

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту
по дисциплине «Инженерное проектирование»

Тема: «Продажа билетов на мероприятия»

Выполнил:

Студент группы 211-322

Сундуреев Т.Ю.

Принял:

Старший преподаватель

Даньшина М.В.

Голубева И.В.

Москва 2022

Оглавление

Введение.....	Error! Bookmark not defined.
Архитектура проекта	3
Приложение main	3
Приложение account	3
Приложение events	3
Заключение	4

Архитектура проекта

Проект разделен на 3 приложения:

- ‘main’ — управление домашней страницей, шаблонов и статичной информации;
- ‘events’ — вывод списка событий, их подробный просмотр, фильтрация по категориям и бронирование;
- ‘account’ — управление авторизацией и регистрацией.

Приложение main

В данном приложении находится базовый шаблон от которого наследуются другие шаблоны с помощью возможности Jinja - `{% extends 'main/layout.html' %}`. Здесь также находится домашняя страница `home.html`, сообщение об ошибке 404 `error_404.html` и информация “О нас” `about-us.html`. Дизайн всех страниц представлен в макете Figma.

Приложение account

В данном приложении реализован авторизация и возможность зарегистрироваться новому пользователю. Имеются шаблоны `login.html` и `register.html`. Также используются формы Django в файле `forms.py` `LoginForm` и `RegisterForm`.

При неправильном вводе пароля или логина при авторизации или регистрации будет выводиться соответствующая информация в виде ошибок.

Приложение events

В приложении ‘events’ реализован показ списка всех событий, показ существующих категорий, возможность подробного просмотра отдельных событий, бронирование нескольких мероприятий и изменять количество билетов.

Тут же определены 5 моделей:

- Event — содержит в себе всю информацию о мероприятии (выступающий, место проведения, время, фото, дата и т.д.)
- Category — модель для определения категорий (названия и слагги (slug) для ЧПУ)
- Order — модель для бронирования мероприятий, заказ
- OrderItem — описаны конкретные события, которые мы добавили в заказ
- Customer — покупатель

Реализованные пункты

- Django REST framework

```
@api_view(['GET'])
def eventList(request):
    events = Event.objects.all()
    serializer = EventSerializer(events, many=True)
    return Response(serializer.data)

@api_view(['GET'])
def eventDetail(request, pk):
    event = Event.objects.get(id=pk)
    serializer = EventSerializer(event, many=False)
    return Response(serializer.data)

@api_view(['POST'])
def eventCreate(request):
    serializer = EventSerializer(data=request.data)
    if serializer.is_valid():
        serializer.save()
    return Response(serializer.data)

@api_view(['POST'])
def eventUpdate(request, pk):
    event = Event.objects.get(id=pk)
    serializer = EventSerializer(instance=event, data=request.data)

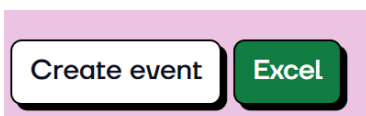
    if serializer.is_valid():
        serializer.save()
```

- Фильтрация

```
def show_events(cat_selected=""):
    if cat_selected:
        events = Event.objects.filter(category__slug=cat_selected)
        if len(events) == 0:
            raise Http404()
        return {'events': events}
    else:
        events = Event.objects.all()
        return {'events': events}
```

- Экспорт данных в Excel

По нажатии на кнопку происходит экспорт данных в events.xlsx



```
def export_data_to_excel(request):
    objs = Event.objects.all()
    data = []

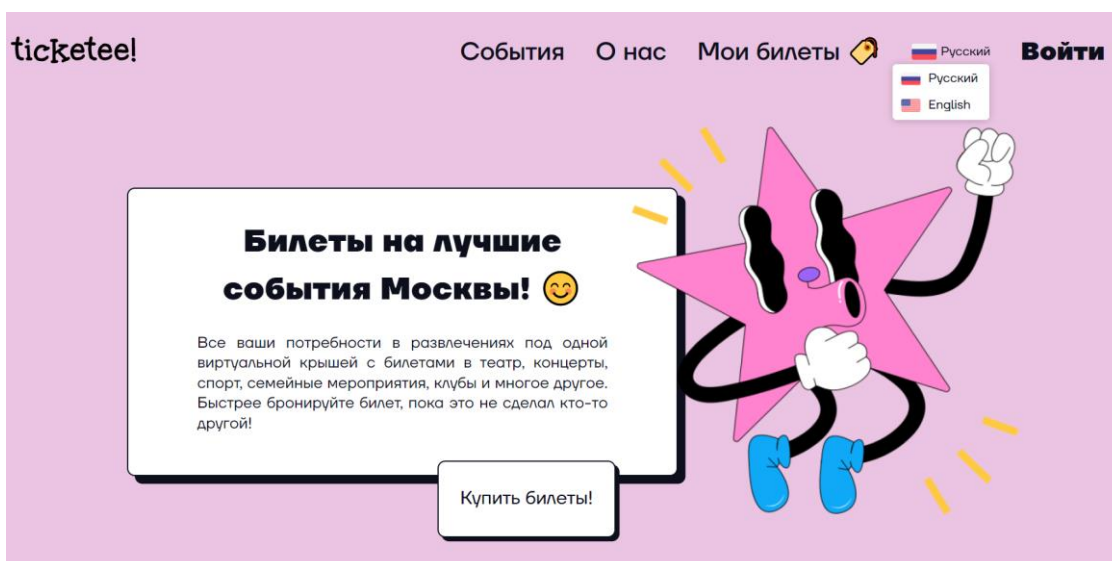
    for obj in objs:
        data.append({
            "Название": obj.name,
            "Описание": obj.description,
            "Выступающий": obj.performer,
            "Дата": obj.date,
            "Время": obj.time,
            "Место": obj.location,
            "Цена": obj.price,
        })

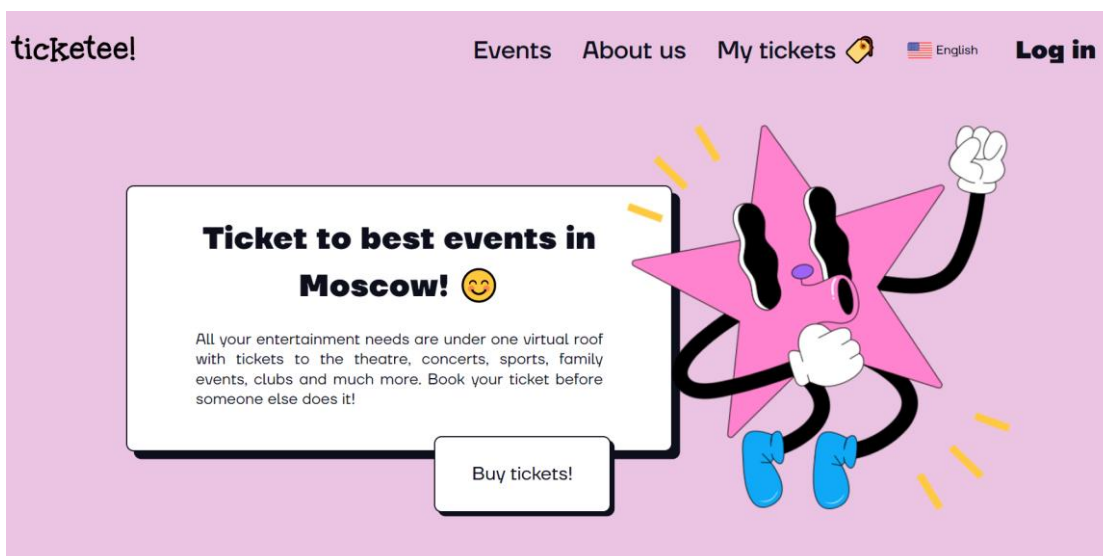
    pd.DataFrame(data).to_excel('events.xlsx')
    return redirect('events')
```

	A	B	C	D	E	F	G	H
		Название	Описание	Выступающий	Дата	Время	Место	Цена
0		The Phantom	The Phantom	Актёрский	2023-02-18	18:00:00	Театр МД	1000
1		Andrew H	Dr. Andrew	Эндрю Х	2023-06-05	14:00:00	Парк «Зар	2500
2		EPICENTER	Epic Sport	16 команд	2023-06-22	00:00:00	ЦСКА Ар	1000
3		Jordan Pet	Jordan Pet	Джордан	2023-08-16	14:00:00	Библиоте	2000
4		Bob Dylan	Bob Dylan	Боб Дила	2023-10-13	19:00:00	Большой	5000
5		Matthew W	Matthew F	Мэттью У	2023-10-24	14:00:00	Научно-п	1000
6		London Gr	London Gr	London Gr	2024-06-10	18:00:00	Main Stage	2750
7		Hayao Miy	Within two	Хаяо Мия	2025-06-15	19:00:00	Кинотеатр	500
8		Bob Dylan	"I'd come	Боб Дила	2025-08-16	20:00:00	Библиоте	2000

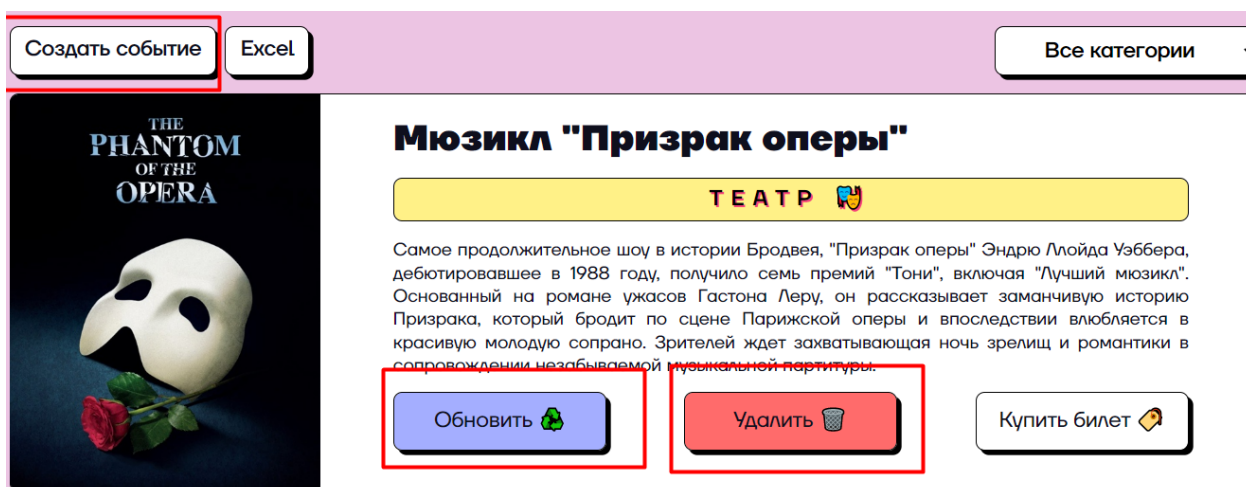
- Переключение языка (Использование Django Model Translation)

Блоки {% translate %} + makemessages + compilemessages





- CRUD на сайте



```
def create_event(request):
    if request.method == 'POST':
        form = AddEventForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('home')
    else:
        form = AddEventForm()
    context = {'form': form, 'title': 'Добавить новое событие', 'action': 'Добавить'}
    return render(request, 'events/create_event.html', context)
```

```
def update_event(request, pk):
    event = Event.objects.get(pk=pk)
    form = AddEventForm(instance=event)

    if request.method == 'POST':
        form = AddEventForm(request.POST, request.FILES, instance=event)
        if form.is_valid():
            form.save()
            return redirect('home')
    context = {'form': form, 'title': 'Редактировать событие', 'action': 'Сохранить', 'event': event}
    return render(request, 'events/update_event.html', context)
```

```
def delete_event(request, pk):  
    event = Event.objects.get(pk = pk)  
    event.delete()  
    return redirect('events')
```

Заключение

По полученной предметной области “Продажа билетов на мероприятия” был создан многостраничный сайт с БД.

При разработке были использованы следующие ресурсы:

- <https://www.youtube.com/c/DennisIvy>
- <https://django.fun/>
- <https://www.youtube.com/c/JustDjango>
- https://www.youtube.com/c/selfedu_rus

Проект доступен на GitHub по [ссылке](#).

Макет в Figma доступен по [ссылке](#).