

# Group-by-group Schedule Planner

네이버 클라우드 11기 이장우

# Contents

- 로그인
- 회원가입
- 비밀번호 찾기
- 메인 페이지
- 개인 일정 캘린더
  - a. Create schedule
  - b. Read and Update schedule
  - c. Delete Schedule
- 그룹 일정 캘린더
  - a. Read and Update schedule
  - b. Delete schedule
- 내 정보 수정
- 전체적인 프로젝트 파일 구성
- Database 구성
- 개선점

# 로그인



Family Connect에 오신 것을 환영합니다!

이메일

비밀번호

Log in

```
userDao dao = new userDao();
try {
    if (dao.searchByEmail(user.getEmail())) {<script>
        alert("이미 존재하는 아이디가 있습니다.");
        history.go(-1);
    </script>
    }
    else{
        try {
            dao.insertUser(user);
        } catch (ParseException e) {
            throw new RuntimeException(e);
        }
    }
} catch (SQLException e) {
    System.out.println(e.getMessage());
}
response.sendRedirect("../view/login.jsp");
```

이메일  
중복체크

require


require

Forward.  
회원가입

Create new account   Forgot password

Forward. 비밀번호  
찾기

# 회원가입



회원가입  
정보를 입력해 주세요.

이름

비밀번호

이메일

핸드폰 번호

초대코드

연도, 월, 일.

Sign Up

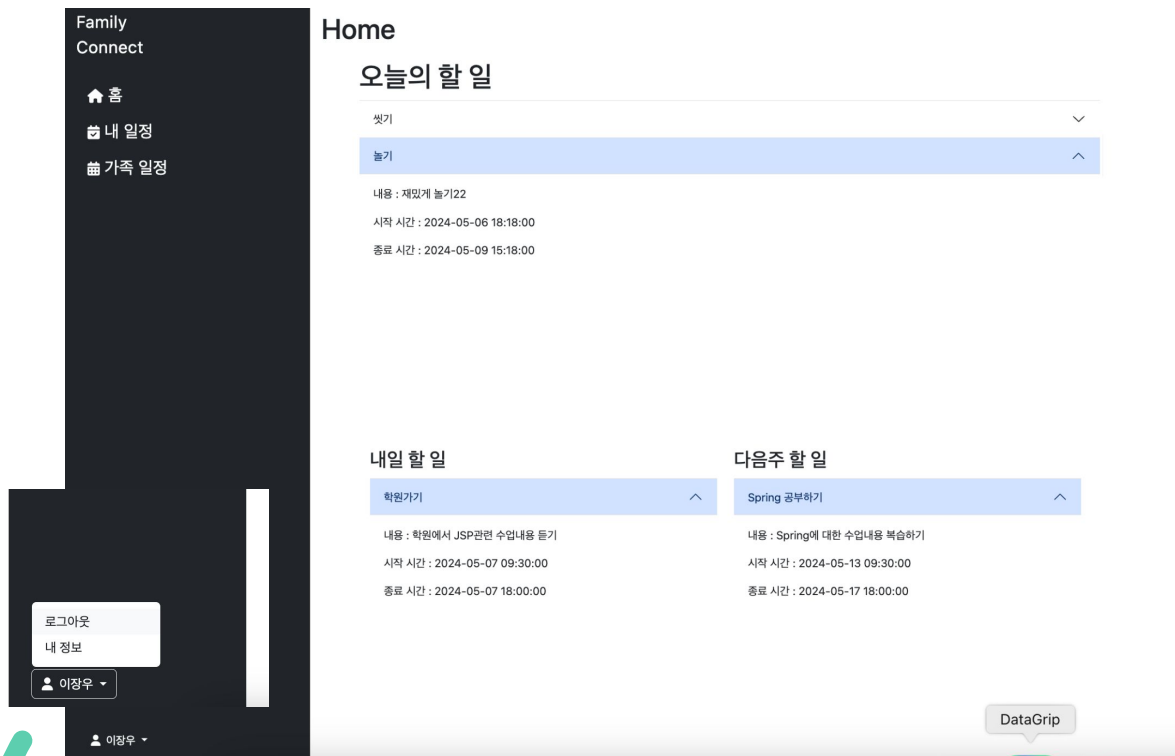
이미 계정을 보유 중입니까? [Sign in](#)

해당 이미지에서 보이는 회원가입에 필요한 요소들 중에 초대코드만을 제외하고 **required**로 표시하여 반드시 입력받게 설정하였습니다.

초대 코드는 입력하지 않을 경우 자동으로 **UUID**값을 생성하여 새로운 그룹을 만들어냅니다.

Forward. 로그인

# 메인 페이지



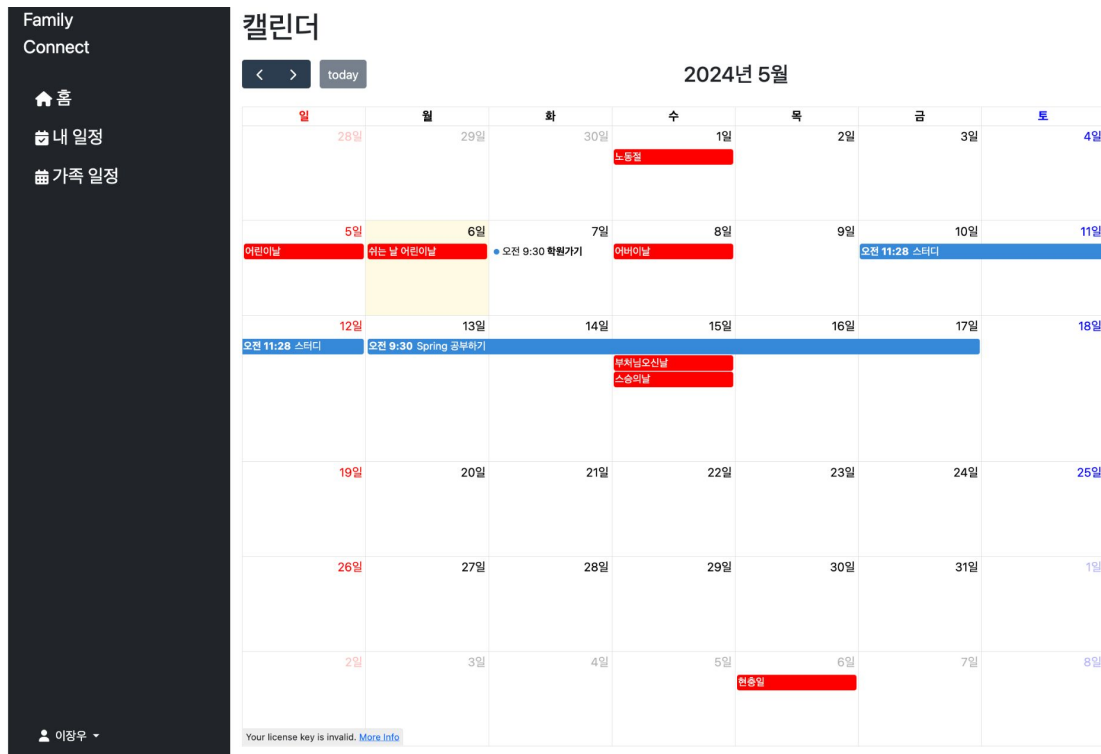
오늘 할 일, 내일 할 일, 다음주 할 일을 각각 **schedule** 테이블의 날짜를 기준으로 한눈으로 볼 수 있게 출력하였습니다.

해당 **UI** 부분은 **bootstrap 5.3** 아코디언을 사용하여 표시하였습니다.

좌측 사이드바의 홈, 내일정, 가족일정을 통해 각각의 메뉴에 접근할 수 있습니다.

밑에 자신의 이름을 눌러 드롭다운 메뉴를 활성화하면 내 정보 수정 및 로그아웃을 진행할 수 있습니다.

# 개인 일정 캘린더

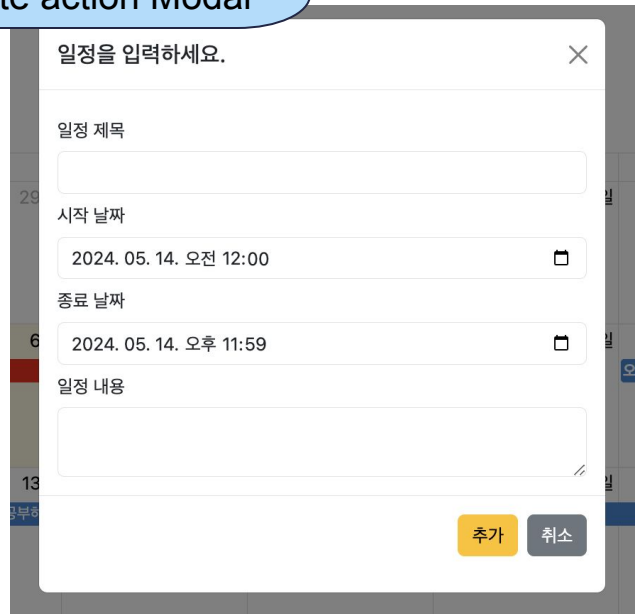


Javascript를 활용하여  
캘린더를 구성하였고 google  
calendar api를 활용하여  
공휴일은 빨간색으로  
처리하여 좀 더 눈에 보기  
좋게 표시하였습니다.

Create : 날짜를 클릭  
Read and Update : 해당 일정을  
클릭  
Delete : 해당 일정 클릭 후 삭제 클릭

# 개인 일정 캘린더 - Modal

Create action Modal



A modal form for creating a new event. It has a title bar with a close button (X). The form contains four input fields: '일정 제목' (Event Title), '시작 날짜' (Start Date) with a calendar icon, '종료 날짜' (End Date) with a calendar icon, and '일정 내용' (Event Content) with a text area. At the bottom right, there are two buttons: '추가' (Add) in yellow and '취소' (Cancel) in gray.

일정을 입력하세요.

일정 제목

시작 날짜

2024. 05. 14. 오전 12:00

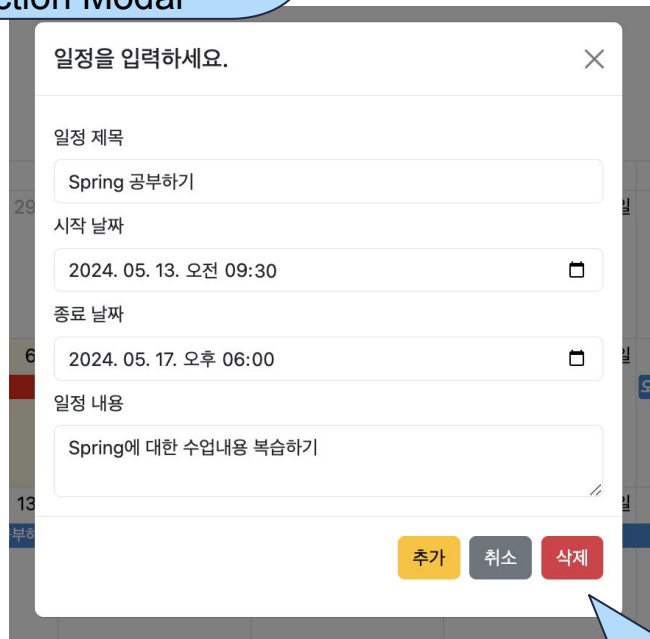
종료 날짜

2024. 05. 14. 오후 11:59

일정 내용

추가 취소

Read and Update  
action Modal



A modal form for reading and updating an event. It has a title bar with a close button (X). The form contains four input fields: '일정 제목' (Event Title) with the value 'Spring 공부하기', '시작 날짜' (Start Date) with the value '2024. 05. 13. 오전 09:30' and a calendar icon, '종료 날짜' (End Date) with the value '2024. 05. 17. 오후 06:00' and a calendar icon, and '일정 내용' (Event Content) with the value 'Spring에 대한 수업내용 복습하기'. At the bottom right, there are three buttons: '추가' (Add) in yellow, '취소' (Cancel) in gray, and '삭제' (Delete) in red.

일정을 입력하세요.

일정 제목

Spring 공부하기

시작 날짜

2024. 05. 13. 오전 09:30

종료 날짜

2024. 05. 17. 오후 06:00

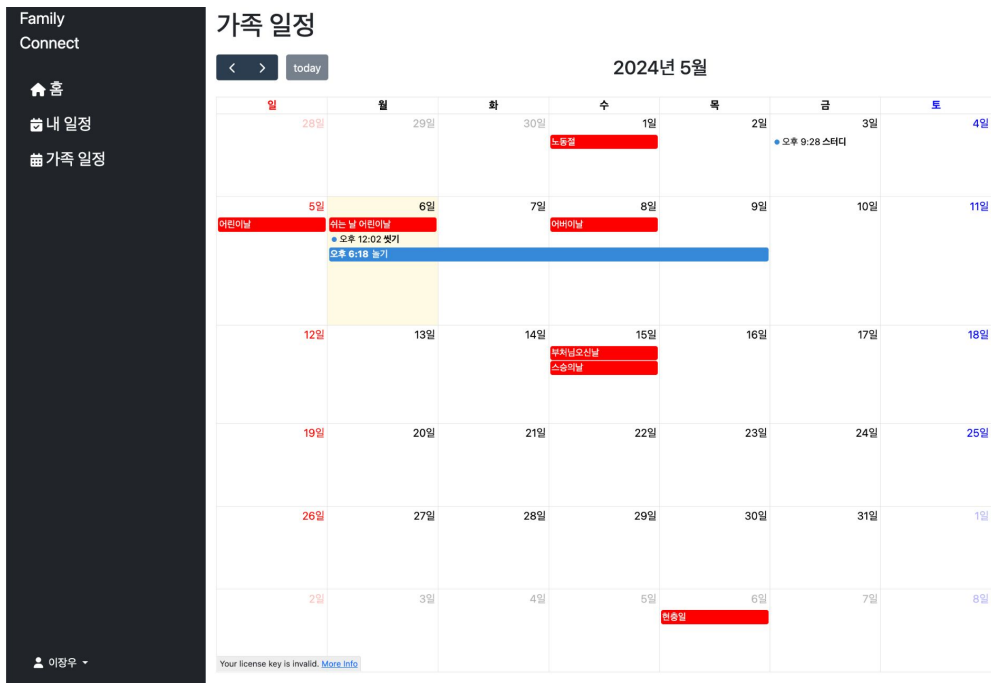
일정 내용

Spring에 대한 수업내용 복습하기

추가 취소 삭제

Delete action  
Modal

# 그룹 일정 캘린더



그 전 캘린더와 크게  
다른점이 없지만 다른  
점은 자신이 속한  
그룹의 모든 일정을  
확인 할 수 있습니다.

단. 자신이 작성하지  
않은 그룹일정은 수정  
및 삭제 할 수 없습니다.



# 그룹 일정 캘린더 - Modal

Create action Modal

일정을 입력하세요.

일정 제목

시작 날짜

2024. 05. 01. 오전 12:00

종료 날짜

2024. 05. 01. 오후 11:59

일정 내용

추가

취소

Read and Update  
action  
Modal(작성자)

일정을 입력하세요.

일정 제목

놀이

시작 날짜

2024. 05. 06. 오후 06:18

종료 날짜

2024. 05. 09. 오후 03:18

일정 내용

재밌게 놀기22

추가

취소

삭제

Read action Modal  
(작성자 X)

일정 확인

일정 제목

씻기

작성자

a1

시작 날짜

2024. 05. 06. 오후 12:02

종료 날짜

2024. 05. 06. 오후 01:02

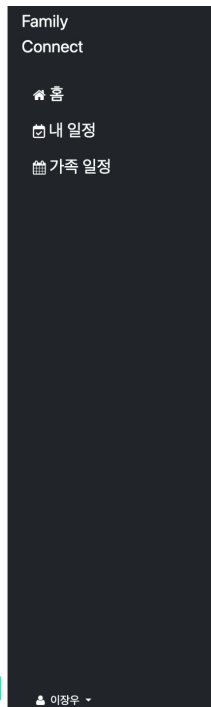
일정 내용

깨끗하게 씻기

취소

Delete action Modal

# 내 정보 수정




회원 정보 수정  
정보를 입력해 주세요.

이름  
이장우

비밀번호  
\*\*\*\*\*

이메일  
8282qwe@gmail.com

핸드폰 번호  
01028081787

1998. 04. 26. 

초대코드  
1d0e17aa-a0c2-43ac-8b2a-a668bc78f4a9

수정하기

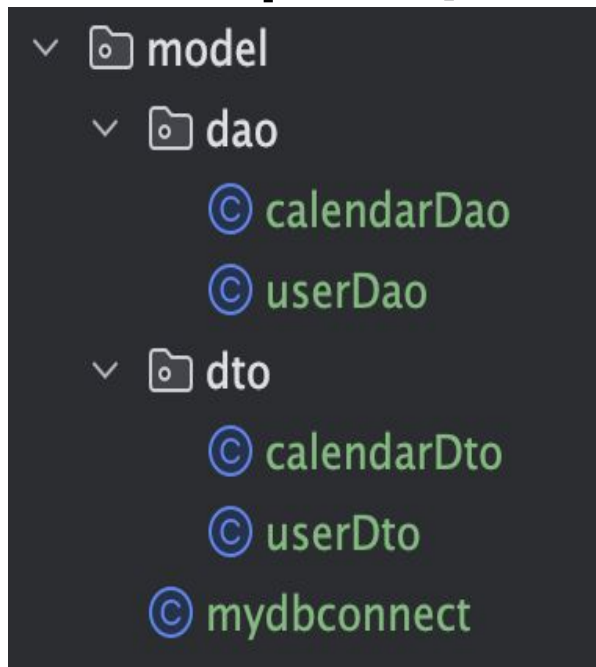
탈퇴하시겠습니까? 탈퇴하기

Update User Form

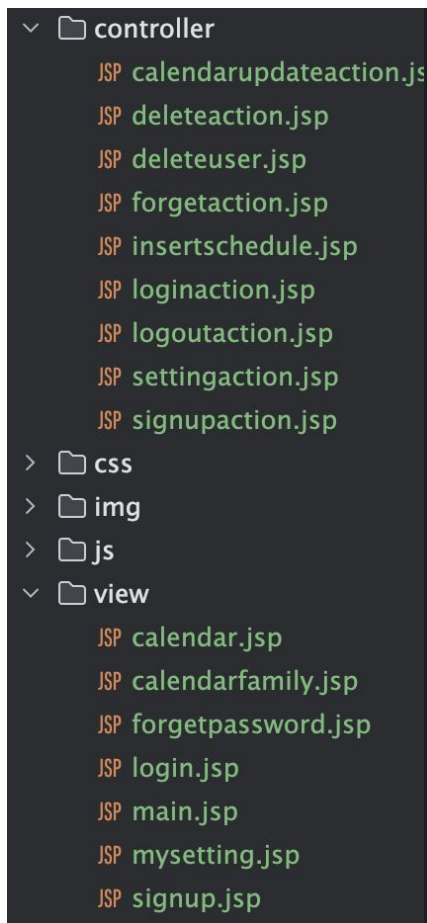
해당 부분  
수정을 통해  
그룹 변경 가능

해당 버튼을 통해 Delete  
User

# 프로젝트 구성



java클래스 부분을 model 패키지에  
담아 dao,dto 부분으로 남아서  
관리하였습니다.



기능을 구현하는 controller와  
단순 페이지 구성을 위한  
view로  
나누어관리하였습니다.

# Dto 구성

```
@Getter 18 usages 신규 *
@Setter
@NoArgsConstructor
public class userDto {
    private int userid;
    private String name;
    private String email;
    private String password;
    private String phone;
    private String birthday;
    private String groupid;
    private String profile;
}
```

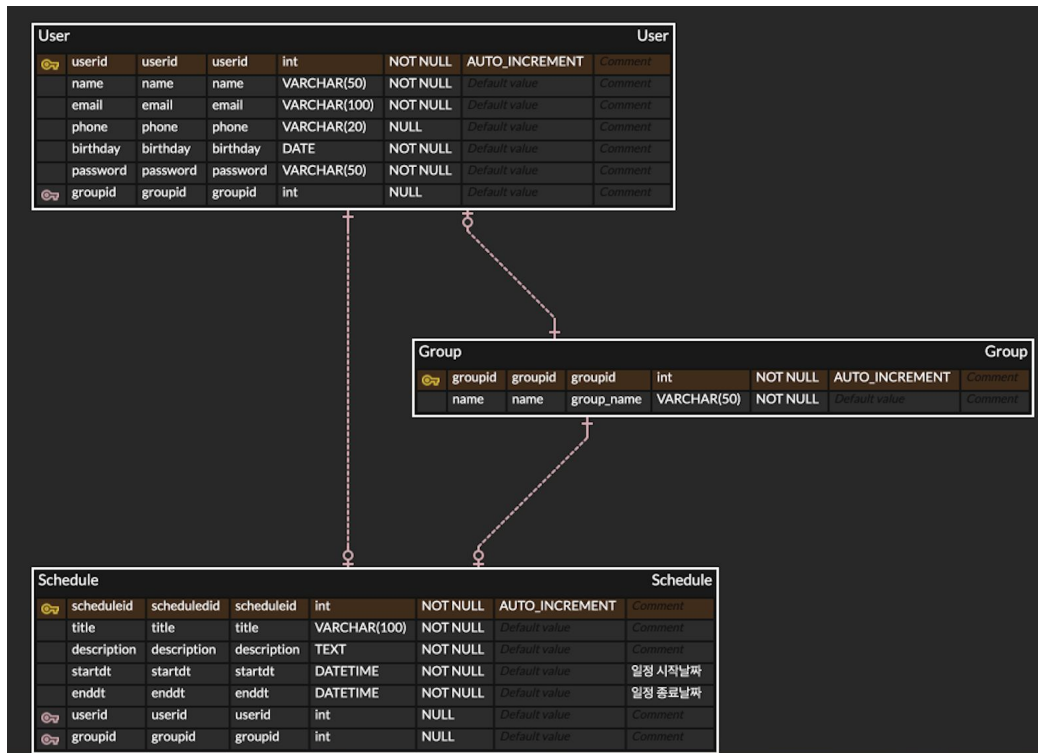
```
@Getter 39 usages 신규 *
@Setter
@NoArgsConstructor
public class calendarDto {
    private int scheduledid;
    private String title;
    private String description;
    private String startdt;
    private String enddt;
    private int userid;
    private int groupid;
}
```

# Dao 구성

```
public class userDao { 22 usages 신규 *  
    mydbconnect mydb = new mydbconnect(); 8 usages  
  
    public boolean insertUser(userDto user) throws SQLException, ParseException {...}  
  
    public boolean searchByEmail(String email) throws SQLException {...}  
  
    public String searchForPassword(String email) throws SQLException, ParseException {...}  
  
    public userDto searchByEmailAndPassword(String email, String password) throws SQLException {...}  
  
    public userDto searchByUserId(int userid) throws SQLException {...}  
  
    public boolean updateByuserid(int userid, userDto userDto) throws SQLException, ParseException {...}  
  
    public boolean deleteByuserid(int userid) throws SQLException {...}  
  
    public String searchByIdForUsername(int id) throws SQLException {...}  
}
```

```
public class calendarDao { 18 usages 신규 *  
    mydbconnect mydb = new mydbconnect(); 8 usages  
  
    public List<calendarDto> searchAllByUserIdNotGroupidToday(int userId, String groupid) throws SQLException {...}  
  
    public List<calendarDto> searchAllByUserIdNotGroupidTomorrow(int userId, String groupid) throws SQLException {...}  
  
    public List<calendarDto> searchAllByUserIdNotGroupidNextWeek(int userId, String groupid) throws SQLException {...}  
  
    public boolean insertSchedule(calendarDto calendarDto, String userid, String groupid) throws SQLException {...}  
  
    public List<calendarDto> searchAlluserid(String userid) throws SQLException {...}  
  
    public List<calendarDto> searchAlluserid(String userid, String groupid) throws SQLException {...}  
  
    public boolean updateSchedule(calendarDto calendarDto) throws SQLException {...}  
  
    public boolean deleteSchedule(int scheduledid) throws SQLException {...}
```

# ERD 다이어그램



# 개선 방안

- A. 캘린더 랜더링 방식이 현재는 해당 페이지에서 폼 액션을 통해 데이터를 주고 처리하여 다시 랜더링하는 방식으로 구현되었지만 **REST** 방식의 서버-클라이언트 방식으로 구성을 변경한다면 좀 더 쉽게 구현할 수 있지 않을까 생각하였습니다.
  
- B. 단순히 일정만 공유하는 방식이 아닌 해당 일정에 대해 주최자가 알람을 통해 해당 모든 그룹원들에게 해당 일정에 대해 메시지를 전송하고, 원하는 그룹원을 초대하기 위한 초대코드를 좀 더 편하게 공유할 수 있게 메신저 플랫폼(**kakao**) API를 활용하여 좀 더 좋은 기능을 구현해보고 싶습니다.

긴 내용 읽어 주셔서 감사합니다.

