**1. 사용한 JVM, 웹서버, WAS제품 등의 종류와 설정값, 버전(IDE 버전 포함) 기재**

1. 프론트엔드 : JavaScript, React 17.0.2 Next.js 12.0.8, chart.js: 3.7.0 react-redux 7.2.6, next-redux-wrapper 7.0.5, redux-persist 6.0.0 HTML5, CSS,
2. 백엔드 : Node Js 14.18.3, Express 4.17.2 3
3. 데이터베이스 : MySQL 8.0.22.0 4, 운영체제
4. 서버 : Window, EC2(Ubuntu), Docker 20.10.7, Nginx 1.20.2
5. 하드웨어: Arduino, Arduino Joystick Shield, HC-06
6. Ide : Visual Studio Code 1.64.2

**2. 빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재**

FE폴더의 .env 파일에 NEXT\_PUBLIC\_SERVER=http://i6a205.p.ssafy.io:8000 작성 확인

(위 주소는, 빌드 시 사용되는 서버 주소 변경에 따라 변경 가능)

BE폴더의 utils/mysql.js 파일에 mysql host 주소 확인 (사용하는 환경에 따라 주소 변경)

**3. 배포 시 특이사항 기재**

**3-1. Docker 를 이용하여 배포 테스트 시 실행 순서**

a) docker 및 docker-compose 다운로드

b) 백엔드 서버 => BE 디렉터리에서 `docker-compose up` 명령어를 통해 실행

c) 프론트엔드 서버 => FE 디렉터리에서 `docker-compose up` 명령어를 통해 실행

**3-2. 로컬에서 테스트 시 실행순서**

- 백엔드 서버

a) BE/utils/mysql.js 에서 로컬에서 사용할 mysql host 주소 및 user, database 정보 설정

b) mysql서버에 접속 후 위에서 설정한 user 정보 및 database 생성

c) BE/db/init.sql 에 작성된 DML 로 테이블 생성

d) BE 디렉터리에서 npm i 명령어로 설치 후 node server.js 명령어로 실행

- 프론트엔드 서버

a) FE 디렉터리에서 npm i 명령어로 라이브러리 설치

b) npm run build 명령어로 빌드파일 생성

c) npm run start 로 빌드파일 실행

\*개발 모드로 실행 시엔 npm run dev 명령어로 실행

**4. DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록**

Docker 를 통해 배포된 mysql database 에 접속할 경우 아래 정보 활용.

Root\_password: ssafy1!

user : a205

password : ssafy1!

접속포트 : 3306