


(https://profile.intra.42.fr)

SCALE FOR PROJECT PISCINE PYTHON DJANGO (/PROJECTS/42CURSUS-PISCINE-PYTHON-DJANGO) / DAY 09 (/PROJECTS/42CURSUS-PISCINE-PYTHON- DJANGO-DAY-09)

You should evaluate 1 student in this team



Git repository

git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-c1119ae2-6b91-41f1-8000-000000000000 

Introduction

For the smooth running of this evaluation, please respect the following rules:

- Remain polite, kind, respectful and constructive whatever happens during this conversation. It's a matter of confidence between you and the 42 community.
- Highlight the potential problems you 've had with the work you're presented to the person or the group you're grading, and take the time to talk about and discuss those issues.
- Accept the fact that the exam subject or required functions might lead to different interpretations. Listen to your discussion partner's perspective with an open mind (are they right or wrong ?) and grade them as fairly as possible.
42's teaching methods can make sense only if peer-evaluation is taken seriously.

Guidelines

- You must only evaluate what you will find in the student's or group's GiT repository.
- Take the time to check that the GiT repository matches the student or group and the project.
- Double check that no malicious alias was used to mislead you and make you grade something different from the official repository content.
- If a script supposed to help evaluate the exam is supplied by either side, the other side will have to strictly check it to avoid nasty surprises.
- If the evaluating student has not yet taken this project, they will have to read the exam subject in its entirety before starting the evaluation.
- Use the flags available on this grading system to signal an empty or non-functional project, a norm flaw, cheating, etc. In that case, evaluation stops and final grade is 0 (or -42 if it's a cheating problem). However, if it's

not a cheating problem, you are invited to keep talking about the work that has been done (or not done, as a matter of fact) in order to identify the issues that lead to this stalemate and avoid it next time.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/22321/en.subject.pdf>)

Foreword

Observing the rules

- The repo contains the evaluated student's or group's work.
- The evaluated student or group can explain their work anytime during the evaluation.
- General and specific instructions of the day will be observed during the whole evaluation.

 Yes

 No

Python-Django D09 Training

- Hard typing the result of an exercise in the code invalidates this exercise. - You must test the exercises with Python3. If one exercise doesn't work with Python3, it's considered invalid, even if it works with Python2. - Every instruction requiring to type a URL consider that the development server of the project concerned by the exercise is launched with the `./manage.py runserver` command. - `jquery` is the only authorized frontend library. - The repo must contain a 'requirement.txt' file allowing the install of the necessary libraries with the command: `mktmpenv -p python3 -r requirement.txt --no-cd`, for instance. (The `virtualenvwrapper` must be installed and configured for it to work)

Exercise 00 - Login with Ajax

The goal of this exercise was to create an authentication system running thanks to AJAX.

If you need it, create an administrator account with the `./manage.py createsuperuser` command.

- The page should never refresh by itself during this evaluation.
- If you're not connected, the `127.0.0.1:8000/account` url must display a standard login form.
- If you misfill this form, errors should be displayed.
- If you fill the form correctly, the form must disappear and a "logout" button must appear, along with the text "Logged as ".
being the name you logged with.
- Close the page and reopen it. You still must be logged in and the login form must not appear.
- Check the form works connecting yourself as an administrator and heading to the admin page. You should not have to log in.
- Clicking "Logout", the form must appear as the rest disappears. Head to the admin page again. This time, you should have to log in.

- If all the conditions are met, this exercise is validated.

 Yes

 No

Exercise 01 - Chat in Websocket

The goal of this series of exercises was to build a functional chat thanks to websockets.

Access the address the student has assigned to the chatroom lists.

- You should see three links leading to three different chatrooms.

Make sure all three chatrooms behave as follows:

- The access should be impossible with logging in.
- The page should never refresh by itself.
- The name of the chatroom must appear somewhere.
- You can send messages. They appear one after the other, top-down, along with the name of the users who posted them. A message must not disappear or must be accessible thanks to a scrollbar if it does.
- Log in with two (or several) different users to the same chatroom. The first one must see ' has joined the chat' with their own , and another user with the of the other user.
- When one logged in user or the other types a message, every user logged to the chatroom can see it. Connect the users to two different chatrooms and make sure this time one can't see the message of the other.

If all the conditions are met, this exercise is validated.

 Yes

 No

Exercise 02 - Message History with Websockets

When you join a chatroom, a history of the last three messages posted in this chatroom appears, top-down, from the oldest to the newest.

If so, the exercise is validated.

 Yes

 No

Exercise 03 - User list with Websockets

- A chatroom page should never refresh by itself.

Join the same chatroom with two different users.

- The list of users currently connected, including you, must appear.
- Leave and return in this chatroom with one of the users. The other one must see the userlist automatically updated.
- When a user leaves the chatroom , ' has left the chat' must

appear after the list of the posted messages. Of course, is the name of the user who just left.

If all the conditions are met, this exercise is validated.

☒ Yes

☐ No

Exercise 04 - Pretty Chat

Join a chatroom:

- The message list must appear in a fixed size 'container'.
- Messages exceeding the 'container' must 'disappear' while remaining available via a scroll bar.
- The scroll bar must always be set to the bottom so the most recent messages always show.

If all the conditions are met, this exercise is validated.

☒ Yes

☐ No

Ratings

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

☒ Ok

☐ Empty work

☐ No author file

☐ Invalid compilation

☐ Norme

☐ Cheat

☐ Crash

☐ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

Terms of use for video surveillance
(<https://signin.intra.42.fr/legal/terms/1>)

Rules of procedure
(<https://signin.intra.42.fr/legal/terms/4>)

Declaration on the use of cookies
(<https://signin.intra.42.fr/legal/terms/2>)

General term of use of the site
(<https://signin.intra.42.fr/legal/terms/6>)

Legal notices
(<https://signin.intra.42.fr/legal/terms/3>)

Privacy policy
(<https://signin.intra.42.fr/legal/terms/5>)