

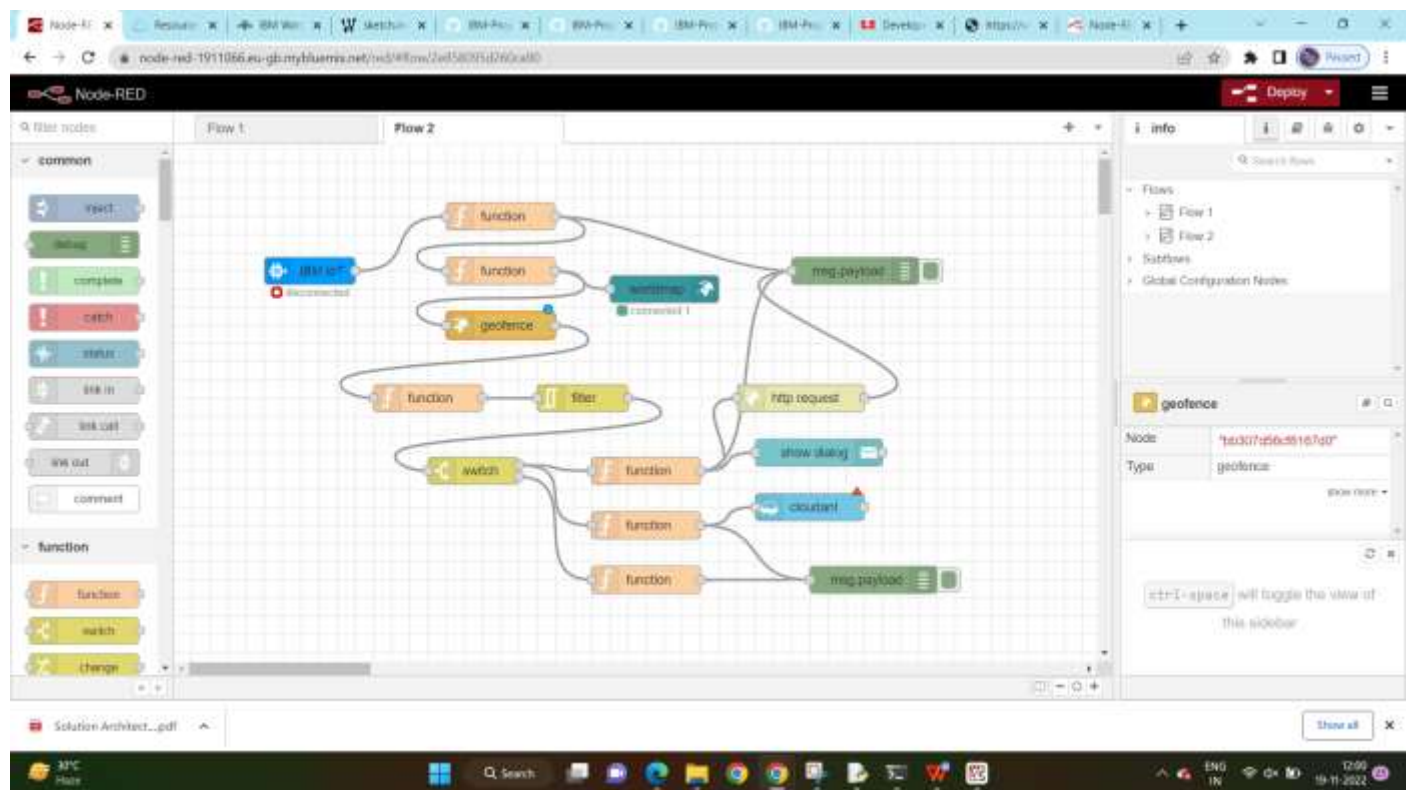
Project Development - Sprint 3

Iot Based Safety For Child Safety Monitoring & Notification

Team ID: PNT2022TMID30207

Creating Node - Red Service and Connecting with IBM cloud

Creating Node-Red Service :



Codes in Each Node:

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with several nodes: a 'msg.payload' node, a 'function' node, a 'switch' node, and several 'function' nodes. The 'Edit function node' dialog box is open on the right, showing the 'Properties' tab. The properties are:

- Authentication: API Key
- API Key: child_id
- Input Type: Device Event
- Device Type: All or IOT_CHILD
- Device Id: All or 16022002
- Event: All or
- Format: All or json
- QoS: 0
- Name: IBM IoT
- Service: registered

The dialog box also has a 'Done' button and a 'Cancel' button.

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with several nodes: a 'msg.payload' node, a 'function' node, a 'switch' node, and several 'function' nodes. The 'Edit function node' dialog box is open on the right, showing the 'Properties' tab. The properties are:

- Name: Name

The dialog box also has a 'Done' button and a 'Cancel' button. Below the properties, there is a 'Setup' tab and a 'Code' tab. The 'Code' tab is selected, showing the following 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;
```

Node-RED interface showing a flow diagram with various nodes (function, geofence, etc.) and a sidebar with storage and IBM Watson components. The right panel displays the 'Edit function node' configuration, including a code editor with the following JavaScript code:

```
1 msg.payload = {
2   'name': global.get('name'),
3   'lat': global.get('latitude'),
4   'lon': global.get('longitude')
5 }
6 return msg;
```

The interface also shows a 'Solution Architect...pdf' document and a 'Show all' button.

Node-RED interface showing the same flow diagram. The right panel displays the 'Edit geofence node' configuration, including a map view of Hyderabad, India, with a geofence boundary. The map shows the city layout with roads and landmarks. Below the map, there are settings for 'Floor' (ground) and 'Ceiling' (infinity).

Node-RED interface showing a flow diagram with various nodes including storage, IBM Watson, and function nodes. The right sidebar displays the "Edit function node" configuration, showing the "On Message" tab with the following code:

```
1 msg.payload = msg.location.traces;
2 return msg;
```

The interface also shows a "Deploy" button and a "Solution Architect...pdf" link.

Node-RED interface showing a flow diagram with various nodes including storage, IBM Watson, and function nodes. The right sidebar displays the "Edit switch node" configuration, showing the "Property" dropdown set to "msg.payload" and the "is true" condition. The "checking all rules" checkbox is checked.

Node-RED interface showing a flow diagram and the Edit function node configuration.

Flow Diagram: The flow starts with a 'msg.payload' input, followed by a 'function' node, then a 'switch' node. The 'switch' node has two outputs: one leading to a 'function' node and another leading to a 'function' node. Both 'function' nodes lead to a 'msg.payload' output.

Edit function node configuration:

- Name: Name
- Properties: Enabled
- On Message:

```
1 var d = new Date();
2 var utc = d.getTime() + (d.getTimezoneOffset()*1000);
3 var offset = 5.5;
4 newDate = new Date(utc + (3600000*offset));
5
6 msg.payload = {
7   "message": "Exit",
8   "time": newDate.toISOString(),
9   "name": global.get('name'),
10  "lat": global.get('latitude'),
11  "lon": global.get('longitude')
12 };
13
14 return msg;
```

Node-RED interface showing a flow diagram and the Edit function node configuration.

Flow Diagram: The flow starts with a 'msg.payload' input, followed by a 'function' node, then a 'switch' node. The 'switch' node has two outputs: one leading to a 'function' node and another leading to a 'function' node. Both 'function' nodes lead to a 'msg.payload' output.

Edit function node configuration:

- Name: Name
- Properties: Enabled
- On Message:

```
1 var d = new Date();
2 var utc = d.getTime() + (d.getTimezoneOffset()*1000);
3 var offset = 5.5;
4 newDate = new Date(utc + (3600000*offset));
5
6 msg.payload = {
7   "message": "Entry",
8   "time": newDate.toISOString(),
9   "name": global.get('name'),
10  "lat": global.get('latitude'),
11  "lon": global.get('longitude')
12 };
13
14 return msg;
```

