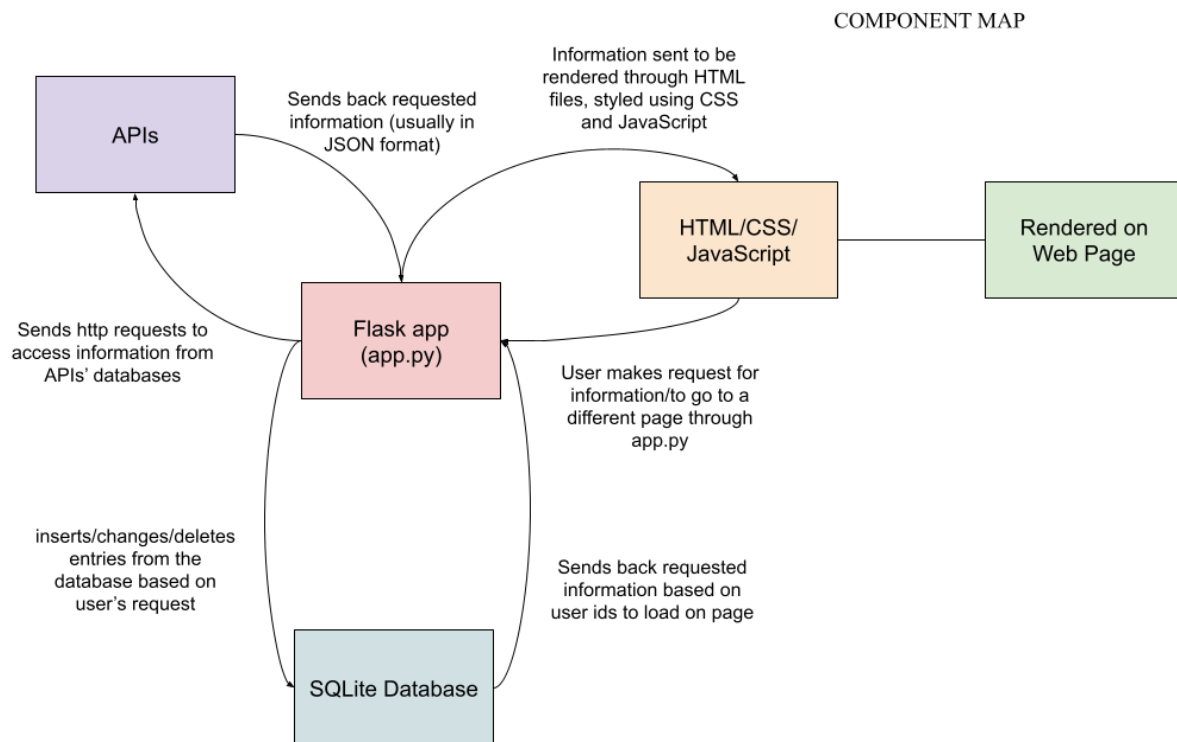


Topic: Study Page/Zone that includes a to-do list, notepad, date, time, a piece of advice, holiday of the day, weather, a cute dog picture.

Program Components and Role of Each

- **SQLite Database:** holds our data and information so we can access it to display on the site
 - keep track of users and user content
 - to-do list content/entries
- **Flask:** the framework for our web-based app, connecting and managing the routes and functions
- **Python:** works with Flask, directs our functions and renders templates
 - Pulls information from SQLite and sends to Flask app to be displayed on page
- **HTML/CSS/Foundation:** styling for our front-end pages
- **JavaScript:** allows users to manipulate/interact with the webpage in real-time
- **RESTful APIs:** retrieve data from external databases and applications, utilize function calls defined within those applications, transmits data back after receiving an http(s) request



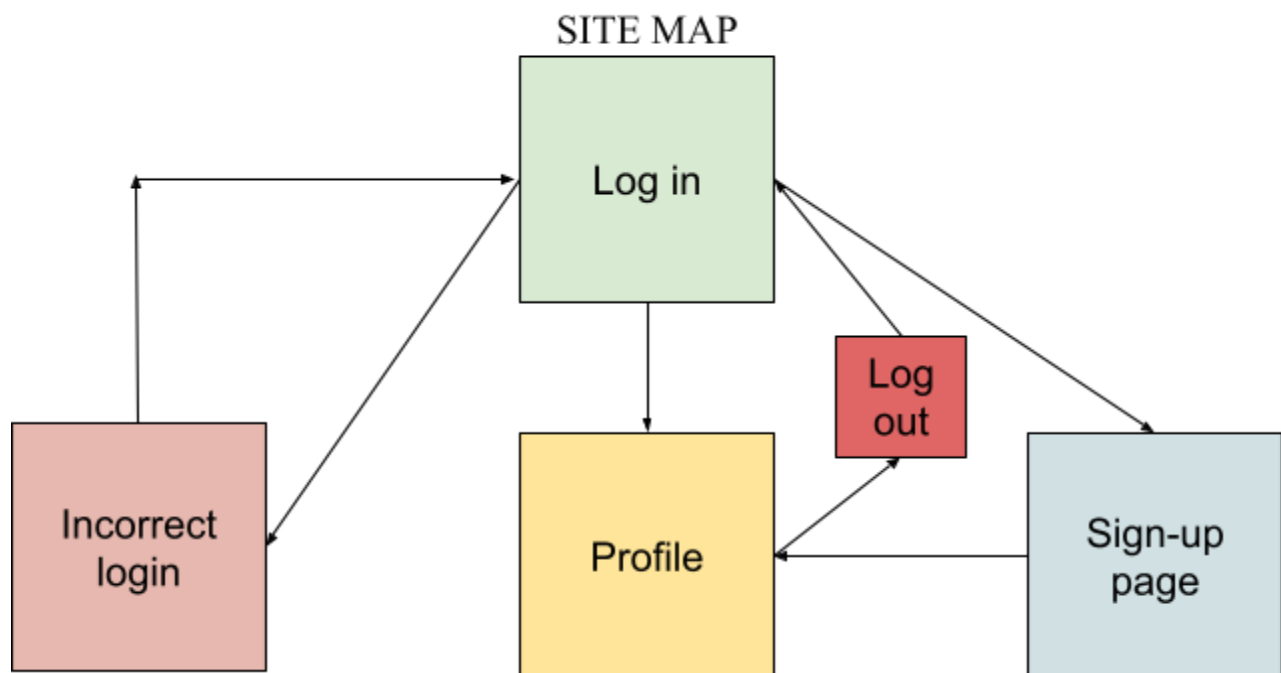
Database Tables

User table

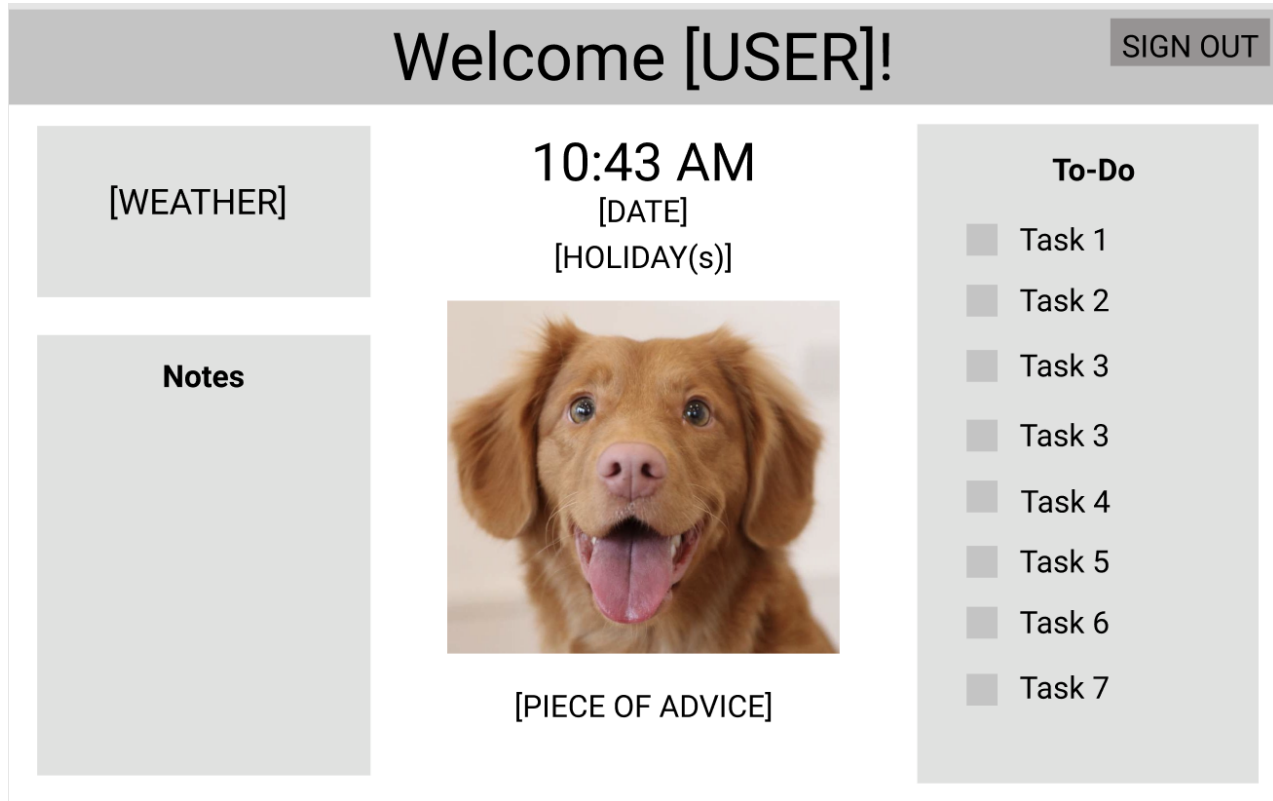
user_id (text, primary key)	username (text, unique)	password (text)
(uuid, ex: 890c238f-e34b-478f-880c-b4f05dfd9d10)	user	123pass

To-Do List table

user_id (text, primary key)	checkbox (binary)	item title (text)	item body (description)
(uuid, ex: 890c238f-e34b-478f-880c-b4f05dfd9d10)	Done? Or not <input type="checkbox"/>	calculus homework	pages 3-4, due tomorrow, notes, etc.



Profile Page:



Task Breakdown

- Login/Logout -- *Ethan*
 - Registering users (setting up database of users and passwords)
 - Set up login page, sign up page, and incorrect login page
 - Check login info with SQLite table of users and passwords
 - Link login page with incorrect login/error page
 - Link login page with sign-up page
 - Link login and sign up pages with profile
 - Link profile with logout page, link logout page with login page
 - Add session to keep track of the logged in user
- Profile/Home Page
 - Create personalized greeting with user's name, ex: "Hello, <person name>" -- *Liam*
 - Show time and date using JavaScript or Python datetime -- *Liam*
 - Display a piece of advice (Advice Slip API) -- *Amelia*
 - Display holiday (Public Holiday API) -- *Amelia*
 - Display today's weather (Open Weather API) -- *Liam*
 - Display dog picture (Dog API) -- *Maddy*

- Display accompanying dog fact if we have time (Dog Facts API)
- Create a to-do list
 - Add entries -- *Ethan*
 - Remove entries -- *Ethan*
 - Check-off entries -- *Amelia*
 - Clear all entries -- *Ethan*
 - Make to-do list visually dynamic using JavaScript -- *Ethan, Liam*
 - When adding items, the user doesn't need to go to another page. → Basically, the user can manipulate the to-do list without leaving their home/profile.
- Create a notepad -- *Ethan*
 - Specifics TBD
- Static Front-End! using Foundation, HTML, CSS -- *Amelia, Maddy*
 - Aesthetics and styling
- Add API cards for Public Holiday, Open Weather, and Dog Facts APIs -- *Amelia, Maddy*

Timeline

- Working Flask app with login and logout functionality -- 4/26
- Personalized greeting, date and time, weather, holiday, advice, dog picture, notepad -- 4/27
- To-do list, front-end work -- 4/28
- Tie up loose ends, API cards -- 4/29