

# 2024-1 자료구조 프로젝트 [ MENTOS ] 계획서

작성일 : 2024-03-31

작성자 : 23-015 김동준, 23-031 김준이, 23-046 문가온

## 1. 문제 상황

‘멘토링’은 멘토가 멘티에게 특정 분야의 지식을 제공하는 제도로, 학생운영위원회 학업봉사부가 관리한다. 멘토링은 수년간 진행된 시스템이지만, 여러 가지 불편한 점이 학업봉사부, 멘토, 멘티로부터 지속해서 제기되었다.

### 1.1. 학업봉사부

멘토링 관리자가 겪는 문제점은 다음과 같았다.

첫째, 우수 멘토를 선정하는 과정에서 문제가 있었다. 우수 멘토는 가장 많은 시간 동안 멘토링을 한 몇 개의 팀에 속한 멘토를 의미한다. 멘토링 시간을 산정하기 위해, 기존 시스템에서는 구글 폼을 이용해 매 멘토링마다 시작 시각과 종료 시각을 제출해야 했다. 관리자는 구글 폼을 이용해 수집한 데이터를 다시 가공하여 총 멘토링 시간을 계산하는 과정에서 많은 시간을 소모할 수밖에 없다.

둘째, 우수 멘티를 선정하는 과정에서 문제가 있었다. 우수 멘티는 2023학년도 2학기에 존재하던 시스템으로, 멘토링을 많이 출석한 멘티에 대해 상품을 주는 제도였다. 하지만 2024학년도 1학기에는 사라졌는데, 학업봉사부는 그 원인에 멘토가 출석 확인을 성실하게 수행하지 않는 것으로 꼽았다.

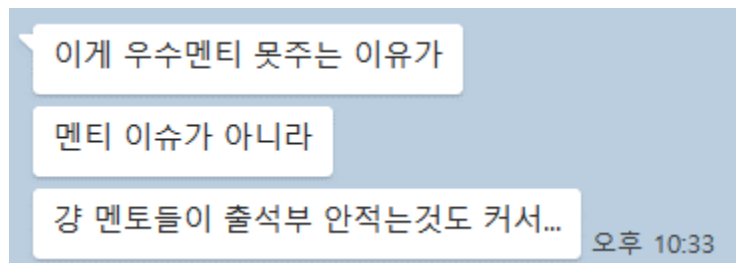


그림 1. 제33대 학생운영위원회 학업봉사부, 우수 멘티 시스템이 사라진 이유

셋째, 멘토링에 관련된 정보가 산재하여 있다. 멘토링 계획서는 구글 드라이브에 게시되어 있으며, 멘토링 시수는 전술했듯 구글 폼을 통해 산정하며, 멘토링 예정 시각 및 장소는 각 멘토링의 카카오톡 방에서만 확인할 수 있다. 한편 위에서 언급한 멘티 출석 확인은 구글 스프레드시트에서만 할 수 있었다. 이렇게 멘토링은 여러 플랫폼에서 관리되므로 관리자가 멘토링 현황을 한눈에 파악하기 어렵게 한다는 문제점을 불러일으켰다.

### 1.2. 멘토

멘토는 멘토링을 계획하고 진행하는 이들이다. 기존 시스템상에서는 멘토들이 멘토링 계획 과정에서 곤란함을 겪는 경우가 많았다. 앞서 언급했듯 멘토링 예정 시각 및 장소는 각 멘토링의 카카오톡 방에서만 공지되었기에, 겨우 잡은 일정이 다른 멘토링과 겹치기 십상이었다. 또한, 멘토링을 기록하는 과정에서도 여러 플랫폼

에 접속하여 정보를 따로 입력하는 번거로움이 발생한다. 이는 멘토링 기록에 실수를 유발하여 우수 멘토 선정 등에서 피해를 불러일으킨다. 실제로 2023학년도 2학기 모 멘토링의 경우 여러 플랫폼에 정보를 입력하다가 종료 시각을 2시간을 이르게 입력하여 세계시민 시수를 2시간 낮게 부여받은 경험이라고 한다.

### 1.3. 멘티

멘티는 멘토링의 수강생이다. 각 멘토링의 시작 시각과 장소는 카카오톡 방에서만 확인할 수 있으므로, 멘토링 일정을 한눈에 확인하기가 어렵다. 멘토가 멘토링 일정을 겹치게 설정했다면 그 피해를 고스란히 보는 것은 멘티이기도 하다.

2023학년도부터 학업봉사부는 멘토링 청강 시스템을 활성화하겠다고 선언하였는데, 사실상 실패한 정책이라고 평가받는다. 그 원인은 다른 멘토링의 일정을 알기 어렵기 때문으로 꼽힌다. 따라서 청강 시스템의 활성화를 위해서는 다른 멘토링의 일정까지 한눈에 볼 수 있는 방법이 필요하다.

## 2. 해결 방안 (프로젝트 목적)

이렇게 멘토링에서 발생하는 많은 문제를 해결하기 위해, 기존 시스템의 문제를 해결한 멘토링 통합 관리 프로그램인 'MENTOS'를 제작할 계획이다. 기존 문제에 대한 해결 방안은 다음과 같다.

먼저, 모든 멘토링 정보를 MENTOS 데이터베이스에 체계적으로 저장하고 이를 사용자 친화적인 인터페이스를 통해 공개할 것이다. 학업봉사부는 MENTOS를 이용해 모든 정보를 효율적으로 관리할 수 있어 멘토링 현황을 한눈에 파악할 수 있을 것이다. 더불어, 멘토는 다른 멘토링 정보를 확인하여 자신의 일정을 조율할 때 다른 멘토링과 일정이 겹치지 않도록 조치할 수 있다. 멘티들은 MENTOS를 통해 모든 멘토링 일정을 한눈에 확인할 수 있을 뿐만 아니라, 신청하지 않은 멘토링 일정도 찾아볼 수 있으므로 원하는 멘토링에 자유롭게 청강할 수 있게 된다.

두 번째로, MENTOS에 멘토와 멘티의 랭킹 시스템을 구축할 것이다. 랭킹 시스템을 이용해 학업봉사부는 우수 멘토와 우수 멘티를 더 쉽게 정할 수 있다. 지금까지는 학기가 끝나야만 멘토링을 다른 사람에 비해 얼마나 많이 했는지 또는 참여했는지를 알 수 있었다. 하지만 랭킹 시스템을 도입하면 실시간으로 다른 사람들과 내 참여도를 비교할 수 있게 된다. 이 시스템은 멘토가 더 열심히 멘토링을 진행할 수 있는, 그리고 멘티가 더 열심히 멘토링에 참여할 수 있는 동기를 부여할 것이다.

마지막으로, MENTOS에 효율적인 멘토링 기록 시스템을 구축할 것이다. 지금까지 출석은 구글 스프레드시트에 멘토가 하나하나 입력하는 방식으로 진행되었는데, 이를 개선하여 멘티가 멘토에게 출석 요청을 보내도록 조치할 것이다. 멘토는 출석 요청을 보낸 멘티의 명단을 보고 출석을 허가할 수 있다. 또한, 멘토링 시간을 구글 폼에 직접 입력하는 것이 아닌 MENTOS에서 자동으로 기록되도록 할 것이다. 이를 통해 학업봉사부는 더 정확한 정보를 확보할 수 있을 것으로 기대된다. 더불어, 멘토의 기록에 대한 의존성을 낮춤으로써 실수로 인한 피해를 방지할 수 있다. 멘티들도 자신의 멘토링 출석 여부를 손쉽게 확인할 수 있을 것이다.

### 3. 최종 구현 범위

위에서 언급한 해결 방안을 모두 구현할 수 있는 MENTOS의 기능을 더 자세하게 기술하겠다. 여기에서 사용된 그림은 이해를 돕기 위한 예시 이미지이며, 실제 프로그램에서는 UI가 이와는 다를 수 있다는 점을 미리 알려둔다.

#### 3.1. 메인 화면 - 멘토링 목록

MENTOS의 메인 화면에서는 그림 2와 같이 해당 학기에 진행 중인 멘토링의 목록을 보여줄 것이다. 멘토링 목록에는 멘토링 코드, 멘토링 이름, 멘토 목록, 멘토링 장소, 멘토링 날짜와 시간 정보가 표시된다. 자신이 참여하고 있는 멘토링은 가장 위에서 배치되고 강조되며, 접속 당일에 멘토링이 시작하는 것으로 예정된 경우에는 시작까지 남은 시간이 표시된다.

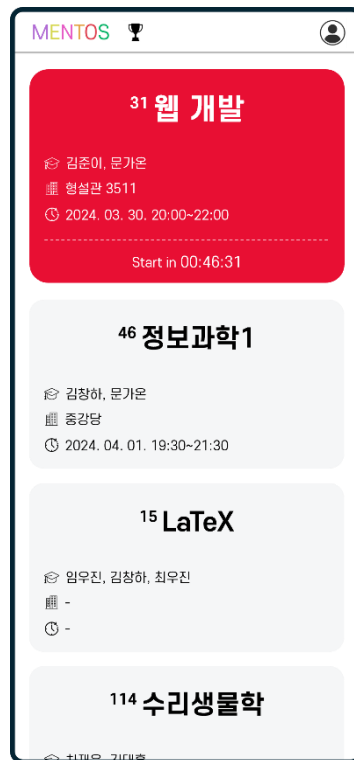


그림 2. MENTOS의 메인 화면

#### 3.2. 멘토링 관리 시스템

메인 화면의 멘토링 목록을 클릭하면 해당 멘토링의 정보 탭으로 넘어갈 수 있다. 여기서는 해당 멘토링의 멘토인 경우와 멘티인 경우로 나뉘어져 다른 기능을 제공한다.

공통적으로는 멘토링 코드, 멘토링 이름, 멘토 명단, 분류, 멘토링 누적 시간의 기본적인 정보를 볼 수 있다. 그리고 화면의 밑부분에서는 멘토링 상세 내역을 볼 수 있다.

멘토는 '시작' 버튼을 누를 수 있다. (그림 3의 두 번째 그림 참고) 시작 버튼을 누르면 멘토링 시간이 기록되기 시작한다. 또한 시작 버튼을 누르면 멘티가 출석 요청을 보낼 수 있는데, 이에 대해서는 후술하겠다. 시작 버튼을 누르면 버튼이 '종료' 버튼으로 바뀐다. 이 버튼에는 멘토링 시간이 같이 기재되어 있다. (그림 3의 세 번째 그림 참고) 종료 버튼을 누르면 멘토링 시간 기록이 종료되며, 데이터베이스로 멘토링 시간이 전송된다.

멘티는 '시작', '종료' 버튼 대신 '출석' 버튼이 존재한다. (그림 3의 첫 번째 그림을 참고) 멘티가 출석 버튼을 누르면 멘토에게 출석 요청을 보낼 수 있다. 멘토는 출석 요청을 보낸 명단을 확인할 수 있으며, 그것을 수락 또는 거절할 수 있다. 멘토가 출석 요청을 한 명도 허가하지 않으면 시수가 산정되지 않기 때문에, 출석 체크 문제를 자연스럽게 막을 수 있다.

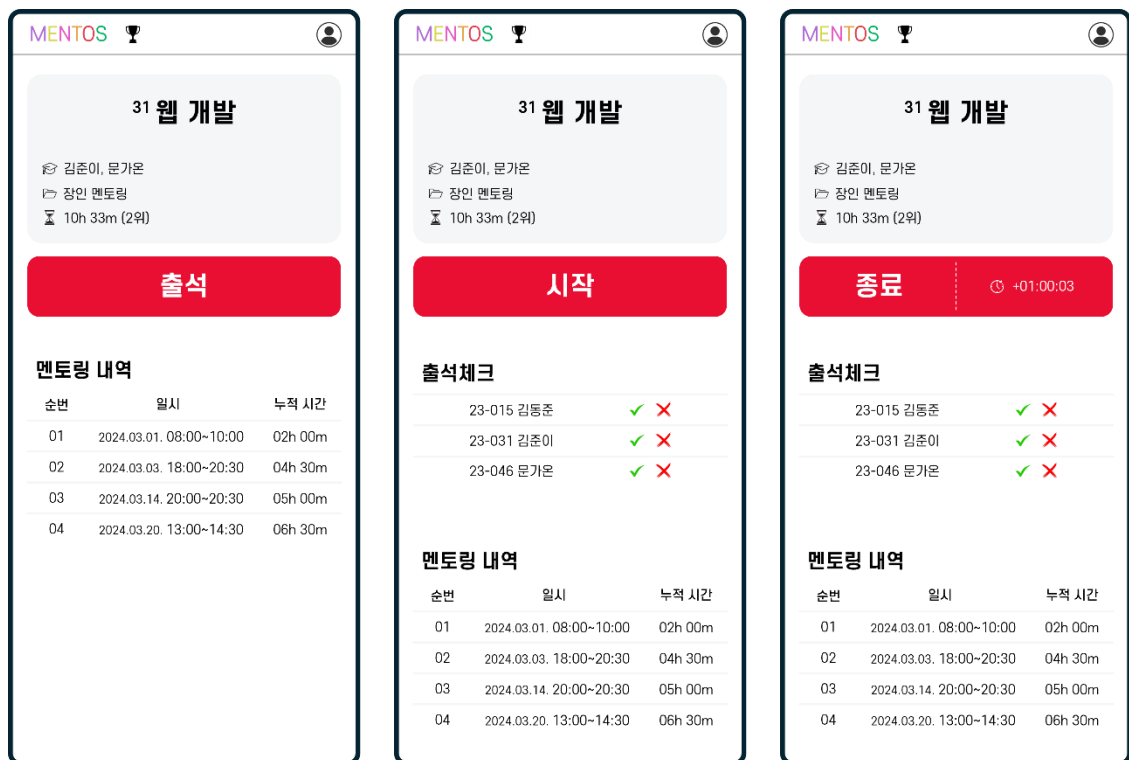


그림 3. MENTOS의 멘토링 관리 화면

### 3.3. 유저 프로필

헤더의 우측 상단 이미지를 누르거나, 다른 학생의 이름을 누르면 각각 자신의, 그 학생의 프로필로 이동하게 된다.

사용자의 프로필에는 기본적으로 해당 사용자의 학번과 이름, 멘토 랭킹(이때, 자신이 참여 중인 멘토링 중 가장 순위가 높은 멘토링의 순위로 반영된다.)과 멘티 랭킹이 적혀 있다. (그림 4 참고) 아래에서는 멘토 활동 내역과 멘티 활동 내역을 확인할 수 있다. 각 활동 내역은 학기별로 나누어 확인할 수 있으며, 어떤 멘토링에 언제 참여했는지 간략하게 볼 수 있다.

### 3.4. 랭킹

헤더의 랭킹 이미지를 누르면 랭킹 화면으로 이동할 수 있다.

랭킹은 멘토와 멘티로 나누어져 있으며, 멘토 랭킹은 멘토링 활동 시간 순서대로, 멘티 랭킹은 멘토링 참여 횟수 순서대로 정렬되어 보인다. (그림 5 참고)



그림 4. MENTOS의 유저 프로필 화면

MENTOS		
랭킹		
멘토	멘티	
순위	멘토링	활동 시간
1	46 정보과학1	23h 46m
2	31 웹 개발	1h 23m
3	15 LaTeX	0h 31m
4	114 수리생물학	0h 15m
5	23 정보과학1	0h 12m
6	45 수학1	0h 11m
7	50 한과영에서 살아남기	0h 9m

MENTOS		
랭킹		
멘토	멘티	
순위	학번/이름	참여 횟수
1	23-046 문가은	23회
2	23-031 김준이	22회
3	23-015 김동준	19회
4	23-035 김창하	17회
5	23-114 차재운	12회
6	23-029 김준성	9회
7	23-063 손민성	6회

그림 5. MENTOS의 랭킹 화면

## 4. 개발 환경

MENTOS는 사용자끼리 상호작용하는 기능을 지원해야 한다. 이는 하나의 서버에 데이터를 저장하고, 그것이 사용자 클라이언트를 연결하는 방법으로 구현할 수 있다. 모바일 앱, 웹 등 다양한 플랫폼이 사용자 클라이언트로서 사용될 수 있지만, 이 프로젝트의 경우 웹사이트를 이용하는 것이 적합하다. 웹사이트는 모바일, 태블릿, PC 등 모든 환경에서 접근할 수 있다는 장점이 있기 때문이다.

웹사이트 환경에서 구동되기에 필요한 하드웨어는 없으며, 웹 호스팅 등 웹을 구동시킬 수 있는 서버를 구하여 배포할 예정이다.

웹사이트는 크게 사용자에게 인터페이스를 보여주는 프론트엔드와, 데이터를 처리하고 클라이언트를 연결해주는 백엔드로 나뉜다. 이제 프론트엔드와 백엔드의 개발 환경에 대해서 각각 이야기해 보겠다.

### 4.1. 프론트엔드

프론트엔드 개발 환경을 세 단어로 요약하자면, React, TypeScript, Vite라고 할 수 있다.

#### 4.1.1. React

React는 Facebook에서 개발한 UI 라이브러리이다. 이것의 가장 큰 장점은 변경된 요소만 감지하여 업데이트

트한다는 것이다. 기존 Vanilla JS를 이용한 웹사이트는 요소를 변경할 때마다 전체 화면을 새로 그리곤 했다. 하지만 React의 경우 가상 DOM을 이용하여 효율적이고 빠르게 변경된 요소만을 다시 렌더링한다. 이외에도 비슷한 요소를 값만 변경하여 다시 사용할 수 있다는 점에서 요소의 재사용성이 높으며, 가장 많이 사용하는 라이브러리인 만큼 개발 생태계가 넓다는 등의 다양한 장점이 존재한다.

#### 4.1.2. TypeScript

React는 기본적으로 JavaScript 라이브러이지만, 우리는 JavaScript가 아닌 TypeScript를 이용할 것이다. TypeScript는 JavaScript의 확장판 격의 언어로, 이름에서 볼 수 있듯 JavaScript에 타입을 부여한 언어이다. JavaScript의 단점은 타입이 동적이고 형 변환이 너무나도 자유롭다는 것이었다. 예를 들어, “3”-“2”를 계산하면 다른 언어의 경우는 에러를 반환하겠지만, JavaScript에서는 Number 타입인 1이 나온다. 이렇게 타입에 얽매이지 않는다는 특징은 JavaScript가 웹 개발에서 엄청난 점유율을 차지할 수 있었던 원인이기도 하지만, 단점도 분명히 존재한다. 컴파일 과정이 엄격하지 않다는 말은 다른 말로 하면 실행 중에 문제가 생길 확률이 높아진다는 뜻이다. 따라서 엄격한 컴파일 검사를 통해 실행 중에 생길 문제를 최소화할 수 있도록 타입의 필요성이 부각된다.

TypeScript는 다른 강타입 언어(C++, Java...)처럼 타입 검사가 엄격할 뿐만 아니라, 또 다른 장점이 존재한다. 첫 번째로는 타입 추론이 가능하다는 것이다. C++을 예로 들자면, a라는 변수에 1을 담는 경우 `int a = 1;`과 같이 써서 반드시 타입을 명시해 두어야 하지만 TypeScript에서는 `const a = 1;`처럼 타입을 꼭 명시해 주지 않아도 컴파일러가 타입을 추론해 준다. 두 번째로는 객체 지향 언어라는 이름에 맞게 동적인 타입을 만들 수 있다는 점이다. 다른 언어는 객체의 타입이 어떤 클래스의 인스턴스라는 정보를 제공하는 것이 고작이지만, TypeScript의 타입은 어떤 key를 선택적으로 가지는지 등의 폭넓은 정보를 제공한다. 이외에도 Microsoft가 개발하여 에디터 연동성이 높고, 개발 생태계가 넓다는 다양한 장점이 존재한다.

#### 4.1.3. Vite

마지막으로, Vite는 React 개발을 돕는 도구이다. React는 Webpack 기반의 라이브러리로, 소스 코드를 컴파일할 때 변경된 코드를 포함하여 전체 코드를 다시 컴파일한다. 하지만 Vite는 Native ESM 기반으로, 소스 코드를 컴파일할 때 변경된 코드가 포함된 부분만 컴파일하여 속도를 비약적으로 높인다. Vite는 컴파일 속도가 빠를 뿐만 아니라, 실시간 코드 변화를 감지하는 개발용 서버 기능도 지원하는 등 React 개발자에게 큰 편의성을 제공한다.

### 4.2. 백엔드

백엔드 개발 환경은 Node.js, TypeScript, Express, MongoDB라는 단어로 요약된다.

#### 4.2.1. Node.js

일반적으로 JavaScript는 웹의 V8 엔진을 이용하여 구동된다. 하지만 Node.js는 브라우저가 아니라 서버에서 JavaScript를 구동할 수 있게 해주는 런타임 환경이다. 이를 이용하면 Python의 Flask처럼 JavaScript로도 서버를 구축할 수 있다. 한편 JavaScript는 Python보다 속도가 훨씬 빠르기 때문에, 대용량 데이터를 다루는 경우 JavaScript를 사용하는 편이 훨씬 유리하다.



#### 4.2.2. TypeScript

TypeScript의 장점은 위에서 충분히 설명했지만, 프론트엔드와 백엔드에서 같은 언어를 쓴다는 것은 또 다른 시너지 효과를 발휘한다. 같은 언어를 사용하기 때문에 한 명의 개발자가 프론트엔드, 백엔드 모두를 개발(풀스택)할 수 있다는 장점이 있으며, 프론트엔드와 백엔드에서 공유하는 유사한 데이터를 하나의 타입으로 묶어서 관리할 수 있기도 하다. 개발자가 적은 MENTOS 프로젝트의 경우에는 모든 팀원이 프론트엔드와 백엔드를 개발할 수 있도록 같은 언어를 사용하는 것이 더욱 유리하다. 또한, 프론트엔드와 백엔드에서 타입을 공유할 수 있다는 것은 각각에서 타입을 체크할 필요가 없다는 뜻이므로 개발 시간을 단축시킬 수 있다.

#### 4.2.3. Express

Express는 Node.js 환경에서 서버를 구축할 수 있게 해주는 프레임워크이다. Express의 장점은 가장 단순한 서버 프레임워크라는 것이다. 우리는 이른 시간 안에 서버를 구축해야 하기에 간단한 프레임워크의 필요성이 강조된다. 또한 Express는 가장 높은 점유율을 보유하고 있어, 개발 생태계가 넓다는 장점이 존재한다.

#### 4.2.4. MongoDB

MongoDB는 데이터베이스 관리 시스템이다. 일반적으로 웹 개발에서 데이터베이스가 필요하다면 SQL을 이용하는 경우가 많은데, SQL은 문법이 너무 어렵고 다른 언어와 연계시키기 어렵다는 단점이 있다. MongoDB는 JSON을 이용하여 데이터베이스를 관리할 수 있고, 우리가 사용하는 TypeScript를 포함하여 여러 언어에서 쉽게 사용할 수 있는 라이브러리를 제공한다.

### 4.3. 팀 작업

마지막으로, 프로젝트 결과물과 직접적으로 관련 있지는 않지만, 이 프로젝트는 팀을 이루어 개발하는 것이기 때문에 파일 관리 시스템이 필요하다. 우리가 사용한 파일 관리 시스템은 Git이다. 한편 코딩 스타일은 개인마다 다르므로 팀 프로젝트에서는 가독성을 위해서 코딩 스타일을 통일시켜 주어야 할 필요가 있는데, ESLint는 그러한 역할을 수행해주는 도구이다. 각각을 소개해 보겠다.

#### 4.3.1. Git

Git은 여러 명의 개발자가 하나의 소스 코드를 개발할 때 사용하는 버전 관리 시스템이다. Git을 이용하면 다음과 같은 방법을 통해 간단하게 파일을 공유할 수 있다. 개발자가 로컬 저장소에서 코드를 편집하면 변경된 줄을 감지하여 원격 저장소로 올린다. 그러면 다른 개발자는 원격 저장소에서 변경된 코드를 가져와 로컬 저장소에 똑같이 적용할 수 있다. 이외에도 Branch 시스템을 통해 병렬 개발을 할 수 있는 등 팀 개발에 있어 유용한 기능을 제공한다.

#### 4.3.2. ESLint

ESLint는 JavaScript(또는 TypeScript)의 문법과 스타일을 검사해 준다. 코드가 지정해 둔 코딩 스타일에 위배되면 ESLint는 에러를 띄운다. 그 때문에 여러 명이 작업한 코드라 해도 일정한 코딩 스타일을 유지할 수 있으며, 한 사람이 짰 것처럼 가독성을 높일 수 있다.

## 5. 개발 일정

계획된 개발 일정을 표로 요약하면 다음과 같다. 프론트엔드는 F, 백엔드는 B로 나타낸다.

1주차 : 04/01~04/07	F. 메인 페이지 제작 B. 데이터베이스 구축
2주차 : 04/08~04/14	F. 메인 페이지 제작 B. 메인 페이지에서 사용되는 쿼리 제작
3주차 : 04/15~04/21	F. 멘토링 관리 페이지 제작 B. 멘토링 관리에 사용되는 쿼리 제작
4주차 : 04/22~04/28	F. 멘토링 관리 페이지 제작 B. 멘토링 관리에 사용되는 쿼리 제작
5주차 : 04/29~05/05	<b>중간 발표회, 피드백 수용</b>
6주차 : 05/06~05/12	F. 랭킹 페이지 제작 B. 랭킹 페이지에서 사용되는 쿼리 제작
7주차 : 05/13~05/19	F. 유저 프로필 제작 B. 유저 프로필에서 사용되는 쿼리 제작
8주차 : 05/20~05/26	베타 테스트, 기타 버그 수정
9주차 : 05/27~05/30	베타 테스트, 기타 버그 수정
05/31	<b>프로젝트 최종 발표회</b>