

coffeebuds: Abid Talukder, Faiza Huda, Ziying Jian, Fang Min Chen

SoftDev

P01 - DESIGN DOC

2022-12-05

time spent: 2.0 hrs

Target ship date: T 2022-12-20

Cafe REST

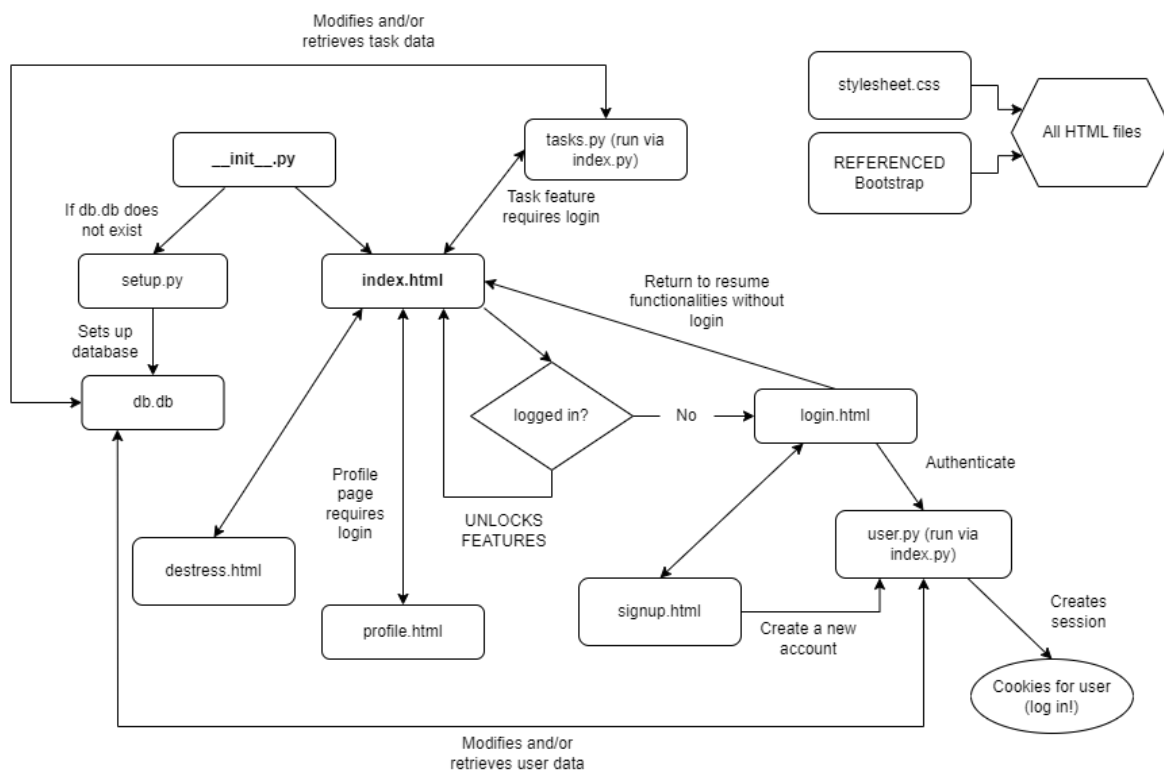
Program Components (role of each specified)

- `__init__.py`
 - serves flask
- `setup.py`
 - setup DB
- `user.py`
 - updates and manages accounts (login, cookies, etc)
- `tasks.py`
 - creates and updates tasks in the to-do list
- `static/`
 - `css/` (if additional frontend css is needed besides referenced FEF)
 - `stylesheet.css`
- `templates/`
 - `signup.html`
 - a form allowing for the user to create an account, providing the username and password stored in a SQLite database (DB)
 - `login.html`
 - a form allowing the user to login to the account, providing the **correct** username and password information
 - `index.html`
 - displays a variety of features for a focused, studying experience; features listed below
 - music
 - background GIF
 - pomodoro
 - to-do list
 - `destress.html`

- displays a variety of features for a relaxing experience; features listed below
 - music
 - background GIF
 - trivia
 - dad jokes
 - cat pics
 - profile.html
 - presents some information about the user
 - displays and tracks the leaderboard, tracked by how many tasks a user has completed
- key/
 - storage for keys; will be added as needed
- js/
 - TBD
- db.db
 - stores all database information

Component Relations

The file `__init__.py` is the main Flask runner, calling `setup` to set up the database with the users and todo tables. Flask renders `index.html` as the main/home page, which in turn can lead to other pages. A user can use the main homepage without a profile with the basic features including the distress tab (rendering `distress.html`). Login components and to-do list components draw upon the database to perform the respective functionality. Once logged in, a session is made to “unlock” features.



Database Organization

users

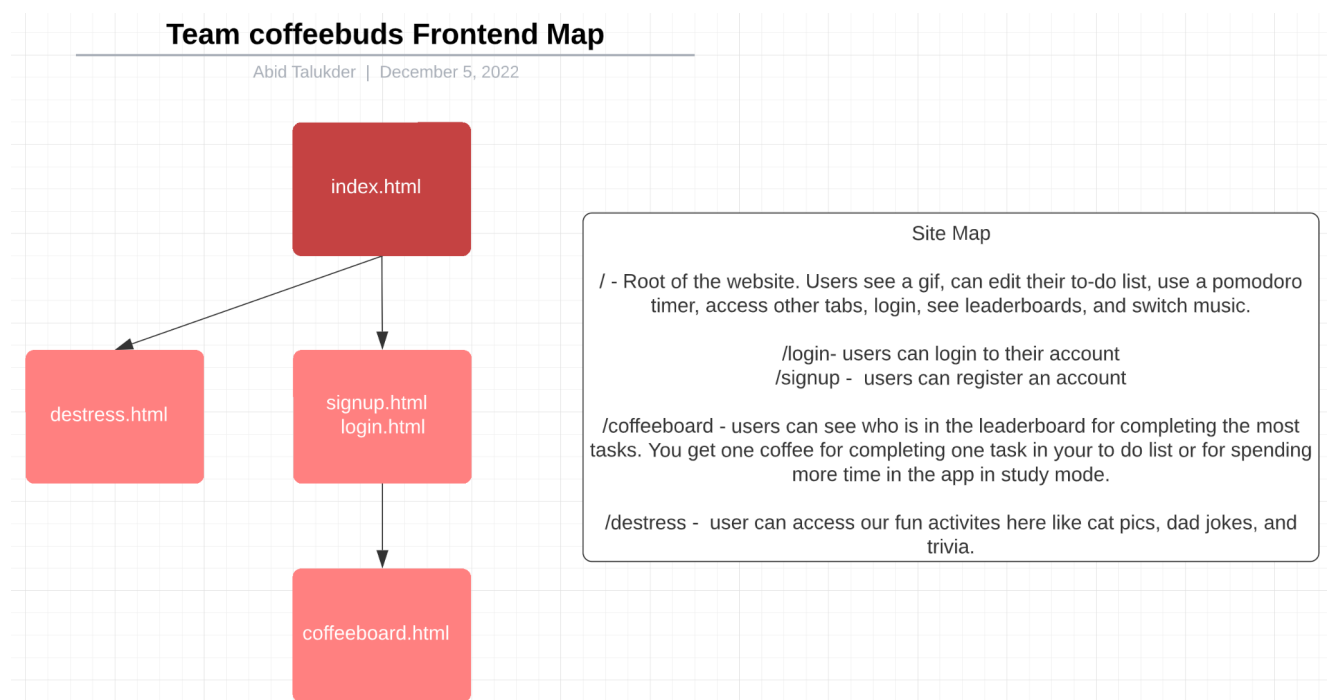
| userID primary int key | username text | password text | coffee int |
|--|--|--|---|
| a unique set of numbers for each user account registered | string of letters and numbers containing no spaces | string of letters and numbers containing no spaces | counts the number of tasks completed as “coffee”s |

todo

| userID int | tasks text |
|--|------------------------------|
| a unique set of numbers for each user account registered | a task that the user creates |

Every time a user completes a task, the corresponding row in the todo db is dropped. Then, the coffee variable in the users table is incremented by one.

Site Map for Frontend



Project Task Breakdown

Fang

- Flask: Routing/linkages and switching tabs without reloading
- Making sure frontend and backend work together

Abid

- Using Bootstrap to build the site
- Building all HTML pages

Ziying

- Constructing a functioning login and registration page

- Finding and incorporating Cat API and Trivia API
- Incorporating YouTube API

Faiza

- Constructing a to-do list db and incorporating its functionality

APIs

- Trivia API
 - <https://the-trivia-api.com/api/questions>
- Giphy API - generates theme-aligning GIFs on the site
 - <https://developers.giphy.com/explorer/>
- Dad Jokes API - cracks random Dad jokes on destress.html
 - TBD
- Cat Pics API - generates random cat pictures/memes on destress.html
 - TBD
- YouTube API - used to get YouTube videos to play
 - <https://developers.google.com/youtube/v3>

Frontend Framework

Bootstrap

- We're using it to beautify our website and to add components like navbars and buttons
- This is because we found the provided components of bootstrap that we can use more aesthetically pleasing and easier to manipulate for our purposes