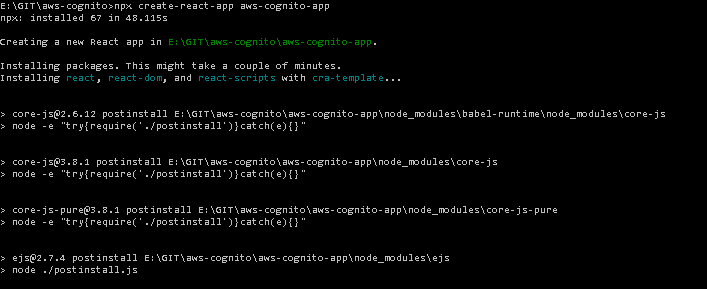
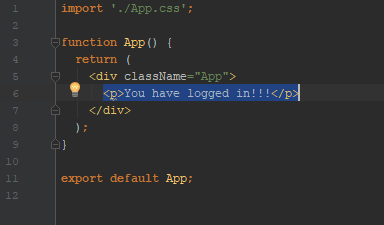
To start creating the application, let us first create a new React application.

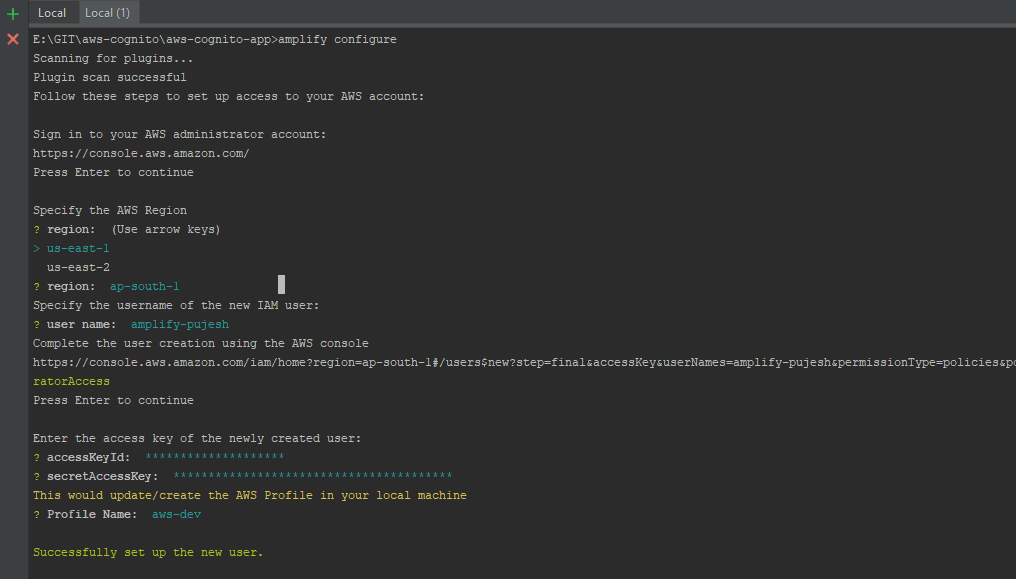
To create a new React app through the command line, let us open a command line prompt and enter the following command:   
> npx create-react-app aws-cognito-app   
And press the Enter button. We will get the following output once it is done creating the application.   
  
  
Fig: Creating a react application from command line.

Once the application is created, let us open our IDE and start the server.   
> npm start  
This hosts the application that we have just created in our localhost. Once it has started hosting in the development server a browser will open up and we can see the React logo.

Now, let us modify the App.js file and let us just print the “You have logged in!!!”. We can simply do this by removing the existing code inside the div[class=”App”] element and adding the <p>You have logged in!!!</p> element in the App.js script file.  
  
  
Fig: Updating the App.js file to display a simple message.

Now, we can see the “You have logged in!!!” message in the browser. Let us now add a login UI before any users can see the message in the browser. To add this we need to first configure AWS CLI. It is a simple process and can be easily configured by following the steps from <https://docs.amplify.aws/start/getting-started/installation/q/integration/react>

To do all this, we need to create an AWS account. Once the account is created we can then configure AWS Amplify.   
We can configure amplify by entering the command:   
> amplify configure

**NOTE:** To proceed with the configuration we need to create a user with Administrator Access IAM role. Access key and secret key are provided by the AWS upon creating a user. Store the secret key somewhere as it cannot be generated again.   
  
Fig: Amplify configure

After we have configured amplify, let us initialize it so that we can create necessary backend services.   
> amplify init

Once we have initialized a new folder amplify in the root directory and a aws-exports.js file under the src directory is created automatically.   
The aws-amplify is the main library for working with Amplify in your apps. So, let us install the dependency.  
> npm install aws-amplify @aws-amplify/ui-react

This package includes React specific UI Components.

**NOTE:** If you have already hosted your app in localhost:3000 you might get a **'Module not found'** error. If you face this issue, terminate the terminal (Ctrl + C) and restart it again.

Once the installation is completed, let us configure amplify on the client so that we can use it to interact with our backend services. Let us open the App.js file under the src directory and paste the following lines of code below the last import:   
 import Amplify from "aws-amplify";  
 import awsExports from "./aws-exports";  
 Amplify.configure(awsExports);

Now, we can start building our authentication page. Here, we will be using the Authentication component provided by the Amazon Amplify. This amplify framework uses Amazon Cognito as the main authentication provider. It handles the user registration, authentication, account recovery and other operations.

To add the authentication, follow the following steps:  
> amplify add auth

Select the options provided for each configuration or we can proceed with default value for now. Few of the settings can be changed later while to change others, it will require to be configured from the scratch. Once it set up we need to deploy it. We can deploy the service by   
> amplify push

Once we push the service it generates a resource name. The details of this configuration will be updated automatically in aws-exports.js file. We can also check that this user pool has been created in the Amplify Console.

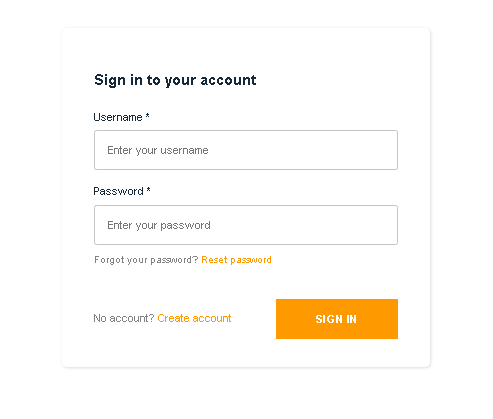
To view the user pool in the console, Login to your Amazon AWS console.   
At the top click Services > Cognito > Manage User Pools.  
Here you can see a user pool that has been created with your resource name.

Now after the configuration, it is time to create the login UI for our application. To use this built-in component from the AWS Cognito in our react app, we need to add the following codes in our App.js file.

First, let’s import the withAuthenticator component from '@aws-amplify/ui-react'  
> import { withAuthenticator } from '@aws-amplify/ui-react';

Then let us wrap our main component with the exported component – withAuthenticator  
> export default withAuthenticator(App)

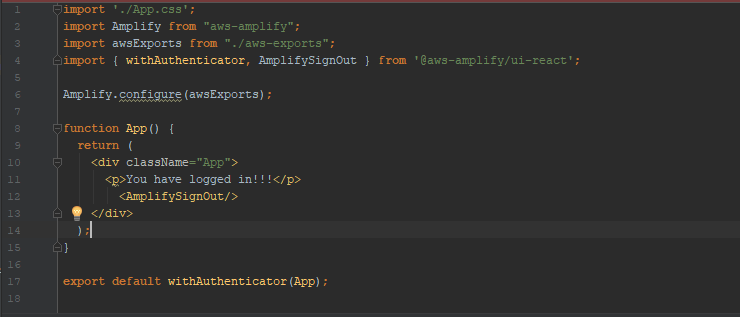
withAuthenticator is a HOC (higher order component) provided by the AWS Cognito.

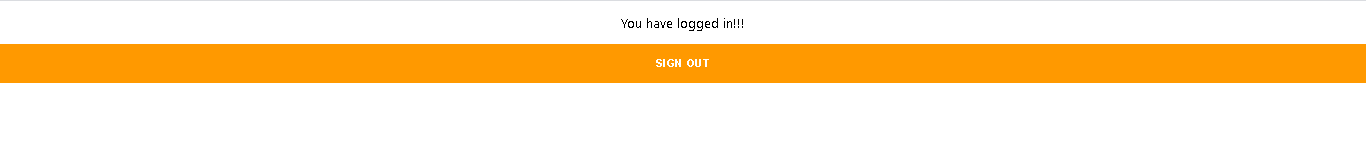
When you load your application you can see something like this -   
  
Fig: Authentication UI Component

Here, we can see a UI component that has all the required authentication functionality such as Sign In, Sign Up, Forgot Password. Now we are ready to sign up a user for us. You can try creating users and logging in with it. This template is based on the Amazon AWS theme. We will shortly look into it to change its appearances.

When you sign up with a user you can check the console to see if the user is created! Also you can create a user group and assign users to the groups too.

Once you create a user and try to log into it, you will then see the "You have logged in!!!" message in the application.

To add a sign out we are just going to add a <AmplifySignOut/> component to our App component.  
The code so far will look something like below:  
  
Fig: App.js Component after adding Sign out component.

After we add this code, we will see that there is a SIGN OUT button added to our application.  
  
Fig: Sign Out button.