SCALE FOR PROJECT PISCINE RUBY ON RAILS (/PROJECTS/42CURSUS-PISCINE-RUBY-ON-RAILS) / DAY 07 (/PROJECTS/42CURSUS-PISCINE-RUBY-ON-RAILS-DAY-07)

You should evaluate 1 student in this team

★

Git repository

`git@vogsphere.msk.21-school.ru:vogsphere/intra-uuid-168c8`   📋

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-funcional 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.

- You must stop grading when one exercise is not correct, even if the other ones are.

- The exercise is not valid if ONE requirement is not perfectly met.

Attachments

⬀ subject.pdf (https://cdn.intra.42.fr/pdf/pdf/46609/en.subject.pdf)

⬀ d07.tar.gz (/uploads/document/document/8260/d07.tar.gz)

Foreword

This section is dedicated to the evaluation start and the checking of prerequisites. It's not graded, but if something's wrong or a condition is not met, here or anytime during the evaluation, the grade is 0 and a flag can be ticked if necessary.

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.

- Le Gemfile is identical to the one provided in the subject

- The code required by the exercise doesn't use any global variable.

- No error page will be tolerated.

- Stories must be observed. With a proper use of English language, the scenarios are verified.

- The 'bundle install && rubycritic' command must return a score of at

least 89/100.

- General and specific instructions are observed all through the evaluation.

     ⊘ Yes   ✕ No

Formation Ruby Rails D07

- D07 is a practical contextualization. - DRY and KISS concepts are honored here.

ex00 mysql is a bad habit

- The database.yml file does describe a db posgtresql.
- The "rake db:create" command passes flawlessly.

     ⊘ Yes   ✕ No

ex01 You Sir?

- The currency gem is properly installed.
- In a User, the name and bio fields are present and editable.
- Authentication is operational.

     ⊘ Yes   ✕ No

ex02 Get me something to sale

- Products are present and are all linked to a brand (see the subject's example).

- Brands and images havec an image.

- Upload works and doesn't write any subject locally.

- The 2500 products are present and the catalog loads (relatively) fast.
Anyway, it doesn't display a 'full-size' version.

     ⊘ Yes   ✕ No

ex03 Cart

- A cart is dislayed on the catalog and you can add products into it.

- You can also increase or reduce the the quantity of products, because,
of course, the same product doesn't appear twice.

- The price increases with the quantity.

- Of course, Ajax should have been used IRL, but it's not on today's bill.

- There must also be a button to empty the cart, and one to validate the order (both are present and operational).

- The order validation command leads to a page that sums up the order and shows the subtotal.

        ☑ Yes   ✕ No

ex04 One panel to rule them all

- The admin panel is operational, and accessible only if you're logged-in.

- The site's resources are manageable or at least: users, products and brands.

        ☑ Yes   ✕ No

ex05 One account to rule them all

- After the seed is executed, at least one user is admin.

- After the seed is executed, at least one user is a mod.

- Site's resources are only manageable by the admin and the mod.

- Mods can only manage brands and products (edition, creaton, deletion).

- Admins can edit anything.

- Creating a user doesn't grant it any privilege.

- Only admins can attribute roles via the interface.

- You cannot modify, create or delete if you have no role. No path allows it (no url-rewriting is possible).

        ☑ Yes   ✕ No

ex06 One account to rule them all

- The project does belong to the student: the URL contains their login.

- All the functionalities are present. They don't bug.

- Images are displayed.

✓ Yes ✕ No

Ratings

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

✔ Ok ★ Outstanding project

■ Empty work ● No author file ☠ Invalid compilation ♪ Norme ➡ Cheat ♣ Crash

👤 Incomplete group ⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation