***Development of early warning platform***

1.First-End User Interface:

Use JavaScript to create a web-based user interface. Popular front-end frameworks or libraries like Angular, Vue.js, or React can be used for this.

2. Visualization of Real-Time Data:

To display sensor data in real-time, such as temperature, humidity, water level, etc., integrate a charting library such as Chart.js or D3.js.

3. Device Integration for IoT:

Using IoT protocols like MQTT, CoAP, or HTTP, connect IoT sensors to your platform. To communicate with these sensors, you might require a microcontroller such as an Arduino or Raspberry Pi.

4. Reverse Server:

To manage data processing, storage, and communication with IoT devices, set up a backend server using Node.js or any other server-side technology.

5.Database:

Select a database system (such as PostgreSQL, MySQL, or MongoDB) to hold flood warning data and previous sensor data.

6. Instantaneous Communication:

For real-time communication between the server and the front-end, use WebSocket or Server-Sent Events (SSE).

7. Warning System for Floods:

Create an algorithm to evaluate sensor data and send out flood alerts based on predetermined thresholds, or incorporate a third-party service.

8.User Notices:

Establish a notification system to notify users of flood warnings by email, SMS, or in-app notifications.

9.Security:

By using appropriate authorization and authentication procedures and encrypting data transferred between devices and the server, you can be sure that your system is safe.

10. Checking and Observing:

Make sure your system is reliable by giving it a comprehensive test, and set up monitoring to find and fix problems.