

The Hamsters of Destiny - Noakai Aronesty, Hebe Huang, Annabel Zhang, Justin Zou

SoftDev

P01 -- The Hamster Wheel | Design Doc

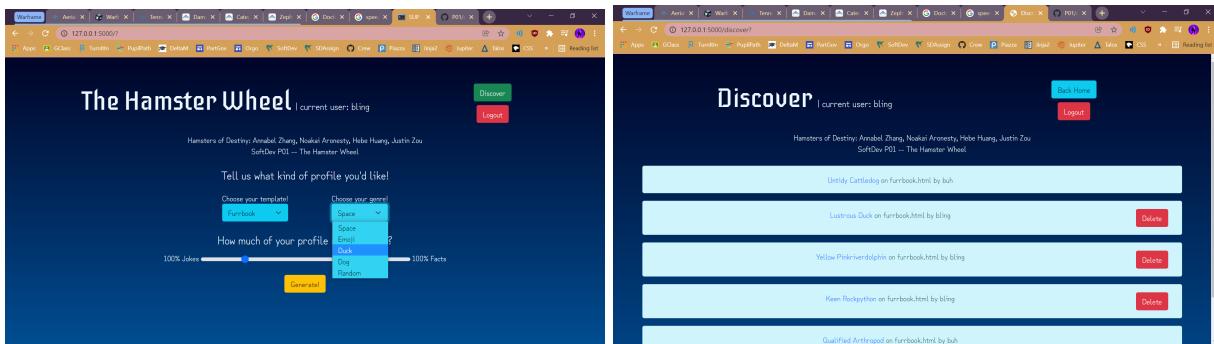
2021-01-04

Project Description:

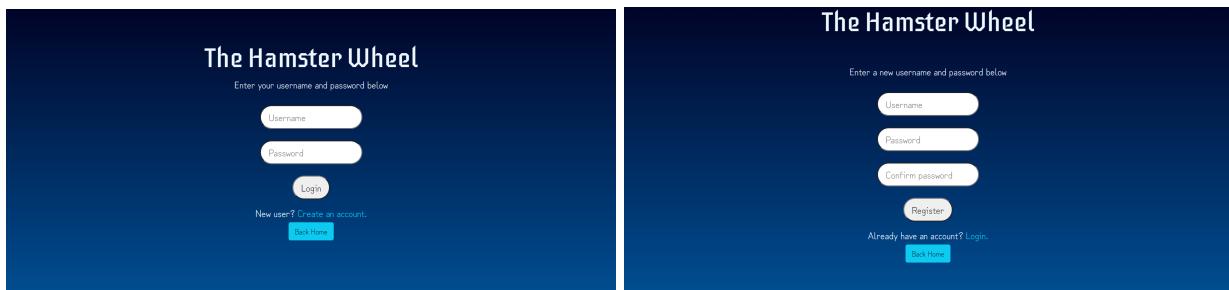
Our project is a random profile generator in which users can select from a variety of templates and genres to generate semi-randomized fictional profile pages. The profile will have a profile picture, banner, bio, and other information such as facts or jokes that are randomly chosen from APIs. The number of random facts/random jokes in the profile as well as the template and genre the profile uses can be customized in the home/generate page.

Templates:

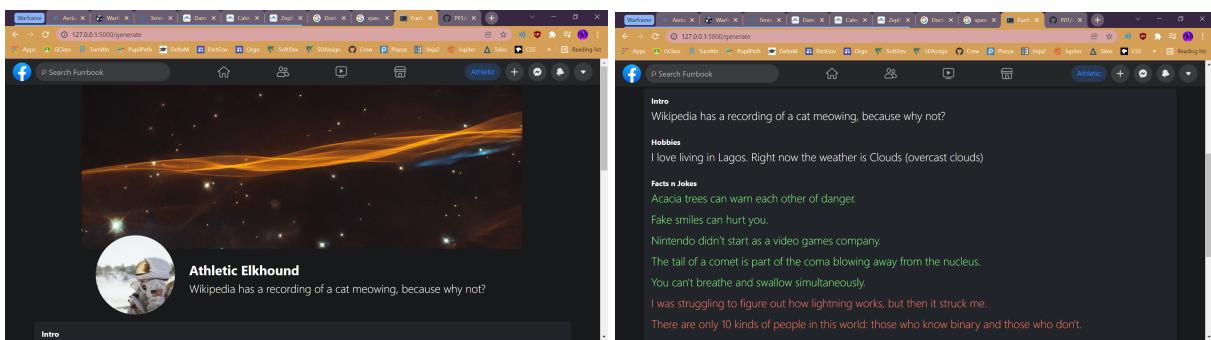
Home and Discover Pages:

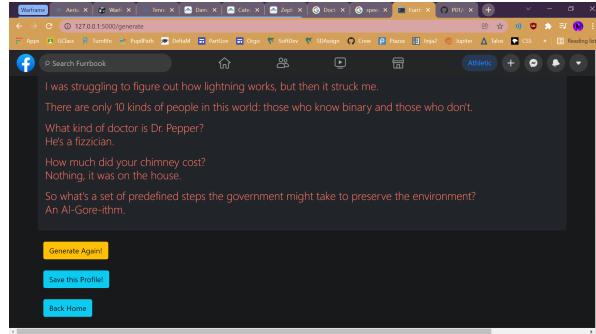


Login and Register Pages:



Furrbook:





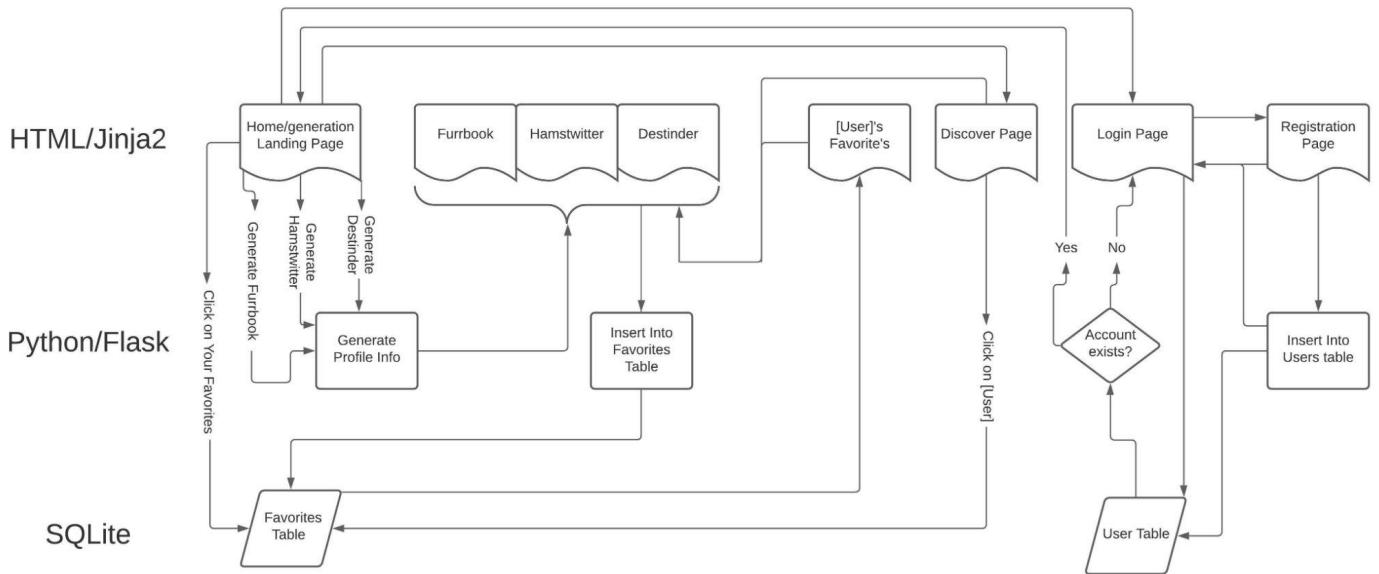
Destinder:

Hamstwitter:

Program Components:

- Templates
 - Home.html: Landing Page with a button that creates a profile. Has control panel that customizes the type of profile that comes out
 - <Template>.html: Page that contains and shows the randomized profile. Users can choose to save their profile and let it be shown on discover.html.
 - Furrbook.html
 - Hamstwitter.html
 - Destinder.html
 - Login.html: Page users use to login
 - Register.html: Page users use to register
 - Discover.html: Page to find other users and profiles and delete or view your own saved profiles
- Flask App
 - `__init__.py`
 - Responsible for *all* flask path functions:
 - “/”: Renders home.html
 - “/generate”: Handles the profile pages
 - “/login”: Handles logging in
 - “/register”: Handles creating new users
 - “/discover”: Renders the discover page
 - “/logout”: Logs the current user out
 - “/save”: Saves a profile
 - “/deleteprofile”: Deletes a profile that the current user saved
 - “/auth” and “/rAuth”: For login/register authentication purposes
 - `api.py`
 - Responsible for most auxiliary functions, communicating with APIs and occasionally the SQLite database.
 - Functions for:
 - Getting information from every API used
 - Saving profiles, rendering profiles, rendering saved profiles and deleting saved profiles
 - `auth.py`
 - Create tables, communicate with SQLite db.
 - Auxiliary functions for creating the database, creating users and authenticating users
- SQLite database
 - User Table
 - Saved Profiles Table

Component Map:



Database Organization:

Users

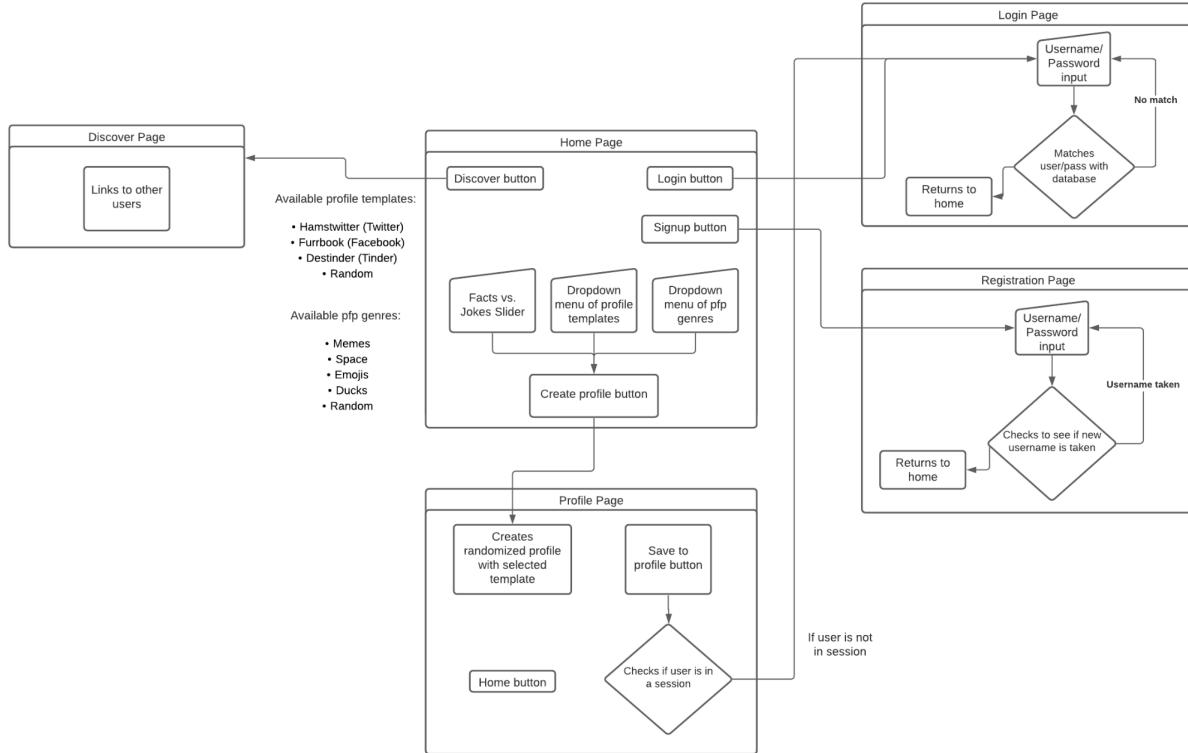
Pid (INT)	Account Name (TEXT)	Username (TEXT)
<pid1>	<name1>	<username1>
<pid2>	<name2>	<username2>
...

Profiles

Fun Cat Fact (TEXT)	Weather Fact (TEXT)	Meme 1 URL (TEXT)	Meme 2 URL (TEXT)	Age (TEXT)	Location (TEXT)	Genre (TEXT)	Date (TEXT)	Year (TEXT)
---------------------	---------------------	-------------------	-------------------	------------	-----------------	--------------	-------------	-------------

<catFact1>	<weatherFact1>	<meme1-1>	<meme2-1>	<age1>	<location1>	<genre1>	<date1>	<year1>
<catFact2>	<weatherFact2>	<meme1-2>	<meme2-2>	<age2>	<location2>	<genre2>	<date2>	<year2>
...

Site Map:



APIs that will be used:

- Text
 - Open Weather Map
 - Joke API
 - Cat Facts
 - Random word
 - Twitter
 - Meme
 - Fun fact
- Images
 - NASA
 - Random Duck
 - The Dog API
 - (<https://api.thedogapi.com/v1/images/search> to get around sign-up requirement)
 - Unsplash API

Frontend Framework:

We plan on using Bootstrap, utilizing the easy to import javascript and the clean and easy to use classes in order to make sure the information is properly laid out.

Good Bootstrap Features

- Containers!
 - Put everything in containers: they're magical.
- Alerts are a nice type of container that (may not be used for intended purpose here but) look very nice and allow access to other colors for containers.
 - Some listed items on main pages will likely be in Alerts, as well as actual alerts, like login errors and such.
- Dropdowns are easy to create, look nice and are user-friendly.
 - Profile templates or other discrete preferences like the genre.
- Buttons have a lot of support classes for aesthetics and links can be made to look like buttons
 - Navigation between pages
 - Form submission
- Display headings are special text classes that allow for much larger and better headings
 - Title or other large headings
- Forms in bootstrap are supported nicely and come with good-looking default classes.
 - Profile settings/preferences
- Tables and auto formatted (but controllable) columns
 - Positioning different profile or page elements.
- Lots of text utility like sizing, spacing, margins, font weight and more.

Task Division:

- Templates
 - Home.html (**Hebe, Justin**)
 - Login.html / Register.html (**Noakai**)
 - Discover.html (**Hebe**)
 - Profile.html (**Justin**)
 - Make sure each essential part of the profile is there (picture, bio, etc)
 - CSS / Bootstrap (**Noakai, Justin**)
- Python
 - Authentication (**Noakai**)
 - db communication (**Annabel**)
 - Slider for the different customization options (**Annabel, Hebe**)
 - html communication / display (**Noakai, Annabel**)
 - Generate profiles (**Justin, Hebe**)
 - Taking information from the APIs and putting them on the profile
 - Different profiles based on Customization

TARGET SHIP DATE: Thursday, 1/5/21

Notes:

Revisit design doc requirements

Project Name: The Hamster Wheel (?) like a lottery wheel for randomized choosing

Ideas: (separate into reqs and nice to have)

- Randomize certain sections
- Preferences
- Control Panel:
 - Sliders that limit the randomizations
 - Different profiles based on website i.e facebook
 - Hamstwitter
 - Furrbook
 - Destinder
 - Possible option: User can perhaps put in their own information and then the profile is (partially) generated from that
- Saving the profile
- Maybe looking at other people's profiles
- Maybe add upvote profile feature

Random Profile Generator:

- The profile picture would be a picture from an animal api
- Could use HelloSalut api (returns "hi" in the user's language) to generate different language profiles?
- User can add parts to their profile
- Random Facts/ things that they like

Post-Mykolyk's message:

- Be able to randomize specific part of profile (aka only pfp or only "About" section for example) - noakai
- Save feature for profiles -> create account to save or download option

<https://www.page2images.com/Convert-HTML-to-Image-or-PDF-online-API> <- we could use this api to save profiles?

Potential Future Additions:

Downloading the Profile

Saving the preferences of each User