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

Data:

Username
First name
Last name
Email
Password
Recipes
My recipes
Recipe name
Ingredients
Instructions
Is recipe public
Grocery list
Occasions

Tables Ideas:
Users: holds name, email, and password
Recipes: holds recipe names and relates to the ingredients and instructions for that recipe
Grocery list: relates to ingredients that are added to the users grocery list
My recipes: contains a users recipes and whether it is public or private
Instructions: holds instructions for a recipe and the ingredients needed
Occasions: contains the date of occasion and will relate with the recipes needed for the occasion
Ingredients: holds a list of ingredients that relates to a recipe

Relationships:

One-to-one:
Recipes and instructions: each recipe will only have one instruction and the instructions will only apply to one recipe
Users and My recipes: each user can only have one list of their recipes and that list will only apply to that user
One-to-many:
Users and Grocery List: each user will have one grocery list but multiple users can have a grocery list
Occasions and users: each user can have multiple occasions but each occasion will only apply to one user

Many-to-many:
Grocery list and ingredients: there are many ingredients on a grocery list and each ingredient can be a part of multiple users list
Occasions and recipes: each occasion will have access to multiple recipes and each recipe can be a part of multiple recipes
My recipes and recipes: each recipe can be a part of different users My Recipes and each My Recipes can contain multiple recipes

Columns:

Users:
First name and last name: VarChar limits length of name and names allow users to find each other
Email: VarChar limits length and email allows user to sign in with correct password
Password: text allows password to be hashed and as complicated as necessary and is necessary for a user to log in using correct email

Recipes:
Ingredients_id: integer since it is relating to the ingredients table and allows all ingredients needed for a recipe to exist in the table
Instructions_id: integer since it is relating to the instructions table and allows the instructions needed for a recipe to exist in the table

Instructions:
Recipe_id: integer because it is relating to the recipes table and allows the instructions to relate to a specific recipe and also obtain the ingredients needed
Content: text because instructions can be long or short and gives the instructions to make a recipe

My Recipes:
User_id: integer because it is relating to the users table and allows a user to add recipes to a My recipes list
recipe_id: integer because it is relating to the recipes table and allows a user to find what recipe they want to add to their list
Public: boolean because it will either be public or private and decides if other users can view the My Recipes of a user

Grocery List:
recipe_id: integer because it is relating to the recipes table and allows a user to find which recipes ingredients they want to add to their grocery list
User_id: integer because it is relating to the users table and allows a user to add ingredients from a recipe their grocery list

Occasions:
User_id: integer because it is relating to the users table and allows a user to add recipes to an occasion
recipe_id: integer because it is relating to the recipes table and allows a user to find what recipe they want to add to their occasion

```
CREATE TABLE `Users` (
      `user_id` INT NOT NULL AUTO_INCREMENT,
      `first_name` varchar(20) NOT NULL,
      `last_name` varchar(20) NOT NULL,
      `email` varchar(50) NOT NULL,
      `password` TEXT NOT NULL,
      PRIMARY KEY (`user_id`)
);

CREATE TABLE `Ingredients` (
      `ingredients_id` INT NOT NULL AUTO_INCREMENT,
      `ingredient_name` varchar(30) NOT NULL,
      PRIMARY KEY (`ingredients_id`)
);

CREATE TABLE `Recipes` (
      `recipes_id` INT NOT NULL AUTO_INCREMENT,
      `ingredients_id` INT NOT NULL,
      `instructions_id` INT NOT NULL,
      PRIMARY KEY (`recipes_id`)
);
```

```
CREATE TABLE `Occasions` (
      `occasions_id` INT NOT NULL AUTO_INCREMENT,
      `recipes_id` INT NOT NULL,
      `user_id` INT NOT NULL,
      PRIMARY KEY (`occasions_id`)
);

CREATE TABLE `GroceryList` (
      `grocery_list_id` INT NOT NULL AUTO_INCREMENT,
      `recipes_id` INT NOT NULL,
      `user_id` INT NOT NULL,
      PRIMARY KEY (`grocery_list_id`)
);

CREATE TABLE `Instructions` (
      `instructions_id` INT NOT NULL AUTO_INCREMENT,
      `recipe_id` INT NOT NULL,
      `content` TEXT NOT NULL,
      PRIMARY KEY (`instructions_id`)
);

CREATE TABLE `MyRecipes` (
      `my_recipe_id` INT NOT NULL AUTO_INCREMENT,
      `user_id` INT NOT NULL,
      `recipe_id` INT NOT NULL,
      `public` BOOLEAN NOT NULL DEFAULT true,
      PRIMARY KEY (`my_recipe_id`)
);

ALTER TABLE `Recipes` ADD CONSTRAINT `Recipes_fk0` FOREIGN KEY
(`ingredients_id`) REFERENCES `Ingredients`(`ingredients_id`);

ALTER TABLE `Recipes` ADD CONSTRAINT `Recipes_fk1` FOREIGN KEY
(`instructions_id`) REFERENCES `Instructions`(`instructions_id`);

ALTER TABLE `Occasions` ADD CONSTRAINT `Occasions_fk0` FOREIGN KEY
(`recipes_id`) REFERENCES `Recipes`(`recipes_id`);

ALTER TABLE `Occasions` ADD CONSTRAINT `Occasions_fk1` FOREIGN KEY
(`user_id`) REFERENCES `Users`(`user_id`);

ALTER TABLE `GroceryList` ADD CONSTRAINT `GroceryList_fk0` FOREIGN
KEY (`recipes_id`) REFERENCES `Recipes`(`recipes_id`);
```

```
ALTER TABLE `GroceryList` ADD CONSTRAINT `GroceryList_fk1` FOREIGN
KEY (`user_id`) REFERENCES `Users`(`user_id`);

ALTER TABLE `Instructions` ADD CONSTRAINT `Instructions_fk0` FOREIGN
KEY (`recipe_id`) REFERENCES `Recipes`(`recipes_id`);

ALTER TABLE `MyRecipes` ADD CONSTRAINT `MyRecipes_fk0` FOREIGN KEY
(`user_id`) REFERENCES `Users`(`user_id`);

ALTER TABLE `MyRecipes` ADD CONSTRAINT `MyRecipes_fk1` FOREIGN KEY
(`recipe_id`) REFERENCES `Recipes`(`recipes_id`);


INSERT INTO users (first_name, last_name, email, password)
VALUES ('Ridge', 'Christensen', 'ridgechristensen28@gmail.com',
'Password1'),
('Jade', 'Christensen', 'jadechristensen16@gmail.com', 'Password2')
```