

What is Amazon Cognito?

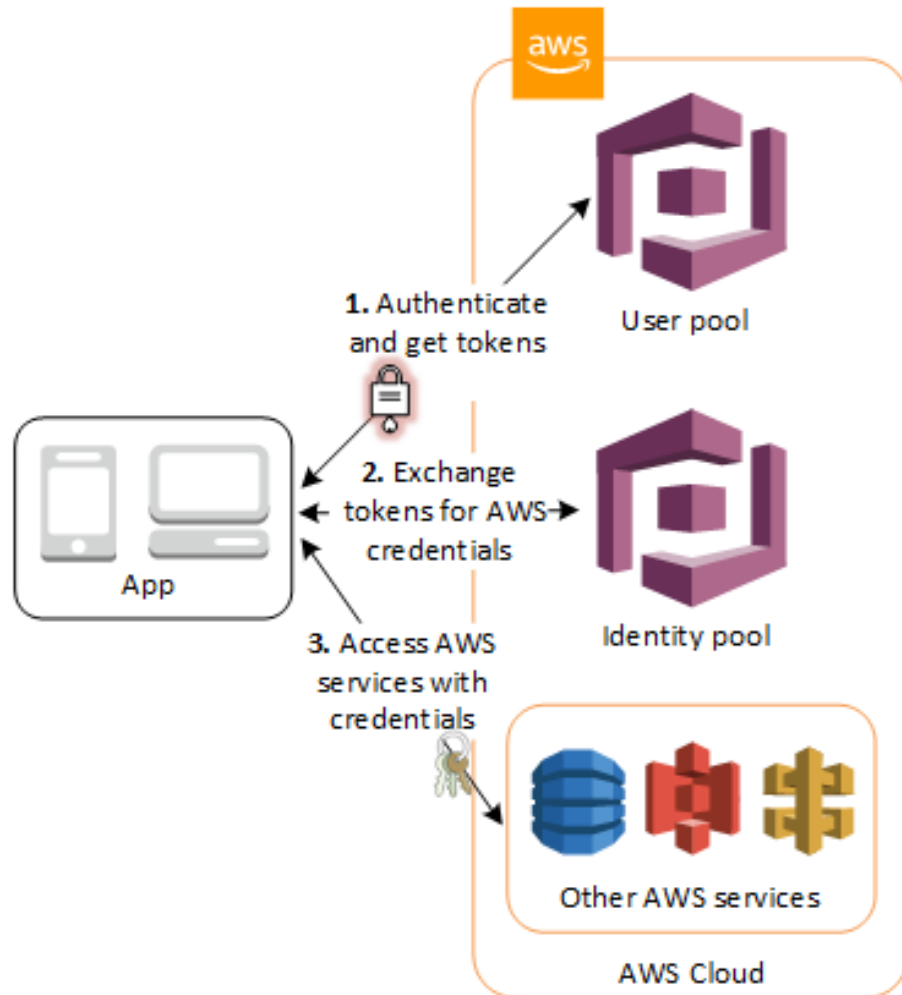
Amazon Cognito provides authentication, authorization, and user management for your web and mobile apps. Your users can sign in directly with a user name and password, or through a third party such as Facebook, Amazon, Google or Apple.

The two main components of Amazon Cognito are user pools and identity pools. User pools are user directories that provide sign-up and sign-in options for your app users. Identity pools enable you to grant your users access to other AWS services. You can use identity pools and user pools separately or together.

An Amazon Cognito user pool and identity pool used together

See the diagram for a common Amazon Cognito scenario. Here the goal is to authenticate your user, and then grant your user access to another AWS service.

1. In the first step your app user signs in through a user pool and receives user pool tokens after a successful authentication.
2. Next, your app exchanges the user pool tokens for AWS credentials through an identity pool.
3. Finally, your app user can then use those AWS credentials to access other AWS services such as Amazon S3 or DynamoDB.



For more examples using identity pools and user pools, see [Common Amazon Cognito scenarios](#).

Amazon Cognito is compliant with SOC 1-3, PCI DSS, ISO 27001, and is HIPAA-BAA eligible. For more information, see [AWS services in scope](#). See also [Regional data considerations](#).

Topics

- [Features of Amazon Cognito](#)
- [Getting started with Amazon Cognito](#)
- [Regional availability](#)
- [Pricing for Amazon Cognito](#)
- [Using the Amazon Cognito console](#)
- [Using this service with an AWS SDK](#)

Features of Amazon Cognito

User pools

A user pool is a user directory in Amazon Cognito. With a user pool, your users can sign in to your web or mobile app through Amazon Cognito, or federate through a third-party identity provider (IdP). Whether your users sign in directly or through a third party, all members of the user pool have a directory profile that you can access through an SDK.

User pools provide:

- Sign-up and sign-in services.
- A built-in, customizable web UI to sign in users.
- Social sign-in with Facebook, Google, Login with Amazon, and Sign in with Apple, and through SAML and OIDC identity providers from your user pool.
- User directory management and user profiles.
- Security features such as multi-factor authentication (MFA), checks for compromised credentials, account takeover protection, and phone and email verification.
- Customized workflows and user migration through AWS Lambda triggers.

For more information about user pools, see [Getting started with user pools](#) and the [Amazon Cognito user pools API reference](#).

Identity pools

With an identity pool, your users can obtain temporary AWS credentials to access AWS services, such as Amazon S3 and DynamoDB. Identity pools support anonymous guest users, as well as the following identity providers that you can use to authenticate users for identity pools:

- Amazon Cognito user pools
- Social sign-in with Facebook, Google, Login with Amazon, and Sign in with Apple
- OpenID Connect (OIDC) providers
- SAML identity providers
- Developer authenticated identities

To save user profile information, your identity pool needs to be integrated with a user pool.

For more information about identity pools, see [Getting started with Amazon Cognito identity pools \(federated identities\)](#) and the [Amazon Cognito identity pools API reference](#).

Implementation Example

1. Create a user pool in AWS Cognito:

The screenshot shows the AWS Management Console interface for the Amazon Cognito 'Users' page. The top navigation bar includes various AWS services like RDS, IAM, S3, CloudFront, Certificate Manager, Route 53, DynamoDB, Simple Notification Service, and API Gateway. The breadcrumb trail indicates the path: Amazon Cognito > User pools > Users.

Users [Info](#) [Delete user pool](#)

User pool overview

User pool name Users	ARN arn:aws:cognito-idp:us-east-1:623549510599:userpool/us-east-1_XzOGPLFVy	Created time February 5, 2023 at 21:34 CST
User pool ID us-east-1_XzOGPLFVy	Estimated number of users 0	Last updated time February 5, 2023 at 21:38 CST

Getting started

[Users](#) | [Groups](#) | [Sign-in experience](#) | [Sign-up experience](#) | [Messaging](#) | [App integration](#)

Users (0) [Info](#)
View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

[Refresh](#) [Delete user](#) [Create user](#)

User name [Filter](#) [Settings](#)

User name	Email address	Email verified	Confirmation status
-----------	---------------	----------------	---------------------

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

[Option+S]

N. Virginia

Admin @ 6235-4951-0599

RDS

IAM

S3

CloudFront

Certificate Manager

Route 53

DynamoDB

Simple Notification Service

API Gateway

Amazon Cognito > User pools > Users > App client: MyAppClient

App client: MyAppClient

Info

Delete

App client information

Edit

App client name MyAppClient	Authentication flow session duration 3 minutes	Created time February 5, 2023 at 21:34 CST
Client ID 3km8gpm8c0g4g950miklfspae v	Refresh token expiration 30 day(s)	Last updated time February 5, 2023 at 21:34 CST
Client secret -	Access token expiration 60 minutes	
Authentication flows ALLOW_REFRESH_TOKEN_AUTH ALLOW_USER_PASSWORD_AUTH ALLOW_USER_SRP_AUTH	ID token expiration 60 minutes	
	Advanced authentication settings Enable token revocation Enable prevent user existence errors	

Attribute read and write permissions

Choose the standard and custom attributes this app can read and write. Required attributes are locked as writable. We recommend that you set immutable custom attributes as writable to allow the app client to set initial values during sign-up.

Edit

Pinpoint analytics

Info

Enable

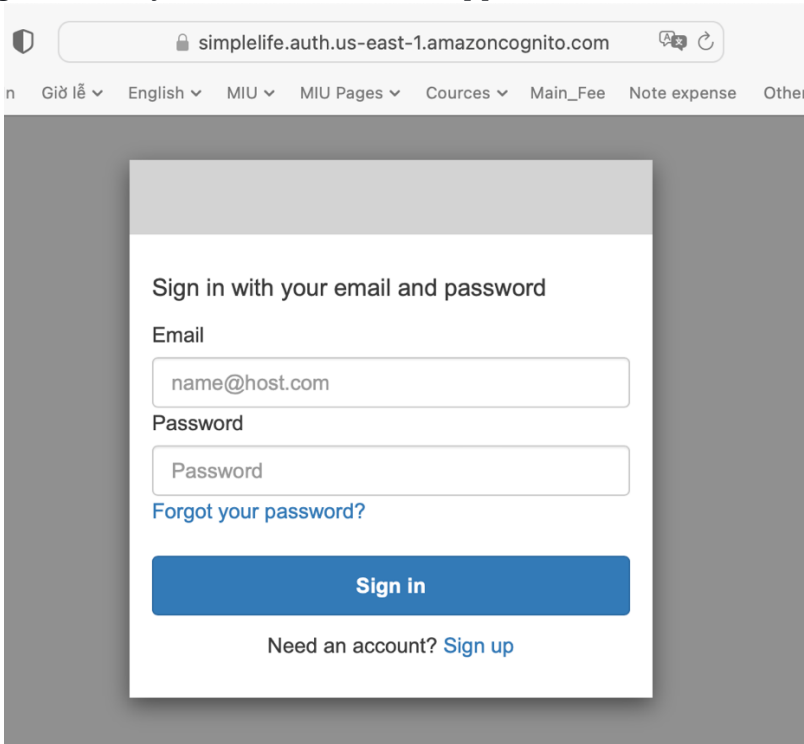
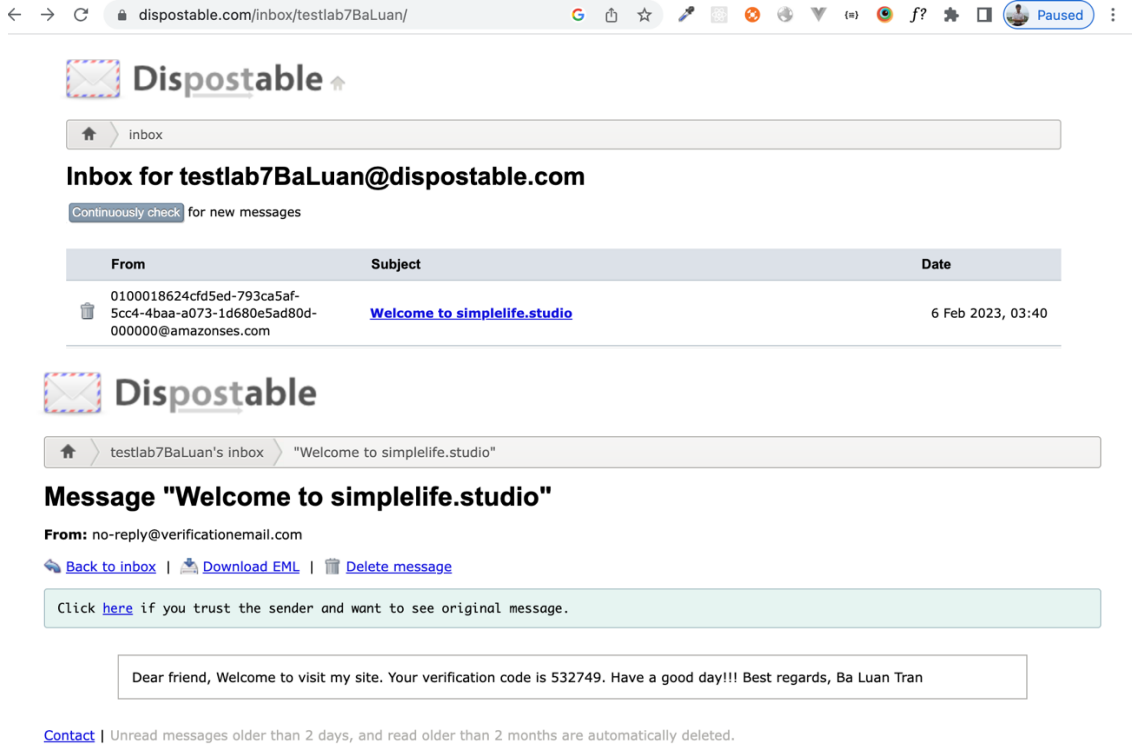
© 2023, Amazon Web Services, Inc. or its affiliates.











Privacy


Terms

Cookie preferences

- After created the AWS Cognito, it will provides authentication, authorization, and user management for your web and mobile apps


← → ↻ [dispostable.com/inbox/testlab7BaLuan/](#)           Paused ⋮


 **Dispostable** ↑

🏠 Inbox

Inbox for testlab7BaLuan@dispostable.com

[Continuously check](#) for new messages

From	Subject	Date
 0100018624cfd5ed-793ca5af-5cc4-4baa-a073-1d680e5ad80d-000000@amazonses.com	Welcome to simplelife.studio	6 Feb 2023, 03:40

 **Dispostable**

🏠 testlab7BaLuan's inbox > "Welcome to simplelife.studio"

Message "Welcome to simplelife.studio"

From: no-reply@verificationemail.com

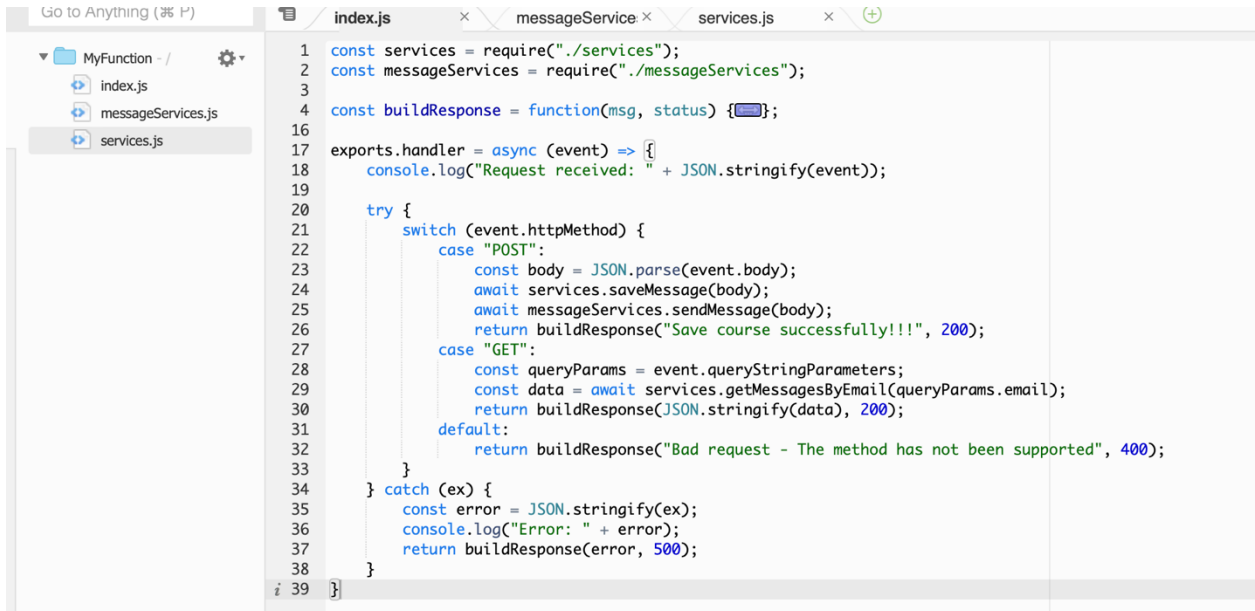
[Back to inbox](#) | [Download EML](#) | [Delete message](#)

Click [here](#) if you trust the sender and want to see original message.

Dear friend, Welcome to visit my site. Your verification code is 532749. Have a good day!!! Best regards, Ba Luan Tran

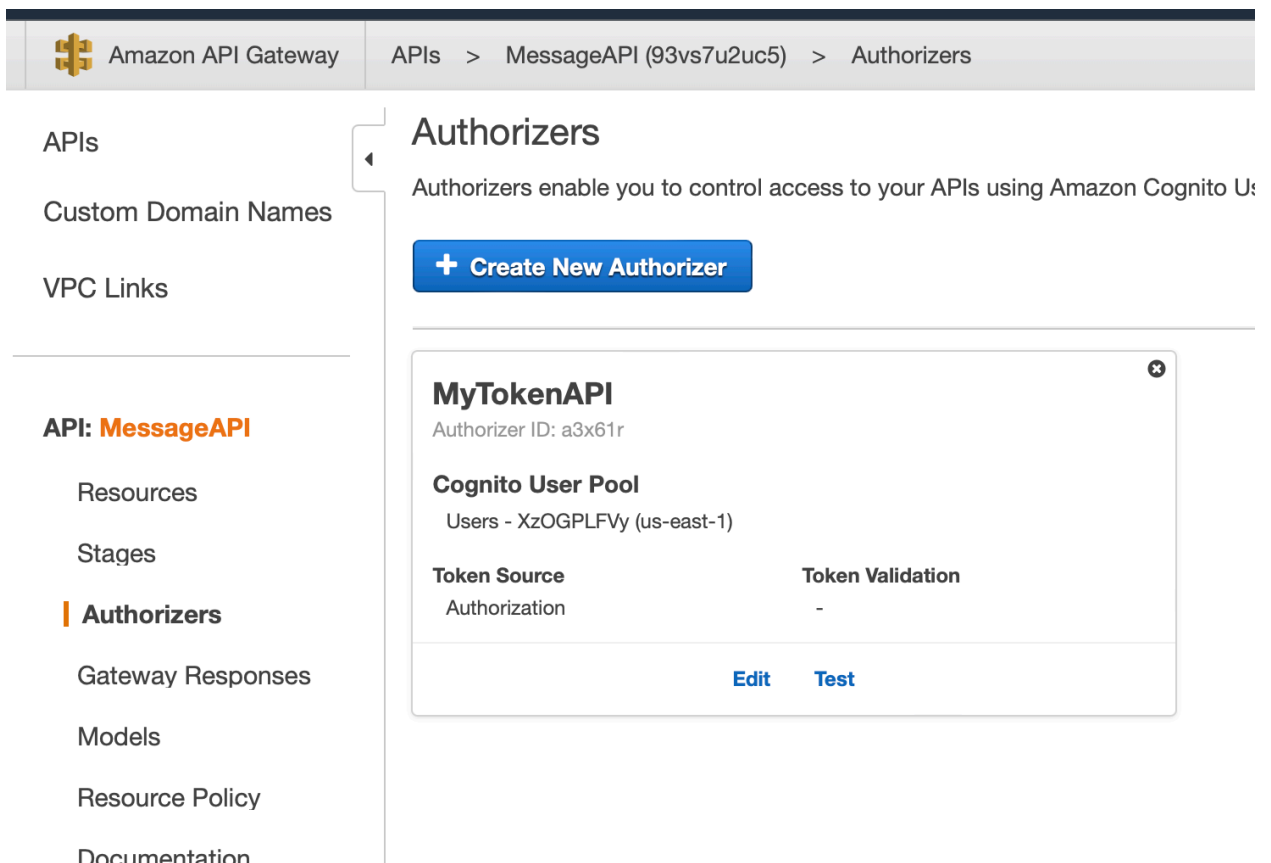
[Contact](#) | Unread messages older than 2 days, and read older than 2 months are automatically deleted.

2. Create a lambda to get/save messages from the website and create API Gateway to the lambda



```
1 const services = require("../services");
2 const messageServices = require("../messageServices");
3
4 const buildResponse = function(msg, status) {
5
6
7
8
9
10
11
12
13
14
15
16
17 exports.handler = async (event) => {
18   console.log("Request received: " + JSON.stringify(event));
19
20   try {
21     switch (event.httpMethod) {
22       case "POST":
23         const body = JSON.parse(event.body);
24         await services.saveMessage(body);
25         await messageServices.sendMessage(body);
26         return buildResponse("Save course successfully!!!", 200);
27       case "GET":
28         const queryParams = event.queryStringParameters;
29         const data = await services.getMessagesByEmail(queryParams.email);
30         return buildResponse(JSON.stringify(data), 200);
31       default:
32         return buildResponse("Bad request - The method has not been supported", 400);
33     }
34   } catch (ex) {
35     const error = JSON.stringify(ex);
36     console.log("Error: " + error);
37     return buildResponse(error, 500);
38   }
39 }
```

3. Create an Authorizers of the Cognito User Pool and assign to the API Gateway



Amazon API Gateway

APIs > MessageAPI (93vs7u2uc5) > Authorizers

APIs

Custom Domain Names

VPC Links

API: **MessageAPI**

Resources

Stages

Authorizers

Gateway Responses

Models

Resource Policy

Documentation

Authorizers

Authorizers enable you to control access to your APIs using Amazon Cognito U:

[+ Create New Authorizer](#)

MyTokenAPI

Authorizer ID: a3x61r

Cognito User Pool

Users - XzOGPLFVy (us-east-1)

Token Source	Token Validation
Authorization	-

[Edit](#) [Test](#)

APIs > MessageAPI (93vs7u2uc5) > Resources > /message (oqowne)

Resources

Actions ▾

/message Methods

▼ /

▼ /message

GET

OPTIONS

POST

GET

arn:aws:lambda:us-east-1:623549510599:functi...

Authorization COGNITO_USER_POOLS

API Key Required

OPTIONS

Mock Endpoint

Authorization None

API Key Not required


POST

Authorization None

API Key Required

Integrating Amazon Cognito into a website

1. Send some messages in the site (<https://www.simplelife.studio/>)

 JAVA DEVELOPER

HOMEABOUTWORKEDUCATIONCONTACTSIGN IN

Send A Message

Name

Thomas Tran

Email

thomas.tran@dispostable.com

Phone Number (optional)

2065937990

Subject

Can I get an appointment with you?

Your Message

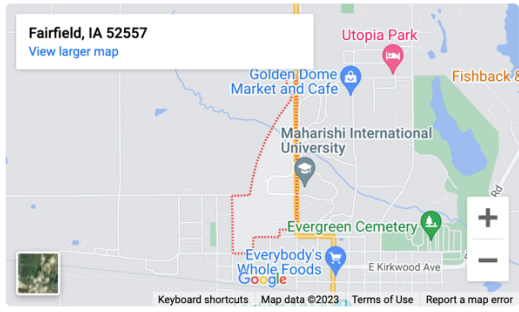
I'm Thomas.
I see you have good experiences about Java and Reactjs.
I want to see you in the next week to discuss technologies,
structure to apply a new project
Have a good day!

Send Message

Get in Touch

Whether you want to get in touch, talk about a project collaboration, or just say hi, I'd love to hear from you.


Simply fill the form and send me an email.



1000 N 4th St, Fairfield, Iowa 52557

(206) 593-7990

ba.tran@miu.edu



2. Sign up to the site with an email which used to send the messages

Sign Up

Email address
thomas.tran@dispostable.com

Password
.....

Verification Code
Your Verification Code

[Sign Up](#) [Cancel](#)

Already have an account? [Sign in](#)

- An verification code will be sent to your email



thomas.tran's inbox > "Welcome to simplelife.studio"

Message "Welcome to simplelife.studio"

From: no-reply@verificationemail.com

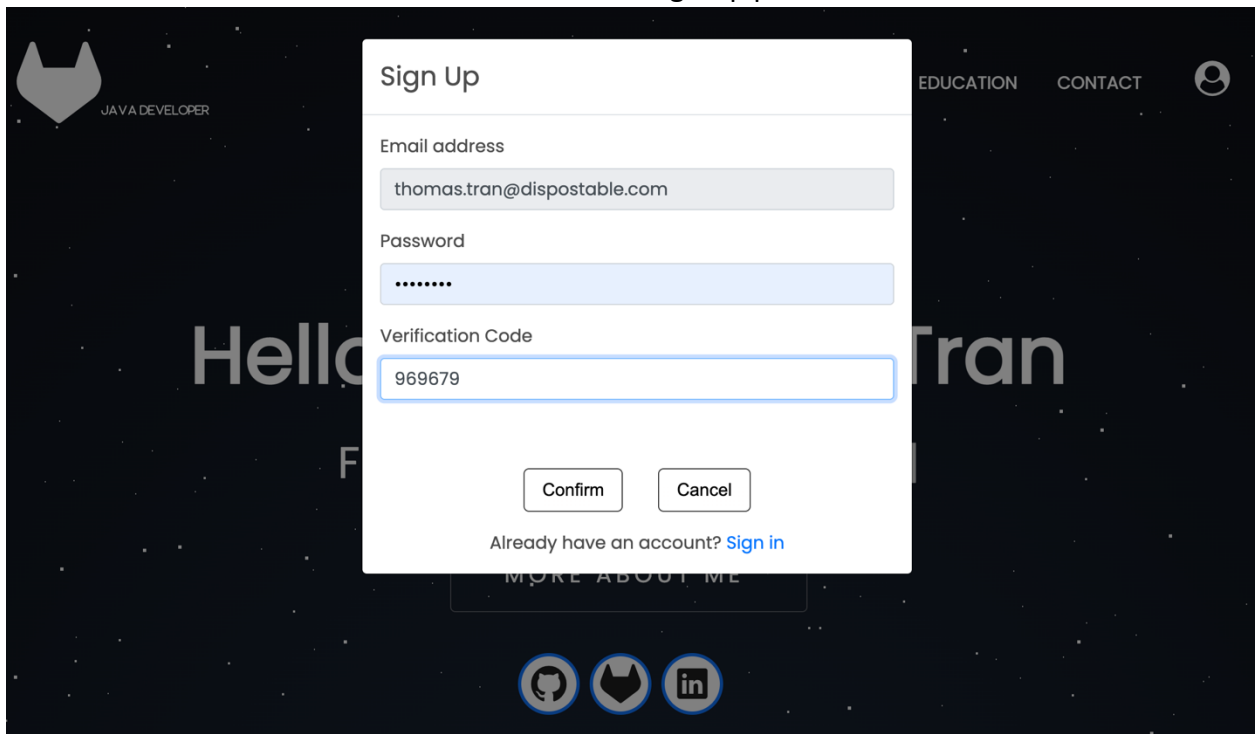
[Back to inbox](#) | [Download EML](#) | [Delete message](#)

Click [here](#) if you trust the sender and want to see original message.

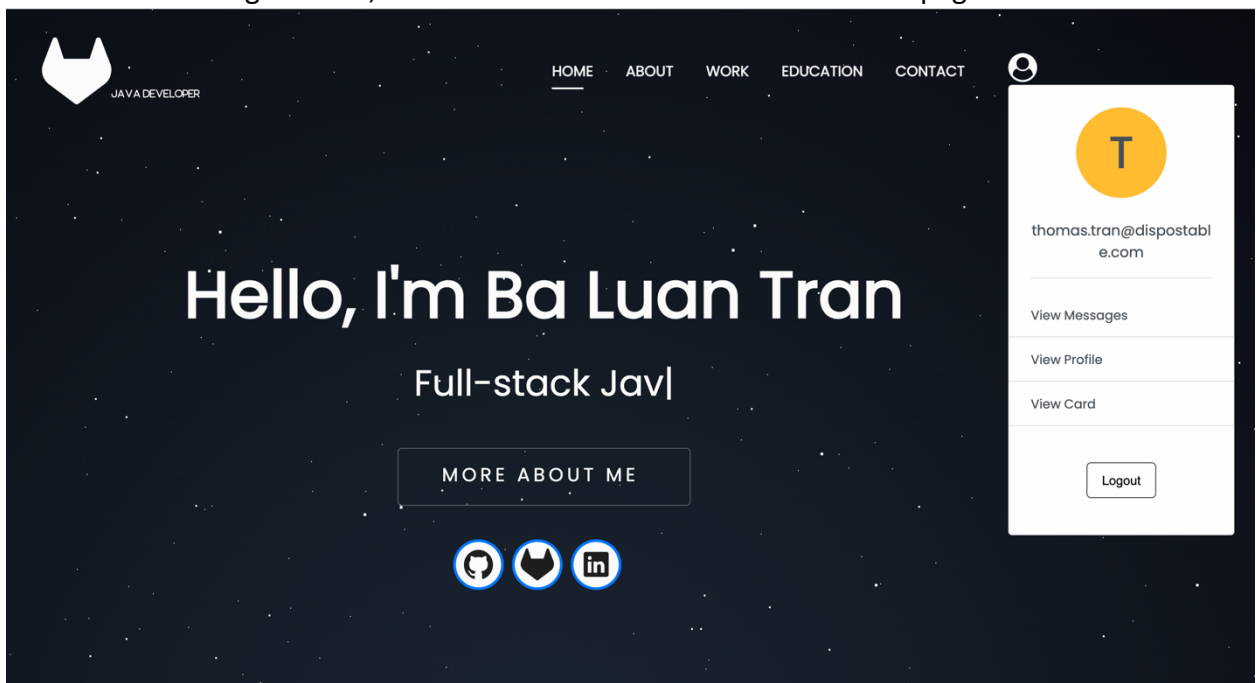
Dear friend, Welcome to visit my site. Your verification code is 969679. Have a good day!!! Best regards, Ba Luan Tran

[Contact](#) | Unread messages older than 2 days, and read older than 2 months are automatically deleted.

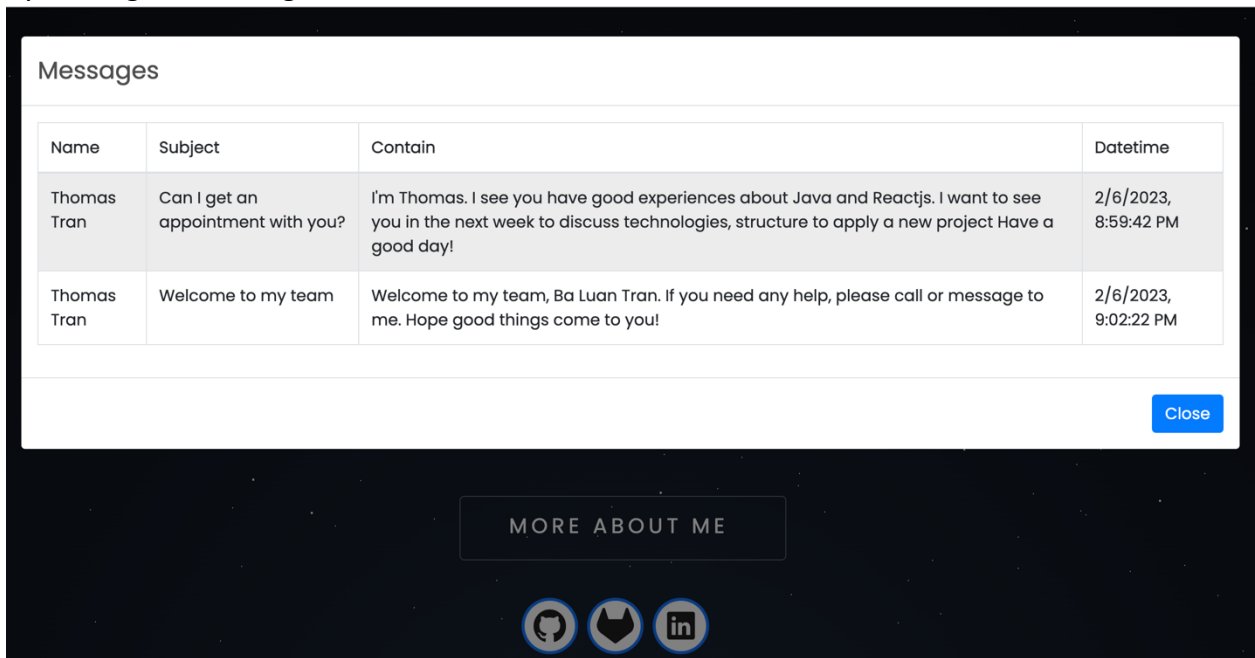
- Enter the code and click Confirm to finish the Sign Up process



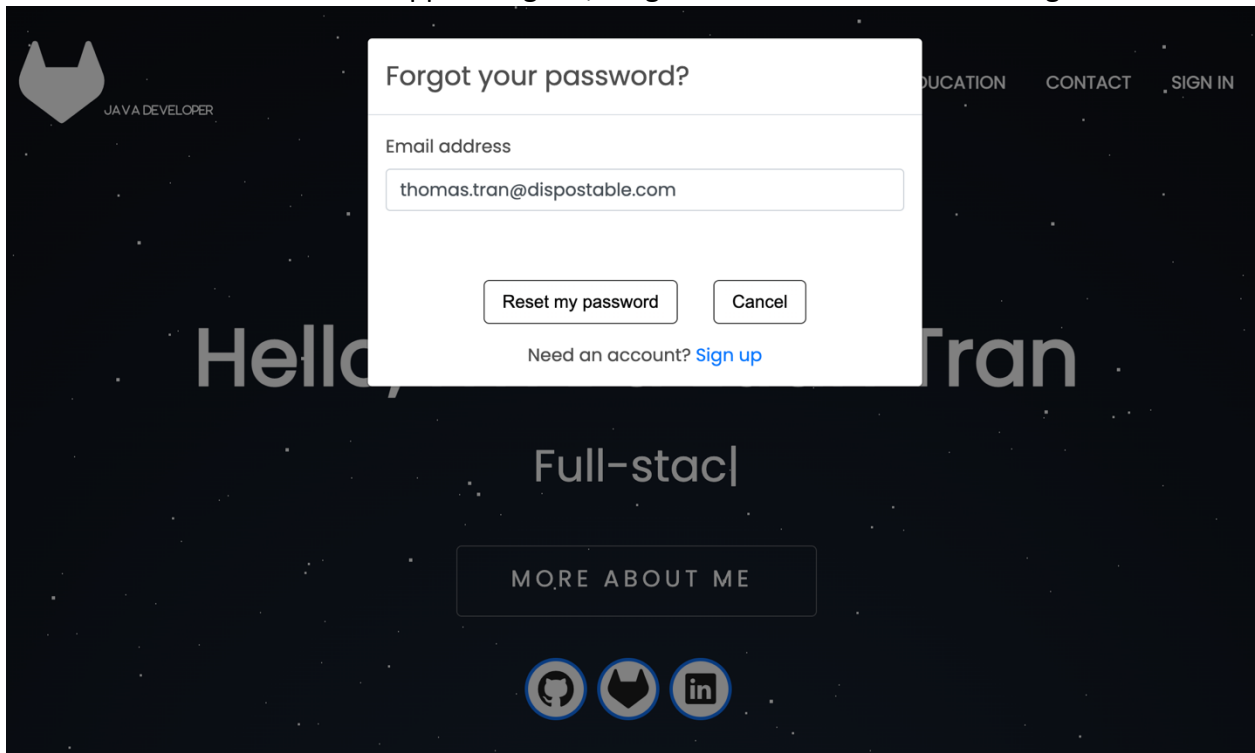
3. After successful registration, we will see the Profile icon in the header page

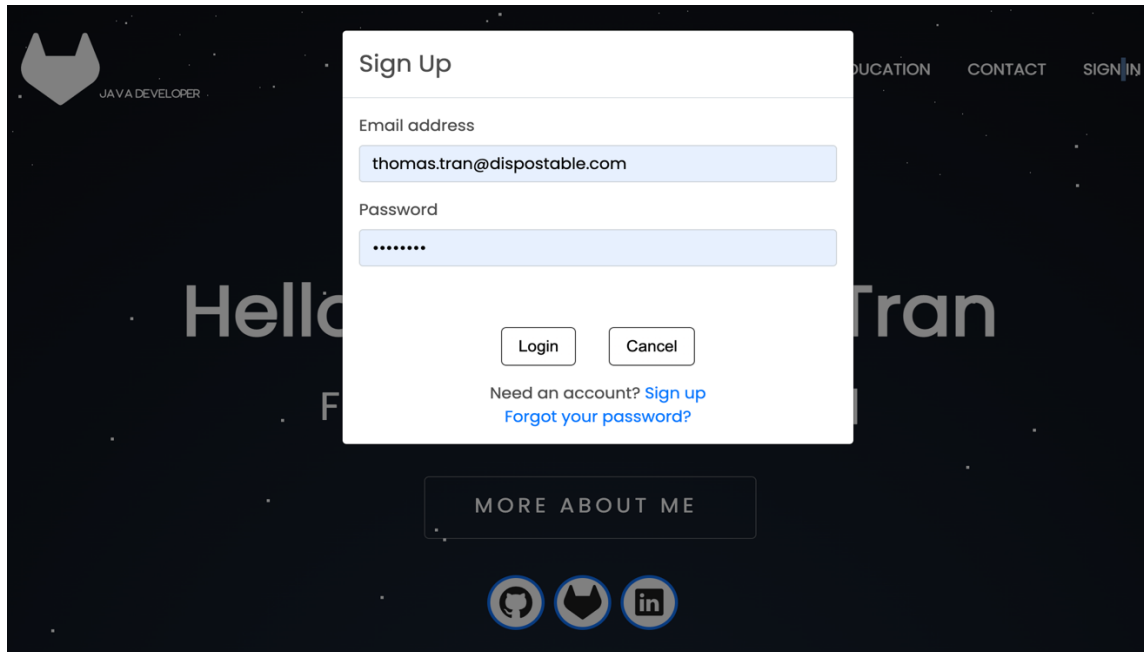


4. If we click the icon, the Profile dropdown will be displayed and we can see all sent messages by clicking the Message menu



5. Beside that the website also supports Sign In, Forgot Password with the AWS Cognito:





Resources:

S3: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>

Lambda https://aws.amazon.com/lambda/getting-started/?trk=dca4b539-ba5f-4c78-bd55-e8e5f7a26221&sc_icampaign=lambda_ict_gs_functions&sc_icontent=awssm-11768_engage&sc_iplace=aws-console-lambda

SNS: <https://docs.aws.amazon.com/sns/latest/dg/welcome.html>

API Gateway: <https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

AWS Cognito: https://docs.amazonaws.cn/en_us/cognito/latest/developerguide/what-is-amazon-cognito.html

Route53: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>

DynamoDB:
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html>

Integration between ReactJS app and AWS Cognito:
<https://www.npmjs.com/package/amazon-cognito-identity-js>

Github: <https://github.com/luan-tran-89/cs516/tree/main/final-project>