

BOOTS: SOPHIA C, EMAAN A, JUN JIE L  
DESIGN DOCUMENT

TARGET SHIP DATE: 2025-12-22

---



## APP OVERVIEW:

Home Sweet Home is a website that allows users to hang out and explore their virtual home. Each tab will represent a different room and will contain its own interactive features, operated using APIs, to explore.

## TASKS/ASSIGNMENTS:

### Sophia C. [PM & Frontend Developer]:

- maintain devlog
- create html templates using jinja:
  - login.html
  - logout.html
  - register.html
  - bedroom.html
  - kitchen.html
  - livingRoom.html
  - Frontlawn.html (home)
  - settings.html
- write all css
- ensure navigation links work between pages

### Emaan A. [Middleware]:

- initialize flask app
  - configure sessions
  - handle APIs
- user registration/login/logout
- Set up routes for html pages
  - /home, /bedroom, /kitchen, /livingroom, /settings, /login, /logout, /create
- requirements.txt

### Jun Jie L. [Backend]:

- database organization (tables)
- write SQLite queries for authorization/login/logout

## APIS:

- [411 on FreeRecipeAPI.md](#) (Recipe Book Kitchen)
- [411 on JokeAPI.md](#) (Joke Book on Shelf in Bedroom)
- [411 on MoviesAPI.md](#) (Movie Shelf in living room)
- [411 on HolidaysAPI.md](#) (Calendar in Kitchen Wall)
- [411 on OpenWeather.md](#) (In the sky in Front lawn)
- [411 on TheCatAPI.md](#) (cat album in living room)
- [411 on The Metropolitan Museum of Art Collection API.md](#) (Museum Book on Shelf in Bedroom)

## FRONTEND FRAMEWORK:

We are using Foundations Framebook for our Frontend. The features that will be included will be Buttons, Dropdowns, Slider, Switch, Menu, Dropdown Menus, Accordion Menus, a Top Bar, Section Alignment, Coloring, Icons, and other Containers like Cards, Images, and Reveal. We used Foundations because its features suited our styling the best.

## DATABASE ORGANIZATION (TABLES):

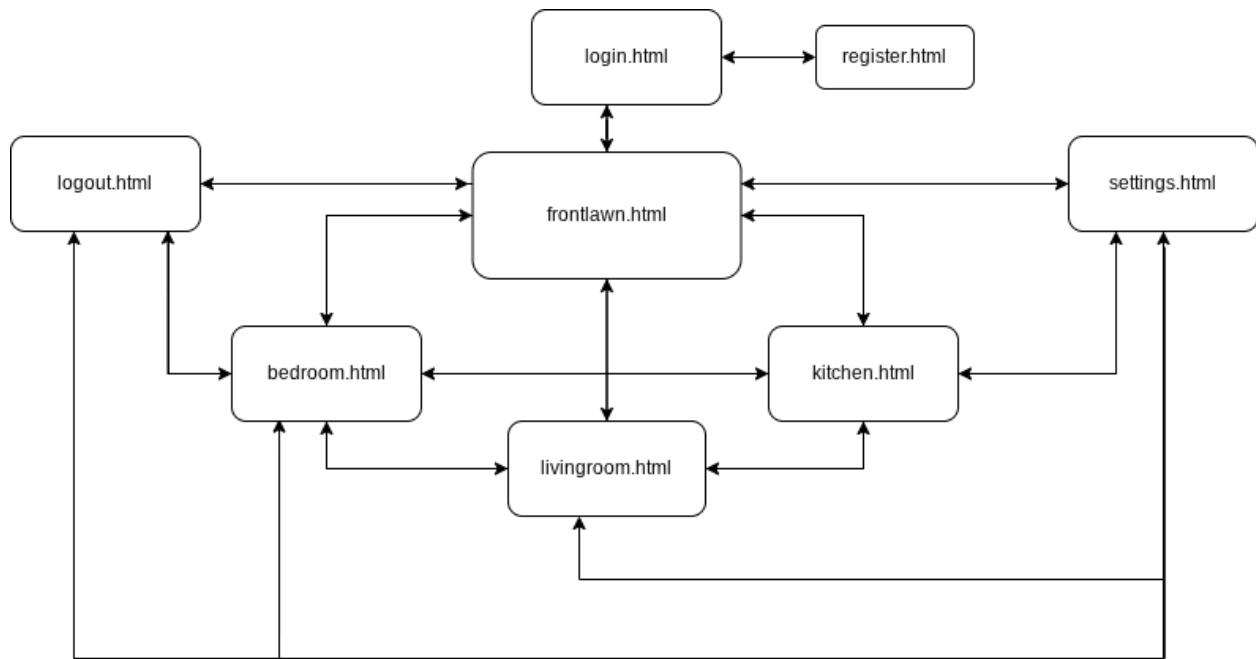
USER		
TEXT	user_id	PK
TEXT	password_hash	
TEXT	session_key	



One user has one API\_cache  
for each API we have.  
Each API\_Cache has one  
user.

API_Cache		
TEXT	id	PK
TEXT	user_id	FK
TEXT	api_name	
TEXT	api_id	
TEXT	types_json	
TEXT	api_response	

## SITE MAP



## COMPONENT MAP:

`__init__.py`: configures flask app, makes routes, handles authorization/login, builds database, handles all api calls  
frontend framework/css styles html templates  
`jinja2` reads database and turns data to webpage  
`data.db`: SQLite3 database

