FLOOD MONITORING SYSTEM AND EARLY WARNING

PROBLEM STATEMENT:

To lessen the effects of floods, an IoT-based flood monitoring system is being created for timely detection, real-time data collecting, and effective sharing of flood-related information to authorities and the public.

PROBLEM DESCRIPTION:

Floods are a type of natural disaster that can seriously harm property and endanger people's lives. For efficient disaster management and risk reduction, flood-related information must be promptly detected, monitored, and communicated. To solve these crucial challenges, this project intends to build an IoT-based Flood Monitoring System.

SOLUTION DESIGN:

1. <u>Network of Sensor</u>: In flood-prone locations, strategically deploy an IoT sensor network. Sensors must monitor things like water levels, precipitation, weather, soil moisture, and water quality. To ensure data reliability and accuracy, use redundant sensors. Examine several sensor kinds, such as optical, radar, and ultrasonic ones, depending on the type of monitoring required.
2. <u>Infrastructure for communication:</u> Establish a reliable, redundant communication system. Utilise a mix of wireless (cellular, satellite, LoRaWAN) and wired (Ethernet, fibre) communication methods. Give your customers choices for backup communication to assure data transmission in case of network disruption.
3. <u>Data Transmission and Collection:</u> Sensors should constantly gather data and communicate it in real-time to a central server or cloud platform. To save bandwidth and transmit data securely, use data compression and encryption.
4. <u>Analytics and data processing</u>: Utilise cloud-based platforms for processing and storing data. For real-time data processing, use machine learning and data analytics. Find anomalies, forecast floods, and produce early warnings.

5. <u>Interfaces for users:</u> Create websites and mobile apps for a range of user groups.The general public has access to flood warnings, evacuation guidelines, and flood maps.
6. <u>Authorities (public) :</u> Keep track of data in real-time, get alerts, and coordinate response activities. Access vital information for quick deployment, emergency personnel. Make sure the interfaces are user-friendly with a clear design and accessibility tools.
7. <u>Alerting Systems:</u> Use a multi-channel alert system (SMS, email, push notifications) to inform consumers of flood warnings in a timely manner. Prioritise warnings according to their importance and proximity to affected locations.
8. <u>Power Control:</u> Create sensors using low-power technology. Utilise energy-efficient technologies to generate electricity, such as solar panels and energy harvesting.
9. <u>Environment-Related Issues:</u> Prior to the deployment of sensors, do environmental impact evaluations. When possible, use eco-friendly supplies of energy and materials. Observe environmental laws and regulations.

OUTCOMES:

The Internet of Things-based Flood Monitoring System (IFMS) will bring about a number of advantages, such as:

- Early flood detection is essential for arranging early warnings and evacuations.
- less fatalities and property losses.
- improved resource allocation and catastrophe management.
- data-driven insights for assessing flood risk and urban planning.
- increased awareness and resilience in the community.
- otential for system expansion and modification to detect additional environmental aspects including water quality and weather.

<u>Innovative Concept:</u>

A network of floating sensors outfitted with cutting-edge technology that can track water levels, rainfall, and meteorological conditions in real-time is a novel IoT-based flood monitoring and early warning system concept. These intelligent buoys would be positioned in water bodies, such as rivers, lakes, and locations vulnerable to flooding, and would provide exact information for flood forecasting and early warning.

<u>Steps to implement the idea:</u>

<u>Step 1:</u> Development of sensors

- Create unique sensors that can float on water and measure a variety of variables, including water level, temperature, rainfall, and even water quality.
- These sensors ought to have GPS and communication tools for data transmission.

<u>Step 2:</u> Sensor Deployment

- In flood-prone waterways, strategically place these smart buoys.
- Make sure they are firmly attached to prevent drifting.

<u>Step 3:</u> Data Transmission and Gathering

- Connect the sensors to a hub in the middle that handles data collection and processing.

<u>Step 4:</u> Prediction and Data Analysis

- Use software to instantly analyse the data. By looking at patterns in water levels, predicted rainfall, and the weather, algorithms can forecast flood conditions.

<u>Step 5:</u> Alert system

- Create a warning system that notifies the government and locals when flood conditions are likely to materialise.
- Text messages, smartphone notifications, or even sirens may serve as alerts.

<u>Step 6:</u> User-friendly App

- Make a user-friendly online interface and mobile application so that anyone may access the sensors' real-time data.

<u>Step 7:</u> calibration and testing

- To verify accuracy, test the sensors and the overall system in various types of weather and bodies of water.

<u>Step 8:</u> Continuous Improvement

- Update the system frequently with the newest tools and data sources.

FLOOD MONITORING SYSTEM USING IOT
PHASE 3
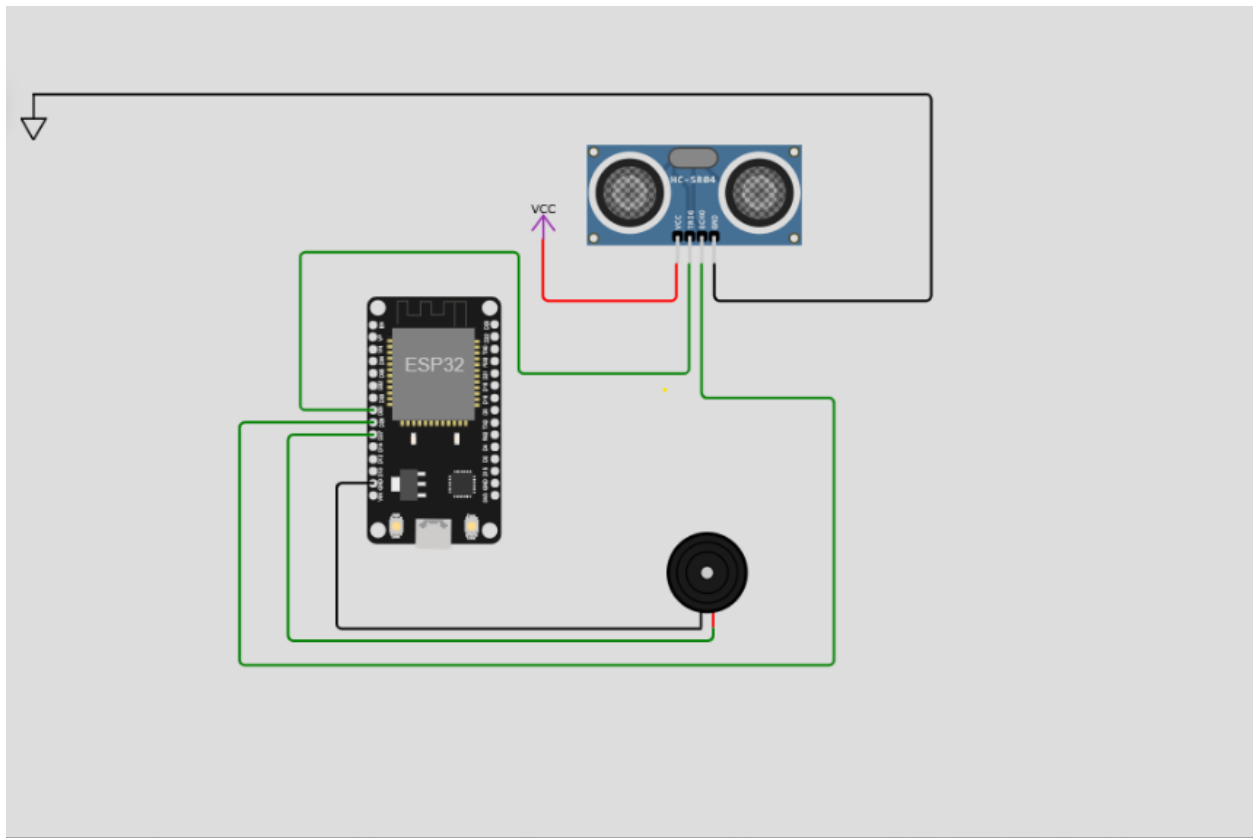# SENSOR DESIGN SIMULATION AND CODE

AIM:

To design and simulate water level monitoring sensor using ESP32 as an IoT Interface with python program.

COMPONENTS USED:

- SOFTWARE USED- WOKWI
- IoT DEVICE -ESP32
- SENSOR USED -Ultrasonic sensor
- Buzzer

CIRCUIT DESIGN:

CODE:

```
import machine
import time

# Define GPIO pin numbers
TRIG_PIN = 2  # GPIO2 for Trigger
ECHO_PIN = 4  # GPIO4 for Echo
BUZZER_PIN = 5  # GPIO5 for Buzzer
WATER_LEVEL_THRESHOLD = 30  # Adjust this value as needed (in
centimeters)

# Initialize GPIO pins
trig = machine.Pin(TRIG_PIN, machine.Pin.OUT)
echo = machine.Pin(ECHO_PIN, machine.Pin.IN)
buzzer = machine.Pin(BUZZER_PIN, machine.Pin.OUT)

while True:
    # Trigger the ultrasonic sensor
    trig.value(0)
    utime.sleep_us(2)
    trig.value(1)
    utime.sleep_us(10)
    trig.value(0)

    # Read the echo pulse duration
    duration = machine.time_pulse_us(echo, 1, 30000)  # 30,000us (30ms)
timeout

    # Convert the duration to distance (in centimeters)
    distance = duration / 58.0  # Speed of sound is approximately 343 m/s (34300
cm/s)

    print("Distance: {:.2f} cm".format(distance))

    # Check if the water level is below the threshold
    if distance < WATER_LEVEL_THRESHOLD:
```
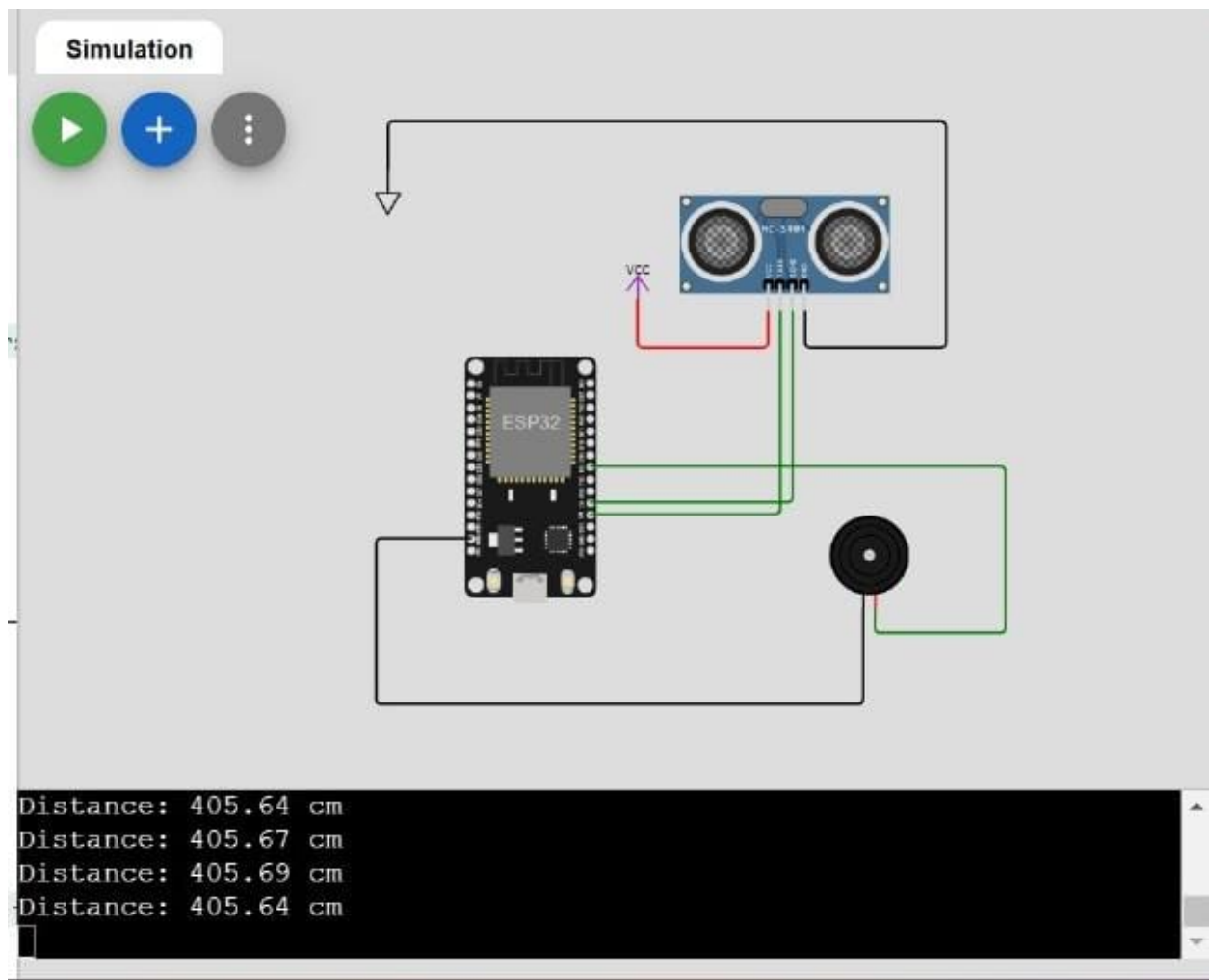
```
    # Water level is below the threshold, sound the buzzer
    buzzer.value(1)
else:
    # Water level is above the threshold, turn off the buzzer
    buzzer.value(0)

utime.sleep(1)  # Delay for 1 second to avoid continuous readings
```
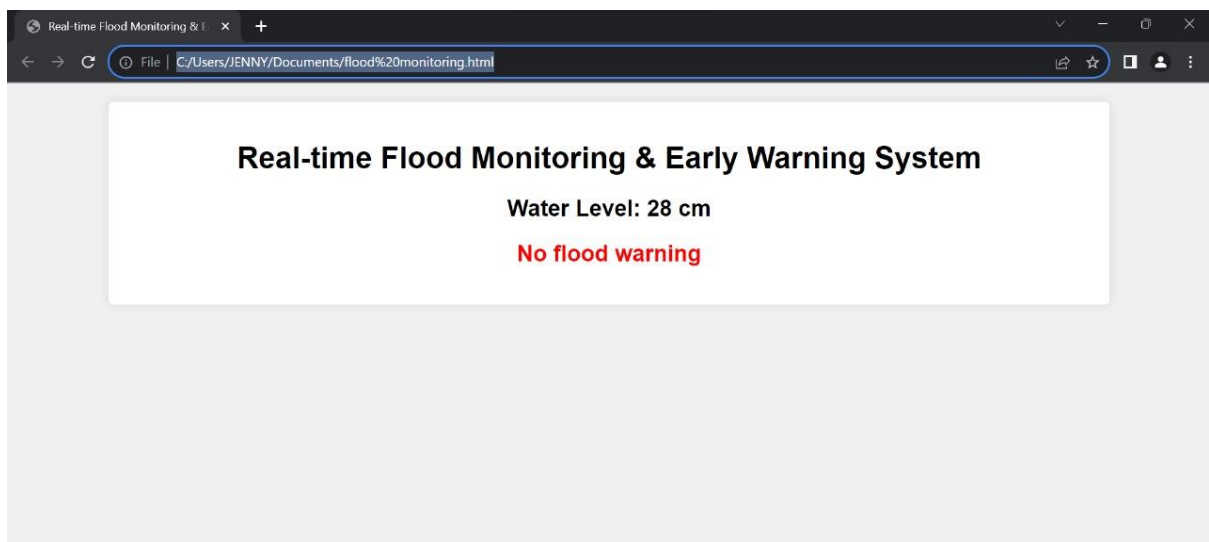
OUTPUT:

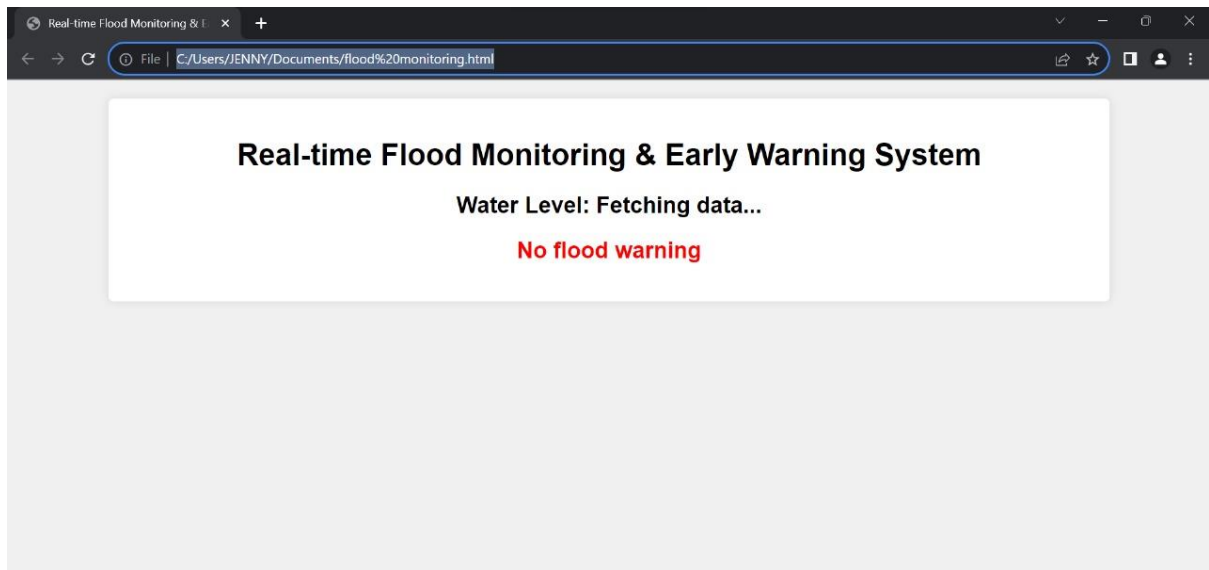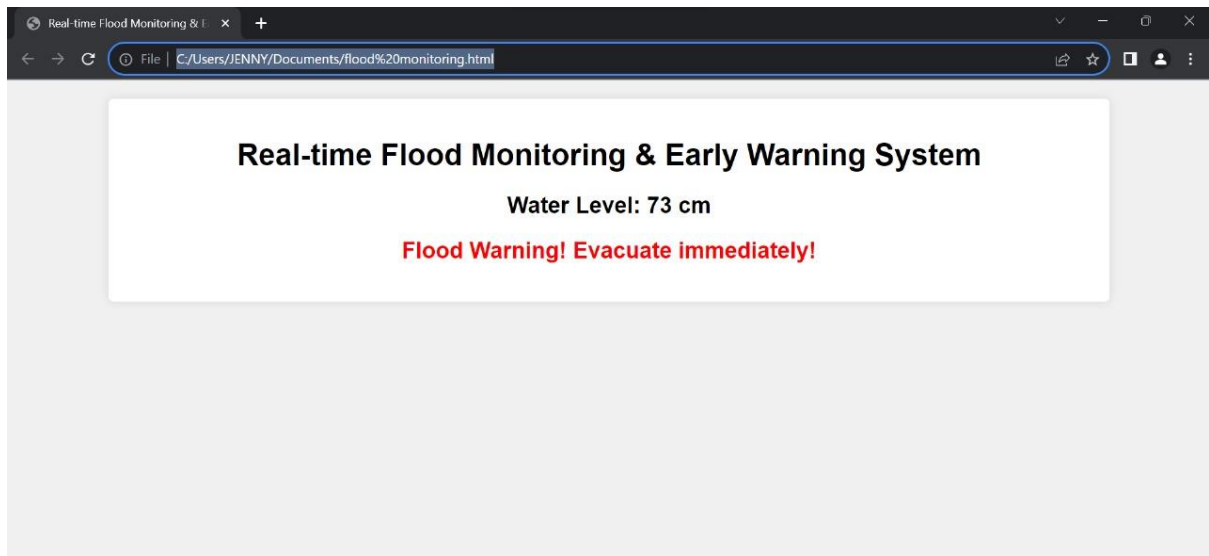# FLOOD MONITORING SYSTEM AND EARLY WARNING

PLATFORM DEVELOPMENT:

To use web development technologies (e.g., HTML, CSS, JavaScript) to create a platform that displays real-time water level data and flood warnings.

To design the platform to receive and display water level data from IoT sensors and issue flood warnings when necessary.

OUTPUTS:

CODE IMPLEMENTATION:

```
<!DOCTYPE html>

<html>

<head>

<title>Real-time Flood Monitoring & Early Warning System</title>

<style>

/* Basic styling for demo purposes - replace with your own styles */

body {
```

```css
      font-family: Arial, sans-serif;

      background-color: #f0f0f0;

      margin: 0;

      padding: 0;

    }

    #data-container {

    width: 80%;

    margin: 20px auto;

    padding: 20px;

    background-color: #fff;

    border-radius: 5px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    }

    h1, h2 {

    text-align: center;

    }

    #warning {

    color: red;

    }

    </style>

    </head>

    <body>

    <div id="data-container">

    <h1>Real-time Flood Monitoring & Early Warning System</h1>

    <h2>Water Level: <span id="waterLevel">Fetching data...</span></h2>

    <h2 id="warning">No flood warning</h2>

    </div>


    <script>

    // Simulating function to fetch real-time data (replace this with your actual data source)
```

```
function getRealTimeData() {

// Simulated data - replace with your data-fetching mechanism

return Math.floor(Math.random() * 100); // Random water level between 0 to 100 (for
demonstration)

}


// Function to update real-time data and warnings

function updateData() {

const waterLevel = getRealTimeData();

document.getElementById('waterLevel').textContent = ${waterLevel} cm;


if (waterLevel > 70) {

document.getElementById('warning').textContent = 'Flood Warning! Evacuate immediately!';

} else {

document.getElementById('warning').textContent = 'No flood warning';

}

}


// Update data and warnings every 5 seconds (for demonstration)

setInterval(updateData, 5000);

</script>

</body>

</html>
```

SUMMARY:

The IoT-based Flood Monitoring and Early Warning System is designed to track and forecast flood conditions in real-time, providing proactive warnings to communities and authorities to minimize the impact of floods. It utilizes IoT sensors, data processing, and web-based interfaces to effectively monitor water levels and issue warnings when flood risks are detected.