



Coding with me

PORTING MAUNAL

SSAFY 7 기 | A304

내용

1. 기술 스택.....	3
2. 서버 아키텍처.....	3
3. Openvidu 서버 설치.....	5
3.1 Openvidu 설치.....	5
3.1.1 도커 설치(설치된 경우 생략).....	5
3.1.2 기존 파일 삭제(docker, openvidu 데이터가 이미 있는 경우).....	6
3.1.3 openvidu On premises 설치 (공식문서).....	7
3.1.4 openvidu 설정 변경.....	7
3.1.5 포트 개방.....	9
3.1.6 Openvidu on promises 실행.....	9
3.2 Nginx 설치.....	10
3.3 Let's encrypt & Nginx 세팅.....	10
4. 컴파일러 서버 설치.....	11
4.1 컴파일러 설치.....	11
4.2 서버 배포.....	12
5. Spring/MySQL/Quasar 서버 설치.....	13
5.1 Mysql 설치.....	13
5.2 (Optional) Mysql 외부 접속 허용.....	13
5.3 Mysql 스키마 생성, 데이터 추가.....	14
5.4 Spring 빌드.....	14
5.5 Spring 배포.....	16
5.6 Quasar 설치.....	16
5.7 Quasar 실행.....	16
6. 시연 시나리오 – 공통.....	18

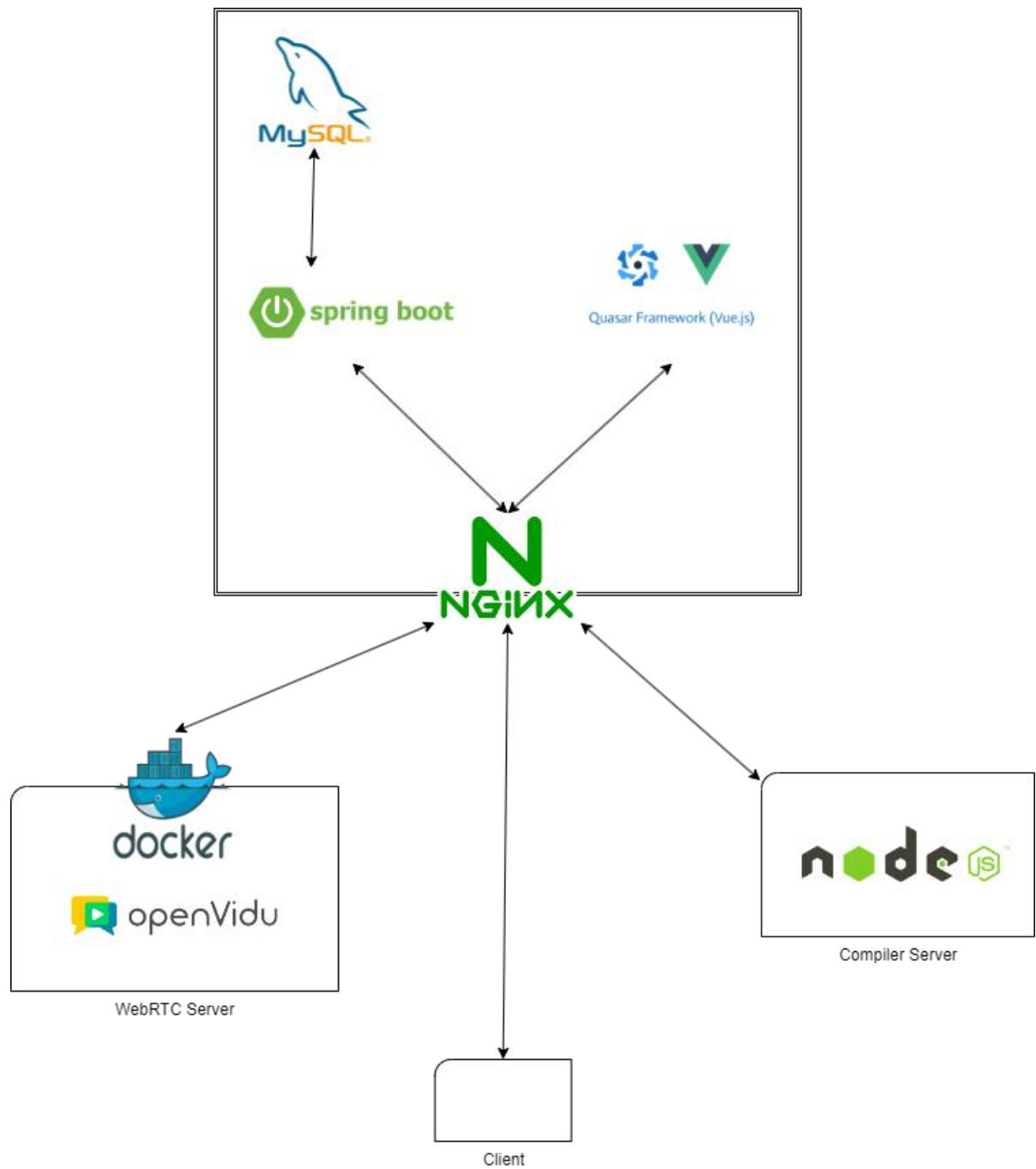
6.1	시작 페이지.....	18
6.2	로그인.....	19
6.3	회원가입.....	19
7.	시연 시나리오 – 강사.....	20
7.1	메인 페이지.....	20
7.2	반 개설 페이지.....	21
7.3	반 관리 페이지.....	21
7.4	시험 등록 페이지.....	22
7.5	강의 페이지.....	22
7.5.1	영상, 사이즈 조절.....	22
7.5.2	웹 IDE 기능.....	23
7.5.3	채팅, 온라인/오프라인 표시, 시험 시작 기능.....	24
7.6	시험 페이지(시험 감독).....	25
8.	시연 시나리오 – 학생.....	26
8.1	메인 페이지.....	26
8.2	강의 페이지.....	26
8.2.1	모드 변환.....	26
8.2.2	웹 IDE & 코드 동기화.....	27
8.3	시험 페이지.....	29
8.4	시험 성적 보기.....	30

1. 기술 스택

- 1.1 Spring/Quasar (Vue) Server: AWS EC2 Ubuntu 20.04 LTS
- 1.2 Openvidu Server: Oracle Cloud Ubuntu 20.04 LTS
(Openvidu Server의 경우 ARM CPU 사용불가)
- 1.3 Compiler Server: Oracle Cloud Ubuntu 20.04 LTS
- 1.4 Mysql: 20.04.2
- 1.5 Spring Boot: 2.4.5
- 1.6 Quasar: 1.0.5
- 1.7 Openvidu: 2.22.0
- 1.8 Nginx: 1.18.0
- 1.9 Node js: 10.19.0
- 1.10 java: 1.8.0
- 1.11 gcc: 9.4.0
- 1.12 g++: 9.4.0
- 1.13 Python: 3.8.10

2. 서버 아키텍처

- 성능을 위해, 컴파일러를 담당하는 서버와 WebRTC 서버를 별도로 분리해 총 3가지 서버로 이루어져 있음.
- Client에서 Nginx를 거쳐 Quasar로 접근해 웹페이지를 받아오고,
Client에서 Nginx를 거쳐 Spring을 통해 API를 호출한다.
- WebRTC Spring API를 통해 인증과 토큰 발급을 진행하지만, 그 이후는 client와 Openvidu 서버끼리만 통신한다.
- Client에서 컴파일을 요청하면 해당 정보를 Spring으로 보내고 Spring에서 컴파일러 서버에 코드 실행을 요청하고, 그 결과를 사용자에게 반환한다.



3. Openvidu 서버 설치

3.1 Openvidu 설치

3.1.1 도커 설치(설치된 경우 생략)

```
# 도커 설치
sudo apt-get update

sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

sudo mkdir -p /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

# sudo apt-get update 실행할 때
GPG error: https://download.docker.com/linux/ubuntu jammy
InRelease: The following signatures couldn't be verified because the public key is not
available: NO_PUBKEY 7EA0A9C3F273FCD8 에러 발생시

# 이 명령어 실행 후 재도전
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
#docker compose 설치
```

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

3.1.2 기존 파일 삭제(docker, openvidu 데이터가 이미 있는 경우)

```
#기존 openvidu 관련 docker image가 존재한다면 모두 삭제해 주고 시작해야한다.
```

```
# docker, openvidu 설치를 한적이 없으면 생략
```

```
sudo docker ps -a
```

```
#openvidu, kurento media server 등의 컨테이너가 존재한다면 삭제한다.
```

```
$ sudo docker rm <ID or Name>
```

```
#컨테이너 모두 삭제를 원할 경우
```

```
$ sudo docker rm $(docker ps -a)
```

```
$ sudo docker images
```

```
# 이미지도 삭제
```

```
$ sudo docker rmi <ID or IMAGE>
```

```
# 이미지 전체 삭제를 원할 경우
```

```
$ sudo docker rmi $(sudo docker images)
```

3.1.3 openvidu On premises 설치 ([공식문서](#))

```
# 관리자 권한
$ sudo su

# openvidu가 설치되는 경로
$ cd /opt

# openvidu on premises 설치
$ curl <https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh> | bash

$ exit
```

3.1.4 openvidu 설정 변경

```
$ cd /opt/openvidu

openvidu 설정 변경
인증서 타입의 기본값은 selfsigned이다.
만약 certbot을 이용해 인증서를 발급받았다면 설정을 변경해 주어야 한다.
참고: vi에서 ':'검색할 키워드'를 통해 빠르게 원하는 내용을 찾을 수 있다.
# /opt/openvidu

$ vi .env

# OpenVidu configuration
# -----
#Documentation: <https://docs.openvidu.io/en/stable/reference-docs/openvidu-config/>

# NOTE: This file doesn't need to quote assignment values, like most shells do.
# All values are stored as-is, even if they contain spaces, so don't quote them.

# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com

# 본인 서버의 도메인(example.com) 또는
# $ curl ifconfig.me 커맨드를 실행한 결과로 얻는 ip를 기입한다.
# openvidu 서버가 여기 작성한 도메인으로 실행된다.
DOMAIN_OR_PUBLIC_IP=<도메인 또는 public IP>
```



```
# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to OpenVidu Dashboard
```

```
OPENVIDU_SECRET=MY_SECRET
```

```
# HTTPS 인증서 별도로 등록안했으면 selfsigned
```

```
# 인증서를 certbot으로 발급받았다면 letsencrypt
```

```
CERTIFICATE_TYPE=letsencrypt
```

```
# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
```

```
# 만약 인증서 타입이 letsencrypt라면 이메일 설정.
```

```
LESENCRYPT_EMAIL=user@example.com
```

```
# Proxy configuration
```

```
# If you want to change the ports on which openvidu listens, uncomment the following lines
```

```
# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/ to be automatically
```

```
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
```

```
# WARNING: the default port 80 cannot be changed during the first boot
```

```
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
```

```
# NGINX의 포트를 변경한다.
```

```
HTTP_PORT=8081
```

```
# Changes the port of all services exposed by OpenVidu.
```

```
# SDKs, REST clients and browsers will have to connect to this port
```

```
HTTPS_PORT=8443
```

```
...
```

.env 파일 저장 후, 기존 openvidu 인증 파일 삭제

만약 /opt/openvidu경로 안에 certificates 폴더가 존재한다면 설정한 환경에 인증서가 제대로 적용되지 않을 때가 있다.

이 경우 폴더 자체를 삭제해 주면 된다.

```
# /opt/openvidu
```

```
$ sudo rm -rf certificates
```

openvidu On Promises 실행

3.1.5 포트 개방

포트 개방

aws도 동일하게 처리 필요

```
sudo apt update
```

```
sudo apt install netfilter-persistent
```

```
sudo iptables -A INPUT -p udp --match multiport --dports 40000:65535 -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp --match multiport --dports 40000:65535 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p tcp --dport 22 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p tcp --dport 443 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p tcp --dport 8081 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p udp --dport 8081 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p tcp --dport 8443 -j ACCEPT
```

```
sudo iptables -I INPUT 1 -p tcp --dport 8443 -j ACCEPT
```

```
service iptables save
```

```
service iptables restart
```

```
sudo netfilter-persistent save
```

```
sudo netfilter-persistent start
```

3.1.6 Openvidu on promises 실행

```
# /opt/openvidu
```

```
$ sudo ./openvidu start
```

```
# 종료할 때는 같은 경로에서
```

```
$ ./openvidu stop을 하면 된다.
```

3.2 Nginx 설치

#운영체제에서 사용 가능한 패키지들과 그 버전에 대한 정보(리스트) 업데이트

```
sudo apt update
```

#nginx 설치하기

```
sudo apt install nginx -y
```

#설치한 nginx 버전 확인

```
nginx -v
```

#nginx running 상태 확인

```
sudo service nginx status
```

3.3 Let's encrypt & Nginx 세팅

```
sudo apt install software-properties-common
```

```
add-apt-repository ppa:certbot/certbot
```

```
apt update
```

```
apt install python3-certbot-nginx
```

```
certbot certonly --nginx -d (도메인)
```

경로 이동

```
cd /etc/nginx/sites-available
```

파일 생성

```
sudo vi 도메인명.conf
```

파일에 아래와 같은 내용을 입력하면 된다.(도메인은 서버에 맞는걸로)

```
upstream openvidu {
```

```
    server 127.0.0.1:8443;
```

```
}
```

```
server {
```

```
    server_name (openvidu 도메인)
```

```
    location / {
```

```
        proxy_pass https://openvidu/;
```

```

        proxy_set_header X-Real-IP $remote_addr;
        #proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/(openvidu 도메인)/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/(openvidu 도메인)/privkey.pem; # managed by
Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

# 생성한 파일의 심볼릭 링크를 생성하고, nginx를 재시작한다.
sudo ln -s /etc/nginx/sites-available/도메인명.conf /etc/nginx/sites-enabled

# nginx 문법 검사
sudo nginx -t

#nginx 재시작
sudo service nginx reload

```

4. 컴파일러 서버 설치

4.1 컴파일러 설치

```

# 컴파일러를 설치하려는 서버에 접속 선행

# 업데이트
sudo apt update
# 서버 실행 & 자바스크립트 컴파일을 위한 nodejs 설치
sudo apt install nodejs
# 파이썬 설치
Sudo apt install python3
# c언어 설치
Sudo apt install gcc

```

```
# c++ 설치
Sudo apt install g++
# openjdk 1.8 설치
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 0x219BD9C9
sudo apt-add-repository 'deb http://repos.azulsystems.com/ubuntu stable main'
sudo apt update
sudo apt install zulu-8
```

4.2 서버 배포

```
# 프로젝트를 받아온다.
Git clone https://lab.ssafy.com/s07-webmobile1-sub2/S07P12A304.git

# 프로젝트 경로도 이동
cd S07P12A304/compiler

# 의존성 설치
npm i

# 서버 시작
node index.js
```

5. Spring/MySql/Quasar 서버 설치

- Spring/MySql/Quasar은 같은 서버에서 작동하는 것으로 설계되어있다.
- 같은 서버에 3개 전부 설치한 뒤에 nginx로 처리한다.

5.1 Mysql 설치

```
sudo apt update  
sudo apt install mysql-server
```

5.2 (Optional) Mysql 외부 접속 허용

- Spring과 mysql이 같은 서버에 있어 생략해도 실행에 문제가 없습니다.
- 개발 목적으로 외부에서 접속할 때 필요한 설정입니다.

```
cd /etc/mysql/mysql.conf.d  
  
sudo vi mysqld.cnf  
  
# mysqld.cnf 내용 변경. => bind-address를 0.0.0.0으로 변경한다.  
#bind-address = 127.0.0.1  
bind-address = 0.0.0.0  
  
# mysql 재시작  
sudo systemctl restart mysql  
  
# mysql 접속  
sudo mysql --defaults-file=/etc/mysql/debian.cnf  
  
# mysql에서 계정만들기  
( 'admin'과 'vn23d@jve'는 각각 아이디, 비밀번호로 원하는 대로 설정가능.)  
create USER 'admin'@'%' IDENTIFIED BY 'vn23d@jve';  
grant all privileges on *.* to 'admin'@'%';  
  
# mysql 종료 후, port 개방  
Exit
```

```
sudo iptables -I INPUT 1 -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
sudo iptables -L
```

```
sudo apt install netfilter-persistent
sudo apt install iptables-persistent
```

```
sudo netfilter-persistent save
sudo netfilter-persistent start
```

```
service iptables save
service iptables restart
```

5.3 Mysql 스키마 생성, 데이터 추가

```
# Mysql 접속 후, 스키마 생성
mysql -u admin -p
(비번 입력)
CREATE SCHEMA 'ssafy_web_db' DEFAULT CHARACTER SET utf8;

# 종료
Exit

# 더미 데이터 추가
(exec에 있는 backup.sql 파일 서버로 업로드 하여 사용)
mysql -u admin -p ssafy_web_db < backup.sql
```

5.4 Spring 빌드

```
# 프로젝트 받아오기
Git clone https://lab.ssafy.com/s07-webmobile1-sub2/S07P12A304.git

# Application.properties 설정 변경
Backend/src/main/resources/application.properties

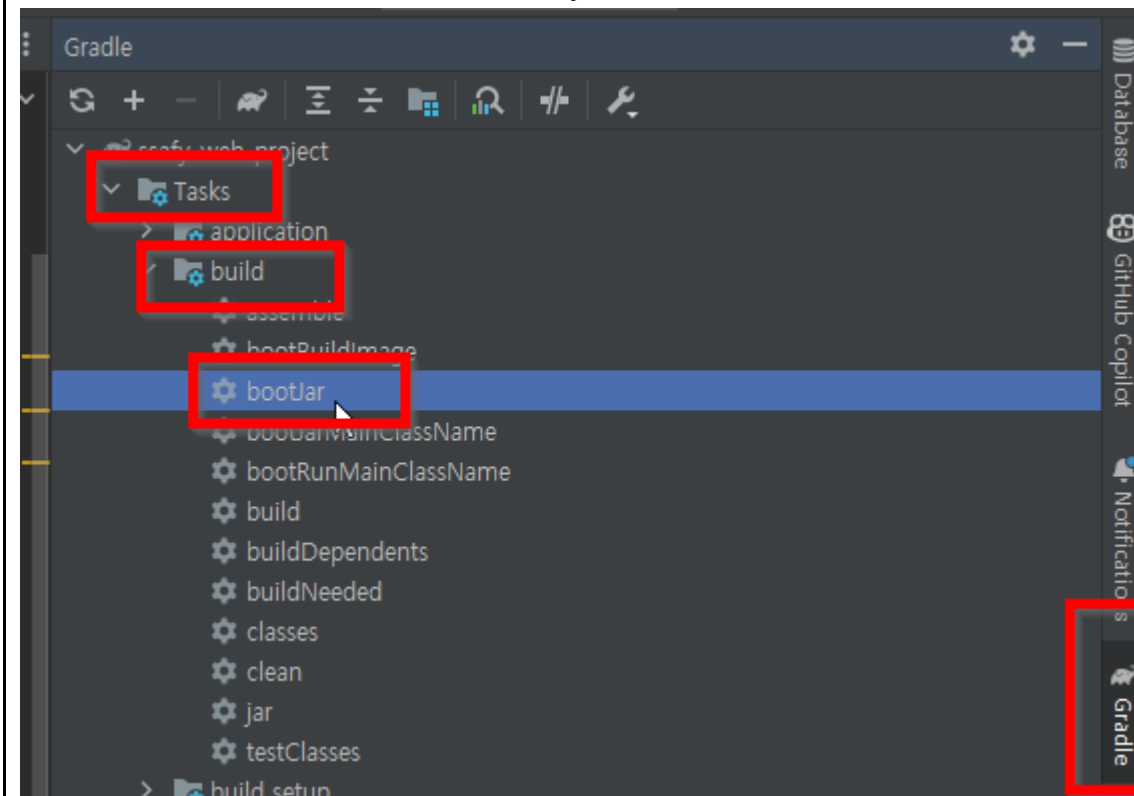
Mysql에서 설정한 스키마명(ssafy_web_db), username/password를 맞춰서 넣는다.
```

```
spring.data.web.pageable.one-indexed-parameters=true
#spring.datasource.url=jdbc:mysql://i7a304.p.ssafy.io:3306/ssafy_web
spring.datasource.url=jdbc:mysql://localhost:3306/ssafy_web_db?useUr
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.hikari.username=admin
spring.datasource.hikari.password=vn23d@jve
#spring.datasource.hikari.username=ssafy
#spring.datasource.hikari.password=ssafy
```

Openvidu가 있는 서버 도메인, .env 파일에 지정한 OPENVIDU_SECRET를 넣는다.

```
# controller?? ??? ???
openvidu.url=https://ssafy7.dev/
openvidu.secret=MY_SECRET
```

- Jar 파일을 생성해준다.(사진은 IntelliJ 기준)



5.5 Spring 배포

```
# openjdk 1.8 설치
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 0x219BD9C9
sudo apt-add-repository 'deb http://repos.azulsystems.com/ubuntu stable main'
sudo apt update
sudo apt install zulu-8

# 이전에 생성한 java 파일 실행
Java -jar ssafy-web-project-1.0-SNAPSHOT.jar
```

5.6 Quasar 설치

```
# 서버 실행 & 자바스크립트 컴파일을 위한 nodejs 설치
sudo apt install nodejs

# quasar 설치
sudo npm install vue
sudo npm install -g @vue/cli
sudo npm install -g @quasar/cli

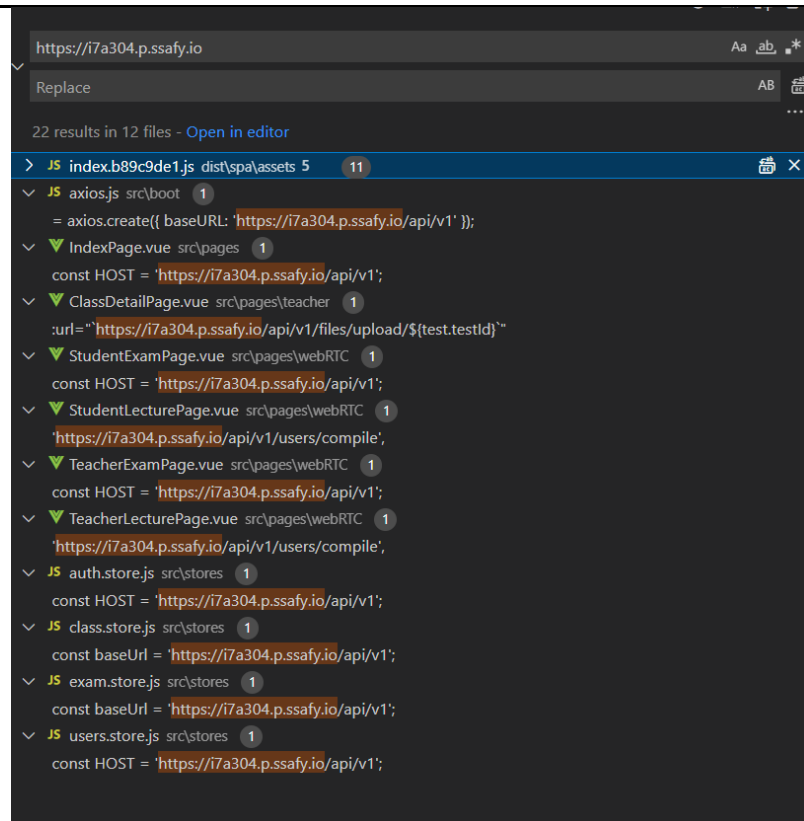
# quasar 설치 확인
quasar -v

# yarn 설치
sudo npm install yarn
```

5.7 Quasar 실행

```
# 프로젝트를 받아온다.
git clone https://lab.ssafy.com/s07-webmobile1-sub2/S07P12A304.git

# API 도메인이 다른 경우 맞는 주소로 변경해준다.
```



cd S07P12A304/FrontEnd

의존성 설치하고, 프로젝트 실행

yarn

sudo quasar dev

5.8 Let's encrypt, Nginx 설치

- [3.2](#), [3.3](#)과 동일하게 하되, nginx 설정만 아래로 한다.

```
upstream spring {
    server 127.0.0.1:8443;
}
server {
    server_name (도메인);
    location /api {
        proxy_pass http://spring;
        proxy_set_header X-Real-IP $remote_addr;
        #proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
    }
    location /{
        proxy_pass https://localhost:9000;
        proxy_set_header X-Real-IP $remote_addr;
        #proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
```

```
        proxy_set_header Connection "Upgrade";
    }
    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/(도메인)/fullchain.pem; # managed by
Certbot
    ssl_certificate_key /etc/letsencrypt/live/(도메인)/privkey.pem; # managed by
Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
```

6. 시연 시나리오 – 공통

6.1 시작 페이지

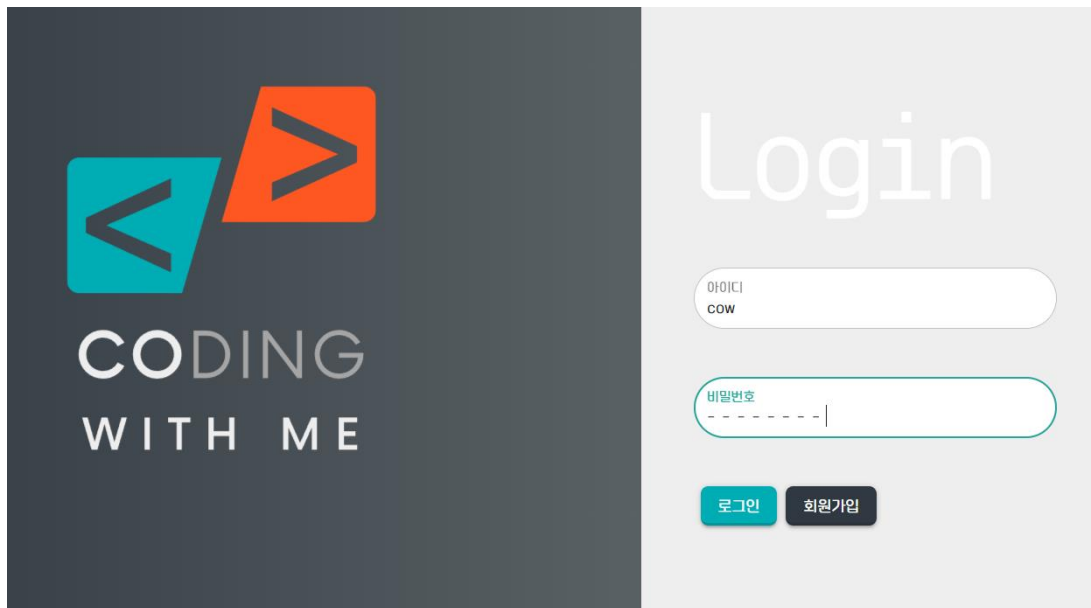
- 아무 곳이나 클릭하면 다음 페이지로 넘어간다.

내 옆의 선생님처럼, Coding with me.



6.2 로그인

- 샘플 계정 - ID : cow / PW : password



The login page features a dark blue header with the 'CODING WITH ME' logo, which consists of a teal less-than sign and an orange greater-than sign. The main content area is light gray and contains the word 'Login' in a large, white, sans-serif font. Below the title, there are two input fields: the first is labeled '아이디' (ID) with the text 'cow' inside, and the second is labeled '비밀번호' (Password) with a masked input field. At the bottom, there are two buttons: a teal '로그인' (Login) button and a dark gray '회원가입' (Sign Up) button.

6.3 회원가입

- 강사 / 학생으로 선택해서 가입할 수 있다.



The sign-in page has a dark blue header with the text 'Sign In' in white. The form is divided into two columns. The left column contains fields for '이름' (Name) with the text '강승길', '아이디' (ID) with 'teacher', '비밀번호' (Password) with a masked input, and '비밀번호 확인' (Confirm Password) with a masked input. The right column contains fields for '이메일' (Email) with 'ksgg1navercom@gmail.com', '전화번호' (Phone Number) with '010-5551-3212', '닉네임' (Nickname) with '강승길', and a date field with '2022-05-04'. Between the columns, there is an orange '중복확인' (Check Duplicate) button. At the bottom, there are two buttons: an orange '가입하기' (Sign Up) button and a teal '로그인하기' (Login) button. Below these buttons, there is a red-bordered box containing the text '강사' (Teacher) with a checked radio button and '학생' (Student) with an unchecked radio button, followed by the text '강사(으)로 가입합니다.' (Sign up as a teacher).

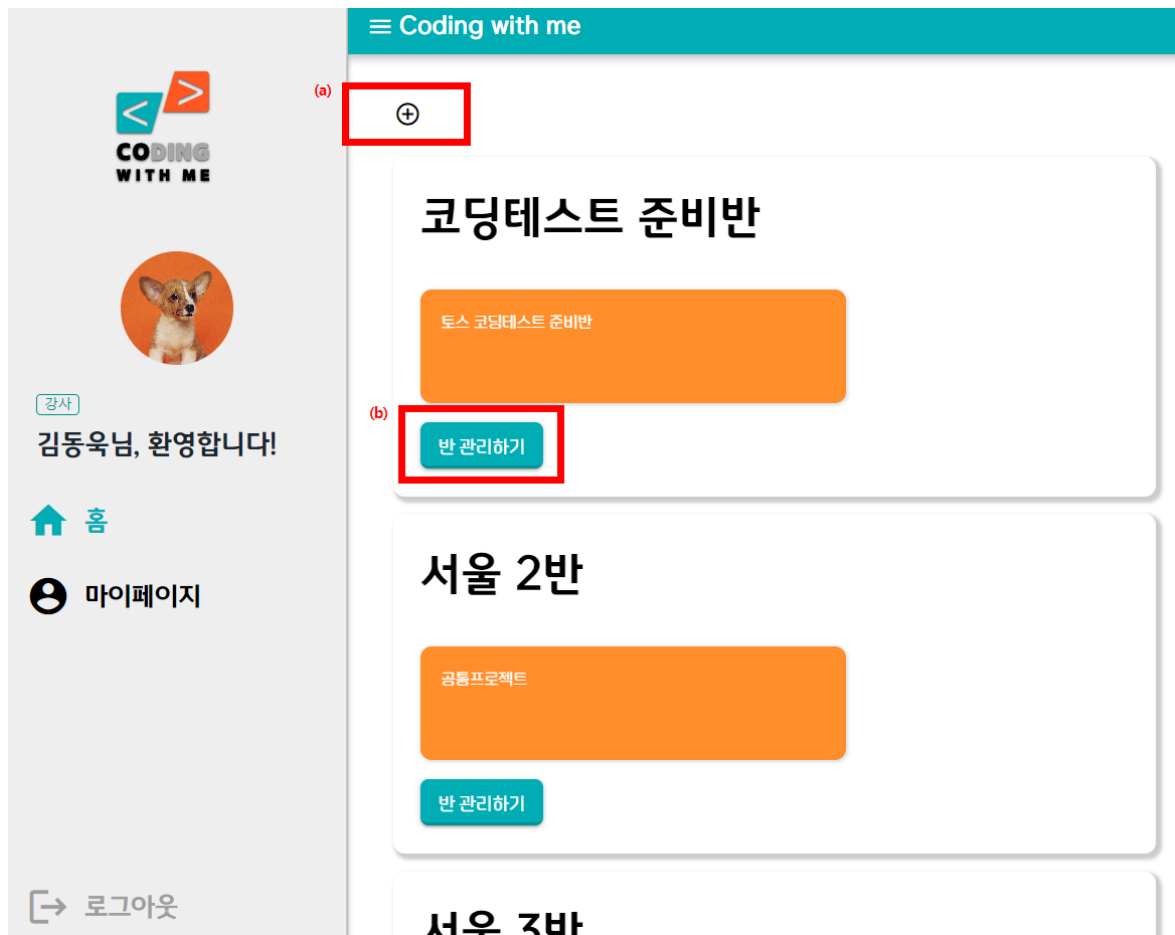
7. 시연 시나리오 – 강사

7.1 메인 페이지

- 강사는 여러 개의 반을 관리할 수 있고, 학생은 한 곳의 반에만 소속될 수 있다
- 강사가 반을 만들고, 학생들을 추가한다.

(a) : 새로운 반을 개설한다.

(b) : 해당 반 페이지로 이동



7.2 반 개설 페이지

≡ Coding with me

새로운 반 만들기

반 이름
서울 공통 3반

반 설명
싸피 서울 공통 3반입니다.

반 만들기

7.3 반 관리 페이지

- (a) 강의 시작 / (b) 반 정보 수정 / (c) 지난 강의 이력 리스트
- (d) 반에 학생 추가
- (e) 새로운 시험 등록
- (f) 시험 시험지 파일(pdf) 등록
- (g) 시험 정보 수정(시험 이름, 테스트 케이스 등등..)

The screenshot shows the '반 관리' (Class Management) interface. At the top, there is a header bar with a red button labeled '코딩테스트 준비반 강의 시작하기' (a) and a settings gear icon (b). Below this is a horizontal scrollable list of lecture cards (c). Each card displays a title, date, and status. The first card is titled 'test' with date '2022-08-17' and status '종료된 강의입니다.'. The other cards are titled '7주차 알고리즘 수업' (2022-08-16), '6주차 알고리즘 수업' (2022-08-16), '5주차 알고리즘 수업' (2022-08-02), and '4주차 알고리즘 수업' (2022-07-26), all with status '종료된 강의입니다.'. Below the lecture list, there are two main sections. The left section, '코딩테스트 준비반의 학생' (d), lists students with their names and IDs. The right section, '코딩테스트 준비반의 시험' (e), shows a list of exams. The first exam is '시험 파일(PDF)' with a progress bar at 8.08 / 8.08%. It has a plus icon (f) and a settings icon (g). Below it is '사칙연산'.

7.4 시험 등록 페이지

- 왼쪽에 시험 이름과 문제 개수를 지정할 수 있고, 각 문제 별로 여러 개의 테스트 케이스를 등록할 수 있다.
- 테스트 케이스 형식은 SW Expert Academy와 동일하게 Input과 Output으로 되어 있으며, 작성한 코드의 실행결과 Output이랑 동일하면 정답으로 처리된다.

시험 만들기

시험 이름: 알고리즘 테스트 증복 체크

문제 개수: 5

가이드

1번 문제

2번 문제

3번 문제

4번 문제

5번 문제

+ 추가

1번째 테스트 케이스 삭제

Input

5
10

Output

15

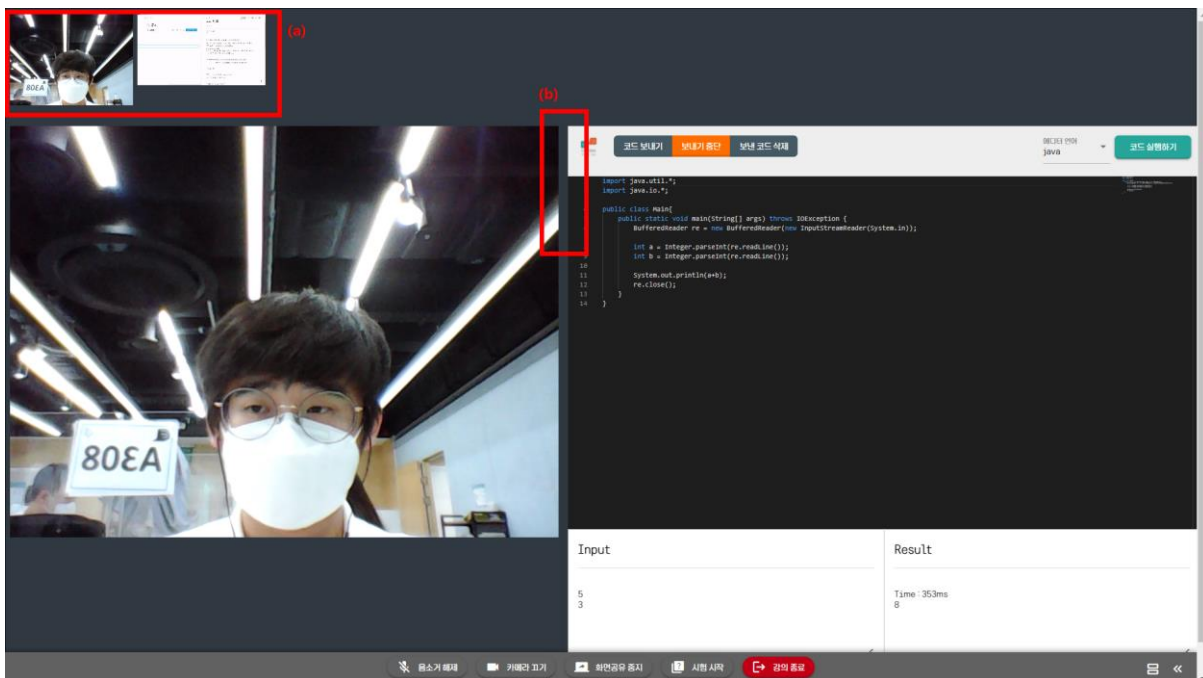
시험 만들기

7.5 강의 페이지

7.5.1 영상, 사이즈 조절

(a) : 영상을 큰 영상으로 표시할 수 있다.

(b) : 좌우 드래그를 통해 너비를 조정할 수 있다.



7.5.2 웹 IDE 기능

- (a) 강사가 작성한 코드를 실시간 동기화를 통해 학생에게 그대로 보여줄 수 있다.

코드 보내기: 실시간 동기화 on

보내기 중단: 실시간 동기화 off, 학생은 마지막으로 받은 코드에서 그대로 유지됨.

- (b) 웹 IDE 언어 선택: 5가지 언어를 지원한다.

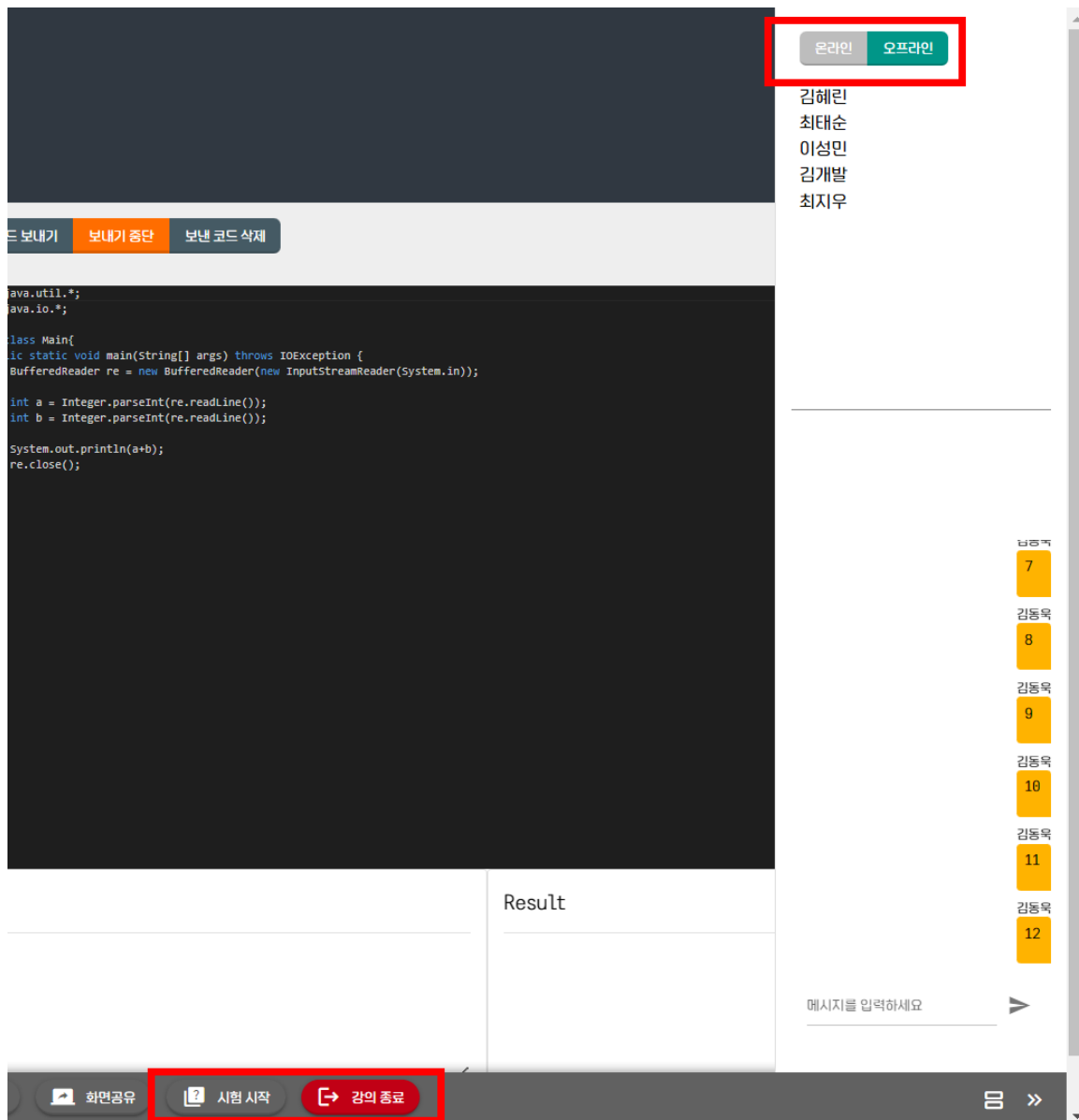
- (c) 코드 실행 결과 표시 : Input 값에 따른 실행 결과가 표시된다. (실행 시간/출력)이 표시되며, 컴파일 오류/런타임 오류/시간 초과 표시도 지원한다.

The screenshot displays a web IDE interface with three main sections highlighted by red boxes and labels (a), (b), and (c).

- (a) Top Bar:** Contains three buttons: "코드 보내기" (Send Code), "보내기 중단" (Stop Sending), and "보낸 코드 삭제" (Delete Sent Code).
- (b) Language Selection:** A dropdown menu labeled "에디터 언어" (Editor Language) is open, showing a list of supported languages: java, python, javascript, c++, and c. The "java" option is currently selected.
- (c) Execution Results:** A panel at the bottom is divided into two columns: "Input" and "Result".
 - Input:** Contains the text "dfds".
 - Result:** Displays the execution output, including the time "Time : 0ms" and an exception message: "Exception in thread 'main' java.lang.NumberFormatException: For input string: 'dfds'". Below this, the full exception details are shown: "java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)".

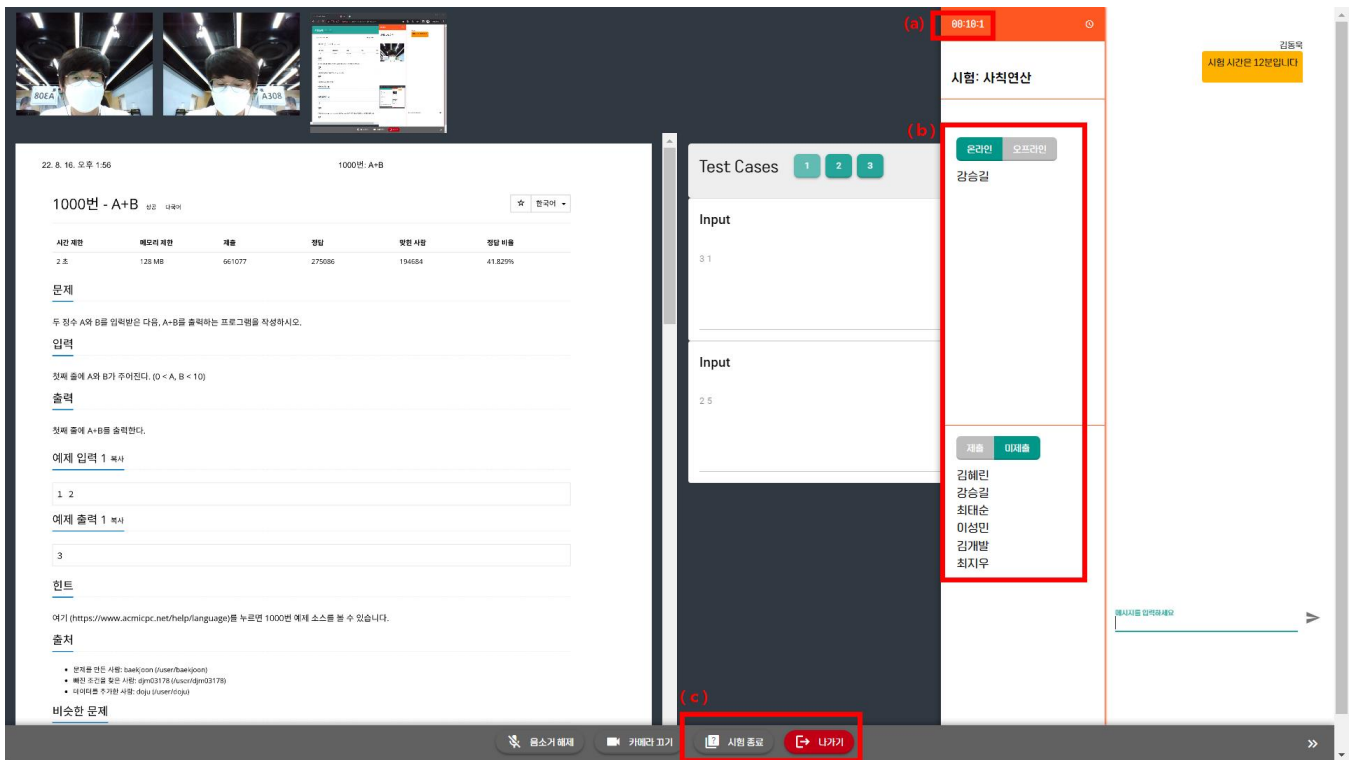
7.5.3 채팅, 온라인/오프라인 표시, 시험 시작 기능

- 화면 우측에 참석자/채팅 창을 표시 할 수 있으며, 그 반 학생들의 온라인/오프라인 여부가 표시된다.
- 화면 하단에는 시험 시작 버튼이 있으며, 강사가 시험과 제한 시간을 고르면, 강사와 학생이 자동으로 시험 페이지로 이동한다.



7.6 시험 페이지(시험 감독)

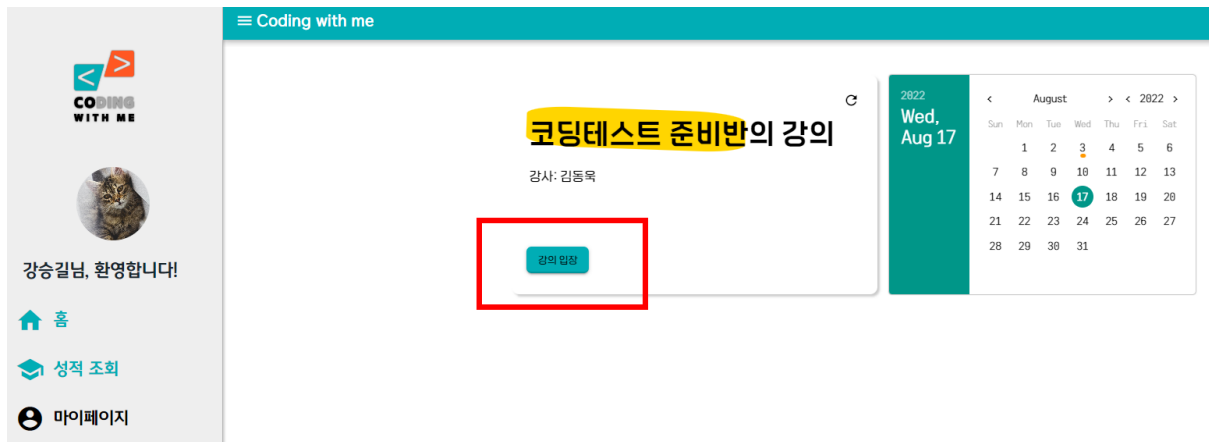
- 상단에 참가자들의 얼굴/화면이 표시되고, 화면 정중앙에는 문제지와 테스트 케이스가 표시된다.
 - (a) 남은 시간이 표시된다.
 - (b) 온라인/오프라인 여부, 학생 제출/미제출 여부가 표시된다.
 - (c) 시험 종료를 누르면 강사/학생이 강의 페이지로 이동되며, 나가기 버튼은 모든 참가자의 화상 회의가 종료된다.



8. 시연 시나리오 – 학생

8.1 메인 페이지

- 강사가 개설한 강의가 있을 경우 입장 버튼이 표시된다.

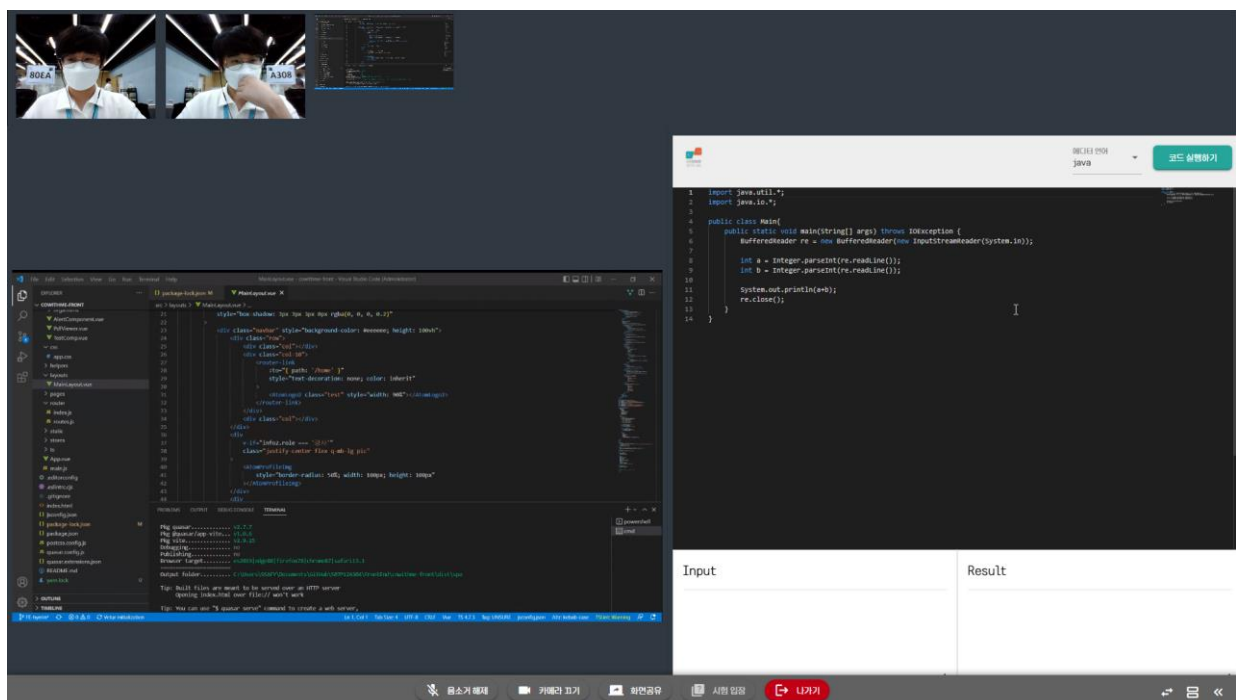


8.2 강의 페이지

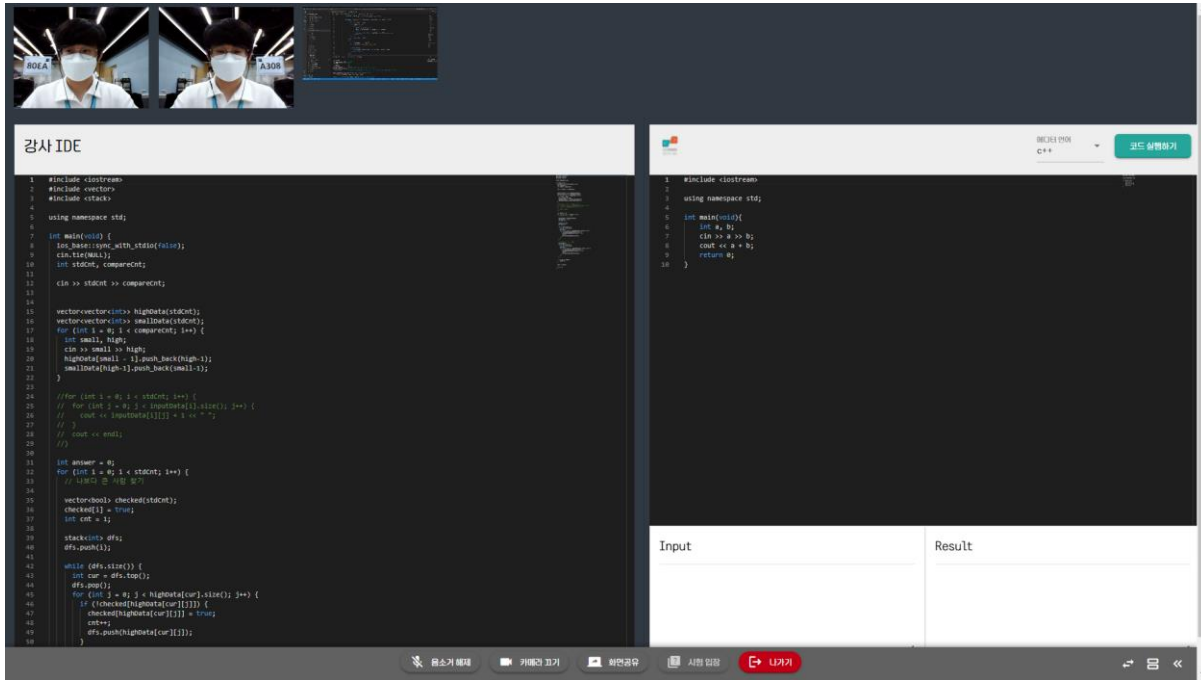
8.2.1 모드 변환

- 강사 얼굴이나 화면 공유를 볼 수 있는 모드1과 강사 IDE를 볼 수 있는 모드2로 자유롭게 변환할 수 있다.

(a) 모드 1

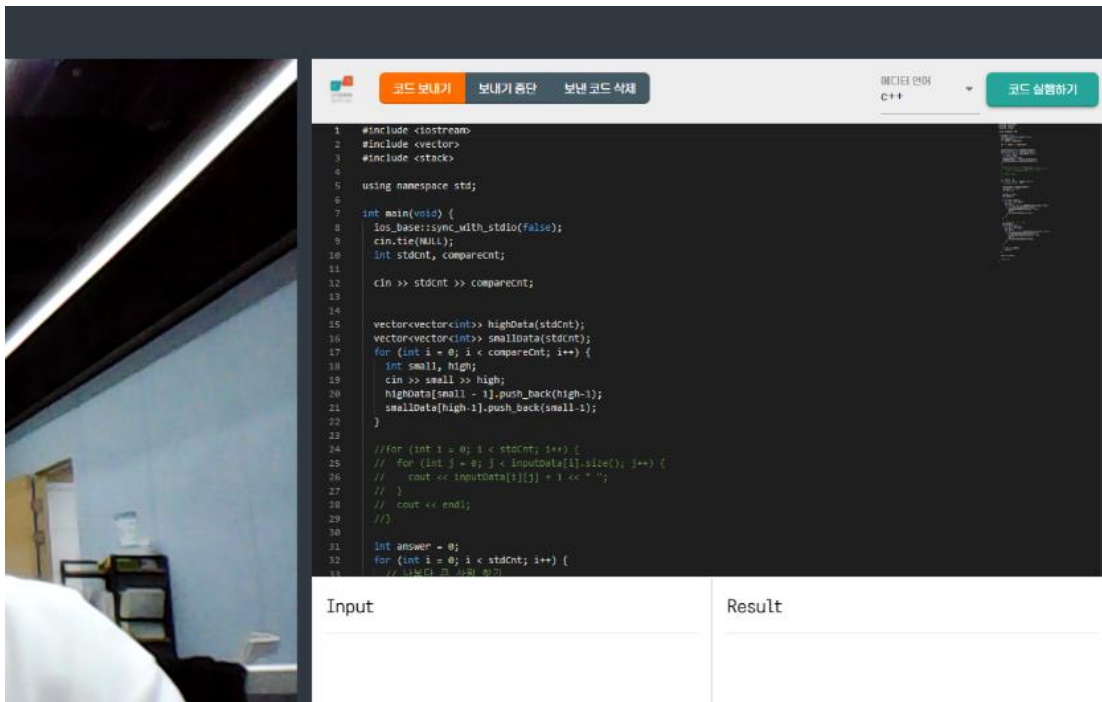


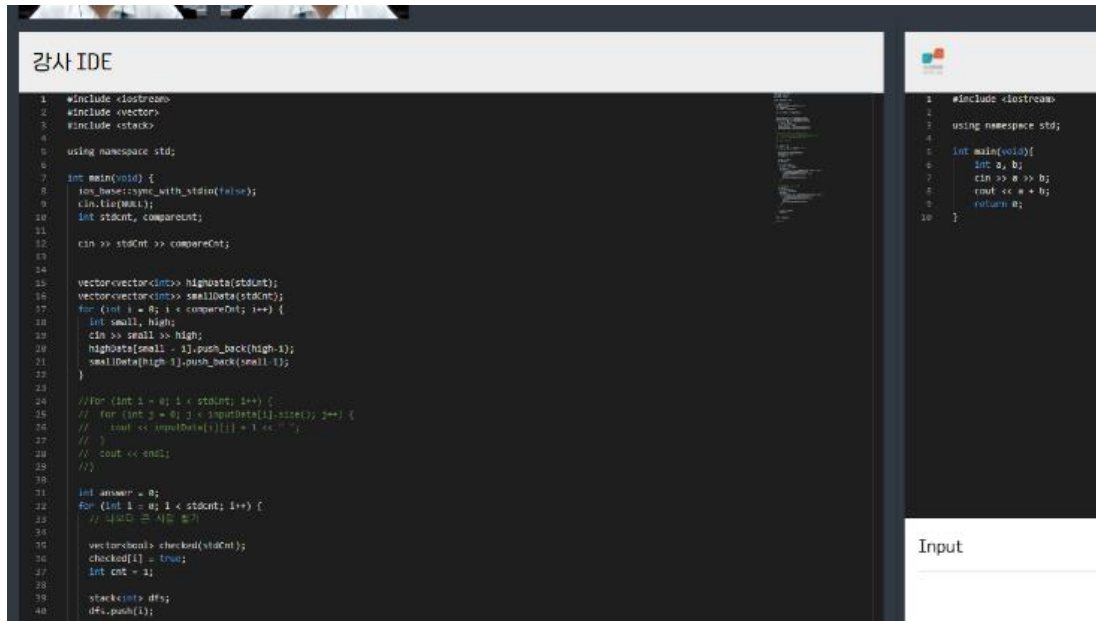
(b) 모드 2



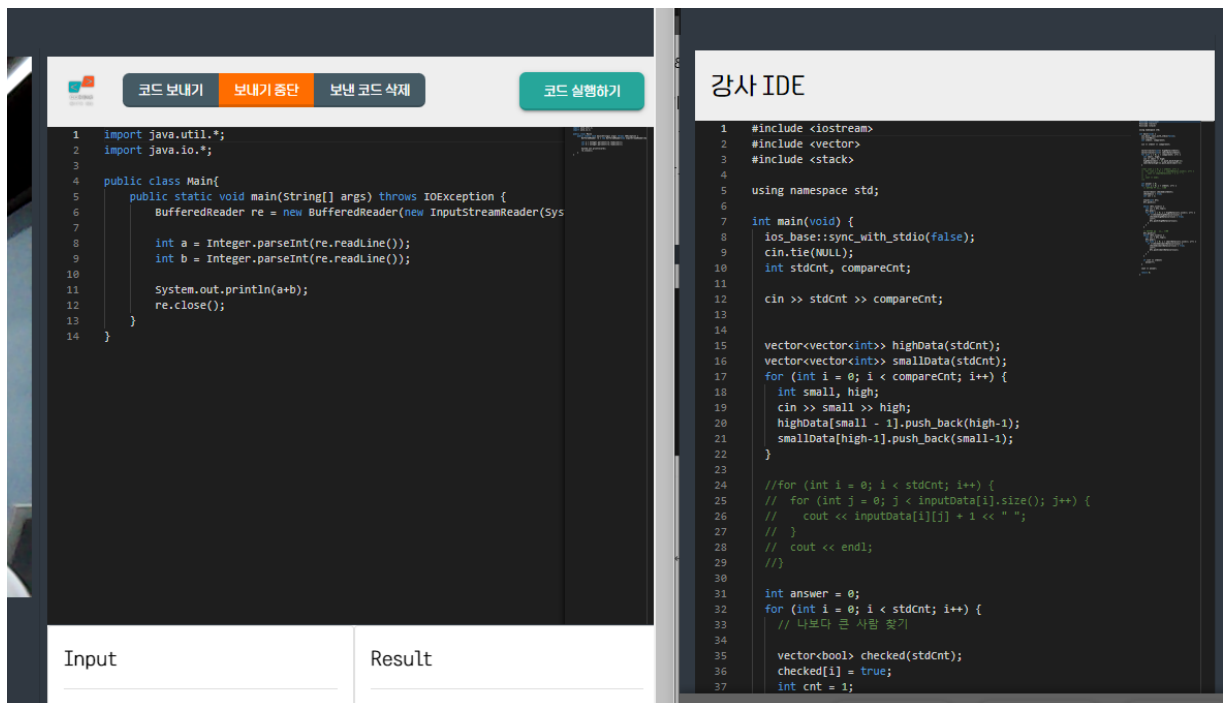
8.2.2 웹 IDE & 코드 동기화

- 강사가 코드 보내기 모드를 했을 때, 강사 화면(1번째 사진)과 학생 화면(2번째)에 있는 IDE가 실시간으로 동기화 된다.

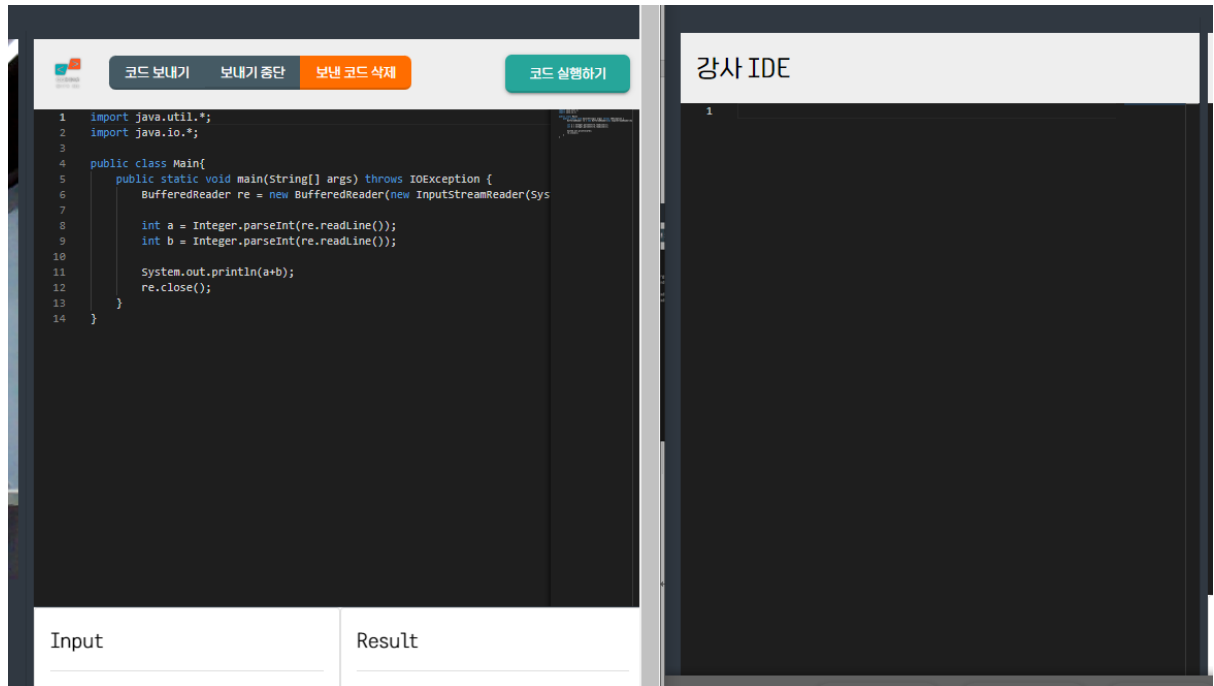




- 모드가 보내기 중단으로 되어 있을 때는 변경사항이 반영 되지 않는다.

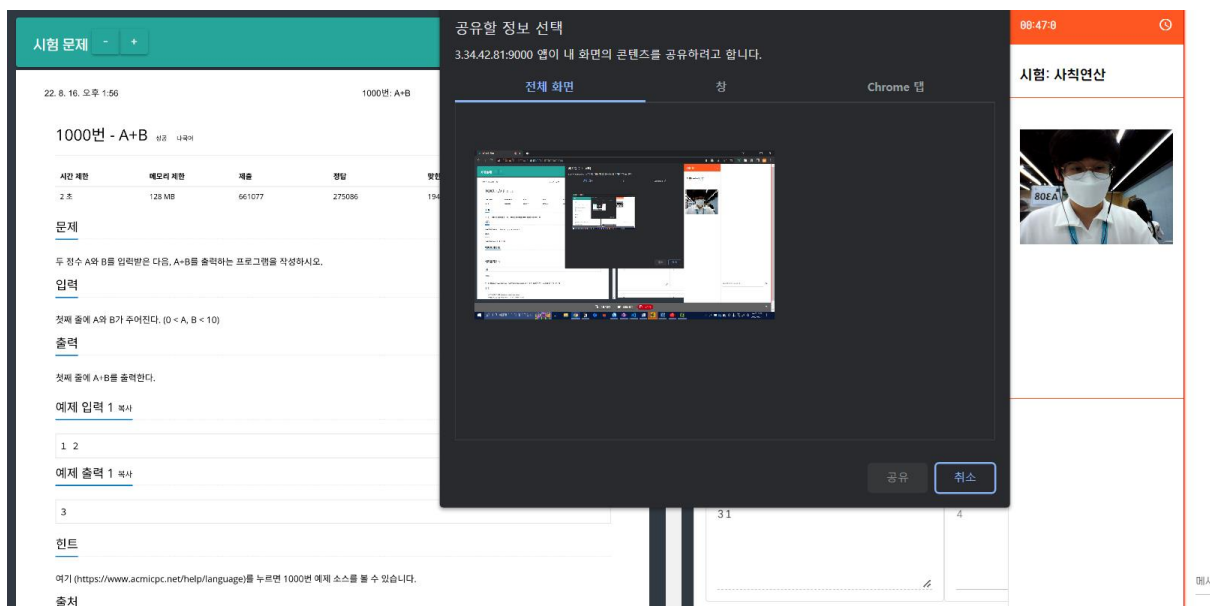


- 보낸 코드 삭제 모드의 경우 학생한테 IDE의 내용이 표시되지 않는다.



8.3 시험 페이지

- 강사가 시험 시작을 누르면 자동으로 이동되며, 화면 공유 요청을 표시됩니다.
- 강사와 동일하게 시험지, 테스트 케이스, 시험 이름과 시간이 표시되지만, 자기 얼굴/화면 외에 다른 사람 화면은 볼 수 없습니다.



(a) 시험지 확대/축소 기능

(b) 시험에 사용할 언어 선택

(c) 작성한 답안 제출 / 코드 실행 기능 => 제출하면 강사화면에서 제출 완료로 표시됨.

(d) 각 테스트 케이스에 대해 실행 시간 / 결과 표시

The screenshot displays the Coding with Me exam interface. On the left, the problem details for '1000번 - A+B' are shown, including a table of statistics and the problem description. The main area contains a code editor with Java code for solving the problem. On the right, the test results are displayed, showing the input, answer, and execution time for each test case. Red boxes highlight specific features: (a) the zoom controls, (b) the language selection dropdown, (c) the submit and run buttons, and (d) the test results table.

8.4 시험 성적 보기

- 홈 화면 - 성적 조회에서 시험 점수 / 작성한 코드를 볼 수 있다.

The screenshot shows the Coding with Me home screen. On the left, there is a sidebar with navigation links: '홈' (Home), '성적 조회' (Check Score), and '마이페이지' (My Page). The '성적 조회' link is highlighted with a red box. The main area displays the exam score for the '2022-08-16 시험' (2022-08-16 Exam), showing a score of 3/3. Below the score, the code for the exam is displayed. Red boxes highlight the '성적 조회' link and the exam score.