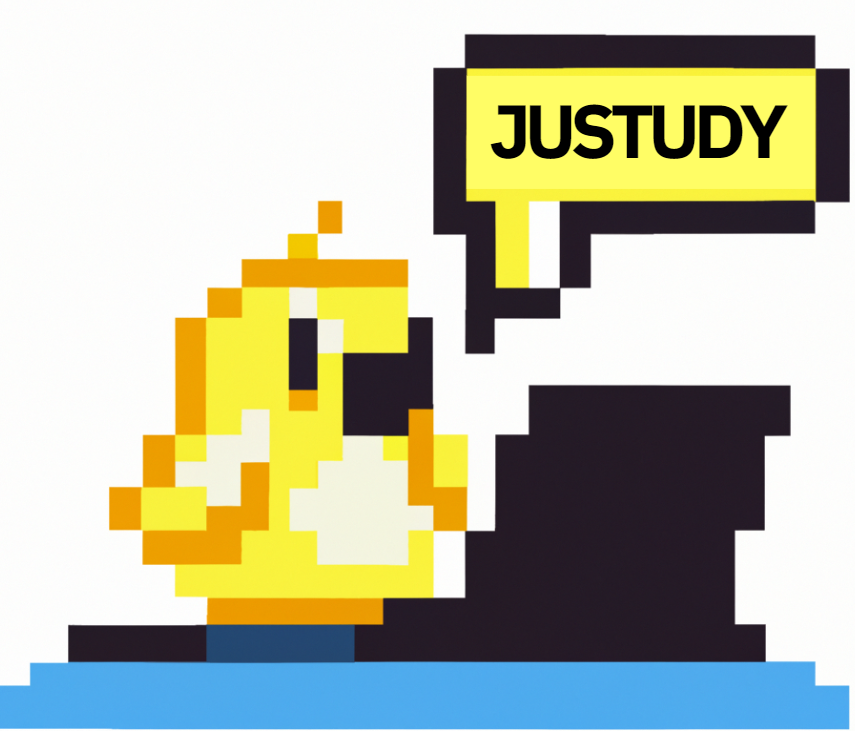
|  |
| --- |
|  |



|  |
| --- |
| **온라인 스터디 사이트** |
|  |
|  |
|  |
|  |
|  |
|  |

**텍스트, 클립아트, 식탁용기구이(가) 표시된 사진

자동 생성된 설명**

목차

[I. 개요 2](#_Toc127119121)

[1. 프로젝트 개요 2](#_Toc127119122)

[2. 프로젝트 사용 도구 2](#_Toc127119123)

[3. 개발환경 2](#_Toc127119124)

[II. 빌드 3](#_Toc127119125)

[1. 환경설정 3](#_Toc127119126)

[2. 라이브러리 설정 4](#_Toc127119127)

[3. 빌드하기 5](#_Toc127119128)

[4. 배포하기 6](#_Toc127119129)

[5. 서비스 아키텍처 8](#_Toc127119130)

[8](#_Toc127119131)

# 개요

## 프로젝트 개요

삼성 청년 SW 아카데미 교육생들을 위한 온라인 스터디 사이트.

프로그래머를 꿈꾸는 교육생들은 평범한 온라인 스터디로는 불편함을 많이 느꼈습니다.

좀 더 효율적으로 나와 비슷한 수준의 교육생들을 찾아 같은 주제를 공부하고 싶었습니다. 온라인상에서 만날 수 있으면서, 같은 코딩을 수정할 수도 있으며, 다양한 의견도 조율하고 싶었습니다. 그런 학생들을 위한 서비스 플랫폼을 구성하였습니다.

더 나아가 초심을 잃는 경우가 많기 때문에, 타이머 측정 기술을 넣어 선의의 경쟁과 자신의 공부량을 확인할 수 있습니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

CI/CD : Jenkins, docker

라이브러리 : Kurento 미디어 서버

## 개발환경

백엔드

* **Java** : azul-11 (java 11.0.17)
* **mysql** : 8.0.30
* **SpringBoot** : 2.7.7
* **IDE** : IntelliJ

프론트엔드

* **node JS** :18.12.1 LTS
* **IDE** : VSCode
  + Extension : ESLint, EditorConfig for VS Code, vscode-icons, Vue VSCode Snippets, Prettier-Code formatter, Vue Language Features (Volar)
* **Vue** 
  + vue 2.6.14, vue-router 3.6.5, vuex 3.6.2, axios 1.2.2

# 빌드

## 환경설정

백엔드(Spring) - Application.yml

|  |
| --- |
| **spring**:  **datasource**:  **driver-class-name**: com.mysql.cj.jdbc.Driver  **username**: 유저이름  **password**: 유저비밀번호  **url**: jdbc:mysql://db:3306/justudy?serverTimezone=UTC&characterEncoding=UTF-8  *# swagger*  **mvc**:  **pathmatch**:  **matching-strategy**: ant\_path\_matcher  **jpa**:  **database**: mysql  **database-platform**: org.hibernate.dialect.MySQL5InnoDBDialect *# 추가 해준 부분*  **hibernate**:  **ddl-auto**: update  **properties**:  **hibernate**:  **show\_sql**: true  **format\_sql**: true  *## page 인덱스를 1부터 시작하게 만듬*  **data**:  **web**:  **pageable**:  **one-indexed-parameters**: true  **logging**:  **level**:  **org.hibernate.sql**: info  **vue**:  **loginUrl**: "localhost:8081/login" |

## 라이브러리 설정

프론트(Vue) - Package.json

|  |
| --- |
| **"dependencies"**: {  **"axios"**: "^1.2.2",  **"chart.js"**: "^3.8.0",  **"core-js"**: "^3.8.3",  **"simple-code-editor"**: "^1.2.2",  **"vue"**: "^2.6.14",  **"vue-chartjs"**: "^4.0.4",  **"vue-router"**: "^3.6.5",  **"vuetify"**: "^2.6.0",  **"vuex"**: "^3.6.2",  **"vuex-persistedstate"**: "^4.1.0"  },  **"devDependencies"**: {  **"@babel/core"**: "^7.12.16",  **"@babel/eslint-parser"**: "^7.12.16",  **"@vue/cli-plugin-babel"**: "~5.0.0",  **"@vue/cli-plugin-eslint"**: "~5.0.0",  **"@vue/cli-plugin-router"**: "^5.0.8",  **"@vue/cli-plugin-vuex"**: "^5.0.8",  **"@vue/cli-service"**: "~5.0.0",  **"eslint"**: "^7.32.0",  **"eslint-plugin-vue"**: "^8.7.1",  **"sass"**: "~1.32.0",  **"sass-loader"**: "^10.0.0",  **"vue-cli-plugin-vuetify"**: "~2.5.8",  **"vue-template-compiler"**: "^2.6.14",  **"vuetify-loader"**: "^1.7.0"  }, |

백엔드(Spring) - build.gradle

|  |
| --- |
| dependencies {  implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  implementation 'org.springframework.boot:spring-boot-starter-security'  implementation 'org.springframework.boot:spring-boot-starter-validation'  implementation 'org.springframework.boot:spring-boot-starter-web'  implementation 'org.springframework.boot:spring-boot-starter-websocket'  *//kurento 설정*  implementation 'org.webjars.bower:kurento-utils:6.7.0'  implementation 'org.kurento:kurento-client:6.18.0'  *//queryDsl 설정*  implementation 'com.querydsl:querydsl-core'  implementation 'com.querydsl:querydsl-jpa'  testImplementation 'org.projectlombok:lombok:1.18.22'  annotationProcessor "com.querydsl:querydsl-apt:${dependencyManagement.importedProperties['querydsl.version']}:jpa" *// querydsl JPAAnnotationProcessor 사용 지정*  annotationProcessor "jakarta.persistence:jakarta.persistence-api" *// java.lang.NoClassDefFoundError(javax.annotation.Entity) 발생 대응*  annotationProcessor "jakarta.annotation:jakarta.annotation-api" *// java.lang.NoClassDefFoundError (javax.annotation.Generated) 발생 대응*  *//swagger 설정*  implementation group: 'io.springfox', name: 'springfox-swagger-ui', version: '2.9.2'  implementation group: 'io.springfox', name: 'springfox-swagger2', version: '2.9.2'  compileOnly 'org.projectlombok:lombok'  developmentOnly 'org.springframework.boot:spring-boot-devtools'  runtimeOnly 'com.h2database:h2'  runtimeOnly 'com.mysql:mysql-connector-j'  annotationProcessor 'org.projectlombok:lombok'  testImplementation 'org.springframework.boot:spring-boot-starter-test'  } |

## 빌드하기

docker-compose.yml

|  |
| --- |
| **version**: "3"  **services**:  **back**:  **container\_name**: be  **build**: ./back  **ports**:  - "8080:8080"  **depends\_on**:  - db  **restart**: on-failure  **front**:  **container\_name**: fe  **build**: ./front  **ports**:  - "8083:5000"  **restart**: on-failure  **db**:  **container\_name**: db  **image**: mysql:8.0.30  **ports**:  - "3306:3306"  **volumes**:  - /mysql:/var/lib/mysql  - /mysql/config/my.cnf:/etc/mysql/my.cnf  **environment**:  **MYSQL\_ROOT\_PASSWORD**: 비밀번호  **restart**: on-failure  **kurento**:  **container\_name**: kurento  **image**: kurento/kurento-media-server:latest  **ports**:  - "8888:8888"  **environment**:  **KMS\_STUN\_IP**: "3.38.116.236"  **KMS\_STUN\_PORT**: "3478"  **KMS\_TURN\_URL**: "a104t:ssafy@3.38.116.236:3478?transport=udp"  **restart**: on-failure |

## 배포하기

Nginx 설정

|  |
| --- |
| worker\_processes 1;  events {  worker\_connections 1024;  }  http {  include /etc/nginx/mime.types;  default\_type application/octet-stream;  upstream front-server {  server 127.0.0.1:8083*;*  keepalive 100;  }  upstream front-server-dev{  server 127.0.0.1:8082*;*  keepalive 100;  }  upstream backend-server {  server 127.0.0.1:8080*;*  keepalive 100;  }  upstream db-server {  server 127.0.0.1:3306*;*  keepalive 100;  }  server {  if ($host = i8a104.p.ssafy.io) {  return 301 https://$host$request\_uri*;*  } # managed by Certbot  server\_name i8a104.p.ssafy.io;  return 404; # managed by Certbot  }  server {  server\_name i8a104.p.ssafy.io;  location / {  proxy\_pass http://front-server*;*  }  location /test/ {  proxy\_pass http://front-server-dev/*;*  }  location /api {  proxy\_pass http://backend-server*;*  proxy\_set\_header HOST $host;  proxy\_set\_header Referer $http\_referer;  }  location /groupcall{  proxy\_pass http://backend-server*;*  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection "upgrade";  proxy\_set\_header Host $host;  proxy\_read\_timeout 21600000; # 6 \* 60 \* 60 \* 1000  proxy\_send\_timeout 21600000; # 6 \* 60 \* 60 \* 1000  }  error\_page 500 502 503 504 /50x.html;  location = 50x.html {  root /usr/share/nginx/html;  }  listen 443 ssl; # managed by Certbot  ssl\_certificate /etc/letsencrypt/live//i8a104.p.ssafy.io/fullchain.pem; # managed by Cert>  ssl\_certificate\_key /etc/letsencrypt/live/i8a104.p.ssafy.io/privkey.pem; # managed by Ce>  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  }  } |

## 서비스 아키텍처

|  |
| --- |
|  |