PROJECT REPORT

Team ID	**PNT2022TMID14424**
Project Name	**IoT Based Safety Gadget for Child Safety Monitoring & Notification**

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The internet of things (IoT) refers to the set of devices and system that stay with real-world sensor and to the internet. During years' Child safety is under threat and it is very important to provide a technology- based solution which will help them under panic situations and monitor them using a smart gadget. Child tracker helps the parents in continuously monitoring the child's location. They can simply leave their children in school or parks and create a geofence around the particular location. By continuously checking the child's location notifications will be generated if the child crosses the geofence. Notifications will be sent according to the child's location to their parents or caretakers. The entire location data will be stored in the database.

1.2 PURPOSE

It assists parents to monitor their children remotely. In case situations happen, notifications will be sent to parents so that actions can be taken. Through this, child safety can be ensured. By continuously checking the child's location notifications will be generated if the child crosses the geofence. Notifications will be sent according to the child's location to their parents or caretakers. The entire location data will be stored in the database. Child can also initiate emergency notification to the parents in-case of unsafe situation. By this, parents know what is happening remotely and can take actions if something goes wrong. It provides parents with the real-time location to monitor the child. It makes parents to make monitor their child from their workplace. Parents can be relax and calm by using this device.

CHAPTER 2

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

As we all know, kids are the heartbeat of every parent, and when it comes to a child with special needs, parents have to be extra careful. They have to take extra care of their child. Parents need to ensure safety of their children but in real time they need to get to work and need to worry about their child whether he/she is safe or not. So to ensure safety they need to monitor & to notify their child what he/she is doing and to know whether they are in safe atmosphere or not to ensure the safety of the child.

2.2 REFERENCES

1. Lai Yi Heng, Intan Farahana Binti Kamsin, "IoT-based Child Security Monitoring System", 2021.

The main aim of this paper is to describes a child safety issues and the need of using child security system. The smart band is waterproof, chargeable and equipped with sensors. All sensors are connected through the internet. This smart band contains GPS for tracking, identifying children's location and setting geofences. Via the smart band, children can also contact parents. Emergency button, a feature in which will automatically record video and automatically call 4 emergency contacts when it is pressed. An alert message along with the video clip is sent to parents' devices It assists parents to monitor their children remotely. In case situations happen, notifications will be sent to parents so that actions can be taken.

2. Poonkuzhlai1 P, Aarthi R, Yaazhini V M, Yuvashri S, Vidhyalakshmi G, "Child Monitoring and Safety System Using Wsn and Iot Technology", 2021.

The main idea of this paper is to provide better and efficient health services and security to the school children by implementing a networked information cloud through IoT so that the experts and doctors could make use of this data and provide a fast and an efficient solution. Like smart watches more advanced wearable models are required to reduce the risks in the human lives by giving hands to the children. The composition of more different purpose sensor may improve the abilities of required system design on the given problem domain. This paper has covered about the issues of children how it can be overcome by using advanced IoT components available in the hand.

3. Arun Francis G, Janani I, Kavya S and Ramiyadevi K, "Child Safety Wearable Device Using Raspberry Pi", 2020.

This paper would explore the idea of a children's smart wearable device. This system has the purpose of helping parents to locate their children. It's focuses on making an SMS text activated to communicate between the wearable child and a parent as the framework for GSM Mobile Communication. The wearable device will respond in real time with a text containing the exact location of a child, which will provide details on the position of the child and the ambient temperature. The new method implemented was using a pi camera to capture the image of a person who is in the opposite position of child. The mail will be send to the parents to track the child location using Raspberry Pi. The pulse sensor is used for monitoring child's pulse rate. The sound sensor is used to differentiate the voice of the child and predict whether the child is in abduction condition. The secondary measure used in this project is the individuals present in the child's surroundings who could respond immediately to the safety of the child until the parents arrive at the place.

2.3 PROBLEM STATEMENT & DEFINITION

The objective of this project is to safeguard the child from threads. Now a days the safety measures of children has been reduced in huge number. Thus the violence against children increasing day by day.Our project mainly focus on sensing the children's Temperature and Heartbeat. By monitoring the activities the state of the child is analyzed. By using GSM, if child reaches the critical state then the latitude and longitude of that particular location is sent as an alert message to the parents.

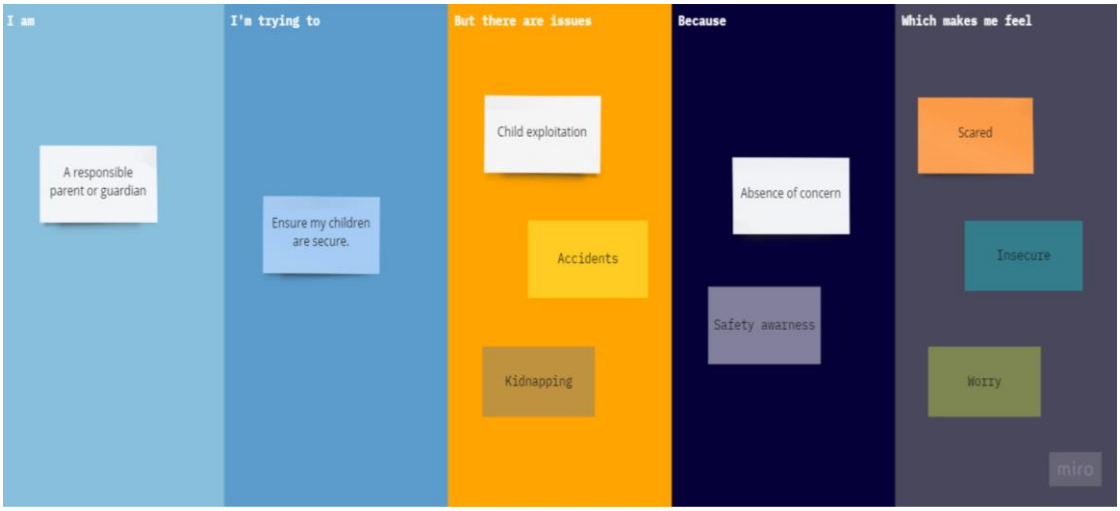


Fig 2.1 Problem Statement

CHAPTER 3

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple,easy –to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it.The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenge.

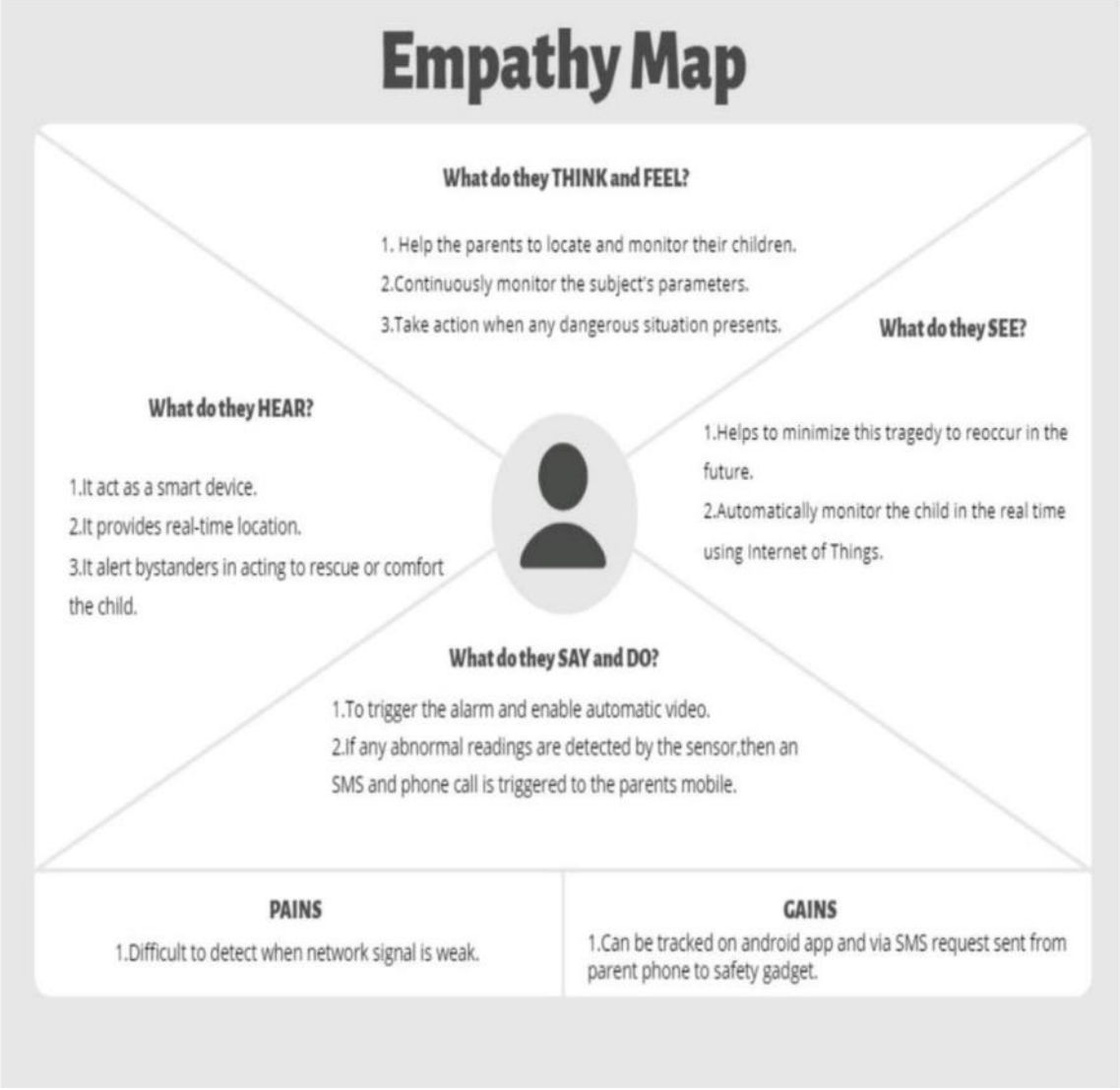


Fig 3.1 Empathy map

3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

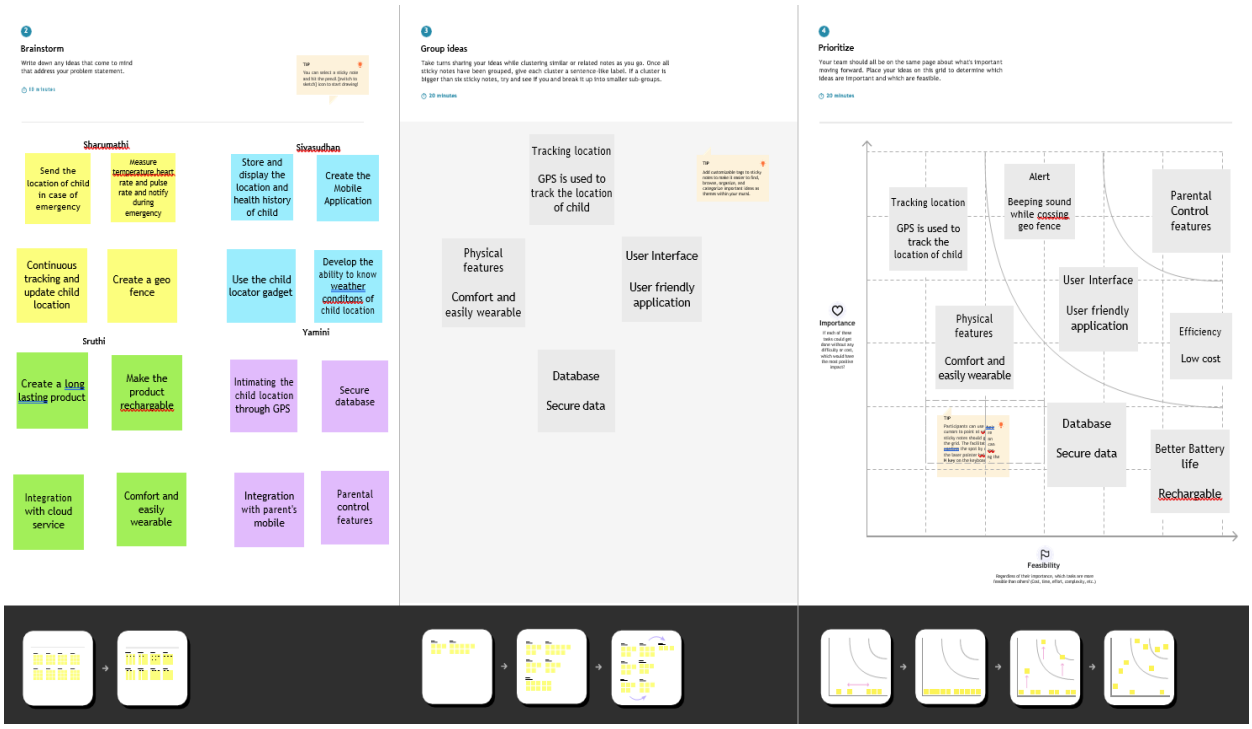


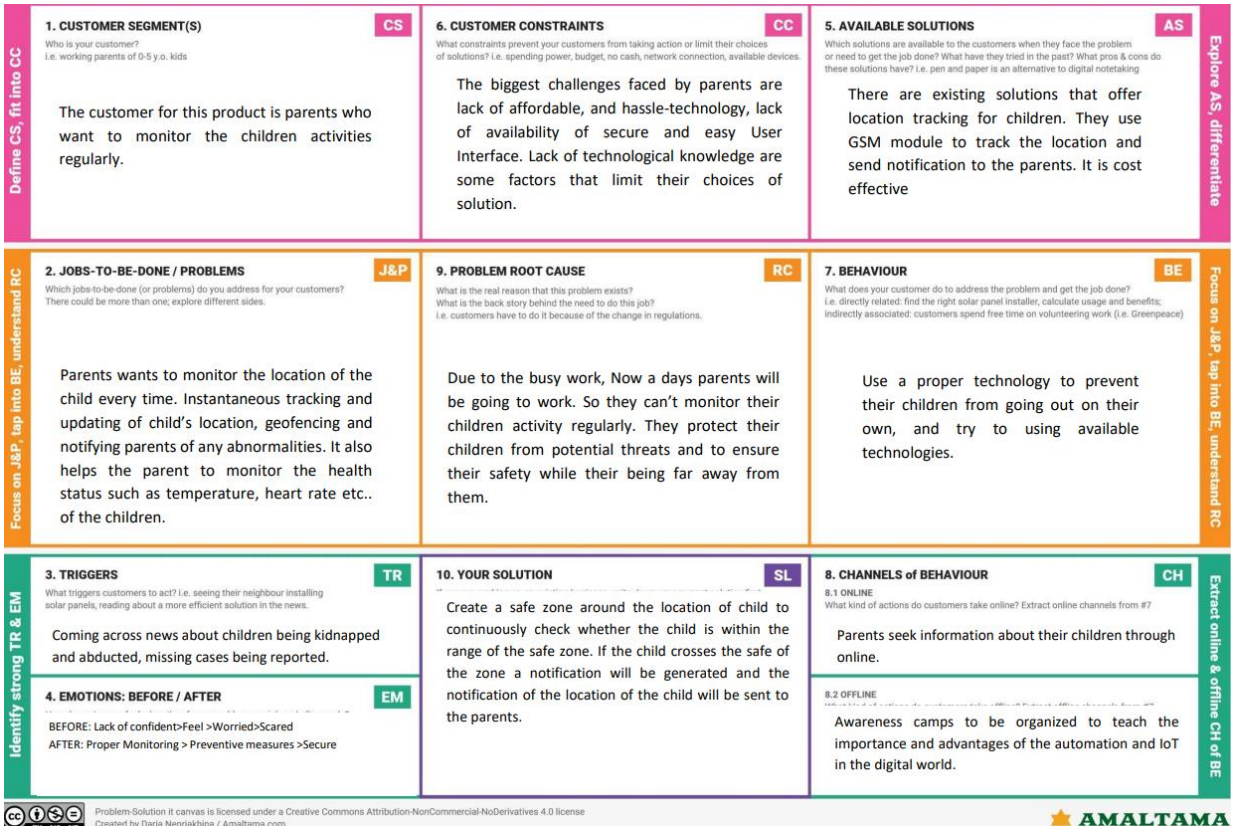
Fig 3.2 Brainstorming

3.3 PROPSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Due to the digital world, Now a days the parents will be going to work.So they can't monitor the child activity regularly. • Child kidnappers continue to kidnap children away from their parents or other legal guardians. More precautioms must be taken against child abuse, accidents and kidnapping and its effects because the crisis that results from it can be extremely harsh.
2.	Idea / Solution description	• Create a safe zone around the location of the child to continuously check whether the child is within the range of the safe zone. If the child crosses the safe of the zone a notification will be generated and the notification of the location of the child will be sent to the parents. • Use of IoT sensors enables to get accurate real time information such as temperature, heart rate, blood pressure and respiratory rate etc..
3.	Novelty / Uniqueness	• IoT devices respond intelligently in emergencies without the need for human intervention, such as the advancement of sensor technology. • It also helps the parents to monitor the health status of the children.

4.	Social Impact / Customer Satisfaction	• When the children cross the safe zone the alerts message will be send to the parents • It will be very useful for parents who don't have time to watch their children. • It is simple to use so anyone can use it.
5.	Business Model (Revenue Model)	• IoT revenue model sit at the heart of business models. They make all difference in considering deployment successful. • The gadget is low cost so all parents can buy it. • It is comfortable to wear.
6.	Scalability of the Solution	• It could be improved any further by installing a little camera within a smart device for exceptional security and protection. • Technical complexity is one of the greatest IoT scalability issues. By using right technology is paramount to safeguard.

3.4 PROBLEM SOLUTION FIT



CHAPTER 4

4. REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	User should register via email.
FR-2	User Authentication	The identity of the user is verified.
FR-3	User Notification	Notification to registered mobile number.
FR-4	Log in to the system	Log in to the system and check the child activity.
FR-5	Manage the activity	Then check the child's location.
FR-6	Sensors	Temperature, Heart rate etc.
FR-7	Log out	Then log out of the system.

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non- Functional Requirement(Epic)	Description
NFR-1	Usability	Usability is about effectiveness and efficiency. Our system provides over all satisfication to the user.
NFR-2	Security	Our security system is to create a secure environment to the children.
NFR-3	Reliability	Our model is reliable because it increased reliability towards technology and reduced reliability towards guardians.
NFR-4	Performance	Using this technology, parent's can monitor their child location and also monitor their child health status.
NFR-5	Availability	The proposed model has suitable recoverability and it is rechargable.
NFR-6	Scalability	High level with increase in performance and it is one of the main concerns of iot domain.

CHAPTER 5

5. PROJECT DESIGN

5.1 DATAFLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow- there are no decision rules and no loops.

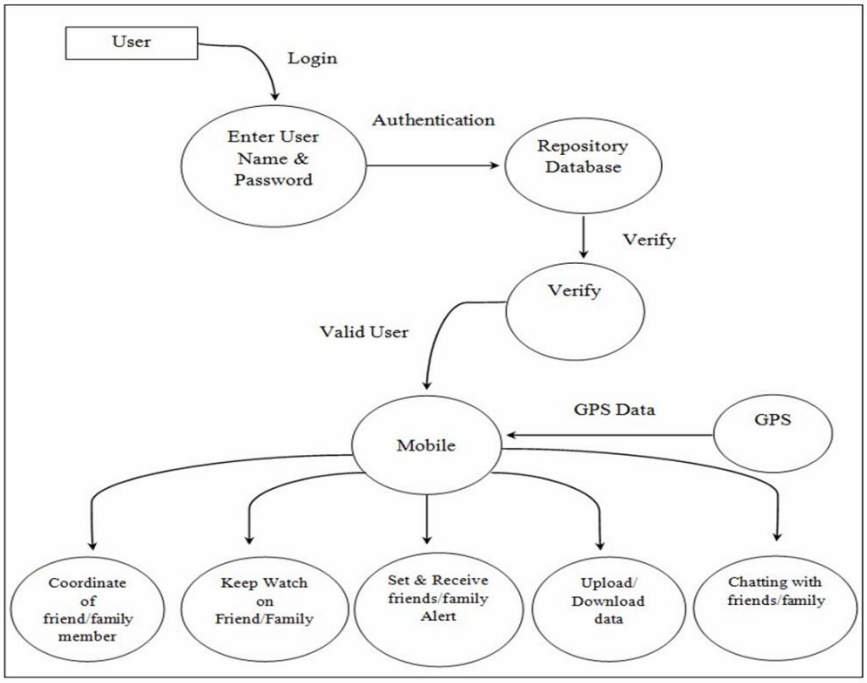


Fig 5.1 Detailed DFD Level 0

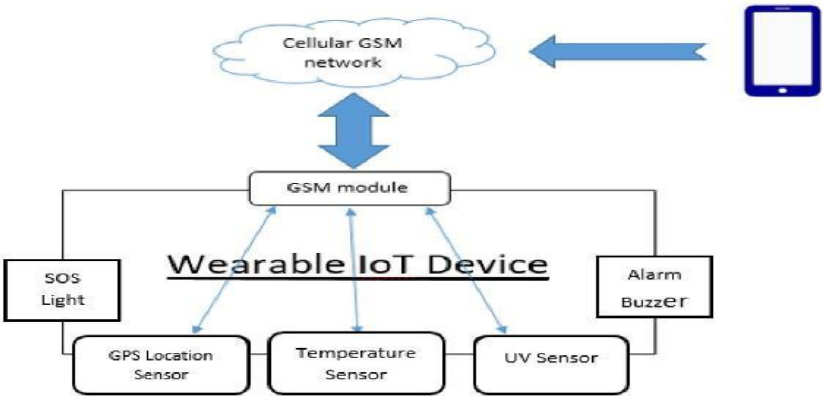


Fig 5.2 Flow Diagram

5.2 SOLUTION & TECHNICAL ARCHITECTURE

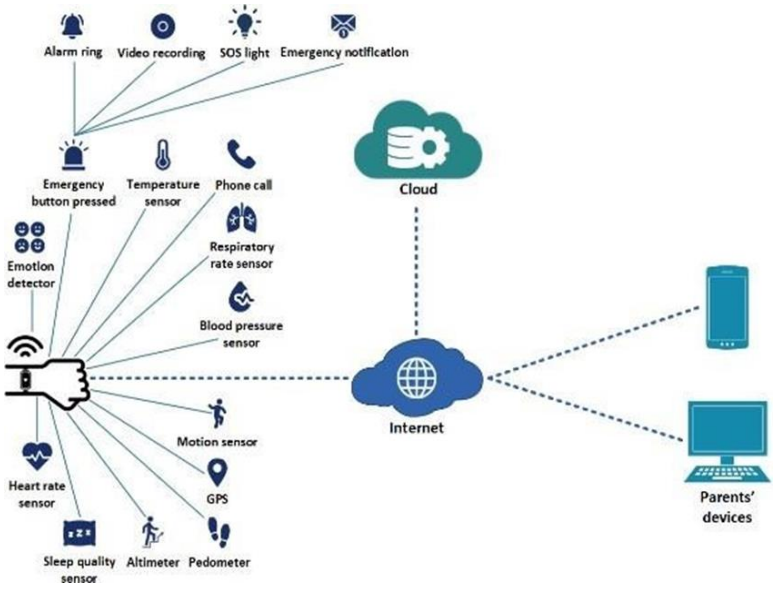


Fig 5.3 Technical Architecture

• In fact, Development of a safety gadget for children to ensure their protection without direct monitoring of the parents which includes various features such as GPS, Blood pressure sensor, Temperature sensor, Emergency alert etc.

• The IoT-enabled digital architecture integrates the Cloud, Mobile and GPS technology to precisely locate the geographical location of a child.

• In GPS, the enrolled device will get alert message or alerts when the device detect activities outside the specified geo fence.If any risk is felt, additional features like notification could be performed.

• In case of emergency, the alarm and SOS light will be activated by parents through their devices.

• The collection of data will be sent to the cloud over the internet for security process, analyze, monitor, store, access and retrieve data remotely.

• After that, the information indicating children's status, along within the range of reference value, notification and some suggestions will be sent to parent's devices.

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register for the application through Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can log into the application by entering email & password	High	Sprint-1
	Dashboard	USN-6	As a user, I can view the dashboard that consists of the current status and alert can also be seen	I can view the dashboard that consists of the current status and alerts can also be seen	High	Sprint-2
Customer (Web user)	Dashboard	USN-7	As a user, I can customize the dashboard according to my needs and preference	I can view the location which is stored in the database of the child via dashboard	Medium	Sprint-3
Customer Care Executive	Application	USN-8	As an customer care executive, I can solve the customer enquiries	I can solve the customer enquiries	High	Sprint-3
Administrator	Application	USN-9	As an admin, I can organise works, check the faults and correct it	I can organisw works and make correction in faults	High	Sprint-3

CHAPTER 6

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration (Parent Mobile User)	USN-1	Registering for an application, As a user we can register by entering our email, password and again we need to confirm the password	3	High	Sivasudhan A
Sprint-1	Login	USN-2	If we have register for the application as a user a confirmation mail will be received to our mail	3	High	Sharumathi S
Sprint-2	User Interface	USN-3	Using Facebook we can register for this application	3	Low	Sruthi R
Sprint-1	Data Visualization	USN-4	We can also register for the application through Gmail	3	Medium	Yamini K
Sprint-3	Login	USN-5	As a user, I can log into the application by entering email & password	3	Low	Sharumathi S
Sprint-1	Dashboard	USN-5	We need to be able to view the function that can perform	4	High	Sivasudhan A
Sprint-2	Notification	USN-1	Using minimum time we should be able to notify their parent and guardian	4	High	Yamini K
Sprint-1	Store data	USN-2	We need to continuously store location data into the database	3	Medium	Yamini K
Sprint-4	Web UI	USN-3	We all will need a friendly interface to view and access the resource easily	3	Medium	Sruthi R
Sprint-3	Registration (Parent Web User)	USN-1	By entering email and password we can log into the application as a user	3	High	Sivasudhan A

6.2 SPRINT DELIVERY SCHEDULE

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	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials organization = "0pycss"

deviceType = "weather_Device1"

deviceId = "weather_deviceid" authMethod = "token"

authToken = "(j!jK*nvh9OKQD9!dJ"

#api key {a-illza1-mbdxqo6z0s}

#api token {zSYzISuAWF&F_x7GkT}

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method":

authMethod, "auth-token": authToken} deviceCli =
ibmiotf.device.Client(deviceOptions)

#.....

except Exception as e:

print("Caught exception connecting device: %s" % str(e)) sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an
event of type

print("POWER ON ") print("CHECKING CONNECTION TO IBM
WATSON...")

time.sleep(2)

deviceCli.connect()
```

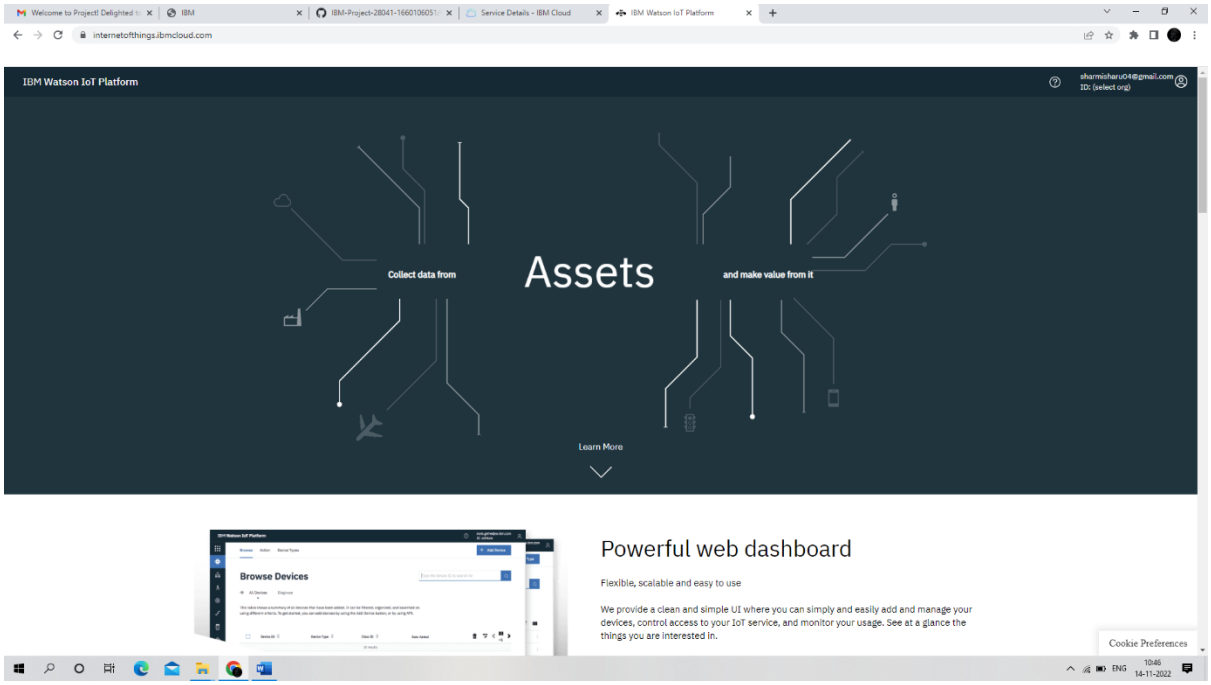
```
print("dear user ... welcome to IBM-IOT ")
print("You can know your child's live location and temperature ")
name=str(input("enter your child name:")) while True:
temperature=random.randint(20,50)
#random temperature for your child
latitude=random.uniform(10.781377,10.78643)
#random latitude for your child
longitude=random.uniform(79.129113,79.134014)
#random longitude for your child
a="Child inside the geofence"
b=" Child outside the geofence"
c="High temperature"
d="Low temperature"
x={'your_child_Zone':a}
y={'your_child_Zone':b}
z={'temp_condition':c}
w={'temp_condition':d}
data = { 'temp' : temperature, 'lat': latitude,'lon':longitude,'name':name }
#print data def
myOnPublishCallback():
print ("Published Temperature = %s C" % temperature, "latitude = %s %%" %
latitude,
"longitude = %s %%" % longitude, "to IBM Watson")
print("\n")
success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data, qos=0,
on_publish=myOnPublishCallback)
if latitude>=10.78200 and latitude<=10.786000 and longitude >=79.130000 and
longitude <=79.133000:
```

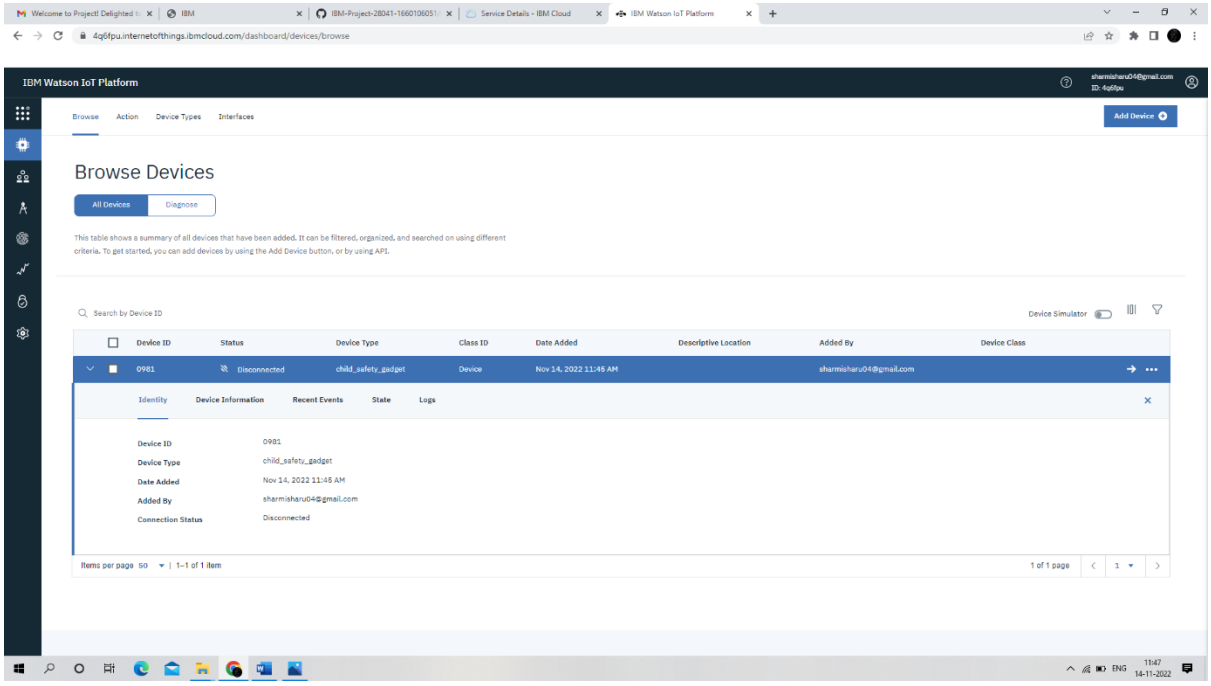
```
deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publish=myOnPublishCallback)

print(x) print("\n") else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=myOnPublishCallback)

print(y)

print("\n")

if (temperature>35):

deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=myOnPublishCallback)

print(z)

print("\n")

else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=myOnPublishCallback)

print(w)

print("\n")

if not success:

print("Not connected to IoTF")

print("\n")

time.sleep(3)

# Disconnect the device and application from the cloud deviceCli.disconnect()

data = { 'temp' : temperature, 'lat': latitude,'lon':longitude,'name':name } #print data def

myOnPublishCallback():

print ("Published Temperature = %s C" % temperature, "latitude = %s %%" % latitude,

"longitude = %s %%" % longitude, "to IBM Watson")

print("\n")
```

```
success = deviceCli.publishEvent("IoTSensorgpsdata", "json", data, qos=0,

on_publish=myOnPublishCallback)

if latitude>=10.78200 and latitude<=10.786000 and longitude >=79.130000 and
longitude

<=79.133000:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=x,qos=0,on_publish=m
yOnPublis hCallback)

print(x)

print("\n")

else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=y,qos=0,on_publish=m
yOnPublishCallback)

print(y)

print("\n")

if (temperature>35):

deviceCli.publishEvent("IoTSensorgpsdata","json",data=z,qos=0,on_publish=m
yOnPublishCallback)

print(z)

print("\n")

else:

deviceCli.publishEvent("IoTSensorgpsdata","json",data=w,qos=0,on_publish=
myOnPublishCallback)

print(w)

print("\n")

if not success:

print("Not connected to IoTF")

print("\n")

time.sleep(3)

# Disconnect the device and application from the cloud deviceCli.disconnect()
```

7.2 FEATURE 2

Creating IBM Cloud Services

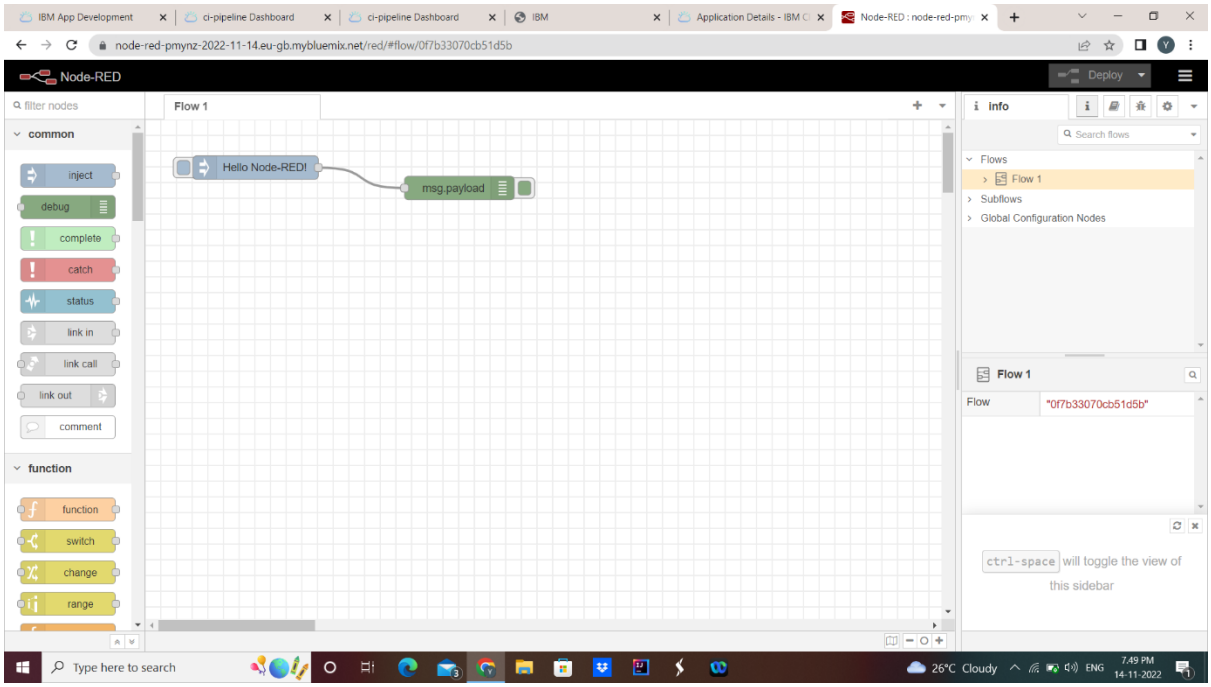


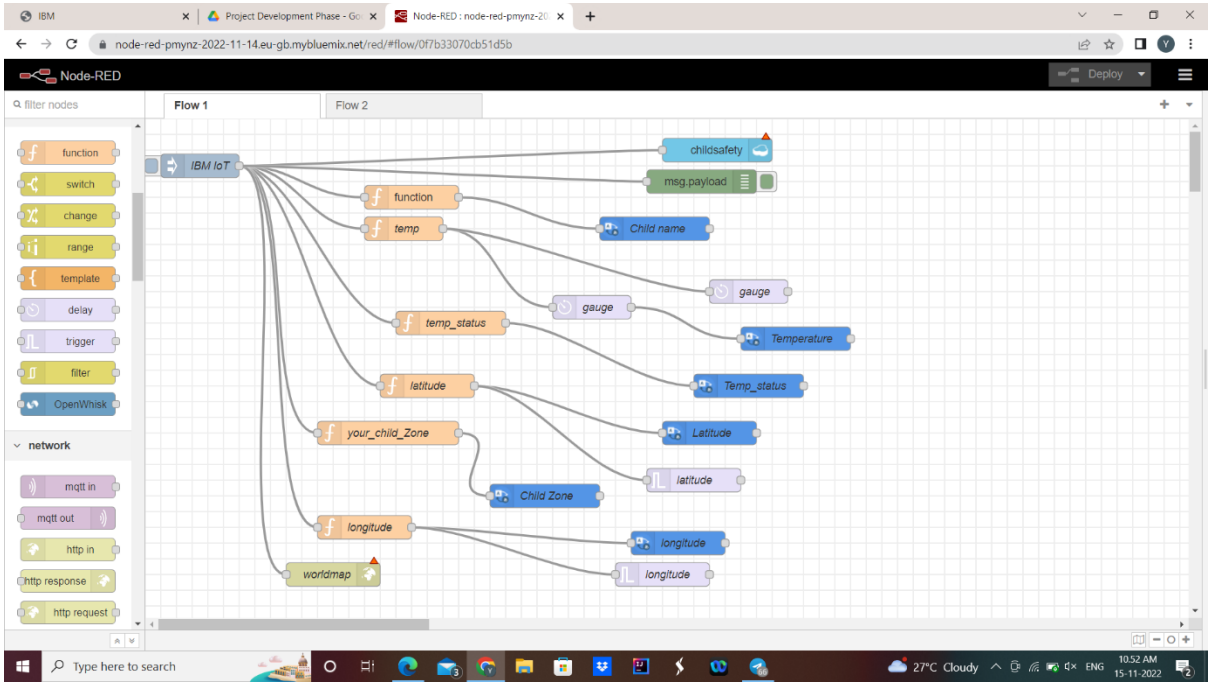


CHAPTER 8

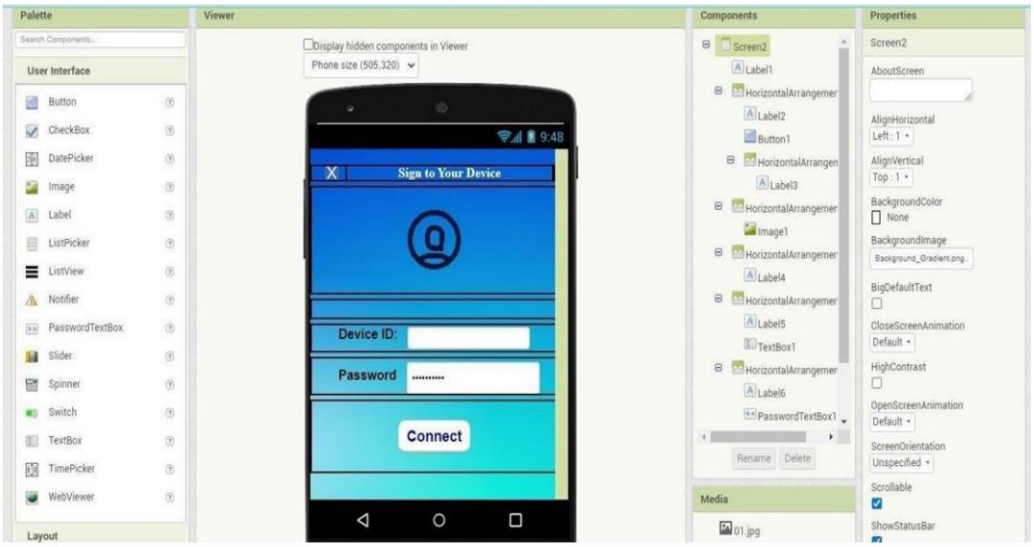
8. TESTING

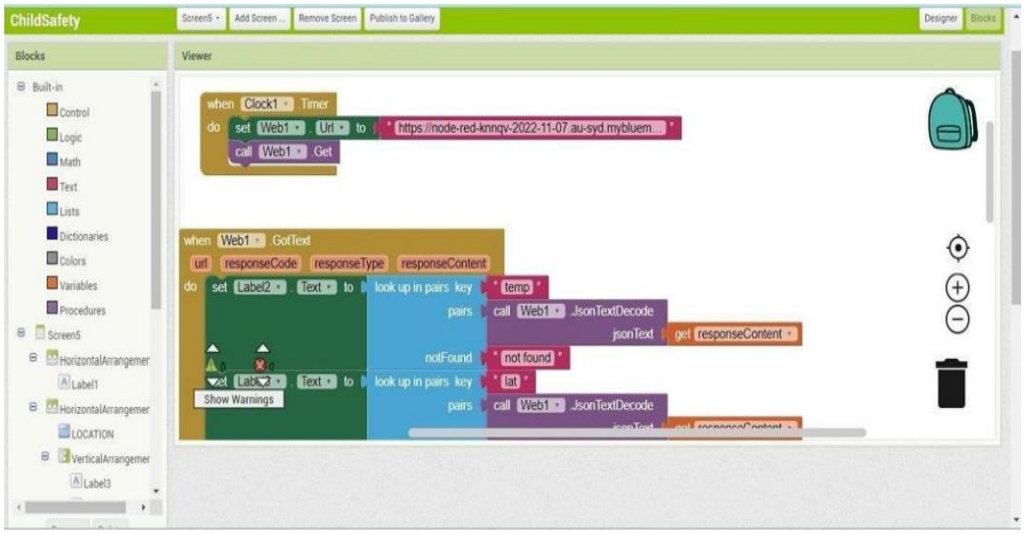
8.1 TEST CASES

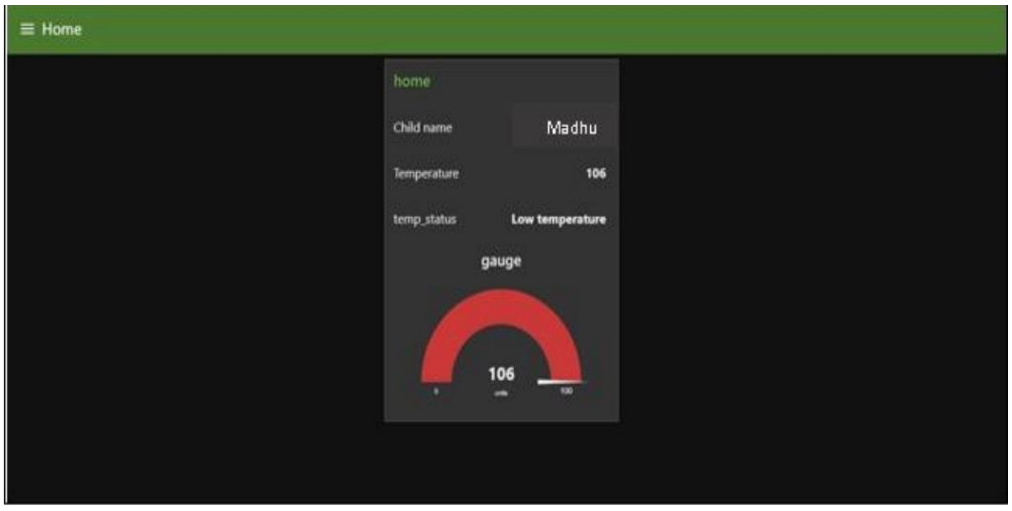




8.2 USER ACCEPTANCE TESTING





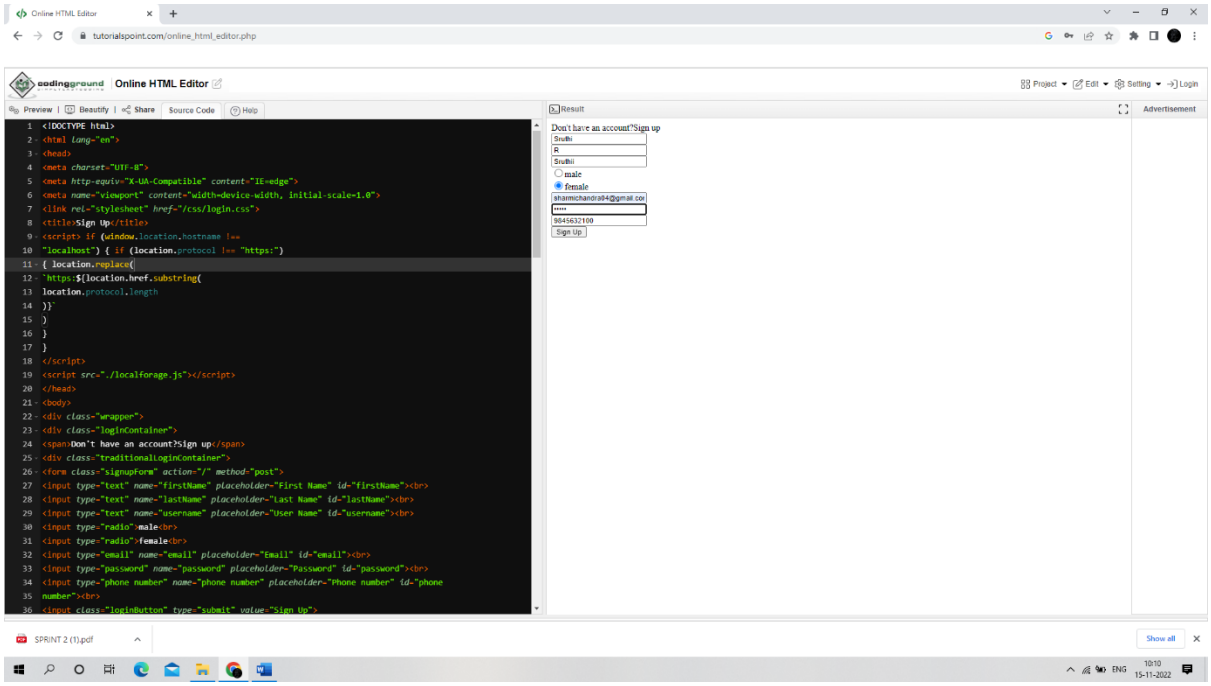


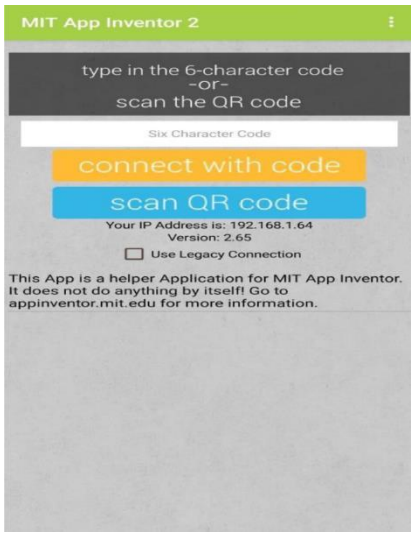
CHAPTER 9

9. RESULTS

9.1 PERFORMANCE METRICS

It is being used as it allows the correct sample of respondents to be selected due to which becomes convenient to obtain results. Besides, the results offered are affordable and usable. Since the respondents are properly chosen, the results tend to be more accurate, precise and reliable.





CHAPTER 10

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

In our system, we provide an environment where this problem can be resolved in an efficient manner. It makes parents to easily monitor their children in real time just like staying beside them as well as focusing on their own career without any manual intervention.

DISADVANTAGES

It can be easily removed or damaged while playing and by any intruders. This requires internet connectivity to get monitored and to notify alert messages to parents.

CHAPTER 11

11. CONCLUSION

The word Future resembles the word Children. As Dr. A.P.J Abdul Kalam’s words “Youngsters are thefuture pillars of one’s nation", today's children are tomorrow's youngsters, preserving their dreams and life for a better future is necessary. Therefore, each and every parent should take care of their own children, without letting them to fall into the dark world of abusements, which entirely ruin them physically, mentally and emotionally destroying our future. Hence, considering the importance of our future, our project makes it easy for parents to track their children and to visually monitor them on regular basis, which makes them ensure the safety of their children and reduces the rate of incidents of child abuse. The system is equipped with GSM and GPS modules for sending and receiving call, SMS between safety gadget and parental phone.The system also consists of Wi-Fi module used to implement IoT and send all the monitored parameters to the cloud for android app monitoring on parental phone. Panic alert system is used during panic situations alerts are sent to the parental phone, seeking for help also the alert parameters are updated to the cloud. Boundary monitoring system is implemented on safety gadget with the help of BEACON technology, as soon as the safety gadget moves far away from the BLE listener gadget an alert is provided to itself.This wearable device has a superior mode for viewing and locating the children\'s whereabouts with correct latitude and longitude, which is especially useful when using Google maps. This could assist to reduce the number of attacks on children while also making them feel protected and secure. The major goal of this project is to create a device that protects youngsters from risky circumstances while also assisting them in combating them.

CHAPTER 12

12. FUTURE SCOPE

A camera module for surveillance of the child's surrounds can be added to improve the system's performance. It's also possible to do it with a Raspberry Pi and Lily pad. It is possible to develop a more energy-efficient type that can keep the battery for a longer period of time. This system can be further enhanced by installation of mini camera inside smart gadget for better security so that live footage can be seen on parental phone during panic situations. The system can be modified by installation of small solar panels for charging the battery of smart gadget to gain maximum battery backup. For surveillance of the child's surroundings, to get a clearer picture of the location, this wearable can also contain a camera module incorporated in it. The camera will be collecting information in the same manner as the GPS module. It will be on stand by conserving power waiting for the particular keyword "SNAPSHOT" to be sent from the user's smart phone to the GSM shield will activate the camera to start clicking a snapshot of the surrounding and save the file temporarily on the external micro SD card. After which Arduino UNO will access the saved image from the micro SD storage and transfer it to the GSM module which send it to the user via SMS/MMS text. This system also requires network connectivity, satellite communication, and high-speed data connection when we use web camera and GPS to lively monitor. It is difficult to monitor when there occurs any hindrance to satellite communication or any network issue. There also occurs time delay in video streaming through the server. Hence in the future, these issues can be overcome by using Zigbee concept or accessing the system without internet and using high-speed server transmission.

SOURCE CODE

```
<!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/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>Don't have an account?Sign up</span>

<div class="traditionalLoginContainer">

<form class="signupForm" action="/" method="post">

<input type="text" name="firstName" placeholder="First Name" id="firstName"><br>

<input type="text" name="lastName" placeholder="Last Name" id="lastName"><br>

<input type="text" name="username" placeholder="User Name" id="username"><br>

<input type="radio">male<br>

<input type="radio">female<br>

<input type="email" name="email" placeholder="Email" id="email"><br>

<input type="password" name="password" placeholder="Password" id="password"><br>

<input type="phone number" name="phone number" placeholder="Phone number" id="phone

number"><br>

<input class="loginButton" type="submit" value="Sign Up">

</form>

</div>

<div class="loginWithFireContainer">
```

```
</div>

<a class="hyperLink" href="<”https://appinventor.mit.edu/”>Account already exists? Login -

></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("/fireoauth.html", "Fire OAuth", 450, 600)-->

<!--})-->

<!--</script>-->

<!--<script>-->

<!--// this.Website's Scripts / App Logic-->

<!--document.querySelector(".signupForm").addEventListener("submit", async (e) =>-->

<!--
{e.preventDefault()letfirstName=document.getElementById("firstName").value
-->

<!--let lastName = document.getElementById("lastName").value let username =-
->

<!--document.getElementById("username").value let email =-->

<!--document.getElementById("email").value let password =-->

<!--document.getElementById("password").value-->

<!--let profilePic =-->

<!--
`https://avatars.dicebear.com/api/adventurerneutral/${firstName}${lastName}.s
vg?backgrou

ndColor=variant01-->

<!--` let data = { firstName, lastName, username, email, password, profilePic-->

<!--}-->

<!--await localforage.setItem("userData", data) let database-->

<!--= await localforage.getItem("userDatabase") if (database-->

<!--== null) { database = []-->

<!--}-->

<!--database.push(data) await-->

```
<!--localforage.setItem("userDatabase", database)-->
<!--window.location.href = "/"-->
<!--})-->
<!--</script>-->
</body>
</html>
```

GITHUBLINK:

https://github.com/IBM-EPBL/IBM-Project-10518-1659184237