Recipe/Grocery List App

Functions

- Create recipes and grocery lists
- Post, comment, like, share
- Follow chefs
- Create events

Users

- User login
    - Username
    - Password
    - Email
- Can create recipes and grocery lists
- Can share recipes
- Can comment on other recipes
- Can communicate with other users

Recipes

- Have instructions
- Have ingredients
- Needed equipment and directions
- Can take images
- Take comments
- Can send ingredients to grocery list
- Can be marked public or private
- Can be added to specific occasions/dates

Grocery Lists

- Have ingredients/products to get
    - Price and amount
- Locations for stores

Tables

Users

- Username
- Email
- Password
- Bio
- User id

Authentication

- Username
- Email
- Password

Recipes

- Pictures

- Description
- Ingredients/equipment
- Instructions
- Recipe Id
- User id
- Tags and comments
- Public/Private

Comments

- Recipe id
- Content
- User id
- tags/categories
- Tag id

Grocery cart

- User id
- Ingredients
- equipment

Tags

- Tag id
- name

Ingredients

- Brand
- Name
- Price
- Ingredient id

Event

- Occasion id
- Name
- Recipes
- Description
- Time
- Date and season
- Tags

Relationships

One to one

- User to authentication
- Grocery cart to user

One to many

- Grocery cart to recipes/ingredients
- User followers
- Occasions to recipes
- Comments to recipe
- Ingredients to recipe

Many to many

- Tags to recipes

Columns

Users

- User_id
  - to keep count of the users on the platform and track relevant things on the app
  - Int serial primary key because it is unique to the user
- Email
  - To contact the user
  - Varchar because it would be a short string
- Password
  - Users password to login to the app
  - Varchar because it is a string of letters, numbers, symbols, etc
- Username
  - User's name on the platform
  - Varchar
- Bio
  - Any info the user may share with others if they want
  - Text so that it isn't limited by character count

Authentication

- User_id
  - Integer number representing a specific user's stored information used for login
  - Foreign key to users table
- Username
  - Stored user's username
  - Varchar to match the user's username they registered with
- Password
  - Stored user's password
  - varchar
- Email
  - Stored user's email
  - varchar

Recipes

- Recipe_id
  - To keep count and track of the specific recipe on the app
  - Int-serial primary key because it is unique to each recipe
- User_id
  - Foreign key from user to tell which user created the recipe
- Tag_id
  - Foreign key from tags table to tell what specific tags/categories the recipe is in
- Ingredient_id

    - Foreign key from ingredients table to tell what specific ingredients are in the recipe
- public/private
    - Boolean true or false to set the recipe to public or private views
- Description
    - Description of the recipe
    - Text so it is not limited by character count
- Instructions
    - Instructions to make the recipe
    - Text so it is not limited by character count
- Comment_id
    - Foreign key linking a specific comment to the recipe

Comments

- Comment_id
    - Int-serial primary key to keep track of specific comments
- User_id
    - Foreign key from users table to tell which user made the comment
- Recipe_id
    - Foreign key from recipe table to tell which recipe the comment was made to
- Comment_content
    - Actual content of the comment
    - Text

Grocery cart

- Cart_id
    - Int-serial primary key to keep track of individual grocery lists
- User_id
    - Foreign key from users because it is a specific user's grocery list
- Ingredient_id
    - Foreign key form ingredients because they are specific ingredients

Ingredients

- Ingredient_id
    - Int-serial primary key to keep track of individual ingredients
- Name
    - Name of ingredient
    - varchar
- Price
    - Price of ingredient
    - float

Occasions

- Occasion_id
    - Int-serial primary key to keep track of specific occasions
- Occasion_name
    - Name of occasion
    - varchar

- Description
    - Description of occasion
    - Text so it isn't limited by character count
- Recipe_id
    - Specific recipes that will be at the occasion
- Date
    - Date of occasion
    - Date data-type
- Tag_id
    - Specific tags/categories the occasion may fall under

Tags

- Trag_id
    - Int-serial primary key to separate specific tags
- Tag_name
    - Name of said tag

SQL Code

```
CREATE TABLE users (
    user_id SERIAL PRIMARY KEY,
  email varchar(50) NOT NULL UNIQUE,
  password varchar(100) NOT NULL,
  username varchar(50) NOT NULL,
  bio TEXT
);

CREATE TABLE authentication(
    user_id INTEGER NOT NULL,
  username VARCHAR(50) NOT NULL UNIQUE,
  password VARCHAR(100) NOT NULL,
  email VARCHAR(50) NOT NULL
);

CREATE TABLE recipes (
    recipe SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL,
  tag_id INTEGER NOT NULL,
  ingredient_id INTEGER NOT NULL,
  Public_Private BOOLEAN NOT NULL,
  Description TEXT NOT NULL,
  Instructions TEXT NOT NULL,
  comment_id INTEGER NOT NULL
);
```

```
CREATE TABLE grocery_cart (
    cart_id SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL,
  ingredient_id INTEGER NOT NULL
);

CREATE TABLE tags (
    tag_id SERIAL PRIMARY KEY NOT NULL,
  tag_name VARCHAR(50) NOT NULL
);

CREATE TABLE ingredients (
            ingredient_id SERIAL NOT NULL,
  name VARCHAR(50) NOT NULL,
  price FLOAT NOT NULL
 );

CREATE TABLE occasion (
    occasion_id SERIAL PRIMARY KEY,
  occasion_name VARCHAR(50) NOT NULL,
  recipe_id INTEGER NOT NULL,
  description TEXT,
  date DATE,
  tag_id INTEGER
);

CREATE TABLE comments (
    comment_id SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL,
  recipe_id INTEGER NOT NULL,
  comment_content TEXT NOT NULL
);

INSERT INTO users(password, email, bio, username)
VALUES('password123', 'abc@123.com', 'This is my bio', '789*');

INSERT INTO tags(tag_name)
VALUES('Breakfast'),
    ('Lunch'),
  ('Dinner');
```