



D06 - Аутентификация и пользователи.

Проект сайта "Советы Life Pro".

Краткое описание: Или как простой проект Django может превратить вас в "черный пояс" по управлению разрешениями и пользователями!

Версия: 3

Содержание

I	Инструкции	2
II	Упражнение 00	3
III	Упражнение 01	4
IV	Упражнение 02	6
V	Упражнение 03	7
VI	Упражнение 04	9
VII	Упражнение 05	10
VIII	Упражнение 06	11

Глава I


Инструкции

Если нет явного противоречия, следующие инструкции будут действительны для всех дней этого Python Django Piscine.

- Только эта страница будет служить в качестве справочника; не доверяйте слухам.
- Осторожно! Этот документ может измениться за час до отправки.
- Эти упражнения тщательно выстроены в порядке возрастания сложности - от самого легкого к самому трудному. Мы не будем принимать во внимание успешно выполненное более трудное упражнение, если более легкое упражнение не вполне функционально.
- Убедитесь, что у вас есть соответствующие разрешения на файлы и каталоги.
- Вы должны следовать процедуре подачи заявок для каждого упражнения.
- Ваши упражнения будут проверены и оценены вашими одноклассниками.
- Кроме того, ваши упражнения будут проверяться и оцениваться программой под названием Moulinette. Moulinette очень скрупулезно и строго оценивает вашу работу. Она полностью автоматизирована, и с ней невозможно договориться. Поэтому, если вы хотите избежать неприятных сюрпризов, будьте как можно более тщательны.
- Упражнения в Shell должны выполняться с помощью /bin/sh.
- Вы не можете оставить в каталоге ни одного дополнительного файла, кроме указанных в теме.
- Есть вопрос? Задайте его своему сверстнику справа. В противном случае обратитесь к сверстнику слева.
- Ваш справочник называется Google / человек / Интернет /
- Не забывайте обсуждать на форуме piscine вашего Intra и на Slack!
- Тщательно изучите примеры. В них вполне могут потребоваться детали, которые прямо не указаны в теме...

Глава II

Упражнение 00

	Упражн ение
Упражнение 00: Анонимные сессии.	
Входящий каталог : <code>ex/</code>	
Файлы для сдачи : Ваш проект, ваш файл <code>requirements.txt</code>	
Разрешенные функции : Все функциональные возможности Django, <code>random</code> , <code>django-bootstrap3</code>	


Вы начинаете проект веб-сайта, который позволит пользователям размещать "Советы профи по жизни".

После создания проекта, приложения и установки домашней страницы по адресу URL `/`, реализуйте следующую систему анонимных сессий:

- Веб-пользователь, посещающий ваш сайт, может получить имя пользователя. Это имя выбирается случайным образом в списке из 10 имен, которые вы определите в файле `settings.py` вашего проекта, срок действия которого составляет всего 42 секунды после его определения.
- Это имя должно быть постоянным и оставаться идентичным в течение 42 секунд. По истечении 42 секунд имя должно быть переопределено, как было описано ранее.
- Выделите реализованное поведение, добавляющее сообщение в элемент `<nav>` в заголовке главной страницы. Это сообщение будет простым "Hello user!", заменяющим пользователя на случайно заданное имя до конца его действия.
- Имя должно мгновенно появляться на посещенных страницах: если для его появления или изменения необходимо перезагрузить страницу, упражнение не выполнено!

Глава III

Упражнение 01

	Упражнение
Упражнение 01: Создание пользователя.	
Входящий каталог : ex/	
Файлы для сдачи : Ваш проект, ваш файл requirements.txt	
Разрешенные функции : Все функциональные возможности Django, django-bootstrap3	

Теперь, когда система анонимных сессий реализована, пришло время добавить управление пользователями с помощью создания интерфейса аутентификации.

В этом упражнении вы шаг за шагом создадите этот интерфейс:

- Начните с добавления страницы регистрации и страницы входа в систему. Добавьте ссылку на каждую из этих страниц в навигационный элемент. Для этого добавьте все необходимые ссылки, шаблоны, представления и т.д..

Вы также добавите название сайта в начало навигатора и превратите его в ссылку на вашу домашнюю страницу.

- Создайте формы, которые вы будете вставлять в эти страницы, следуя этим спецификациям:
 - Регистрационная форма:
 - * Вы вставите 3 различных поля, включая:
 - Поле для имени пользователя.
 - Поле для пароля.
 - Поле для подтверждения пароля.
 - * Проверка будет проведена только в том случае, если каждое поле заполнено правильно, что данное имя уже не используется, а оба поля пароля заполнены строго одинаково.

- Форма для входа в систему:
 - * Вы должны вставить 2 разных поля, включая:
 - Поле для имени пользователя.
 - Поле для пароля.
 - * Проверка будет проведена только в том случае, если каждое поле заполнено и что используемая пара имя пользователя/пароль уже существует в базе данных и может быть идентифицирована.
- Теперь реализуйте поведение ваших страниц:
 - В случае проверки данных:

Регистрация: Создание и активация нового пользователя с помощью предоставленных ин- формаций. Затем соединение с этим пользователем и перенаправление на домашнюю страницу.

Подключение: Соединение с пользователем, соответствующим указанным данным, и перенаправление на домашнюю страницу.
 - В случае отсутствия валидации:

В любом случае, вы заново откроете страницу с формой, содержащей информацию, которую вы ввели ранее (за вычетом пароля).

Регистрация: Отображение одного (или нескольких) сообщений об ошибке с объяснением ошибки (обязательное поле, существующее имя пользователя, неправильный пароль).

Вход в систему: Отображение сообщения об ошибке в форме (обязательное поле или неверные данные для входа в систему).
- Наконец, когда пользователь вошел в систему:
 - Ссылки на страницы регистрации и входа в систему заменены ссылкой log-out (функциональной, да), которая выводит пользователя из системы и перенаправляет его на домашнюю страницу после нажатия на нее.
 - Имя анонимной сессии в элементе nav заменяется именем пользователя.
 - Посетитель не должен иметь доступ к страницам регистрации и входа в систему. Если они попытаются зайти на них, то будут перенаправлены на домашнюю страницу.
- Разумеется, пароли будут скрыты при вводе.




В другом упражнении вам нужно будет создать свой собственный класс пользователя, чтобы заменить предоставленный по умолчанию класс.

Возможно, будет разумно (но не обязательно) предусмотреть это изменение в вашем проекте, чтобы вам не пришлось вручную переписывать каждый вызов пользовательского класса в вашем коде.

Глава IV

Упражнение 02

	Упражн ение
Упражнение 02: Наши советы!	
Входящий каталог : <i>ex/</i>	
Файлы для сдачи : Ваш проект, ваш файл requirements.txt	
Разрешенные функции : Все функциональные возможности Django, django-bootstrap3.	

Пришло время реализовать основную функциональность этого проекта:

систему чаевых! Для этого вам необходимо создать модель Tip с:

- Поле содержимого с... содержимым подсказки (разумеется, в текстовом виде).
- Поле автора (пользователь, создавший совет).
- Поле даты (дата создания записи).


Советы должны быть перечислены на главной странице, и все атрибуты, указанные в этом разделе, должны быть отображены.

После входа в систему пользователь должен иметь возможность создать совет с помощью формы, установленной на главной странице. (используйте модельную форму!)

Если пользователь не вошел в систему, форма не должна появляться, и посетитель не должен иметь возможности создать чаевые.

Глава V

Упражнение 03

	Упражн ение
Упражнение 03: Голоса.	
Входящий каталог : <code>ex/</code>	
Файлы для сдачи : Ваш проект, ваш файл <code>requirements.txt</code>	
Разрешенные функции : Все функциональные возможности Django, django-bootstrap3.	

Теперь, когда ваша система советов готова, было бы неплохо иметь возможность собирать лучшие из них. Да, настало время установить систему голосования "за" и "против"!

Для этого вы должны изменить свою модель `Tip`, чтобы реализовать эти функции.

Вы также должны реализовать возможность удаления советов, которые были признаны неинтересными. Вы должны добавить следующие элементы к советам, перечисленным на вашей домашней странице:

- Кнопка удаления
- Кнопка повышения голоса
- Кнопка понижения голоса
- Количество голосов "за"
- Количество "даунвотсов"

На данный момент единственным ограничением, которое вы введете, будет необходимость входа в систему для того, чтобы проголосовать или удалить. Если пользователь не вошел в систему, ни одно из этих действий будет невозможно.

Каждое из этих действий приводит к перезагрузке страницы.

Очевидно, что только что созданный совет будет иметь 0 upvotes и 0 downvotes.

Предупреждение:

- Одновременное повышение и понижение голосов за один и тот же совет должно быть невозможным.
- Пользователь может проголосовать за один и тот же совет только один раз. При двойном нажатии на одну и ту же кнопку голосование отменяется.
- Если пользователь понижает голос за совет, за который он ранее проголосовал, то повышение голоса заменяется понижением голоса (и наоборот).




Внимательно следите за реализацией системы upvote. Несоответствие, несовершенство (количество голосов, не уменьшающееся после отмены голосования) сделает упражнение недействительным. Вам следует рассмотреть возможность использования полей manytomany, а не "счетчиков" для реализации системы голосования.

На этом этапе каждое действие должно работать только в том случае, если пользователь вошел в систему.

Вы должны иметь возможность отобразить или отключить элемент, который пользователь не сможет использовать.

Глава VI

Упражнение 04

	Упражн ение
Упражнение 04: Первичное использование разрешений.	
Входящий каталог : <code>ex/</code>	
Файлы для сдачи : Ваш проект, ваш файл <code>requirements.txt</code>	
Разрешенные функции : Все функциональные возможности Django, django-bootstrap3.	

Теперь наш проект стал немного более существенным, и мы понимаем, что каждый может делать все, что угодно, пока он входит в систему. Возможно, пришло время добавить несколько небольших ограничений.

Теперь пользователю требуется разрешение на удаление советов, если он хочет... удалить советы.

Активируйте интерфейс администрирования, который поставляется с Django по умолчанию, и используйте суперпользователя для изменения разрешений и выделения поведения вашего проекта.


Конечно, это требует большего, чем просто удалить кнопку "удалить" страницы (хотя вы могли бы сделать это). Найдите правильный способ ограничить эту функциональность, правильно проверив разрешения.

NB: Конечно, автор совета может удалить свой совет без какого-либо разрешения.

Это будет единственным исключением из правила.

Глава VII

Упражнение 05

	Упражн ение
Упражнение 05: персонализированная авторизация.	
Входящий каталог : ex/	
Файлы для сдачи : Ваш проект, ваш файл requirements.txt	
Разрешенные функции : Все функциональные возможности Django.	

Учитывая негативный характер "downvote", его использование должно быть зарезервировано для конкретных пользователей.


Примените тот же контроль, что и в предыдущем упражнении, к downvotes. Поскольку это разрешение не является разрешением по умолчанию, и было бы обидно, если бы вы использовали другое разрешение, которое не подходит, создайте собственное разрешение, чтобы ограничить использование downvote!

Как и в предыдущем упражнении, дело не только в удалении кнопки downvote. Вы можете попробовать, верно? Однако успех не гарантирован.

NB: как и в случае с подавлением советов, автор совета всегда может понизить голос за свой собственный совет, если считает его, например, тупым или что-то в этом роде.

Глава VIII

Упражнение 06

	Упражн ение
Упражнение 06: Автоматизация и репутация	
Входящий каталог : <code>ex/</code>	
Файлы для сдачи : Ваш проект, ваш файл <code>requirements.txt</code>	
Разрешенные функции : Все функциональные возможности Django, django-bootstrap3.	

Теперь у нас есть большинство необходимых инструментов для запуска этого проекта, и было бы хорошо, если бы нам не приходилось вручную управлять этими разрешениями. Некоторые сайты начали выдавать привилегии по шкале репутации. Это дает достойным пользователям права на использование новых функций! Именно этому посвящено данное упражнение. Вам нужно будет создать систему репутации в зависимости от голосов, которые получили Советы, размещенные пользователем.

Замените стандартный класс пользователя своим собственным классом пользователя. В этом классе вы создадите способ, с помощью которого полномочия на понижение голосов и удаление советов будут зависеть от репутации пользователя. Репутация этого пользователя будет зависеть от количества голосов "за" и "против", которые получили его советы.

Каждый новый пользователь начинает с 0 очков репутации. Каждое повышение голоса за советы дает ему 5 очков репутации. При понижении голоса снимается 2. Если совет удаляется, влияние его голосов на репутацию пользователя исчезает.

Пользователь получает право удалять советы (не только свои), набрав 15 очков репутации. Чтобы получить право удалять советы, необходимо набрать 30 очков репутации. Если репутация пользователя снижается, он может потерять полномочия, которые давала ему репутация.

Вы также измените внешний вид имени пользователя, отображая его репутацию сбоку (между скобками), когда оно появляется на ваших страницах.

