**DETAILED EXPLANATION**

To develop a flood monitoring and early warning system, several components and steps need to be considered:

1. **Data collection**: The system should collect data from various sources such as rainfall gauges, river level sensors, weather forecasts, and satellite imagery. This data will be used to monitor the current flood conditions and predict future flood events.

2. **Data processing and analysis**: The collected data needs to be processed and analyzed to identify patterns and trends that indicate potential flood risks. This can involve techniques such as data fusion, statistical analysis, and machine learning algorithms.

3***.* Real-time monitoring**: The system should continuously monitor the data and update the flood conditions in real-time. This can involve the use of IoT devices and sensors to collect data and transmit it to a central monitoring system.

4. **Early warning system**: Based on the analysed data and predefined thresholds, the system should issue timely warnings to the authorities and affected communities. These warnings can be sent through various channels such as SMS notifications, mobile apps, sirens, and public address systems

Implementation of IoT Sensors: IoT sensors are devices designed to measure and collect data related to flood monitoring system.

**PROBLEM DEFINITION:**

The project focuses on implementing IoT sensors to develop a system for flood monitoring and early warning. This system should be able to accurately detect and monitor flood conditions in real-time, and provide timely warnings to affected areas to minimize the potential damage and loss of life.

**FLOOD MONITORING AND EARLY WARNINGSYSTEMS**

5. **Visualization and reporting**: The system should provide a user-friendly interface that allows authorities and the general public to access and understand the flood information easily. This can include interactive maps, graphs, and reports that show the current flood conditions, predicted flood areas, and recommended evacuation routes.

6. **Integration with existing systems**: The flood monitoring and early warning system should be integrated with existing disaster management systems and emergency response mechanisms. This enables effective coordination and response during flood emergencies.

7. **Continuous improvement**: The system should be regularly updated and improved based on feedback and lessons learned from previous flood events. This can involve incorporating new data sources, refining algorithms, and enhancing the user interface.

Design thinking is a user-centric approach that can be applied to the development of a flood monitoring and early warning system. Here is a step-by-step process using design thinking principles:

1***.* Empathize**: Start by understanding the needs and challenges of the end-users, such as authorities responsible for flood management and the general public. Conduct interviews, surveys, and observations to gain insights into their experiences, expectations, and pain points related to flood monitoring and early warning.

2. **Define**: Based on the information gathered, define the specific problems and goals of the project. This could include identifying the key data sources, the accuracy and timeliness requirements of the system, and the desired user experience. Clearly define the scope and objectives of the project.

3. **Ideate**: Brainstorm and generate innovative ideas to address the defined problems and goals. Encourage creativity and collaboration among the team members. Explore different concepts, technologies, and approaches that can be incorporated into the flood monitoringand early warning system.

**DESIGN THINKING**

4. **Prototype**: Create a low-fidelity prototype of the system to visualize and test the proposed ideas. This can be in the form of wireframes, mockups, or even a simple interactive demo. The prototype should focus on the key features and functionality of the system, allowing stakeholders to provide feedback and make necessary improvements.

5. **Test**: Gather feedback from end-users and stakeholders by testing the prototype. Conduct usability tests, interviews, and surveys to understand how well the system meets their needs and expectations. Identify any usability issues, gaps, or areas for improvement.

6. **Iterate**: Based on the feedback and insights gathered during the testing phase, refine and iterate on the design of the system. Make adjustments to the features, user interface, and functionality to address the identified issues and enhance the user experience.

7***.* Implement**: Once the design has been finalized, proceed with the implementation of the flood monitoring and early warning system. Develop the necessary software, hardware, and infrastructure components, ensuring that they align with the user requirements and design specifications.

8.**Evaluate**: Continuously evaluate the performance and effectiveness of the implemented system. Monitor its functionality, accuracy, and reliability in real-world scenarios. Collect feedback from end-users and stakeholders to identify areas for further improvement.

**IMPORTANT COMPONENTS**

# Several important components are necessary for the development of a flood monitoring and early warning system:

* Data collection devices:
* Data transmission and communication:
* Central monitoring system:
* Early warning system:
* User interface and visualization:
* Integration with existing systems
* Continuous monitoring and evaluation:

.****

After thorough research and analysis, we arrived at an innovative solution to solve the above problem as detailed in phase 1 of our project

1. We will be using the ESP32 micro controller as well as Arduino UNO microcontroller as both these suit the best for our project

2. We chose this because we only need water level data and post it to a public platform.

3. Non local processing of data is required and hence we chose not to use Raspberry Pi Single board computer

**INNOVATION:**

**SENSORS:**

**Ultrasonic distance sensor HC-SR04**

An ultrasonic distance sensor is a device used in flood monitoring system to measure water level in rivers, lakes, and other bodies of water. It operates by emitting high –frequency sound waves that bounce off the water’s surface and return to the sensor. By measuring the time it takes for the sound waves to travel and return, the sensor calculates the distance to the Water’s surface. The real-time , non-contact measurement data s crucial for early flood detection and issuing timely warnings, helping to mitigate the impact of flooding on communities and infrastructure.



The ESP32 is a popular microcontroller chip known for its integrated Wi-Fi and Bluetooth capabilities. It is widely used in varies internet of things (IOT) and embedded system projects.ESP32 -based development boards provide an affordable and accessible solution for connecting devices to Wi-Fi networks, enabling communication, data transfer, and control over the internet. The ESP32’s Wi-Fi functionality makes it suitable for application such as home automation, sensor monitoring, remote control and more. We can program the ESP32 using the Arduino IDE .

**CONNECTIVITY:**

**Wi-Fi:**

**Sensors**

**CLOUD**

**Beeceptor:**

Beeceptor is a tool that allows you to easily create HTTP endpoints to collect and inspect HTTP requests. It's useful for testing and debugging.

**PROTOCOL**

**HTTP(Hypertext Transfer Protocol):**

HTTP is a foundational communication protocol used in flood monitoring and early warning systems. It facilitates the retrieval ,transmission and sharing of critical data related top water levels, weather conditions and flood risk. HTTP enables remote monitoring real time alerts data visualization and integrating with external services, enhancing the system’s ability to provide timely information and support decision-making for flood managements and early warnings.

One public platform commonly used in flood monitoring and early warning systems is the Global Flood Awareness System (GloFAS). Here are some of its key features:

1. Real-time Monitoring: GloFAS provides real-time monitoring of river flows and flood potential across the globe. It uses weather forecasts, satellite data, and hydrological models to generate flood forecasts and updates.

**PUBLIC TRANSFORM**

**Features:**

2.Early Warning Systems: The platform issues early warnings by predicting flood events up to 30 days in advance. It helps authorities and communities to prepare and respond effectively.

3. Global Coverage: GloFAS covers multiple countries and regions worldwide, allowing for a comprehensive assessment of flood risk on a global scale.

4.Interactive Maps: The platform offers interactive maps that display flood forecast information, river flows, and flood extent. Users can access these maps to visualize flood risk and plan accordingly

5.Data Sharing: GloFAS promotes data sharing and collaboration by providing access to its flood forecast data and information to governments, organizations, and researchers. This facilitates informed decision-making and supports the development of localized flood early warning systems.

6.Integration with Disaster Management Systems: The platform can be integrated with existing disaster management systems and emergency response frameworks, enabling seamless coordination and communication during flood events.

**PHASE 3:**

**Problem:**

To develop the python script on IOT devices for flood monitoring and early warning as per the project requirement

**Solution:**

Flood monitoring and early warning can be implemented using an

ESP32 and sensor like ultrasonic sensor to detect the

water level and to find out the distance of water level.

**Source code:**

#define BLYNK\_TEMPLATE\_ID "TMPL3tobBFjj-"

#define BLYNK\_TEMPLATE\_NAME“IOT FLOOD MONITORING”

#define BLYNK\_AUTH\_TOKEN"gy2bzRRbPW3oWOpAiDgr6sSVzIHVZ"

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <ESP32Servo.h>

Servo gate;

const int trigPin=2;//d2

const int echoPin=4;//d4

const int servoPin = 18;//d18

long duration;

int distance;

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

gate.attach(servoPin, 500, 2400);

}

void loop()

{

digitalWrite(trigPin, LOW);

delay(2);

digitalWrite(trigPin,HIGH);

delay(10);

digitalWrite(trigPin, LOW);

if(distance<100)

{

gate.write(90);

Blynk.virtualWrite(V1," SAFE CONDITIONS GATES CLOSED ");

}

else

{

gate.write(0);

Blynk.virtualWrite(V1," FLOOD DETECTED GATES OPENED ");

}

**Include Libraries:**

Wi-Fi.h - This library provides the necessary functions to connect the ESP32 to a Wi-Fi network.

Wi-FiClient.h0-This library allows the ESP32 to create a client that can connect to a specified internet IP address and port.

BlynkSimpleEsp32.h - This library provides the Blynk-related functions for the ESP32. It allows you to easily connect your ESP32 to the Blynk IoT platform.

ESP32Servo.h - This library enables you to control servo motors with an ESP32

Alerts will be given based on the reading of ultrasonic sensor to

the authorized person bt blynk app.

**Alerts and Notifications:**

Read the distance from the ultrasonic sensor using the sonar.ping\_cm() function. Check if the flood level exceeds the defined threshold by calculating distance from ultrasonic sensor. If the conditions are match, send Alert function by blynk.

**Loop Function:**

Initialize the ultrasonic sensor with its D2 as the output pin and D4 as the output pin. Sent the comments using Blynk.

**Initialize Objects and Variables:**

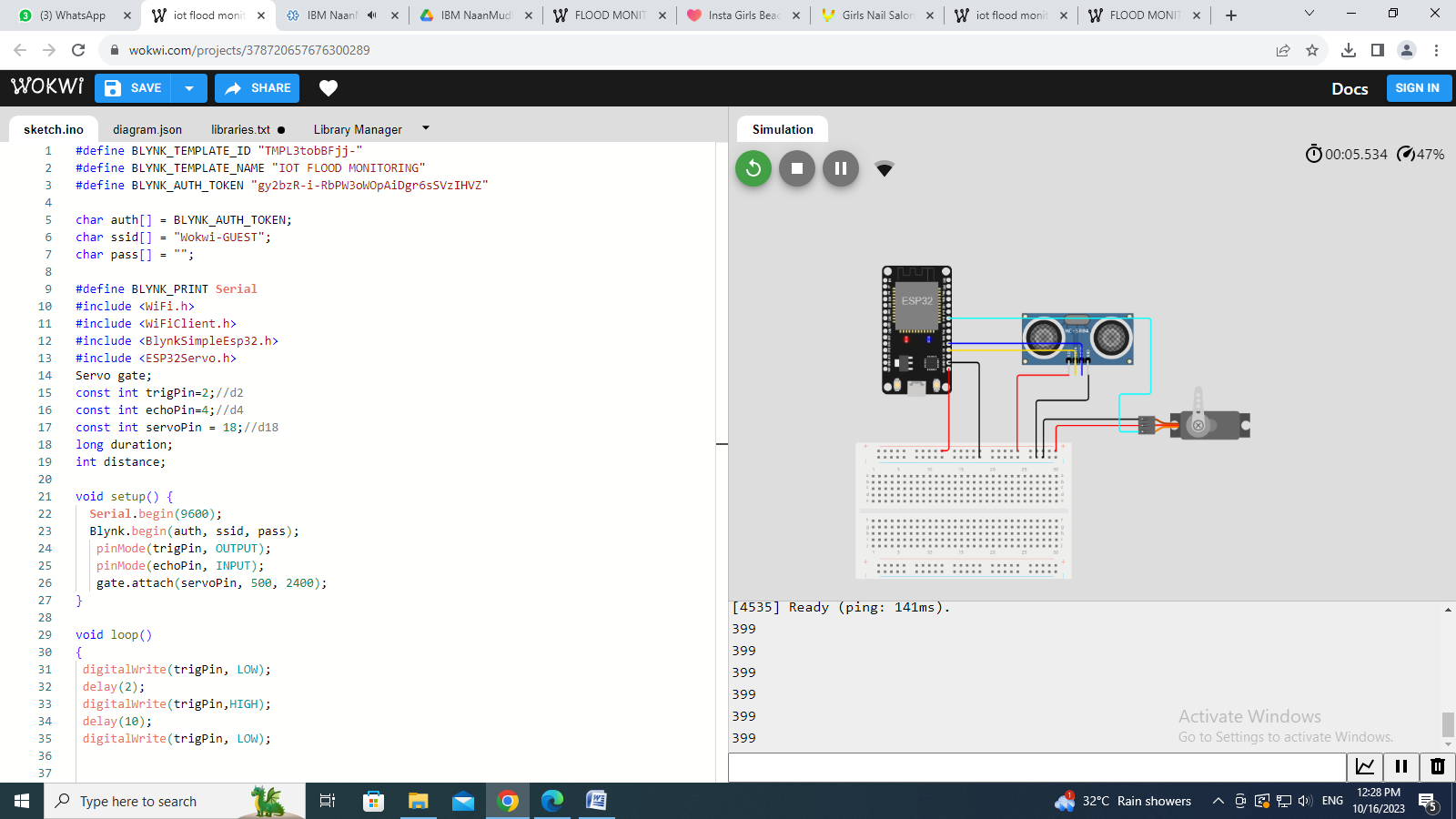
**Setup Function:**

Define the pins for the ultrasonic sensor (TRIGGER\_PIN and ECHO\_PIN).

Define the pin for the servo for the gates to be opened and closed.

<https://wokwi.com/projects/378731625245404161>

**Simulation link:**



**Simulation Image:**

**BY,**

**Fathima Nabisathul Sameera S(2021504006)**

**Megashree G (2021505022)**

**Srithika S (2021504046)**

**Keerthana K A (2021504017)**

**NaveenaaSuresh Babu (2021504027)**