

## Features¶

- 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

- User ID
- User email
- User Password
- User First name
- User Last name
- Recipe ID
- Recipe Name
- Recipe Ingredients
- Recipe Steps
- Privacy settings of recipe
- Grocery List ID
- Occasions

## Tables

### User

- User ID
- User email
- User Password
- User First name
- User Last name

### Recipe

- Recipe ID
- Recipe Name
- Privacy settings of recipe
- User ID of creator

### Grocery List

- Grocery List ID
- Grocery List Creator ID
- Ingredient ID

### Occasions

- Occasion ID
- Occasion Name

- User ID of Occasion Creator

#### Instruction

- Instruction ID
- Instruction Name

#### Ingredients

- Ingredient ID
- Ingredient Name

#### Recipe Ingredients

- Recipe Ingredient ID
- Recipe ID
- Ingredient ID

#### Grocery Ingredients

- Grocery Ingredient ID
- Grocery List ID
- Ingredient ID

#### Recipe Occasions

- Recipe Occasion ID
- Occasion ID
- Recipe ID

#### Recipe Instructions

- Recipe Instruction ID
- Recipe ID
- Instruction ID

#### Relationships

##### One To One

##### One To Many

- User=>Recipe: Each recipe can only have one user creator, however each user can create many different recipes.
- User=>Grocery List: Each user can create many different grocery lists, however each grocery list can only have one  
User=>Occasion. In the context of this app, each occasion can only have one creator. However each user can have many occasions saved.

##### Many To Many

- Grocery List =>Ingredients: Each ingredient can appear on many grocery lists. Each grocery list will contain many ingredients.
- Occasion =>Recipe: Each occasion can contain many potential recipes. Recipes can fit into and be listed on many occasions.
- Recipe => Ingredients: Each recipe can contain many ingredients. Each ingredient can be contained in many recipes.

- Recipe=> Instructions: Each recipe can contain many instructions. Each instruction can be contained in many recipes

## **Columns**

### **User**

- User ID: We will store this because it is the necessary item to keep track of the data for this specific user. This is why we will store it as the primary key.
- User email: We will store this because the user needs to use their email to login to the app. We will store it using Varchar, as this allows it to be stored as a string.
- User Password: We will store this because the user needs to use their password to login to the app. We will store it using Varchar, as this allows it to be stored as a string.
- User First name: We will store this because it's important to keep track of the user's profile information. We will store it using Varchar, as this allows it to be stored as a string.
- User Last name: We will store this because it's important to keep track of the user's profile information. We will store it using Varchar, as this allows it to be stored as a string.

### **Recipe**

- Recipe ID: We will store this because it is the necessary item to keep track of the data for this specific recipe. This is why we will store it as the primary key.
- Recipe Name: We need to store this as the user will want to know what the name of the recipe is. We will store it as a varchar because it is a string.
- Privacy settings of recipe: We will store this because it's critical to know if the user wants their recipe to be displayed publicly or kept private. We will store this as a boolean, because it is either true that the user wants to have their recipes be private, or it is not.
- User ID of creator: We need to store this, because we need to link the Recipe with the user that created it. Because of this we need to store it as a foreign key.

### **Grocery List**

- Grocery List ID: We will store this because it is the necessary item to keep track of the data for this specific grocery list. This is why we will store it as the primary key.
- Grocery List Creator ID: We need to store this, because we need to link the grocery list with the user that created it. Because of this we need to store it as a foreign key.

### **Occasions**

- Occasion ID: We will store this because it is the necessary item to keep track of the data for this specific occasion. This is why we will store it as the primary key.
- Occasion Name: We need to store this because this is the occasion that the user is looking for. We will store this as a Varchar because it is a string.

- User ID of Occasion Creator: We need to store this, because we need to link the occasion with the user that created it. Because of this we need to store it as a foreign key.

#### Instruction

- Instruction ID: We will store this because it is the necessary item to later call upon this specific instruction. This is why we will store it as the primary key.
- Instruction Name: We need to store this because this is the specific instruction that needs to be completed in order to make the recipe app functional. We're going to store this as text because that is the format most convenient to enter the instruction in.

#### Ingredients

- Ingredient ID: We will store this because it is the necessary item to later call upon this specific ingredient. This is why we will store it as the primary key.
- Ingredient Name: We will store this because we need to know what ingredient we are looking for. We will store this as a varchar because it is a string.

#### Recipe Ingredients

- Recipe Ingredient ID: We will store this because it is the necessary item to keep track of the data for this specific middle table. This is why we will store it as the primary key.
- Recipe ID: To have this middle table show the necessary data we need to call upon the Recipe ID from another table. Because of this we will store this as the foreign key.
- Ingredient ID: To have this middle table show the necessary data we need to call upon the Ingredient ID from another table. Because of this we will store this as the foreign key.

#### Grocery Ingredients

- Grocery Ingredient ID: We will store this because it is the necessary item to keep track of the data for this specific middle table. This is why we will store it as the primary key.
- Grocery List ID: To have this middle table show the necessary data we need to call upon the Grocery List ID from another table. Because of this we will store this as the foreign key.
- Ingredient ID: To have this middle table show the necessary data we need to call upon the Ingredient ID from another table. Because of this we will store this as the foreign key.

#### Recipe Occasions

- Recipe Occasion ID: We will store this because it is the necessary item to keep track of the data for this specific middle table. This is why we will store it as the primary key.

- Occasion ID: To have this middle table show the necessary data we need to call upon the Occasion ID from another table. Because of this we will store this as the foreign key.
- Recipe ID: To have this middle table show the necessary data we need to call upon the Recipe ID from another table. Because of this we will store this as the foreign key.

#### Recipe Instructions

- Recipe Instruction ID: We will store this because it is the necessary item to keep track of the data for this specific middle table. This is why we will store it as the primary key.
- Recipe ID: To have this middle table show the necessary data we need to call upon the Recipe ID from another table. Because of this we will store this as the foreign key.
- Instruction ID: To have this middle table show the necessary data we need to call upon the Instruction ID from another table. Because of this we will store this as the foreign key.

#### **POSTGRESQL Code**

```
CREATE TABLE users (
user_id SERIAL PRIMARY KEY,
user_email VARCHAR(30),
user_password VARCHAR(50),
user_first_name VARCHAR(30),
user_last_name VARCHAR(30)
);
```

```
CREATE TABLE recipes (
recipe_id SERIAL PRIMARY KEY,
recipe_name VARCHAR(50),
recipe_privacy BOOLEAN,
author_id INT NOT NULL REFERENCES users(user_id)
);
```

```
CREATE TABLE grocery_list (
grocery_list_id SERIAL PRIMARY KEY,
creator_id INT NOT NULL REFERENCES users(user_id)
);
```

```
CREATE TABLE occasions (
occasion_id SERIAL PRIMARY KEY,
```

```
occasion_name VARCHAR(30),  
occasion_creator_id INT NOT NULL REFERENCES users(user_id)  
);
```

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

```
CREATE TABLE instructions (  
instruction_id SERIAL PRIMARY KEY,  
instruction_text TEXT  
);
```

```
CREATE TABLE recipe_ingredients (  
recipe_ingredient_id SERIAL PRIMARY KEY,  
recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id),  
ingredient_id INTEGER NOT NULL REFERENCES ingredients(ingredient_id)  
);
```

```
CREATE TABLE grocery_ingredients (  
grocery_ingredient_id SERIAL PRIMARY KEY,  
grocery_id INTEGER NOT NULL REFERENCES grocery_list(grocery_list_id),  
ingredient_id INTEGER NOT NULL REFERENCES ingredients(ingredient_id)  
);
```

```
CREATE TABLE recipe_occasions (  
recipe_occasion_id SERIAL PRIMARY KEY,  
recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id),  
occasion_id INTEGER NOT NULL REFERENCES occasions(occasion_id)  
);
```

```
CREATE TABLE recipe_instructions (  
recipe_instruction_id SERIAL PRIMARY KEY,  
recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id),  
instruction_id INTEGER NOT NULL REFERENCES instructions(instruction_id)  
);
```

**Inserting Data into Tables:**

```
INSERT INTO users (user_last_name, user_first_name, user_password, user_email)
VALUES ('Smith', 'Harry', 'password10', 'harrysmith@yahoo.com');
```

```
INSERT INTO recipes (recipe_name, recipe_privacy, author_id)
VALUES ('Toast', FALSE, 1);
```