



Rush01

Home stretch !

Vincent Montécot vmonteco@student.42.fr

Summary: It's time to rub in a real project!

Contents

I	instructions	2
II	Specific Rules of the day	4
III	Preamble	5
Part IV mandatory		6
IV.1	Base Project.	6
IV.1.1	A con fi HTTP configuration.	6
IV.1.2	An authentication system.	6
IV.1.3	A homepage	7
IV.1.4	Manage user profiles.	7
IV.2	Functionality choice.	8
IV.2.1	Forum.	8
IV.2.2	A messaging system.	8
V	optional part	10
V.0.1	Bonus Forum: implementation of AJAX.	10
V.0.2	messaging Bonus: use websockets.	10
V.0.3	Push notifications.	10
V.0.4	SSL.	11
V.0.5	Package.	11
V.0.6	system votes.	11

Chapter I

Instructions

- Only this page serve as a reference: Do not fi rm the hallway noise.
- The subject can change up to an hour before rendering.
- If no contrary information is explicitly present, you should assume that the versions of languages and frameworks used are (or later):
 - Python 3
 - HTML5
 - CSS 3
 - Javascript EM6
 - Django 1.9
 - psycpg2 2.6
- Unless otherwise indicated in the subject, the fi les in python each year on **Python one (d01, d02 and d03) must have their end block if `__name__ == '__main__':` in order to insert the entry point in the case of a program, or tests in the case of a module.**
- Unless otherwise indicated in the subject, each year days on **Django will have its own application in the project to make for reasons peda- gogiques.**
- The exercises are precisely ordered from simple to more complex. In no event shall we bear no attention or take into account a complex exercise if a simpler exercise is not very successful.
- Attention to the rights of your fi les and your directories.
- You must follow the rendering process for all your exercise: only the present work on your GIT repository will be evaluated in defense.
- Your exercises will be evaluated by your pool mates.
- You should leave in your repertoire no other fi le than those explicitly speci fi ed by the forward drills.
- Unless otherwise specified in the topic you should not include in your rendering:

- The files `__pycache__`.

- Any migration.

Warning, you are still advised to make the file migration / `__init__.py`, it is not necessary, but simplifies the construction of migration. Do not add this file will not invalidate your rendering but you *must* be able to manage migration for correction in this case.

- The file created by the command `collectstatic` of `manage.py` (with the way the value of the variable `STATIC_ROOT`).

- The file Python bytecode (files with extension. `.pyc`).

- database files (especially with `sqlite`).

- Any file or folder must or can be created by the normal behavior of the rendering work.

It is recommended that you modify your. `gitignore` in order to avoid accidents.

- When you need to get an accurate output in your programs, it is of course forbidden to a ffi expensive output precalculated instead of performing the exercise cor- rectly.
- You have a question ? Ask your neighbor to the right. Otherwise, try your left neighbor.
- Your reference manual called Google / man / Internet /
- Think discuss on the forum of your pool Intra!
- Read the examples carefully. They might require things that are not otherwise specified in the subject ...
- Please, by Thor and Odin! Re fl échissez dammit!

chapter II

Specific Rules of the day

- You can use it to this day-the following packages (over your own packages to build and install when correcting):

```
asgi-repeat == asgiref
0.14.1 0.14.0 0.16.0 autobahn
== channels 0.17.3 0.15.0
daphne == Django 1.9.5

django-bootstrap3 == 7.1.0 gunicorn
19.6.0 msgpack == python-0.4.8 Pillow
== repeat 3.4.1 2.10.5 1.10.0 Twisted
six == 2.5 16.4.1 txao .1 uwsgi ==
== 4.3.2 2.0.14 zope.interface
```

- You can also use JQuery and Nginx as you like.
- You are free to organize your project as you wish.
- Remember your file dependencies pip.
- When correcting the correction will completely alter the naviga- gateur side code to try to put in your rendering failure. You should take this into consideration to make your project sufficiently robust.
- Your rendering will be cloned with the command `git clone <repo> ~/rush01`. So you have to set your project runs at that location.

Chapter III

Preamble

Here is a selection of Ig Nobel prize awarded in recent years:

2013 odds Price : Bert Tolkamp Mary Haskell, Fritha Langford, David Roberts and Colin Morgan, for discovering that the probability that a cow rises increases with the time during which it remains lying, and it is almost impossible to otherwise predict when it recouchera.

Prize for Peace in 2013 : Alexander Lukashenko, President of Belarus to making illegal applause in the public and the Belarusian police for arresting a unimanchiste accused of this crime.

2013 Psychology Price : Laurent Begue, Brad Bushman, Oulmann Zerhouni, Baptiste Subra and Medhi Ourabah to have experimentally confirmed that people thought to be drunk also think is attractive.

Physics Award 2014 : to Kiyoshi Mabuchi, Kensei Tanaka Daichi Uchijima and Rina Sakai studying the coefficient of friction between soil, a banana peel and a shoe sole.

Neuroscience Prize 2014 : to Jiangang Liu Jun Li, Lu Feng, Ling Li, Tian Jie Kang and Lee, for trying to understand how some people see the face of Jesus in a piece of toast.

Physics Award 2015 : Patricia Hu Yang and David for showing that mammalian more than 3kg emptied their bladder 21s (+/- 13s).

2015 Biology Price : Bruno Grossi, Omar Larach Mauricio Canals Rodrigo A. Vásquez and José Iriarte-Díaz, having seen a chicken with a baton fixed rump adopted a similar approach to the one that have probably had non-avian dinosaurs.

2015 Literature Prize : Mark Dingemanse Francisco Torreira and Nick J. Infield for discovering that the word "huh" existed anywhere without one really knows why.

chapter IV

mandatory part

You have arrived at this second rush that will close two weeks of swimming. This is an opportunity for you to demonstrate your skills with Django through the realization of an intranet.

IV.1 Base Project

You must first perform a basic project with Django with several parts, among which:

IV.1.1 A configuration HTTP

A configuration server to serve your project with Nginx (You should *NOT* use the development server, *DEBUG MUST* be at False).

All features should work with this configuration. (If a functionality works with `./manage.py runserver` but not with Nginx, this functionality is invalid).

IV.1.2 An authentication system

Will only be available offline:

- A registration page.

The form on this page must include a field for the user name and two fields for the password (which should not appear when typed). The validation will require that the user name is not already used and the values entered for the password are the same.

Once the validation is effected, a user is created with the input and active data. The visitor is then connected with this user is redirected to the home page.



The password should obviously not be stored unencrypted in database.

- A login page.

The form on this page must include a field for the user name and a field for the password (which should not appear when typed). The validation will require the user exists, and that the authentication with him is possible.

Once the form is validated, the visitor is connected with this user is redirected to the home page.

Once connected, the user must be redirected to the homepage. Once connected, these pages are no longer accessible. The user will then have access to a disconnect button.

IV.1.3 A homepage

A home page to access all the features that you can implement.

IV.1.4 Manage user profiles

You must associate each user some information among which:

- His name.
- His first name.
- His email.
- Its description.
- His picture of profile.

You must provide on the page profile of each user the option (for this user and administrator) to view and edit the information.

Administrators must also be able to assign or remove a user administrative rights (It must be able to know whether a given user has administrative rights).

An administrator can very well be a superuser, a user with a set of permissions or part of a group. At your convenience.

IV.2 Functionality choice

Once this project is, you will implement at least one of the following features (if you do both you extra points).

IV.2.1 Forum

This forum, although simple, is to implement a system of posts. These posts will be listed by 10 on a page (bring up a paging system when the posts are too many).

Links to the detail of the listed posts will be present.

A post should have a title, author, content, date and time of creation and a set of comments that will be associated with it.

The comments have content, date and time of creation as well as an author.

The detail of a post affords all information of this post and all the comments (and their respective information as well).

It shall be possible to comment on a post as it will be possible to comment Comment (Your comment will be organized as a tree). The affords change indicating clearly what a comment is a comment (post or this or that comment).

All users must have the ability to create posts and comment.

IV.2.2 A messaging system

Your mail system will allow a user to hold conversations with other users by exchanging messages.

A conversation between two users will always *different* (And the content of that conversation is accessible only to those users only).

Your mail system will include two pages:

- A page listing conversations *started* involving the current user. You need to display the last message of information received / sent (date / time, content truncated to 30 characters, party ...) for each conversation. Afford by 10 and bring up a paging system if the messages are too many to be affords on a single page.

also to implement each conversation listed a link to the detail of it.

Also add a link to start a conversation with a user *different* on his page of profile (If the conversation does not exist).

- The details of each conversation.

You will have to list all the messages exchanged by chronological order with their contents and the date and time of issue. The interface must also clearly indicate whether a message was sent or received. The user with whom the conversation is held must also appear.

chapter V

optional part

Once the mandatory part performed, you can achieve bonus among the following:

V.0.1 Bonus Forum: implementation of AJAX

Refresh this page should not be necessary for a form for commentary, fill, submit it and see the changes was expensive after validation by the server, and repeat.

Of course, this bonus is only feasible if you realized the forum.

V.0.2 messaging Bonus: use websockets

Using websockets in your messaging system:

The messages you are sent by another user should appear instantaneous in the thread conversation without having to reload the page.

Of course, this bonus is only feasible if you have implemented a messaging system.

V.0.3 Push notifications

You must, using websockets, implement a system of push notifications, two cases may arise:

- You must show a notification when a new post is created if you did the week. This notification must at least bring up the post title truncated to 30 characters.
- You must show a notification when a new message is sent to you if you realized the mail system. You need at least a form

the name of the sender and the message truncated to 30 characters. If you have validated both the messaging system and the forum, treat only one of the two cases sufficient to validate this bonus.

V.0.4 SSL

Use your site to HTTPS and use your potential websockets in WSS with a key and an SSL certificate.

While accessing your site via HTTP, redirection to the HTTPS version is automatically effectuée.

V.0.5 Package

Make one of your application as a package that will be built and installed with pip when correcting.

V.0.6 System votes

Implement a polling system to submit elements of your site users vote.

These votes will be positive votes (upvotes) or negative (downvotes). This feature will allow a

change of some information:

- voting buttons (for downvoter and upvoter the element).
- A summary of the votes (positive and negative) for this element.
- The current vote of the user for this element (upvote, downvote or no vote) A user can vote only once on an item and can not simultaneously upvoter and downvoter this element.

If the user clicks a second time on the same vote button, the previous vote is simply canceled.

If the user clicks an item for downvoter after upvote, downvote and the item is the previous upvote is canceled, the reverse is also true.

Once the click a button that is effectuée, information are updated on a change.

Once your done voting system, test it on your forum (On posts and comments). Check that having more element to be voted on the same

page does not disturb the different voting buttons, and each behaves correctly (eg click a button to upvote not upvote all page elements).