You're a backend developer at Newsifier and you've been asked to build a Karma user ranking API to be used by our front end and mobile developers. Each user in our database has a karma score, the higher the karma score they have, the better ranking position they get. Usually, they get a higher karma score by commenting and receiving likes from other users.

Now, given the fact that we have these 2 tables:

users (
 `id`: Primary key
 `username`: Unique
 `karma_score`: positive integer, default is 0.
 `image_id`: foreign key.
)

images (
 `id`: primary key
 `url`: string
)

We would like to get the **overall user position** compared to all users depending on the karma score, in addition to the 2 users right before him and the 2 users right after him.

For example:

Users tables (example of the table the DB, sorted by id ASC)

Id	username	karma_score	image_id
1	troglecartier	4202	1
2	towpocket	5000	2
3	snorklingbinnacle	5500	3
4	fitwhatever	4201	4
5	singeofficer	3203	5
6	perhapsupstage	4205	6
7	chokefiery	4200	7
8	artistmath	4204	8

So if you query the API for the user of id 1 which his karma score 4202 and his overall position is 5.

GET **api/v1/user/1/karma-position** it should return something like

```
[
        // The 2 users who are in the higher ranking than the user 1 and right before him.
        {
           id:  6,
           position: 3,
           karma_score: 4205,
           Image_url: https://….
        },
        {
           id:  8,
           position: 4,
           karma_score: 4204,
           Image_url: https://….

        },
        // overall user position is 5 out of 8
        {
           id:  1,
           position: 5
           karma_score: 4202
           Image_url: https://….

        },
        //  The 2 users who are lower in ranking to the user 1 and right after him.
        {
           id:  4,
           position: 6
           karma_score: 4201
           Image_url: https://….

        },
        {
           id:  7,
           position: 7
           karma_score: 4200
           Image_url: https://….
        },

]
```

You will be scored to the following criteria:

- Create the database using MySQL 8 (you are not allowed to change the DB table structure)
- Write the API endpoint that accepts any user id and it returns his overall position plus the next 2 higher users in the ranking and the next 2 lower users in the ranking.

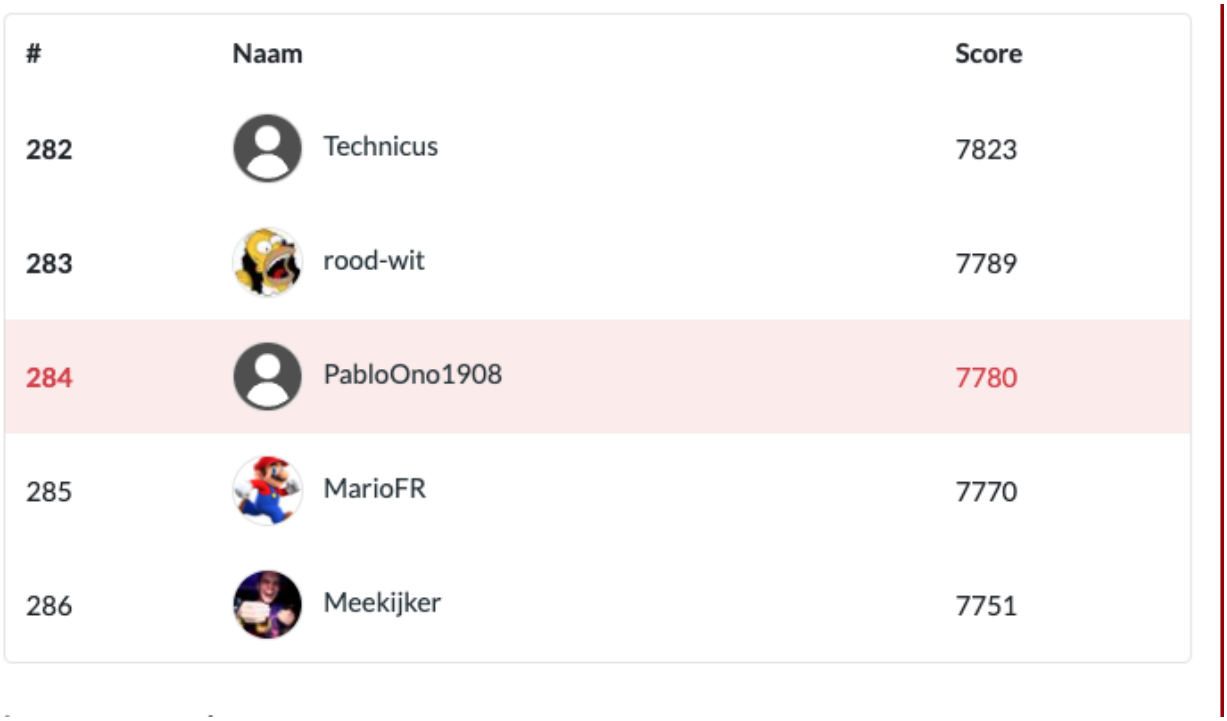
  The user has the highest karma score in the table has the position 1 and the lowest karma score has the last position.
- For each user, you need to return the image URL instead of the image_id.
- API is public, no need for authentication.
- Write a function that fills the database with fake users and random scores, at least 100,000 users.
- Make sure the queries are fast and optimized. They need to work fast on a large scale of data.
- You can use Php Laravel OR Node js express for the backend.
- Please push your code to any public git repository, it should include the database scheme, a readme file on how to run the project, and any notes that are helpful for the reviewer.

Bonus (not required but will have a higher grade if you do it)

- Always show 5 users, if the requested user has the position 1, show him first, and show the next 4 lower users in ranking, also if the user is the lowest in ranking, show the user at the end of the list and the next 4 users are higher than him in the ranking.
- Write an automated test to check your API results.
- Use the API to render a simple HTML table that shows the user ranking
- Make the number of users being returned instead of 5 customizable from the API endpoint.

Any questions regarding the test project feel free to email kamal@newsifier.com

Screenshot from the live feature.



#	Naam	Score
282	Technicus	7823
283	rood-wit	7789
284	PabloOno1908	7780
285	MarioFR	7770
286	Meekijker	7751