

# Dating Suggestions

## Context

In recent times, some niche online dating apps have seen massive success. At the heart of these apps is the algorithm to recommend best match profiles. In order to ensure that users do not end up browsing large amounts of irrelevant results, the algorithm must return the results based on ranking the matches using weightage of individual matching requirements.

## Problem Statement

Write code for a dating recommendation engine which would find the closest match for any user. Each registered user is expected to have provided:

- Name
- Gender
- Age
- Interests

Recommendation engine applies following rules in given order while identifying the closest matches:

- Gender Rule: Opposite gender is given a preference.
- Age rule: Closest match in terms of age is given a preference.
- Interest rule: Closest match in terms of interests is given a preference.

## Example

Let us say the system has following registered users:

Name	Gender	Age	Interest
User 1	Female	25	Cricket, Chess
User 2	Male	27	Cricket, Football, Movies
User 3	Male	26	Movies, Tennis, Football, Cricket
User 4	Female	24	Tennis, Football, Badminton
User 5	Female	32	Cricket, Football, Movies, Badminton

If system is asked to fetch top 2 matches of User 2, the output should be the following list having matches ordered by their closeness with User 2:

**Output:** [User 1, User 4]

## Explanation

- Though User 5 and User 2 has maximum number of interests matching, age is given preference over interests.
- Similarly, User 2 is closest to User 3 in terms of age and all their interests match, but gender is given preference over age & interests.

- Interest match counts of User 1 and User 4 with User 2 are identical. However, User 1 is closer to User 2 in terms of age when compared to User 4. Therefore, User 1 is given more preference.

### Implicit Requirements

1. Solution must have a robust design.
2. Solution must be extensible i.e. we may need to add a new requirement later.
3. Unit tests should be present.

### How to share your solution with us?

1. Create a public github repository.
2. Commit your code there with a README file describing how to run the application.
3. Share the repository url with us.

### Note:

- You are free to expand/extend the scope of the project.
- Use latest Java to develop the solution
- Develop using Spring Boot framework
- Include all necessary framework/artifacts like DB, Objects, Messaging, Caching, etc.,
- Emphasis will be given to your coding practices & approach
- You can even include front end/UI pages as well.