Espresso | *Daniel Sooknanan, Sophie Liu, Joshua Kloepfer, Patrick Ging*

SoftDev

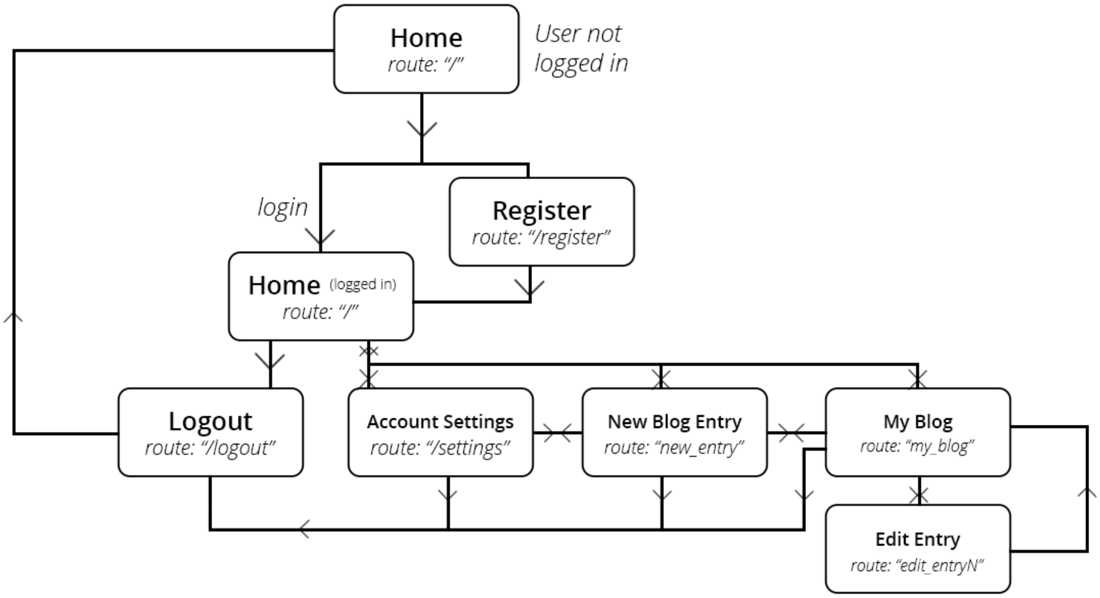
P00 - Expresso Blogging Design Document

10-27-2021

File Structure

```
flask_app
├─ app.py
├─ database.db
├─ templates/
│   ├─ login.html
│   ├─ register.html
│   ├─ home.html
│   ├─ settings.html
│   ├─ blog.html
│   ├─ new_blog.html
│   ├─ new_entry.html
├─ static/
│   ├─ css/
│   │   ├─ style.css
├─ users/
│   ├─ <userID>/
│   │   ├─ 0001.txt
│   │   ├─ 0002.txt
│   │   ├─ n.txt
```

Front-End Sitemap



Page Descriptions

- Home (not logged in) - Prompt the user to login if they have an account, or register if they don’t
- Register - Add new user credentials to database (create an account)
- Home (logged in) - Displays blogs of other users
- Account Settings - Change username & passwords for a user
- New Blog Entry - Create a new blog entry
- My Blog - View your blog (lists blog entries)
- Edit Entry - Edit a pre-existing blog entry
- Logout - Logout the user from their account

Component List:

- Python
  - This component would be used to create the required database, populate it with information, and to access/edit the inputted data.
- 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, site will still work without (though it will not look as nice)

Component Map

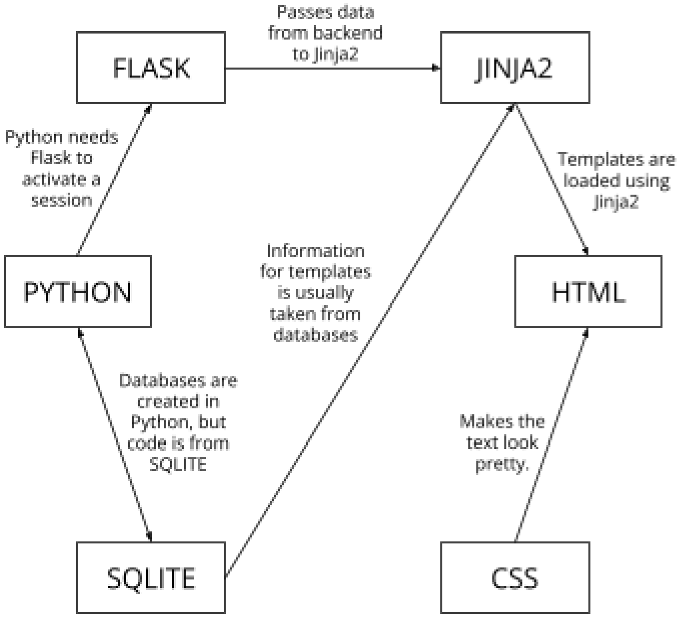

FLASK
JINJA2
PYTHON
HTML
SQLITE
CSS
Passes data from backend to Jinja2
Python needs Flask to activate a session
Templates are loaded using Jinja2
Information for templates is usually taken from databases
Databases are created in Python, but code is from SQLITE
Makes the text look pretty.

Table organization:

User_Id INTEGER PRIMARY KEY ASC	Username TEXT	Password TEXT	Blog_name TEXT	Last post number INTEGER
Makes every account unique	Allows user to change their username	Allows user to change their passcode	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.

Blog text will be stored in the directory blogs/ and each user will have a directory in blogs/ that has all of their entries

Role:

- Sophie will manage the database and how it interacts with with the website
- Daniel will create the templates for each page of the website
- Joshua will manage how the user can view and edit blogs