

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

Факультет «Инфокоммуникационных технологий»  
Направление подготовки «09.03.03 Мобильные и сетевые технологии»

## **О Т Ч Е Т**

---

**Тема задания:** Веб-приложение на Django Framework

---

**Выполнил:**  
Студент Шкикавый И.Я. К33402

**Проверил:**  
Преподаватель Говоров А. И.

**Санкт-Петербург  
2020**

#### Вариант 4:

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.

**Цель:** создать веб-приложение средствами Django Framework

#### Рисунок 1 - models.py

```
class User(AbstractUser):
    username = models.CharField(max_length=15, blank=True, null=True, unique=True)
    passport = models.CharField(max_length=100, blank=True, null=True)
    email = models.CharField(max_length=40, blank=True, null=True)
    tour = models.ManyToManyField('Tour', through="Reservation")

class Tour(models.Model):
    PAYMENT = [(('Банковская карта', 'Банковская карта'), ('Наличные деньги', 'Наличные деньги')), (('Веб-сервисы', 'Веб-сервисы'))]
    name = models.CharField(max_length=80)
    tour_agency = models.CharField(max_length=80)
    country = models.CharField(max_length=50)
    begin_date = models.DateField()
    end_date = models.DateField()
    description = models.CharField(max_length=500)
    pay = models.CharField(max_length=25, choices=PAYMENT)

class Comment(models.Model):
    RATING = [(1, '1'), (2, '2'), (3, '3'), (4, '4'), (5, '5'), (6, '6'), (7, '7'), (8, '8'), (9, '9'), (10, '10'), ]
    begin_date = models.DateField()
    end_date = models.DateField()
    commentator = models.ForeignKey(User, on_delete=models.CASCADE)
    text = models.CharField(max_length=500)
    rating = models.IntegerField(choices=RATING)

class Reservation(models.Model):
    username = models.ForeignKey(User, on_delete=models.CASCADE)
    nameoftour = models.ForeignKey(Tour, on_delete=models.CASCADE)
    begin_date = models.DateField(blank=True, null=True)
    end_date = models.DateField(blank=True, null=True)
    status = models.BooleanField(blank=True, null=True)
```

#### Рисунок 2 - urls.py

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

urlpatterns = [
    path('', views.main),
    path('lc/', views.lc),
    path('register/', register, name='register'),
    path('reservedtour/', views.reservedtourlist),
    path('commentlist/', views.commentlist),
    path('createcomment/', CreateComment.as_view(), name='comment'),
    path('tours/', views.tourlist),
    path('tours/<int:pk>/reservation', CreateReservation.as_view(), name='reservation'),
    path('lcreservations/', listreservations.as_view(), name='reservations'),
    path('lcreservations/lcreservations/deleterreservation/<int:pk>/', views.DeleteReserveView.as_view()),
    path('lcreservations/lcreservations/updaterreservation/<int:pk>/', views.UpdateReserveView.as_view())
]
```

## Рисунок 3 - forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from .models import *

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

class CreateCommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = ('commentator', 'begin_date', 'end_date', 'rating', 'text')

    def __init__(self, *args, **kwargs):
        super(CreateCommentForm, self).__init__(*args, **kwargs)
        self.fields['commentator'].disabled = True

class CreateReservationForm(forms.ModelForm):
    class Meta:
        model = Reservation
        fields = ('username', 'nameoftour', 'begin_date', 'end_date')

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

## Рисунок 4 - views.py

```
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 lc(request):
    return render(request, 'lc.html')

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('/lc')
    else:
        form = RegistrationForm()
        visual = {'form': form}
        return render(request, 'registration/signin.html', visual)
```

```

class CreateReservation(CreateView):
    form_class = CreateReservationForm
    model = Reservation
    template_name = 'reservation.html'
    context_object_name = 'reservation'
    success_url = '/lc'

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

class UpdateReserveView(UpdateView):
    model = Reservation
    fields = ['begin_date', 'end_date']
    template_name = 'upreservation.html'
    context_object_name = 'reservation'
    success_url = '/lc'

class DeleteReserveView(DeleteView):
    model = Reservation
    template_name = 'delreservation.html'
    context_object_name = 'reservation'
    success_url = '/lc'

def reservedtourlist(request):
    visual = {"reservations": Reservation.objects.all()}
    return render(request, 'reservedtour.html', visual)

class listreservations(ListView):
    template_name = 'lcreservations.html'
    context_object_name = 'reservation_list'
    def get_queryset(self):
        self.user = self.request.user.pk
        return Reservation.objects.filter(username=self.user)

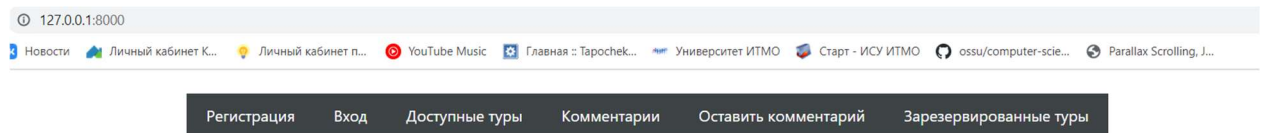
def commentlist(request):
    visual = {"comments": Comment.objects.all()}
    return render(request, 'comments.html', visual)

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

class CreateComment(CreateView):
    form_class = CreateCommentForm
    model = Comment
    template_name = 'comment.html'
    context_object_name = 'comment'
    success_url = '/comment'
    def get_initial(self):
        initial = super(CreateComment, self).get_initial()
        initial = initial.copy()
        initial['commentator'] = self.request.user.pk
        return initial

```

## Рисунок 5 - главная страница сайта



## Рисунок 6 - окно регистрации

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:

## Рисунок 7 - окно авторизации

Логин:

Пароль:

## Рисунок 8 - личный кабинет

**Ник:** pushin

**Электронная почта:** pushkin@mail.ru

**Имя:** Александр

**Фамилия:** Пушкин

**Паспорт:** None

[Мои брони](#)

## Рисунок 9 - Кабинет брони пользователя

# Кабинет pushin'a

**Тур:** Суровый Берлин

**Начало брони:** Nov. 5, 2020

**Конец брони:** Nov. 15, 2020

**Статус брони:** Ждите ответа администратора

Удалить

Обновить

[Назад в ЛК](#)

[Главная страница](#)

## Рисунок 10 - лента всех заказов

---

# Лента заказов

Суровый Берлин

pushin

## Рисунок 11 - доступные туры

---

ID: 1

[Book now!](#)

**Название:** Суровый Берлин

**Описание:** Погрузитесь в суровую историю Берлина, сердца Германии

**Турагентство:** Самолёт

**Начинается:** Nov. 17, 2020

**Заканчивается:** Nov. 20, 2020

**Страна:** Германия

**Метод оплаты:** Банковская карта

## Рисунок 12 - страница комментариев

**Ник:** а

**Воспользовался услугой от:** Jan. 1, 2020

**До:** Jan. 7, 2020

**Оценка:** 8

**Комментарий:** Было неплохо

**Ник:** pushin

**Воспользовался услугой от:** Nov. 17, 2020

**До:** Nov. 25, 2020

**Оценка:** 8

**Комментарий:** Всё было окейно, хорошая компания

**Рисунок 13 - окно оставления комментария**

Commentator:

Begin date:

End date:

Rating:

Text:

**Рисунок 14 - окно оформления брони**

---

Username:

Nameoftour:

Begin date:

End date:

**Вывод:** в ходе выполнения лабораторной работы мы научились делать простое веб-приложение на Django Framework, закрепив знания, полученные при выполнении практической работы.