

최종 보고서



부산대학교 소프트웨어교육센터
PUSAN NATIONAL UNIVERSITY SOFTWARE EDUCATION CENTER

CCM 멘토링 시스템 <멘토와 멘티의 연결고리>



팀	명	M & M		
팀	원	201645825	이승윤	(인)
		201645819	심재영	(인)

목차

1. 개발 배경	2
가. 개발 배경	2
나. 개발 필요성	3
2. 개발개요	5
가. 개발목표	5
나. 개발기간	5
다. 개발내용	6
라. 활용방안	8
3. 상세설계	9
가. 시스템구조	9
나. 페이지 구조	10
다. 사용기술 및 알고리즘	11
4. 개발결과	18
가. 전체 시스템 흐름도	18
나. 개발 기능	19
5. 개발 일정 및 역할	29
가. 주요 개발 일정	29
나. 세부 서비스 세부 개발 일정	29
다. 구성원별 역할	30
6. 참고문헌	31

1. 개발 배경

가. 개발 배경

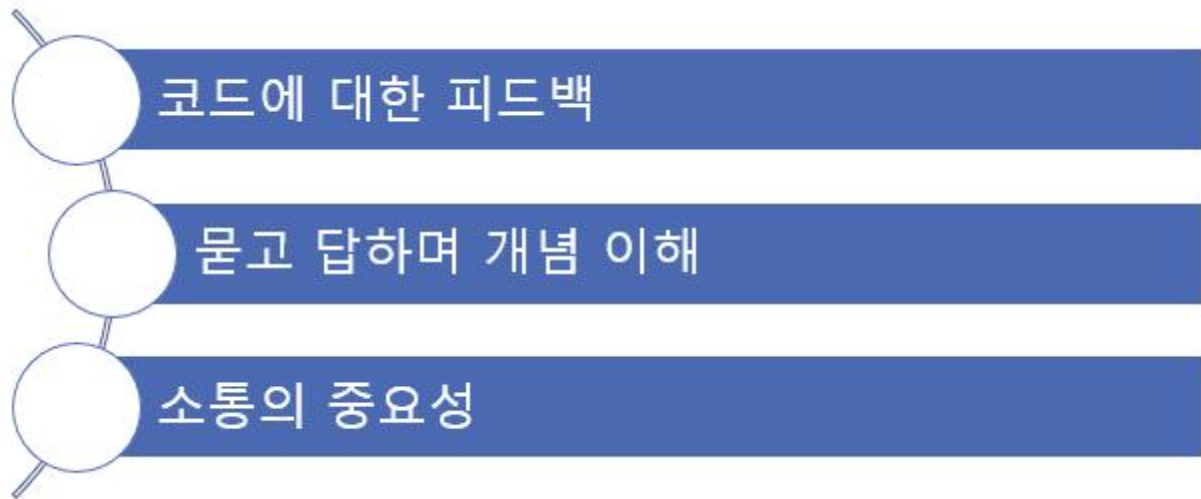


그림 1 개발 배경

평상시 과제를 하거나 수업에서 배운 이론을 가지고 실습할 때 혼자 해결하지 못하거나, 궁금한 코드에 대해 간단한 피드백을 받고 싶을 때가 많았다. 궁금한 부분에 대해 검색을 해봐도 부분적으로는 해결되지만, 완전히 해결하지 못하고 넘어가는 경우가 꽤 있었다. 내가 궁금한 부분에 대해 가장 잘 이해하고 답해줄 수 있는 사람은 같은 수업을 듣는 학생들이나 수업을 해주고 과제를 내주시는 교수님이라는 생각이 들었다. 그리하여 많은 학생이 궁금한 부분에 대해 빠르고 정확하게 피드백을 받고 소통함으로써 수업에 대한 이해도가 향상될 뿐만 아니라 학생 본인의 실력증진에 도움을 줄 것이라는 생각으로 멘토링 시스템을 개발하기로 결심하였다.

한때 혼자서도 모든 프로그램을 만들 수 있다고 생각한 적이 있었다. 혼자서 프로그램을 개발하면 처음부터 끝까지 구현에 대해 내가 책임질 수 있고 모든 기능 및 동작을 이해하고 있어 기능을 추가 또는 수정하기 쉽다. 그런데 개발하다 보니 여기저기서 코드를 가져오기 시작한다. 코드를 가져와서 내 프로그램에 맞게 수정하고 어떤 경우엔 라이브러리를 통째로 가져와 남이 구현해놓은 코드 덩어리를 사용하기도 한다. 1인 개발자는 정말 혼자 개발하는 것일까? 다른 사람과 함께 개발하는 것일까? 오랜 고민 끝에 함께 개발한다는 쪽의 손을 들어주기로 생각을 굳혔다. 처음부터 끝까지 모두 혼자 해결한다는 것은 흔치 않다. 결국엔 함께 만들어가게 된다. 그래서 소통의 의미가 중요하다.

학생 간에 서로 소통할 수 있는 어떤 시스템, 부산대판 스택 오버플로우를 만들고 싶었다. 서로 질문하고 답변하고 어떤 주제에 대해서 서로 대화하듯 의견을 주고받을 수 있는 웹사이트를 만들고 싶었다.

나. 개발 필요성

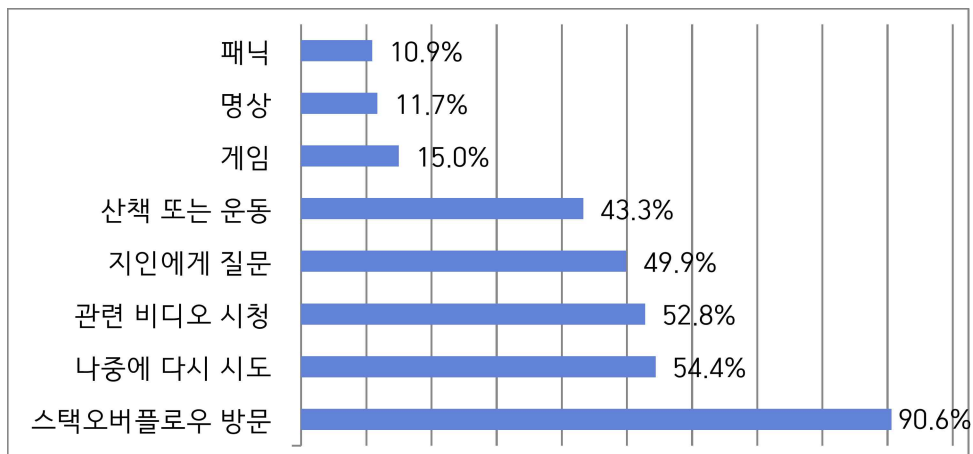


그림 2 프로그램 작성 중 막혔을 때 대처 방법 (중복 응답) [1]

스택오버플로우에서 개발자 64,416명을 대상으로 진행한 2020 개발자 설문에 따르면 프로그램 작성 중 막히는 것이 있으면 90.6%가 스택오버플로우에 방문하여 문제를 해결하며 이어서 54.4%는 나중에 다시 시도, 52.8%는 관련 비디오 시청, 49.9 %는 지인에게 질문하여 문제를 해결하는 것으로 나타났다.

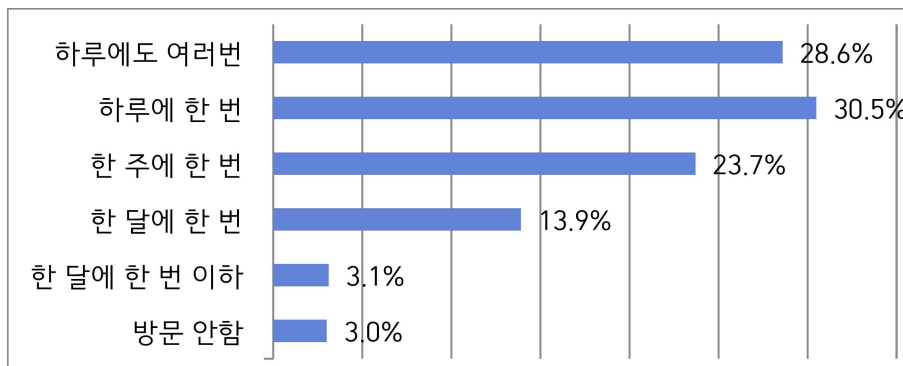


그림 3 스택오버플로우 방문 주이 [1]

또한, 같은 조사에서 개발자들의 28.6%가 하루에도 여러 번, 30.5%가 하루에 한 번, 23.7%가 한 주에 한 번 스택오버플로우에 방문하는 것으로 드러났다.

앞의 두 그래프를 통해 90.6%의 개발자들이 문제 해결을 위해 질의응답 사이트를 찾으며 82.8%의 개발자가 적어도 일주일에 한 번은 질의응답 사이트를 방문하는 것을 파악할 수 있다. 위 조사를 통해 개발자를 위한 질의응답 사이트가 필요하다는 사실을 알 수 있다.

코딩하면서 궁금한 점이 생겼을 때 스택오버플로우 같은 범용 질의응답 사이트에서는 본인이 궁금한 상황과 일치하는 해답을 찾을 수 없는 경우가 많다. 부분적으로 검색을 하여 비슷한 내용을 찾아 문제를 해결해야 하기에 완전한 해결이 힘든 경우도 있다. 이러한 부분을 멘토링 시스템을 활용해서 해결하고자 한다.

가장 근접하게 학생의 상황을 이해할 수 있는 사람은 같은 수업을 듣는 학생일 수 있다. 특히 같은 과제를 수행하고 있거나 관심 있는 분야일 경우 다른 학생들의 문제를 해결하는데 큰 도움을 줄 수 있다. 학교 내 학생들만 참여하는 공간을 통해 목표 대상이 분명한 질문에 대해 더 정확한 대답을 확보할 수 있다.

기존의 질의응답 사이트는 한 질문에 대해 원하는 대답이 나오지 않거나 답변에 대해 이해하지 못할 경우 다시 질문하기가 어렵게 구성되어 있으며 비슷한 질문 업로드 금지 등 질문과 답변에 관한 규정으로 인한 제약사항이 많다. 대화형 사용자 인터페이스를 통해 한 질문에 여러 학생이 답변하고 문제가 해결되지 않으면 다시 질문할 수 있으며 잘못된 부분에 관해서도 토론하고 서로 도움을 받을 수 있는 공간을 만들어 해결하려 한다.

또 단순한 질의응답 시스템을 넘어 멘토링 팀 기능을 통해 어떤 분야에 대해 사용자가 멘토가 되어 팀을 만들면 다른 사용자들이 팀에 참가하여 함께 주제에 대해 토론하고 문제를 해결할 수 있는 환경을 제공하고자 한다. 같은 분야에 대한 그룹화를 통해 발생하는 문제점들을 공유함으로써 훨씬 더 빠르게 학생들의 문제를 해결할 수 있다.

요즘 원격수업으로 인해 직접 교수님에게 질문하기 어려운 부분이 있다. 멘토링 시스템을 통해서 직접 대면하지 않고도 궁금한 부분에 대해 소통함으로 본인의 문제를 해결할 수 있다.

멘토링의 의미도 확장되고 있다. 과거 풍부한 경험과 지혜를 가진 사람이 1:1로 전수해주는 것에서 현대에는 매우 뛰어나지 않은 사람도 멘토 역할을 수행할 수 있고 비슷한 실력을 갖춘 사람끼리 멘토, 멘티 역할을 바꾸어가며 진행하기도 하며 다수가 참여하는 팀 멘토링 프로그램이 활성화되고 있다. 이에 맞게 멘토-멘티 간의 관계에 얽매이지 않고 누구나 질문하고 대답할 수 있으며 어떤 주제에 대해 토론할 수 있는 새로운 모임 공간을 구상하였다.

멘토링을 통해 소통의 장이 활성화되어 멘티는 프로그래밍에 대한 흥미와 자존감을 높이고 멘토는 본인의 지식을 정리하고 다시금 되짚어볼 수 있는 서로에게 윈-윈(Win-Win) 효과를 일으킨다. 비슷한 실력을 갖춘 사람끼리도 서로 도움을 받으며 문제를 해결해나가는 성취감을 맛볼 수 있다. 멘토링 프로그램을 통해 학생들은 자신의 관심 분야에 더욱 집중할 수 있고 토론을 통한 문제 해결로 프로그래밍 실력 향상을 기대할 수 있다.

2. 개발개요

가. 개발목표

학생들이 프로그래밍을 하다 궁금한 것이 생기거나 막혔을 때, 프로그래밍에 대한 질문을 하고 답변을 받는 질의응답 기능과 서로 주제에 대해 토론하고 팀 프로젝트를 진행할 수 있는 멘토링 기능을 갖춘 웹사이트이다.

기존의 질의응답 사이트(대표적으로 스택오버플로우)에 존재하는 질의응답 기능에 국한되는 것이 아닌 <CCM 멘토링 시스템>은 멘토링 프로그램을 통해 학생-학생 간의 멘토-멘티 시스템을 활성화한다. 학교 내 학생들만 참여하는 멘토링 프로그램을 통해 본인 상황을 가장 근접하게 이해할 수 있는 멘토들에게 질문함으로써 질문에 대한 부분적 해결보다는 더 정확한 대답에 근접하여 완벽한 해결로 나아갈 수 있도록 한다.

또한, <CCM 멘토링 시스템>에 제공되는 팀 기능을 통하여 본인이 공부하고 싶은 분야나 관심 있는 분야의 팀에 참가하여 같은 분야에 대한 멘토링(그룹화)을 한다. 더욱 세밀한 그룹화를 통해 세부적인 분야의 주제에 대해 토론하고 발생한 문제점들을 공유함으로써 훨씬 더 빠르게 문제를 해결할 수 있도록 돕는다.

나. 개발기간

- 2020-10-14~2020-11-17 (약 한 달간)



그림 4 개발 기간에 따른 개발 순서

- 1) 주요 제공 서비스 확정
- 2) 시스템 설계
- 3) 개발환경구축
- 4) 1차 시연
- 5) 레이아웃 제작 완료
- 6) 데이터베이스 제작 완료
- 7) 최종 시연
- 8) 개발 완료 및 결과 보고서 제출

다. 개발내용

1) 주요 서비스



질문



답변



멘토

그림 5 주요 서비스

가) 질문 하기

학생이 프로그래밍 중 궁금한 것에 대한 질문을 등록한다. '제목', '내용', '태그' 3가지를 입력하여 질문 등록이 가능하다. 특히 '태그' 기능을 통해 질문내용에 해당하는 분야를 태그로 등록함으로써 그 분야에 관심 있는 사람들에게 답변을 얻을 수 있도록 유도할 수 있다.

나) 답변 하기

사이트에 등록된 회원이면 누구나 등록된 질문에 대해 답변을 할 수 있다. 답변하기 창에 들어가면 사이트에 등록된 질문 목록을 볼 수 있다. 추가로 '일반' 질문 목록과 '팀' 질문 목록을 분류하여 팀에 등록된 회원은 팀에 해당하는 질문 목록을 추가로 볼 수 있으며 그 질문들에 대한 답변 또한 가능하다.

또한, 각 질문과 질문에 달린 답변에는 '좋아요' 아이콘과 '싫어요' 아이콘이 존재하고 본인이 만족스러운 답변을 얻었거나 본인에게 공감되는 질문, 등 원하는 아이콘을 선택하여 각 질문과 답변에 대한 점수를 매길 수 있다. 아이콘('좋아요', '싫어요')을 통해 매긴 '질문 점수'와 '답변 점수'는 각 질문과 답변을 작성한 회원들의 데이터베이스에 저장된다.

다) 멘토 찾기

사이트에 등록된 멘토들의 간단한 프로필 정보(프로필 사진, 이름, 질문 점수, 답변 점수)를 확인할 수 있는 목록이 출력된다. 출력은 '답변 점수'가 높은 기준으로 정렬이 되어있다. 멘티들은 멘토들의 '답변 점수'를 통해 신뢰도가 높은 멘토들을 선택하여 멘토가 운영하는 팀에 등록하여 질문할 수 있다. 각 멘토에게는 '프로필 더보기'라는 버튼이 존재하고 그 버튼을 통해서 멘토의 세부정보(회원 기본정보(이름, ID, 이메일, 전화번호), 멘토가 등록된 팀, 가장 최근에 작성한 각 3가지 질문, 답변 리스트)들을 확인할 수 있다.

2) 추가 기능



검색



로그인



팀



데이터베이스

그림 6 추가 기능

가) 검색 기능

모든 창에서 검색 기능을 이용할 수 있다. 본인이 관심 있는 분야에 대한 검색을 통해 그 분야에 해당하는 질문을 볼 수 있고 그 질문들에 답변할 수 있다. 또한, 질문자의 입장에서 볼 땐 본인과 비슷한 상황에 있는 질문을 검색함으로 궁금한 점에 대한 답변을 손쉽게 얻을 수 있다.

나) 로그인 기능

기본적으로 질문 및 답변 작성과 수정, 삭제를 위해 로그인 기능을 구현하였다. 멘토링 기능을 위해 사용자별로 팀을 분류하고 정보를 저장하기 위해서도 로그인 기능이 반드시 필요하다.

다) 팀 기능

본인이 관심 있거나 배우고 싶은 팀에 등록하거나 본인이 원하는 멘토가 운영하는 팀에 학생들이 등록하여 팀이 다루는 분야에 대한 질문과 답변의 과정을 거치며 각 주제에 대해 토론함으로 세부분야에 대한 지식을 습득하고 이해도를 높일 수 있다.

또한, 팀을 운영하는 멘토는 본인이 운영하는 팀에 대해 책임감을 느끼고 운영함으로써 더욱 멘토-멘티 관계를 돈독하게 하며 멘토링 시스템을 활성화할 수 있도록 한다.

라) 데이터베이스

각 사용자의 해당하는 데이터는 서버의 데이터베이스에 기록한다. 데이터를 서버 중심으로 통합하여 저장하고 관리함으로써 사이트 이용자로부터의 서로 다른 처리요구를 받아들여 각각의 이용 방법에 적합한 조작방식을 제공하기 위해서 데이터베이스 시스템을 이용하여 사이트 내의 데이터 처리기능을 수행한다.

라. 활용방안

1) 학생들의 관심도 동향 파악

<CCM 멘토링 프로그램> 사이트 내에 등록된 질문과 답변을 파악함으로써 학생들의 질문 질의 빈도를 통해 학생들의 관심도 동향을 파악해볼 수 있다.

그 예로 아래 [그림 7]에 나와 있는 그래프에서 볼 수 있듯이 개발자들이 가장 많이 사용하는 프로그래밍 질의응답 사이트인 스택오버플로우에서도 사용자들의 프로그래밍 언어 관련 질의 빈도 추세를 매년 파악하는 것을 볼 수 있다.

학생들의 관심 분야에 대한 동향을 파악하여 학교 차원에서 관심도가 높은 항목에 대해 새로운 과목을 개설하거나 특강을 열어 학생들의 요구를 충족해줄 수 있다.

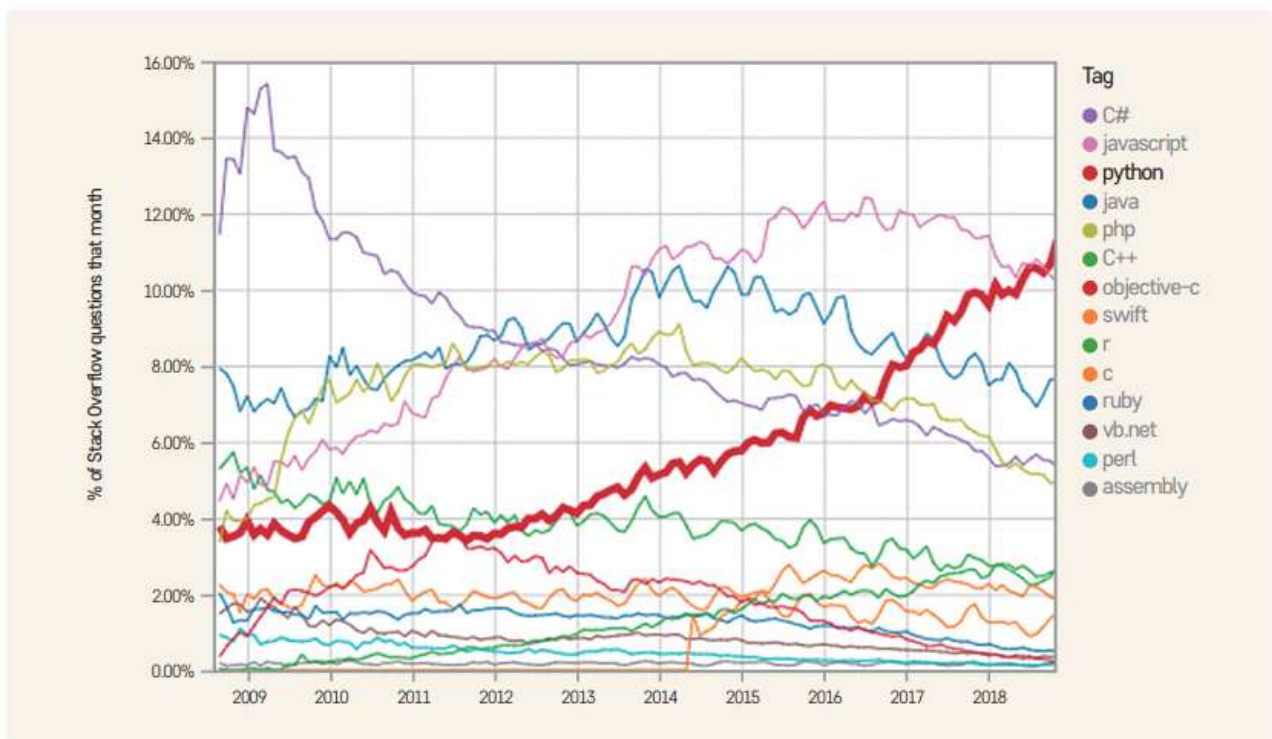


그림 7 Stack Overflow 내 프로그래밍 언어 관련 질의 빈도 추세 [1][2]

2) 수업의 질 향상

<CCM 멘토링 시스템> 사이트에 존재하는 멘토링 프로그램을 교수들이 멘토로 등록하여 진행 가능하다. 교수들은 본인이 수업에 관련한 팀을 생성하여 학생들이 그 팀에 등록할 경우, 팀 기능 안에서 학생들은 서로 소통하며 질의응답을 진행할 것이다. 교수는 자주 나오는 질문이나 학생(즉, 멘티)들이 이해도가 떨어진다고 생각하는 부분에 대해 앞으로 진행되는 수업에 추가로 설명함으로써 수업의 완성도를 높여갈 수 있다.

3. 상세설계

가. 시스템구조



사용자는 웹 브라우저를 통해 CCM 멘토링 시스템에 접속할 수 있다. CCM 멘토링 시스템은 웹 서버와 데이터베이스 서버로 구성되며 사용자가 각 페이지에 접근할 때 웹 서버에서 필요한 파일을 전송받으며 데이터베이스에 저장된 데이터가 필요한 경우 웹 서버를 통해 데이터베이스 서버에서 데이터를 동적으로 전송받는다.

웹 서버는 Node.js와 express를 기반으로 동작하며 페이지 주소에 따라 구분하여 적절한 동작을 수행한다. 데이터베이스 서버는 Mysql을 기반으로 동작하며 express가 판별한 페이지 주소가 post 요청이고 데이터를 요청하도록 설정된 경우 데이터베이스에 저장된 데이터를 전송한다.

각 페이지는 html 또는 ejs 파일을 통해 표시(render)되고 페이지 스타일은 해당 파일 또는 별도의 css 파일에 정의된 스타일을 적용한다. 동적으로 수행되어야 할 작업은 해당 파일 또는 자바스크립트(js) 파일에 작성된 코드를 바탕으로 수행된다.

나. 페이지 구조

멘토링 시스템은 크게 기본 구조, 로그인, 질문 및 답변, 멘토(팀) 찾기로 구분할 수 있다.



그림 9 페이지 구조

1) 기본 구조

기본 구조는 모든 페이지에 적용되는 헤더(header), 네비게이션(navigation), 푸터/footer)로 구성된다. 헤더는 네비게이션, 사이트 로고, 로그인 팝업, 알림, 검색 기능을 제공한다. 헤더의 네비게이션 버튼을 클릭하면 네비게이션이 출력되고 메인 페이지와 사이트 핵심 기능인 질문 하기, 답변 하기, 멘토 찾기로 바로 이동할 수 있다. 푸터는 아래쪽에 제작자 정보를 출력한다.

2) 로그인

멘토링에서 사용자 구분을 위해 로그인 기능이 필요하다. 처음 사용자는 정보를 입력하고 회원 가입 후 로그인할 수 있다. 로그인한 사용자는 자신의 정보를 조회할 수 있으며 가입한 팀과 질문, 답변 점수 및 최근 항목도 조회할 수 있다.

3) 질문 및 답변

사용자가 질문을 게시하면 질문 목록에서 질문을 조회할 수 있다. 조회된 질문 중 관심 있는 질문을 클릭하면 상세 질문이 표시되며 해당 질문에 대한 수정 및 삭제가 가능하다. 질문 하단에는 게시된 답변이 출력되며 자신의 답변을 게시할 수 있다. 단, 사용자가 팀 게시판에 게시한 질문은 표시되지 않으며 해당 팀이 아니면 상세 질문 내용 및 답변을 확인할 수 없다.

4) 멘토(팀) 찾기

전체 멘토들의 목록을 살펴보고 검색할 수 있다. 멘토 목록에서 원하는 멘토를 찾아 정보를 확인하여 팀에 가입할 수도 있고 팀 목록에서 원하는 팀을 골라 팀에 가입할 수도 있다.

다. 사용기술 및 알고리즘

1. material-components-web

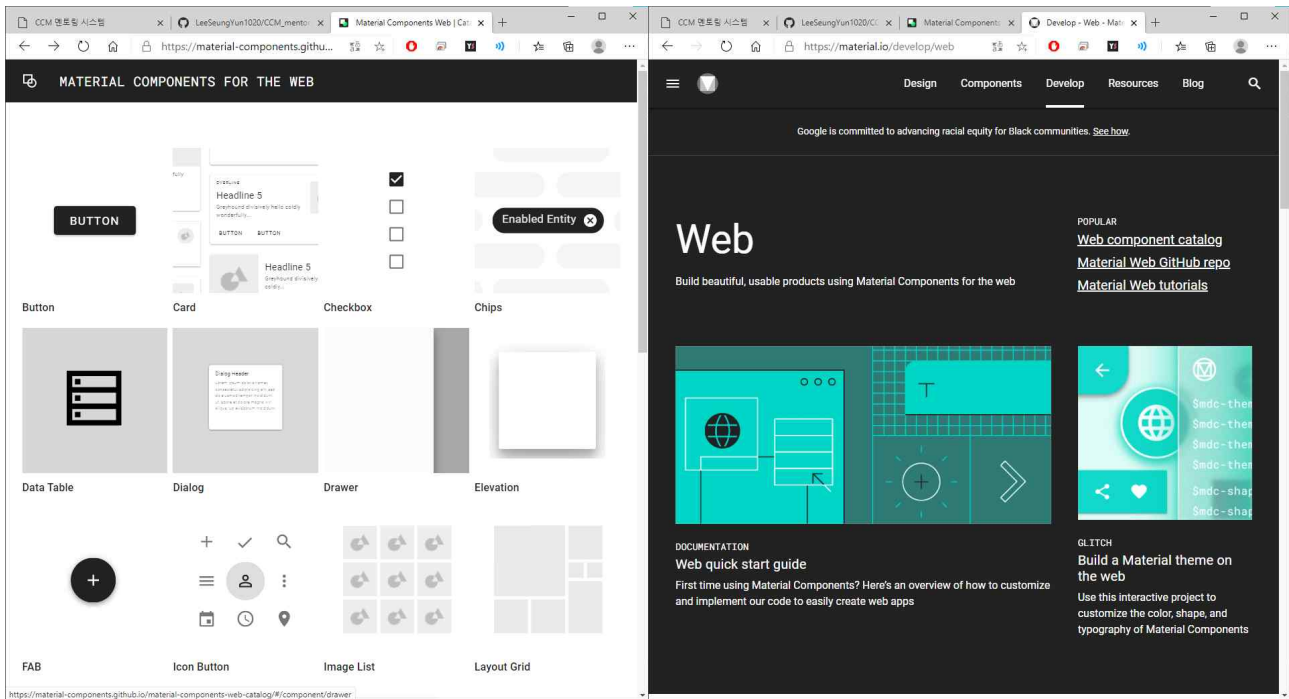


그림 10 material-components-web

전반적인 페이지 디자인을 위해 사용하였다. 네비게이션 서랍, 버튼, 텍스트 필드, 카드, 칩, 아이콘, 메뉴, 스위치 등 요소의 레이아웃 디자인 적용을 위해 사용하였다.

2. Node.js

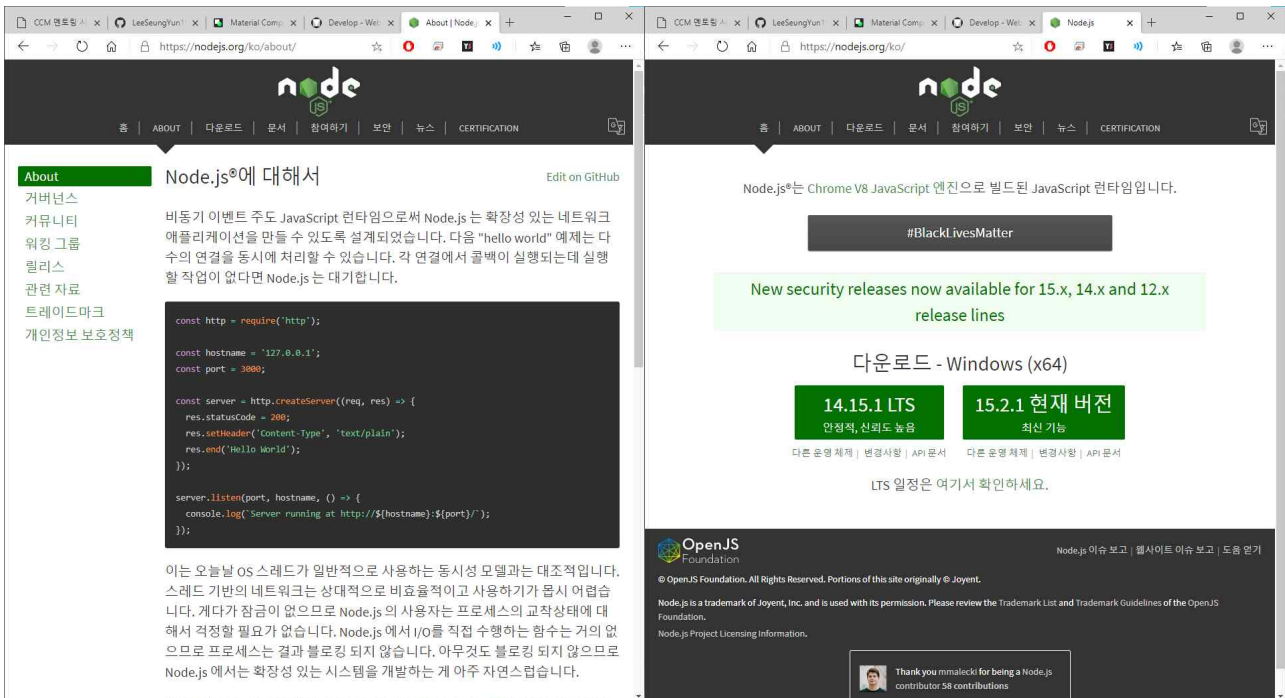


그림 11 Node.js

자바스크립트로 서버 코드를 작성할 수 있도록 하는 라이브러리 비동기 이벤트 기반 자바스크립트 런타임으로 네트워크 어플리케이션 설계에 유용하다.

3. Express

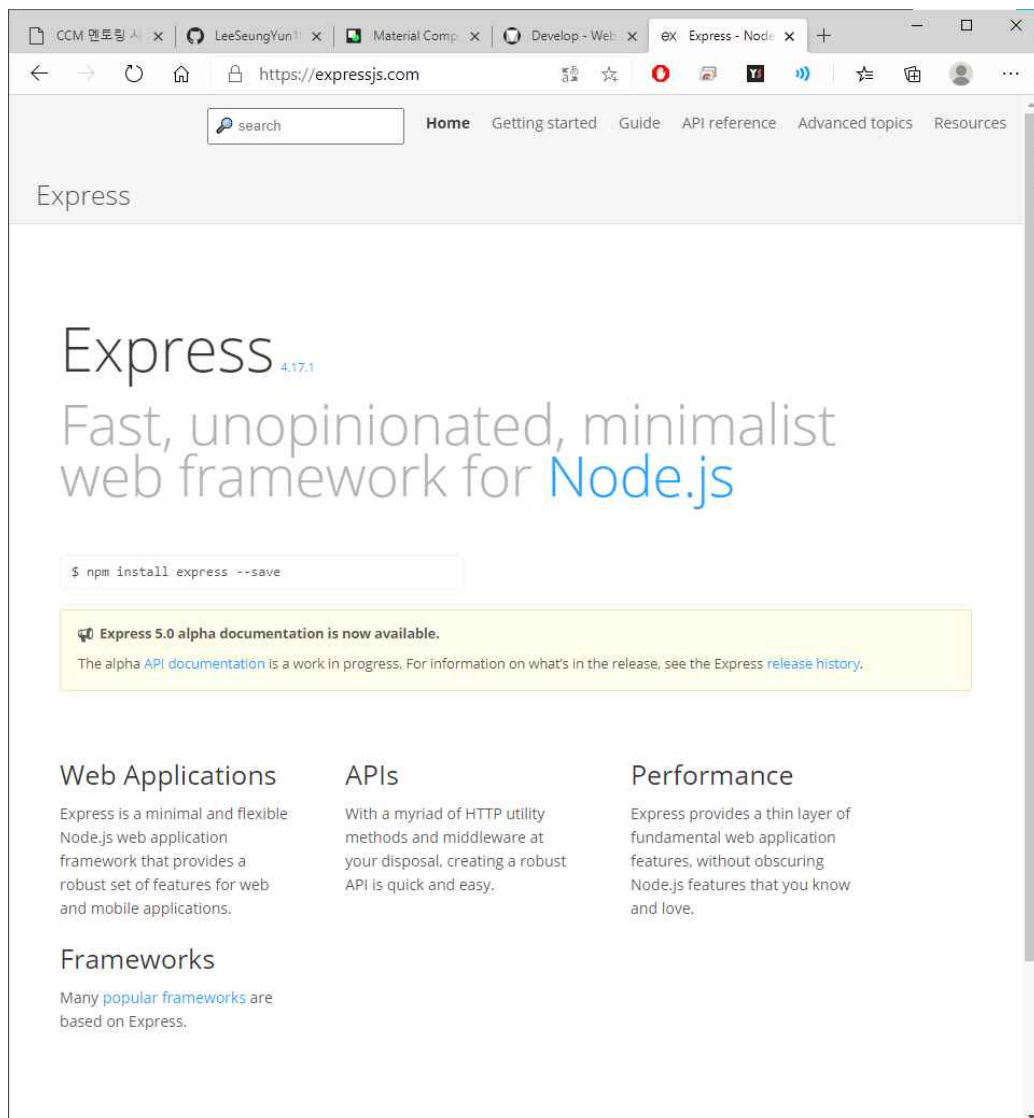


그림 12 Express

Express는 Node.js 기반 웹프레임워크이다. 미들웨어와 주소 기반의 라우팅 기능을 지원한다. Node.js를 이용하여 주소를 기반으로 각 페이지가 작동하도록 구현하였다. 특히 데이터를 받아올 때 라우팅 기능과 인자(parameter) 전달 기능을 유용하게 사용하였다.

4. Quill.js

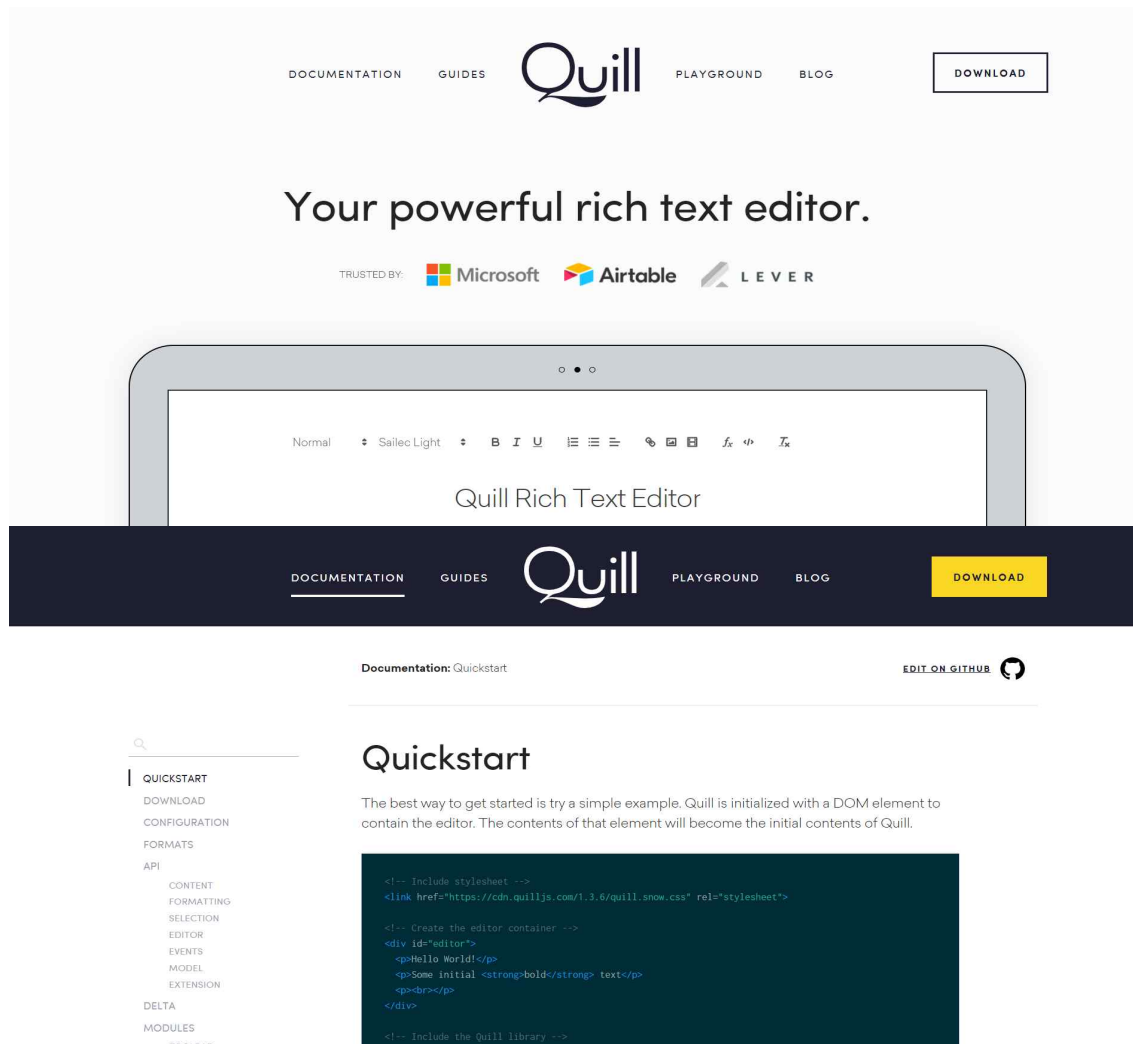


그림 13 Quill.js

Quill은 현대 웹을 위해 제작된 오픈소스 편집기다. 모듈식 아키텍처와 표현형 API를 통해 필요에 따라 완벽하게 맞춤 설정을 한다. Quill.js를 이용하여 풍부한 텍스트 에디터 기능 특히 가장 중요한 게시글에 코드 입력 기능을 구현하였다.

5. highlight.js

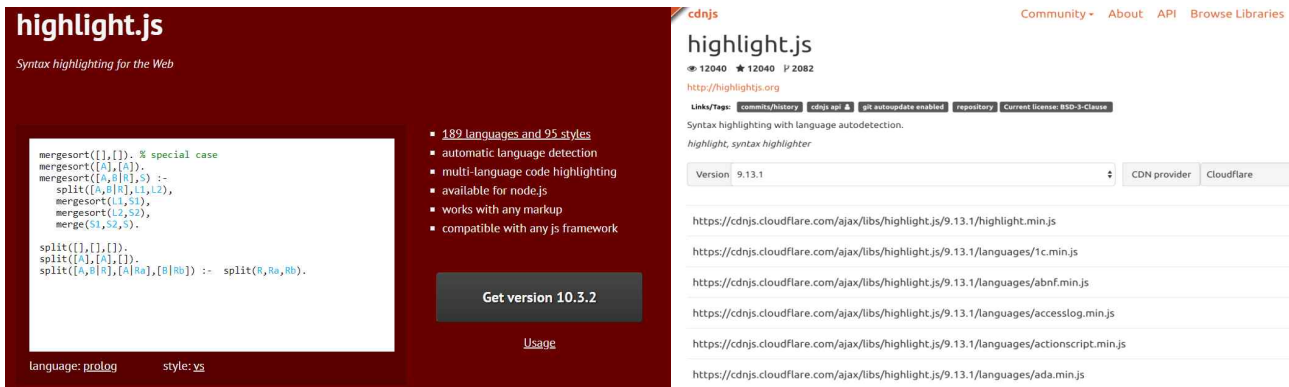


그림 14 highlight.js

글을 작성할 때 코드 부분이 실제 에디터에서 보이는 것처럼 강조하는 방법으로 highlight.js를 사용하였다.

6. Passport

Passport

Simple, unobtrusive authentication for Node.js

Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped in to any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more.

```
app.js - vim

passport.authenticate('facebook');
```

Overview

Passport is authentication middleware for Node. It is designed to serve a singular purpose: authenticate requests. When writing modules, encapsulation is a virtue, so Passport delegates all other functionality to the application. This separation of concerns keeps code clean and maintainable, and makes Passport extremely easy to integrate into an application.

In modern web applications, authentication can take a variety of forms. Traditionally, users log in by providing a username and password. With the rise of social networking, single sign-on using an OAuth provider such as Facebook or Twitter has become a popular authentication method. Services that expose an API often require token-based credentials to protect access.

Passport recognizes that each application has unique authentication requirements. Authentication mechanisms, known as *strategies*, are packaged as individual modules. Applications can choose which strategies to employ, without creating unnecessary dependencies.

Despite the complexities involved in authentication, code does not have to be complicated.

```
app.post('/login', passport.authenticate('local', { successRedirect: '/',
failureRedirect: '/login' }));
```

그림 15 Passport.js

Passport는 Node.js용 인증 미들웨어다. 로그인 / 로그아웃 / 접근제한과 같은 인증 기능을 구현할 때 사용하였다.

7. MySQL

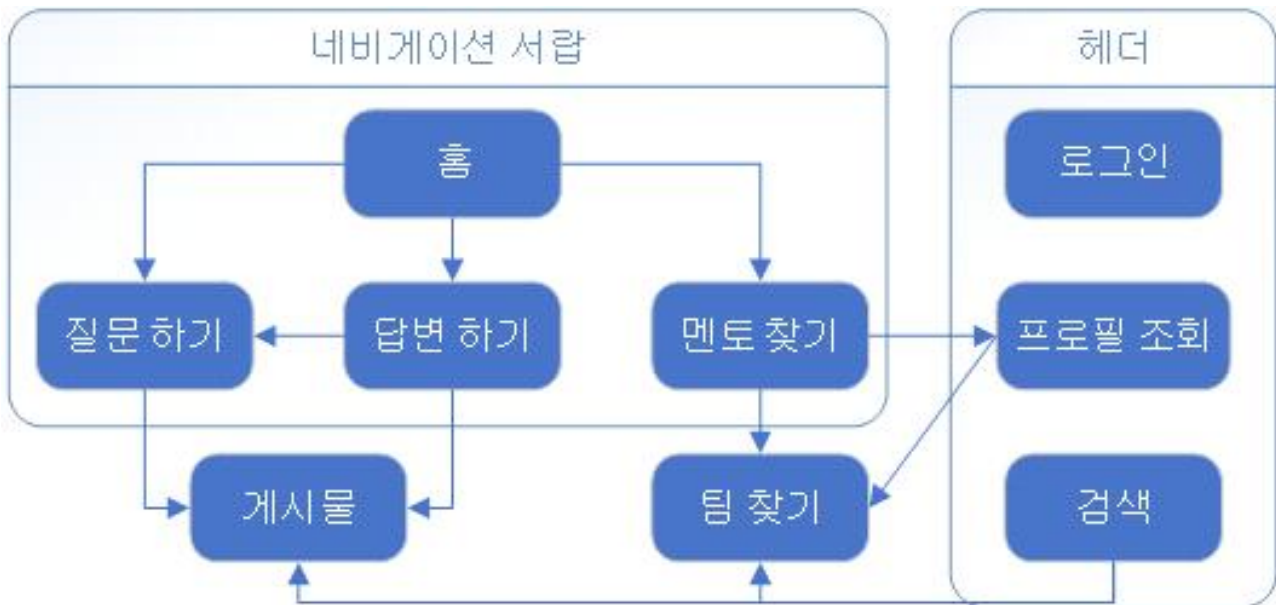


그림 16 MySQL

MySQL은 세계에서 가장 많이 쓰이는 오픈소스 관계형 데이터베이스 관리 시스템(RDBMS)이다. MySQL을 기본 데이터베이스로 채택하여 사용하였다.

4. 개발결과

가. 전체 시스템 흐름도



홈, 질문 하기, 답변 하기, 멘토 찾기는 네비게이션 서랍에 포함되어 어디에서든 서랍을 열어 바로 이동할 수 있다. 또한 로그인(로그인하지 않은 경우), 프로필 조회(로그인한 경우), 검색은 헤더에 포함되어 있어 어떤 페이지에서나 이동할 수 있다. 검색 기능을 사용하면 게시물과 팀이 검색되고 각 항목으로 이동할 수 있다.

홈 화면은 질문 하기, 답변 하기, 멘토 찾기로 구성되며 네비게이션 서랍에 있는 항목과 같다. 질문 하기에서는 게시물 작성 페이지가 표시되고 작성 완료 시 게시물이 표시되며 답변 하기에서는 게시물 목록 페이지가 표시되고 게시물 페이지로 이동할 수 있다. 멘토 찾기에서는 멘토 목록이 표시되고 각 멘토의 프로필 정보를 조회할 수 있으며 팀 찾기를 통해 원하는 팀을 찾고 가입할 수 있다.

나. 개발 기능

1) 헤더 (Top App Bar)

헤더는 크게 3가지 타입으로 구성된다.

1. 로그인하지 않은 경우

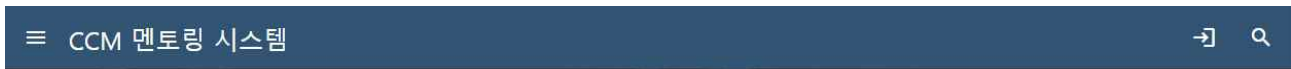


그림 18 헤더(처음) - 좌측부터 네비게이션 버튼, 로고, 로그인 버튼, 검색 버튼

2. 로그인한 경우



그림 19 헤더(로그인) - 좌측부터 네비게이션 버튼, 로고, 로그인 팝업 버튼, 알림, 검색 버튼

3. 검색 버튼을 누른 경우

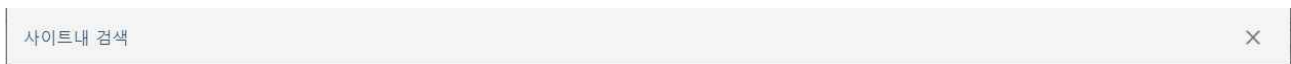


그림 20 헤더(검색) 좌측부터 검색 창, 취소 버튼

네비게이션 버튼을 클릭하면 화면 좌측에 네비게이션 서랍(navigation drawer)이 표시된다. 로고를 클릭하면 홈으로 이동한다. 로그인 버튼은 로그인하지 않은 경우에만 표시되며 로그인 창으로 이동한다. 로그인 팝업 버튼은 로그인한 경우에만 표시되며 클릭하면 사용자 사진, 이름, ID, 로그아웃 버튼, 프로필 정보 버튼이 있는 팝업이 나타난다. 검색 버튼을 누르면 검색 창과 취소 버튼이 나타나며 검색 창에 원하는 항목 입력 후 엔터 키를 누르면 검색 창으로 이동한다.



그림 21 헤더(로그인 팝업)

2) 네비게이션

헤더에 표시되는 네비게이션 버튼을 누르면 어느 화면에서나 네비게이션 서랍을 열고 닫을 수 있다. 홈, 질문 하기, 답변 하기, 멘토 찾기로 바로 이동할 수 있다.



그림 22 네비게이션 서랍

3) 홈 화면

메인 화면은 3가지 주요 기능 질문하기, 답변하기, 멘토 찾기로 구성된다.

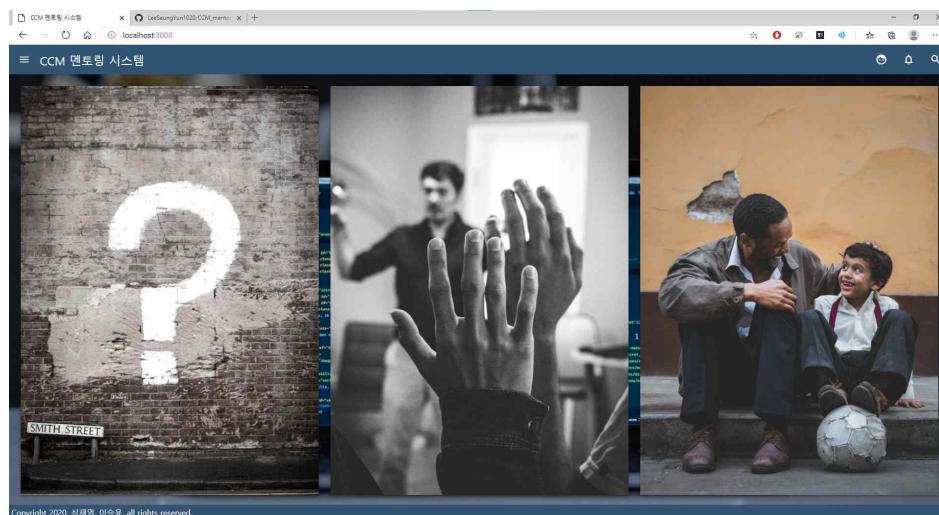


그림 23 홈 화면 – 좌측부터 질문 하기, 답변 하기, 멘토 찾기

각 기능을 강조하기 위해 관련 깊은 이미지를 사용하였으며 마우스를 해당 사진에 가져가면 아이콘과 이름이 표시된다.

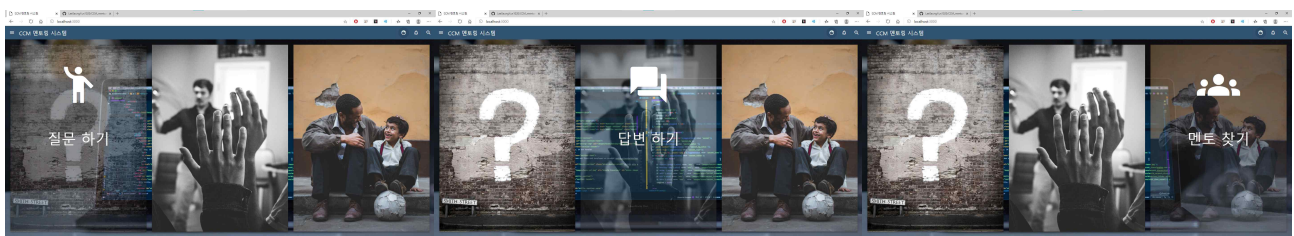


그림 24 홈 화면 – 마우스 호버링

4) 로그인

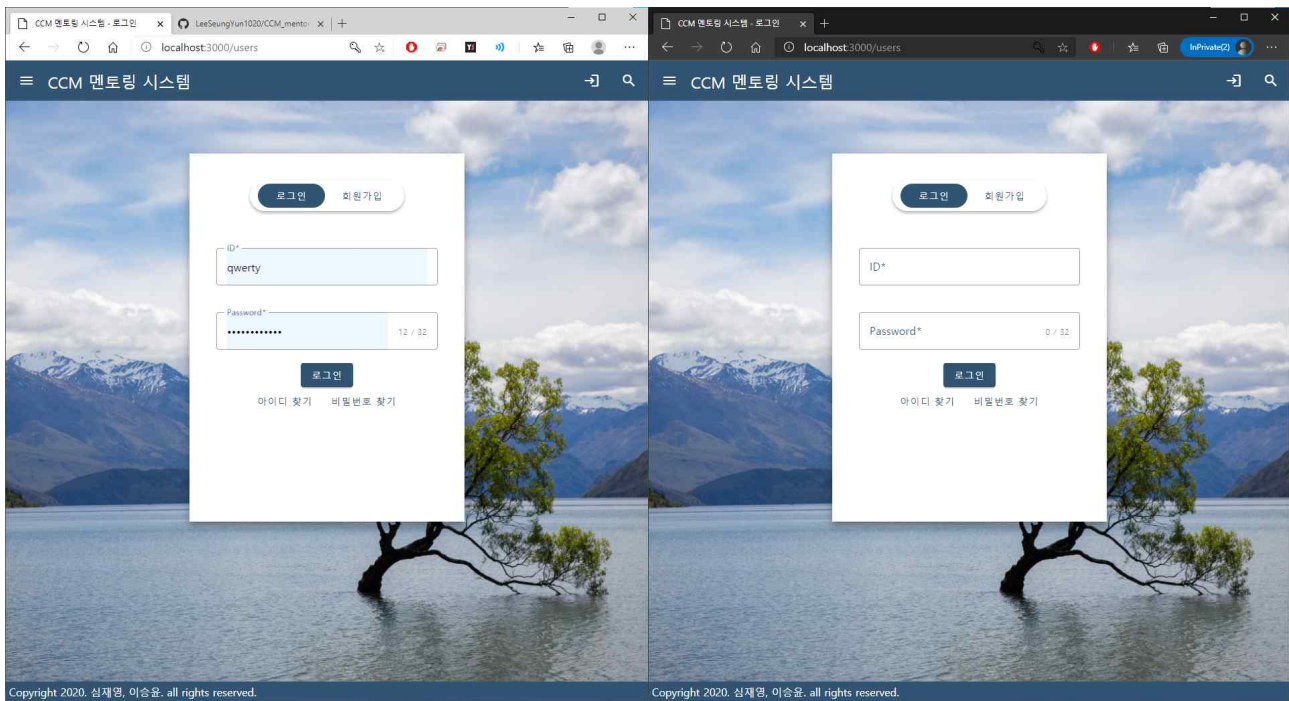


그림 25 로그인 - 좌측은 자동 완성이 적용된 경우 우측은 적용되지 않은 경우

회원 가입한 경우 아이디와 비밀번호를 입력하면 로그인할 수 있다.

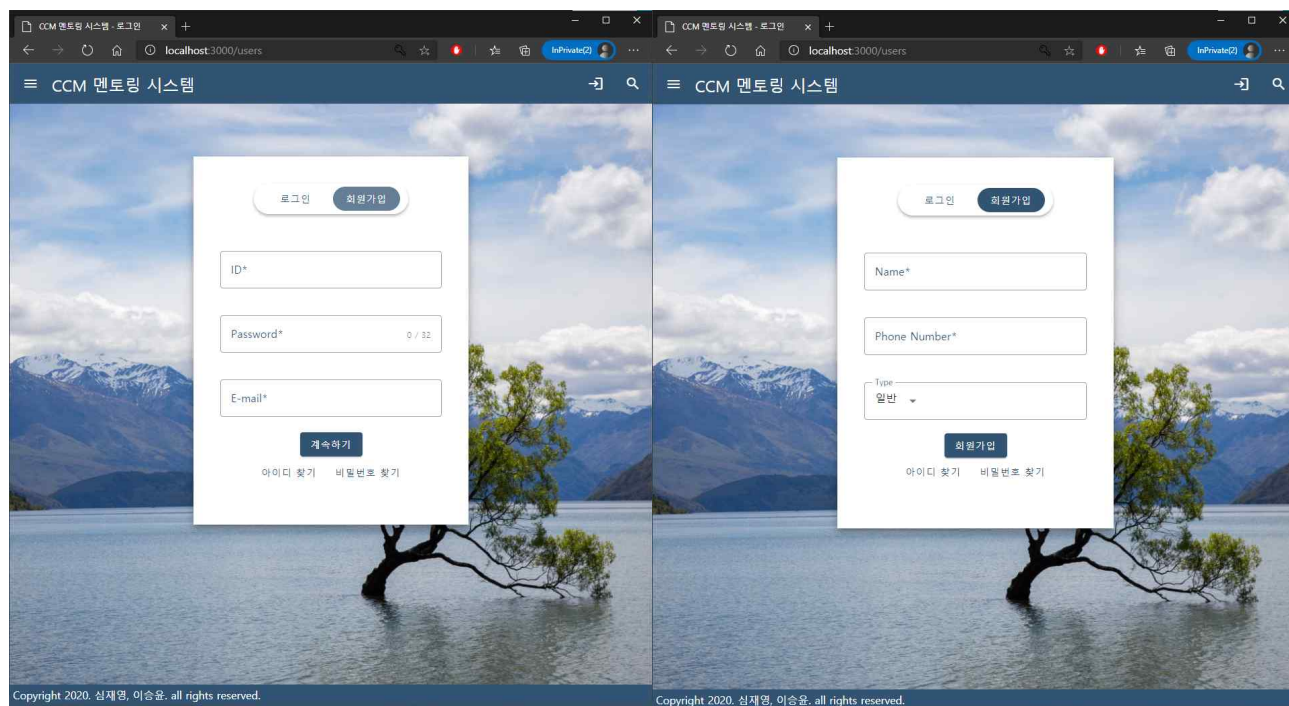


그림 26 회원 가입

회원 가입에 필요한 정보를 입력하면 회원 가입된다. 각 필드를 사용자가 입력하였는지 확인하고 ID 중복 체크 후 상세 정보 입력 필드가 개방되고 모든 필드에 대해 입력을 마치면 회원 가입이 완료된다.

5) 검색

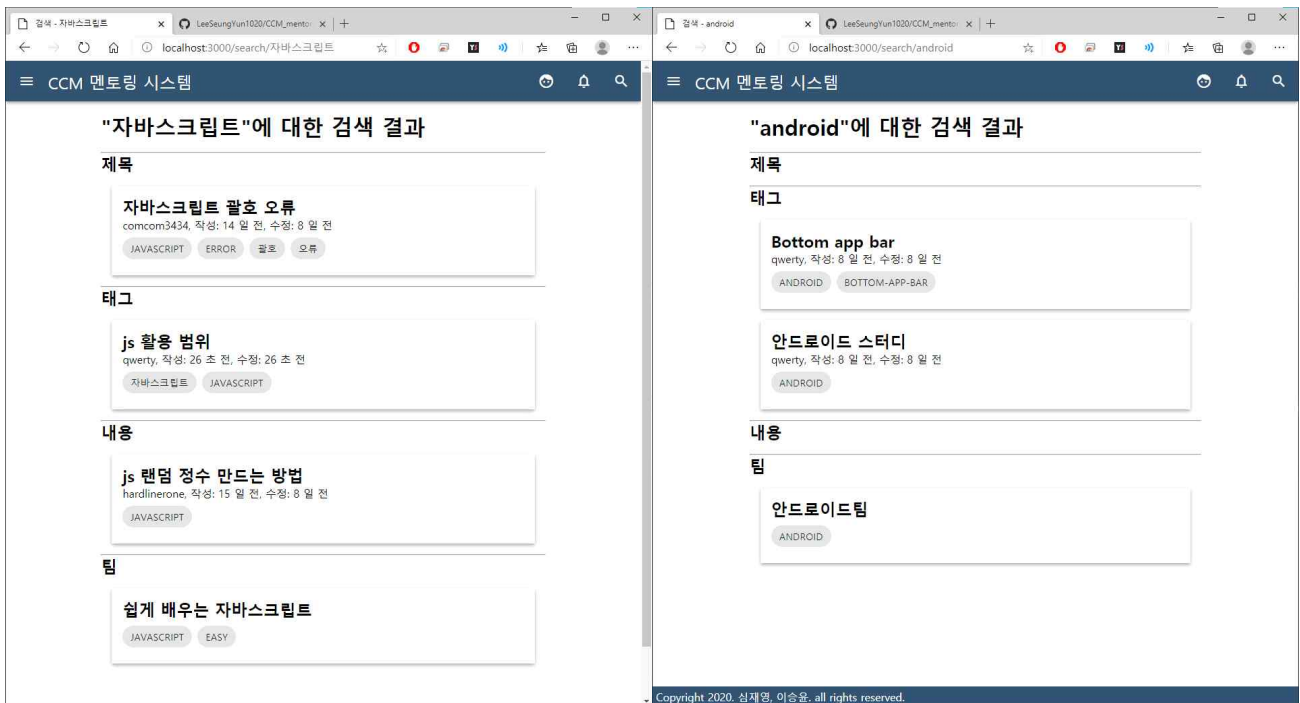


그림 27 검색

검색 창을 이용하거나 게시물 또는 사용자 정보 등에서 태그를 클릭하여 검색 결과 화면으로 이동할 수 있다. 각 게시물이 제목, 태그, 내용에 따라 표시되며 멘토링 팀도 이름, 태그가 포함된 경우 표시된다. 쉽고 빠르게 사이트 내의 항목들을 찾아갈 수 있다.

6) 질문 하기 - 게시글 작성, 수정

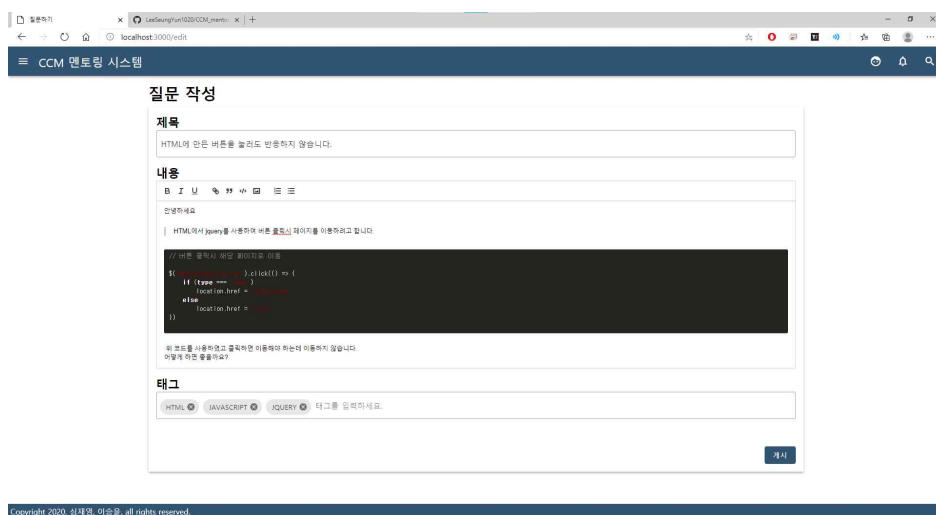


그림 28 게시글 작성

주요 기능 중 하나인 질문하기는 제목, 내용, 태그를 지원하며 특히 내용에는 코드를 삽입하거나 그림을 올리는 등 코드 멘토링에 필요한 핵심적인 내용을 삽입할 수 있고 인용, 링크, 차례 등 다양한 텍스트 에디터 기능을 지원한다.

7) 질문 하기 – 게시글 상세

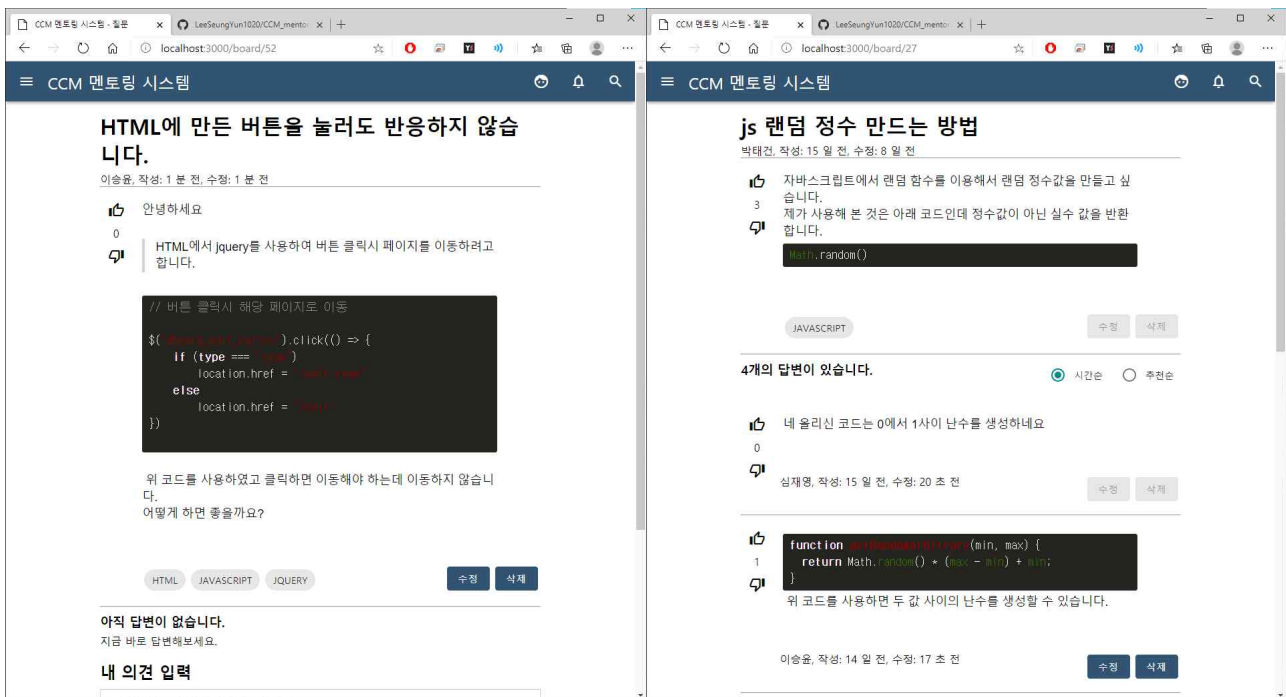


그림 29 게시글 상세

질문을 작성하면 작성된 게시물을 바로 확인할 수 있다. 아직까지 답변이 달리지 않은 질문은 수정, 삭제가 가능하고 답변이 달린 게시물은 수정 또는 삭제할 수 없다. 또한, 자신이 작성한 게시물만 수정 가능하며 자신이 작성한 게시물이 아닌 경우 수정, 삭제 버튼이 비활성화된다. 질문자가 추가한 태그를 클릭하면 검색 화면으로 이동한다.

8) 게시글 추천/비추천



그림 30 게시글 추천/비추천

작성한 질문과 답변은 추천 또는 비추천할 수 있다. 추천 또는 비추천 버튼을 한번 누르면 추천 또는 비추천되어 각 질문 및 답변에 부여된 점수가 상승 또는 하락한다. 누른 추천/비추천 버튼을 다시 누를 경우 추천/비추천이 취소된다. 부여된 점수는 각 사용자별 질문 점수 및 답변 점수 계산에 반영되어 멘토 찾기 및 사용자 정보 조회 시 표시된다. 답변의 경우 추천순으로 정렬하면 추천을 많이 받은 답변이 우선적으로 표시된다.

9) 답변 하기 - 답변 정렬, 게시, 수정, 삭제

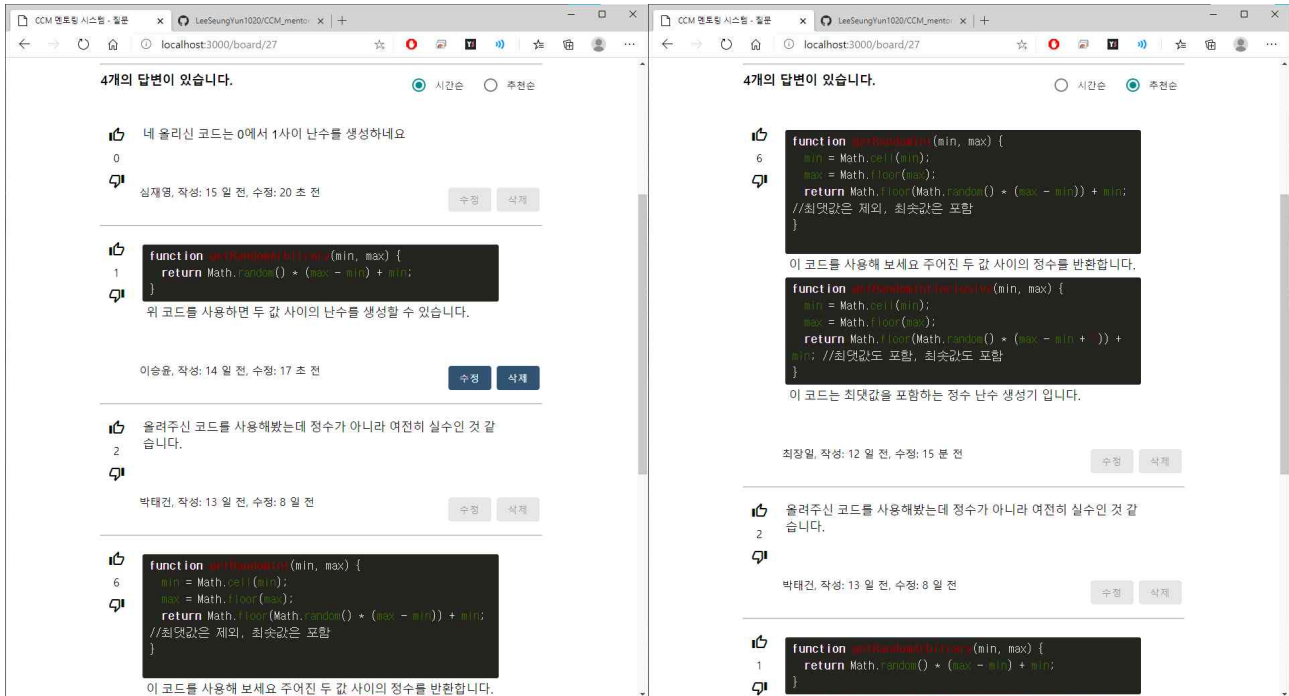


그림 31 답변 상세 - 시간순 정렬과 추천순 정렬

답변은 작성 순서대로 또는 추천이 높은 순서대로 표시하도록 할 수 있다.

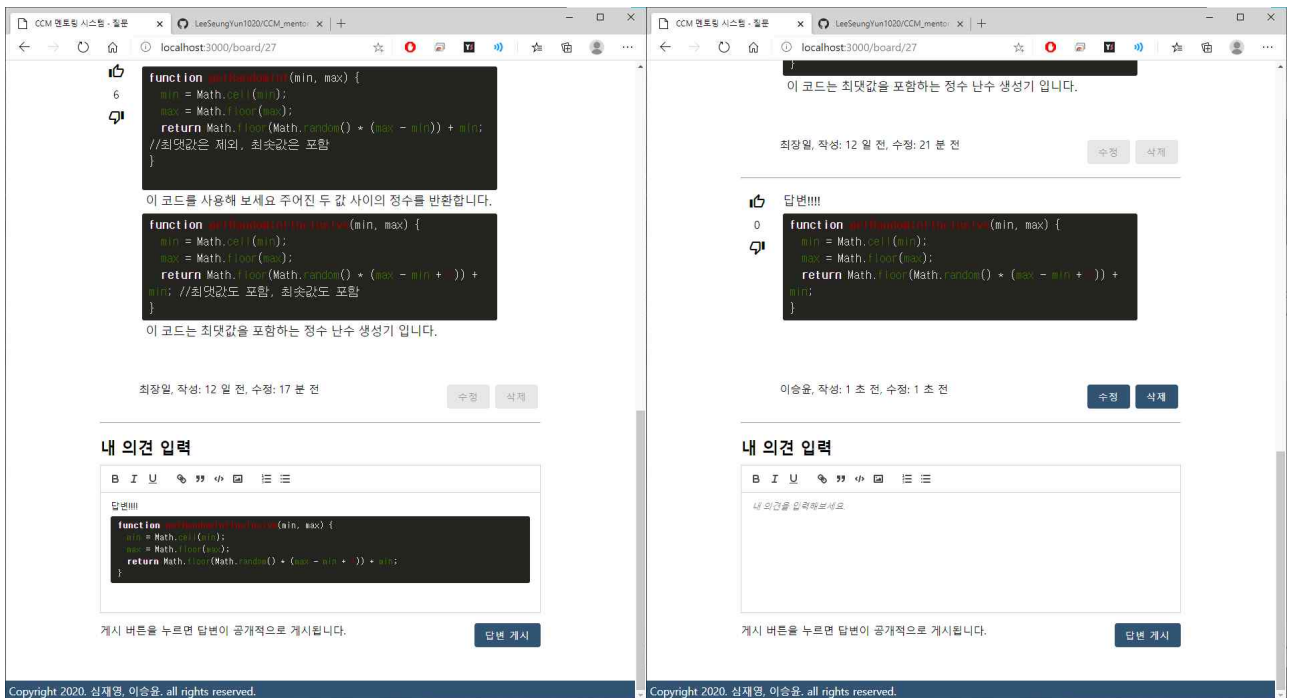


그림 32 답변 게시

질문 작성에서와 동일한 텍스트 에디터를 지원한다. 의견을 입력하고 답변 게시 버튼을 누르면 답변이 게시된 것을 확인할 수 있다.

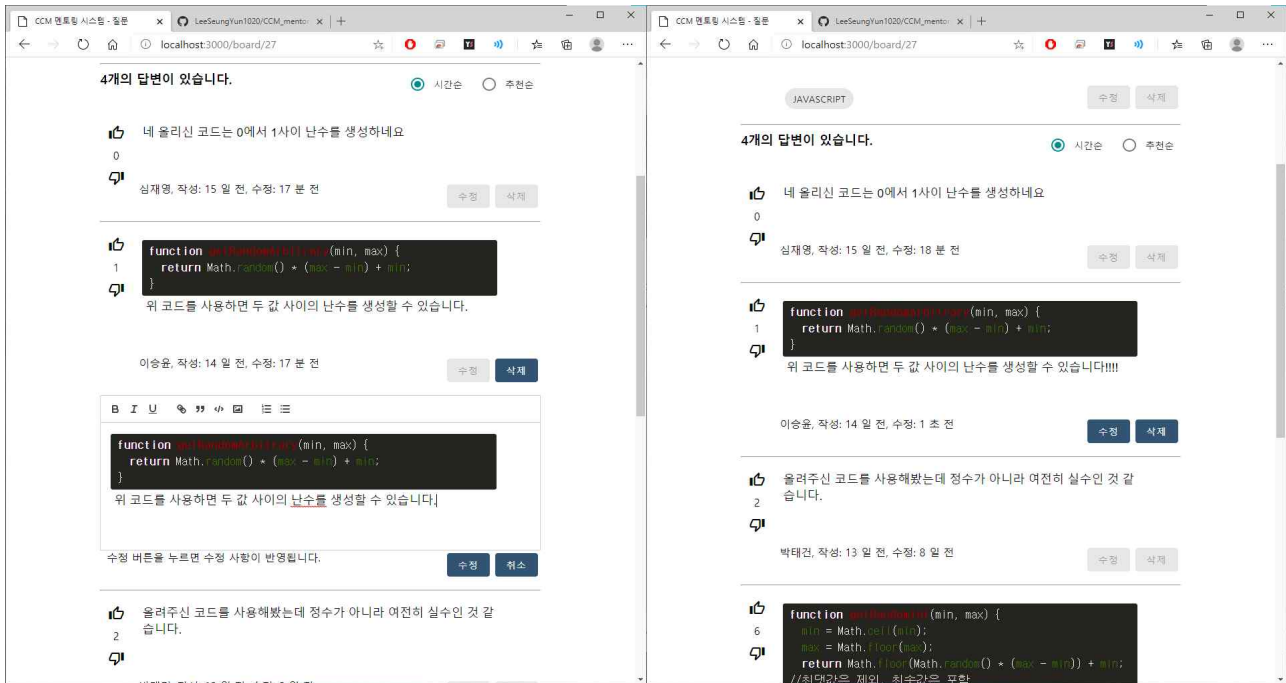


그림 33 답변 수정

자신이 작성한 답변은 수정할 수 있다. 수정 버튼을 누르면 아래에 작성한 답변이 표시되고 즉시 내용을 수정할 수 있다. 수정 버튼을 누르면 변경 사항이 최종적으로 적용된다. 취소 버튼을 누르면 변경이 취소된다. 삭제 버튼을 누르면 작성한 답변이 삭제된다. 수정 및 삭제 버튼은 자신이 작성한 답변에만 활성화된다.

10) 답변 하기 – 질문 목록

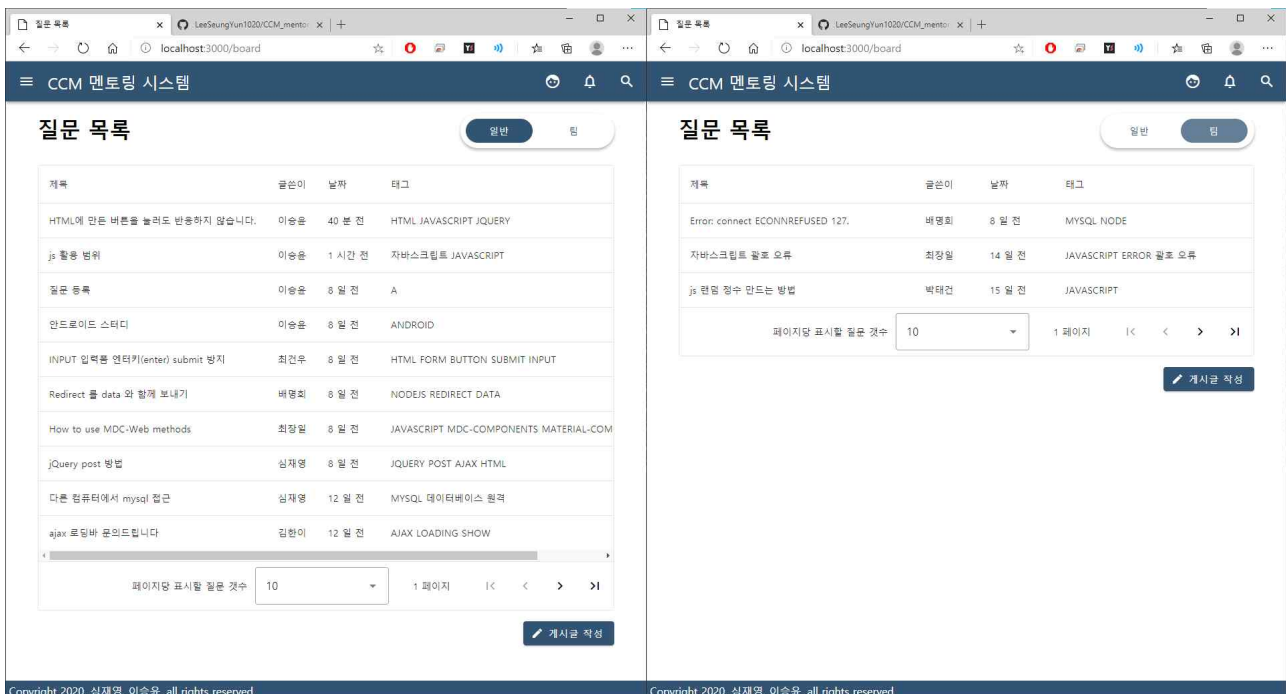


그림 34 질문 목록 – 일반 게시물 및 팀 게시판 게시물

답변 하기를 누르면 질문 목록으로 이동하며 우측 상단의 버튼을 통해 전체 공개 게시글과 팀 공개 게시글을 조회할 수 있다. 팀 공개 게시글은 가입한 팀의 게시물만 공개되며 게시판 아래 게시글 작성을 통해 작성할 수 있다. 각 질문을 클릭하면 앞에서 살펴본 질문 상세 보기 화면으로 이동한다.

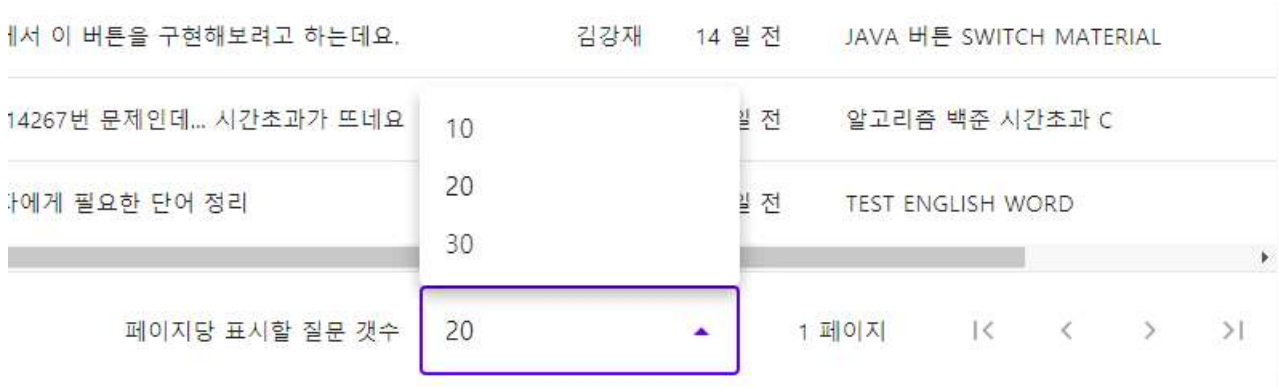


그림 35 질문 목록 – 페이지 (표시할 질문 수)



그림 36 질문 목록 – 페이지 (첫 페이지와 마지막 페이지)

아래 페이지당 표시할 질문 갯수를 눌러 표시할 질문의 갯수를 변경할 수 있다. 오른쪽에는 현재 페이지가 표시되며 현재 페이지 오른쪽 각 버튼을 통해 각각 첫 페이지, 이전 페이지, 다음 페이지, 마지막 페이지로 이동할 수 있다.

11) 멘토 찾기

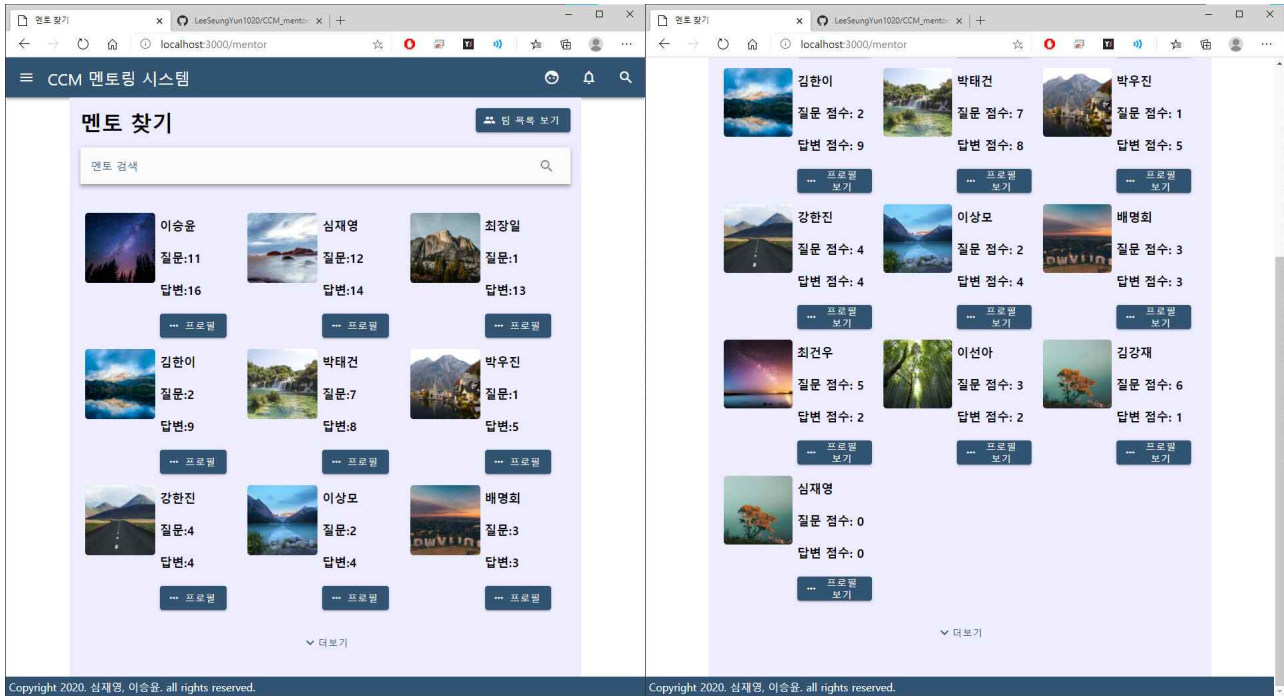


그림 37 멘토 찾기

멘토들이 답변 점수 순서로 표시된다. 9명씩 출력되며 더보기 버튼을 눌러 계속해서 답변 점수순으로 조회할 수 있다. 팀 목록 보기를 눌러 팀 전체 목록을 확인할 수 있다.

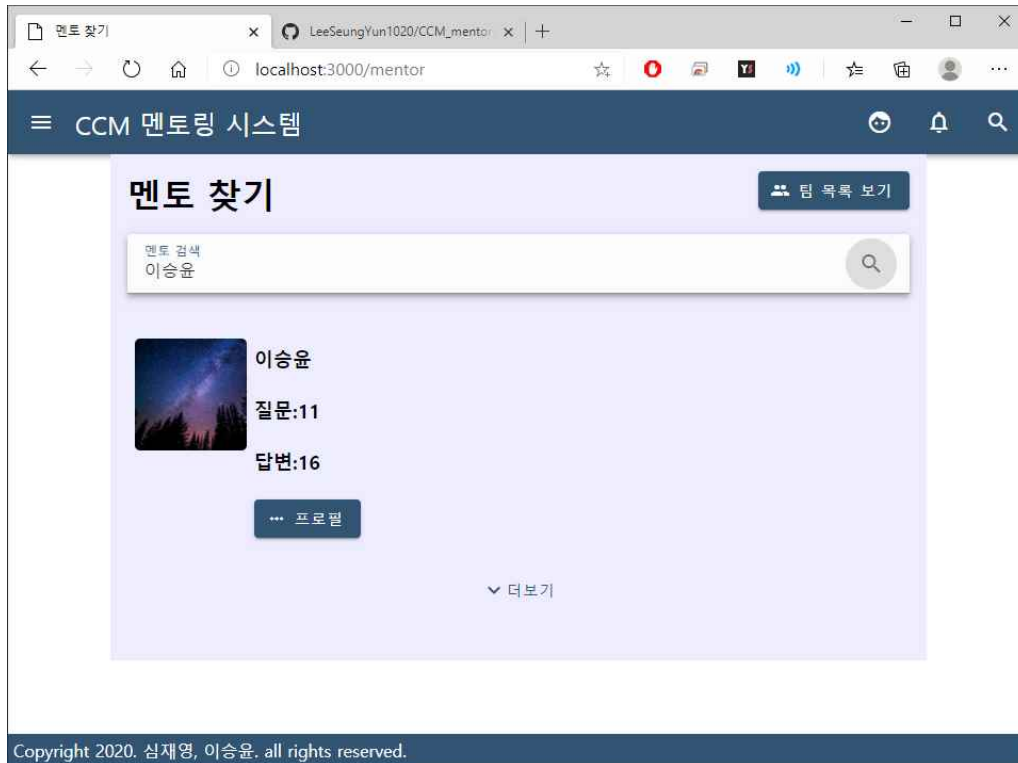


그림 38 멘토 찾기 - 검색(필터링)

검색 창에 멘토 이름을 입력하여 필터링할 수 있다.

12) 팀 목록

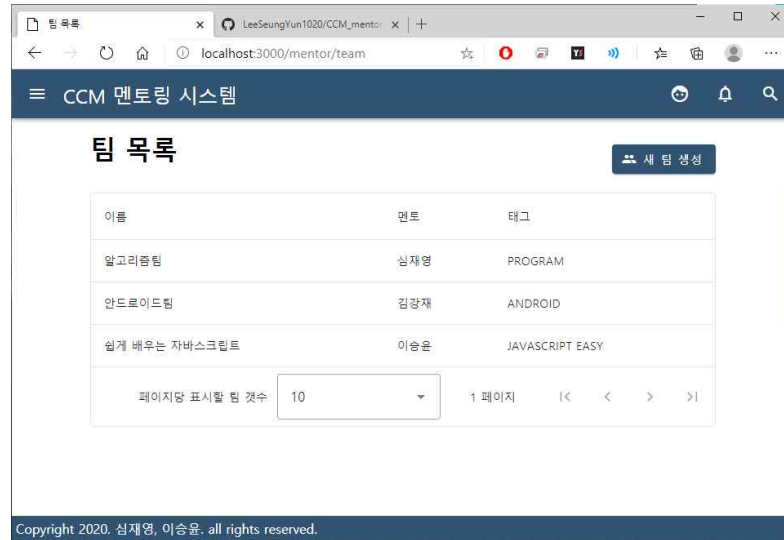


그림 39 팀 목록

팀 목록이 표시되고 원하는 팀을 눌러 가입할 수 있다. 마음에 드는 팀이 없다면 새 팀을 생성할 수 있다. 이미 가입한 팀이 있다면 현재 팀을 탈퇴할 것인지 사용자에게 확인한다.

13) 프로필 조회



그림 40 프로필 조회

로그인한 사용자는 로그인 팝업에서 프로필 정보 버튼을 눌러 프로필 정보를 조회할 수 있다. 멘토 찾기에서 프로필 보기 버튼을 눌러 다른 사용자의 프로필 정보도 조회할 수 있다. 프로필 정보에는 개인 정보와 팀 정보, 질문 및 답변 점수와 최근 질문 또는 답변한 게시물이 표시된다. 팀 정보의 가입 버튼을 눌러 해당 팀에 가입할 수 있고 각 질문 및 답변을 눌러 해당 게시물로 이동할 수 있다.

5. 개발 일정 및 역할

가. 주요 개발 일정

이름	기간	상태
주요 제공 서비스 확정	2020/10/14 – 10/21	완료
레이아웃 설계	2020/10/14 – 10/29	완료
개발 환경 구축	2020/10/21 – 10/23	완료
사용자 데이터베이스 설계	2020/10/23 – 10/30	완료
서비스 데이터베이스 설계	2020/10/23 – 10/30	완료
레이아웃 제작	2020/10/23 – 11/17	완료
데이터베이스 제작	2020/10/31 – 11/04	완료
서비스 제작	2020/10/29 – 11/17	완료
1차 시연	2020/11/06	완료
최종 시연	2020/11/13	완료
결과 보고서 작성	2020/11/17	완료

나. 세부 서비스 세부 개발 일정

이름	기간	상태
헤더	2020/10/23 – 2020/11/13	완료
네비게이션	2020/10/23 – 2020/11/05	완료
푸터	2020/10/23 – 2020/11/04	완료
알림	2020/11/10 – 2020/11/10	완료
검색	2020/11/13 – 2020/11/15	완료
홈	2020/10/23 – 2020/11/13	완료
회원가입	2020/10/23 – 2020/11/06	완료
로그인	2020/10/23 – 2020/11/05	완료
로그인 팝업	2020/11/05 – 2020/11/14	완료
사용자 프로필 조회	2020/11/15 – 2020/11/17	완료
질문 목록	2020/11/10 – 2020/11/11	완료
질문 작성	2020/11/04 – 2020/11/14	완료
질문 수정	2020/11/02 – 2020/11/13	완료
질문 삭제	2020/11/02 – 2020/11/13	완료
질문 상세	2020/11/02 – 2020/11/11	완료
답변 작성	2020/11/02 – 2020/11/11	완료
답변 수정	2020/11/02 – 2020/11/12	완료
답변 삭제	2020/11/02 – 2020/11/12	완료
질문 및 답변 - 팀	2020/11/10 – 2020/11/17	완료
멘토 목록	2020/11/13 – 2020/11/17	완료
팀 목록	2020/11/15 – 2020/11/17	완료
팀 생성	2020/11/16 – 2020/11/17	완료
팀 가입, 탈퇴	2020/11/15 – 2020/11/17	완료
팀 삭제	2020/11/15 – 2020/11/17	완료

다. 구성원별 역할

1) 이승윤

가) 레이아웃

헤더, 네비게이션, 푸터, 검색, 질문 목록, 질문 상세, 답변, 팀

나) 기능

헤더, 네비게이션, 푸터, 검색, 회원 가입 로그인, 질문 목록, 질문 상세, 답변,
팀

다) 데이터베이스

라) 보고서

개발 배경, 상세 설계, 개발 결과, 개발 일정, 참고 문헌

2) 심재영

가) 레이아웃

회원 가입, 로그인, 홈, 로그인 팝업, 프로필 조회, 질문 작성, 멘토 목록, 사진

나) 기능

홈, 로그인 팝업, 프로필 조회, 질문 작성, 멘토 목록

다) 보고서

개발 배경, 개발 개요, 상세 설계, 참고 문헌

6. 참고문헌

- [1] "Stack Overflow Annual Developer Survey," Stack overflow. 2020.
- [2] "Stack Overflow의 글로벌 SW개발자 현황 조사 결과 분석," 소프트웨어정책연구소. 2019.