Summary

We want to create a recipe creating/sharing and grocery list app. You'll be planning out what tables we'll need, what information they'll store, and how the data will relate to each other.

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

Part 1: Conceptual Planning - Word/Google/Pages Doc

Step 1

- In your doc, start a list and title it "Brainstorming".
- Take 10 minutes individually and brainstorm all the different things you might need to keep track of for this app.
- Don't worry if you repeat something or if it seems dumb as soon as you think it – **write it down anyway**.
- Walk through some different user flows to help you think of all the data they might come across. You could also quickly sketch out some different views of the app and figure out what data would be needed for each view.
- After 10 minutes, reconvene with your partner and share your ideas.

Brainstorming

- Id
- Email
- Password
- Recipes
    - Public
    - Private
- Ingredients
- Instructions
- Grocery list
- Occasions'

Step 2

- Title this next section "Table Ideas".

- Based off the ideas you just brainstormed, list out what tables you think you'll need. It's okay if you change it up later.
- Write a brief description of each table. For example: "products: this table will hold information about the products offered, each row will be an individual product".

Table Ideas

users
- id: SERIAL PRIMARY KEY
- email: VARCHAR(255)
- password_hash: VARCHAR(255)

recipes
- id: SERIAL PRIMARY KEY
- name: VARCHAR(255)
- instructions: VARCHAR(255)
- public_private: BOOL

saved_recipes
- id: SERIAL PRIMARY KEY
- user_id: INTEGER
- recipe_id: INTEGER

ingredients
- id: SERIAL PRIMARY KEY
- name: VARCHAR(255)

groceries_list
- id: SERIAL PRIMARY KEY
- user_id: INTEGER
- ingredients_id: INTEGER

occasions
- id: SERIAL PRIMARY KEY
- name: VARCHAR(255)
- user_id: INTEGER

occasions_recipe
- id: SERIAL PRIMARY KEY

- user_id: INTEGER
- recipe_id: INTEGER

Step 3

- Next figure out what relationships should exist among the data.
- Title this section “Relationships” and create 3 sub-sections as well – “one-to-one”, “one-to-many”, and “many-to-many”.
- In each subsection, list the tables that have that relationship **and explain why you chose that relationship.**
- For example, let’s say I’m planning an ecommerce app. In the one-to-many section, I could have: “products table and review table because each product can have multiple reviews, but a review is only for one specific product”.
- Note: remember tables can relate to multiple other tables. In the product/reviews example above, reviews would also have a relationship with a users table. And product could be related to a cart table. There could be others as well.

Step 4

- Now that you know what relationships you’ll have, go back to your “Table Ideas” and **in a different text color** add in any additional tables and their descriptions that you’ll need.

Part 2: Table Planning - DB Designer & Word/Google/Pages Doc

Step 1 - DB Designer

- Create a new schema.
- Insert all your tables.
- Go back through each table, figure out what columns you need, and add them in.
- As you’re adding in columns, don’t forget to select the data types you want and set any primary or foreign keys.
- When you’re done, go to the Schema menu and “Save As …”, then go to the Export menu and select “PDF …”.

Step 2 - Word/Google/Pages Doc

- Make a new section below “Relationships” called “Columns”. Create a sub-section for each of your tables.

- **List out each table's respective columns in the table's sub-section and explain for each column:**
    - why you'll be storing that data
    - and why you chose the data type you did

Part 3: Create Tables in SQL - Postgres Sandbox & Word/Google/Pages Doc

- Go to https://postgres.devmountain.com/
- Write a create table statement for each of your tables
- Copy and paste each of the statements into your doc so you can keep track of them and turn them in

Intermediate

- Try inserting some data into your tables on https://postgres.devmountain.com/
- Make sure to save your SQL code in your Word/Google/Pages doc

```
---------------------
-- Create Tables

CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  email VARCHAR(255),
  password_hash VARCHAR(255)
)

CREATE TABLE recipes (
  id SERIAL PRIMARY KEY,
  FOREIGN KEY (id) REFERENCES users(id),
  name VARCHAR(255),
  instructions TEXT,
  isPublic BOOLEAN NOT NULL DEFAULT FALSE
)

CREATE TABLE saved_recipes (
  id SERIAL PRIMARY KEY,
  FOREIGN KEY (id) REFERENCES users(id),
```

```
  FOREIGN KEY(id) REFERENCES recipes(id)
)

CREATE TABLE ingredients (
  id SERIAL PRIMARY KEY,
  name VARCHAR(255)
)

CREATE TABLE recipe_ingredients (
  id SERIAL PRIMARY KEY,
  FOREIGN KEY(id) REFERENCES ingredients(id),
  FOREIGN KEY(id) REFERENCES recipes(id)
)

CREATE TABLE groceries_list (
  id SERIAL PRIMARY KEY,
  FOREIGN KEY (id) REFERENCES users(id),
  FOREIGN KEY(id) REFERENCES ingredients(id)
)

CREATE TABLE occasions (
  id SERIAL PRIMARY KEY,
  FOREIGN KEY (id) REFERENCES users(id),
  name VARCHAR(255)
)

CREATE TABLE occasions_recipe (
  id SERIAL PRIMARY KEY,
  FOREIGN KEY (id) REFERENCES users(id),
  FOREIGN KEY(id) REFERENCES recipes(id)
)


-------------------------------------

-- Insert Data
```

```
INSERT INTO users (email, password_hash)
VALUES ('DwayneForemanJr@yahoo.com', 123456),
('Jon@yahoo.com', 4567),
('Kim@yahoo.com', 1111);


INSERT INTO recipes (name, instructions)
VALUES ('Sams Grill Cheese', 'Use butter and bread'),
('Special Stew', 'Marinate beef for 3 hours');

INSERT INTO ingredients (name)
VALUES ('butter'),
('rice'),
('bacon'),
('bread'),
('milk');

INSERT INTO occasions(name)
VALUES ('birthday'),
('christmas'),
('halloween');
```

Submit To GitHub

Save your document as a PDF. Export your database diagram as a PDF. Create a repo for them on GitHub and upload both documents.