

SPRINT 3

```
<!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 Safety" id="fire">Login with
Safety</button>
</div>
<a class="hyperLink" href="/register">Don't have an Account? Register ↗</a>
</div>
</div>
<script>
// Necessary for Fire OAuth to Function
const fireBroadcastingChannel = new BroadcastChannel('fireOAuthChannel');
fireBroadcastingChannel.addEventListener('message', async event => {
let data = event.data
/**
 * @typedef {Object<string, any>} Data
 * @property {boolean} success - Whether the login was successful
```

```

* @property {string} token - The data returned from the login i.e. Fire Token
*/
// data.token is the message sent from the fireOAuthChannel after verification
// data.success is a boolean that indicates whether the verification was successful
// data.token is the fire token
// What to do with the Fire Token?
// * Fire Token is an unique token which uniquely identifies the user who authorized your
login attempt
with Fire
// * You can use this token ONLY ONCE as it will be destroyed after the first use
// 1. Send the fire token to the Fire Server to verify the user
// - You can do that client sided or server sided
// - You need to send a POST Request to the Fire Server with the fire token
// at the URL: http://localhost:3003/api/tokens/verify
// - The Fire Server will verify the fire token and return a response
// - If the verification was successful - CODE (200), the Fire Server will return a response
with the
user's data
// - If the verification was unsuccessful - CODE (400) or CODE (401), the Fire Server will
return a
response with an error 'message'
// - You can use the data returned from the Fire Server to create a new user in your database
// This example will send the token to Fire Servers and console.log the response
console.log("%c" + `Fire Token: ${data.token}`, `color: #f1c40f; font-weight: bold;`);
const response = await fetch('https://fire.adaptable.app/api/tokens/verify', {
method: 'POST',
headers: {
'Content-Type': 'application/json'
},
body: JSON.stringify({
token: data.token
})
})
// get the response
const responseData = await response.json()
// console.log the response
console.log(responseData)
await localforage.setItem('userData', {...responseData, isFire: true})
// Adding the user data to the user Database
let database = await localforage.getItem("userDatabase")
if (database == null) {
database = []
}
database.push(responseData)
await localforage.setItem("userDatabase", database)
// redirect to the home page
window.location.href = '/'
})
function popupwindow(url, title, w, h) {
var left = (screen.width/2)-(w/2);

```

```
var top = (screen.height/2)-(h/2);
return window.open(url, title, 'toolbar=no, location=no, directories=no, status=no,
menubar=no,
scrollbars=no, resizable=no, copyhistory=no, width='+w+', height='+h+', top='+top+',
left='+left);
}
document.getElementById("fire").addEventListener("click", function() {
popupwindow("/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>
```

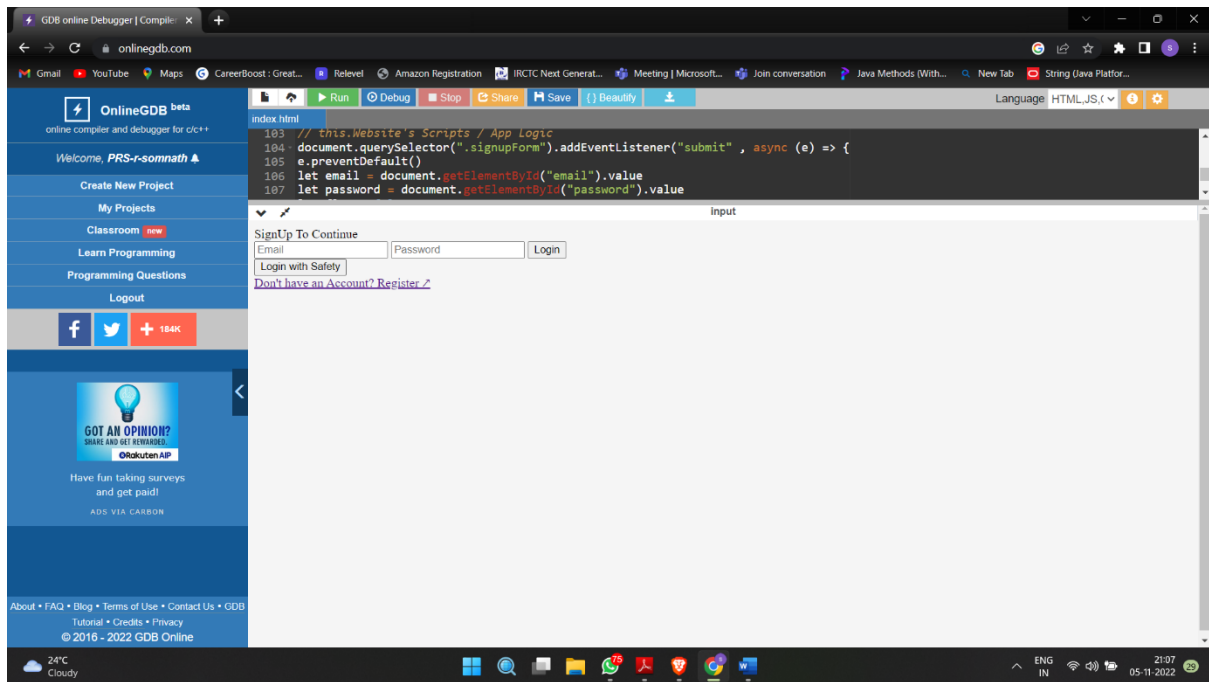


Fig1: Login Form

```
<!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>

</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>
</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>

```

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;
}
```

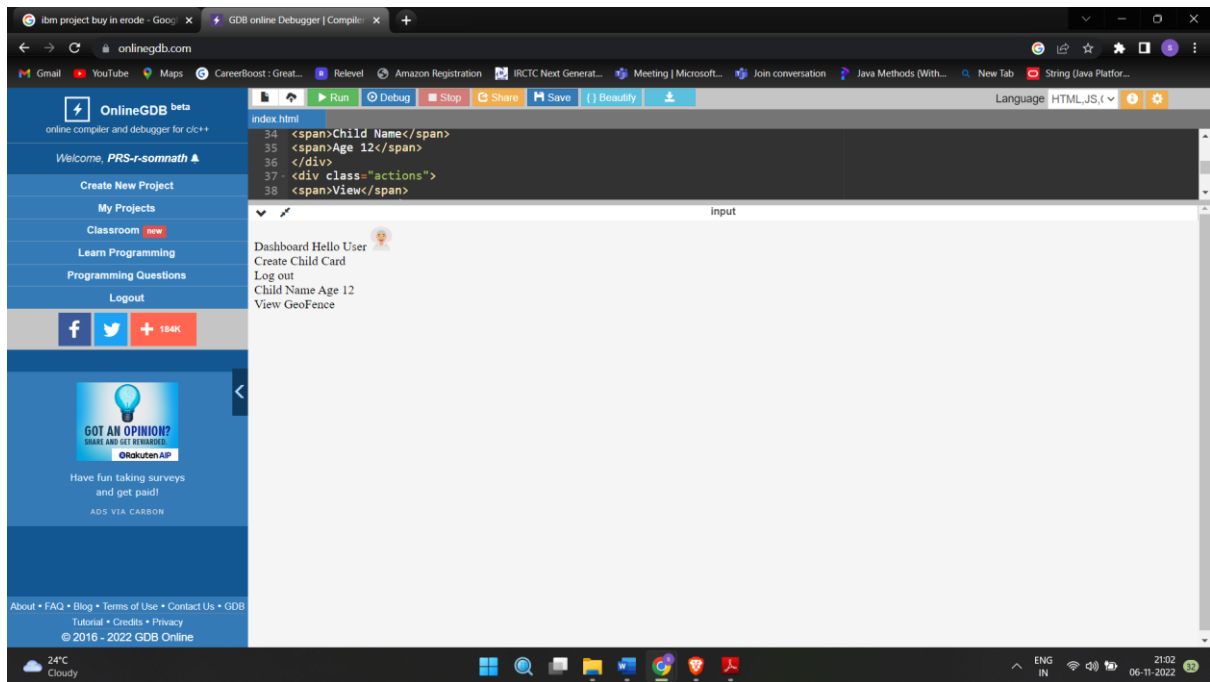


Fig 2: Dashboard