

General instructions

Preliminary tests

- 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, Firefox or Safari 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 encrypted. 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

The website

Security concerns

Ensure that the website is secured. Check the database to verify that passwords are encrypted. 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

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

Spectator mode

There is a spectator mode. The user can watch any live games. They can be accessed through the chat interface or the friend interface. There can also be a page dedicated to live games from which the user can access any of them.

✓ 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

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