BRAINSTORM

User sign in info with username, email, password
List of ingredients
List of instructions
List of recipes
Display grocery list
List of occasions and the associated recipes
Like/share other user recipes
Save other user recipes

TABLE IDEAS

Users – store user info
Ingredients – store all ingredients used across app
Instructions – store all instructions
Recipes – store all recipes created by users (Middle between ingredients and instructions)
Grocery List – collection of ingredients (Associated between users and ingredients)
Occasions – collection of recipes
Likes – which user likes a recipe (Associated between users and recipes)
Saved Recipes – store recipes users want to see again but don't want to add to an occasion

RELATIONSHIPS

One-to-one:
Users and Grocery List: users can have one grocery list, and a grocery list belongs to one user
Instructions and Recipes: Recipes will have one instruction, instructions can only belong to one recipe

One-to-many:
Users and Recipes: Users can create many recipes, those recipes are all tied to that user
Users and Likes: Users can like many recipes, those likes are all tied to that user

Many-to-many:
Users and Occasions: Users can have many occasions, occasions can be part of many users
Users and Saved Recipes: Users can save multiple recipes, those recipes can be saved to multiple users
Ingredients and Instructions: Ingredients can be part of many instructions, instructions include many ingredients
Ingredients and Recipes: Ingredients can be part of many recipes, recipes include many ingredients
Ingredients and Grocery List: Ingredients can be on multiple grocery lists, grocery lists can have many ingredients
Recipes and Occasions: Recipes can be part of many occasions, occasions can have multiple recipes
Recipes and Likes: Recipes can have many likes, likes can be part of many different recipes
Recipes and Saved Recipes: Many recipes can be saved, those saved recipes will also be in recipes

COLUMNS

Users:

- User_id: keep track of users with whole number for ID
- Username: give each user a unique username, limit to 20 characters to limit length
- Email: associate email address with user, limit to 40 characters to limit length
- Password: establish password to login, limit to 20 characters to limit length

Ingredients

- Ingredient_id: keep track of each ingredient with whole number for ID
- Ingredient_name: give each ingredient a name, limit to 100 characters
- Recipe_id: show which recipes each ingredient is a part of, integer to match recipe table id

Instructions

- Instruction_id: keep track of each instruction with whole number for ID
- Instruction: the verbiage of the instruction, limit to 255 characters
- Recipe_id: show which recipe the instruction goes with, integer matches recipe table id
- Ingredient_id: show which ingredients are part of the instruction, matches ingredients table id

Recipes

- Recipe_id: keep track of each recipe with whole number for ID
- Recipe_name: what the recipe is called, limit to 50 characters so name isn't too long
- Instruction_id: link the correct instruction and ingredients to recipe, matches instruction table id
- User_id: link to the user who created the recipe, matches user table id
- Occasion_id: show which occasions each recipe is part of, matches occasion table id
- Like_id: show how many and what users have liked the recipe, matches like table id
- Saved_id: show which recipes have been saved, how many times, and by what user, matches saved recipes table id

Grocery List

- Grocery_list_id: keep track of each list with whole number for ID
- User_id: show which user the list is associated with, matches user table id
- Ingredient_id: show which ingredients in the list, matches ingredient table id

Occasions

- Occasion_id: keep track of each occasion with whole number for id
- Occasion_name: assign a name to the occasion, limit character number to 100
- User_id: show which occasions are associated with which users, matches user table id
- Recipe_id: show which recipes are included in the occasion, matches recipe table id

Likes

- Like_id: keep track of each like with whole number for id
- User_id: show which user each like is associated with, matches user table id
- Recipe_id: show which recipe has been liked, matches recipe table id

Saved Recipes

- Saved_id: keep track of each saved recipe, whole number for id
- User_id: show which users have that saved recipe, matches user table id
- Recipe_id: show which recipe has been saved, matches recipe table id

CREATING TABLES

```
CREATE TABLE users(
 user_id SERIAL PRIMARY KEY,
 username VARCHAR(20),
 email VARCHAR(40),
 password VARCHAR(20)
);

CREATE TABLE ingredients(
  ingredient_id SERIAL PRIMARY KEY,
  ingredient_name VARCHAR(100),
  recipe_id INTEGER REFERENCES recipes(recipe_id)
 );

CREATE TABLE instructions (
 instruction_id SERIAL PRIMARY KEY,
 instruction VARCHAR(255),
 ingredient_id INTEGER REFERENCES ingredients(ingredient_id),
 recipe_id INTEGER REFERENCES recipes(recipe_id)
);

CREATE TABLE grocery_list(
 grocery_list_id SERIAL PRIMARY KEY,
 user_id INTEGER REFERENCES users(user_id),
 ingredient_id INTEGER REFERENCES ingredients(ingredient_id)
);

CREATE TABLE occasions (
  occasion_id SERIAL PRIMARY KEY,
  occasion_name VARCHAR(100),
  user_id INTEGER REFERENCES users(user_id),
  recipe_id INTEGER REFERENCES recipes(recipe_id)
);

CREATE TABLE likes(
 like_id SERIAL PRIMARY KEY,
 user_id INTEGER REFERENCES users(user_id),
 recipe_id INTEGER REFERENCES recipes(recipe_id)
);
```

```
CREATE TABLE recipes(
 recipe_id SERIAL PRIMARY KEY,
 recipe_name VARCHAR(50),
 instruction_id INTEGER REFERENCES instructions(instruction_id),
 user_id INTEGER REFERENCES users(user_id),
 occasion_id INTEGER REFERENCES occasions(occasion_id),
 like_id INTEGER REFERENCES likes(like_id),
 saved_id INTEGER REFERENCES saved_recipes(saved_id)
);


CREATE TABLE saved_recipes (
 saved_id SERIAL PRIMARY KEY,
 user_id INTEGER REFERENCES users(user_id),
 recipe_id INTEGER REFERENCES recipes(recipe_id)
);
```

ADDING DATA

```
INSERT INTO users (username, email, password)
 VALUES('Michael', 'mscott@df.com', '123456789'),
        ('Jim', 'jhalperet@df.com', 'abcdefg'),
        ('Dwight', 'dschrute@df.com', 'beats'),
        ('Pam', 'pbeasley@df.com', 'ilovejim'),
        ('Kevin', 'kmalone@df.com', 'chili');

INSERT INTO instructions(instruction)
VALUES('Everyone gets to know each other in the pot. Its probably the thing I do best');

INSERT INTO recipes(user_id, instruction_id, recipe_name)
VALUES(5, 2, 'Kevins Famous Chili')

INSERT INTO occasions(user_id, occasion_name)
VALUES (5, 'Office Party');
```

```
UPDATE recipes
SET occasion_id = 2
WHERE recipe_id = 2;

UPDATE instructions
SET recipe_id = 2
WHERE instruction_id = 2;

INSERT INTO likes(user_id, recipe_id)
VALUES(1, 2);

UPDATE recipes
SET like_id = 1
WHERE recipe_id = 2;

INSERT INTO saved_recipes(user_id, recipe_id)
VALUES(2, 2);

UPDATE recipes
SET saved_id = 1
WHERE recipe_id = 2;
```