



D06 - Authentication and users.

Site Project "Life Pro Tips".

Vincent Montécot vmonteco@student.42.fr

Summary: Where how to become a black belt permissions and user management with a simple Django project.

Contents

I	Preamble	2
I.1	TREATY From the art of making just hypées addresses.	2
I.1.1	The aornées die beautifully.	2
II	instructions	3
III	00 year	5
IV	Exercise 01	6
V	exercise 02	9
VI	exercise 03	10
VII	exercise 04	12
VIII	Exercise 05	13
Exercise IX	06	14

Chapter I

Preamble

Here are a few lines from a book talking artisanal manufacture of IP addresses to the former:

I.1 TREATY From the art of making just hypées Address

As this will be found to written real art & infrequently exposed, its manufacture tion of hypées addresses, as the ancients taught in the workshops of the Great Vinton. Would to God that every one knew cestus just beautiful & much neglected subject, producing addresses if messy to use in trade in then.

Souventement we read and even manufacturing will practice both hated modern addresses. From the latter does not traictée ycelieu, for what it pro- few addresses Duit use, which wither and rot in no time as chiabrenas of virgins. Those explained in Vices pages haulte ma- SPECT and profound science, say real hypées are only able to withstand fl ancient scourge, Faillibus-aulniem, Altus-fi droopium epithelium. We must adoner songeusment both studie and under parfaicte workers in the preparation of ingredients and tools, easily save to pro- duce acute addresses, subtle, deep and seraines, and beautifully proportioned to achieve the highest degree exchanges.

I.1.1 The die beautifully aornées

As the faid saige Solomon, Wisdom points entrant in malivole soul, it is an important part of the influence of the manufacturer on manufactured like a mirror representing the most haulte virtue. So were it not that the strongest and healthiest workers who can complete that business and consume.

You can find the rest of this healthy reading [right here](#) .

Chapter II

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 III

Exercise 00

	Exercise :
00 Exercise: anonymous sessions.	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django, random, django-bootstrap3	
Remarks: n / A	


You start here a website project to provide users to post "Life Pro Tips".

After creating your project, an application and set up a homepage accessible to the URL /, implement the following sessions of anonymous system:

- A user visiting your site should be assigned a username. This name is randomly selected from a list of 10 names that you must define in the file `settings.py`. Your project has a duration of 42 seconds after its definition.
- This name must be persistent and remain the same for 42 seconds, once this period has elapsed, the name must be redefined in the same way as before.
- Highlight the behavior and implemented by adding a message in a `< nav>` top of page. This message will be just as "Hello user" in replacement by the name randomly assigned to visitors until the end of validity of the name in question.
- The name should appear immediately on the pages visited: If you need further update once the page so that it appears or is changed is that the exercise is not successful!

Chapter IV

Exercise 01

	Exercise :
01 Exercise: Creating users.	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django, django-bootstrap3	
Remarks: n / A	

Now that the system of anonymous session is added, it is time to implement user management with the creation of an authentication interface.

You will realize this interface in several stages in this exercise:

- Start by adding a registration page and a login page. After a link to each page in your element nav. Add to it the necessary URLs, templates for you, views, etc ... Add the site name at the beginning of your nav and make it a link to the home page.
- Create forms to insert in these pages with the specific following cities:
 - Registration Form :
 - You will need to insert three fields different including:
 - A field for the user name.
 - A field for the password.
 - A field for the verification password.
 - The validation shall be effected if every field is filled, the informed name does not already exist beforehand and the password fields

pass 2 will have identical values.

- Login Form:
 - You will need to insert two fields different including:
 - A field for the user name.
 - A field for the password.
 - The validation will not 'Effectuer' if every field is filled, and the torque user / password filled exists in your database and allows an authentication.
- Now implement the behavior of pages:
 - If data validation:
 - registration:** Creation and activation of a new user with the informations provided. Then connect to that user and redirect to the home page.
 - Login:** Login with the corresponding user information in-Trees, and then redirect to the home page.
 - In case of non-validation of the data:

In all cases, you must first sheave ffi the page with the form containing the information previously entered (except passwords).


 - registration:** A ffi chage of one (or more) messages (s) explaining error the error (Required field, username already used, dif- ferent passwords)
 - Login:** A ffi chage an error message with the form (field re- quired or incorrect connection information).
- To fi ne when the user is logged:
 - links to registration and login pages are replaced with a logout link (obviously functional) that disconnects the visitors and redirects to the homepage once it clicked.
 - the name of the anonymous session in element nav is replaced with the name of the user.
 - visitors should no longer have access to the registration and login pages, if he tries to access it, it is redirected to the home page.
- Obviously, passwords should be hidden when they are typed.



In another exercise, you have to make your own user class to replace that provided by default. So you would do well (but not obliged) to anticipate this change in your project does not have to manually rewrite each call to your user class in your code.

Chapter V

Exercise 02

	Exercise: Exercise 02:
Our Tips!	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django, django-bootstrap3.	
Remarks: n / A	

It is time to implement the main feature of this project: A tuces as- system!

For this you will need to design a Model Tip with:

- A field contents ... with the contents of the tip (obviously as text).
- A field author (the user who created the tip).
- A field Date (the date and time of creation).


The tips will be listed on the home page and all the attributes mentioned in this statement should appear.

Once connected, the user must have the ability to create a Tip formulated with a lar on the home page (use ModelForm!)

If the user is not connected, the form should not appear and the visitor is not able to create tip.

Chapter VI

Exercise 03

	Exercise: Exercise
03: Votes	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django, django-bootstrap3.	
Remarks: n / A	

Tips Now that our system is in place, it would be nice to identify the best: It is time to put in place a system upvotes and downvotes!

For this, you will alter your Model Tip in order to implement these fonctionnali- tees.

You will also need to implement a way to remove the tips that we con- dérerà not as interesting.

You need to add the tips listed on the home page:

- A delete button
- A button upvote
- A button downvote
- The number of upvotes
- The number of downvotes.

For now, the only restriction will implement the need to be connected to e ff ectuer shares voting or deletion. If the user is not logged in, none of these actions will be possible.

Each of these actions reloads the page. Obviously, a Tip freshly created will have 0 upvotes and 0

downvotes.

Warning :

- It should not be possible to upvote and downvote along the same tip.
- A user can upvote or downvote once on a tip. Re-click on the button simply cancels the downvote or upvote.
- If a user downvotes a tip after having upvoted, the upvote is canceled in addition to the downvote (and vice versa).




Pay particular attention to the implementation of your upvote system. An inconsistency, an imperfection (e.g., a number of votes not lowering after cancellation of the vote) invalidates the exercise. Consider using ManyToMany fields rather than "counters" to implement the voting system.

At this point, every action must only work if the visitor is logged.

You cannot disable or make expensive items unusable by the user.

Chapter VII

Exercise 04

	Exercise :
Exercise 04: First use of permissions.	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django, django-bootstrap3.	
Remarks: n / A	

Now that our project is a bit more complete we realize that anyone-can do anything provided it is connected. It is probably time to add some restrictions.

The user must now have permission to delete ... delete Tips for tips.


Enable the default administrative interface included with Django and use a su- perutilisateur to alter the permissions and thus highlight the behavior of your project.

Obviously, remove the "Remove" button on the page is not su ffi health (although you can). To you to find the right way to restrict this functionality by checking the permissions correctly.

Note: Obviously, the author of a tip can remove the tip even if he does not au torisation to remove in general. This is the only exception.

Chapter VIII

Exercise 05

	Exercise :
Exercise 05: custom permissions.	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django.	
Remarks: n / A	

Given the negative nature of the downvote, it would also restrict its use to certain well selected users.


Implement the same control as the previous year, but for downvotes. As such permission does not exist by default and it would be ugly that you read another UTI not adapted permission, make your own permission to restrict the use of the downvote!

Just like last year, he is not here simply to remove a page button.

Note: As with the removal of a tip, the author can downvoter own tips even if it is stupid.

Chapter IX

Exercise 06

	Exercise :
Exercise 06: Automation and reputation	
Render Folder: <i>ex/</i>	
Files to make: Your project, your file requirements.txt	
Permitted functions: All the features of Django, django-bootstrap3.	
Remarks: n / A	

Now we have most of the tools needed to run this project, it would be nice to not have to manually manage permissions. Some sites have implemented a gain privileges going with a reputation gain. This means, users showing worthy unlock new actions! This is the subject of this exercise where you have to build a reputation system dependent on the votes received by Tips that a user has posted.

Replace the user class standard by your own user class. Implementing in this class a way to depend on the reputation of a user's permissions downvoter and remove tips. This user's reputation depends on itself the sum of upvotes and downvotes received by Tips posted by that user.

Each new user starts with 0 reputation points. Every upvote received on one of its pay him 5 Tips reputation points while a downvote him in- rise 2. If a tip is removed, the influence of his votes on the reputation of its author is canceled.

A user unlocks the permission of the Tips downvoter other than its own from 15 points of reputation, and that of removing from 30 to reputation. If a user's reputation down, it may lose the permissions that his reputation had made him.

Modify it also has to change the name of the user in a field beside his reputation

in your pages in parentheses.