- The project must demonstrate knowledge of the concepts covered in class. To that end:
    - You must use python/Flask
    - You must use a front-end framework (probably bootstrap or foundation)
    - You must provide custom CSS
    - You must have your own javascript code that does something meaningful for your project
        - Include at least two examples of:
            - making ajax calls
            - using canvas or svg
            - using data visualization (most likely d3)
    - You must use at least one of the following:
        - databases
        - APIs
            - If you are using an API, research it to make sure it is stable.

- Create a PDF named DESIGN.pdf in your repository (See other projects for detailed Design Document guidelines).
    - Include a component map, site map and database schema (if applicable), as well as any necessary supporting documentation.
    - Include a component description
    - Divide the tasks among your group members. Include a Project Manager.
    - Include a tentative timeline
    - Include a style guide

FINAL PROJECT: Organizer

Pd 8 | 3Y1B: Md Abedin, Karen Chen, Jenny Gao, Yiduo Ke

DESCRIPTION

The goal of this website is to help users stay organized. Users can add tasks to their to-do list and simultaneously view tasks in list form and calendar form (on the same page). The leaderboard will create incentive to complete tasks in a timely manner.

PROGRAM COMPONENTS

FRONT END (Bootstrap)

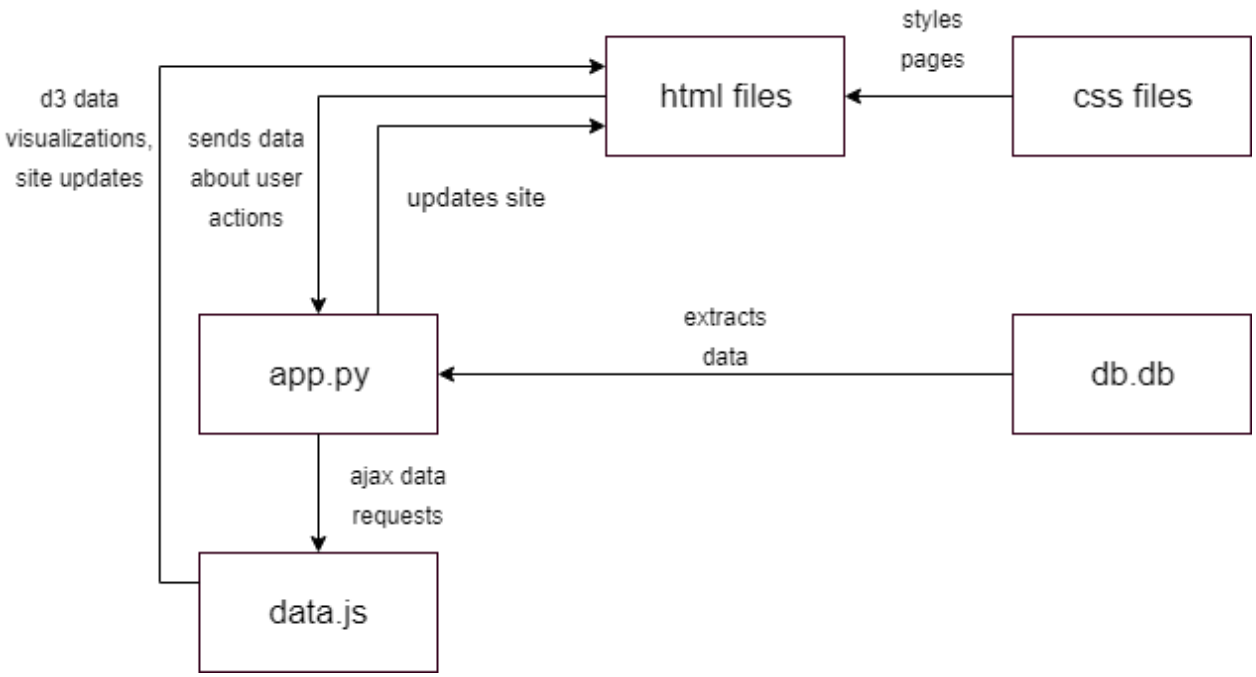
- ***index.html***
    a) Template for users' main page (option to create account/log in)
- ***create.html***
    a) Users creates account
- ***login.html***
    a) Users log in

- ***leaderboard.html***
    a) Displays ranks of users based on how quickly they finish their tasks
- ***Home.html***
    a) Shows to do list and calendar, as well as completed tasks
- ***style.css***
    a) Maintains consistent style throughout site
- ***Bootstrap.min.css***
    a) Makes site responsive

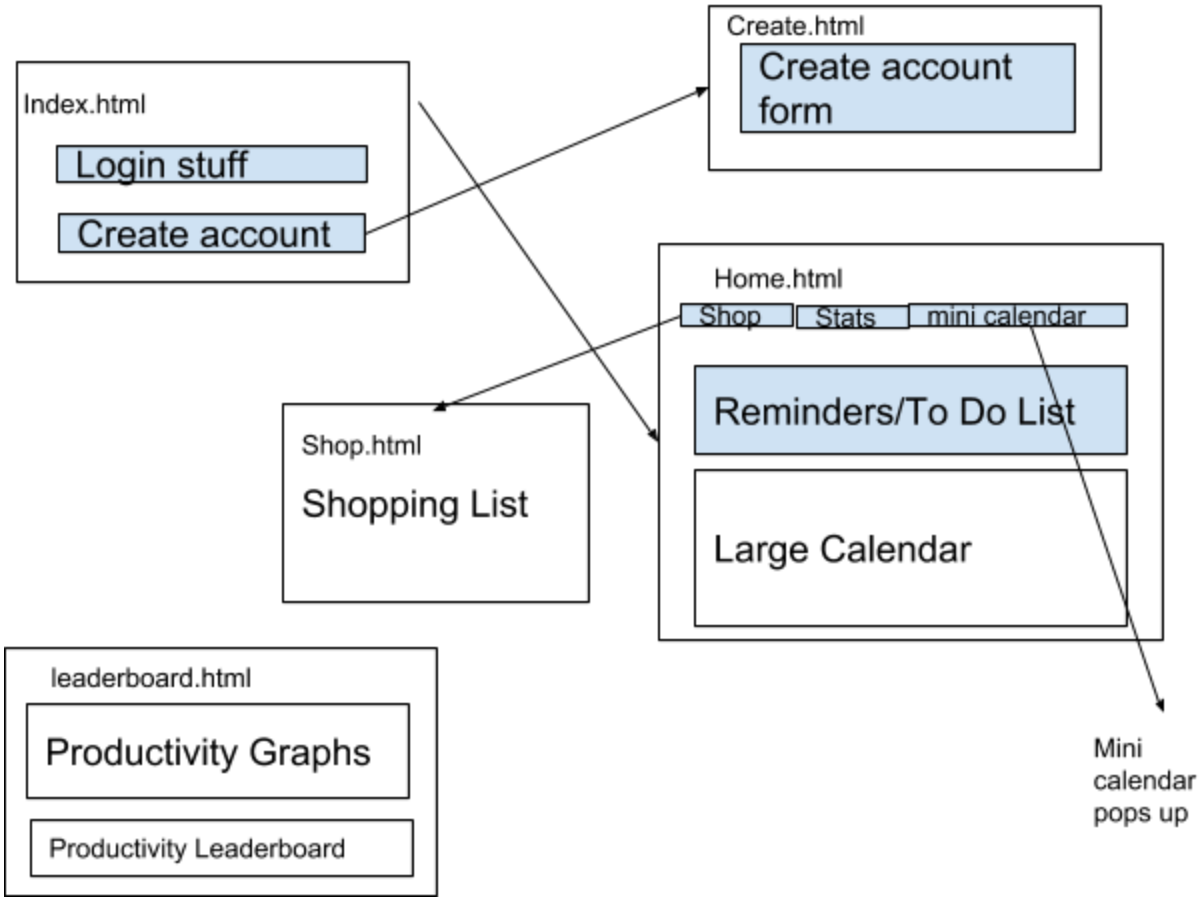
BACK END

- ***app.py***
    b) Serves website, handles url requests/form responses, updates site
- ***db.db***
    a) Stores all ongoing and completed tasks of all users
    b) Stores user task completion stats:
        i) How long they expected/how long they took for a task
        ii) Time started and time ended
- ***data.js***
    a) Acts as a bridge between the UI and the database of user info
    b) Performs d3.js data visualization

COMPONENT MAP



SITEMAP



ROUTES

- **Home Page (“/”)** - *renders “home.html”*
    - Shows to do list and calendar if logged in, otherwise redirects to login
- **Create Account (“/account/create”)**
    - asks for username and password, will redirect to login on success
- **Shop (“/shop”)** - renders *“shop.html”*
    - Allows users to add items to shopping list
- **Leaderboard (“/leaderboard”)** - renders “*leaderboard.html”*
    - Displays ranks of users based on how quickly they finish their tasks
- **Login (“/account/login”) -** *renders “login.html”*
    - Login or create an account
- **Logout (“/account/logout”)** - *redirects to “/”*
    - Logs out account

DATABASE SCHEMA

TABLE: users

Column Name	Type	Example
username	STRING PRIMARY KEY	‘drawbot5000’
password	STRING	‘as23da’
score	INTEGER	100

TASK ASSIGNMENT

Project Manager: Yiduo Ke

Specific Tasks: (these are tentative; you can change them if you want)

- Front end: Karen Chen
    - making things look aesthetic (e.g. how tasks look on to-do list/calendar)
- Programming the to-do list and calendars:
    - Md Abedin: database (handling user info like what tasks and when), calendar display
    - Jenny Gao: HTML files/Flask, to-do list, leaderboard/d3 visualization
- Incorporating APIs to help users shop: Yiduo Ke

TIMELINE

Week of May 13/20: To-do list, calendar, shopping system
Week of May 27: Timer
Week of June 4: Aesthetics