WHAT WILL IT DO

- Users sign in with email and pass
- Users can create recipes with ingredients and instructions
- Recipes can be marked as public or private
- Users can view other people's recipes
- Ingredients from recipes can be added to a user's grocery list
- Users can create their own occasions and assign recipes to occasions

What tables will I need

- User table –user_id, first_name, last_name, username, bio, profile_pic
- Auth table – user_id, user_email, pass_hash
- Recipe table – recipe_id, recipe_name, user_id, ingredient_id, instructions, is_public, pic_url
- Comment table – user_id, post_id, comment, timestamp, like/dislike
- Favorites table – user_id, recipe_id
- Ingredient table – ingredient_id, ingredient, ingredient_price
- Grocery table –ingredient_id, user_id
- Event table – recipe_id, event_id, event_name, event_location, event_time, event_date, user_id
- Group table – event_id, user_id

Relationships

- One to one: User & auth, event & group, user & grocery
- One to many: Ingredients & Recipe, recipe & event, recipe & comments, user & comments, user & event
- Many to many: group & users

Columns

Auth –

- Auth id in order to keep track of the user's information by id
- User email for authentication purposes
- User password for authentication purposes
- User id from the users table for username info

Users –

- User id to keep track of the user by id
- First name and last name for other users to view when looking at a user's recipes
- Username for login and commenting
- Bio for other users to get more info about the poster
- Profile picture for added personalization

Recipe –

- Recipe id to keep track of the recipe by id
- Recipe name for display and query

- User_id to know who posted the recipe
- Ingredient id to call on ingredients for the recipe
- Instructions for the user to write out a lengthy manual for their recipe
- is_public to toggle the posts between public and private (true/false)
- pic_url so that everyone can see your delicious treats

Ingredient –

- ingredient id to keep track of ingredients by id
- ingredient for the name of the ingredient
- ingredient price for the price of the ingredient, which will be used in the grocery cart.

Grocery List –

- grocery list id to keep track of individual grocery lists
- ingredient id to pull individual ingredients into the grocery list

Favorites –

- favorites id to keep track of a list of favorite recipes by id
- user id to know whose favorites list it is
- recipe id to add recipes to the favorite list

Comment –

- post id to keep track of each post by id
- user id to keep track of who made the post
- comment for the actual post
- timestamp to know when the post was made
- like with an initial value of NULL with a bool value for likes/dislikes

Event –

- event id to keep track of each event by id
- user id to keep track of who made the event
- recipes to add recipes to the event
- event name for the name of the event
- event location for the location
- event time for the time the event will be held
- event date for the date the event will be held

Group –

- group id to keep track of each group by id
- event id to keep track of the event the group is pertaining to
- user id to keep track of who is in the event group.

The tables in SQL:

Users --

```
create table Users (
        user_id serial primary key,
 first_name varchar(50) NOT NULL,
 last_name varchar(50) NOT NULL,
 username varchar(50) NOT NULL UNIQUE,
 bio varchar(3000),
 profile_pic text
);
```

Auth --

```
create table Auth (
        auth_id serial primary key,
 user_email varchar(255) NOT NULL UNIQUE,
 pass_hash varchar(1000) NOT NULL,
 user_id int references Users(user_id)
);
```

Ingredient --

```
create table Ingredient (
        ingredient_id serial primary key,
 ingredient varchar(50) NOT NULL UNIQUE,
 ingredient_price int NOT NULL
);
```

Recipe --

```
create table Recipe (
        recipe_id serial primary key,
 recipe_name varchar(50) NOT NULL,
 user_id int references Users(user_id),
 ingredient_id int references Ingredient(ingredient_id),
```

```
  is_public bool NOT NULL default 'true',
  pic_url text
);
```

Favorites --

```
create table Favorites (
  favorites_id serial primary key,
  user_id int references Users(user_id),
  recipe_id int references Recipe(recipe_id)
);
```

GroceryList --

```
create table GroceryList (
        grocery_list_id serial primary key,
  ingredient_id int references Ingredient(ingredient_id)
);
```

Comment --

```
create table Comment (
  post_id serial primary key,
  user_id int references Users(user_id),
  comment varchar(300) NOT NULL,
  typed_on timestamp with time zone NOT NULL,
  upvote bool DEFAULT 'false'
);
```

Event --

```
create table Event (
  event_id serial primary key,
  user_id int references Users(user_id),
  recipes int references Recipe(recipe_id),
  event_name varchar(50) NOT NULL,
  event_location varchar(50) NOT NULL,
```

```
  event_time varchar(50) NOT NULL,
  event_date varchar(50) NOT NULL
);
```

Groups --

```
create table Groups (
  group_id serial primary key,
  event_id int references Event(event_id),
  user_id int references Users(user_id)
);
```