

Campus Green Automation Using IBM Cloud

Abstract:

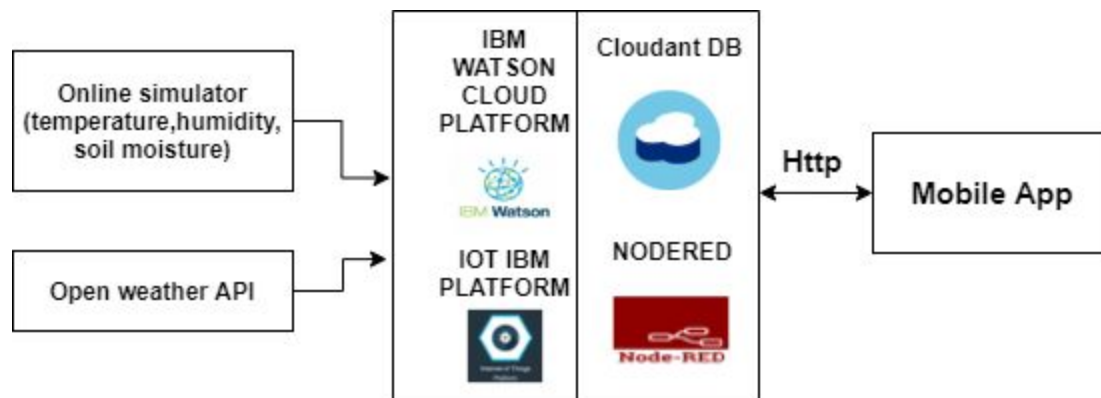
Campus Green project aims at developing and maintaining a greening project in Sreepathy Institute of Management and Technology, Vavanoor, Palakkad. In this project, we propose a **Real-Time Weather Based Smart Sprinkler System** to maintain the greenery of the campus. The main objective of the system is real-time monitoring of the proposed area over which the campus green project is set up. The system monitors the soil and environment conditions in realtime and also gets weather forecast data from OpenWeather API. Based on the sensor data and the weather forecast data, automated sprinklers will be operated which maintains a healthy environment for plants and trees. The data are stored in the IBM cloud and visual analysis is provided with the help of a web application and android application. Manual operation of motors is possible through the web and mobile UIs. Also, the system provides real-time notification over user mail about the present conditions of weather and sprinkler systems.

Introduction:

Currently, the campus green project is realized using manual effort. A person in charge needs to visit the place and manually inspect whether the plants and trees are having a healthy environment. If there is high temperature and less soil moisture then the person needs to manually operate the watering system or motors to provide healthy growing conditions for the trees. The watering system needs to be monitored as long as it is turned ON which increases the human load. If not monitored correctly, it may result in loss of water. To avoid these drawbacks, a **Real-Time Weather Based Smart Sprinkler System** is proposed. The IoT based system makes the watering process more efficient and reduces a lot of human load.

Proposed Solution:

In the proposed project the soil moisture, temperature and humidity levels at Sreepathy are continuously monitored with the help of sensors. These values are updated to the IBM IoT platform and the data is stored in Cloudant DB. The system also fetches data from open weather APIs and automatically controls the sprinkler system. The data can be visualized with the help of web and android application interfaces.



Features of the Proposed Solution:

- Real-time weather data (Sensor & API Data)
- Automated operation of sprinklers

- Alert notifications based on weather conditions
- Web/App interface to visualize weather conditions
- Cloud storage to keep track of weather conditions.
- Remote operation of Sprinklers using Android application

Project Flow:

- Configure and connect the online simulator to publish temperature, humidity, and soil moisture values to the IBM IoT Platform.
- Create a Node-RED flow to get the data from the IBM IoT platform and store it in Cloudant DB.
- Create HTTP APIs in Node-RED to send the sensor data to the mobile app and also to get the commands from the mobile applications.
- Create a mobile app to visualize the sensor parameters and also to get the open weather data.
- Configure the mobile app to send commands to the IBM IoT Platform to control the sprinklers based on the sensor values and weather details.
- Send notifications to users based on weather conditions.

The screenshot shows the IBM Watson IoT Platform interface. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. Below this, there's a 'All Devices' button and a 'Diagnose' button. A message states: 'This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.' Below this is a search bar 'Search by Device ID' and a 'Device Simulator' toggle switch. The main table lists devices with columns: Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. The table contains four rows of data, all with a status of 'Disconnected'. At the bottom right, there's a status indicator '1 Simulation running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	node_mcu	Device	Oct 1, 2020 6:19 PM	
sprinkler	Disconnected	node_mcu	Device	Oct 7, 2020 8:18 PM	
node_mcu_1	Disconnected	node_mcu	Device	Oct 7, 2020 8:45 PM	
motor	Disconnected	actuators	Device	Oct 12, 2020 8:28 PM	

Fig 1: IBM Watson IoT Platform

The screenshot shows the 'Recent Events' tab in the IBM Watson IoT Platform interface. It displays a table of recent events for a device. The table has columns: Event, Value, Format, and Last Received. The events are all from the 'iotsensor' and represent temperature and humidity data. The format is 'json' and the last received time is 'a few seconds ago'.

Event	Value	Format	Last Received
iotsensor	{"d":{"name":"motor","temperature":18,"humidit..."}	json	a few seconds ago
iotsensor	{"d":{"name":"motor","temperature":18,"humidit..."}	json	a few seconds ago
iotsensor	{"d":{"name":"motor","temperature":18,"humidit..."}	json	a few seconds ago
iotsensor	{"d":{"name":"motor","temperature":18,"humidit..."}	json	a few seconds ago

Fig 2: Values obtained from IBM online simulator

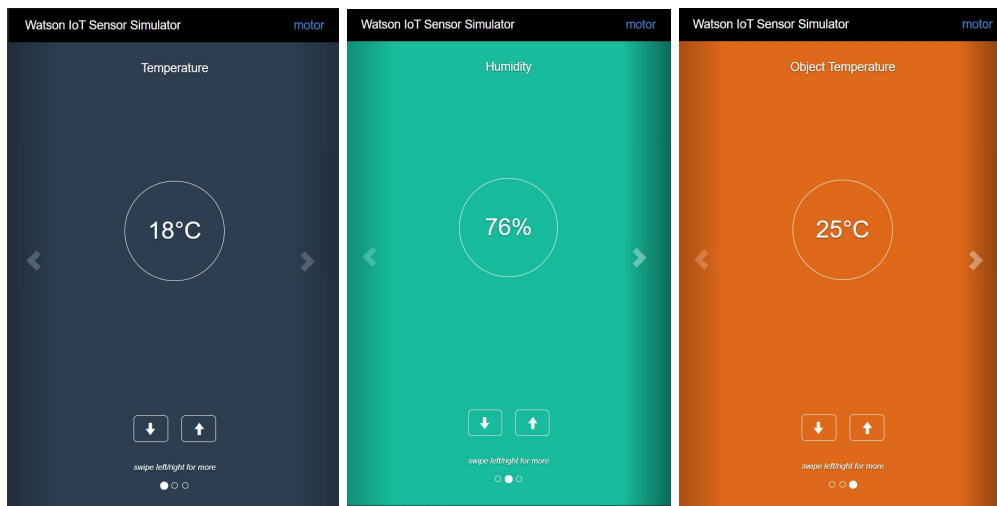


Fig 3: IBM Watson IoT Sensor Simulation

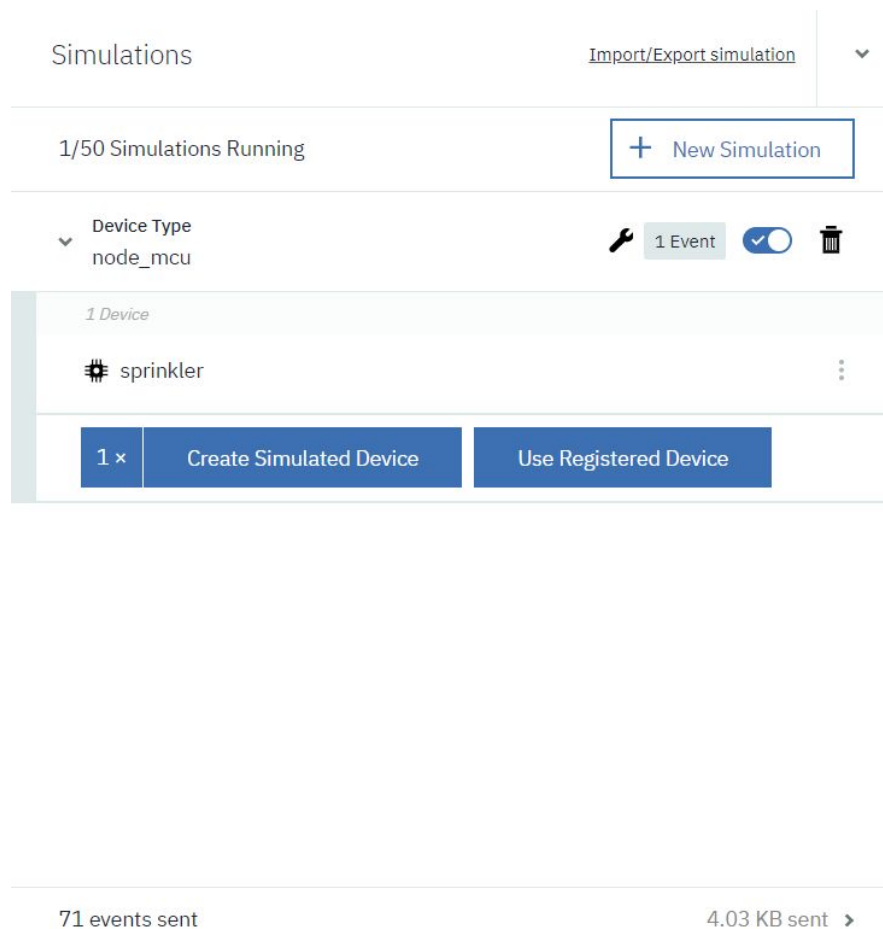


Fig 4: IBM Watson IoT Platform Sensor Simulation

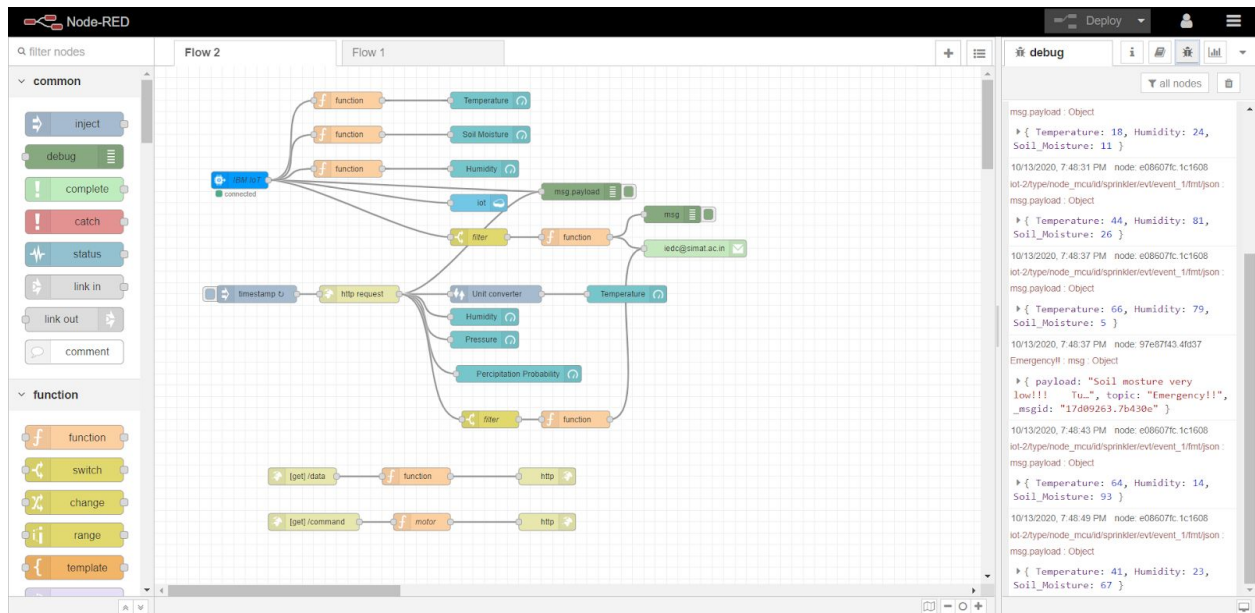


Fig 5: Node-Red Flow

Weather Data Obtained from OpenWeathe API at Sreepathy Institute of Management And Technology.

```
{
  "lat":10.76,
  "lon":76.14,
  "timezone":"Asia/Kolkata",
  "timezone_offset":19800,
  "current":{
    "dt":1602598792,
    "sunrise":1602549845,
    "sunset":1602592733,
    "temp":297.15,
    "feels_like":300.91,
    "pressure":1004,
    "humidity":94,
    "dew_point":296.12,
    "uvi":12.49,
    "clouds":75,
    "visibility":3000,
    "wind_speed":2.1,
    "wind_deg":360,
    "weather":[
      {
        "id":701,
        "main":"Mist",
        "description":"mist",
        "icon":"50n"
      }
    ]
  }
}
```

}

all nodes

msg.payload : Object

▶ { Temperature: 26, Humidity: 8, Soil_Moisture: 41 }

10/13/2020, 7:52:01 PM node: e08607fc.1c1608

iot-2/type/node_mcu/id/sprinkler/evt/event_1/fmt/json :

msg.payload : Object

▶ { Temperature: 83, Humidity: 53, Soil_Moisture: 74 }

10/13/2020, 7:52:07 PM node: e08607fc.1c1608

iot-2/type/node_mcu/id/sprinkler/evt/event_1/fmt/json :

msg.payload : Object

▶ { Temperature: 73, Humidity: 86, Soil_Moisture: 78 }

10/13/2020, 7:52:19 PM node: e08607fc.1c1608

iot-2/type/node_mcu/id/sprinkler/evt/event_1/fmt/json :

msg.payload : Object

▶ { Temperature: 75, Humidity: 14, Soil_Moisture: 95 }

10/13/2020, 7:52:24 PM node: e08607fc.1c1608

msg.payload : Object

▶ { lat: 10.76, lon: 76.14, timezone: "Asia/Kolkata", timezone_offset: 19800, current: object }

10/13/2020, 7:52:25 PM node: e08607fc.1c1608

iot-2/type/node_mcu/id/sprinkler/evt/event_1/fmt/json :

msg.payload : Object

▶ { Temperature: 62, Humidity: 58, Soil_Moisture: 63 }

Fig 6: Values obtained from IBM IoT Platform & Weather API

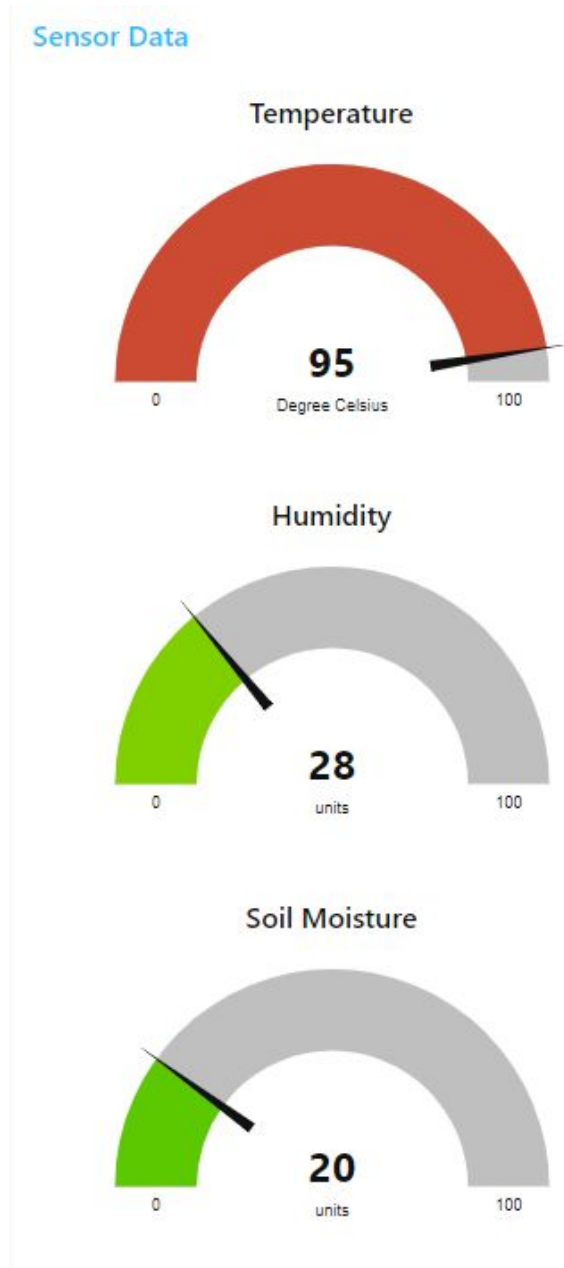


Fig 7a: Web Interface - Simulated Sensor Data

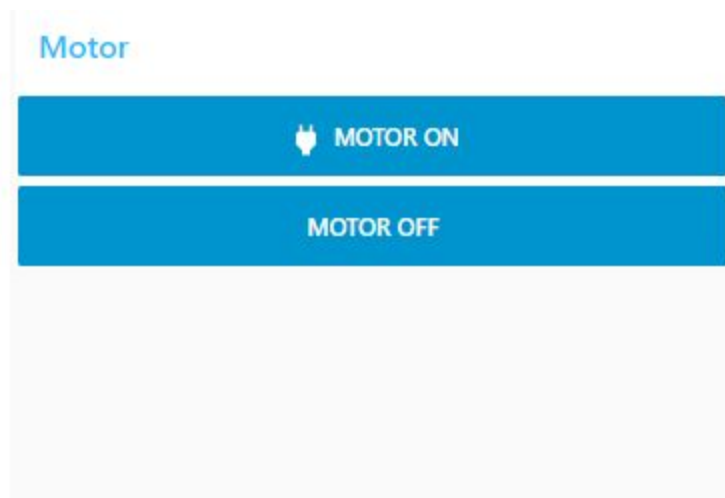


Fig 7b: Web Interface - Motor ON/OFF Control

Weather API

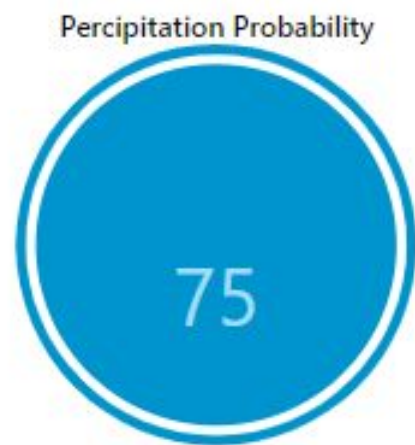
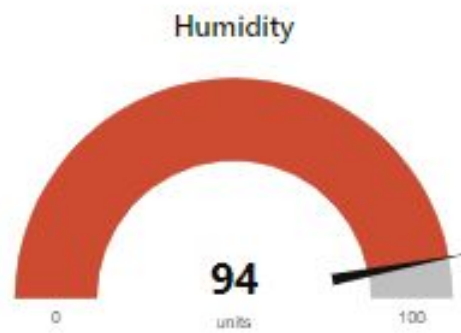
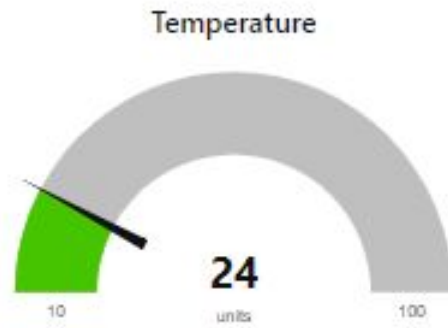


Fig 7c: Web Interface - API Weather Data

weather_data

Document ID

Options

{ } JSON

All Documents

Query

Permissions

Changes

Design Documents

Table

Metadata

{ } JSON

Create Document

	id	key	value
<input type="checkbox"/>	0c1820f197dfa6d85915c77900bdf3f	0c1820f197dfa6d85915c77900bdf3f	{ "_rev": "1-6c312b4d644fbd217d1f2dfdbfc57..."
<input type="checkbox"/>	d3a53d8cc71462d799ade339f6bf3468	d3a53d8cc71462d799ade339f6bf3468	{ "_rev": "1-870a98ed74d178c69eceb9a5d84b..."
<input type="checkbox"/>	ecd49a6f47f35c89a87d8de81fba6149	ecd49a6f47f35c89a87d8de81fba6149	{ "_rev": "1-870a98ed74d178c69eceb9a5d84b..."
<input type="checkbox"/>	f596fb762284638db38e6267b0c3cd4a	f596fb762284638db38e6267b0c3cd4a	{ "_rev": "1-870a98ed74d178c69eceb9a5d84b..."

iot > 01ddf47fd045e519289c04001e728f97

{ } JSON

Save Changes

Cancel

Upload Attachment

Clone Document

Delete

```

1
2 { "_id": "01ddf47fd045e519289c04001e728f97",
3   "_rev": "1-e1ef8d0659c2a6398c638011b107aaf8",
4   "Temperature": 64,
5   "Humidity": 9,
6   "Soil_Moisture": 43
7 }

```

Fig 8: Data Stored in IBM Cloud

Emergency!!

Inbox x

guidance@simat.ac.in

to me

Soil moisture very low!!!

Tue Oct 13 2020 14:09:00 GMT+0000 (Coordinated Universal Time)

Fig 9: e-mail Notification on Low Soil Moisture Levels

Android Application Interface



Campus Green
Sreepathy Institute of Management & Technology

Powered By
IBM Cloud

Sensor Data

Data

Weather Data

UI



Temperature

41

Humidity

44

Soil Moisture

27

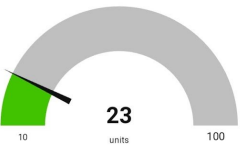
Motor ON

Motor OFF

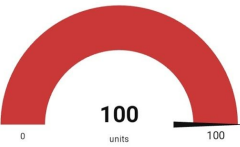
Weather Data

Weather Data

Temperature



Humidity



Pressure

