

Open-Source Software Practice Final Project - Proposal Phase

Donghun Jung, 2020312141¹

Jinhwa Hong, 2017310820²

hyeonjeong Ko, 2020315791³

¹Sungkyunkwan University, Department of Physics, atompioneer@g.skku.edu

²Sungkyunkwan University, Systems Management Engineering, bout123456@naver.com

³Sungkyunkwan University, Department of Software, toyu7870@naver.com

1 Application type

해당 프로젝트는 웹페이지 기반 오픈소스로, 튜터링 신청을 받는 웹사이트를 만들 것이다.

2 Theme

주제 : 성균튜터링 신청 사이트

해당과목에서 A이상 학점을 받은 튜터와 학습에 도움을 받고자 하는 튜티들이 그룹학습을 할 수 있도록 매칭하고 지원하는 프로그램이다. 매학기 약 500여명이, 100여 개의 팀으로 참여하여, 튜터를 중심으로 스터디를 이끌어간다. 많은 학생들이 참여하는 활동임에도 불구하고 신청 과정은 꽤 불편했다.

튜터 모집 과목	튜티 모집 과목
AI기초와 활용	Financial Management
General Physics 1	Introduction to Microeconomics
거시경제학	고급생명과학2
경제수학	기초중국어1
경제통계분석	기초프랑스어1
고분자공학입문	뇌와 마음
고전영저작클럽	비판적사고
고체역학실계실습	소비자행동
공업물리화학2	스포츠재무관리
공업유기화학2	영화의이해
공정설계입문	예술로보는동아시아

Figure 1: 성균튜터링 신청 과목 현황 예시

첫째로, 어떤 과목이 개설되어 있는지 확인하기 어렵다. 그림 1에서 확인할 수 있는 것처럼 교육개발센터에서 신청 마감 약 이틀전부터 모집 과목을 업로드해준다. 튜터와 튜티 모두 어떤 과목이 필요하고 개설되는지 확인하기 어렵다.

둘째로, 튜터링을 신청하면서도 성사여부를 확신할 수 없다. 개설여부를 즉시 확인하기 어렵기도 하고, 모집마감 여부 역시 불투명하기 때문이다. 이는 튜터링 신청을 주저하는 원인 중 하나이기도 하다.

셋째로, 튜터가 튜터링을 신청할 때에 제공하는 많은 정보를 제공하는 것에 반해, 튜티는 랜덤으로 튜터를 배정받는다. 튜터로 튜터링을 신청할 때, 본인의 성적, 수강한 과목의 담당 교수, 커리큘럼, 지원동기 등을 입력하나 튜티가 신청할 때에 관련된 정보를 전혀 얻을 수 없다. 교육개발센터에서는 담당 교수만을 중심으로 학생을 배정하고 있다. 해당 정보들을 모두 공개하여 튜티가 튜터를 선택하게 할 수 있는 것이 바람직하다.

마지막으로, 공지는 학교 홈페이지, 신청은 구글폼, 결과 확인은 교육개발센터 네이버 카페에서 이루어지는 번거로움이 있다.

이런 이유로, 성균튜터링 신청 사이트를 이번 프로젝트 주제로 선정하여 성균튜터링 신청 사이트를 제작하고자 한다.

3 Features

3.1 로그인 기능

학번(id), 이름(혹은 비밀번호)를 통해 로그인한다.

3.2 튜터링 과목 등록

튜터가 튜터링할 과목을 선택하고 튜터링을 신청한다. 신청과 동시에 튜터링 과목 목록에 추가되며, 튜터로부터 튜터링 신청을 받을 수 있다. 이때 튜터로부터 입력받아야 할 정보는 다음과 같다.

1. 취득 성적(A+/A)
2. 과목명
3. 수강한 교수명
4. 튜터링 계획(Syllabus)
5. 지원 동기
6. 그룹 지원 여부 및 튜티 학번
7. 가능한 시간 등등

3.3 튜터링 수강 신청

튜티는 튜터가 등록한 정보를 확인할 수 있고, 신청 버튼을 누르기만 하면 신청할 수 있다. 이때, 튜티에게 보여져야 하는 정보는 다음과 같음.

1. 과목명, 튜터명, 시간 등 기본적인 정보
2. 자세히 보기와 같은 버튼을 통해 열람 가능한 취득성적, 교수명, 튜터링 계획 등 자세한 정보
3. 신청 현황 (n/5와 같이 표기)

4 Usage scenario

4.1 회원가입

학번과 이름(혹은 pw)를 입력받아 회원가입한다. 만약 이미 존재하는 학번이면 → 실패하고, 그렇지 않다면 유저목록에 새로 등록한다.

4.2 로그인

학번과 이름(혹은 pw)를 입력받고 일치하면 튜터링 신청 화면으로 전환한다. 이때 https request를 통해 입력 받은 정보를 보내고 일치여부를 수신 받아야 한다. 일치하지 않을 시 로그인 할 수 없다는 알림을 띄운다.

4.3 튜티 과목 등록

과목 등록 시, 위에서 언급한 정보를 입력 받아야 한다. 이때 확인해야 할 정보는 다음과 같다. 먼저 취득성적이 A 미만인 경우, 과목 등록이 불가능하다. 둘째, 과목명은 학교 GLS에서 크롤링해온 과목 목록에 존재해야 한다. 튜터링 과목을 직접 입력받는 것이 아닌 선택하도록 할 수 있다. 마지막으로, 튜티 학번은 유저 목록에 존재해야 함.

4.4 튜티 수강 신청

튜티는 과목 목록에서 과목명, 튜터명, 튜터링 시간 등 기본적인 정보를 확인할 수 있다. 이때 신청 버튼과 자세히 보기 버튼을 누름으로써 인터페이스와 상호작용할 수 있다. 신청하기 버튼을 누르면 해당과목에 신청한다는 request를 서버에 보내고 가능여부를 서버로부터 수신 받고 알림을 통해 확인한다. 만약 튜터링 신청 인원이 초과되었거나 이미 신청한 튜터링이라면 불가능 메시지 띄운다. 그렇지 않다면, 수락 메시지 출력하고 신청 인원을 한 명 늘린다. 한편, 자세히 보기 버튼을 누르면 취득성적, 교수명, 튜터링 계획 등 자세한 정보 확인할 수 있다.

5 Task assignment

위 기능을 구현하기 위해 다음 작업이 필요할 것으로 예상된다.

5.1 AWS 서버 구축

학교에 개설된 과목 목록, 유저 목록, 튜터링 과목 목록을 저장하고 http request를 송수신할 서버 구축한다. python Flask를 활용해야 할 것이다.

5.2 디자인 패턴 : MVC, MVP, MVVP 모델

AWS 서버로부터 튜터링 과목 목록을 수신받고 화면에 출력할 cell을 만들기 위해 필요하다. 한편, 튜터링 신청이 승인되면, 수강인원을 하나 늘린다.

5.3 okhttp

HTTP는 웹 서비스를 위한 응용 프로그램 통신 프로토콜이다. 'get' method는 서버에 특정 데이터를 수신받기위해, 'post'

method는 서버의 데이터를 변경할 때 사용한다. 둘 모두 json string을 받으며, 데이터를 포함하거나 수정 성공여부를 받게 된다.

JS code에 http request 송수신과 관련된 코드가 추가되어야 한다. 해당 문서를 참고할 것이다.

(<https://www.npmjs.com/package/okhttp>)

5.4 Webpage design with HTML, CSS, JS

크게 네 개의 페이지가 필요할 것이다. 먼저 로그인 페이지가 있다. 학번과 이름을 입력받고, 로그인 버튼을 누르면 서버에 'post' request를 보낸다.

둘째로, 회원가입 페이지가 필요하다. 마찬가지로 학번과 이름을 입력받고, 'post' request를 통해 서버에 저장한다.

셋째로, 튜터링 신청 페이지이다. 로그인 직후 'get' request 튜터링 과목 목록을 받아온다. 이때 받아온 목록을 mvp model을 활용해 테이블 형식으로 보여줘야한다. 페이지의 형태는 수강신청 페이지와 유사할 것이다. 각각의 cell에는 신청 버튼이 있으며, 신청 버튼을 누르면 AWS 서버에 학번과 함께 'post' request를 보낸다. 해당 학번이 이미 등록되어 있거나, 신청 인원이 가득차있지 않으면 True를, 아닌 경우 False를 돌려준다.

마지막으로, 튜터링 과목 등록이다. 튜터로서 튜터링을 신청하는 것으로, 위에서 언급한 정보를 입력 받아야한다. 입력 받은 정보를 'post' request를 통해 서버에 등록한다.

5.5 crawling

학교 GLS에 존재하는 과목을 끌어와 AWS 서버에 저장할 것이다. 튜터링 과목명을 튜터가 직접 입력하도록 할 경우, 같은 과목이 다르게 저장될 수 있기 때문이다.

6 Sketch

그림 2 참고.

Figure 2 shows three web forms from SKKU. (a) is the login form with fields for 학번 (Student ID) and 이름 (Name), and a 로그인 (Login) button. (b) is the tutor application form, which includes a 회원가입 (Sign Up) section with fields for 학번, 이름, and a 비밀번호 (Password) field, and a 로그인 (Login) button. (c) is the tutor application form, which includes a 회원가입 (Sign Up) section with fields for 학번, 이름, and a 비밀번호 (Password) field, and a 로그인 (Login) button. Below the login forms, there are two tables showing the results of the application process. The first table, titled '튜터링 신청(튜터)', shows the results for the tutor application, with columns for 수강신청 (Application), 학수번호 (Student ID), 과목명 (Subject), 교수명 (Professor), 신청/제 (Application/Withdrawal), 튜터명 (Tutor Name), 시간/강의실 (Time/Lecture Room), 상세 보기 (Details), and 비고 (Remarks). The second table, titled '튜터링 신청(튜티)', shows the results for the tutee application, with the same columns. Both tables show a list of applications with status indicators (e.g., 신청, 신청완료, 신청취소) and buttons for details and withdrawal.

Figure 2: (a) 로그인, (b) 튜터링 신청(튜터), (c) 튜터링 신청(튜티)