

Coding with me

PORTING MAUNAL

SSAFY 7기 a304

2022

내용

[1. 기술 스택 3](#_Toc111648573)

[2. 서버 아키텍쳐 3](#_Toc111648574)

[3. Openvidu 서버 설치 5](#_Toc111648575)

[3.1 Openvidu 설치 5](#_Toc111648576)

[3.1.1 도커 설치(설치된 경우 생략) 5](#_Toc111648577)

[3.1.2 기존 파일 삭제(docker, openvidu 데이터가 이미 있는 경우 6](#_Toc111648578)

[3.1.3 openvidu On premises 설치 **(공식문서)** 7](#_Toc111648579)

[3.1.4 openvidu 설정 변경 7](#_Toc111648580)

[3.1.5 포트 개방 9](#_Toc111648581)

[3.1.6 Openvidu on promises 실행 9](#_Toc111648582)

[3.2 Nginx 설치 10](#_Toc111648583)

[3.3 Let’s encrypt & Nginx 세팅 10](#_Toc111648584)

[4. 컴파일러 서버 설치 11](#_Toc111648585)

[4.1 컴파일러 설치 11](#_Toc111648586)

[4.2 서버 배포 12](#_Toc111648587)

[5. Spring/MySql/Quasar 서버 설치 13](#_Toc111648588)

[5.1 Mysql 설치 13](#_Toc111648589)

[5.2 (Optional) Mysql 외부 접속 허용 13](#_Toc111648590)

[5.3 Mysql 스키마 생성, 데이터 추가 14](#_Toc111648591)

[5.4 Spring 빌드 14](#_Toc111648592)

[5.5 Spring 배포 16](#_Toc111648593)

[5.6 Quasar 설치 16](#_Toc111648594)

[5.7 Quasar 실행 16](#_Toc111648595)

[6. 시연 시나리오 – 공통 18](#_Toc111648596)

[6.1 시작 페이지 18](#_Toc111648597)

[6.2 로그인 19](#_Toc111648598)

[6.3 회원가입 19](#_Toc111648599)

[7. 시연 시나리오 – 강사 20](#_Toc111648600)

[7.1 메인 페이지 20](#_Toc111648601)

[7.2 반 개설 페이지 21](#_Toc111648602)

[7.3 반 관리 페이지 21](#_Toc111648603)

[7.4 시험 등록 페이지 22](#_Toc111648604)

[7.5 강의 페이지 22](#_Toc111648605)

[7.5.1 영상, 사이즈 조절 22](#_Toc111648606)

[7.5.2 웹 IDE 기능 23](#_Toc111648607)

[7.5.3 채팅, 온라인/오프라인 표시, 시험 시작 기능 24](#_Toc111648608)

[7.6 시험 페이지(시험 감독) 25](#_Toc111648609)

[8. 시연 시나리오 – 학생 26](#_Toc111648610)

[8.1 메인 페이지 26](#_Toc111648611)

[8.2 강의 페이지 26](#_Toc111648612)

[8.2.1 모드 변환 26](#_Toc111648613)

[8.2.2 웹 IDE & 코드 동기화 27](#_Toc111648614)

[8.3 시험 페이지 29](#_Toc111648615)

[8.4 시험 성적 보기 30](#_Toc111648616)

# 기술 스택

* 1. Spring/Quasar (Vue) Server: AWS EC2 Ubuntu 20.04 LTS
  2. Openvidu Server: Oracle Cloud Ubuntu 20.04 LTS

(Openvidu Server의 경우 ARM CPU 사용불가)

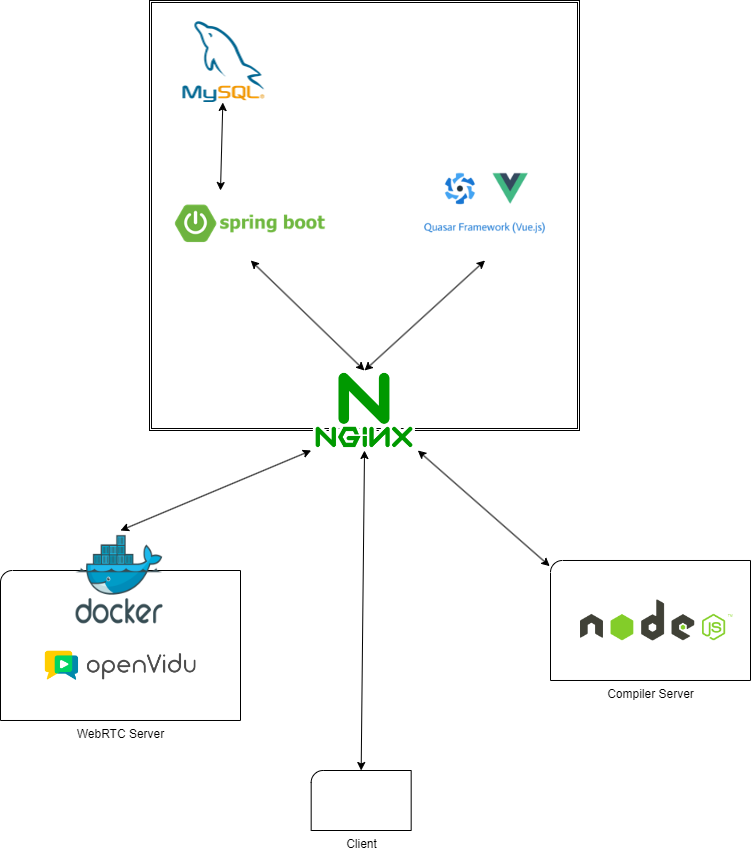
* 1. Compiler Server: Oracle Cloud Ubuntu 20.04 LTS
  2. Mysql: 20.04.2
  3. Spring Boot: 2.4.5
  4. Quasar: 1.0.5
  5. Openvidu: 2.22.0
  6. Nginx: 1.18.0
  7. Node js: 10.19.0
  8. java: 1.8.0
  9. gcc: 9.4.0
  10. g++: 9.4.0
  11. Python: 3.8.10

# 서버 아키텍쳐

* 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.



# Openvidu 서버 설치

### Openvidu 설치

### 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 \  $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  sudo apt-get update  sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin  # **sudo apt-get update 실행할 때**  **GPG error: https://download.docker.com/linux/ubuntu jammy**  **InRelease: The foll owing signatures couldn't be verified because the public key is not available: N O\_PUBKEY 7EA0A9C3F273FCD8 에러 발생시**  # 이 명령어 실행 후 재도전  sudo chmod a+r /etc/apt/keyrings/docker.gpg  #docker compose 설치  sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  sudo chmod +x /usr/local/bin/docker-compose |

### 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 server등의 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) |

### **openvidu On premises 설치**[**(공식문서)**](https://docs.openvidu.io/en/stable/deployment/ce/on-premises/)

|  |
| --- |
| # 관리자 권한  $ sudo su  # openvidu가 설치되는 경로  $ cd /opt  # openvidu on promises 설치  $ curl <https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install\_openvidu\_latest.sh> | bash  $ exit |

### 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.  LETSENCRYPT\_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** |

### 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 |

### Openvidu on promises 실행

|  |
| --- |
| # /opt/openvidu  $ sudo ./openvidu start  # 종료할 때는 같은 경로에서  $ ./openvidu stop을 하면 된다. |

### 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 |

### 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 |

# 컴파일러 서버 설치

### 컴파일러 설치

|  |
| --- |
| # 컴파일러를 설치하려는 서버에 접속 선행  # 업데이트  sudo apt update  # 서버 실행 & 자바스크립트 컴파일을 위한 node js 설치  sudo apt install nodejs  # 파이썬 설치  Sudo apt install python3  # c언어 설치  Sudo apt install gcc  # c++ 설치  Sudo apt install g++  # open jdk 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 |

|  |
| --- |
| # 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 |

### 서버 배포

# Spring/MySql/Quasar 서버 설치

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

### Mysql 설치

|  |
| --- |
| sudo apt update  sudo apt install mysql-server |

### (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 |

### 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 |

### 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.    Openvidu가 있는 서버 도메인, .env 파일에 지정한 OPENVIDU\_SECRET를 넣는다.     * Jar 파일을 생성해준다.(사진은 Intellij 기준) |

### Spring 배포

|  |
| --- |
| # open jdk 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 |

### Quasar 설치

|  |
| --- |
| # 서버 실행 & 자바스크립트 컴파일을 위한 node js 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 |

### 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 |

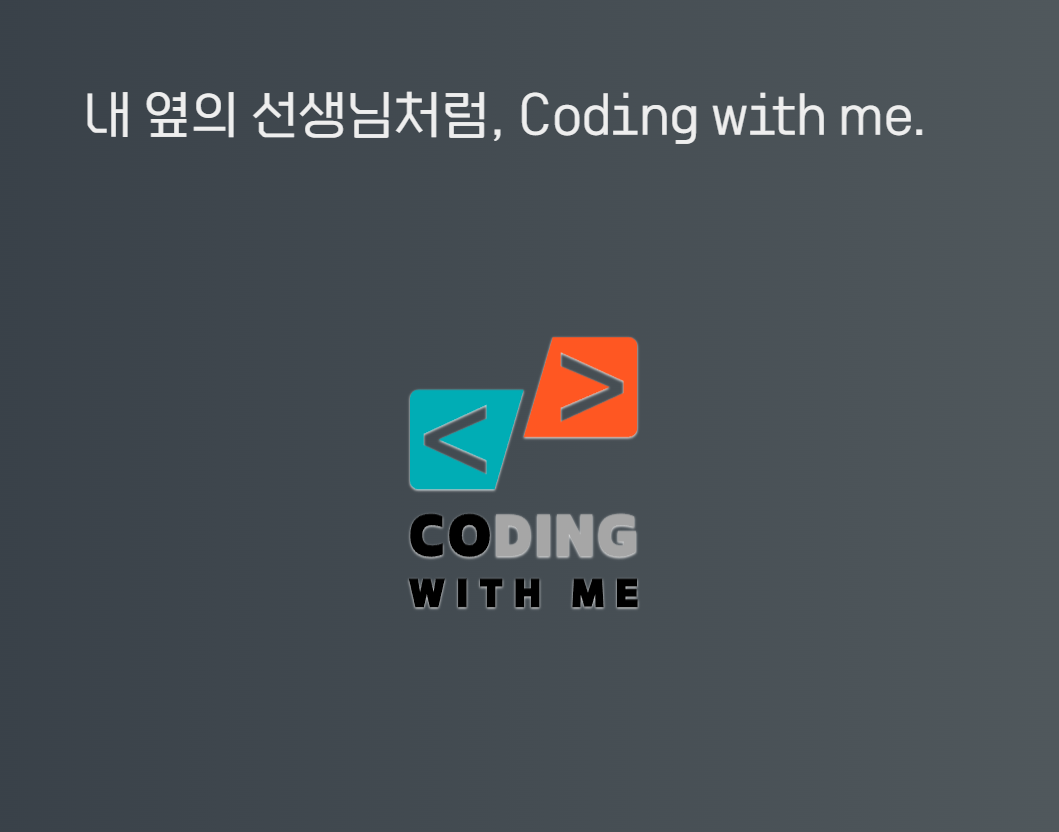
* 1. Let’s encrypt, Nginx 설치
* [3.2](#_Nginx_설치), [3.3](#_Let’s_encrypt_&)과 동일하게 하되, nginx설정만 아래로 한다.

|  |
| --- |
| 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 |

# 시연 시나리오 – 공통

### 시작 페이지

* 아무 곳이나 클릭하면 다음 페이지로 넘어간다.



### 로그인

* 샘플 계정 - ID : cow / PW : password



### 회원가입

* 강사 / 학생으로 선택해서 가입할 수 있다.



# 시연 시나리오 – 강사

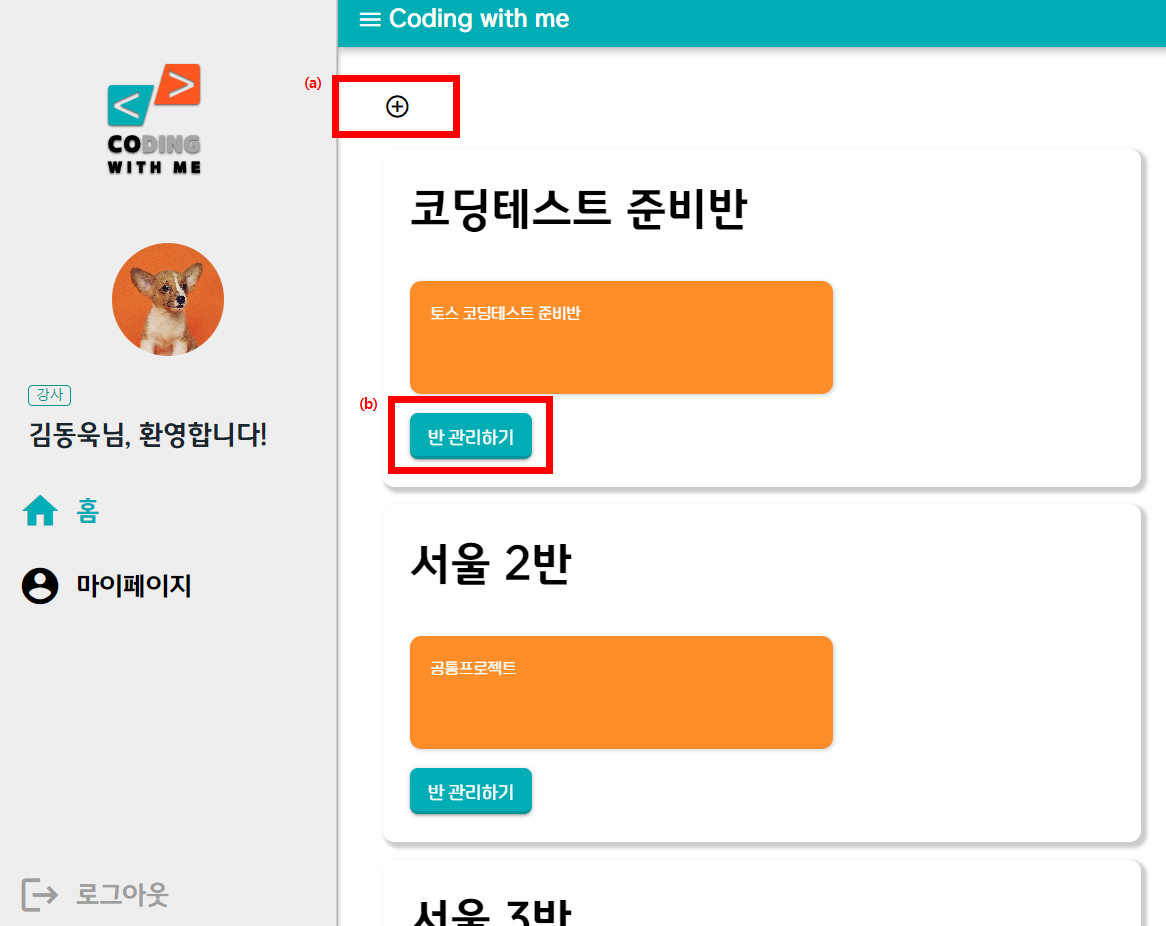
### 메인 페이지

- 강사는 여러 개의 반을 관리할 수 있고, 학생은 한 곳의 반에만 소속될 수 있다

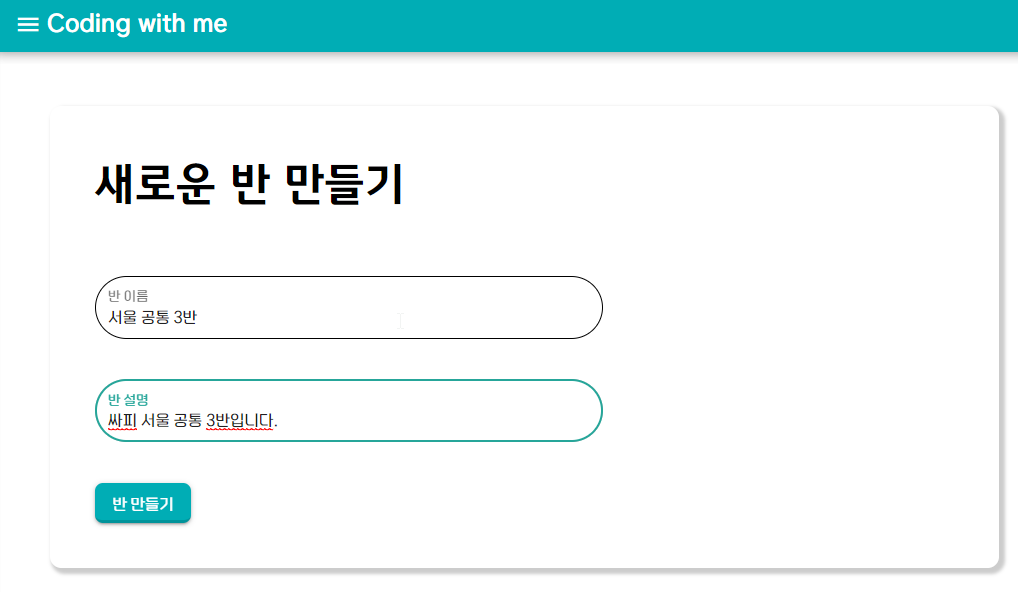
- 강사가 반을 만들고, 학생들을 추가한다.

(a) : 새로운 반을 개설한다.

(b) : 해당 반 페이지로 이동



### 반 개설 페이지



### 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..)



### 시험 등록 페이지

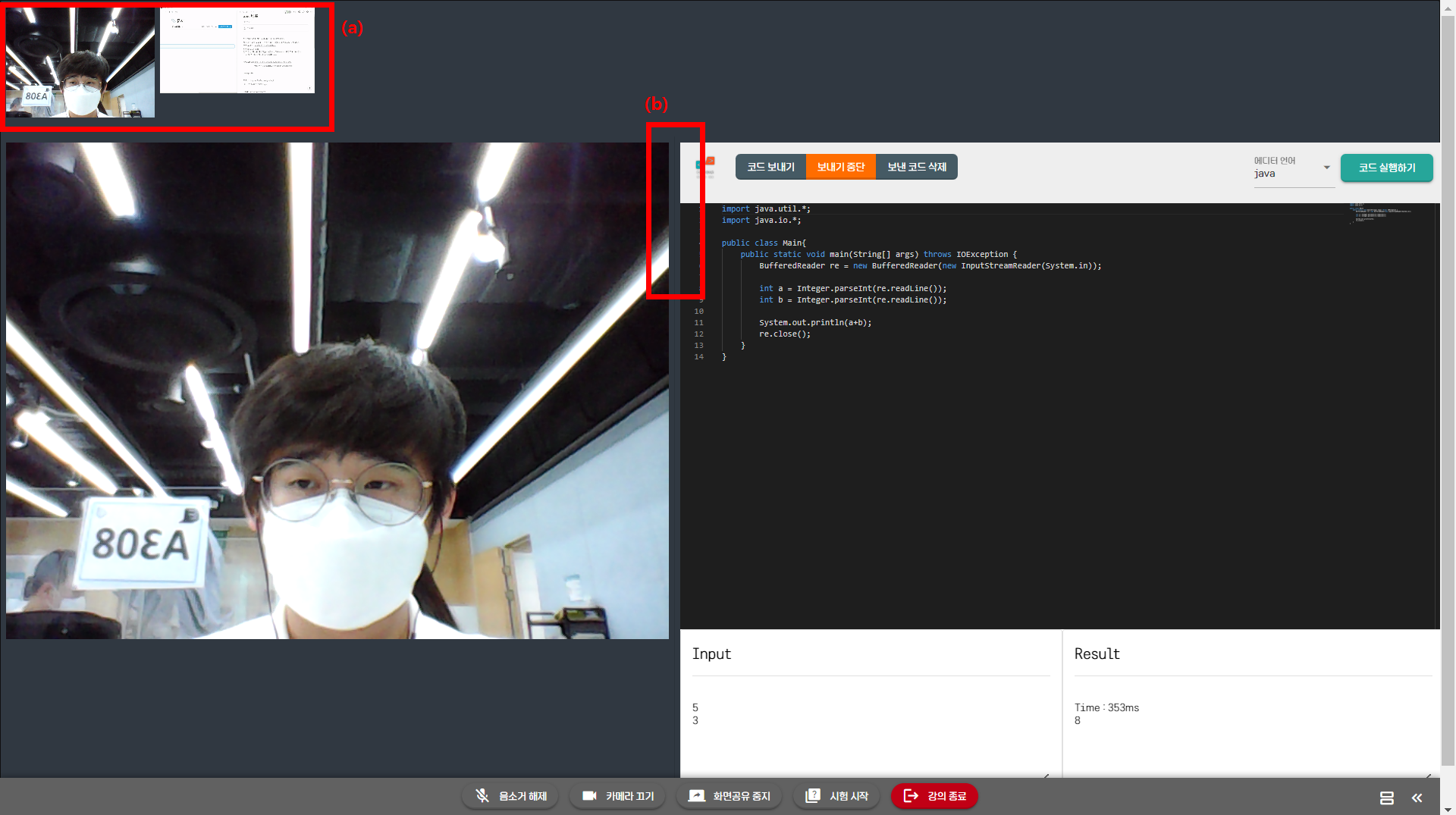
* 왼쪽에 시험 이름과 문제 개수를 지정할 수 있고, 각 문제 별로 여러 개의 테스트 케이스를 등록할 수 있다.
* 테스트 케이스 형식은 SW Expert Academy와 동일하게 Input과 Output으로 되어 있으며, 작성한 코드의 실행결과 Output이랑 동일하면 정답으로 처리된다.



### 강의 페이지

### 영상, 사이즈 조절

1. : 영상을 큰 영상으로 표시할 수 있다.
2. : 좌우 드래그를 통해 너비를 조정할 수 있다.



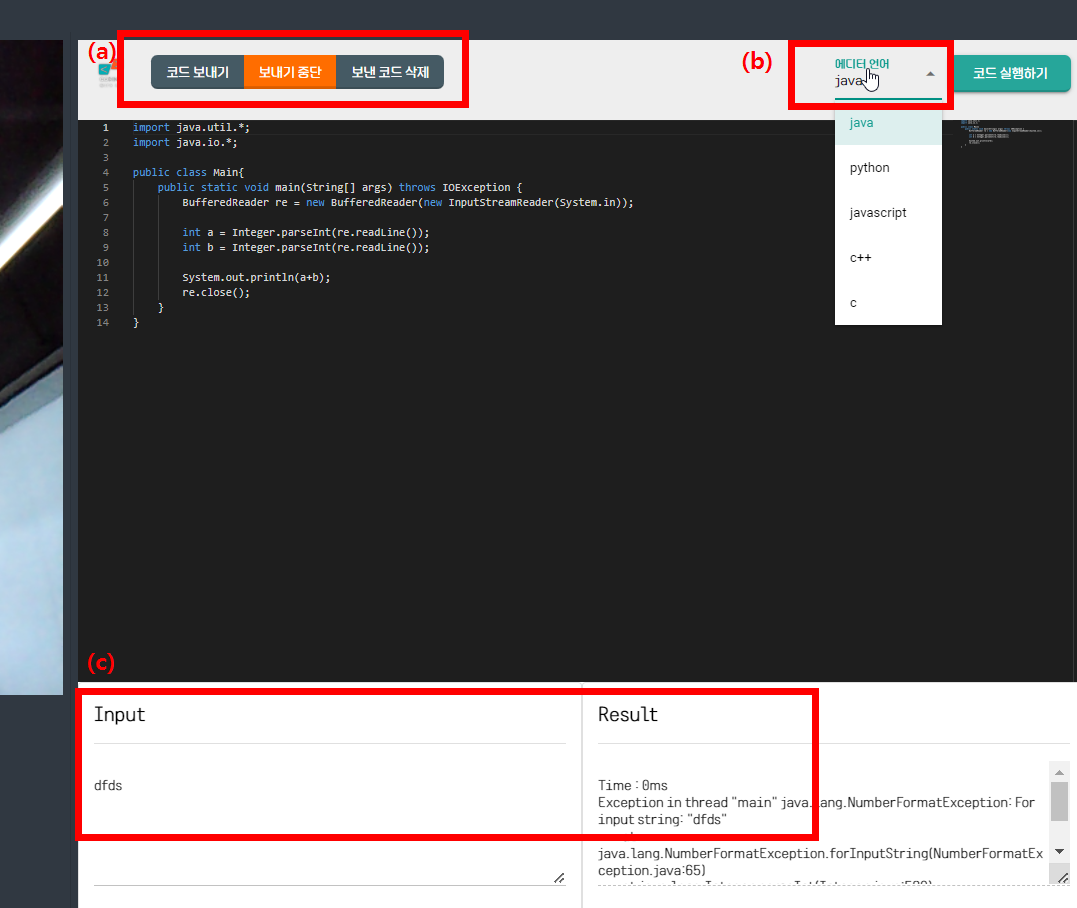
### 웹 IDE 기능

1. 강사가 작성한 코드를 실시간 동기화를 통해 학생에게 그대로 보여줄 수 있다.

코드 보내기: 실시간 동기화 on

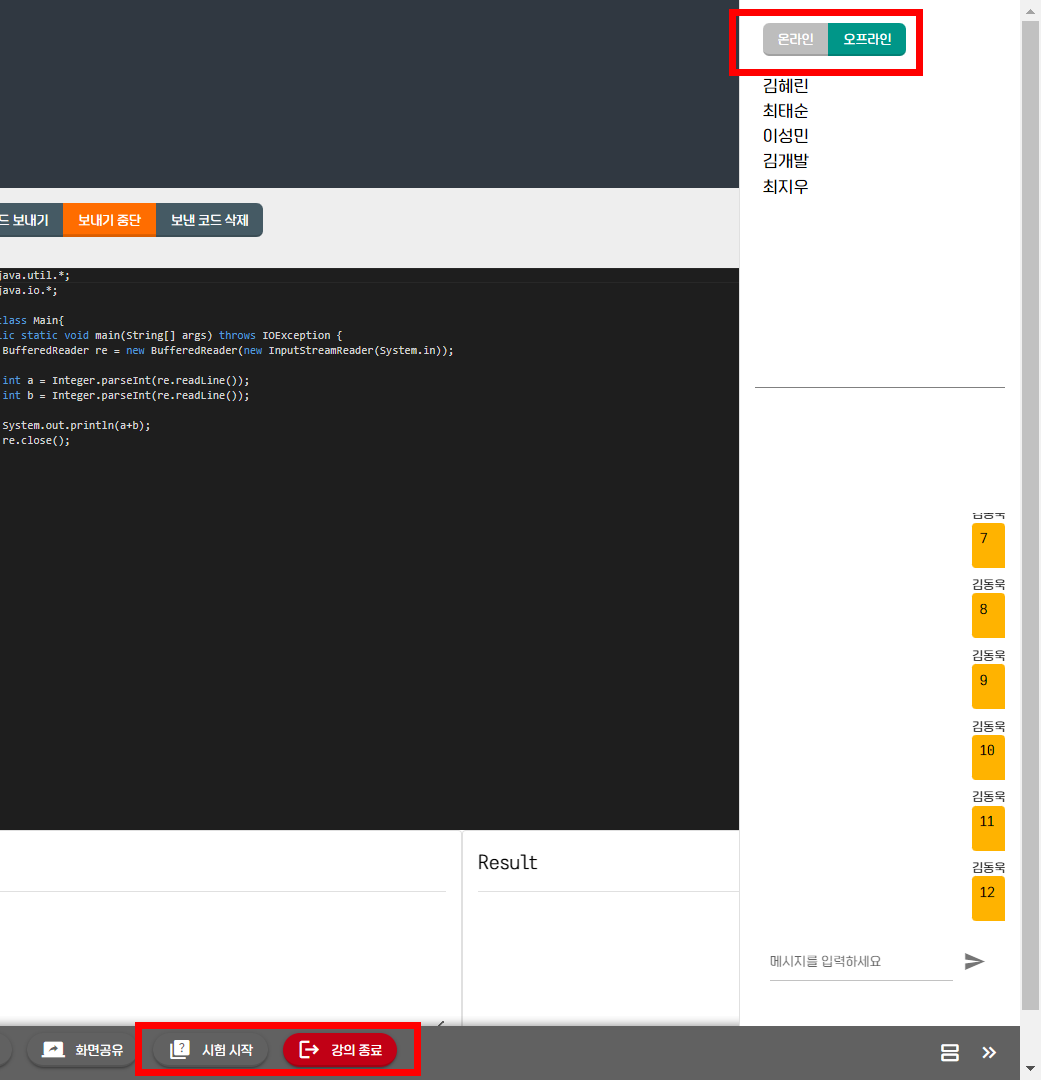
보내기 중단: 실시간 동기화 off, 학생은 마지막으로 받은 코드에서 그대로 유지됨.

1. 웹 IDE 언어 선택: 5가지 언어를 지원한다.
2. 코드 실행 결과 표시 : Input 값에 따른 실행 결과가 표시된다. (실행 시간/출력)이 표시되며, 컴파일 오류/런타임 오류/시간 초과 표시도 지원한다.



### 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, 강사과 학생이 자동으로 시험 페이지로 이동한다.



### 시험 페이지(시험 감독)

* 상단에 참가자들의 얼굴/화면이 표시되고, 화면 정중앙에는 문제지와 테스트 케이스가 표시된다.

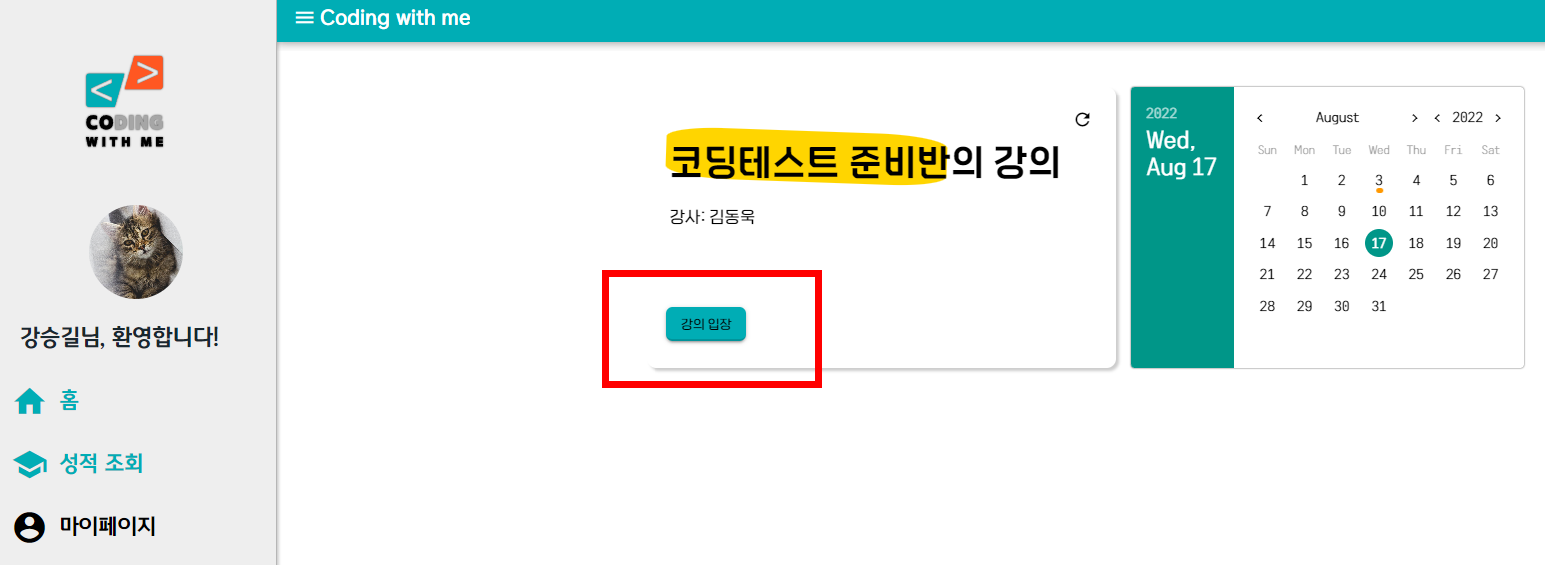
1. 남은 시간이 표시된다.
2. 온라인/오프라인 여부, 학생 제출/미제출 여부가 표시된다.
3. 시험 종료를 누르면 강사/학생이 강의 페이지로 이동되며, 나가기 버튼은 모든 참가자의 화상 회의가 종료된다.



# 시연 시나리오 – 학생

### 메인 페이지

* 강사가 개설한 강의가 있을 경우 입장 버튼이 표시된다.

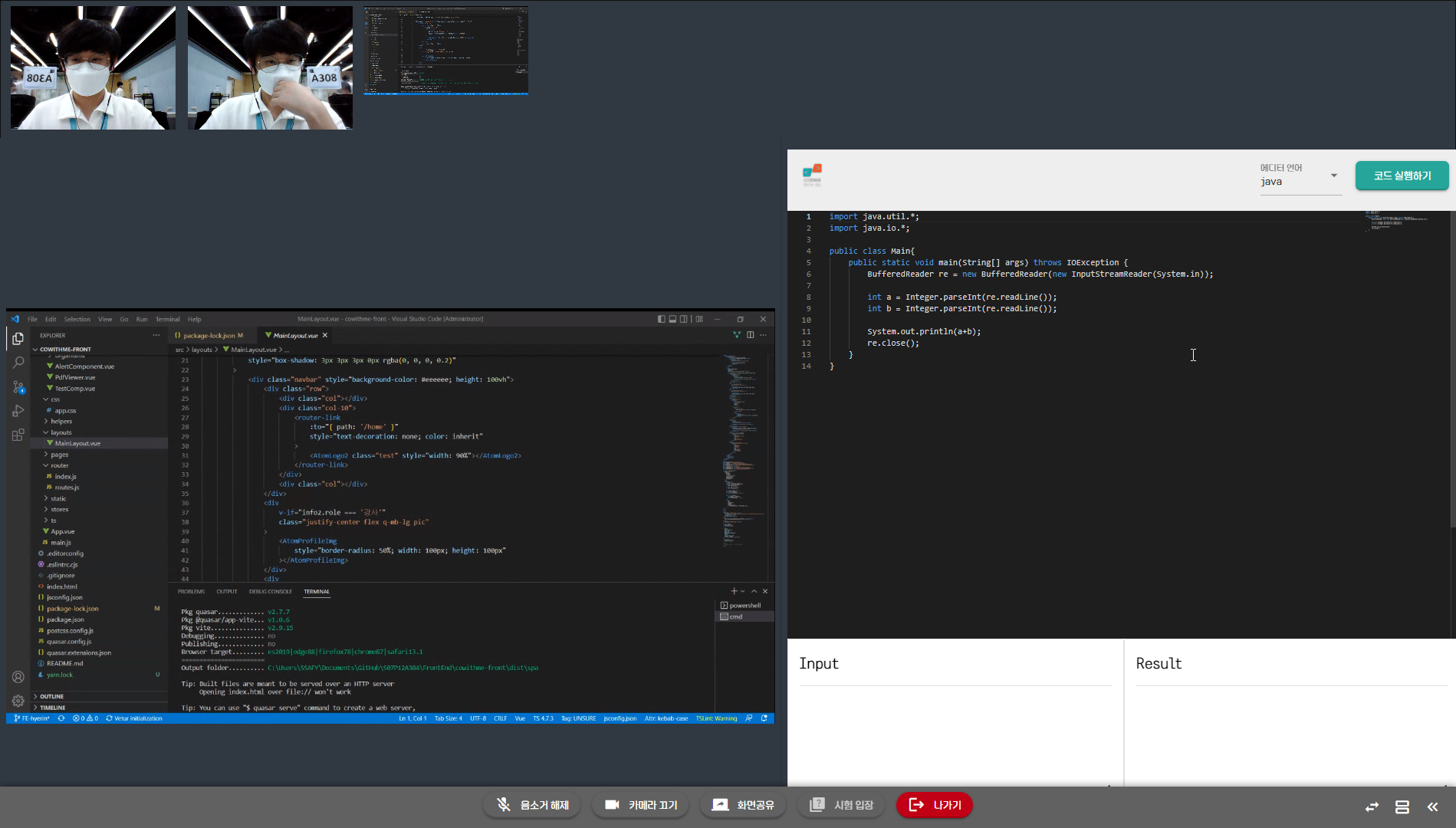


### 강의 페이지

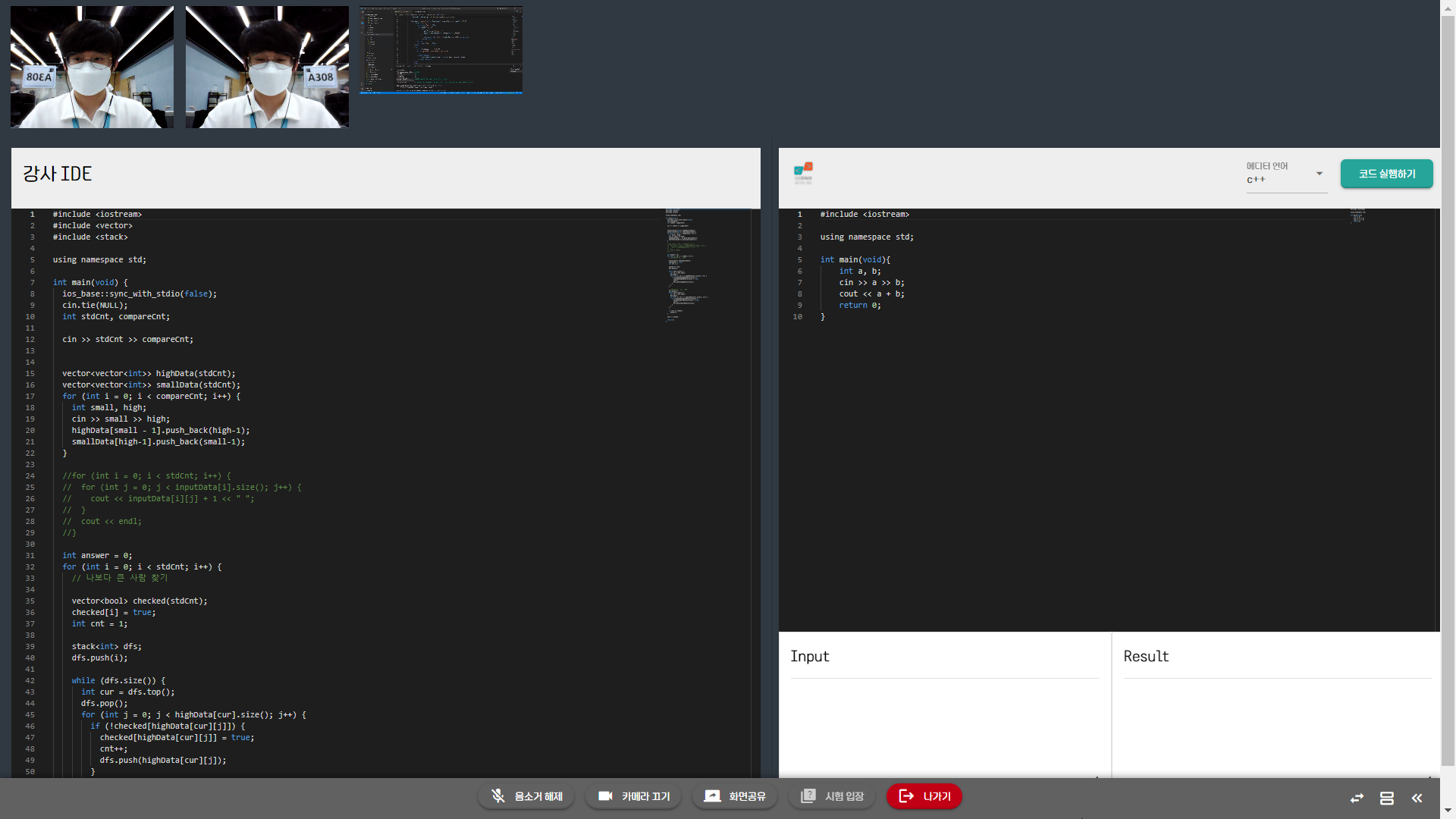
### 모드 변환

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

1. 모드 1



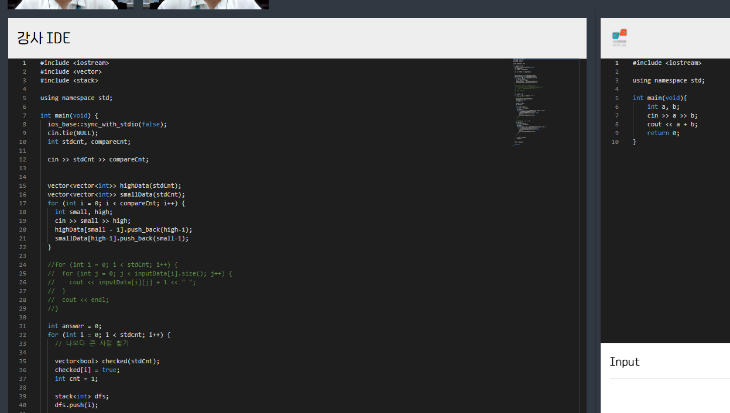
1. 모드 2



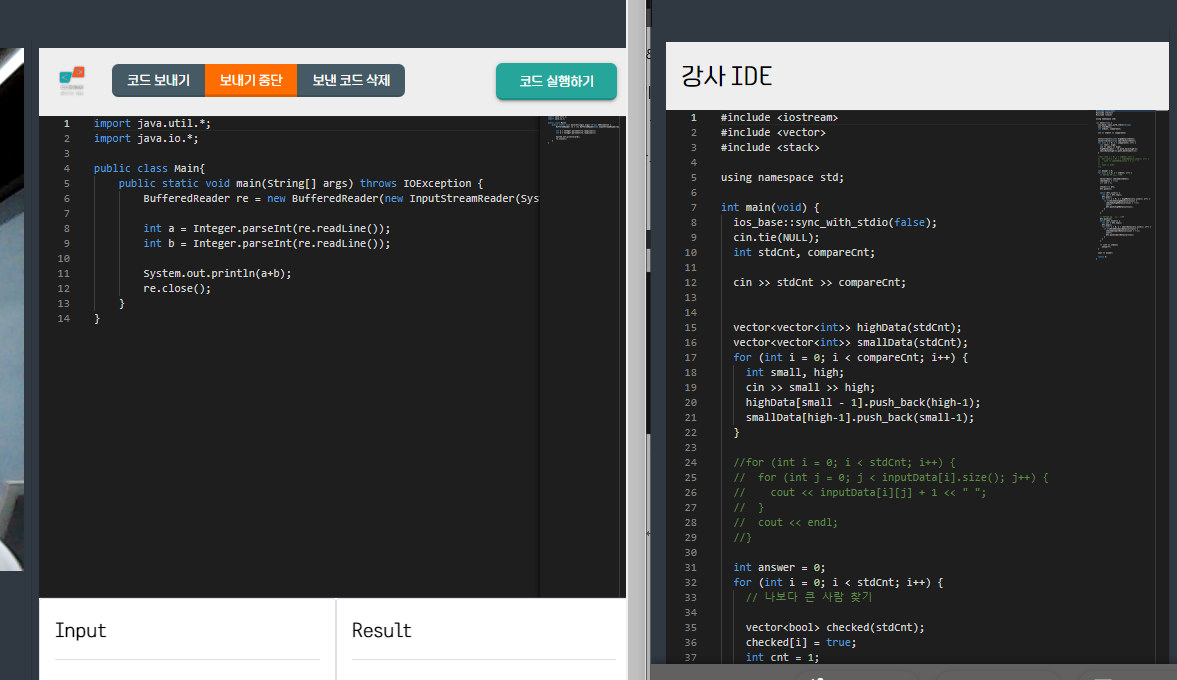
### 웹 IDE & 코드 동기화

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

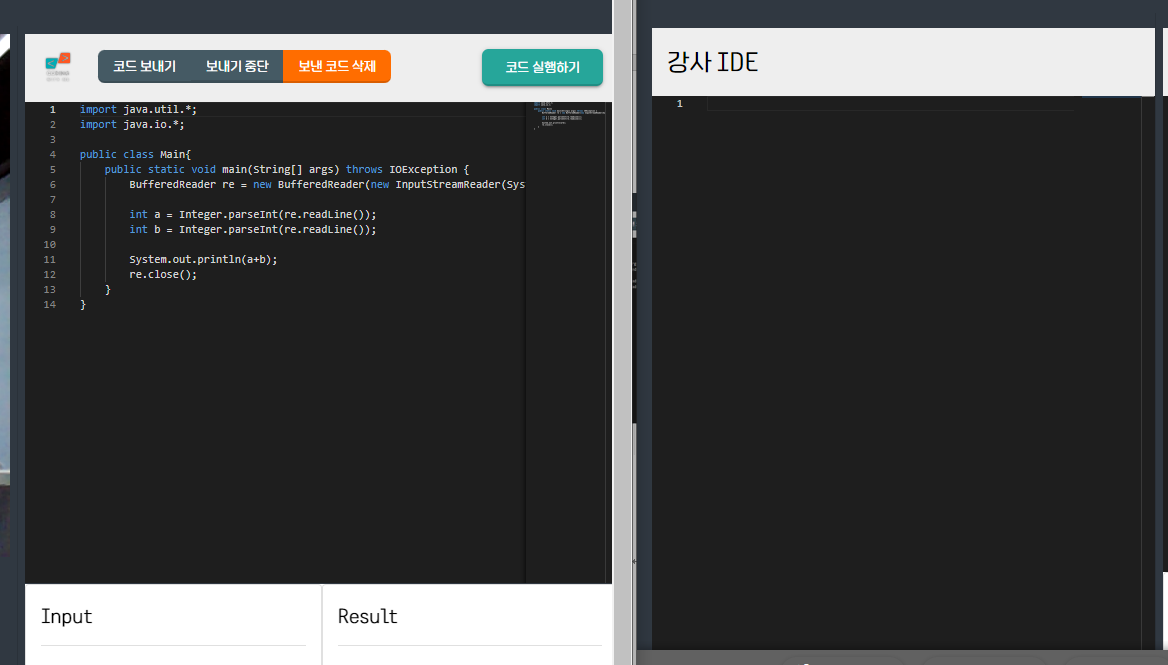




* 모드가 보내기 중단으로 되어 있을 때는 변경사항이 반영 되지 않는다.

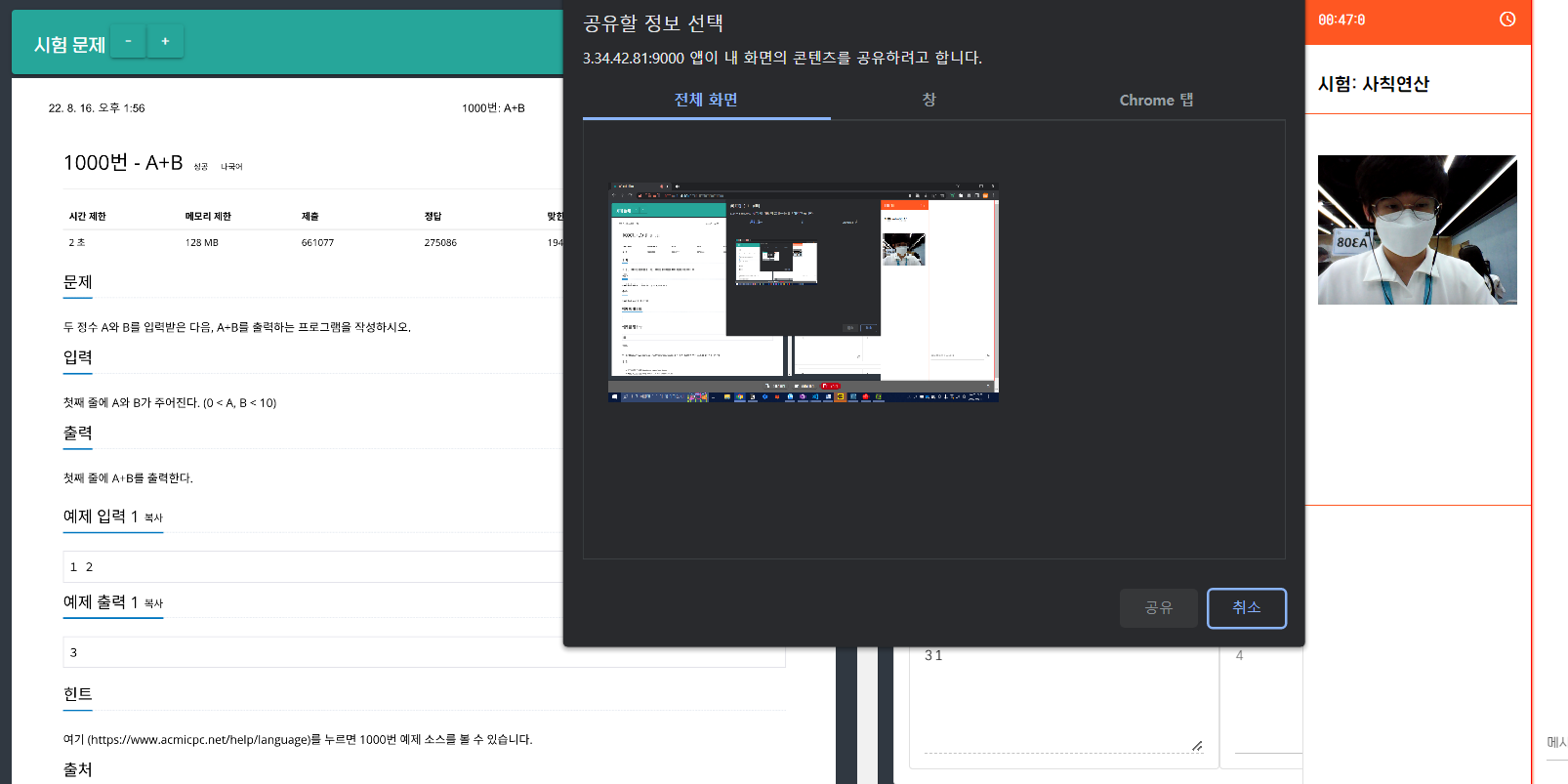


* 보낸 코드 삭제 모드의 경우 학생한테 IDE의 내용이 표시되지 않는다.

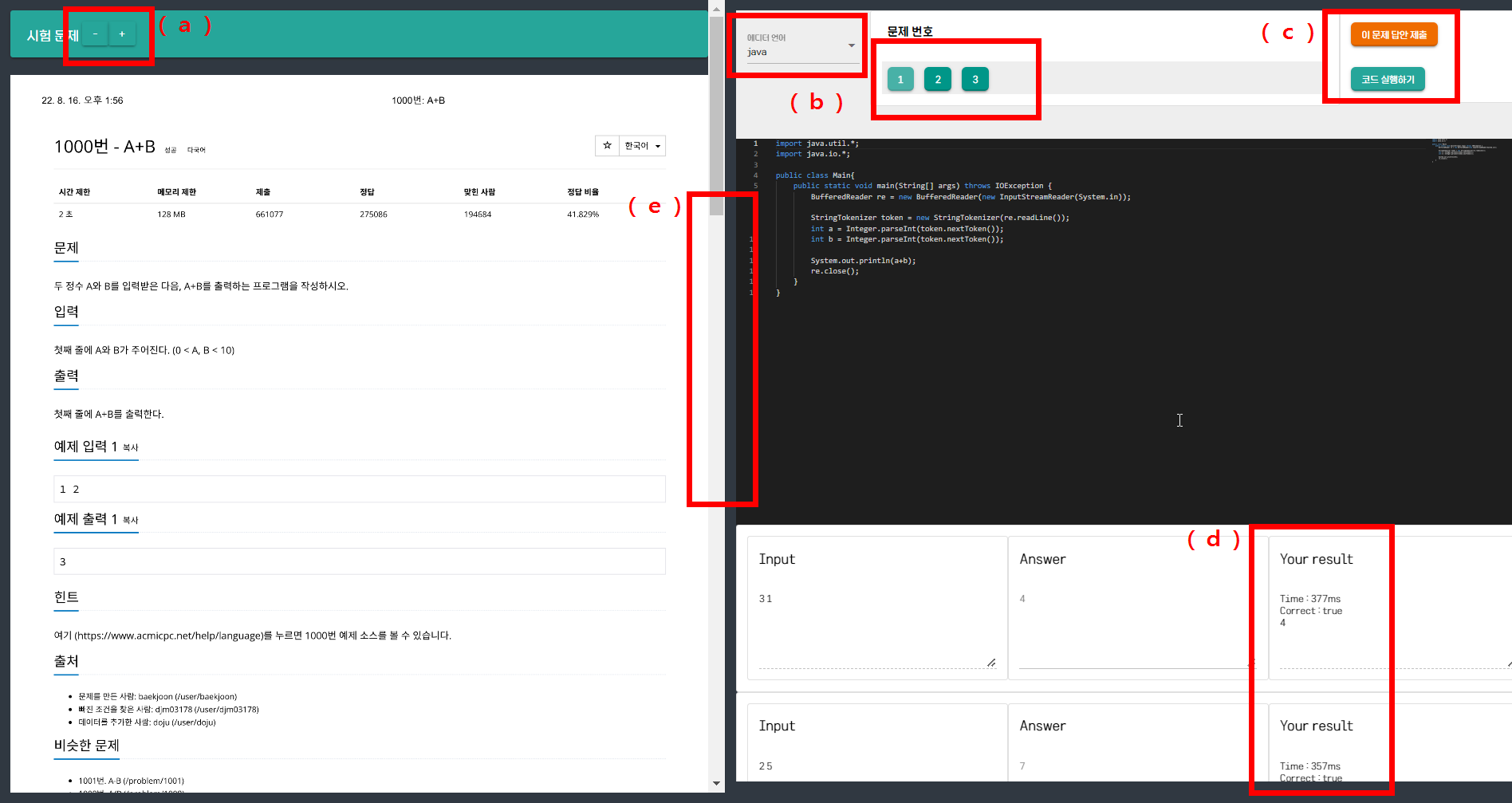


### 시험 페이지

* 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.



1. 시험지 확대/축소 기능
2. 시험에 사용할 언어 선택
3. 작성한 답안 제출 / 코드 실행 기능 => 제출하면 강사화면에서 제출 완료로 표시됨.
4. 각 테스트 케이스에 대해 실행 시간 / 결과 표시



### 시험 성적 보기

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

