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 the user's grocery lists
- Users can create their own occasions and assign recipes to occasions

Brainstorming/Data:

- Users
    - Password
    - Email/username
    - Date of birth
    - Favorite holidays/occasions
    - Phone number
    - Website
    - Number of recipes
    - Number of followers
- Recipes
    - Diet type/meal plan (paleo, vegetarian, vegan, keto, Atkins, etc.)
    - Other categories (breakfast, lunch, dinner, snack)
    - Recipe step number
    - Recipe name
    - Recipe description
    - Ingredients
    - Public? boolean
    - Amounts for ingredients (integer)
    - Unit of measurement (# teaspoon, # lbs, # apples, # potatoes, etc.)
    - Notes
    - Needs_shopped Boolean
    - author
- Connections
    - Who's following who
    - Follow a person or follow a meal plan type
- Grocery
    - References selected ingredients
    - Amounts for ingredients (integer)
    - Unit of measurement (# teaspoon, # lbs, # apples, # potatoes, etc.)
    - Is bought
- Occasions
    - Holidays
    - Other categories (breakfast, lunch, dinner, snack) REFERENCE RECIPES TABLE

Table Ideas:

- User – holds all user data; each row is a different user
    - User_id
    -
- Recipe – holds all recipe data; each row is a different recipe (how to solve for many steps?)
    - Recipe_id
- Following – bridge between users; each row is a different connection
    - Primary key
    - The follower
    - The person being followed
- Grocery – provides a subset of recipe items that need to be purchased; each row is a different item with amounts
- Occasion – provides holidays and their dates which can be tied back to recipes; each row is a different holiday/occasion and its date

Relationships (explain the relationships later):

- One to One
- One to Many
    - User ➔ recipes
- Many to Many
    - Who's following who

```
CREATE TABLE user (
  user_id SERIAL PRIMARY KEY,
  user_fname VARCHAR(30) NOT NULL,
  user_lname VARCHAR(30) NOT NULL,
  user_email VARCHAR(30) NOT NULL,
  user_password VARCHAR(500) NOT NULL,
  user_bio VARCHAR(1000),
  user_phone INT(10),
  user_company VARCHAR(30),
  user_website VARCHAR(100),
  account_created TIMESTAMP NOT NULL
);
```

```
CREATE TABLE recipe (
  recipe_id SERIAL PRIMARY KEY,
  recipe_name VARCHAR(50) NOT NULL,
  recipe_ingredient_001 VARCHAR(30) NOT NULL, -- apple
  recipe_ingredient_002 VARCHAR(30),
  recipe_ingredient_003 VARCHAR(30), -- sugar
  recipe_ingredient_004 VARCHAR(30),
  recipe_ingredient_005 VARCHAR(30),
  recipe_ingredient_amount_001 INT NOT NULL, -- 4
  recipe_ingredient_amount_002 INT,
  recipe_ingredient_amount_003 INT, -- 3
  recipe_ingredient_amount_004 INT,
  recipe_ingredient_amount_005 INT,
  recipe_ingredient_unit_001 VARCHAR(30) NOT NULL, -- whole
  recipe_ingredient_unit_002 VARCHAR(30),
  recipe_ingredient_unit_003 VARCHAR(30), -- tablespoon(s)
  recipe_ingredient_unit_004 VARCHAR(30),
  recipe_ingredient_unit_005 VARCHAR(30),
  recipe_special_instructions VARCHAR(1000), -- time in oven, stir here, etc.
  recipe_notes VARCHAR(1000), -- anything else
  meal_plan VARCHAR(30), -- keto, vegan, etc.
  meal_type VARCHAR(30), -- breakfast, lunch, etc.
  occasion_id INT REFERENCES occasion(occasion_id),
  is_public BOOLEAN DEFAULT FALSE,
  needs_purchased_001 BOOLEAN DEFAULT FALSE,
  needs_purchased_002 BOOLEAN DEFAULT FALSE,
  needs_purchased_003 BOOLEAN DEFAULT FALSE,
  needs_purchased_004 BOOLEAN DEFAULT FALSE,
```

```
  needs_purchased_005 BOOLEAN DEFAULT FALSE,
  author_id INT NOT NULL REFERENCES user(user_id)
);

CREATE TABLE following (
  follow_id SERIAL PRIMARY KEY,
  follower_id INT NOT NULL REFERENCES user(user_id),
  following_id INT NOT NULL REFERENCES user(user_id)
);

CREATE TABLE grocery (
  grocery_list_id SERIAL PRIMARY KEY,
  grocery_recipie_id INT NOT NULL REFERENCES recipe(recipe_id),
  grocery_recipie VARCHAR(50) NOT NULL REFERENCES recipe(recipe_name),
  ingredient_name_001 VARCHAR(30) NOT NULL REFERENCES recipe(recipe_ingredient_001),
  ingredient_name_002 VARCHAR(30) NOT NULL REFERENCES recipe(recipe_ingredient_002),
  ingredient_name_003 VARCHAR(30) NOT NULL REFERENCES recipe(recipe_ingredient_003),
  ingredient_name_004 VARCHAR(30) NOT NULL REFERENCES recipe(recipe_ingredient_004),
  ingredient_name_005 VARCHAR(30) NOT NULL REFERENCES recipe(recipe_ingredient_005),
  ingredient_amount_001 INT NOT NULL REFERENCES recipe(recipe_ingredient_amount_001),
  ingredient_amount_002 INT NOT NULL REFERENCES recipe(recipe_ingredient_amount_002),
  ingredient_amount_003 INT NOT NULL REFERENCES recipe(recipe_ingredient_amount_003),
  ingredient_amount_004 INT NOT NULL REFERENCES recipe(recipe_ingredient_amount_004),
  ingredient_amount_005 INT NOT NULL REFERENCES recipe(recipe_ingredient_amount_005),
  ingredient_measurement_001 VARCHAR(30) NOT NULL REFERENCES
recipe(recipe_ingredient_unit_001),
  ingredient_measurement_002 VARCHAR(30) NOT NULL REFERENCES
recipe(recipe_ingredient_unit_002),
  ingredient_measurement_003 VARCHAR(30) NOT NULL REFERENCES
recipe(recipe_ingredient_unit_003),
```

```
  ingredient_measurement_004 VARCHAR(30) NOT NULL REFERENCES
recipe(recipe_ingredient_unit_004),
  ingredient_measurement_005 VARCHAR(30) NOT NULL REFERENCES
recipe(recipe_ingredient_unit_005),
  needs_purchased_001 BOOLEAN NOT NULL REFERENCES recipe(needs_purchased_001),
  needs_purchased_002 BOOLEAN NOT NULL REFERENCES recipe(needs_purchased_002),
  needs_purchased_003 BOOLEAN NOT NULL REFERENCES recipe(needs_purchased_003),
  needs_purchased_004 BOOLEAN NOT NULL REFERENCES recipe(needs_purchased_004),
  needs_purchased_005 BOOLEAN NOT NULL REFERENCES recipe(needs_purchased_005),
  buyer_id INT NOT NULL REFERENCES user(user_id)
);

CREATE TABLE occasion (
    occasion_id SERIAL PRIMARY KEY,
    occasion_name VARCHAR(30) NOT NULL,
    occasion_date DATE
);
```