

## DELIVERY OF SPRINT 1

<u>Team ID</u>	PNT2002TMID10533
<u>Project Name</u> <u>Project</u>	<i>IOT Based Safety Gadget for Child Safety Monitoring&amp;Notification</i>

## **REGISTRATION AND LOGIN**

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <link rel="stylesheet" href="/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() 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/adventurerneutral/${firstName}${lastName}.svg?backgroundColor=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>
```

## OUTPUT STIMULATION

The screenshot displays an online HTML editor interface. The left pane shows the source code for a login form, and the right pane shows the rendered output.

**Source Code (HTML):**

```
25 <form class="signupform" action="/" method="post">
26 <input type="text" name="firstName" placeholder="First Name" id="firstName">
27 <br>
28 <input type="text" name="lastName" placeholder="Last Name" id="lastName">
29 <input type="text" name="username" placeholder="User Name" id="username">
30 <input type="email" name="email" placeholder="Email" id="email">
31 <input type="password" name="password" placeholder="Password" id="password">
32 <input type="text" name="phone number" placeholder="Phone number" id="phone
number">
33 <input type="radio" name="gender" value="Male" id="Male"> Male
34 <input type="radio" name="gender" value="Female" id="Female"> Female
35 <input class="loginButton" type="submit" value="SIGN UP">
36 </form>
37 </div>
38 <div class="loginWithFireContainer">
39 <button type="button" class="fire" title="Login" id="fire">LOGIN</button>
40 </div>
41 <a class="hyperLink" href="https://careereducation.smartinternz.com/student-enroll
-login">Already have an Account? Login </a>
42 </div>
43 </div>
44 <script>
45 // Necessary for Fire Auth to Function
46 const fireBroadcastingChannel = new BroadcastChannel('fireAuthChannel');
47 fireBroadcastingChannel.addEventListener('message', async event => {
48 let data = event.data
49 // **
50 * @typedef {Object<string, any>} Data
51 * @property {boolean} success - Whether the login was successful
52
```

**Rendered Output:**

Login to Continue

Madhumitha
G
Madhu1310
211419106155@smartinternz
****
7896543456

☐ Male  
☒ Female

[Already have an Account? Login >](#)

**Advertisement:** Geek Goddess 2022 Grand Finale. India's biggest programming event for women.

The screenshot shows the MIT App Inventor 2 mobile application interface. At the top, it displays the time (2:35 PM) and network status (183KB/s). The main screen has a green header with the app name and a menu icon.

**Instructions:**

type in the 6-character code  
-or-  
scan the QR code

**Input Fields:**

Six Character Code

**Buttons:**

connect with code  
scan QR code

**Information:**

Your IP Address is: 192.168.1.64  
Version: 2.65

☐ Use Legacy Connection

**Footer:**

This App is a helper Application for MIT App Inventor. It does not do anything by itself! Go to [appinventor.mit.edu](http://appinventor.mit.edu) for more information.