

NALAIYA THIRAN

**IoT Based Safety Gadget For Child Safety
Monitoring & Notification**

PRESENTED BY,

1. **DICKSON**
2. **GOKULAKANNAN**
3. **DHARSHANPONSAMI**
4. **HARIHARAN**

AREA OF THE PROJECT

INTERNET OF THINGS(IOT)

OBJECTIVES

- ❖ Enables tracking of the child's location and capturing of data remotely such as where the child located, distance, etc.
- ❖ To show the child's actual data with reference values.
- ❖ Enables sending of notification if the child is out of location or when the device realizes abnormal conditions/situations.
- ❖ Develop a prototype of IoT wearable smart band connected to parent's mobile apps so that they can monitor the actual condition of children at anytime and anyplace.

LITERATURE SURVEY

Author	Title	Proposed Methods	Journal, Year
David Hanes, Gonzalo, Patrick Grosetete, Robert, Barton, Jerome.	“IoT Fundamental and Networking Technologies, Protocols”	During an emergency, mobile apps alert the control room of nearby police stations or caretakers of children. The literature shows that location tracking devices are available in the market but it does not provide a complete solution to the problem. The solution to this problem is to design an IoT device, which senses the child’s location and environment and during an emergency, it should send the alert to the parents automatically.	Cisco,2017
Aditi Gupta, Vibhor Harit.	Child Safety & Tracking Management System by using GPS.	This paper proposed a model for child safety through smartphones that provide the option to track the location of their children as well as in case of emergency children are able to send a quick message and its current location via Short Message Services. Merits: The advantages of smart phones they offer rich features like Google maps, GPS, SMS etc. Demerits: This system is unable to sense the human behaviour of children.	IEEE,2016
K. N. H. Srinivas, T. D. S. Sarveswara Rao, E. Kusuma Kumari.	Smart IoT Device for Child Safety and Tracking.	The system is developed using Link-It ONE board programmed in embedded C and interfaced with temperature, heartbeat, touch sensors and also GPS, GSM & digital camera modules. The novelty of the work is that the system automatically alerts the parent/caretaker by sending SMS when immediate attention is required for the child during an emergency.	IEEE,2019

METHODOLOGY

- ✓ It focuses on the key aspect that a missing child can be assisted by the people around the child and can play a remarkable role in the child's safety until reunited with the parents.
- ✓ If any deviant readings are disclosed by the sensor, then an SMS and phone calls are set off to the parent's mobile. Also, it overhauls the parental app through the cloud.
- ✓ The technique is equipped with GSM and GPS modules for sending and receiving calls, and SMS between the safety gadget and the parental phones.

- ✓ The system also consists of a Wi-Fi/cellular data module used to implement IoT and send all the monitored parameters to the cloud for android app monitoring on the parental phones.
- ✓ The panic alert system is used during panic situations alerts are sent to the parental phone, seeking help also the alert parameters are updated to the cloud. Most of the wearables available today are focused on providing the location, and activity of the child to the parents.

METHODOLOGY

HARDWARE REQUIREMENTS:

IoT device

GSM

Mobile(Notification)

GPS

SOFTWARE REQUIREMENTS:

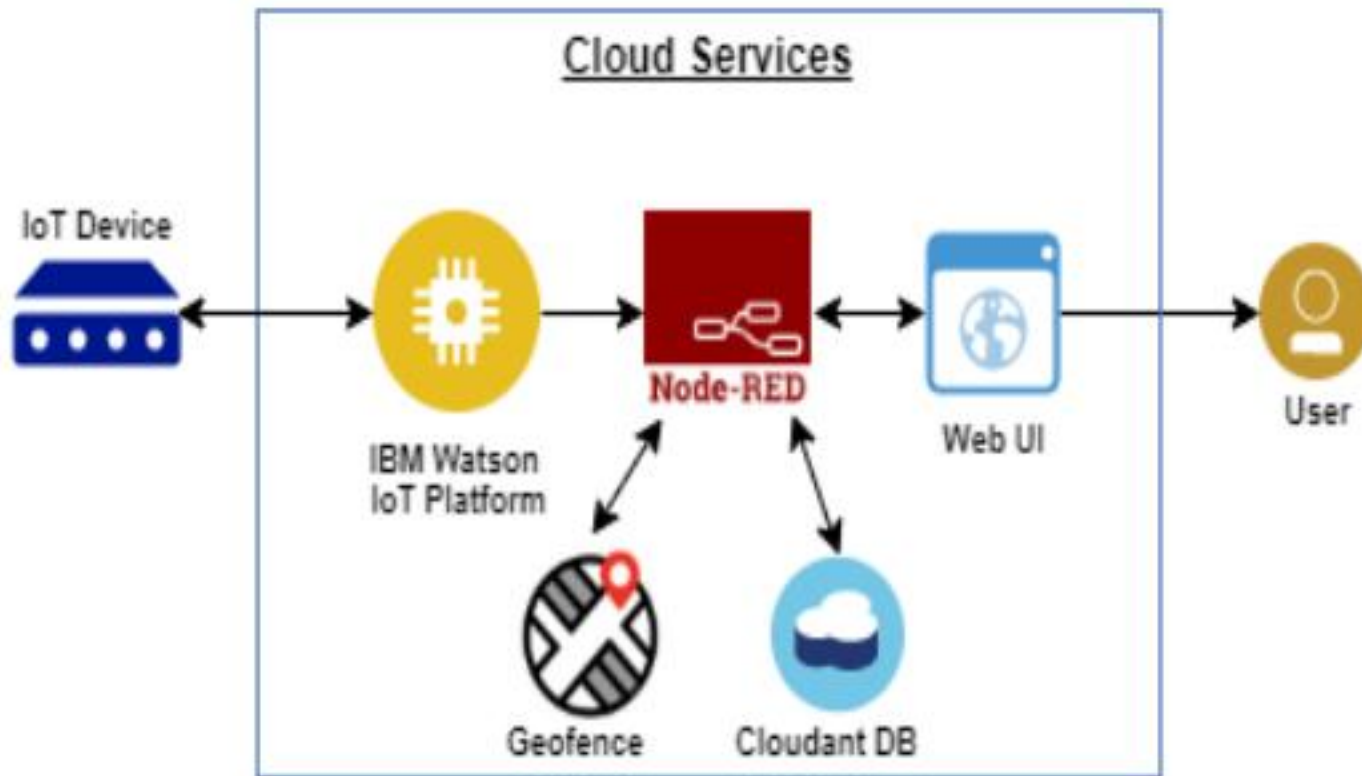
IBM Cloud

IBM IoT Platform

IBM Node red

IBM Cloudant DB

PROPOSED BLOCK DIAGRAM/ TECHNICAL ARCHITECTURE



SIMULATION AND RESULTS

IBM WATSON IOT PLATFORM

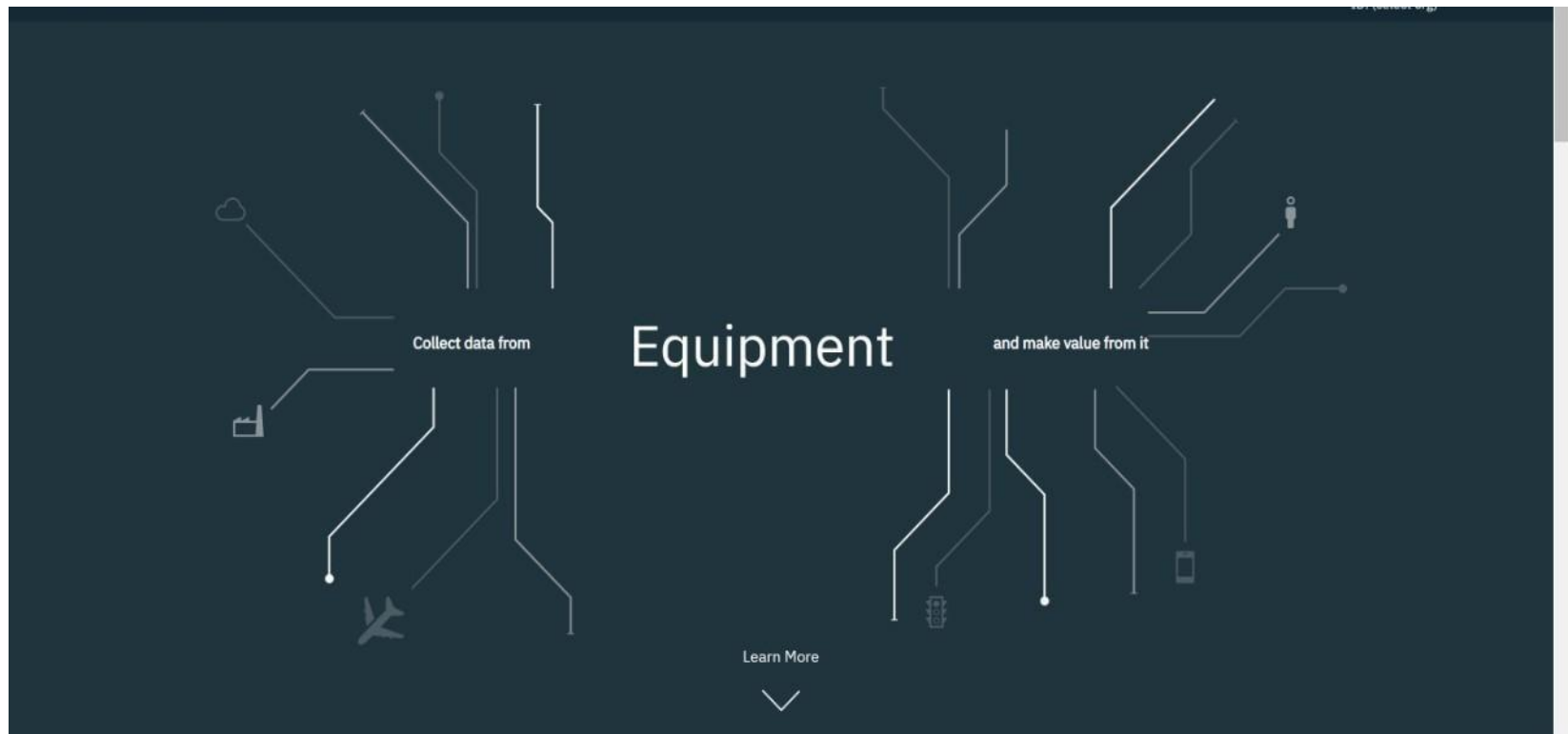


Fig.1,Creating IBM Cloud Service and creating the device

Creating Python Code:

```
import json
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity":{"orgId": "4o1qxb",
    "typeId": "TestDeviceType",
    "deviceId": "12345"},
    "auth": {"token":"pnhXvzN-
sWMKv&hxyi"}} }
client =
wiotp.sdk.device.DeviceClient
(config=myConfig,
logHandlers=None)
client.connect()
```

```
while True:
    name= "Smartbridge"
    #in area location
    latitude= 17.4225176
    longitude= 78.5458842
    #out area location
    #latitude= 17.4219272
    #longitude= 78.5488783
    myData={'name': name,'lat':latitude,
    'lon':longitude}
    client.publishEvent(eventId="status",
    msgFormat="json", data=myData,
    qos=0, onPublish=None)
    print("Data Published to IBM IoT
    platfrom: ", myData)
    time.sleep(5)
    client.disconnect()
```


Browse Action Device Types Interfaces

Add Device +

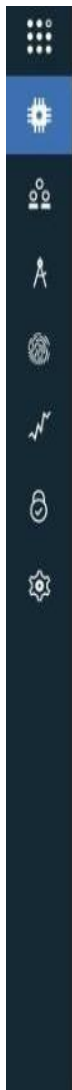
	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
	12345	Connected	TestDeviceType	Device	Oct 29, 2022 12:42 PM	...

Identity Device Information Recent Events State Logs X

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
status	{"name":"Smartbridge","lat":17.4225176,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
status	{"name":"Smartbridge","lat":17.4225176,"lon":7...	json	a few seconds ago

1 Simulation running



Browse Action Device Types Interfaces

Add Device +

Identity Device Information **Recent Events** State Logs X

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"name":"Smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
status	{"name":"Smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago
event_1	{"name":"smartbridge","lat":17.4219272,"lon":7...	json	a few seconds ago

Items per page 50 | 1-1 of 1 item 1 of 1 page < 1 >

1 Simulation running

NODE-RED SERVICE

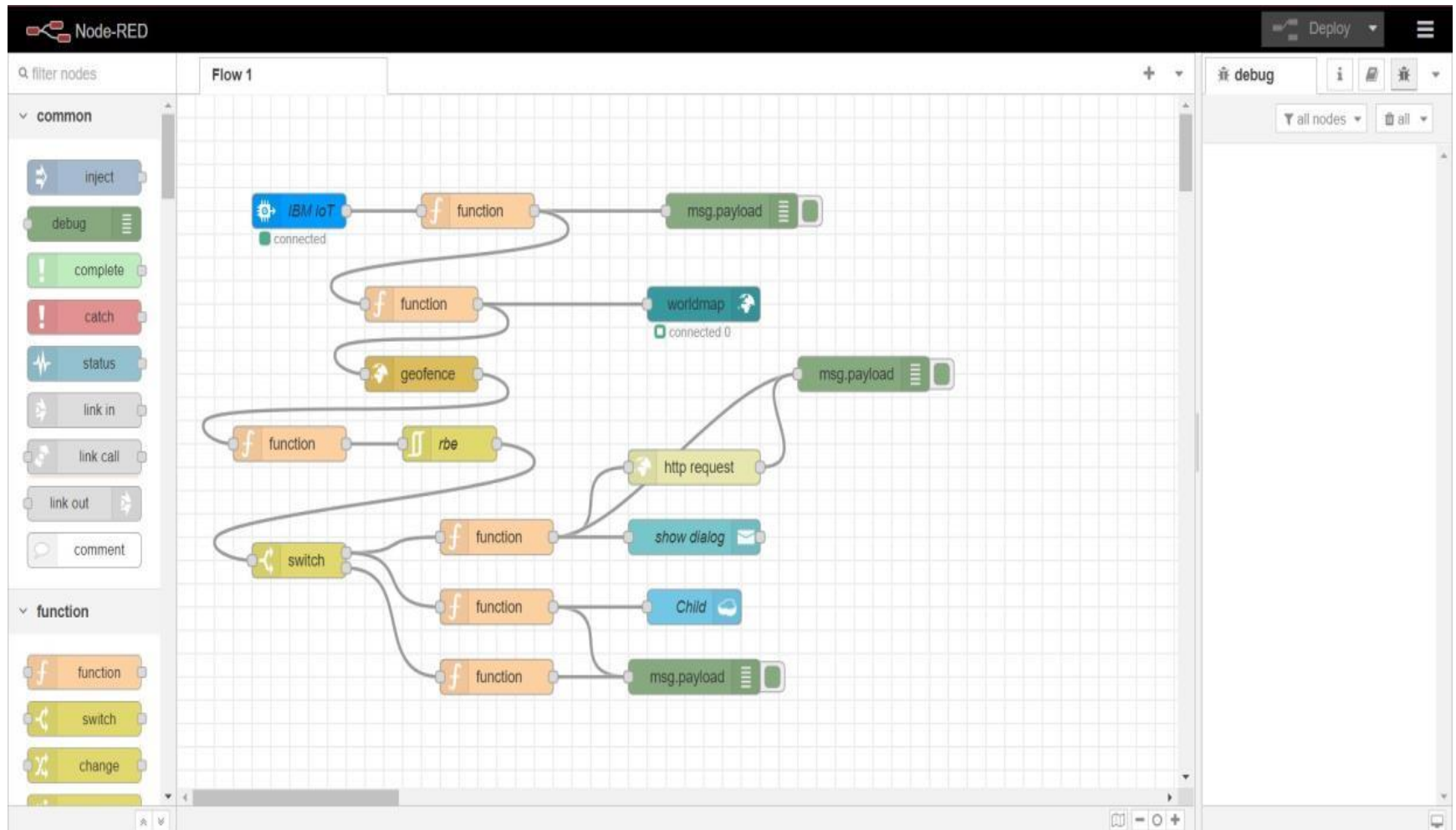


Fig.4,Create Node red service

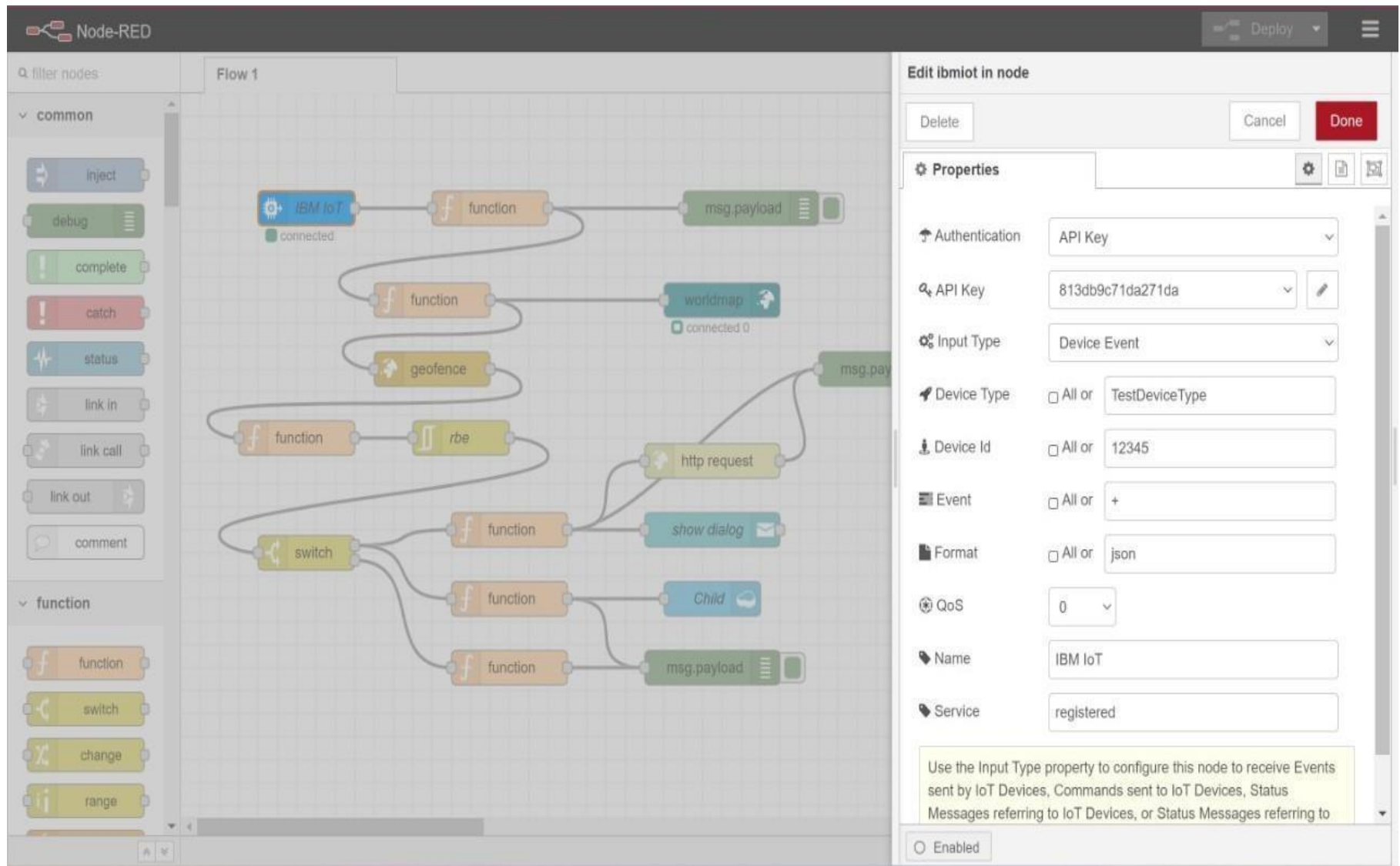


Fig.5,Code in each nodes

Node-RED interface showing a flow titled "Child Tracker" and the "Edit function node" dialog.

Flow Diagram:

- Input: **inject** node.
- Node: **debug** node (labeled "connected").
- Node: **IBM IoT** node.
- Node: **function** node (selected for editing).
- Node: **function** node.
- Node: **geofence** node.
- Node: **function** node.

Edit function node dialog:

- Properties:** Name (empty).
- Setup:** On Start, On Message (selected), On Stop.
- Code:**

```
1 var name = msg.payload.name
2 var lat = msg.payload.lat
3 var lon = msg.payload.lon
4 global.set('latitude',lat)
5 global.set('longitude',lon)
6 global.set('name',name)
7 return msg;
```
- Enabled:** ☐ Enabled

Dashboard:

- Layout: Site, Theme.
- Child Tracker: Map.

Node-RED interface showing a flow titled "Child Tracker" and the "Edit debug node" configuration panel.

Flow 1: Child Tracker

- Input: **inject** (blue)
- Node: **debug** (green, labeled "connected")
- Function: **function** (orange)
- Output: **msg.payload** (green)
- Function: **function** (orange)
- Output: **worldmap** (teal, labeled "connected 0")
- Function: **geofence** (yellow)
- Function: **function** (orange)
- Output: **rbe** (yellow)
- Output: **switch** (yellow)

Edit debug node configuration:

- Output:** msg. payload
- To:** ☒ debug window, ☐ system console, ☐ node status (32 characters)
- Name:** Name
- Enabled:** ☐ Enabled

Dashboard:

- Layout: Site Theme
- Tabs & Links: Child Tracker (Map)

Node-RED

Deploy

filter nodes

Child Tracker

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

Child Tracker

Flow 1

IBM IoT

connected

function

function

geofence

function

Edit function node

Delete

Cancel

Done

Properties

Name

Setup

On Start

On Message

On Stop

1 • msg.payload = {

2 "lat": global.get('latitude'),

3 "lon": global.get('longitude'),

4 "lat": global.get('latitude'),

5 • }

6 return msg;

Enabled

dashboard

Layout

Site

Theme

Tabs & Links

Child Tracker

Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

Flow 1 nodes and connections:

- IBM IoT (connected) → function → msg.payload
- function → worldmap (connected 0)
- worldmap → geofence
- geofence → function → rbe
- rbe → http request
- http request → show dialog
- http request → Child
- http request → msg.payload
- switch → function → show dialog
- switch → function → Child
- switch → function → msg.payload

Edit worldmap node

Properties

- Group: [Child Tracker] Map
- Size: auto
- Start: Latitude: 17.4226372, Longitude: 78.5456505, Zoom: 16
- Map list: 7 selected
- Base map: ESRI Satellite
- Overlays: 5 selected
- Cluster when: zoom level is less than 0 (0, off - 19)
- Max age: Remove markers after 600 seconds
- User menu: Show, Layer menu: Hide
- Lock map: False, Lock zoom: False
- Auto-pan: Disable, Right click: Disable

Enabled

Node-RED interface showing a flow and the 'Edit geofence node' panel.

Flow 1:

- IBM IoT (connected) → function → msg.payload
- function → worldmap (connected 0)
- function → geofence
- function → rbe
- switch → function → http request
- switch → function → show dialog
- switch → function → Child
- switch → function → msg.payload

Edit geofence node:

- Buttons: Delete, Cancel, Done
- Properties: Map view showing a geofence area (purple outline) on a map of Chennai, India. The map includes labels for various locations like Cantonment, Adyar, Besant Nagar, Thiruvannamipur, Kottivakkam, Palavakkam, Neelankarai, Perungudi, Taramani, Adambakkam, Ullagaram, Puzhuthivakkam, Madipakkam, and Kandanchoy.
- Controls: Floor (ground), Ceiling (infinity), Action (add "Inarea" property), and a checkbox for 'Enable output of zones to WorldMap node' (Enabled).

Node-RED interface showing a flow titled "Child Tracker" and the "Edit function node" dialog.

Flow Diagram:

- Input: **IBM IoT** (connected)
- Function Node 1: **function**
- Function Node 2: **function**
- Geofence Node: **geofence**
- Function Node 3: **function**

Edit function node dialog:

- Properties:** Name:
- Setup** | **On Start** | **On Message** | **On Stop**
- Code:**

```
1 msg.payload=msg.location.inarea
2 return msg;
```
- Enabled** (radio button)

Dashboard:

- dashboard** | **Layout** | **Site** | **Theme**
- Tabs & Links:**
 - Child Tracker
 - Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED interface showing a flow editor and the 'Edit filter node' dialog.

Flow Editor:

- Search: filter nodes
- Child Tracker
- Flow 1
- Flow 2
- Common nodes: inject, debug, complete, catch, status, link in, link call, link out, comment
- Function nodes: function, switch

Flow Diagram:

```
graph LR; IoT[IBM IoT] --> F1[function]; F1 --> Payload[msg.payload]; Payload --> F2[function]; F2 --> Worldmap[wordmap]; Worldmap --> Geofence[geofence]; Geofence --> F3[function]; F3 --> RBE[rbe]; RBE --> Switch[switch];
```

Edit filter node dialog:

- Buttons: Delete, Cancel, Done
- Properties:
 - Mode: block unless value changes
 - Property: msg.payload
 - ☒ Apply mode separately for each
 - Property: msg.topic
 - Name: rbe
- Enabled: ☐ Enabled

Dashboard:

- Layout, Site, Theme
- Tabs & Links:
 - Child Tracker
 - Map

<https://node-red-opszk-2022-11-04.eu-gb.bluemix.net/red/#editor-tab-properties>

Node-RED

filter nodes Child Tracker Flow 1 Flow 2

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch

IBM IoT

connected

function

msg.payload

function

worldmap

connected

geofence

function

rbe

switch

Edit switch node

Delete Cancel Done

Properties

Name

Property

msg. payload

is false

→ 1

is true

→ 2

+ add

checking all rules

☐ recreate message sequences

☐ Enabled

dashboard

Layout Site Theme

Tabs & Links

- Child Tracker
 - Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED

filter nodes Child Tracker Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch

function

msg payload

worldmap

connected 0

geofence

rbe

switch

Edit function node

Delete Cancel Done

Properties

Name

Setup On Start On Message On Stop

```
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1)
6
7 newDate = new Date(utc + (3600000 * offset));
8
9 msg.payload = {
10   "message": "Exit",
11   "Time": newDate.toLocaleString(),
12   "name": global.get('name'),
13   "lat": global.get('latitude'),
14   "lon": global.get('longitude')
15 };
16
17 return msg;
```

Enabled

dashboard

Layout Site Theme

Tabs & Links

- Child Tracker
- Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED interface showing a flow titled "Child Tracker" and the "Edit function node" panel.

Flow 1:

- Inject node
- Function node (msg.payload)
- Function node (worldmap)
- Geofence node
- Function node (rbe)
- Switch node

Edit function node:

Properties: Name (Name)

Setup | On Start | **On Message** | On Stop

```
1 var d = new Date();
2
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4
5 var offset = 5.5; // This is the offset for UTC*3, in your case (UTC*1)
6
7 newDate = new Date(utc + (3600000* offset));
8
9 msg.payload={
10   "message": "Entry",
11   "Time": newDate.toLocaleString(),
12   "name": global.get('name'),
13   "lat": global.get('latitude'),
14   "lon": global.get('longitude')
15 };
16
17 return msg;
```

Enabled

Dashboard: Layout | Site | Theme

Tabs & Links: Child Tracker, Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED

Deploy

filter nodes:

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

change

range

```
graph LR; IoT[IBM IoT] --> F1[function]; F1 --> P1[msg.payload]; P1 --> F2[function]; F2 --> WM[wordmap]; WM --> GE[geofence]; GE --> F3[function]; F3 --> RBE[rbe]; RBE --> HR[http request]; HR --> SW[switch]; SW --> F4[function]; F4 --> SD[show dialog]; SW --> F5[function]; F5 --> CH[Child]; SW --> F6[function]; F6 --> P2[msg.payload];
```

Edit http request node

Delete

Cancel

Done

Properties

Method

GET

URL

https://www.fast2sms.com/dev/bulkV2?authorizati

Payload

Ignore

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

Return

a UTF-8 string

Name

Name

Enabled

Node-RED

Child Tracker

Flow 1

Flow 2

filter nodes

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

switch

function

function

msg.payload

worldmap

connected 0

geofence

rbe

switch

function

function

function

function

Edit notification node

Delete

Cancel

Done

Properties

Layout

OK / Cancel Dialog

Send to all browser sessions. ☒

Default action label

OK

Secondary action label

(optional label for Cancel button)

Accept raw HTML/JavaScript input in msg.payload to format popup. ☐

Class

[msg.className]

Topic

[msg.topic]

Name

Show Dialoge

Note: checking Accept raw HTML/JavaScript can allow injection of

Enabled

dashboard

Layout

Site

Theme

Tabs & Links

Child Tracker

Map

<https://node-red-opszk-2022-11-04.eu-gb.mybluemix.net/red/#editor-tab-properties>

Node-RED

filter nodes

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

IBM IoT

connected

function

msg.payload

function

worldmap

connected 0

geofence

function

rbe

http request

msg.payload

switch

function

show dialog

function

Child

function

msg.payload

Edit cloudant out node

Delete

Cancel

Done

Properties

Service

External cloudant or couchdb service

Server

https://93dc87d4-df75-4ee1-a851-2026

Database

childsafetydb

Operation

insert


☐ Only store msg.payload object?

Name

Child

Enabled

Connecting with IBM Cloud: Using IBM IOT node through the API key



Browse IBM Cloud Apps

+ Generate API Key

Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

Type the app description to search for

<input type="checkbox"/> Key	Description	Role	Expires	
2 results				
<input type="checkbox"/> a-4o1qxb-d5wguvebrf	-	Standard Application	-	
<input type="checkbox"/> a-4o1qxb-ecmygwzdce	API Key for the device simulator	Standard Application	-	

1 Simulation running

Apps using your microphone: Google Chrome

Browse IBM Cloud Apps

+

Generate API Key

Browse API Keys

Type the app description to search for

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

Key

Description

Role

Expires

2 results

a-4o1qxb-d5wguvebrf

-

Standard Application

-

API Key Information

Access Control/Permissions

X

Key

a-4o1qxb-d5wguvebrf

Last Edited By

310819106007@smartinternz.com

Description

-

Expires

Never

Date Added

Nov 10, 2022 2:20 PM

Last Update

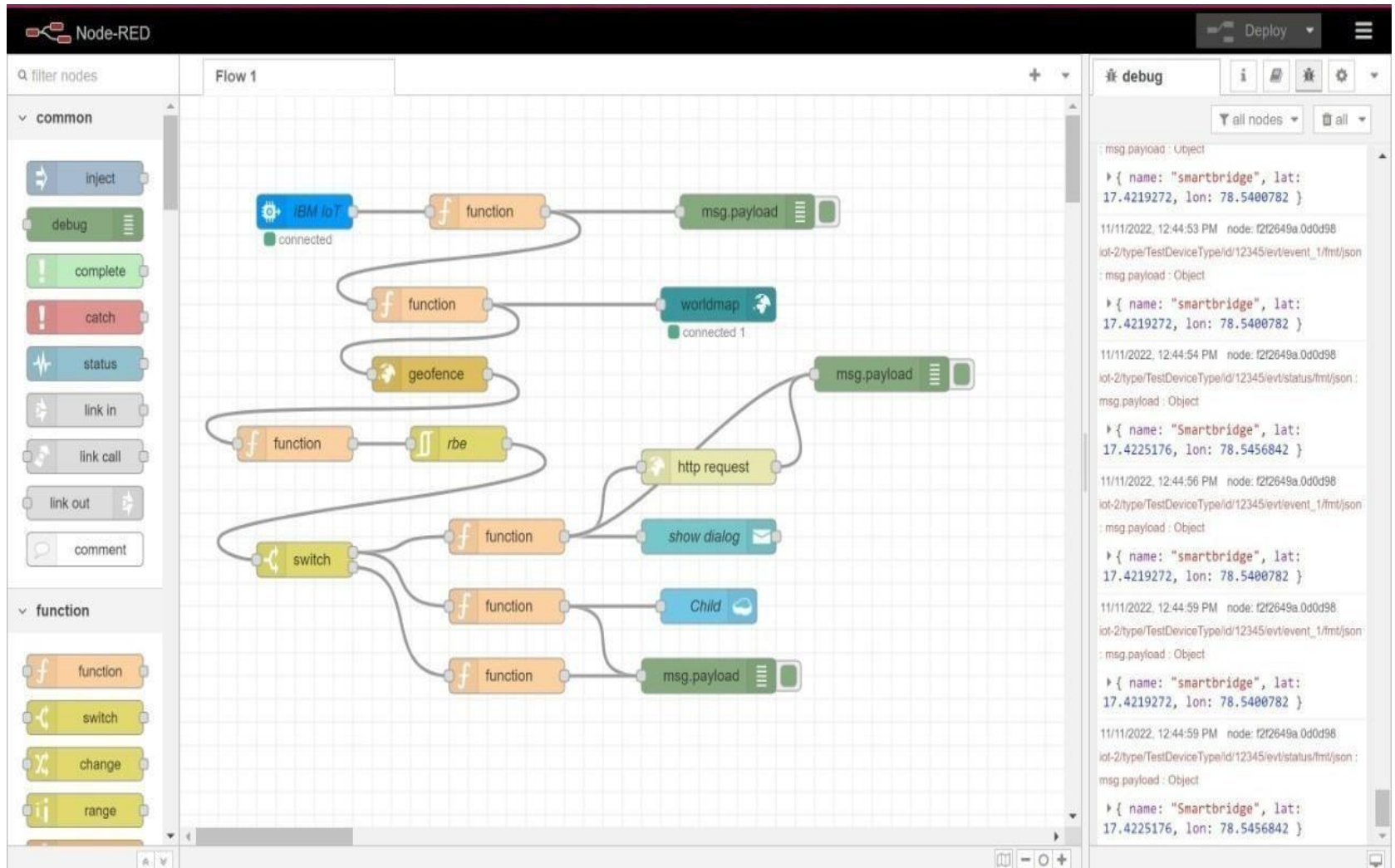
Nov 10, 2022 2:20 PM

1 Simulation running

Transferring values from Python Code:

The image shows a Windows desktop with two open Python 3.7.0 Shell windows. The left window displays a Python script for connecting to the IBM IoT platform using the wiotp.sdk.device module. The script defines a configuration object, creates a DeviceClient, connects, and enters a loop publishing location data (latitude, longitude) as JSON messages. The right window shows the execution output, indicating a successful connection and a series of 'Data published to IBM IoT platform' messages with timestamps and coordinates.

Node-Red:



Creating Cloudant DB and integrating Node-Red with the Web UI

The screenshot displays the Cloudant Databases web interface. On the left is a dark sidebar with navigation icons. The main header area includes the title 'Databases', a 'Database name' dropdown, a 'Create Database' button, and links for JSON, documentation, and notifications. Below the header, a section titled 'Your Databases' contains a table with two entries:

Name	Size	# of Docs	Partitioned	Actions
childsafety	14 bytes	1	No	[Icons for edit, lock, delete]
childsafetydb	15 bytes	1	No	[Icons for edit, lock, delete]

At the bottom of the interface, a status bar indicates 'Showing 1-2 of 2 databases.' and 'Databases per page 20'. A 'Log Out' button is located in the bottom-left corner of the sidebar.

child safety db
Document ID
Options
{ } JSON

All Documents +

Query

Permissions

Changes

Design Documents +

☐

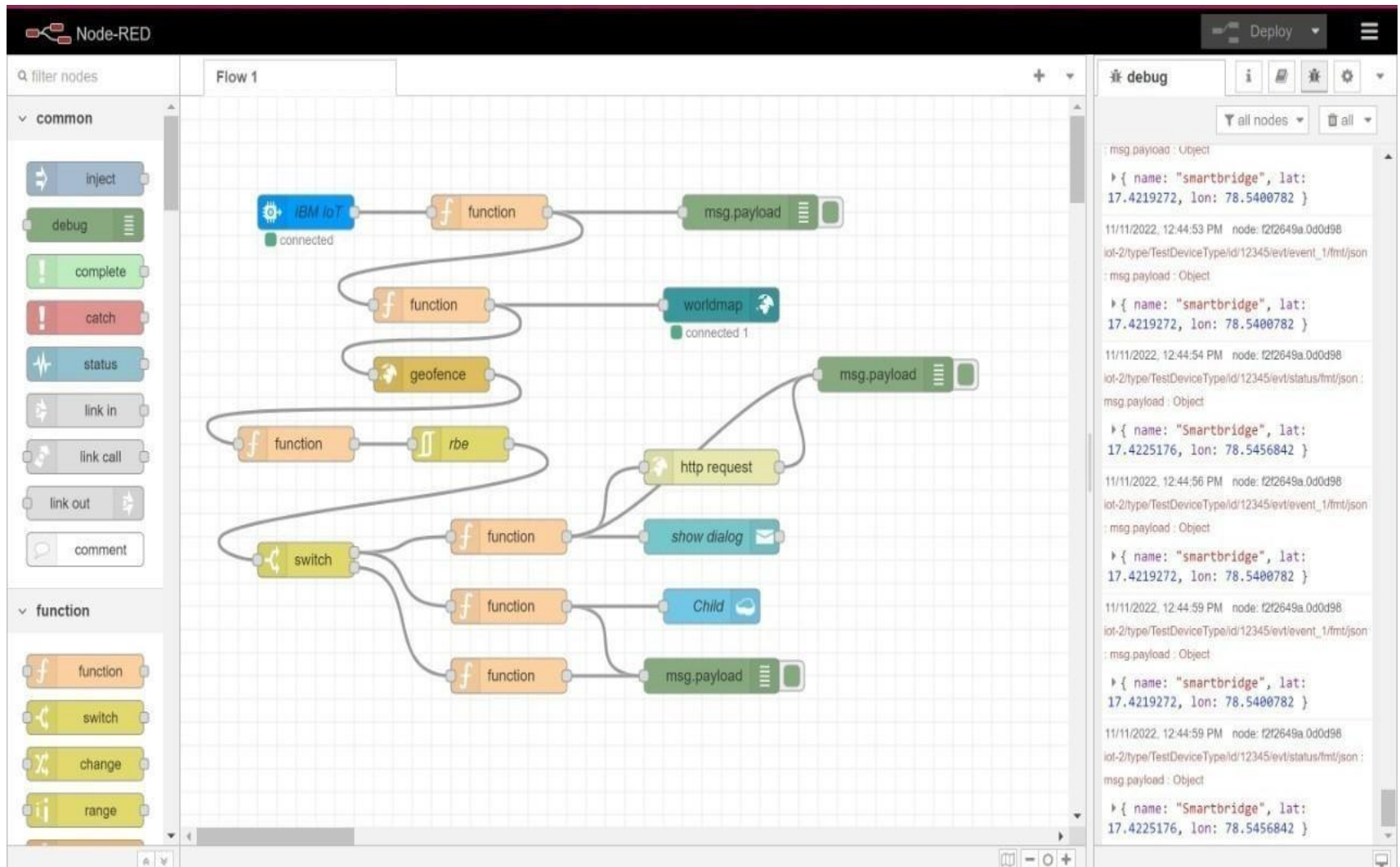
Table	Metadata	{ } JSON
-------	-----------------	----------

	id	key	value
<input type="checkbox"/>	e375b703e3d589d1a5e0108d5c7fd2...	e375b703e3d589d1a5e0108d5c7fd2...	{"rev": "2-3f59b999d6fe5a17091eba..."}

Log Out

Showing document 1 - 1. Documents per page: 20 >

Node-Red Service with Cloudant Database:



Node-RED

filter nodes

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

Flow 1 diagram:

```
graph LR
    IBM[IBM IoT7] --> F1[function]
    F1 --> MP1[msg.payload]
    F1 --> F2[function]
    F2 --> WM[worldmap]
    WM --> GE[geofence]
    GE --> F3[function]
    F3 --> RBE[rbe]
    RBE --> F4[function]
    F4 --> HR[http request]
    HR --> MP2[msg.payload]
    HR --> SD[show dialog]
    HR --> CH[Child]
    HR --> F5[function]
    F5 --> MP3[msg.payload]
    HR --> SW[switch]
    SW --> F6[function]
    SW --> F7[function]
    SW --> F8[function]
```

Edit cloudant out node

Delete Cancel Done

Properties

Service: External cloudant or couchdb service

Server: https://93dc87d4-df75-4ee1-a851-2026

Database: childsafetfdb

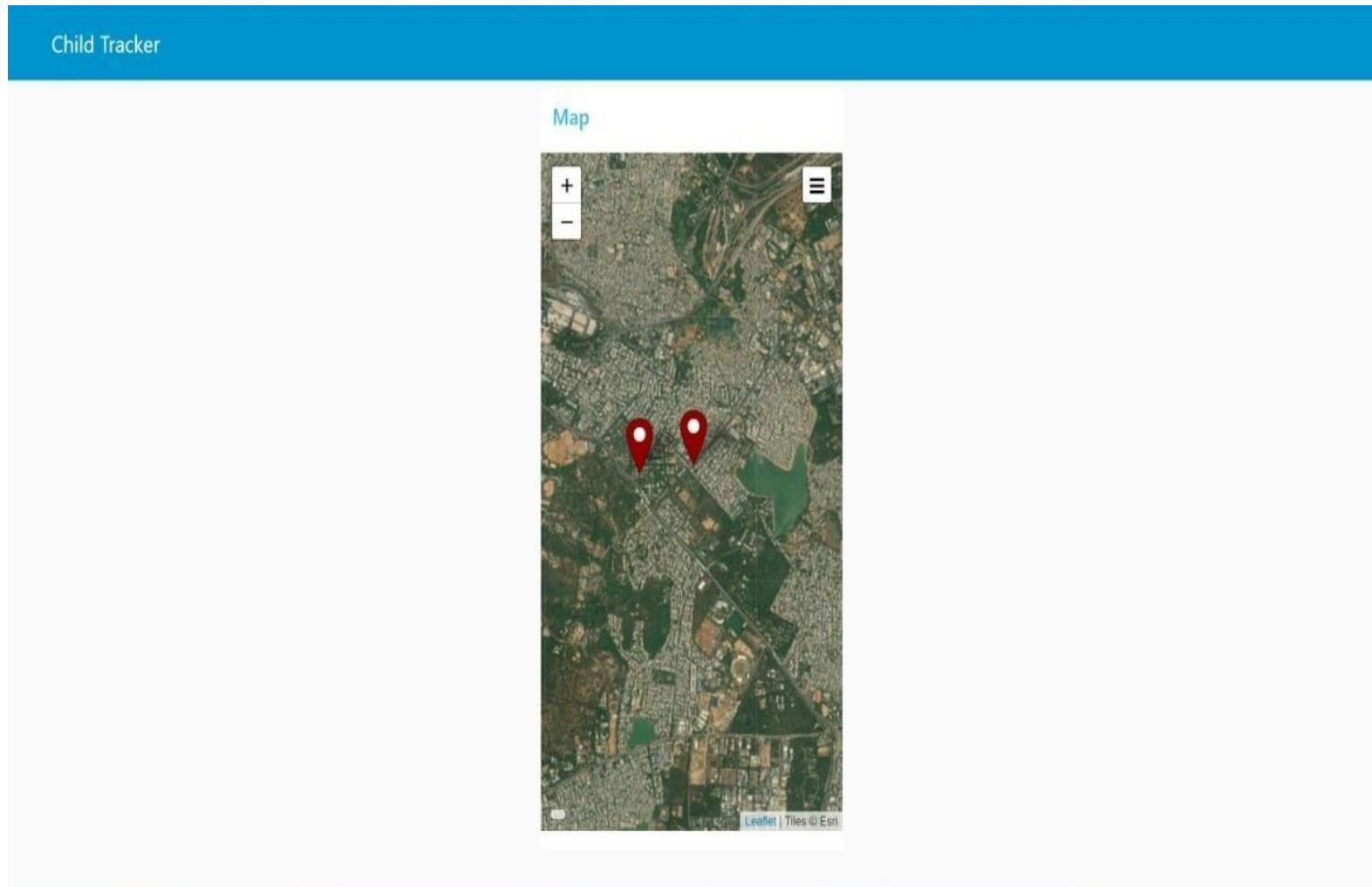
Operation: Insert

☐ Only store msg.payload object?

Name: Child

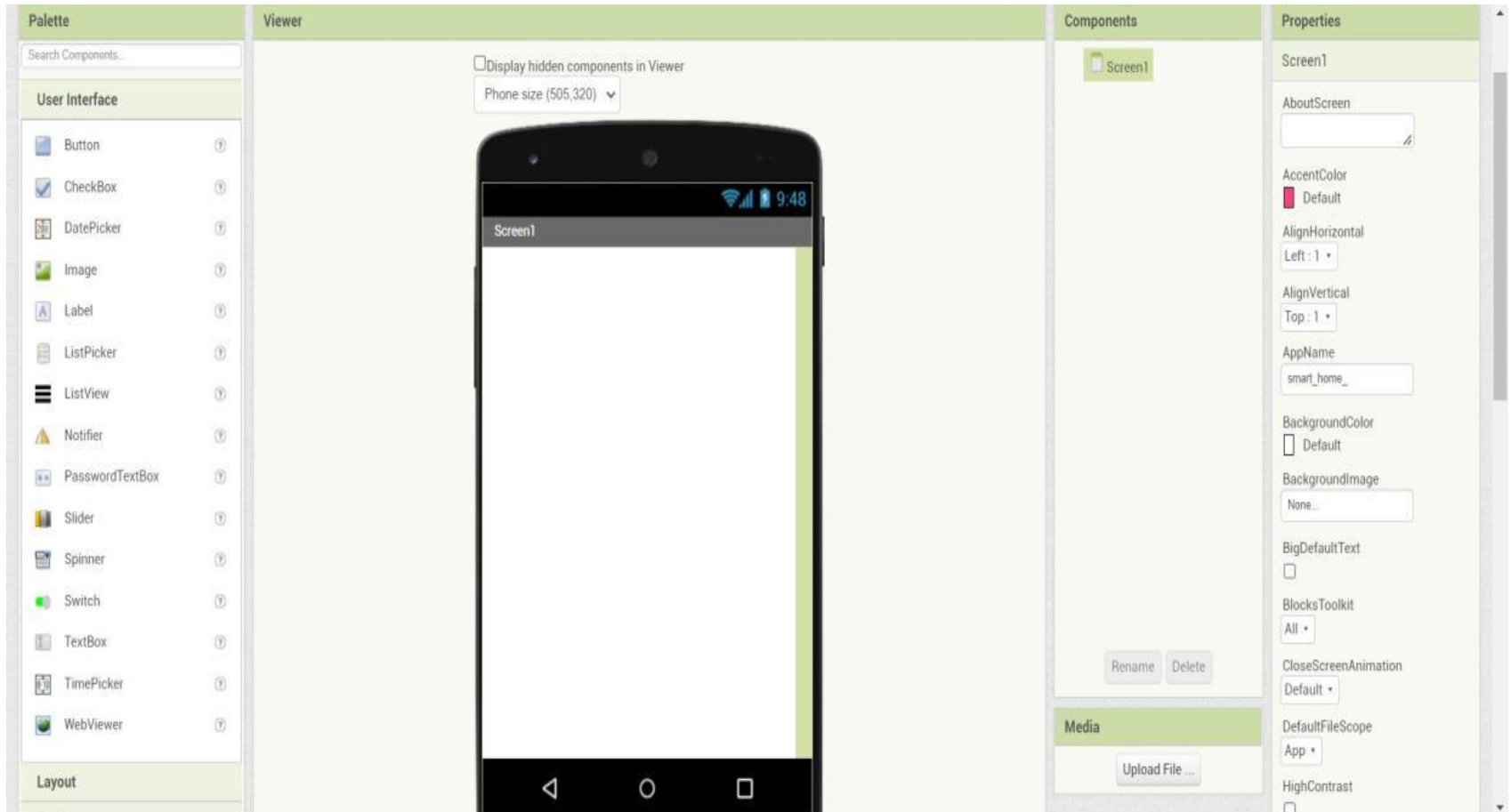
Enabled

Node-Red Dashboard(Web ui):



Creating the MIT app and Showing the child's location

Create App in MIT App inventor:



Block Configuration:

The screenshot displays the 'ChildTracker' application interface. At the top, a green header bar contains the app name 'ChildTracker' and navigation buttons: 'Screen1', 'Add Screen ...', 'Remove Screen', and 'Publish to Gallery'. On the right side of the header are 'Designer' and 'Blocks' tabs.

The left sidebar is divided into two sections: 'Blocks' and 'Media'. The 'Blocks' section lists components for 'Screen1', including 'VerticalArrangement1' (containing 'TextBox1', 'Label1', 'Button1', 'Label2', 'Label3', 'Label4', and 'Button2') and 'Map1' (containing 'Circle1', 'Marker1', 'Marker2', 'LocationSensor1', 'FirebaseDB1', and 'Navigation1'). Below this list are 'Rename' and 'Delete' buttons. The 'Media' section at the bottom has an 'Upload File' button.

The main 'Viewer' area shows a visual programming workspace. It begins with an 'Initialize global flag to 0' block. This is followed by a 'when FirebaseDB1 .DataChanged' event listener. Inside the listener's 'do' block, there is an 'if' statement. The 'if' statement has three conditions: 'get value' equals 'lat', 'get value' equals 'long', and 'get value' equals 'address'. Each condition has a corresponding 'then' block: 'set Label3 . Text' to 'get value' for 'lat', 'set Label3 . Text' to 'get value' for 'long', and 'set Label2 . Text' to 'get value' for 'address'. A yellow tooltip is visible over the 'long' condition, stating: 'If the first value is true, then do the first block of statements. Otherwise, if the second value is true, do the second block of statements.'

At the bottom left of the workspace, there are two warning icons (a yellow triangle and a red X) with a 'Show Warnings' button below them. On the right side of the workspace, there is a backpack icon and a vertical toolbar with a target icon, plus/minus buttons, and a trash can icon.

At the bottom of the application window, there is a link for 'Privacy Policy and Terms of Use'.

ChildTracker

Screen1Add Screen ...Remove ScreenPublish to Gallery

DesignerBlocks

Blocks

Screen1

- VerticalArrangement1
 - TextBox1
 - Label1
 - Button1
 - Label2
 - Label3
 - Label4
 - Button2
- Map1
 - Circle1
 - Marker1
 - Marker2
- LocationSensor1
- FirebaseDB1
- Navigation1

Any component

RenameDelete

Media

Upload File ...

Viewer

when Button2.Click

do

- set Label3.Text to "17.4219272"
- set Label4.Text to "78.5488783"
- call Map1.PanTo
 - latitude LocationSensor1.Latitude
 - longitude LocationSensor1.Longitude
 - zoom 15

when Navigation1.GotDirections

directions points distance duration

do

- set Navigation1.StartLocation to Marker1
- set Navigation1.EndLocation to Marker2
- call Navigation1.RequestDirections

00

Show Warnings

Privacy Policy and Terms of Use



ChildTracker Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks

- Screen1
 - VerticalArrangement1
 - TextBox1
 - Label1
 - Button1
 - Label2
 - Label3
 - Label4
 - Button2
 - Map1
 - Circle1
 - Marker1
 - Marker2
 - LocationSensor1
 - FirebaseDB1
 - Navigation1
- Any component

Rename Delete

Media

Upload File ...

Viewer

when LocationSensor1 . LocationChanged

latitude longitude altitude speed

do set Label1 . Text to "Updated"

if get global flag = 0

then

call FirebaseDB1 . StoreValue

tag "lat"

valueToStore get latitude

call FirebaseDB1 . StoreValue

tag "long"

valueToStore get longitude

call FirebaseDB1 . StoreValue

tag "address"

valueToStore LocationSensor1 . CurrentAddress

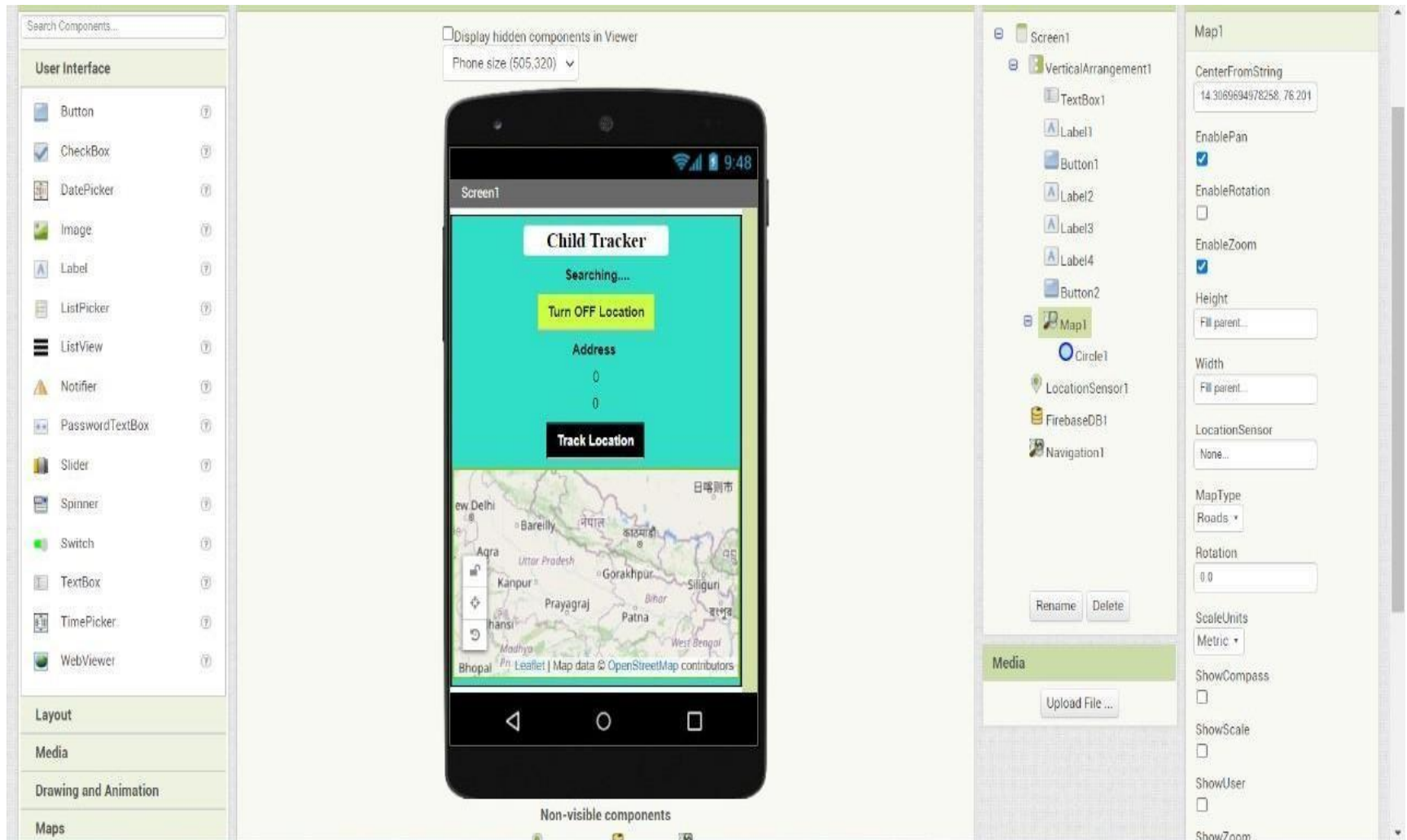
set latitude to "17.4219272"

set longitude to "78.5488783"

Show Warnings

Privacy Policy and Terms of Use

Output(App inventor):



Location Status:



CONCLUSION

This paper to ensure the safety of children and increase their confidence. Many experimenters are operating in this area and have formulated different technologies to aid children. The key represented in this paper takes the advantage of smartphones which proposes affluent elements like Google maps, SMS, etc. The child safety and protection device is proficient in acting as a smart IoT device. It equips parents with real-time location, the surrounding temperature, and along with an alarm buzzer for their child's circumstances and the capability to locate their child. This paper depicts the fundamental design concept and functionality along with the anticipated consequences.