

Wiki Project Design Document

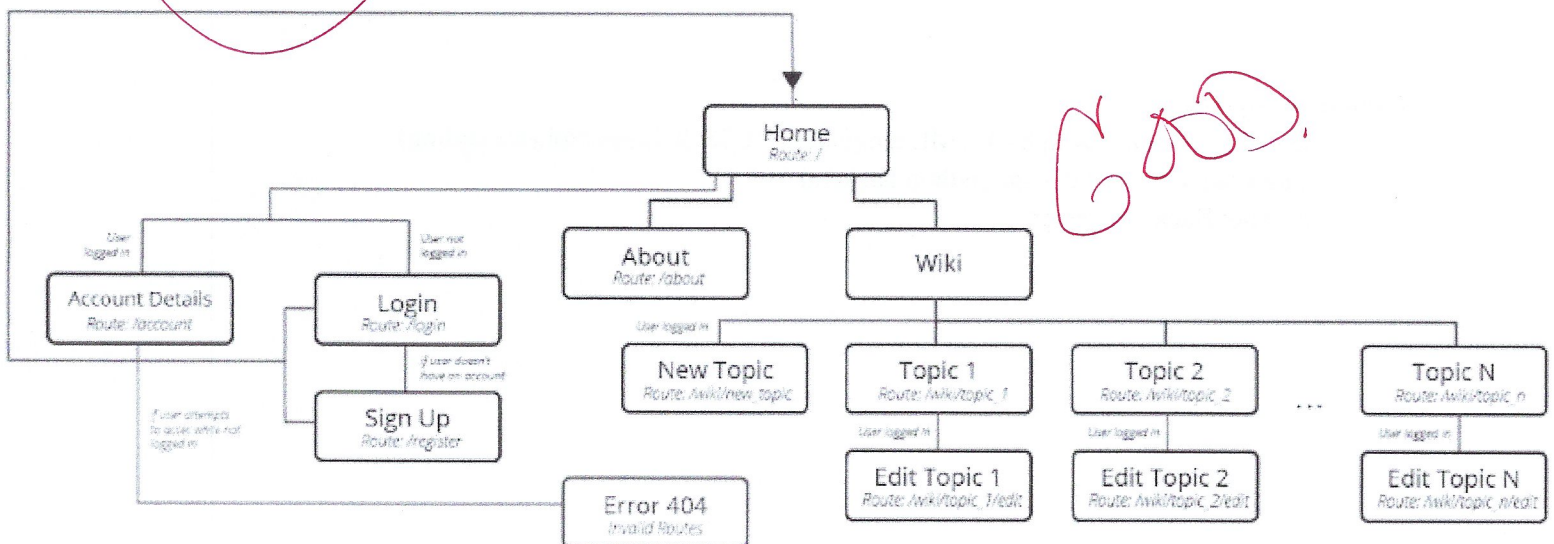
Daniel Soaknanan, Cameron Nelson, Sophie Liu

Std Heading Format?

File Tree:

```
flask_app
├── app.py
├── wiki.db
├── templates/
│   ├── home.html
│   ├── login.html
│   └── topic.html
├── static/
│   ├── css/
│   │   ├── style.css
│   │   ├── javascript/
│   │   └── script.js
│   └── page_info/
│       ├── topic_1.txt
│       ├── topic_2.txt
│       └── topic_n.txt
```

Front-End Sitemap



GOOD

More

Program Component List:

- Python
 - The python app would be used for the creation of the database itself and for inserting data / pulling data from it.
- SQLite
 - We use SQLite to interact with our database. The commands used in the python app would come from this component.
- Jinja
 - Jinja takes the data from the backend for the wiki contents, page title, and other bits and puts them where we want in our webpage.
- Flask
 - Flask serves the actual web pages jinja will be rendering (does network stuff).
 - Session allows for the site to allow a logged in user to edit existing pages and add new ones. If someone attempts to edit/add a page without being logged in, nothing happens.
- CSS
 - CSS makes our site look pretty to the user on the frontend.

Database Organization

- Tables:

1. PAGES

- a. Stores the url by which the page can be accessed and the number pointing to where the contents of the page can be retrieved.
- b. Three columns, ROWID, NAME, and AUTHOR.

2. USERS

- a. Stores user data
- b. Holds username, hashed password, special permissions (mod/admin) + ROWID

Member Assignments

Daniel: Frontend Design - HTML templates + CSS & Javascript (for styling)

Cameron: database design, python backend

Sophie: Backend design