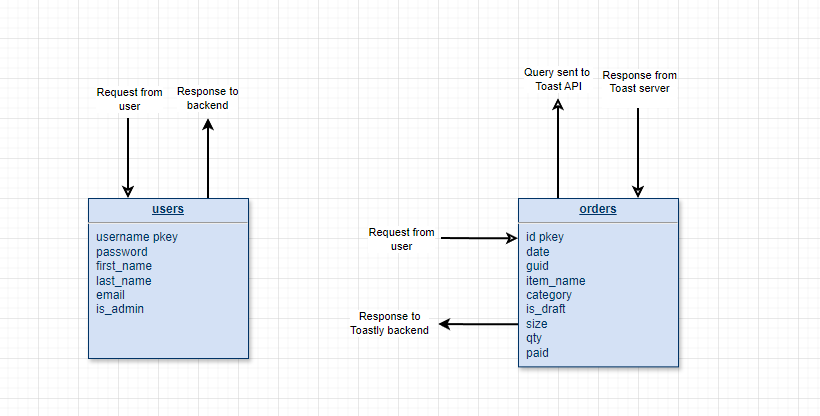
**Toastly Database Schema**

My project has a simple database that stores two separate tables. The first stores user information for logging in and profile. The second main table stores order info retrieved from the Toast API.



Users Table

CREATE TABLE users (

username VARCHAR(25) PRIMARY KEY,

password TEXT NOT NULL,

first\_name TEXT NOT NULL,

last\_name TEXT NOT NULL,

email TEXT NOT NULL

CHECK (position('@' IN email) > 1),

is\_admin BOOLEAN NOT NULL DEFAULT FALSE

);

username - primary key to check for other users with the same username  
password - stores bcrypt hashed password  
first\_name  
last\_name  
email  
is\_admin - admin rights allow creation of other users

Orders Table

The Toast API and provides a massive amount of data for each order. Originally, I was just requesting the API data through the backend, and then providing that data to the frontend to process. I decided to create a table to store only the necessary data from the API. This decreases the amount of data sent between the frontend and backend, and it gives the database a bit more usage than just storing user credentials.

The backend filters and parses the data sent from the API in order to check if it already exists in the database. It does this by checking if the date of the order already exists in the DB. If it does, it skips adding it.

CREATE TABLE orders (

id SERIAL PRIMARY KEY,

date TEXT NOT NULL,

guid TEXT,

item\_name TEXT NOT NULL,

category TEXT,

is\_draft BOOLEAN NOT NULL DEFAULT FALSE,

size TEXT NOT NULL,

qty NUMERIC,

paid NUMERIC CHECK (paid >= 0)

);

id - primary key and auto-populated when the item is added  
date - stored as yyyymmdd, the date of the order  
guid - unique identifier of the order from Toast  
item\_name - name of the item ordered  
category - the menu item category, such as “Draft Beer” “Merchandise” etc

is\_draft - used in front end if the item is draft since a draft item can have multiple sizes  
size - if draft, will store a size, otherwise stores “”  
qty - the item quantity of the order  
paid - net amount paid for the order