Design Doc for p00 by DECK, pd. 5

↳ *Roster: Cindy Liu, David Lee, Ethan Cheung, Kalimul Kaif*

PROJECT NAME: “LogDECK”
A Flask application utilizing SQLite databases to store and display blogs. Readers will create or log into accounts to read others’ blogs or create and edit their own.

TARGET SHIP DATE: 2025-11-10

Program Components:

- [ ] sqlite3 (backend data storage system)
  - [ ] user_master_list (stores relevant information for each user)
  - [ ] blog_master_list (stores relevant information for each blog)
  - [ ] edit_history (stores edit history of all blogs)
- [ ] Python (application layer)
  - [ ] app.py (runs Python)
- [ ] Flask (web server/delivery framework)
- [ ] html (frontend display)
  - [ ] homepage.html (displays existing “blog” text)
  - [ ] login.html (checks entered login info against database)
  - [ ] register.html (adds new login info to database)
  - [ ] profile.html (displays user history)
  - [ ] logout.html (logs user out of session)
  - [ ] create_blog.html (enables user “blog” creation)
  - [ ] edit_blog.html (enables user “blog” creation)

Our MVP: A functional blogging platform with user login functionalities (including viewable profiles) and a functional logging system to track contributions to each blog.

We are using 3-to-4 technologies: SQLite, Python (to execute scripts to serve HTML web pages), Flask (to serve dynamic HTML pages), and CSS (to style the served HTML pages).

We want our backend data storage system (using SQLite) to store multiple types of information that make up our blogging platform and its functionalities, including but not limited to (for now): **user_master_list**, which is essentially a master list of all users and related information (username, password, profile biographies, date of creation); **blog_master_list**, which is a master list of all the blogs with related information (name, creator, link, content, and date of the most recent edit); and **edit_history**, which is a master list of all edits made to individual blogs (name of blog, blog creator, timestamp).

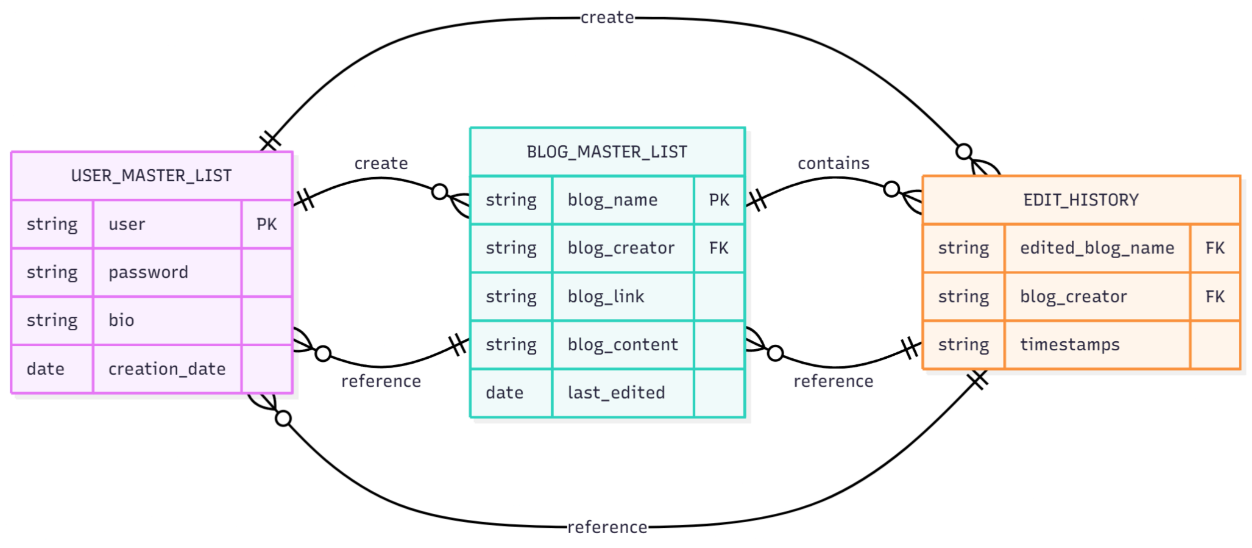
user_master_list is utilized by Flask though a Python script for the pages *login.html* (to retrieve and compare user input login to system stored login info), *register.html* (to see if user already exists, then to append new information), and *profile.html* (which displays the contents of user_master_list).

blog_master_list is used for *homepage.html* (displays all blogs available for viewing), *profile.html* (displays the user's own blog information), *edit_blog.html* (to access and update blog_content & update last_edited), and *create_blog.html* (to access and update blog_content & update last_edited).

edit_history is used for *profile.html* (loop through all contributions, look for matching username to append to a list of that user's edit history), *create_blog.html* (append first contribution after create_blog.html form is submitted), and *edit_blog.html* (append new contributions after edit_blog.html form is submitted).

All pages are dynamic in nature, because they each have user-specific outcomes (changes made to user state) and/or interact with at least one database for the addition/removal of displayed contents on each page.

Component Map:



Database Organization:

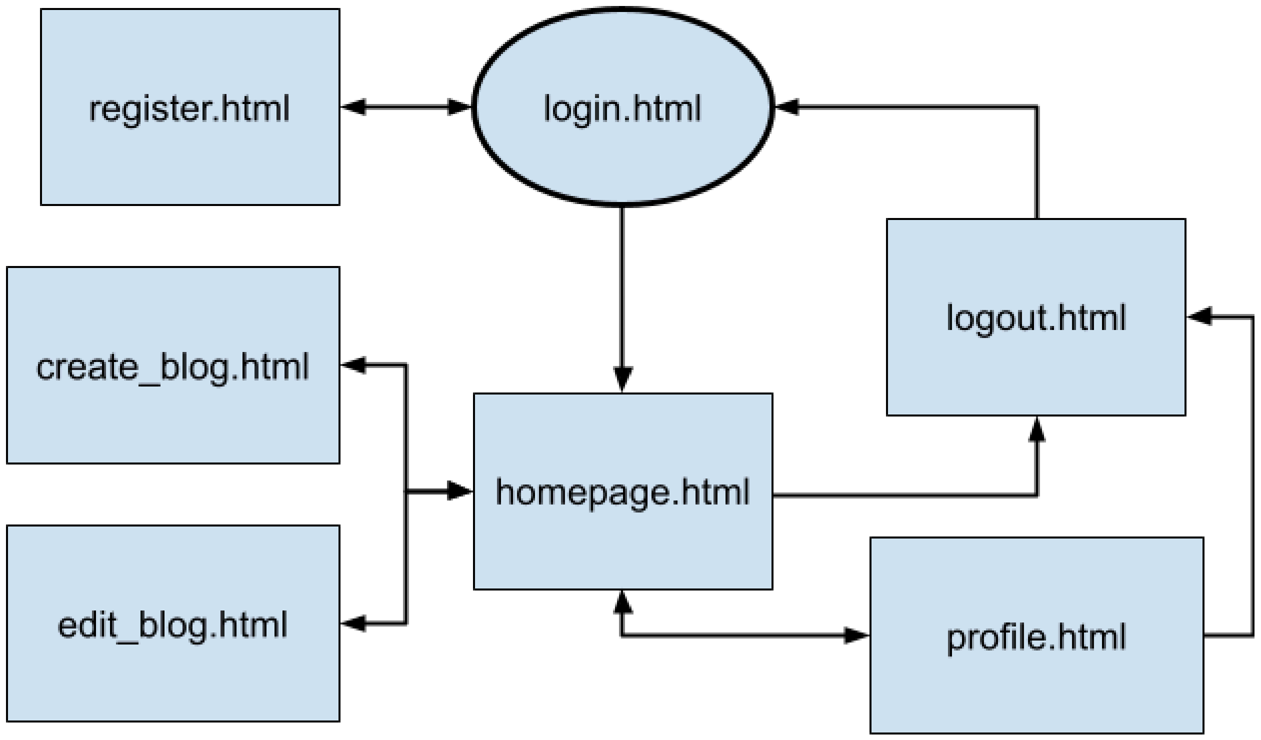
USER_MASTER_LIST			
string	user	PK	unique identifier of blog user
string	password		
string	bio		user customization of profile
date	creation_date		only populated once during acc. creation

BLOG_MASTER_LIST			
string	blog_name	PK	unique blog names
string	blog_creator	FK	references users in user_master_list
string	blog_link		
string	blog_content		
date	last_edited		

Database Organization (cont.):

EDIT_HISTORY			
string	edited_blog_name	FK	reference blog names in blog_master_list
string	blog_creator	FK	references users in user_master_list
string	timestamps		

Site Map:



Breakdown of Tasks:

- ☐ create __init__.py (ethan)
- ☐ create csv files (david)
- ☐ setting up sqlite3 databases in app.py (david)

creating templates + corresponding functions in app.py:

- [ ] homepage.html (cindy)
- [ ] login.html (kalimul)
- [ ] logout.html (cindy)
- [ ] register.html (kalimul)
- [ ] profile.html (david)
- [ ] create_blog.html (ethan)
- [ ] edit_blog.html (ethan)