

Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики.  
Факультет инфокоммуникационных технологий.

Лабораторная работа №2

Выполнила студентка группы К33402 Куценко В.С.  
Преподаватель: Говоров А.И.

Санкт-Петербург, 2020

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

### **Ход работы**

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.

## Описание модели данных.

Пользователь – содержит логин, электронную почту, имя, фамилию и паспортные данные.

```
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")
```

Тур – название, описание, турагенство, даты проведения, страна, описание и способ оплаты.

```
class Tour(models.Model):
    PAYMENT = [('Банковская карта', 'Банковская карта'), ('Наличные', 'Наличные')]
    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)
```

Отзыв – имя пользователя, даты, рейтинг, комментарий.

```
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)
tour = models.ForeignKey("Tour", on_delete=models.CASCADE)
E)
text = models.CharField(max_length=500)
rating = models.IntegerField(choices=RATING)

```

Резерв тура – имя пользователя, название тура, даты и статус.

```

class Reservation(models.Model):
    username = models.ForeignKey("User", on_delete=models.CA
SCADE)
    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)

```

## Описание контроллеров в файле views

Регистрация пользователя. Проверка данных и 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('/lc')
        else:
            form = RegistrationForm()

```

```

        visual = {'form': form}
        return render(request, 'registration/signin.html', visual)
1)

```

Создание бронирования

```

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

```

Обновление бронирования

```

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

```

Удаление бронирования

```

class DeleteReserveView(DeleteView):

```

```
model = Reservation
template_name = 'delreservation.html'
context_object_name = 'reservation'
success_url = '/lc'
```

Зарегистрированные туры

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

Написание комментария

```
class CreateComment(CreateView):
    form_class = CreateCommentForm
    model = Comment
    template_name = 'comment.html'
    context_object_name = 'comment'
    success_url = '/tours'
    def get_initial(self):
        initial = super(CreateComment, self).get_initial()
        initial = initial.copy()
        initial['commentator'] = self.request.user.pk
        initial['tour'] = get_object_or_404(Tour, pk=self.kwargs['pk'])
        return initial
```

Список комментариев:

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

urls

```
path('', views.main),
path('lc/', views.lc),
path('register/', register, name='register'),
path('reservedtour/', views.reservedtourlist),
path('commentlist/', views.commentlist),
path('tours/<int:pk>/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())
```

Получившиеся страницы:

Главная страница

[Регистрация](#)

[Вход](#)

[Доступные туры](#)

[Зарезервированные туры](#)

## Регистрация

---

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:

Sign in



Вход

---

Логин:

Пароль:

Войти

Профиль

**Ник:** lena

**Электронная почта:**

**Имя:**

**Фамилия:**

**Паспорт:** None

[Мои брони](#)

## Просмотр доступных туров

ID: 1

[Book now!](#)

**Название:** Исторический Петербург

**Описание:** Красиво и исторически

**Турагентство:** Спбавиа

**Начинается:** July 3, 2021

**Заканчивается:** July 10, 2021

**Страна:** Россия

**Метод оплаты:** Наличные

[Оставить комментарий](#)

**Ник:** lena

**Воспользовался услугой от:** July 3, 2021

**До:** July 10, 2021

**Оценка:** 9

**Комментарий:** узнала много про историю

ID: 2

[Book now!](#)

**Название:** Легендарная Москва

**Описание:** Легендарно и эпатажно

**Турагентство:** Мскавиа

**Начинается:** March 19, 2021

**Заканчивается:** March 28, 2021

**Страна:** Россия

**Метод оплаты:** Банковская карта

## Бронирование тура

Username:

Name of tour:

Begin date:

End date:

Просмотр забронированных туров:

## Кабинет lena

**Тур:** Исторический Петербург

**Начало брони:** July 3, 2021

**Конец брони:** July 10, 2021

**Статус брони:** Ждите ответа администратора

[Назад в ЛК](#)

[Главная страница](#)

Оставить комментарий:

Commentator: lena ▼

Tour: Исторический Петербург ▼

Begin date: 2021-07-03

End date: 2021-07-10

Rating: 9 ▼

Text: узнала много про истори

Публикация