

CryptoWatcher Project Report

For the large project in this class, I've built the CryptoWatcher, a serverless single-page application that lets people see the prices of cryptocurrencies, convert their prices, and see their portfolios with charts. My goal in this project was to learn and apply cloud technologies while building an application that people can actually use.

Development

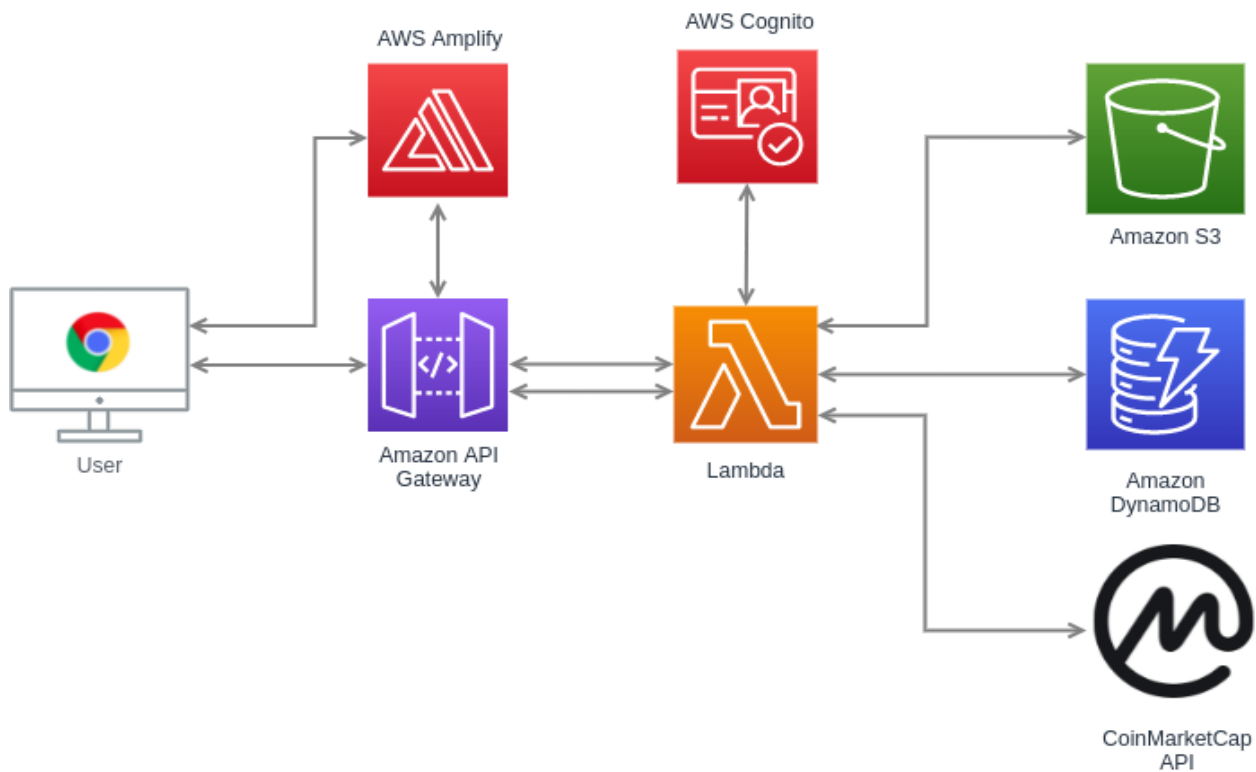
I've used Amazon API Gateway, lambda functions, and DynamoDB for the backend. I've used a javascript library called React in the frontend for building the Single Page Application with AJAX calls. Also, I've used Amplify on the backend for the authentication and deployment, and I've used it on the frontend for UI components of the authentication.

First of all, I initiated the front-end development environment and the AWS environment separately. On the front end part, I set up the React environment and installed the required libraries and components. On the backend part, I started with creating the AWS account and user accounts. Then I read the documentation of AWS Lambda functions. After that, I've created initial test functions and tested them to get familiar with the environment. I integrated the test functions with the API Gateway to access those functions publicly. Then I got the API key from CoinMarketCap and started implementing the functions on Lambda. Meanwhile, I've integrated the front-end environment with AWS Amplify to handle authentication and continuous deployment of the application. This integration included front-end components of the authentication flow and their integration into React architecture.

Workflow of the Application

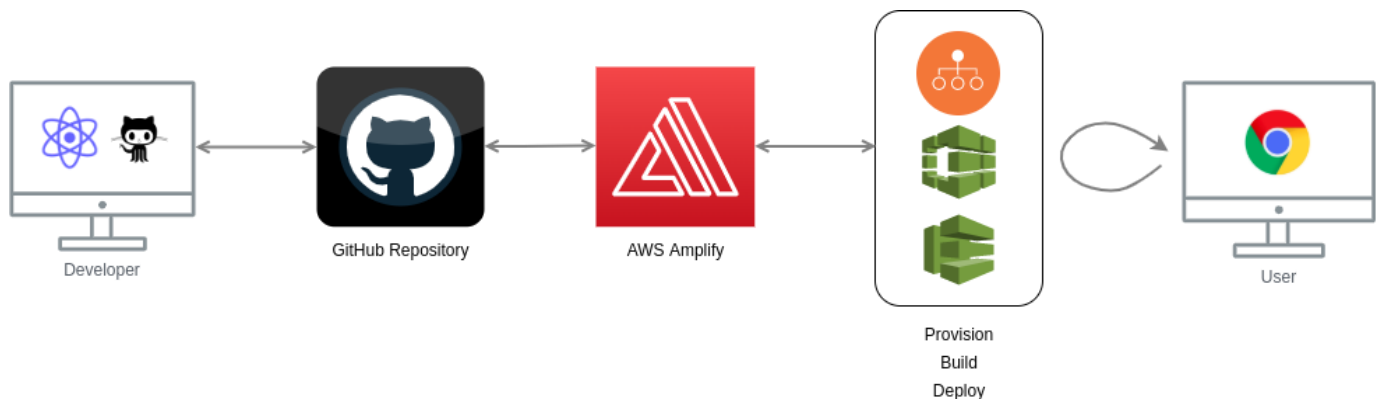
Cryptowatcher is a web application that people can access through web browsers. When a user first loads the web application, initial API call to fetch the current prices of cryptocurrencies are made to the specified Lambda function through API gateway. The lambda function then makes another API call to CoinMarketCap to get the current data of the currencies. The application supports 12 cryptocurrencies at the moment and in each load of the web page, 12 API calls are made to the CoinMarketCap API by a lambda function. After that, the front-end receives the data and displays it in a table that shows various information about the currencies such as their prices, daily change, monthly change, and market cap along with the logo and the weekly graph of the currency.

Authentication is done through AWS Amplify which also uses the Lambda functions to handle the authentication process. I used S3 to store the images. Wallet function is responsible for storing and retrieving a user's wallet from the Amazon DynamoDB



Continuous Deployment

The figure below shows the workflow of the continuous deployment mechanism. I used AWS Amplify to set this up. I specified the main branch of the Github repository as the trigger to use AWS Amplify to build, provision, and deploy the latest version of the application automatically. I've tested this mechanism 10 times and the average build time was about six minutes.



The Result

As a result, I've built a serverless web application that currently offers three features to users. First of these features is the table of cryptocurrencies that people can use to see statistical data in a visual manner. Users are able to sort or filter any of the given columns and make searches in a column. Most of the endpoints are updated every minute according to the CoinMarketCap API, therefore the data is pretty reliable and up to date.

CRYPTOWATCHER

PRICE CONVERTER

SHOW MY WALLET

EXCHANGE INFO














SIGN OUT

<input type="checkbox"/>	ID	Logo	Name	Price	Daily Change %	30d Change %	Market Cap	Daily Volume %	Daily Change %	Last 7 Days
<input type="checkbox"/>	1		Bitcoin	\$49194.7922...	-7.03603915	-20.06977232	\$929359088236.8575	\$61714476437.99495	52.3837	
<input type="checkbox"/>	2		Ethereum	\$4121.44216...	-1.07179353	-8.75002133	\$488838229467.57666	\$38780058191.72989	45.889	
<input type="checkbox"/>	3		Litecoin	\$162.491261...	-11.62214624	-19.08295145	\$11230999069.639814	\$3690494832.085583	98.2051	
<input type="checkbox"/>	4		XRP	\$0.83518608...	-8.15027703	-29.48787957	\$39460284116.28928	\$6972188662.987531	147.9944	
<input type="checkbox"/>	5		Binance C...	\$561.995785...	-3.92822158	0.30046737	\$93741542119.34131	\$4381027021.660338	70.2823	
<input type="checkbox"/>	6		Dogecoin	\$0.17400331...	-11.45708405	-33.04521135	\$23035627902.429237	\$4405291701.3513...	227.3565	
<input type="checkbox"/>	7		Cardano	\$1.39311065...	-8.46298247	-29.57252086	\$46409039116.545784	\$4038478009.3994...	52.1857	
<input type="checkbox"/>	8		Solana	\$196.645801...	-3.16894387	-19.46991685	\$60103014573.19141	\$5766708224.803383	38.2473	

1-12 of 12

The second feature is the price converter. In this component, users are able to convert any of their cryptocurrencies to any other currency including the USD. The updates to the input fields are triggered at every input and results are directly displayed by React. Users are also able to change inputs in each direction, change the currency option, add and remove converters.

Convert Different Currencies



Bitcoin

Number

2

↻

USD

Number

98389.5844564

Ethereum

Number

10

↻

Binance Coin

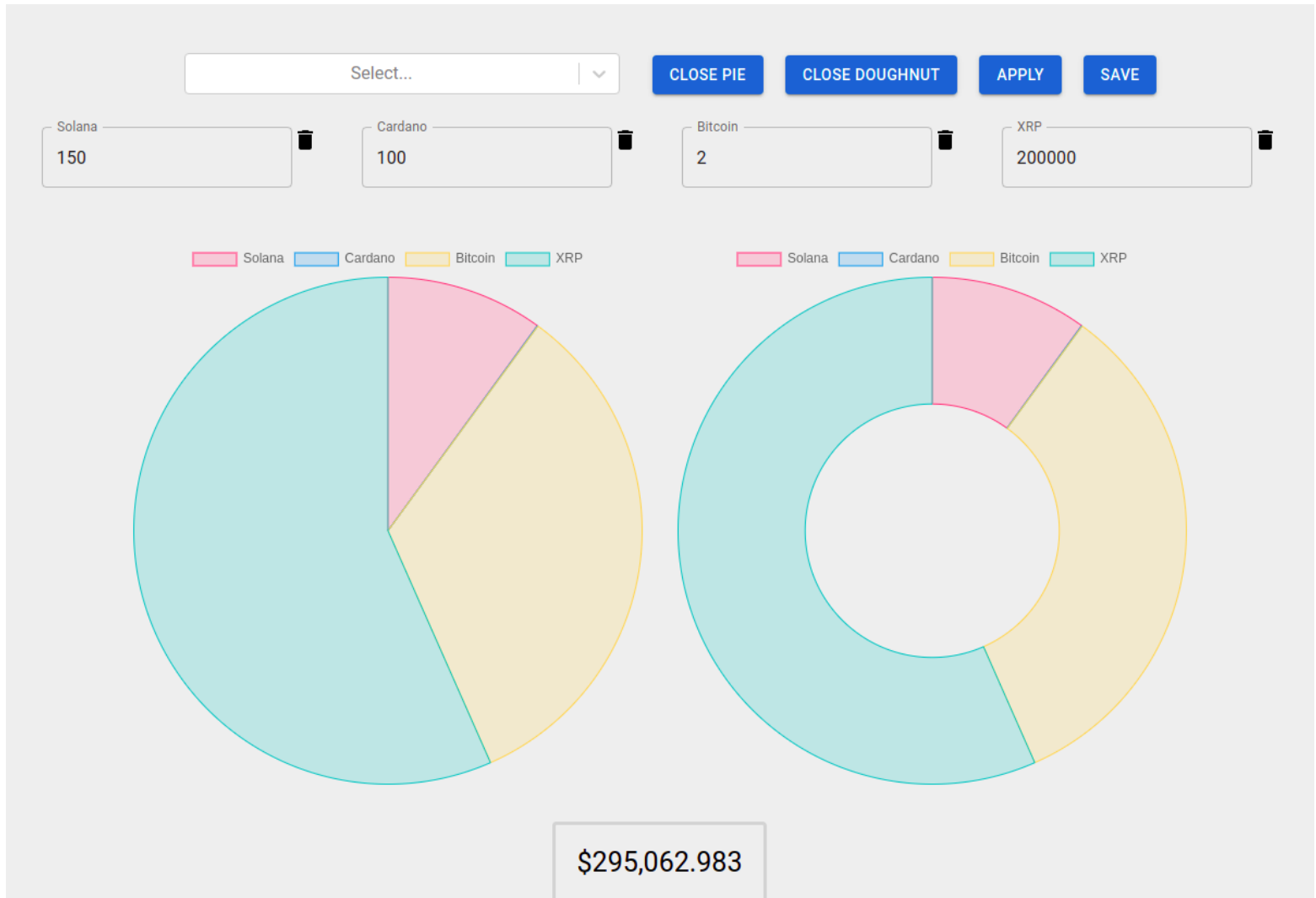
Number

73.33582

ADD CONVERTER

REMOVE CONVERTER

The third feature is the wallet page. On this page, users are able to see the distribution of their cryptocurrencies in terms of their values. Users can add or remove any given amount of the given coins and see the result of distribution with visual components. I've included two kinds of charts which are the pie chart and the doughnut chart to represent the data. Users can also save their wallets after logging in so that they will be able to see the same wallet next time they log in. Users can also see the sum value of their wallets at the bottom.



Conclusion

In this project, I was able to learn and experience how convenient it has become to leverage the use of cloud computing services to build applications. I took advantage of the serverless lambda functions to build an application without the burden of renting, configuring and maintaining servers. I've also made use of countless advantages of AWS services such as AWS Amplify, DynamoDB, API Gateway, and S3 buckets. I was able to build this application without any budget by using the free tiers of both the CoinMarketCap API and AWS Services.

I was able to get the backend up and running with multiple API endpoints which are REST and HTTP with API Gateway. I was able to store and retrieve data using DynamoDB. I was able to run backend functions without hosting them through lambda

functions. I was able to authenticate people by integrating the service into the application instead of building authentication from scratch.

The application is hosted by AWS Amplify and is publicly available through the following URL: <https://main.dae70rht8vanf.amplifyapp.com/>

The project source code is maintained in GitHub:
<https://github.com/unsel/cryptowatcher>

APPENDIX

1) Lambda Functions

Functions (9)							
Last fetched in 0 seconds ↻ Actions Create func							
<input type="text" value="Filter by tags and attributes or search by keyword"/>							
	Function name	Description	Package type	Runtime	Code size	Last modified	
<input type="checkbox"/>	amplify-login-define-auth-challenge-ca5d919b	-	Zip	Node.js 12.x	1.7 kB	17 days ago	
<input type="checkbox"/>	amplify-login-create-auth-challenge-ca5d919b	-	Zip	Node.js 12.x	1.1 kB	17 days ago	
<input type="checkbox"/>	amplify-amplify872e5ac1a3-UpdateRolesWithIDPFuncti-NdBUWZaNlZq7	-	Zip	Node.js 12.x	2.8 kB	7 days ago	
<input type="checkbox"/>	walletFunction	-	Zip	Node.js 14.x	777.0 byte	2 days ago	
<input type="checkbox"/>	amplify-login-custom-message-ca5d919b	-	Zip	Node.js 12.x	2.2 kB	17 days ago	
<input type="checkbox"/>	getAllCurrencies2	-	Zip	Python 3.6	1.0 kB	2 days ago	
<input type="checkbox"/>	getMedalPrices	-	Zip	Python 3.6	600.0 byte	2 days ago	
<input type="checkbox"/>	amplify-login-verify-auth-challenge-ca5d919b	-	Zip	Node.js 12.x	2.5 kB	4 days ago	
<input type="checkbox"/>	amplify-amplify872e5ac1a35e4--UserPoolClientLambda-bAZM1kZwoCQv	-	Zip	Node.js 12.x	2.4 kB	7 days ago	

2) API Gateway

APIs (2)				
<input type="text" value="Find APIs"/>				
	Name	Description	ID	Protocol
<input type="radio"/>	getAllCurrencies2-API	Created by AWS Lambda	99gz9lge5l	REST
<input type="radio"/>	walletHttpApi		2jbjhydie7	HTTP

3) DynamoDB Wallet Table

<input type="checkbox"/>	id	wallet
<input type="checkbox"/>	cfdb691e-5...	{ "Litecoin" : { "N" : "5" }, "Solana" : { "N" : "5" }, "Ethereum" : { "N" : "5" }, "Bitcoin" : { "N" : "5" } }
<input type="checkbox"/>	five	{ "Litecoin" : { "N" : "300" }, "Solana" : { "N" : "500" }, "Bitcoin" : { "N" : "2" }, "Ethereum" : { "N" : "120" } }
<input type="checkbox"/>	514e94e6-f...	{ "Solana" : { "N" : "150" }, "Cardano" : { "S" : "100" }, "Bitcoin" : { "N" : "2" }, "XRP" : { "S" : "200000" } }

4) S3 Buckets

coinpng

Info

Publicly accessible

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (9)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.

↺

Copy S3 URI

Copy URL

Download

Open

Delete










Actions

Create folder

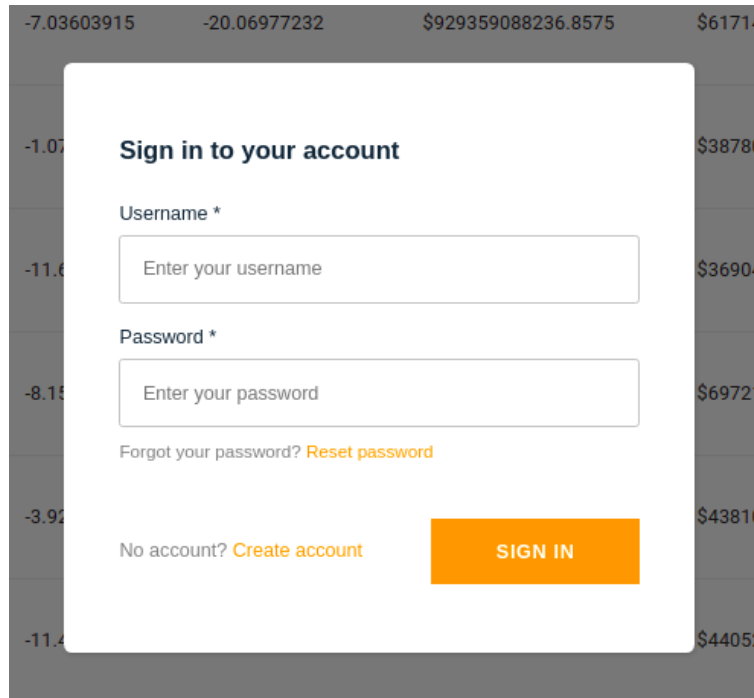
Upload

Find objects by prefix

Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	 Avalanche.png	png	December 1, 2021, 00:54:25 (UTC+03:00)	
<input type="checkbox"/>	 Binance Coin.png	png	December 1, 2021, 00:54:25 (UTC+03:00)	
<input type="checkbox"/>	 bitcoin.png	png	December 1, 2021, 00:54:25 (UTC+03:00)	
<input type="checkbox"/>	 Cardano.png	png	December 1, 2021, 00:54:26 (UTC+03:00)	
<input type="checkbox"/>	 Ethereum.png	png	December 1, 2021, 00:54:26 (UTC+03:00)	
<input type="checkbox"/>	 Litecoin.png	png	December 1, 2021, 00:54:27 (UTC+03:00)	
<input type="checkbox"/>	 SHIBA INU.png	png	December 1, 2021, 00:54:27 (UTC+03:00)	
<input type="checkbox"/>	 Solana.png	png	December 1, 2021, 00:54:27 (UTC+03:00)	
<input type="checkbox"/>	 XRP.png	png	December 1, 2021, 00:54:28 (UTC+03:00)	

5) Sign In Component



A mockup of a sign-in component. It features a white card with rounded corners on a dark gray background. The card has a title "Sign in to your account" in bold. Below the title are two input fields: "Username *" and "Password *", each with a placeholder "Enter your username" and "Enter your password" respectively. Below the password field is a link "Forgot your password? Reset password" in orange. At the bottom left is a link "No account? Create account" in orange. At the bottom right is an orange button labeled "SIGN IN".

Sign in to your account

Username *

Enter your username

Password *

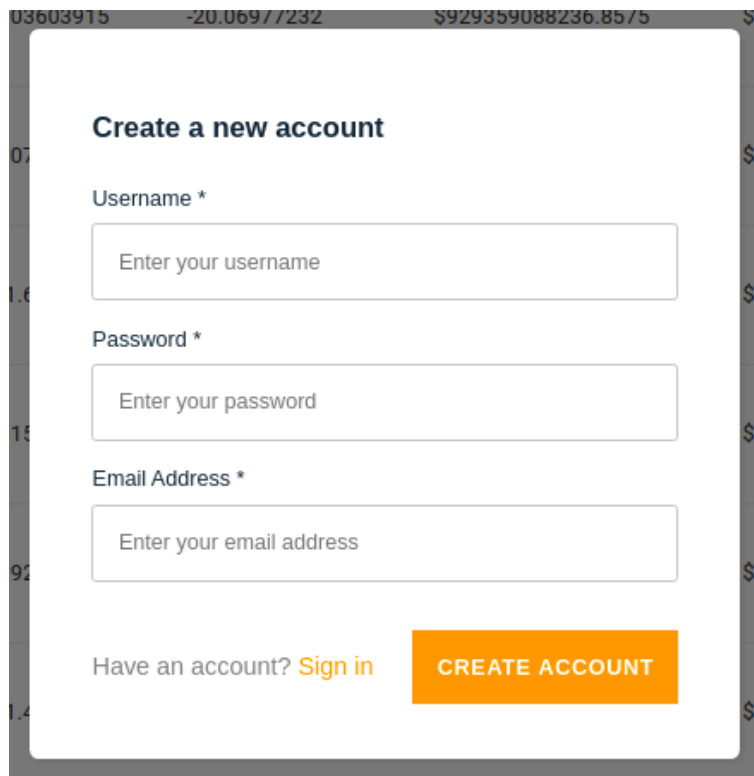
Enter your password

Forgot your password? [Reset password](#)

No account? [Create account](#)

SIGN IN

6) Sign Up Component



A mockup of a sign-up component. It features a white card with rounded corners on a dark gray background. The card has a title "Create a new account" in bold. Below the title are three input fields: "Username *" with placeholder "Enter your username", "Password *" with placeholder "Enter your password", and "Email Address *" with placeholder "Enter your email address". At the bottom left is a link "Have an account? Sign in" in orange. At the bottom right is an orange button labeled "CREATE ACCOUNT".

Create a new account

Username *

Enter your username

Password *

Enter your password

Email Address *

Enter your email address

Have an account? [Sign in](#)

CREATE ACCOUNT