



---

# I Am Your APP (IAYA)

건양인을 위한 APP

<개발 완료 보고서>

팀명	멋쟁이 괴짜들
작성자	신준호, 성현석, 정권호, 원우영, 권택상
최종 작성일	2020.06.16

# 목 차

---

## 1. 제안

- 1.1 팀 구성 및 역할 선정
- 1.2 회의 시간 및 방식 선정
- 1.3 문제 인식 및 주제 선정
- 1.4 프로젝트 바탕 숙지

## 2. 구상 & 분석

- 2.1 프로젝트 목표 인식
- 2.2 콘텐츠 개요 작성

## 3. 설계

- 3.1 프로젝트 목표 선정
- 3.2 프로그램 개발 방향 선정
- 3.3 개발 툴 학습 – 파이참, pyQT, crawling

## 4. 구현 및 메뉴얼

- 4.1 개발 환경 디자인 구현
- 4.2 디자인
- 4.3 구현내용

## 5. 테스트

- 단위 테스트 및 통합 테스트

## 6. 완료

- 6.1 자료 통합
- 6.2 실행 파일 구현
- 6.3 프로젝트 완료 소감

## 1. 제안

### 가. 팀 구성 및 역할 선정

신준호(팀장)	디자인, 약도 크롤링
성현석(팀원)	디자인, 공지사항 크롤링
정권호(팀원)	코로나, 영화 순위 크롤링
원우영(팀원)	날씨, IT뉴스 크롤링
권택상(팀원)	식단, 연락처 크롤링

### 나. 회의 방식 및 시간 선정

마이크로소프트 teams(온라인), 대면회의(오프라인)

1차 회의	2020.05.21	주제선정, 요구사항 분석
2차 회의	2020.05.25	디자인 Tool 선정, 아이디어 시각화
3차 회의	2020.06.12	구현 전략 회의, 관련 자료 학습
4차 회의	2020.06.14	설계-구현 보완점 파악 및 개선
5차 회의	2020.06.15	크롤링 코드 종합, 오류수정, ppt 제작
6차 회의	2020.06.16	최종 결과 보고서 작성 및 발표 진행 회의

### 다. 개요 및 목적

개요	구성원들의 공통 관심사 또는 찾아보는 빈도가 높은 정보들을 종합하여 하나의 프로그램에서 그러한 모든 정보들을 볼 수 있게 만들었습니다.
목적	1. 건양대 내부 약도와 현재 위치에서 건양대 까지 가는 경로를 알 수 있습니다. 2. 건양대 의료공과대학의 학부와 학과 그리고 주요 민원부서의 전화번호를 알 수 있습니다. 3. 건양대 공지사항을 알 수 있습니다. 4. 네이버 누적 예매율 top5 영화에 대한 정보를 알 수 있습니다. 5. 건양대 죽헌정보관과 건양회관의 금주 식단표를 알 수 있습니다. 6. 네이버 최신 헤드라인 IT 기사 top3를 알 수 있습니다.

### 라. 문제인식 및 소주제 선정 배경

- 문제인식 : 팀원들끼리 건양대 학생으로서 불편했던 점과 필요한 내용을 고민했습니다.

건양대 주요 연락처와 학과 연락처, 건양대 약도와 찾아오는 길, 건양대 공지사항, 건양대 식단을 일일이 찾아봐야 하는 점이 불편했습니다.

네이버 IT 기사 소식, 오늘의 날씨, 현재 코로나 정보, 현재 상영하는 영화 정보를 일일이 찾아봐야 하는 점이 불편했습니다.

- 프로젝트 소주제 선정 배경 : 크롤링의 장점인 실시간 변화하는 정보를 검색하는 것을 이용하여 구성원이 자주 접하고 필요로 하는 정보들을 한곳에 모아서 제공해주는 크롤러 프로그램을 만드는 것을 소주제로 선정

## 2. 구상 & 분석

### 프로젝트 목표 인식

팀원들 각자가 원하는 그리고 관심이 있는 공통의 정보를 하나의 프로그램 안에서 한 눈에 볼 수 있도록 만들었습니다.

### 콘텐츠 개요 작성

메인화면	
타이틀	프로그램 이름(I am your APP)
코로나 확진자 현황	코로나 관련 전일대비 증감, 누적 확진환자, 누적 격리해제, 격리 중, 사망자 정보
오늘의 날씨	오늘의 강수확률, 기온, 날씨정보
오늘 뭐 볼까?	네이버 주간 영화 순위
건양대 학식	논산, 대전, 학식 정보
IT기사	네이버 상위 IT 기사 정보 3개
건양대 정보	
건양대 약도	입력한 현재 위치에서 건양대까지의 자동 길찾기
건양대 연락처	의료공대, 교무팀, 장학팀, 학생지원팀, 인성관 전화번호
건양대 공지사항	학사, 장학, 학생, 취업, 신청, 채용, 일반 건양대 각종 공지

### 3. 설계

#### 프로젝트 목표 선정

저희 팀은 이번 설계 프로젝트를 진행함에 있어서 저희의 1학년 때의 경험을 통해 필요하다고 생각하는 홈페이지의 내용들과 현재 저희에게 필요하다고 생각되어지는 내용들을 하나의 프로그램에 모아서 일상생활 속 정보 수집의 번거로움을 조금 더 편하게 만들어줄 프로그램을 작성하는 것을 중점으로 두었습니다.

이번 설계 프로젝트는 “프로젝트를 통해 정보 수집의 번거로움을 덜 수 있는지”, “사용자가 필요로 하는 프로그램을 제공할 것인가”에 대한 부분에 중점을 두었습니다. 전체적인 프로젝트 목표에 대한 팀원들 간의 토의 및 브레인스토밍을 통하여 설계 과정을 구체적으로 문서화하며, 프로젝트의 대한 실행결과를 지속적으로 관찰, 개발하는 것으로 선정하였습니다.

프로젝트의 개발은 각자 역할을 분담하고 이후 내용들을 전체적으로 병합하여 최종적으로는 전체적인 코드의 오류 발생을 처리하고 개발된 프로그램을 문서화하는 작업을 거치는 것을 목표로 하였습니다

#### 프로그램 개발 방향 선정

프로그램으로 하여금 웹에 흩어져 있는 여러 가지의 정보들을 웹 크롤링을 통하여 수집, 분석, 처리하여 하나의 디자인되어진 화면에 출력하고, 그로 인하여 프로그램을 사용하는 사용자가 필요한 정보들을 한눈에 혹은 최소한의 클릭을 통하여 확인할 수 있도록 선정하였습니다.

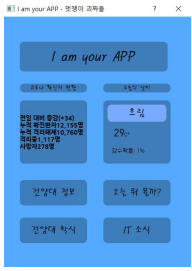
필요 이유	일상생활 속에서 자주 찾아보는 정보를 각각의 검색을 통하여 수집해야 하고, 건양대 관련 필요한 정보들은 신입생들이 아닐지라도 잘 알지 못하는 사람들이 존재하기 때문에 그러한 사용자들을 위해서 필요 정보들을 보기 쉽게 화면에 출력하여 편리성을 추구하는 데에 있습니다.
개발 방향	프로그램을 실행함에 있어서 네이버 페이지와 건양대 홈페이지에 대한 크롤링을 통하여, 최근 자주 검색하게 되는 내용인 코로나와 원래 사람들이 자주 찾아보는 내용인 날씨를 메인 화면에 띄우도록 하며, 건양대 관련 정보와 IT기사, 영화 추천 내용들은 버튼의 클릭을 통하여 볼 수 있도록 합니다. 또한, Selenium을 통한 홈페이지 매크로를 통하여 영화와 학교 식단, IT뉴스에 대한 정보들을 크롤링하여 최신 정보와 그에 대한 이미지를 가져올 수 있도록 방향을 설정하였습니다.

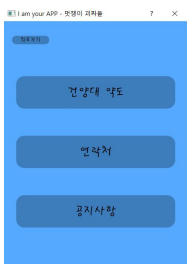
#### 개발 툴 학습 – 파이썬, pyQT, crawling

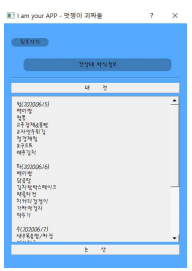
Crawling의 경우에 강의 학습을 통한 내용을 토대로 복습하여 학습하고, 잘 알지 못하는 크롤링 코드들에 대해서는 이번 학기에 처음 접하게 된 프로그램인 Pycharm과 pyQT는 검색을 통하여 학습하고, 부족한 부분은 영상을 통하여 학습하였습니다.

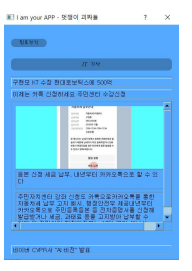



< 구현 내용 >


 <p>main</p>	코로나 현재 상태	금일 확진자, 누적 확진자 수, 사망자 등의 정보를 크롤링하여 코로나 라벨에 출력한다.
	기상 상태	현재 온도 및 날씨 상태, 강수량 등을 구글, 네이버에서 크롤링하여 가져온 뒤 그것을 각각의 라벨 에 출력한다.
	건강대 정보 버튼	각각의 Form으로 이동할 수 있는 버튼이며 각각의 Form은 패널로 이루어져 있어 그 크기를 resize 하는방식으로 열고 닫을 수 있다.
	건강대 학식 버튼	
	오늘 뭐먹지 버튼	
	IT뉴스 버튼	

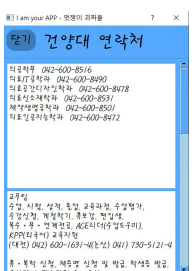
 <p>공지사항</p>	건강대 약도 버튼	각각의 영역의 Form을 열어주는 버튼이며 그 폼은 main을 부모로 하기 때문에 main이 닫히면 저절로 종료된다.
	연락처 버튼	
	공지사항 버튼	
	뒤로가기 버튼	공지사항 패널을 resize해서 크기를 0으로 줄임으로 다시 main창이 나올수 있게 한다.


 <p>오늘의 식단</p>	page-대전	셀레니움을 통해서 건강대 식단 사이트에 들어가서 식단 데이터 크롤링하여 각각 대전, 논산 식단의 데이터를 page의 식단 라벨에 저장한다.
	page-논산	
	Toolbox	각 페이지 공간을 나눠주는 역할을 하며 그 페이지는 대전, 논산이다
	뒤로가기 버튼	오늘의 식단 패널을 resize해서 크기를 0으로 줄임으로 다시 main창이 나올수 있게 한다.

 <p>it 뉴스</p>	page-news1	네이버 뉴스의 제목을 크롤링하여 각 페이지의 이름으로 설정하고 그 페이지 안에는 각각 라벨이 3개가 있으며 하나는 사진이 들어가는 라벨,본 내용을 축약시킨 본문 라벨,본 내용이 들어가는 라벨으로 나누어져 있다. 이 라벨들은 선택된 뉴스의 주소를 크롤링하여 그 속의 사진, 내용 등을 크롤링하여 가지고 온다.
	page-news2	
	page-news3	
	뒤로가기 버튼	it뉴스 패널을 resize해서 크기를 0으로 줄임으로 다시 main창이 나올수 있게 한다.

 <p>영화 추천</p>	page-영화 추천1	<p>스택패널을 이용하여 각각의 페이지들을 분리하고 페이지마다 서로 다른 영화를 추천할 수 있도록 구간을 나눈다. 영화 추천 데이터는 네이버 영화 페이지에서 크롤링하여 1~5 순위까지의 데이터를 끌어모아서 영화 이름, 예매율, 사진 등을 각각의 라벨에 출력한다.</p>
	page-영화 추천2	
	page-영화 추천3	
	page-영화 추천4	
	page-영화 추천5	
	다음 버튼	현재 보고있는 스택패널의 페이지를 다음 페이지로 이동시킨다.
	이전 버튼	현재 보고있는 스택패널의 페이지를 이전 페이지로 이동시킨다.
	뒤로가기 버튼	영화추천 패널을 resize해서 크기를 0으로 줄임으로 다시 main창이 나올수 있게 한다.

 <p>건양대 약도</p>	검색 버튼	셀레니움으로 구글 길찾기 사이트를 불러와서 자동으로 내 현재 위치와 건양대병원의 위치를 입력시키고 길찾기를 실행한다.
	닫기 버튼	건양대 약도 Form을 종료한다.

 <p>건양대연락처</p>	연락처 Text Edit	Text Edit에 건양대학교의 자주 쓰여지는 연락처들을 크롤링하여 출력한다.
	닫기	건양대 연락처 Form을 종료한다.

 <p>공지사항</p>	Tap Area	학사, 장학,신청,취업 등 각각의 공간을 할당하게 해준다.
	Tap1- 학사	<p>건양대의 학사, 장학 등등의 주소들을 크롤링 하여 그 주소에서 제목을 따고, 그 제목을 누르면 이동하는 주소와 함께 하이퍼 링크로 구현하여 라벨에 저장한다. 라벨을 클릭할 시 그 라벨의 하이퍼 링크 주소로 연결된다. 각각의 Tap는 10개의 하이퍼 링크 라벨로 구현이 되어 있다.</p>
	Tap2- 장학	
	Tap3- 학생	
	Tap4- 취업	
	Tap5- 신청	
	Tap6- 채용	
	뒤로가기 버튼	공지사항 Form을 종료한다.



## 5. 테스트

---

### 단위 테스트 및 통합 테스트

2020.05.26 : 파이참 및 qt designer

성현석: 메인화면, 코로나, 날씨, 네이버 영화, 건양대 식단, 네이버 IT기사 페이지 디자인 제작

신준호: 건양대 정보, 건양대 약도, 건양대 공지사항, 건양대 연락처 페이지 디자인 제작

각자 만든 페이지 들을 파이참을 이용하여 하나로 이어서 프로그램으로 구현하여 테스트

2020.06.12 : 파이참 및 qt designer

성현석: 크롤링에 맞춘 디자인의 label 크기 버튼 효과 등 수정 및 테스트

원우영: 디자인 페이지를 이용하여 날씨, IT기사 정보에 대하여 파이참에 크롤링 코드 구현 및 테스트

권택상: 디자인 페이지를 이용하여 식단, 연락처 정보에 대하여 파이참에 크롤링 코드 구현 및 테스트

정권호: 디자인 페이지를 이용하여 코로나, 영화 순위 정보에 대하여 파이참에 크롤링 코드 구현 및 테스트

2020.06.14 : 파이참 및 qt designer

성현석: 디자인의 label 크기 버튼 효과 등 수정 및 테스트 및 공지사항 크롤링

원우영: 디자인 페이지를 이용하여 날씨, IT기사 정보에 대하여 파이참에 크롤링 코드 구현 및 반복적인 테스트

권택상: 디자인 페이지를 이용하여 식단, 연락처 정보에 대하여 파이참에 크롤링 코드 구현 및 반복적인 테스트

정권호: 디자인 페이지를 이용하여 코로나, 영화 순위 정보에 대하여 파이참에 크롤링 코드 구현 및 반복적인 테스트

신준호: 디자인 페이지를 이용하여 약도 크롤링 및 IT뉴스 디자인 수정 및 테스트

2020.06.15 : 파이참 및 qt designer 및 exe 파일 생성 및 프로그램 구현 완료

성현석: 건양대 공지사항 크롤링 코드 재수정, 크롤링 코드 결합 및 수정

원우영: IT뉴스 페이지 기존 코드에 대한 예외사항 처리 및 수정 및 테스트

권택상: 디자인 페이지를 이용하여 식단, 연락처 정보에 대하여 파이참에 크롤링 코드 구현 및 반복적인 테스트

정권호: 영화 정보 페이지 수정 및 테스트, 연락처 정보 협업

신준호: 모든 코드 결합 협업 및 exe 파일 생성 및 테스트

## 6. 완료

### < 자료 통합 >

# .py 파일 통합

```
1  import ...
19
20  ui = uic.loadUiType("./main.ui")[0]
21  uiMap = uic.loadUiType("./KonyangUniversityMap.ui")[0]
22  uiInfo = uic.loadUiType("./KonyangInfo.ui")[0]
23  uiCon = uic.loadUiType("./KonyangContract.ui")[0]
24
25  class CoronaPanel:...
47  def indicateMoviePanel():...
65  class WeatherBox:...
12  class ITNews:...
18  class MealTable:...
85  class MakeMoviePanel:...
12  def createObject():...
19  createObject()
28  class windows(QDialog,ui):...
41  class Map_Area(QDialog, uiMap):...
77  class Info_Area(QDialog,uiInfo):...
46  class Con_Area(QDialog,uiCon):...
63
64  app = QApplication(sys.argv)
65  w1 = windows()
66  w1.show()
67  app.exec()
68
```

과정 및 순서

A : 단일 테스트를 통한 각 모듈의 기능 점검








B : 모듈의 부분 통합 후 에러 디버깅

C : 전체 모듈 통합 후 테스트

단일 테스트를 거쳐 성공적으로 빌드가 된 모듈들을 하나씩 통합시키고 과정에서 에러가 나는 부분은 디버깅을 통해 다시 수정 후 테스트, 이후 전체 모듈을 하나의 .py 코드로 통합하였음.

전체 모듈을 통합하여 실행된 프로그램의 기능을 점검하고 이상없이 작동하는 것을 확인한 뒤 프로그램의 실행파일 제작 단계로 이동

## 2. .ui & 프로그램 확장

	KonyangContract.ui	2020-06-15 오후 8:15	UI 파일	9KB
	KonyangInfo.ui	2020-06-15 오후 8:15	UI 파일	22KB
	KonyangUniversityMap.ui	2020-06-15 오후 8:15	UI 파일	6KB
	main.ui	2020-06-15 오후 8:24	UI 파일	49KB
	foreicon.ico	2020-06-15 오후 8:56	아이콘	5KB
	icon.jpg	2020-06-15 오후 6:20	일체 JPG 파일	6KB
	chromedriver.exe	2020-05-01 오후 5:56	응용 프로그램	8,666KB

디자이너로 구현한 .ui 파일 및 기본적으로 실행파일로 만들기 위해 필요한 필수 파일을 묶어 응용프로그램(.exe)으로 확장하기위한 포맷을 생성

## < 실행 파일 구현 >

```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\IL\Desktop\최종의 최종 의 최종 >
```














pyinstaller 설치 후 pycharm 하단 terminal 클릭

```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\IL\Desktop\최종의 최종 의 최종 >pyinstaller --icon=foreicon.ico --onefile main.py
```

pyinstaller --icon=foreicon.ico --onefile main.py 입력

foreicon.ico = icon 파일 형식

main.py = pycharm 코드 파일

	.idea	2020-06-16 오후 8:40	파일 폴더	
	__pycache__	2020-06-15 오후 9:21	파일 폴더	
	build	2020-06-15 오후 9:20	파일 폴더	
	dist	2020-06-15 오후 9:30	파일 폴더	
	main.py	2020-06-15 오후 9:19	JetBrains PyCharm	30KB
	main.spec	2020-06-15 오후 9:22	SPEC 파일	2KB
	KonyangContract.ui	2020-06-15 오후 8:15	UI 파일	9KB
	KonyangInfo.ui	2020-06-15 오후 8:15	UI 파일	22KB
	KonyangUniversityMap.ui	2020-06-15 오후 8:15	UI 파일	6KB
	main.ui	2020-06-15 오후 8:24	UI 파일	49KB
	foreicon.ico	2020-06-15 오후 8:56	아이콘	5KB
	icon.jpg	2020-06-15 오후 6:20	알씨 JPG 파일	6KB
	chromedriver.exe	2020-05-01 오후 5:56	응용 프로그램	8,666KB

dist파일안의 폴더 확인후 main.exe 파일 확인 - > exe파일 생성완료 및 실행파일 구현 완료

## 프로젝트 완료 소감

인원	소감내용
신준호	대학교 들어온 이래로 팀 프로젝트를 학교에서 진행한 적은 있으나 계획한 것을 실제로 구현 해본적은 없었습니다. 그래서 이번에 처음 팀의 팀장을 해보았고 팀프로젝트를 구현 까지 완료 하였습니다. 이 과정에서 의견 충돌도 있었으나 팀장으로서 팀원들의 의견을 하나로 모아서 프로젝트를 완료 구현 까지 해보니 다음 프로젝트를 진행할 때는 좀 더 잘 할 수 있을거라는 확신이 들었고 팀프로젝트로 처음 제대로 만들어본 프로그램이라 뿌듯하다는 생각도 들었습니다. 제가 디자인을 맡았으나 QT디자인을 따로 배워 본적은 없어서 하나하나 인터넷에서 찾아서 배워서 디자인을 현석이와 구현하는 과정을 통해서도 이러한 식으로 다른 프로그래밍 언어들도 익힐 수 있고 또 QT 디자이너를 앞으로 활용 할 수도 있을 것이라는 생각이 들었습니다. 그리고 프로그래밍 특히 디자인 부분을 구현 할 때 정말 재미있고 한번 하면 제대로 빠져들 수 있을거라는 확신이 들었습니다.
성현석	이번 프로젝트를 통해서 제대로 된 프로그램을 만들어 낸 것 같아서 재미있었습니다. 처음다뤄보는 QT 디자이너도 조장과 함께 상의를 해보면서 고민하고 머리를 쥐어 짜내서 하는 것에 조금 힘들기도 했지만 그것도 결국 성공해낸 것에 대해서 성취감은 매우 컷습니다. 그리고 디자인을 끝낸 뒤 조원들의 크롤링을 도와주었을때 하이퍼 링크를 만드는 것에 어려움이 있었는데. 조원과의 의견을 서로 주고받고 서로 찾아가면서 결국 성공해 낸 것에 대해서 기뻐합니다. 그렇게 다음 회의 에서 서로 모든 걸 끝내고 서로가 맡았던 구간들을 이어붙이는 것도 몇가지 오류가 생기긴 했지만 그 오류들을 해결해가면서 모든 것을 해결했을 때 안도의 한숨과 이 프로젝트를 끝냈다는 행복이 너무 향홀했습니다. 어렵기는 했지만 성취감도 그만큼 커서 빠져나가기 힘든 그런 감정을 이번 프로젝트를 통해서 배운 것 같습니다.
정권호	이번 프로젝트를 진행함에 있어서 팀원들과 토의를 통해 주제를 선정하고, 하나의 프로그램을 개발해 나가면서 발생하는 여러가지 오류들을 파악하여 개선하고, 발전시켜나아가는 과정에서 여러가지 많은 점들을 배울 수 있었습니다. 우선 이번 설계 수업을 수강하면서 1학년 때에는 가볍게 배웠던 파이썬에 대하여 좀 더 문법적인 구조, 여러 가지 기능들에 대해서 학습하고 실행해보면서 파이썬에 대해서 더 자세하게 알아갈 수 있었고, 이번 프로그램의 개발을 참여, 진행하면서 수업시간에 배운 내용들을 직접 사용하면서 부족한 부분은 더 보완되어지고 알고 있던 부분들은 더 단단하게 굳어지게 되었습니다. 모든 팀원들이 그렇겠지만 짧다면 짧고 길다면 긴 시간동안에 만든 이 프로그램에 굉장히 애착이 가고 한편으로는 다음에 하게 될 프로그램의 개발을 좀 더 자신감 있게 진행해 나갈 수 있을 것 같습니다.

원우영	<p>크롤링이라는 정보처리 기법을 이용해 내가 원하는 데이터를 추출하고 가공하여 새로운 정보를 만들어 보았던 이번 프로젝트는 나에게 있어서 웹에 업로드되는 데이터의 구조를 파악하고, 다양한 데이터들을 분석하고 처리하는 과정을 조금이나마 이해하고 학습하는 데에 큰 도움이 되었다고 생각한다.</p> <p>이번에 팀으로 설계를 진행하면서 인상적이었던 것은 전체에 대한 일부 작업을 소분하여 팀원 각자가 담당하였다. 그리고 각자의 의견을 내고 아이디어를 공유하여 개선해야 할 점, 부족한 점들을 서로의 피드백을 통해 채우며 작업이 진행되었다. 두 과정을 통해 부분에 대한 집중적인 시간 할애로 작업 효율이 증가하였고, 프로젝트의 전반적인 측면에서 개인의 역량으로 갖는 기대 이상의 결과물을 얻는 데에 크게 작용했다고 생각한다. 무엇보다도 팀 구성원 모두가 맡은 바 역할을 성실히 수행했을 때 비로소 이러한 효과가 크게 나타난다는 점을 이번 프로젝트를 진행하면서 다시금 인식할 수 있었다는 점에서 의미 있는 시간이었다.</p>
권택상	<p>대학 복학 후 첫 프로젝트를 다른 팀원들과 함께 진행하였습니다.</p> <p>코로나로 인해 제한되는 점이 많아서 교수님과 수업 중 자세한 내용이 교류되지 못한 점이 걱정되고 아쉬웠지만 부족한 점은 팀원들과 서로 의지하며 문제를 해결해 나갈 수 있었습니다.</p> <p>팀원으로서 단기 프로젝트를 진행하였는데 처음 사용하는 개발환경이 익숙하지 않아서 불편했지만 교수님 수업 외의 어려운 내용은 구글 검색이나 유튜브 강의를 찾아보고 팀원들과 정보 공유를 통해 문제를 해결해 나갈 수 있었습니다.</p> <p>팀원들과 프로젝트를 진행하면서 다른 팀원들의 작업 방식들 중 유익한 방법들을 배울 수 있었고 저의 실수와 팀원들의 실수를 서로 찾아줌으로써 프로젝트에서는 공동체 의식이 매우 중요하다는 것을 알게 되었습니다.</p> <p>2개월 정도 진행되었던 비교적 짧은 프로젝트였지만 첫 정식 프로젝트로써 프로젝트에 임하는 자세를 배울 수 있었고 Pyqt5, Pycharm과 같은 새로운 개발환경을 경험함으로써 다음 프로젝트를 진행하는데 좋은 자양분이 될 것 같습니다.</p>

```

import sys
import time
import PyQt5
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5 import uic
from bs4 import BeautifulSoup
import requests
from urllib.request import urlopen
from urllib.request import urlretrieve
from urllib.parse import quote_plus
from selenium import webdriver
from PyQt5.QtGui import QDesktopServices
from PyQt5.QtCore import QUrl
import urllib.request
from PyQt5.QtGui import QPixmap
from selenium.webdriver.common.keys import Keys

# 기본 Ui
ui = uic.loadUiType("./main.ui")[0]
uiMap = uic.loadUiType("./KonyangUniversityMap.ui")[0]
uiInfo = uic.loadUiType("./KonyangInfo.ui")[0]
uiCon = uic.loadUiType("./KonyangContract.ui")[0]

#COVID_19 코로나 구현
page =
requests.get('http://ncov.mohw.go.kr/bdBoardList_Real.do?brdld=1&brdGubun=13&ncv
ContSeq=&contSeq=&board_id=&gubun=')
soup = BeautifulSoup(page.content, 'html.parser')

data = soup.find('div', class_='mapview')

data_text = data.find_all('span', class_='tit')
data_value = data.find_all('span', class_='num')
data_value_num = data.find_all('em')

data_sub_text = data.find('span', class_='sub_tit red')
data_sub_num = data.find('span', class_='sub_num red')

for i in range(len(data_text)):
    corona_today = data_sub_text.get_text() + data_sub_num.get_text()
    coronaA = data_text[0].get_text() + data_value[0].get_text() +
data_value_num[0].get_text()
    coronaB = data_text[2].get_text() + data_value[2].get_text() +
data_value_num[2].get_text()
    coronaC = data_text[1].get_text() + data_value[1].get_text() +
data_value_num[1].get_text()
    coronaD = data_text[3].get_text() + data_value[3].get_text() +
data_value_num[3].get_text()
    corona = corona_today + '\n' + coronaA + '\n' + coronaB + '\n' + coronaC + '\n'
+ coronaD

# 영화 구현
page =
requests.get('https://movie.naver.com/movie/running/current.nhn?view=image&tab=n

```

```

ormal&order=reserve')
soup = BeautifulSoup(page.content, 'html.parser')

movie = soup.find('div', class_='lst_wrap')

movie_name = movie.find_all('strong', class_='tit') #영화제목
movie_num = movie.find_all('span', class_='dsc') # 영화별에매울
#movie_image = movie.find_all('img', onerror_='this.src') #영화이미지

movie_result = [0, 1, 2, 3, 4]

for i in range(len(movie_name)):
    if(i <= 4):
        movie_result[i] = movie_name[i].get_text() + '\n' +
movie_num[i].get_text(": ")
    else:
        break

class WeatherBox:
    naver =
'https://search.naver.com/search.naver?sm=tab_hyt.top&where=nexearch&query='

    keyword = quote_plus('내위치 날씨')
    url = naver + keyword
    page = requests.get(url)
    soup = BeautifulSoup(page.content, 'html.parser')

    google = 'https://www.google.com/search?q='
    url2 = google + keyword
    page2 = requests.get(url2)
    soup2 = BeautifulSoup(page2.content, 'html.parser')

    #온도 수치 구하기
    def WeatherValue(self):
        # 온도 값 가져오기
        soup = self.soup
        temperature = soup.select('div > p > span.todaytemp')
        return temperature[0].get_text()
    #celsius 구하기
    def WeatherCelsius(self):
        soup = self.soup
        tmp = soup.select('div > p > span.tempmark')
        s = tmp[0].get_text()
        tempmark = s[2:]
        return tempmark

    #맑음, 비, 흐림 등등
    def WeatherState(self):
        soup = self.soup
        state = soup.select_one('.cast_txt')
        s = str(state.get_text())
        pos = s.find(',')
        weather = s[0:pos]
        return weather

    # 강수확률 구하기
    def rainProb(self):
        browser = webdriver.Chrome('./chromedriver.exe')
        browser.get(self.url2) #구글 내위치 날씨 열기

```

```

        time.sleep(2)
        html = browser.page_source
        soup = BeautifulSoup(html)
        datas = soup.select('.wob-dtl > div')
        # soup = self.soup2
        # datas = soup.select('.wob-dtl > div')
        return datas[0].get_text()

class ITNews:
    url = 'https://news.naver.com/main/main.nhn?mode=LSD&mid=shm&sid1=105'
    page = requests.get(url)
    soup = BeautifulSoup(page.content, 'html.parser')

    def getHeadline(self, index):
        soup = self.soup
        # 헤드라인 앞, 뒤 크롤링
        headF = soup.select('div.cluster_head > div > div > h2 > a >
span:nth-child(1)')
        headR = soup.select('div.cluster_head > div > div > h2 > a >
span:nth-child(2)')

        headline = []
        li1 = []
        li2 = []

        # 헤드라인 앞부분 리스트 저장
        for data in headF:
            li1.append(data.get_text())

        # 헤드라인 뒷부분 리스트 저장
        for data in headR:
            li2.append(data.get_text())

        # 헤드라인 앞, 뒤 묶음 리스트 저장
        for i in range(len(li1)):
            headline.append(li1[i] + li2[i])

        return headline[index]

    def getHeadLink(self):
        soup = self.soup
        headlink = soup.select('div.cluster_head > div > div > h2 > a')
        results = []
        prefix = 'http://news.naver.com'
        for data in headlink:
            link = data.attrs['href']
            results.append(prefix + link)
        return results

    def getArticleLink(self):
        # 각 헤드라인의 첫번째 기사를 스크랩
        soup = self.soup
        titles = soup.select('ul.cluster_list')
        linklist = []
        # 기사 링크 가져오기
        for data in titles:
            link = data.a['href']
            linklist.append(link)

```



```

        return linklist

def getImageLinks(self, linklist):
    imgLinks = []

    # 링크 접속 후 사진 가져오기
    for i in range(len(linklist)):
        try:
            eachNewsUrl = linklist[i]
            eachNewspage = requests.get(eachNewsUrl)
            soup = BeautifulSoup(eachNewspage.content, 'html.parser')
            imglink = soup.select('span > img')
            imgsrc = imglink[0].attrs['src']
            imgLinks.append(imgsrc)
            #이미지 없는 기사 만났을 시 임의 이미지로 대체
        except:

imgLinks.insert(i, 'https://www.seekpng.com/png/full/423-4235598_no-image-for-noi
mage-icon.png')
    return imgLinks

def getArticleImage(self, index): #imagelinks = getImageLinks
    targetImageLink = self.getImageLinks(self.getArticleLink())[index]
    imageFromWeb = urllib.request.urlopen(targetImageLink).read() #웹 이미지
url을 읽어서 파일로 열기
    ArticleImage = QPixmap()
    ArticleImage.loadFromData(imageFromWeb)
    ArticleImage.scaledToHeight(160)

    return ArticleImage

def getArticleTitle(self, linklist, index):
    eachNewsTitles = []
    # 링크에서 뉴스 제목 가져오기
    for i in range(len(linklist)):
        eachNewsUrl = linklist[i]
        eachNewspage = requests.get(eachNewsUrl)
        soup = BeautifulSoup(eachNewspage.content, 'html.parser')
        title = soup.select('h3')
        s = str(title[1].get_text())
        title = s[:30] + '\n' + s[30:] #줄바꿈문자 추가
        eachNewsTitles.append(title)

    return eachNewsTitles[index]

def getArticleContents(self, linklist, index, MaxScrapSize):
    eachNewsContents = []
    for i in range(len(linklist)):
        eachNewsUrl = linklist[i]
        eachNewspage = requests.get(eachNewsUrl)
        soup = BeautifulSoup(eachNewspage.content, 'html.parser')
        article = soup.select('#articleBodyContents')

        s = str(article[0].get_text())
        content = s[5:MaxScrapSize] + '...' #앞 줄바꿈 문자 제거 + 끝 중략 삽입
        eachNewsContents.append(content)

    return eachNewsContents[index]

```

```

class MealTable:
    # 식단표 만들기
    def MakeMealTable(self):
        # 대전 식단표
        driver = webdriver.Chrome('./chromedriver.exe')

driver.get('https://www.konyang.ac.kr/prog/sikdan/kor/sub06_06_03/list.do')
time.sleep(3)

source = driver.page_source
soup = BeautifulSoup(source, 'html.parser')

meals = soup.find(class_='basic_table center')
meals_date = meals.find_all('th')
meals_meal = meals.find_all('td')

list = [0,1,2,3,4]
listMeals = []

for i in range(5):
    list[i] = meals_date[i + 1].get_text() + '\n' + meals_meal[i +
7].get_text('\n') + '\n'
    meal_table1 = list[0] + '\n' + list[1] + '\n' + list[2] + '\n' + list[3]
+ '\n' + list[4]

    # 논산 식단표
    listMeals.append(meal_table1)
    # 논산 식단 크롤링 위한 옵션 변경
    sikdang = "//option[@value='" + '01' + '"'
driver.find_element_by_xpath(sikdang).click()
time.sleep(3)

source = driver.page_source
soup = BeautifulSoup(source, 'html.parser')

meals = soup.find(class_='basic_table center')
meals_date = meals.find_all('th')
meals_meal = meals.find_all('td')

for i in range(5):
    list[i] = meals_date[i + 1].get_text() + '\n' + meals_meal[i +
7].get_text('\n') + '\n'
    meal_table2 = list[0] + '\n' + list[1] + '\n' + list[2] + '\n' + list[3]
+ '\n' + list[4]
    listMeals.append(meal_table2)
    driver.close()
    return listMeals

# 영화페이지 만들기 클래스
class MakeMoviePanel:
    browser = webdriver.Chrome('./chromedriver.exe')
    browser.implicitly_wait(30)
    url =
'https://movie.naver.com/movie/running/current.nhn?view=image&tab=normal&order=r
eserve'
    browser.get(url)
    block_imgs = browser.find_element_by_css_selector('div.top_poster_area')
    imgs = block_imgs.find_elements_by_css_selector('img')

```

```

results = []

for img in imgs:
    link = img.get_attribute('src')
    results.append(link)
browser.close()
#이미지 URL 가져오기
def loadMovieImageLink(self):
    imagelinks = []
    for data in self.results:
        imagelinks.append(data)
    return imagelinks
#이미지 가져오기
def loadMovieImage(self, mvImageLinks, index):
    urlstring = mvImageLinks[index]
    imageFromWeb = urllib.request.urlopen(urlstring).read()
    QPixmapWebVar = QPixmap()
    QPixmapWebVar.loadFromData(imageFromWeb)
    QPixmapWebVar = QPixmapWebVar.scaledToWidth(200)

    return QPixmapWebVar

# 뉴스, 식단, 영화 객체
newsIt = ITNews(); mealT = MealTable(); myMovie = MakeMoviePanel()

class windows(QDialog,ui):
    # 날씨 클래스 객체 wb 생성
    wb = WeatherBox()
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.show()

    #area open
    self.btn_Main_it_info.clicked.connect(self.itAreaOpen)
    self.btn_Main_movie_info.clicked.connect(self.MvAreaOpen)
    self.btn_Main_school_meal.clicked.connect(self.MealAreaOpen)
    self.btn_Main_konyang_info.clicked.connect(self.infoAreaOpen)

    #back btn
    self.btn_it_info_back.clicked.connect(self.Allclose)
    self.btn_Main_movie_info_back.clicked.connect(self.Allclose)
    self.btn_Main_school_meal_back.clicked.connect(self.Allclose)
    self.btn_info_back.clicked.connect(self.Allclose)

    #konyang info area
    self.btn_Konyang_Minimap_3.clicked.connect(self.MapOpen)
    self.btn_Konyang_info_3.clicked.connect(self.InfoOpen)
    self.btn_Konyang_Contract_3.clicked.connect(self.ConOpen)

    #All Area Close
    self.it_info_Area.setFixedSize(450, 0)
    self.movie_Area.setFixedSize(450, 0)
    self.school_meal_Area.setFixedSize(450, 0)
    self.info_Area.setFixedSize(450, 0)

    #movie area
    self.btn_Main_movie_info_next_1.clicked.connect(self.pageChangedNext)
    self.btn_Main_movie_info_next_2.clicked.connect(self.pageChangedNext)

```

```

self.btn_Main_movie_info_next_3.clicked.connect(self.pageChangedNext)
self.btn_Main_movie_info_next_4.clicked.connect(self.pageChangedNext)
self.btn_Main_movie_info_next_5.clicked.connect(self.pageChangedNext)
self.idx = 0
self.btn_Main_movie_info_before_1.clicked.connect(self.pageChangedBefore)
self.btn_Main_movie_info_before_2.clicked.connect(self.pageChangedBefore)
self.btn_Main_movie_info_before_3.clicked.connect(self.pageChangedBefore)
self.btn_Main_movie_info_before_4.clicked.connect(self.pageChangedBefore)
self.btn_Main_movie_info_before_5.clicked.connect(self.pageChangedBefore)

#corona, movie
self.lbl_Main_corona.setText(corona)
self.lbl_movie_name_1.setText(movie_result[0])
self.lbl_movie_name_2.setText(movie_result[1])
self.lbl_movie_name_3.setText(movie_result[2])
self.lbl_movie_name_4.setText(movie_result[3])
self.lbl_movie_name_5.setText(movie_result[4])

# 영화
movieImages = [self.lbl_movie_image_1, self.lbl_movie_image_2,
self.lbl_movie_image_3, self.lbl_movie_image_4,
                self.lbl_movie_image_5]
for i in range(len(movieImages)):
movieImages[i].setPixmap(myMovie.loadMovieImage(myMovie.loadMovieImageLink(),
i))

#날씨
self.lbl_weather_weather.setText(self.wb.WeatherState())
self.lbl_weather_temp.setText(self.wb.WeatherValue())
self.lbl_weather_rainfall.setText(self.wb.rainProb())

#뉴스
newsimgs = [self.lbl_photo_1, self.lbl_photo_2, self.lbl_photo_3]
newstitles = [self.lbl_contents_1_1, self.lbl_contents_2_1,
self.lbl_contents_3_1]
newscontents = [self.lbl_contents_1_2, self.lbl_contents_2_2,
self.lbl_contents_3_2]
for i in range(3):
    self.page_headline_3.setItemText(i, newsIt.getHeadline(i))
    newsimgs[i].setPixmap(newsIt.getArticleImage(i)) # 이미지 레이블 값
    newstitles[i].setText(newsIt.getArticleTitle(newsIt.getArticleLink(),
i))
    newscontents[i].setText(newsIt.getArticleContents(newsIt.getArticleLink(), i,
300))

#식단
Tables = []
for data in mealT.MakeMealTable():
    Tables.append(data)
self.lbl_school_meal_deajon.setText(Tables[0])
self.lbl_school_meal_nonsan.setText(Tables[1])

def pageChangedBefore(self):
    if(self.idx == 0 or self.idx == -1):
        self.idx = 4
    else:

```

```

        self.idx -= 1
        self.movie_Area_stack.setCurrentIndex(self.idx)

def pageChangedNext(self):
    self.idx += 1
    if((self.idx//6) != 0):
        self.idx = 0
    self.movie_Area_stack.setCurrentIndex(self.idx)

def itAreaOpen(self):
    self.it_info_Area.setFixedSize(450, 630)
def MvAreaOpen(self):
    self.movie_Area.setFixedSize(450, 630)
def MealAreaOpen(self):
    self.school_meal_Area.setFixedSize(450, 630)

def infoAreaOpen(self):
    self.info_Area.setFixedSize(450, 630)

def Allclose(self):
    self.idx = 0 #영화 스택패널 인덱스
    self.movie_Area_stack.setCurrentIndex(self.idx) #스택패널 위치 초기화
    self.it_info_Area.setFixedSize(450, 0)
    self.movie_Area.setFixedSize(450, 0)
    self.school_meal_Area.setFixedSize(450, 0)
    self.info_Area.setFixedSize(450, 0)

def MapOpen(self):
    Map_Area(self)
def InfoOpen(self):
    Info_Area(self)
def ConOpen(self):
    Con_Area(self)

class Map_Area(QDialog, uiMap):

    def __init__(self, parent):
        super(Map_Area, self).__init__(parent)
        self.setupUi(self)
        self.show()

    def loadImageFromWeb(self):
        urlstring =
'http://www.konyang.ac.kr/cyber/images/campus/medi_gyu.png'
        imageFromWeb = urllib.request.urlopen(urlstring).read()
        self.qPixmapWebVar = QPixmap()
        self.qPixmapWebVar.loadFromData(imageFromWeb)
        self.qPixmapWebVar = self.qPixmapWebVar.scaledToWidth(600)
        self.label_Minimap.setPixmap(self.qPixmapWebVar)

        loadImageFromWeb(self)
        self.btn_FindwayToKonyang.clicked.connect(self.FindwayToKonyang)
        self.btn_Minimap_close.clicked.connect(self.thisClose)

    def thisClose(self):
        self.close()

```

```

def FindwayToKonyang(self):
    n = self.btn_CurrentLocation.text()
    browser = webdriver.Chrome("./chromedriver.exe")

browser.get('https://www.google.com/maps/dir///@36.3272011,127.3727549,14z')
    time.sleep(0.1)

    search = browser.find_element_by_css_selector("#sb_ifc50 > input")
    search.send_keys(n)
    search.send_keys(Keys.ENTER)

    d = "건양대학교병원"
    search2 = browser.find_element_by_css_selector("#sb_ifc51 > input")
    search2.send_keys(d)
    search2.send_keys(Keys.ENTER)

class Info_Area(QDialog, uiInfo):
    def __init__(self, parent):
        super(Info_Area, self).__init__(parent)
        self.setupUi(self)
        self.show()

        ## 공지사항 페이지

        # 장학
        page =
requests.get("https://www.konyang.ac.kr/cop/bbs/BBSMSTR_000000000584/selectBoard
List.do")
        soup = BeautifulSoup(page.content, "html.parser")

        lst = []
        hacdata2 = soup.select("td.left > div > span.link > a")
        for data in hacdata2:
            link = data['href']
            lst.append(link)
        lst2 = []
        hacdata3 = soup.select("td.left > div > span.link > a > b")
        for data in hacdata3:
            lst2.append(data.get_text())

        lbl1st =
[self.label_2, self.label_3, self.label_4, self.label_5, self.label_6, self.label_7, s
elf.label_8, self.label_9, self.label_10, self.label_11]

        for i in range(len(lbl1st)):
            lbl1st[i].setOpenExternalLinks(True)
            lbl1st[i].setText('<a
href="https://www.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
            lbl1st[i].linkActivated.connect(self.link)

        page2 =
requests.get("https://www.konyang.ac.kr/cop/bbs/BBSMSTR_000000000585/selectBoard
List.do")
        soup2 = BeautifulSoup(page2.content, "html.parser")

        lbl1st = [self.label_12, self.label_13, self.label_14, self.label_15,

```

```

self.label_16, self.label_17, self.label_18,
        self.label_19, self.label_20, self.label_21]
    lst = []
    jangdata2 = soup2.select("td.left > div > span.link > a")
    for data in jangdata2:
        link = data['href']
        lst.append(link)

    lst2 = []
    jangdata3 = soup2.select("td.left > div > span.link > a > b")
    for data in jangdata3:
        lst2.append(data.get_text())

    for i in range(len(lbllst)):
        lbllst[i].setOpenExternalLinks(True)
        lbllst[i].setText('<a
href="https://www.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
        lbllst[i].linkActivated.connect(self.link)

    page3 =
requests.get("https://www.konyang.ac.kr/cop/bbs/BBSMSTR_000000000883/selectBoard
List.do")
    soup3 = BeautifulSoup(page3.content, "html.parser")

    lbllst = [self.label_22, self.label_23, self.label_24, self.label_25,
self.label_26, self.label_27,
        self.label_28, self.label_29, self.label_30, self.label_31]
    lst = []
    studentdata = soup3.select("td.left > div > span.link > a")
    for data in studentdata:
        link = data['href']
        lst.append(link)

    lst2 = []
    studentdata2 = soup3.select("td.left > div > span.link > a > b")
    for data in studentdata2:
        lst2.append(data.get_text())

    for i in range(len(lbllst)):
        lbllst[i].setOpenExternalLinks(True)
        lbllst[i].setText('<a
href="https://www.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
        lbllst[i].linkActivated.connect(self.link)

    page4 =
requests.get("https://job.konyang.ac.kr/cop/bbs/BBSMSTR_000000001381/selectBoard
List.do")
    soup4 = BeautifulSoup(page4.content, "html.parser")

    lbllst = [self.label_32, self.label_33, self.label_34, self.label_35,
self.label_36, self.label_37,
        self.label_38, self.label_39, self.label_40, self.label_41]
    lst = []
    getjobdata = soup4.select("td.left > div > span.link > a")
    for data in getjobdata:
        link = data['href']
        lst.append(link)

    lst2 = []

```

```

getjobdata2 = soup4.select("td.left > div > span.link > a")
for data in getjobdata2:
    lst2.append(data.get_text())

for i in range(len(lbllst)):
    lbllst[i].setOpenExternalLinks(True)
    lbllst[i].setText('<a
href="https://job.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
    lbllst[i].linkActivated.connect(self.link)

page5 =
requests.get("https://www.konyang.ac.kr/cop/bbs/BBSMSTR_000000000587/selectBoard
List.do")
soup5 = BeautifulSoup(page5.content, "html.parser")

lbllst = [self.label_42, self.label_43, self.label_44, self.label_45,
self.label_46, self.label_47,
        self.label_48, self.label_49, self.label_50, self.label_51]

lst = []
applicationdata = soup5.select("td.left > div > span.link > a")
for data in applicationdata:
    link = data['href']
    lst.append(link)

lst2 = []
applicationdata2 = soup5.select("td.left > div > span.link > a > b")
for data in applicationdata2:
    lst2.append(data.get_text())

for i in range(len(lbllst)):
    lbllst[i].setOpenExternalLinks(True)
    lbllst[i].setText('<a
href="https://www.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
    lbllst[i].linkActivated.connect(self.link)

page6 =
requests.get("https://www.konyang.ac.kr/cop/bbs/BBSMSTR_000000000588/selectBoard
List.do")
soup6 = BeautifulSoup(page6.content, "html.parser")

lbllst = [self.label_52, self.label_53, self.label_54, self.label_55,
self.label_56, self.label_57,
        self.label_58, self.label_59, self.label_60, self.label_61]

lst = []
Hiredata = soup6.select("td.left > div > span.link > a")
for data in Hiredata:
    link = data['href']
    lst.append(link)

lst2 = []
Hiredata2 = soup6.select("td.left > div > span.link > a > b")
for data in Hiredata2:
    lst2.append(data.get_text())

for i in range(len(lbllst)):

```



```

        lbl1st[i].setOpenExternalLinks(True)
        lbl1st[i].setText('<a
href="https://www.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
        lbl1st[i].linkActivated.connect(self.link)

    page7 =
requests.get("https://www.konyang.ac.kr/cop/bbs/BBSMSTR_000000000583/selectBoard
List.do")
    soup7 = BeautifulSoup(page7.content, "html.parser")

    lbl1st = [self.label_62, self.label_63, self.label_64, self.label_65,
self.label_66, self.label_67,
                self.label_68, self.label_69, self.label_70, self.label_71]
    lst = []
    Nonmaldata = soup7.select("td.left > div > span.link > a")
    for data in Nonmaldata:
        link = data['href']
        lst.append(link)

    lst2 = []
    Nonmaldata2 = soup7.select("td.left > div > span.link > a > b")
    for data in Nonmaldata2:
        lst2.append(data.get_text())

    for i in range(len(lbl1st)):
        lbl1st[i].setOpenExternalLinks(True)
        lbl1st[i].setText('<a
href="https://www.konyang.ac.kr{0}">{1}</a>'.format(lst[i], lst2[i]))
        lbl1st[i].linkActivated.connect(self.link)

    self.btn_Info_Close.clicked.connect(self.thisClose)

    def link(self, strlink):
        QDesktopServices.openUrl(QUrl(strlink))
        print(strlink)

    def thisClose(self):
        self.close()

class Con_Area(QDialog, uiCon):
    def __init__(self, parent):
        super(Con_Area, self).__init__(parent)
        self.setupUi(self)
        self.show()
        self.btn_Contract_Close.clicked.connect(self.thisClose)

    ### 의료공대 행정실 & 주요 민원부서 전화번호 (의료공대 행정실, 교무팀, 장학팀,
기타)
    ## 의료공대 행정실 전화번호 모음
    # 의공학부
    page2 = requests.get('https://www.konyang.ac.kr/kor/sub03_05_01_01.do')
    soup2 = BeautifulSoup(page2.content, 'html.parser')

    data2 = soup2.find(class_='content')
    data2_p = data2.find_all('p')
    data2_call = data2.find_all('a')

```

```

BE = data2_p[0].get_text() + ' ' + data2_call[4].get_text()

# 의료 IT 공학과
page3 = requests.get('https://www.konyang.ac.kr/kor/sub03_05_02_01.do')
soup3 = BeautifulSoup(page3.content, 'html.parser')

data3 = soup3.find(class_='content')
data3_p = data3.find_all('p')
data3_call = data3.find_all('a')

MITE = data3_p[0].get_text() + ' ' + data3_call[4].get_text()

# 의료 공간 디자인 학과
page4 = requests.get('https://www.konyang.ac.kr/kor/sub03_05_03_01.do')
soup4 = BeautifulSoup(page4.content, 'html.parser')

data4 = soup4.find(class_='content')
data4_p = data4.find_all('p')
data4_call = data4.find_all('a')

DMSDM = data4_p[0].get_text() + ' ' + data4_call[4].get_text()

# 의료 신소재 학과
page5 = requests.get('https://www.konyang.ac.kr/kor/sub03_05_04_01.do')
soup5 = BeautifulSoup(page5.content, 'html.parser')

data5 = soup5.find(class_='content')
data5_p = data5.find_all('p')
data5_call = data5.find_all('a')

DBM = data5_p[0].get_text() + ' ' + data5_call[4].get_text()

# 제약 생명 공학과
page6 = requests.get('https://www.konyang.ac.kr/kor/sub03_05_05_01.do')
soup6 = BeautifulSoup(page6.content, 'html.parser')

data6 = soup6.find(class_='content')
data6_p = data6.find_all('p')
data6_call = data6.find_all('a')

DPB = data6_p[0].get_text() + ' ' + data6_call[4].get_text()

# 의료 인공지능 학과
page7 = requests.get('https://www.konyang.ac.kr/kor/sub03_05_07_01.do')
soup7 = BeautifulSoup(page7.content, 'html.parser')

data7 = soup7.find(class_='content')
data7_p = data7.find_all('p')
data7_call = data7.find_all('a')

MAI = data7_p[0].get_text() + ' ' + data7_call[1].get_text()

Contract = BE + '\n' + MITE + '\n' + DMSDM + '\n' + DBM + '\n' + DPB +
'\n' + MAI
self.textEdit_Contract1.setText(Contract)

## 주요 민원부서 전화번호

# 교무팀 전화번호

```

```

page = requests.get('https://www.konyang.ac.kr/kor/sub06_09_01.do')
soup = BeautifulSoup(page.content, 'html.parser')

data = soup.find(class_='tbl_basic')
data_th = data.find_all('th')
data_td = data.find_all('td')

gyomu = data_th[3].get_text() + '\n' + data_td[0].get_text() + '\n' +
data_td[1].get_text() \
        + '\n\n' + data_td[2].get_text().replace('\t', ' ') + '\n' +
data_td[3].get_text()
self.textEdit_Contract2.setText(gyomu)

## 장학팀 전화번호
janghak = data_th[5].get_text() + '\n' + data_td[8].get_text() + '\n' +
data_td[9].get_text()
self.textEdit_Contract3.setText(janghak)

# 학생지원팀
jiwon = data_th[4].get_text() + '\n' + data_td[10].get_text() + '\n' +
data_td[11].get_text()
self.textEdit_Contract4.setText(jiwon)

# 인성관
insung = data_th[20].get_text() + '\n' + data_td[38].get_text() + '\n' +
data_td[39].get_text()
self.textEdit_Contract5.setText(insung)

def thisClose(self):
    self.close()

app = QApplication(sys.argv)
w1 = windows()
w1.show()
app.exec()

```