Group name: PASSIO (Adams, Yuki, Kaycee, Jared, Bryce)
Lab time: Friday 8-10am

Milestone #4: Testing Update Report (Due March 29, 11:59pm)

For this milestone, you need to submit a pdf report with the following information:

1. **An update of your testing implementation so far:**

- **describe all the approaches you have done to test your project.**
    - Given the learning curve of the project, we have not been able to implement all the testing parameters we initially set out to complete.
    - We wanted to utilize a testing framework that gave us the most flexibility and options with our testing, so we landed on trying to use the pytest framework within python. Unfortunately, this proved to be a more difficult task than anticipated, so after spending a week trying to implement this, we reverted back to the standard unittest library within python.
    - Currently, we are implementing validation testing as we go to try and get our system to a functional level. This is predominantly in the form of white box testing. The main types of white box testing we are implementing are:
        - Statement coverage and branch coverage:
            - We want to not only see
            - As a means to make sure the smaller local paths within a method are working effectively.
        - Path and linear code path coverage:
            - We are then testing more functionality as a whole to see how different methods and components work together.
            - Also keeping in mind our goal of wanting to maintain clean code, we are also trying to utilize this as a refactoring technique to help isolate any unused or duplicate code that we can get rid of.
            - One issue we have run into with this is creating mock objects has proven difficult to implement within the standard unittest library within python, which does have a .mock extension. This is something we are still working on creating for our individual project components.
    - In addition to what is mentioned above, we are also trying to do system testing with the project components that are currently functional. By this, we are simply utilizing the system as a user to utilize the basic functions that are currently functional (login and page selection currently).
    - These above mentioned features have not been implemented in all pages of the project at present. Currently they are being implemented on the

login pages, in some of the account editing components and some of the collections pages.
    - We wanted to utilize black box testing as a means of test driven development, but given the unfamiliarity we had with the platforms we are using, this was not feasible for us to accomplish.
- **describe what else you will be doing to improve the testing in the future weeks**
    - We hope to implement the testing procedures discussed in the previous section in all of our pages where applicable. We hope in doing this, it will allow for a more seamless transition as we merge our different project components together; as it will allow us to determine if the addition is altering the functionality of the individual pages.
    - As we begin to create a functional webpage, we then hope to create tests that are testing the components via linear code and all-definition-use path coverage. This will allow us to test all paths of individual variables and confirm that they are running as desired. This will also include doing system testing as a whole, which we will discuss in a few bullet points below.
    - As we do this, we may have to go back and possibly remove previous tests or sections of code that may become redundant or unneeded. This is to get in the habit of proper code refactoring and keeping the code clean, which should help make implementing any new tests an easier process.
        - This alludes to another item we are hoping to work on in our code review. As we are getting a bit more familiar with our design, we are hoping that the test cases will help with our peer code review. Our goal is for this to occur in 3 main ways: 1. The individual doing the review can first look at the code and then trial the tests to see if we are testing what we are wanting to test. They can then check if there are other tests that should be implemented or if there are redundant tests that should be removed. Ideally this will make our code review have an additional level of validation and, ideally, clarity.
    - As the system comes together, we will be able to begin testing the overall system functionality. This system testing will ideally include the following:
        - First we want to have not only basic statement, branch and path coverage, but we will also want to have defect testing. This will be in the form of testing edge cases and any anomalies which could occur during the user interaction. This will ensure the user has a smooth experience from login to logout.

        - We will then progress from a development testing stage to working towards a user acceptance stage of the current product**. This would include getting some of our classmates and/or friends to trial our design and give us feedback on their thoughts of the interaction and if they ran into any issues.
            - **Do note, there are some functional features we may not get to implement given the time left in the semester. We want to set realistic goals for achievement for this stage and some of these (like processing payment and completing purchase) may not be completed by this time.
        - In order to complete full testing, we will likely need to create mock objects, as alluded to in the first section of this page. We have not isolated our exact means of doing this yet, but we are thinking of either continuing to trial the .mock function of the basic unittest library within Python. Or our other thought is that we could use mongomock which is a library used to test the code interactions between Pymongo and MongoDB. This is something we have not solidified as we are not at a point to begin this testing, but it is something that is on our radar. We are hoping this will allow us better insight into our testing and database interactivity.
            - MongoMock: https://github.com/mongomock/mongomock

2. **An update of the functionality that has been tested so far,**

- **<u>What's working, what's passing, what's not?</u>**
    - What's working:
        - Logging in and registering a user works as intended.
        - Logged in user data persisting between pages works as intended.
        - Entering an event into the system works as intended.
        - Displaying all available events with details works as intended.
        - Searching events by keyword works as intended.
    - What's in the works:
        - Actually purchasing the event via the checkout page and linking this to the attendee's profile.
        - Making all pages link together fluently for attendees to navigate the website fluently.
        - Finalize testing protocols.

3. **<u>What automation have you implemented in your project so far? How is it working?</u>**

- Added an automation into github that makes every new issue end up in the in progress column in project, this is done in the github/workflows folder
- Besides this we won't elaborate on this too much as a lot of it was mentioned in the previous sections. The main automation that is implemented so far are the unit tests we have implemented via the unittest library in Python.
- This works well for local testing, but once we have a more functional system, this will only be able to validate the local items within each page and not test the interactions are occurring as we want them to. This is going to be where we will utilize the aforementioned .mock and/or MongoMock to test this functionality is occurring appropriately.
- Kaycee - I have done 0 automation, learning 3 languages for a fully functional web app is enough. I do not have the time or the will to spend more time learning to automatically test my code when it'll probably take longer to learn the testing and set it up than just manually trying cases.

Also, add a summary of the project so far, where you are in the project, and reflect on how it is functioning. Make sure you add some comments on:

- **<u>the process; how is the process you selected as a team working? Does anything need to change?</u>**
  - The process we selected as a team has worked well overall. We have done a good job of working together and we have tried to pull from the principles of agile methods as we go. An example of this is we have focussed on the "people, not process" principle where we have tried to delegate tasks in a manner where we play to the strengths of our group members. If time allows, we will pair someone who has less experience in this area with the more knowledgeable person on that task to allow for collaborative learning to occur. Although this is not always feasible due to time constraints.
  - In terms of things that could be changed with our process, we are slowly working these out as we go. We have been struggling with the code review as we are learning how to do the coding, so it makes reviewing code quite difficult from a functional standpoint. But now that we have had last weeks lectures we can start implementing some of the code review strategies we have learned. The only other thing that we would like to change is our contracts as we feel we could be paid more.

- **<u>how do you ensure the quality of your source code? What has your team done to improve the quality?</u>**

- To ensure the quality of our source code we have been trying to have at least 2 people review the code before a pull request is merged into the main branch. However, this is something we could have been better at documenting and noting within github. To improve on this we are planning to start assigning reviewers in github, as well as implementing thorough comments for reviews that are done.
- We also have been trying to do code reviews with basic code structure in mind (as we learned from the summarized inspection checklist for common errors, provided to us on slides 35/36 in Lecture 1 of week 5 for COSC 310). This is something we have been doing in person as we work on our respective components during our weekly meetings. Again, something we had not been noting on Github, which we are planning to get better at. As discussed previously, a difficulty with code review is we are still learning how to make the code work, so reviewing code without understanding how to use it is quite difficult. However, we have been studying the ideas presented in last week's lectures around refactoring and maintaining clean code, which are strategies we plan to start implementing into our upcoming code reviews.