- users can sign into the app with their email and password
- 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 user's grocery lists
- users can create their own occasions and assign recipes to occasions

Brainstorming:

-keep track of user accounts
-keep track of what recipes belong to which user
-keep track of what ingredients a user put in their grocery list
-keep track of how many views a recipe has
- " of what occasions a recipe has been assigned to
- " of what occasions a user has created
- " of which recipes are public vs private
- " of which instructions belong to which recipe

Table Ideas:

Profile: this table will hold a certain user's info. Each row will be a different user. It'll contain user_id, first_name, last_name, email, for every person registered in the app

Passwords: this table will hold passwords for each user. It'll contain user_id and password

Recipes: this table will hold info on recipe names and the people that created them. It'll contain recipe_name, recipe_id, occasion_id, and the user_id, isPublic (boolean) that they are connected to

Occasions: this table will contain occasion_id, occasion_name

Grocery List: this table will hold different grocery lists for each user. It will contain grocerylist_id, user_id, recipe_id, ingredient_id

Ingredients: this table will hold a list of ingredients for each recipe. It'll contain recipe_id, ingredient_id, ingredients

Relationships:

One-to-one:
-one password is assigned to one user
-one email is assigned to one user
-each recipe is assigned to one user
-each recipe is either public or private

One-to-many:
-one recipe can have many views
-one user can have many recipes
-

Many-to-many:
-recipes can be associated with different occasions

Columns:

Profile:

user_id	I will be storing this data as an integer with a primary key to ensure the uniqueness of every user
first_name	I will be storing this data as a varchar(30) so that we can know who the user_id belongs to
last_name	I will be storing this data as a varchar(30) so that we can know who the user_id belongs to
email	I will be storing this data as a varchar(30) so that we have the email address of each user in getting registered/logged in

```
CREATE TABLE profile (
        user_id SERIAL PRIMARY KEY,
        first_name VARCHAR(30),
        last_name VARCHAR(30),
        email VARCHAR(30)
);
```

Passwords:

user_id	I will be referencing this data from the profile table so that we know which passwords belong to whom
password	I will be storing this data as a varchar(30)

```
CREATE TABLE passwords (
        user_id INTEGER NOT NULL REFERENCES profile(user_id),
        password VARCHAR(30)
);
```

Recipes:

recipe_id	I will be storing this data as an integer with a primary key to ensure the uniqueness for every user
recipe_name	I will be storing this data as a varchar(30) so that we can know what the recipe is named
user_id	I will be referencing this data from the profile table so that we know which passwords belong to whom
isPublic	I will be storing the data as a boolean value for whether a recipe is marked public or private
directions	I will be storing this data as text so that we can have explicit directions for each recipe

```
CREATE TABLE recipes (
        recipe_id SERIAL PRIMARY KEY,
         recipe_name VARCHAR(30),
        user_id INTEGER NOT NULL REFERENCES profile(user_id),
        is_public BOOLEAN,
        directions TEXT
);
```

Occasions:

occasion_id	I will be storing this data as an integer with a primary key to ensure the uniqueness of every occasion made by a user
occasion_name	I will be storing this data as a varchar(30) so that we can know what the occasion is named

```
CREATE TABLE occasions (
        occasion_id SERIAL PRIMARY KEY,
         occasion_name VARCHAR(30)
);
```

Grocery List:

grocerylist_id	I will be storing this data as an integer with a primary key to ensure the uniqueness of every grocery list made by a user
recipe_id	I will be referencing this data from the profile table so that we know which list belongs to whom

```
CREATE TABLE grocery list (
        grocerylist_id SERIAL PRIMARY KEY,
        user_id INTEGER NOT NULL REFERENCES profile(user_id),
);
```

Ingredients:

ingredient _id	I will be storing this data as an integer with a primary key to ensure the uniqueness of every recipe ingredient list made by a user
recipe_id	I will be referencing this data from the profile table so that we know which recipe

	the ingredients belong to
ingredients	I will be storing this data as text so that we can have explicit ingredients for each recipe

```
CREATE TABLE ingredients (
        ingredient_id SERIAL PRIMARY KEY,
        user_id INTEGER NOT NULL REFERENCES profile(user_id),
         ingredients TEXT
);
```