Brainstorming:

Users:
- -id
- -password
- -email
- -first name
- -last name
- -bio
- -image

Ingredients:
- -id
- -name

Recipes:
- -id
- -name
- -ingredients
- -instructions
- -image
- -privacy
- -servings
- -source

Shopping:
- -id
- -recipes
- -entries

Occasions:
- -id
- -recipe

Tables:

Users:
- -User_id
- -User_password
- -User_email
- -first_name
- -last_name
- -bio
- -user_image

Ingredients Table:
- -ingredient_id
- -ingredient_name

Recipes Table:
- -recipe_id
- -recipe_name
- -recipe_ingredients
- -recipe_instructions
- -recipe_image
- -recipe_privacy
- -user_id

Shopping Table:
- -shopping_id
- -user_id
- -list_name
- -shopping_recipes
- -shopping_ingredients

Occasions Table:
- -occasion_id
- -recipe_id
- -user_id

Relationships:

One to one:

One to many:
user to recipes… because a user can have many recipes but a recipe can only have one user

user to shopping… because a user can have many shopping lists but a shopping list can only have one user

user to occasion… because a user can have many occasions but an occasion can only have one user

Many to many:
ingredients to recipes… because an ingredient can have many recipes and a recipe can have many users

recipes to shopping… because a recipe can have many shopping lists and a shopping list can have many recipes
occasions to recipes… because an occasion can have many recipes and a recipe can have many occasions

Columns:

Users:
-User_id - SERIAL PRIMARY KEY allows unique identifier for a user and will automatically increment every time a new user is added
-User_password - VARCHAR(50) allows for a password of up to 50 characters
-User_email - VARCHAR(50) allows for an email of up to 50 characters
-first_name - VARCHAR(50) allows for a first name of up to 50 characters
-last_name - VARCHAR(50) allows for a last name of up to 50 characters
-bio - VARCHAR(2000) allows for a biography of up to 2000 characters
-user_image - VARCHAR(1000) allows for an image URL of up to 1000 characters

Ingredients Table:
-ingredient_id - SERIAL PRIMARY KEY allows unique identifier for an ingredient and will automatically increment every time a new ingredient is added
-ingredient_name VARCHAR(50) allows for an ingredient name of up to 50 characters

Recipes Table:
-recipe_id - SERIAL PRIMARY KEY allows unique identifier for a recipe and will automatically increment every time a new recipe is added
-recipe_name - VARCHAR(50) allows for a last name of up to 50 characters
-recipe_ingredients - VARCHAR(50) allows for a last name of up to 50 characters
-recipe_instructions - VARCHAR(2000) allows for an instructions section of up to 2000 characters
-recipe_image - VARCHAR(1000) allows for an image URL of up to 1000 characters
-recipe_privacy - BOOLEAN allows for true or false value for privacy

Shopping Table:
-shopping_id - SERIAL PRIMARY KEY allows unique identifier for a shopping list and will automatically increment every time a new shopping list is added
-user_id - INT NOT NULL REFERENCES users(user_id) links to the users table to get the value of user_id
-list_name VARCHAR(50) allows for a list name of up to 50 characters
-shopping_recipes VARCHAR(50) allows for recipes to be added to list
-shopping_entries VARCHAR(50) allows for individual items to be added to the list

Occasions Table:
-occasion_id - SERIAL PRIMARY KEY allows unique identifier for an occasion and will automatically increment every time a new occasion is added

-recipe_id - INT NOT NULL REFERENCES links to recipe id in the recipes table
-user_id - INT NOT NULL REFERENCES links to user id in the users table

recipe_ingredients Table:
- recipe_ingredients_id - SERIAL PRIMARY KEY
-recipe_id - INT NOT NULL REFERENCES links to recipe id in the recipes table
-ingredient_id - INT NOT NULL REFERENCES links to ingredient id in the ingredients table

shopping_ingredients_id Table:
-shopping_ingredient_id - SERIAL PRIMARY KEY allows unique identifier for shopping list ingredients and will automatically increment every time a new shopping list ingredient is added
-shopping_id - INT NOT NULL REFERENCES links to shopping id in the shopping table
-ingredient_id - INT NOT NULL REFERENCES links to ingredient id in the ingredients Table

shopping_recipes Table:
-shopping_recipes_id - SERIAL PRIMARY KEY allows unique identifier for shopping list recipes and will automatically increment every time a new recipe is added to the shopping list.
-shopping_id - INT NOT NULL REFERENCES links to shopping id in the shopping table
-occasion_id - INT NOT NULL REFERENCES links to occasion id in the occasions table

Postgres:

```
CREATE TABLE users (
  user_id SERIAL PRIMARY KEY,
  user_password VARCHAR(50),
  user_email VARCHAR(50),
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  bio VARCHAR(2000),
  user_image VARCHAR(1000)
);

CREATE TABLE ingredients (
  ingredient_id SERIAL PRIMARY KEY,
  ingredient_name VARCHAR(50)
);

CREATE TABLE recipes (
  recipe_id SERIAL PRIMARY KEY,
```

```
  recipe_name VARCHAR(50),
  recipe_instructions VARCHAR(50),
  recipe_image VARCHAR(1000),
  recipe_source VARCHAR(50),
  recipe_privacy BOOLEAN,
  user_id INT NOT NULL REFERENCES users(user_id)
);

CREATE TABLE shopping (
  shopping_id SERIAL PRIMARY KEY,
  user_id INT NOT NULL REFERENCES users(user_id),
  shopping_name VARCHAR(50),
  shopping_recipes VARCHAR(50),
  shopping_ingredients VARCHAR(50)
);

CREATE TABLE occasions (
  occasion_id SERIAL PRIMARY KEY,
  occasion_name VARCHAR(100),
  user_id INT NOT NULL REFERENCES users(user_id)
);

CREATE TABLE recipe_ingredients (
  recipe_ingredients_id SERIAL PRIMARY KEY,
  recipe_id INT NOT NULL REFERENCES recipes(recipe_id),
  ingredient INT NOT NULL REFERENCES ingredients(ingredient_id)
);

CREATE TABLE shopping_ingredients_id (
  shopping_ingredient_id SERIAL PRIMARY KEY,
  shopping_id INT NOT NULL REFERENCES shopping(shopping_id),
  ingredient_id INT NOT NULL REFERENCES ingredients(ingredient_id)
);

CREATE TABLE shopping_recipes (
  shopping_recipes_id SERIAL PRIMARY KEY,
  shopping_id INT NOT NULL REFERENCES shopping(shopping_id),
  occasion_id INT NOT NULL REFERENCES occasions(occasion_id)
);
```