

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет «Инфокоммуникационных технологий»  
Направление подготовки «45.03.04 Интеллектуальные системы в гуманитарной сфере»

**О Т Ч Е Т**

**по лабораторной работе №2**

**Тема задания:** Реализация простого сайта средствами Django

**Выполнил:**

**Студент** Ларионова А. В. К33422  
(Фамилия И.О.) номер группы

**Проверил:**

**Преподаватель** Говоров А. И.  
(Фамилия И.О.)

**Санкт-Петербург  
2020**

## Цель работы

Овладеть практическими навыками и умениями реализации web-серверов средствами Django.

## Описание варианта (№4):

Список туров туристической фирмы

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:

### Описание модели данных

- Сущность «Пользователь»

Имя пользователя, паспортные данные и электронная почта

```
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 «Тур»

Название тура, описание, турагентство, период тура, страна и способ оплаты

```
class Tour(models.Model):
    PAY = (
        ("card", "card"),
        ("cash", "cash")
    )
    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=4, choices=PAY)
```

- Сущность «Отзыв»

Имя пользователя (FK), период тура, комментариев и рейтинг (1-10)

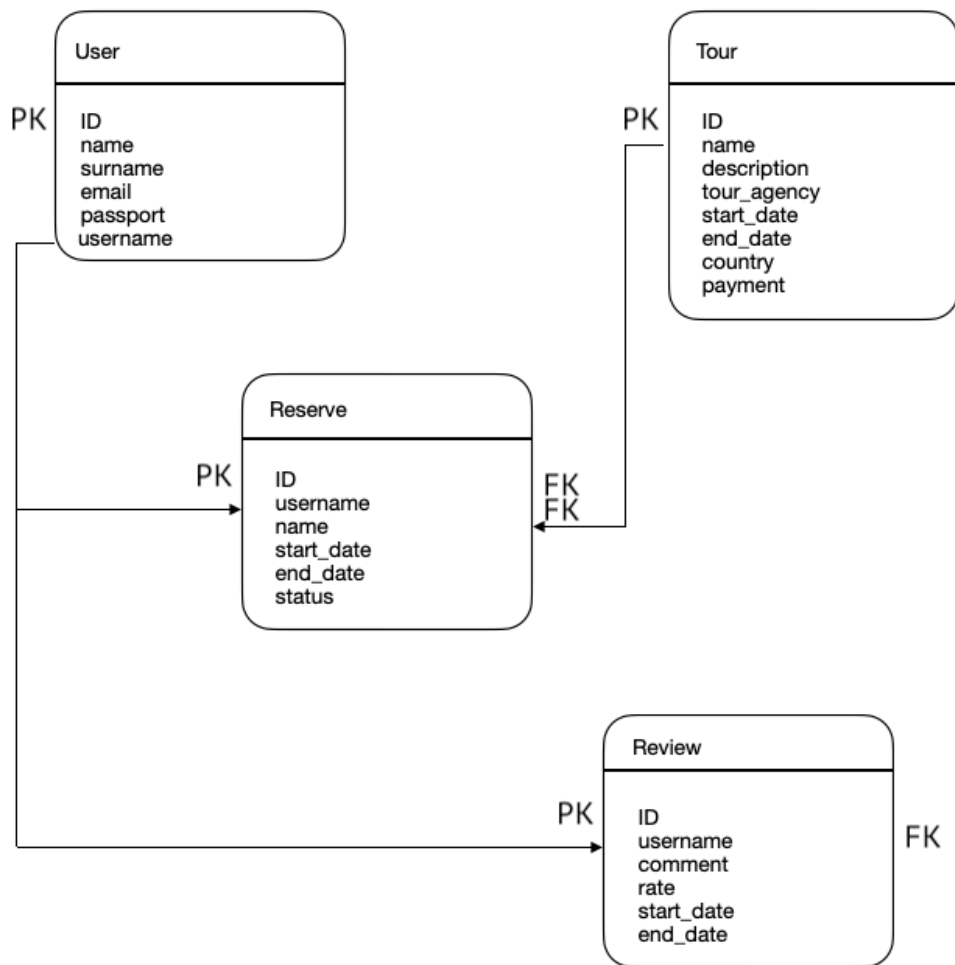
```
class Review(models.Model):
    MARKS = (
        ("1", "1"),
        ("2", "2"),
        ("3", "3"),
        ("4", "4"),
        ("5", "5"),
        ("6", "6"),
        ("7", "7"),
        ("8", "8"),
        ("9", "9"),
        ("10", "10")
    )
    start_date = models.DateField()
    end_date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

- Связующая сущность «Резервирование тура»

Имя пользователя (FK), название тура (FK), период тура и статус резервирования (подтверждается админом)

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

## Модель базы данных



## Описание контроллеров в файле views

Функции: основная страница сайта и страница профиля авторизованного пользователя

```
from django.shortcuts import render, redirect, get_object_or_404
from django.views.generic import CreateView, ListView, UpdateView, DeleteView
from .form import *
from .models import *
from django.contrib.auth import authenticate, login

def main(request):
    return render(request, 'main.html')

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

Регистрация нового пользователя. Осуществляется проверка данных, при успешном завершении – создание пользователя

```
def register(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)

        if form.is_valid():
            form.save()
            username = form.cleaned_data['username']
            password = form.cleaned_data['password1']
            user = authenticate(username=username, password=password)
            login(request, user)
            return redirect('/profile')

    else:
        form = RegistrationForm()

    visual = {'form': form}
    return render(request, 'registration/register.html', visual)
```

Форма регистрации

```
class RegistrationForm(UserCreationForm):
    class Meta:
        model = User
        fields = [
            'username',
            'password1',
            'password2',
            'email',
            'first_name',
            'last_name',
            'passport'
        ]
```

Список доступных туров

```
def tourslist(request):
    visual = {"tours": Tour.objects.all()}
    return render(request, 'tours.html', visual)
```

Список зарезервированных туров

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

## Резервирование тура для авторизованного пользователя

```
class CreateReserve(CreateView):
    form_class = CreateReserveForm
    model = Reserve
    template_name = 'reserve.html'
    context_object_name = 'reserve'
    success_url = '/profile'

    def get_initial(self):
        initial = super(CreateReserve, self).get_initial()
        initial = initial.copy()
        initial['username'] = self.request.user.pk
        initial['name'] = get_object_or_404(Tour, pk=self.kwargs['pk'])
        return initial
```

## Форма резервирования

```
class CreateReserveForm(forms.ModelForm):
    class Meta:
        model = Reserve
        fields = (
            'username',
            'name',
            'start_date',
            'end_date'
        )

    def __init__(self, *args, **kwargs):
        super(CreateReserveForm, self).__init__(*args, **kwargs)
        self.fields['username'].disabled = True
        self.fields['name'].disabled = True
```

## Оставление отзыва для авторизованного пользователя

```
class CreateReview(CreateView):
    form_class = CreateReviewForm
    model = Review
    template_name = 'review.html'
    context_object_name = 'review'
    success_url = '/reviews'

    def get_initial(self):
        initial = super(CreateReview, self).get_initial()
        initial = initial.copy()
        initial['username'] = self.request.user.pk
        return initial
```

## Форма оставления отзыва

```
class CreateReviewForm(forms.ModelForm):
    class Meta:
        model = Review
        fields = (
            'username',
            'start_date',
            'end_date',
            'rate',
            'comment'
        )

    def __init__(self, *args, **kwargs):
        super(CreateReviewForm, self).__init__(*args, **kwargs)
        self.fields['username'].disabled = True
```

## Список всех отзывов

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

## Список зарезервированных туров в личном кабинете авторизованного пользователя

```
class listreserves(ListView):
    template_name = 'myreserves.html'
    context_object_name = 'reserves_list'

    def get_queryset(self):
        self.user = self.request.user.pk
        return Reserve.objects.filter(username=self.user)
```

## Удаление зарезервированного тура в личном кабинете авторизованного пользователя

```
class deletereserveView(DeleteView):
    model = Reserve
    template_name = 'deletereserve.html'
    context_object_name = 'reserve'
    success_url = '/profile'
```

Обновление зарезервированного тура в личном кабинете авторизованного пользователя

```
class updatereserveView(UpdateView):
    model = Reserve
    fields = ['start_date', 'end_date']
    template_name = 'updatereserve.html'
    context_object_name = 'reserve'
    success_url = '/profile'
```

## Описание роутеров файла urls

Urls проекта, где accounts/ отвечают за авторизацию пользователей

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('django.contrib.auth.urls')),
    path('', include('toursite.urls'))
]
```

Urls приложения, где реализованы основные страницы сайта


```
from django.urls import path
from . import views
from .views import *

urlpatterns = [
    path('', views.main),
    path('profile/', views.profile),
    path('myreserves/', listreserves.as_view(), name='reserves'),
    path('myreserves/myreserves/deletereserve/<int:pk>/', views.deletereserveView.as_view()),
    path('myreserves/myreserves/updatereserve/<int:pk>/', views.updatereserveView.as_view()),
    path('register/', register, name='register'),
    path('tours/', views.tourslist),
    path('tours/<int:pk>/reserve/', CreateReserve.as_view(), name='reserve'),
    path('review/', CreateReview.as_view(), name='review'),
    path('soldtours/', views.soldtourslist),
    path('reviews/', views.reviewslist)
]
```



## Скрины работы программы

Главная страница неавторизованного пользователя



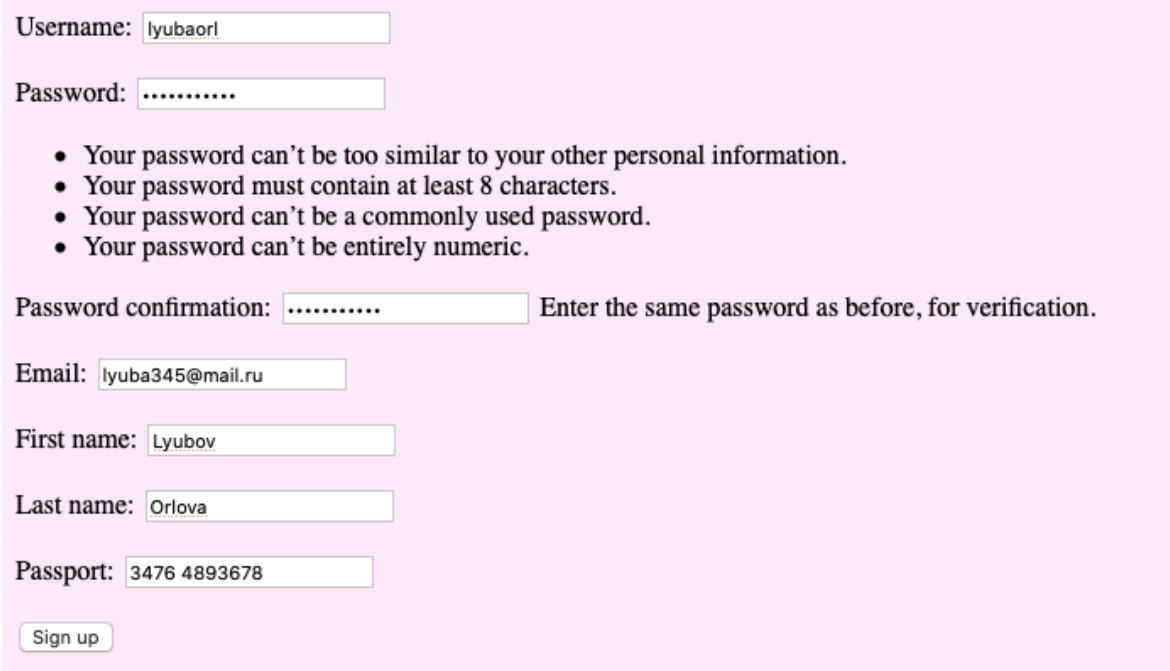
Not logged in

Sign up Login

Available tours All reviews

Give feedback Sold tours

Регистрация нового пользователя



Username:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

Email:

First name:

Last name:

Passport:

Главная страница авторизованного пользователя и его профиль

## Logged in as lanastya

[Sign up](#)[Login](#)[Available tours](#)[All reviews](#)[Give feedback](#)[Sold tours](#)

### lanastya's profile

**Username:** lanastya

**Email:** lanastya@gmail.com

**Name:** Nastya

**Surname:** Larionova

**Passport:** None

[To my bookings](#)

Зарезервированные туры в личном кабинете

### lanastya's bookings

**Tour:** Romantic Paris

**Start\_date:** Dec. 2, 2020

**End\_date:** Dec. 5, 2020

**Booking status:** Wait for the admin's reply

[Delete](#)[Update](#)

**Tour:** Perfect Apollo

**Start\_date:** Nov. 15, 2020

**End\_date:** Nov. 21, 2020

**Booking status:** Booked succesfully

[Delete](#)[Update](#)

[Back to profile](#)

[Main page](#)

### Обновление дат зарезервированного тура

**Start date:**

**End date:**

### Удаление зарезервированного тура

Do you want to delete your booking "Reserve object (16)"?

[Back to my bookings](#)

### Доступные туры для резервирования

ID: 1

[Book now!](#)

**Name:** Perfect Apollo

**Description:** Exploring the archaeological area you will meet the Temple of Apollo (built on a rocky hill)

**Tour agency:** Passion for travel

**Starts on** Nov. 1, 2020

**Ends on** Nov. 7, 2020

**Country:** Greece

**Payment method:** card

ID: 2

[Book now!](#)

**Name:** Romantic Paris

**Description:** Explore the district of Notre Dame, Sorbonne University and the Pantheon

**Tour agency:** Passion for travel

**Starts on** Nov. 16, 2020

**Ends on** Nov. 19, 2020

**Country:** France

**Payment method:** cash

ID: 3

[Book now!](#)

**Name:** Magical Helsinki

**Description:** Things to see in Helsinki during the winter months

**Tour agency:** Passion for travel

**Starts on** Dec. 1, 2020

**Ends on** Dec. 5, 2020

**Country:** Finland

**Payment method:** card

## Резервирование тура для авторизованного пользователя

Username:

Name:

Start date:

End date:

## Примеры подтвержденных и не подтвержденных админом туров

**Booking status:** Booked succesfully

**Booking status:** Wait for the admin's reply

## Отзывы пользователей о турах

**Username:** lanastya

**Started on** Dec. 1, 2019

**Ended on** Dec. 5, 2019

**Rate:** 9

**Comment:** I liked everything about my stay & tour agency! Thanks for a warm trip.

**Username:** georgebov

**Started on** Feb. 3, 2019

**Ended on** Feb. 15, 2019

**Rate:** 8

**Comment:** Your tour agency was a hidden gem for me. Thank you!

**Username:** coolguy

**Started on** Nov. 1, 2019

**Ended on** Nov. 7, 2019

**Rate:** 4

**Comment:** I'm sad there was no free sweets.

**Username:** sofya

**Started on** April 3, 2019

**Ended on** April 15, 2019

**Rate:** 7

**Comment:** I really enjoyed my trip to Bahamas. Can't wait to book a new one!

Оставление отзыва от авторизованного пользователя

Username:

Start date:

End date:

Rate:

Comment:

Таблица с зарезервированными турами и их обладателями

Booked tours	Happy owners
Romantic Paris	lanastya
Romantic Paris	coolguy
Perfect Apollo	annatour
Magical Helsinki	prettymari
Perfect Apollo	lanastya
Perfect Apollo	sofya

## Выводы

В результате выполнения лабораторной работы были получены практические навыки и умения реализации web-серверов средствами Django; создан простой сайт для туристической фирмы.