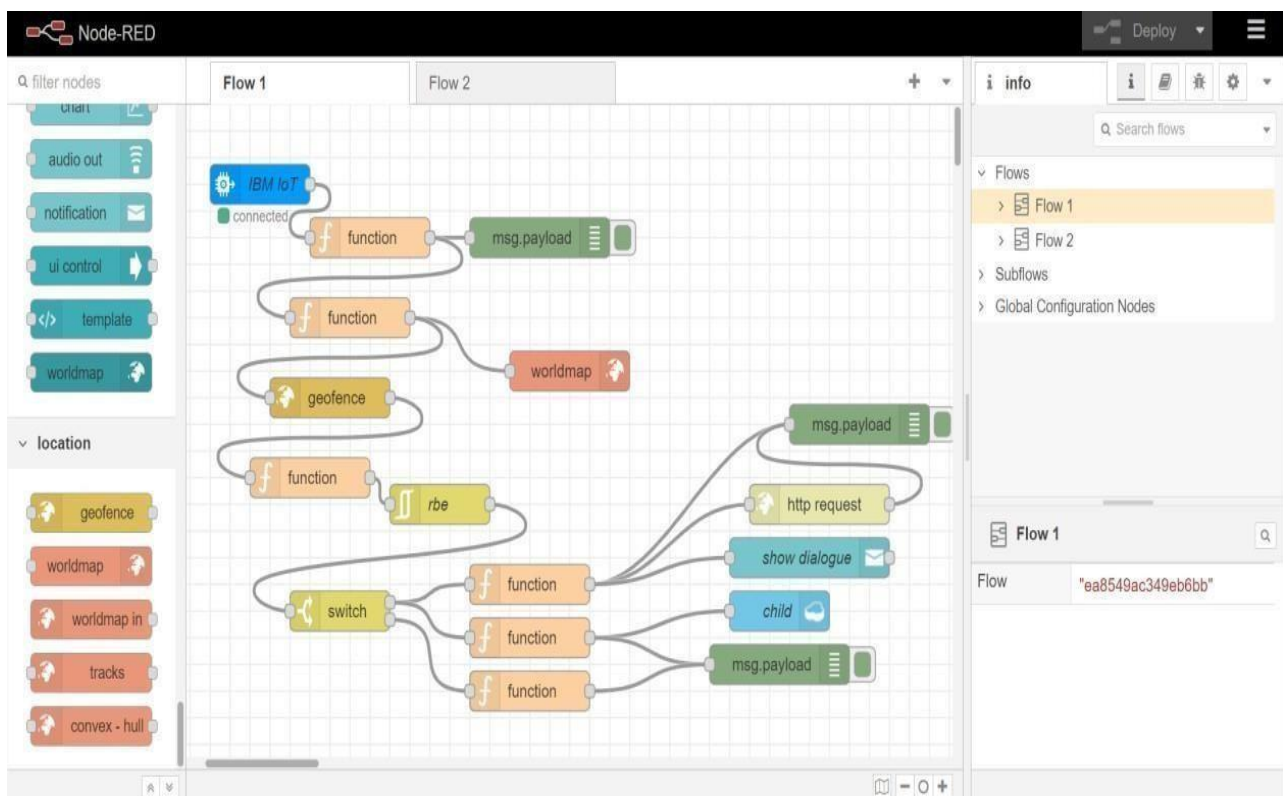


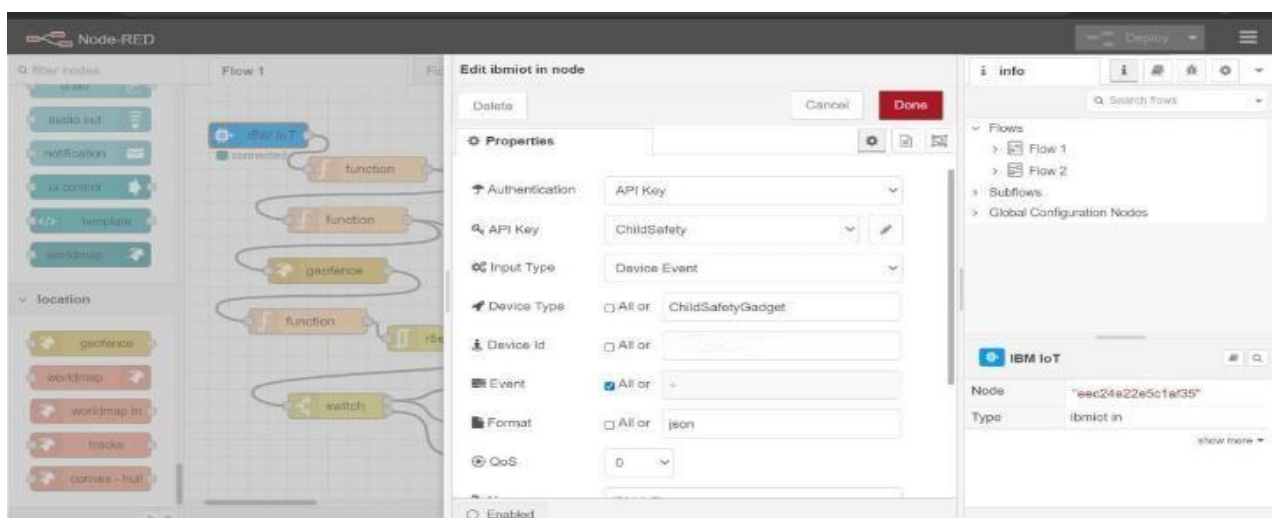
Node Red Cloudant DB-Communication

Date	19-11-2022
Team ID	PNT2022TMID37130
Project Name	Project - IOT based safety gadget for child safety monitoring and notification
Maximum Marks	4 Marks

Step 1: Connections.



Step 2: Code in nodes



Step 3:Edit function mode(code)

The screenshot shows the Node-RED interface with the 'Edit function node' dialog open. The dialog has a 'Properties' tab and a 'Name' field. Below these are four tabs: 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected, and it contains the following JavaScript 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;
```

The right sidebar shows the 'info' panel with a search bar and a list of flows. The node is identified as 'function: 76ec77ccd270c2c7'.

Step 4:Changing code

The screenshot shows the Node-RED interface with the 'Edit function node' dialog open. The dialog has a 'Properties' tab and a 'Name' field. Below these are four tabs: 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected, and it contains 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 right sidebar shows the 'info' panel with a search bar and a list of flows. The node is identified as 'function: 77b707bf262aa6f4'.

Step 5: Geofencing mode

The screenshot shows the Node-RED interface with the 'Edit geofence node' dialog open. The dialog has a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' tab is active, showing a map with a purple geofence circle. The map includes labels for 'St. pius pg college', 'St pius apurva's hostel', 'Vedic Systems', 'Sneha Apartments', 'Nacharam road', 'Dolphin', 'Mamas Kitchen', 'house', 'SHREE POOJA HOSPITAL', 'Bapuji Maternity and Nursing Home', and 'bapuji nagar park'. The 'Node' field in the properties panel contains the ID 'ee4695a46b4858f1' and the 'Type' is 'geofence'.

Step 6: Worldmap node with properties

The screenshot shows the Node-RED interface with the 'Edit worldmap node' dialog open. The dialog has a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' tab is active, showing various settings for the worldmap node. The settings include: 'Start' (Latitude: 17.4219272, Longitude: 78.5400783, Zoom: 16), 'Map list' (1 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), and 'Lock zoom' (False). The 'Node' field in the properties panel contains the ID '5ed11e914b5614ea' and the 'Type' is 'worldmap'. A note at the bottom states: 'The map can be found [here](#).'

Step 7: Apply the code with different node

The screenshot shows the Node-RED interface with the 'Edit function node' dialog open. The 'On Message' tab is selected, and the code is as follows:

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

The right sidebar shows the 'info' panel with the following details:

- Node: "3585a4ac8098085f"
- Type: function

Step 8: Code with different node

The screenshot shows the Node-RED interface with the 'Edit switch node' dialog open. The 'Property' is set to 'msg.payload'. The rules are configured as follows:

- Rule 1: is false → 1
- Rule 2: is true → 2

The right sidebar shows the 'info' panel with the following details:

- Node: "16b63596382b4cc6"
- Type: switch

Step 9: Code with different node

The screenshot shows the Node-RED web interface. On the left, a sidebar lists various nodes under categories like 'location'. The main workspace shows a flow with an 'IBM IoT' node connected to a 'function' node. The 'Edit function node' panel is open, showing the following JavaScript code:

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

The 'info' panel on the right displays the node's details:

- Node: "322663c3205cc798"
- Type: function

Step 10: code with different node

The screenshot shows the Node-RED web interface, similar to Step 9. The 'Edit function node' panel is open, displaying the same JavaScript code as in Step 9:

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

The 'info' panel on the right displays the node's details for a different node:

- Node: "dfef78abe1ae2fd4"
- Type: function

Step 11: Code with different node

Node-RED interface showing the 'Edit function node' dialog. The code in the function node is as follows:

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

The right sidebar shows the 'info' panel with a search bar and a list of flows: Flow 1, Flow 2, Subflows, and Global Configuration Nodes. Below the list, the selected node is identified as 'function: 36df960d0bd4f874' with a 'show more' link.

Step 12: Http request node->properties

Node-RED interface showing the 'Edit http request node' dialog. The properties are:

- Method: GET
- URL: <https://www.fast2sms.com/dev/wallet?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

The right sidebar shows the 'info' panel with a search bar and a list of flows: Flow 1, Flow 2, Subflows, and Global Configuration Nodes. Below the list, the selected node is identified as 'http request' with a 'show more' link.

Step 13: Notification node

The screenshot shows the Node-RED web interface. On the left, a palette of nodes is visible, including 'notification'. In the center workspace, a flow is being edited, featuring an 'IBM IoT' node connected to a 'function' node, which then connects to a 'notification' node. The right-hand panel is titled 'Edit notification node' and contains the following configuration options:

- 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 dialogue
- Enabled:** ☐

The right sidebar shows the 'info' panel with a search bar and a list of flows. Below this, the 'show dialogue' node is highlighted, showing its Node ID as '388bcb2648d1348b' and its Type as 'ui_toast'.

Step 14: Cloudant node

The screenshot shows the Node-RED web interface. In the center workspace, the same flow as in Step 13 is visible. The right-hand panel is titled 'Edit cloudant out node' and contains the following configuration options:

- Service:** Cloudant-pn-99850
- Database:** noderedapqzh20221108
- Operation:** insert
- Only store msg.payload object?:** ☐
- Name:** child
- Enabled:** ☐

The right sidebar shows the 'info' panel with a search bar and a list of flows. Below this, the 'child' node is highlighted, showing its Node ID as '7f199a311bbfac36' and its Type as 'cloudant out'.