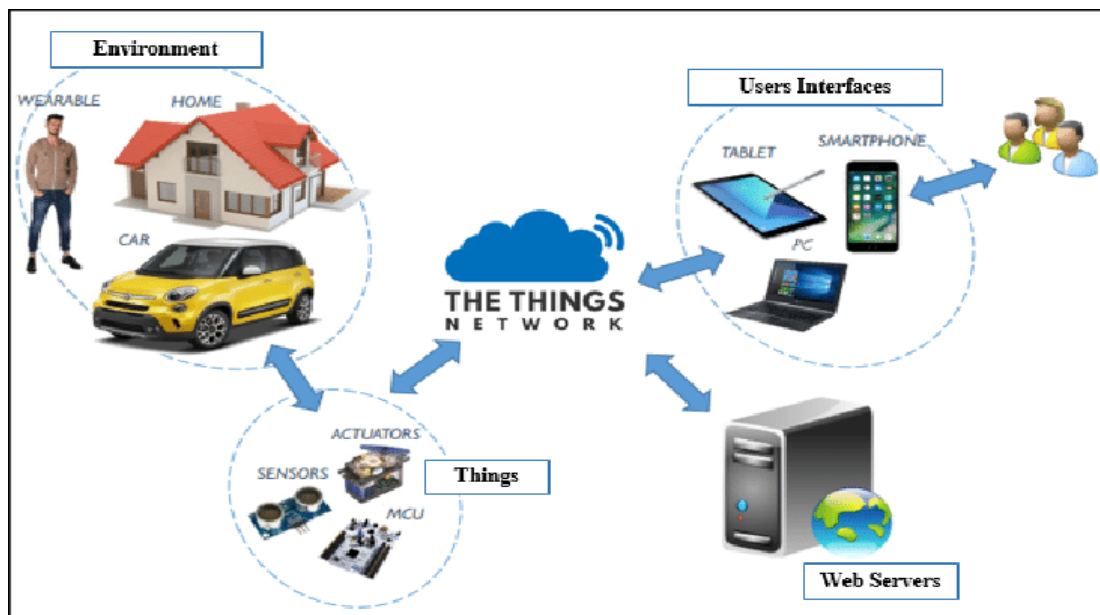


# Environmental monitoring

Creating an environmental monitoring platform that displays real-time temperature and humidity data from IoT devices involves several steps. Here's a high-level overview:



## Setting up the IoT Devices:

First, ensure that you have IoT devices (e.g., sensors) capable of measuring temperature and humidity.

Configure the IoT devices to transmit data to a server using protocols like MQTT or HTTP.

### Server-Side Development:

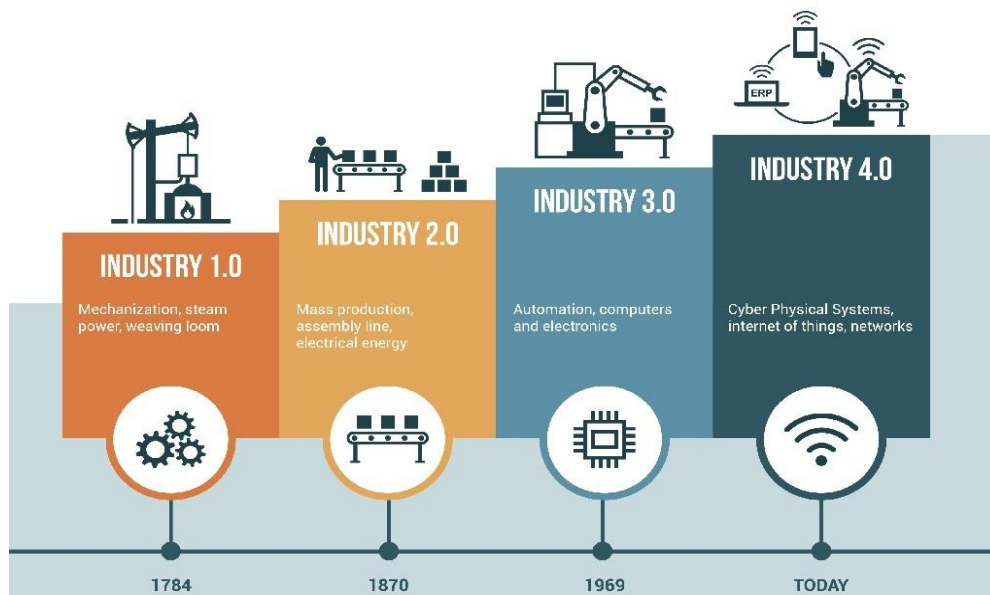
Develop a server application (e.g., Node.js) that receives data from the IoT devices.

Store this data in a database for real-time access.

### Database Design:

Choose a database system (e.g., MySQL, MongoDB) to store the environmental data.

Create a database schema to organize the data.



**Fig: environmental monitoring on industries**

### API Development:

Create APIs that allow IoT devices to send data to the server.  
Implement authentication and security measures to protect the data.

### Front-End Development:

Use HTML, CSS, and JavaScript to create a web-based user interface for the platform.  
Utilize frameworks like React, Angular, or Vue.js for a more dynamic interface.

### Real-Time Data Display:

Implement real-time data updates using technologies like WebSocket or Server-Sent Events.  
Use JavaScript libraries (e.g., Chart.js) to create interactive charts for temperature and humidity.

### User Authentication:

Implement user authentication to control access to the platform.  
Allow users to register and log in to view the environmental data.

### Visualization and User Interface:

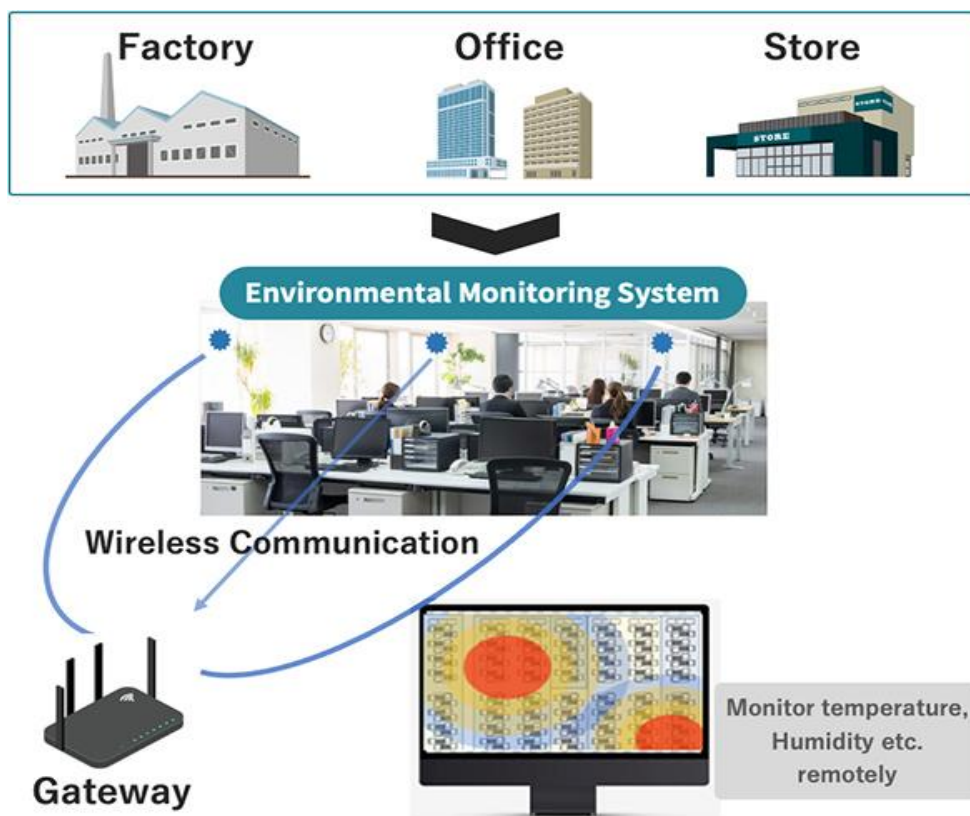
Design a user-friendly interface that displays real-time temperature and humidity data.  
Include features like data filtering, time-based graphs, and alerts.

## Mobile Responsiveness:

Ensure that the platform is responsive and works well on various devices, including smartphones and tablets.

## Testing and Debugging:

Thoroughly test the platform to identify and resolve any issues. Consider load testing to ensure it can handle a significant number of IoT devices and users.



**Fig: environmental monitoring systems**

### Deployment:

Deploy the server application and database on a cloud platform or a dedicated server.

Ensure proper scalability and redundancy.

### Security Measures:

Implement security measures to protect the platform from unauthorized access and data breaches.

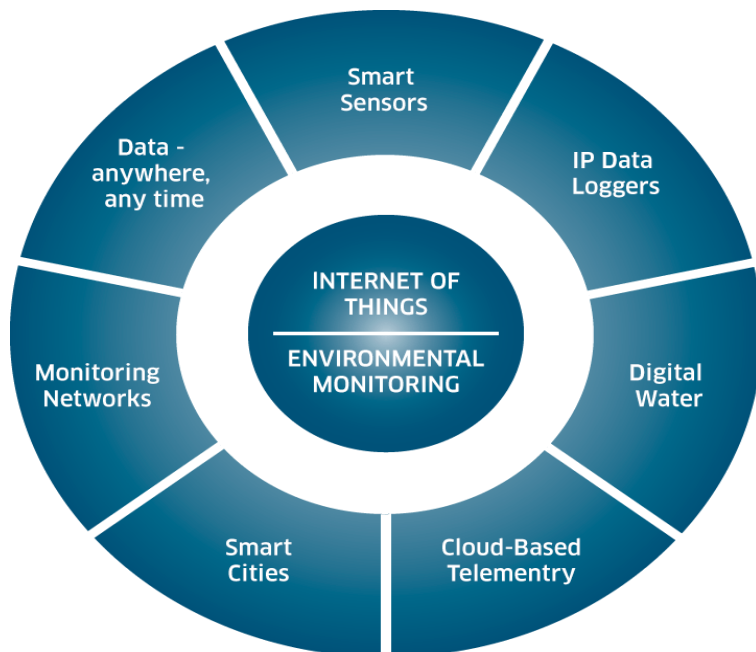


Fig: environmental monitoring in iot

### Documentation:

Create documentation for users and administrators on how to use and maintain the platform.

## Monitoring and Maintenance:

Set up monitoring tools to track system health and performance. Regularly update and maintain the platform to address security vulnerabilities and improve functionality.

## Integration:

Consider integrating with third-party services for additional features, such as weather forecasts or data analysis.

This is a complex project that requires careful planning and development. Be sure to follow best practices in web development, security, and IoT to create a robust and reliable environmental monitoring platform.

## Coding:

### HTML:

```
<!DOCTYPE html>  
<html lang="en">  
<head><meta charset="UTF-8">
```

```
<metaname="viewport" content="width=device-widthinitial
-scale="1.0">
<link rel="stylesheet" href="style.css">
<title>Smart Water Management System</title>
</head>
<body>
<header>
<h1>Environmental monitoring System
</h1>
</header>
<main>
<section class="sensors">
<h2>
Sensor Data
</h2>
<div id="sensorData">
<!--Sensor data will be displayed here dynamically
</div>
</section>
<section class="control-panel">
<h2>Control Panel</h2>
<button id="startMonitoring">Start Monitoring>
</button>
<button id="stopMonitoring">Stop Monitoring>
</button>
</section>
</main>
<script src="script.js">
</script>
</body>
</html>
Css:
```

```
Body {
Font-family: Arial, sans-serif;
}
Header
{Background-color: #3498db;
Color: #fff;
Text-align: center;
Padding: 20px;
}
Main
{
Display: flex;
Justify-content: space-around;
Margin: 20px;
}
Section
{
Border: 1px solid #ddd;
Padding: 20px;
Margin: 10px;
}
Button
{
Display: block;
Background-color: #3498db;
Color: #fff;
Border: none;
Padding: 10px 20px;
Margin: 10px;
Cursor: pointer;
}
Button:hover
```



```

{
Background-color: #2980b9;}
body {Font-family: Arial, sans-serif;
}
JAVASCRIPT:
// Simulated sensor data
Const sensors = [
{ name: 'Temperature', value: '25°C' },

{ name: 'Humidity', value: '50%' },
{ name: 'Flow Rate', value: '10 L/min' }];
// Display initial sensor data
Const sensorDataContainer =
document.getElementById('sensorData');
Sensors.forEach(sensor =>
{
Const sensorDiv = document.createElement('div');
sensorDiv.innerHTML =
`<strong>${sensor.name}</strong>${sensor.value}
`;sensorDataContainer.appendChild(sensorDiv);
}
);
// Start and stop monitoring buttons
Const
startMonitoringButton=document.getElementById('startMonitoring');
Const stopMonitoringButton =
document.getElementById('stopMonitoring');
startMonitoringButton.addEventListener('click', () =>
{
// Start monitoring logicAlert('Monitoring started');
});
stopMonitoringButton.addEventListener('click', () =>

```

```
{// Stop monitoring logicAlert('Monitoring stopped');  
});
```

## **Conclusion:**

Environmental monitoring is critical to the protection of human health and the environment. As the human population continues to increase, as industrial development and energy use continues to expand, and despite advances in pollution control, the continued production of pollution remains inevitable. Thus the need for environmental monitoring is still as great as ever. Continued advances in the development, application, and automation of monitoring devices are needed to enhance the accuracy and cost-effectiveness of monitoring programs. Equally as important is the need to produce more scientists and engineers that have the knowledge and training required to successfully develop and operate monitoring devices and manage monitoring programs.