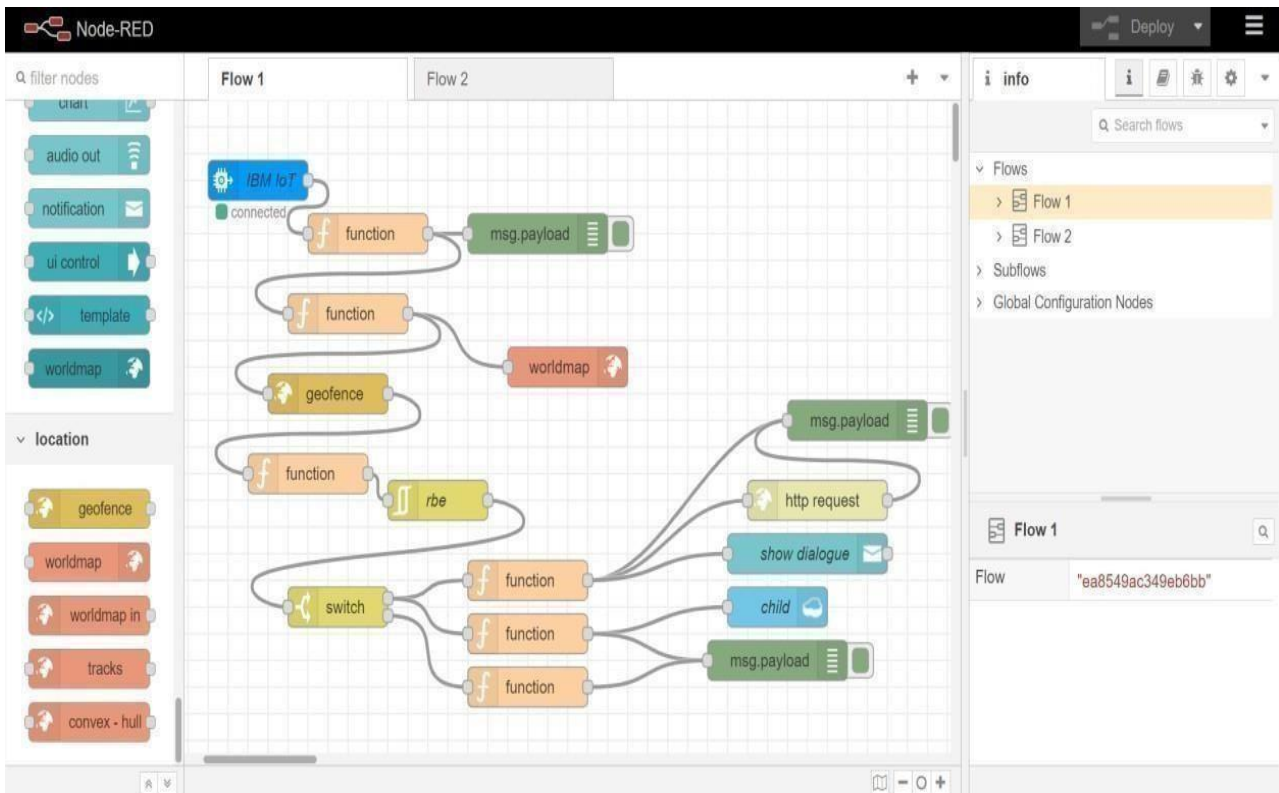


Project Development Phase SPRINT-3

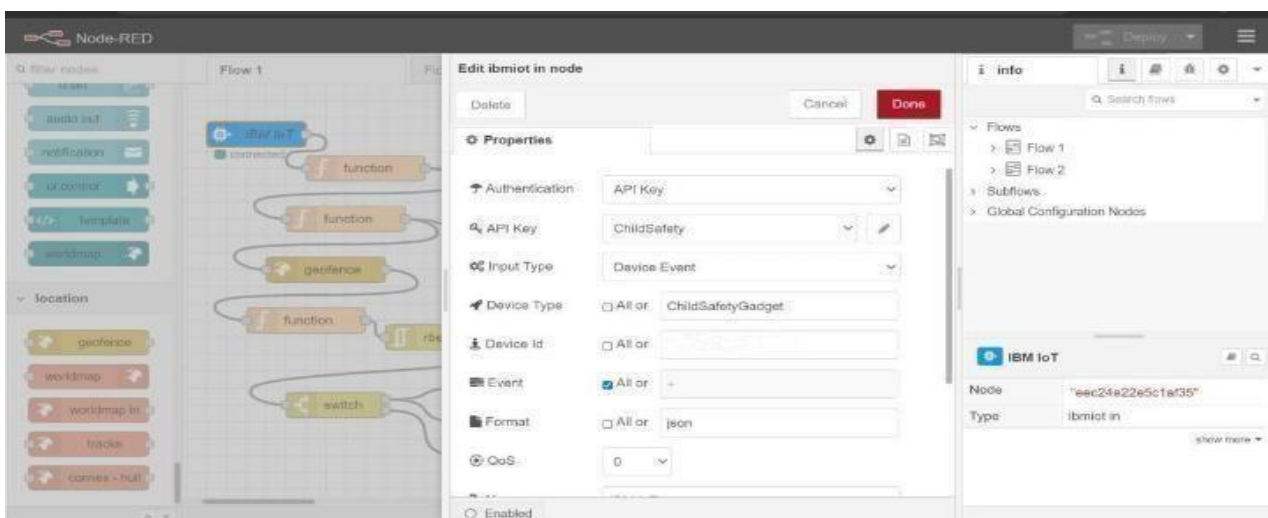
Date	8 November 2022
Team ID	PNT2022TMID23712
Project Name	Project - IOT Based Safety Gadget for Child Safety Monitoring & Notification

NODE RED- Cloudant DB Communication:

Step 1: Connections



Step 2: Code in nodes



Step 3: Edit function mode(code)

The screenshot shows the Node-RED web interface. On the left, a palette of nodes is visible, including 'audio out', 'notification', 'ui control', 'template', 'worldmap', and a 'location' category with 'geofence', 'worldmap', 'worldmap in', 'tracks', and 'convex-hull'. The main workspace shows a flow with an 'IBM IoT' node connected to a 'function' node. The 'Edit function node' dialog is open, displaying 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 dialog also shows tabs for 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected. The 'Properties' section shows a 'Name' field. The right sidebar shows the 'info' panel with a search bar and a list of flows (Flow 1, Flow 2, Subflows, Global Configuration Nodes). Below this, a node information panel shows the node ID '76ec77ccd270c2c7' and its type 'function'.

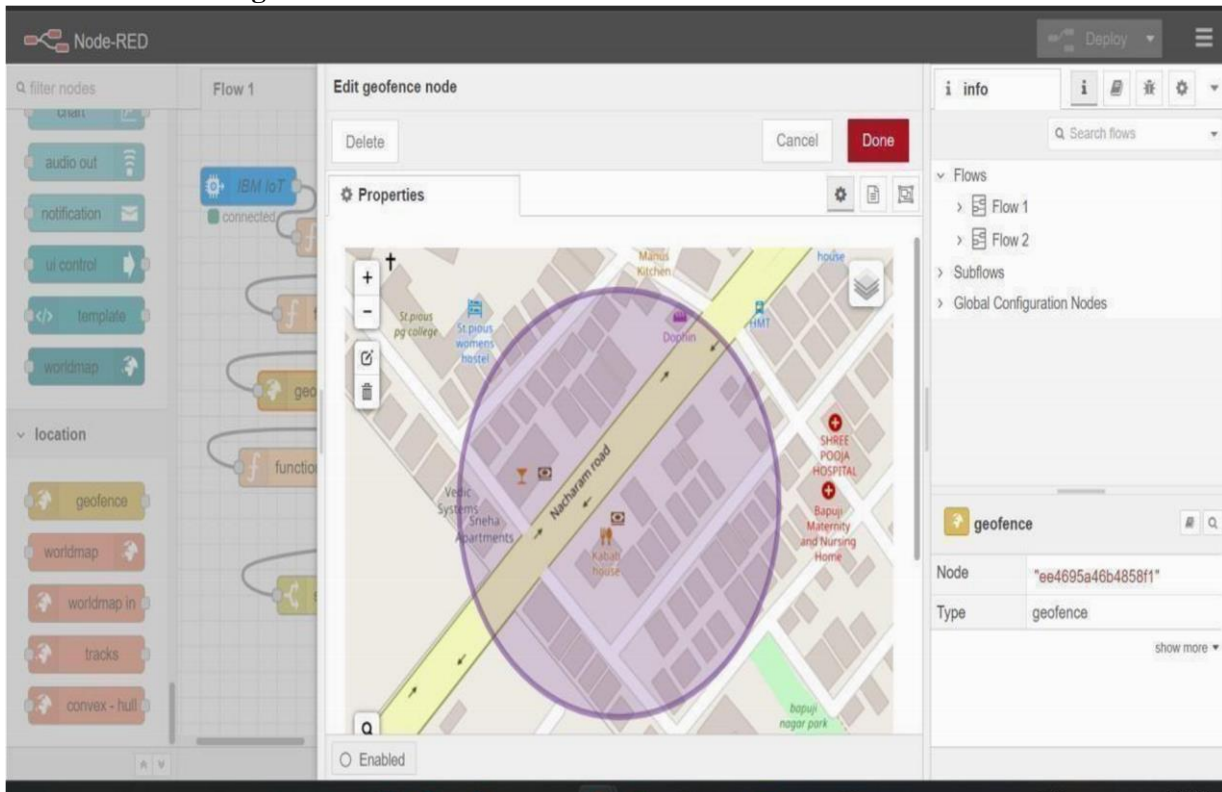
Step 4:

The screenshot shows the Node-RED web interface, similar to the previous one. The 'Edit function node' dialog is open, displaying the following JavaScript code:

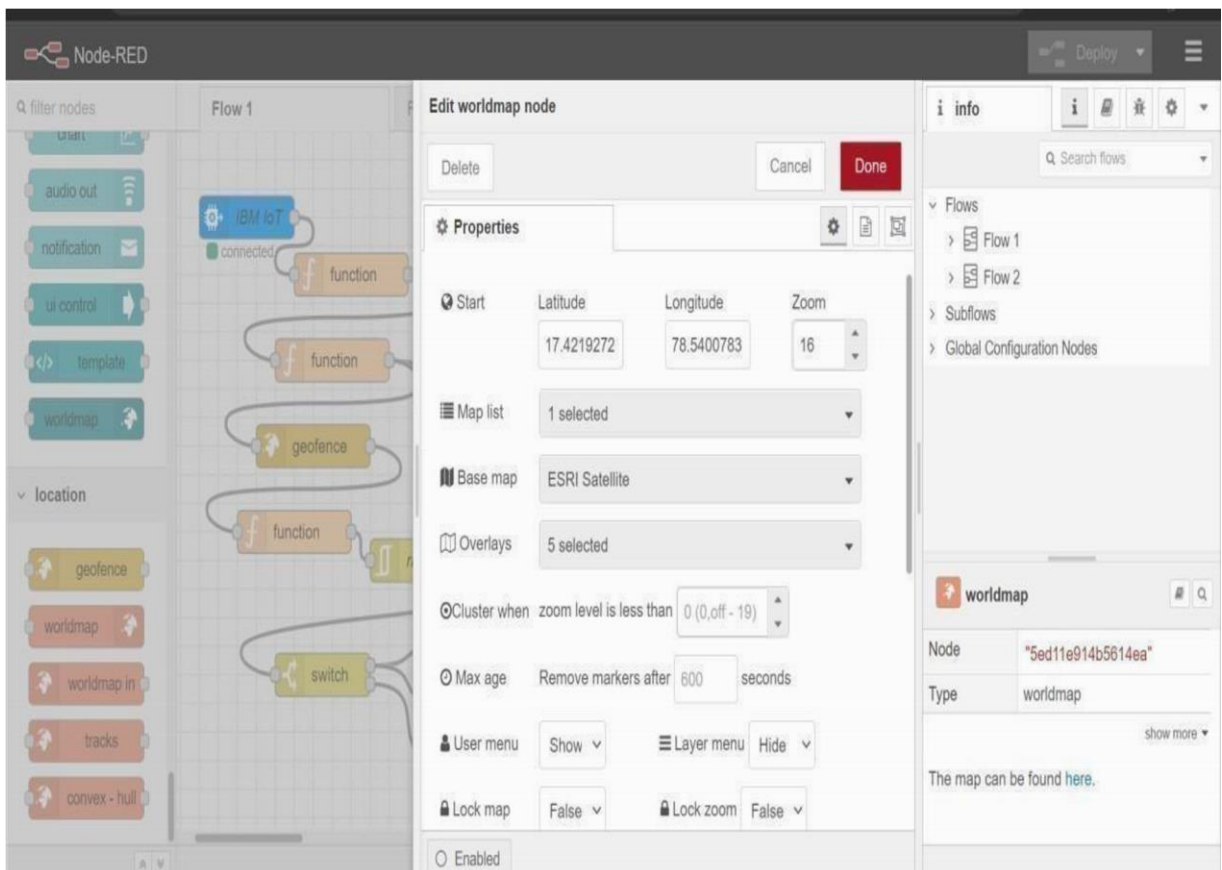
```
1 msg.payload={
2   'name':global.get('name'),
3   'lat':global.get('latitude'),
4   'lon': global.get('longitude')
5 }
6 return msg;
```

The 'On Message' tab is selected. The right sidebar shows the 'info' panel with a search bar and a list of flows (Flow 1, Flow 2, Subflows, Global Configuration Nodes). Below this, a node information panel shows the node ID '77b707bf262aa6f4' and its type 'function'.

5: Geofencing mode



Step 6: World map node with properties



7: Apply the code with different node

The screenshot shows the Node-RED web interface. On the left, a sidebar contains a 'filter nodes' search bar and a list of nodes categorized by type (e.g., 'location' nodes like geofence, worldmap, tracks). The main workspace displays 'Flow 1' with a sequence of nodes: IBM IoT (connected), a function node, and another function node. The 'Edit function node' dialog is open, showing the 'Properties' tab. The 'Name' field is empty. The 'On Message' tab is selected, and the code editor contains the following JavaScript code:

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

The right sidebar shows the 'info' panel with a search bar and a list of flows (Flow 1, Flow 2). Below this, the 'function' node is highlighted, showing its ID '3585a4ac8098085f' and type 'function'.

Step 8: Code with different node

The screenshot shows the Node-RED web interface. The main workspace displays 'Flow 1' with a sequence of nodes: IBM IoT (connected), a function node, a geofence node, another function node, and a switch node. The 'Edit switch node' dialog is open, showing the 'Properties' tab. The 'Name' field is empty. The 'Property' dropdown is set to 'msg.payload'. The 'is false' rule is selected, and the output is set to '1'. The 'is true' rule is also visible, with the output set to '2'. The 'checking all rules' checkbox is checked. The right sidebar shows the 'info' panel with a search bar and a list of flows (Flow 1, Flow 2). Below this, the 'switch' node is highlighted, showing its ID '16b63596382b4cc6' and type 'switch'.

9: Code with different node

The screenshot shows the Node-RED web interface. On the left, a palette of nodes is visible, including 'filter nodes', 'location', and 'worldmap'. The main workspace shows a flow with an 'IBM IoT' node connected to a 'function' node. The 'Edit function node' dialog is open, displaying the following JavaScript code:

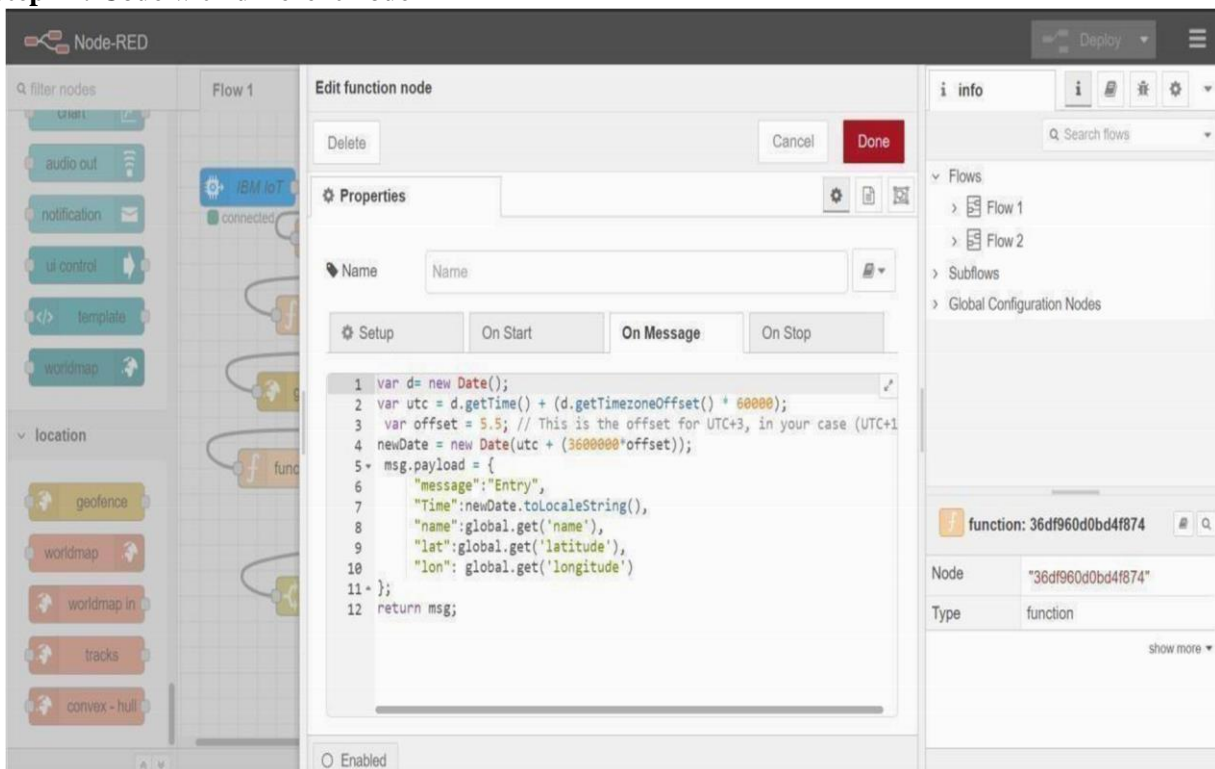
```
1 var d= new Date();
2 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
3 var offset = 5.5; // This is the offset for UTC+3, in your case (UTC+1
4 newDate = new Date(utc + (3600000*offset));
5 msg.payload = {
6   "message": "Entry",
7   "Time": newDate.toLocaleString(),
8   "name": global.get('name'),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

The right sidebar shows the 'info' tab with a search bar and a list of flows. Below this, the node details are shown for a function node with ID '322663c3205cc798'.

Step 10: code with different node

This screenshot is identical to the previous one, showing the same Node-RED interface and JavaScript code in the 'Edit function node' dialog. The only difference is in the right sidebar, where the node details are shown for a function node with ID 'dfe78abe1ae2fd4'.

Step 11: Code with different node



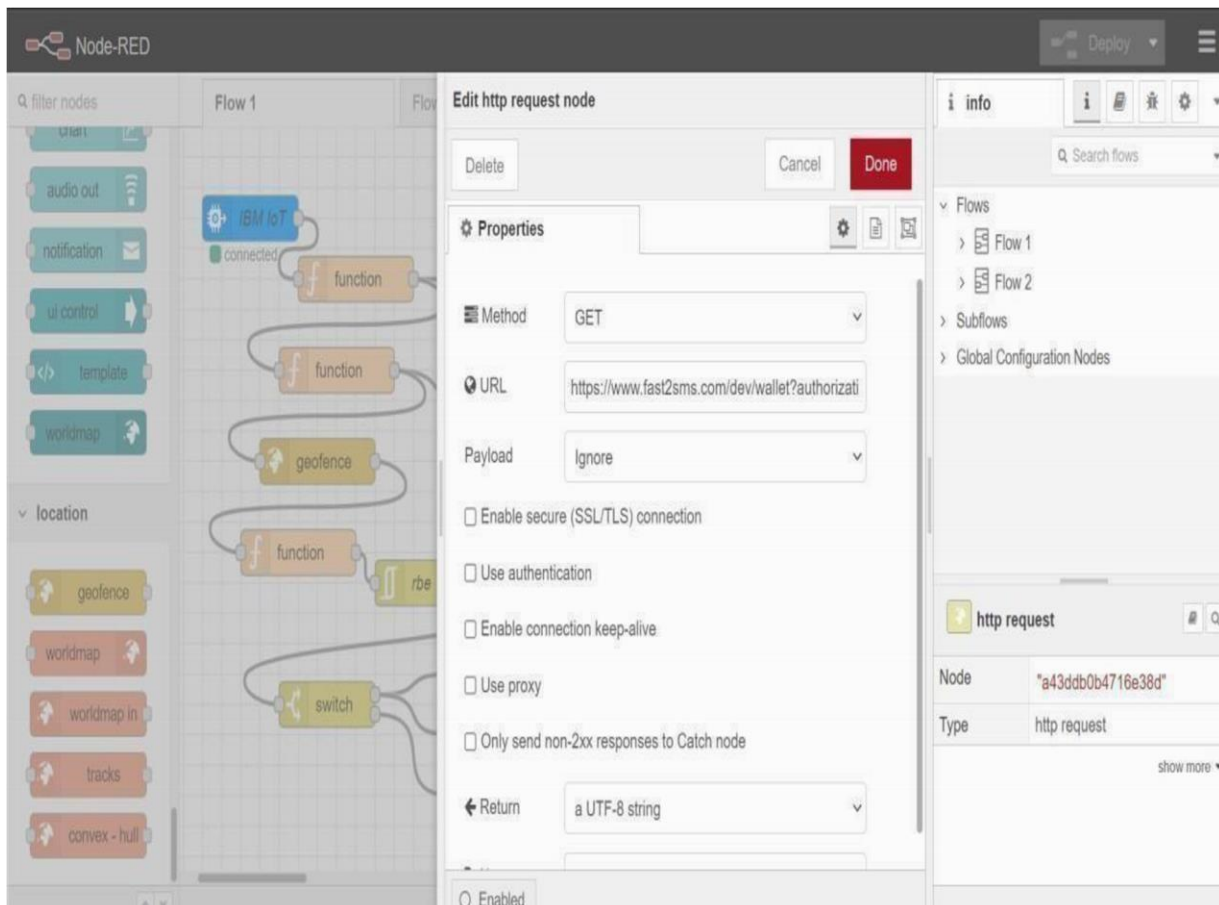
The screenshot shows the Node-RED interface with the 'Edit function node' dialog open. The 'On Message' tab is selected, and the following JavaScript code is entered:

```
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 the following details:

- Node: "36df960d0bd4f874"
- Type: function

Step 12: Http request node->properties



The screenshot shows the Node-RED interface with the 'Edit http request node' dialog open. The 'Properties' tab is selected, and the following properties are configured:

- 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 the following details:

- Node: "a43ddb0b4716e38d"
- Type: http request

Step 13: Notification node

Node-RED interface showing the 'Edit notification node' dialog. The dialog is titled 'Edit notification node' and has buttons for 'Delete', 'Cancel', and 'Done'. It contains a 'Properties' section with fields for 'Layout' (set to 'OK / Cancel Dialog'), 'Send to all browser sessions.' (unchecked), 'Default action label' (set to 'OK'), 'Secondary action label' (optional label for Cancel button), and 'Accept raw HTML/JavaScript input in msg.payload to format popup.' (unchecked). There are also fields for 'Class' (set to '[msg.className]'), 'Topic' (set to '[msg.topic]'), and 'Name' (set to 'show dialogue'). The 'Enabled' checkbox is checked. On the right, the 'info' panel shows the node's details: 'Node' is '388bcb2648d1348b' and 'Type' is 'ui_toast'.

Step 14: Cloudant node

Node-RED interface showing the 'Edit cloudant out node' dialog. The dialog is titled 'Edit cloudant out node' and has buttons for 'Delete', 'Cancel', and 'Done'. It contains a 'Properties' section with fields for 'Service' (set to 'Cloudant-pn-99850'), 'Database' (set to 'noderedapqzh20221108'), 'Operation' (set to 'insert'), and 'Only store msg.payload object?' (unchecked). There is also a field for 'Name' (set to 'child'). The 'Enabled' checkbox is checked. On the right, the 'info' panel shows the node's details: 'Node' is '7f199a311bbfac36' and 'Type' is 'cloudant out'.