


DEM PUMPKINS: Diana Akhmedova, Emily Ortiz, May Qiu
SoftDev
P00 -- Move Slowly and Fix Things
2022-11-10
time spent: 0.7 hrs

Target  Date: 2022-11-15

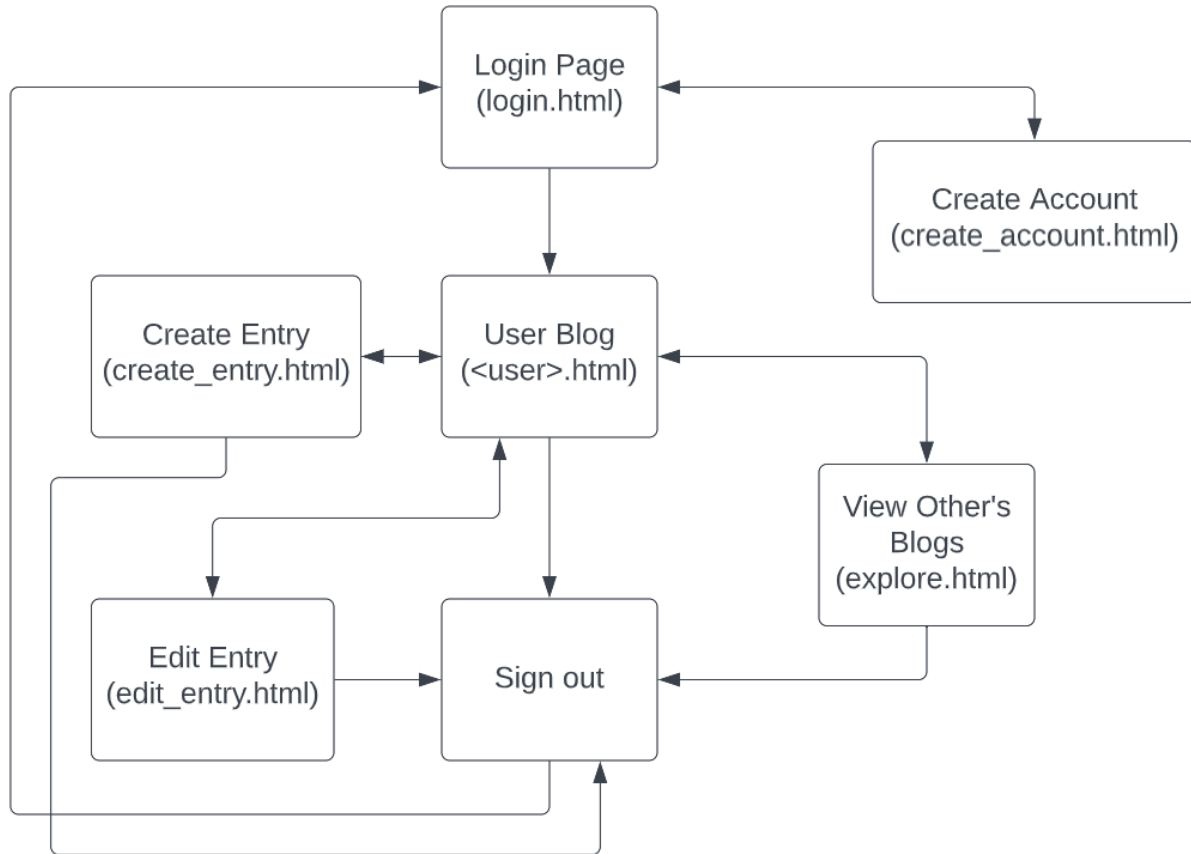
Scenario Two: Your team has been contracted to create a web log hosting site.

- Users will have to register to use the site.
- A logged-in user will be able to
 - Create a new blog
 - Update their blog by adding a new entry
 - View and edit their own past entries
 - View the blogs of other users

Files Needed:

- 7 HTML files
 - login.html
 - create_account.html
 - blog_template.html
 - <user>.html
 - explore.html
 - create_entry.html
 - edit_entry.html
- 3 Python files
 - __init__.py
 - edit_html.py
 - database.py
- SQLite3 database
 - tables.db

Site Map:



login.html:

- Look at K10-12 for reference
- Check database if username and password match
 - Have text on page tell user if/which ones don't match
- Have a button/link to the create account page
- After login, direct to <user>.html

create_account.html:

- Form with username, password, and blog_name input
 - Each username must be unique because the username is how we identify who made changes
- Check database to make sure the username doesn't exist
 - If it exists, prompt user to select new username
 - If it doesn't exist, add username and password to login table

- Make <user>.html file (copy of blog_template.html) and store in database
- blog_name replaces these fields in <user>.html
 - <title>{blog_name}</title>
 - <h1>{blog_name}</h1>
- Direct to <user>.html when done

blog_template.html

- Blog template with basic formatting <!DOCTYPE>, <head>, <body>, etc.
- Replaceable fields:
 - <title>{blog_name}</title>
 - <h1>{blog_name}</h1>

<user>.html:

- 2 buttons at top of page:
 - Create entry button → create_entry.html
 - Edit entry button → edit_entry.html
- Name of blog at top (h1)
- All entries are on the same page, separated by a line like this or other method:

-
- Newest entries at the top
 - Entry contains:
 - Title of entry (h2)
 - Original date published (h3)
 - Date most recently updated (h3)
 - Paragraph of the blog entry
 - Example Blog (bmcc_my_fav.html)

My Blog by bmcc_my_fav

0: First entry!

2022-11-03 01:07 (Last Edited 2022-11-03 01:07)

BMCC is the best school ever.

explore.html:

- Button to go to <user>.html, login.html (sign out)
- List of links to other blogs
- Latest blog created at top of list
- Example formatting
 - Best Ramen Brands by pickleSURPRISE
 - Brings user to pickleSURPRISE.html

create_entry.html/edit_html.py:

Current Method, will change if something better is discovered

- Button to go back to <user>.html and button to submit edits(also directs back to <user>.html after changes)
- Instructions for how to use
, \t, \n, etc.
- Form asking for title and paragraph
- Import html file into edit_html.py as a string
- Split the html file by lines and store each line as an element in an array
 - Ex. html[0] = "<!DOCTYPE>"
- Create strings with proper html syntax, using the user input
 - title_line = "<h>" + user_title_input + "</h>" → "<h>TITLE</h>"
 - paragraph_line = "<p>paragraph content</p>"
- Insert the strings into the array at the proper index
 - Always the same index, to keep the newest entry at the top
- Combine the elements and put them back into the html file

edit_entry.html/edit_html.py:

- Instructions clarifying that if the user wants to include old content in their update, they have to copy and paste it themselves. Whatever is submitted in the form replaces the old entry.
- Form asking for entry id, title, paragraph
 - Tell user if entry id doesn't exist
- Similar method used after create_entry.html is submitted

- Go to a certain index (need to create a function to count this) and replace the strings there for heading and paragraphs
- update the string with latest time edited

database.py

- anything dealing with sqlite3

One Database, Contains Login and Content Tables:

logins		
username	password	blog_name
i_hate_coding123	HeheHAHA1010	
bmcc_my_fav	Myparentshatemecusbmcc	My Blog
pickleSURPRISE	think_simple_think_smart	

The HTML file will hold each individual blog.

content				
username	html_file	entry_id	title	paragraph
bmcc_my_fav	bmcc_my_fav_blog.html	bmcc_my_fav_0	First entry!	BMCC is the best school ever.

Tasks:

- HTML Content:
 - login.html (K19)
 - create_account.html (Diana)
 - blog_template.html (Emily)
 - <user>.html (Everyone)
 - explore.html (May)
 - create_entry.html (Emily)

- `edit_entry.html` (Emily)
- Linking HTML forms to `app.py` (Everyone)
- `__init__.py` communicating with `tables.db` via `database.py` (Everyone)
- User editing of their blog (Everyone)

Extra Elements:

- Deleting an entry
- Allowing user to put lists, images, links, etc. in their entry
- CSS
- Wiki Syntax
- Reset username and password
 - Security questions
 - Password requirements
- Sort the blogs using hottest or latest
- Users will be able to make more than 1 blog
- Users should be able to name their blog
- Commenting
- Editing name of blog