

도와줘! 수강타임

소프트웨어코딩과적용 5팀

김현수
유지윤
유체린
정지석

Concept

어떻게 해야 원하는 강의를
더 쉽게 찾을 수 있을까?

수강신청, 더 편하게 하자!

내가 들어야 하는 이 강의 학생들은
어떻게 평가 했을까?

Naming

 학생들이 만든  응용 프로그램



학생들의 수강신청 고민과 시간을 줄여줄 프로그램

Differentiation

기존의 Everytime 프로그램과 앱은
과목명과 교수명을 통한 검색 밖에 되지 않는 상황

" 수강타임 " 은
교수명, 강의명, 총점, 과제, 팀플, 학점, 출결, 시험...

Everytime에서 강의평을 작성할 때 쓰는 익숙한 필터들을 사용해
내가 원하는 강의를 쓱쓱!
검색할 수 있다.

Function

도와집! 수강타임

— □ ×

①

교수명

강의명

소속

과제

팀플

학점

출결

시험

② 검색

③

④

UPDATE!

	교수	과목명	평점	과제	팀플	학점	출석	시험횟수
1	최민지	ACT	4.57	보통	많음	학점느님	직접호명	없음
2	도선재	ACT	3.69	많음	많음	비율 채워줌	전자출결	없음
3	김홍중	ACT	3.92	보통	많음	비율 채워줌	직접호명	없음
4	김선미	ACT	4.07	보통	많음	비율 채워줌	직접호명	없음
5	최민지	ACT	4.57	보통	많음	학점느님	직접호명	없음
6	도선재	ACT	3.69	많음	많음	비율 채워줌	전자출결	없음
7	최민지	ACT	4.57	보통	많음	학점느님	직접호명	없음
8	김홍중	ACT	3.92	보통	많음	비율 채워줌	직접호명	없음
9	도선재	ACT	3.7	많음	많음	비율 채워줌	전자출결	없음
10	최민지	ACT	4.57	보통	많음	학점느님	직접호명	없음
11	김재경	ACT	3.99	보통	많음	비율 채워줌	직접호명	없음
12	김선미	ACT	4.05	보통	많음	비율 채워줌	직접호명	없음

Source: Client

```
import json_reader
from PyQt5 import QtWidgets
import everytime

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Form = QtWidgets.QWidget()
    ui = everytime.Everytime()
    ui.setupUi(Form)
    Form.show()
    sys.exit(app.exec_())
```

Source: Host

```
from selenium import webdriver
import time
import pyautogui
import json
from bs4 import BeautifulSoup
```

```
# 스크롤 내리기 오랜 시간이 걸리고 다수의 마우스 클릭을 사용하므로 시연 시에는 주석을 안 풋습니다.
...
b = 0
while b <= 2500:
    pyautogui.click()
    time.sleep(0.01)
    b = b + 1
...
```

```
#입력되는 강의의 정보를 하나씩 띄워줍니다.
print(lecture_name, prof, time_table, score, assign, team, grade, attendance, test_count)

list.append(
    {"lecture_name": lecture_name, "prof": prof, "time_table": time_table, "full_link": full_link,
     "score": score, "assign": assign, "team": team, "grade": grade, "attendance": attendance,
     "test_count": test_count})

print(list)
with open('list.json', 'w') as writer:
    #모든 강의의 리스트를 각 강의별로 Dictionary로 저장하고 list의 하나의 요소로써 담아줍니다.
    writer.write(json.dumps(list))
driver.close();
```

Source: json_reader

```
import json # import json module

class Lecture_list:
    result_list = []
    def __init__(self, prof, lect_name, score, assign, team, grade, attendance, test):#리스트를 불러오기 전 각 변수들을 모두 None으로 받아옵니다
        self.prof = prof
        self.lect_name = lect_name
        self.score = score
        self.assign = assign
        self.team = team
        self.grade = grade
        self.attendance = attendance
        self.test = test

    def update(self):#Host가 만들어준 list.json 파일을 열어서 클래스멤버변수로 저장합니다.
        with open('list.json') as json_file:
            json_data = json.load(json_file)
        self.list = json_data
        print("Lecture Data Loaded")

    def search(self, prof, lect_name, score, assign, team, grade, attendance, test):#gui에서 입력받은 값으로 각 강의를 분류해냅니다.
        self.prof = prof
        self.lect_name = lect_name
        self.score = score
        self.assign = assign
        self.team = team
        self.grade = grade
        self.attendance = attendance
        self.test = test

        i = 0
        #검색 조건에 걸리지 않을 경우가 있으므로 초기화를 시켜줍니다.
        self.result_list = [{"prof":None, "lecture_name":None, "score":None, "assign":None, "team": None, "grade": None, "attendance":None, "time_table": None,"full_link":None, 'test_count':None}]

        while i<len(self.list)-1:
            item = self.list[i]
            if self.prof == None or self.prof == '' or item["prof"] == self.prof:
                if self.lect_name == None or self.lect_name == '' or item["lecture_name"] == self.lect_name:
                    if self.score == None or self.score == '' or item["score"] >= self.score:
                        if self.assign == None or self.assign == '' or item["assign"] == self.assign:
                            if self.team == None or self.team == '' or item["team"] == self.team:
                                if self.grade == None or self.grade == '' or item["grade"] == self.grade:
                                    if self.attendance == None or self.attendance == '' or item["attendance"] == self.attendance:
                                        if self.test == None or self.test == '' or item["test_count"] == self.test:
                                            self.result_list.append(item)#조건을 모두 무합하는 결과 리스트를 작성합니다.

            i = i+1
```


Source: everytime

```
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QTableWidgetItem
import json_reader
```

```
def setupUi(self, Form):#gui를 띄울 시 창을 구성하는 모든 정보가 담겨 있습니다. 단, 각 버튼에 대한 이름은 retranslateUi에서 다시 바꿔 줍니다
    Form.setObjectName("Form")
    Form.setWindowModality(QtCore.Qt.WindowModal)
    Form.setEnabled(True)
    Form.resize(1405, 627)
    sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Minimum)
    sizePolicy.setHorizontalStretch(0)
    sizePolicy.setVerticalStretch(0)
    sizePolicy.setHeightForWidth(Form.sizePolicy().hasHeightForWidth())
    Form.setSizePolicy(sizePolicy)
    Form.setMinimumSize(QtCore.QSize(700, 500))
    Form.setMaximumSize(QtCore.QSize(1500, 950))
```

```
# 각 조건을 입력 받을 때마다 변수의 값을 변경해줍니다.
def inputProfessor(self):
    self.prof = self.lineEdit_3.text()
def inputLectureName(self):
    self.lect_name = self.lineEdit_4.text()
def inputScore(self):
    self.score = self.comboBox_12.currentText()
def inputAssignment(self):
    self.assign = self.comboBox_9.currentText()
```

Thank You!