

https://projects.intra.42.fr/scale_teams/3296748/edit#<https://profile.intra.42.fr/><https://profile.intra.42.fr/searches>

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT PISCINE PYTHON DJANGO (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/42CURSUS- PISCINE-PYTHON-DJANGO) / DAY 07 (HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/42CURSUS- PISCINE-PYTHON-DJANGO-DAY-07)

You should evaluate 1 student in this team



Git repository

`git@vogsphere-v2.s19.be:vogsphere/intra-uuid-5b66701a-afba-41ed-bcd1-`



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/24861/en.subject.pdf>)
- ☐ d07.tar.gz (<https://projects.intra.42.fr/uploads/document/document/3956/d07.tar.gz>)

Foreword

description

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 Training D07

- The repo must be a single Django project. - Implementing any "hard" URL is prohibited. In the templates, the URLs must use the 'URL' template tag. In the views, they must use the reverse or reverse_lazy functions. - A view cannot be implemented as a function. - English will be the default language.

Exercise 00 -

- Check that the models are implemented as required in the subject.
- Check that the Articles, Home and Login views are classes that inherit the generic views. None of them must directly be inheriting from 'View'.
- Home functionality must redirect to Article. Its URL must be 127.0.0.1:8000.
- The Article functionality must display an HTML table that contains at least 5 different articles from 3 different users. For each

article, every field - except the 'content' one - must be displayed.

- Login displays a form requiring a username and a password. If the form is filled correctly, you must be redirected to Home (and logically, Article). If the form is invalid, you must be redirected to the Login URL and error message must explain what happened.

If one test fails, the exercise is invalid. Go to the next one.

☐ Yes

☐ No

Exercise 01 -

- Check that the views of the Publications, Detail, Logout and Favourites functionalities are classes inheriting from generic views. None of them must directly be inheriting from 'View'.

- The Publications functionality must display an HTML table presenting the fields 'title', 'synopsis', and 'created' of every article recorded in the 'Article' model whose author is the current logged user, as well as the link leading to this article's Detail functionality.

- Click on the link listed above. You must reach a page showing every field of the listed article.

- The Logout functionality must log you out and redirect you to the Home functionality. Check that you're logged out trying to access the Publication functionality. Whichever the result (crash, redirection, empty page), you must not have the same result as the one you had previously.

- The Favourite functionality must display a list of articles titles that are actual links to the articles' details. These articles are the ones saved as favorite by the current user. The student must have provided one user with at least 2 different favorite articles with their work.

If one test fails, the exercise is invalid. Go to the next one.

☐ Yes

☐ No

Exercise 02 -

- Check that the Register, Publish And Add to favourite functionalities are classes that inherit from CreateView.

- Register must display a POST type form allowing to register. Try to create a new user and then try to log with this user's name and password.

- Publish must display a POST form allowing to type a new article. The Author field must not be displayed, it must be filled when validating. Enter a new article and check it does appear in the table of the Articles functionality with the right author.

- Add to Favourite must be a hidden form located in the detail page of

an article. You must only see a button that allows to submit it. Try to put an article that not already in the favorite list, in the favorite list. It must appear in the list of the Favorites functionality, being registered with the same user.

If one test fails, the exercise is invalid. Go to the next one.

☐ Yes

☐ No

Exercise 03 -

- Check that a menu is present and displayed on every page of the site.
- Check that it changes according to your log status.
- Thanks to the subject it holds, check that all the required links and only these links lead to the right functionalities.
- Check that from any page, an invalid form does lead to a page indicating the form's errors.
- On the Article functionality, the HTML table must be sorted out by creation date from the newest to the oldest.
- On the Article functionality, an abstract that's more than 20 characters is cut off. The following part is replaced by suspension points.
- On the Article functionality, an additional column shows how long the article has been published for.

If one test fails, the exercise is invalid. Go to the next one..

☐ Yes

☐ No

Exercise 04 -

- Check that the menu is formatted like the one visible on the image provided with today's resources.

If the test fails, the exercise is invalid. Go to the next one.

☐ Yes

☐ No

Exercise 05 -

- Check that the default URL contains an 'en' prefix.
- Check that a button allowing to switch language is present on the Article functionality.
- Check that all the display of the Article functionality is translated - that includes the menu and the HTML table header - whether you're

logged or not.

If one test fails, the exercise is invalid. Go to the next one.

☐ Yes

☐ No

Exercise 06 -

- Check that it is impossible to access the Favourites, Publications and Publish functionalities without being logged.

- Check that it is not possible to access the Register functionality being logged.

- Check that it's impossible to set the same article twice in the favorite list.

- Check that all the test files are present and that they do simulate the required behaviors.

- Check that the tests simulate the behaviors listed above and they all return 'OK' when you run them.

If one test fails, the exercise is invalid. Go to the next one.

☐ Yes

☐ No

Ratings

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

☐ Ok

☐ Empty work

☐ No author file

☐ W Invalid compilation

☐ Norme

☐ Cheat

☐ d Crash

☐ I Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

terms & conditions (<https://signin.intra.42.fr/legal>)