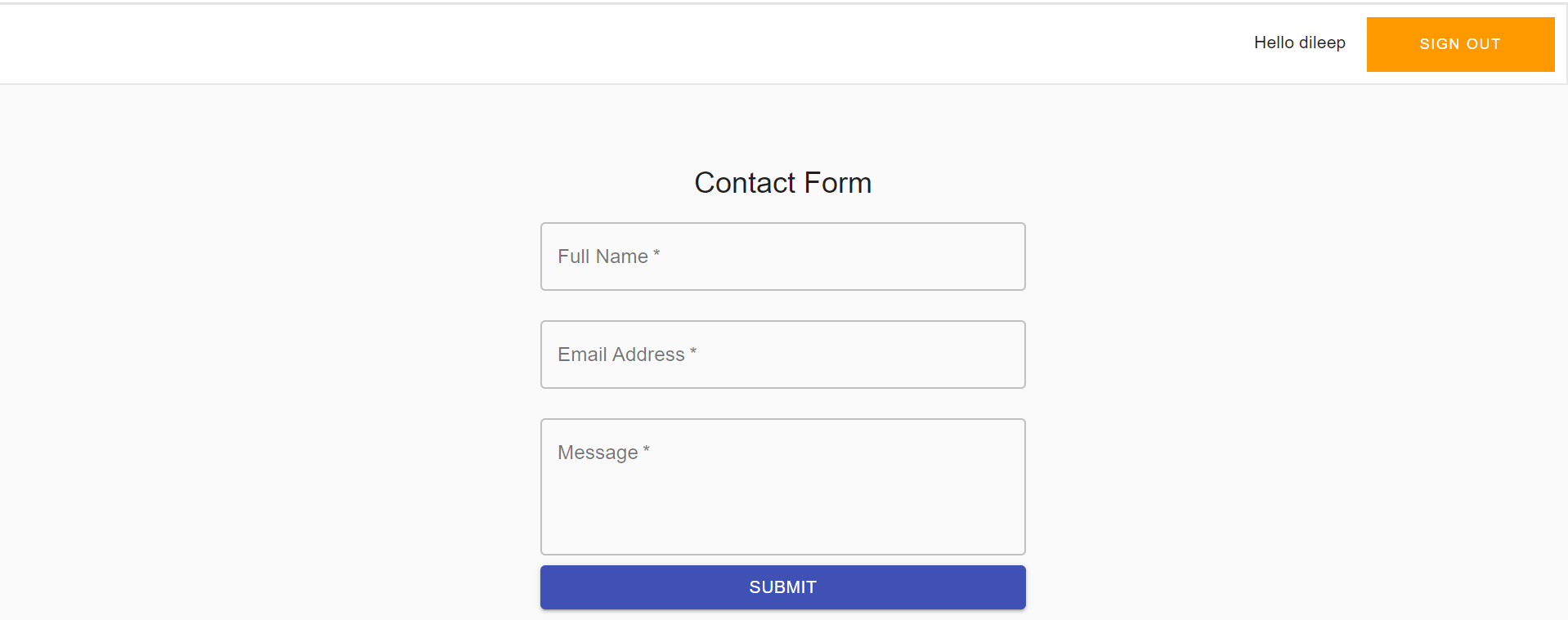
**Contact Form App**

**Project Outline**: Build a website contact form. The form should contain fields for Name, Email and message at a minimum. Upon submit, the form should send the message to the email provided in the form, with the name as the email subject.



**Project Requirements:**

1. Set up a free account with Amazon Web Services (AWS) (<https://aws.amazon.com/>)
2. Using Material UI, Build a form (name, email, message, submit button). The page must be a static website, hosted in an S3 bucket. Ensure proper field level validation (E.g. Please enter a name.) and action confirmation user experiences are properly handled.

Web Page Link: <https://d2gxp6vg87knst.cloudfront.net/>

1. On submit, the UI should call API Gateway to trigger a Lambda function.

=>

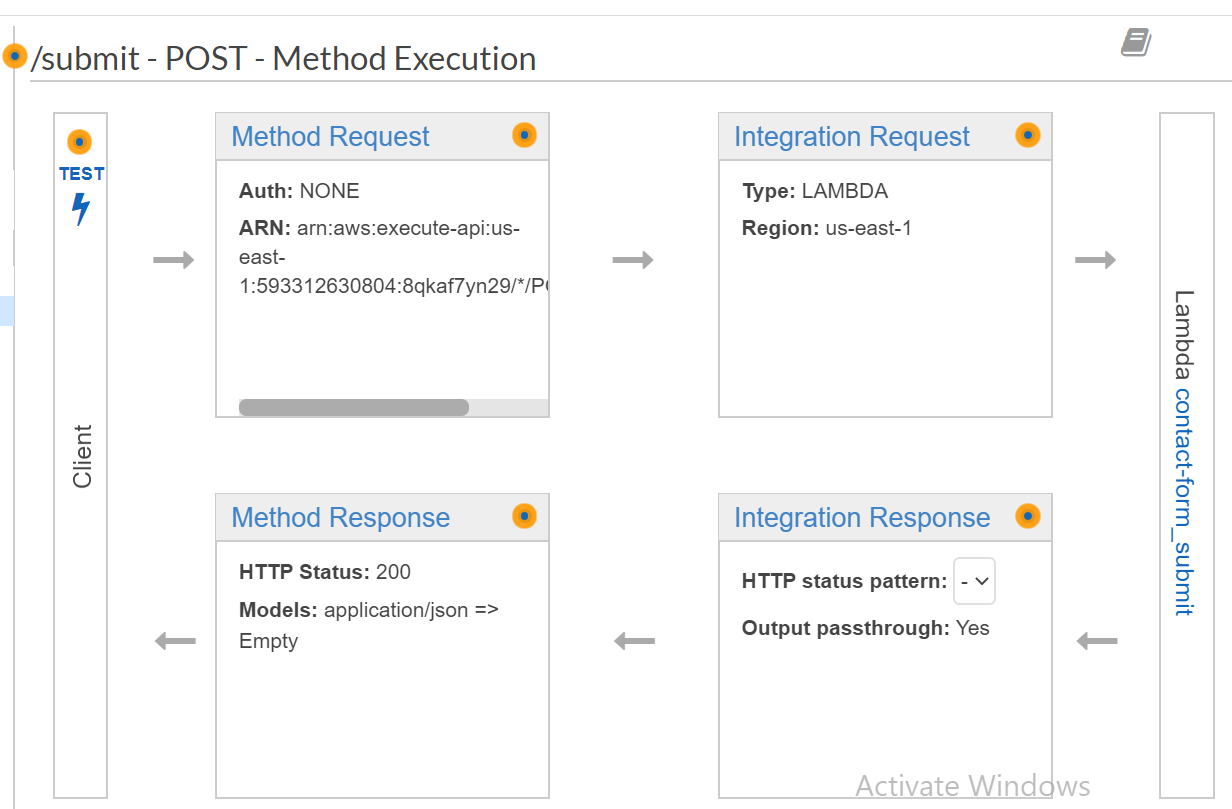
API Gateway Name: [send-email-api](https://console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/8qkaf7yn29/resources)

1. Send Email:

Resource Name: submit

Method: POST

Lambda Function: contact-form\_submit

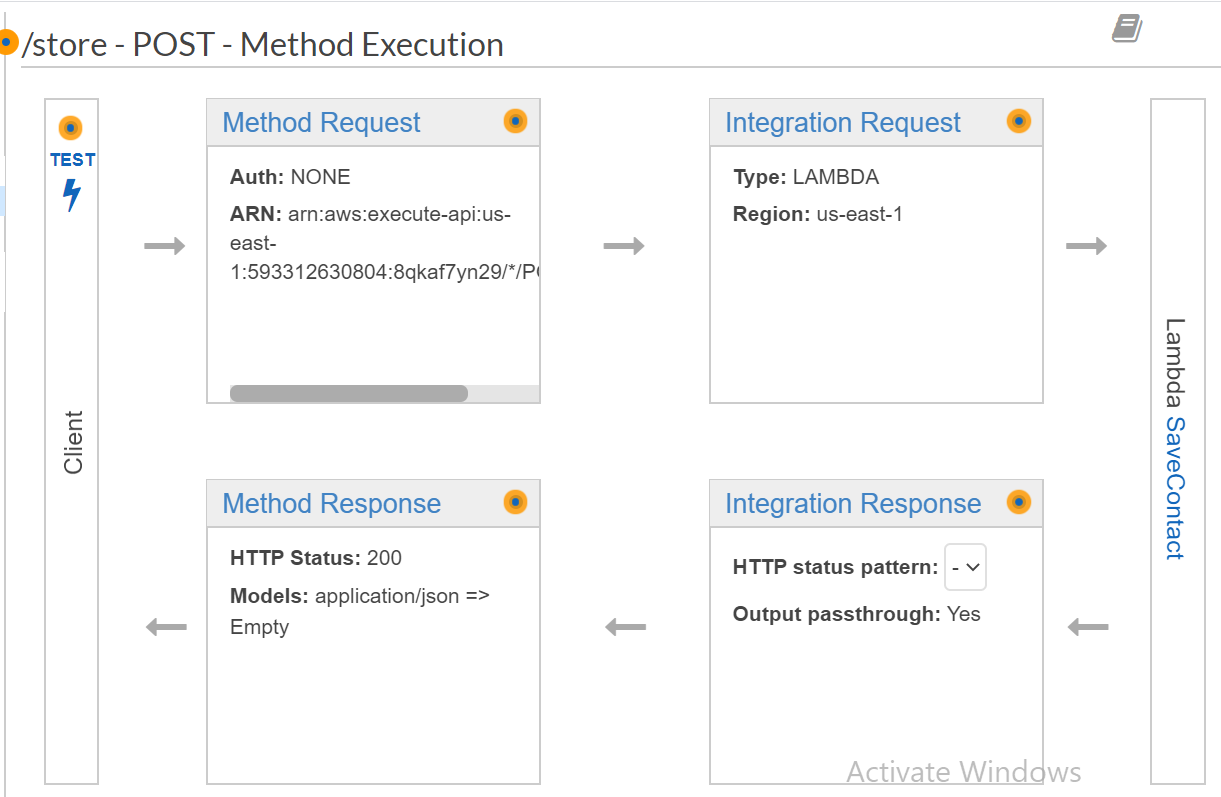


1. Send Email:

Resource Name: store

Method: POST

Lambda Function: SaveContact



1. The Lambda function should send an email using SES, and store the message in DynamoDB.

**Lambda Function Name: contact-form\_submit**

1. Index.js

const sendEmail = require('./send')

exports.handle = function(e, ctx, cb) {

if (e.toEmail && e.subject && e.body) {

sendEmail(e.toEmail, e.subject, e.body).then((result) => {

cb(null, result)

}, cb)

} else {

cb(new Error('Invalid payload: ' + JSON.stringify(e)))

}}

1. Send.js

const AWS = require('aws-sdk')

AWS.config.update({

region: process.env.SES\_REGION

})

const ses = new AWS.SES({

apiVersion: '2010-12-01'

})

const ddb = new AWS.DynamoDB.DocumentClient({

region: 'us-east-1'

});

module.exports = function sendEmail(toEmail, subject, body) {

return new Promise((resolve, reject) => {

// this must relate to a verified SES account

const from = process.env.FROM\_NAME + ' <' + process.env.FROM\_EMAIL + '>'

const params = {

Source: from,

Destination: {

ToAddresses: [toEmail]

},

Message: {

Subject: {

Data: subject,

Charset: 'UTF-8'

},

Body: {

Text: {

Data: body,

Charset: 'UTF-8'

}

}

}

} //console.log('e-mail params:', params)

ses.sendEmail(params, (err, data) => {

if (err) {

console.error('Failed to send the email:', err.stack || err)

reject(err)

} else {

console.log('e-mail sent:', data)

resolve(data)

}

})

})

}

**Lambda Function Name: SaveContact**

1. Index.js

const AWS = require('aws-sdk');

const ddb = new AWS.DynamoDB.DocumentClient({region: 'us-east-1'});

exports.handler = (event, context, callback) => {

// TODO implement

if (event.emailid && event.name && event.message) {

var params = {

Item: {

emailid: event.emailid,

name: event.name,

message: event.message

},

TableName: 'contactform'

}

ddb.put(params, function(err, data) {

if(err){

callback(err,null);

}else{

callback(null,data);

}

})

} else {

callback(new Error('Invalid payload: ' + JSON.stringify(event)))

}

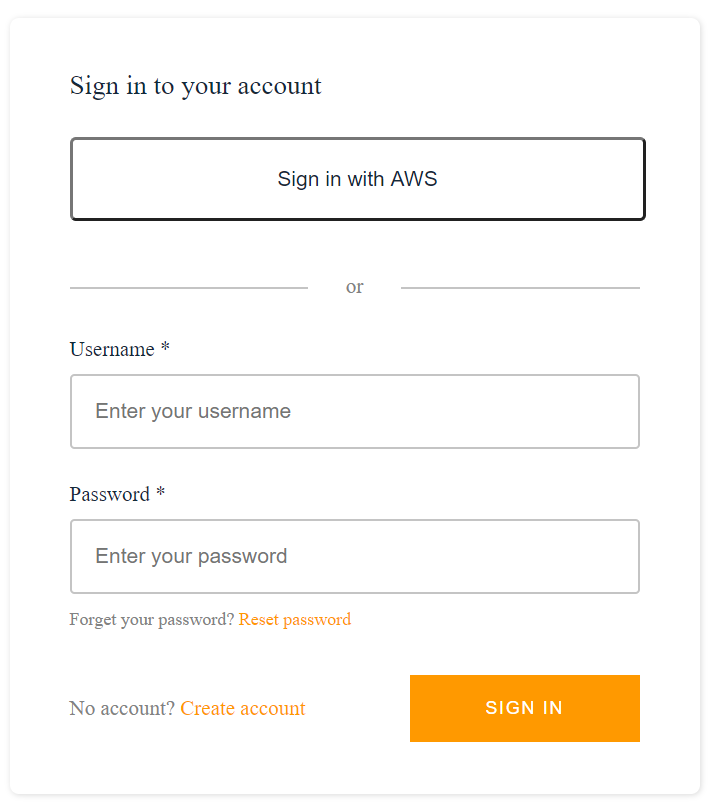
};

1. Push your code to a public GitHub repository

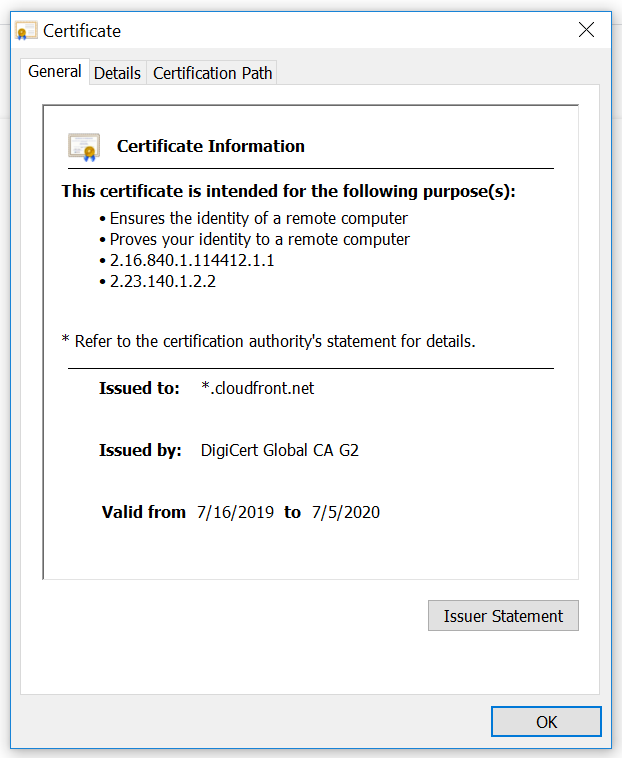
Link: <https://github.com/dileepdail/contact-form-app.git>

**Bonus Points:**

1. Access to the form should be protected via a login screen. Set up a login page with AWS Cognito User Pools and Federated Identities.



1. Enable SSL (TLS 1.2) for your page.



1. Allow authentication via Facebook, Google or another third-party provider.

