1조: 3rd Project (Spring Framework)



팀원: 한동길(조장) | 김승룡 | 문호정 | 이건민 | 김경빈

2025.02.28.

# Category

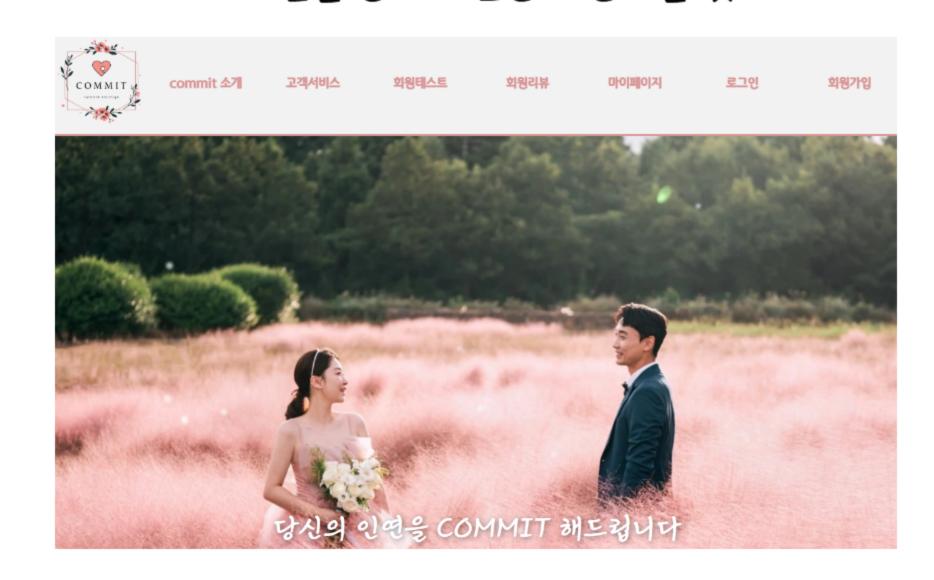
- 1. 프로그램 소개
- 2. ERD
- 3. 주요기능
- 4. 간단한 코딩 소개
- 5. 시연
- 6. 어려웠던 사항들 (느낀점), QNA



# 1. 프로그램 소개



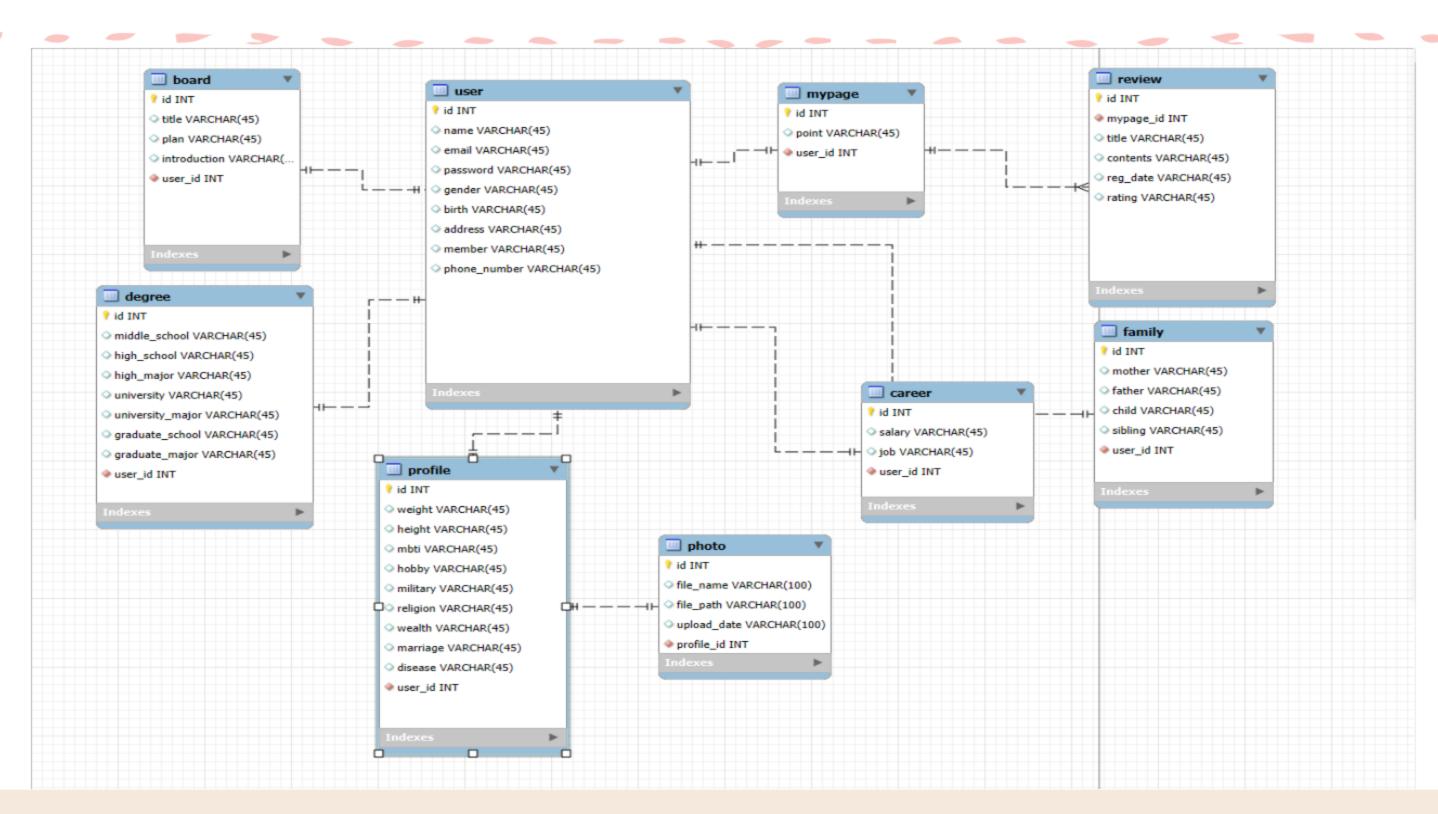
저출산 시대! 결혼이 힘들고 아무나 만나고 싶지 않은 여러분을 위해 만들었습니다. 저희 프로그램을 통해서 운명의 상대를 찾으세요!





### 2. ERD





## 3. 주요기능



- 로그인 & 회원가입
- 회원 기능: 마이페이지 관리, 리뷰 작성
- 비회원 기능 : 연애 테스트
- 관리자 기능: 고객 정보 관리

## 4.간단한 코딩 소개



```
1 # 1.베이스 이미지 선정
                                               2 FROM eclipse-temurin:17-jdk
> 🕌 > CommitSpring [boot] [devtools] [CommitSpring main]
 CommitSpringApi
                                               4 # 2.작업 디렉터리 생성

→ 

CommitSpringpro [boot] [devtools]

                                               5 WORKDIR /app
 > 🎛 com.commit.controller
                                               7 # 3.jsp 파일을 컨테이너로 복사
                                               8 ARG JAR_FILE=target/commitSpringPro.jar
   > ## com.commit.mappers
                                               9 COPY ${JAR_FILE} .
   > # com.commit.repository.vo
   > # com.commit.restserver
   > 🔠 com.commit.service
                                               12 ENV SPRING_DATASOURCE_URL=jdbc:mysql://localhost:3306/mydb
  > 🌁 src/main/resources
                                               13
                                               14 EXPOSE 8927
  > 乃 src/test/java
                                               15
  > M JRE System Library [JavaSE-17]
                                               16 # 6.컨테이너가 실행될 때 JAR를 실행
  > Maven Dependencies
                                               17 ENTRYPOINT ["java", "-jar", "/app/commitSpringPro.jar"]
  > 🕮 target/generated-sources/annotations
  > # target/generated-test-sources/test-annotations
                                               19⊖ # build
                                               20 # docker build -t myrestserver .
 > 🗁 src
  target
                                              22⊖ # run
   > Begenerated-sources
                                               23 # mysql-container 먼저 실행
   > Begenerated-test-sources
                                               24 # docker run -d -p 18088:8088 --name myrestserver -e SPRING_I
   > > maven-archiver
                                              25 # docker run -d -p 18090:8927 commitspring
   > > maven-status
      commitSpringPro.jar
      commitSpringPro.jar.original
    Dockerfile
    M HELP.md
    mvnw
    mvnw.cmd
   lmx.moq 🔬
```

```
COMMITREACT
                                         FROM node:22 AS build

    Admin_components
    AdminDashboard.jsx

                                     4 WORKDIR /app
 AdminFilter.jsx
 AdminUserList.jsx
 AdminUserModal.jsx
                                         COPY package*.json .
 career_components
                                     8 RUN npm install
 CareerButton.jsx
 CareerContent.jsx
 ContentBox.jsx
 NoInfo.jsx
 > company_components
 components
                                    15 RUN npm run build
 Footer.jsx
 # Header.jsx
                                    18 FROM nginx:alpine
 TopBar.jsx
                                    20 COPY --from-build /app/dist /usr/share/nginx/html
 > family_components
                                          EXPOSE 80
 > main_components
                                    24 ENTRYPOINT ["nginx", "-g", "daemon off;"]
 > modal_components
 > mypage_components
 > pages

→ profile_components

 ContentBox.jsx
 Molnfo.jsx
 ProfileButton.jsx
 ProfileContent.jsx
 > review_components
 > signin_components
 > signup_components
 TestMain.jsx
 🗱 TestPages.jsx
 TestQuestion.jsx
                                   문제 🔼 출력 디버그 콘출 터미널 포트
 TestResult.jsx
xzi.qqA 🎡
                                     VITE v6.2.0 ready in 305 ms
main.jsx
                                      → Local: http://localhost:5173/
                                     → Network: use --host to expose
{} .package-lock.json
                                      → press h + enter to show help
```

## 4.간단한 코딩 소개



```
compose > # compose.yml
O .emv

    oitattributes

                                             - my-network
                                          container name: spring-commit-container
                                            SPRING DATASOURCE USERNAME: ${MYSQL USER}
                                           SPRING_DATASOURCE_PASSWORD: ${MYSQL_PASSWORD}
                                          depends_on:
- db
                                           MYSQL_DATABASE: ${MYSQL_DATABASE}
                                         external: true # 기존 볼륨을 사용할 경우 true
                                        external: true # 기존 네트워크에 완류
```

```
✓ COMMITMYSQL
                                   Dockerfile
.env
                                       FROM mysql:latest
$ .env.template
Dockerfile
                                    4 # 초기화 SQL
mydb_board.sql
                                    5 # /docker-entrypoint-initdb.d
mydb_career.sql
mydb_degree.sql
                                    8 COPY mydb_board.sql /docker-entrypoint-initdb.d/
mydb_family.sql
                                    9 COPY mydb_career.sql /docker-entrypoint-initdb.d/
mydb_grant.sql
                                   10 COPY mydb_degree.sql /docker-entrypoint-initdb.d/
mydb_mypage.sql
                                   11 COPY mydb_family.sql /docker-entrypoint-initdb.d/
mydb_photo.sql
                                    12 COPY mydb_mypage.sql /docker-entrypoint-initdb.d/
mydb_profile.sql
                                    13 COPY mydb_photo.sql /docker-entrypoint-initdb.d/
mydb_review.sql
                                   14 COPY mydb_profile.sql /docker-entrypoint-initdb.d/
mydb_routines.sql
                                   15 COPY mydb_review.sql /docker-entrypoint-initdb.d/
mydb_user.sql
                                   17 COPY mydb_user.sql /docker-entrypoint-initdb.d/
                                   18 COPY mydb_grant.sql /docker-entrypoint-initdb.d/
                                   22 ENV MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
                                   23 ENV MYSQL_DATABASE=${MYSQL_DATABASE}
                                   24 ENV MYSQL_USER=${MYSQL_USER}
                                   25 ENV MYSQL PASSWORD=${MYSQL PASSWORD}
                                   문제 출력 디버그 콘슬 터미널 포트
                                     UndefinedVar: Usage of undefined variable '$MYSQL_USER' (line 24)

    UndefinedVar: Usage of undefined variable '$MYSQL_PASSWORD' (line 25

    SecretsUsedInArgOrEnv: Do not use ARG or EW instructions for sensit;

                                  C:\Users\202-26\Documents\GitHub\commitMysql>docker run --name mysql-com
                                   314932bbcb201fc0fc7c7e8ba7523decd55dc0cc7e8cee7c81fa855690c4634c
                                  C:\Users\202-26\Documents\GitHub\commitMysql>docker run --name mysql-com
```

# 5. 시연



村村建筑村村

## 6. 어려웠던 사항들 (느낀점), QNA



#### 호정 -

Spring, DB, React 를 모두 사용해서 만들어봤더니 저에게 맞는 방향을 찾을 수 있게 됐습니다. 개발자로서 성공하기 위해서 지금의 수준보다 더 레벨 업! 하면 좋을 것 같다는 생각이 들었습니다. 시중에 있는 홈페이지를 만든 개발자들이 참 대단하다는 생각이 들었습니다.

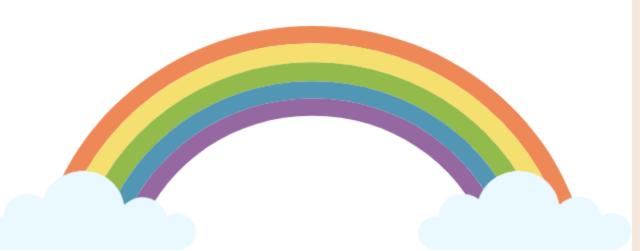
#### 건민 -

이번 프로젝트에 참여하면서 스스로의 부족한 점을 많이 느꼈지만, 좌절할 때마다 조원들의 응원과 협력이 큰 힘이 되었습니다 함께 노력해 준 팀원들에게 감사의 마음을 전하며, 이번 경험을 바탕으로 더욱 발전해 나가고 싶습니다.

#### 경빈 -

프로젝트에 참여하지 못 하고

기초부터 공부하면서 팀원들의 코드를 보며 많은 공부와 대단함을 느꼈습니다 다음 프로젝트에서는 간단한 부분이라도 꼭 참여 할 수 있도록 노력해야겠다는 생각이 참 많이 들었습니다.



## 6. 어려웠던 사항들 (느낀점), QNA



#### 승룡 -

알고있다고 생각했던 부분들이 사실은 아는게 아니었습니다. 촉박한 시간속에서 열심히 했지만, 한계도 명확히 느낀 시간이었습니다. 팀원들은 매번 너무 좋아서 즐거웠지만, 개발은 쉽지않았습니다.

#### 동길 -

프론트랑 백엔드 연결시키는 작업중에 Jsession의 쿠기 허용이 에러 발생의 요인 -> credentials 옵션을 설정해줘야함 ?옵션을 사용해서 에러 발생을 방지 할 수 있음, 이미지 파일을 post할 때 변경되는 데이터형식을 알 수 있었음 컴포넌트의 구조의 중요성에 대해 다시 한번 알게되었음



# ーなかむにトー