

# 2021 스터디상생플러스 결과 보고서



2021.08.22.

넥스트레블 팀

# 1. 프로젝트 개요

=====

## 1) '상명 알리미'란?

상명대학교는 지속되는 코로나 상황 속에서 e-campus 시스템을 활용해 강의를 업로드하고, 강의 진도율 및 주차별 과제를 통해 출석 체크를 하는 방식을 활용하고 있다. 이 시스템을 직접 경험하고 있는 학생들은 그동안 자신이 수강하는 모든 수업의 강의 목록을 일일이 클릭해 강의와 과제를 하나씩 직접 확인해야 한다는 점에 굉장한 불편함을 호소해왔다. 또한, 그런 불편함 때문에 확인을 제대로 하지 못해 마감 기한에 맞추어 제출하지 못하고, 실제로 이로 인해 출석 부분에서 점수가 깎이는 주변 학생들을 많이 볼 수 있었다. 상명 알리미는 이러한 스트레스를 받는 학생들을 위해 구상된 프로젝트이다.

모두가 처음 겪는 이 상황 속에서 학생들이 겪고 있는 불편함을 해소해줄 수 있는 디스코드 봇이 바로 '상명 알리미'이다. 학생들이 많이 사용하고 있는 SNS 앱 중 하나인 Discord를 통해 명령어 하나만 간단하게 입력하면, 우리가 현재 듣고 있는 강의 중 진도율 100% 미만인 강의 목록과 제출하지 못한 과제 목록을 한눈에 살펴볼 수 있다. 이는 상명대학교 학생들의 편리함을 보장해줄 최고의 디스코드 봇이다.

## 2) 목적 및 필요성

코로나 19로 인해 전면 비대면 또는 융합 수업을 진행하면서, 주로 e-campus와 코스모스 앱을 이용하여 온라인 강의를 진행하게 되었다. 그런데 e-campus 웹사이트 또는 코스모스 앱을 이용할 때 학교 진도 및 과제 제출 현황을 확인하기 위해서는 강좌를 일일이 클릭해야 했다. 이러한 점에 대해 우리 팀원들뿐만 아니라 다른 학우들도 불편함을 호소하였

고, 학업적인 면에서 스트레스를 받는다는 커뮤니티 게시물을 많이 볼 수 있었다.

따라서 사용자가 e-campus 사이트에 직접 접속하지 않고도 강의 진도와 과제 제출 현황을 한눈에 볼 수 있는 서비스를 구현하여, 학우들의 불편함을 해소하고자 한다. 또한 이 기능과 더불어 매일 사용자가 설정한 시각마다 미수강, 미제출 상태인 강의 및 과제 목록을 전송하여 학우들이 비대면 상황 속에서도 강의를 더욱 수월하게 들을 수 있도록, 학습 효율을 증가시킬 수 있도록 편의를 제공하는 것이 이 프로젝트의 진행 목적 및 필요성이라고 할 수 있다.

## 2. 연구내용

### 1) 웹 스크레이핑(Web Scraping)

상명대학교 e-campus의 사용자 정보에 따른 강의 목록 리스트를 끌어오고, 그 목록 각각의 강의 진도 현황과 과제 제출 현황의 필요한 부분만 가져오기 위해 웹 스크레이핑 기법을 활용하였다.

```
import requests
from bs4 import BeautifulSoup
```

[사진] 크롤링(Crawling)을 위해 사용한 모듈

웹 스크레이핑(Web Scraping)이란 웹에 존재하는 정보를 수집하는 작업이다. 다양한 웹 스크레이핑 기법 중에서, 우리는 HTML의 태그를 파싱하여 필요한 데이터만을 추출할 수 있는 함수를 제공해주는 BeautifulSoup 라이브러리를 활용해 스크레이핑 작업을 진행하였다. 동시에 하이퍼 텍스트 형식으로 주고 받는 프로토콜인 http의 요청을 웹 브라우저가 아닌 python에서 모듈로서 가능하게 해주는 requests 모듈 또한 활용하였다. 이 두 모듈을 통해 크게 두 단계로 크롤링을 진행하였다.

첫 번째, 웹 스크레이핑을 위한 모든 url의 주소를 정의한 후, 상명대학교 e-campus의 사용자 정보를 user\_info 리스트에 넣어주고, 그 리스트를 통해 BeautifulSoup 라이브러리의 함수를 통해 끌어온 로그인 창으로 로그인을 해 준 다음, 해당 사용자가 듣고 있는 강의의 목록을 스크레이핑 해온다.

두 번째, 사용자의 정보에 따라 듣고 있는 강의의 종류와 개수가 다르기 때문에, 강의 하나가 가지고 있는 고유 id 주소를 스크레이핑 함으로써 해당 사용자의 강의 목록을 파싱하고, 각각의 강의 진도 현황표에서 필요한 요소와 과제

제출 현황표에서 필요한 요소인 '제목', '현황', '마감 기한'을 스크레이핑 한 결과를 text 처리하여 정리한 모습으로 출력한다.

## 2) 디스코드 봇

강의 진도 현황과 과제 제출현황을 스크레이핑 한 코드를 디스코드 봇에 적용한다. 디스코드 봇에 구현한 코드를 적용하기 위해 다음과 같은 모듈을 사용하였다.

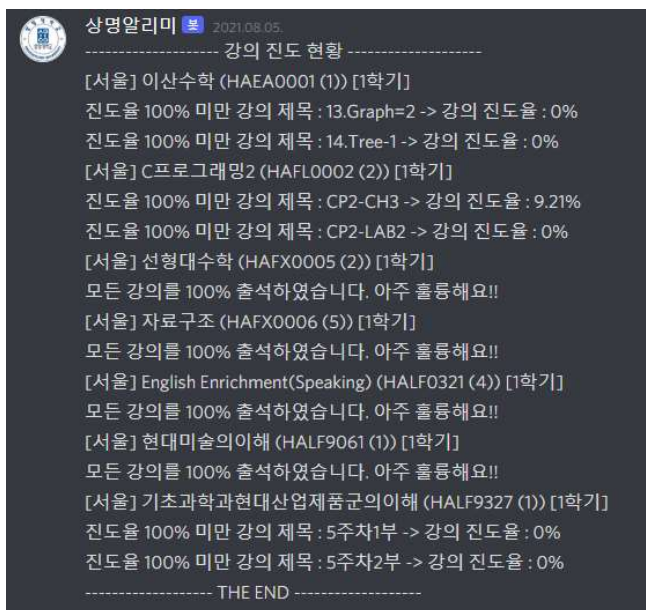
'discord'는 디스코드 봇 실행을 위한 모듈, 'asyncio'는 명령어 실행을 위한 모듈, 'datetime'은 알람 기능 구현을 위한 모듈이고, 'urllib'는 URL 발생 처리를 위한 모듈이다.

```
import discord
from discord.ext import commands
import asyncio
import datetime as dt
from datetime import datetime, timedelta
from urllib.request import urlopen
from urllib.error import URLError
```

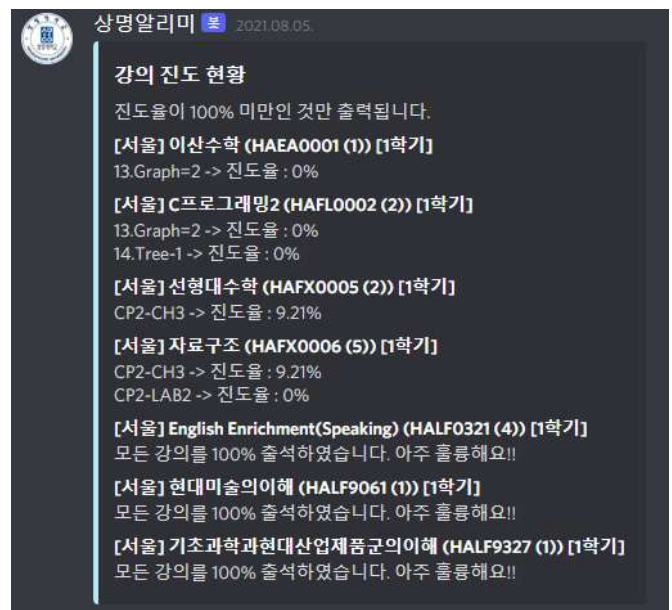
[사진] 디스코드 봇 실행을 위해 사용한 모듈

일단 디스코드 개발자 페이지에서 디스코드 봇을 생성한다. 생성한 봇의 토큰을 코드에 넣음으로써 파이썬 코드가 디스코드 봇에서 실행이 되게 한다. @bot.event 함수를 사용해 디스코드 봇을 실행시키고, status 함수로 현재 봇의 상태를 출력해주게 한다. 또한 @bot.command() 함수를 통해 명령어의 기능을 수행하는 코드를 넣어준다. 파이썬 함수에서 사용하던 print 함수는 디스코드 봇에서 사용자에게 메시지를 전달하는 await ctx.send() 함수를 사용하였다.

‘상명 알리미’ 봇은 e-campus를 활용하는 상명대학교 학생들이 불편함으로 인해 제대로 확인하지 못하여 생기는 피해를 해소해주기 위한 시스템이다. 따라서 모든 강의와 과제의 현황을 출력하는 것이 아닌, 조건문과 반복문을 통해 진도율이 100% 미만인 강의와 미제출 상태인 과제만을 출력하게 하여 가장 필요한 정보들만 출력하게 하였다. 이때, 해당 수업의 모든 강의와 과제를 100% 완료했다면 미완료인 강의나 과제가 없음을 사용자에게 알리도록 했다. 사용자들이 한눈에 확인할 수 있도록 디스코드 기능 중 임베드 기능을 활용하였다. 임베드 기능은 디스코드에 존재하는 기능으로, 사용자가 보다 편하게 볼 수 있도록 도와주는 기능이다. 이해를 돕기 위해 아래 예시를 첨부하였다. 임베드 기능을 활용하기 위해 Embed 함수를 통해 상자 안에 출력할 내용들만을 보기 좋게 정리하였다.



예시 1 임베드 기능을 활용하지 않고 보내는 경우



예시 2 임베드 기능을 활용하여 보내는 경우

‘상명 알리미’ 봇은 기본적으로 명령어와 함께 e-campus 아이디와 비밀번호를 입력하면 임베드를 통해 미완료인 강의와 과제를 출력한다. 그 외에도 datetime 모듈을 통해 알람 기능을 수행하게 된다. 알람 기능을 위해서는 사용자에게 아이디, 비밀번호, 알람을 받을 시간을 입력받아, 알람 기능을 수행하는

함수의 매개 변수로 전달한다. datetime 모듈을 통해 현재 시각을 출력하고, 알람 기능에서 매개 변수로 받은 시간과 현재 시각을 비교하여, 현재 시각과 알람을 받을 시간이 일치한다면 사용자에게 강의 진도 현황과 과제 제출 현황을 출력해준다. 이 알람은 while문을 돌며 정해진 시간마다 사용자에게 메시지를 보낸다.

이러한 기능들을 수행하는 디스코드 봇은 vs code로 코드를 동시에 실행하고 컴퓨터가 계속 켜져 있을 때에만 실행이 된다는 단점이 있다. 이를 보완하기 위하여 24시간 동안 디스코드 봇을 구동시킬 수 있는 방법으로 'heroku'를 선택하였다. 'heroku'란 무료 호스팅이자 서버를 대체할 수 있는 클라우드 플랫폼이다. 깃허브와 연결하여 몇 가지 설정만으로 봇을 24시간동안 구동시킬 수 있었다.

이때, 'heroku'를 통해 디스코드 봇을 24시간 동안 실행시키면 예상치 못한 시간에 임베드가 출력된다는 것을 깨달았다. 한참의 조사와 고민 끝에 이는 'heroku'가 미국 서버를 사용하기 때문에 우리나라의 서버 시간과 차이가 있기 때문이라는 점을 발견하였다. 이를 해결하기 위해 현재 시각을 계산하기 위한 함수에서 (hours = 9) 코드를 추가하여 미국 서버 시간과 한국 서버 시간을 일치시켜주는 작업을 진행하였다.

이 외에도 간단한 명령어들에 대한 설명이 담긴 임베드를 출력해주는 명령어를 수행하도록 embed 함수를 한 번 더 활용했으며, 에러 발생에 대한 메시지 출력도 해줄 수 있게 한다. 스크레이핑 하는 과정에서 서버를 찾을 수 없는 상황이 발생하면 URL ERROR가 발생하게 되는데 이를 try-except 함수를 통해 해결했다. try 함수 속의 코드를 통해 로그인 창 의 url 주소를 열어주는 작업을 수행한다. 이 때 URL ERROR가 발생하게 되면 e-campus 페이지에 오류가 발생했다는 메시지를 전달하고, 에러가 발생하지 않은 채 url 주소가 성공적으로 열린다면 지금까지의 코드를 무사히 수행할 수 있도록 하였다.

### 3. 프로젝트 결론

=====

#### 1) 결과물

##### ● 디스코드봇 결론 코드

##### 1> 봇 준비 상태

```
prefix = '!'
bot = commands.Bot(command_prefix = prefix)

@bot.event
async def on_ready():
    await bot.change_presence(status = discord.Status.online, activity = discord.Game("열공하는 스무들 기다리는 중 :-"))
    print("Bot is ready!")
```

##### 2> 알람 기능 & 확인 기능 (시간 제외 원리 같음)

```
#강의진도현황크롤링하기
if (request3.status_code == 200):
    value_lst = []
    for i in range(count):
        request3 = s.post(url_lst[i])
        name_lst = []
        rate_lst = []
        a_li = []
        bs = BeautifulSoup(request3.text, 'html.parser')
        lecture_name = bs.find_all("td", {"class", "text-left"})
        rates = bs.find_all("td", {"class", "text-center"})

        for name in lecture_name:
            name_lst.append(name.text.strip())
        name_lst = name_lst[3:]

        for rate in rates:
            rate_lst.append(str(rate.text))

        for word in rate_lst:
            if '%' in word:
                a_li.append(word)

        size = len(a_li)

        i += 1
        absent_lst = []
        value = ''
        for i in range(0, size - 1, 1):
            if float(a_li[i][:-1]) < 100:
                string = name_lst[i] + ' -> 진도를 : ' + a_li[i]
                absent_lst.append(string)

        if len(absent_lst) == 0:
            value_lst.append("모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!")
        else:
            for i in absent_lst:
                value += (i + '\n')
            value_lst.append(value)

    value = ''
    embed = discord.Embed(title = "진도를 100% 미만 강의 목록(2021학년도 1학기)", description = t, color = 0xB2EBF4)
    for i in range(count):
        embed.add_field(name = class_name_lst[i], value = value_lst[i], inline = False)
```



```

#과제진도현황크롤링하기
if (request.status_code == 200):
    value2_lst = []
    for i in range(count):
        request4 = s.post(url_lst_assign[i])

        bs = BeautifulSoup(request4.text, 'html.parser')
        assignment_name = bs.find_all("td", {"class", "cell c1"})
        assignment_rates = bs.find_all("td", {"class", "cell c3"})
        assignment_close = bs.find_all("td", {"class", "cell c2"})

        assignment_name_lst = []
        for name in assignment_name:
            assignment_name_lst.append(name.text.strip())

        assignment_rate_lst = []
        for rate in assignment_rates:
            assignment_rate_lst.append(rate.text.strip())

        assignment_close_lst = []
        for close in assignment_close:
            assignment_close_lst.append(close.text.strip())
        size = len(assignment_name_lst)

        i += 1

        absent = '미제출'
        unsubmit_lst = []
        value2 = ''

        for i in range(0, size):
            if absent in assignment_rate_lst[i]:
                string = assignment_name_lst[i] + ' / 현황: ' + assignment_rate_lst[i] + ' / 마감일: ' + assignment_close_lst[i]
                unsubmit_lst.append(string)

            if absent not in assignment_rate_lst:
                value2_lst.append('모두 제출 완료!!')
            elif absent in assignment_rate_lst:
                for i in unsubmit_lst:
                    value2 += (i + '\n')
                value2_lst.append(value2)

        embed2 = discord.Embed(title = '미제출 현황 과제 목록(2021학년도 1학기)', description = t, color = 0xFFD8D8)
        for i in range(count):
            embed2.add_field(name = class_name_lst[i], value = value2_lst[i], inline = False)
        await ctx.send(embed = embed)
        await ctx.send(embed = embed2)

    await asyncio.sleep(60)

# 원하는 시간에 강의 진도 현황과 과제 제출 현황 출력
@bot.command()
async def 알람(ctx, id, pw, alarm):
    login_url = 'https://ecampus.smu.ac.kr/login/index.php' #로그인 창 주소
    class2021_url = 'https://ecampus.smu.ac.kr/local/ubion/user/?year=2021&semester=10' #강의 인덱스 가져오기 위한 주소
    url_lst = []
    url_lst_assign = []

    user_info = {}

    user_info['username'] = id
    user_info['password'] = pw

    try:
        html = urlopen(login_url)
    except URLError as e:
        await ctx.send('Ecampus ERROR: 이캠퍼스 페이지에 오류가 있습니다. \ (ㄹ_ㄹ)!!!')
    else:
        while True:
            current_time = dt.datetime.now() + dt.timedelta(hours=9)
            t = "현재 시각: " + current_time.strftime('%Y-%m-%d %H:%M')
            hour = current_time.strftime('%H:%M')

            if alarm == hour:
                with requests.Session() as s:
                    #필요한 주소들 & selector들 미리 설정
                    request = s.post(login_url, data = user_info)
                    request3 = s.post(class2021_url, data = user_info)
                    source = request3.text
                    soup = BeautifulSoup(source, 'html.parser')
                    items = soup.find_all("a", {"class", "coursefullname"})

                    #강의 제목 가져오기
                    class_name_lst = []
                    for name in items:
                        arrange = name.get_text()
                        class_name_lst.append(arrange)

                    #url에 있는 강의가 갖고 있는 id코드 크롤링하기
                    if (request.status_code == 200):
                        bs = BeautifulSoup(request3.text, 'html.parser')

                        lectures = bs.select("#region-main > div > div > div.course_lists > div > table > tbody > tr > td > div > a")

                        class_list = []
                        for lecture in lectures:
                            class_list.append(str(lecture))
                        lst = []
                        for i in class_list:
                            index = i.find('id')
                            lst.append(int(i[index+3:index+8]))
                        count = len(lst)
                        for i in range(count):
                            url_lst_assign.append('https://ecampus.smu.ac.kr/mod/assign/index.php?id='+str(lst[i]))
                            url_lst.append('https://ecampus.smu.ac.kr/report/ubcompletion/user_progress.php?id='+str(lst[i]))

```

### 3> 설명 기능

```
#명령어를 설명해주는 [[설명]] 기능

@bot.command()
async def 설명(ctx):

    embed3 = discord.Embed(title="명령어 설명", description="꺄바로우봇의 명령어에 대한 설명입니다 *~ (> , a) ", color = 0x4030EA)
    embed3.add_field(name="!알람 (아이디) (비밀번호) (알람을 울릴 시간)", value = "알람을 설정해두면 매일 그 시간에 메시지를 보내드려요! (알람 시간 예시: 00:20, 09:00, 22:00 등)", inline = False)
    embed3.add_field(name="!확인 (아이디) (비밀번호)", value="100% 미만 진도율의 강의 목록과 미제출 과제 목록을 언제든 확인할 수 있어요! o(*'▽'*)/~")
    embed3.set_thumbnail(url="상명대로고이미지주소")

    await ctx.send(embed=embed3)

bot.run(token)
```

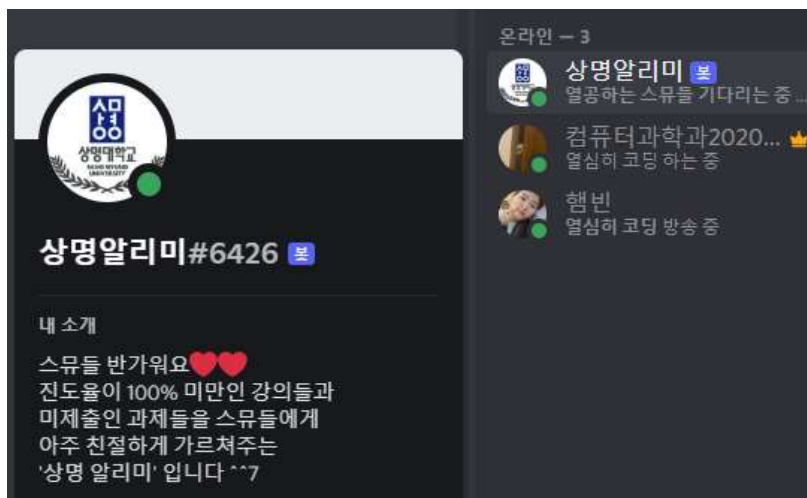
### 4> 에러 발생 메시지 출력 (2번 코드 부가 설명)

```
try:
    html = urlopen(login_url)
except URLError as e:
    await ctx.send('Ecampus ERROR: 이캠퍼스 페이지에 오류가 있습니다. \ (ㄹㅇ)/!!!')
else:
    while True:
```

### 5> 'heroku' 실행 위한 코드 추가 (2번 코드 부가 설명)

```
current_time = dt.datetime.now() + dt.timedelta(hours=9)
t = "현재 시각: " + current_time.strftime('%Y-%m-%d %H:%M')
hour = current_time.strftime('%H:%M')
```

## ● ‘상명 알리미’의 프로필과 상태메세지



## ● ‘!설명’ 명령어 실행

컴퓨터과학과202010904이혜린  
!설명

상명알리미 **보** 오늘 오전 1:29

**명령어 설명**

쿼바로우봇의 명령어에 대한 설명입니다 ☆ ~ ( > . θ )

**! 알람 (아이디) (비밀번호) (알람을 올릴 시간)**  
알람을 설정해두면 매일 그 시간에 메시지를 보내드려요! (알람 시간 예시: 00:20, 09:00, 22:00 등)

**! 확인 (아이디) (비밀번호)**  
100% 미만 진도율의 강의 목록과 미제출 과제 목록을 언제나 확인할 수 있어요! o(▽)/☆"

## ● ‘!알람’ 명령어 실행

햄빈 2021.08.20.  
!알람 202010869 01:54

상명알리미 **보** 2021.08.20.

**진도율 100% 미만 강의 목록(2021학년도 1학기)**  
현재 시각: 2021-08-20 01:54

[서울] 이산수학 (HAEA0001 (1)) [1학기]  
13.Graph=2 -> 진도율 : 0%  
14.Tree-1 -> 진도율 : 0%

[서울] C프로그래밍2 (HAFI0002 (2)) [1학기]  
CP2-CH3 -> 진도율 : 9.21%  
CP2-LAB2 -> 진도율 : 0%

[서울] 선형대수학 (HAFX0005 (2)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] 자료구조 (HAFX0006 (5)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] English Enrichment(Speaking) (HALF0321 (4)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] 현대미술의이해 (HALF9061 (1)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] 기초과학과현대산업제품공의이해 (HALF9327 (1)) [1학기]  
5주차1부 -> 진도율 : 0%  
5주차2부 -> 진도율 : 0%

**미제출 현황 과제 목록(2021학년도 1학기)**  
현재 시각: 2021-08-20 01:54

[서울] 이산수학 (HAEA0001 (1)) [1학기]  
모두 제출 완료!!

[서울] C프로그래밍2 (HAFI0002 (2)) [1학기]  
모두 제출 완료!!

[서울] 선형대수학 (HAFX0005 (2)) [1학기]  
모두 제출 완료!!

[서울] 자료구조 (HAFX0006 (5)) [1학기]  
비대면 기말고사 / 현황: 미제출 / 마감일: 2021-06-18 16:00

[서울] English Enrichment(Speaking) (HALF0321 (4)) [1학기]  
Group Presentation I (Week 7) Scores / 현황: 미제출 / 마감일: 2021-04-15 11:00  
Individual Presentation (Scores) / 현황: 미제출 / 마감일: 2021-04-29 13:00  
Group Presentation II (Scores) / 현황: 미제출 / 마감일: 2021-05-27 13:00

[서울] 현대미술의이해 (HALF9061 (1)) [1학기]  
조별활동결과 제출 / 현황: 미제출 / 마감일: 2021-05-14 15:00

[서울] 기초과학과현대산업제품공의이해 (HALF9327 (1)) [1학기]  
모두 제출 완료!!

## ● ‘!확인’ 명령어 실행

햄빈 2021.08.20.  
!확인 202010869

상명알리미 **보** 2021.08.20.

**진도율 100% 미만 강의 목록(2021학년도 1학기)**  
현재 시각: 2021-08-20 02:16

[서울] 이산수학 (HAEA0001 (1)) [1학기]  
13.Graph=2 -> 진도율 : 0%  
14.Tree-1 -> 진도율 : 0%

[서울] C프로그래밍2 (HAFI0002 (2)) [1학기]  
CP2-CH3 -> 진도율 : 9.21%  
CP2-LAB2 -> 진도율 : 0%

[서울] 선형대수학 (HAFX0005 (2)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] 자료구조 (HAFX0006 (5)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] English Enrichment(Speaking) (HALF0321 (4)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] 현대미술의이해 (HALF9061 (1)) [1학기]  
모든 강의를 100% 출석하였습니다. 아주 훌륭해요!!

[서울] 기초과학과현대산업제품공의이해 (HALF9327 (1)) [1학기]  
5주차1부 -> 진도율 : 0%  
5주차2부 -> 진도율 : 0%

**미제출 현황 과제 목록**  
현재 시각: 2021-08-20 02:16

[서울] 이산수학 (HAEA0001 (1)) [1학기]  
모두 제출 완료!!

[서울] C프로그래밍2 (HAFI0002 (2)) [1학기]  
모두 제출 완료!!

[서울] 선형대수학 (HAFX0005 (2)) [1학기]  
모두 제출 완료!!

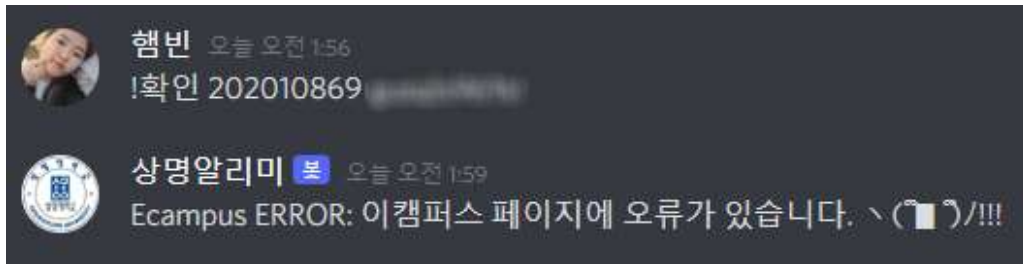
[서울] 자료구조 (HAFX0006 (5)) [1학기]  
비대면 기말고사 / 현황: 미제출 / 마감일: 2021-06-18 16:00

[서울] English Enrichment(Speaking) (HALF0321 (4)) [1학기]  
Group Presentation I (Week 7) Scores / 현황: 미제출 / 마감일: 2021-04-15 11:00  
Individual Presentation (Scores) / 현황: 미제출 / 마감일: 2021-04-29 13:00  
Group Presentation II (Scores) / 현황: 미제출 / 마감일: 2021-05-27 13:00

[서울] 현대미술의이해 (HALF9061 (1)) [1학기]  
조별활동결과 제출 / 현황: 미제출 / 마감일: 2021-05-14 15:00

[서울] 기초과학과현대산업제품공의이해 (HALF9327 (1)) [1학기]  
모두 제출 완료!!

● URL ERROR 발생 출력 (e-campus 서버 이전 중일 때 실행)



## 2) 기대효과 & 방향성

‘상명 알리미’가 상명대학교 학생들의 큰 불편함을 줄여주기 때문에 그동안 불편함으로 인해 받았던 스트레스가 감소하고, 그에 따라 학습 효율을 증가시킬 수 있다. 정해진 시간에 주기적으로 알람을 보냄으로써 학우들에게 학습 동기를 부여하고, 학우들의 성적 또한 증가시킬 수 있는 좋은 시스템이라고 생각한다.

이처럼 ‘상명 알리미’ 봇은 상명대학교 학생들이 비대면 강의 수업을 듣게 되는 동안 활용할 수 있는 가장 효율적인 시스템이라고 생각한다. 이 프로젝트는 방학 동안 진행되어 1학기 때의 정보를 바탕으로 스크레이핑을 진행하였다. 따라서 9월이 되어 개강을 하게 되었을 때, 현재 코드에서 url의 주소와 크롤링 할 selector만 수정하면 실제로 학기 중에 사용할 수 있는 시스템이다. ‘명령어 하나만 입력하면 되는 간단함’과 ‘한눈에 볼 수 있다는 편리함’이 두 가지의 가장 큰 장점들 때문에 실제 학생들이 많이 사용할 것이라고 예상한다.

그러므로 시스템의 완결성을 더 높이기 위해 이전에 입력했던 채팅을 지워주는 청소 기능과 같은 명령어의 종류도 추가하고, URL 에러뿐만 아니라 로그인 정보를 잘못 입력했을 때 발생하는 에러 등 다양한 에러를 해결할 수 있도록 하여 학기 중에도 프로젝트를 꾸준히 진행해보고자 한다. 또한, 비동기 프로그래밍을 아직 완벽히 익히지 못해 현재 프로그램은 효율성이 조금 떨어지는데,

2학기에도 꾸준히 공부하고, 프로젝트를 보완하여 더 완벽한 프로그램을 만들고자 한다. 2학기 중간고사가 끝나기 전에 프로젝트를 완료하여 실제로 다양한 학과, 다양한 학번의 학우들에게 ‘상명 알리미’를 소개할 예정이다.

## 4. 참고 문헌

- =====
1. 라이언 미젤(2019), 파이썬으로 웹 크롤러 만들기, 한빛미디어
  2. 왕초보 웹크롤링 무작정 따라하기, “크롤링”, <https://charimlab.tistory.com>, (2021.07.07)
  3. [Python] Discord 봇 만들기-시작, “디스코드 봇 만들기”, <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogID=6116949&logNo=221901926848>, (2021.08.11.)
  4. 디스코드 봇 만들기, “Heroku”, <https://lektion-von-erfolglosigkeit.tistory.com/97>, (2021.08.11.)