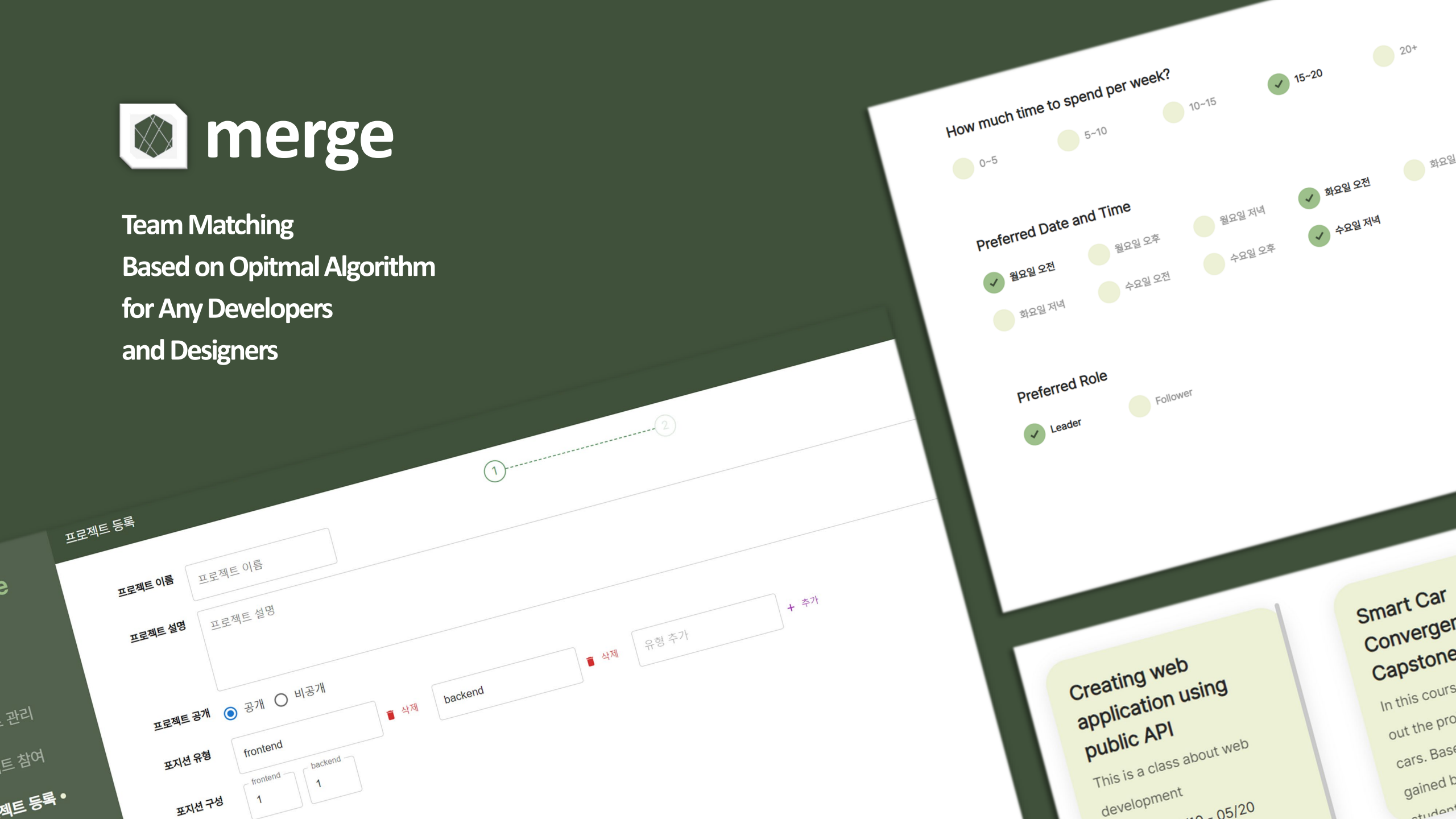
merge

Team Matching
Based on Opitmal Algorithm
for Any Developers
and Designers



Objective and Motivation

Optimal Matching

Anyone Register

Team Chatting

Project Progress

Week	3	4	5	6	7	8	9	10	11	12	13	14
Planning and DB design												
Register / Login												
UI / UX												
Create / Join Project												
Matching Algorithm												
Team chat												
Usability test												

Final Design: Backend

Final Design: Algorithm

Step 1

- Position composition

Step 2

- Leaders assigned to each team

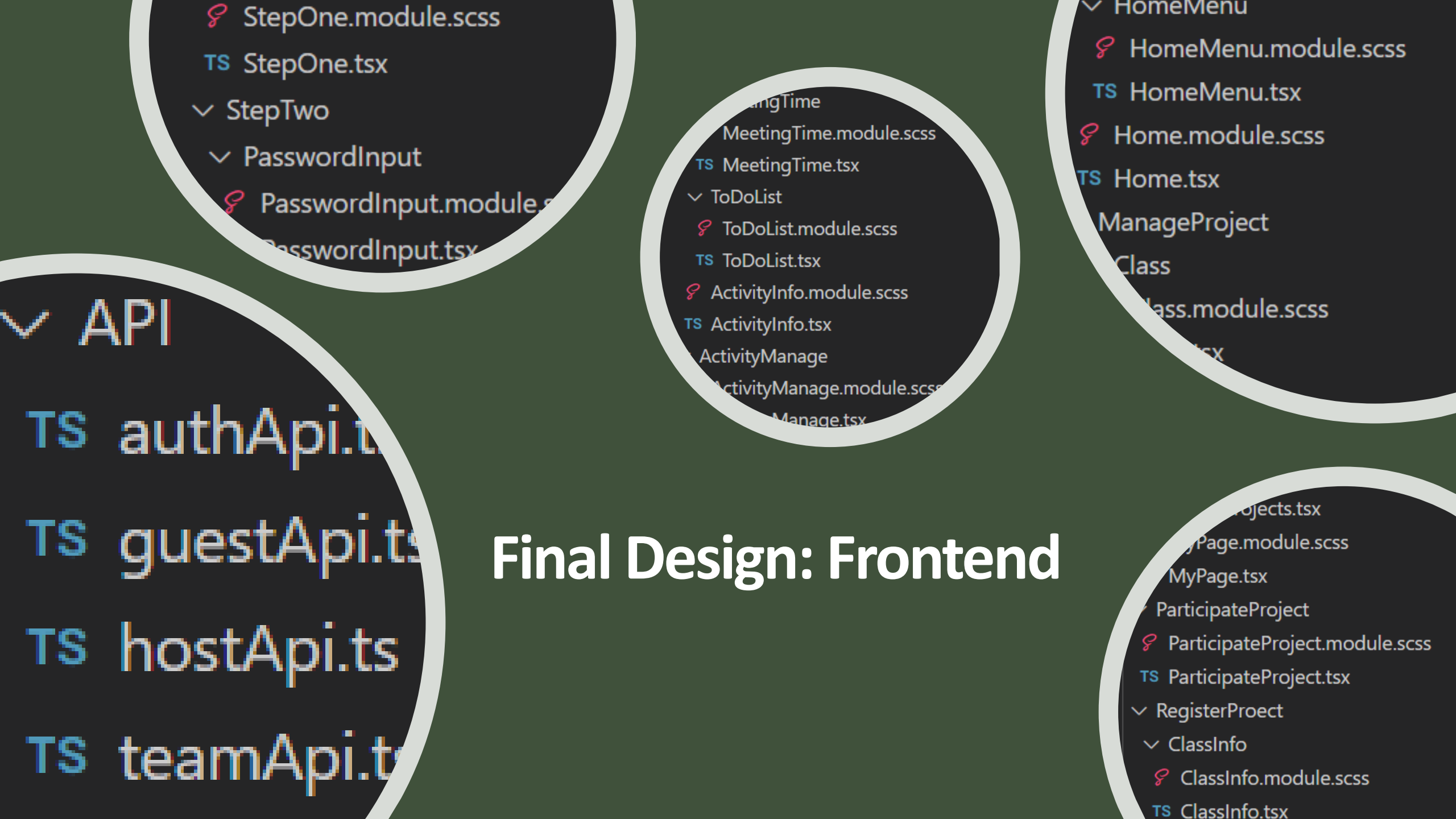
Step 3

- 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' }
```



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