

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

A cute and comforting study website with a radio, to-do lists, Pomodoro timers, coffee, Dad jokes, and more!

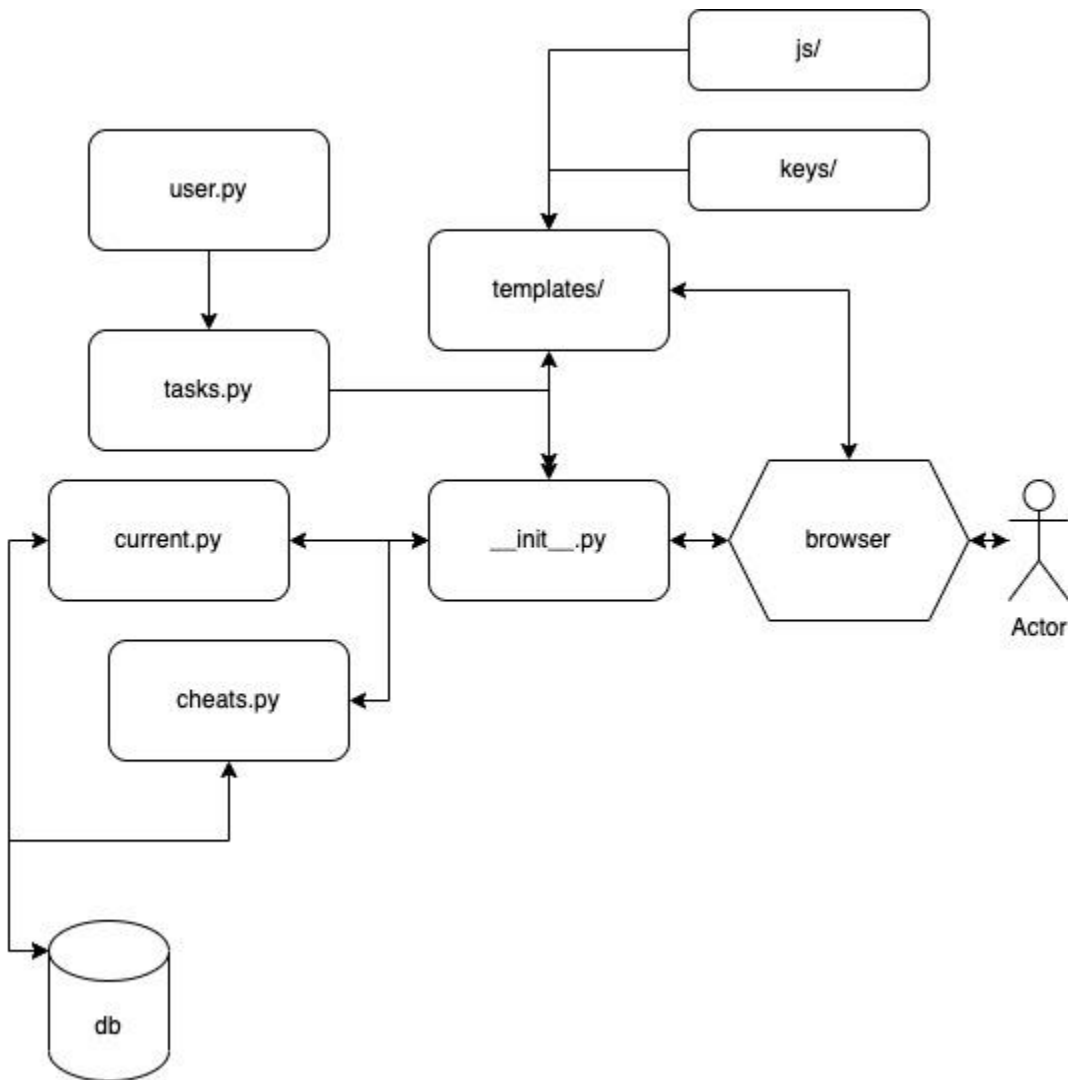
Program Components (role of each specified)

- `__init__.py`
 - serves flask
- `cheats.py`
 - setup DB
- `current.py`
 - updates and manages accounts (login, cookies, etc)
- `user.py`
- `tasks.py`
 - creates and updates tasks in the to-do list
- `static/`
 - `css/` (if additional frontend css is needed besides referenced FEF)
 - `assets/`
 - `js/`
- `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 timer
 - to-do list
- destress.html
 - displays a variety of features for a relaxing experience; features listed below
 - trivia (trivia.html)
 - dad jokes (dadjokes.html)
 - cat pics (catpics.html)
- 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
- static/
 - assets
 - images used
 - js
 - player.js
 - tasks.js
 - timer.js
- coffeebuds.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 “remember the user”.



Database Organization

users

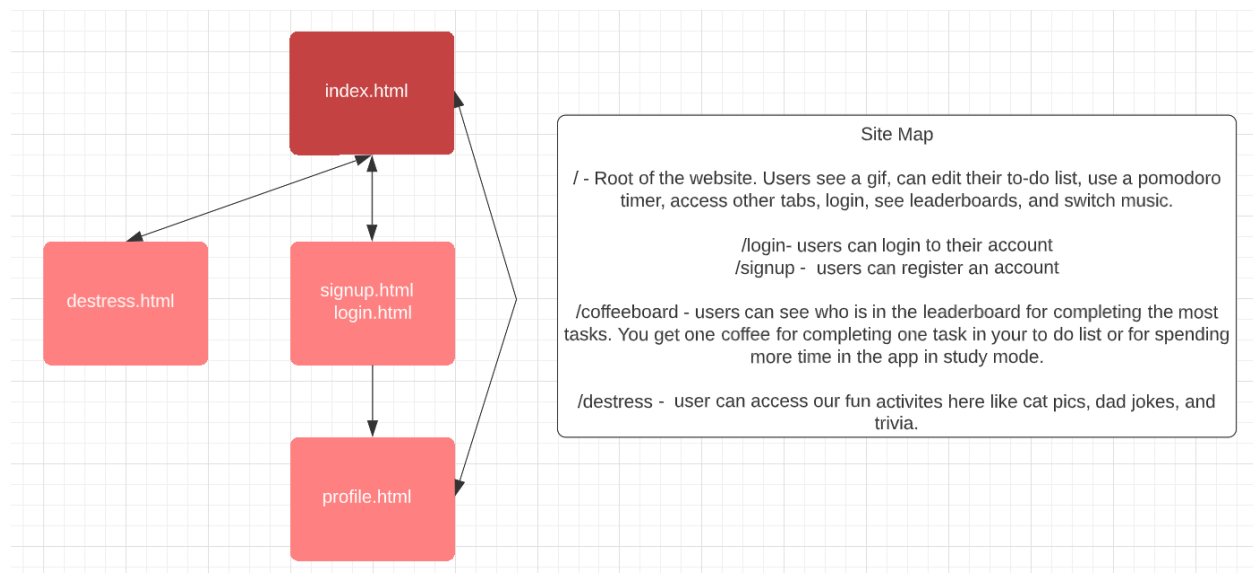
userID primary int key	username text	password text
a unique set of numbers for each user account registered	string of letters and numbers containing no spaces	string of letters and numbers containing no spaces

todo

ID primary int key	userID int	task text
a unique set of numbers for each task	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.

Site Map for Frontend



Project Task Breakdown

Fang Min Chen (PM)

- Flask: Routing/linkages
- Incorporating apis for the destress page
- Building HTML pages

Abid Talukder

- Using Bootstrap to build the site
- Building HTML pages

Faiza Huda

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

Ziying Jian

- Constructing a functioning login and registration page
- Incorporating YouTube embed to create a radio

APIs

- CatAPI
 - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_Cat_API.md
- DadJokes
 - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_DadJokes.md
- Giphy
 - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_Giphy.md
- OpenTriviaDB
 - https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_OpenTriviaDB.md

Frontend Framework

Bootstrap

- We're using it to beautify our website
- This is because we found the provided components of bootstrap that we can use more aesthetically pleasing and easier to manipulate for our purposes
- Features: navbar, custom buttons, stylize text and containers