RU Classroom

Group Members:  Gianna Aprile and James Kyle Litton

Our project RU Classroom is a website that allows Rutgers students to search for an empty classroom to study in. The user can choose a campus (Busch, Livingston, etc.), a day of the week, and a time to find a room and we return a list of rooms and what time they're open until with a hyperlink to the map of where they are on the campus. We've been in the situation in the past where we've wanted to find an empty classroom to study in and have never found a way to do this without physically going to the room to check. This motivated us to create this service that makes the room search process much more convenient and simpler for students.

Our first step of creating this website was using Python to scrape the Schedule of Classes for the current class schedule for both the Spring and Fall semester and storing this information that's gathered in text files. We scraped this website using Selenium which allowed us to iterate through the drop-down menu that contains all the subject and used the click method to open the various subjects and classes. This was all done in the files: courseScraper.py, fall2019.txt, and spring2019.txt

Once we had the spring.txt and fall.txt files, we began setting up the HTML for the site. The first HTML form called my-form.html collects the initial input from the user and is essentially the homepage. The second HTML file is called result.html and it is where the final results of our searching through classes appear. These two forms are run using Flask in the file app.py.
In app.py we took the user data in through a form request and saved these as variables of campus, day, time (note we worked in military time in all of the backend work). These variables are sent into a method called "checkTime" in the search.py file.

In search.py we used the information in the spring2019.txt file to create a list of lists of objects for each campus so there are four lists total which was stored in one main list. Each object contains a day, a start time, an end time, and a room. This was all done in search.py in the method "createList" in search.py.

Once we had this list of four lists (one for each campus), we identified which campus the user requested and worked with that list from that point on. In the method "checkTime" in search.py, we created our initial return list called openRooms by starting with a list full of the rooms on that campus, and then removing a room if our method "vacant" returned false which means the room is occupied at that given day and time of the user.

Now that we have obtained the list openRooms, we used our method called “modifyList” in order to add the “open until: “ message that you see on our final webpage. Once this was accomplished, any room that did not have an open until message concatenated to the openRooms string for each room was given a message that said “open all day” because there was no classes after the requested time. At this point, we added called our “addRoomLinks” method which adds a hyperlink from the Rutgers Maps website. We did this by scraping this website to collect links for each room on each campus in the file createRoomInfoFile.py and stored the links in a text file called spring2019rmInfo.txt. Finally, we sorted this final list and returned it back to the app.py method which then deploys this final list as a table in result.html.

One of the challenges we faced along the way was learning to use a time.sleep method within the webscaping aspect of our program. Sometimes when we were still first learning to use Selenium, we would get an error and the program would stop running. It took us a while to realize that this was actually because our internet was too slow loading the page so some commands were trying to execute before the page was even loaded. We fixed this by importing a time method that we were able to use by calling time.sleep which gave the webpages time to load before our code continued.

Another more general challenge we faced was learning HTML and Flask, as neither one of us have worked with either in the past. However, we believe that this course (314) has helped us learn to adapt and languages quickly and this project pushed that even further.