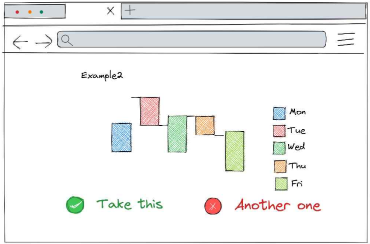
Timetable Recommendation Program

for Sungkyunkwan University students

2018312342 Kim Yerim
2021313553 Oh Wonjun

Contents

Recommendation



1 Introduction

2 DB design and modelling

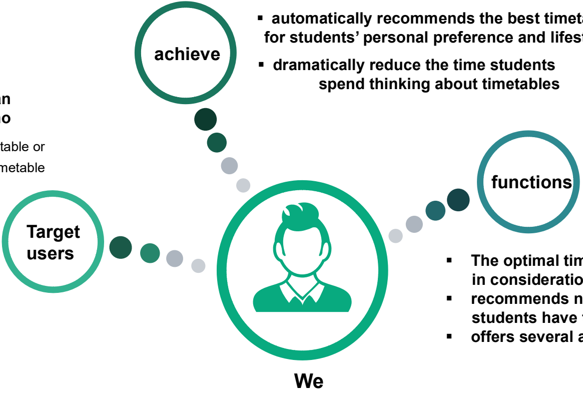
3 Demo of UI and functions

4 Lessons Learned

1. Introduction

01

Introduction



Target users

all the Sungkyunkwan university students who

are bothered to make a timetable or want to get a fast and efficient timetable

achieve

- **automatically recommends the best timetable for students' personal preference and lifestyle**
- **dramatically reduce the time students spend thinking about timetables**

functions

- **The optimal timetable is recommended in consideration of personal information**
- **recommends not to repeat the subjects students have taken before**
- **offers several alternatives**

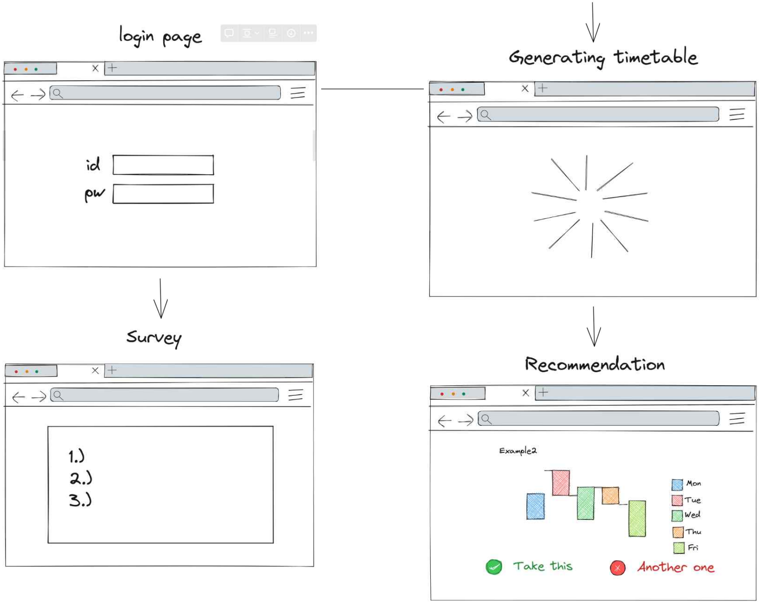
We

Introduction

- languages
  - Webframework: django
  - DB: mysql
  - Language: python
  - Datacleaning: pandas
  - Front-end design: html, css, tailwindcss

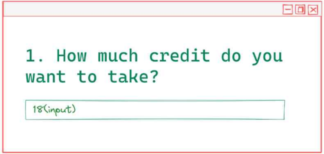
Introduction

- Flowchart



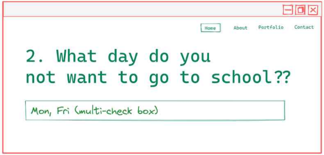
Introduction

- survey



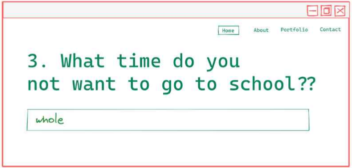
1. How much credit do you want to take?

18(input)



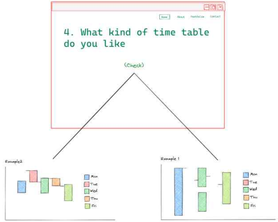
2. What day do you not want to go to school??

Mon, Fri (multi-check box)

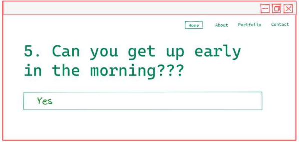


3. What time do you not want to go to school??

whole

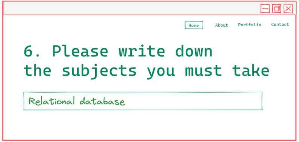


4. What kind of time table do you like



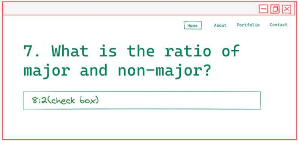
5. Can you get up early in the morning???

Yes



6. Please write down the subjects you must take

Relational database



7. What is the ratio of major and non-major?

8:2(check box)

2. DB Design and Modelling

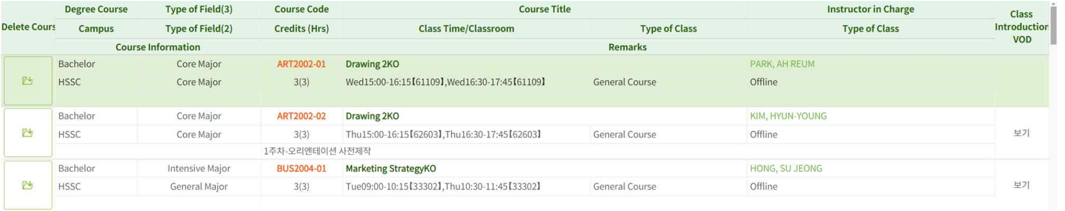
02

DB Design and Modelling

- Data acquisition
  - Subject data

  - Cannot be obtained by search → obtained by crawling at GLS (school website)

  - available online freely

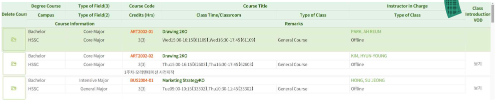


Delete Cours	Degree Course	Type of Field(3)	Course Code	Course Title		Instructor in Charge	Class Introduction VOD
	Campus	Type of Field(2)	Credits (Hrs)	Class Time/Classroom	Type of Class	Type of Class	
	Course Information			Remarks			
🗁	Bachelor	Core Major	ART2002-01	Drawing 2KO		PARK, AH REUM	
	HSSC	Core Major	3(3)	Wed15:00-16:15【61109】,Wed16:30-17:45【61109】	General Course	Offline	
🗁	Bachelor	Core Major	ART2002-02	Drawing 2KO		KIM, HYUN-YOUNG	보기
	HSSC	Core Major	3(3)	Thu15:00-16:15【62603】,Thu16:30-17:45【62603】	General Course	Offline	
			1주차-오리엔테이션 사전제작				
🗁	Bachelor	Intensive Major	BUS2004-01	Marketing StrategyKO		HONG, SU JEONG	보기
	HSSC	General Major	3(3)	Tue09:00-10:15【33302】,Thu10:30-11:45【33302】	General Course	Offline	

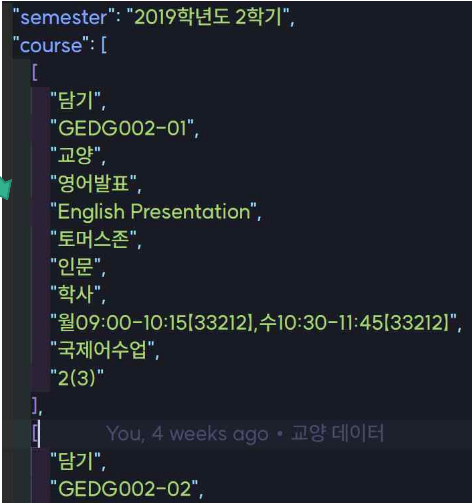
DB Design and Modelling

- Data acquisition
  - Subject data

  - At first, take all the letters in the table and save them in json format



Delete Cours	Degree Course	Type of Field(3)	Course Code	Course Title		Instructor in Charge	Class Introduction VOD
	Campus	Type of Field(2)	Credits (Hrs)	Class Time/Classroom	Type of Class	Type of Class	
	Course Information			Remarks			
	Bachelor	Core Major	ART2002-01	Drawing 2KO		PARK, AH REUM	
	HSSC	Core Major	3(3)	Wed15:00-16:15【61109】,Wed16:30-17:45【61109】	General Course	Offline	
	Bachelor	Core Major	ART2002-02	Drawing 2KO		KIM, HYUN-YOUNG	보기
	HSSC	Core Major	3(3)	Thu15:00-16:15【62603】,Thu16:30-17:45【62603】	General Course	Offline	
			1주차-오리엔테이션 사전제작				
	Bachelor	Intensive Major	BUS2004-01	Marketing StrategyKO		HONG, SU JEONG	보기
	HSSC	General Major	3(3)	Tue09:00-10:15【33302】,Thu10:30-11:45【33302】	General Course	Offline	

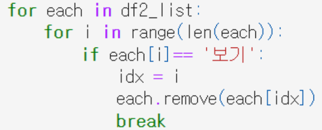


```
"semester": "2019학년도 2학기",
"course": [
  [
    "담기",
    "GEDG002-01",
    "교양",
    "영어발표",
    "English Presentation",
    "토머스존",
    "인문",
    "학사",
    "월09:00-10:15[33212],수10:30-11:45[33212]",
    "국제어수업",
    "2(3)"
  ],
  [          You, 4 weeks ago • 교양 데이터
    "담기",
    "GEDG002-02",
```

DB Design and Modelling

- Data cleaning and import methods

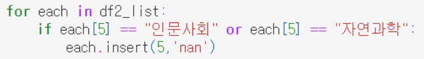
  - delete ‘보기(View)‘



```python
for each in df2_list:
    for i in range(len(each)):
        if each[i]== '보기':
            idx = i
            each.remove(each[idx])
            break
```

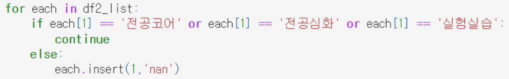
- Column length varies from row to row
→ need to fill in the ‘nan’ value where there is no value

  - If the name of the professor does not exist
→ insert ‘nan’ at the corresponding column location



```python
for each in df2_list:
    if each[5] == "인문사회" or each[5] == "자연과학":
        each.insert(5,'nan')
```

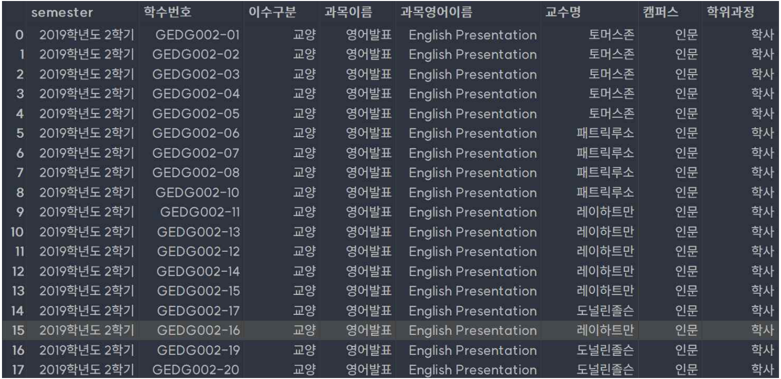
  - filling type of class3 values



```python
for each in df2_list:
    if each[1] == '전공코어' or each[1] == '전공심화' or each[1] == '실험실습':
        continue
    else:
        each.insert(1,'nan')
```

DB Design and Modelling

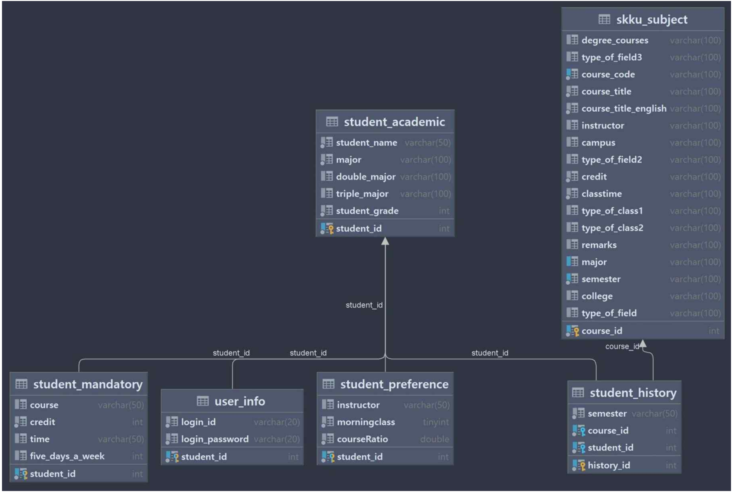
- Data cleaning and import methods
  - Preprocessed Data Frames



	semester	학수번호	이수구분	과목이름	과목영어이름	교수명	캠퍼스	학위과정
0	2019학년도 2학기	GEDG002-01	교양	영어발표	English Presentation	토머스존	인문	학사
1	2019학년도 2학기	GEDG002-02	교양	영어발표	English Presentation	토머스존	인문	학사
2	2019학년도 2학기	GEDG002-03	교양	영어발표	English Presentation	토머스존	인문	학사
3	2019학년도 2학기	GEDG002-04	교양	영어발표	English Presentation	토머스존	인문	학사
4	2019학년도 2학기	GEDG002-05	교양	영어발표	English Presentation	토머스존	인문	학사
5	2019학년도 2학기	GEDG002-06	교양	영어발표	English Presentation	패트릭루소	인문	학사
6	2019학년도 2학기	GEDG002-07	교양	영어발표	English Presentation	패트릭루소	인문	학사
7	2019학년도 2학기	GEDG002-08	교양	영어발표	English Presentation	패트릭루소	인문	학사
8	2019학년도 2학기	GEDG002-10	교양	영어발표	English Presentation	패트릭루소	인문	학사
9	2019학년도 2학기	GEDG002-11	교양	영어발표	English Presentation	레이하트만	인문	학사
10	2019학년도 2학기	GEDG002-13	교양	영어발표	English Presentation	레이하트만	인문	학사
11	2019학년도 2학기	GEDG002-12	교양	영어발표	English Presentation	레이하트만	인문	학사
12	2019학년도 2학기	GEDG002-14	교양	영어발표	English Presentation	레이하트만	인문	학사
13	2019학년도 2학기	GEDG002-15	교양	영어발표	English Presentation	레이하트만	인문	학사
14	2019학년도 2학기	GEDG002-17	교양	영어발표	English Presentation	도널린졸슨	인문	학사
15	2019학년도 2학기	GEDG002-16	교양	영어발표	English Presentation	레이하트만	인문	학사
16	2019학년도 2학기	GEDG002-19	교양	영어발표	English Presentation	도널린졸슨	인문	학사
17	2019학년도 2학기	GEDG002-20	교양	영어발표	English Presentation	도널린졸슨	인문	학사

DB Design and Modelling

- DB model
  - Full database



skku_subject	
degree_courses	varchar(100)
type_of_field3	varchar(100)
course_code	varchar(100)
course_title	varchar(100)
course_title_english	varchar(100)
instructor	varchar(100)
campus	varchar(100)
type_of_field2	varchar(100)
credit	varchar(100)
classtime	varchar(100)
type_of_class1	varchar(100)
type_of_class2	varchar(100)
remarks	varchar(100)
major	varchar(100)
semester	varchar(100)
college	varchar(100)
type_of_field	varchar(100)
course_id	int

student_academic	
student_name	varchar(50)
major	varchar(100)
double_major	varchar(100)
triple_major	varchar(100)
student_grade	int
student_id	int

student_mandatory	
course	varchar(50)
credit	int
time	varchar(50)
five_days_a_week	int
student_id	int

user_info	
login_id	varchar(20)
login_password	varchar(20)
student_id	int

student_preference	
instructor	varchar(50)
morningclass	tinyint
courseRatio	double
student_id	int

student_history	
semester	varchar(50)
course_id	int
student_id	int
history_id	int

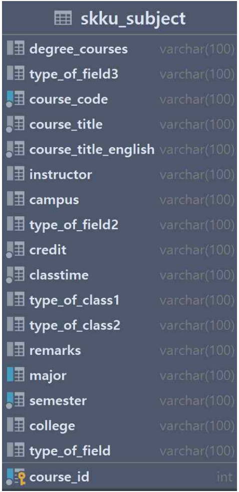
DB Design and Modelling

- DB model

1. skku_subject table

    - Information on previously crawled and pre-processed subjects

    - 35949 rows



skku_subject

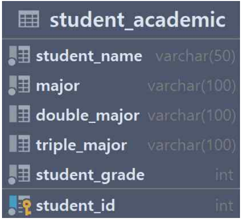
degree_courses	varchar(100)
type_of_field3	varchar(100)
course_code	varchar(100)
course_title	varchar(100)
course_title_english	varchar(100)
instructor	varchar(100)
campus	varchar(100)
type_of_field2	varchar(100)
credit	varchar(100)
classtime	varchar(100)
type_of_class1	varchar(100)
type_of_class2	varchar(100)
remarks	varchar(100)
major	varchar(100)
semester	varchar(100)
college	varchar(100)
type_of_field	varchar(100)
course_id	int

DB Design and Modelling

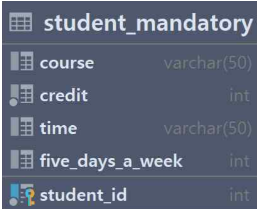
- DB model

  2. tables about students

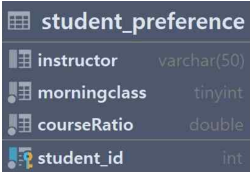
     - Data was randomly generated or filled with user's input



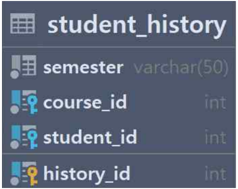
student_academic	
student_name	varchar(50)
major	varchar(100)
double_major	varchar(100)
triple_major	varchar(100)
student_grade	int
student_id	int



student_mandatory	
course	varchar(50)
credit	int
time	varchar(50)
five_days_a_week	int
student_id	int



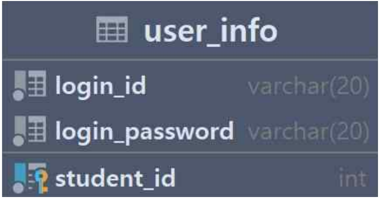
student_preference	
instructor	varchar(50)
morningclass	tinyint
courseRatio	double
student_id	int



student_history	
semester	varchar(50)
course_id	int
student_id	int
history_id	int

DB Design and Modelling

- DB model
  3. user table
    - Login features
    - Make it accessible only to oneself



user_info

login_id	varchar(20)
login_password	varchar(20)
student_id	int

3. Demo of UI and functions

- Run application
- functions

03

4. Lessons Learned

04

Lessons Learned

Data:
Collected through crawling & changed over time
→ difficulty of preprocessing

the difficulty of one person controlling the whole process

Thank you for listening!

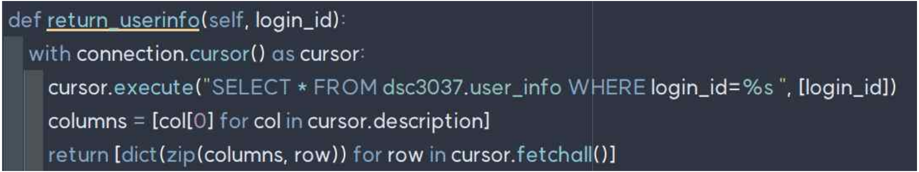
After this page, we will cover details mentioned in notification.

SQL- Quries

all the sql queries are in sql.py which has 4 functions.

- return_userinfo
- insert_survey
- subject_available
- student_info_join

Return_userinfo (SELECT)



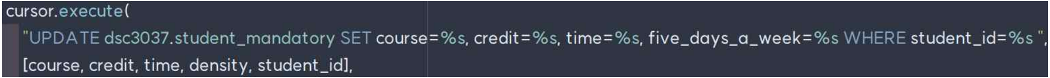
```python
def return_userinfo(self, login_id):
    with connection.cursor() as cursor:
        cursor.execute("SELECT * FROM dsc3037.user_info WHERE login_id=%s ", [login_id])
        columns = [col[0] for col in cursor.description]
        return [dict(zip(columns, row)) for row in cursor.fetchall()]
```

this function returns user's info when
login_id is given

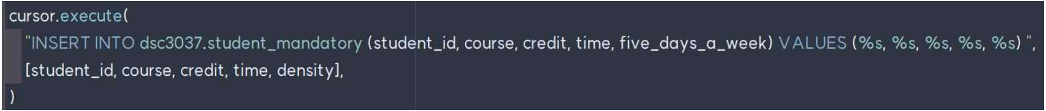
insert_survey (Insert & Update)

this function stores user's info into the db
when survey is done

If there already exists a student_id in the table,
update it. If not, insert it.



```
cursor.execute(
    "UPDATE dsc3037.student_mandatory SET course=%s, credit=%s, time=%s, five_days_a_week=%s WHERE student_id=%s ",
    [course, credit, time, density, student_id],
```



```
cursor.execute(
    "INSERT INTO dsc3037.student_mandatory (student_id, course, credit, time, five_days_a_week) VALUES (%s, %s, %s, %s, %s) ",
    [student_id, course, credit, time, density],
)
```

whole code is in sql.py

subject_available (SELECT, GROUP BY, JOIN)

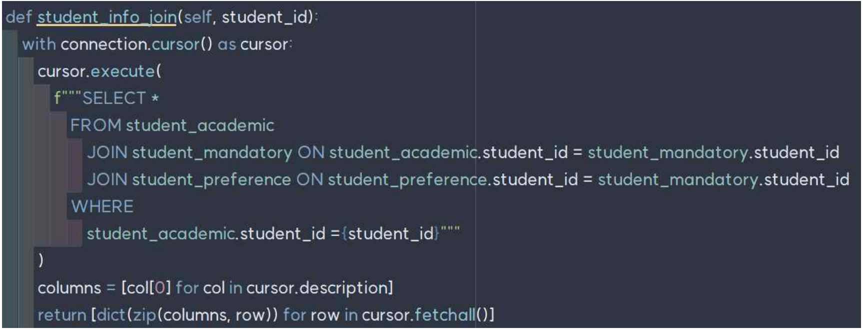
- this function returns all the courses that student can listen
- student only can listen courses that match his/her major, double major
- Students can take only subjects that they have not taken before.=> We filter this by using sub-query

this function returns user's info when login_id is given

whole code is in sql.py

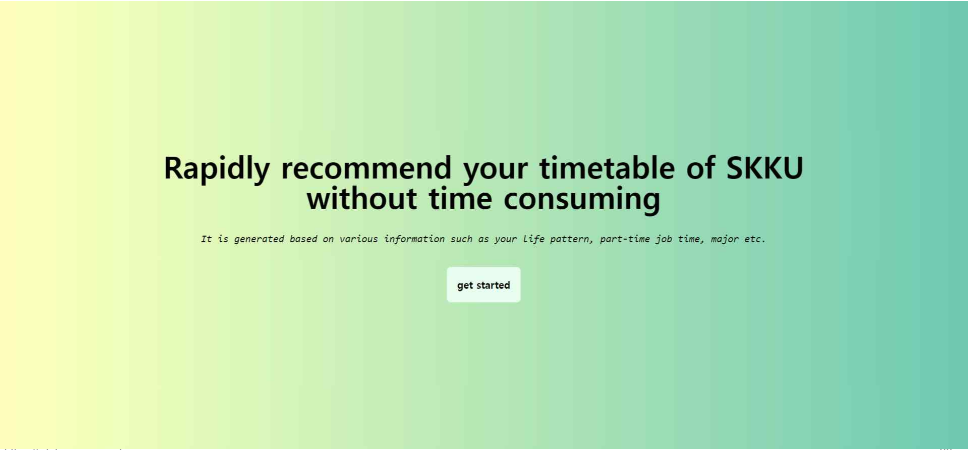
student_info_join (SELECT, JOIN)

This function joins all the information about student



```
def student_info_join(self, student_id):
    with connection.cursor() as cursor:
        cursor.execute(
            f"""SELECT *
                FROM student_academic
                    JOIN student_mandatory ON student_academic.student_id = student_mandatory.student_id
                    JOIN student_preference ON student_preference.student_id = student_mandatory.student_id
                WHERE
                    student_academic.student_id ={student_id}"""
        )
        columns = [col[0] for col in cursor.description]
        return [dict(zip(columns, row)) for row in cursor.fetchall()]
```

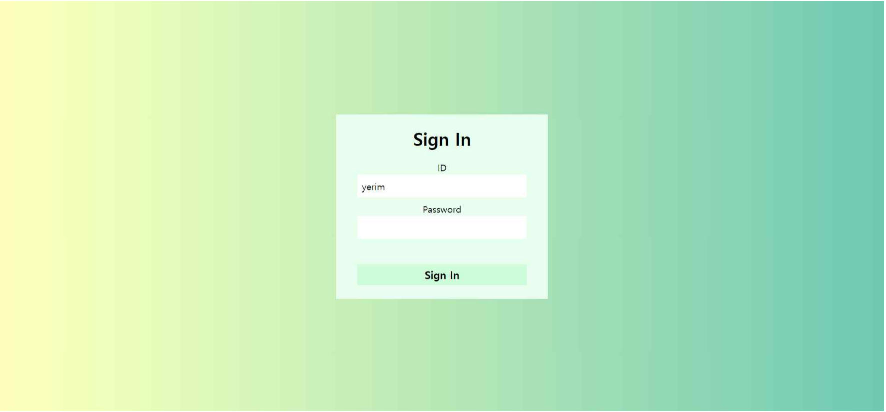
Screenshots1



Rapidly recommend your timetable of SKKU without time consuming

It is generated based on various information such as your life pattern, part-time job time, major etc.

Screenshots2



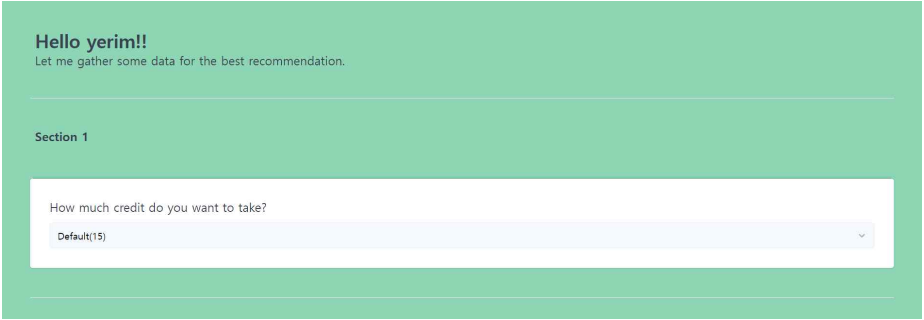
Sign In

ID

yerim

Password

Screenshots3



Hello yerim!!

Let me gather some data for the best recommendation.

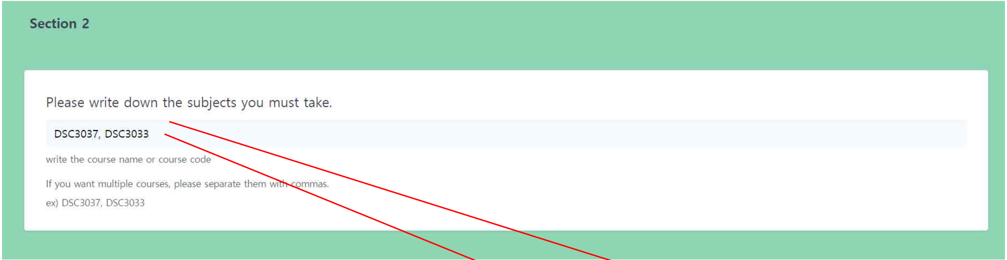
Section 1

How much credit do you want to take?

Default(15)

Feedback 1: The **default value** was explicitly written.

Screenshot4



Section 2

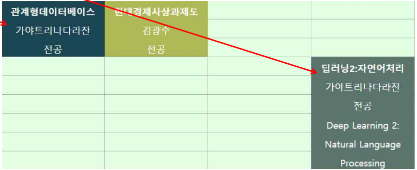
Please write down the subjects you must take.

DSC3037, DSC3033

write the course name or course code

If you want multiple courses, please separate them with commas.

ex) DSC3037, DSC3033



관계형데이터베이스
가야트리나다라잔
전공

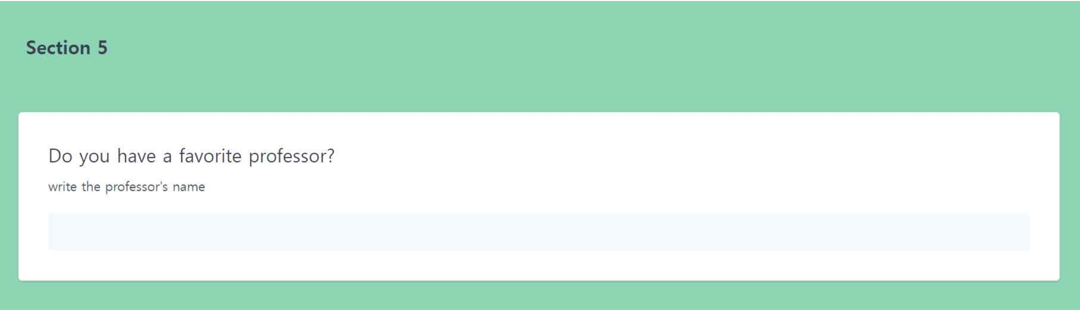
현대경제사상과제도
김광수
전공

딥러닝2:자연어처리
가야트리나다라잔
전공
Deep Learning 2:
Natural Language
Processing

Feedback 2: Users can write **course codes** as well as subject names.

Feedback 3: If user want **multiple subjects**, they can add them through commas.

Screenshots 3

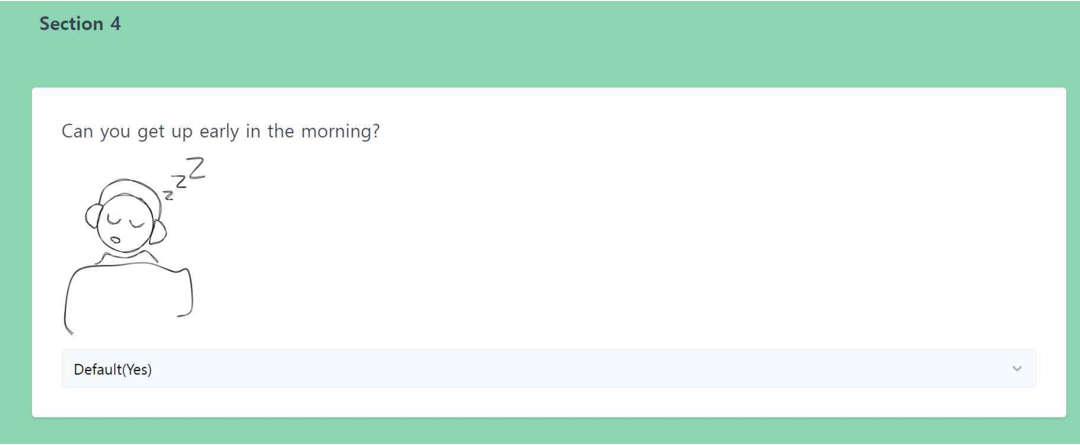


Section 5

Do you have a favorite professor?

write the professor's name

Screenshots 3

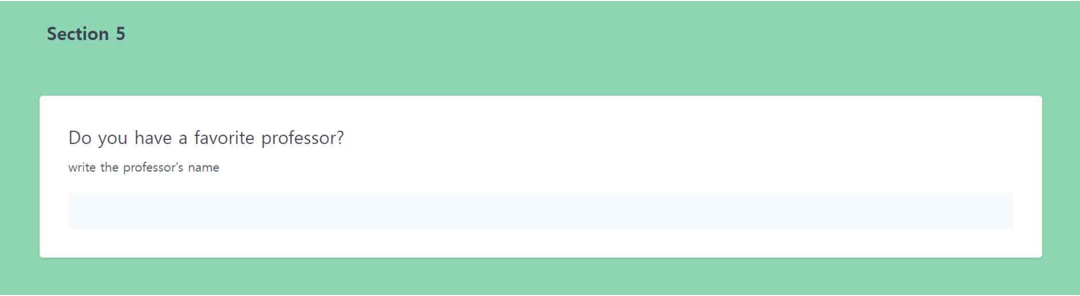


Section 4

Can you get up early in the morning?

Default(Yes)

Screenshots 3

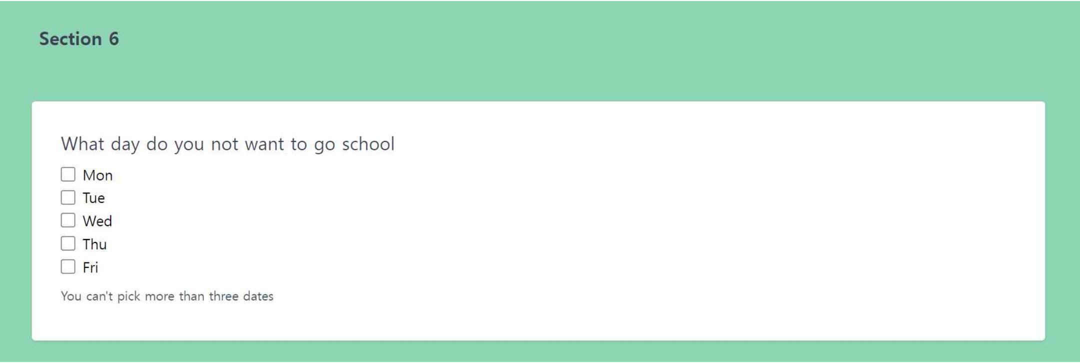


Section 5

Do you have a favorite professor?

write the professor's name

Screenshots 3



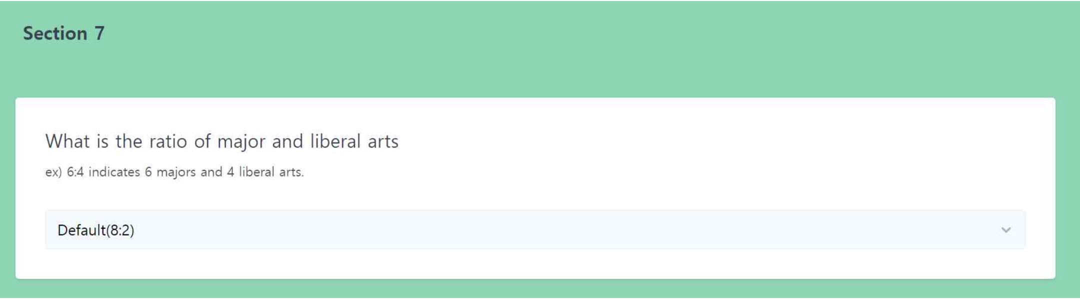
Section 6

What day do you not want to go school

- □ Mon
- □ Tue
- □ Wed
- □ Thu
- □ Fri

You can't pick more than three dates

Screenshots4



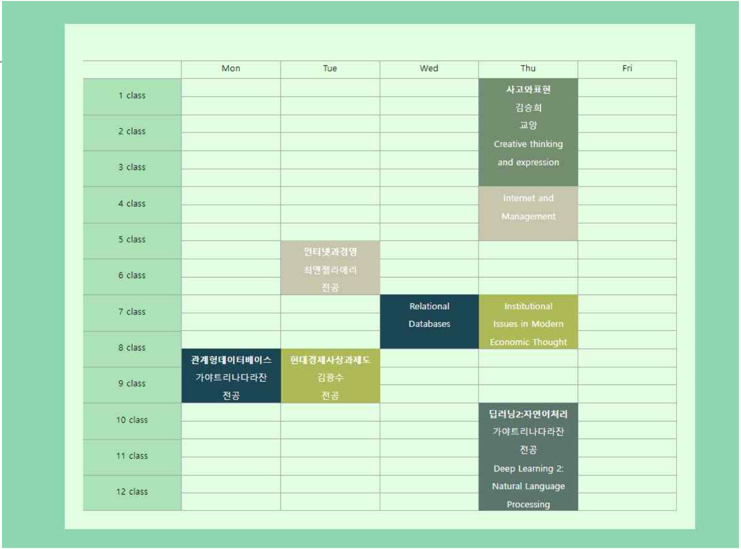
Section 7

What is the ratio of major and liberal arts

ex) 6:4 indicates 6 majors and 4 liberal arts.

Default(8:2)

Screenshots4



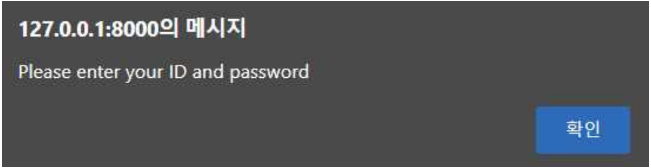
	Mon	Tue	Wed	Thu	Fri
1 class				사고와표현 김승희 교양 Creative thinking and expression	
2 class					
3 class					
4 class				Internet and Management	
5 class		인터넷과경영 최엔젤라애리 전공			
6 class					
7 class			Relational Databases	Institutional Issues in Modern Economic Thought	
8 class	관계형데이터베이스 가야트리나다라잔 전공	현대경제사상과제도 김광수 전공			
9 class					
10 class				딥러닝2:자연어처리 가야트리나다라잔 전공 Deep Learning 2: Natural Language Processing	
11 class					
12 class					

Recommend another timetable

User can refresh the page clicking this button or F5

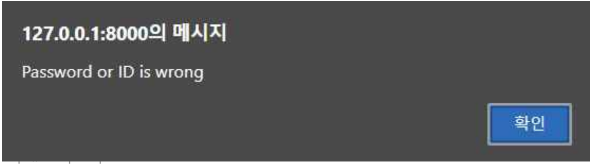
Extra updates for 2days

On the login page, give a message if the password and ID are incorrect.



127.0.0.1:8000의 메시지

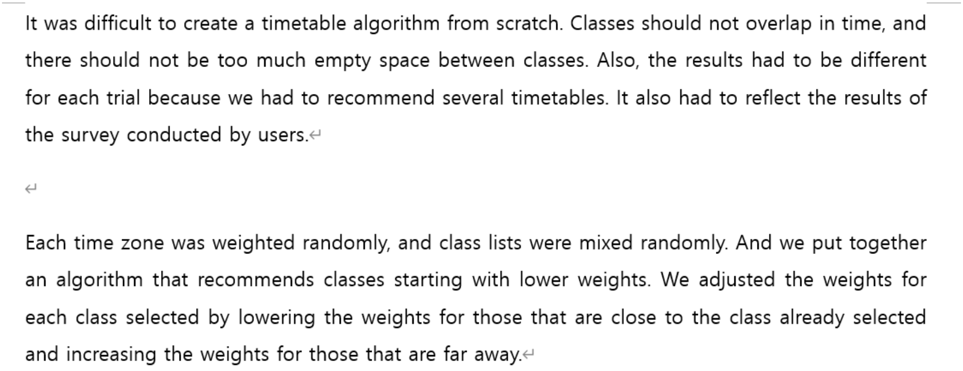
Please enter your ID and password



127.0.0.1:8000의 메시지

Password or ID is wrong

Specific features (timetable algorithm)

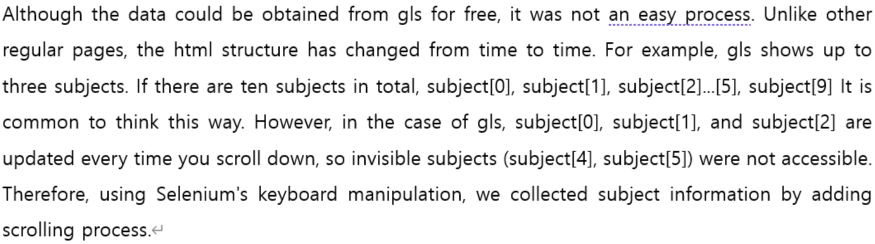


It was difficult to create a timetable algorithm from scratch. Classes should not overlap in time, and there should not be too much empty space between classes. Also, the results had to be different for each trial because we had to recommend several timetables. It also had to reflect the results of the survey conducted by users.↵

↵

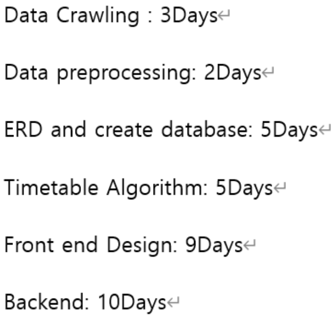
Each time zone was weighted randomly, and class lists were mixed randomly. And we put together an algorithm that recommends classes starting with lower weights. We adjusted the weights for each class selected by lowering the weights for those that are close to the class already selected and increasing the weights for those that are far away.↵

Specific features (Crawling)



Although the data could be obtained from gls for free, it was not an easy process. Unlike other regular pages, the html structure has changed from time to time. For example, gls shows up to three subjects. If there are ten subjects in total, subject[0], subject[1], subject[2]...[5], subject[9] It is common to think this way. However, in the case of gls, subject[0], subject[1], and subject[2] are updated every time you scroll down, so invisible subjects (subject[4], subject[5]) were not accessible. Therefore, using Selenium's keyboard manipulation, we collected subject information by adding scrolling process.↵

Time line of events



Data Crawling : 3Days↵

Data preprocessing: 2Days↵

ERD and create database: 5Days↵

Timetable Algorithm: 5Days↵

Front end Design: 9Days↵

Backend: 10Days↵

Challenges

- Optimizing the Time Tableting Algorithm

- Create a membership screen and allow users to add previously taken subjects themselves.