

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Кафедра Интеллектуальных систем в
гуманитарной сфере, факультет
Инфокоммуникационных технологий

Лабораторная работа №2

Реализация простого сайта средствами Django

В

ыполнил студент гр.

№ K33422

Кузьмичев Кирилл

Максимович

Проверил преподаватель

Говоров А.И.

Санкт-Петербург

2020

Вариант №4: Список туров туристической фирмы

Цель: Овладеть практическими навыками и умениями реализации web-сервисов средствами Django 2.2.

Оборудование: Компьютерный класс.

Программное обеспечение: Python 3.6+, Django 3, PostgreSQL *.

Текст задания:

Хранится информация о названии тура, турагентстве, описании тура, периоде проведения тура, условиях оплаты.

Необходимо реализовать следующий функционал:

- Регистрация новых пользователей.
- Просмотр и резервирование туров. Пользователь должен иметь возможность

редактирования и удаления своих резервирований.

- Написание отзывов к турам. При добавлении комментариев, должны сохраняться даты тура, текст комментария, рейтинг (1–10), информация о комментаторе.

- Администратор должен иметь возможность подтвердить резервирование

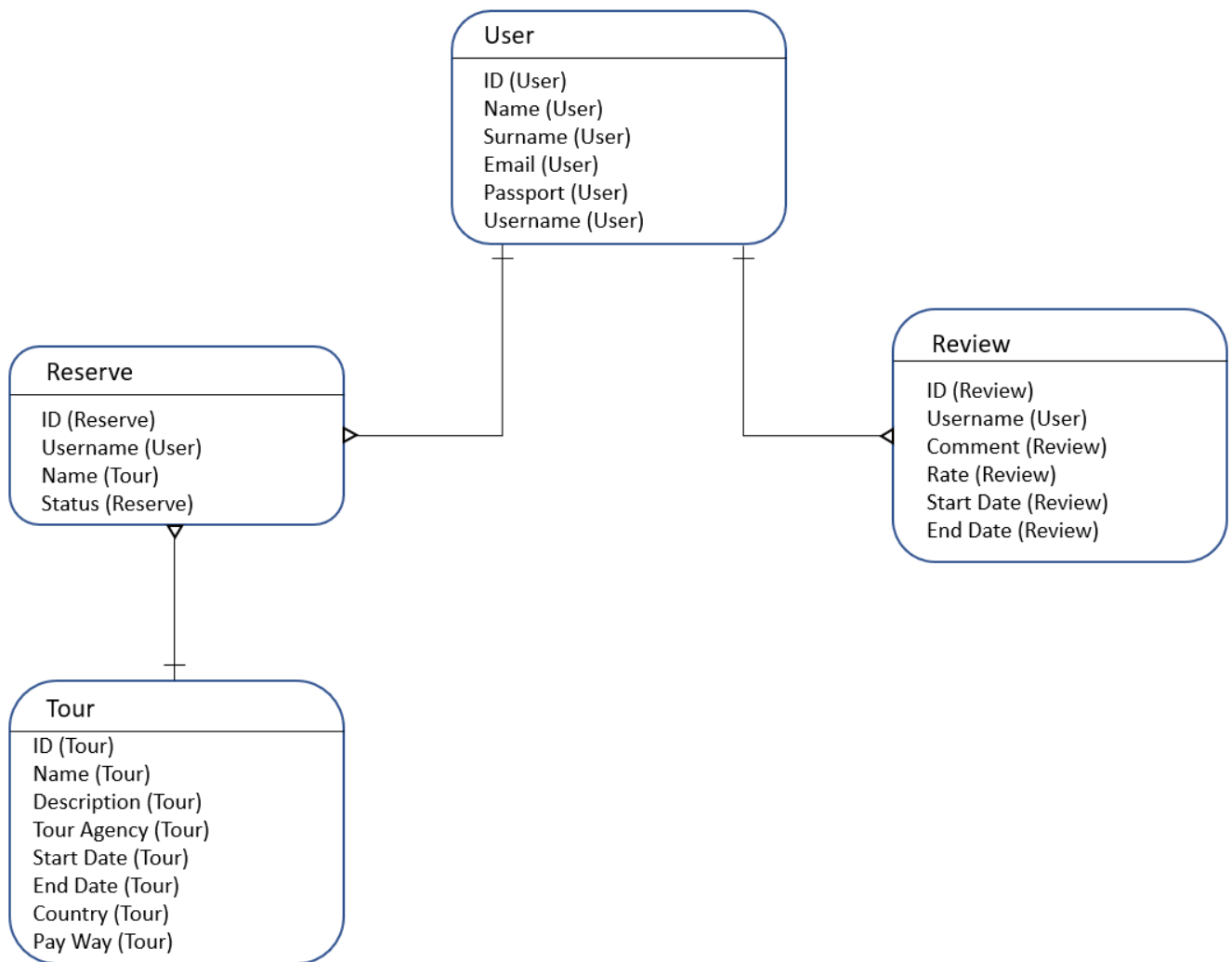
тура средствами Django-admin.

- В клиентской части должна формироваться таблица, отображающая все проданные туры по странам.

Выполнение работы:

Подготовительная часть (Часть №1):

Модель базы:



Созданные сущности:

Сущность “User”:

```
class User(AbstractUser):
```

```
    username = models.CharField(max_length=10, blank=True, null=True, unique=True)
    passport = models.CharField(max_length=100, blank=True, null=True)
    email = models.CharField(max_length=30, blank=True, null=True)
    tour = models.ManyToManyField('Tour', through="Reserve")
```

Сущность “Review”:

```
class Review(models.Model):
    MARKS = (
        ("1", "1"),
        ("2", "2"),
        ("3", "3"),
        ("4", "4"),
        ("5", "5"),
        ("6", "6"),
        ("7", "7"),
        ("8", "8"),
        ("9", "9"),
        ("10", "10")
    )
    start_date = models.DateField()
    end_date = models.DateField()
    username = models.ForeignKey('User', on_delete=models.CASCADE)
    comment = models.TextField(max_length=300)
    rate = models.CharField(max_length=2, choices=MARKS)
```

Сущность “Reserve”:

```
class Reserve(models.Model):
    username = models.ForeignKey('User', on_delete=models.CASCADE)
    name = models.ForeignKey('Tour', on_delete=models.CASCADE)
    status = models.BooleanField(blank=True, null=True)
```

Сущность “Tour”:

```
class Tour(models.Model):
    PAY = (
        ("полная оплата", "полная оплата"),
        ("рассрочка", "рассрочка")
    )
    name = models.CharField(max_length=100)
    description = models.CharField(max_length=100)
    tour_agency = models.CharField(max_length=30)
    start_date = models.DateField()
    end_date = models.DateField()
    country = models.CharField(max_length=30)
    payment = models.CharField(max_length=13, choices=PAY)
```

Функции:

Главная страница:

```
1 from django.shortcuts import render, redirect, get_object_or_404
2 from django.views.generic import CreateView, ListView, UpdateView, DeleteView
3 from .form import *
4 from .models import *
5 from django.contrib.auth import authenticate, login
6
7
8 def main(request):
9     return render(request, 'main.html')
```

Main.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Турфирма</title>
6 </head>
7 <body>
8
9     {% if user.is_authenticated %}
10     <p><h1>Вы вошли как <a href="/profile/"> {{ user.username }} </a></h1></p>
11     {% else %}
12     <p>Вход не выполнен</p>
13     {% endif %}
14
15 <a href="/accounts/login/" style ="..."
23 >Войти</a>
24
25 <a href="/register/" style ="..."
33 >Зарегистрироваться</a><br><br><br>
34
35 <a href="/tours/" style ="..."
43 >Тур</a>
44
45 <a href="/soldtours/" style ="..."
53 >Проданные туры</a><br><br><br>
54
55 <a href="/reviews/" style ="..."
63 >Все отзывы</a>
64
65 <a href="/review/" style ="..."
73 >Оставить отзыв</a>
74
75
76 </body>
77 </html>
```

Профиль пользователя:

```
12 def profile(request):
13     return render(request, 'profile.html')
14
```

Profile.html:

```
1 |!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Профиль</title>
6 </head>
7 <body>
8
9     <h2>Профиль</h2>
10
11     <strong>Логин: </strong>{{ user.username }} <br>
12     <strong>Почта: </strong>{{ user.email }} <br>
13     <strong>Имя: </strong>{{ user.first_name }} <br>
14     <strong>Фамилия: </strong>{{ user.last_name }} <br>
15     <strong>Паспорт: </strong>{{ user.passport }} <br><br>
16
17     <a href="/myreserves">Мои бронирования</a><br>
18     <a href="/">Главная</a>
19
20 </body>
21 </html>
```

Регистрация пользователя:

```
16 def register(request):
17     if request.method == 'POST':
18         form = RegistrationForm(request.POST)
19
20         if form.is_valid():
21             form.save()
22             username = form.cleaned_data['username']
23             password = form.cleaned_data['password1']
24             user = authenticate(username=username, password=password)
25             login(request, user)
26             return redirect('/profile')
27
28     else:
29         form = RegistrationForm()
30
31     visual = {'form': form}
32     return render(request, 'registration/register.html', visual)
```

Register.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Регистрация</title>
6 </head>
7 <body>
8     <form method="POST" action="{% url 'register' %}">
9         {% csrf_token %}
10
11         {% if form.errors %}
12         <p>Что-то пошло не так</p>
13         {% endif %}
14
15         {{form.as_p}}
16
17         <input type="submit" value="Sign up"> <br><br>
18     </form>
19 </body>
20 </html>
```

Выбор туров:

```
35 def tourslis(request):
36     visual = {"tours": Tour.objects.all()}
37     return render(request, 'tours.html', visual)
38
```

Tours.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Турь</title>
6     <style type="text/css">
7         .main {
8             width: 500px;
9             background: transparent;
10            outline-color: Black;
11            outline-style: solid;
12            color: Black;
13        }
14    </style>
15 </head>
16 <body>
17 <div class="main">
18     {% for tour in tours %}
19     <strong>Название: </strong>{{tour.name}} <br>
20     <strong>Описание: </strong>{{tour.description}} <br>
21     <strong>Агентство: </strong>{{tour.tour_agency}} <br>
22     <strong>Дата начала:</strong>{{tour.start_date}} <br>
23     <strong>Дата окончания:</strong>{{tour.end_date}} <br>
24     <strong>Страна: </strong>{{tour.country}} <br>
25     <strong>Способ оплаты: </strong>{{tour.payment}} <br>
26     <a href="/tours/{{ tour.id }}/reserve">Забронировать</a><br><br>
27     {% endfor %}
28 </div>
29
30 </body>
31 </html>
```


Бронирование туров:

```
40 class CreateReserve(CreateView):
41     form_class = CreateReserveForm
42     model = Reserve
43     template_name = 'reserve.html'
44     context_object_name = 'reserve'
45     success_url = '/profile'
46
47     def get_initial(self):
48         initial = super(CreateReserve, self).get_initial()
49         initial = initial.copy()
50         initial['username'] = self.request.user.pk
51         initial['name'] = get_object_or_404(Tour, pk=self.kwargs['pk'])
52         return initial
```

Reserve.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Бронирование</title>
6 </head>
7 <body>
8
9 <form method="POST" enctype="multipart/form-data">
10
11     {% csrf_token %}
12     {{ form.as_p }}
13
14     <button class="btn btn-secondary">Забронировать</button>
15
16 </form>
17
18 </body>
19 </html>
```

Проданные туры:

```
55 def soldtourslist(request):
56     visual = {"reserves": Reserve.objects.all()}
57     return render(request, 'soldtours.html', visual)
```

Soldtours.html:

```
4     <meta charset="UTF-8">
5     <title>Проданные туры</title>
6 </head>
7 <body>
8
9     {% block content %}
10
11     <table id="myTable" bordercolor="Black" border="5" width="45%">
12     <thead>
13     <tr>
14         <th scope="colgroup"><h1>Проданные туры:</h1></th>
15         <th scope="colgroup"><h1>Пользователь:</h1></th>
16     </tr>
17     </thead>
18
19     <tbody>
20
21     {% for reserve in reserves %}
22
23     <tr>
24         <td align="center"> {{ reserve.name.name }} </td>
25         <td align="center"> {{ reserve.username.username }} </td>
26     </tr>
27
28     {% endfor %}
29     </tbody>
30 </table>
31 {% endblock %}
32
33 </body>
34 </html>
```

Оставить отзыв о турах:

```
60 class CreateReview(CreateView):
61     form_class = CreateReviewForm
62     model = Review
63     template_name = 'review.html'
64     context_object_name = 'review'
65     success_url = '/reviews'
66
67     def get_initial(self):
68         initial = super(CreateReview, self).get_initial()
69         initial = initial.copy()
70         initial['username'] = self.request.user.pk
71         return initial
```

Review.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Отзыв</title>
6 </head>
7 <body>
8
9 <form method="POST" enctype="multipart/form-data">
10
11     {% csrf_token %}
12     {{ form.as_p }}
13
14     <button class="btn btn-secondary">Отправить</button>
15
16 </form>
17
18 </body>
19 </html>
```

Посмотреть отзывы о турах:

```
74 def reviewslist(request):
75     visual = {"reviews": Review.objects.all()}
76     return render(request, 'reviews.html', visual)
77
```

Reviews.html:

```
1 |<!DOCTYPE html>
2 |<html lang="en">
3 |<head>
4 |    <meta charset="UTF-8">
5 |    <title>Отзывы</title>
6 |    <style type="text/css">
7 |        .main {
8 |            width: 500px;
9 |            background: transparent;
10 |            outline-color: Black;
11 |            outline-style: solid;
12 |            color: Black;
13 |        }
14 |    </style>
15 |</head>
16 |<body>
17 |<div class="main">
18 |    {% for review in reviews %}
19 |    <strong>Имя: </strong>{{review.username}} <br>
20 |    <strong>Дата начала: </strong>{{review.start_date}} <br>
21 |    <strong>Дата окончания: </strong>{{review.end_date}} <br>
22 |    <strong>Рейтинг: </strong>{{review.rate}} <br>
23 |    <strong>Комментарий: </strong>{{review.comment}} <br><br>
24 |    {% endfor %}
25 |</div>
26 |
27 |</body>
28 |</html>
```

Просмотр/удаление/изменение забронированных туров пользователя:

```
79 class listreserves(ListView):
80     template_name = 'myreserves.html'
81     context_object_name = 'reserves_list'
82
83     def get_queryset(self):
84         self.user = self.request.user.pk
85         return Reserve.objects.filter(username=self.user)
86
87
88 class deletereserveView(DeleteView):
89     model = Reserve
90     template_name = 'deletereserve.html'
91     context_object_name = 'reserve'
92     success_url = '/profile'
93
94
95 class updatereserveView(UpdateView):
96     model = Reserve
97     fields = ['start_date', 'end_date']
98     template_name = 'updatereserve.html'
99     context_object_name = 'reserve'
100     success_url = '/profile'
101
```

Myreserves.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Забронированные туры</title>
6 </head>
7 <body>
8
9     <h1>Бронирования {{user.username}}</h1>
10
11     {% for reserve in reserves_list %}
12
13         <strong>Тип: </strong>{{ reserve.name.name }} <br>
14         <strong>Дата начала: </strong>{{ reserve.name.start_date }} <br>
15         <strong>Дата окончания: </strong>{{ reserve.name.end_date }} <br>
16         <strong>Статус бронирования: </strong>{{ reserve.status|yesno:"Успешное бронирование,Ждет проверки администратора" }}<br/>
17
18
19         <a href="myreserves/deletereserve/{{ reserve.id }}"><button class="btn btn-secondary">Удалить</button></a>
20         <a href="myreserves/updatereserve/{{ reserve.id }}"><button class="btn btn-secondary">Изменить</button></a>
21     <br><br><br>
22     {% empty %}
23     <li>Нет бронирований</li>
24     {% endfor %}
25
26
27     <a href="/profile">Вернуться в профиль</a><br>
28     <a href="/">Главная</a>
29
30 </body>
31 </html>
```

Deletereserve.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Удаление бронирования</title>
6 </head>
7 <body>
8
9     <form method="post">{% csrf_token %}
10         <p>Хотите удалить бронирование "{{object}}"?</p>
11         <input type="submit" value="Удалить"> <br><br>
12     </form>
13     <a href="/myreserves/">Вернуться к забронированным турам</a>
14
15 </body>
16 </html>
```

Updatereserve.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Изменение бронирования</title>
6 </head>
7 <body>
8
9     <form method="POST" enctype="multipart/form-data">
10
11         <!-- Security token -->
12         {% csrf_token %}
13
14         <!-- Using the formset -->
15         {{form.as_p}}
16
17         <input type="submit" value="Сохранить"> <br><br>
18     </form>
19
20 </body>
21 </html>
```

Формы:

Регистрации:

```
1 from django import forms
2 from django.contrib.auth.forms import UserCreationForm
3 from .models import *
4
5
6 class RegistrationForm(UserCreationForm):
7     username = forms.CharField(label='Логин', widget=forms.TextInput(attrs={'class': 'form-input'}))
8     password1 = forms.CharField(label='Пароль', widget=forms.PasswordInput(attrs={'class': 'form-input'}))
9     password2 = forms.CharField(label='Повтор пароля', widget=forms.PasswordInput(attrs={'class': 'form-input'}))
10    email = forms.CharField(label='Почта', widget=forms.TextInput(attrs={'class': 'form-input'}))
11    first_name = forms.CharField(label='Имя', widget=forms.TextInput(attrs={'class': 'form-input'}))
12    last_name = forms.CharField(label='Фамилия', widget=forms.TextInput(attrs={'class': 'form-input'}))
13    passport = forms.CharField(label='Паспорт', widget=forms.TextInput(attrs={'class': 'form-input'}))
14
15    class Meta:
16        model = User
17        fields = [
18            'username',
19            'password1',
20            'password2',
21            'email',
22            'first_name',
23            'last_name',
24            'passport'
25        ]
26
```

Login.html:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Вход</title>
6 </head>
7 <body>
8     {% if form.errors %}
9     <p>Что-то пошло не так</p>
10    {% endif %}
11
12    {% if next %}
13    <p>У Вас нет прав доступа</p>
14    {% endif %}
15
16    <form method="POST" action="{% url 'login' %}">
17        {% csrf_token %}
18
19        <p>Имя: {{ form.username }}</p>
20        <p>Пароль: {{ form.password }}</p>
21
22        <p><input type="submit" value='Войти'></p>
23        <input type="hidden" name="next" value="{{ next }}">
24    </form>
25
26 </body>
27 </html>
```

Резервирования:

```
28 class CreateReserveForm(forms.ModelForm):
29     class Meta:
30         model = Reserve
31         fields = (
32             'username',
33             'name'
34         )
35
36     def __init__(self, *args, **kwargs):
37         super(CreateReserveForm, self).__init__(*args, **kwargs)
38         self.fields['username'].disabled = True
39         self.fields['name'].disabled = True
40
41
```

Написания отзыва:

```
42 class CreateReviewForm(forms.ModelForm):
43
44     class Meta:
45         model = Review
46         fields = (
47             'username',
48             'start_date',
49             'end_date',
50             'rate',
51             'comment'
52         )
53
54     def __init__(self, *args, **kwargs):
55         super(CreateReviewForm, self).__init__(*args, **kwargs)
56         self.fields['username'].disabled = True
```


Urls:

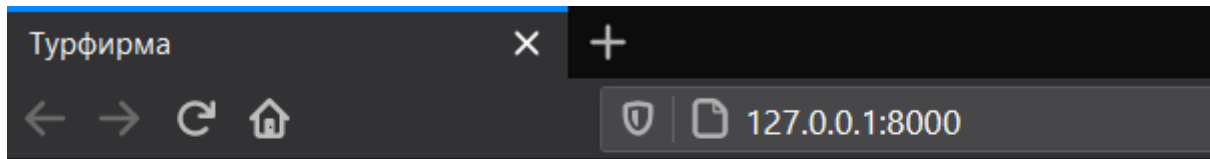
```
1 from django.urls import path
2 from . import views
3 from .views import *
4
5
6 urlpatterns = [
7     path('', views.main),
8     path('profile/', views.profile),
9     path('myreserves/', listreserves.as_view(), name='reserves'),
10    path('myreserves/myreserves/deletereserve/<int:pk>/', views.deletereserveView.as_view()),
11    path('myreserves/myreserves/updatereserve/<int:pk>/', views.updatereserveView.as_view()),
12    path('register/', register, name='register'),
13    path('tours/', views.tourslist),
14    path('tours/<int:pk>/reserve/', CreateReserve.as_view(), name='reserve'),
15    path('review/', CreateReview.as_view(), name='review'),
16    path('soldtours/', views.soldtourslist),
17    path('reviews/', views.reviewslist)
18 ]
```

Admin.py:

```
1 from django.contrib import admin
2 from .models import User, Tour, Review, Reserve
3 from django.contrib.auth.admin import UserAdmin
4
5 admin.site.register(User, UserAdmin)
6 admin.site.register(Tour)
7 admin.site.register(Review)
8 admin.site.register(Reserve)
```

Работа программы:

Без зарегистрированного пользователя:



Вход не выполнен



С зарегистрированным пользователем:



Вы вошли как kirmrik



Регистрация:

Регистрация × +

← → ↻ 🏠 🔒 📄 127.0.0.1:8000/register/

Логин:

Пароль:

Повтор пароля:

Почта:

Имя:

Фамилия:

Паспорт:

Sign up

Профиль:

Профиль × +

← → ↻ 🏠 🔒 📄 127.0.0.1:8000/profile/

Профиль

Логин: kirmrik
Почта: kirmrik@gmail.com
Имя:
Фамилия:
Паспорт: None

[Мои бронирования](#)
[Главная](#)

Бронирования:



Бронирования kirmrik

Тур: Тур - Италия

Дата начала: April 12, 2021

Дата окончания: April 19, 2021

Статус бронирования: Успешное бронирование

Удалить

Изменить

Тур: Тур - Париж

Дата начала: May 10, 2021

Дата окончания: May 17, 2021

Статус бронирования: Успешное бронирование

Удалить

Изменить

Тур: Тур - Россия

Дата начала: April 5, 2021

Дата окончания: April 12, 2021

Статус бронирования: Ждет проверки администратора

Удалить

Изменить

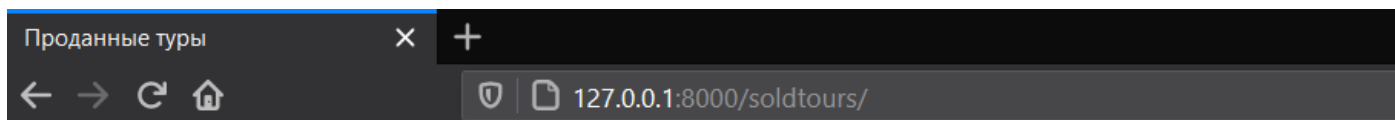
[Вернуться в профиль](#)

[Главная](#)

Забронировать тур и их список:

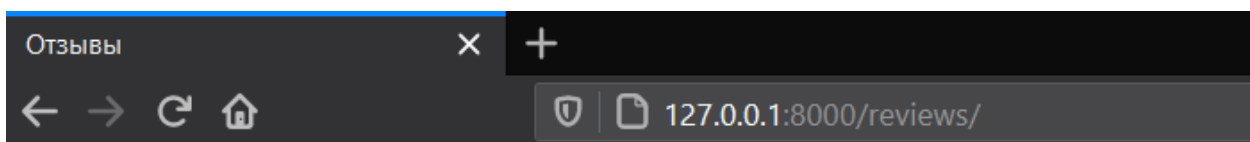
Туры	×	+
←	→	↻
🔒	📄	127.0.0.1:8000/tours/
<div><p>Название: Тур - Италия</p><p>Описание: Тур по итальянским провинциям и Риму</p><p>Агентство: ItalianoDaTuro</p><p>Дата начала: April 12, 2021</p><p>Дата окончания: April 19, 2021</p><p>Страна: Италия</p><p>Способ оплаты: полная оплата</p><p>Забронировать</p></div> <div><p>Название: Тур - Париж</p><p>Описание: Тур по французским провинциям</p><p>Агентство: TuroDaFrance</p><p>Дата начала: May 10, 2021</p><p>Дата окончания: May 17, 2021</p><p>Страна: Франция</p><p>Способ оплаты: рассрочка</p><p>Забронировать</p></div> <div><p>Название: Тур - Россия</p><p>Описание: Тур по золотому кольцу</p><p>Агентство: ТурРосс</p><p>Дата начала: April 5, 2021</p><p>Дата окончания: April 12, 2021</p><p>Страна: Россия</p><p>Способ оплаты: полная оплата</p><p>Забронировать</p></div>		

Проданные туры:



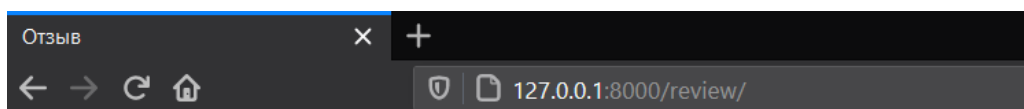
Проданные туры:	Пользователь:
Тур - Италия	kirmrik
Тур - Париж	kirmrik
Тур - Россия	kirmrik
Тур - Италия	User1

Отзывы:



Имя: kirmrik
Дата начала: March 1, 2021
Дата окончания: March 6, 2021
Рейтинг: 9
Комментарий: Все отлично, погода немного подвела

Оставить отзыв:



Username:

Start date:

End date:

Rate:

Comment:

Вывод:

В результате выполнения лабораторной работы №2 был реализован простой сайт для турфирмы, с регистрацией разных пользователей, бронированием/ удалением/ изменением различных туров, создание отзывов, просмотр проданных туров; отзывов. Получены практические навыки работы с Django.