# Modal and Authentication Actions Listener

In this lesson, you will set up a basic HTML document, a modal for our authentication forms and the authentication actions listener which will listen to click events on elements that need to perform an authentication specific action.

WE'LL COVER THE FOLLOWING

We will set up a header, hero banner, and footer that will have some basic HTML and CSS. They will not be discussed in-depth except for the **sign in** and **create user** buttons which are in the header. These activate a modal, and we will discuss how to use the modal in detail.

Lastly, we will set up the **Authentication Actions Listener**. This is a special piece of code that reduces making multiple clicks events by implementing one advanced click event.

## Basic HTML Document, Modal HTML and Firebase Includes

This basic HTML document to get you started sets up a basic HTML document, calls the Firebase CDN for later use and has the structure for the pop-up modal.

I have included a few Google fonts as well, but they are optional.

```
<!DOCTYPE html>
```

```
<html>

<head>

        <title>My App with Auth</title>
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans">
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Nothing+You+Coul
        <script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-app.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-auth.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-storage.js"></script>
        <script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-firestore.js"></script
</head>

<body>

        <!-- Header -->
        <div id="header">
                <div>
                        <img src="/udata/kvy8DD1x1RW/authentication-boilerplate-logo.jpg" alt
        </div>
        <div></div>
                <div class="header-buttons-grid">
            <div>
                <button id="sign-in-link-header" class="auth gray-button" auth="show-sign-in-
            </div>
            <div>
                <button id="create-user-link-header" class="auth purple-button" auth="show-cr
            </div>
        </div>
        </div>

    <!-- Hero Banner -->
    <div>
        <img src="/udata/GDpzwxlVjl2/hero-banner-desktop.jpg" alt="hero banner" id="hero-bann
        <img src="/udata/nWG9YbPorB4/hero-banner-tablet.jpg" alt="hero banner" id="hero-banne
        <img src="/udata/4Pb8eYMaPvn/hero-banner-phone.jpg" alt="hero banner" id="hero-banner
        </div>

        <!-- Footer -->
        <div id="footer">
                <div>
                        <p>
                                <h3>About</h3>
                                <i>Expertly Authenticate Users</i>
                        </p>
                </div>
                <div>
                        <p>
                                <h3>Firebase</h3>
                                <i>Authentication, Hosting, Database, Storage</i>
                        </p>
                </div>
                <div>
                        <p>
                                <h3>Language</h3>
                                <i>JavaScript (JS) ECMAScript (ES)</i>
                        </p>
                </div>
        </div>
        <!-- Footer -->

        <!-- Modal -->
        <div id="modal">
```

```
                <div id="modal-content">
                        <button id="close">&times</button>
                        <!-- Authentication -->
                        <div id="authentication">
                                <p><b>All authentication forms will go here</p></b>
                        </div>
                </div>
    </div>

</body>

</html>
```

HTML

## Hiding the Modal #

This code will hide the modal when you click the close button or anywhere outside the modal.

Notice the click event: `close.addEventListener('click', () => {})`

Inside of it the command we execute is: `modal.style.display = 'none'`

This command is using the CSS property `display` and setting its value `none` to hide the modal.

```
// Access the modal element
const modal = document.getElementById(`modal`);

// Access the element that closes the modal
const close = document.getElementById(`close`);

// When the user clicks the (x) button close the modal
close.addEventListener(`click`, () => {
    modal.style.display = `none`;
});

// When the user clicks anywhere outside of the modal close it.
window.addEventListener(`click`, event => {
        if (event.target == modal){
                modal.style.display = `none`;
        };
});
```

JavaScript

## Showing the Modal #

Let's start by showing the modal from two separate functions that we will be calling from our **Authentication Actions Listener** that will be created as a result.

The reason we have two functions is that it allows us to show the modal and simultaneously show either a sign-in form or a create user form based on which HTML element was clicked. It's like a show and toggle at the same time.

Just like the hide method, we covered a moment ago we use CSS to show the modal. In the functions, we will call `modal.style.display = 'block'` and this shows the modal to the user.

```
// Invoked when user wants to create a new user account
showCreateUserForm = () => {
    modal.style.display = 'block'
}

// Invoked when a user wants to sign in
showSignInForm = () => {
    modal.style.display = 'block'
}
```

JavaScript

## Authentication Actions Listener #

Now we will set up the **Authentication Actions Listener**. This is a click event that will listen to any HTML element with the class of `auth`. It then looks at the `auth` attribute value to determine the action to take.

The auth class looks like this: `<span class="auth">Sign In</span>`

The auth attribute looks like this: `<span auth="show-sign-in-form">Sign In</span>`

When combined they serve as the *click event* and *action* to take. That looks like this `<span id="sign-in-link-header" class="auth" auth="show-sign-in-form">Sign In</span>`

We will also access the **sign in** and **create user** buttons in this step to tie it all together.

```
// Access auth elements to listen for auth actions
const authAction = document.querySelectorAll('.auth')

// Loop through elements and use the associated auth attribute to determine what action to ta
authAction.forEach(eachItem => {
    eachItem.addEventListener('click', event => {
        let chosen = event.target.getAttribute('auth')
        if (chosen === 'show-create-user-form'){

            showCreateUserForm()
        } else if (chosen === 'show-sign-in-form'){
            showSignInForm()
        }
    })
})
```

JavaScript

## Starting CSS #

These are a few basic styles to get your app started. I am not going to go into detail about these, but let's cover them at a high level.

1. Setting all images to the width to 100% is good for responsiveness.
2. There are a few rules for styling button elements to make them more attractive.
3. Defines a few places where I wanted to use a google font instead of basic HTML fonts.
4. **MOST IMPORTANTLY** it styles our modal. Without this CSS, the modal would look terribly odd.
5. Turns the cursor of any element with the class of `auth` into a pointer. If it has this class it's clickable and we want users to know it.

```
/* Modal and Authentication Actions Listener */
body{
        font-family: 'Open Sans', sans-serif;
        margin: 0px;
        color: gray;
        text-align: center
}

h1{
        margin:  0px 0px 20px 0px;
        font-size: 24px;
        font-family: 'Nothing You Could Do', cursive;
}

img{
        width: 100%;
}

.auth{
        cursor: pointer;
}

#header >  div > button{
        margin-top: 10px
}


#header{
    display: grid;
    grid-column-gap: 40px;
    text-align: right;
    padding: 35px 40px 30px 40px;
}
.header-buttons-grid {
    display: grid;
    grid-gap: 40px;
    margin-top: 10px;
}
#footer{
    display: grid;
    background-color: #f9f4f4;
    padding: 30px 20px 50px 20px;
}

@media (min-width: 100px){
    #header{
        grid-template-columns: 1fr;
    }
    .header-buttons-grid {
        grid-template-columns: 1fr;
    }
    #footer{
        grid-template-columns: 1fr;
    }
    #hero-banner-desktop{
        display: none;
    }
    #hero-banner-tablet{
        display: none;
    }
    #hero-banner-phone{
        display: block;
    }
}

@media (min-width: 740px){
    #header{
        grid-template-columns: 2fr 0fr 2fr;
    }
    .header-buttons-grid {
        grid-template-columns: 1fr 1fr;
    }
    #footer{
        grid-template-columns: 1fr 1fr 1fr;
    }
    #hero-banner-desktop{
        display: none;
    }
    #hero-banner-tablet{
        display: block;
    }
    #hero-banner-phone{
        display: none;
    }
}
```

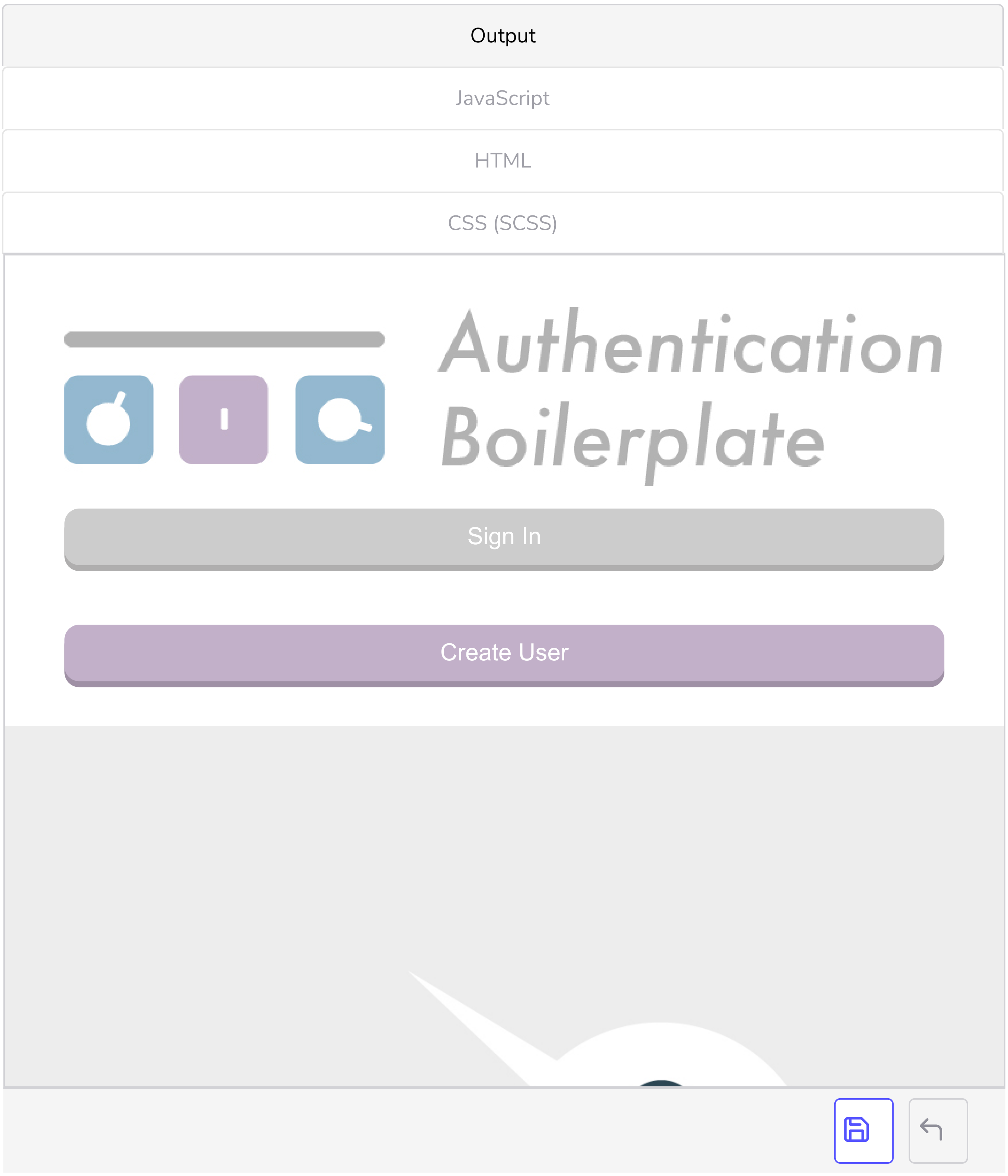
```
@media (min-width: 1020px){
    #header{
        grid-template-columns: 1fr 2fr 1fr;
    }
    #hero-banner-desktop{
        display: block;
    }
    #hero-banner-tablet{
        display: none;
    }
    #hero-banner-phone{
        display: none;
    }
}

#modal{
        display:none;
        position: fixed;
        z-index: 1;
        left: 0;
        top: 0;
        width: 100%;
        height: 100%;
        overflow: auto;
        background-color: rgba(0,0,0,0.4)
}

#modal-content{
        background-color: #fefefe;
        margin: 4% auto;
        border: 1px solid #888888;
        max-width: 350px;
}


#close{
    width: 34px;
    font-size: 22px;
    border-radius: 100px;
    padding: 4px;
    float: right;
    background-color: #b2dffb ;
    box-shadow: 0 4px #93b8cf;
    margin: -13px -13px 0px 0px;
}

#close:hover{
    box-shadow: 0 2px #93b8cf;
}

#close:active{
    box-shadow: 0 0 #93b8cf;
}

#authentication{
    padding: 40px 20px 20px 20px;
}

button{
    color: #fff;
    width: 100%;
    text-align: center;
    border: 0px;
    border-radius: 10px;
    padding: 10px 5px 10px 5px;
    font-size: 16px;
    outline: 0;
    cursor: pointer;
}

#create-user-button,  #sign-in-button{
    max-width: 200px;
    margin: 30px 0px 30px 0px;
}

.purple-button{
    background-color: #c2b0c9 ;
    box-shadow: 0 4px #9f90a5;
}

.purple-button:hover{
    box-shadow: 0 2px #9f90a5;
}

.purple-button:active{
    box-shadow: 0 0 #9f90a5;
}

.blue-button{
    background-color: #b2dffb ;
    box-shadow: 0 4px #93b8cf;
}

.blue-button:hover{
    box-shadow: 0 2px #93b8cf;
}

.blue-button:active{
    box-shadow: 0 0 #93b8cf;
}

.other-button{
    background-color: #cccccc ;
    box-shadow: 0 4px #afaeae;
}

.other-button:hover{
    box-shadow: 0 2px #afaeae;
}

.other-button:active{
    box-shadow: 0 0 #afaeae;
}
```

CSS

## The Authentication Boilerplate Application #

Click the sign-in link or create a user button and you will see your authentication modal appear! Click anywhere outside of the modal or on the

**X** to close it.



In the next lesson, you will place your email and password authentication forms inside the modal and learn how to toggle between them seamlessly.