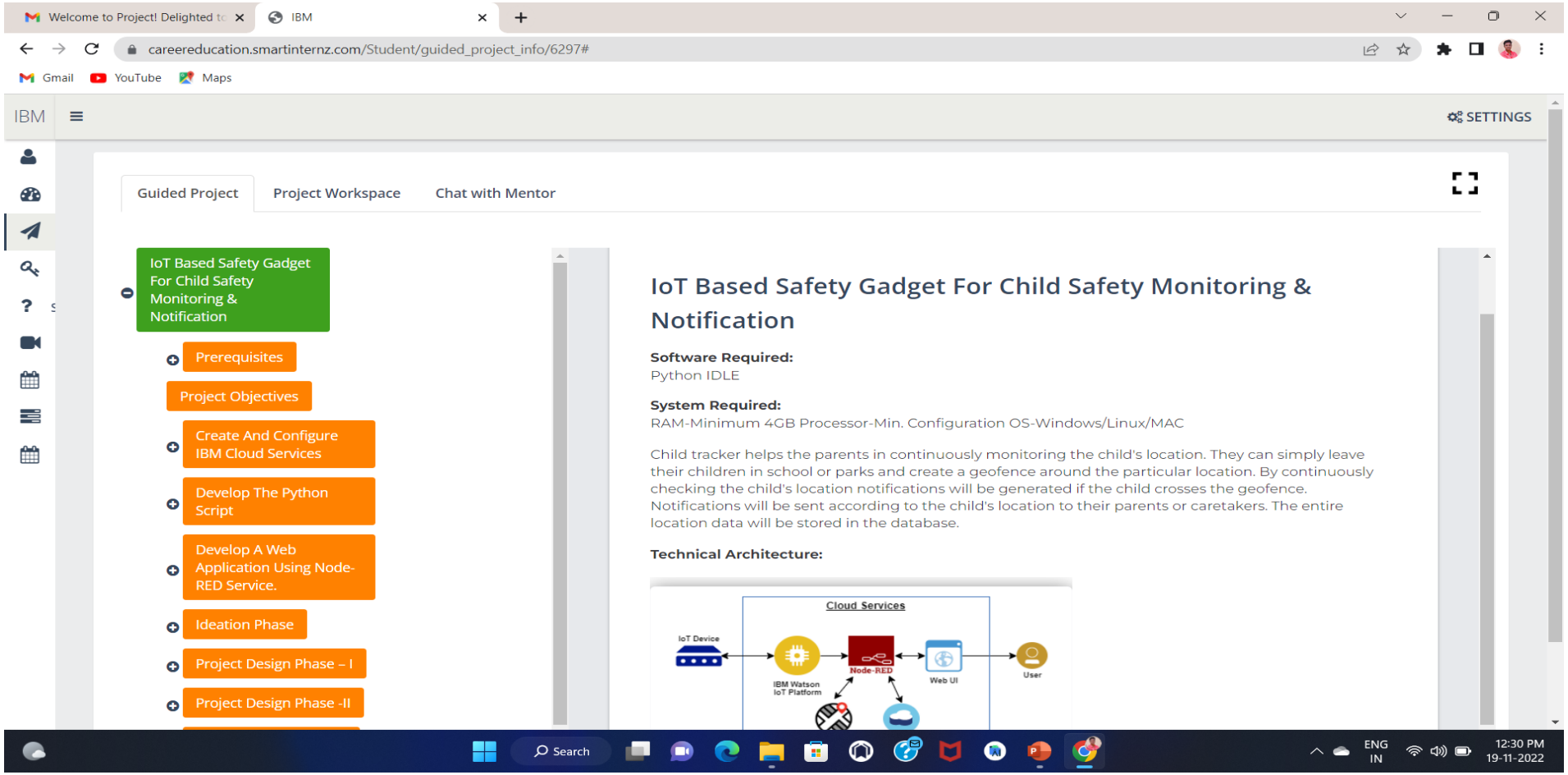
TEAM ID : PNT2022TMID11121

PROJECT NAME : IOT BASE CHILD SAFETY MONITORING AND NOTIFICATION

TEAM MEMBERS :

- VIGNESHWER.S
- SUJEEVE.A
- SUBASH RAM.S
- VASIKARAN.J

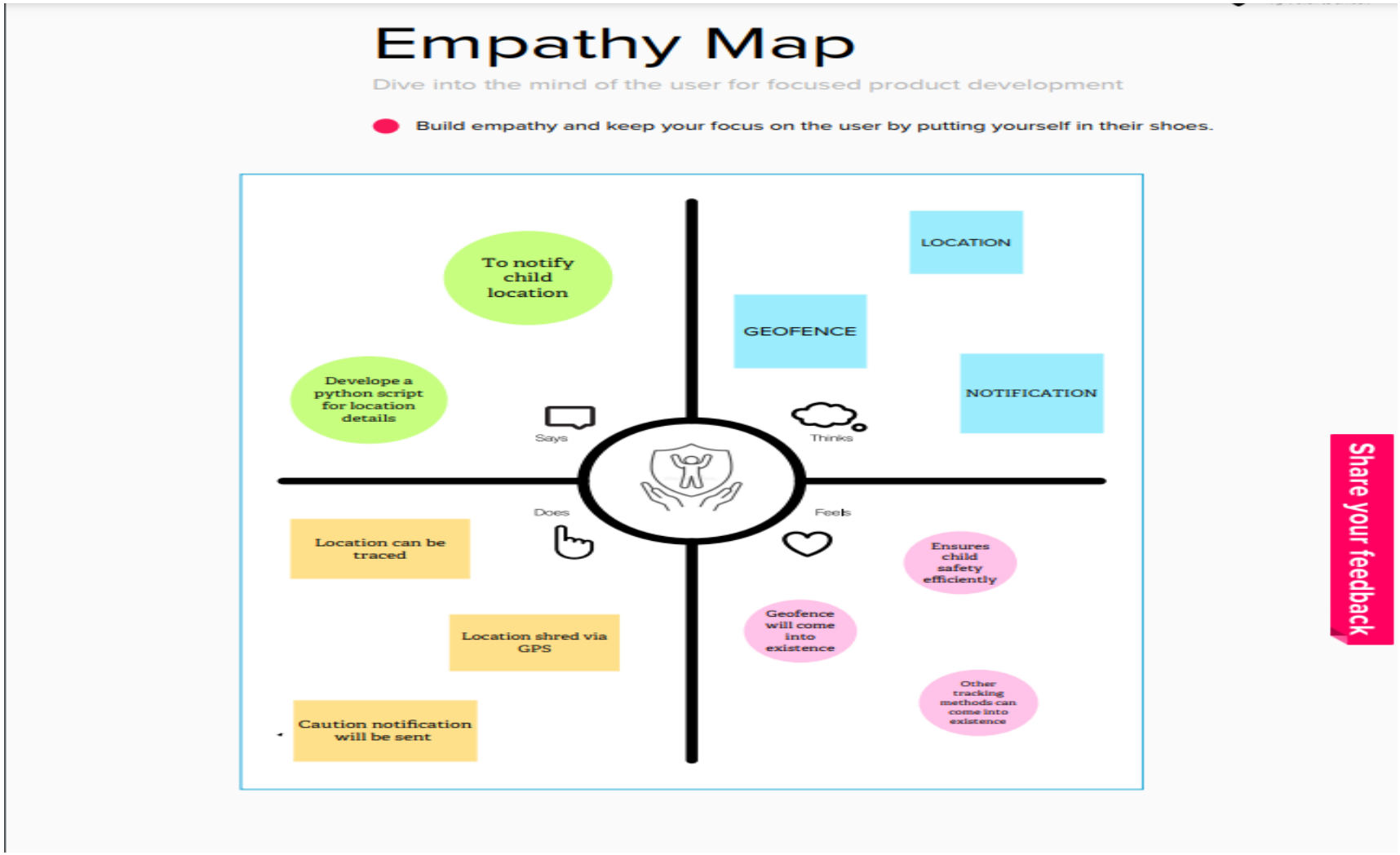
IBM NALAIYATHIRAN PROFILE:



IDEATION PHASE:

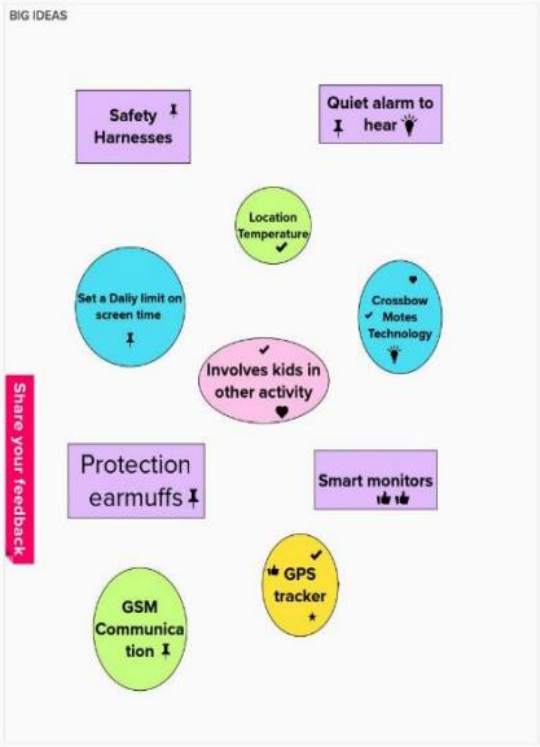
- EMPATHY MAP
- IDEATION OF PROJECT
- LITERATURE SURVEY

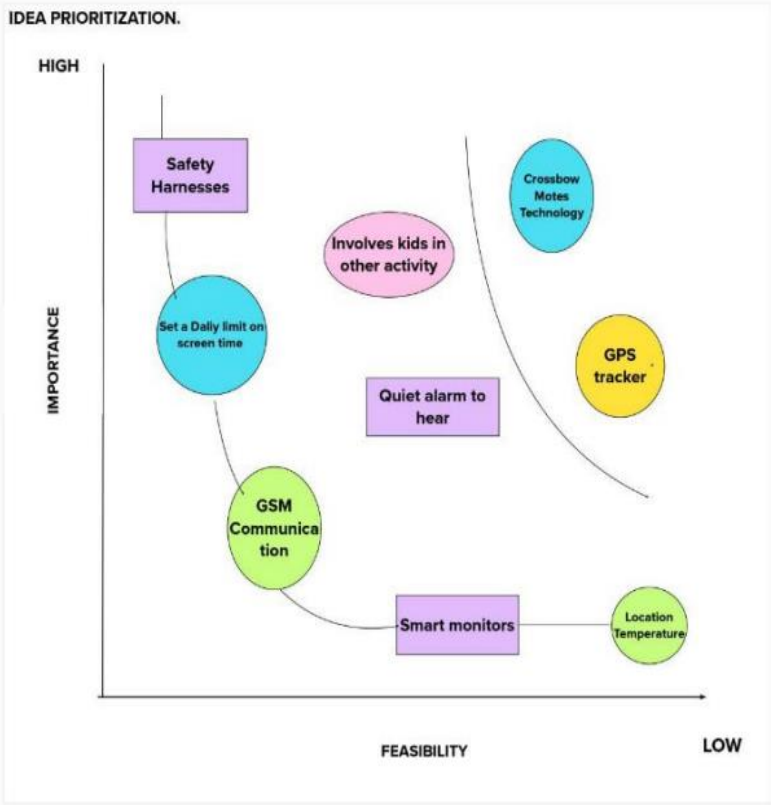
EMPATHY MAP


Empathy Map
Dive into the mind of the user for focused product development
Build empathy and keep your focus on the user by putting yourself in their shoes.
To notify child location
Develope a python script for location details
Says
Thinks
LOCATION
GEOFENCE
NOTIFICATION
Does
Feels
Location can be traced
Location shred via GPS
Caution notification will be sent
Ensures child safety efficiently
Geofence will come into existence
Other tracking methods can come into existence
Share your feedback

IDEATION


Empathy Map
Dive into the mind of the user for focused product development
Build empathy and keep your focus on the user by putting yourself in their shoes.
Carpal Tunnel Syndrome
Physical Fatigue
Adequate Immunity
Enhance Communication
Boost Collaboration
Educate Students
Child Breath monitoring
Informative and secure
Intuitive
Early Intervention
DropCam Video monitoring
Child tracker using metal ornaments
Need Statement
CHILD SAFETY
24 Hours Protection
Highly secured
Maintain child's physical mental and physiological integrity . Protection against infection and sudden catastrophe .


BIG IDEAS
Safety Harnesses
Quiet alarm to hear
Location Temperature
Set a Daily limit on screen time
Crossbow Motes Technology
Involves kids in other activity
Share your feedback
Protection earmuffs
Smart monitors
GPS tracker
GSM Communica tion


IDEA PRIORITIZATION.
HIGH
IMPORTANCE
Safety Harnesses
Crossbow Motes Technology
Involves kids in other activity
Set a Daily limit on screen time
GPS tracker
Quiet alarm to hear
GSM Communica tion
Smart monitors
Location Temperature
FEASIBILITY
LOW

LITERATURE SURVEY

- Literature survey of our project enclose the references of various IEEE papers related to our topics that was published in various conferences.
- We have taken ten papers and drive the literature survey to our project.

PROJECT DESIGN PHASE-I

- PROPOSED SOLUTION
- SOLUTION FIT
- SOLUTION ARCHITECTURE

PROPOSED SOLUTION

Project Design Phase-I
Proposed Solution Template

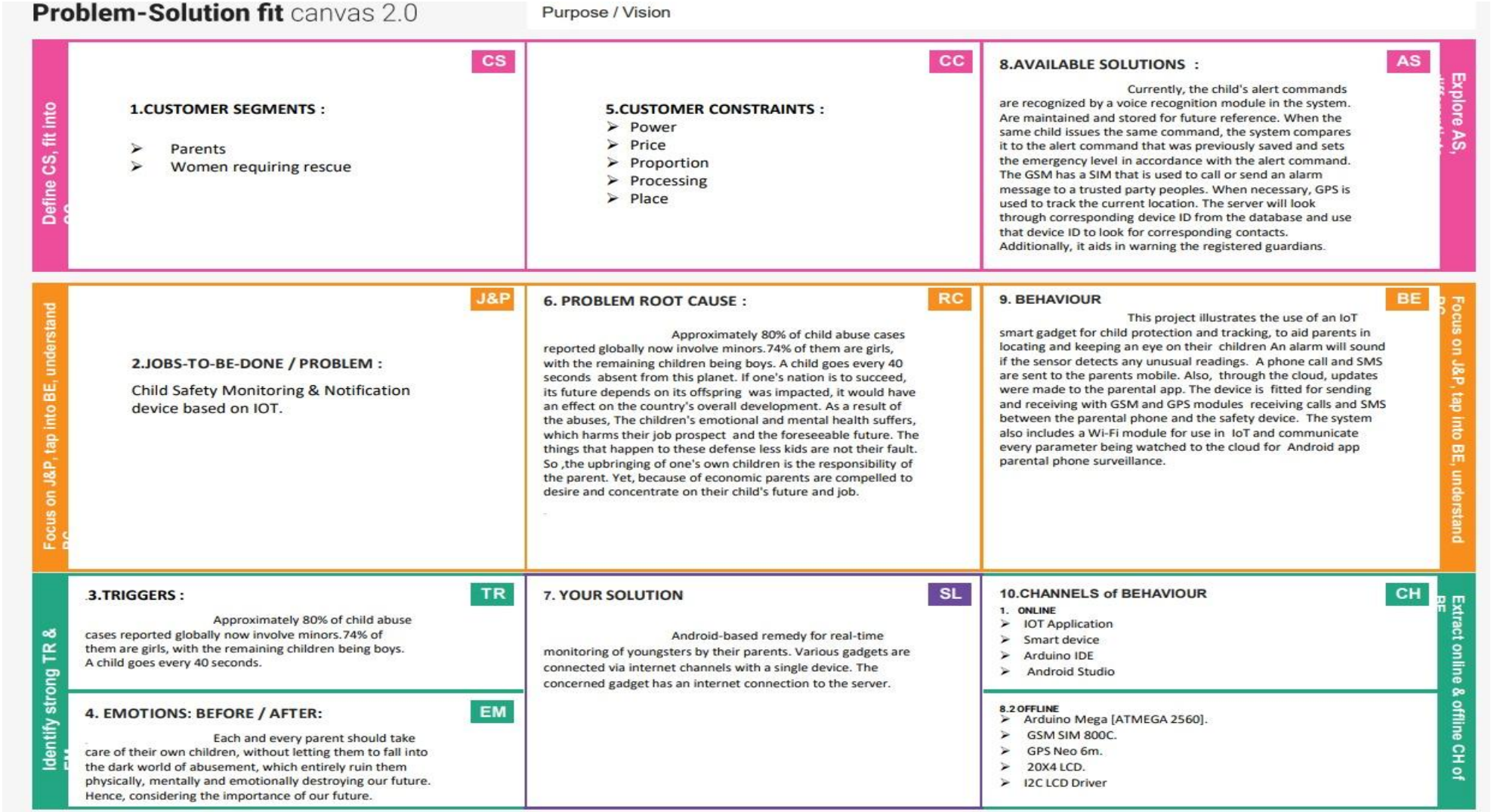
Date	19 September 2022
Team ID	PNT2022TMID11121
Project Name	IOT Based device for child safety Monitoring and Notification
Maximum Marks	2 Marks

Proposed Solution Template:

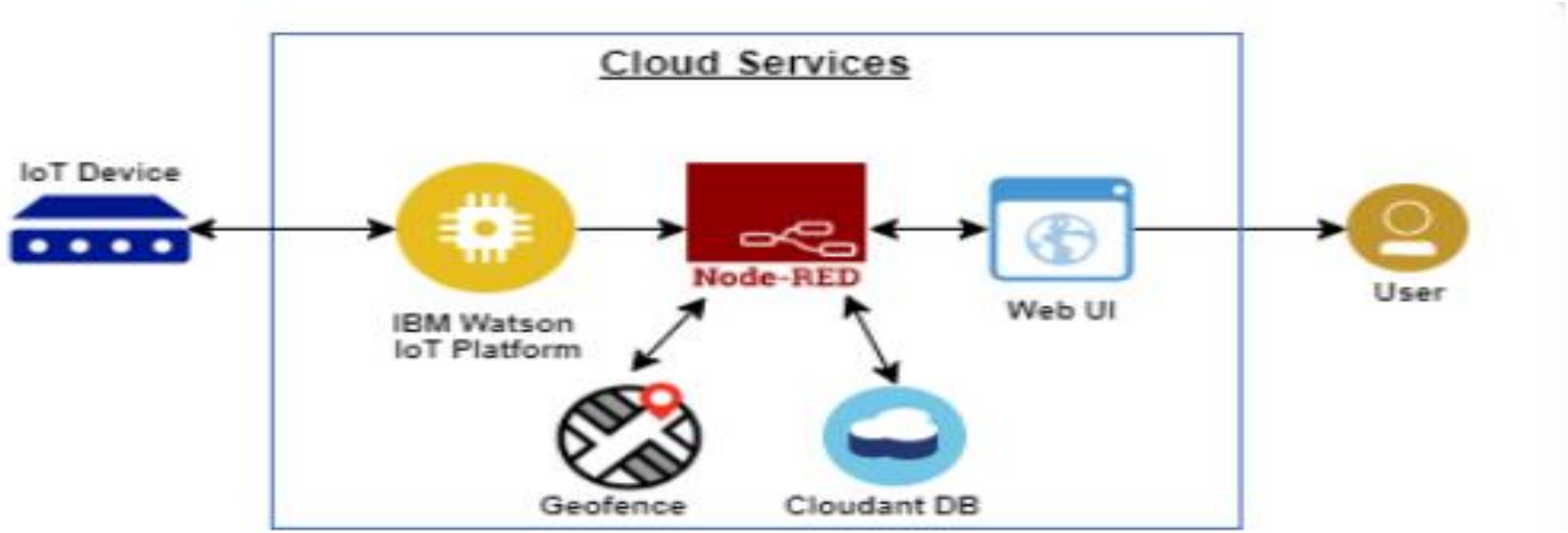
Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Child Safety Monitoring & Notification device based on IOT.
2.	Idea / Solution description	Android-based remedy for real-time monitoring of youngsters by their parents. Various gadgets are connected via internet channels with a single device. The concerned gadget has an internet connection to the server.
3.	Novelty / Uniqueness	child safety measures that contain two major gadgets: BLE and smart devices. Listener apparatus The system also has an Android application. Specifically, the Parental app that will be created and deployed on family phone.
4.	Social Impact / Customer Satisfaction	The parents can use a tool to track their kids in real time or to protect women. The suggested remedy moves the problem location offerings from the GSM module. It permits the parents to receive the whereabouts of their child through SMS.
5.	Business Model (Revenue Model)	➢ Live Location Tracking ➢ Panic Alert Systems ➢ Stay Connected Feature ➢ Health Monitoring System
6.	Scalability of the Solution	Small solar panels can be installed to change the system. For maximising the power of a smart device's battery and backup batteries.

SOLUTION FIT



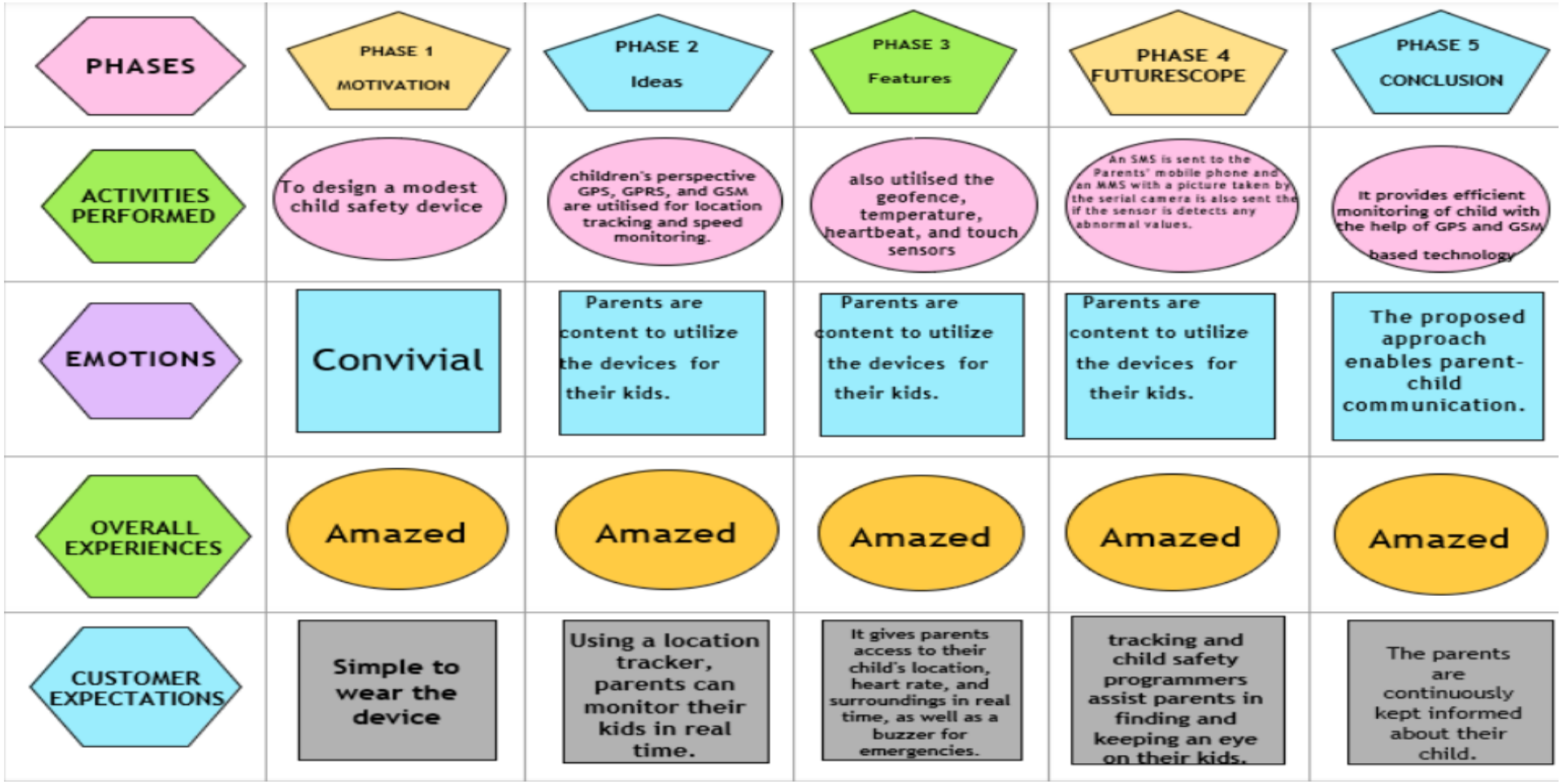
SOLUTION ARCHITECTURE



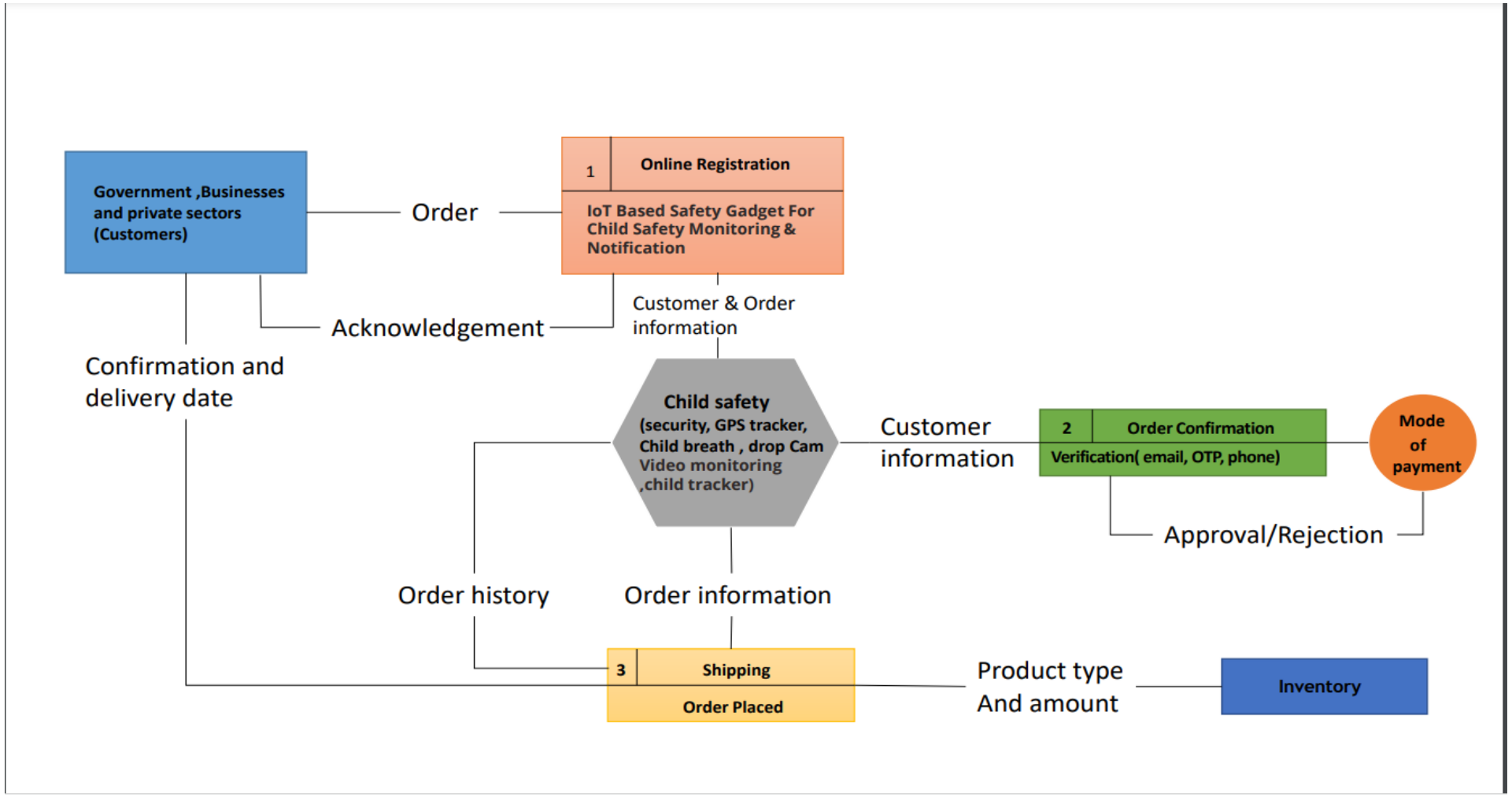
PROJECT DESIGN PHASE II

- CUSTOMER JOURNEY
- DATA FLOW DIAGRAM
- FUNCTIONAL REQUIREMENT
- TECHNOLOGY ARCHITECTURE

CUSTOMER JOURNEY



DATA FLOW DIAGRAM



FUNCTIONAL REQUIREMENT

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	11 October 2022
Team ID	PNT2022TMID11121
Project Name	IOT BASED DEVICE FOR CHILD SAFETY MONITORING AND NOTIFICATION
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

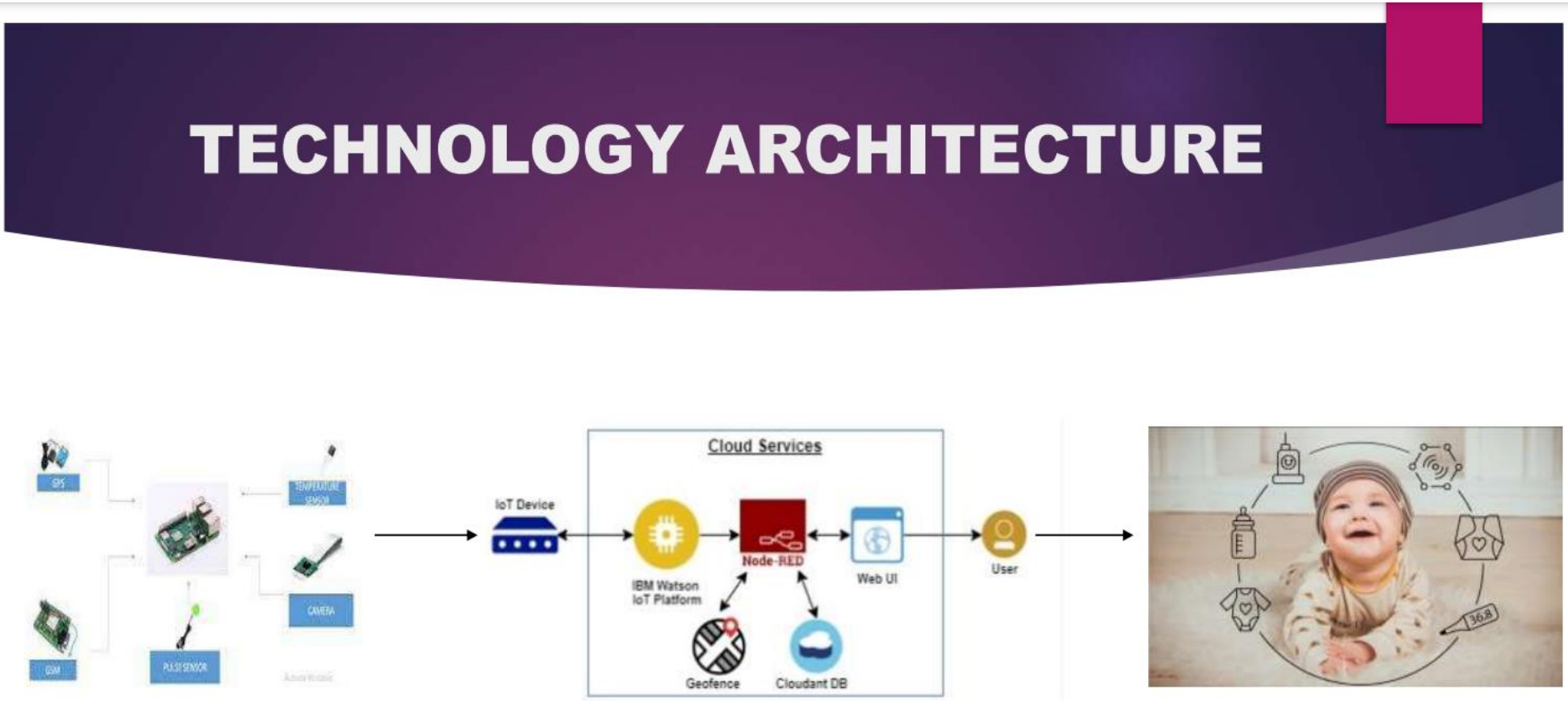
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirements	A smart device will be given to the parents/guardian in order to ensure the safety of the children.
FR-2	User Registration	Manual Registration Through a Website or Gmail
FR-3	User Confirmation	Phone Confirmation Email confirmation OTP authentication
FR-4	Payments options	No payment required
FR-5	Product Delivery and installation	The installation fee will be determined with respect to the circumstances of the children and the parent.
FR-6	Product Feedback	Through a website via Gmail

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have clear product instructions and a self-explanatory product that is simple to use.
NFR-2	**Security**	Cloud data must be contained within the network, collapsing to be avoided, Real-time avoidance should be avoided, and the device will be constantly monitored.
NFR-3	**Reliability**	Hardware is frequently tested.
NFR-4	**Performance**	The smart device will provide a better user experience and deliver accuracy output.
NFR-5	**Availability**	All of the functions that the user demands will be provided, depending on the needs of the consumer.
NFR-6	**Scalability**	The product is based on child safety so it must ensures all types of child safety parameters are true.

TECHNOLOGY ARCHITECTURE



PROJECT PLANNING PHASE

- MILESTONE
- SPRINT DELIVERY PLAN

MILESTONE

MILESTONE LIST AND ACTIVITY LIST

Date	09th NOVEMBER 2022
Team ID	PNT2022TMID11121
Project Name	IOT based safety Gadget for child safety monitoring and notification
Maximum Marks	2 Marks

Milestone Name	Activities	Milestone Number	Description	Completion Date	Status
Prerequisites			Create the IBM account and download the necessary software for your chosen category of the project	28/08.2022	Completed
Ideation Phase	Literature Survey	1	Literature survey on the selected project by gathering and referring research paper and publications	17/09/2022	Completed
	Empathy Map	1	Create an empathy map that list the user's pains and gains	16/10/2022	Completed
Project Design Phase -1	Solution Architecture	2	Prepare Solution architecture diagram for the proposed solution	01/10/2022	Completed
	Problem Solution Fit	2	Prepare Solution Fit Document for the proposed solution	01/10/2022	Completed
Project Design Phase -2	Customer Journey Map	3	Prepare a customer journey map to understand how the user interact and experience your product	08/10/2022	Completed

	Data Flow Diagram	3	Draw the data flow diagram for you proposed solution	16/10/2022	Completed

	Solution Requirements	3	Create a solution requirement document for the proposed solution	11/10/2022	Completed
	Technology Stack	3	Prepare the technology stack diagram for the proposed solution	16/10/2022	Completed
Project Planning	Milestone And Activity List	4	Create a document to show your milestones as well as activity in your development cycle	08/11/2022	Completed
	Sprint Delivery Plan	4	Create a sprint plan for the project	07/11/2022	Completed
Project Developmen t Phase	Sprint-1	5	Delivery of the sprint-1	09/11/2022	On Going
	Sprint-2	6	Delivery of the sprint-2	09/11/2022	On Going
	Sprint-3	7	Delivery of the sprint-3	14/11/2022	On Going
	Sprint-4	8	Delivery of the sprint-4	19/11/2022	On Going

SPRINT DELIVERY PLAN

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	20 October 2022
Team ID	PNT2022TMID11121
Project Name	Project- IoT Based Safety Gadget for Child Safety Monitoring & Notification
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	Registration through websiteRegistration through app	2	High	VIGNESHWER S
Sprint-1	User Confirmation	USN-2	Confirmation via EmailConfirmation via OTP	1	High	SUJEEVE A
Sprint-2	User login	USN-3	Setting up User Id and password	2	Low	SUBASH RAM S
Sprint-2	App permission	USN-4	Grant the permission for the app to access location,contact etc..	2	Medium	VASIKARAN J
Sprint-3	Interface with the Device	USN-5	Connecting the device with the registered app with thedevice ID.	1	High	VIGNESHWER SUJEEVE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Setting Geo-location	USN-6	Creating the Geo-location area in the map	2	Low	VASIKARAN J
Sprint-4	Database	USN-7	Location history is stored in the cloud Can be accessed from the dashboard.	2	High	SUBASH RAM S
Sprint-4	Tracking location	USN-8	Tracking the location through app. Tracking the location through website.	2	High	SUJEEVE A

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	05 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	08 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CREATE AND CONFIGURE

- CREATE A CLOUDANT IN DB
- CREATE A IBM WATSON PLATFORM
- CREATE A NODE RED SERVICE

CLOUDANT IN DB

CREATE A DATABASE IN CLOUDANT DB

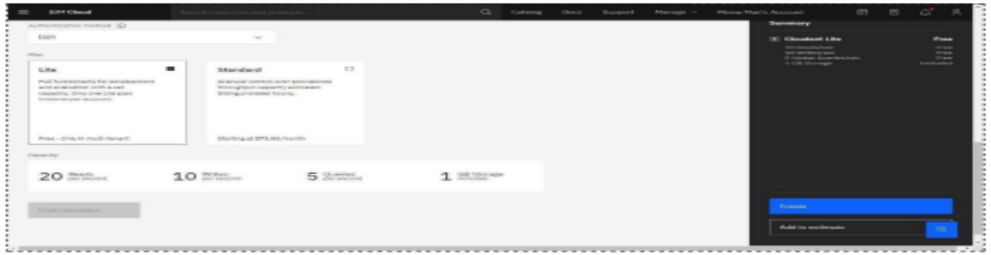
Date	21-10-2022
Team ID	PNT2022TMID11121
Project Name	IOT based safety gadget for child safety monitoring and notification
Maximum Marks	4 Marks
Submitted By	VIGNESHWER.S SUJEEVE.A SUBASH RAM.S VASIKARAN.J

Create A Database In Cloudant DB

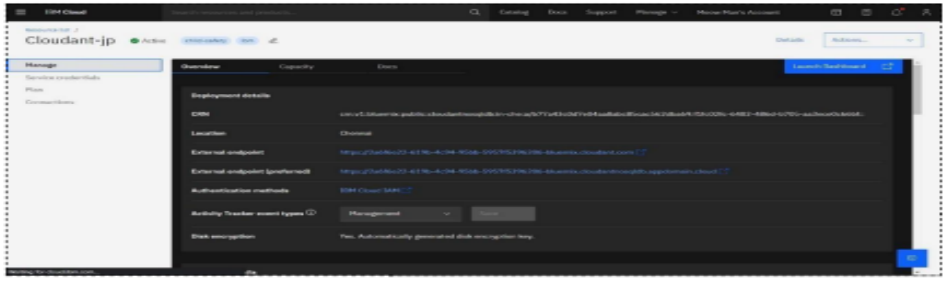
Aim: To create a database in Cloudant DB to store location data.

Steps followed:

- Logged in to IBM Cloud account
- Navigated to `./resources`
- Clicked on the "Create Resource +" button
- Searched for "Cloudant"
- Chose the "Lite Version" and clicked on "Create"



- The Cloudant database resource was created successfully



- Clicked on Launch Dashboard

- Clicked on "Create Database". Entered "meowman" as the database name and the "Non-partitioned" option

- The database “meowman” was created successfully

Result:
A database to store the location data was created successfully on Cloudant DB

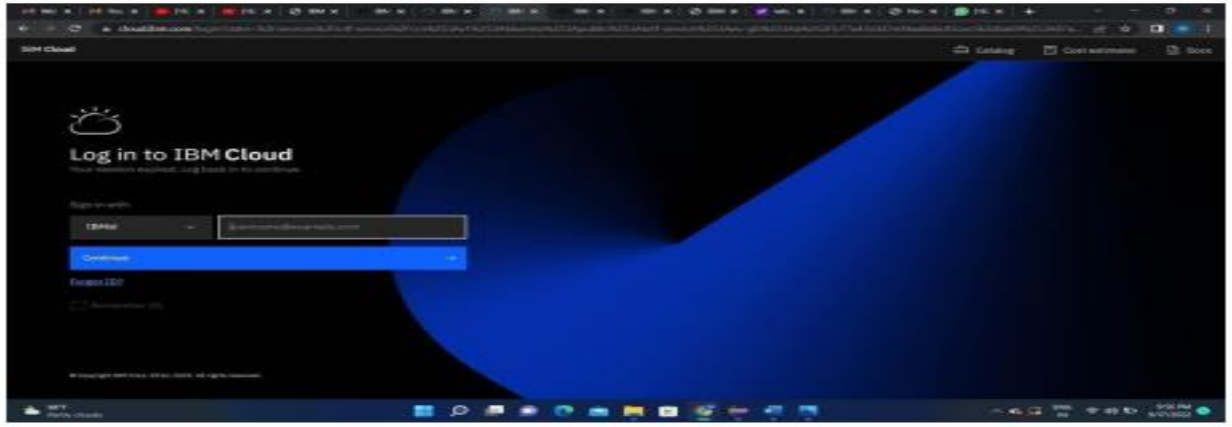
CREATE A IBM WATSON PLATFORM

CREATE IBM WATSON IOT PLATFORM AND DEVICE

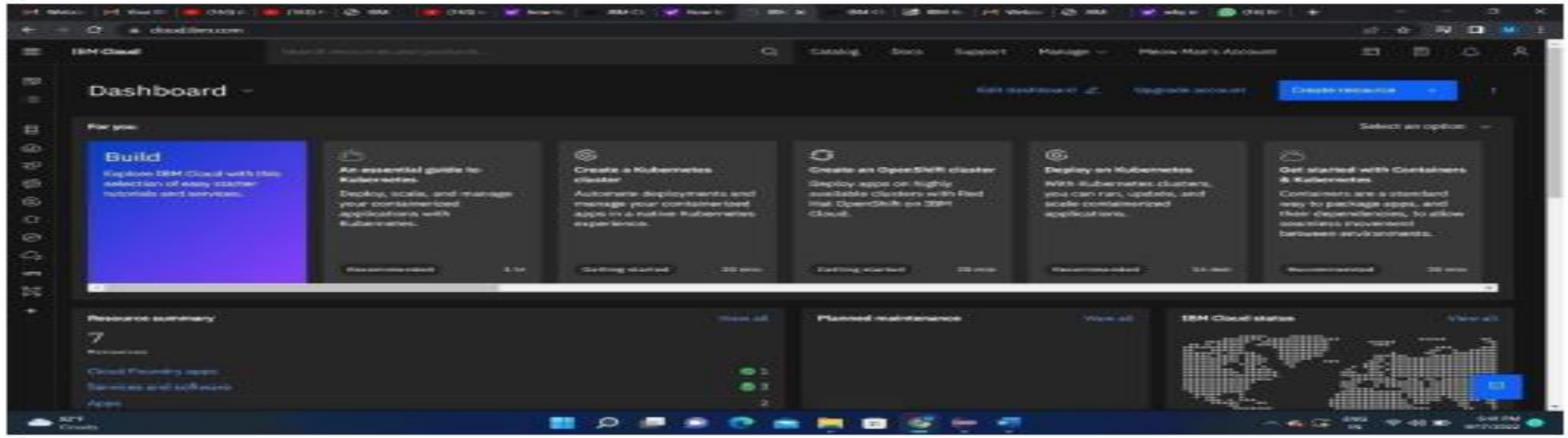
Date	22-10-2022
Team ID	PNT2022TMID11121
Project Name	IOT based safety gadgetfor child safety monitoring and notification
Maximum Marks	4 Marks
Submitted By	VIGNESHWER S SUJEEVE A SUBASHRAM S VASIKARAN J

STEPS:

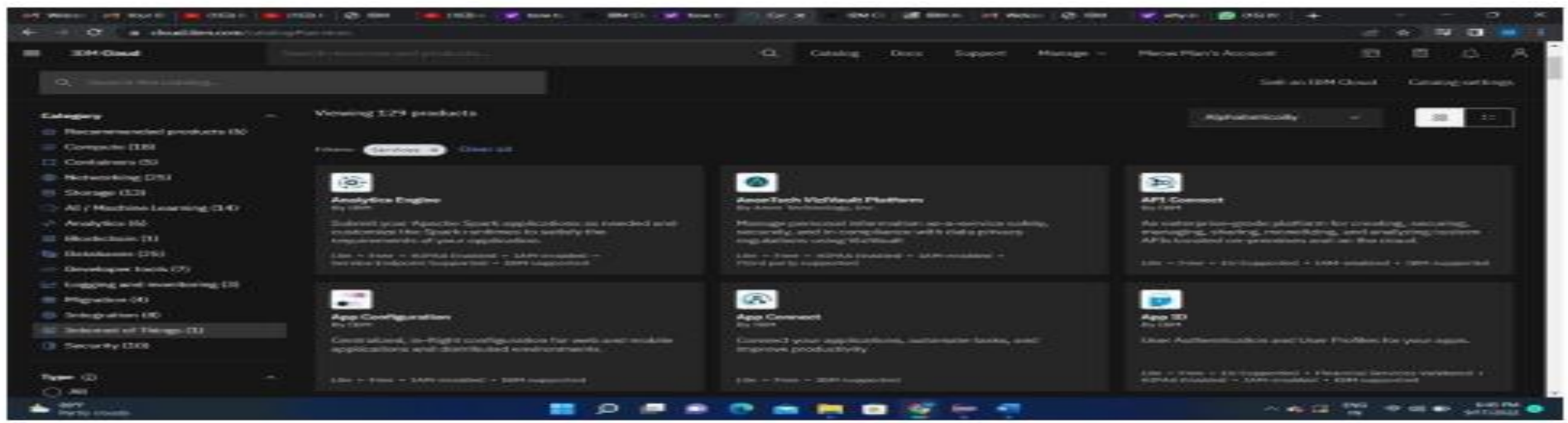
1. Firstly create an IBM cloud account with IBMid and password



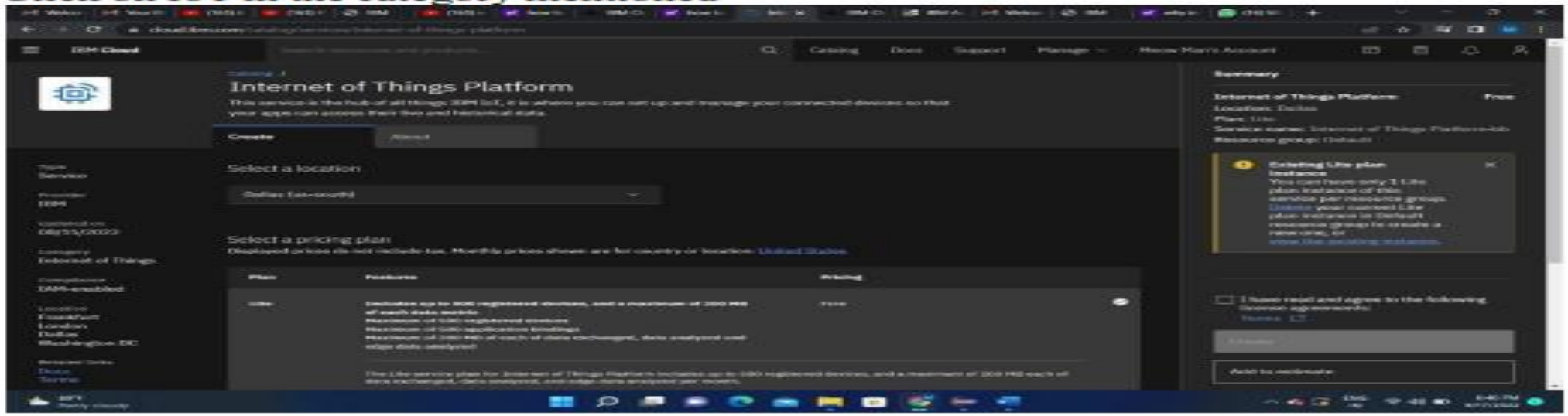
2. Home page of IBM cloud



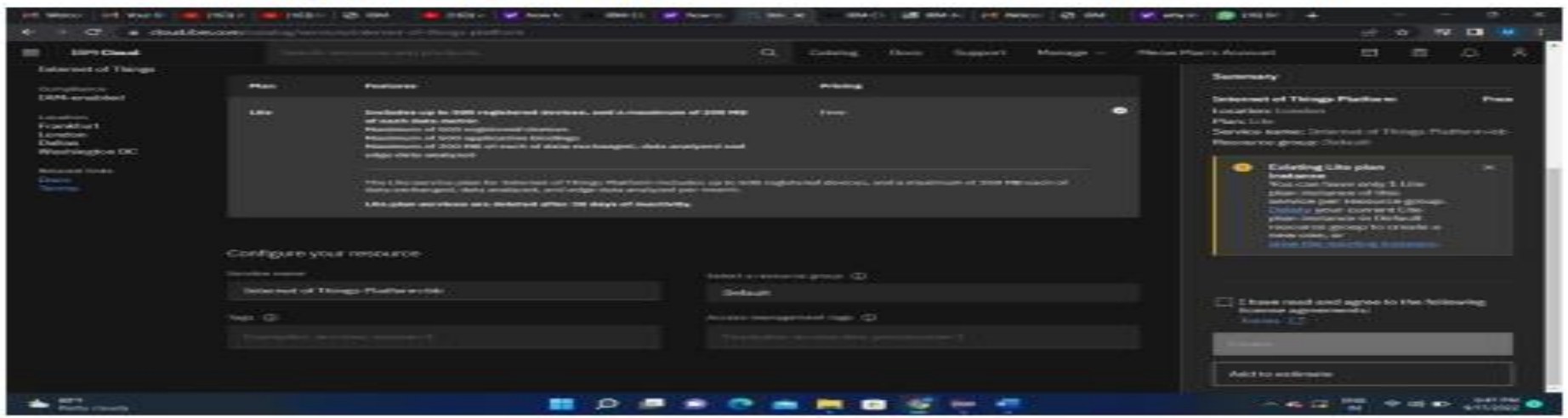
3. Click on the catalog on the top



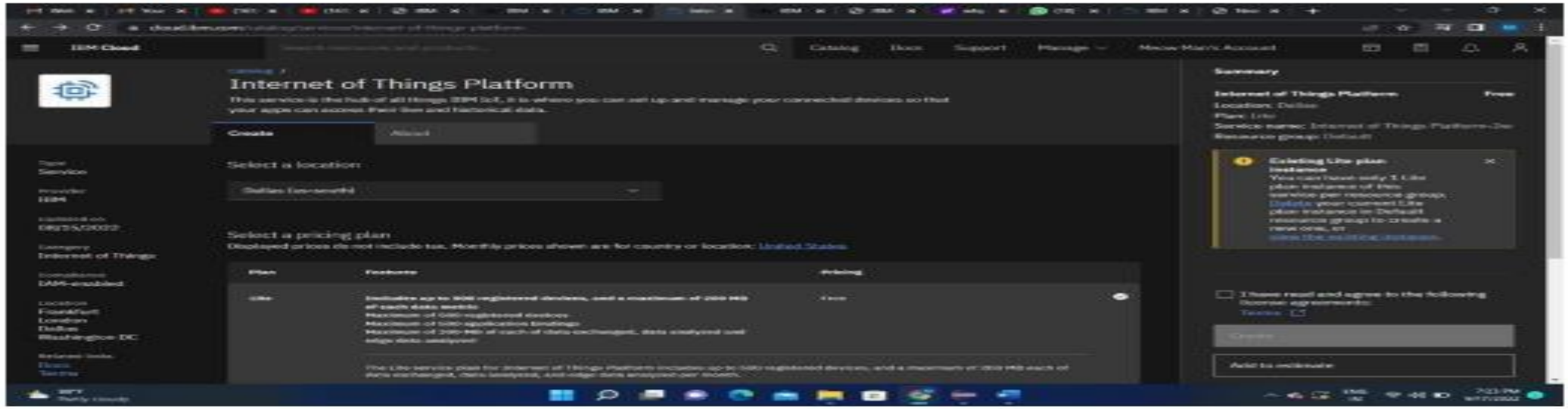
Click on IoT in the category mentioned

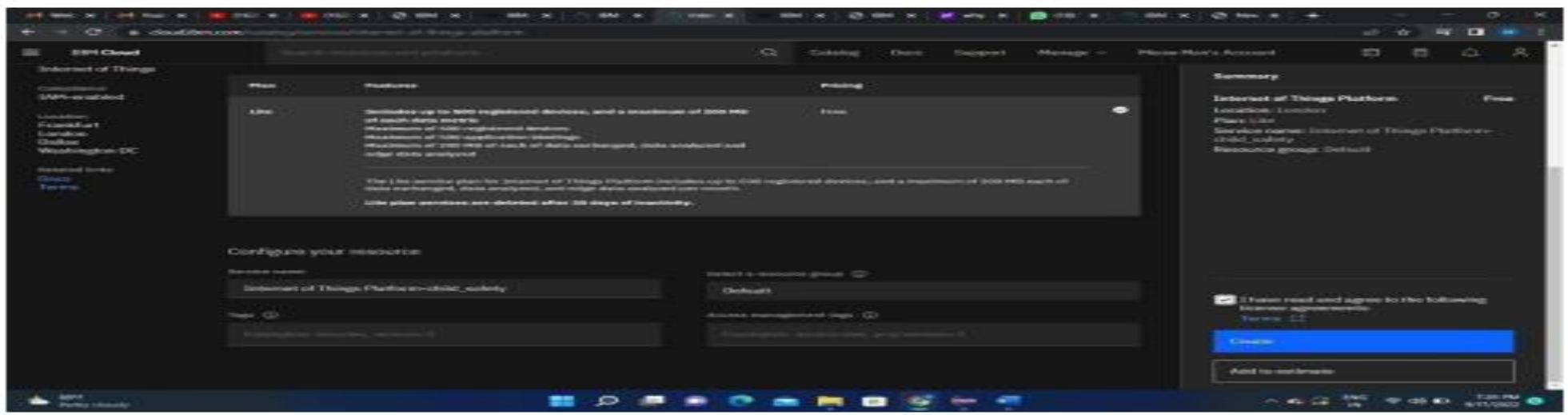


4. If already a lite is present delete it else u can't create another

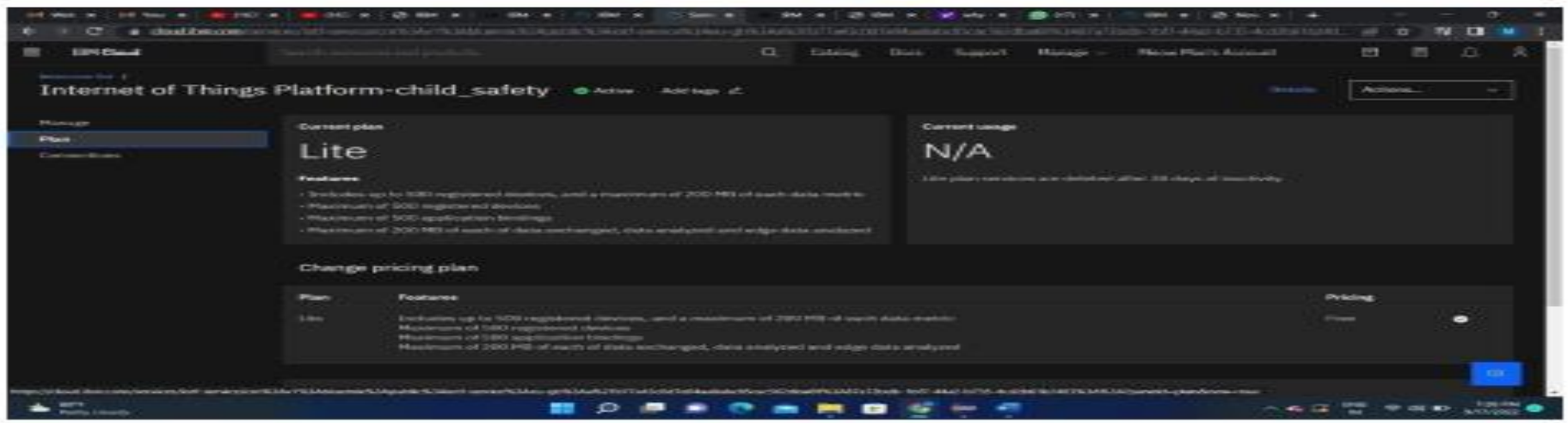


5. Enter the location and in the configure your resource type the service name and choose the plan, tick the agree with agreements and then click on create



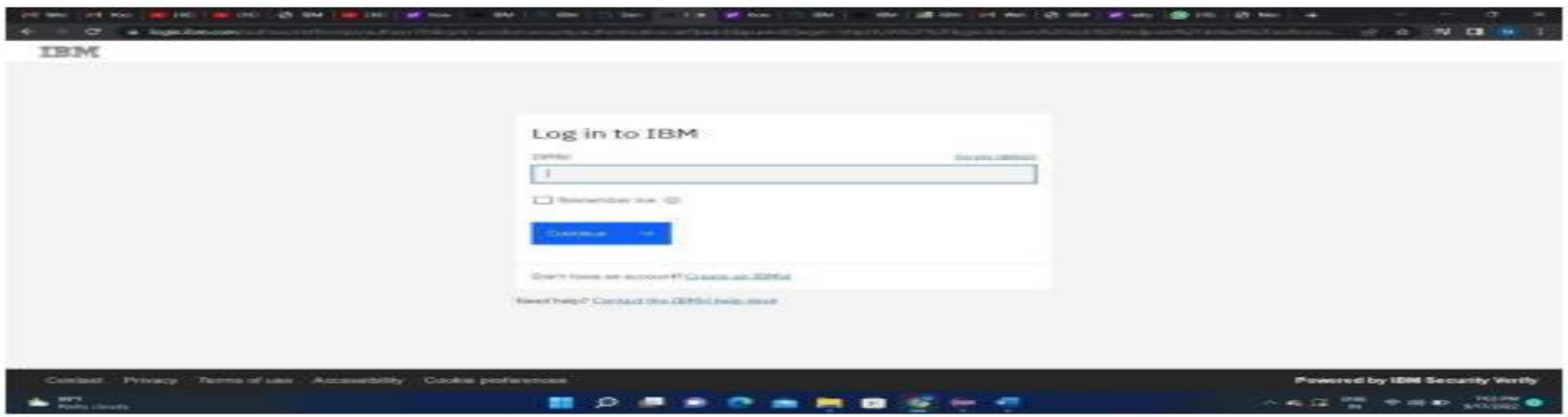


6. Internet of Things Platform Child_safety will be created, where there are different options like manage, plan, and connection (manage is for launch, Plan gives us the idea about the payment package and its upgrades, and lastly the connection is for to connect IoT with other servies)



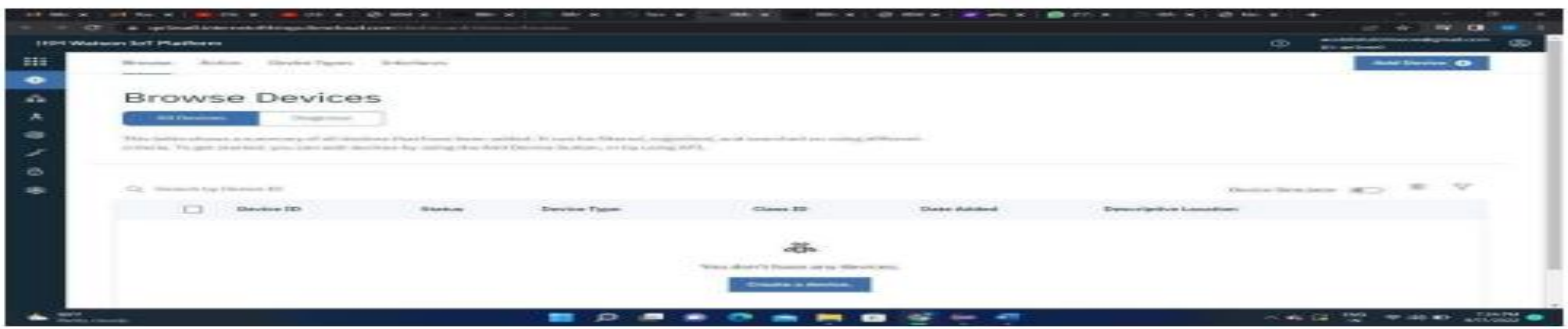
7. Clicking on the launch button in the manage tab, it will open to this

8. Enter the details to sign in to the Watson Cloud to create a device

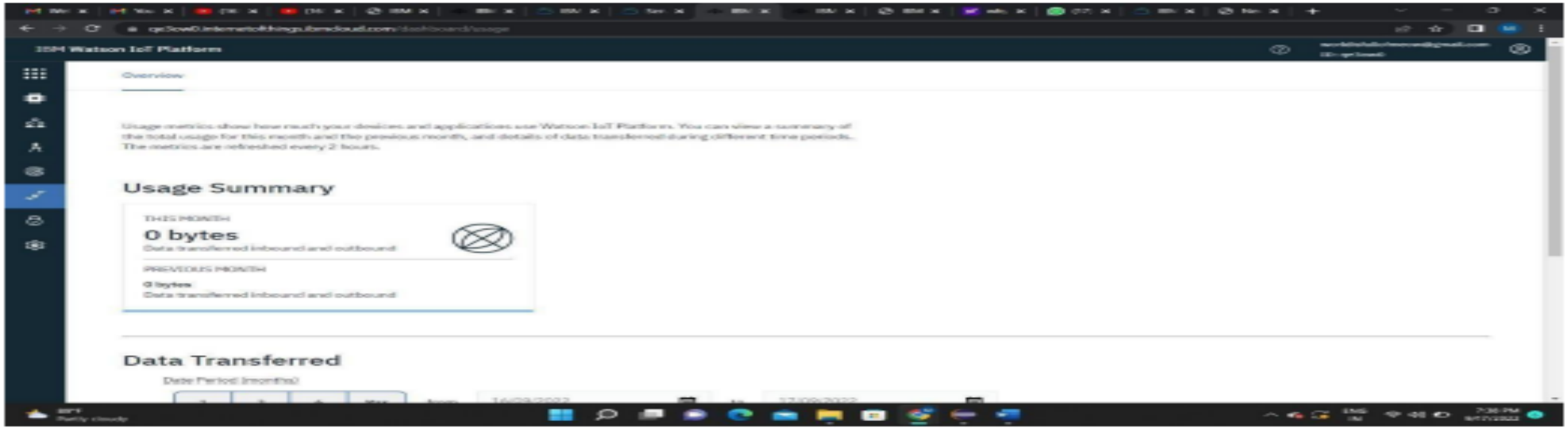


9. Once logged in the name will be displayed and it goes back to the first page

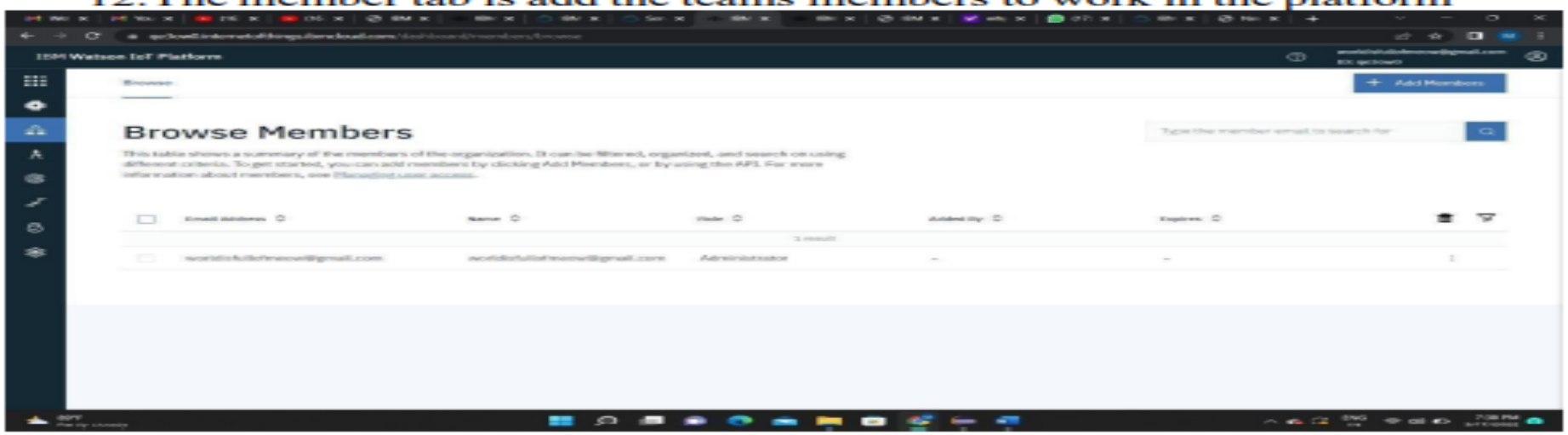
10. And again clicking on the launch button will open this tab, the device will help in the creation of the devices, the addition of devices, and the display ofdetails of the devices

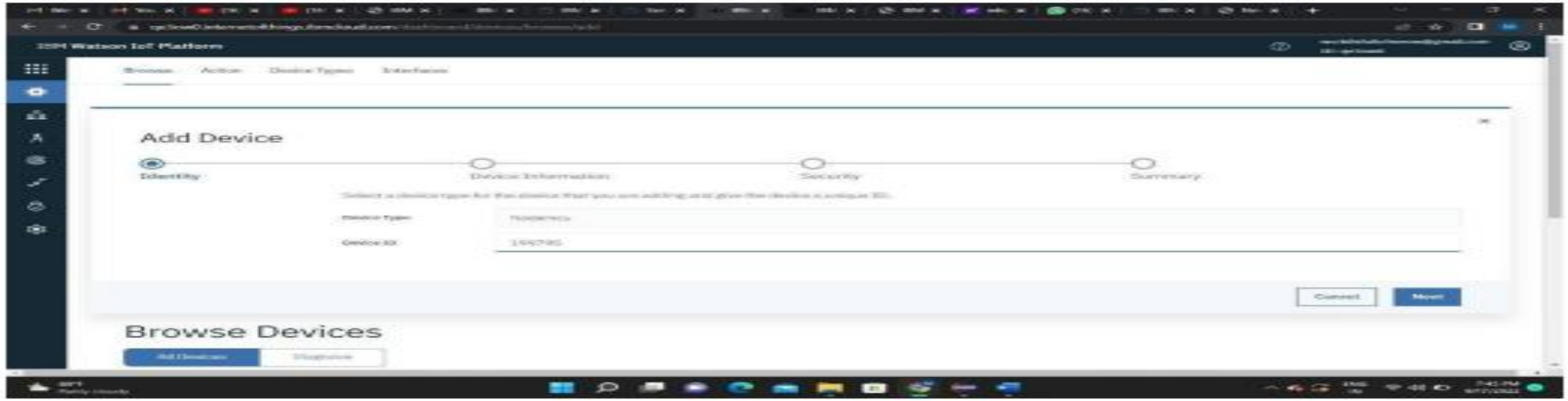


11. Usage gives the summary of how many bytes are used between the devices and the IBM cloud.

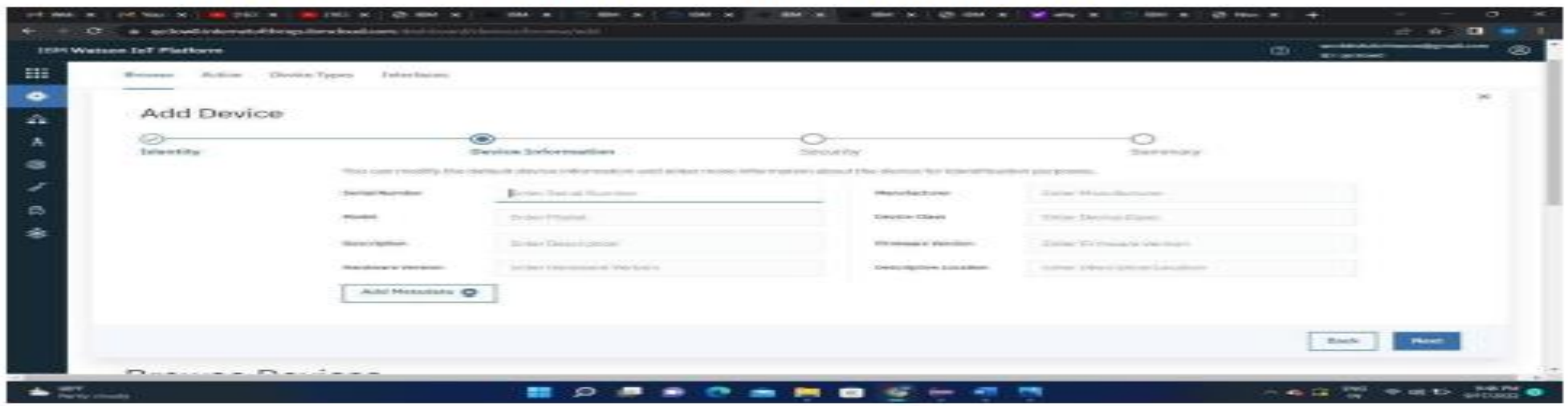


12. The member tab is add the teams members to work in the platform

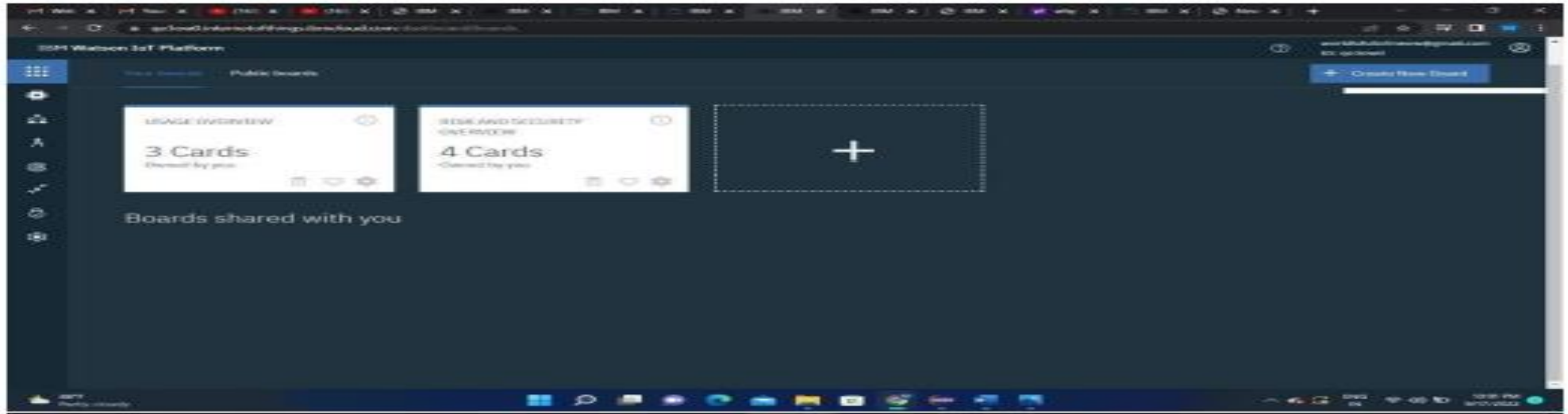


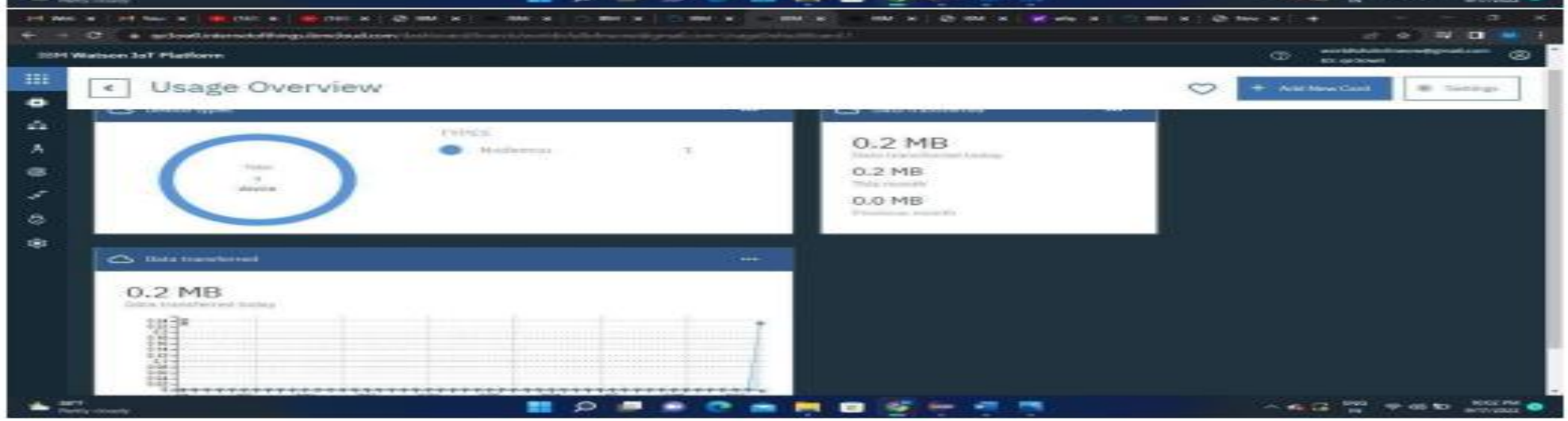


15. This page to enter extra details and of the hardware



21.The Boards will display card for the project.





RESULT:

An IBM Watson cloud for IoT and a device is created

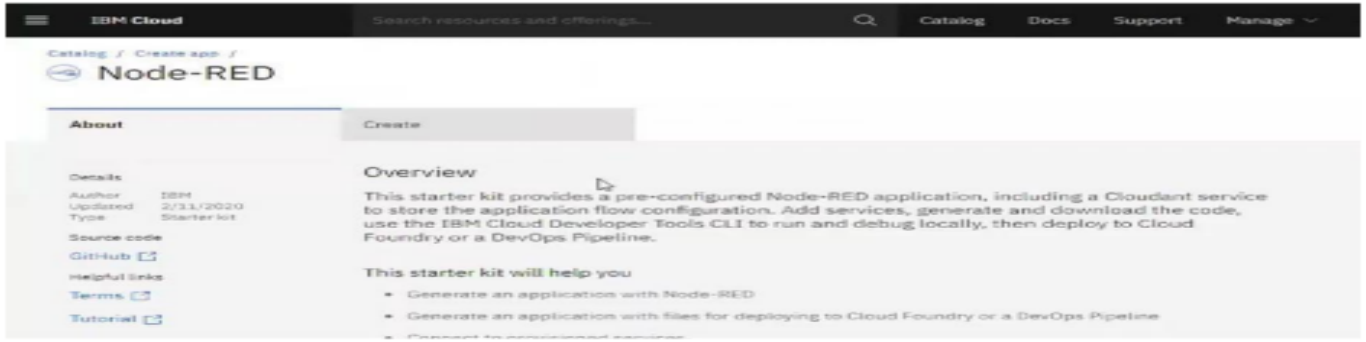
CREATE A NODE RED SERVICE

Create Node-RED Service

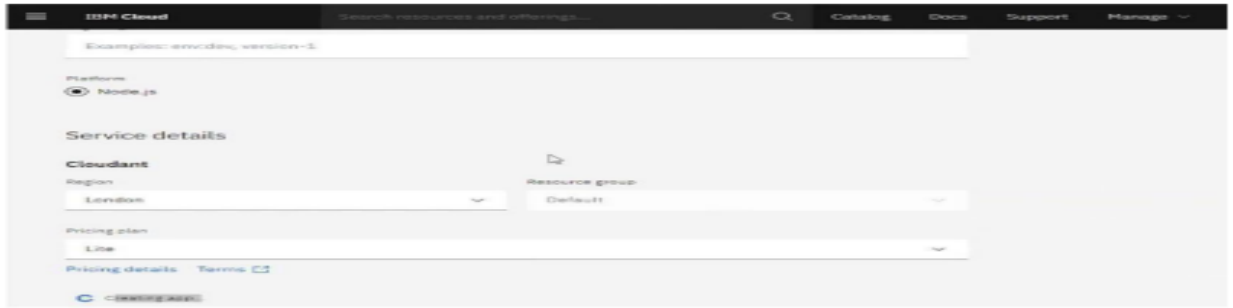
Date	**21-10-2022**
Team ID	**PNT2022TMID11121**
Project Name	**IOT based safety gadget for child safety monitoring and notification**
Maximum Marks	**4 Marks**
Submitted By	➢ VIGNESHWER.S ➢ SUJEEVE.A ➢ SUBASH RAM.S ➢ VASIKARAN.J

Aim: To create a web applications create a Node-RED service.Steps followed:

- Navigated to the App creation page



- Entered project details and clicked on create



- Clicking on the "Deploy your App" Button

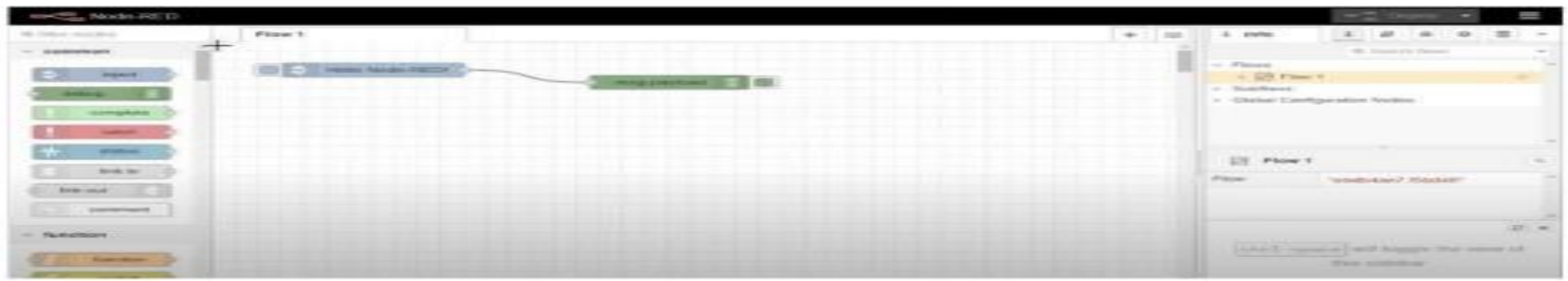
- Setting up the environment and deploying the app

- Successfully deployed the app

Delivery Pipelines

Name	ci-pipeline
Status	Success
Last input	Last commit by IBM Cloud DevOps Services (7 minutes ago) Clone from zip

- Dragged and dropped components into the editor



- Editing some values of the properties

- Successfully deployed the app

Result: Successfully created a NodeRED service on IBM Cloud

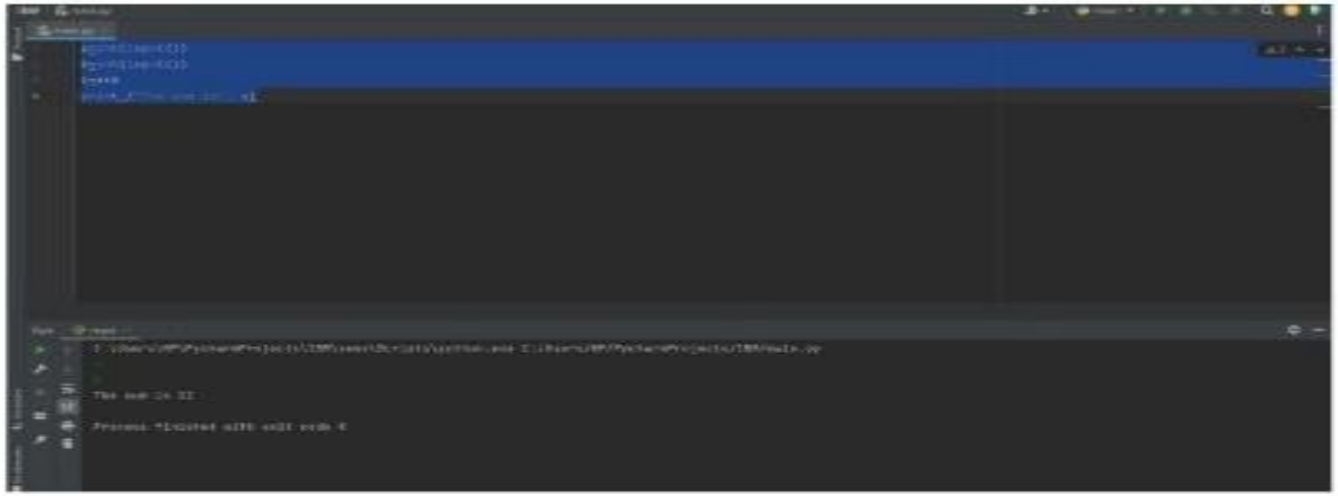
DEVELOP A PYTHON SCRIPT

DEVELOP A PYTHON SCRIPT

Date	**21-10-2022**
Team ID	**PNT2022TMID11121**
Project Name	**IOT based safety gadget for child safety monitoring and notification**
Maximum Marks	**4 Marks**
Submitted By	**VIGNESHWER.S SUJEEVE..A SUBASH RAM.S VASIKARAN.J**

SCRIPT:

```
a=int(input())
b=int(input())
c=a+b
print ("The sum is", c)
```



PROJECT DEVELOPMENT PHASE

- SPRINT DELIVERY-1
- SPRINT DELIVERY-2
- SPRINT DELIVERY-3
- SPRINT DELIVERY-4

SPRINT DELIVERY 1

SPRINT 1

Date	29 October 2022
Team ID	PNT2022TMID11121
Project Name	IOT BASED DEVICE FOR CHILD SAFETY MONITORING AND NOTIFICATION
Maximum Marks	20 marks
Team members	S VIGNESHWER A SUJEEV J VASIKARAN S SUBASH RAM

CREATE A LOGIN

```
<!DOCTYPE
html>
            <html lang="en">
            <head>
                <meta charset="UTF-8">
                <meta http-equiv="X-UA-Compatible" content="IE=edge">
                <meta name="viewport" content="width=device-width, initial-scale=1.0">
                <link rel="stylesheet" href="/LOGIN.css">
                <title>Sign Up</title>
                <script>
                    if (window.location.hostname !== "localhost") {
                        if (location.protocol !== "https:") {
                            location.replace(
                                `https:${location.href.substring(
                                    location.protocol.length
                                )}`
                            )
                        }
                    }
                </script>
                <script src="./localforage.js"></script>
            </head>
            <body>
                <div class="wrapper">
                    <div class="loginContainer">
                        <span>Signup To Continue</span>
                        <div class="traditionalLoginContainer">
                            <form class="signupForm" action="/" method="post">
                                <input type="email" name="email" placeholder="Email"
            id="email">
```

```
                    <input type="password" name="password"
placeholder="Password" id="password">
                    <input class="loginButton" type="submit" value="Login">
                </form>
            </div>
            <div class="loginWithFireContainer">
                <button type="button" class="fire" title="Login with Safety"
id="fire">Login with Safety</button>
            </div>
            <a class="hyperlink" href="/register">Don't have an Account?
Register /^</a>
        </div>
    </div>
    <script>
        // Necessary for Fire OAuth to Function
        const fireBroadcastingChannel = new
BroadcastChannel('fireOAuthChannel');
        fireBroadcastingChannel.addEventListener('message', async event => {

            let data = event.data
            /**
            * @typedef {Object<string, any>} Data
            * @property {boolean} success - Whether the login was successful
            * @property {string} token - The data returned from the login
i.e. Fire Token
            */



            // data.token is the message sent from the fireOAuthChannel
after verification
            // data.success is a boolean that indicates whether the
verification was successful
            // data.token is the fire token

            // What to do with the Fire Token?
            // * Fire Token is an unique token which uniquely identifies the
user who authorized your login attempt with Fire
            // * You can use this token ONLY ONCE as it will be destroyed
after the first use

            // 1. Send the fire token to the Fire Server to verify the user
            //    - You can do that client sided or server sided
            //    - You need to send a POST Request to the Fire Server with
the fire token
            //      at the URL: http://localhost:3003/api/tokens/verify
```

```
            //    - The Fire Server will verify the fire token and return a
response
            //      - If the verification was successful - CODE (200), the
Fire Server will return a response with the user's data
            //      - If the verification was unsuccessful - CODE (400) or
CODE (401), the Fire Server will return a response with an error `message`
            //    - You can use the data returned from the Fire Server to
create a new user in your database

            // This example will send the token to Fire Servers and
console.log the response
            console.log("%c" + `Fire Token: ${data.token}`, `color: #f1c40f;
font-weight: bold;` );
            const response = await
fetch('https://fire.adaptable.app/api/tokens/verify', {
                method: `POST`,
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify({
                    token: data.token
                })
            })
            // get the response
            const responseData = await response.json()
            // console.log the response
            console.log(responseData)
            await localforage.setItem('userData', {...responseData, isFire:
true})
            // Adding the user data to the user Database
            let database = await localforage.getItem("userDatabase")
            if (database == null) {
                database = []
            }
            database.push(responseData)
            await localforage.setItem("userDatabase", database)
            // redirect to the home page
            window.location.href = '/'

        })

        function popupwindow(url, title, w, h) {
            var left = (screen.width/2)-(w/2);
            var top = (screen.height/2)-(h/2);
```

```
            return window.open(url, title, 'toolbar=no, location=no,
directories=no, status=no, menubar=no, scrollbars=no, resizable=no,
copyhistory=no, width='+w+', height='+h+', top='+top+', left='+left);
        }
        document.getElementById("fire").addEventListener("click", function()
{
            popupwindow("/firebauth.html", "Fire OAuth", 450, 600)

        })
    </script>

    <script>
        // this.website's Scripts / App Logic
        document.querySelector(".signupForm").addEventListener("submit" ,
async (e) => {
            e.preventDefault()
            let email = document.getElementById("email").value
            let password = document.getElementById("password").value
            let flag = false
            let userData = await localforage.getItem("userDatabase")
            if(userData) {

                userData.forEach(e => {
                    if(e.email === email) {
                        if(e.password === password || e.isFire === true) {
                            localforage.setItem("userData", e)
                            flag = true
                            window.location.href = "/"
                        }
                    }
                })

            } else {
                alert("User Not Found")
            }

            if(!flag) {
                alert("Invalid Credentials")
            }
        })
    </script>
</body>
</html>
```

OUTPUT:

SPRINT PLAN 2

SPRINT 2

Date	5 November 2022
Team ID	PNT2022TMID11121
Project Name	IOT BASED DEVICE FOR CHILD SAFETY MONITORING AND NOTIFICATION
Marks	20 marks
Members	S VIGNESHWER A SUJEEV J VASIKARAN S SUBASH RAM

Sprint 2 is about **LOGIN and NOTIFIACATION** of the IoT device in Parent's Web Application for getting information about Child's Status.

LOGIN:

This Coding is to built login page of parent's application to get information about child's condition.

Coding:

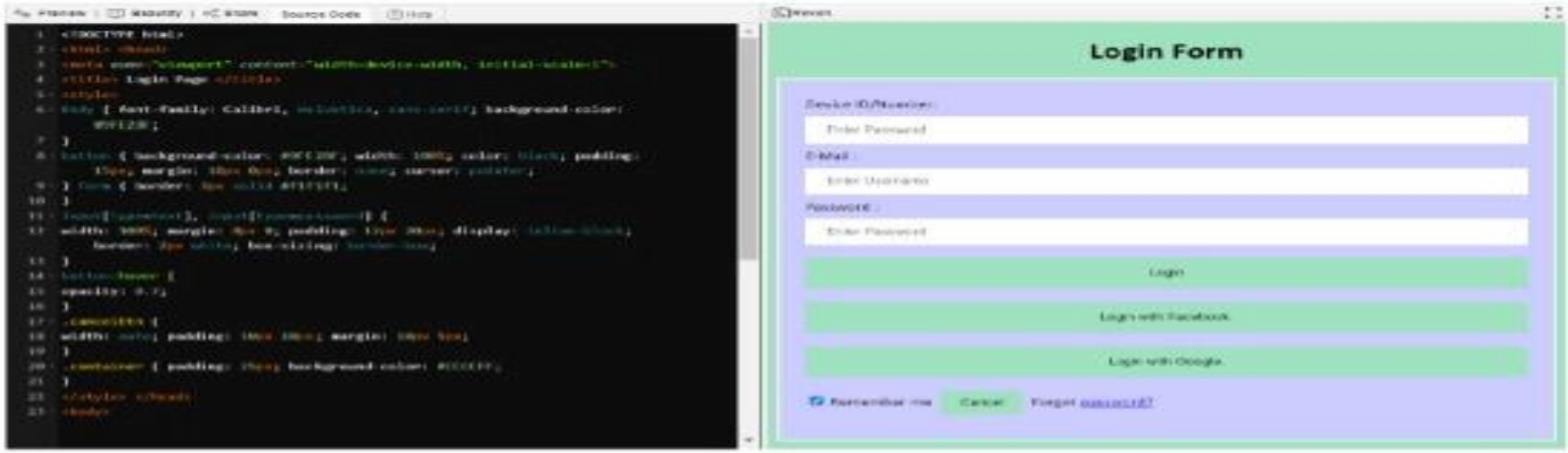
```
<!DOCTYPE html>
<html> <head>
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
.cancelbtn {
    width:          auto;
    padding: 10px 18px;
    margin: 10px 5px;
  }
.container { padding: 25px;
    background-color: #CCCCFF;
  }
</style> </head>
<body>
  <center> <h1> Login Form </h1> </center>
  <form>
    <div class="container">
    <label>Device ID/Number: </label>
      <input type="password" placeholder="Enter Password" name="password" required>
      <label>E-Mail : </label>
      <input type="text" placeholder="Enter Username" name="username" required>
      <label>Password : </label>
      <input type="password" placeholder="Enter Password" name="password" required>
      <button type="submit">Login</button>
      <button class="loginBtn loginBtn--facebook">Login with Facebook.</button>
      <button class="loginBtn loginBtn--google">Login with Google.</button>
      <input type="checkbox" checked="checked"> Remember me
```

```
.cancelbtn {
    width:          auto;
    padding: 10px 18px;
    margin: 10px 5px;
  }
.container { padding: 25px;
    background-color: #CCCCFF;
  }
</style> </head>
<body>
  <center> <h1> Login Form </h1> </center>
  <form>
    <div class="container">
    <label>Device ID/Number: </label>
      <input type="password" placeholder="Enter Password" name="password" required>
      <label>E-Mail : </label>
      <input type="text" placeholder="Enter Username" name="username" required>
      <label>Password : </label>
      <input type="password" placeholder="Enter Password" name="password" required>
      <button type="submit">Login</button>
      <button class="loginBtn loginBtn--facebook">Login with Facebook.</button>
      <button class="loginBtn loginBtn--google">Login with Google.</button>
      <input type="checkbox" checked="checked"> Remember me
```

```
        <button type="button" class="cancelbtn"> Cancel</button> Forgot
        <a href="#"> password? </a>
      </div>
    </form>
</body>
</html>
```

OUTPUT


Login Form
Enter Password
Enter Username
Enter Password
Login
Login with Facebook
Login with Google
Cancel

NOTIFICATION:

This coding will make connection between IoT Device & Parent's application. When the child cross across the geofence message will be notified on parent's application.

Coding:

```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload,unsigned int payloadlength);
//----------credentials of IBM Account------
#define ORG "45z3o2"// IBM ORGANIZATION ID
#define DEVICE_TYPE "ESP32_Controller"//DEVICE TYPE MENTIONED IN IOT WATSON PLATFORM
#define DEVICE_ID "bme2"//DEVICE ID MENTIONED IN IOT WATSON PLATEFORM
#define                                    TOKEN
"OKZ+g@JfPWDOd6wBT"//Token String data3;
float dist;
//---------customize the above value---------
char server[]=ORG ".messaging.internetofthings.ibmcloud.com";//server name
char publishtopic[]="ultrasonic/evt/Data/fmt/json";/*topic name and type of event perform and format in which
 data to be send*/
char subscribetopic[]="ultrasonic/cmd/test/fmt/String";/*cmd REPRESENT Command tupe and
COMMAND IS TEST OF FORMAT STRING*/
```

```
char       authMethod[]="use-token-auth";//authentication       method    char
token[]=TOKEN;
char clientid[]="d:" ORG ":" DEVICE_TYPE":" DEVICE_ID;//CLIENT ID
// ..........
WiFiClient wifiClient;// creating an instance for wificlient
PubSubClient client(server, 1883 , callback , wifiClient);/*calling the predefined client id by passing parameter like
server id,portand wificredential*/ int LED =4; int trig =5; int echo=18; void setup(){ Serial.begin(115200);
pinMode(trig,OUTPUT);     pinMode(echo,INPUT);     pinMode(LED,OUTPUT);     delay(10);     Serial.println();
wificonnect(); mqttconnect();
}
void loop() {
  digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW); float
  dur=pulseIn(echo,HIGH); float
  dist=(dur * 0.0343)/2;
  Serial.print("distance in cm");
  Serial.println(dist);
  PublishData(dist);
  delay(1000); if
  (!client.loop()){
```

```
        mqttconnect();
    }
}

/*.....retriving to cloud ..... */ void PublishData(float dist){ mqttconnect();//function
call for connecting to ibm
  /*creating the string in form of JSON to update the data to ibm cloud*/
  String object;
  if(dist<100)
  {

    digitalWrite(LED,HIGH); Serial.println("no object is
    near"); object="Near";
  }

  else
  {

    digitalWrite(LED,LOW); Serial.println("no object
    found"); object="No";
  }
```

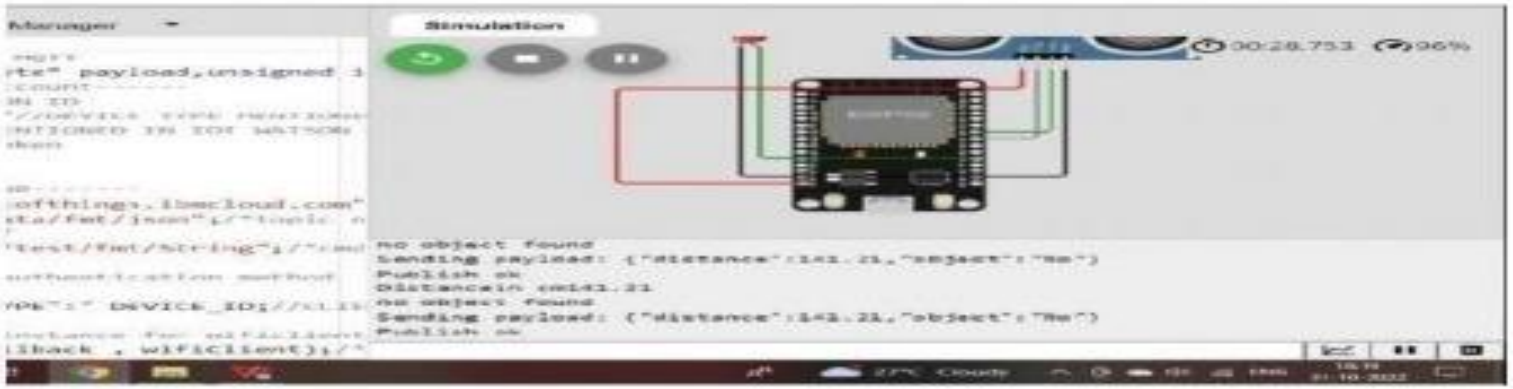
```
    mqttconnect();
  }
}

/*.....retriving to cloud ..... */ void PublishData(float dist){ mqttconnect();//function
call for connecting to ibm
  /*creating the string in form of JSON to update the data to ibm cloud*/
  String object;
  if(dist<100)
  {

    digitalWrite(LED,HIGH); Serial.println("no object is
    near"); object="Near";
  }

  else
  {

    digitalWrite(LED,LOW); Serial.println("no object
    found"); object="No";
  }
```

```
  String payload="{\"distance\":"; payload
  +=dist;  payload  +="," "\"object\":\"";
  payload += object;
  payload += "\"}";

  Serial.print("Sending payload: ");
  Serial.println(payload); if(client.publish(publishtopic, (char*) payload.c_str())){
    Serial.println("Publish ok");/* if its sucessfully upload data on the cloud then it will print publish ok in serial
    monitor or else it will print publish failed"/
  } else{
    Serial.println("Publish failed");

  }
}

void mqttconnect(){
  if(!client.connected()){
    Serial.print("Reconnecting client to "); Serial.println(server);
    while(!!client.connect(clientid,authMethod, token)){
      Serial.print("."); delay(500);
    }
    initManagedDevice();
    Serial.println();
```

```
    }
}

void wificonnect()//function defenition for wificonnect
{
   Serial.println();
   Serial.print("Connecting to ");
   WIFI.begin("vivo 1816", "taetae95",6);//PASSING THE WIFI CREDIDENTIALS TO ESTABLISH CONNECTION
   while (WiFi.status() !=WL_CONNECTED){
      delay(500);
      Serial.print(".");

   }
   Serial.println("");
   Serial.println("WiFi connected");
   Serial.println("IP address");
   Serial.println(WiFi.localIP());
}

void initManagedDevice(){
   if(client.subscribe(subscribetopic)){
      Serial.println((subscribetopic));
      Serial.println("subscribe to cmd OK");
```

OUTPUT

Output:



SPRINT PLAN-3

SPRINT 3

Date	12 November 2022
Team ID	PNT2022TMID11121
Project Name	IOT Based Safety Gadget for Child Safety Monitoring and notification
Maximum Marks	20 Marks
Team Members	S VIGNESHWER A SUJEEV J VASIKARAN S SUBASH RAM

LOCAL FORAGE:

```
!function(a)
{if("object"==typeof exports&&"undefined"!=typeof module)module.exports=a(); else
if("function"==typeof define&&define.amd)define([],a); else{var b; b="undefined"!=typeof
window?window:"undefined"!=typeof global?global:"undefined"!=typeof
self?self:this,b.localforage=a()}}(function(){
 return function a(b,c,d){
  function e(g,h){if(!c[g]){if(!b[g]){
```

```
var i="function"==typeof require&&require;
if(!h&&i)return i(g,!0);if(f)return f(g,!0); var
j=new Error("Cannot find module '"+g+"'");
throw j.code="MODULE_NOT_FOUND",j}var k=c[g]={exports:{}};
b[g][0].call(k.exports,function(a){
var c=b[g][1][a];return e(c||a)},k,k.exports,a,b,c,d)}
return c[g].exports} for(var f="function"==typeof
require&&require,g=0;g<d.length;g++)e(d[g]); return
e}({1:[function(a,b,c){(function(a){"use strict"; function c(){k=!0;for(var
a,b,c=l.length;c;){
for(b=l,l=[],a=-1;++a<c;)b[a]();c=l.length}k=!1}
function d(a){1!==l.push(a)||k||e()}var
e,f=a.MutationObserver||a.WebKitMutationObserver; if(f){var
g=0,h=new
f(c),i=a.document.createTextNode("");h.observe(i,{characterData:!0}),e=function(){i.data=g=++g%2}} else
if(a.setImmediate void ea.MessageChannel)e="document"in all"onreadystatechange in
a.document.createElement("script") function(){var b=a.
document.createElement("script");b.onreadystatechange-function()(c(),b.
onreadystatechange=null,b.parentNode.removeChild
(b),b=null),a.document.documentElement.appendChild(b)}:function() (setTimeout(c,8));else{var j-new
a.MessageChannel;j.port1.onmessage=c,e-function(){j.port2.postMessage(0)}}var k,1-[];b.exports-
d)).call(this, "undefined"!=typeof global?global: "undefined"!=typeof self?self: "undefined"!=typeof
window?window: {})},{}],2: [function(a,b,c){"use strict"; function d()() function e(a){if("function" I-typeof
a) throw new TypeError("resolver must be a function"); this.states, this.queue=[], this.outcome vald
0,aldi(this,a)} function f(a,b,c){this.promise-a, "function"==typeof b&&(this.onFulfilled-b,
this.callFulfilled- this.otherCallFulfilled), "function"typeof c&&(this.onRejected=c,
```

```
    }
  }
  check()
      </script>
</head>
<body style=" height:
      100%; margin: 0;
            font-weight: 300;
font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto,        Oxygen,
              Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue',
                sans-serif;
      "
>
      <div      class="wrapper"   style="
            height: 90%; display: flex;
            flex-direction: column; align-
            items:     center;     justify-
            content: center;
    text-align: center;
            "
      >
            <div      class="details"   style="
                  display: flex; flex-direction:
                  column; align-items: center;
                  gap: 20px; padding: 1rem;
                  border-radius: 5px;
```

```
                box-shadow: 0 0 8px 0px #44444444;
        max-width: 80%;
                "
            >
                <h1 class="name" style="margin: 0"></h1>
                <div class="imageContainer"
                    style="padding: 10px; height: 10rem; width: 10rem"
                >
                    <img class="image" alt="profile picture" />
                </div>
                <h2 class="email" style="margin: 0"></h2>

    <a style="text-decoration: none;text-align: center;font-size: 1.2rem;color: #0070f3;font-weight:
400;" href="./dashboard">Go to Dashboard ?</a>
                </div>
        </div>
        <script> async function
            main() {
    let name = document.querySelector(".name") let
    image = document.querySelector(".image") let email
    = document.querySelector(".email") let userData =
    await localforage.getItem("userData") if(userData
    == null) {
        window.location.href = "/login"
    }
```

OUTPUT


Preview | Beautify | Share
Source Code
Result
profile picture
Go to Dashboard ?

SPRINT 4

SPRINT 4

Date	19 November 2022
Team ID	PNT2022TMID11121
Project Name	IoT Based Safety Gadget for Child Safety Monitoring and notification
Maximum Marks	20 Marks
Team Members	S VIGNESHWER A SUJEEV J VASIKARAN S SUBASH RAM

FIREOAUTH:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/css/fireoauth.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.css">
    <link rel="shortcut icon" href="https://raw.githubusercontent.com/tharunoptimus-
pd/firepwa/main/favicon.ico?token=GHSAT0AAAAAABR46HVJ5M5L3QGFRZRQXOISYUJUWAA"
type="image/x-icon">
```

```
<style> html,
  body {
    height: 100%;
    margin: 0;
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen,
      Ubuntu, Cantarell, "Open Sans", "Helvetica Neue", sans-serif; font-
    weight: 300;
  }

  a {
    text-decoration: none;
    color: #007bff; font-
    weight: 500;
    font-size: 1.2rem;
  }

  h3 {
    font-size: 1.4rem;
  }

  h3, h4 {
    margin: 0;
    padding: 0.3rem 0;
  }

  .wrapper { display: flex;
    flex-direction: column;
```

```
  align-items:      center;
  justify-content: center;
  height:   100%;   text-
  align: center;
}

.oneClickSignin   {   padding:   0.5rem;
  border: 1px solid #44444444; border-
  radius: 5px; box-shadow: 0 0 3px 0px
  #44444444;
  opacity: 0.2;
  pointer-events: none;
}

.qrcode { opacity:
  0.1;
}

.learnAboutFire { padding-top:
  1.25em;
}

.qrHolder { display:
  none;
  margin-top: 3rem;
}
```

```
    .qrContainer { align-items:
      center; display: flex;
      justify-content: center;
      padding: 8px; margin:
      2rem auto; box-
      shadow: 0 0px 6px 1px
      rgb(0 0 0 / 16%);
      border: 1px solid
      #44444444; border-
      radius: 6px; width:
      200px;
      height: 200px;
    }
  </style>
  <title>Fire OAuth</title>
  <script> if (window.location.hostname !==
    "localhost") {
      if (location.protocol !== "https:") {
        location.replace(
          `https:${location.href.substring(
            location.protocol.length
          )}`
        )
      }
    }
  </script>
</head>
```

```
<body>
  <div class="wrapper">
    <h3 class="pageTitle">Login with Fire ??</h3>
    <div class="qrAuthorize">
      <h4 class="subTitle">Scan QR from your Fire OAuth App??</h4>

      <div class="qrContainer">
        <canvas id="qr-code" class="qrcode"></canvas>
      </div>
    </div>

    <div class="oneClickSignin">
      <h4>Have Fire PWA on this device?</h4>
      <a target="_blank" id="authorizeOverLink"
href="https://firepwa.netlify.app/authorize?sessionId" rel="noopener">Click to Authorize ?? </a>
</div>

    <div class="learnAboutFire">
      <a target="_blank" href="https://fireoauth.netlify.app" rel="noopener">Learn More about Fire
??</a>
    </div>
  </div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/qrious/4.0.2/qrious.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.2.0/socket.io.js"></script>
  <script>
```

```
const FIRE_API_KEY = "635b790a3bcc6b59c4b772d0"
const FIRE_ENDPOINT = "https://fire.adaptable.app/api/apis/generate" const
CHANNEL_NAME = "fireOAuthChannel"
const broadCastingChannel = new BroadcastChannel(CHANNEL_NAME)
const FIRE_SERVER_SOCKET_ENDPOINT = "https://fire.adaptable.app" let
socket = io(FIRE_SERVER_SOCKET_ENDPOINT)

let qr

let qrcode = document.querySelector(".qrcode") let
oneClickSignin = document.querySelector(".oneClickSignin") let
pageTitle = document.querySelector(".pageTitle")
let subTitle = document.querySelector(".subTitle")

function setOpacity(opacity) {
  oneClickSignin.style.opacity = opacity
  oneClickSignin.style.pointerEvents = opacity === "1" ? "auto" : "none"
  qrcode.style.opacity = opacity
}

async function getSessionID() {
  let response
  try {
    response = await fetch(`${FIRE_ENDPOINT}/${FIRE_API_KEY}`, {
      method: "GET",
      headers: {
        "Content-Type": "application/json",
```

```
            }
          })
        } catch (error) {
          console.log(error) return null
        }

        let data = await response.json() let
        { sessionId, chatRoomId } = data
        return { sessionId, chatRoomId }
      }

      function generateQR(value) {
        (qr = new QRious({ element:
          document.getElementById("qr-code"), size:
          200, level: 'M',
          value: value,
        }))
      }

      function changeHREF ({sessionId, chatRoomId}) {
        let firePwaUrlHostname = "https://firepwa.netlify.app" let
        originURL = encodeURIComponent(window.location.origin)
let url =
 `${firePwaUrlHostname}/authorize.html?sessionId=${sessionId}&chatRoomId=${chatRoomId}&url=${o
 ri ginURL}` let a = document.getElementById("authorizeOverLink") a.href = url
      }
      async function fire() { NProgress.set(0.4)
```

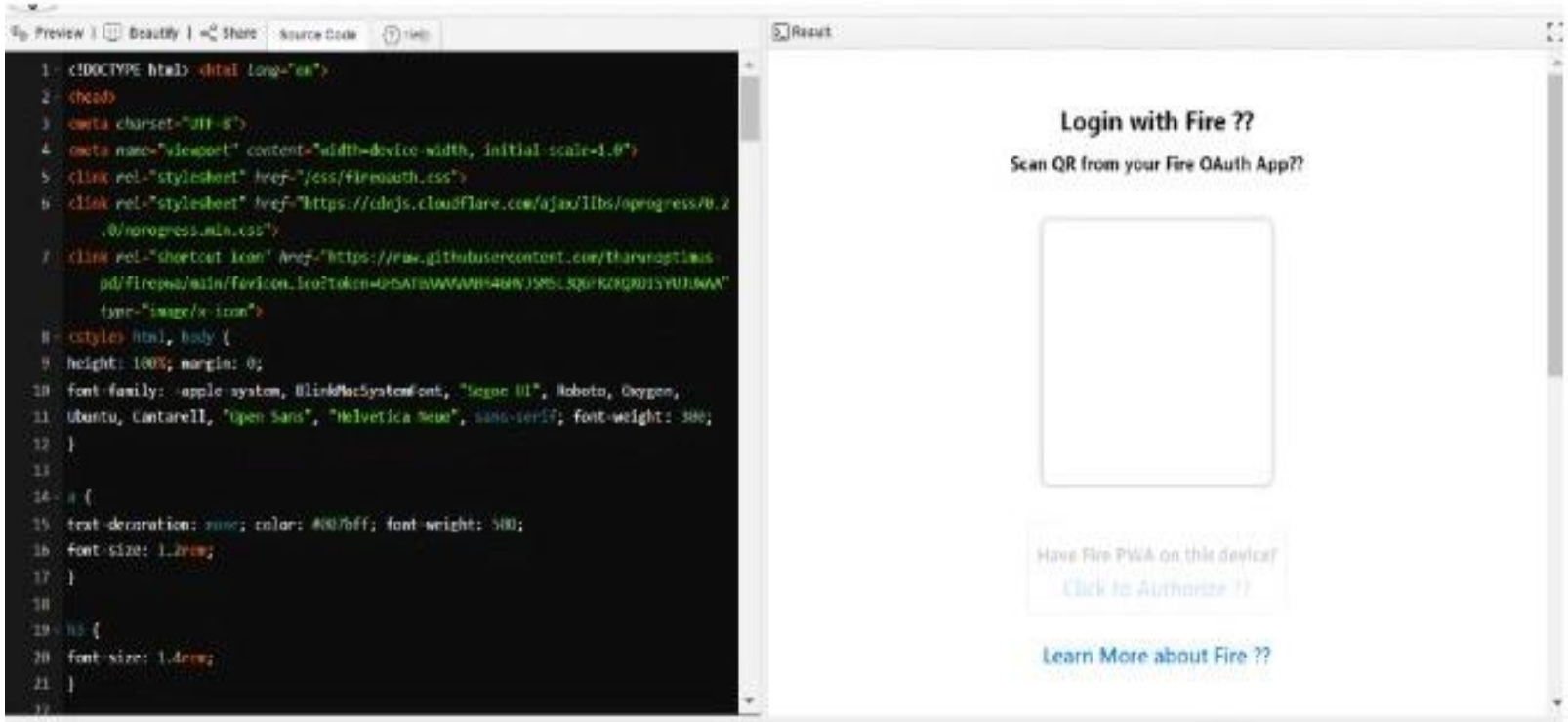
```
        broadCastingChannel.postMessage(data)

        window.close()
      })
    </script>
  </body>
</html>
```

DASHBOARD:

```
 <!DOCTYPE html>
 <html lang="en">
 <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./css/dashboard.css">
    <title>Dashboard</title>
    <script src="./localforage.js"></script>
 </head>
 <body>
    <div class="wrapper">
      <div class="header">
        <span class="heading">Dashboard</span>
        <span class="right">
          <span class="username">Hello User</span>
          <span>
```

```
    </div>
    <script> async function main() { let userData = await
      localforage.getItem('userData')
        if(userData == null) {
          window.location.href = "/login"
        }
        document.querySelector(".username").innerHTML = `Hello ${userData.firstName}`
        document.querySelector(".profilePic").src = userData.profilePic
      }
      main()
      document.querySelector(".logout").addEventListener("click", async () => {
        await localforage.setItem('userData', null)
        window.location.href = "/login"
      })
    </script>
  </body>
</html>
```

OUTPUT


Preview
Beautify
Share
Source Code
Result
<!DOCTYPE html> <html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="/css/firesauth.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.css">
<link rel="shortcut icon" href="https://raw.githubusercontent.com/tharunoptimus-pd/firepwa/main/favicon.ico?token=GHSAT0AAAAAABH4BRNJ5MSLBQUFKZKQBOISYUJAAA" type="image/x-icon">
<style> html, body {
height: 100%; margin: 0;
font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen,
Ubuntu, Cantarell, "Open Sans", "Helvetica Neue", sans-serif; font-weight: 300;
}
a {
text-decoration: none; color: #007bff; font-weight: 500;
font-size: 1.2rem;
}
h1 {
font-size: 1.6rem;
}
Login with Fire ??
Scan QR from your Fire OAuth App??
Have Fire PWA on this device?
Click to Authorize ??
Learn More about Fire ??