Project Development Phase

SPRINT – 1

Team ID	**PNT2022TMID05117**
Project Name	**IoT Based Safety Gadget For Child Safety Monitoring and Notification**

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>
    <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>
```

```
}
```

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;

    }

}
```

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>
    <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>
```

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>
    <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>Child Name</span>

        <span>Age 12</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>
```

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;

}
```

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>
  <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>
```