회원가입 기능 구현

Config

Cors

```
package com.project.ssafy.config;

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.Cors
import org.springframework.web.servlet.config.annotation.WebM

@Configuration
public class WebMvcConfig implements WebMvcConfigurer {
    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
                .allowedOrigins("*")
                .allowedMethods("*")
                .allowedHeaders("*");
    }
}
```

Security

```
package com.project.ssafy.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.bui
import org.springframework.security.config.annotation.web.con
import org.springframework.security.crypto.bcrypt.BCryptPassw
import org.springframework.security.web.SecurityFilterChain;

@EnableWebSecurity
```

```
@Configuration
public class SecurityConfig {

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecuri
        http
                .csrf(csrf -> csrf.disable())
                .authorizeHttpRequests(authorizeRequests ->
                        authorizeRequests.requestMatchers("/*
                                .anyRequest().authenticated()
                );
        return http.build();
    }

}
```

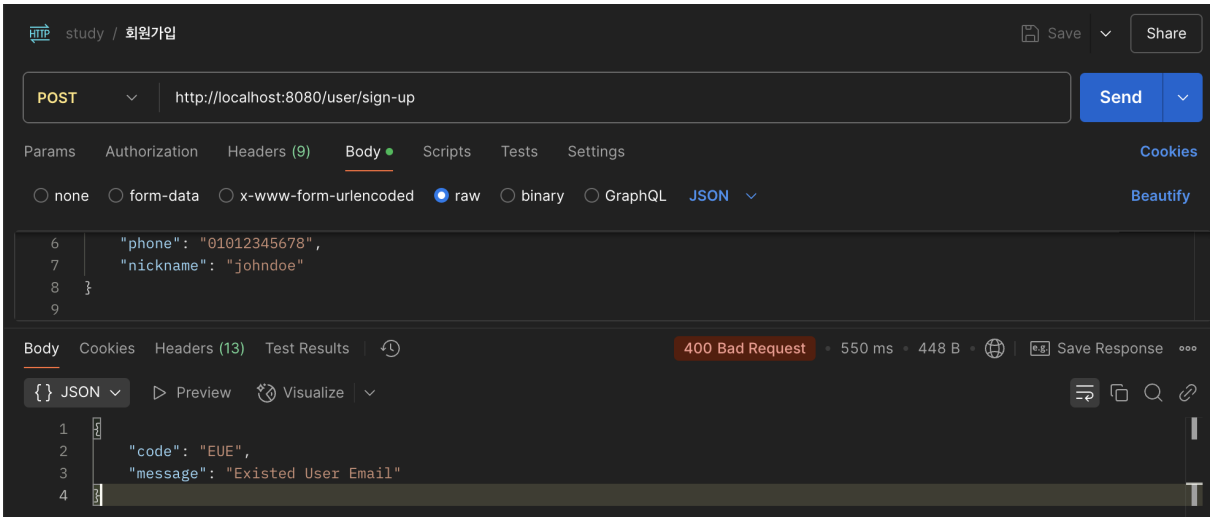
Controller

```
package com.project.ssafy.controller;

import com.project.ssafy.dto.user.request.SignUpRequestDto;
import com.project.ssafy.service.UserService;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping
import org.springframework.web.bind.annotation.RestController

@RestController
@RequestMapping("/user")
public class UserController {

    // 일반 사용자 회원가입
```

```
    @PostMapping("/sign-up")
    public ResponseEntity<?> signUp(@RequestBody SignUpReques
        ResponseEntity<?> response = userService.signUp(signU
        return response;
    }

}
```

공통 Response

> 프론트(모바일) 개발 팀원들간 api 연결 시 에러 메세지를 해석해서 문제점을 찾는 것이 아니라, 백엔드 측에서 발생될만한 에러들을 미리 바로 보기 쉽게 알려주기 위한 코드 Set



Response Code

```
package com.project.ssafy.common;

public interface ResponseCode {
    String SUCCESS = "SU";
    String EXISTED_USER_EMAIL = "EUE";
    String NO_EXISTED_USER_EMAIL = "NEUE";
    String EXISTED_USER_NICKNAME = "EUN";
    String NO_EXISTED_USER_NICKNAME = "NEUN";
```

```
    String EXISTED_USER_PHONE = "EUP";
}
```

Response Message

```
package com.project.ssafy.common;

public interface ResponseMessage {
    String SUCCESS = "Success";
    String EXISTED_USER_EMAIL = "Existed User Email";
    String NO_EXISTED_USER_EMAIL = "No Existed User Email";
    String EXISTED_USER_NICKNAME = "Existed User Nickname";
    String NO_EXISTED_USER_NICKNAME = "No Existed User Nickna
    String EXISTED_USER_PHONE = "Existed User Phone";
}
```

Request Dto

```
package com.project.ssafy.dto.user.request;

import com.project.ssafy.entity.User;
import lombok.Data;
import org.springframework.security.crypto.bcrypt.BCryptPassw

import java.time.LocalDateTime;

@Data
public class SignUpRequestDto {
    private String email;
    private String password;
    private String name;
    private Character gender;
    private String phone;
    private String nickname;

    public User toUserEntity(BCryptPasswordEncoder passwordEn
```

```
        return User
                .builder()
                .email(email)
                .password(passwordEncoder.encode(password))
                .name(name)
                .gender(gender)
                .phone(phone)
                .nickname(nickname)
                .createDate(LocalDateTime.now())
                .updateDate(LocalDateTime.now())
                .build();
    }
}
```

Response Dto

```
package com.project.ssafy.dto.user.response;

import com.project.ssafy.common.ResponseCode;
import com.project.ssafy.common.ResponseMessage;
import com.project.ssafy.dto.ResponseDto;
import lombok.AllArgsConstructor;
import lombok.Getter;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;

@Getter
@AllArgsConstructor
public class SignUpResponseDto extends ResponseDto {

    private SignUpResponseDto(String code, String message) {
        super(code, message);
    }

    public static ResponseEntity<?> success() {
        SignUpResponseDto result = new SignUpResponseDto(Resp
        return ResponseEntity.status(HttpStatus.OK).body(resu
```

```
    }

    public static ResponseEntity<?> existedUserEmail() {
        SignUpResponseDto result = new SignUpResponseDto(Resp
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).
    }

    public static ResponseEntity<?> existedUserPhone() {
        SignUpResponseDto result = new SignUpResponseDto(Resp
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).
    }

    public static ResponseEntity<?> existedUserNickname() {
        SignUpResponseDto result = new SignUpResponseDto(Resp
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).
    }

}
```

Service

```
package com.project.ssafy.service;

import com.project.ssafy.dto.user.request.SignUpRequestDto;
import org.springframework.http.ResponseEntity;

public interface UserService {
    public ResponseEntity<?> signUp(SignUpRequestDto signUpRe
}
```

ServiceImpl

```
package com.project.ssafy.service.serviceImpl;

import com.project.ssafy.dto.user.request.SignUpRequestDto;
import com.project.ssafy.dto.user.response.SignUpResponseDto;
```

```
import com.project.ssafy.entity.RoleRegister;
import com.project.ssafy.entity.User;
import com.project.ssafy.repository.RoleRegisterRepository;
import com.project.ssafy.repository.UserRepository;
import com.project.ssafy.service.UserService;
import jakarta.transaction.Transactional;
import org.springframework.beans.factory.annotation.Autowired
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.bcrypt.BCryptPassw
import org.springframework.stereotype.Service;

import java.time.LocalDateTime;

@Transactional
@Service
public class UserServiceImpl implements UserService {
    @Autowired
    UserRepository userRepository;

    @Autowired
    RoleRegisterRepository roleRegisterRepository;

    @Autowired
    BCryptPasswordEncoder passwordEncoder;

    @Override
    public ResponseEntity<?> signUp(SignUpRequestDto signUpRe
        // 1) 해당 사용자의 입력 이메일이 이미 존재하는 계정인지 확인
        User existedUser = userRepository.findByEmail(signUpR
        if(existedUser != null) return SignUpResponseDto.exis
        // 2) 해당 사용자의 입력 연락처가 이미 존재하는 계정인지 확인
        User existedUserPhone = userRepository.findByPhone(si
        if(existedUserPhone != null) return SignUpResponseDto
        // 3) 해당 사용자의 입력 닉네임이 이미 존재하는 계정인지 확인
        User existedUserNickname = userRepository.findByNickn
        if(existedUserNickname != null) return SignUpResponse

        // 1, 2, 3 해당 사항이 없다면 정상적으로 회원가입 진행
```

```
        userRepository.save(signUpRequestDto.toUserEntity(pas

        // role_register 테이블에도 해당 사항 등록
        User saveUser = userRepository.findByEmail(signUpRequ
        roleRegisterRepository.save(RoleRegister
                .builder()
                .userId(saveUser.getUserId())
                .roleId(2L)
                .createDate(LocalDateTime.now())
                .updateDate(LocalDateTime.now())
                .build());
        return SignUpResponseDto.success();
    }
}
```

Entity

User Entity

```
package com.project.ssafy.entity;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.time.LocalDateTime;

@Builder
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity(name = "user")
@Table(name = "user")
public class User {
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long userId;
    private String email;
    private String password;
    private String name;
    private Character gender;
    private String phone;
    private String nickname;
    private String profile;
    private String statusMessage;
    private String height;
    private String reach;
    private LocalDateTime createDate;
    private LocalDateTime updateDate;
}
```

Role Entity

```
package com.project.ssafy.entity;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.time.LocalDateTime;

@Builder
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity(name = "role")
@Table(name = "role")
public class Role {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Long roleId;
    private String roleNameEng;
    private String roleNameKor;
    private LocalDateTime createDate;
    private LocalDateTime updateDate;
}
```

Role Register Entity

```
package com.project.ssafy.entity;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.time.LocalDateTime;

@Builder
@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity(name = "roleRegister")
@Table(name = "role_register")
public class RoleRegister {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long roleRegisterId;
    private Long userId;
    private Long roleId;
    private LocalDateTime createDate;
    private LocalDateTime updateDate;
}
```

Repository

User Repository

```
package com.project.ssafy.repository;

import com.project.ssafy.entity.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface UserRepository extends JpaRepository<User, L
    User findByEmail(String email);
    User findByPhone(String phone);
    User findByNickname(String nickname);
}
```

Role Repository

```
package com.project.ssafy.repository;

import com.project.ssafy.entity.Role;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface RoleRepository extends JpaRepository<Role, L
}
```

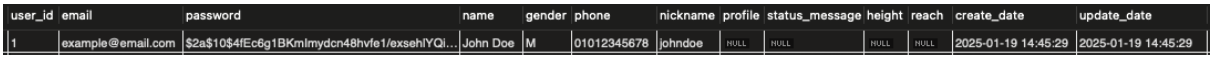
Role Register Repository

```
package com.project.ssafy.repository;

import com.project.ssafy.entity.RoleRegister;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
```

```
public interface RoleRegisterRepository extends JpaRepository
}
```

구현

User



user_id	email	password	name	gender	phone	nickname	profile	status_message	height	reach	create_date	update_date
1	example@email.com	$2a$10$4fEc6g1BKmlmydcn48hvfe1/exsehlYQi...	John Doe	M	01012345678	johndoe	NULL	NULL	NULL	NULL	2025-01-19 14:45:29	2025-01-19 14:45:29

Role Register

role_register...	user_id	role_id	create_date	update_date
1	1	2	2025-01-19 14:45:29	2025-01-19 14:45:29