pyCalPlan

Project Description

pyCal is an innovative and reimagined idea of a calendar planner. pyCal allows users to not only track their scheduled events throughout days, weeks, months, and years, but also keeps track of to-do items. Up-comming events and to-do items are automatically ordered based on due date, availability, and priority into a handy sidebar that allows users to see what they need to be doing next.

Competitive Analysis

Similar projects have appeared as past 112 term projects as many variations of to-do planners and general week calendars. Their features focus on the presentation of just to-do items and receiving some type of reward by completing the items, or if its a general week calendar, it only keeps track of events during the current week. However, my project will focus on keeping track of all events, no matter what day, week, month, or year. The project will also be able to suggest the order that the to-do items must be completed given by the parameters such as time needed to complete task, availability and priority. The goal for this project is to be able to fully replace the user's current to-do planner and calendar applications.

Structural Plan

pyCal.py - this is the python file that will contain functions to display and keep track of the week view of the events and mini calendar view by month.
pyPlanner.py - this is the python file that will contain functions to display and keep track of the to-do items along with the scheduled events from pyCal. The main function this file will house is the function that will automatically order the to-do items.
event_data.json - this is a json file that will keep track of all event data
planner_data.json - this is a json file that will keep track of all to-do data

Algorithmic Planning

- The calendar dates will be drawn onto the canvas by looping through real calendar year dates
- The user's events will be stored as a dictionary with the keys being the dates and values being a list of names, times, locations, etc of the event. This will be looped through and displayed on the calendar by whether or not the current view of the calendar is set within the bounds of the current week, month, and year
- The user's to-do items will be stored as a list of name, priority, time needed to complete, etc of the task. This will be looped through and displayed on the sidebar in order of suggested completion. The order of the to-do items will follow a backtracking algorithm that will place weights on current availability, deadline, priority, and time needed.

Timeline Plan

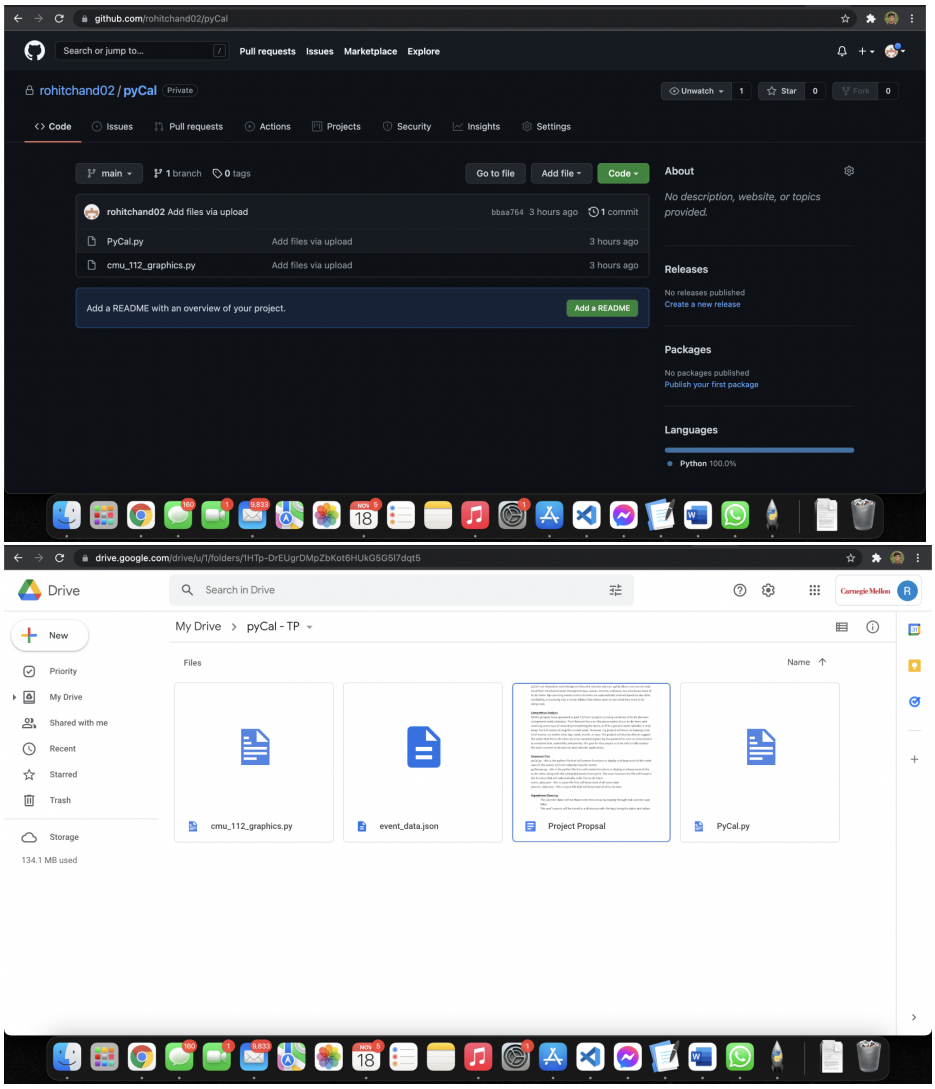
TP1 - Have a simple GUI that allows for users to add/delete calendar events that will be displayed and stored, have ability to sync with real year calendar

TP2 - Have ability to add to-do items, the items are ordered with backtracking algorithm

TP3 - Adding better GUI experience for users - smooth scrolling, ability to open zoom links and download pdfs for classes, ability to add locations and open directions to location, ability to set outdoor events that will show how the weather will be like, etc

Version Control Plan

Daily backup on GitHub and Google Drive



Module List

calendar, json, and os

TP2 Update

One part of the design that changed in TP2 is how the data for events and tasks are being stored. While I opted to store events as tuples inside a dictionary initially, json doesn't have the capability to store tuples. So instead, I am now pickling the tuples in order to store the events and tasks across reruns.

Additionally, TP2 features the addition of basic features. These features include adding/deleting calendar events, adding/completing to-do tasks, and automatically sorting the to-do tasks according to when it is due and priority so the user is shown what order to complete them.