IDG2001 Assignment 1 report

TEA - The Everything App (the twitter clone) was developed with python and FastAPI in the backend, React in the frontend and a PostgreSQL database. These are all deployed as 3 different services on render.

The database is a free PostgreSQL instance on render, which means it will automatically be deleted 30 days after creation, in my case april 20th. This means if you try to go to the site after april 20th, it will not work properly.

Additionally, the backend spins down pretty quickly with inactivity, so when first accessing the site you will likely have to wait ~1-2 minutes for it to load.

Frontend URL:
https://tea-dofo.onrender.com/

Backend URL:
https://tea-theeverythingapp.onrender.com/

API Endpoints

GET https://tea-theeverythingapp.onrender.com/tables
Gets a list of all the tables in the database with their columns.

GET https://tea-theeverythingapp.onrender.com/accountposts
Gets a list of every account along with all their posts.

GET https://tea-theeverythingapp.onrender.com/accounts
Gets a list of every account.

GET https://tea-theeverythingapp.onrender.com/posts
Gets a list of every post.

GET https://tea-theeverythingapp.onrender.com/myposts
Gets a list of the logged in user's posts.

POST https://tea-theeverythingapp.onrender.com/posts
Creates a new post.

PUT https://tea-theeverythingapp.onrender.com/posts/{post_id}
Edits a post by post_id. Logged in user must be the same as the author of the original post.

DELETE https://tea-theeverythingapp.onrender.com/posts/{post_id}
Deletes a post by post_id. Logged in user must be the same as the author of the original post.

GET https://tea-theeverythingapp.onrender.com/search/posts
Gets a list of posts matching the query.

GET https://tea-theeverythingapp.onrender.com/search/accounts
Gets a list of accounts matching the query.

POST https://tea-theeverythingapp.onrender.com/user/register
Registers a new user.

POST https://tea-theeverythingapp.onrender.com/user/login
Logs in an existing user.

Database structure

Accounts
- Accountid, integer (primary key)
- Email, varchar
- Username, varchar
- Passwordhash, text
- Createdat, timestamp without time zone

Posts
- Postid, integer (primary key)
- Accountid, integer (foreign key)
- Parentpostid, integer (foreign key)
- Content, varchar
- createdat, timestamp without time zone
- Lastedited, timestamp without time zone

Followers
- followerid , integer
- Followingid, integer

Likes
- Accountid, integer
- Postid, integer

Followers and likes are currently not implemented, so those tables are unused for the time being. Parentpostid in the posts table exists to be used for replies, but replies have not been implemented either. These were included in the database structure based on an earlier draft of the assignment where these features seemed to be required.