Phase 1: Problem Definition and Design Thinking

PROJECT TITLE	IOT FLOOD MONITORING AND EARLY WARNING
NAME	R.PRIYADHARSNI
TEAM ID	5246
TEAM NAME	PROJ_204183_TEAM_1
COLLEGE CODE - NAME	9238 – MANGAYARKARASI COLLEGE OF ENGINEERING PARAVAI,MADURAI.
GROUP	5
REPOSITORY	https://github.com/priyadharsni/democode.git

1. Project Definition

1.1 Project Overview:

In this project we are chosen to reduce the dead rate during the occurrence of the flood and measure the water level in the dam . We includes alert message to all the mobile phones .so our main aim is to alert the people before the flood.so we will block all the server of other apps except phone, whatspps.

1.2 Project Objectives:

1. **DATA COLLECTION :** To gather real-time data on rainfall ,river levels and other relevant meteorological and hydrological parameters

2. **EARLY WARNING:** To provide early warning and alerts to communities in flood- prone areas,helping them prepare and evacuate in advance of flooding events

3.RISK ASSESSMENT: To assess and map flood risk area,identifying vulnerable population, critical infrastructure and valuable assets that may by affected by floods.

4. PUBLIC AWARENESS: To raise awareness among the public about flood risks, safety measures and evacuation procedures through educational campaings and community out reach

5. MOBILE APP INTERGRATION: Develop a user-friendly mobile application that provide awareness information to the user.

2. DESIGN THINKING

2.1 IOT SENSOR DESIGN

We select the IOT sensors for monitoring the water quantity. These sensors are used in our project

- Rain Gauges
- Weather Stations
- Flow Meters
- Water Temperature Sensors
- GPS Sensors
- Cameras and Image Sensors
- Cellular Modems
- Doppler Radar Sensors
- Barometric Pressure Sensors
- LiDAR (Light Detection and Ranging)
- Motion Sensors
- Ultrasonic sensor
- Float sensor

2.1.2 SENSOR DEPLOYMENT:

1. Sensor Placement: Strategically place sensors in each dams. Sensor ensures alert messages with images ,water level, temperature, rainfall range,location.

2. Communication: Set up a wireless communication network (e.g., Wi-Fi) for sensors to transmit data to all people

3. Power Supply: Ensure a reliable power source for sensors through batteries,wired connection.

2.2. Real-Time Transit Information Platform

2.2.1. Mobile App Interface

1. User Interface (UI): Design an intuitive and user-friendly interface for the mobile app. Include features such as a map view in flood occurring place

2. Real-time Updates: Implement a mechanism to continuously update the app with real-time.

2.3. Integration Approach

2.3.1. Data Collection with Arduino

1. Sensor Selection: We choose a suitable Arduino-compatible sensors for monitoring flood level

2. Arduino Board Selection: We Select an Arduino board that meets our project&#39;s requirements.

3. Sensor Wiring: We Connect the sensors to the Arduino board according to the manufacturer&#39;s specifications..

4. Data Acquisition: We Write the Arduino code to read data from the sensors. Arduino&#39;s analog and digital pins can be used to interface with various sensors.

5. Data Processing: We Implement the code on the Arduino to process sensor data .

2.3.2. IoT Cloud Integration

1. Cloud Platform Selection: We Choose an IoT cloud platform that fits our needs and include the AWS IoT, Google Cloud IoT, and Microsoft Azure IoT, IoT WATSON.

2. Cloud Account Setup: We Create an account and set up our IoT project on the chosen cloud platform.

3. Device Registration: We Register our Arduino-based device as an IoT thing on the cloud platform. You will receive credentials (such as certificates and keys) for secure communication.

4. Data Transmission: we Configure the Arduino to use the MQTT (Message Queuing Telemetry Transport) or HTTPS protocol to send sensor data to the cloud platform securely.

2.3.3. Data Processing in the Cloud

1. Cloud Functionality: Set up cloud functions or serverless computing services on our cloud platform to receive and process incoming sensor data.

2. Data Storage: Store processed data in a cloud-based database or storage service (e.g. Amazon DynamoDB, Google Cloud Firestore, or Azure Cosmos DB).

3. Real-time Processing: We Implement the real-time data processing to determine the flood level.

2.3.4. Mobile App Integration

1. Mobile App Development: We develop a mobile app (iOS/Android) that communicates with the cloud-based IoT platform.

2. API Communication: Integrate APIs or SDKs provided by the cloud platform to enable secure communication between the mobile app and the cloud.

3. Real-time Updates: We Implement the real-time updates within the mobile app.

2.3.5. User Interface

1. App Interface: Design an user-friendly interface for the mobile app.

2.3.6. Testing and Deployment

1. Testing: Conduct thorough testing of the integrated system, including Arduino sensor data collection, cloud-based data processing, and mobile app functionality. Address any issues or bugs encountered during testing.

2. Deployment: Once testing is successful, deploy the integrated system . Ensure all sensors are correctly placed and communicate effectively with the cloud platform.

3. User Training: If necessary, provide user training on how to use the mobile app