

P02: Makers Makin' It, Act I

Project: p02 ChoreRulean

Team: cerulean

Roster: Natalie Keiger (PM), Michelle Chen, Maya Berchin

PROJECT DESCRIPTION:

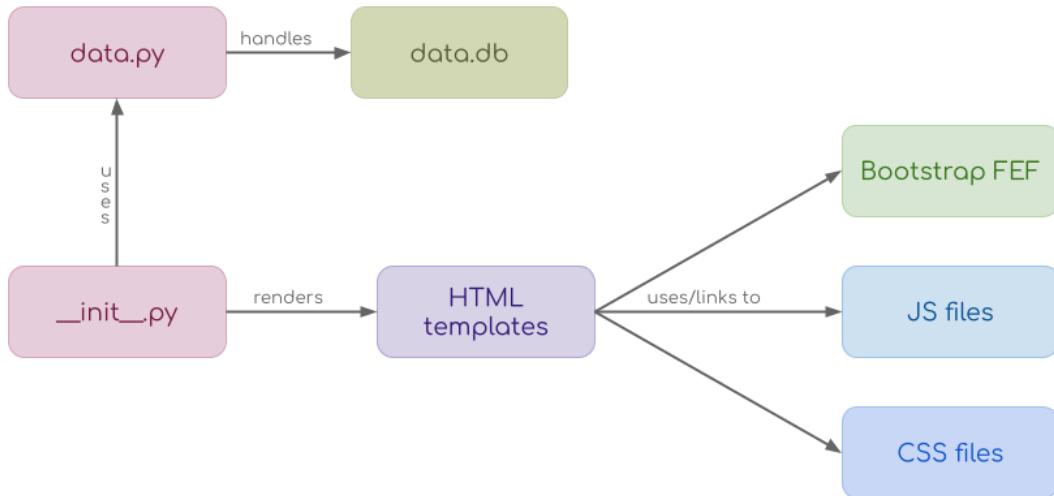
Task manager cerulean™ to create, delete, and “accomplish” tasks. Users can login and save their tasks as well as adding other users to tasks to share them. Tasks can be sorted by categories and use a calendar for organization.

PROGRAM COMPONENTS + EXPLANATION:

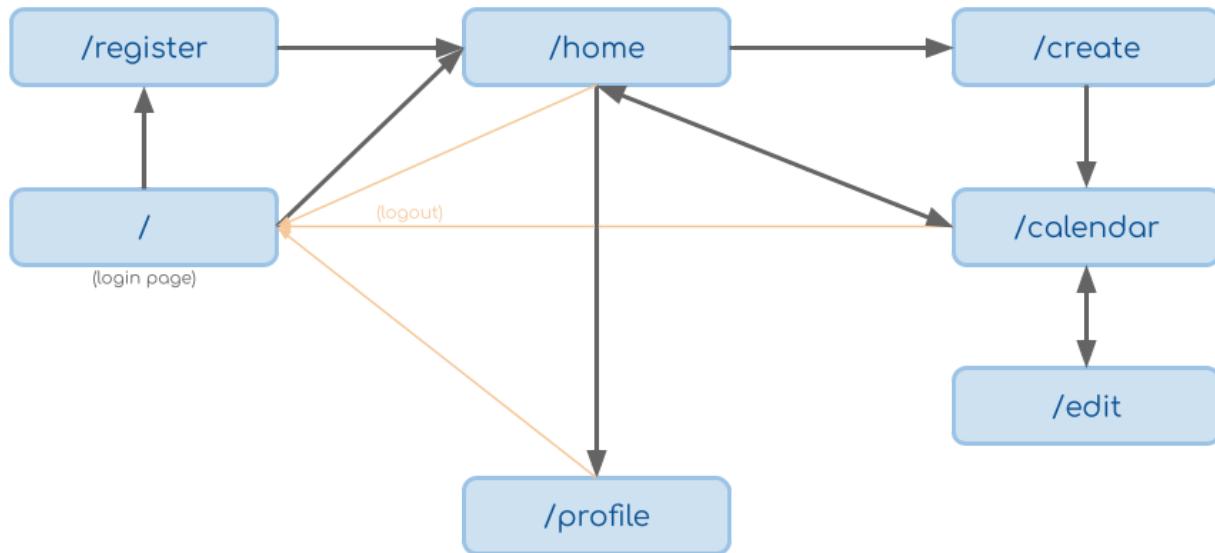
- Python files
 - `__init__.py`: the main file; serves app
 - `data.py`: handles SQLite3 database
- Templates
 - `login.html`: the user will be directed onto the login page first. They will be redirected to the homepage once they are logged in (automatic if they are already logged in). If they don't have an account, they can be redirected to register
 - `register.html`: the user will be able to register. They will be redirected to the homepage once they do this
 - `home.html`: the homepage, which will show the user's upcoming tasks, tasks they've been invited to, and tasks other users are up to which are visible to them (determined by a task's visibility: public, friends only, private). They can also join a task without being invited provided the task has permissions that allow them to do so (determined by a task's `join_perms`: `invite_only`, `friends_only`, or `everyone`)
 - `create.html`: where the user can create tasks, set things like the category (chore, HW, etc), task

- description, deadline, permissions, and visibility, and invite other users
- `calendar.html`: displays a calendar with the user's tasks by date
- `edit.html`: lets the user edit a task by its ID provided they are the task owner. They can change the deadline, assign other users, delete the task, edit the title, edit the visibility or join permissions, and edit the description. If the user is not the owner of the task, they can only leave the task
- `profile.html`: lets a user edit their profile picture, password, and other user settings, like who can invite them to join tasks
- CSS files
 - `calendar.css`
 - Possibly others as needed, probably using bootstrap for most
- JavaScript files
 - `home.js`: connected to the home page, allows for drag-and-drop of tasks into "completed" or "incomplete"
 - `calendar.js`: connected to calendar HTML, handles dates and setup for the calendar page
- Database (stored in data.db)
 - `users`: includes username, ID, password, pfp (out of a select few), and who can invite them to tasks
 - `tasks`: includes task name, ID, description, deadline, status, type, and users assigned to it
- Front-end Framework
 - `bootstrap`: 2/3 devos have the most experience with this FEF, which is why we chose it

COMPONENT MAP :



FRONT END DIAGRAM:



DATABASE ORGANIZATION:

users

Variable Type	Variable Name	Variable Attribute(s)
string	username	PK NOT NULL
string	password	NOT NULL
string	friends	
string	friend_reqs	
string	pfp	
string	invite_perms	
string	pending_invites	

tasks

Variable Type	Variable Name	Variable Attribute(s)
string	name	NOT NULL
string	id	PK NOT NULL
string	description	
string	deadline	
string	status	NOT NULL
string	category	NOT NULL
string	users	NOT NULL
string	visibility	
string	join_perms	
string	owner	NOT NULL

TASK DELEGATION:

Task	Devo (s)	Deadline	Status
DD: Description	Natalie	1/12m	done ▾
DD: DB organization	Maya	1/12m	done ▾
DD: Component map	Maya	1/12m	done ▾
DD: Frontend map	Michelle	1/12m	done ▾
data.py: setup, tables, functions	Maya	1/13t	IP ▾
Implement login, logout (backend)	Maya	1/13t	done ▾
login.html, register.html (frontend)	Michelle	1/13t	done ▾
Pages linked according to front-end diagram, navbar	Natalie	1/13t	IP ▾
README.md	Natalie	1/13t	done ▾
requirements.txt	Michelle	1/13t	NS ▾
create.html, calendar.html (frontend)	Michelle	1/14w	NS ▾
Create tasks, see tasks on calendar (backend)	Natalie	1/14w	NS ▾
profile (frontend+backend)	Maya	1/14w	NS ▾
home.html, edit.html (frontend)	Michelle	1/15r	NS ▾
Homepage (backend)	Maya	1/15r	NS ▾
Homepage JS	Natalie	1/15r	NS ▾
Finish all styling	Michelle	1/16f	NS ▾
Calendar JS	Natalie	1/16f	NS ▾
Demo vid	ALL	1/16f	NS ▾