Team POGGERS -- Kelvin Ng (PM), Tanzim Elahi, Kevin Li, Justin Shaw
SoftDev1 pd1
P02 -- The End
2020-01-02

Cookie Clicker

API(s) used:

- speedrun.com REST API (no API key required - just don't spam it with requests, it does time you out for a few minutes for doing so)

Minimum Viable Product

We will be recreating the Cookie Clicker game with the skills we've learned over the course of this semester. This project will minimally have features such as:

- Making an account to save progress
- Registering that the cookie on screen has been clicked and recording it into the database
    - Update every 5 seconds to avoid high resource usage
    - Also include a manual save progress button
- A shop that can convert cookie clicks into perks (such as auto clickers) - those perks will also be stored in some way.
    - Maybe different backgrounds for the website? Different fonts?
- The cookie clicker leaderboards will be retrieved via the speedrun.com API from here: https://www.speedrun.com/cclicker#1_Million_Cookies

Extra Stuff

- Recording data
    - Milestones/achievements (e.g. "1000 Cookies Earned!")
    - Time and date of said achievements (database layout will get even more complicated though)
    - The time it took to achieve said achievements after creating your account (same as above)
- Track clicks per second
- Timed trials (30s, 1min, 2min) - no perks allowed in those games.
    - Compare your time trial stats with those of others on the leaderboards page
- Allow change in username and password in profiles

Bootstrap or Foundation?

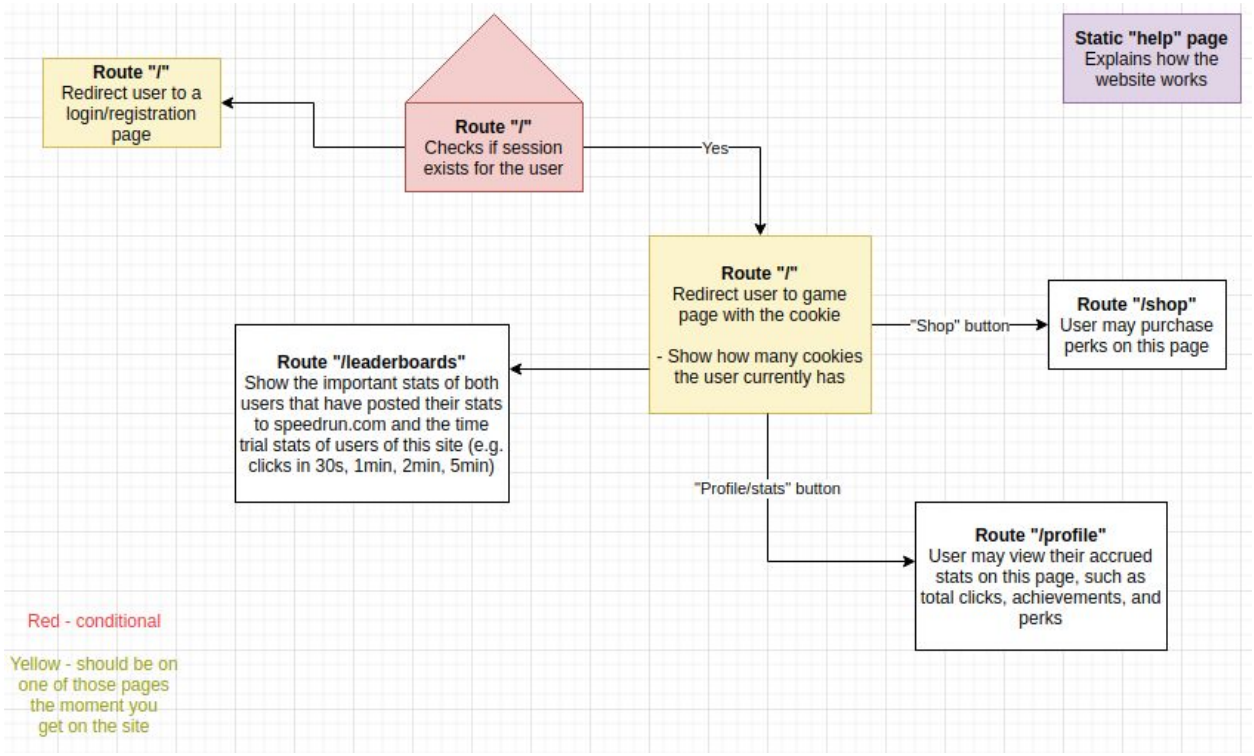
We will be using Bootstrap for our front-end styling purposes. The majority of the group is familiar with Bootstrap, and as such we believe our project will look nicer using it compared to using Foundation.

Task Division

- Kelvin Ng
    - Project Manager
    - Update README, make sure design doc is current, manage devlog
    - Minor coding tasks if necessary
- Tanzim Elahi
    - Backend implementation of databases and interactions with frontend
    - Help with frontend (templates, routes, styling with Bootstrap)
- Kevin Li
    - Interfacing with the API to retrieve the necessary data
    - Backend interactions with the frontend cookie clicker
    - Backend implementation of accounts
- Justin Shaw
    - JavaScript necessary for clicking the cookie and purchasing perks
    - Frontend (templates, routes, styling with Bootstrap)

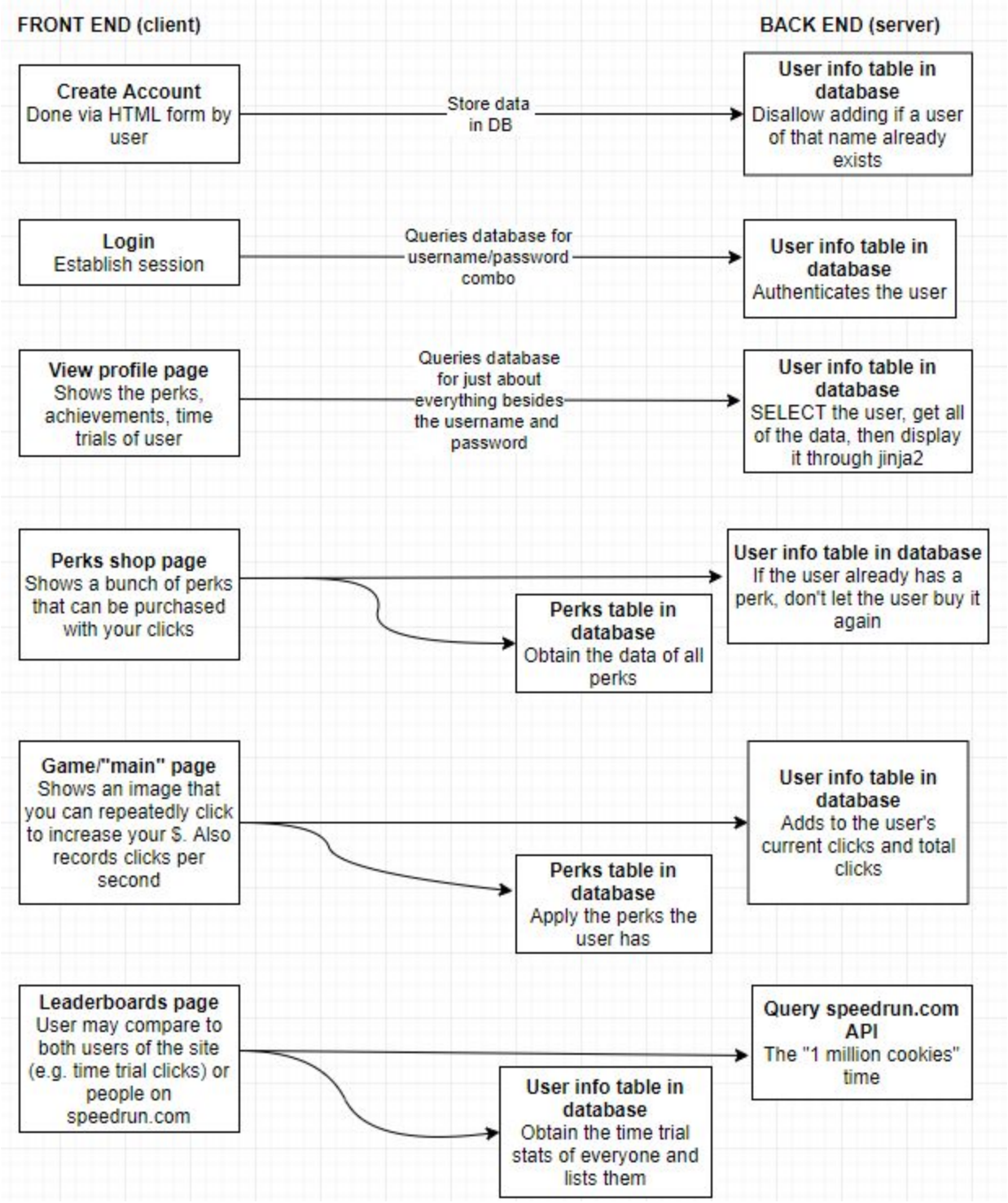
Site Map

https://drive.google.com/file/d/1OT7McKfBYJeqV-S1p1pyWjG-XrJa73J-/view?usp=sharing

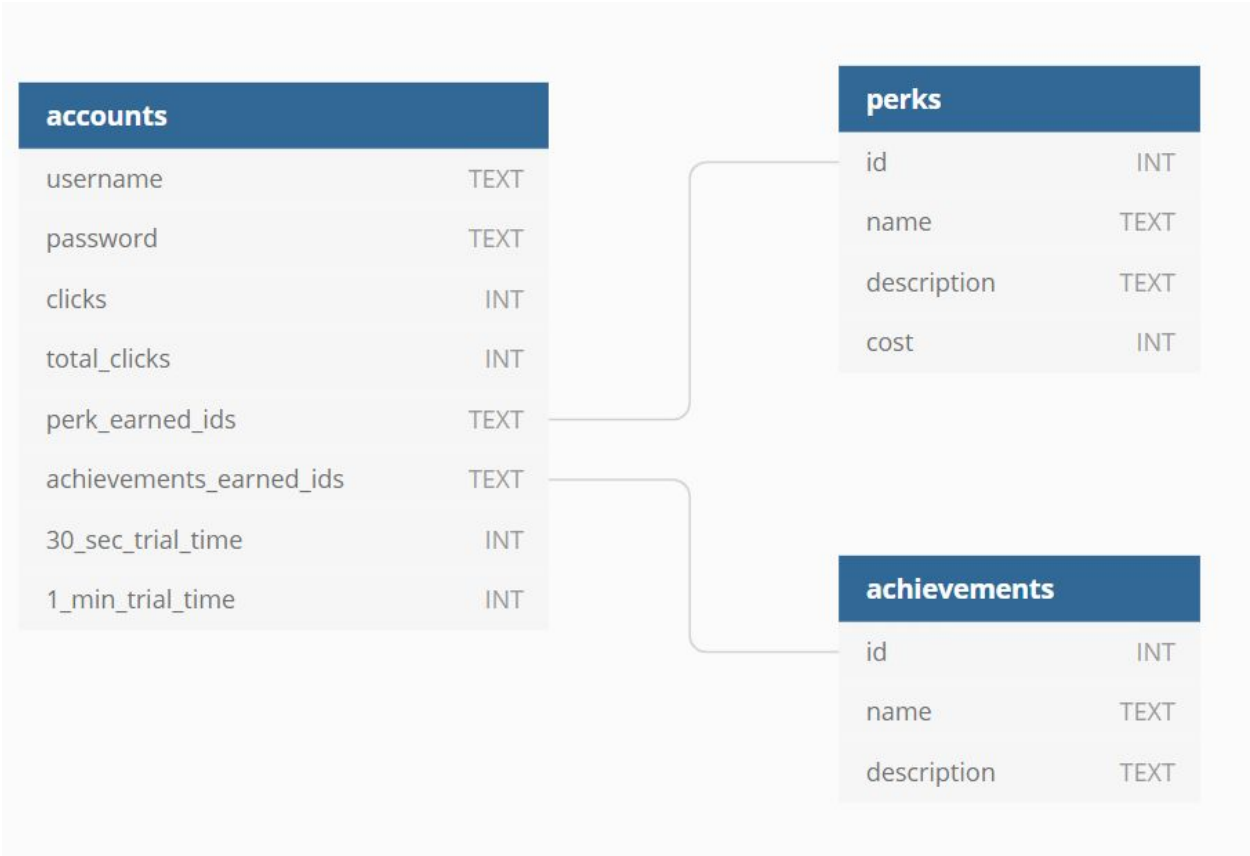


Component Map

https://drive.google.com/file/d/1hO6Qf2li9C-VKgmstcfaT3zDvUZAu-CB/view?usp=sharing



Database Layout



Note: perked_earned_ids and achievements_earned_ids will store a string separated by commas (i.e. “0, 2, 3”). These IDs correspond to the perks or achievements with that ID.

Note 2: clicks is how much “money” you have. It is different from total clicks in that it can be lowered after buying stuff, whereas total clicks never go down.

Note 3: the trial times will be in clicks. The leaderboards page can retrieve the clicks everyone has and compare them.

Note 4: will have to parse the perked_earned_ids and achievements_earned_ids and edit every time. Not sure if there’s a better solution for this.

Note 5: if we have time, we should have a separate records table rather than stuffing everything in the accounts table.