Team 07
Project Reflection

Front End Tools

jQuery - jQuery allowed us to simplify the scripting of our front end system which was especially useful for our AJAX requests to our API. Not having to deal with vanilla async functions was very beneficial. On the other hand, jQuery abstracted away what was going on behind the scenes. Because of this, we didn't learn as much as we could have learned if we used plain vanilla JS or another framework.

Bootstrap - Bootstrap was very beneficial. It allowed us to quickly prototype our ideas and build our web pages using its built in components. It also made it easier for our front end to be responsive and work on a variety of devices. The downside to using Bootstrap is that it was difficult to customize and in the end gave our site more generic look.

Back End Tools

Flask - Using Flask was ultimately a good choice because of its ease of setup. Running our Flask app from one Python file allowed us to have a more unified place to code site operations. One of the big disadvantages was that in order to add a page to the site or access the database, the front-end team would have to wait for the back-end team to code the functionality into the Flask app.

Raspberry Pi + Nginx - We decided early on to host our own site locally, which ultimately gave us control over every tool we used, while being practically free. Without being tied to some external service, this allowed our team and our app to remain very self-contained.

MySQL - We initially had some problems using sql procedures, but procedures wound up being very helpful for us being able to split the load in automating tasks between Flask and MySQL to keep our code modular while avoiding bloat.

Productivity Tools

Slack - Slack was an essential tool for the completion of our project. None of us had used Slack before but appreciated the simplistic design. We found that the interface was very effective and user friendly. By creating subgroups for the front end and back end groups, we could effectively collaborate within the different groups without creating a cluttered workspace for everyone else. We found there to be some issues with communication via Slack as sometimes team members were logged out and didn't realize it and then missed out on a few days of communication.

Google Docs - Google Docs proved to be a very handy tool for pushing out presentation slides, documents, and collaborative group work. We found that when having the whole team editing a Google Doc, it would allow for maximum coverage and efficiency when completing milestones. There was never any issues with Google Docs as everything is automatically backed up, and easily accessible.

Github - By utilizing Github, our group found it very easy to keep up-to-date with our projects progress. The pull and push features of Github proved to be very beneficial, especially when working collaboratively with the front end and back end sides. Github however, did have a small learning curve as to all the commands and how to pull, commit, and push changes, however, it was a very powerful tool and we all enjoyed it.

Trello - Trello was an essential progress manager for our team and proved to be a very beneficial tool. A big issue for most groups was time management and keeping up to speed and on task. Trello allowed us to essentially keep on track while organizing what has been done/needs to be done. Trello's interface however was lacking, we found that the comment system to be obscure and hidden. Despite the miniscule flaws, it was a very quick and convenient tool.

Project Report

- Our original goals and objectives
  - Originally we had planned to create an online app split into 3 primary sections as "To-do", "Notes", and "Events".
  - The events section would have the ability to register an event and assign multiple invitees to the event, perhaps even allowing to attach notes
  - We intended to have the ability to uploaded photos and links as well
- Project and its success as it stands right now
  - As of right now the functionality for creating notes is working, however we were not able to get to adding advanced features to the "events" category apart from functionality similar to the "To-do" list.
  - ...
- Accomplishments
  - Among the things we did complete successfully, we were able to establish a stable system for adding features and further developing our app, with front end interfaces functioning reliably and back end/deployment services running as they should, accepting new changes and features is easy as they are being created.
- Outstanding Issues:
  - Our events section is sorted by the time the user submitted the event instead of the time of the actual event

    - Events system does not allow for inviting others

- Future Plans
    - We are planning on implementing a very simplistic calendar that the user can incorporate with their notes. In doing so, we hope to bring in services such as Google's Calendar into our app.

<u>Project Report Write Up</u>

Our original main goal/objective was to create an online app split into 3 primary sections: 'To-do,' 'Notes,' and 'Events.' The idea is that these are the three main areas of life college students need to keep track of in a simple, intuitive way. The event section would have the ability to register for an event, assign multiple invitees to the event, and to attach notes to the event. We also had a goal of being able to upload photos/links to the notes section; however, this was postponed as we didn't see this as essential.

As of right now, the functionality for creating and modifying notes, events, and the to-do list is working; however, we were unable to get the added functionality of the event section (e.g. inviting other users) and the ability to upload photos/links to the note section.

Among our accomplishments, we were able to establish a stable, core system for adding features and further developing our project. Our front end systems are functioning reliably and the back end and deployment services are running as they should. With all the groundwork we have accomplished, it makes accepting new features and features easy. We are happy with how we designed our overall system and learned a lot doing it.

Our biggest issue is that our events section is sorting event entries by the time they were submitted instead of the date/time they are actually at. As we were building our core system, this was an oversight that never got fixed.

We plan to solve this issue by allowing the user to select the date and time of the event and add this field to our database. On the front end once we receive all of the events, we then will sort them by their date by grabbing the date and time for the JSON returned from our API. We also want to implement the ability to upload photos and links to the notes section. We let the ability to add other users to an event fall to the side as this would mean creating another table in the database and designing how we would store, modify, and match events and their attendees. Now that we don't have time pressure, we intend to explore this more as we believe it can teach us even more about how web services are created and developed.