expresso

(Team Flag)

by Espresso | *Daniel Sooknanan, Sophie Liu, Joshua Kloepfer*

SoftDev

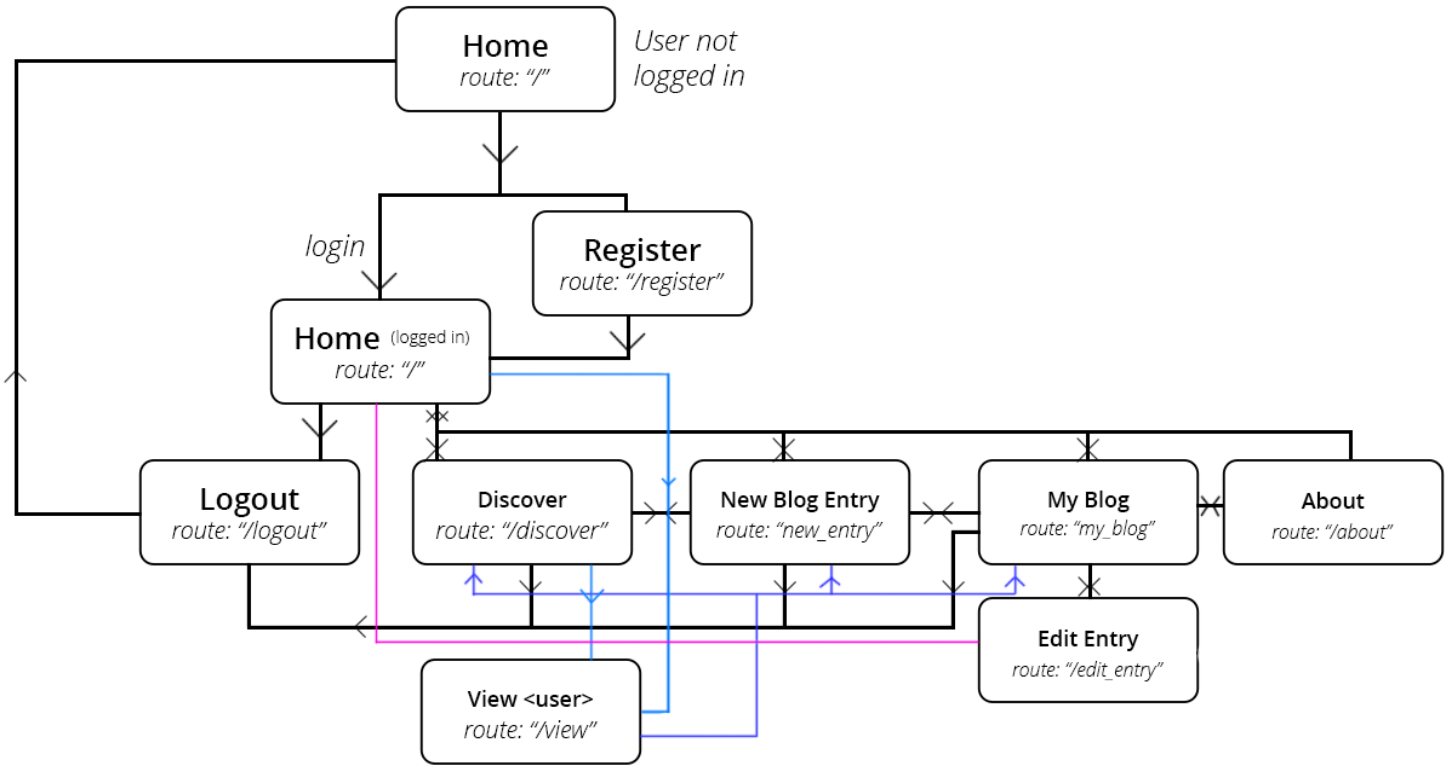
P00 - Expresso Blogging Design Document

11-11-2021

File Structure

```
flask_app
├─ app.py
├─ users.db
├─ templates/
│   ├─ login.html
│   ├─ register.html
│   ├─ home.html
│   ├─ blog.html
│   ├─ name_blog.html
│   ├─ new_entry.html
│   ├─ about.html
│   ├─ error.html
│   ├─ look.html
│   ├─ discover.html
│   ├─ edit.html
│   ├─ my_blog.html
├─ static/
│   ├─ css/
│   │   ├─ about.css
│   │   ├─ blog.css
│   │   ├─ discover.css
│   │   ├─ entry.css
│   │   ├─ error.css
│   │   ├─ login.css
│   │   ├─ style.css
├─ blogs/
│   ├─ <user_id>/
│   │   ├─ <post_num>.txt
```

Front-End Sitemap



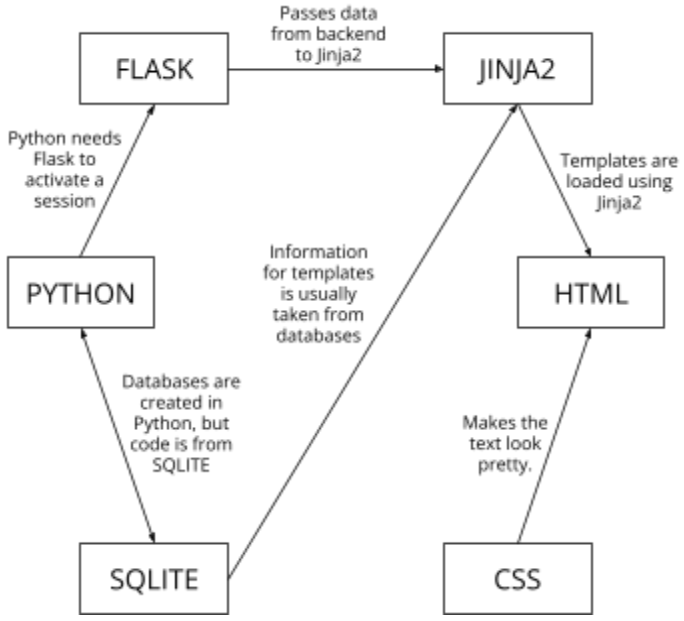
Page Descriptions

- Home (not logged in) - Prompts the user to login if they have an account, or register if they don't.
- Register - Adds new user credentials to the database (create an account).
- Home (logged in) - Displays 20 of the most recent posts, made by all users.
- New Blog Entry - Create a new blog entry.
- My Blog - View your blog (lists blog posts).
- Edit Entry - Edit a pre-existing blog entry.
- Discover - Lists the blogs of all users who have created a blog.
- Logout - Logs out the user from their account.

Component List:

- Python
    - This component is used to create the required database, populate it with information, and to access/edit the inputted data. It is also used to interact with Flask in order to set up the webpages.
- SQLite
    - This component allows for database interaction. It is where the commands for creating/interacting with databases actually come from.
- Flask
    - This component hosts the webpage where the user can interact with it. Also allows for individual sessions so multiple users can be logged in at once.
- Jinja2
    - This component renders templates using inputted information in order to populate the webpage with text.
- HTML
    - This component is essentially the skeleton of the webpage. It is what the user sees.
- CSS
    - This component is used to make the website itself look pretty. However, the site will still work without it (though it will not look as nice).

Component Map



Database Organization:

USERS

User_Id INTEGER PRIMARY KEY ASC	Username TEXT	Password TEXT	Blog_name TEXT	Last post number INTEGER
Makes every account unique.	Allows user to log into the site.		Allows user to change their blog name.	Allows the website to keep track of the number of posts on the blog. Also allows posts to be displayed in reverse chronological order.

POSTS

DATE TEXT	UID INTEGER	POST_NUM INTEGER	POST_TITLE TEXT	POST_ID INTEGER PRIMARY KEY ASC
Stores the date and time the post was made	Stores the UID of the user who made the post	Stores the number post of that user	Stores the title of the post	Stores an id for each post

Blog text is stored in the directory blogs/.
Each user has a directory in blogs/ that contains all of their entries as text files.

Roles:

- Daniel (Project Manager): Managed front-end development (templating & styling) & worked on backend (routing, login & register, listing blog posts).
- Joshua: Coded the basic functionality of the website as well as the file structure for storing blogs.
- Sophie: Did lots of bugtesting, and other miscellaneous things.