

Final Report

8조 – 김동현, 김종진, 오지현, 이동민

Abstract

본 시스템은 서울대학교 재학생들에게 더 좋은 배움의 기회를 제공하기 위해 튜터와 튜티를 효율적으로 매칭시켜주는데 목표를 두고 있다. 튜터링을 해주기를 희망하는 학생들은 자신의 전공과 소개, 이력 등을 시스템에 등록하여 튜티들이 그 정보를 기반으로 적절한 튜터를 찾을 수 있도록 했다. 이후 튜티가 튜터에게 튜터링 요청을 보내면 튜터가 이를 수락하여 둘 사이에 튜터링이 이루어질 수 있도록 했다.

Motivation & Problem

서울대학교에서 제공하는 다양한 강의들을 듣다 보면 혼자 공부하기 힘들어 도움을 받고 싶은 과목들이 종종 있곤 하다. 그러나 같은 강의를 같이 수강하는 친구가 없거나 같이 수강하는 친구조차 큰 도움이 되지 못한다면 어떻게 해야 할까? 이 시스템은 이런 경우에 도움을 주고자 하는 학생들과 도움을 받고자 하는 학생들을 연결해주기 위한 목적을 갖고 있다.

Related Work

본 시스템과 가장 비슷한 프로그램은 '서울대학교 기초교육원 교양피어튜터링 프로그램'이다. 이 프로그램은 3~4학년 학생들이 튜터가 되어 한 학기동안 주 1~2회씩 튜티에게 수업 진도에 맞춘 개인적인 도움을 제공하는 프로그램이다. 튜터와 튜티는 홈페이지를 통해 온라인으로 모집하며 튜터의 경우 성적이 우수한 3~4학년 학생 한정으로 모집한다. 이후 수시로 튜터와 튜티를 매칭시켜 주며 튜터에게는 활동비를 지원해주는 형식이다.

그러나 본 시스템은 교양피어튜터링 프로그램과 비교해 몇 가지 차별점을 가지고 있다. 먼저 대상 교과목의 범주이다. 교양피어튜터링 프로그램은 한국어/영어, 글쓰기 및 핵심 교양 교과목 혹은 기초수학, 기초물리, 기초화학, 기초생물 교과목에 대해서만 튜터-튜티 매칭 서비스를 제공하고 있다. 반면에 본 시스템에서 목표로 하는 바는 기초 교양 교과목들 뿐만 아니라 전공 과목 등 모든 과목에 대한 튜터-튜티 매칭 서비스를 제공하는 것이다.

또 다른 큰 차별점은 튜터-튜티 간 교육이 이루어지는 기간에 있다. 교양피어튜터링 프로그램에서는 1학기 동안 매주 1~2회의 정기적인 만남이 이루어지는 것을 상정하며 그에 따라 매 만남마다 보고서를 제출하여야 한다. 하지만 본 시스템에서는 튜티의 필요에 따라 튜터와 튜티의 합의하에 자유롭게 기간과 빈도 설정이 가능하다. 이는 다가오는 퀴즈나 시험, 과제에 대한 도움이 필요한 튜티에게 필수적인 요소이다.

본 시스템과 비슷한 서비스를 제공하는 다른 프로그램으로는 여러 온라인 과외 사이트를 생각해 볼 수 있다. '과외바다', '오르비 과외시장' 등과 같은 사이트는 과외를 잡고 싶은 예비 선생님들을 등록받아 과외를 받기를 희망하는 학생들이 적당한 선생님을 검색해 찾아볼 수 있도록 하는 서비스를 제공하고 있다. 이는 등록된 튜터들 중 자신이 원하는 사람을 선택해 연락할 수 있으며 자세한 사항은 튜터와 튜티끼리 합의하도록 한다는 점에서 본 시스템과 비슷하다.

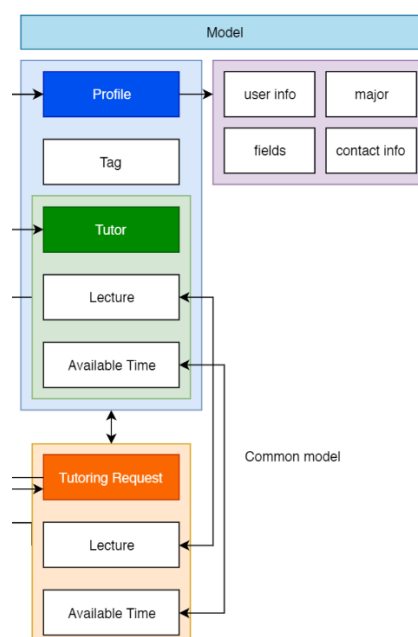
하지만 이러한 과외사이트들은 대부분 대입을 준비하는 중고등학생들을 타겟으로 한 것이라는 점이 본 시스템과 다르다. 본 시스템은 주로 대학생들을 타겟으로 할 것이므로 기존의 과외 사이트들과 차별화가 가능하다. 이러한 과외 사이트 뿐만이 아니라 유사한 튜터-튜티 매칭 서비스를 제공하는 '탈잉'과 같은 서비스들 역시 타겟 수요층이 본 시스템과 달라 차별화가 가능할 것이다.

Functionality

먼저 이 시스템은 서울대학교 학생들을 대상으로 하고 있기 때문에 회원가입을 할 때 서울대학교 이메일을 통한 인증을 요구한다. 이메일 인증 이후 이용자는 자신의 이름, 연락처, 전공, 간략한 소개와 이력 등을 입력하고 자신이 수강한 과목들을 등록한다. 이후 각 사용자는 자신이 도움을 받고자 하는 과목에 대해 적절한 전공, 수강 과목, 소개 등으로 튜터를 검색해 해당하는 튜터들의 리스트를 확인할 수 있다. 이렇게 튜터를 찾은 후 사용자는 해당 튜터에게 자신이 원하는 조건으로 튜터링 리퀘스트를 보낼 수 있다. 리퀘스트로 인한 알림을 받은 튜터는 이를 승인하거나 거부할 수 있고 만일 승인했다면 튜터링이 서버에 등록되어 이루어지게 된다.

Design & Implementation

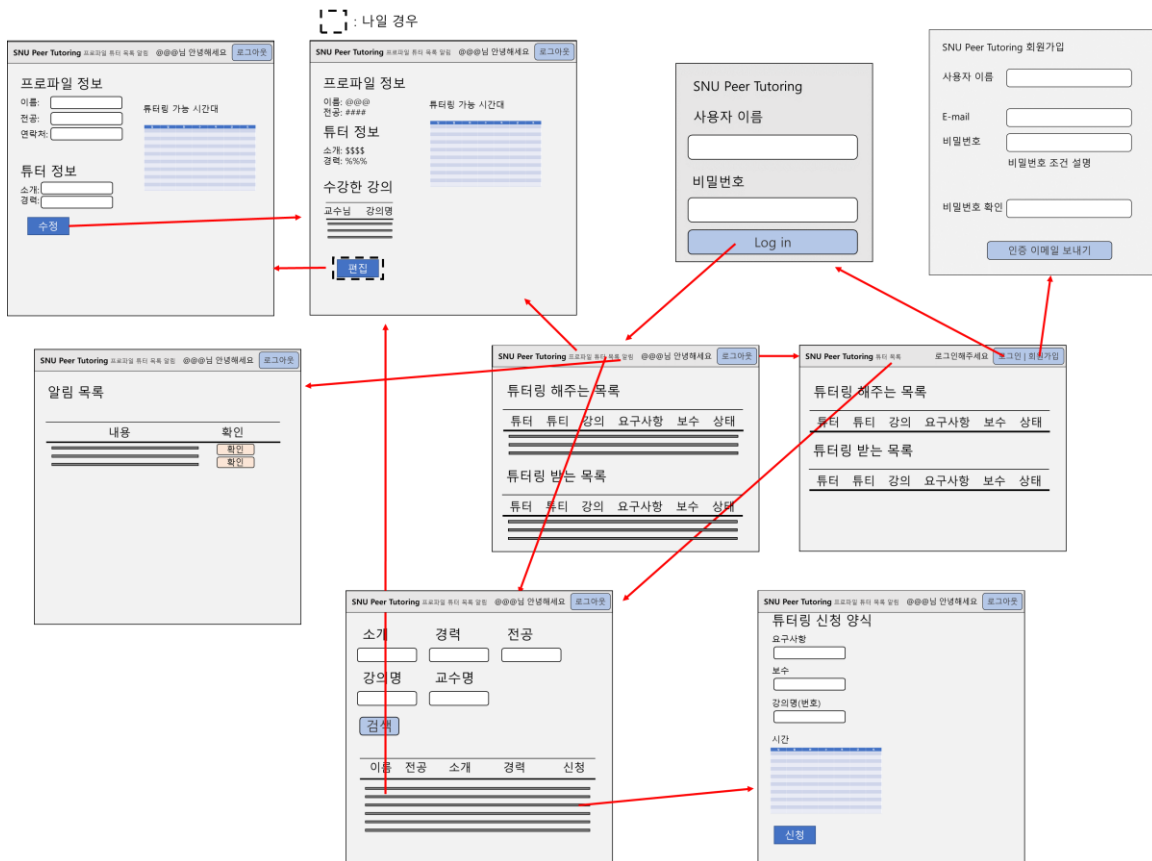
위의 기능을 구현하기 위한 백엔드 모델의 대략적인 구성은 다음과 같다.



그에 따른 백엔드 API의 대략적인 구성은 다음과 같다.




Model	API	GET	POST	PUT	DELETE
Profile	/profiles	전체 프로필 보기			
	/profile/:id	특정인 프로필 보기		프로필 수정	
Tutor	/tutors	필터링한 튜터 보기			
	/tutor/:id	특정인 튜터 보기		튜터 정보 수정	
Request	/requests	리퀘스트 전체 보기	리퀘스트 생성		
	/request/:id	특정 리퀘스트 보기			리퀘스트 삭제
Notification	/notification/:id	알림 내용 보기			알림 삭제
Auth	/login		로그인		
	/register		회원가입		

이를 활용하기 위한 프론트엔드의 대략적인 구성은 다음과 같다.



이를 구현하기 위해 Iteration별로 적절히 분량을 나누어 첫 주기에는 모델 설계 및 개발 환경 구성을 하였으며 두 번째 주기에는 기본적인 모델들과 뷰를 작성하였고 세 번째 주기에 필터 기능과 몇몇 버그 픽스, 보안 문제를 해결하였으며 네 번째 주기에 수강 과목에 대한 데이터베이스 구현 및 필터링 기능 구현, 프론트엔드 페이지 몇몇을 구현하였고 마지막 주기에 자잘한 버그 픽스와 구현되지 않은 기능들을 마저 구현하였다.

UI/UX

← → ↺ <https://ldm2468.com/swpp/> ☆    ⋮

SNU Peer Tutoring 튜터 목록 로그인해주세요. [로그인](#) [회원가입](#)




튜터링 해주는 목록

튜터	튜티	강의	요구사항	보수	시간	상태
----	----	----	------	----	----	----

튜터링 받는 목록

튜터	튜티	강의	요구사항	보수	시간	상태
----	----	----	------	----	----	----

처음 접속하면 이러한 홈페이지가 사용자를 맞이한다. 회원가입 이후 로그인을 하면 각 목록이 표시되게 된다. 한편, 튜터 목록을 확인해보면 아래와 같은 화면이 나타나는데

← → ↺ <https://ldm2468.com/swpp/tutors>  ☆   ⋮

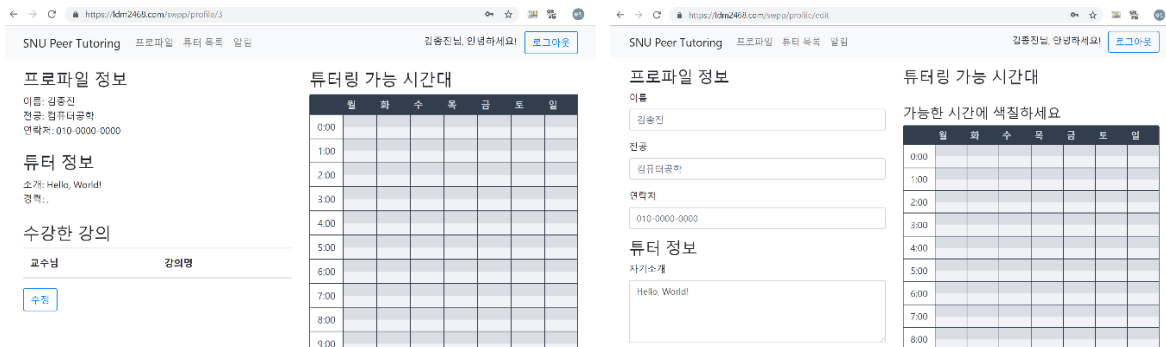
SNU Peer Tutoring 프로필 튜터 목록 알림 김종진님, 안녕하세요! [로그아웃](#)

소개	경력	전공	수강 강의명
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
강의 교수	최소 시간 (0.5시간 단위)	최소 모임 시간 (0.5시간 단위)	
<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="0.5"/>	시간대 설정

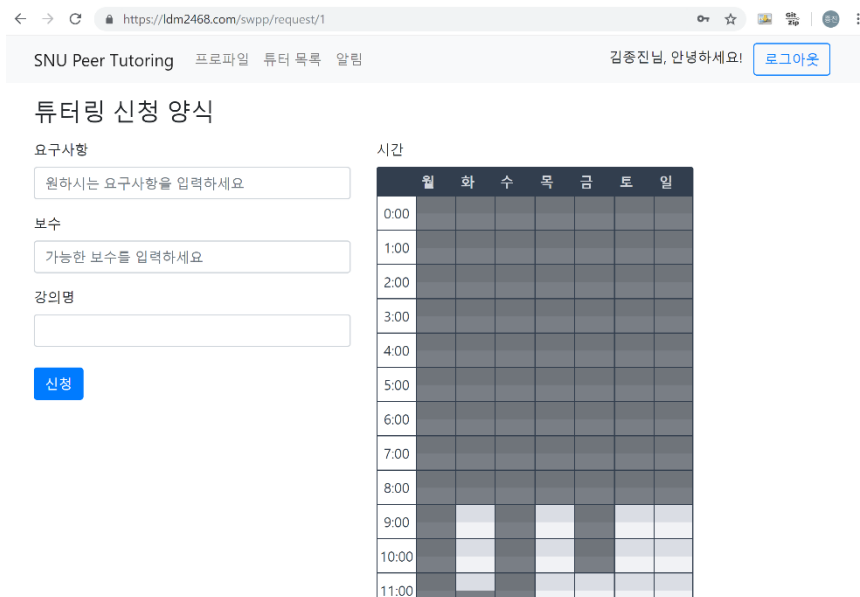
[검색](#)

이동민 전공: 컴퓨터공학부	히 히힝ㅎㅎ히	경력: 메이플 좀 했다가 접었음.	튜터링 신청하기
동현 전공: 컴공	안녕하세요 하하	경력: 3학년..	튜터링 신청하기
김종진 전공: 컴퓨터공학	Hello, World!	경력: .	튜터링 신청하기
오지현	I'm fine thank you and you?	경력: 백수입니다	튜터링 신청

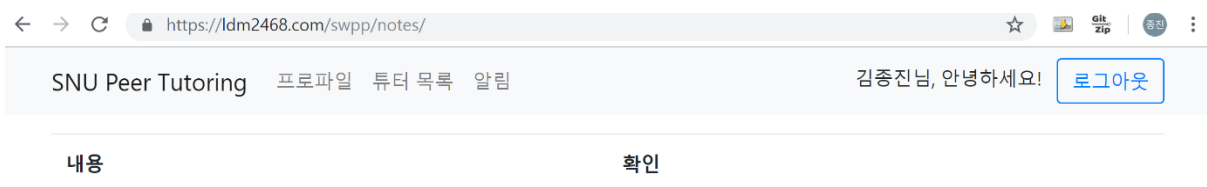
저 필터에 적절한 값을 입력하여 검색하면 자신에게 적합한 튜터를 손쉽게 찾을 수 있다.



이 때 프로파일 버튼 혹은 튜터 리스트에서 튜터의 이름을 클릭하면 좌상단과 같이 해당 인물의 정보를 확인할 수 있다. 이 때 해당 페이지가 자신의 것일 경우 수정 버튼을 눌러 우상단의 화면과 같이 자신의 정보를 수정할 수도 있다. 한편 튜터 리스트에서 튜터의 이름 대신 튜터링 신청 버튼을 누르면 다음과 같은 화면이 나오는데



여기에 적절한 내용을 기입하고 원하는 시간대를 칠한 후 신청 버튼을 눌러 해당 튜터에게 튜터링 요청을 보낼 수 있다. 이 요청을 받은 튜터는 위의 알림 버튼을 클릭하면 나오는



이러한 알림 페이지에서 알림을 받아볼 수 있으며 맨 처음에 등장했던 메인 페이지에서 해당 리퀘스트를 수락하거나 거절할 수 있다. 리퀘스트를 수락하면 튜터링 정보가 서버에 등록되어 튜터링이 진행되게 되며 거절하면 튜티에게 마찬가지로 거절되었다는 알림이 보내지게 된다.

Conclusion

한 학기에 걸친 웹 어플리케이션 개발을 통해 프론트엔드와 백엔드를 분리해 구조적으로 안정적이고 잘 짜여진 어플리케이션 설계 방법을 익힐 수 있었으며 django를 통해 원하는 데이터베이스를 백엔드에 구축하는 방법과 react-redux를 이용해 백엔드의 내용을 보기 좋은 페이지에서 다룰 수 있는 프론트엔드 구축 방법을 익힐 수 있었다. 뿐만 아니라 다양한 테스트 방법으로 소프트웨어를 완벽하게 검증하는 법을 익힐 수 있었다. 이번 프로젝트의 경험을 바탕으로 이후에 접할 소프트웨어 개발에서 보다 안정적이고 명료한 소프트웨어의 구조를 설계할 수 있을 것이며 다양한 테스트 환경을 통해 소프트웨어의 완성도를 높일 수 있을 것이다.