**SEVAK EXECUTION GUIDE**

**CONTENTS**

* **Introduction**
* **Prerequisites**
* **Hardware**
* **Hardware setup**
* **IBM account**
* **How to use Sevak?**
* **How to access uploaded files on Cloud?**
* **Conclusion**

**Introduction**

Sevak is an IoT based project developed for the sole purpose of providing assistance and aid to the underground sanitation workers forced to lead unhealthy lives filled with dangerous risks due to the nature of their profession. In India, poverty is the main factor that pushes the people into this hazardous profession. The government is unable to provide them better facilities to ensure the safety of these workers who are responsible for the clean and disease free towns and cities in the country. In most cases, they enter the underground drainage without any protective equipment. We aim to develop a head gear and a watch that can monitor the environment conditions in which they work.

This project includes the following features:

**● Sensor to detect fall**

**● Harmful gas detection**

**● Heart rate sensor detection**

**● Temperature detection**

**● Obstacle detection**

**● Location of the worker**

**● Alerts when the harmful gas levels goes above threshold level**

**● Alerts when the pulse of the worker falls below normal BPM**

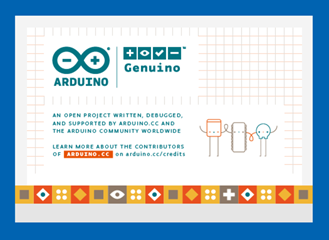
**● Emergency button for the drainage worker in case he gets trapped**

**● Employee Database**

**Prerequisites**

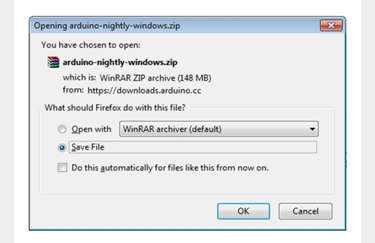
**Software**

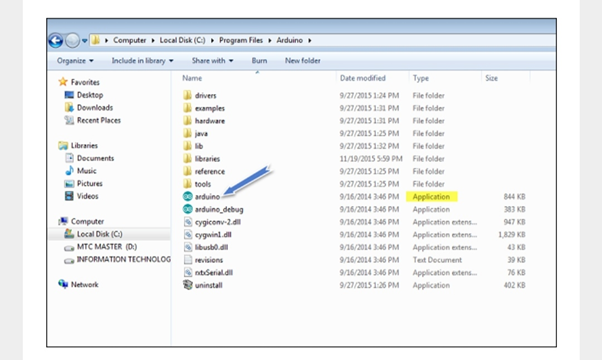
* Arduino IDE



How to install Arduino IDE?

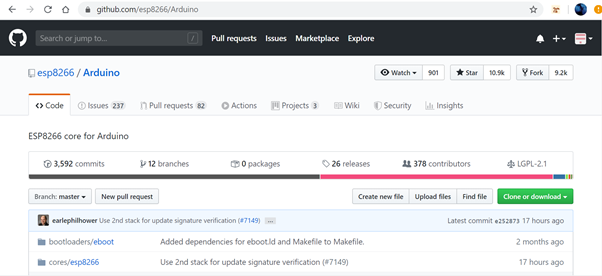
Step 1-Download the most suitable version from the Arduino official website. Unzip the file.



Step 2: Power up your Nodemcu.Inside the unzipped folder you can find the application icon with infinity label.Double-click to start the IDE.**ARDUINO 1.8.12 is installed.**

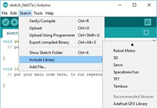
* Library for NodeMCU

Link:”<https://github.com/esp8266/Arduino>”

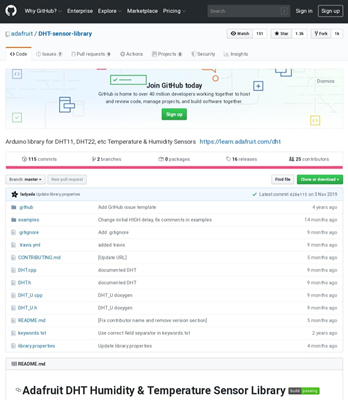


Library for NodeMCU is installed from the given link as shown above. This Library has to be used for implementing the features of the project.

To include the library:- Sketch->include library->Click on the required one or Add .ZIP library.



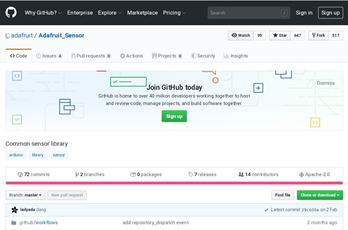
* Library for DHT 22 is installed

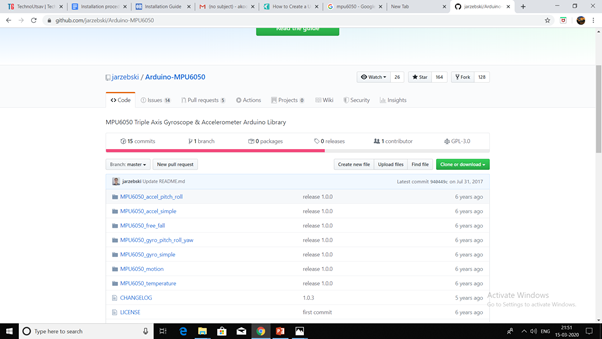
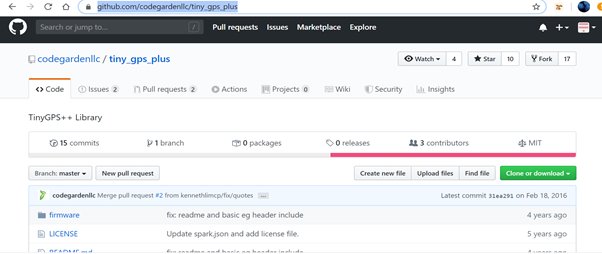


Link:<https://github.com/adafruit/DHT-sensor-library>

Adafruit Unified sensor library is also downloaded for using dht 22 Library.

Link:<https://github.com/adafruit/Adafruit_Sensor>



* Library for MPU6050 is installed
* Tinygps+ (for NEO 6M module) Library

Install the Tinygps+ library from the link<https://github.com/codegardenllc/tiny_gps_plus> to include the library used to implement the gps tracking feature which would help determine the location of the worker.

### Arduino Environment

In tools menu set the following:

* Board : NodeMCU 1.0 (ESP-12E Module)
* Upload Speed : 115200
* CPU Frequency : 160 MHz
* Flash Size : 4M (1M SPIFFS)
* Debug Port : Disabled
* Debug Level : None
* IwIP Variant : v2 Lower Memory
* vTables: Flash
* Exceptions: Disabled
* Erase Flash : Only Sketch
* SSL Support : Basic SSL ciphers (lower ROM use)
* Port : *Connect the ESP8266 to your laptop using a MicroUSB cable and then select your port, depending on OS*

Once the environment is set up, choose the code file, compile it and then flash it to the board. We will also need a local WiFi network to connect nodemcu.

Also add the following lines to the arduino code.

// Add WiFi connection information

char ssid[] = " "; // your network SSID (name)

char pass[] = " "; // your network password

* Working IBM cloud account

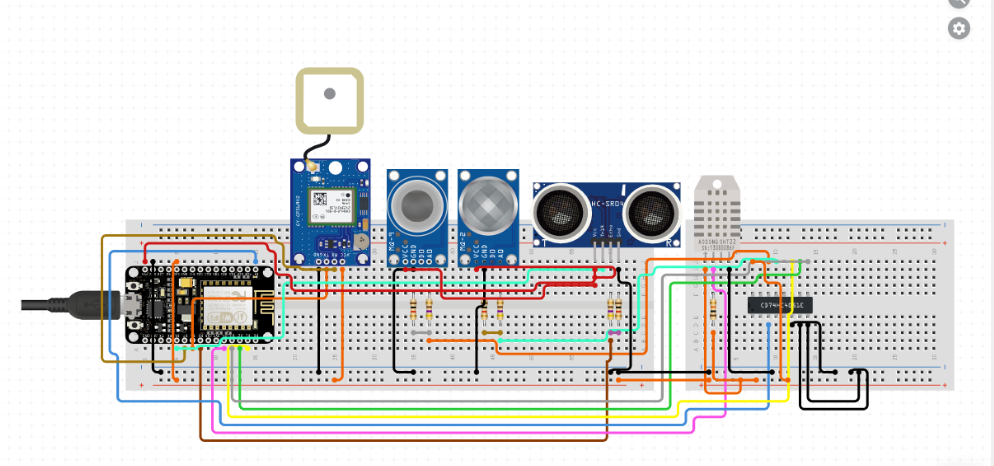
**Hardware**

The hardware required is as follows:-

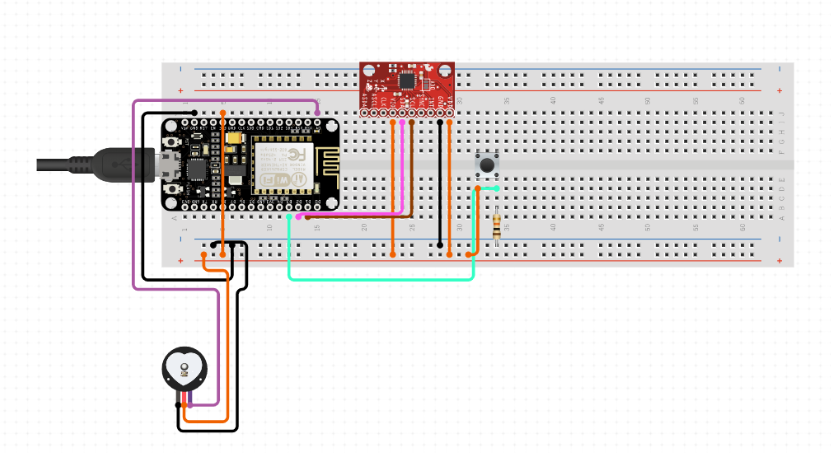
1. Nodemcu
2. DHT22
3. Ultrasonic sensor HC-SR04
4. Neo 6-M
5. Mq9 gas sensor-Monitors level of carbon monoxide.
6. Mq2 gas sensor-Monitors level of methane.
7. Mpu6050
8. Sen 111574
9. Push button
10. Jumper wires

**HARDWARE SETUP**

**HEAD GEAR**

****

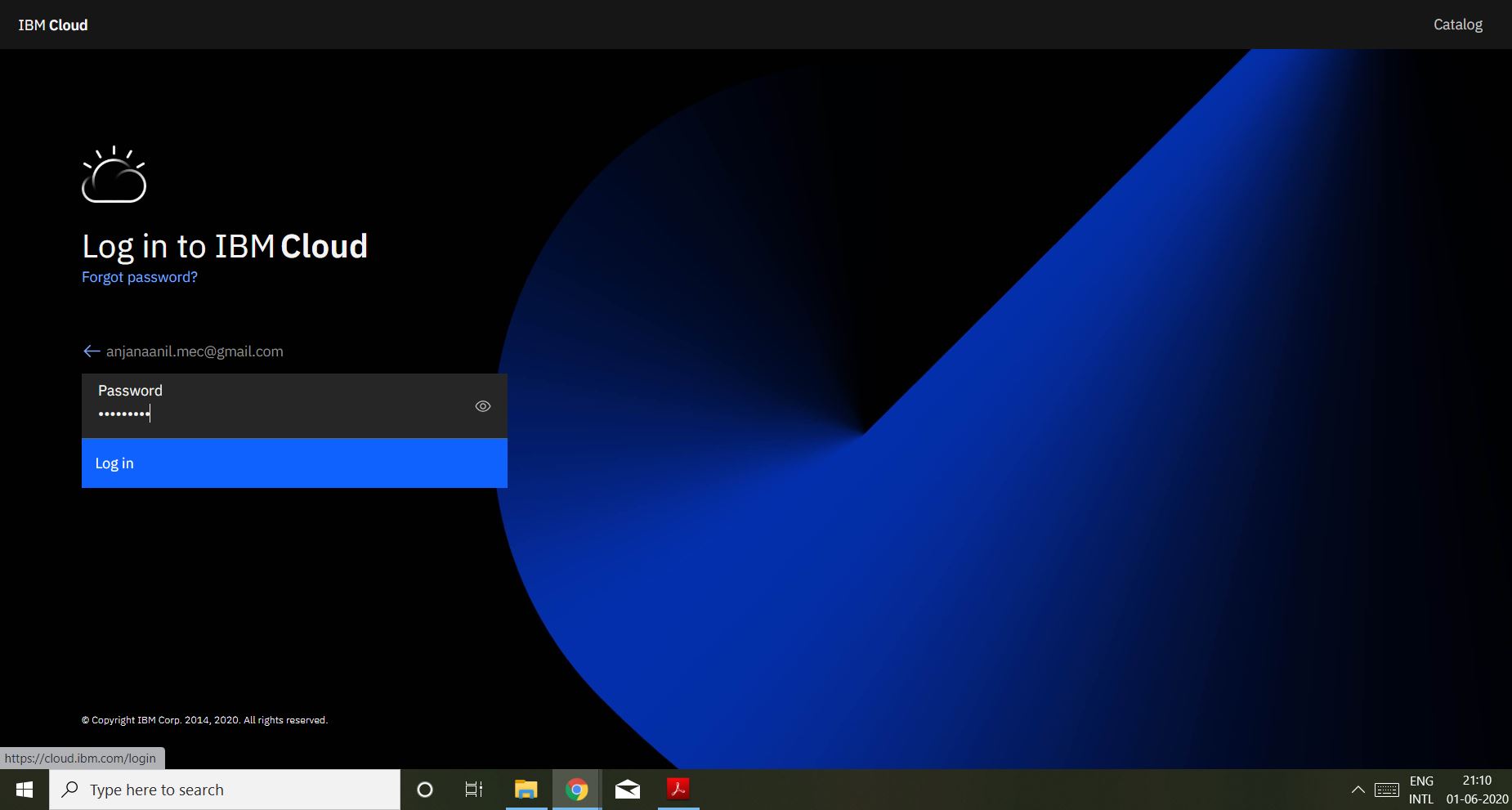
**WATCH**

****

**IBM ACCOUNT**

**Create IBM Account:**

Go to [IBM Platform](https://cloud.ibm.com/) to create an IBM account.



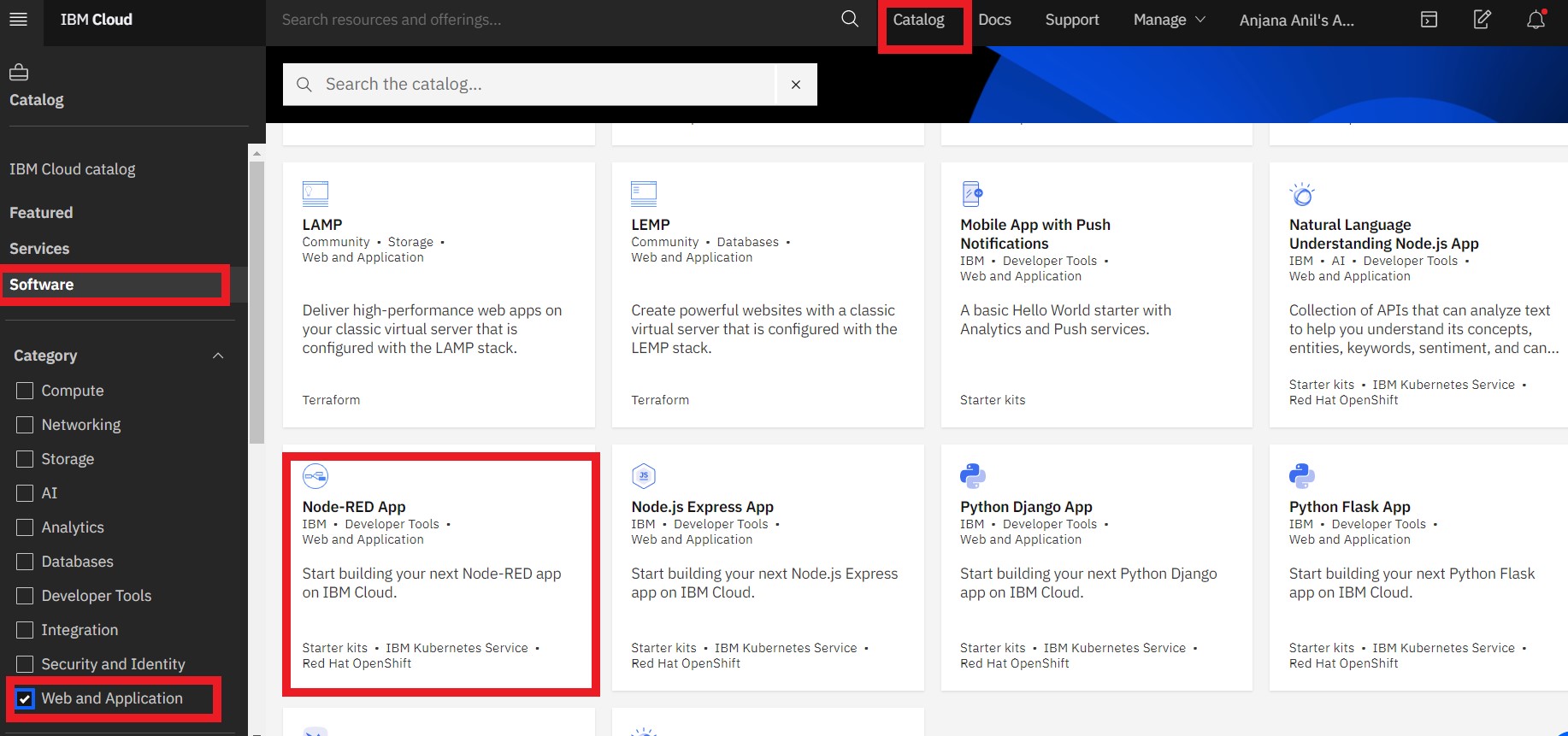
**To create Node-Red Application**

1.From Dashboard select catalog.

2.Select Software

3.Select Web and Application

4.Select Node-Red App



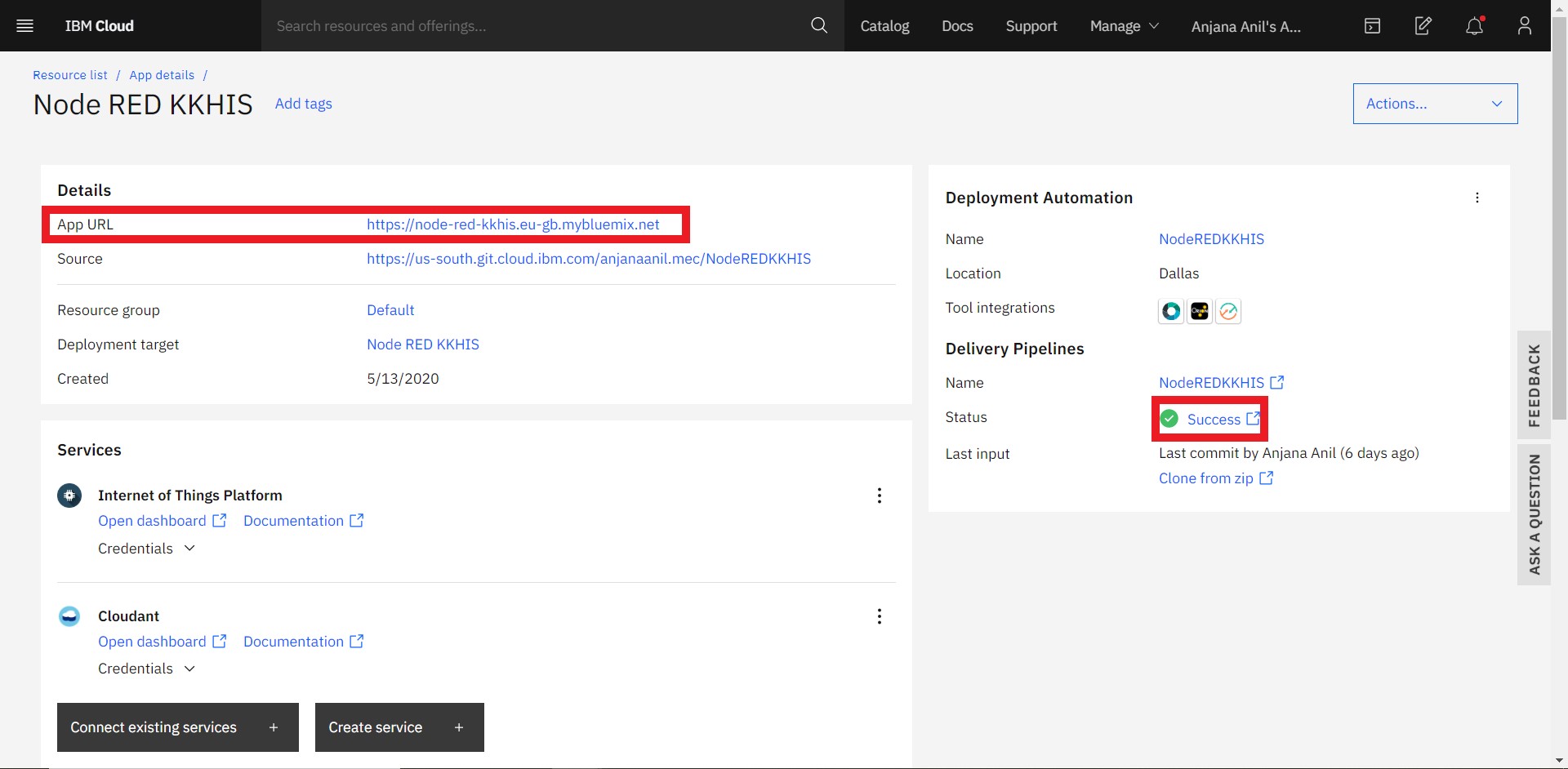
5.Keep all the values to default and continue with the App development.

* Wait until the Cloudant service has been deployed
* Press the Create Service + button, then select the Internet of Things section and press the Next button.
* Select the Internet of Things Platform then press the Next button
* Press Create to add the Internet of Things Platform to your application.
* Also generate the IBM cloud API key.

6.Once you select on click **Create app**, then proceed to click on **Deploy App.**

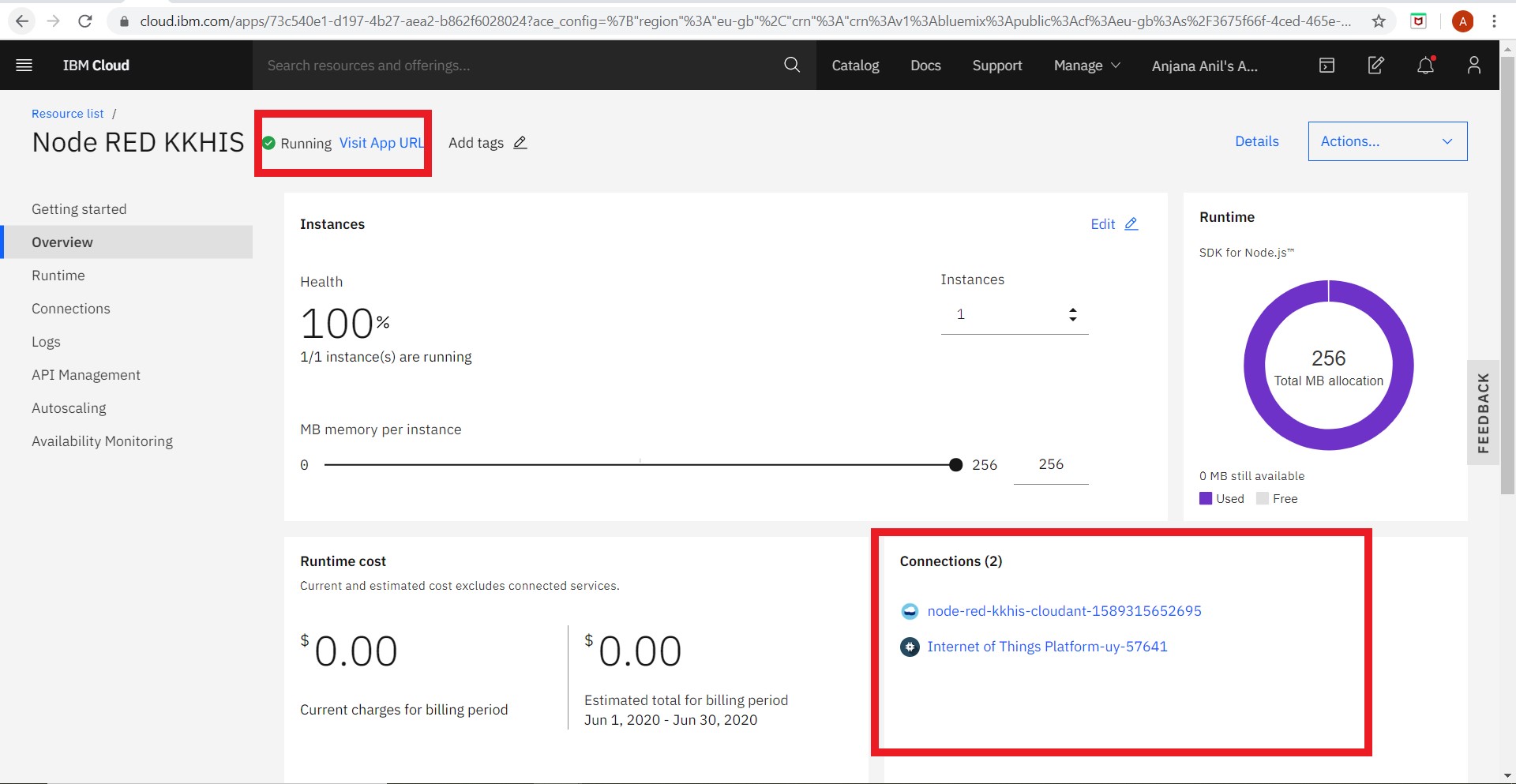
7.After deployment wait for the status to turn into **Success.**

8.App URL will appear after successful deployment . Click on the App URL

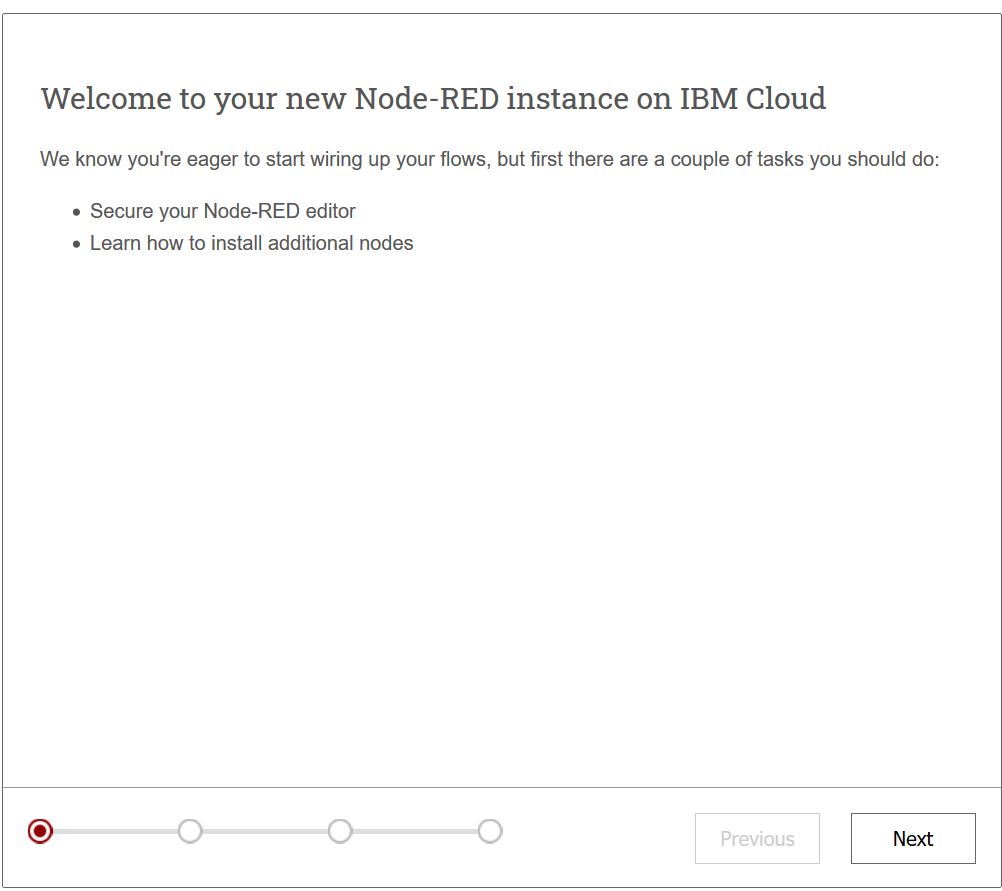


9.Add connections to the associated services clicking Connections --> Add Connections.

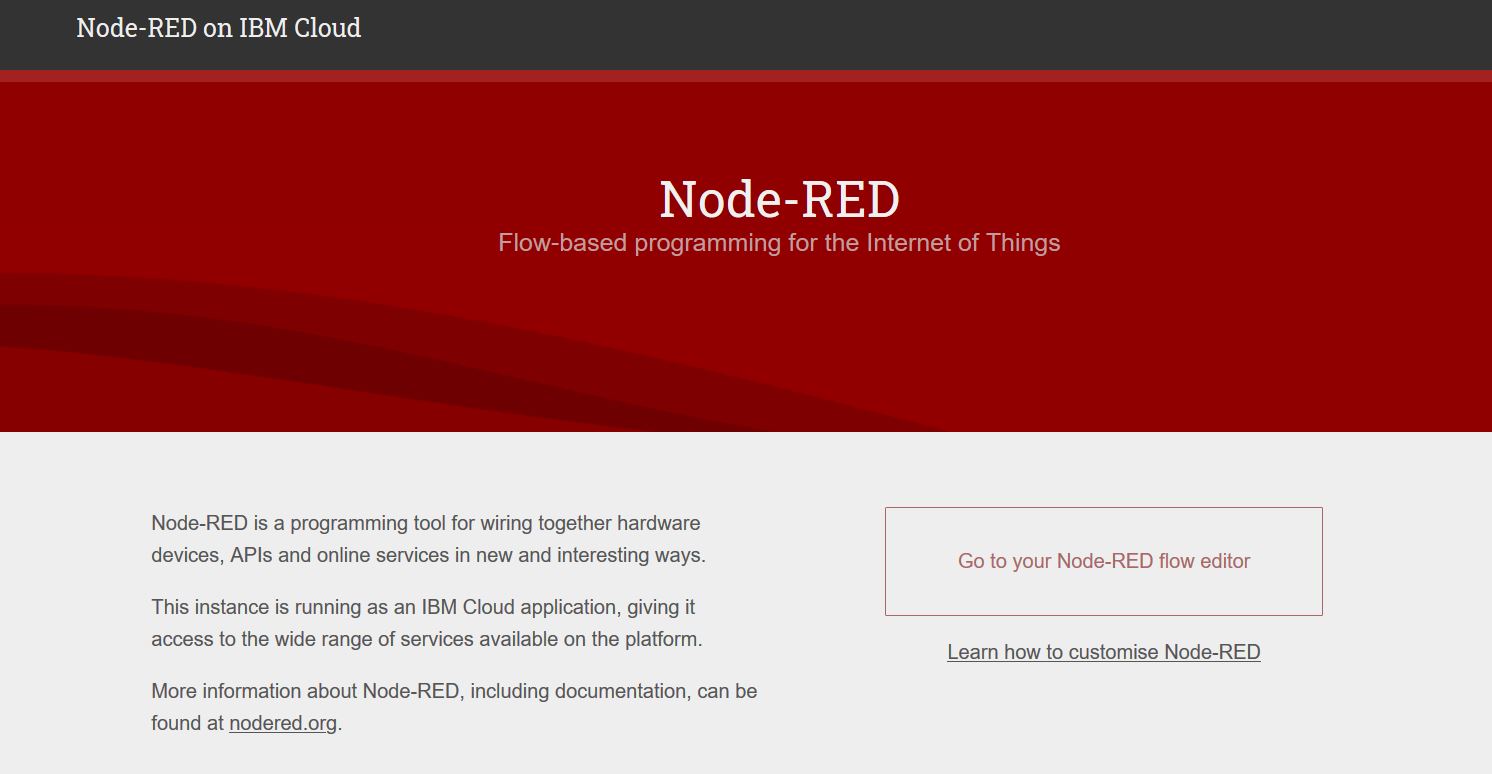
10.Click on Visit App URL in the Overview page.



11.Setup the Node RED editor.



12.Continue clicking on Next and click on Finish at the end to launch the Node RED editor.



13.Click on Node-RED flow editor.

14. The Node-Red Editor window opens:

Before importing the flow following has to be installed:-

15.Install-

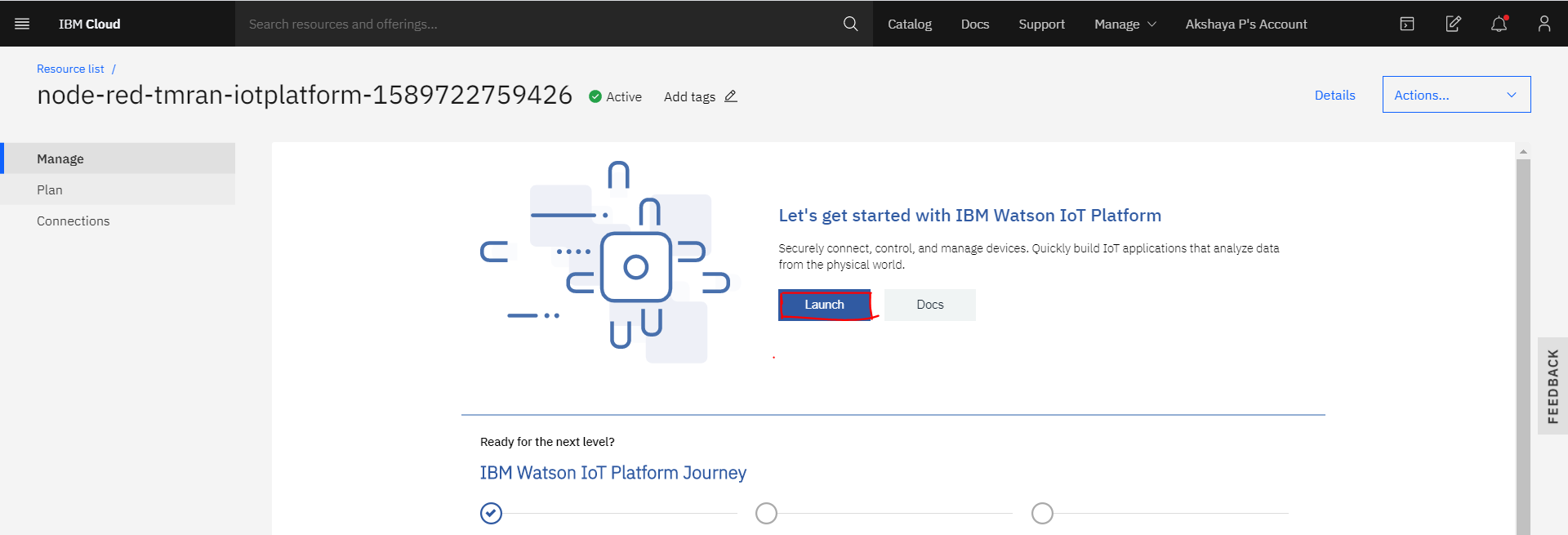
1. [node-red-contrib-ibm-watson-iot](https://npmjs.com/package/node-red-contrib-ibm-watson-iot)
2. [node-red-contrib-scx-ibmiotapp](https://npmjs.com/package/node-red-contrib-scx-ibmiotapp)
3. [node-red-dashboard](https://npmjs.com/package/node-red-dashboard)
4. [node-red-node-twilio](https://npmjs.com/package/node-red-node-twilio)
5. [node-red-contrib-ibmpush](https://npmjs.com/package/node-red-contrib-ibmpush)

Incase it fails, add the following to package.json file

1. "[node-red-contrib-ibm-watson-iot](https://npmjs.com/package/node-red-contrib-ibm-watson-iot)":"[0.x](https://npmjs.com/package/node-red-contrib-ibm-watson-iot)",
2. "[node-red-contrib-scx-ibmiotapp](https://npmjs.com/package/node-red-contrib-scx-ibmiotapp)":"[0.x](https://npmjs.com/package/node-red-contrib-scx-ibmiotapp)",
3. "[node-red-dashboard](https://npmjs.com/package/node-red-dashboard)":"[2.x](https://npmjs.com/package/node-red-dashboard)",
4. "[node-red-node-twilio](https://npmjs.com/package/node-red-node-twilio)":"[0.x](https://npmjs.com/package/node-red-node-twilio)",
5. "[node-red-contrib-ibmpush](https://npmjs.com/package/node-red-contrib-ibmpush)":"[0.x](https://npmjs.com/package/node-red-contrib-ibmpush)",

**Watson IoT Platform**

Go to top menu *≡* -> Resource list->IoT Platform.



Launch the platform.

Press the + Add Device and create a Device id-”your\_choice” and Device type-”esp8266”. Skip over and press next until the token is prompted. Give one of your choice and you will finally reach the device drilldown page.

**Note down the Org-ID, Device Type, Device ID and Authentication Token.It cannot be retrieved once you leave the page.**

**MQTT**

We use MQTT to send data to IoT platform.

Add the following code in the Arduino IDE:

#define MQTT\_HOST "ORG\_ID.messaging.internetofthings.ibmcloud.com"

#define MQTT\_PORT 1883

#define MQTT\_DEVICEID "d:ORG\_ID:DEVICE\_TYPE:DEVICE\_ID"

#define MQTT\_USER "use-token-auth"

#define MQTT\_TOKEN " "//TOKEN THAT YOU ENTERED WHILE ADDING DEVICE IN IoT PLATFORM

#define MQTT\_TOPIC "iot-2/evt/status/fmt/json"

#define MQTT\_TOPIC\_DISPLAY "iot-2/cmd/display/fmt/json"

In place of ORG\_ID,DEVIE\_ID,DEVICE\_TYPE and TOKEN, add the data you stored from the device drilldown page.

Open Connection Security Policy. Press the pencil icon next to Connection Security to edit the settings. Change the Default Security Level to TLS Optional, accept the Warning message by pressing the Ok button, then Save the change.This will use unsecured connection.

The rest of the mqtt code can be copied from the arduino code given.

On flashing the code, the data will be sent to the IoT platform.

Link to the code-” “

**Generating your own API Key**

Go to IoT platform.Go to the App section and click generate API Key.

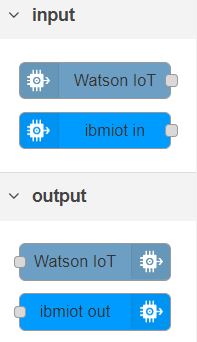
The API Key is used as the username when connecting and the API Token is the password.



**Setting up the nodes**

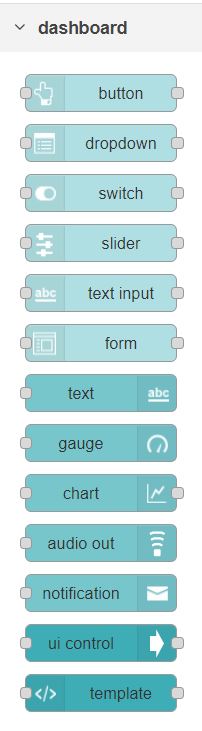
Following nodes are installed in Node-RED

* **Node-red-contrib-scx-ibmiotapp and Node-red-contrib-ibm-watson iot**

****

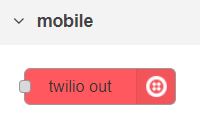
* **Node-red-dashboard**

The following nodes are installed in the Node-RED to provide widgets that show up in your application user interface (UI).



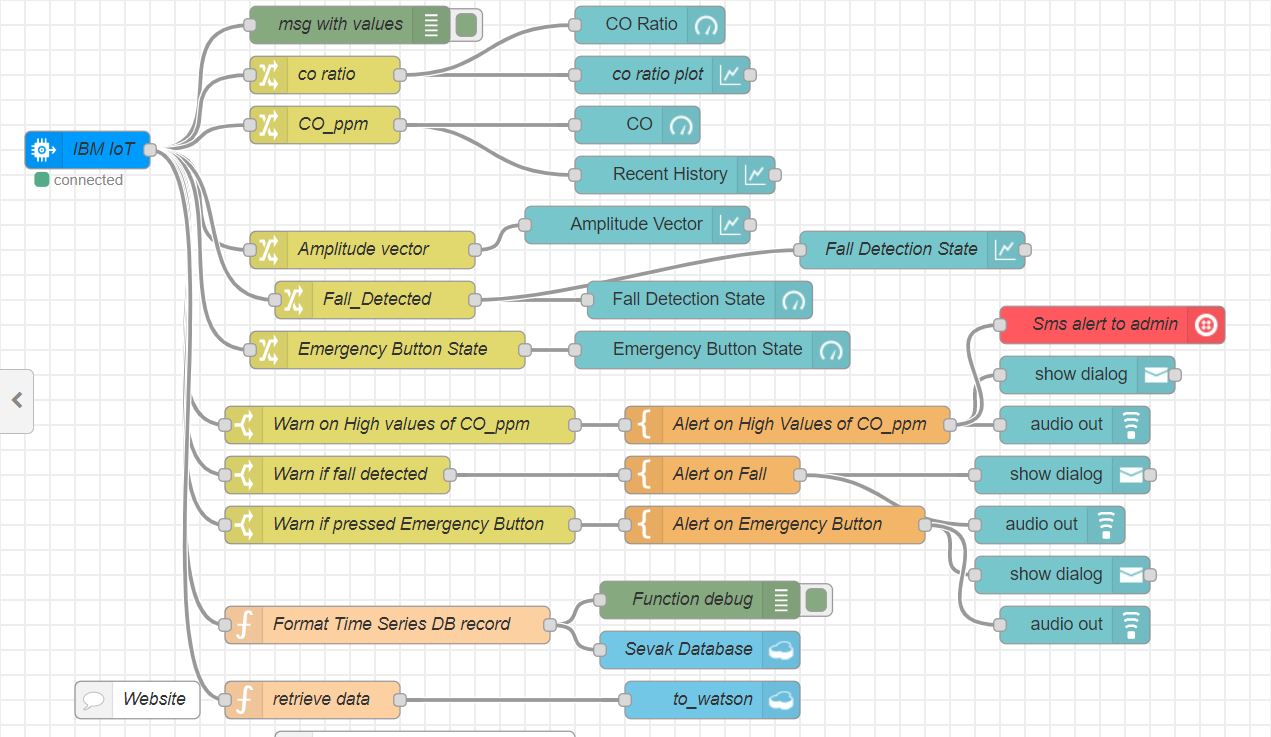
* **Node-red-node-twilio-**twilio

TwilioAPIs can add capabilities like voice, video, and messaging to applications.



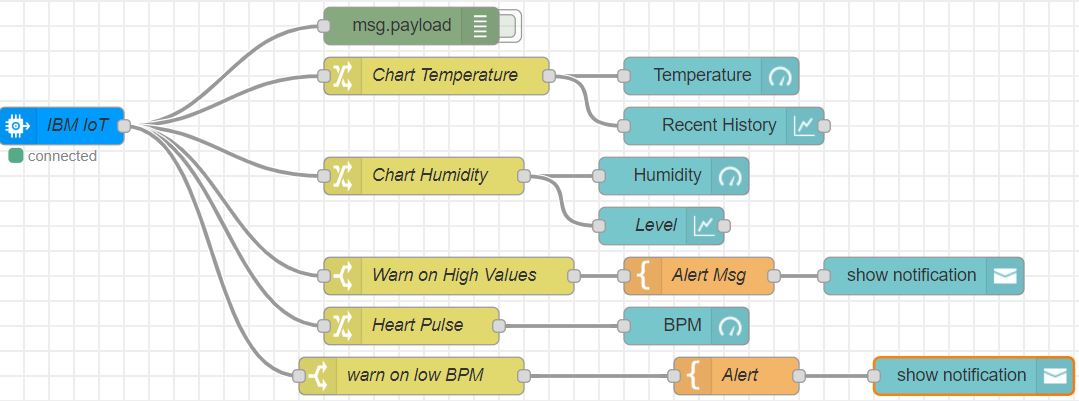
**FLOW OF NODE-RED**

* **CO concentration and fall detection flow**

****

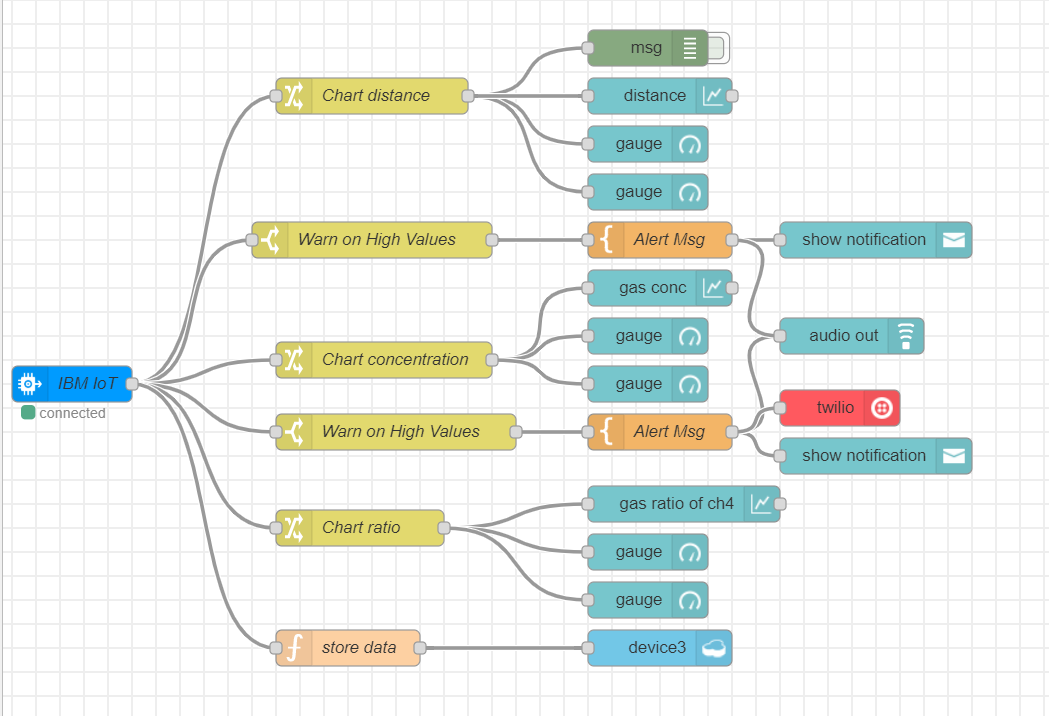
Link to flow code :<https://github.com/Sevak-Mec-Electros/Sevak/blob/master/Flows/CO_fall.json>

* **DHT 22 and pulse sensor flow**

****

Link to flow code : <https://github.com/Sevak-Mec-Electros/Sevak/blob/master/Flows/dht22_pulse.json>

* **Obstacle detection and CH4 gas sensor flow**

****

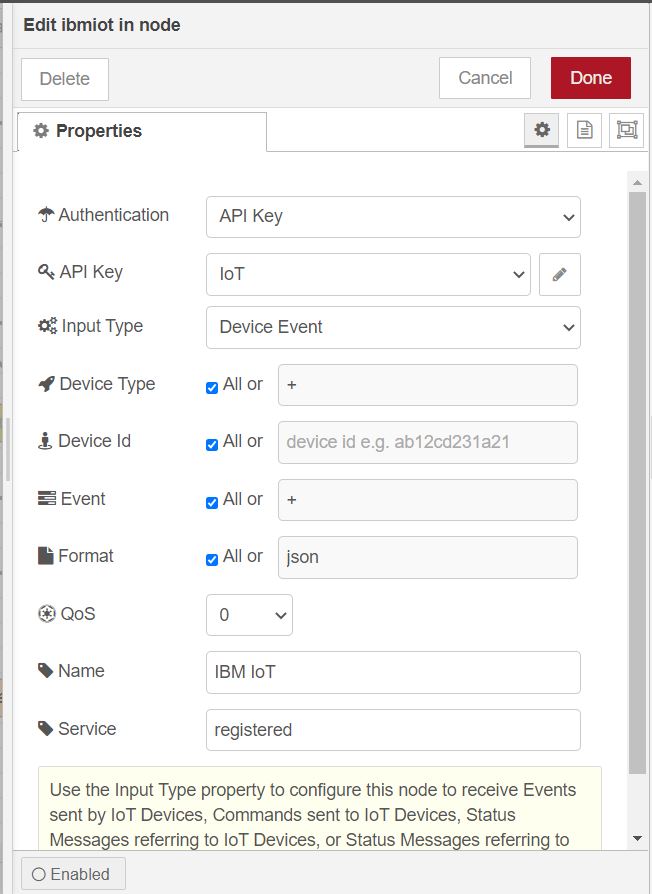
Link to flow code :

<https://github.com/Sevak-Mec-Electros/Sevak/blob/master/Flows/obj_detection%20and%20gas_sensor.json>

**IBM IoT input node**

****

1. Drag an input > ibmiot in node from the palette on the left into the flow (the large blank area).
2. Double-click the new item.



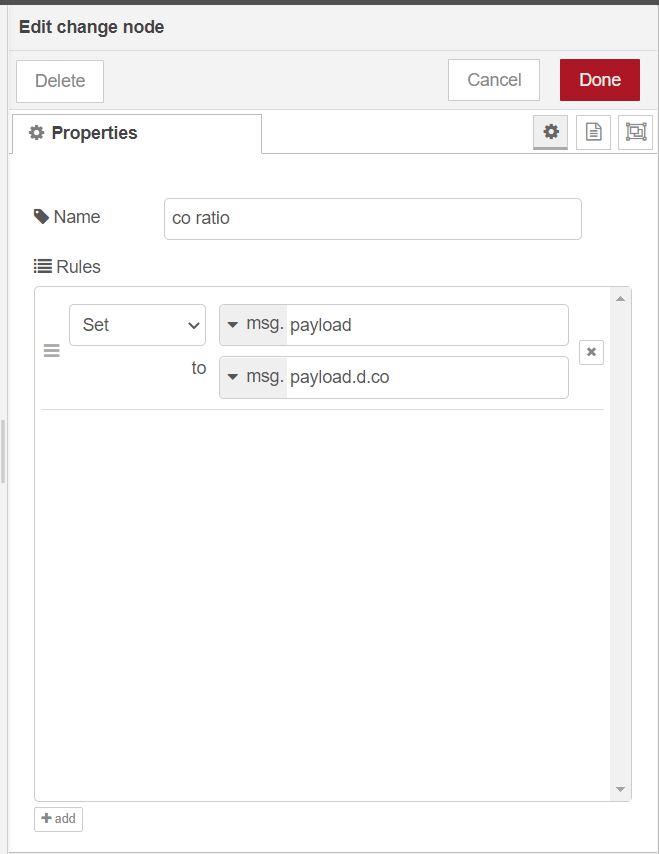
1. Specify these properties:
   * Authentication: API Key
   * API Key: Add new ibmiot
   * Input Type: Device Event
   * Device Type: All
   * Device Id: All
   * Event: All
   * Format: All
2. Click the pencil icon.
3. Create an IoT with these parameters:
   * Name: “your\_choice”
   * API Key and API Token: The values you created for the API Key previously
   * Server-Name: The host name you used (six characters for your specific ID followed by .messaging.internetofthings.ibmcloud.com)
4. Click Add and then Done

**Change Node**

****

Change node allow you to change a message payload or add new properties. You can use this node to affect the properties in a message, either by changing existing ones, deleting them or adding new properties.

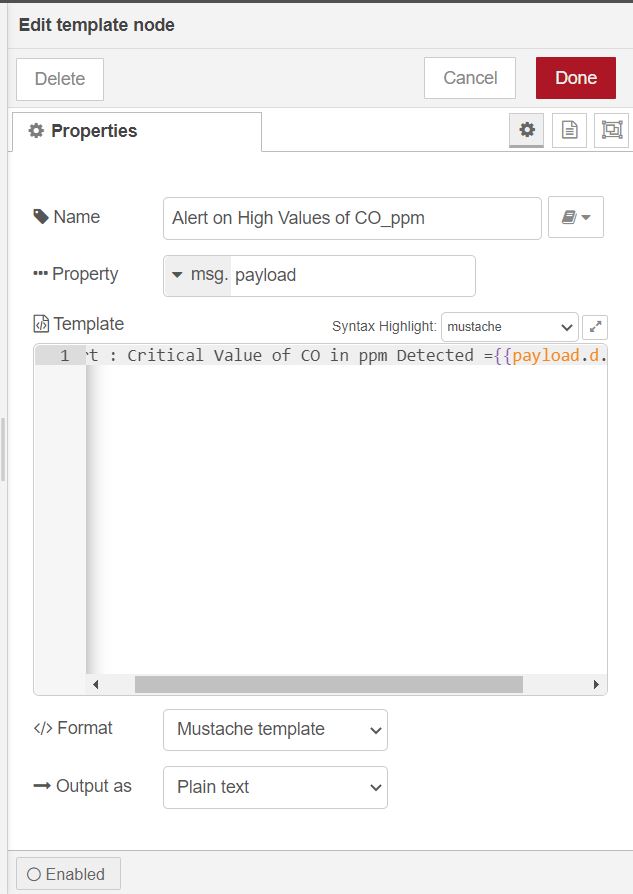
1. Drag and drop and change node
2. Double click and add properties based on the input to the node



**Template Node**

****

1. Drag and drop template node
2. Double click and add the template to display if a value goes above or below a threshold.

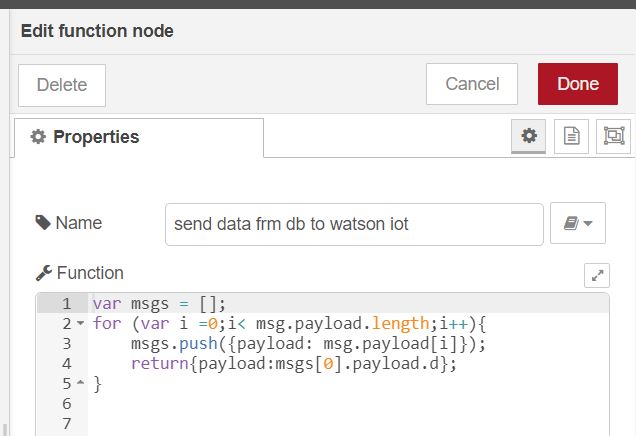


**Function Node**

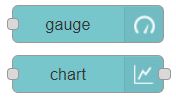
****

Function node are used to perform a specific function.

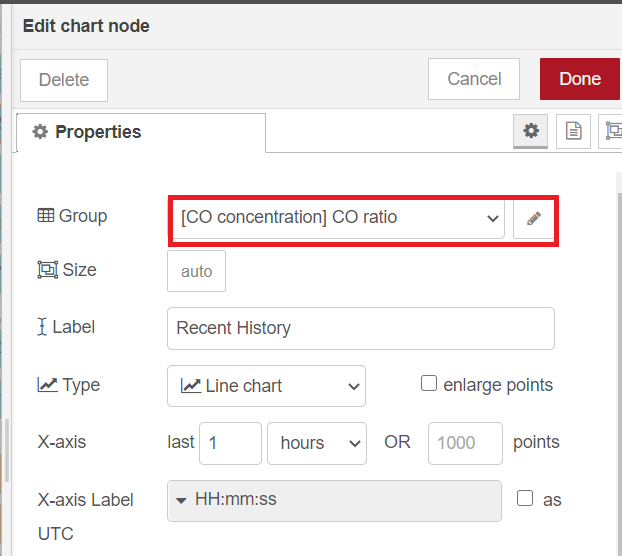
1. Drag and drop Function node
2. Double click on the node
3. Enter the code for the function to perform

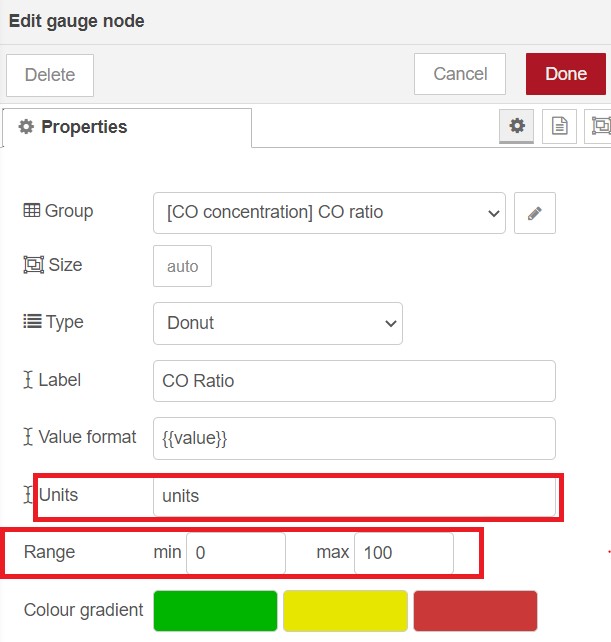


**Chart and gauge node**

****

These node are used to display the obtained values in dashboard

1. Drag and drop the chart or gauge node.
2. Double click on the node
3. Put the group name is set as the name of the tab the graph or gauge belongs
4. Gauge node can be used to set max and min value for the input and can also be used to set the units of the input.

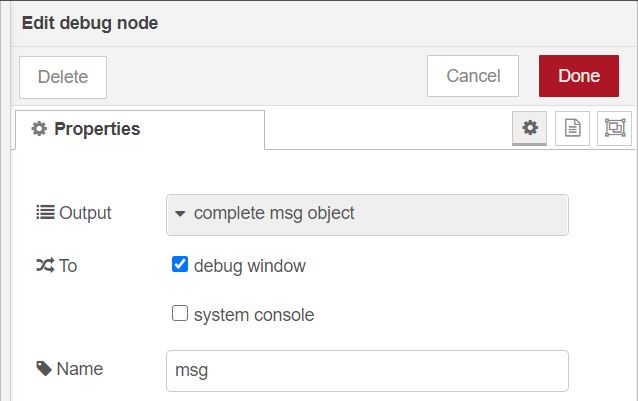


**Debug Node**

****

Debug nodes are used to check errors and values that are sent to different nodes.

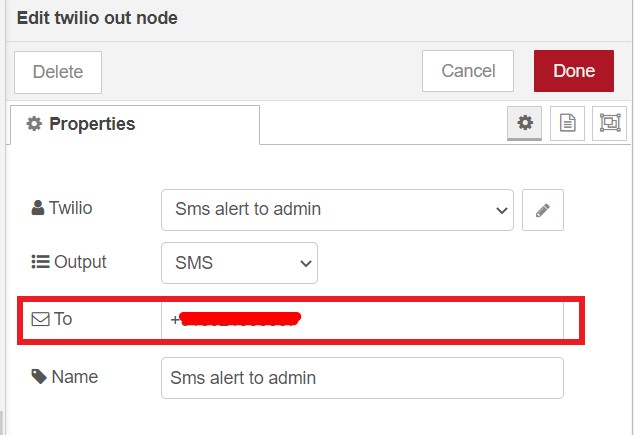
1. Drag and drop debug node
2. Double click on the node
3. Error message if any will be shown in the debug messages



**Twilio out node**

****

Add Account SID ,the number from which you will receive the call and Token. Also specify output as either sms or call and the number to which you will receive the alert.



**Database**

A database is created using the cloudant service with all the possible values that will or may occur for each feature that is included in the device.By clicking the load data button in the dashboard the user can view the stored data in the database and by clicking each tab in the dashboard the user can view the individual values for each feature that is included in the device.

**How to use Sevak?**

The user and admin can visit Sevak’s website:

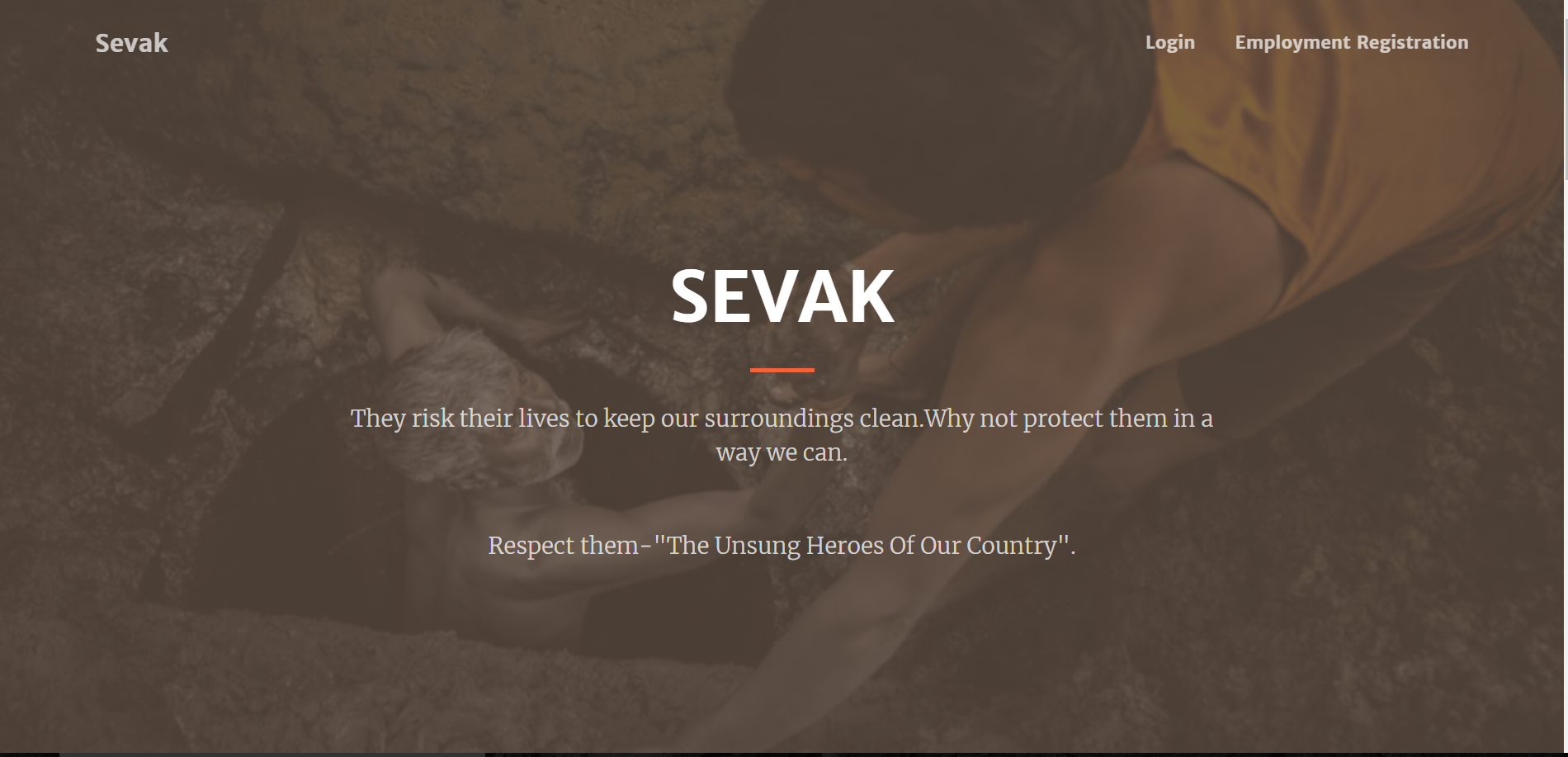
Link : <https://sevak-mec-electros.github.io/Sevak/>

Demonstration video link : <https://drive.google.com/file/d/12jSCLLRkIYVElgH2v4nyqEZTRtDAny5c/view?usp=drivesdk>

The user can either view the dashboard by logging in or can visit the Employment registration to register for the same.

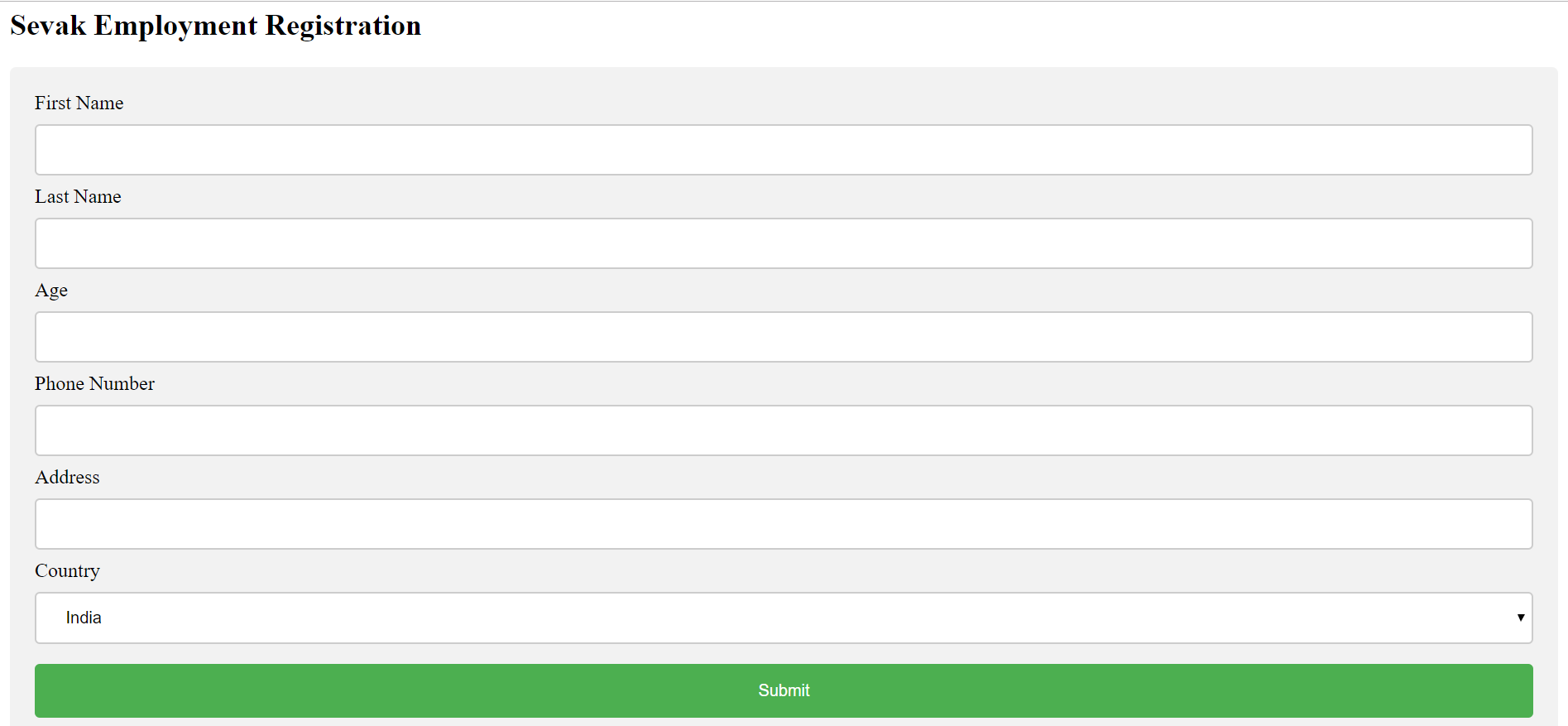
For the purpose of attaining our goal to help as many people as we could, the Sevak Service is implemented as a user friendly website on a custom domain configured using a leading software development platform such as Github.

This ensures widening the reach of our service to millions of people through a widely supported and accessible hosting platform in a variety of browsers including Google Chrome, Mozilla Firefox and Microsoft edge.



We have also taken great care regarding the Friendliness and Accessibility of our website by testing it against the standards stated by the Wave Accessibility Evaluation tool [ [https://wave.webaim.org](https://wave.webaim.org/) ] and receiving acceptable results.

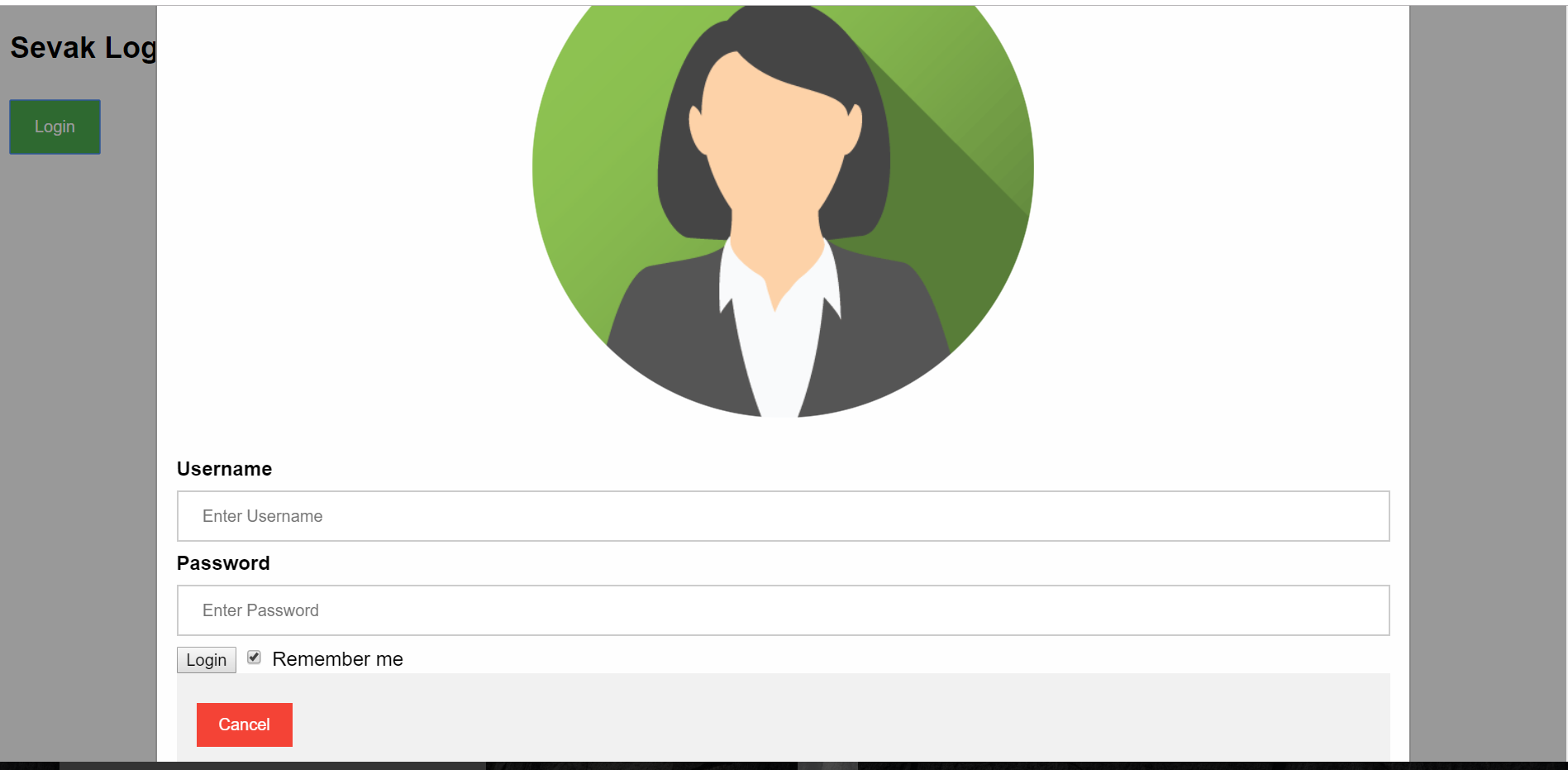
* Use the link provided to reach the welcome page of our website .Scroll down to find a brief description regarding our service.
* At the top right corner click on the **Employment Registration** option to be presented with an employee registration form .The details filled in this page can be accessed from the cloudant nosql database service provided by the ibm cloud service , credentials of which will be provided at the end of this document. Employee management is a future scope of our project.

****

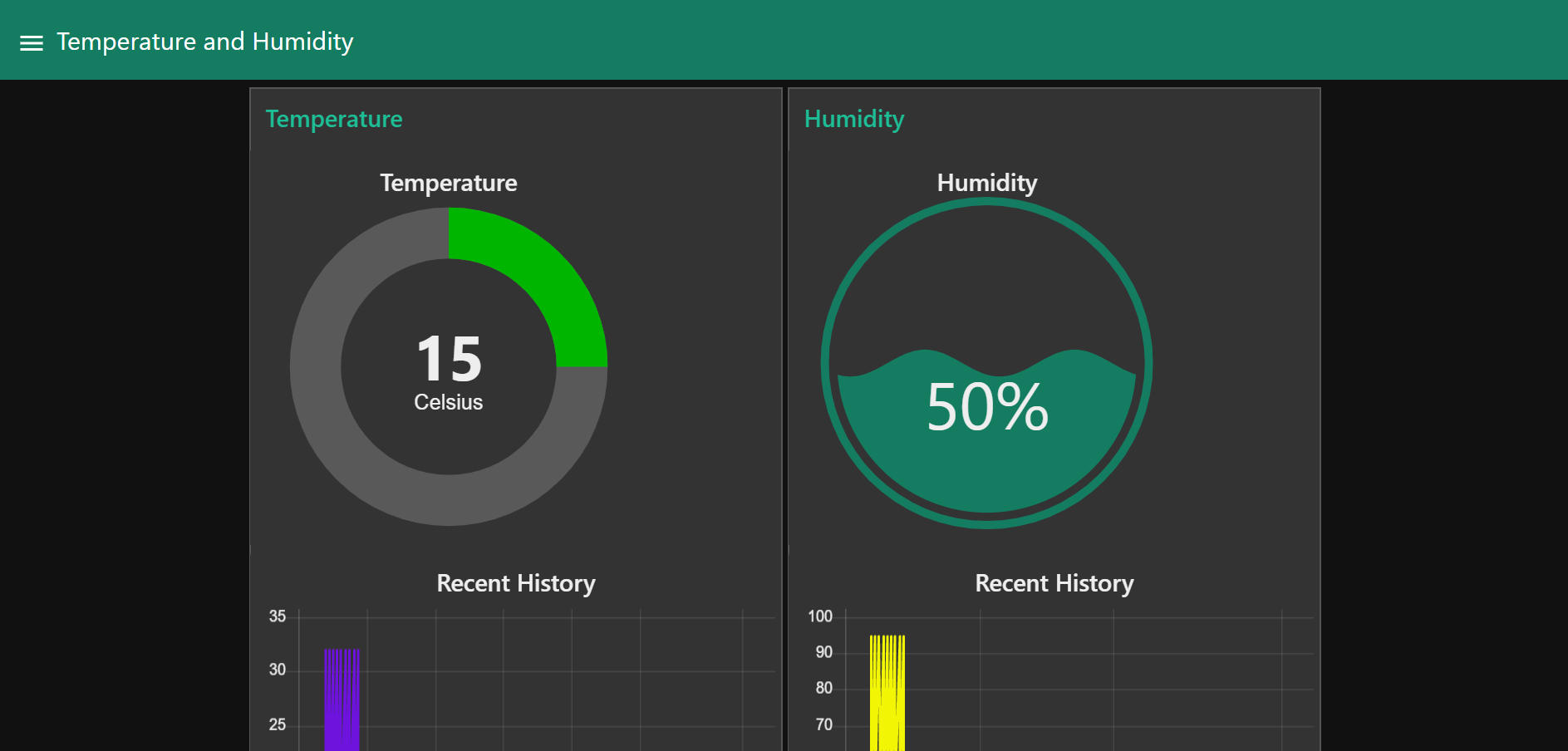
* Return to the home page to click on the **Login** option . You will be taken to a new page with a login button in the top left corner.
* Click on this button to get a login form .Give the credentials:Sample username and password given below

**Username :a**

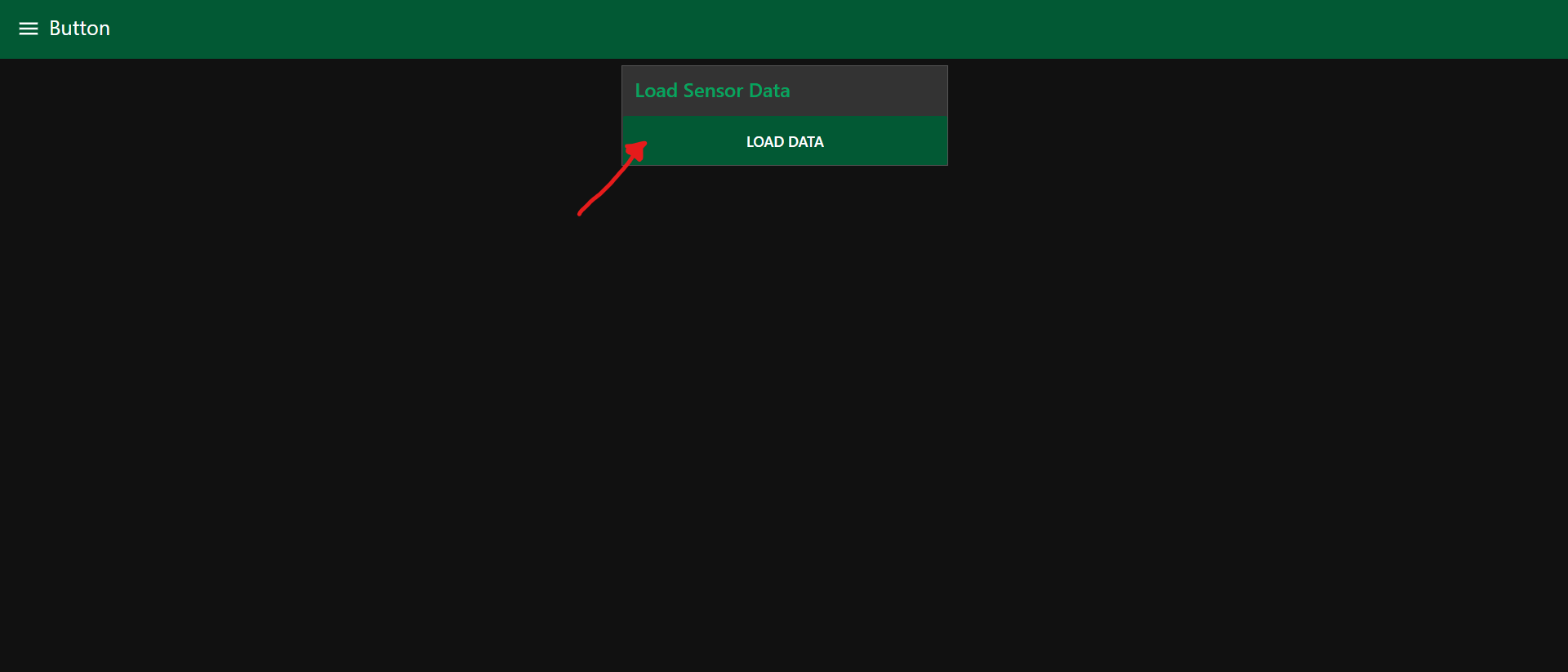
**Password :a**



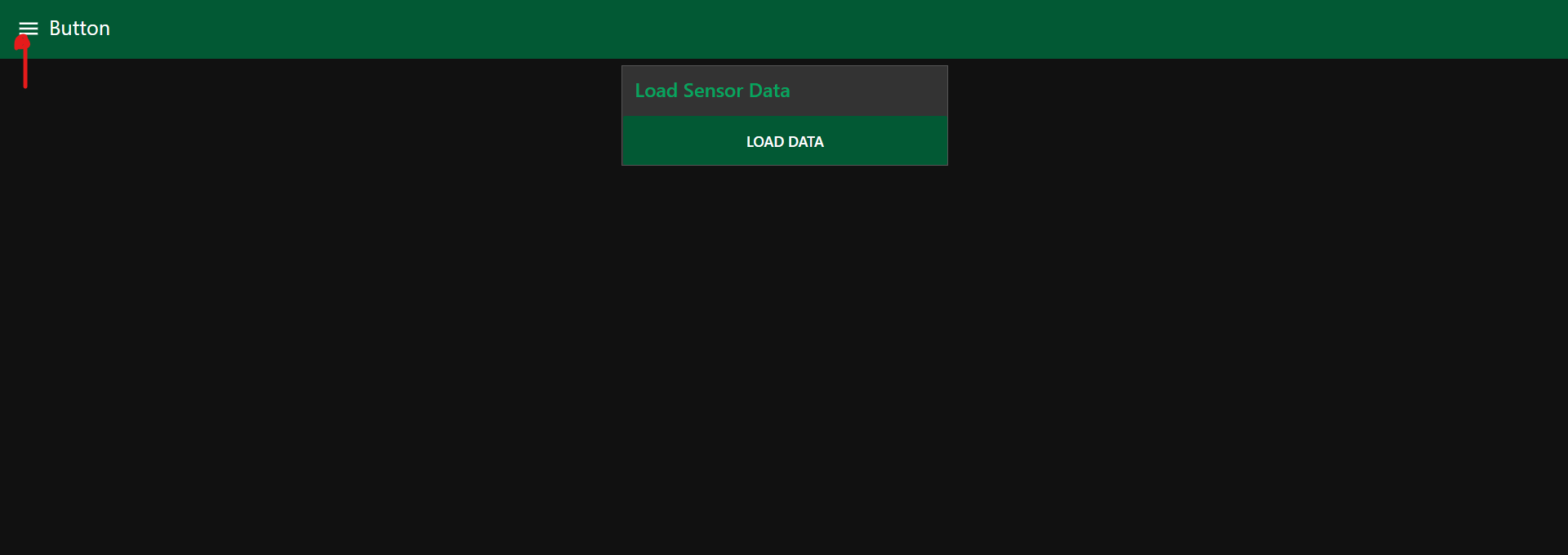
* Be careful with this form as u get only **three attempts** to login .On successful login, you will be taken to the node red dashboard page of Sevak.
* This page is complete with the visualisations and alert systems to monitor the health conditions and surroundings of the employee.



* On logging in you are taken to the Button tab. On clicking on the load sensor data button, you can have a real time view of how Sevak works.



* The sensor data is loaded from a database to dashboard to have real time view.
* On clicking the sandwich button, you can navigate to different tabs containing information about different parameters measured.



* The dashboard tabs can be selected and viewed from the list of tabs visible on clicking the option available in the top left corner.the list of tabs include:
  + CO concentration [carbon monoxide present in the surroundings]
  + Temperature and Humidity [as experienced by the employee worker]
  + Methane concentration [CH4 present in the surroundings ].
  + Pulse rate [pulse rate of the employee worker].
  + Fall detection[based on observation of the pressure]
  + Distance [Distance of the nearby obstacles from the employee worker]
* All the above mentioned monitoring systems are equipped with an automatic alert system which
  + Sends the coordinates of the location on emergency situations
  + Sends voice alert in a **preferred language** as opted by the user.
  + Sends **sms or calls** the number provided with required information.

The user can either view the dashboard by logging in or can visit the Employment registration to register for the same.

In order to address the various accessibility issues that would be encountered due to any unfortunate compatibility errors [the occurrence of which is extremely rare],we have also provided an additional link to access to the monitor dashboard .

Additional Link: <https://node-red-deloitte.eu-gb.mybluemix.net/ui/>

**CLOUD LOGIN CREDENTIALS**

Main cloud account:

Username: [anjanajayan.mec@gmail.com](mailto:anjanajayan.mec@gmail.com)

Password: Sevak@167

Node-red

Username: Aj

Password: surya0484

Others:

Username: anjanashiny.mec@gmail.com

Password: Sevak@123

Username: akshayapraba.mec@gmail.com

Password: DanBrown123\*

**HOW TO ACCESS UPLOADED FILES ON CLOUD?**

* Login to our main cloud account.

Username: [anjanajayan.mec@gmail.com](mailto:anjanajayan.mec@gmail.com)

Password: Sevak@167

* Go to resource list.
* Under storage click on cloud object storage.(cloud-object-storage-kg-watson)
* Click on the bucket. Here you can find all the uploaded files(sevak-documents).

**CONCLUSION**

As conclusion let us first consider the following points,

* Health and safety of sanitation workers are inadequately addressed in the public health research.
* Sanitation work lacks specific protective regulatory guidelines to address health hazards unlike other hazardous occupations.
* Most workers work without appropriate treatment as they ignored their illness, and did not want to miss their wages or lose their job. Self-medication was common.
* Intake of alcohol was prevalent to cope with the inhuman task of cleaning filthy sewage, and as a modality to forget their health problems.
* The pattern of illnesses reported during monthly monitoring was also reported as long-standing illnesses.
* Health and safety mechanisms at the workplace did not exist and were not mandated by regulatory bodies.

**Sevak** is a device that was born from our efforts and hardwork ,that could be the next step in helping these sanitation workers, a section of the society trapped in a cycle and forced to work in poor conditions due to societal pressure and poverty.

This device will monitor the unhealthy living conditions that the workers face every day with fear and great risk. Our device will reduce the risk by alerting the authorities whenever a variable falls below threshold.

We sincerely hope this step contributes to bringing down their fear and helps them work in a safe and monitored environment.