

Here is some information on you can access our project, and other miscellaneous information.

LIVE BUILD:

The link to the live site is:

<https://master.d2nmsllsuguwvm.amplifyapp.com/>

This is the current build of the application that runs the code seen in the repo. To use the site, you can sign up for an account. You'll need to use a real Email because we send a verification code. You may have to reload the page after logging in, or just click the sign in button a second time.

SELLING ITEMS:

To sell an item, go to the add item tab, and at the bottom you should see a create stripe account button, if you are a new user who hasn't set up a stripe account with us yet. First click on that button and set up your stripe account. You may have to disable your ad blocker/popup blocker, because we will be opening a new window to direct you to sign up through stripe. When you are prompted for a phone number, there's a button that gives you the option of using a test phone number. Make sure to click that. Then when asked for a verification code, click the button that says test code. Finally, when setting up a bank account for payment, click the button to set up a dummy/test bank account. All the other fields can be fake emails, names, websites, etc. After setting up the stripe account you should see some text that says whether it was set up correctly or not. You will not be able to sell items, until you set up your stripe account, because that is how we know where to send the money to. After setting up your account, you can go back to our page, and the create stripe account button should be replaced with a submit button. Now you can fill out the form with the info of the item you want to sell and click submit. You can see a list of items you are currently selling on the selling page.

BIDDING/PAYING:

From the home page you can click on an item and enter a bid for it. You can see the items you are bidding on, in your bid cart. You can see your current bid on that item there. You can see what the highest bid on the item is back on the item's details page and adjust your bid if the new bid is higher than your current bid. When the timer is up, the person with the highest bid should be able to pay for the item in their bid cart. No other people will be able to pay for that item except the winner. When you click on the pay button, enter in this credit card for testing: 4000 0025 0000 3155. You can use any 3 digit CVC, future expiration date, and zip code with this test card.

CHECKING IF THE PAYMENT WENT THROUGH:

Email/Text us your Emails so we can send you an invitation to join our stripe team. There will be a link in the email to access the dashboard for our stripe app. You will be able to see the current users registered in the app, and how much money they have been transferred so you can verify if payments went through. You can email yeontaekj@gmail.com or text 7143924742 so we can set you guys up in our stripe console.

TLDR;

Live site link: <https://master.d2nmsllsuguwvm.amplifyapp.com/>

Disable adblock/popup block

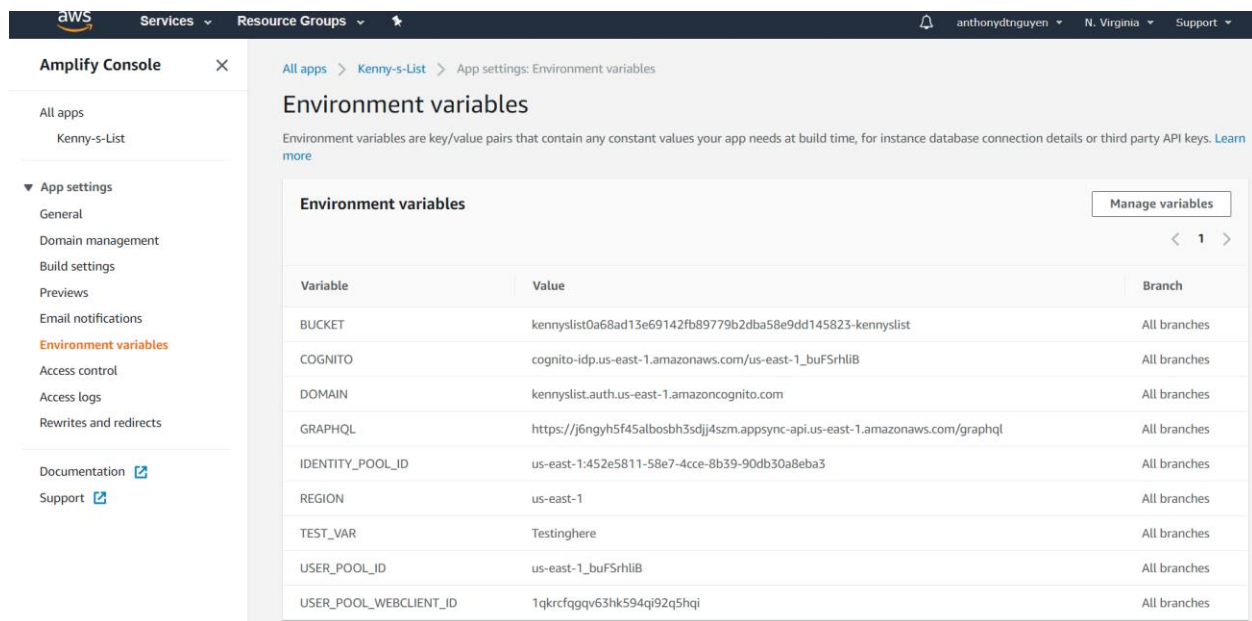
Pay with 4000 0025 0000 3155, any 3 digit cvc, future expiration date, and zip code.

Send your emails to yeontaekj@gmail.com so we can set you up in our stripe console if you want to verify that your payments went through.

Feel free to ask us if you have any questions on actually using the site or if you have any trouble accessing the live build.

MISCELLANEOUS INFORMATION:

A lot of our backend is set up and configured through the AWS Amplify framework, and so our backend is set up mostly through our AWS Console. There really isn't any backend "code" for us to have in the repo for setting up our backend resources. It's mostly just configuration, and it gets built and deployed by AWS. So we will screenshot some things to show you how the backend is set up and configured. We consider the following things within our trust boundaries.



The screenshot shows the AWS Amplify Console interface. On the left is a sidebar with navigation links: 'All apps', 'Kenny-s-List', 'App settings' (expanded), 'General', 'Domain management', 'Build settings', 'Previews', 'Email notifications', 'Environment variables' (highlighted), 'Access control', 'Access logs', 'Rewrites and redirects', 'Documentation', and 'Support'. The main content area is titled 'Environment variables' and shows a table of variables for the 'Kenny-s-List' app. The table has three columns: 'Variable', 'Value', and 'Branch'. The variables listed are BUCKET, COGNITO, DOMAIN, GRAPHQL, IDENTITY_POOL_ID, REGION, TEST_VAR, USER_POOL_ID, and USER_POOL_WEBCLIENT_ID. All variables are set for 'All branches'.

Variable	Value	Branch
BUCKET	kennyslist0a68ad13e69142fb9779b2dba58e9dd145823-kennyslist	All branches
COGNITO	cognito-idp.us-east-1.amazonaws.com/us-east-1_buF5rhliB	All branches
DOMAIN	kennyslist.auth.us-east-1.amazoncognito.com	All branches
GRAPHQL	https://f6ngyh5f45albosh3sdij4szm.appsync-api.us-east-1.amazonaws.com/graphql	All branches
IDENTITY_POOL_ID	us-east-1:452e5811-58e7-4cce-8b39-90db30a8eba3	All branches
REGION	us-east-1	All branches
TEST_VAR	Testinghere	All branches
USER_POOL_ID	us-east-1_buF5rhliB	All branches
USER_POOL_WEBCLIENT_ID	1qkrqgqv63hk594qi92q5hqi	All branches

Environment variables set up in our backend to configure our resources and link them together.

aws Services Resource Groups

anthonydtnghuyen N. Virginia Support

AWS AppSync X

APIs
kennyslist-kennyslist
Schema
Data Sources
Functions
Queries
Caching
Settings
Monitoring

AWS AppSync > Kennyslist-kennyslist > Data Sources

Data Sources

Connect existing AWS resources to your API. [Info](#)

Data Sources Edit Delete Create data source

Q

	Name ▲	Type ▼	Resource ▼
<input type="radio"/>	ItemTable	AMAZON_DYNAMODB	ItemTable
<input type="radio"/>	KennysListUserTable	AMAZON_DYNAMODB	KennysListUserTable
<input type="radio"/>	LatestUserBidTable	AMAZON_DYNAMODB	LatestUserBidTable
<input type="radio"/>	UserBidsTable	AMAZON_DYNAMODB	UserBidsTable

AWS AppSync X

APIs
kennyslist-kennyslist
Schema
Data Sources
Functions
Queries
Caching
Settings
Monitoring

API Details

API URL
 Copy

API ID
 Copy

Primary auth mode
AMAZON COGNITO_USER_POOLS

Edit API name

Edit API name
Enter a name for your API.

Default authorization mode

API-level
The primary authorization mode for your schema. This applies to all fields and types by default.

Configuration

GraphQL API set up to work only with a specific AWS Amplify application build.

aws

Services

Resource Groups

anthonydtnghuyen

N. Virginia

Support

DynamoDB

Dashboard

Tables

Backups

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Delete table

Filter by table name

Choose a table ...

Actions

Name

AllUsers

FoodTable

ItemTable

kennyslist-cplfw262jndzhilo67cmwd7

KennysListUserTable

LatestUserBidTable

RecipeTable

RestaurantTable

Todo-nmfooxd6drfanigazov7hkb16m-l

TodoTable

UpdatedUserTable

UserBidsTable

UserTable

ItemTable

Close

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

Backups

Contributor Insights

More

Create item

Actions

Scan: [Table] ItemTable: itemID

Viewing 1 to 9 Items

Scan

[Table] ItemTable: itemID

Add filter

Start search

itemID	category	condition	description
180276ct-009d-4c9c-a703-8a644999fddf	Other	New	png
2d16b1c3-a4ef-4e96-a3da-796505f93338	Electronics	Used	Live out your romantic fantasies in this game
4c8f1353-f581-490d-8a44-a168e1dee5cb	Other	New	dog
53f0328c-99af-4e50-a848-9bf35bcebe9	Other	New	123444
63638750-e284-43c3-a9cb-e401852ee993	Other	New	wed
7bfca59b-82ca-461d-960a-b5f6c9abfe65	Other	New	wef
9756b07-25f2-4b83-9329-3f3c56b49dcb	Other	New	asdfa
dff31813-93e0-4b73-ad12-411b8d22eb63	Other	Used	wwef
fd2cef5e-5c93-402a-b014-fb7bfa80e2e	Other	New	asdfadsf

aws

Services

Resource Groups

anthonydtnghuyen

N. Virginia

Support

DynamoDB

Dashboard

Tables

Backups

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Delete table

Filter by table name

Choose a table ...

Actions

Name

AllUsers

FoodTable

ItemTable

kennyslist-cplfw262jndzhilo67cmwd7

KennysListUserTable

LatestUserBidTable

RecipeTable

RestaurantTable

Todo-nmfooxd6drfanigazov7hkb16m-l

TodoTable

UpdatedUserTable

UserBidsTable

UserTable

KennysListUserTable

Close

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

Backups

Contributor Insights

More

Create item

Actions

Scan: [Table] KennysListUserTable: username

Viewing 1 to 19 Items

Scan

[Table] KennysListUserTable: username

Add filter

Start search

username	accountID	randstring
Example	acct_1GEIYVJnqkbaON6	
ExampleForClass	acct_1GEja4EA4WBU7oVl	
FoodBar	acct_1GGLysA7VIMrB1H8	
Joe		2042704837339060795
Leroy_S1	acct_1GlrIBB7z8fORYVx	
Tiger	acct_1GEglKErGPMtucBO	
admin	acct_1GEIGkHUHKV9kJW6	
cat	acct_1GEg9oAP8HqyjIAM	
dog	acct_1GEIgel9OCUNJKBW	
kevfffin	acct_1GEgalDatAhGM2ld	

LatestUserBidTable Close

Overview **Items** Metrics Alarms Capacity Indexes Global Tables Backups Contributor Insights More

Create item Actions

Scan: [Table] LatestUserBidTable: lubtProductID Viewing 1 to 23 items

Scan [Table] LatestUserBidTable: lubtProductID

Add filter

Start search

lubtProductID	BidAmt	Username
0909c914-de69-4fdb-8eb5-82dbfc2305e1	12	Leroy_S1, (seller)
180276cf-009d-4c9c-a703-8a644999fbdf	11	newuserstest
192bbcf2-7e5b-491e-bd9e-deaf3126a53b	10	Tiger
29ee4460-b1f0-4772-94ed-bfb9f684d82e	12	Leroy_S1, (seller)
2d16b1c3-a4ef-4e96-a3da-796505f93338	25	khzheng24998, (seller)
34854271-deae-44cc-8279-5ef4c9eccdd9	5	Tiger
4c8f1353-f581-490d-8a44-a168e1dee5cb	0	Tiger, (seller)
523038f2-e1cb-4004-9d06-d58f3ae7554e	14	Leroy_S1
53f0328c-99af-4e50-a848-9bff35bcebe9	17	Leroy_S1

DynamoDB Tables

Amazon S3 > kennyslist0a68ad13e69142fb89779b2dba58e9dd145823-kennyslist > protected

kennyslist0a68ad13e69142fb89779b2dba58e9dd145823-kennyslist

Overview

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions

US East (N. Virginia)

Viewing 1 to 9

Name	Last modified	Size	Storage class
us-east-1:1ba861b3-c077-45bf-870b-63e90d039e8c	--	--	--
us-east-1:24f390be-e6f8-4280-b858-54a412177070	--	--	--
us-east-1:39bbda72-347f-4831-bdcf-735fc7739434	--	--	--
us-east-1:88c6f4ab-f95a-4119-ac90-3bde5a121bfc	--	--	--
us-east-1:8f9b47f7-cb98-493e-bd3e-19eb3e734401	--	--	--
us-east-1:b50e563f-bfe8-4d47-a48e-da014df6a343	--	--	--
us-east-1:f8e6263b-0d44-41f5-9208-1c9a427eedd2	--	--	--
us-east-1:f98acb29-4058-4d0d-8a2a-96f653501049	--	--	--

AWS S3 buckets for images. Each user has a protected folder only they can write to, but is public read.

The screenshot displays the AWS IAM console interface for a User Pool. The top navigation bar shows the AWS logo, 'Services', 'Resource Groups', and a user profile 'anthonydtnghuyen' in 'N. Virginia'. The main header indicates 'User Pools | Federated Identities' and the specific pool name 'kennyslistc8b0ac57_userpool_c8b0ac57-kennyslist'. A 'Delete pool' button is visible in the top right.

The left sidebar lists various settings categories: General settings (selected), Users and groups, Attributes, Policies, MFA and verifications, Advanced security, Message customizations, Tags, Devices, App clients, Triggers, Analytics, App integration, App client settings, Domain name, UI customization, Resource servers, Federation, Identity providers, and Attribute mapping.

The main content area shows the 'General settings' for the selected User Pool. The settings are organized into sections:

- Pool ID:** us-east-1_buFSrhilB
- Pool ARN:** arn:aws:cognito-idp:us-east-1:428663290127:userpool/us-east-1_buFSrhilB
- Estimated number of users:** 20
- Required attributes:** name, email
- Alias attributes:** none
- Username attributes:** none
- Enable case insensitivity?:** No
- Custom attributes:** Choose custom attributes...
- Minimum password length:** 15
- Password policy:** uppercase letters, special characters, numbers
- User sign ups allowed?:** Users can sign themselves up
- FROM email address:** Default
- Email Delivery through Amazon SES:** No

A note at the bottom states: 'Note: You have chosen to have Cognito send emails on your behalf. Best practices suggest that customers send emails through Amazon SES for production User Pools due to a daily email limit. Learn more about email best practices.'

Our log in general settings and password requirements.

Again all of this information is supposed to be trusted by our application, so please don't go exposing all our information.

We've also included in the repo two backend lambda functions that are in use. They are in the backendLambdaFunctions folder. Again they aren't really compiled from here, but from our AWS Console. We have them here so you can see it.