



Team Matching Based on Optimal Algorithm for Any Developers and Designers

프로젝트 등록

프로젝트 이름

프로젝트 이름

프로젝트 설명

프로젝트 설명

프로젝트 공개

☒ 공개

☐ 비공개

포지션 유형

frontend

frontend

1

backend

1

포지션 구성

backend

유형 추가

+ 추가

How much time to spend per week?

0~5

5~10

10~15

15~20

20+

Preferred Date and Time

☒ 월요일 오전

☐ 월요일 오후

☐ 월요일 저녁

☒ 화요일 오전

☐ 화요일 저녁

☐ 수요일 오전

☐ 수요일 오후

☒ 수요일 저녁

Preferred Role

☒ Leader

☐ Follower

Creating web
application using
public API

This is a class about web
development

10 - 05/20

Smart Car
Convergen
Capstone

In this cours
out the pro
cars. Base
gained b
student

Objective and Motivation



Optimal Matching



Anyone Register



Team Chatting

Project Progress

[illegible]

JS class.formTeam.js
JS class.getAllClasses.js
JS class.getClass.js
JS class.getFormTeamWithOption.js
JS class.getGuestClasses.js
JS class.getHostClasses.js
JS class.joinClass.js

JS class.test.js
JS createClass.test.js
JS createMultipleUsers.js
JS createUserWithoutVerify.js
JS formTeam.test.js
JS getClass.test.js
JS getTeam.test.js
JS joinClass.test.js
JS login.test.js
JS condition.js

JS formTeam.js
JS getAllClasses.js
JS getClass.js
JS getFormTeamWithOption.js
JS getGuestClasses.js
JS getHostClasses.js
JS joinClass.js
JS global
JS createFeedback.js

models

JS Answer.js
JS Class.js
JS Feedback.js
JS Question.js
JS Team.js
JS User.js

Final Design: Backend

JS createGroup.js
JS createGraph.js
JS createGroup.js
JS createGroupsGreedy.js
JS createGroupsGreedyOptimal.js
JS createTeam.js
JS deleteRandomData.js
JS getMaxPositionCounter.js
JS resetGroups.js
JS resetPositionCounter.js

Final Design: Algorithm

Step 1

- Position composition

Step 2

- Leaders assigned to each team

Step 3

- Assign remaining users to each team one by one
- Relax conditions of each questions if no more match -> sub optimal solution

Cases for team matching iteration

- **Case 0: Match all** This case aims to match all conditions specified by users' answers.
- **Case 1: Match all with lower condition** Similar to Case 0, but with relaxed conditions.
- **Case 2: Match preferred time and experience** This case focuses on matching users based on their preferred time availability and experience.
- **Case 3: Match preferred time and experience with lower condition** Similar to Case 2, but with relaxed conditions.
- **Case 4: Match preferred time and time spend** This case considers users' preferred time availability and the amount of time they are willing to invest.
- **Case 5: Match preferred time and time spend with lower condition** Similar to Case 4, but with relaxed conditions.
- **Case 6: Match experience and time spend** This case aims to match users based on their experience and the amount of time they are willing to invest.
- **Case 7: Match experience and time spend with lower condition** Similar to Case 6, but with relaxed conditions.
- **Case 8: Match time spend** This case focuses on matching users based on the amount of time they are willing to invest.
- **Case 9: Match time spend with lower condition** Similar to Case 8, but with relaxed conditions.
- **Case 10: Match preferred time** This case considers users' preferred time availability.
- **Case 11: Match experience** This case focuses on matching users based on their experience.
- **Case 12: Match experience with lower condition** Similar to Case 11, but with relaxed conditions.

```
prevTeamConditionId: 0,1,2,3,4,5,12,-1
left followers: 0
{ message: 'Successfully formed a team' }
```

StepOne.module.scss

TS StepOne.tsx

✓ StepTwo

✓ PasswordInput

PasswordInput.module.scss

PasswordInput.tsx

✓ API

TS authApi.ts

TS guestApi.ts

TS hostApi.ts

TS teamApi.ts

MeetingTime

MeetingTime.module.scss

TS MeetingTime.tsx

✓ ToDoList

ToDoList.module.scss

TS ToDoList.tsx

ActivityInfo.module.scss

TS ActivityInfo.tsx

ActivityManage

ActivityManage.module.scss

ActivityManage.tsx

HomeMenu

HomeMenu.module.scss

TS HomeMenu.tsx

Home.module.scss

TS Home.tsx

ManageProject

Class

Class.module.scss

Class.tsx

Projects.tsx

MyPage.module.scss

MyPage.tsx

ParticipateProject

ParticipateProject.module.scss

TS ParticipateProject.tsx

✓ RegisterProect

✓ ClassInfo

ClassInfo.module.scss

TS ClassInfo.tsx

Final Design: Frontend

Demo

Challenges

Backend

- Testing Environment
- Team forming Algorithm Implementation

Frontend

- Project Management
- Team forming Algorithm Implementation

Limitation

- **Extreme Cases**
- **Predefined Questions**

Evaluations

- **Effectiveness of our algorithm**
- **Error-free client server interaction**
- **Complex algorithm but easy UX**
- **Clear matching result feedback**

Thank you