Data Modeling

STEP 1

Brainstorm

- Sign in: users can sign into the app with their email and password.
    - Sign in with email and password
- Users can have a "library" of collected recipes.
    - Each recipe will have a list of ingredients and cooking instructions
- Users can have a grocery list to buy the ingredients that they need
- Users can can put their recipe as either private or public
- Users can have a board of events that they can add recipes to for that occasion(like pinterest)
- Users can view other users public recipes
- Users can search for recipes
- Users can post recipes
- Users can search recipes by name or type (appetizers, desserts, dinner, lunch, breakfast, cook time)
- Users can review recipes
    - Users can add comments to recipe reviews

Table Ideas

User sign table: User ID(PK), email, password
Recipe Table: id, title, ingredients_id, ingredient amount, instructions userID, private/public, cook time
Grocery List: id, user id, ingredient id
Ingredients table: ingredient ID, ingredient name, ingredient amount.
Category: category id, recipe ID, category type
Board: id, user ID, name, recipe id
Reviews: id, user id, body, rating
Comments: id, review id, body, userID

Relationships

One-to-One	One-to-Many	Many-to-Many
Grocery	User	Recipe table
Category	board	Ingredients table
	review	Comments

STEP 2

Columns

__User sign table__:
Columns & datatype: User ID(PK) serial, email varchar, password varchar
Why store data: To track user
Why datatype: User ID should be serial PK so that it increments and it's a PK. Email and password are varchar because they could include numbers and letters

__Recipe Table__:
Columns & datatype: ID serial, title varchar, ingredients_id serial, ingredient amount integer, instructions var char userID serial , private boolean, cook time integer
Why store data: To track the recipes that are created and that we will create
Why datatype: *Recipe ID* should be serial PK so that it increments and it's a PK. *Title* and *Instructions* are varchar because they could include numbers and letters *Ingredients ID* is an Int because it is auto-generated and referenced in other tables, *Ingredient amount* is an int because it is a number, *UserID* is int because it is referenced in the user table and doesn't have to be generated/incremented again, Private/public is boolean because there can only be two options & cook time is an in because it should be entered as a number.

__Grocery List__:
Columns & datatypes: ID serial, user id int, ingredient id int
Why store data: to make a grocery list that users can add ingredients to.

Why data type: ID is serial so each one can have a unique ID to refer to. User ID is integer so that each user continues to have a unique ID. Ingredient ID is int so each different ingredient will have a unique ID.

Ingredients table:
Columns & data types: ingredient ID int, ingredient name varchar, ingredient amount int.
Why store data: ingredients should be stored in order to pull them into grocery lists
Why data type: ingredient id should be unique to the table, ingredient name should be entered as text and amount is int because it should be entered as a number

Category:
Columns & datatypes: category id serial, recipe ID integer , category type varchar
Why store data: so that recipes can be searched by category
Why data type: Category id needs to be a unique id for the table. Recipe ID is referenced from the recipe table and category type is varchar because it should be letters

Board:
Columns & data types: Id serial, user ID int, name, recipe id int
Why store data: each user can have a board for different events they can save recipes to.
Why data type: board id is serial so it can generate a new ID for each board made. User ID is integer so it can refer back to the user table. Name is varchar so you can add what you want to name the board. Recipe ID is int so it can refer back to a pre made recipe.

Reviews:
Columns & data types: ID serial, user id int, body varchar, rating int
Why store data: reviews will be stored so that users can see how recipes are rated
Why data type: Review ID has to be unique to the table, user ID is referenced in the user table, body should be varchar so that numbers and letters can be used and the rating is int because users should enter it as a number

Comments:
Columns & data type: Id serial, review id int, body, userID int
Why store data: Comments will be stored so that other people can view them and not just the person who types it.
Why data type: the comment ID is serial so it can generate a unique ID for each comment. Review ID is integer so it can pull the unique ID from the review table. Body

is varchar so the user can type in what they want for the comment. User ID is integer so it can pull the users unique ID from the users table.

```sql
create table users (
  user_ID serial PRIMARY KEY,
  email varchar(255),
  password varchar(255)
);
create table recipeTable (
  user_ID int references users(user_ID),
  recipe_ID serial PRIMARY KEY,
  title varchar(255),
  ingredient_ID int references ingredients(ingredient_ID),
  ingredient_amount integer,
  ingredient_name varchar(255),
  private boolean default false,
  cook_time integer
);
create table groceryList (
  grocery_ID serial PRIMARY KEY,
  user_ID integer references users(user_ID),
  ingredient_ID integer
);

create table ingredients (
  ingredient_ID serial PRIMARY KEY,
  ingredient_name varchar(255),
  ingredient_amount integer
);
create table reviews (
  review_ID serial PRIMARY KEY,
  user_ID integer references users(user_ID),
  body varchar(10000),
  rating integer
);
```

```sql
create table comments (
  comment_ID serial PRIMARY KEY,
  review_ID integer references reviews(review_ID),
  body varchar(10000),
  user_ID integer references users(user_ID)
 );
create table board (
  board_ID serial PRIMARY KEY,
  user_ID integer references users(user_ID),
  recipe_ID integer references recipeTable(recipe_ID),
  name varchar(1000)
 );
create table category (
        category_ID serial PRIMARY KEY,
  recipe_ID integer references recipeTable(recipe_ID),
  category_type varchar(255)
);


insert into users (password, email)
values ('blue', 'email@email.com');

insert into groceryList (user_id, ingredient_id)
values (1, 3);

insert into recipetable (user_id, title, ingredient_id, ingredient_amount, ingredient_name,
cook_time)
values (1, 'This is my Table!', 4, 1, 'sugar', 75);

insert into reviews (body, rating, user_id)
values ('Wow! What a great thing!', 5, 1);

select * from reviews;
insert into comments (review_id, body, user_id)
values (1, 'Wow you are so right.  It is a great thing!', 2);
```