



박근영과누나들

---

# TimeCodi

---

홍시현

김소영

이현영

조서연

박근영



**01. Objective & Milestone**

**02. Role of each member**

**03. Implementation**

**Frontend : Design UX/UI**  
**Backend**

**04. Challenges**

**05. Limitation**

**06. Demo**

**07. Next Week**



# **01**

## **OBJECTIVE & MILESTONE**

---



## OBJECTIVE

---

- **Easily coordinate schedules with members**
  - **Recommend time zones when all members are available**
  - **No need to re-enter schedules when creating other meetings**
-

MILESTONE											
Schedule	5	6	7	8	9	10	11	12	13	14	15
Frontend	Design Prototype					Friends' Time Zone Page		Other UI	Deploy & Review		
Backend	Design DB Structure					Invite & Notification Function		Recommendation & Vote Function	Deploy & Review		



# **02**

## **Role of each member**

---



## Role of each member

---

### Front-end

김소영

- Design: UX/UI
- Implement
  - Start Page
  - Login & Register Page
  - Friends Page

조서연

- Design: UX/UI
- Implement
  - User schedule Page

홍시현

- Design API
- Implement
  - My Page
  - Group Page

### Back-end

이현영

- API
  - Google map & calendar
  - KakaoTalk share
- Implement
  - Schedule functions

박근영

- Design DB structure
  - Connect React and DB
  - Implement
    - Login & Register functions
-

# **03**

## **IMPLEMENTATION**

---





## FRONT-END

---

- **Implement : ‘Start Page’**
    - ‘Login & Register Page’**
    - ‘User Schedule Page’ (schedule functions)**
    - ‘My Page’ (My group list, Invited group list)**
    - ‘Group Page’ (Members, Meeting Info, Vote)**
    - ‘Friends Page’ (Friends List, Add button)**
-



# FRONT-END: Design UX/UI

## Start Page

A design mockup of a start page. It features a light purple background with a white rectangular card in the center. The card contains the text "Hello! Welcome to TimeCodi." in a bold, black, sans-serif font. Below the text are two rounded rectangular buttons with purple borders and purple text: "Login" and "Register".

## Login & Register Page

A design mockup of a login and register page. It features a light purple background with two white rectangular cards side-by-side. The left card is for login, with the heading "Hello! let's get started" and the subtext "Sign in to continue." It includes input fields for "Username" and "Password", a purple "SIGN IN" button, a checkbox for "Keep me signed in", and a link "Don't have an account? Create". The right card is for registration, with the heading "New here?" and the subtext "Signing up is easy. It only takes a few steps". It includes input fields for "Username", "Email", and "Password", a checkbox for "I agree to all Terms & Conditions", a purple "SIGN UP" button, and a link "Already have an account? Login".

# FRONT-END: Design UX/UI

## User Personal Page

- **Open API**
- **Add, Delete, Modify schedules**
- **Set visibility**
- **Repeat events**
- **Google account linkable**

## User schedule Page

User Schedule

April 4 / 2023

< TODAY >

SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

Add Event

Event Name:

capstone

Visibility:

private

Start Date:

2023-04-29 오후 03:00

End Date:

2023-05-06 오전 04:00

Repeat:

none

Repeat End Date:

년-월-일

Submit

Cancel

or

Fetch schedule via Google Calendar

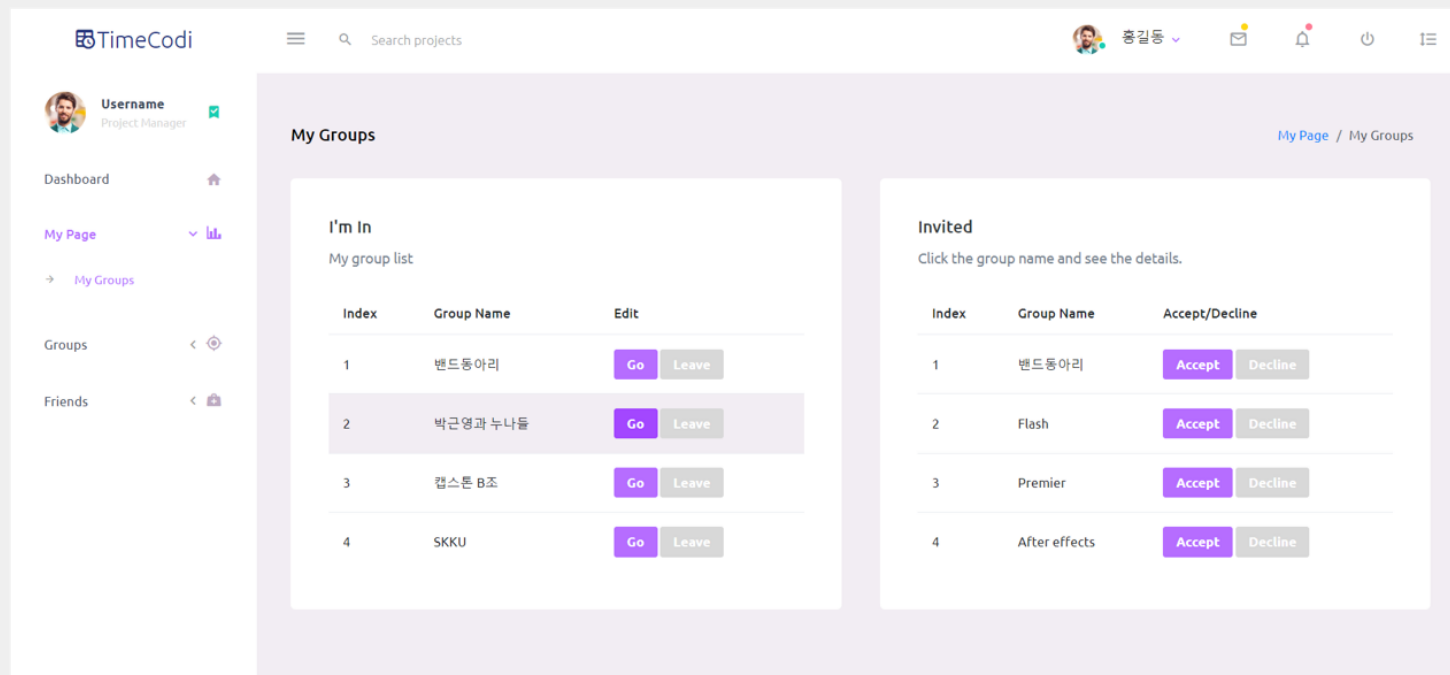
Upload Google Calendar

# FRONT-END: Design UX/UI

## My Page

### User My Page

- **My group list**
- **Invited group list**

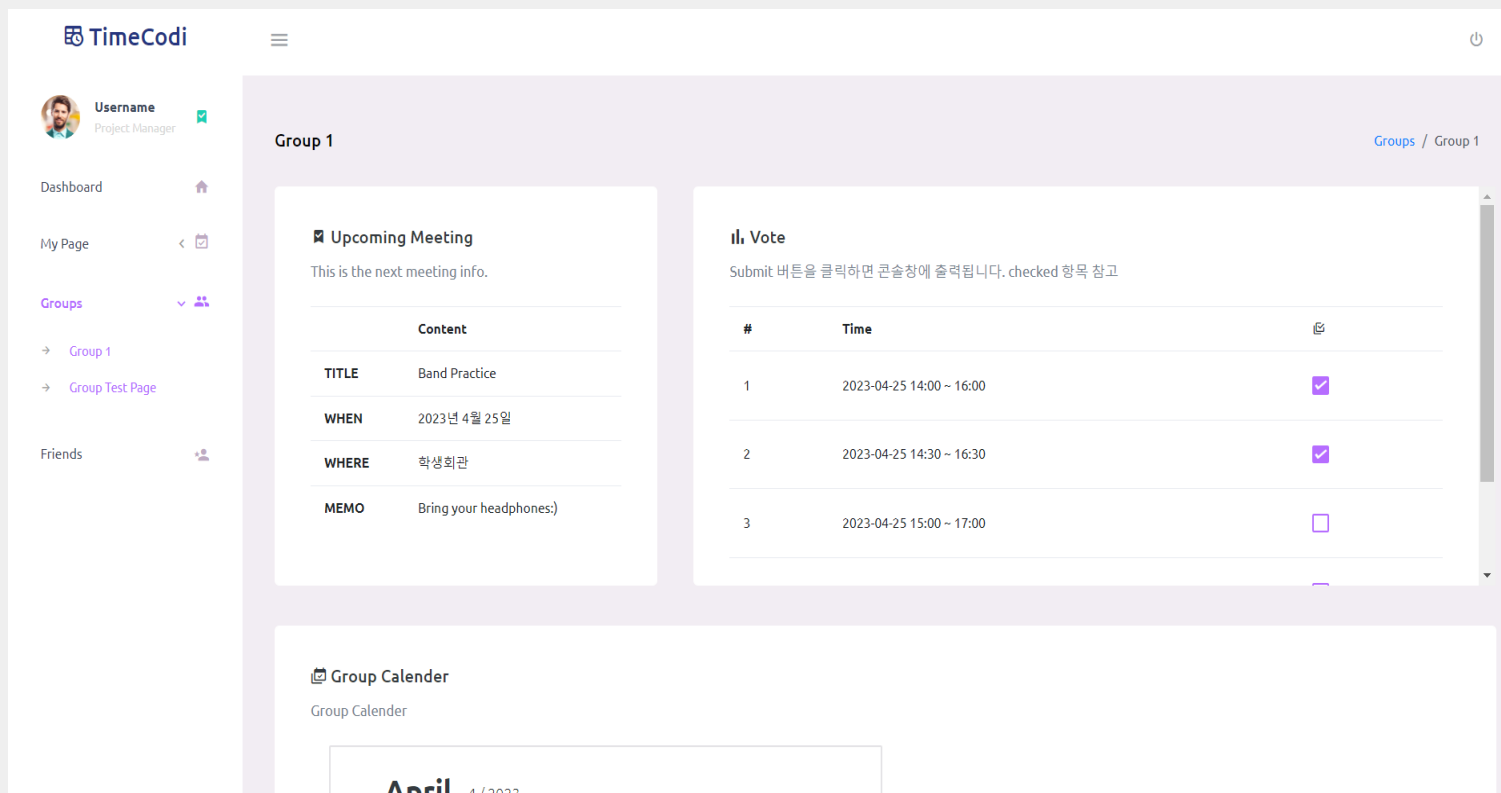


# FRONT-END: Design UX/UI

## Group Page

### Each Group Page

- **Members**
- **Group Meeting information**
- **Vote time zone**
- **Group Calendar**




# FRONT-END: Design UX/UI

## Group Page

### Each Group Page

- **Members**
- **Group Meeting information**
- **Vote time zone**
- **Group Calendar**

 Group Calender

Group Calender

April 4 / 2023							< TODAY >	
SUN	MON	TUE	WED	THU	FRI	SAT		
26	27	28	29	30	31	1		
2	3	4	5	6	7	8		
9	10	11	12	13	14	15		
16	17	18	19	20	21	22		
23	24	25	26	27	28	29		
30	1	2	3	4	5	6		

 Edit Meeting Info

 Create New Meeting

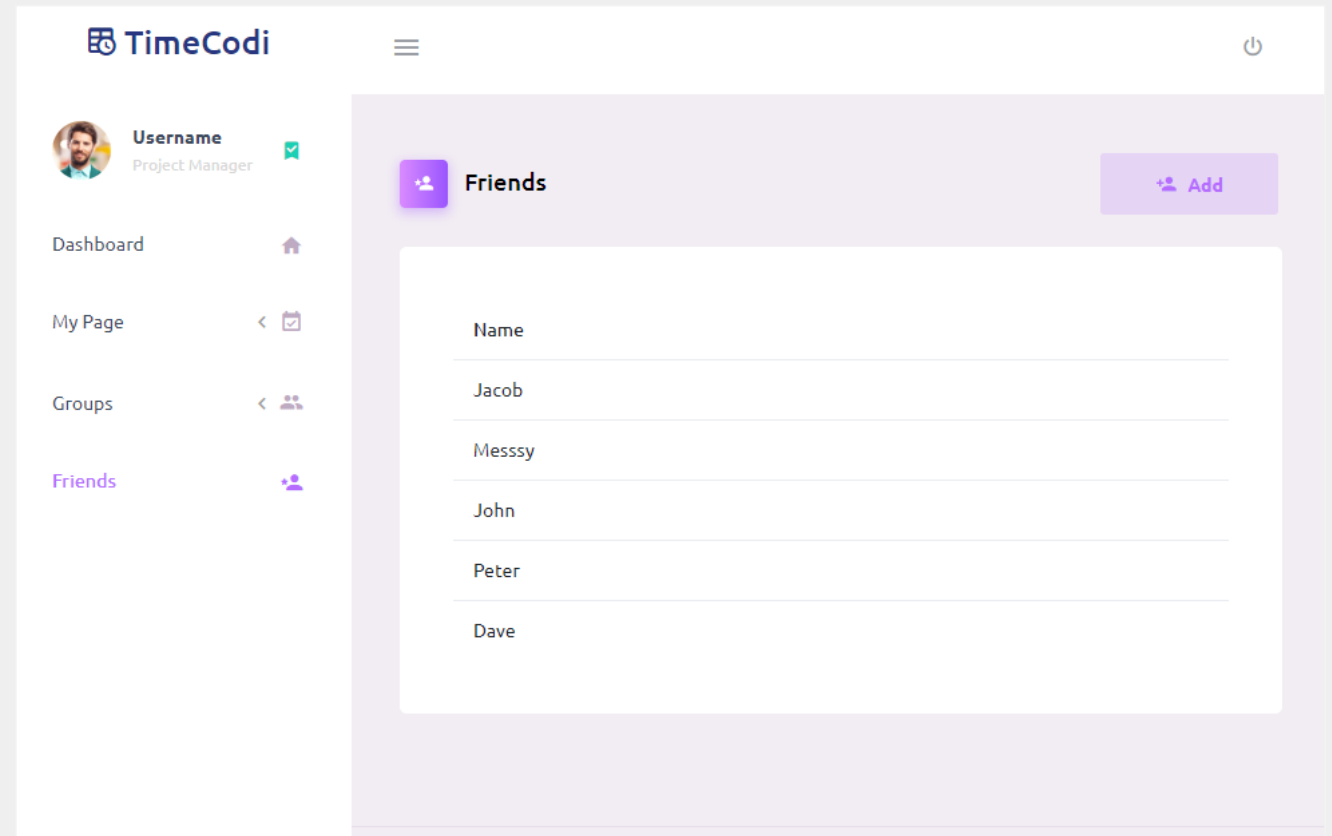
timetable

# FRONT-END: Design UX/UI

## Friends Page

### User Friends Page

- **Add, Delete Friend list**
- **Check your friend's schedule**





## BACK-END

---

- **Verify Login account**
- **Link Login & Register, User schedule page to MariaDB**
- **Implement : Time-Zone setting functions**

**(Add & Delete & Modify event)**

**(Repeat events)**

---



# Back-end: Login account Authorized

The screenshot displays the Swagger UI for a FastAPI application. The top left corner shows 'FastAPI 0.1.0 OAS3' and the URL '/openapi.json'. The main interface is divided into two columns. The left column lists API endpoints under the 'default' tag: GET / (Get Home), GET /login (Login), POST /signin (Signin User), POST /signup (Signup User), and GET /event (Get Event). The right column shows the details for the selected endpoint, including parameters, responses, and a 'Try it out' button. A modal dialog titled 'Available authorizations' is open in the center. It contains the following text: 'Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. API requires the following scopes. Select which ones you want to grant to Swagger UI.' Below this, it lists 'OAuth2PasswordBearer (OAuth2, password)' with the status 'Authorized'. The details for this authorization are: Token URL: /signin, Flow: password, username: test, password: \*\*\*\*\*, Client credentials location: basic, and client\_secret: \*\*\*\*\*. At the bottom of the dialog are 'Logout' and 'Close' buttons.

FastAPI 0.1.0 OAS3  
/openapi.json

Authorize

### default

GET / Get Home

GET /login Login

POST /signin Signin User

POST /signup Signup User

GET /event Get Event

Parameters

No parameters

Responses

Code Description Links

### Available authorizations

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. API requires the following scopes. Select which ones you want to grant to Swagger UI.

**OAuth2PasswordBearer (OAuth2, password)**

Authorized

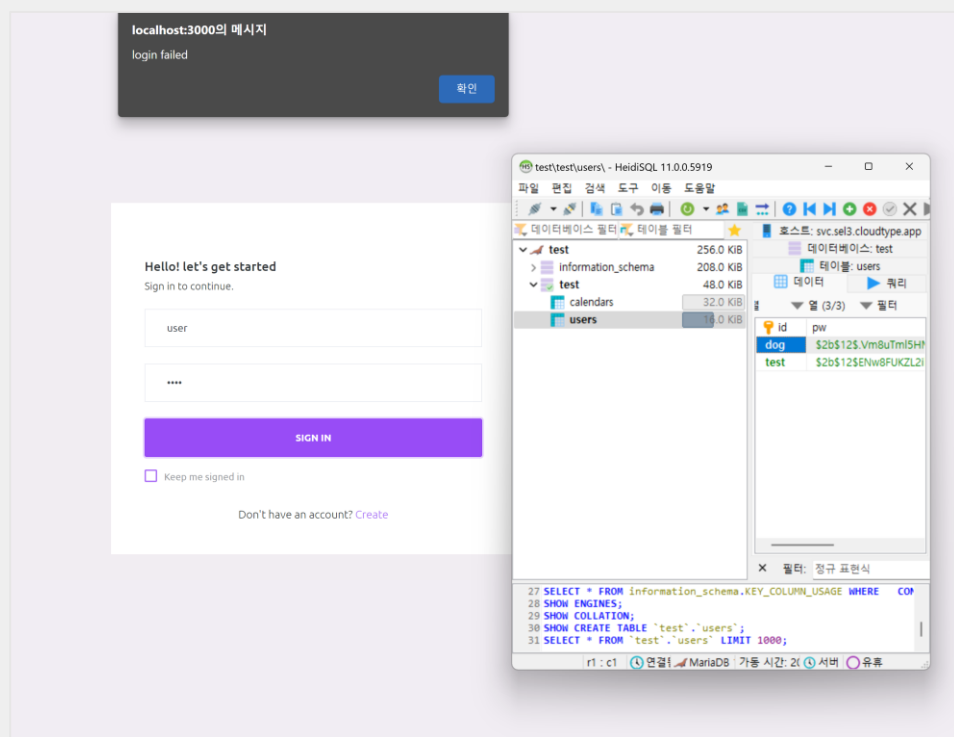
Token URL: /signin  
Flow: password  
username: test  
password: \*\*\*\*\*  
Client credentials location: basic  
client\_secret: \*\*\*\*\*

Logout Close

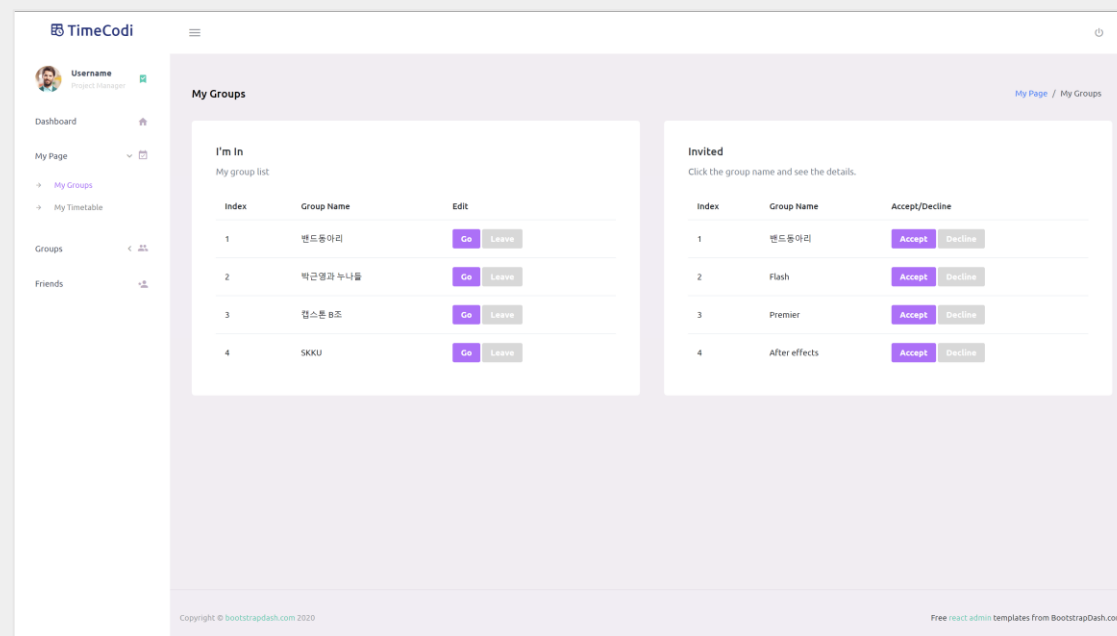
Try it out

# Back-end: Verifying Login account

## No account



## Success login (Main Page)



# Back-end: Link React ↔ Maria DB

## Schedule Page (React)

User Schedule

April 4 / 2023

SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

**Add Event**

Event Name:  
capstone

Visibility:  
private

Start Date:  
2023-04-29 오후 03:00

End Date:  
2023-05-06 오전 04:00

Repeat:  
none

Repeat End Date:  
년-월-일

**Submit** Cancel

Upload Google Calendar



## Add event (MariaDB)

test(test\calendars\ - HeidiSQL 11.0.0.5919)

test

test.calendars: 1 행 (총) (대략적)

cid	uid	cname	visibility	sdatetime	edatetime
18	test	capstone	0	2023-04-29 15:00:00	2023-05-06 04:00:00

```

486 SHOW INDEXES FROM `calendars` FROM `test`;
487 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='test' AND TABLE_NAME='calendars' AND F
488 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE CONSTRAINT_SCHEMA='test' AND TABLE_NAME='calendars' AND REFERENC
489 SHOW CREATE TABLE `test`.`calendars`;
490 SELECT * FROM `test`.`calendars` LIMIT 1000;
  
```

r1 : c5 연결됨: 00:24 MariaDB 10.5.19 가동 시간: 13:04 h 서버 시간: 오 유틸

## Get event

**(Login account's ID = Event's uid)**

The screenshot shows a REST client interface with the following details:

- Request:**
  - Method: GET
  - URL: http://localhost:8080/test/calendars
  - Headers:
    - Host: localhost:8080
    - Accept: \*/\*
- Response:**
  - Status: 200 OK
  - Content-Type: application/json
  - Body:
 

```
[{"cid": 18, "cname": "capstone", "sdatetime": "2023-04-29T15:00:00", "visibility": false, "uid": "test", "edatetime": "2023-05-06T04:00:00"}, {"cid": 19, "cname": "OS", "sdatetime": "2023-04-13T15:00:00", "visibility": true, "uid": "test", "edatetime": "2023-04-19T04:00:00"}, {"cid": 20, "cname": "CG", "sdatetime": "2023-05-15T15:30:00", "visibility": true, "uid": "test", "edatetime": "2023-05-31T15:30:00"}]
```
- Headers:**
  - content-length: 377
  - content-type: application/json
  - date: Thu, 27 Apr 2023 08:34:40 GMT
  - strict-transport-security: max-age=15724800; includeSubDomains; preload

## Delete event

[illegible]

# **04**

## **CHALLENGES**

---



## Challenge

---

- 1. Version Conflict between node and node-sass**  
**Compatibility Issue between node.js and outdated node-sass**  
**which used in our Bootstrap template**
  - 2. Version Conflict between team members**
-



## Solution

---

- **Solution 1: Manual Installation**

**Node-sass that can work with our template**

**Downgraded node.js to a compatible version**

**Python 2, which is required by some libraries used in node-sass**

---



## Solution

- **Solution2: In order to match versions between team members, all of our members matched the versions of node.js, python, node-sass and npm (instead of using docker)**

9 lines (9 sloc) | 943 Bytes

```
1  1. node js를 14버전으로 재설치해주세요. 아마 node 아예 삭제하고 재설치해야 하는 것 같습니다.  
2  (Windows 기준) https://nodejs.org/download/release/v14.21.3/ 여기서 확장자가 .msi로 끝나는 거 다운받아서 설치하면 됩니다.  
3  2. python 3 말고 2가 필요합니다. (sass 라이브러리 중에 그걸 쓰는 게 있다고 합니다)  
4  설치하시고, 윈도우는 환경변수 설정할때 `C:\Python27\Scripts` 이렇게 Scripts 폴더의 위치로 설정해주세요.  
5  3. 터미널에서 현재 위치를 frontend 폴더로 두세요.  
6  `npm install` 하세요: 에러가 뿔 겁니다. sass 때문입니다.  
7  `npm install node-sass@4` 이렇게 따로 설치합니다. @4는 버전 때문입니다.  
8  4. 이제 npm start하면 예쁜 화면이 떠야 합니다. 안 되면 말씀해주세요  
9  5. 'npm install node-sass@4'에서 오류가 발생하면 'npm install -g npm@7.24.2' 을 해주세요.
```



# **05**

## **LIMITATION**

## Limitation

---

- **We plan to implement Notification function using KakaoTalk message**

**Send the Meeting Info**

17:54

Here's your next meeting:  
Band Practice 🎸  
4:00 p.m. ~ 6:00 p.m. every Tuesday  
at the 03100 학생회관 1층 00호  
Bring your headphones 🎧

**Automatically send meeting information on a meeting day**

---



## Limitation

---

**However, we can't be implemented**

**because it is possible only when a business is registered**

- **Create a KakaoTalk channel account (It's okay)**
  - **Send a notification message through that channel**  
**(It requires business registration)**
-

# **06**

## **DEMO**

---

**07** **NEXT WEEK**

---

# WEEKLY PROGRESS

Schedule	5	6	7	8	9	10	11	12	13	14	15
Frontend	<div>Design Prototype</div>		<div>Login &amp; User Page</div>	<div>Time Zone Setting Page</div>		<div>Friends' Time Zone Page</div>		<div>Other UI</div>		<div>Deploy &amp; Review</div>	
Backend	<div>Design DB Structure</div>		<div>Login &amp; Register Function</div>	<div>Time Zone Setting Function</div>		<div>Invite &amp; Notification Function</div>		<div>Recommendation &amp; Vote Function</div>		<div>Deploy &amp; Review</div>	



## NEXT WEEK

---

<b>Frontend</b>	<ul style="list-style-type: none"><li>• <b>Implement Friend's Time Zone(Schedule) Page</b></li></ul>
<b>Backend</b>	<ul style="list-style-type: none"><li>• <b>Implement Invite function using KakaoTalk message</b></li></ul>