

# SCALE FOR PROJECT FT\_TRANSCENDENCE (/PROJECTS/FT\_TRANSCENDENCE)

You should evaluate 3 students in this team



Git repository

git@vogsphere-v2.codam.nl:vogsphere/intra-uuid-a8dbf4bb-bee1-4f70-a5

## Introduction


Please comply with the following rules:

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.
- Identify with the student or group whose work is evaluated the possible dysfunctions in their project. Take the time to discuss and debate the problems that may have been identified.
- You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. The pedagogy is useful only and only if the peer-evaluation is done seriously.

## Guidelines

- Only grade the work that was turned in the Git repository of the evaluated student or group.
- Double-check that the Git repository belongs to the student(s). Ensure that the project is the one expected. Also, check that 'git clone' is used in an empty folder.
- Check carefully that no malicious aliases was used to fool you and make you evaluate something that is not the content of the official repository.
- To avoid any surprises and if applicable, review together any scripts used to facilitate the grading (scripts for testing or automation).
- If you have not completed the assignment you are going to evaluate, you have to read the entire subject prior to starting the evaluation process.
- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, and so forth.  
In these cases, the evaluation process ends and the final grade is 0, or -42 in case of cheating. However, except for cheating, student are strongly encouraged to review together the work that was turned in, in order to identify any mistakes that shouldn't be repeated in the future.
- Remember that for the duration of the defence, no segfault, no other unexpected, premature, uncontrolled or unexpected termination of the program, else the final grade is 0. Use the appropriate flag.  
You should never have to edit any file except the configuration file if it exists. If you want to edit a file, take the time to explicit the reasons with the evaluated student and make sure both of you are okay with this.
- You must also verify the absence of memory leaks. Any memory allocated on the heap must be properly freed before the end of execution.  
You are allowed to use any of the different tools available on the computer, such as leaks, valgrind, or e\_fence. In case of memory leaks, tick the appropriate flag.

## Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/80444/en.subject.pdf>)

## General instructions

### Preliminary tests

- Any credentials, API keys, environment variables must be set inside a .env file during the evaluation. In case any credentials, API keys are available in the git repository and outside of the .env file created during the evaluation, the evaluation stops and the mark is 0.
- Ensure the docker compose file is at the root of the repository.
- Run the "docker-compose up --build" command.
- Since the rating of this project is more flexible, do not stop the evaluation process unless you encounter a 500 error, a crash, or anything that actually doesn't work within the project scope.

☒ Yes

☐ No

### Backend

- The backend must be developed using the NestJS framework.
- The database must be a PostgreSQL database.
- During the whole evaluation process, there must be no unhandled warning or error.

☒ Yes

☐ No

### Frontend

- The frontend must be done using a TypeScript framework.
- Any TypeScript/JavaScript library is allowed.
- During the whole evaluation process, there must be no unhandled warning or error.

☒ Yes

☐ No

### Basic checks

- The website is available at the address chosen by the students.
- The user can login using the 42 intranet OAuth feature.
- When logged for the first time, the user is prompted to add information to their account (display name/nickname, avatar, and so forth).
- If not logged, the user has access to only little or no information and is prompted to sign in.
- The website is a Single Page Application. The user can use the "Back" and "Forward" buttons of the web browser.
- You can browse the website using the latest version of Chrome and one additional browser without encountering any problems or errors.

☒ Yes

☐ No

## The website

### Security concerns

Ensure that the website is secured. Check the database to verify that passwords are hashed. Check the server for server-side validation/sanitization on forms and any user input. If this isn't done, the evaluation ends now.

☒ Yes

☐ No

### User profile - private

When logged in, the user has access to their profile where they can edit their information. For instance, they can change their nickname (which must be unique) or their avatar (which is a default avatar if not set).

✓ Yes

✗ No

#### User profile - public

Users can see the profile of other users. A profile contains basic informations such as their nickname, their avatar, or a button to add them as friends.

User can block other users. This means they won't receive private messages from the accounts they blocked nor view their messages in public/private channels.

✓ Yes

✗ No

#### Friend interface

The user has access to a friends interface, where they can see their friends and their status (offline/online/in a game/and so forth). They also have access to basics informations about them (name/nickname, avatar and so forth).

✓ Yes

✗ No

#### 2FA

The user can enable/disable 2FA (two-factor authentication). If enabled, they must pass it in order to sign in. For example, 2FA can use Google Authenticator, a text message, an email, and so forth.

✓ Yes

✗ No

## Chat interface

#### Join/leave channels

A logged in user can access the website chat service. Joining/leaving channels is a manual action. For example, this means it must not be done on logout (the user has to click a "Leave channel" button or something else). The user can join channels (that can be already created) to have a chat. Some of them can be password-protected. If so, the user has to enter the correct password in order to join the channel.

✓ Yes

✗ No

#### Chat usage

The users can chat. Messages must be sent/received instantly. If the user blocked another user, the messages from the blocked person must be hidden. The user can access the user profile of other players from the chat interface and also invite them for a Pong duel.

✓ Yes

✗ No

#### Creating channels

The user can create new channels. The channel creator is set as the channel owner and has basic moderations rights (ban/mute users, add a password to protect the channel, set new administrators, and so forth).

✓ Yes

✗ No

#### Channel roles

A user who is an owner of a channel can kick, ban, mute other users and the channel administrators.

A user who is an administrator of a channel can kick, ban mute other users, but not the channel owners.

✓ Yes

✗ No

## The game

### Matchmaking system

When logged in, the user has access to a matchmaking system so they can play Pong 1v1 games versus other players on the website. When they get matched, a new game is loaded and the two users can start playing.

✓ Yes

✗ No

### Gameplay

The game itself must be playable and respect the original Pong game. The controls must be intuitive or correctly explained (with some rules or manual). When a game is over, either a kind of end-game screen is displayed or the game page just exits.

✓ Yes

✗ No

### Lags & disconnects

Unexpected disconnections and lags have to be handled. The game and the website must not crash when a user is experiencing lags or is disconnected. Handling such issues in an efficient way is appreciated but not mandatory:

- Pause the game for a defined duration.
- Disconnected users can reconnect.
- Lagging users can catch up to the match.
- And so forth. Any solution is acceptable. The only requirement is: the game should not crash.

✓ Yes

✗ No

### Additional features

The user can enjoy extra features such as power-ups, different maps, achievements, and so forth.

✓ Yes

✗ No

## Ratings

Don't forget to check the flag corresponding to the defense

✓ Ok

★ Outstanding project

Empty work

📁 Incomplete work

📄 Cheat

💥 Crash

👤 Incomplete group

⚠ Concerning situation

🚫 Forbidden function

## Conclusion

Leave a comment on this evaluation

Finish evaluation