1. Для добавления новых загадок администраторами вместе с вариантами ответа **добавим новый шаблон riddles/templates/admin.html**:

{% **extends 'base.html'** %}  
{% **block main** %}  
 {% **if user**.**is\_staff** %}  
 {% **if message** %}  
 <**p class="alert alert-success"**>  
 <**strong**>{{ **message** }}</**strong**>  
 </**p**>  
 {% **endif** %}  
 <**div class="card mb-3"**>  
 <**div class="card-body"**>  
 <**form action="../post\_riddle/" method="post"**>  
 {% **csrf\_token** %}  
 <**h5**>Add riddle:</**h5**>  
 <**div class="mb-3"**>  
 <**input class="form-control"  
 placeholder="Riddle text"  
 name="text"  
 required**/>  
 </**div**>  
 <**div class="card mb-3"**>  
 <**div class="card-body" id="options\_new"**>  
 <**div class="mb-3"**>  
 <**input class="form-control"  
 placeholder="Option 1 (correct)"  
 name="option1"  
 required**/>  
 </**div**>  
 <**div class="mb-3"**>  
 <**input class="form-control"  
 placeholder="Option 2"  
 name="option2"  
 required**/>  
 </**div**>  
 <**script**>  
 **var *option\_counter*** = 2;  
 </**script**>  
 </**div**>  
 </**div**>  
 <**div class="mb-3"**>  
 <**button class="btn btn-success"  
 onclick="***addOption*(**'options\_new'**, 0)**"  
 type="button"**>  
 + Add option  
 </**button**>  
 <**button class="btn btn-success"**>  
 Save  
 </**button**>  
 </**div**>  
 <**input type="hidden"  
 name="riddle\_id"  
 value="0"**/>  
 </**form**>  
 </**div**>  
 </**div**>  
 <**div class="card mb-3"**>  
 <**div class="card-body"**>  
 <**h4 class="card-title"**>Available Riddles</**h4**>  
 {% **if latest\_riddles** %}  
 <**ul**>  
 {% **for riddle in latest\_riddles** %}  
 <**li**>  
 <**a href="/riddles/**{{ **riddle**.**id** }}**/"**>  
 {{ **riddle**.**riddle\_text** }}  
 </**a**>  
 </**li**>  
 {% **endfor** %}  
 </**ul**>  
 {% **else** %}  
 <**p**>No riddles are available right now.</**p**>  
 {% **endif** %}  
 </**div**>  
 </**div**>  
  
 <**script**>  
 *//добавление блока для нового ответа* **function** *addOption*(target, pos\_in\_array){  
 **var** option=***document***.createElement(**"div"**);  
 ***option\_counter*** += 1;  
 option.**className**=**"input-group mb-3"**;  
 option.**innerHTML**=  
 **'<input class="form-control"'** +  
 **' placeholder="Option '** +   
 ***option\_counter*** + **'"'** +   
 **' name="option'** + ***option\_counter*** + **'"/>'** +  
 **'<div class="input-group-append">'** +  
 **' <button class="btn btn-danger"'** +  
 **' type="button"'** +  
 **' onclick="removeParent('** +  
 **' this.parentNode)">'** +  
 **' Delete'** +  
 **' </button>'** +  
 **'</div>'**;  
 ***document***.getElementById(target).appendChild(option);  
 }  
 *//удаление блока ответа с поправкой номеров других ответов* **function** *removeParent*(child){  
 **var** optionList=child.**parentNode**.**parentNode**;  
 **var** currentOption=child.**parentNode**;  
 optionList.*removeChild*(currentOption);  
 **for**(**var** i=3; i<optionList.**children**.**length**; i+=1){  
 **var** opt=optionList.**children**[i].**children**[0];  
 opt.**name**=**"option"**+(i);  
 opt.**placeholder**=**"Option "**+(i);  
 }  
 ***option\_counter*** -= 1;  
 }  
 </**script**>  
 {% **else** %}  
 <**h4**>Access denied.</**h4**>  
 {% **endif** %}  
{% **endblock** %}

1. Для отображения администраторской панели добавления загадок **допишем новую функцию, аналогичную функции index, в существующем файле riddles/views.py:**

**def** admin(request):  
 message = **None  
 if "message" in** request.GET:  
 message = request.GET[**"message"**]  
 *# создание HTML-страницы по шаблону admin.html  
 # с заданными параметрами latest\_riddles и message* **return** render(  
 request,  
 **"admin.html"**,  
 {  
 **"latest\_riddles"**:  
 Riddle.objects.order\_by(**'-pub\_date'**)[:5],  
 **"message"**: message,  
 }  
 )

1. **Привяжем созданную функцию к соответствующему URL, на который будет заходить администратор, в riddles/urls.py:**

urlpatterns = [  
 re\_path(**r'^$'**, views.index, name=**'index'**),  
 re\_path(**r'^([0-9]+)/$'**, views.detail, name=**'detail'**),  
 re\_path(**r'^([0-9]+)/answer/$'**, views.answer, name=**'answer'**),  
 *# ...*

re\_path(**r'^admin/$'**, views.admin, name=**'admin'**),   
]

1. Просто отобразить пустую форму добавления новой загадки недостаточно – необходимо реализовать обработку отправляемых заполненных форм. Для этого **допишем новую функцию в существующем файле riddles/views.py:**

**def** post\_riddle(request):  
 *# защита от добавления загадок неадминистраторами* author = request.user  
 **if not** (author.is\_authenticated **and** author.is\_staff):  
 **return** HttpResponseRedirect(app\_url+**"admin"**)  
 *# добавление загадки* rid = Riddle()  
 rid.riddle\_text = request.POST[**'text'**]  
 rid.pub\_date = datetime.now()  
 rid.save()  
 *# добавление вариантов ответа* i = 1 *# нумерация вариантов на форме начинается с 1*  
 *# количество вариантов неизвестно,  
 # поэтому ожидаем возникновение исключения,  
 # когда варианты кончатся* **try**:  
 **while** request.POST[**'option'**+str(i)]:  
 opt = Option()  
 opt.riddle = rid  
 opt.text = request.POST[**'option'**+str(i)]  
 opt.correct = (i == 1)  
 opt.save()  
 i += 1  
 *# это ожидаемое исключение,  
 # при котором ничего делать не надо* **except**:  
 **pass  
  
 return** HttpResponseRedirect(app\_url+str(rid.id))

1. **Привяжем созданную функцию к соответствующему URL, на который будут отправляться заполненные формы, в riddles/urls.py:**

urlpatterns = [  
 re\_path(**r'^$'**, views.index, name=**'index'**),  
 re\_path(**r'^([0-9]+)/$'**, views.detail, name=**'detail'**),  
 re\_path(**r'^([0-9]+)/answer/$'**, views.answer, name=**'answer'**),  
 *# ...*

re\_path(**r'^post\_riddle/$'**,   
 views.post\_riddle,   
 name=**’post\_riddle’**),  
]

1. **Запустим приложение, зайдем на адрес http://127.0.0.1:8000/riddles/admin/ , добавим 2 загадки с вариантами ответа, ответим на них правильно. Обратите внимание, что правильный вариант всегда оказывается первым в списке вариантов ответа.**
2. Неинтересно, когда правильный вариант при выборе ответа всегда оказывается первым из-за того, что стоит первым в форме добавления загадки! Реализуем перестановку вариантов ответа с помощью **модуля numpy – установим его к используемому интерпретатору через меню File – Settings.**
3. **Переделаем старую функцию detail с помощью установленного модуля numpy в файле riddles/views.py:**

**import** numpy

*# ...*

*# страница загадки со списком ответов***def** detail(request, riddle\_id):  
 error\_message = **None  
 if "error\_message" in** request.GET:  
 error\_message = request.GET[**"error\_message"**]  
 *# формируем список ответов* ordered\_option\_set = \  
 list(Option.objects.filter(riddle\_id=riddle\_id))  
 *# формируем случайный порядок номеров ответов* option\_iter = \  
 numpy.random.permutation(len(ordered\_option\_set))  
 *# формируем новый список, в который выписываем ответы  
 # в сформированном случайном порядке* option\_set = []  
 **for** num **in** option\_iter:  
 option\_set.append(ordered\_option\_set[num])  
 **return** render(  
 request,  
 **"answer.html"**,  
 {  
 *# передаем список ответов в случайном порядке*  
 **"option\_set"**: option\_set,  
  
 **"riddle"**: get\_object\_or\_404(  
 Riddle, pk=riddle\_id),  
 **"error\_message"**: error\_message,  
 **"latest\_messages"**:  
 Message.objects  
 .filter(chat\_id=riddle\_id)  
 .order\_by(**'-pub\_date'**)[:5],  
 *# кол-во оценок, выставленных пользователем* **"already\_rated\_by\_user"**:  
 Mark.objects  
 .filter(author\_id=request.user.id)  
 .filter(riddle\_id=riddle\_id)  
 .count(),  
 *# оценка текущего пользователя* **"user\_rating"**:  
 Mark.objects  
 .filter(author\_id=request.user.id)  
 .filter(riddle\_id=riddle\_id)  
 .aggregate(Avg(**'mark'**))  
 [**"mark\_\_avg"**],  
 *# средняя по всем пользователям оценка* **"avg\_mark"**:  
 Mark.objects  
 .filter(riddle\_id=riddle\_id)  
 .aggregate(Avg(**'mark'**))  
 [**"mark\_\_avg"**]  
 }  
 )

1. **Подставим сформированный список ответов в случайном порядке в форму ответа в шаблоне riddles/templates/answer.html:**

...

<**form action="answer/" method="post"**>  
 {% **csrf\_token** %}  
 {% **for option in option\_set** %}  
 <**input type="radio"  
 name="option"  
 id="option**{{ **forloop**.**counter** }}**"  
 value="**{{ **option**.**id** }}**"**/>  
 <**label for="option**{{ **forloop**.**counter** }}**"**>  
 {{ **option**.**text** }}  
 </**label**><**br**>  
 {% **endfor** %}  
 <**input type="submit" value="Answer"**/>  
</**form**>

...

1. **Запустим приложение, зайдем на страницу любой загадки, несколько раз обновим страницу в браузере. Обратите внимание, что при каждом обновлении страницы порядок вариантов ответа изменяется.**
2. Администраторам неудобно набирать адрес панели добавления загадок в адресной строке – поэтому **допишем ссылку на панель добавления загадки, которая будет появляться только у администраторов, после ссылок Log out и Change password в шаблоне riddles/templates/base.html:**

...

<**p class="text-right"**>  
 Hello, {{ **user**.**username** }}!  
 <**a href="/riddles/logout"**>Log out</**a**> |  
 <**a href="/riddles/password-change"**>Change password</**a**>  
 {% **if user**.**is\_staff** %}  
 | <**a href="/riddles/admin"**>Admin panel</**a**>  
 {% **endif** %}  
</**p**>

...

1. **Заходим в приложение под пользователем-администратором, проверяем наличие ссылки, переходим по ней, чтобы убедится, что откроется страница добавления загадки, выходим из учетной записи администратора и заходим под обычным пользователем, проверяем, что ссылка на администраторскую панель не отображается.**