

The Onions - Ishraq Mahid, Lucas Tom-Wong (LTW), Tomas Acuna
SoftDev

P00 -- Half Quick - Planning before prototyping, outlining our objectives.

2021-10-25

Program Components

Front end: Flask (html, Jinja 2); we will create the websites themselves, load them, and then deliver to the user utilizing Flask. Jinja2 and html templates and static pages will be useful for this as well, as Flask will deliver this material.

Database: SQLite (SQ but light) and CSV (separated values, but with commas); from the term “database,” it is implied that these tools will help us create a base for our data. SQLite will be our primary database, as implied by the assignment, and CSV will serve whatever function it may end up being useful for (although it is entirely possible we don’t use CSV at all).

Back end: snake (*ssssssssss*); *the python slithers in the background*. It will allow us to do whatever data processing we may need in conjunction with Flask. We are literally coding in Python, and utilizing the different components within this language, connecting them to be one.

What should directory look like:

App.py

- Transfers information between database and flask

Supporting python files

- Helps clean up app.py, can be used to store methods
- Should have a method for creating templates????

Templates

- See frontend
- Should contain a temporary template as a template for templates

Statics

- Pages that would be beneficial to not change at all and simply load

DB

- Stores important information, that would be inputted into our templates and other important regions
-

What should frontend look like - tentative ideas on current paths:

Homepage

- Search button to go to specific page
- Otherwise use route if you want specific stuff

- Random page button sends you to random page
- Only unchangeable page
- Homepage should be accessible from every page on the Wiki

Wiki Page

- Shows info from database, inputting it utilizing whatever templates provided
- Create changes/edit button, to potentially create a request to add on information
- When changes are made, create a backup of the database and make changes.

Python Code Ideas:

Backend and its interaction with python:

Other Stuff

Different routes would be a different wiki page with different info?????????

Or would there be a variable/key to the info and the path is constant.

How would you create additional routes?

Use a database to store data to a specific route or to a key which can be brought up when you search or random.

Additionally, you can use a database to create copies every time you make an edit to the wiki.

Labor of the Division (tentative):

- Importation of desired information into database to get started with working data (Tomas)
- Setup of the database and methods to access it (Tomas)
- Setup of templates, statics, and Flask app (Ishraq)
- Setup of the UI, management of HTML, and Jinja of the pages (Lucas TW)