

# Recipe App

By: Jackson, Rene, Isabella, and Sydney



# Introduction

- This is an AWS web app that is implemented using a web server and a database in a properly configured network environment.
- It is implemented using a simple web page to simulate the apps interactions with the database.
- The app collects data about the user and allows them to search the database.

# Background

The reason we chose to do a recipe based app was because:

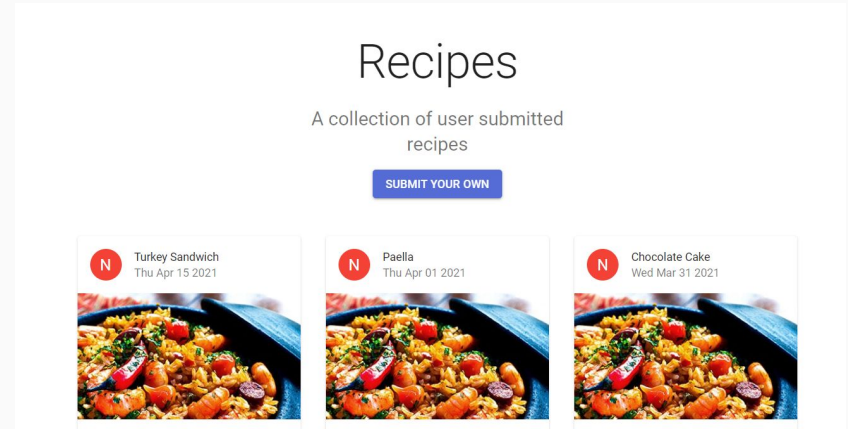
- All group members are from different cultures
- Food blogs are cool
- Simple enough to implement

# Objective

The goal of this app is to allow users to interact with the recipe database.

Users can:

- Sign up/Login
- Search for recipes
- Create/Delete recipes
- Add tags
- Filter by tags
- Add/Delete images
- Favorite recipes



# Constraints, Assumptions, Risks and Dependencies

**Constraints** - The tools that will be used are on the Free Tier on AWS.

**Assumptions** - Tools that will be available to deploy the web app, including Database, Storage, and Public Access.

**Risks and Dependencies** - Selecting a large project scope could result in an incomplete project and communication issues.

# User Table

Using the examples from class, we created a DynamoDB table for users.

- Username
- Name
- Passwords

Scan: [Table] cts-users: username ^ Viewing 1 to 4 items

Scan [Table] cts-users: username ^

+ Add filter

Start search

<input type="checkbox"/>	username	name ⓘ	password
<input type="checkbox"/>	Noskcaj19	Jackson Nunley	\$2b\$10\$LHvLu5Guvlbwllee1EU.9O31SxsvGN5h/.VFNO.KhKnFgwfp4Kzfu

<input type="checkbox"/>	username	name ⓘ	password
<input type="checkbox"/>	Slieske	Sydney	\$2a\$10\$MQS43ooL5LYGX0D/VBUrWOgXCqN.6/cCPguD6my.pxa1eRX
<input type="checkbox"/>	Noskcaj19	Jackson Nunley	\$2b\$10\$LHvLu5Guvlbwllee1EU.9O31SxsvGN5h/.VFNO.KhKnFgwfp4Kz
<input type="checkbox"/>	Bella	Isabella Morales	\$2a\$10\$4B1kEc/MERdii55D9NxrZO91AvF2.wu1WmhmyialQwXLcnA6Y

# Recipe Table

DynamoDB table for recipes.

- Author
- Instructions
- Date created
- Recipe name
- Tags
- Images

Scan: [Table] cts-recipes: id ^

Viewing 1 to 8 items

Scan [Table] cts-recipes: id ^

+ Add filter

Start search

	id ⓘ	_dummy	author_username	body	created	description	name	tags	image
<input type="checkbox"/>	3c145b87	dummy	Slieske	Preheat the o...	2021-04-15T19:46:14.000Z	This vegetar...	Vegetable Lasagna	[{"S": "vegetarian...]	
<input type="checkbox"/>	80ebd513	dummy	Noskcaj19	<Instructions ...	2021-03-31T21:04:12.000Z	A chocolate ...	Chocolate Cake	[{"S": "desert"}, {...	
<input type="checkbox"/>	87e508d8	dummy	Noskcaj19	Heat 1/2 cup ...	2021-04-01T17:23:13.000Z	This impres...	Paella	[{"S": "meal"}]	paella.jpg
<input type="checkbox"/>	9b75ba81	dummy	Slieske	Preheat oven ...	2021-04-15T19:57:41.139Z	Vegan Tacos	Chipotle Portob...	[{"S": "vegan"}, {...	
<input type="checkbox"/>	b4a213fe	dummy	Slieske	Ingredients: 3...	2021-04-15T19:41:33.841Z	Hashbrown ...	Hashbrown Cas...	[{"S": "meal"}, {...	
<input type="checkbox"/>	ce6ff008	dummy	Noskcaj19	Add various fr...	2021-04-15T18:56:51.431Z	A mixed frui...	Fruit Salad	[{"S": "vegetarian...]	
<input type="checkbox"/>	ce8dc582	dummy	Bella	Season sliced...	2021-04-15T19:39:44.760Z	Breaded Chi...	Pollo Empanizado	[{"S": "meal"}, {*...	

# Cognito Unauthenticated IAM Role

Roles > Cognito\_Server\_DynamoUnauth\_Role Delete role

## Summary

Role ARN	arn:aws:iam::214084644761:role/Cognito_Server_DynamoUnauth_Role <a href="#">🔗</a>
Role description	<a href="#">Edit</a>
Instance Profile ARNs	<a href="#">🔗</a>
Path	/
Creation time	2021-03-15 20:13 EDT
Last activity	2021-04-18 12:26 EDT (Today)
Maximum session duration	1 hour <a href="#">Edit</a>

Permissions Trust relationships Tags Access Advisor Revoke sessions

▼ Permissions policies (3 policies applied)

[Attach policies](#) [+ Add inline policy](#)

Policy name ▼	Policy type ▼	
▶ <a href="#">AmazonS3FullAccess</a>	AWS managed policy	✕
▶ <a href="#">AmazonDynamoDBFullAccess</a>	AWS managed policy	✕



# S3 & Image Uploading

Using S3, we created an image uploading feature.

**Block public access (bucket settings)**  
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Edit

Block all public access

Block public access to buckets and objects granted through new access control lists (ACLs)

Block public access to buckets and objects granted through any access control lists (ACLs)

Block public access to buckets and objects granted through new public bucket or access point policies

Block public and cross-account access to buckets and objects through any public bucket or access point policies

**Bucket policy**  
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

EditDelete

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::3-cts-images*"
    }
  ]
}
```

Copy

```
public async addRecipe(recipe: DBRecipe) {
  let imageKey = undefined;
  if (recipe.image) {
    imageKey = uuidv4();
    try {
      await this.s3.send(
        new PutObjectCommand({ input: {
          Bucket: DB.IMAGES_BUCKET,
          Key: imageKey,
          Body: Buffer.from(recipe.image, encoding: "base64"),
        }})
      );
    } catch (err) {
      console.log("Error", err);
    }
  }
}

await this.docClient.put({ args: {
  Item: {
    id: uuidv4(),
    ...recipe,
    ...(imageKey && { image: imageKey }),
    created: recipe.created.toISOString(),
    _dummy: "dummy",
  },
  TableName: DB.RECIPES_TABLE,
});
}
```

# Conclusion

In our introduction, we had outlined the project of creating an AWS web app that collected data about the user and let them interact with the database. This app we felt accomplished that.

Some difficulties encountered:

- Communication
- Different levels of experience
- Some issues understanding Dynamo

What we would do differently:

- Create a clear plan
- Communicate effectively
- Maybe use an SQL database

# Live Demo