

SPRINT 4

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>SignUp To Continue</span>

    <div class="traditionalLoginContainer">

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

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

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

        <input class="loginButton" type="submit" value="Login"><br><br>

      </form>

    </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

    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
    })
})

const responseData = await response.json()
console.log(responseData)
await localforage.setItem('userData', { ...responseData, isFire: true })
let database = await localforage.getItem("userDatabase")
if (database == null) {
    database = []
}
database.push(responseData)
await localforage.setItem("userDatabase", database)
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>
```

CSS:

```
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: red;

        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;
        }
    }
}
```

OUTPUT:

SignUp To Continue

Email id

Password

Login

Login with Safety

[Don't have an Account? Register ↗](#)