Backend Problem Statement

1. There are Users and Blogs, and any user can comment on any blog.
2. Create a database with sample data, use the database of your choice.
3. Consider all users who have commented on the same blog as friends (1st level friend).
4. A friend is 2nd level friend if he has commented on a blog where a 1st level friend has also commented but has not commented on any common blog.
5. Example - Blog1 has the comment of {User1, User2}, Blog2 has the comment of {User1, User3} here User2 and User3 are 2nd level friend if there exists no blog which has the comment of User2 and User3.
6. Similar to above there can be third level friend and k-th level friend (LinkedIn shows this kind of friend level)
7. Create a REST api GET /users/<userId>/level/<levelNo> which should give list of all friends of that level for given userId (ex- /users/1234/level/1 for first level friend)
8. Use high standard design principles while implementing the solution
9. Write modular and clean code with comments keeping in mind scalability and manageability of code.

Judging criteria will be

1. Quality of the solution
2. Quality of code
3. Bonus if you use nodeJs, mongoDb, ReactJs, and Graphql

Submission:

1. Github link with steps to run and execute the code
2. A readme on the github explaining the approach taken