Devo Council : Alex Luo, Tawab Berri, Nia Lam, Ivan Gontchar
P00: Scenario Two
TARGET SHIP DATE: 2024-11-08
Devo council, unite!

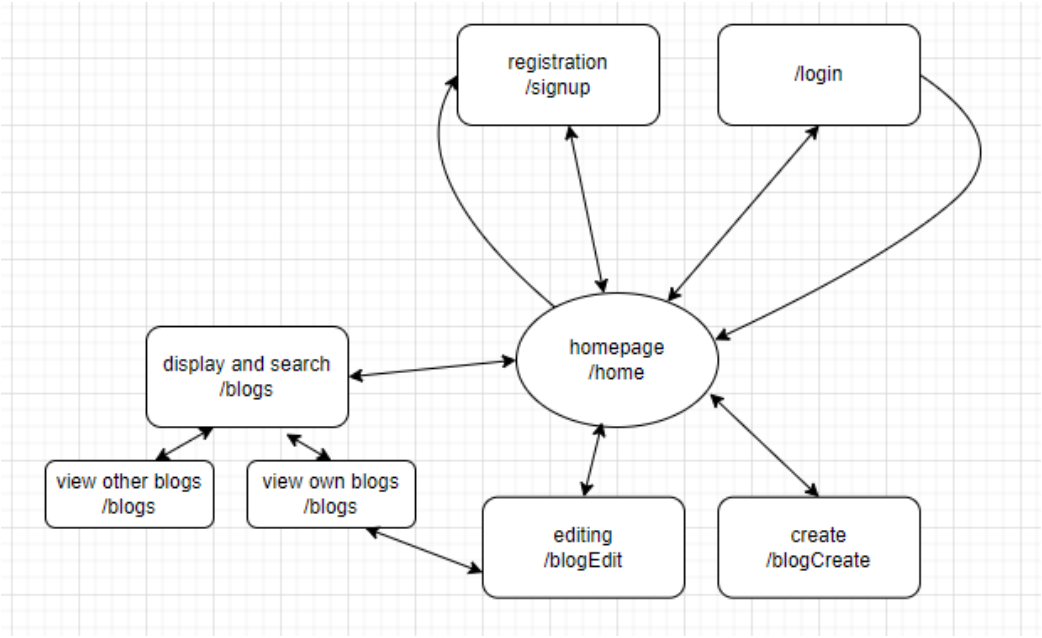
Scenario Two: Your team has been contracted to create a we**b log** hosting site, with the following features:

- Users will have to register to use the site.
- A logged-in user will be able to
    - Create a new blog
    - Update their blog by adding a new entry
    - View and edit their own past entries
    - View the blogs of other users

Design Document Specifications:

Site Map/Site Preview:

1. Home
    - Link to other blogs
    - Link to own blog (with past entries)
    - Symbol that shows whether you are logged in or not
    - Search bar (to search for other users, other blogs, own blogs)
2. Registration/Login
    - Allows users to register an account with username/password?
    - After logging in, redirects user to home page
3. Search Results
    - Shows results of the search requested from home with links, thumbnails, and summary of content for each result
4. Personal Blog
    - Link to create new blog (maybe save drafts)
    - Links to existing entries
5. Edit Personal Blogs
    - User interface to edit the entry or new blog you requested from the personal blog page
6. Other users' Blogs
    - Links to other blogs



Database Organization:

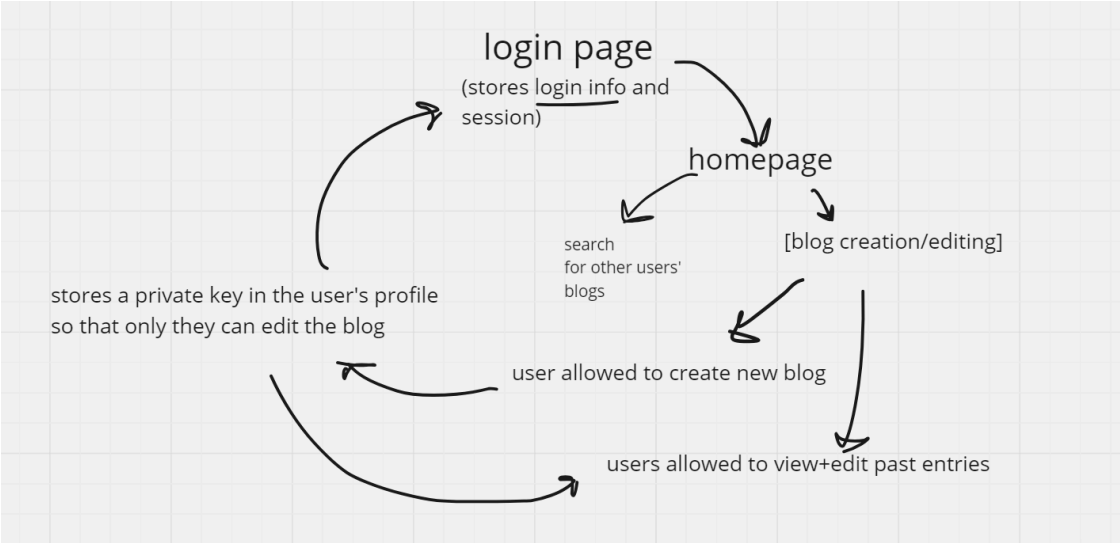
- Login data storage will be similar to in-class session activity
- If the user is the author of a blog, they will have access to add a new entry onto said blog, as well as view past entries.
- This means that blog will have to be defined as completely random keys (os.urandom(32)) and a user will have possession of the keys of the blogs that they are allowed to edit (they are the author of these blogs)
- If the user does not possess the private key of a blog they are viewing, they do not have edit permissions on said blog nor do they have permission to view past entries of this blog.
- Creation of a blog will consist of creation of a new private key that refers to said blog as well as linkage of said private key to the creator of the blog (currently logged-in user)
- Viewing of past entries will require storage of said entries, so maybe we could store entries in a list (connected to the secure private key given to the user), and append new entries to the list as the user creates them, as well as access older list items (entries) upon the user's request

Program Components:

1. blog.py: this will be the main python file that we will run to have it load all the other components of our project. It will be responsible for handling user input, storing users, and redirecting and rendering pages
2. home.html: this will be our template to display the landing page, where the user can navigate the website, as described in site map, as well as login/register
3. login.html: the template that is generated when a user is attempting to log in or register an account.

4. blogs.html: this template will load all the user's blogs from the databases and will be searchable for specific ones.
5. editing.html: template for being the user interface where one can choose if they want to edit one of their past blogs or make a new one
6. blogCreate.html: template for the page where a user could be able to create a new blog entry with prompts for things like title, short summary, actual writing for the blog, etc.
7. blogEdit.html: template for the page where a user can edit their blogs. We will also use this same framework to load the info of a past blog they may have chosen to edit. The user can then submit their entry or changes, which will be updated in the database.
8. userData.db: database where we will have all our information stored on users: their usernames, passwords, blog entries, and more
9. table users: table where we will store all our information on users with their usernames, passwords, and private keys (as mentioned above)
10. table blogs: table where we will store all our blogs with information about their general necessities, such as title, summary, content, etc, and also their private key, so we can see which user may, or may not access them.
11. style.css: makin' everything pretty

Linkage diagram of program components:



Assignments of each group member:

Tawab (PM): Back End(SQLITE/databases) + Anything Else
Ivan: Back End(SQLITE/databases)
Alex: Front End (HTML+CSS)
Nia: User Sessions (Flask) + HTML