Project #2: Web Client – Computer Network – 2021 Spring Semester

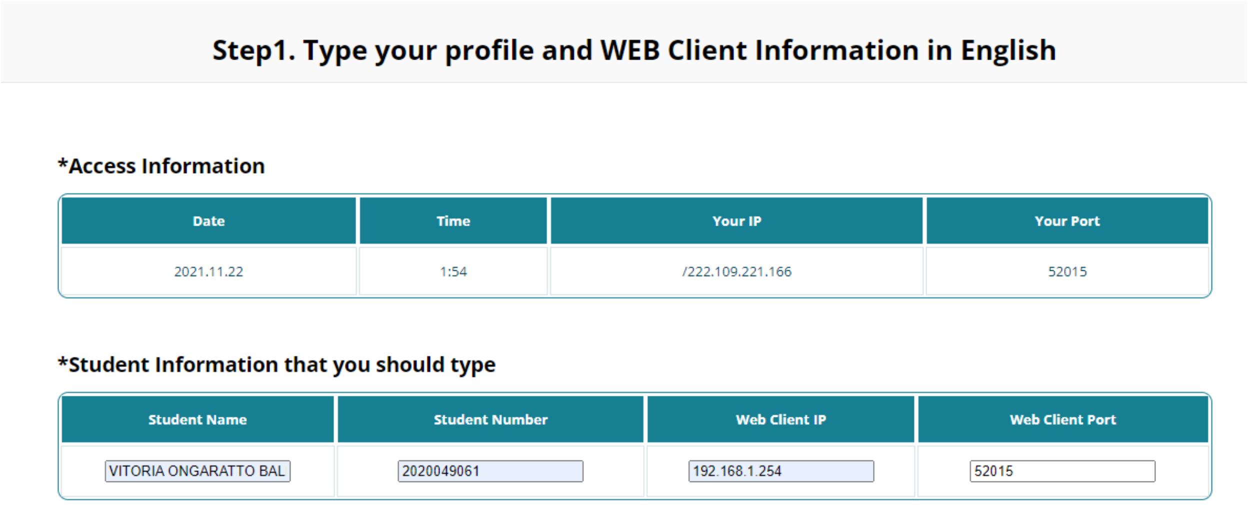
이름 : Vitoria Baldan / 비토리아
학번 : 2020049061

Introduction

This Project was about creating a Web Client application, connect with the previous project, the Web Server, and using both, accomplish some missions given by the instructions paper. The missions were displayed in the subject website with an automatic scoring program. The language in which we should be developing this application was free, therefore, I made my Web Server in Python language.

Instructions

First, it might be helpful to do port-forwarding in your router connecting with the Lab ports and IP. After that, you log-in on the website for the auto-marking program and write your information.



Step1. Type your profile and WEB Client Information in English

***Access Information**

Date	Time	Your IP	Your Port
2021.11.22	1:54	/222.109.221.166	52015

***Student Information that you should type**

Student Name	Student Number	Web Client IP	Web Client Port
VITORIA ONGARATTO BAL	2020049061	192.168.1.254	52015

Then, it's time to run your Web Server and Client applications.

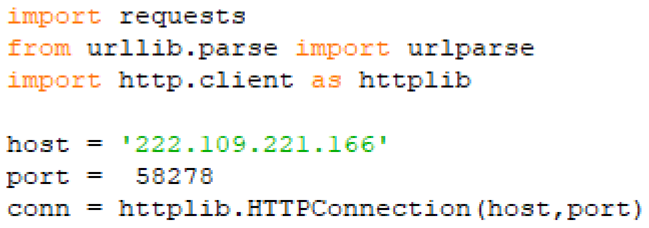
Once you run your applications you can start testing your programs in the auto-marking website, following the instructions for each mission.

It's important to pay attention if you're using the right port, the right personal information and also, the right IP addresses, so your application can connect and run without major issues.

How the program works

The Web Client program was made in python and the packages I used for making it work where the ‘request’ and ‘urllib’ (just ulrparse package). To make the connection I used the http.client package (Python3 version of the httplib package). I choose these packages for convenience, after doing some research over python packages for web client developing, I found out these two seem to be easier to deal with. Here is the official documentation for each package:

Python Request: HTTP for Humans: https://docs.python-requests.org/en/latest/
Urllib Package: https://docs.python.org/3/howto/urllib2.html
Httplib.client: https://docs.python.org/3/library/http.client.html



```
import requests
from urllib.parse import urlparse
import http.client as httplib

host = '222.109.221.166'
port =  58278
conn = httplib.HTTPConnection(host,port)
```

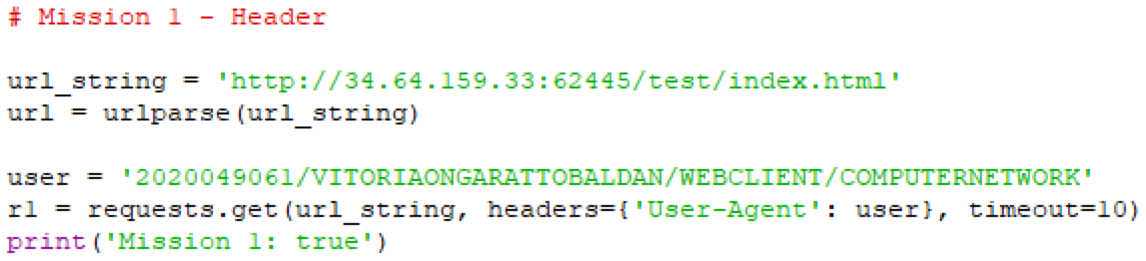
In the next parts of the code, it will be easy to see how simple and efficient these packages were to implement the entire algorithm.

It’s necessary to specify the host IP and the port for the connection. I did not used a fixed port while testing, so the values might differ along the rest of the prints in this document since I wrote all this Project while taking several tests with several different port numbers and Ips (tested in different places).

a) Mission 1 – The Header

First mission is about making a Header for the request. The user agent was built in the format ‘StudentID/NAME/PROJECTNAME/SUBJECTNAME’.

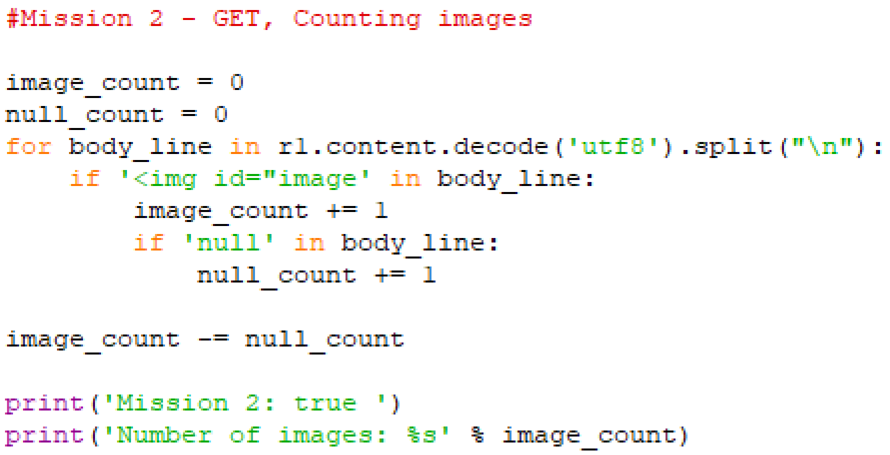
Using the method in the request and url parse packages was possible to send a GET request with user agent as determined before. I set a timeout of 10 seconds just to check if the program was working properly in a reasonable amount of time.



```
# Mission 1 - Header

url_string = 'http://34.64.159.33:62445/test/index.html'
url = urlparse(url_string)

user = '2020049061/VITORIAONGARATTOBALDAN/WEBCLIENT/COMPUTERNETWORK'
r1 = requests.get(url_string, headers={'User-Agent': user}, timeout=10)
print('Mission 1: true')
```

b) Mission 2 – GET method and image counter

For mission 2, we needed to count how many images where being shown in the page, also using GET method that I showed in the scope of mission 1 above. For counting the images, I used two counters, one to count how many images were displayed and other to count how many null images were displayed. Then, the final image count would be the total amount of images minus the number of null images, giving as result just the number of valid (visible) images in the page.

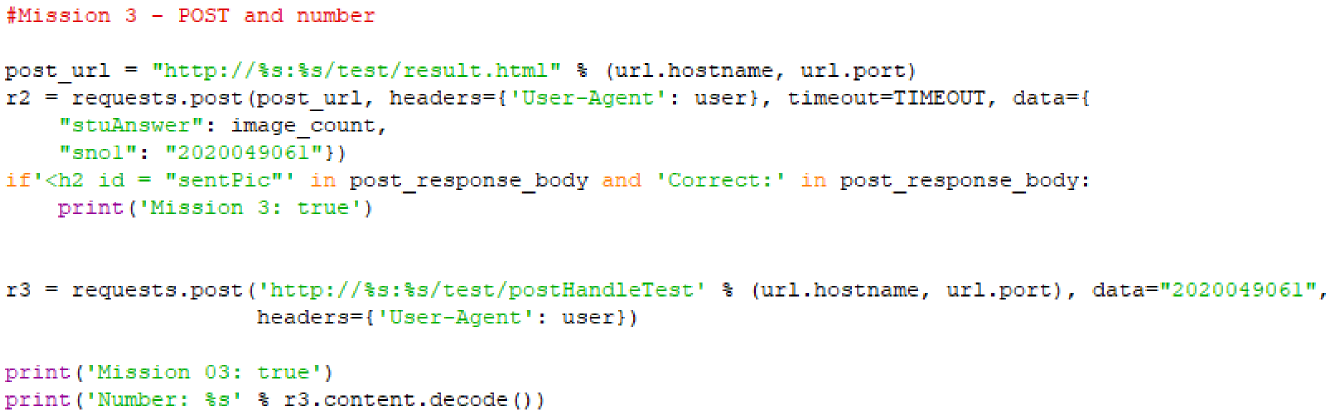


```
#Mission 2 - GET, Counting images

image_count = 0
null_count = 0
for body_line in r1.content.decode('utf8').split("\n"):
    if '<img id="image' in body_line:
        image_count += 1
        if 'null' in body_line:
            null_count += 1

image_count -= null_count

print('Mission 2: true ')
print('Number of images: %s' % image_count)
```

c) Mission 3 – POST method and number identification

For mission 3, we were required to send the URL message by POST method, assigning the specific url with its hostname and the port.

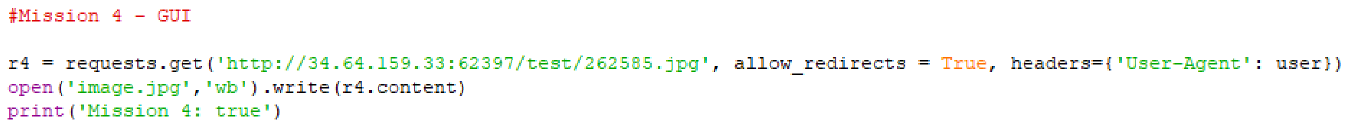
After checking if the information is correct and decoding the content from the answer r2 using utf8, r3 will send the post request to the URL build with the student number and User Agent, and then, display the number sequence from the Auto Marking Program on the page.



```
#Mission 3 - POST and number

post_url = "http://%s:%s/test/result.html" % (url.hostname, url.port)
r2 = requests.post(post_url, headers={'User-Agent': user}, timeout=TIMEOUT, data={
    "stuAnswer": image_count,
    "snol": "2020049061"})
if'<h2 id = "sentPic"' in post_response_body and 'Correct:' in post_response_body:
    print('Mission 3: true')


r3 = requests.post('http://%s:%s/test/postHandleTest' % (url.hostname, url.port), data="2020049061",
                   headers={'User-Agent': user})

print('Mission 03: true')
print('Number: %s' % r3.content.decode())
```

d) Mission 4 (Optional) – GUI and image identification

To build the part of the code for this mission it was simple. It was possible by sending a GET method request to the host, with User-Agent headers, allowing redirections and then opening the image received writing the r4 request content. This image will be saved in the user's computer and also be visible through the page in the link given in the Auto Marking program.

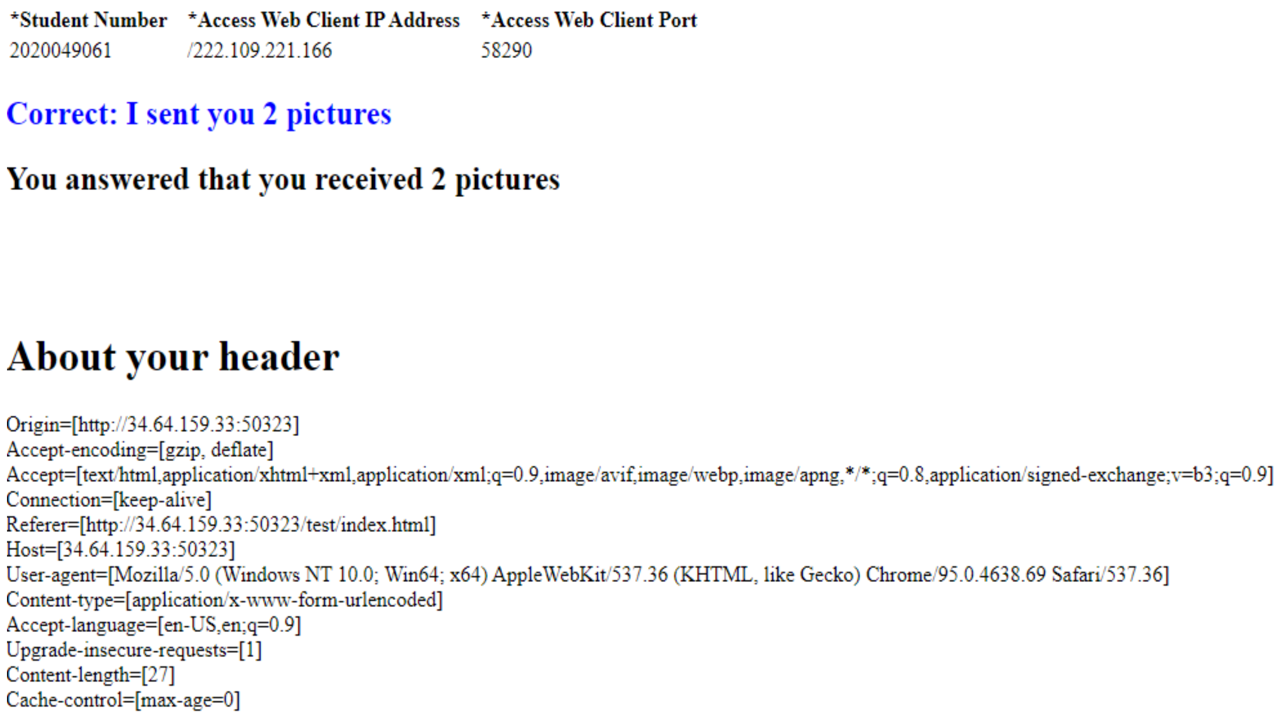


```
#Mission 4 - GUI

r4 = requests.get('http://34.64.159.33:62397/test/262585.jpg', allow_redirects = True, headers={'User-Agent': user})
open('image.jpg','wb').write(r4.content)
print('Mission 4: true')
```

Results

a) Auto Marking Program

First, for Mission 1 and 2, after answering the images question, the result was:



*Student Number	*Access Web Client IP Address	*Access Web Client Port
2020049061	/222.109.221.166	58290

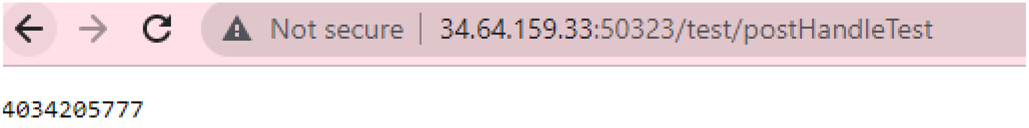
Correct: I sent you 2 pictures

You answered that you received 2 pictures

About your header

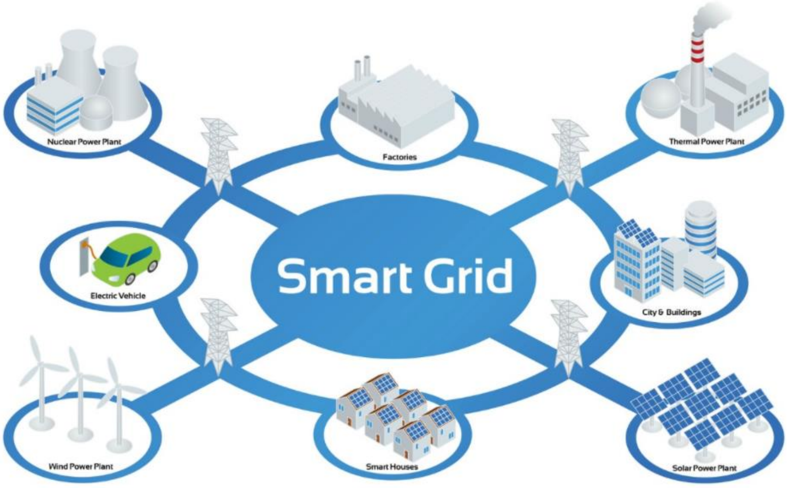
```
Origin=[http://34.64.159.33:50323]
Accept-encoding=[gzip, deflate]
Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9]
Connection=[keep-alive]
Referer=[http://34.64.159.33:50323/test/index.html]
Host=[34.64.159.33:50323]
User-agent=[Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36]
Content-type=[application/x-www-form-urlencoded]
Accept-language=[en-US,en;q=0.9]
Upgrade-insecure-requests=[1]
Content-length=[27]
Cache-control=[max-age=0]
```

After that, for Mission 3, the number received was:

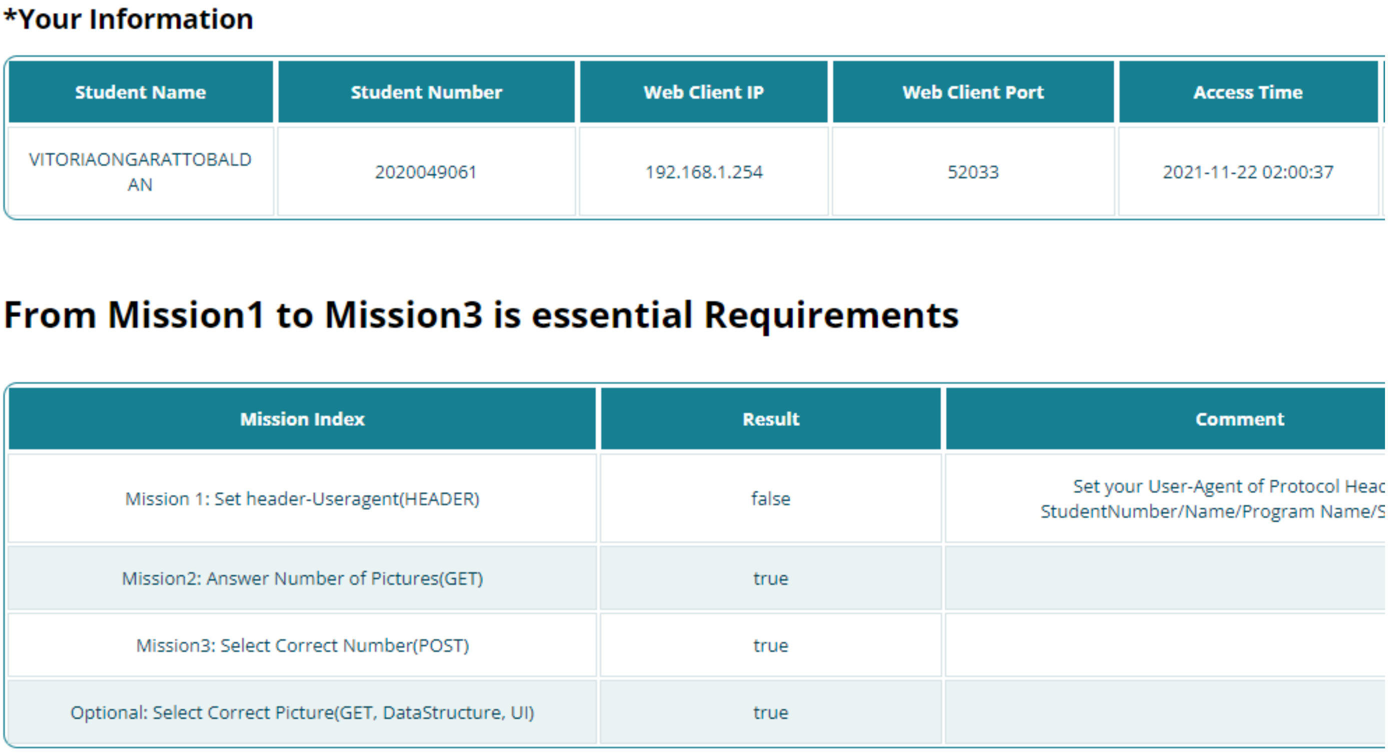


```
4034205777
```

Finally, for Mission 4, the image received was



And the result of the Auto Marking program I received was:



***Your Information**

Student Name	Student Number	Web Client IP	Web Client Port	Access Time
VITORIAONGARATTOBALD AN	2020049061	192.168.1.254	52033	2021-11-22 02:00:37

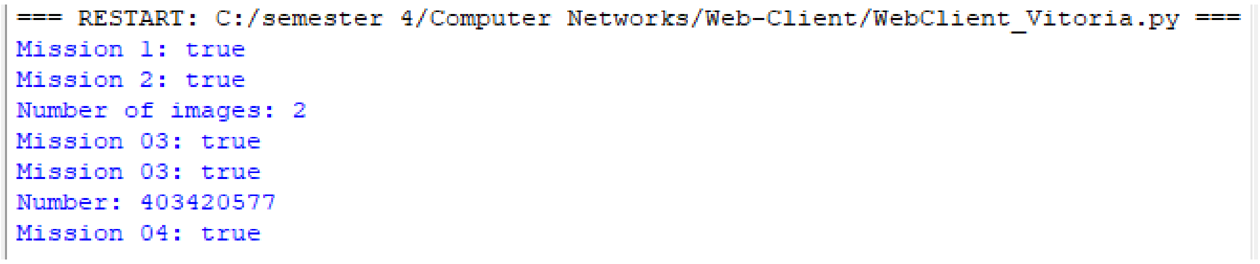
From Mission1 to Mission3 is essential Requirements

Mission Index	Result	Comment
Mission 1: Set header-Useragent(HEADER)	false	Set your User-Agent of Protocol Heac StudentNumber/Name/Program Name/S
Mission2: Answer Number of Pictures(GET)	true	
Mission3: Select Correct Number(POST)	true	
Optional: Select Correct Picture(GET, DataStructure, UI)	true	

The mission 1 seems to be false, I tried to fix several times, but I suspect is some issue related to firewall, even though I tried by port forwarding, tried in a noncommercial web browser (used Brave). Maybe it could also be a problem with my name format, even though I tried several different possibilities. I revised my code several times and it doesn't seem to be a coding issue, more like a connection issue. However, in the next print it's possible to see that in my python shell the mission 1 seems to be true, but I'm not sure if it's a viable answer from my code either.

b) Results from Python Shell

This is a shell capture from my results



```
=== RESTART: C:/semester 4/Computer Networks/Web-Client/WebClient_Vitoria.py ===
Mission 1: true
Mission 2: true
Number of images: 2
Mission 03: true
Mission 03: true
Number: 403420577
Mission 04: true
```

As we can see, my shell result shows that all the missions where accomplished and displays the right values for the answers.

About Mission 1, as I mentioned before, is not clear if the message in the shell is valid since the Auto Marking program marked that as 'false'. I was not able to identify the issue.

**Obs: The ports might be different from a capture to another because they are from different testing, and I changed ports several times.*

Personal thoughts

This Project was really interesting to do, I came to learn a lot of features and packages Python has to deal with HTTP and Web Client requests, and I learned the mechanism behind the web server and client connections and the all the data exchange. After doing the Project 1 and Project 2 I feel that all the Internet, that once was something very abstract to me, is now gaining form, and I can see and understand how things works in a way I never could before.

Regarding my results, I'm triggered about the Mission 1 'false' sign in the Auto Marking program while marking 'true' in my Python Shell results. I was not able to identify the issue, what made me a little bit unsatisfied. I wish I could have gone to the Lab to test my code there, but for personal reasons I could not start the Project earlier and I've been self-quarantining for a while because I was exposed to a high risk of COVID infection situation and even though I tested negative, I decided to keep myself far from other people to don't put them in risk of infection. If circumstances where different I would definitely have gone to the Lab and ask help for the Assistants.

I wish this class and all the labs could have been offline, I'm learning a lot, but as I don't have contact with any classmate, I could never make any friends to discuss project matters with, or to just talk about the subject and class contents. I wonder how fun it would be to do these projects in a group and discuss about it!

In general, I learned a lot while working on this Project and I hope in the future we can have the opportunity to learn and explore more projects like this, especially offline. I feel much more qualified to work with Web and network related jobs than ever. Looking forward to the next projects!