

# 최종 발표

졸업 프로젝트 주제: SW 자율학습 강의 Docker+웹 서비스 개발

종합설계1\_02분반

야간자율학습

고태완  
김지혜  
최수연

# 목차

---

01

디자인  
스프린트

02

멘토링

03

문서

04

설문조사

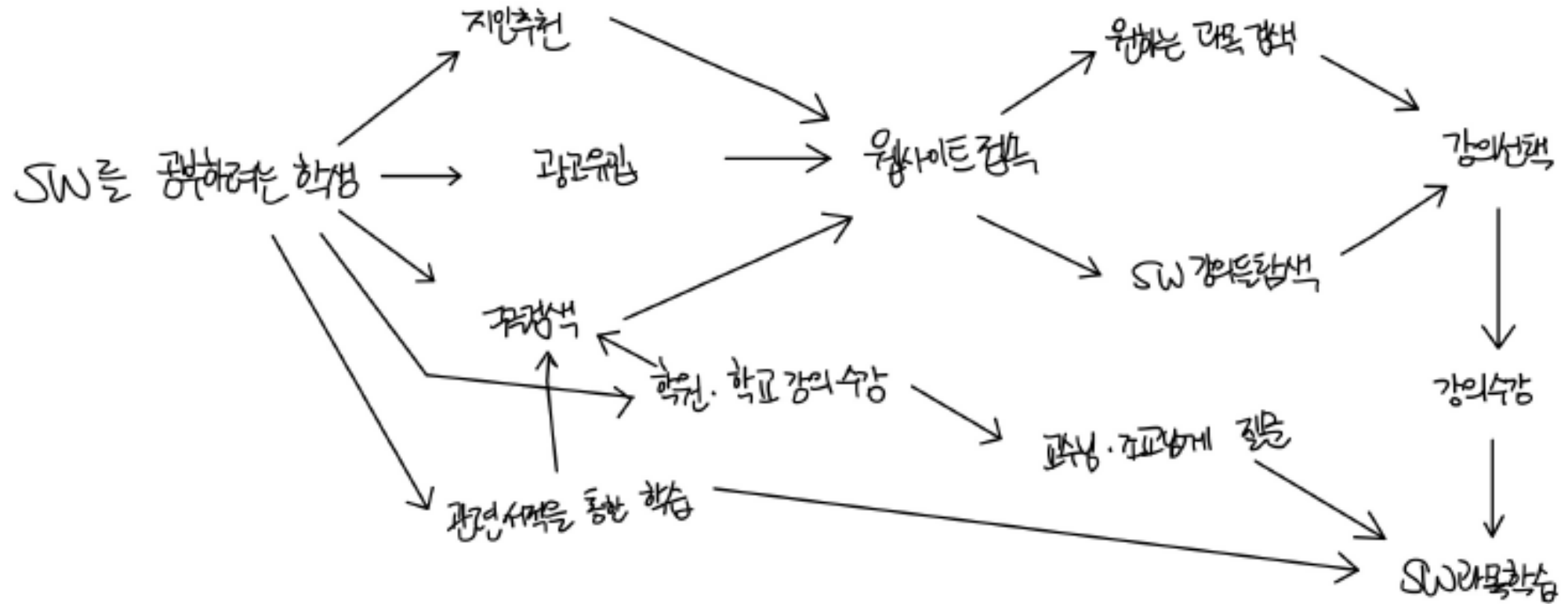
05

프로토타입  
데모

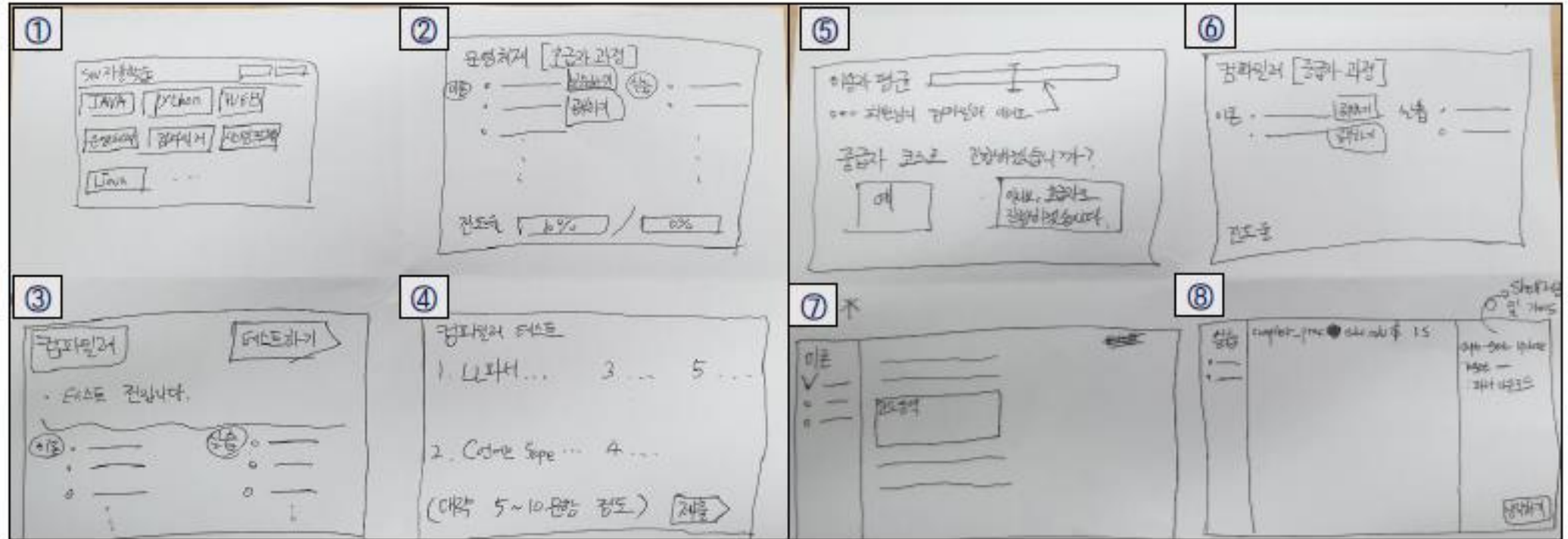
01

디자인 스프린트

# 1일차 - Map

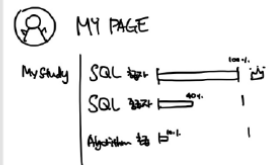
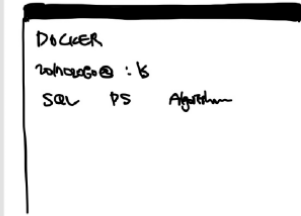
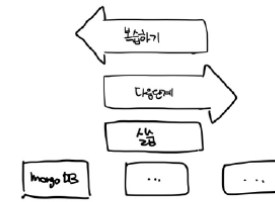
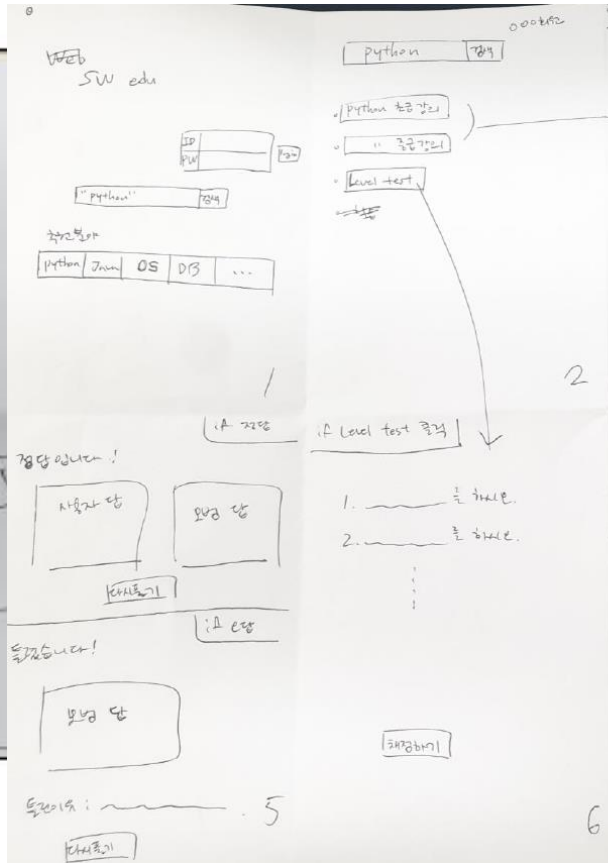


# 2일차 - Sketch





## 야간자율학습



# 3일차 – Sketch

## Solution 도출 – 팀에서 자체적 투표

순서대로 투표했는데, 다른 사람이 어디에 그 의견에 동조할 수도 있기 때문에 각자

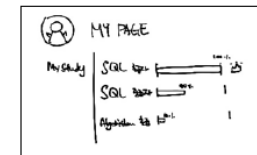
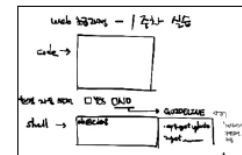
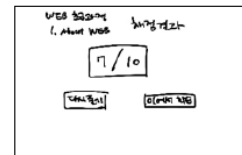
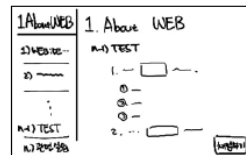
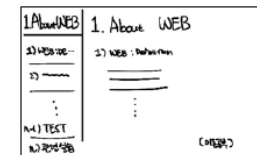
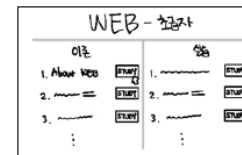
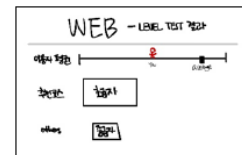
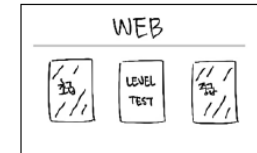
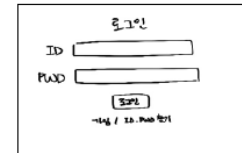
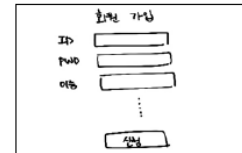
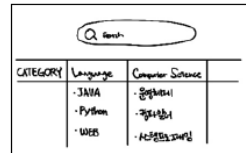
|   | A | B |
|---|---|---|
| 1 | 1 | 1 |
| 2 |   |   |
| 3 | 1 |   |
| 4 |   | 1 |
| 5 | 1 | 2 |
| 6 | 1 |   |
| 7 |   |   |
| 8 | 3 | 3 |

투표 결과

## 완성된 스토리보드

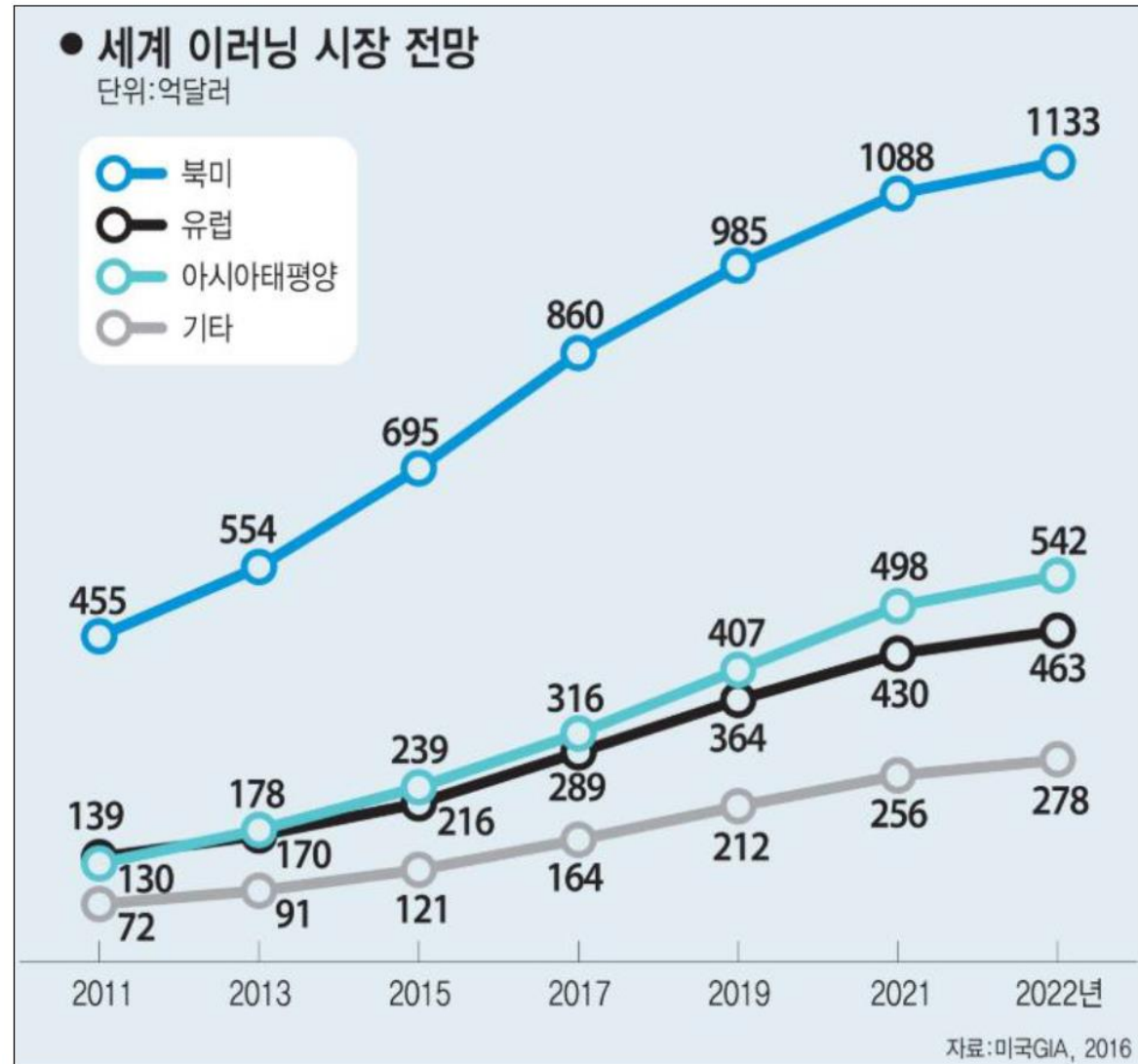
12슬라이드. 3표, 2표를 받았던 Sketch 반영

그 외에 필요하다고 생각하는 부분들(로그인, 회원가입 등)을 협의를 통해 추가





# Test – 시장조사



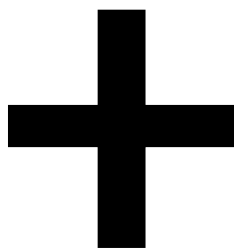
# Test – 시장조사

유데미는 MOOC 중에서도 개방형 플랫폼 모델을 도입한 유일한 성공 사례로 주목할 만하다. 2010년 창업한 이래 최근까지 8개의 벤처캐피털로부터 1억7300만 달러를 투자받으며 성장세를 이어가고 있다. 미국 샌프란시스코에 본사를 설립하고 2014년 아일랜드 더블린, 2017년 말 브라질 상파울루에 지사를 설립했다. 최근에는 기업 단위의 맞춤형 교육 프로그램을 제공하면서 B2C에서 B2B로 수익 모델을 확장하고 있다.

| 서비스\기능     | 실습 환경 제공 | 사용자 업로드 가능 | level test | 복습 기능 |
|------------|----------|------------|------------|-------|
| inlearn    | x        | o          | x          | x     |
| codecademy | o        | x          | x          | o     |
| coursera   | x        | o          | x          | o     |
| udemy      | x        | o          | x          | o     |
| 생활코딩       | x        | x          | x          | x     |

# Test – Feedback

- 콘텐츠의 지속적 업데이트
- 자료의 신뢰성 여부
- 교과목에서 막히는 부분 (Q&A, 튜터링을 통한 도움)
- 만들면 실제로 쓸 것 같은 커리큘럼



[미래설계상담]

## 설문 결과

| 질문  | 예  | 예  | 예   |
|---|--|--|---|
| 1-1. 기존 sw 학습 사이트 (예: 인프런, codecademy)를 이용하는 장단기는 무엇인가?                   |  |  |   |
| 1-2. 위와 질문(1-1)에 '예'라고 대답한 경우, 아래에 학습 사이트의 특징과 어떤 점이 장점을 꼽을 수 있겠나고 생각하는가? | 학이 단편적인 지식으로 그들끼리 sw 프로그램을 사먹을 특별한 지식을 느끼지 못하였다. | 장강강의 수강 전과후 레퍼토리들 먼저 지루해 해볼 수 있는 점과 좋은강의이다. 이론 지식은 1년이상도 두 달이여 관련정보를 가져와야 받았다지만, 인터넷에서 좋은 강의는 정말 특색이 없는것이 있는것고, 인터넷은 정말 수월이해서 불만정보를 알고 싶으면 불만일 것 같다. | 장강 level 테스트를 통한 자기 의 수준에 맞는 강의가 좋은것 같다. 다른 사이트에서는 모르는게 많아 지게 될때를 매우 위해서 조금은 더 잘 알고있는데 좋겠다.<br>level-테스트 기능 |
| 2-1. 영상에서 소개한 프로젝트에 나오는 기능 32에 추가가 있으면 더 좋을것 같은 기능은 무엇인가?                 | 예  | 예  | 예   |
| 2-2. 위와 질문(2-1)에 '예'라고 대답한 경우, 추가가 있으면 좋을 것 같은 기능은 무엇인가?                  | QnA 기능   |  |   |
| 3. 프로젝트에 나오는 기능에 어떤 개선안이 있다고 생각하는가?                                       |  |  |   |
| 4. 현재 제3자에게 배포하지 않고도 돌려받을 수 있겠다.  |  |  |   |
| 5. 프로젝트에 나오는 기능에 어떤 개선안이 있다고 생각하는가?                                       | 형인   | 만약 다른사람 강의를 듣고싶은경우에 자기 학기 기능이 있으면 좋겠다.   | 레퍼토리 테스트를 할 필요는 없지만, 어떤 기능으로 나누어지기를 하고 선택하여 할 수 있으면 좋겠다.<br>예를 들면, 강의를 가르기 위해 메뉴를 만들어 내겠다.                  |

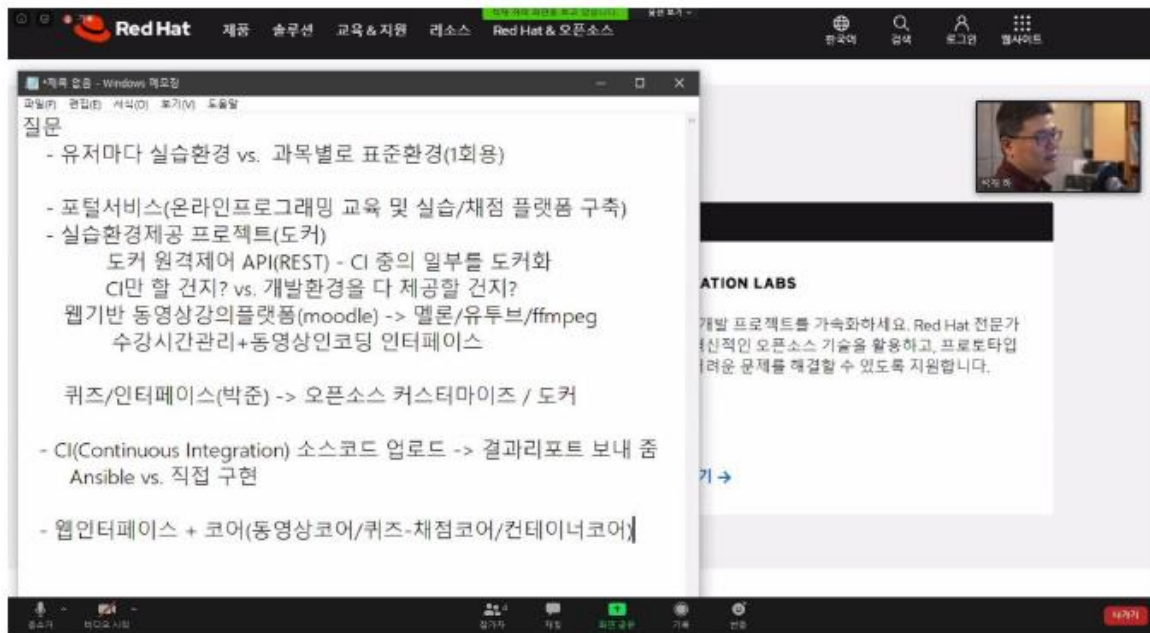
[설문조사]

“ *Udemy, Coursera*처럼  
개방형 SW 강의 플랫폼을 만들자

02

멘토링

# 멘토링



05/22 하석재 멘토님과..

- 유저마다 실습환경 vs. 과목별로 표준환경(1회용)
  - 포털서비스(온라인프로그래밍 교육 및 실습/채점 플랫폼 구축)
  - 실습환경제공 프로젝트(도커)
    - 도커 원격제어 REST API - CI 중의 일부를 도커화
    - CI만 할 건지? vs. 개발환경을 다 제공할 건지?
  - 웹 기반 동영상강의플랫폼(moodle) -> 멜론/유튜브/ffmpeg
  - 수강시간관리+동영상인코딩 인터페이스
  - 퀴즈/인터페이스(박준) -> 오픈소스 커스터마이즈 / 도커
  - CI(Continuous Integration) 소스코드 업로드 -> 결과리포트 보내 줌
  - Ansible vs. Docker REST API
  - 웹인터페이스 + 코어(동영상코어/퀴즈-채점코어/컨테이너코어)
    - 인터페이스만 정의한 상태로 보이나
    - 코어를 어떻게 구축할 건지(오픈소스기반 vs. 자체구축)

객체지향기술을 사용한다면 UML(필수), cf. Agile을 하면 TDD(Test case를 작성)  
쿠버네티스 - 도커로 여러 대를 사용한 서비스를 운영(Orchestration)할 때 사용하는 기술

1. VMWare Workstation(standalone)
2. VMWare ESXi + vCenter(여러 대 관리-> orchestration) -> vSphere

Docker + Kubernetes(orchestration)  
vSphere(기존) + Kubernetes(도입) - 같이 어떻게 사용할 수 있나?  
orchestration(DB이중화/WAS)

회의록

# 멘토링 그 이후

- 현재 메인 시스템이 3개로 규모가 너무 큼 (웹기반 동영상강의플랫폼, 퀴즈-채점, 도커를 이용한 실습 환경)
- 도커를 이용한 실습 환경에 초점을 맞추기로 결정(선택과 집중)
- 이미 시장에서 잘 사용되고 있는 이론 강의에 대한 서비스 및 퀴즈는 최종 프로젝트에서 삭제
- 실습 환경을 이미지로 저장하여 수강생들 개개인에게 하나의 컨테이너로 제공
  - => OS에 대한 접근이 필요한 실습에서 수강생은 root 권한으로 자유롭게 실습 진행 가능, 단 1회용 컨테이너, 실습이 끝난 후에 삭제됨
  - => Shell과 Code Editor를 웹페이지에서 제공.
    - 단 Shell을 사용하지 않아도 되는 경우(OS에 접근하지 않아도 되는 경우) Shell을 제공하지 않음
    - Code Editor를 사용할 경우에 채점을 하지 않고 컴파일 된 결과만 보여줌.
  - => 기본적으로 사용자가 업로드한 실습 강의에 대해서 저장된 실습의 난이도를 이용해 사용자에게 수준별 학습을 제공할 수 있도록 Curation한다.
  - => 특정 강사가 특정 사용자들에 대해서만 실습을 가능하게 할 것을 원하는 경우, 이 실습은 private 상태가 되며 특정 강사가 저장한 환경으로 사용 권한을 가진 수강생 들에 한해서 해당 환경으로 바로 실습을 진행 할 수 있도록 한다.
- Docker와 Kubernetes를 이용한 Orchestration 방식의 유연한 서버 운영
- CI(Continuous Integration) 기반의 소스 코드 에디터 사용. 소스코드를 올리면 빌드 후 에러 포함한 결과 리포트를 사용자에게 보여줌.

위와 같은 이유로 웹기반 동영상강의플랫폼, 퀴즈-채점은 제거  
도커를 이용한 실습 환경 중점적으로 구현 => SE문서에 좀 더 상세하게 기록



문서 재작성 및 프로토타입 재제작

03

문서

# Little Changed

## 문제정의서

### 1. 연구의 필요성

현재 세계 이력서 시장이 성장세를 이루면서 SW 자율학습 플랫폼 또한 증가하고 있다. 흔히 자주 쓰이는 SW 자율학습 플랫폼으로 udemy, coursera와 같은 MOOC와, codecademy, inflearn, 생탈로딩 등이 있다. 이러한 SW 자율학습 플랫폼들로 인해 많은 사람들이 인터넷으로도 쉽고 간편하게 SW 학습을 할 수 있게 되었다. 하지만 이를 포함한 많은 SW 자율학습 플랫폼들에서 몇 가지 한계점이 드러난다.

먼저, SW 학습의 상당 부분이 실습 위주인 것에도 불구하고 실습 환경을 제공하는 서비스가 드물다는 것이다. 그래서 많은 사용자들이 SW 실습 강의를 따라가기도 전에 환경 설정에서 막히는 경험을 한다. 학습 강사가 환경 설정에 대한 강의를 해 주는 경우도 있으나, 사용자들마다 자신의 개발 환경 상태가 다를 것이므로 환경 설정 강요는 한계가 있다. 또한 사용자가 자신의 수준을 인지하기 어려워 수준에 맞지 않는 학습을 할 수 있다. 강의의 수준과 자신의 수준을 정확히 파악하지 못하고 강의를 학습하는 것에 어려움을 겪는 경우도 있다. 이렇게 사용자의 자율학습을 돕는 것에도 한계가 드러난다. 자율학습을 돕는 것에서 뿐만 아니라, 서비스 자체적으로도 한계점이 있다. 사용자가 강의를 직접 업로드 함으로써 서비스 사용자들이 자체적으로 서비스를 업데이트하는 기능 또한 모든 SW 자율학습 플랫폼들이 갖고 있지 않다. 이렇게 되면 서비스 운영자가 직접 서비스를 업데이트해야 하는 데, 그것 만으로는 지속적으로 업데이트를 하는 데 한계가 있다.

4차산업혁명이 현실화가 되어가면서 세계 각국에서 SW 역량을 갖춘 인재 양성에 힘쓰고 있다. 국내에서도 코딩 교육의 중요성을 인지하여 코딩 교육을 의무화하고 있다. 많은 재능 있는 SW 인재를 육성하는 것이 사회의 발전에 기여하는 것 뿐만 아니라, 더 나아가 국가 경쟁력 또한 확보 할 수 있다. 우리는 이러한 현재 SW 시장 흐름에 맞춰 기존에 SW 자율학습 플랫폼들의 한계점을 개선하여 SW 자율학습을 돕고 경쟁력 있는 SW 인재를 육성하는 SW 자율학습 플랫폼 서비스를 개발할 것이다.

## 요구사항명세서

### 1.1. Purpose

SW 자율학습을 돕는 서비스를 개발한다. 여기서 stakeholder는 SW 교육 서비스를 제공하는 주체와 SW 학습을 하고자 하는 고객들을 말한다.

### 1.2. Scope

이 SW 자율학습 플랫폼은 사용자에게 이론 및 실습에 대한 학습을 제공하는 기능, 강의 큐레이션 기능과 사용자의 수준을 측정하는 레벨테스트 기능, 실습 환경 제공 기능, 학습 한 내용에 대한 테스트 기능, 사용자의 학습 현황을 볼 수 있는 기능, 강의 업로드 기능을 갖추으로써 사용자의 SW 자율학습을 돕는 웹 서비스이다. 이 서비스는 사용자 맞춤 양질의 콘텐츠를 제공할 수 있으며, 자동 실습 환경 제공과 강의 큐레이션을 함으로써 편리함 또한 갖출 수 있다. 4차산업혁명에 따른 코딩 교육 시장 또한 상승세를 이루고 있는 요즘 많은 사람들의 SW 자율학습을 도우면서 콘텐츠의 지속적인 업데이트와 경쟁력을 갖춘 SW 자율학습 플랫폼을 개발하는 것이 목표이다.

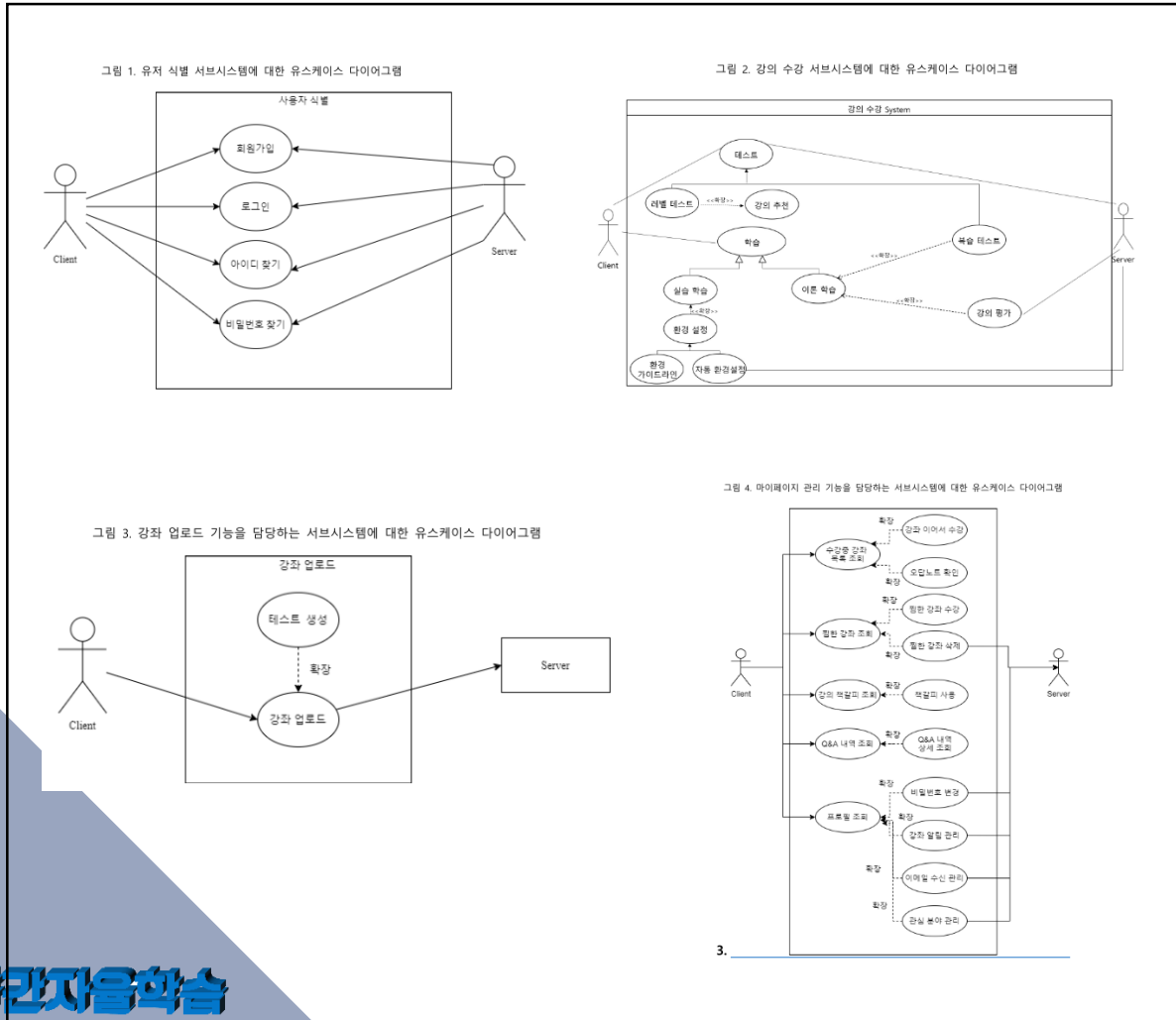
### 1.3. Definitions, acronyms, and abbreviations

N/A: Not Applicable  
ES6: Ecma script 6 (Javascript version)  
HTTP: HyperText Transfer Protocol  
JWT: Json Web Token  
React: Javascript 라이브러리  
Linux: 운영체제 중 하나  
MySQL: 데이터베이스 중 하나  
django: Python 백엔드 라이브러리

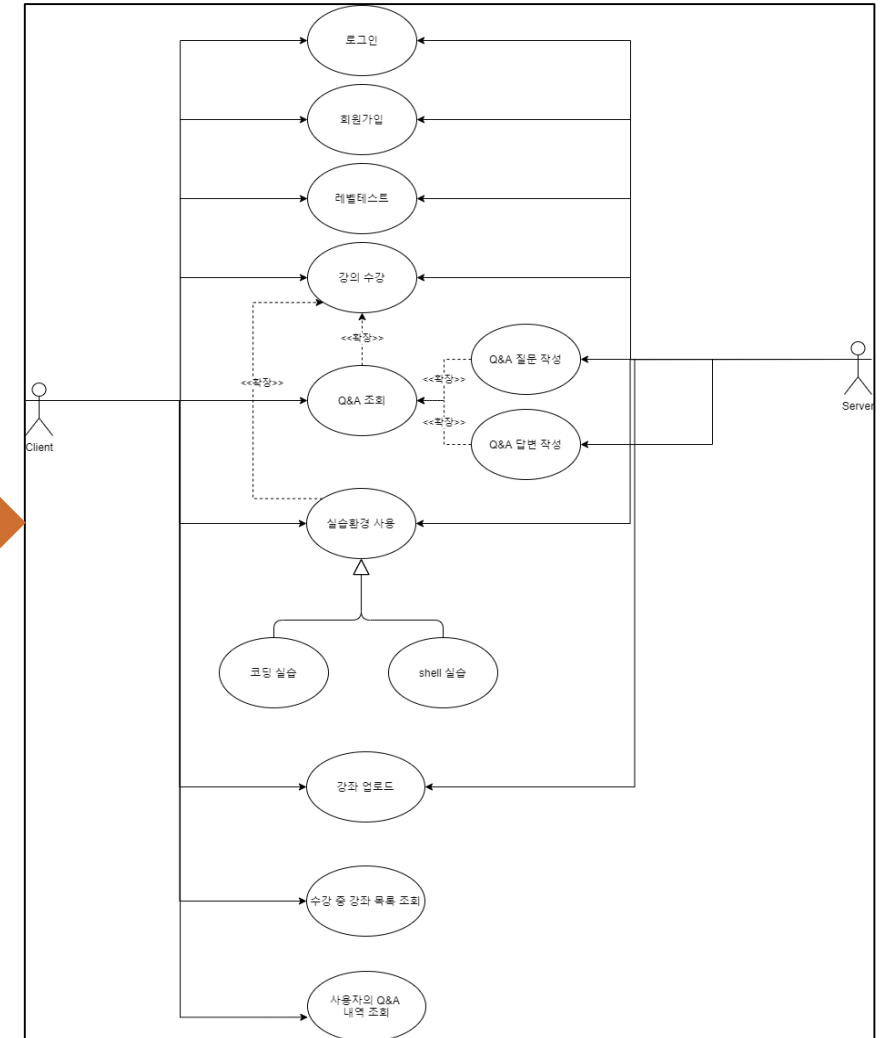


# Changed - 유스케이스

28개 -> 12개의 Specification, 핵심(실습 환경 구축)에 집중

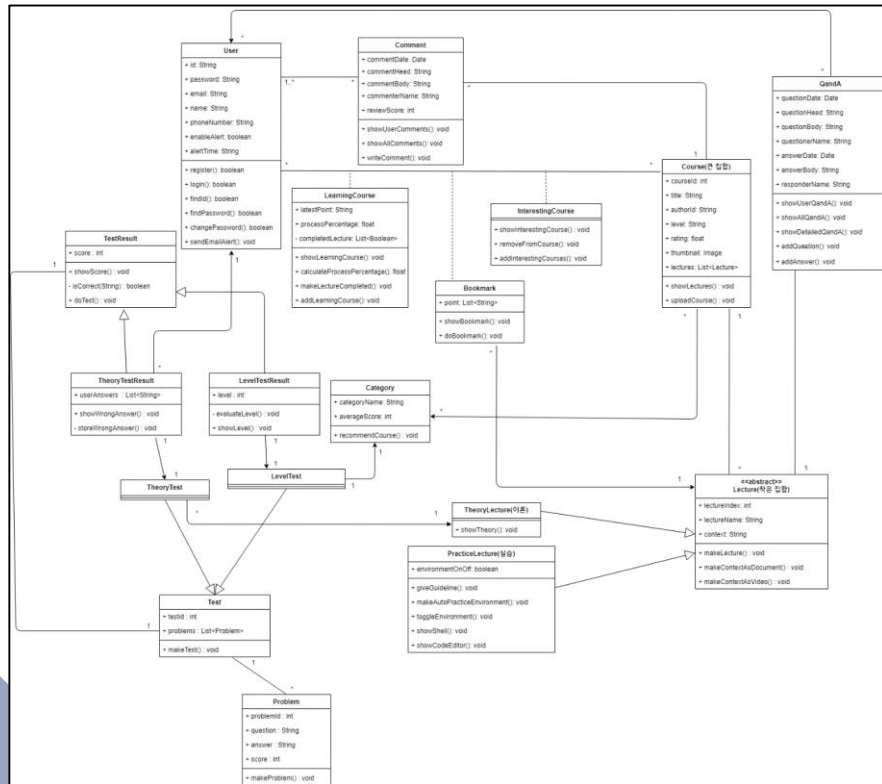


간소화

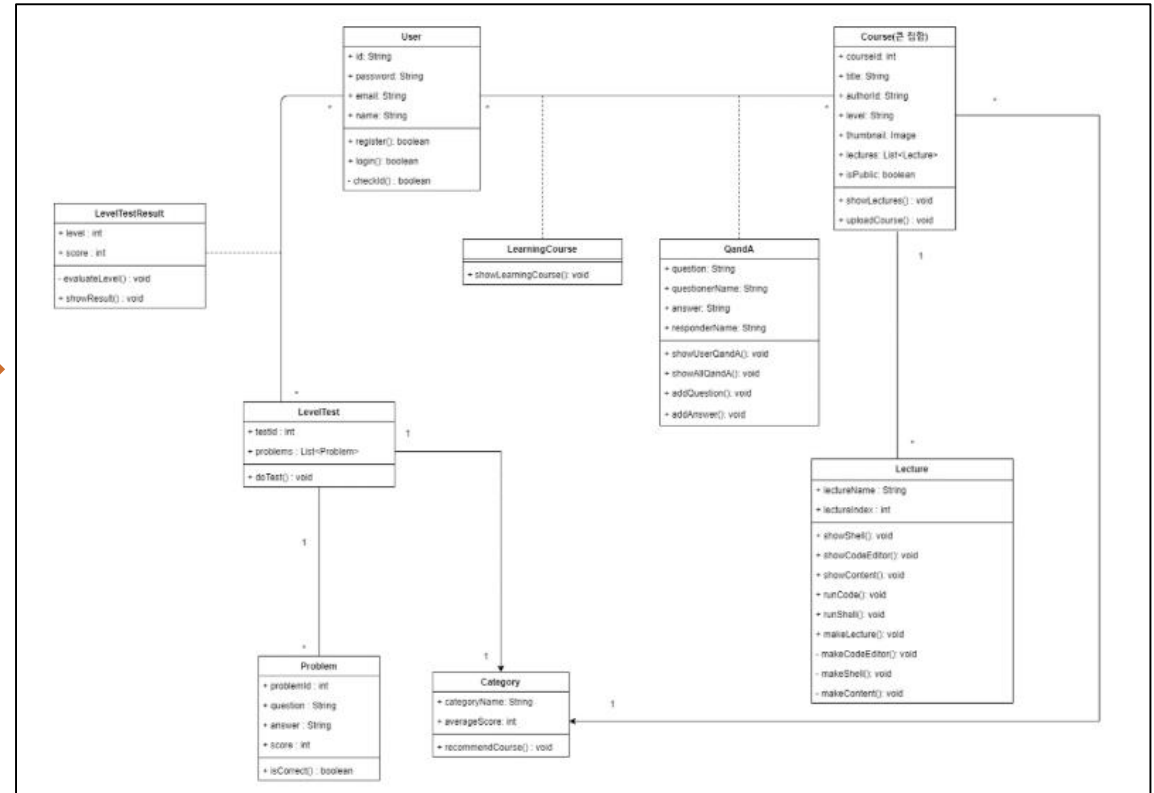


# Changed - 클래스다이어그램

18개 -> 9개의 Specification, 핵심(실습 환경 구축)에 집중



간소화



# Changed - 시퀀스다이어그램

## Table of Contents

### 목차

|                           |    |
|---------------------------|----|
| 1. INTRODUCTION .....     | 5  |
| 1.1. OBJECTIVE .....      | 5  |
| 2. USE CASE DIAGRAM ..... | 6  |
| 3. SEQUENCE DIAGRAM ..... | 9  |
| 3.1. 회원가입 .....           | 9  |
| 3.2. 로그인 .....            | 11 |
| 3.3. 아이디 찾기 .....         | 12 |
| 3.4. 비밀번호 찾기 .....        | 13 |
| 3.5. 레벨 테스트 .....         | 14 |
| 3.6. 강의 추천 .....          | 15 |
| 3.7. 이론 학습 .....          | 16 |
| 3.8. 복습 테스트 .....         | 17 |
| 3.9. 강의 평가 .....          | 18 |
| 3.10. 실습 학습 .....         | 19 |
| 3.11. 환경 가이드라인 .....      | 21 |
| 3.12. 자동 환경설정 .....       | 22 |
| 3.13. 강좌 업로드 .....        | 23 |
| 3.14. 테스트 생성 .....        | 26 |
| 3.15. 수강 중 강좌 목록 조회 ..... | 28 |
| 3.16. 강좌 이어서 수강 .....     | 29 |
| 3.17. 오답노트 확인 .....       | 30 |
| 3.18. 찜한 강좌 조회 .....      | 31 |
| 3.19. 찜한강좌 수강 .....       | 32 |
| 3.20. 찜한 강좌 삭제 .....      | 33 |
| 3.21. 강좌 재검색 조회 .....     | 34 |
| 3.22. 강의 재검색 사용 .....     | 35 |
| 3.23. Q&A 내역 조회 .....     |    |
| 3.24. Q&A 내역 상세 조회 .....  |    |
| 3.25. 비밀번호 변경 .....       |    |

간소화

|                            |    |
|----------------------------|----|
| 1.1. OBJECTIVE .....       | 4  |
| 2. USE CASE DIAGRAM .....  | 5  |
| 3. SEQUENCE DIAGRAM .....  | 6  |
| 3.1. 회원가입 .....            | 6  |
| 3.2. 로그인 .....             | 8  |
| 3.3. 레벨 테스트 .....          | 9  |
| 3.4. 강의 수강 .....           | 11 |
| 3.5. 코딩 실습 .....           | 12 |
| 3.6. SHELL 실습 .....        | 13 |
| 3.7. Q&A 조회 .....          | 14 |
| 3.8. Q&A 질문 작성 .....       | 15 |
| 3.9. Q&A 답변 작성 .....       | 16 |
| 3.10. 강좌 업로드 .....         | 17 |
| 3.11. 수강 중 강좌 목록 조회 .....  | 20 |
| 3.12. 사용자의 Q&A 내역 조회 ..... | 21 |

|                       |    |
|-----------------------|----|
| 3.26. 강좌 알림 관리 .....  | 40 |
| 3.27. 이메일 수신 관리 ..... | 41 |

04

설문조사

# 설문조사

SW와 관련이 있는 지인들에게 요청, 6명이 응답

## Code-At-Once 프로토타입 설문조사

안녕하세요. 종합설계 02분반 야간자율학습 팀입니다.

저희는 "SW자율학습 웹서비스+ 실습 환경 제공"이라는 주제로 졸업프로젝트를 진행하고 있습니다.

첨부된 영상은 4분 15초로, 한번 시청하신 후 설문해주시면 감사하겠습니다.

code-at-once는 codacademy, codesandbox, 백준 사이트처럼 실습 환경 및 코드 에디터를 제공합니다. 사용자는 환경설정에 대한 시간을 투자하지 않고 온전히 실습에 집중할 수 있습니다.

위 서비스들은 코드 에디터만 제공하지만, code-at-once는 shell을 제공합니다. 따라서 진짜 OS를 사용하는 것처럼 직접 Library를 설치할 수 있고 환경 변수, 커널, OS단에서 설정할 수 있는 값들 또한 수정할 수 있습니다. 각각의 shell은 유저 개개인에게 독립적으로 동작합니다.

코드 에디터에서는 내가 입력한 코드를 가상의 환경에서 컴파일하여 결과 값을 웹에서 보여줍니다.

code-at-once는 특정 유저가 환경을 설정하여 배포할 수 있습니다. 이를 Private으로 설정하여 강좌를 생성할 경우 공유 가능한 링크를 통해 배포된 환경이 복제되어 링크를 이용하는 사용자가 사용할 수 있습니다.

\* 필수항목

# 설문조사

SW와 관련이 있는 지인들에게 요청, 6명이 응답

## Code-At-Once

안녕하세요. 종합설계 02분반 야간자  
저희는 "SW자율학습 웹서비스+ 실습  
입니다.

첨부된 영상은 4분 15초로, 한번 시청

code-at-once는 codacademy, code  
공합니다. 사용자는 환경설정에 대한

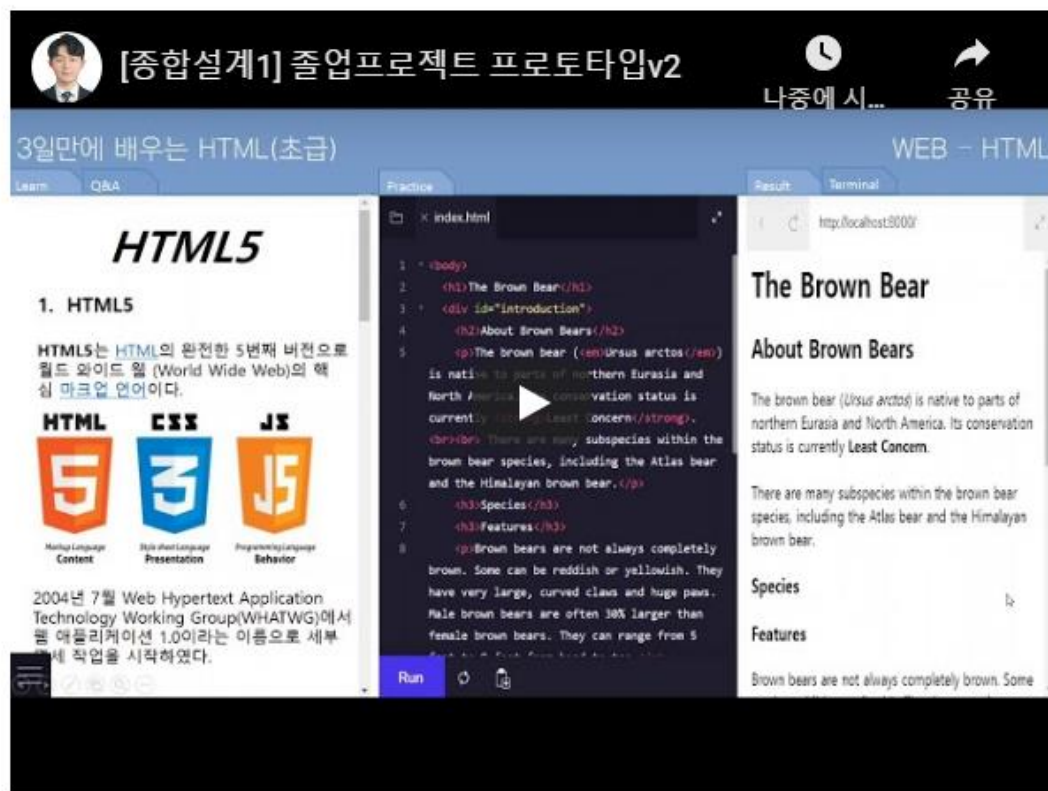
위 서비스들은 코드 에디터만 제공하  
를 사용하는 것처럼 직접 Library를  
값들 또한 수정할 수 있습니다. 각각

코드 에디터에서는 내가 입력한 코드가  
니다.

code-at-once는 특정 유저가 환경을  
강좌를 생성할 경우 공유 가능한 링  
가 사용할 수 있습니다.

\* 필수항목

## 프로토타입 영상



# 설문조사

---

1. 만약 환경설정의 번거로움을 해소하여 바로 코딩할 수 있는 서비스가 있다면 사용하시겠습니까?
2. codecademy, codesandbox.io, 백준과 같은 SW 학습 서비스를 사용해본 적이 있습니까?
  - 2-1. 위와 같은 SW 학습 서비스를 사용하면서 불편한 점이 있다면 적어주세요.
3. 영상에서 본 프로토타입 중 학습 과정에 대해서 추가되면 좋겠다, 혹은 없어도 되겠다라고 생각하는 기능이 있습니까?
4. 본 서비스가 출시된다면 사용할 의향이 있습니까?

# 설문조사

## 설문 결과



## 결론

사용자들은 기존의 다양한 기능을 제공할 때 보다 부족한 점을 느끼지 못하였고, 좀 더 전문화된 실습 환경 서비스 구축으로 프로젝트가 전환이 되었다.



05

프로토타입 데모

**야간 자율학습**