2 Whites And A Gray – Nada Hameed, Gitae Park, Brianna Tieu

SoftDev

P00: Move Slowly and Fix Things

2022-11-03

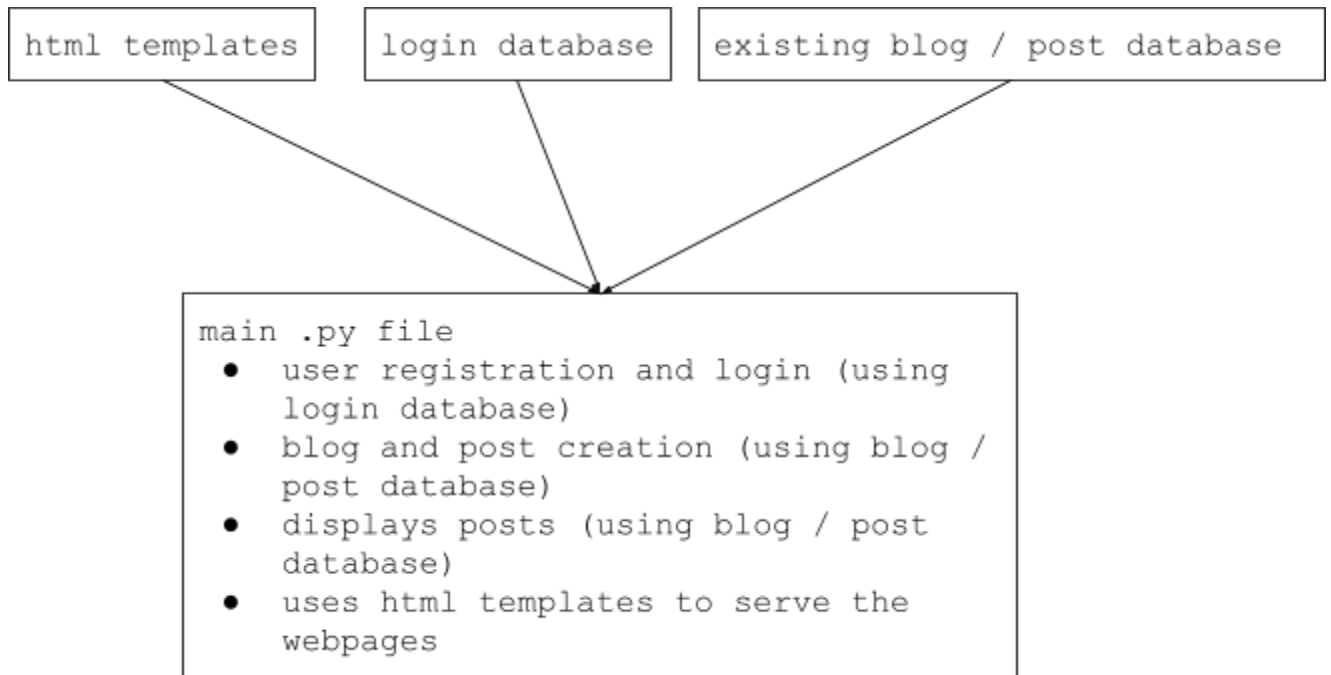time spent: 1 hr

Target Ship Date: 2022-11-28

<u>Program Components:</u>

- html templates for the login / registration page, main blog, blog creation page, blog editing page
- database of logins (usernames and passwords)
    - inputs from registration form submission
    - used to check for any existing users when creating new accounts and correct password input
- databases of user blogs and posts
    - allows user to edit their previous posts
- python file to bring all the components together
    - uses Flask to deploy website
    - allows users to create new accounts and/or log into existing accounts
    - allows users to create new blogs and posts
    - displays other posts

<u>Component Relationships:</u>

- html templates - create the skeleton for the website's pages (what we want users to see on the local page)
- login database - has the data for the user logins, necessary for the user registration and login feature
- blog database - necessary for user editing

- main python file - takes elements from the other files, calls Flask to render the templates to serve a webpage for the user
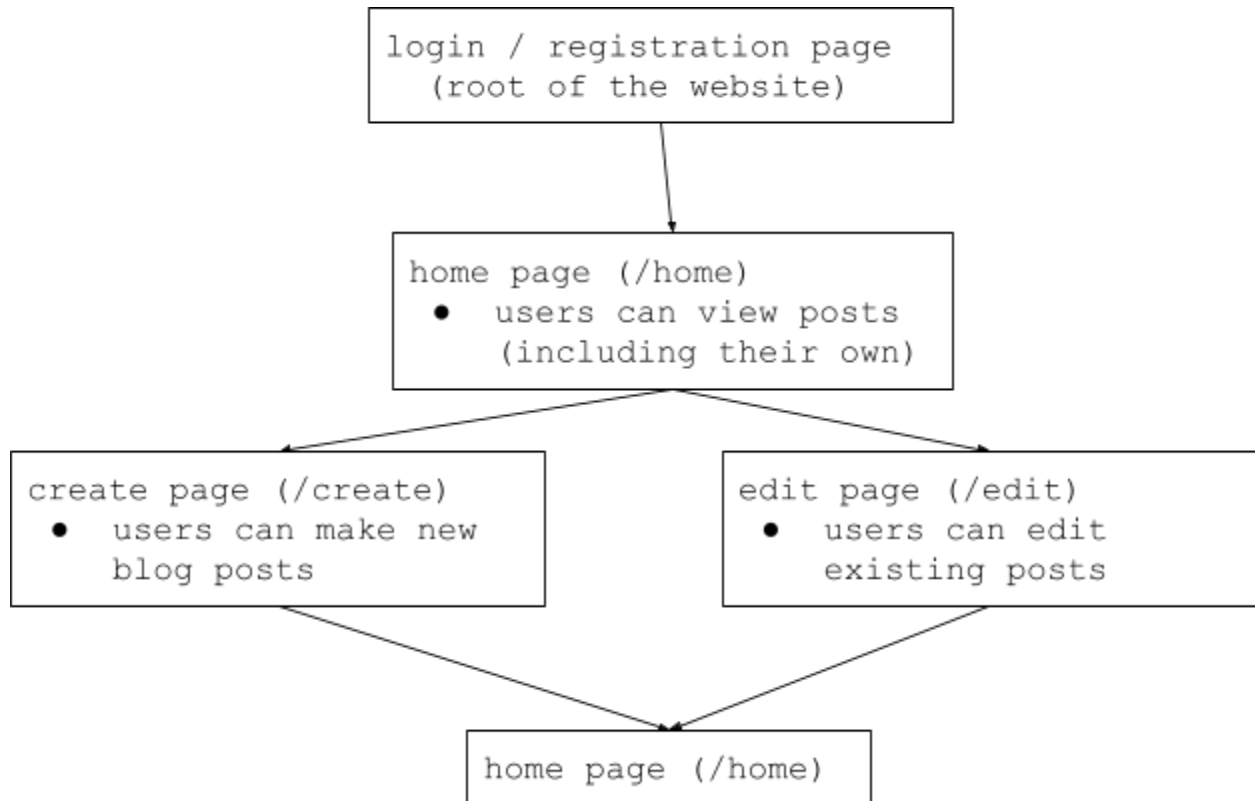   - **Component Map:**

```
┌──────────────────┐  ┌────────────────┐  ┌──────────────────────────────┐
│  html templates  │  │ login database │  │ existing blog / post database │
└──────────────────┘  └────────────────┘  └──────────────────────────────┘

┌────────────────────────────────────────────────────────┐
│  main .py file                                          │
│   •  user registration and login (using                │
│      login database)                                     │
│   •  blog and post creation (using blog /               │
│      post database)                                      │
│   •  displays posts (using blog / post                  │
│      database)                                           │
│   •  uses html templates to serve the                  │
│      webpages                                            │
└────────────────────────────────────────────────────────┘
```

Database Organization:
- Login database
    - Username
    - Password
- Post Database
    - Name
    - Author
    - Body / Content of the Post
    - Time of post publication
- Blog Database
    - Blog ID (counting numbers / ints. assigned by order or time of creation)
    - Username

○ Posts

Site map for the front end:

```
┌─────────────────────────────┐
│ login / registration page   │
│    (root of the website)    │
└─────────────────────────────┘
                │
                ▼
┌─────────────────────────────┐
│ home page (/home)           │
│   ● users can view posts    │
│      (including their own)   │
└─────────────────────────────┘
        ╱              ╲
       ▼                ▼
┌──────────────────┐  ┌──────────────────┐
│ create page      │  │ edit page (/edit)│
│ (/create)        │  │  ● users can edit│
│  ● users can make│  │    existing posts│
│    new blog posts│  │                  │
└──────────────────┘  └──────────────────┘
        ╲              ╱
         ▼            ▼
      ┌─────────────────────┐
      │ home page (/home)   │
      └─────────────────────┘
```

Required Tasks:

- Create html templates (Brianna)
  - login/registration page
  - new blog creation page
  - home blog page (where users can view other blogs)
  - blog editing page
- Login / registration capabilities (Nada)
  - check for existing usernames and/or password requirements
- SQLite database population / creation (Gitae)

- Write main python code to deploy Flask (all)