Ample Examples by Team Joe Cubed
Ian Chen-Adamczyk, Eric Lo, Michelle Thaung, Jessica Yeung
04-22-2021
P3

Description: A website to contain lists about topics, meant to serve as creative inspiration. This website will be useful for creating in-depth universes about things you don't know that well, like how *Zootopia* is all about animals, or *Steven Universe* is all about gemstones. Users can log in to save certain items on lists as well.

Project Timeline
DUE 4/29/21
Requirements (4/25):
- Digital Ocean (4/21)
- Login features (4/22)
- API implementation and lists (4/23)
- Favorite features (4/24)
- Styling and JS (4/25)

Additional Features (4/27):
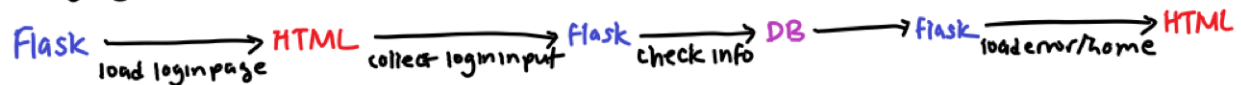- Flairs (4/26)
- More lists (4/27)

Components
- SQLite Database
  - Two user-related tables
    - One meant for logging in that contains username, user id, and password
    - One meant for storing all of the users' favorited list items
  - Multiple tables containing topic contents
    - One table per topic, containing entry name, id, picture, description, etc
- Flask (app/__init__.py)
  - Serves as the middleman between the database and templates and between the APIs and the user
  - All information being sent/received by the database, templates, APIs, and the user will pass through the application.
- HTML templates (app/templates)
  - login.html
  - register.html
  - error.html
  - home.html

- JavaScript will be used to modify the visibility of login/profile buttons
- generic_list.html
    - JavaScript will be used to tailor the template for each list
- profile.html
- CSS (app/static/css)
    - Contains CSS for all HTML templates to use
- Foundation
    - Improves the
- JavaScript (app/static/js)
    - Send requests without reloading the page
- RESTful APIs used:
    - Dog API
    - Fruityvice (kb)
    - Edamam (kb)
    - Trefle (kb)
    - MediaWiki/Wikipedia
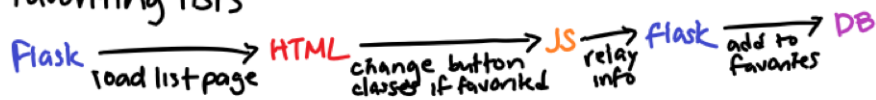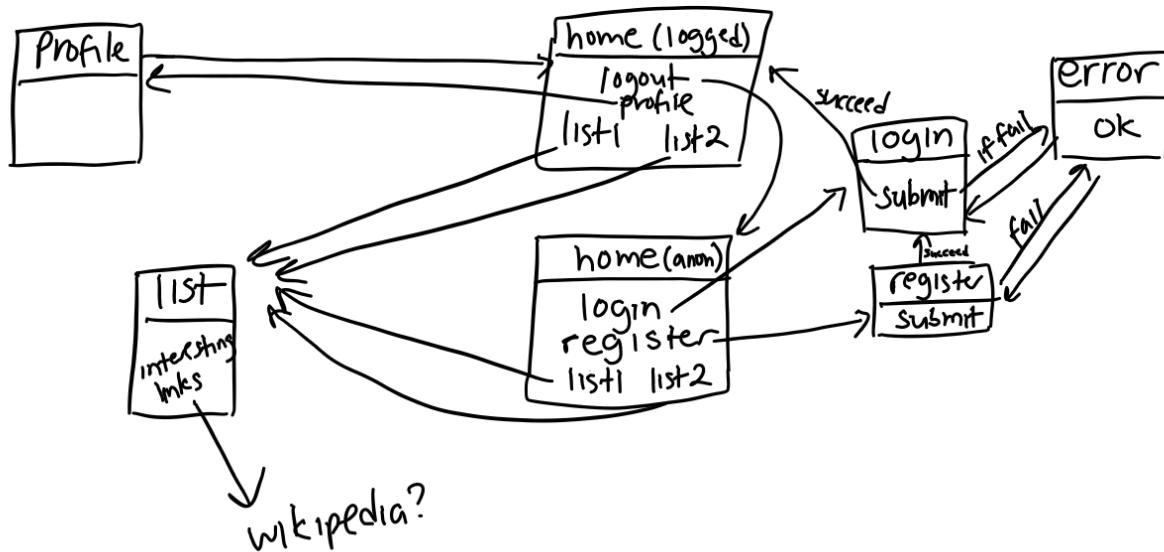
Component Visualization

Logging in

Flask →(load login page)→ HTML →(collect login input)→ Flask →(check info)→ DB →→ Flask →(load error/home)→ HTML

Fetching lists

Flask →(check update periodically)→ API →→ flask →→ DB →→ flask →(load list pages)→ HTML

Creating accounts

Flask →(load page to create account)→ HTML →(collect input)→ flask →(add entry)→ DB

Favoriting lists

Flask →(load list page)→ HTML →(change button classes if favorited)→ JS →(relay info)→ flask →(add to favorites)→ DB

Site Map

Database Organization

users		
username	password	ID
TEXT	TEXT	INTEGER primary key

favorited		
userID	listID	ID
INTEGER	INTEGER	INTEGER primary key

list<n>				
title	picture	descriptions	flairs	ID
TEXT	TEXT	TEXT	TEXT	INTEGER primary key

Tasks and Assignments

- Ian
  - Login and registration
  - Templates: login.html, register.html
  - Styling
- Eric
  - SQL
  - Templates: error.html, generic_list.html
  - Styling
- Jessica
  - APIs
  - Flask
  - Template: profile.html
  - Styling
- Michelle
  - SQL
  - Template: home.html
  - Styling

Folder Organization

- app/
    - __init__.py
    - app.py
    - data.db
    - db_builder.py
    - static/
        - css
            - styles.css
        - js
            - scripts.js
    - templates/
        - login.html
        - register.html
        - error.html
        - home.html
        - generic_list.html
        - profile.html
    - keys/
        - readme
        - key_api0.txt
        - key_api1.txt
        - key_api2.txt
- design.pdf
- devlog.txt
- flag.jpg
- README.md
- requirements.txt