Project Report

Date	19 November 2022
Team ID	PNT2022TMID06857
Project Name	IoT Based Safety Gadget for Child Safety Monitoring & Notification

1. INTRODUCTION

1.1 Project Overview

This project aims to create a method through which the parents can create a geofence for their children and mark it as a safe zone. Whenever the child goes out of that zone the parents will get notified that the child has left the safe zone, after which the parent can track the exact location of the child.

1.2 Purpose

By the end of this project, we will:

- Gain knowledge of Watson IoT Platform.
- Connecting IoT devices to the Watson IoT platform and exchanging the sensor data.
- Gain knowledge on Cloudant DB
- Gain Knowledge of geofence
- Creating a Web Application through which the user interacts with the device

Project Flow:

- The GPS coordinates of the child will be sent to the IBM IoT platform
- Location can be viewed in the Web Application
- A parent can create a geofence in the web application
- The web application will check if the child is inside or outside the geofence
- Notifies the parents if the child goes out of the geofence

To accomplish this, we must complete all the activities and tasks listed below:

- Create and configure IBM Cloud Services
  - Create IBM Watson IoT Platform

    - Create a device & configure the IBM IoT Platform
    - Create Node-RED service
    - Create a database in Cloudant DB to store location data
- Develop a web Application using Node-RED Service.
    - Develop the web application using Node-RED o Integrate the geofence & google map
- Develop a python script to publish the location details to the IBM IoT platform

2. LITERATURE SURVEY

2.1 Existing problem

The existing solution problems are that they are costly and not reliable.

2.2 References

2019- Child Safety Wearable Device

V .Lavanya, C.Meenambigai, M.Suriyaa, S.Kavya

Department of Computer Science and Engineering, Velammal College of Engineering and Technology, Madurai

2019- Multi-sensor Wearable for Child Safety

Ushashi Chowdhury, Pranjal Chowdhury, Sourav Paul, Anwesha Sen, Partho Protim Sarkar, Shubhankur Basak, Abari Bhattacharya

Department of Computer Science and Engineering, Institute of Engineering and Management

2.3 Problem Statement Definition

- In today’s world there are lots of ongoing crime related to children like child trafficking and kidnapping.

- Statistics will show how large the number of these crimes are but there are bound to be more cases that go unreported and it’s painful to know that many of those cases remain unsolved to this day.

- So, parents want a reliable way through which they can ensure their child’s safety when they are not under their supervision.

- Thus, our project aims to create a method through which the parents will be able to track the location of their child.

- How it works is that, after they leave their children in school or parks, they can create a geofence around that particular location.

- The child’s location will be continuously monitored to check if the child is still inside the safe zone.

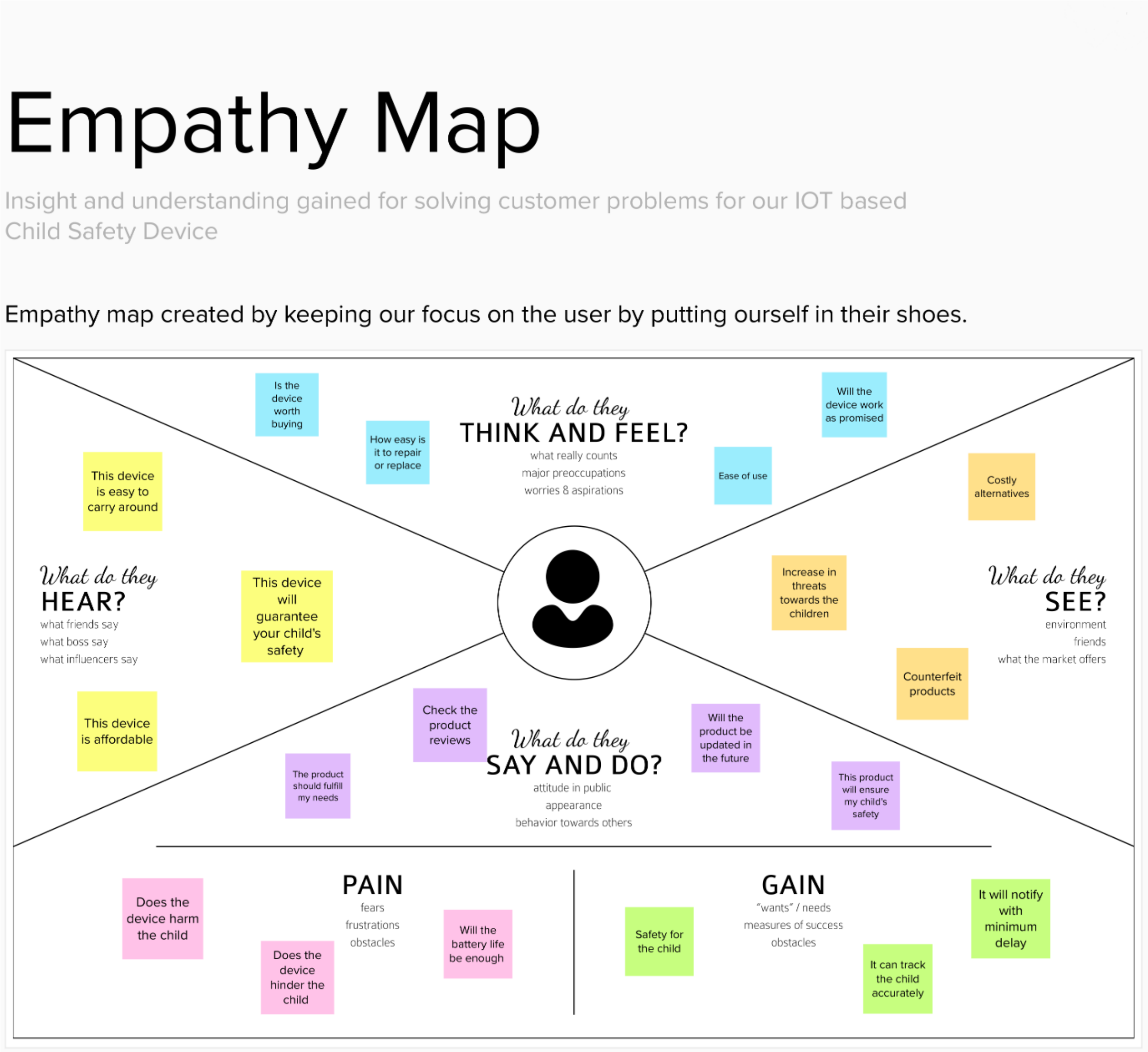
- Whenever the child will go out of that geofenced area i.e. the safe area, the parents or the caretaker will be notified through the app that the child is out of the safe zone.

- The parent or the caretaker can then act to ensure the child’s safety.

- The entire location data will be stored in the database.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Empathy Map

Insight and understanding gained for solving customer problems for our IOT based Child Safety Device

Empathy map created by keeping our focus on the user by putting ourself in their shoes.

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Is the device worth buying

How easy is it to repair or replace

Ease of use

Will the device work as promised

What do they HEAR?
what friends say
what boss say
what influencers say

This device is easy to carry around

This device will guarantee your child's safety

This device is affordable

What do they SEE?
environment
friends
what the market offers

Costly alternatives

Increase in threats towards the children

Counterfeit products

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

Check the product reviews

The product should fulfill my needs

Will the product be updated in the future

This product will ensure my child's safety

PAIN
fears
frustrations
obstacles

Does the device harm the child

Does the device hinder the child

Will the battery life be enough

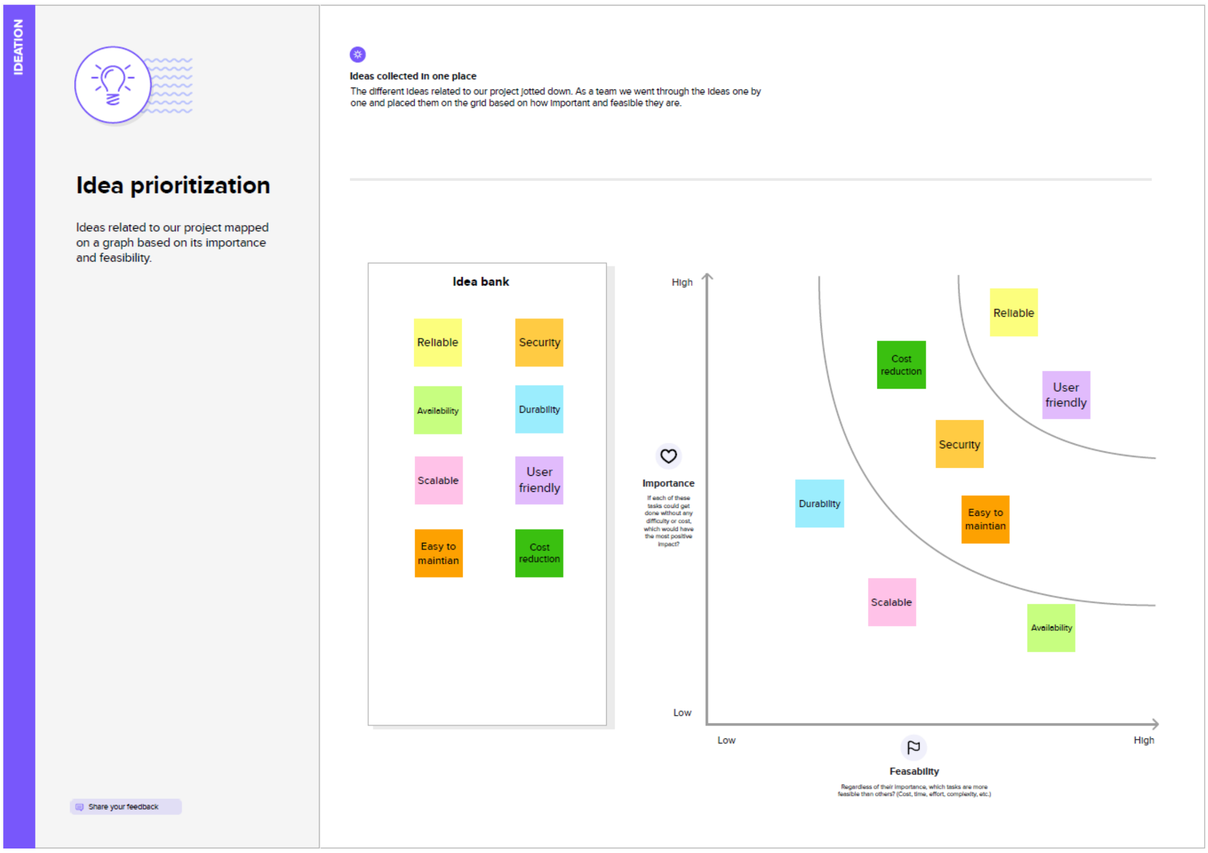
GAIN
"wants" / needs
measures of success
obstacles

Safety for the child

It can track the child accurately

It will notify with minimum delay

3.2 Ideation & Brainstorming



IDEATION

Idea prioritization

Ideas related to our project mapped on a graph based on its importance and feasibility.

Ideas collected in one place

The different ideas related to our project jotted down. As a team we went through the ideas one by one and placed them on the grid based on how important and feasible they are.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasability

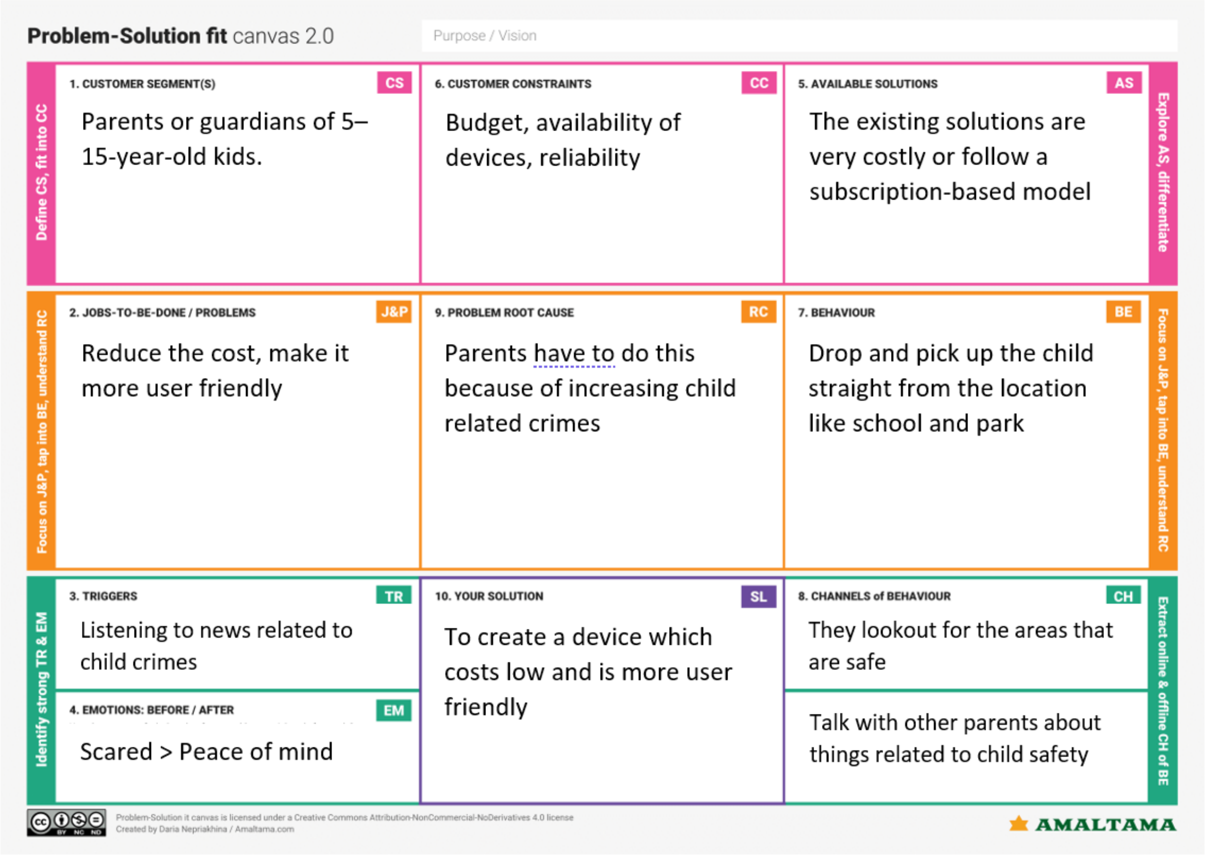
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	In today's world with lots of ongoing crime related to children, parents want a reliable way through which they can ensure their child's safety.
2.	Idea / Solution description	To create an IOT device through which the parents can set geofence for their child.
3.	Novelty / Uniqueness	To send accurate location data and send notification as soon as child goes out of the safe zone.
4.	Social Impact / Customer Satisfaction	• Reduce in crime rate related to children. • Peace of mind for parents.
5.	Business Model (Revenue Model)	• This device can be used for school going children.

		• The device can also be used for tracking a vehicle in case of a theft. • The device can be modified for the use of women.
6.	Scalability of the Solution	• The battery life of the device can be improved by a lot. • The location accuracy can be improved. • The size of the device can be reduced.

3.4 Problem Solution fit



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Parents or guardians of 5–15-year-old kids.

6. CUSTOMER CONSTRAINTS CC

Budget, availability of devices, reliability

5. AVAILABLE SOLUTIONS AS

The existing solutions are very costly or follow a subscription-based model

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Reduce the cost, make it more user friendly

9. PROBLEM ROOT CAUSE RC

Parents have to do this because of increasing child related crimes

7. BEHAVIOUR BE

Drop and pick up the child straight from the location like school and park

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Listening to news related to child crimes

4. EMOTIONS: BEFORE / AFTER EM

Scared > Peace of mind

10. YOUR SOLUTION SL

To create a device which costs low and is more user friendly

8. CHANNELS of BEHAVIOUR CH

They lookout for the areas that are safe

Talk with other parents about things related to child safety

Extract online & offline CH of BE

AMALTAMA

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

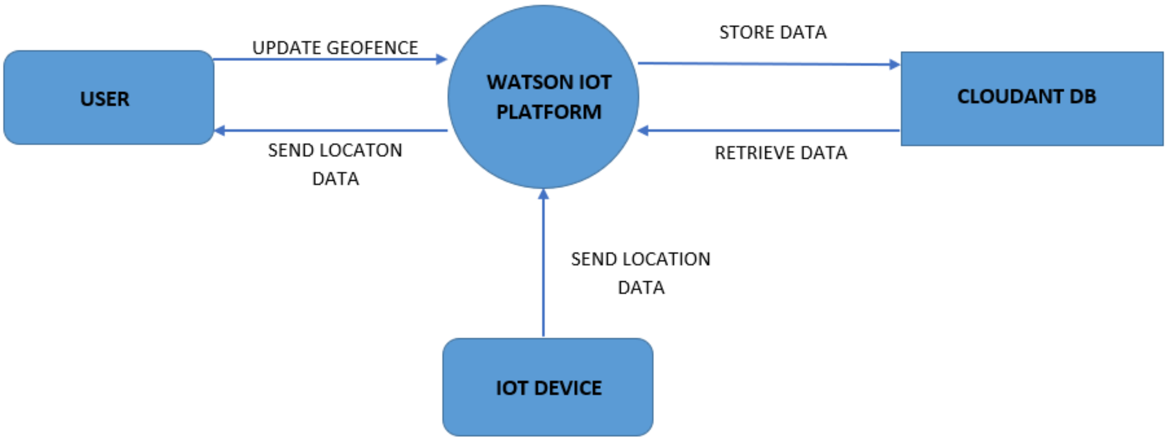
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication	Confirming if the email id and password combination is correct
FR-4	User Verification	Sending the verification mail whenever the account is logged in
FR-5	Diverse Connectivity	Wi-Fi, LTE
FR-6	Compliance requirements	The solution complies with related regulations

4.2 Non-Functional requirements

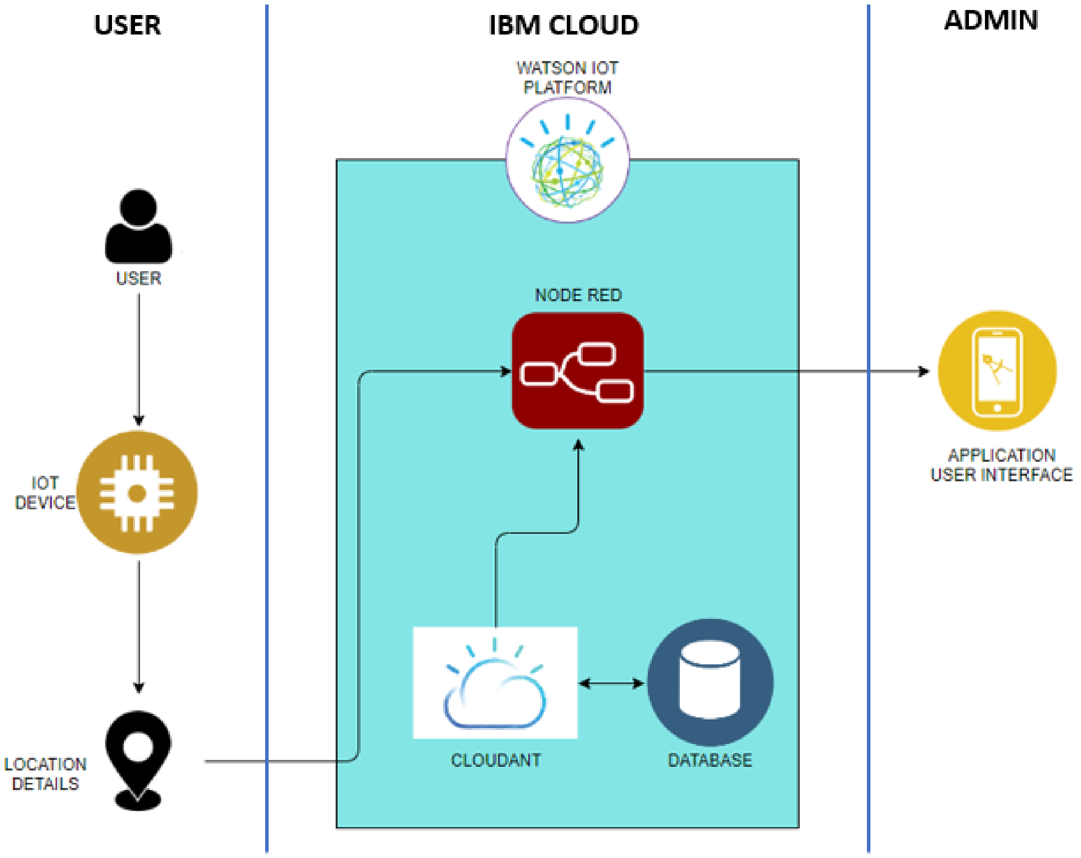
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Users can easily navigate its interface. • Users can easily determine what a feature is and what it can do
NFR-2	**Security**	• Systems require users to create accounts to access applications that store information and display profiles • The application will not grant access until the user creates a strong password.
NFR-3	**Reliability**	Highly reliable functions with the same or similar efficiency after extensive use.
NFR-4	**Performance**	The device performs well i.e. features of a system are functioning well based on what a developer predicted.
NFR-5	**Availability**	Platforms must offer exceptionally high availability
NFR-6	**Scalability**	Platform is fully able to support a small, initial implementation, but also able to scale out as our business needs grow.

5. PROJECT DESIGN

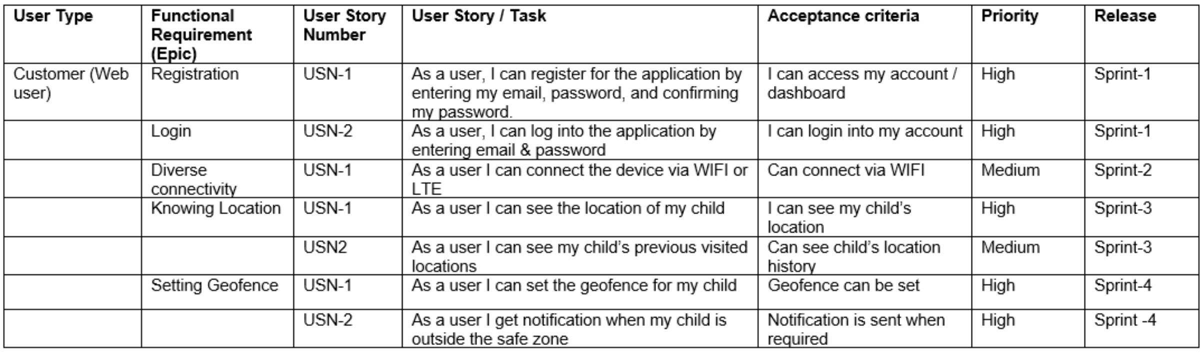
5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web 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
	Login	USN-2	As a user, I can log into the application by entering email & password	I can login into my account	High	Sprint-1
	Diverse connectivity	USN-1	As a user I can connect the device via WIFI or LTE	Can connect via WIFI	Medium	Sprint-2
	Knowing Location	USN-1	As a user I can see the location of my child	I can see my child’s location	High	Sprint-3
		USN2	As a user I can see my child’s previous visited locations	Can see child’s location history	Medium	Sprint-3
	Setting Geofence	USN-1	As a user I can set the geofence for my child	Geofence can be set	High	Sprint-4
		USN-2	As a user I get notification when my child is outside the safe zone	Notification is sent when required	High	Sprint -4

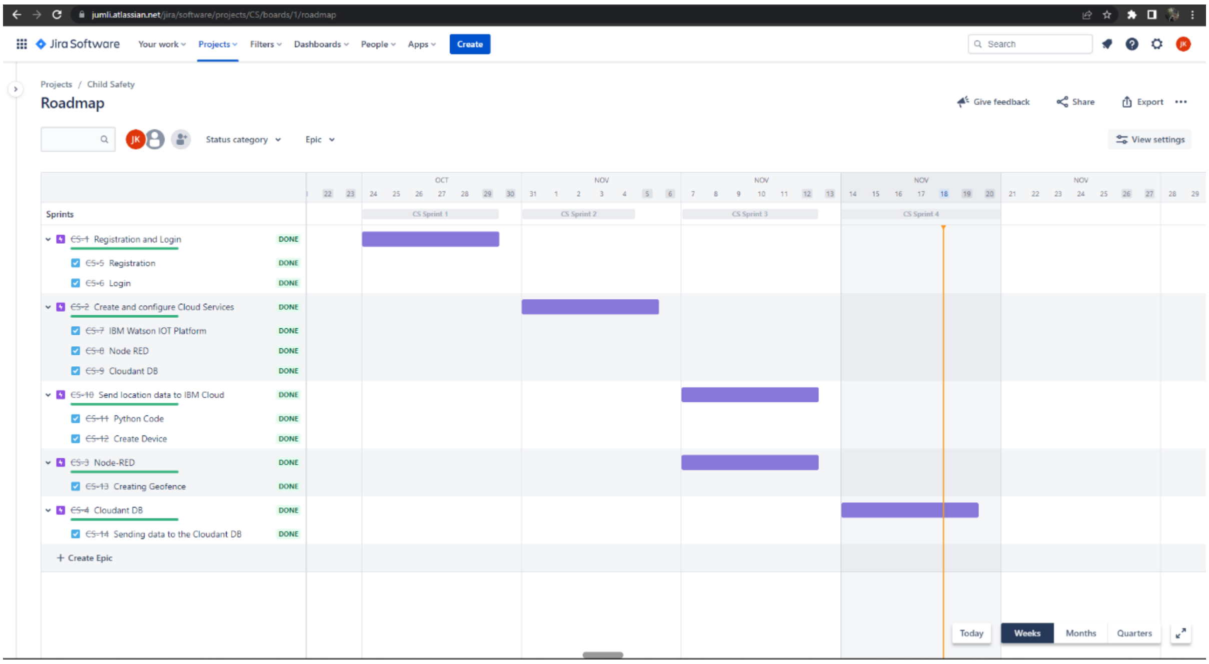
6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	4
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	2	High	4
Sprint-2	Diverse connectivity	USN-1	As a user I can connect the device via WIFI or LTE	2	Medium	4
Sprint-3	Knowing Location	USN-1	As a user I can see the location of my child	3	High	4
Sprint-3		USN2	As a user I can see my child's previous visited locations	2	Medium	4
Sprint-4	Setting Geofence	USN-1	As a user I can set the geofence for my child	3	High	4
Sprint -4		USN-2	As a user I get notification when my child is outside the safe zone	2	High	4

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	4	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	2	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	5	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	5	19 Nov 2022



Projects / Child Safety

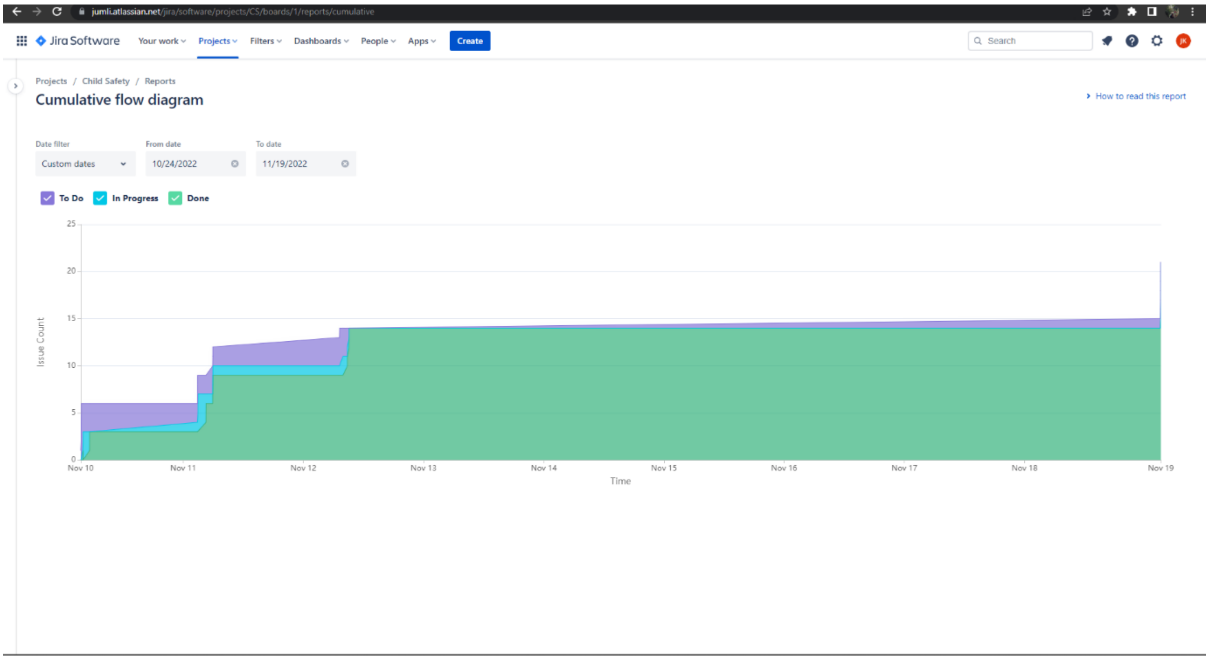
Roadmap

Sprints

- ~~CS-1~~ Registration and Login DONE
  - ☑ ~~CS-5~~ Registration DONE
  - ☑ ~~CS-6~~ Login DONE
- ~~CS-2~~ Create and configure Cloud Services DONE
  - ☑ ~~CS-7~~ IBM Watson IOT Platform DONE
  - ☑ ~~CS-8~~ Node RED DONE
  - ☑ ~~CS-9~~ Cloudant DB DONE
- ~~CS-10~~ Send location data to IBM Cloud DONE
  - ☑ ~~CS-11~~ Python Code DONE
  - ☑ ~~CS-12~~ Create Device DONE
- ~~CS-3~~ Node-RED DONE
  - ☑ ~~CS-13~~ Creating Geofence DONE
- ~~CS-4~~ Cloudant DB DONE
  - ☑ ~~CS-14~~ Sending data to the Cloudant DB DONE

+ Create Epic

6.3 Reports from JIRA



jumli.atlassian.net/jira/software/projects/CS/boards/1/reports/cumulative

Projects / Child Safety / Reports

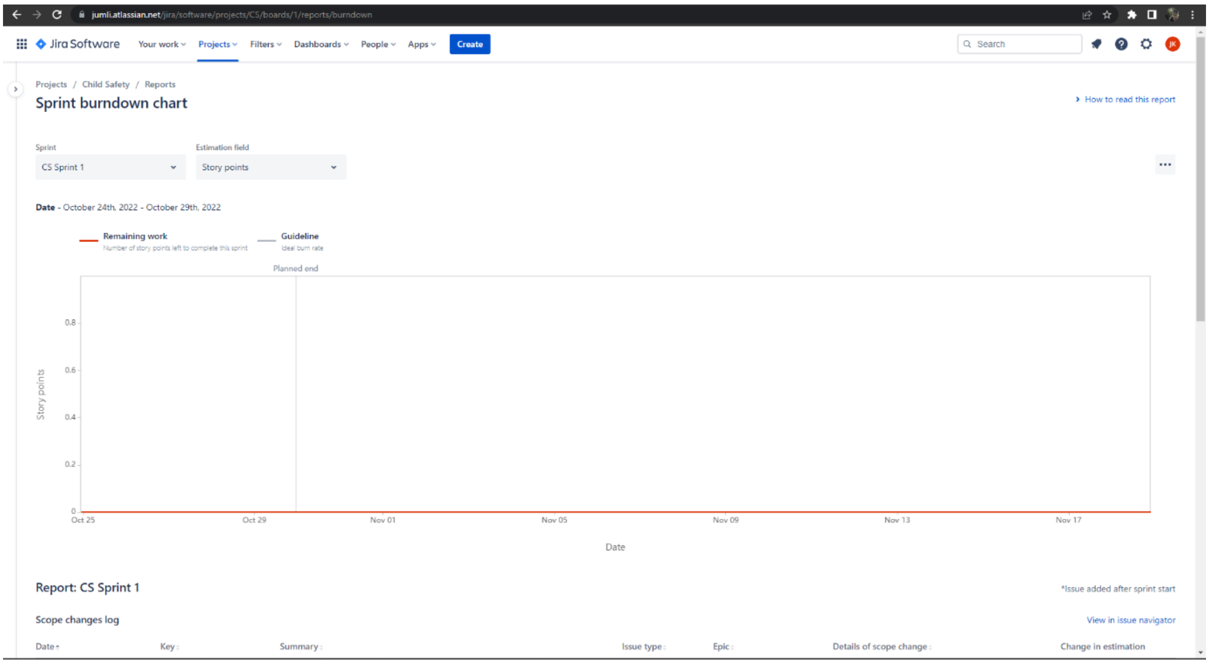
Cumulative flow diagram

Date filter: Custom dates

From date: 10/24/2022

To date: 11/19/2022

☑ To Do ☑ In Progress ☑ Done



jumli.atlassian.net/jira/software/projects/CS/boards/1/reports/burndown

Projects / Child Safety / Reports

Sprint burndown chart

Sprint: CS Sprint 1

Estimation field: Story points

Date - October 24th, 2022 - October 29th, 2022

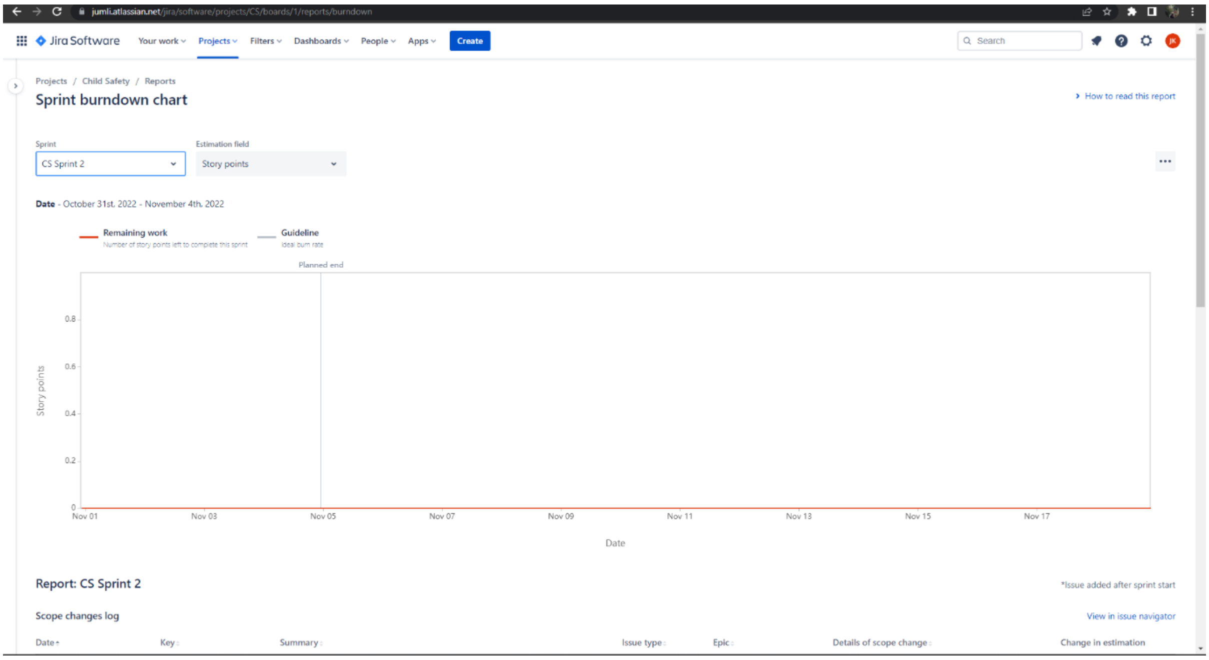
Report: CS Sprint 1

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation



Sprint burndown chart

Sprint: CS Sprint 2

Estimation field: Story points

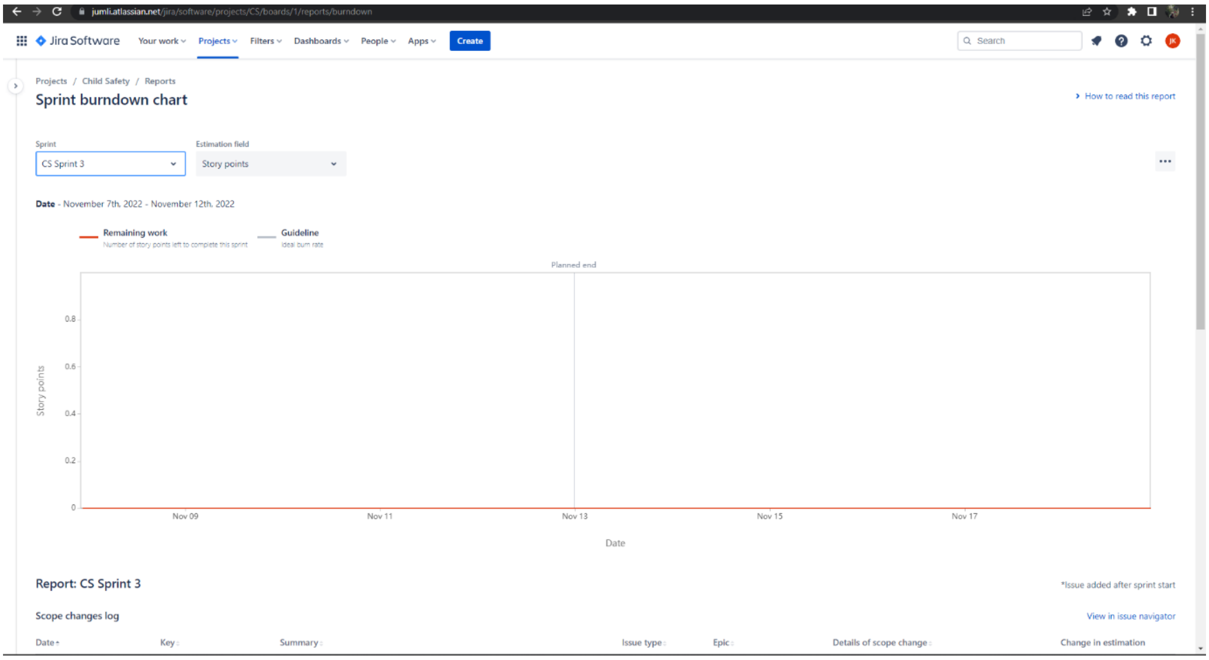
Date - October 31st, 2022 - November 4th, 2022

Report: CS Sprint 2

*Issue added after sprint start

Scope changes log

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation



Sprint burndown chart

Sprint: CS Sprint 3

Estimation field: Story points

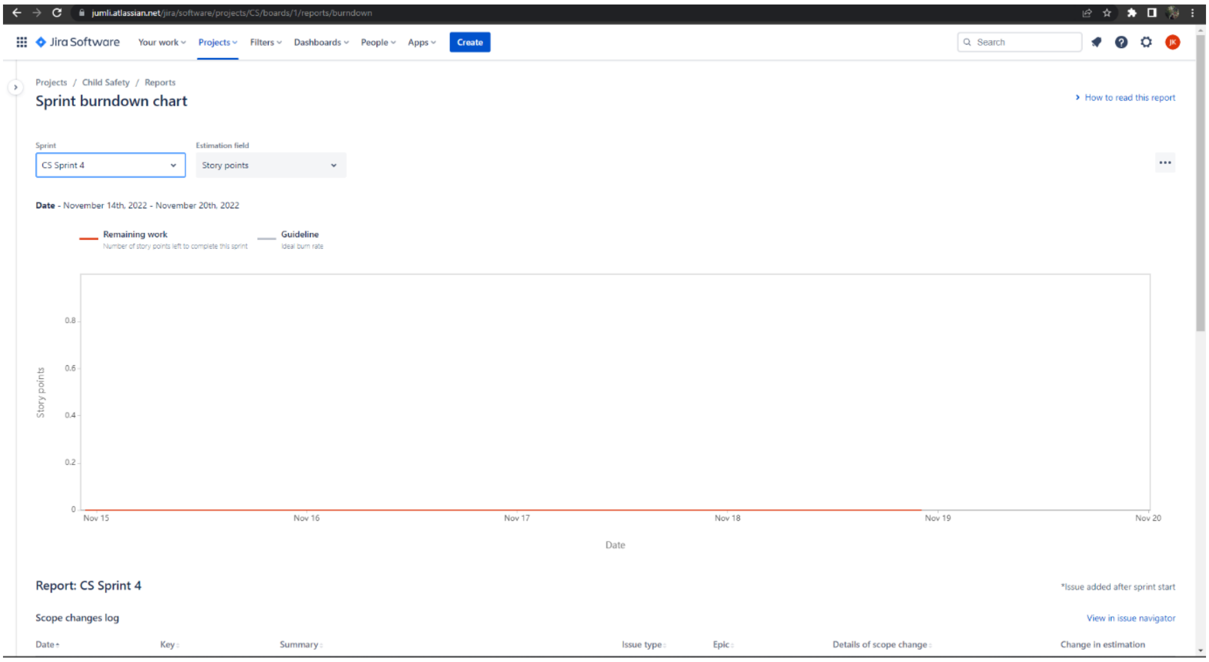
Date - November 7th, 2022 - November 12th, 2022

Report: CS Sprint 3

*Issue added after sprint start

Scope changes log

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation



jumli.atlassian.net/jira/software/projects/CS/boards/1/reports/burndown
Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
JK

Projects / Child Safety / Reports

Sprint burndown chart

› How to read this report

Sprint: CS Sprint 4

Estimation field: Story points

Date - November 14th, 2022 - November 20th, 2022

Report: CS Sprint 4

*Issue added after sprint start

Scope changes log

View in issue navigator

Date	Key	Summary	Issue type	Epic	Details of scope change	Change in estimation

7. CODING & SOLUTIONING

7.1 Feature 1(Login)

CSS

Dashboard.css

```
html, body {
   margin: 0;
   padding: 0;
   font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI',
Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;


}
.wrapper {
   display: flex;
   flex-direction: column;
   height: 100vh;
   margin: 0 auto;
}
.header {
   display: flex;
   justify-content: space-between;
   align-items: center;
   padding: 1rem;
   background-color: #f1f1f1;
}
.heading {
   font-size: 1.3rem;
}
```

```
.right {
    display: flex;
    align-items: center;
    gap: 0.5rem;
}
.actionCenter {
    display: flex;
    justify-content: flex-end;
    align-items: center;
    gap: 1rem;
    padding: 0.5rem 0;
}

.actionCenter .action {
    padding: 0.5rem 0.8rem;
    border: 1px solid #44444444;
    border-radius: 12px;
    margin-right: 0.5rem;
    cursor: pointer;
}

.actionCenter .action:hover {
    background-color: #0070f3;
    color: white;
}
```

```
.childCardContainer {
    display: flex;
    flex-wrap: wrap;
    padding: 0.5rem 1rem;
}

.childCardContainer .childCard {
    display: flex;
    flex-direction: column;
    border: 1px solid #44444444;
    border-radius: 12px;
    max-width: 300px;
    font-size: 1.4rem;
}

.childCardContainer .childCard .childCardHeader {
    display: flex;
    gap: 3rem;
    margin: 0.1rem 0.1rem 0.8rem;
    padding: 1rem 0.8rem 0;
}

.childCardContainer .childCard .actions {
    background-color: #0070f3;
    padding: 2rem 0.5rem 1rem;
    color: white;
```

```
   display: flex;
   justify-content: space-around;
   border-radius: 12px;
   border-top-left-radius: 0;
   border-top-right-radius: 0;
}

.childCardContainer .childCard .actions span {
   border: 1px solid #0070f3;
   background-color: white;
   color: #0070f3;
   padding: 0.3rem 0.9rem;
   border-radius: 12px;
   cursor: pointer;
}
```

Login.css

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

```
.wrapper {
        height: 100%;
        display: flex;
        align-items: center;
        justify-content: center;
}

.loginContainer {
        display: flex;
        flex-direction: column;
        gap: 1rem;
        min-width: 25rem;
        padding: 1rem 3rem;
        border: 1px solid #44444444;
        box-shadow: 0px 3px 2px 1px #44444444;
        border-radius: 8px;
}

.loginContainer span {
        text-align: center;
        font-size: 3rem;
        font-weight: 500;
        margin: 1rem 1rem 3rem;
}

.traditionalLoginContainer form {
        display: flex;

        flex-direction: column;

        align-items: center;

        justify-content: center;

}

.traditionalLoginContainer :is(input[type="text"], input[type="password"], input[type="email"]) {

        margin: 0.3rem;

        padding: 0.3em 0.5em;

        border: 1px solid #44444444;

        border-radius: 5px;

        outline: none;

        min-width: 200px;

        font-size: 1.3rem;

}

.traditionalLoginContainer .loginButton {

        background-color: #0070f3;

        font-size: 1.6rem;

        padding: 0.2em 0.8em;

        color: white;

        margin: 0.4rem;

        border: none;

        border-radius: 5px;

  cursor: pointer;

        margin-top: 2rem;
```

```
}

.traditionalLoginContainer .loginButton:hover {
   background-color: #0071f3d6;
}

.loginWithFireContainer {
        display: grid;
        display: -ms-grid;
        place-items: center;
}

.fire {
        background-color: #f8f9fa;
        border: 1px solid #3c404321;
        border-radius: 4px;
        color: #3c4043;
        font-family: arial, sans-serif;

        margin: 11px 4px;
        padding: 0.4em 0.8em;
        line-height: 27px;

        min-width: 54px;
        text-align: center;
        cursor: pointer;
        user-select: none;
        font-size: 1.3rem;
        font-weight: 500;
}

.hyperLink {
        text-decoration: none;
  text-align: center;
  font-size: 1.2rem;
  color: #0070f3;
  font-weight: 400;
}

@media screen and (max-width: 480px) {
  .loginContainer {
    border: none;
    box-shadow: none;
    min-width: fit-content;
    min-width: -moz-fit-content;
    min-width: -webkit-fill-available;
    padding: 1rem;
  }
}
```

Dashboard.html

```
<!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 bgcolor="#A5CBC6">
    <div class="wrapper">
        <div class="header">
            <span class="heading">Dashboard</span>
            <span class="right">
                <span class="username">Hello User</span>
                <span>
                    <img class="profilePic"
src="https://avatars.dicebear.com/api/avataaars/asdfasdfds.svg" alt="User
Profile" height="30" width="30">
                </span>
            </span>
        </div>

        <div class="actionCenter">
            <div class="action">
                <span>Create Child Card</span>
            </div>
            <div class="action">
        <span class="logout">Log out</span>
      </div>
    </div>

    <div class="childCardContainer">
      <div class="childCard">
        <div class="childCardHeader">
          <span>Varun</span>
          <span>Age 13</span>
        </div>
        <div class="actions">
          <span>View</span>
          <span>GeoFence</span>
        </div>
      </div>
    </div>
  </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>
```

Fireoauth.html

```
<!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 bgcolor="#A5CBC6">
  <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=
${chatRoomId}&url=${originURL}`
        let a = document.getElementById("authorizeOverLink")
        a.href = url
    }

    async function fire() {
        NProgress.set(0.4)
        let { sessionId, chatRoomId } = await getSessionID()

        if(sessionId === undefined || chatRoomId === undefined || sessionId
=== null || chatRoomId === null) {
            pageTitle.innerHTML = "Something went wrong"
            subTitle.innerHTML = "Please try again later"
            return
        }

        setOpacity("1")
```

```
      NProgress.done()
      let data = {
        sessionId,
        url: encodeURIComponent(window.location.origin)
      }
      data = JSON.stringify(data)
      generateQR(data)
      changeHREF({sessionId, chatRoomId})
      socket.emit("join room", sessionId)
    }

    fire()

    socket.on("trusted token", (token) => {

      let data = {}
      data.success = true
      data.token = token

      broadCastingChannel.postMessage(data)

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

```
</html>
```

Index.html

```
<!DOCTYPE html>
<html lang="en" style="height: 100%; margin: 0;">
    <head>
        <meta charset="UTF-8" />
        <meta name="description" content="The Home Page after Logged In" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>IOT Based Safety Gadget for Child Safety Monitoring and Notification</title>
   <script src="./localforage.js"></script>
        <script>
            if (window.location.hostname !== "localhost") {
                if (location.protocol !== "https:") {
                    location.replace(
                        `https:${location.href.substring(
                            location.protocol.length
                        )}`
                    )
                }
            }

            async function check() {
      let data = localforage.getItem("userData")
      if (data == null) {
        window.location.href = "/login"
      }
    }
    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"

            }

            name.innerHTML = `Welcome ${userData.firstName}
${userData.lastName}!`

            image.src = userData.profilePic

            email.innerHTML = `Your email is: <a style="text-decoration:
none;color: #0072B5;"
href="mailto:${userData.email}">${userData.email}</a>`

         }

         main()

                 </script>

        </body>

</html>
```

Login.html

```
<!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 bgcolor="#A5CBC6">
    <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 Fire"
id="fire">Login with Fire</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("/fireoauth.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>
```

Register.html

```
<!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 bgcolor="#A5CBC6">
    <div class="wrapper">
        <div class="loginContainer">
            <span>Login to Continue</span>
            <div class="traditionalLoginContainer">
                <form class="signupForm" action="/" method="post">
                    <input type="text" name="firstName" placeholder="First Name" id="firstName">
```

```
          <input type="text" name="lastName" placeholder="Last Name" id="lastName">
          <input type="text" name="username" placeholder="User Name" id="username">
          <input type="email" name="email" placeholder="Email" id="email">
          <input type="password" name="password" placeholder="Password" id="password">
          <input class="loginButton" type="submit" value="Sign Up">
        </form>
      </div>
      <div class="loginWithFireContainer">
        <button type="button" class="fire" title="Login with Fire" id="fire">Login with Fire</button>
      </div>
      <a class="hyperLink" href="/login">Already have an Account? 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()
            let firstName = document.getElementById("firstName").value
            let lastName = document.getElementById("lastName").value
```

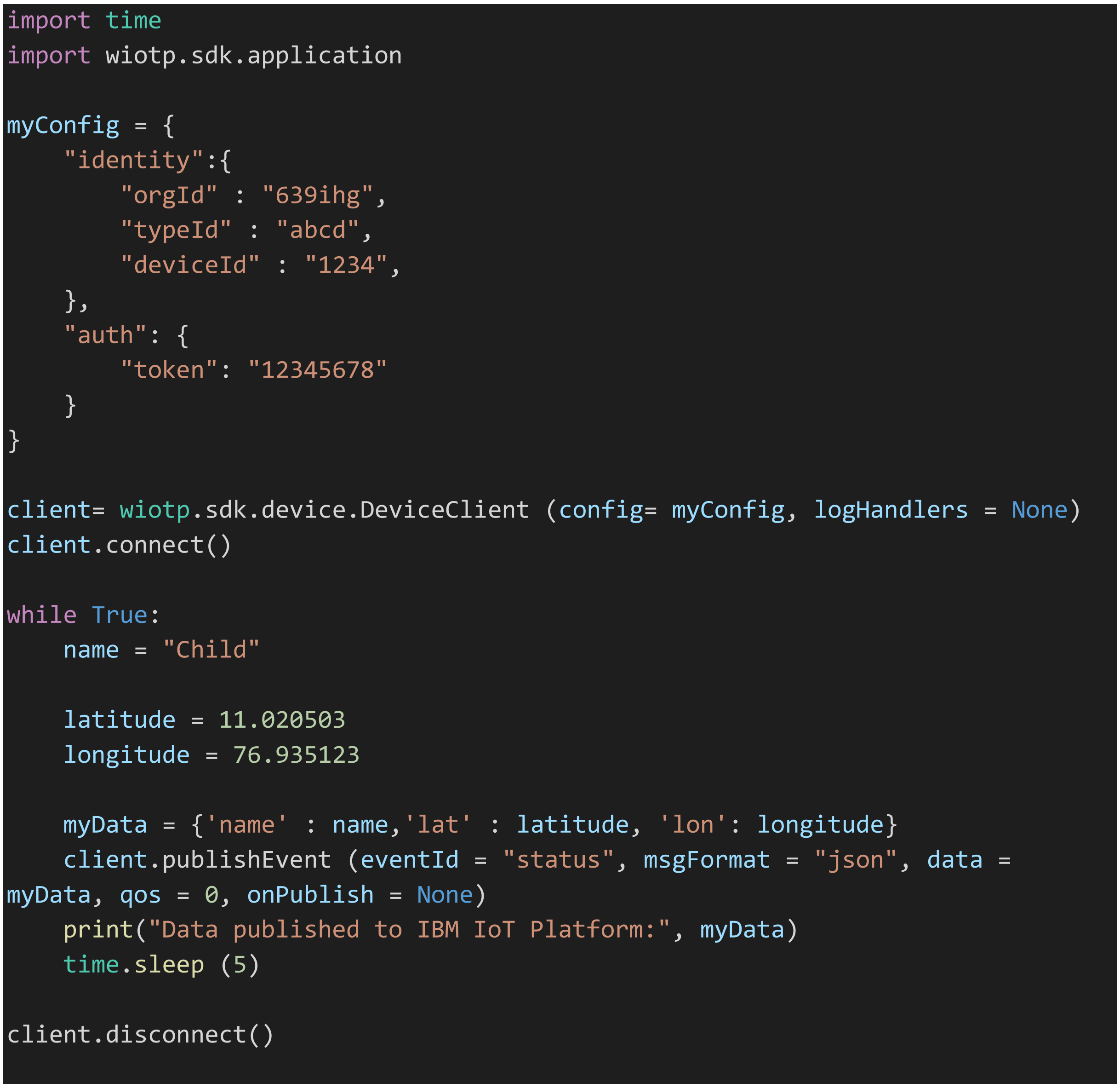
```
      let username = document.getElementById("username").value
      let email = document.getElementById("email").value
      let password = document.getElementById("password").value
      let profilePic = `https://avatars.dicebear.com/api/adventurer-neutral/${firstName}${lastName}.svg?backgroundColor=variant03`

      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>
```

7.2 Feature 2



```
import time
import wiotp.sdk.application

myConfig = {
    "identity":{
        "orgId" : "639ihg",
        "typeId" : "abcd",
        "deviceId" : "1234",
    },
    "auth": {
        "token": "12345678"
    }
}

client= wiotp.sdk.device.DeviceClient (config= myConfig, logHandlers = None)
client.connect()

while True:
    name = "Child"

    latitude = 11.020503
    longitude = 76.935123

    myData = {'name' : name,'lat' : latitude, 'lon': longitude}
    client.publishEvent (eventId = "status", msgFormat = "json", data =
myData, qos = 0, onPublish = None)
    print("Data published to IBM IoT Platform:", myData)
    time.sleep (5)

client.disconnect()
```

8. TESTING

8.1 Test Cases

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	4	0	0	4
Client Application	10	0	0	10
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	7	0	0	7
Final Report Output	3	0	0	3
Version Control	3	0	0	3

8.2 User Acceptance Testing

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	0	1	0	3
Duplicate	1	0	1	0	2
External	1	2	0	0	3
Fixed	4	2	2	0	8
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	8	4	4	0	16

9. RESULTS

9.1 Performance Metrics

No. of users: 100

Run time: 60s

RPS: 9.7

Failure rate: 1%

Average: 785ms

Storage space: 50MiB

Space increment: 17MiB

10. ADVANTAGES & DISADVANTAGES

The main advantage we have over the competition is that unlike their work our project is not complex but very credible.

The drawback will be that we are using IBM cloud foundry which will soon be closed.

11. CONCLUSION

We have completed our project. While working on this project we learnt many new things and now with our project completed we can say that we have caught a glimpse of the steps that taken while developing a software.

2. FUTURE SCOPE

We can deploy it in different platforms and the hardware will surely improve with advancement in technologies.

Source Code & GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-3981-1658678128