CREATING CLIENT APPLICATION USING REACTJS USING AWS COGNITO FOR SIGNUP and SIGNIN

1) Setting Up Web App : The web app create-react-app

AWS Amplify makes it easy to create, configure, and implement scalable mobile and **web apps** powered by **AWS**

2) AWS Amplify SDK : npm install aws-amplify –save

3) Now create a new file call config.js in our src folder

And open the config.js file our IDE and

Place the configurations from aws apigateway and from cognito....

https://console.aws.amazon.com/apigateway/home?region=us-east-1#/apis

Go to API gateway find the end point and go to stages inside it and after that you will find the url over there and copy that url and paste it in url and copy the region from that url and place at region...

And go to the cognito and go to user pool and get ur pool id and place it in user pool and place there and after that the get the identity pool id from the federated identities and place it in the config file.

Inside the config.js file ….

```
export default {
apigateway: {
REGION: 'us-east-1',
URL: 'https://mznrnoy5i0.execute-api.us-east-1.amazonaws.com/dev'
},
cognito: {
REGION: 'us-east-1',
USER_POOL_ID: 'us-east-1_exKWCRida',
APP_CLIENT_ID: '5kq6ms9r08dk3bv5rhk94km265',
IDENTITY_POOL_ID: 'us-east-1:38a79da5-1409-46ed-8efb-df90ca4a1e80'
}
}
```

And now the configuration will be done.

And Open your reactjs index file inside src folder and import the amplify library from aws-amplify

And import the config file aswell and and the configure amplify like this in the file...

```
Amplify.configure({
Auth:{
mandatorySignIn: true,
region: config.cognito.REGION,
userPoolId:config.cognito.USER_POOL_ID,
identityPoolId:config.cognito.IDENTITY_POOL_ID,
userPoolWebClientId:config.cognito.APP_CLIENT_ID
},
api:{
endpoints:[
{
name: 'testApi',
endpoint: config.apigateway.URL,
region: config.apigateway.REGION
}
]
}
});
```

And after that do npm install from the terminal so that it will install if any configurations are to be made.

And do npm start to start the service so that web app will work now.

And we have to place SignUp and SignIn Buttons over there and

4) Create a test user to start with.

For Sign-Up we need ….

API

Region

Client-id

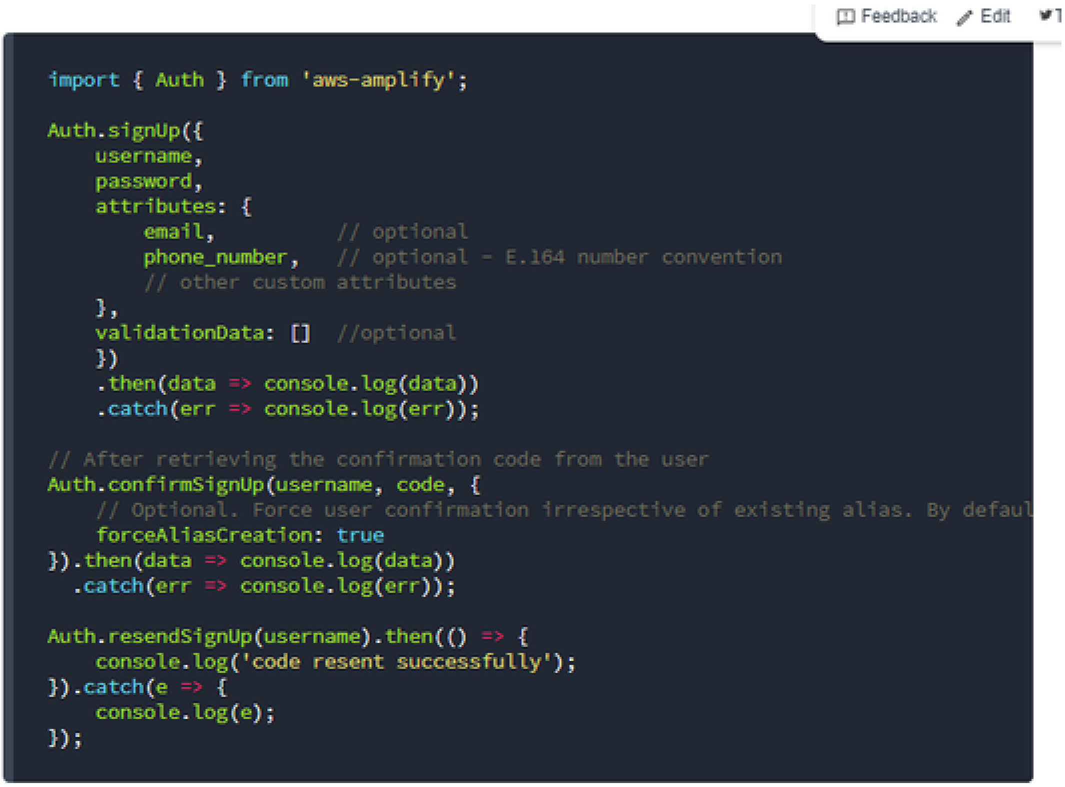
Username or Email

Password

Session Storage of Tokens

Create a login Page with template and with validation using react formik ui...

Once Done with the SignUp screen



```
import { Auth } from 'aws-amplify';

Auth.signUp({
    username,
    password,
    attributes: {
        email,          // optional
        phone_number,   // optional - E.164 number convention
        // other custom attributes
    },
    validationData: []  //optional
    })
    .then(data => console.log(data))
    .catch(err => console.log(err));

// After retrieving the confirmation code from the user
Auth.confirmSignUp(username, code, {
    // Optional. Force user confirmation irrespective of existing alias. By defaul
    forceAliasCreation: true
}).then(data => console.log(data))
  .catch(err => console.log(err));

Auth.resendSignUp(username).then(() => {
    console.log('code resent successfully');
}).catch(e => {
    console.log(e);
});
```

Once done with the signup going to signin by using email and password...

import the Auth from aws-amplify library and put that Auth in handle submit function like this....

handleSubmit = async event => {

try{

await Auth.signIn(this.state.email, this.state.password);

alert(‘Logged In’);

} catch (e) {

alert(e.message);

}

}

We have to do lot many things like...

- Change Password
- Forgot Password
- Validating User
- Verify Phone number or email id
- Create a New Password
- SignOut
- Retrive Current Session
- Managing Security Tokens
- Using Auth Components in Reactjs by ‘withAuthonticator’ Higher Order Component.

To Create authentication with AWS Cognito in a web app with React...

We have found two best aws libraries...

One is AWS Amplify and another is AWS AppSync...

https://aws.amazon.com/about-aws/whats-new/2018/04/aws-amplify-adds-support-for-graphql-and-aws-appsync-enabling-re/

*****

Instead of using AWS Amplify we can deal with AWS Appsync which will reduce the cost and bandwidth by using graphql query language. AWS AppSync extends GraphQL's capabilities even more by offering enterprise-level security, real-time data synchronization.