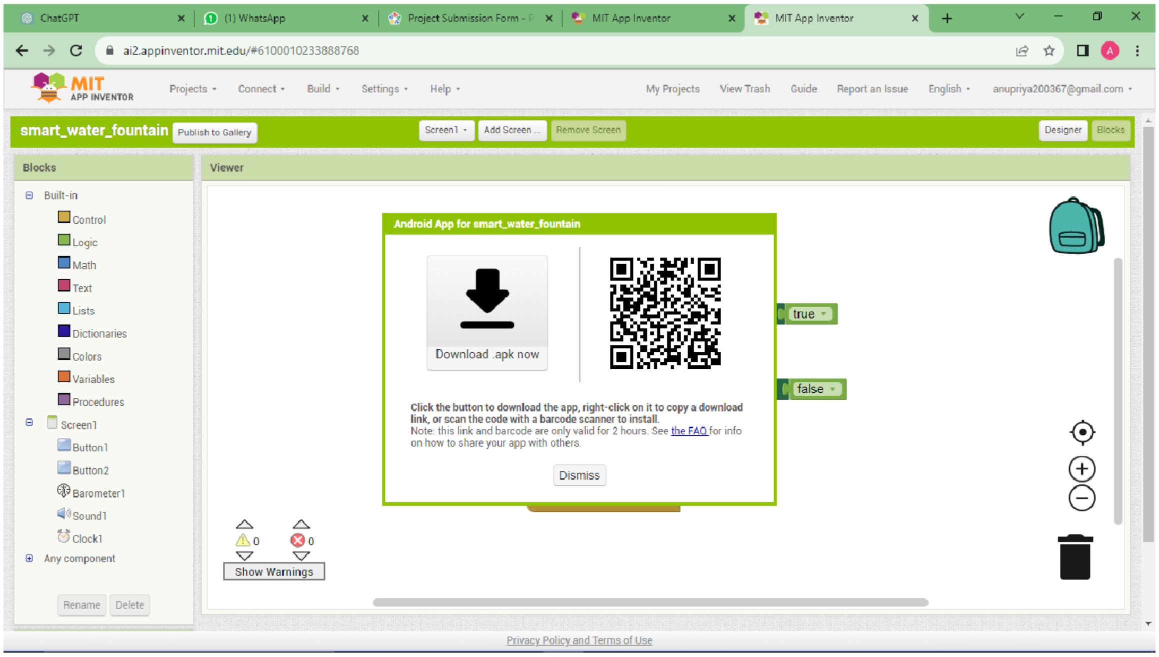




```
when Button1 .Click
do  set Clock1 . TimerEnabled to true

when Button2 .Click
do  set Clock1 . TimerEnabled to false
    call Sound1 .Stop

when Clock1 .Timer
do  call Sound1 .Play
```

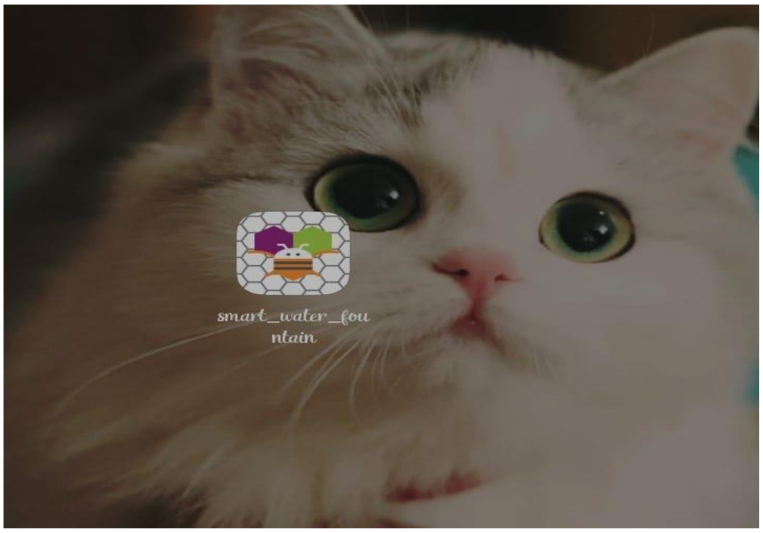


Android App for smart_water_fountain

Download .apk now

Click the button to download the app, right-click on it to copy a download link, or scan the code with a barcode scanner to install.
Note: this link and barcode are only valid for 2 hours. See the FAQ for info on how to share your app with others.

Dismiss



Code Implementation:

IoT Sensor Code: Each sensor's code includes data collection, processing, and Wi-Fi communication. The code may be written in C/C++ or a suitable IoT platform like Arduino IDE.

Raspberry Pi Code: Python scripts are used to receive data from sensors, update the database, and communicate with the mobile app via WebSockets.

Mobile App Code: The app is developed using relevant programming languages (e.g., Java for Android, Swift for iOS). It communicates with the Raspberry Pi using HTTP requests and WebSockets.

```
#include <Servo.h>
```

```
const int pumpPin = 9; // Pin for controlling the water pump
const int triggerPin = 10; // Ultrasonic sensor trigger pin
const int echoPin = 11; // Ultrasonic sensor echo pin
const int redPin = 5; // Red LED pin
const int greenPin = 6; // Green LED pin
const int bluePin = 3; // Blue LED pin
Servo myservo;

void setup() {
  pinMode(pumpPin, OUTPUT);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  myservo.attach(12);
  myservo.write(90); // Adjust servo position for water flow control
  Serial.begin(9600);
}

void loop() {
  long duration, distance;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration / 58.2;

  if (distance < 10) { // If water level is low
    digitalWrite(pumpPin, HIGH); // Turn on the water pump
    analogWrite(redPin, 255); // Red LED on
    analogWrite(greenPin, 0); // Green LED off
    analogWrite(bluePin, 0); // Blue LED off
  } else {
    digitalWrite(pumpPin, LOW); // Turn off the water pump
    analogWrite(redPin, 0); // Red LED off
    analogWrite(greenPin, 255); // Green LED on
    analogWrite(bluePin, 0); // Blue LED off
  }

  delay(1000); // Check water level every second
}
```

Promotion of Water Efficiency and Public Awareness:

The real-time water fountain status system promotes water efficiency and public awareness in several ways:

Fountain Accessibility: Users can quickly locate functioning fountains, reducing the need for disposable water bottles.

Data Visualization: Historical data and trends encourage users to make informed choices about water usage.

Real-Time Updates: Real-time status updates remind users to conserve water by using only operational fountains.

Feedback Mechanism: Users can report non-working fountains through the app, facilitating timely repairs.

Public Awareness: The app can include water conservation tips and information, raising awareness about the importance of responsible water usage.

By combining IoT sensor technology, mobile app development, and real-time data, this project actively contributes to both water conservation and public education on the subject.