Приложение Б. Текст программы

АННОТАЦИЯ

В данном программном документе приведён текст программы для программного комплекса для доставки посылок «DlBox».

В разделе «Текст программы» указано наименование программы, область применения программы, модули программы в виде таблицы с указанием описания и размера каждого модуля, код программы.

СОДЕРЖАНИЕ

[1. ТЕКСТ ПРОГРАММЫ 5](#_Toc167057548)

[1.1. Наименование объекта 5](#_Toc167057549)

[1.2. Область применения объекта 5](#_Toc167057550)

[1.3. Модули 5](#_Toc167057551)

[1.4. Код программного комплекса 8](#_Toc167057552)

[1.4.1. Код веб-сайта 8](#_Toc167057553)

[1.4.2. Код мобильного приложения 49](#_Toc167057595)

1. ТЕКСТ ПРОГРАММЫ
   1. Наименование объекта

Наименование - «Программный комплекс для доставки посылок DlBox».

* 1. Область применения объекта

Программа «DlBox» предназначена для автоматизации процесса заказа доставки посылки пользователем, а также работы сотрудников. Она будет использоваться в различных организациях для оптимизации внутренних процессов.

* 1. Модули

Таблица 1. Модули программного комплекса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название | Описание | Количество строк | Размер файла в Кб |
| 1 | 2 | 3 | 4 | 5 |
| Модули веб-сайта | | | | |
| 1 | admin.py | Регистрация моделей | 17 | 1 |
| 2 | forms.py | Создание форм для ввода данных | 101 | 3 |
| 3 | models.py | Создание моделей | 174 | 8 |
| 4 | routing.py | Настройка маршрутизации | 102 | 6 |
| 5 | serializers.py | Cериализация моделей для API | 75 | 2 |
| 6 | views.py | Определение представлений | 893 | 32 |
| 7 | admin\_BoxCRUD.html | Вывод всех элементов таблицы Box | 23 | 1 |
| 8 | admin\_BoxDetailView.html | Вывод одного объекта таблицы Box | 27 | 1 |
| 9 | admin\_index.html | Главная страница администратора | 60 | 1 |
| 10 | admin\_OrderCRUD.html | Вывод всех элементов таблицы Order | 22 | 1 |
| 11 | admin\_OrderDetailView.html | Вывод одного объекта таблицы Order | 31 | 1 |
| 12 | admin\_RoleCRUD.html | Вывод всех элементов таблицы Role | 22 | 1 |
| 13 | admin\_RoleDetailView.html | Вывод одного объекта таблицы Role | 24 | 1 |
| 14 | admin\_SizePackCRUD.html | Вывод всех элементов таблицы SizePackage | 22 | 1 |
| 15 | admin\_SizePackDetailView.html | Вывод одного объекта таблицы SizePackage | 24 | 1 |
| 16 | admin\_StatusOrderCRUD.html | Вывод всех элементов таблицы StatusOrder | 22 | 1 |
| 17 | admin\_StatusOrderDetalView.html | Вывод одного объекта таблицы StatusOrder | 24 | 1 |
| 18 | admin\_TypeOfDelveryCRUD.html | Вывод всех элементов таблицы TypeOfDelivery | 22 | 1 |
| 19 | admin\_TypeOfDeliveryDetailView.html | Вывод одного объекта таблицы TypeOfDelivery | 25 | 1 |
| 20 | admin\_TypeOfPayCRUD.html | Вывод всех элементов таблицы TypeOfPay | 22 | 1 |
| 21 | admin\_TypeOfPayDetailView.html | Вывод одного объекта таблицы TypeOfPay | 24 | 1 |
| 22 | admin\_UserCRUD.html | Вывод всех элементов таблицы MyUser | 22 | 1 |
| 23 | admin\_UserDetailView.html | Вывод одного объекта таблицы MyUser | 29 | 2 |
| 24 | auth.html | Страница авторизации | 23 | 2 |
| 25 | box\_reg.html | Страница оформления посылки | 15 | 1 |
| 26 | create.html | Страница, принимающая формы | 15 | 1 |
| 27 | delete.html | Страница подтверждения удаления | 15 | 1 |
| 28 | error.html | Страница с выводом ошибок | 8 | 1 |
| 29 | index.html | Главная страница | 71 | 5 |
| 30 | info.html | Страница с информацией | 94 | 5 |
| 31 | order\_detail.html | Страница просмотра деталей заказа | 20 | 1 |
| 32 | order\_reg.html | Страница оформления заказа | 47 | 2 |
| 33 | status\_track\_order.html | Страница просмотра статуса доставки | 12 | 1 |
| 34 | tracking\_order.html | Страница ввода трек-номера для отслеживания | 40 | 1 |
| 35 | user\_orders.html | Страница заказов пользователя | 27 | 1 |
| 36 | adminHeader.html | Верхнее меню навигации администратора | 50 | 1 |
| 37 | footer.html | Нижнее меню навигации | 11 | 1 |
| 38 | header. html | Верхнее меню навигации | 44 | 4 |
| 39 | admin\_base.html | Базовый шаблон страниц администратора | 260 | 5 |
| 40 | base. html | Базовый шаблон страниц пользователя | 350 | 7 |
| 41 | useraccount.html | Страница аккаунта пользователя | 42 | 2 |
| Модули мобильного приложения | | | | |
| 1 | OrderAdapter.java | Класс-адаптер для работы со списком «Заказы» | 69 | 2 |
| 2 | TrackingAdapter.java | Класс-адаптер для работы со списком «Отслеживание» | 68 | 2 |
| 3 | ItemAdapter.java | Класс-адаптер для работы со списком «Пользователи» | 73 | 2 |
| 4 | MyUser.java | Класс-модель для работы с таблицей из базы данных «Пользователи» | 98 | 3 |
| 5 | Order.java | Класс-модель для работы с таблицей из базы данных «Заказы» | 124 | 3 |
| 6 | Tracking.java | Класс-модель для работы с таблицей из базы данных «Отслеживание» | 34 | 1 |
| 7 | MainActivity.java | Класс для авторизации пользователя | 107 | 4 |
| 8 | HomeActivity.java | Класс для навигации между окнами | 67 | 2 |
| 9 | AccountActivity.java | Класс аккаунта авторизованного пользователя | 44 | 2 |
| 10 | OrdersActivity.java | Класс для вывода списка заказов | 69 | 2 |
| 11 | TrackingActivity.java | Класс для вывода списка отслеживаний | 60 | 2 |
| 12 | UsersActivity.java | Класс для вывода списка пользователей | 65 | 2 |
| 13 | OrderDetailsActivity.java | Класс для вывода информации о заказе | 56 | 2 |
| 14 | UserDetailsActivity.java | Класс для вывода информации о пользователе | 42 | 2 |
| 15 | TrackingDetailsActivity.java | Класс для вывода информации об отслеживании | 117 | 4 |
| 16 | ApiClient.java | Класс для подключения к API | 30 | 1 |
| 17 | api.java | Класс-интерфейс для подключения к API | 39 | 1 |
| 18 | UserSaveInfo.java | Класс для сохранения данных авторизованного пользователя | 35 | 1 |
| 19 | activity\_main.xml | Разметка для авторизации пользователя | 52 | 2 |
| 20 | activity\_home.xml | Разметка для навигации | 73 | 3 |
| 21 | activity\_account.xml | Разметка для вывода данных авторизованного пользователя | 127 | 5 |
| 22 | activity\_orders.xml | Разметка для вывода заказов | 19 | 1 |
| 23 | activity\_tracking.xml | Разметка для вывода отслеживаний | 19 | 1 |
| 24 | activity\_users.xml | Разметка для вывода пользователей | 19 | 1 |
| 25 | activity\_order\_details.xml | Разметка для вывода данных заказа | 154 | 6 |
| 26 | activity\_tracking\_details.xml | Разметка для вывода данных отслеживания | 87 | 3 |
| 27 | activity\_user\_details.xml | Разметка для вывода данных пользователя | 118 | 4 |
| 28 | item\_card.xml | Разметка элементов списка пользователей | 25 | 1 |
| 29 | item\_order\_card.xml | Разметка элементов списка заказов | 25 | 1 |
| 30 | item\_tracking\_card.xml | Разметка элементов списка отслеживаний | 25 | 1 |

* 1. Код программного комплекса
     1. Код веб-сайта
        1. View.py

from django.contrib import messages  
from django.http import HttpResponseRedirect, Http404  
from django.shortcuts import render  
from django.shortcuts import render, redirect, get\_object\_or\_404  
from django.views import View  
  
from .forms import \*  
from django.views.generic import ListView, DetailView, UpdateView, CreateView, DeleteView  
from django.urls import reverse\_lazy, reverse  
from django.core.paginator import Paginator  
from django.contrib.auth.hashers import make\_password, check\_password  
from django.contrib.auth.models import User  
from django.core.mail import send\_mail  
  
import ssl  
import certifi  
  
from django.db import connection  
from django.shortcuts import render  
  
from rest\_framework.decorators import api\_view  
from rest\_framework.response import Response  
from django.contrib.auth.hashers import check\_password  
  
ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context  
ssl\_context = ssl.create\_default\_context(cafile=certifi.where())  
  
from rest\_framework import viewsets  
from .serializers import \*  
  
  
# ViewSet для работы с пакетами  
class PackageBoxViewSet(viewsets.ModelViewSet):  
 queryset = PackageBox.objects.all()  
 serializer\_class = PackageBoxSerializer  
  
  
# ViewSet для работы с типами доставки  
class TypeOfDeliveryViewSet(viewsets.ModelViewSet):  
 queryset = TypeOfDelivery.objects.all()  
 serializer\_class = TypeOfDeliverySerializer  
  
  
# ViewSet для работы с типами оплаты  
class TypeOfPayViewSet(viewsets.ModelViewSet):  
 queryset = TypeOfPay.objects.all()  
 serializer\_class = TypeOfPaySerializer  
  
  
# ViewSet для работы с размерами пакетов  
class SizePackageViewSet(viewsets.ModelViewSet):  
 queryset = SizePackage.objects.all()  
 serializer\_class = SizePackageSerializer  
  
  
# ViewSet для работы с пользователями  
class MyUserViewSet(viewsets.ModelViewSet):  
 queryset = MyUser.objects.all()  
 serializer\_class = MyUserSerializer  
  
  
# ViewSet для работы с мобильными пользователями  
class MyUserMobileViewSet(viewsets.ModelViewSet):  
 queryset = MyUser.objects.all()  
 serializer\_class = MyUserMobileSerializer  
  
  
# ViewSet для работы с ролями  
class RoleViewSet(viewsets.ModelViewSet):  
 queryset = Role.objects.all()  
 serializer\_class = RoleSerializer  
  
  
# ViewSet для работы со статусами заказов  
class StatusOrderViewSet(viewsets.ModelViewSet):  
 queryset = StatusOrder.objects.all()  
 serializer\_class = StatusOrderSerializer  
  
  
# ViewSet для работы с трекингами  
class TrackingViewSet(viewsets.ModelViewSet):  
 queryset = Tracking.objects.all()  
 serializer\_class = TrackingSerializer  
  
  
# ViewSet для работы с коробками  
class BoxViewSet(viewsets.ModelViewSet):  
 queryset = Box.objects.all()  
 serializer\_class = BoxSerializer  
  
 def get\_serializer\_context(self):  
 context = super().get\_serializer\_context()  
 context['request'] = self.request  
 return context  
  
  
# ViewSet для работы с заказами  
class OrderViewSet(viewsets.ModelViewSet):  
 queryset = Order.objects.all()  
 serializer\_class = OrderSerializer  
  
  
# ViewSet для работы с заказами пользователей  
class UserOrdersViewSet(viewsets.ModelViewSet):  
 queryset = UserOrders.objects.all()  
 serializer\_class = UserOrdersSerializer  
  
  
# TEMPLATES  
  
# Инициализация главной страницы  
def index(request):  
 return render(request, 'DlBox/index.html')  
  
  
# Инициализация главной страницы Администратора  
def admin\_index(request):  
 return render(request, 'DlBox/admin\_index.html')  
  
  
# Вывод данных пользователя из сессии  
def useraccount(request):  
 # Проверка, авторизован ли пользователь  
 if request.session.get('login'):  
 # Если пользователь авторизован, отображаем личный кабинет  
 return render(request, 'DlBox/useraccount.html')  
 else:  
 # Если пользователь не авторизован, перенаправляем на страницу авторизации  
 return redirect('authPage')  
  
  
# Вывод заказов пользователя  
def user\_orders(request):  
 # Получаем пользователя из сессии  
 user = request.session.get('login')  
 if user:  
 user = get\_object\_or\_404(MyUser, login=user)  
 # Получаем все заказы для этого пользователя  
 orders = Order.objects.filter(orders=user)  
 return render(request, 'DlBox/user\_orders.html', {'orders': orders})  
 else:  
 messages.error(request, 'User not found in session.')  
 return redirect('authPage')  
  
  
# Реализация выхода из сессии пользователя  
def logout\_view(request):  
 if request.session.get('login'):  
 request.session.flush()  
 messages.success(request, 'Вы успешно вышли из аккаунта.')  
 return redirect('authPage')  
 else:  
 messages.info(request, 'Вы не были в системе.')  
 return HttpResponseRedirect("error httpresponse logout view")  
  
  
# Инициализация страницы с выводом ошибок  
def error(request):  
 return render(request, 'DlBox/error.html')  
  
  
# Реализация функции отслеживания заказа  
def tracking(request):  
 if request.method == 'POST':  
 form = TrackingForm(request.POST)  
 if form.is\_valid():  
 tracking\_number = form.cleaned\_data['uniqueOrderNum']  
 try:  
 package = Tracking.objects.get(uniqueOrderNum=tracking\_number)  
 return render(request, 'DlBox/status\_track\_order.html', {'package': package})  
 except Tracking.DoesNotExist:  
 return render(request, 'DlBox/tracking\_order.html', {'form': form, 'error': 'Трек номер не найден'})  
 else:  
 form = TrackingForm()  
 return render(request, 'DlBox/tracking\_order.html', {'form': form})  
  
  
# Вывод статуса заказа  
def status\_track(request):  
 unique\_order\_num = request.session.get('unique\_order\_num')  
 return render(request, 'DlBox/status\_track\_order.html', {'unique\_order\_num': unique\_order\_num})  
  
  
# Реализация функции оформления посылки  
def box\_reg(request):  
 if request.method == 'POST':  
 form = UserBoxForm(request.POST)  
 if form.is\_valid():  
 # Получаем пользователя из сессии  
 user = request.session.get('login')  
 if user:  
 user = MyUser.objects.get(login=user)  
 box = form.save(commit=False)  
 box.user\_sender = user  
 box.save()  
 return redirect('orderRegPage', pk=box.pk)  
 else:  
 messages.error(request, 'User not found.')  
 return redirect('authPage')  
 else:  
 form = UserBoxForm()  
 return render(request, 'DlBox/box\_reg.html', {'form': form})  
  
  
# Добавление заказа в таблицу базы данных для отслеживания и назначение изначального статуса  
def create\_tracking\_record(order\_number):  
 # Получаем или создаем статус "Отправлен"  
 status, \_ = StatusOrder.objects.get\_or\_create(name='Оформлен')  
  
 # Создаем запись в таблице Tracking  
 Tracking.objects.create(  
 uniqueOrderNum=order\_number,  
 status=status  
 )  
 return "Теперь вы можете отслеживать свою посылку!"  
  
  
# Реализация оформления заказа и добавление его к заказам пользователя  
def order\_reg(request, pk):  
 box = get\_object\_or\_404(Box, pk=pk)  
 try:  
 order = Order.objects.get(box=box)  
 except Order.DoesNotExist:  
 order = None  
  
 if request.method == 'POST':  
 form = OrderForm(request.POST, instance=order)  
 if form.is\_valid():  
 order = form.save(commit=False)  
 order.box = box  
 order.save()  
 create\_tracking\_record(order.uniqueOrderNum)  
 user = request.session.get('login')  
 if user:  
 user = get\_object\_or\_404(MyUser, login=user)  
 # Создаем новый экземпляр UserOrders  
 user\_order = UserOrders(user=user, order=order)  
 user\_order.save()  
 return redirect('orderDetailPage', pk=order.pk)  
 else:  
 messages.error(request, 'User not found in session.')  
 return redirect('authPage')  
 else:  
 form = OrderForm(instance=order)  
  
 return render(request, 'DlBox/order\_reg.html', {'form': form, 'box': box})  
  
  
# Вывод информации о деталях заказа  
def order\_detail(request, pk):  
 order = get\_object\_or\_404(Order, pk=pk)  
 return render(request, 'DlBox/order\_detail.html', {'order': order})  
  
  
# Инициализация страницы для работы с forms.py  
def create(request):  
 if request.method == 'POST':  
 form = RoleForm(request.POST, request.FILES)  
 if form.is\_valid():  
 form.save()  
 return redirect('indexPage')  
 else:  
 form = RoleForm()  
 return render(request, 'DlBox/create.html', {'form': form})  
  
  
# Реализация функции регистрации пользователя и сохранение его в сессии  
def register(request):  
 if request.method == 'POST':  
 form = UserRegisterForm(request.POST)  
 if form.is\_valid():  
 user = form.save(commit=False)  
 # Получаем роль "Пользователь" из базы данных  
 user\_role = Role.objects.get(name='Пользователь')  
 # Присваиваем роль пользователю  
 user.role = user\_role  
 # Шифруем пароль перед сохранением  
 user.password = make\_password(user.password)  
 user.save()  
 username = form.cleaned\_data.get('login')  
 request.session['login'] = user.login  
 request.session['user\_name'] = user.user\_name  
 request.session['user\_surname'] = user.user\_surname  
 request.session['user\_patronymic'] = user.user\_patronymic  
 request.session['telephoneNumber'] = user.telephoneNumber  
 return redirect('indexPage')  
 else:  
 form = UserRegisterForm()  
 return render(request, 'DlBox/create.html', {'form': form})  
  
  
# Реализация функции авторизации пользователя и сохранение его в сессии  
def auth(request):  
 if request.method == 'POST':  
 form = LoginForm(request.POST)  
 if form.is\_valid():  
 username = form.cleaned\_data['login']  
 password = form.cleaned\_data['password']  
 try:  
 user = MyUser.objects.get(login=username)  
 if check\_password(password, user.password):  
 # Сохранение информации о пользователе в сеансе  
 request.session['login'] = user.login  
 request.session['user\_name'] = user.user\_name  
 request.session['user\_surname'] = user.user\_surname  
 request.session['user\_patronymic'] = user.user\_patronymic  
 request.session['telephoneNumber'] = user.telephoneNumber  
 request.session.modified = True  
  
 role\_user = Role.objects.get(name='Пользователь')  
 role\_admin = Role.objects.get(name='Администратор')  
  
 if user.role == role\_user:  
 return redirect('indexPage')  
 elif user.role == role\_admin:  
 return redirect('adminIndexPade')  
 else:  
 return redirect('errorPage')  
 else:  
 messages.error(request, 'Invalid username or password.')  
 except MyUser.DoesNotExist:  
 messages.error(request, 'Invalid username or password.')  
 else:  
 form = LoginForm()  
 return render(request, 'DlBox/auth.html', {'form': form})  
  
  
# MOBILE  
@api\_view(['POST'])  
def check\_password\_view(request):  
 login = request.data.get('login')  
 raw\_password = request.data.get('password')  
 user = get\_object\_or\_404(MyUser, login=login)  
 if user:  
 hash\_pass = make\_password(raw\_password)  
 if user.password == hash\_pass:  
 return Response(status=200)  
 else:  
 return Response(status=404)  
  
  
# AAAAAADMIIIIIIIIIINNNNNNNNNN  
  
  
# View для отображения списка пользователей  
class MyUserListView(ListView):  
 model = MyUser  
 template\_name = 'DlBox/admin\_UserCRUD.html'  
 context\_object\_name = 'list\_users'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 myusers = self.get\_queryset()  
 paginator = Paginator(myusers, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 myusers = paginator.get\_page(page)  
 context['myusers'] = myusers  
 return context  
  
  
# View для отображения детальной информации о пользователе  
class MyUserDetailView(DetailView):  
 model = MyUser  
 template\_name = 'DlBox/admin\_UserDetailView.html'  
  
  
# View для создания нового пользователя  
class MyUserCreateView(CreateView):  
 model = MyUser  
 form\_class = MyUserForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def form\_valid(self, form):  
 user = form.save(commit=False)  
 user.password = make\_password(user.password) # Хэширование пароля  
 user.save()  
 return super().form\_valid(form)  
  
  
# View для обновления информации о пользователе  
class MyUserUpdateView(UpdateView):  
 model = MyUser  
 form\_class = MyUserForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def form\_valid(self, form):  
 user = form.save(commit=False)  
 if 'password' in form.changed\_data: # Проверяем, изменился ли пароль  
 user.password = make\_password(user.password) # Хэширование пароля  
 user.save()  
 return super().form\_valid(form)  
  
  
# View для удаления пользователя  
class MyUserDeleteView(DeleteView):  
 model = MyUser  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("MyUserDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No MyUser matches the given query.")  
  
  
# ROOOOOLEEESSSSSSS  
# View для отображения списка ролей  
class RoleListView(ListView):  
 model = Role  
 template\_name = 'DlBox/admin\_RoleCRUD.html'  
 context\_object\_name = 'list\_roles'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 roles = self.get\_queryset()  
 paginator = Paginator(roles, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 roles = paginator.get\_page(page)  
 context['roles'] = roles  
 return context  
  
  
# View для отображения детальной информации о роли  
class RoleDetailView(DetailView):  
 model = Role  
 template\_name = 'DlBox/admin\_RoleDetailView.html'  
  
  
# View для обновления информации о роли  
class RoleUpdateView(UpdateView):  
 model = Role  
 form\_class = RoleForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания новой роли  
class RoleCreateView(CreateView):  
 model = Role  
 form\_class = RoleForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления роли  
class RoleDeleteView(DeleteView):  
 model = Role  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("RoleDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No Role matches the given query.")  
  
  
# SIZEEEEE PACKAGEEEE  
# View для отображения списка размеров пакетов  
class SizePackRoleListView(ListView):  
 model = SizePackage  
 template\_name = 'DlBox/admin\_SizePackCRUD.html'  
 context\_object\_name = 'list\_sizepack'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 sizepacks = self.get\_queryset()  
 paginator = Paginator(sizepacks, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 sizepacks = paginator.get\_page(page)  
 context['sizepacks'] = sizepacks  
 return context  
  
  
# View для отображения детальной информации о размере пакета  
class SizePackDetailView(DetailView):  
 model = SizePackage  
 template\_name = 'DlBox/admin\_SizePackDetailView.html'  
  
  
# View для обновления информации о размере пакета  
class SizePackUpdateView(UpdateView):  
 model = SizePackage  
 form\_class = SizePackForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания нового размера пакета  
class SizePackCreateView(CreateView):  
 model = SizePackage  
 form\_class = SizePackForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления размера пакета  
class SizePackDeleteView(DeleteView):  
 model = SizePackage  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("SizePackDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No SizePackage matches the given query.")  
  
  
# STAAAATUUUUSUSSSSS  
# View для отображения списка статусов заказов  
class StatusOrderListView(ListView):  
 model = StatusOrder  
 template\_name = 'DlBox/admin\_StatusOrderCRUD.html'  
 context\_object\_name = 'list\_status'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 statuses = self.get\_queryset()  
 paginator = Paginator(statuses, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 statuses = paginator.get\_page(page)  
 context['statuses'] = statuses  
 return context  
  
  
# View для отображения детальной информации о статусе заказа  
class StatusOrderDetailView(DetailView):  
 model = StatusOrder  
 template\_name = 'DlBox/admin\_StatusOrderDetailView.html'  
  
  
# View для обновления информации о статусе заказа  
class StatusOrderUpdateView(UpdateView):  
 model = StatusOrder  
 form\_class = StatusOrderForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания нового статуса заказа  
class StatusOrderCreateView(CreateView):  
 model = StatusOrder  
 form\_class = StatusOrderForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления статуса заказа  
class StatusOrderDeleteView(DeleteView):  
 model = StatusOrder  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("StatusOrderDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No StatusOrder matches the given query.")  
  
  
# TYPEEEEEEE OFFFFFFFFF DELIVERRRRSSS  
# View для отображения списка типов доставки  
class TypeOfDeliveryListView(ListView):  
 model = TypeOfDelivery  
 template\_name = 'DlBox/admin\_TypeOfDeliveryCRUD.html'  
 context\_object\_name = 'list\_typesofdeliv'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 types = self.get\_queryset()  
 paginator = Paginator(types, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 types = paginator.get\_page(page)  
 context['types'] = types  
 return context  
  
  
# View для отображения детальной информации о типе доставки  
class TypeOfDeliveryDetailView(DetailView):  
 model = TypeOfDelivery  
 template\_name = 'DlBox/admin\_TypeOfDeliveryDetailView.html'  
  
  
# View для обновления информации о типе доставки  
class TypeOfDeliveryUpdateView(UpdateView):  
 model = TypeOfDelivery  
 form\_class = TypeOfDeliveryForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания нового типа доставки  
class TypeOfDeliveryCreateView(CreateView):  
 model = TypeOfDelivery  
 form\_class = TypeOfDeliveryForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления типа доставки  
class TypeOfDeliveryDeleteView(DeleteView):  
 model = TypeOfDelivery  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("TypeOfDeliveryDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No TypeOfDelivery matches the given query.")  
  
  
# TYPEEEEEEE OFFFFFFFFF PAAAAAAAYYYYYYYS  
# View для отображения списка типов оплаты  
class TypeOfPayListView(ListView):  
 model = TypeOfPay  
 template\_name = 'DlBox/admin\_TypeOfPayCRUD.html'  
 context\_object\_name = 'list\_typesofpay'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 types = self.get\_queryset()  
 paginator = Paginator(types, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 types = paginator.get\_page(page)  
 context['types'] = types  
 return context  
  
  
# View для отображения детальной информации о типе оплаты  
class TypeOfPayDetailView(DetailView):  
 model = TypeOfPay  
 template\_name = 'DlBox/admin\_TypeOfPayDetailView.html'  
  
  
# View для обновления информации о типе оплаты  
class TypeOfPayUpdateView(UpdateView):  
 model = TypeOfPay  
 form\_class = TypeOfPayForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания нового типа оплаты  
class TypeOfPayCreateView(CreateView):  
 model = TypeOfPay  
 form\_class = TypeOfPayForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления типа оплаты  
class TypeOfPayDeleteView(DeleteView):  
 model = TypeOfPay  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("TypeOfPayDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No TypeOfPay matches the given query.")  
  
  
# BBBBBOOOOOOXXXXXXX  
# View для отображения списка коробок  
class BoxListView(ListView):  
 model = Box  
 template\_name = 'DlBox/admin\_BoxCRUD.html'  
 context\_object\_name = 'list\_box'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 boxes = self.get\_queryset()  
 paginator = Paginator(boxes, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 boxes = paginator.get\_page(page)  
 context['boxes'] = boxes  
 return context  
  
  
# View для отображения детальной информации о коробке  
class BoxDetailView(DetailView):  
 model = Box  
 template\_name = 'DlBox/admin\_BoxDetailView.html'  
  
  
# View для обновления информации о коробке  
class BoxUpdateView(UpdateView):  
 model = Box  
 form\_class = BoxForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания новой коробки  
class BoxCreateView(CreateView):  
 model = Box  
 form\_class = BoxForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления коробки  
class BoxDeleteView(DeleteView):  
 model = Box  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("BoxDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No Box matches the given query.")  
  
  
# OOOORRRDEDEEEEERRRRRSSSS  
# View для отображения списка заказов  
class OrderListView(ListView):  
 model = Order  
 template\_name = 'DlBox/admin\_OrderCRUD.html'  
 context\_object\_name = 'list\_order'  
 paginate\_by = 50  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super().get\_context\_data(\*\*kwargs)  
 orders = self.get\_queryset()  
 paginator = Paginator(orders, self.paginate\_by)  
  
 page = self.request.GET.get('page')  
 orders = paginator.get\_page(page)  
 context['orders'] = orders  
 return context  
  
  
# View для отображения детальной информации о заказе  
class OrderDetailView(DetailView):  
 model = Order  
 template\_name = 'DlBox/admin\_OrderDetailView.html'  
  
  
# View для обновления информации о заказе  
class OrderUpdateView(UpdateView):  
 model = Order  
 form\_class = OrderAdminForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для создания нового заказа  
class OrderCreateView(CreateView):  
 model = Order  
 form\_class = OrderAdminForm  
 template\_name = 'DlBox/create.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
  
# View для удаления заказа  
class OrderDeleteView(DeleteView):  
 model = Order  
 template\_name = 'DlBox/delete.html'  
 success\_url = reverse\_lazy('adminIndexPade')  
  
 def get\_object(self, queryset=None):  
 # Получаем значение pk из URL-адреса  
 pk = self.kwargs.get('pk')  
 if pk is None:  
 raise AttributeError("OrderDeleteView must be called with an object pk.")  
 try:  
 # Пытаемся получить объект по pk  
 return self.model.objects.get(pk=pk)  
 except (self.model.DoesNotExist, ValueError):  
 # Если объект не найден или pk не может быть преобразован в число, вызываем исключение 404  
 raise Http404("No Order matches the given query.")  
  
  
# Инициализация страницы подтверждения удаления  
def delete(request):  
 return render(request, 'DlBox/delete.html')  
  
  
# Инициализация страницы информации  
def info(request):  
 return render(request, 'DlBox/info.html')

* + - 1. Admin.py

from django.contrib import admin

from .models import \*

# Register your models here.

admin.site.register(PackageBox)

admin.site.register(TypeOfDelivery)

admin.site.register(TypeOfPay)

admin.site.register(Box)

admin.site.register(Order)

admin.site.register(Role)

admin.site.register(MyUser)

admin.site.register(UserOrders)

admin.site.register(StatusOrder)

admin.site.register(Tracking)

admin.site.register(SizePackage)

* + - 1. Forms.py

from django import forms

from django.contrib.auth.hashers import make\_password

from django.db.migrations import serializer

from .models import \*

class UserRegisterForm(forms.ModelForm):

password = forms.CharField(widget=forms.PasswordInput)

confirm\_password = forms.CharField(widget=forms.PasswordInput)

class Meta:

model = MyUser

fields = ['login', 'password', 'confirm\_password', 'user\_surname', 'user\_name', 'user\_patronymic', 'telephoneNumber']

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

self.fields['confirm\_password'].label = 'Подтверждение пароля'

self.fields['password'].label = 'Пароль'

def clean(self):

cleaned\_data = super().clean()

password = cleaned\_data.get("password")

confirm\_password = cleaned\_data.get("confirm\_password")

if password != confirm\_password:

raise forms.ValidationError("Пароли не совпадают.")

class MyUserForm(forms.ModelForm):

class Meta:

model = MyUser

fields = ['login', 'password', 'user\_surname', 'user\_name', 'user\_patronymic', 'telephoneNumber', 'role']

class LoginForm(forms.Form):

login = forms.CharField(label='Введите логин', max\_length=100)

password = forms.CharField(label='Введите пароль', max\_length=255, widget=forms.PasswordInput)

class TrackingForm(forms.Form):

# class Meta:

# model = Tracking

uniqueOrderNum = forms.CharField(label='Трек-номер', max\_length=200)

class OrderForm(forms.ModelForm):

class Meta:

model = Order

fields = ['delivery', 'doorTodoor', 'delicate', 'pay', 'sender\_address', 'recipient\_address', 'postmat']

widgets = {

'sender\_address': forms.HiddenInput(),

'recipient\_address': forms.HiddenInput(),

}

class UserBoxForm(forms.ModelForm):

class Meta:

model = Box

fields = ['valueOfContent', 'package', 'size\_package']

class RoleForm(forms.ModelForm):

class Meta:

model = Role

fields = ['name', 'permission']

class SizePackForm(forms.ModelForm):

class Meta:

model = StatusOrder

fields = ['name', 'description']

class StatusOrderForm(forms.ModelForm):

class Meta:

model = StatusOrder

fields = ['name', 'description']

class TypeOfDeliveryForm(forms.ModelForm):

class Meta:

model = TypeOfDelivery

fields = ['name', 'delivery\_cost', 'description']

class TypeOfPayForm(forms.ModelForm):

class Meta:

model = TypeOfPay

fields = ['name', 'description']

class BoxForm(forms.ModelForm):

class Meta:

model = Box

fields = ['user\_sender', 'valueOfContent', 'package', 'size\_package', 'price']

class OrderAdminForm(forms.ModelForm):

class Meta:

model = Order

fields = ['uniqueOrderNum', 'box', 'delivery', 'doorTodoor', 'delicate', 'cost', 'delicate', 'pay', 'sender\_address', 'recipient\_address', 'postmat', 'orders']

* + - 1. Routing.py

from django.urls import path, re\_path, include

import DlBox.views as v

from rest\_framework import routers

router = routers.DefaultRouter()

router.register(r'package', v.PackageBoxViewSet)

router.register(r'role', v.RoleViewSet)

router.register(r'typeofdelivery', v.TypeOfDeliveryViewSet)

router.register(r'typeofpay', v.TypeOfPayViewSet)

router.register(r'sizepackage', v.SizePackageViewSet)

router.register(r'statusorder', v.StatusOrderViewSet)

router.register(r'tracking', v.TrackingViewSet)

router.register(r'users', v.MyUserViewSet)

router.register(r'box', v.BoxViewSet)

router.register(r'order', v.OrderViewSet)

router.register(r'userorders', v.UserOrdersViewSet)

# router.register(r'autnmobile', v.MyUserMobileViewSet)

urlpatterns = [

path('api/', include(router.urls)),

# path('api-auth/', include('rest\_framework.urls', namespace='rest\_framework')),

path('', v.index, name='indexPage'),

path('account/', v.useraccount, name='useraccountPage'),

# path('update/<str:pk>/', v.UserUpdateSelfView.as\_view(), name='UserUpdateSelfView'),

path('info/', v.info, name='infoPage'),

path('adminindex/', v.admin\_index, name='adminIndexPade'),

path('adminusers/', v.MyUserListView.as\_view(), name='adminUsersListView'),

path('adminuserdetail/<str:pk>/', v.MyUserDetailView.as\_view(), name='adminUserDetailListView'),

path('adminuseradd/', v.MyUserCreateView.as\_view(), name='adminUserCreateView'),

path('adminuserupdate/<str:pk>/', v.MyUserUpdateView.as\_view(), name='adminUserUpdateView'),

path('adminuserdelete/<int:pk>/', v.MyUserDeleteView.as\_view(), name='adminUserDeleteView'),

path('adminrole/', v.RoleListView.as\_view(), name='adminRoleListView'),

path('adminroledetail/<str:pk>/', v.RoleDetailView.as\_view(), name='adminRoleDetailView'),

path('adminroleupdate/<str:pk>/', v.RoleUpdateView.as\_view(), name='adminRoleUpdateView'),

path('adminroleadd/', v.RoleCreateView.as\_view(), name='adminRoleCreateView'),

path('adminroledelete/<int:pk>/', v.RoleDeleteView.as\_view(), name='adminRoleDeleteView'),

path('adminsizepackadge/', v.SizePackRoleListView.as\_view(), name='adminSizePackListView'),

path('adminsizepackadgedetail/<str:pk>/', v.SizePackDetailView.as\_view(), name='adminSizePackDetailView'),

path('adminsizepackadgeupdate/<str:pk>/', v.SizePackUpdateView.as\_view(), name='adminSizePackUpdateView'),

path('adminsizepackadgeadd/', v.SizePackCreateView.as\_view(), name='adminSizePackCreateView'),

path('adminsizepackadgedelete/<int:pk>/', v.SizePackDeleteView.as\_view(), name='adminSizePackDeleteView'),

path('adminstatusorder/', v.StatusOrderListView.as\_view(), name='adminStatusOrderListView'),

path('adminstatusorderdetail/<str:pk>/', v.StatusOrderDetailView.as\_view(), name='adminStatusOrderDetailView'),

path('adminstatusorderupdate/<str:pk>/', v.StatusOrderUpdateView.as\_view(), name='adminStatusOrderUpdateView'),

path('adminstatusorderadd/', v.StatusOrderCreateView.as\_view(), name='adminStatusOrderCreateView'),

path('adminstatusorderdelete/<int:pk>/', v.StatusOrderDeleteView.as\_view(), name='adminStatusOrderDeleteView'),

path('admintypeofdelvery/', v.TypeOfDeliveryListView.as\_view(), name='adminTypeofDeliveryListView'),

path('admintypeofdelverydetail/<str:pk>/', v.TypeOfDeliveryDetailView.as\_view(), name='adminTypeofDeliveryDetailView'),

path('admintypeofdelveryupdate/<str:pk>/', v.TypeOfDeliveryUpdateView.as\_view(), name='adminTypeofDeliveryUpdateView'),

path('admintypeofdelveryadd/', v.TypeOfDeliveryCreateView.as\_view(), name='adminTypeofDeliveryCreateView'),

path('admintypeofdelverydelete/<int:pk>/', v.TypeOfDeliveryDeleteView.as\_view(), name='adminTypeofDeliveryDeleteView'),

path('admintypeofpay/', v.TypeOfPayListView.as\_view(), name='adminTypeofPayListView'),

path('admintypeofpaydetail/<str:pk>/', v.TypeOfPayDetailView.as\_view(), name='adminTypeofPayDetailView'),

path('admintypeofpayupdate/<str:pk>/', v.TypeOfPayUpdateView.as\_view(), name='adminTypeofPayUpdateView'),

path('admintypeofpayadd/', v.TypeOfPayCreateView.as\_view(), name='adminTypeofPayCreateView'),

path('admintypeofpaydelete/<int:pk>/', v.TypeOfPayDeleteView.as\_view(), name='adminTypeofPayDeleteView'),

path('adminbox/', v.BoxListView.as\_view(), name='adminBoxListView'),

path('adminboxdetail/<str:pk>/', v.BoxDetailView.as\_view(), name='adminBoxDetailView'),

path('adminboxupdate/<str:pk>/', v.BoxUpdateView.as\_view(), name='adminBoxUpdateView'),

path('adminboxadd/', v.BoxCreateView.as\_view(), name='adminBoxCreateView'),

path('adminboxdelete/<int:pk>/', v.BoxDeleteView.as\_view(), name='adminBoxDeleteView'),

path('adminorder/', v.OrderListView.as\_view(), name='adminOrderListView'),

path('adminorderdetail/<str:pk>/', v.OrderDetailView.as\_view(), name='adminOrderDetailView'),

path('adminorderupdate/<str:pk>/', v.OrderUpdateView.as\_view(), name='adminOrderUpdateView'),

path('adminorderadd/', v.OrderCreateView.as\_view(), name='adminOrderCreateView'),

path('adminorderdelete/<int:pk>/', v.OrderDeleteView.as\_view(), name='adminOrderDeleteView'),

re\_path('delete/id=\d+', v.delete, name='deletePage'),

path('auth/', v.auth, name='authPage'),

path('registration/', v.register, name='regPage'),

path('logout/', v.logout\_view, name='logoutPage'),

path('tracking/', v.tracking, name='trackingPage'),

path('statustracking/', v.status\_track, name='statusTrackPage'),

path('orderreg/<int:pk>/', v.order\_reg, name='orderRegPage'),

path('boxreg/', v.box\_reg, name='boxRegPage'),

path('orders/<int:pk>/', v.order\_detail, name='orderDetailPage'),

path('myorders/', v.user\_orders, name='userOrdersPage'),

path('add/', v.create, name='createPage'),

path('error/', v.error, name='errorPage'),

path('checkpassword/', v.check\_password\_view, name='check\_password'),

]

* + - 1. Models.py

import re

from django.core.exceptions import ValidationError

from django.core.validators import MinValueValidator

from django.db import models

import uuid

from django.db.models.signals import post\_save

class PackageBox(models.Model):

name = models.CharField(max\_length=255, verbose\_name='Название')

description = models.CharField(blank=True, verbose\_name='Описание')

def \_\_str\_\_(self):

return self.name

class TypeOfDelivery(models.Model):

name = models.CharField(max\_length=255, verbose\_name='Название')

delivery\_cost = models.FloatField(default=0.0, verbose\_name='Стоимость доставки')

description = models.CharField(blank=True, verbose\_name='Описание')

def \_\_str\_\_(self):

return self.name

class TypeOfPay(models.Model):

name = models.CharField(max\_length=255, verbose\_name='Название')

description = models.CharField(blank=True, verbose\_name='Описание')

def \_\_str\_\_(self):

return self.name

class SizePackage(models.Model):

name = models.CharField(max\_length=255, verbose\_name='Размер')

description = models.CharField(blank=True, verbose\_name='Описание')

def \_\_str\_\_(self):

return f"{self.name} - {self.description}"

# пользователи

class Role(models.Model):

name = models.CharField(max\_length=255, verbose\_name='Название')

permission = models.CharField(max\_length=255, verbose\_name='Права доступа')

def \_\_str\_\_(self):

return self.name

# Отслеживание

class StatusOrder(models.Model):

name = models.CharField(max\_length=255, verbose\_name='Название')

description = models.CharField(blank=True, verbose\_name='Описание')

def \_\_str\_\_(self):

return self.name

class Tracking(models.Model):

# нельзя удалить статус отслеживания если он используется

uniqueOrderNum = models.CharField(max\_length=255, unique=True, verbose\_name='Номер заказа')

status = models.ForeignKey(StatusOrder, on\_delete=models.PROTECT, verbose\_name='Статус')

def \_\_str\_\_(self):

return self.uniqueOrderNum

class MyUser(models.Model):

# нельзя удалить роль если она используется

login = models.CharField(max\_length=50, unique=True, verbose\_name='Логин')

password = models.CharField(max\_length=255, verbose\_name='Пароль')

user\_surname = models.CharField(max\_length=255, verbose\_name='Фамилия')

user\_name = models.CharField(max\_length=255, verbose\_name='Имя')

user\_patronymic = models.CharField(max\_length=255, blank=True, verbose\_name='Отчество')

telephoneNumber = models.CharField(max\_length=20, verbose\_name='Номер телефона')

role = models.ForeignKey(Role, on\_delete=models.PROTECT, verbose\_name='Роль')

def \_\_str\_\_(self):

return self.login

def clean(self):

# Валидация логина

if len(self.login) < 3 or not self.login.isalpha():

raise ValidationError(

{'login': 'Логин должен содержать не менее 3 символов и состоять только из латинских букв.'})

# Валидация пароля

password\_validation = r'^(?=.\*[A-Z])(?=.\*\d)[A-Za-z\d]{6,}$'

if not re.match(password\_validation, self.password):

raise ValidationError({

'password': 'Пароль должен содержать не менее 6 символов, состоять только из латинских букв и содержать хотя бы одну заглавную букву и цифру.'})

# Валидация номера телефона

phone\_validation = r'^8\d{3}\d{3}\d{2}\d{2}$'

if not re.match(phone\_validation, self.telephoneNumber):

raise ValidationError(

{'telephoneNumber': 'Номер телефона должен соответствовать шаблону '

'8 (...) ...-..-..'})

super().clean()

class Box(models.Model):

user\_sender = models.ForeignKey(MyUser, on\_delete=models.PROTECT, verbose\_name='Пользователь')

valueOfContent = models.FloatField(max\_length=10, validators=[MinValueValidator(10)], verbose\_name='Ценность содержимого посылки')

package = models.ForeignKey(PackageBox, null=True, on\_delete=models.SET\_NULL, verbose\_name='Упаковка')

size\_package = models.ForeignKey(SizePackage, null=True, on\_delete=models.SET\_NULL,

verbose\_name='Размер упаковки')

price = models.FloatField(default=0.0, verbose\_name='Базовая стоимость')

def \_\_str\_\_(self):

return f"{self.user\_sender}"

class Order(models.Model):

uniqueOrderNum = models.UUIDField(default=uuid.uuid4, unique=True, verbose\_name='Номер заказа')

box = models.ForeignKey(Box, on\_delete=models.PROTECT, verbose\_name='Посылка')

delivery = models.ForeignKey(TypeOfDelivery, null=True, on\_delete=models.SET\_NULL,

verbose\_name='Тип доставки')

doorTodoor = models.BooleanField(default=False, verbose\_name='От двери до двери')

delicate = models.BooleanField(default=False, verbose\_name='Хрупкая посылка')

cost = models.FloatField(verbose\_name='Цена')

pay = models.ForeignKey(TypeOfPay, null=True, on\_delete=models.SET\_NULL, verbose\_name='Тип оплаты')

timeAdd = models.DateTimeField(auto\_now\_add=True, verbose\_name='Дата заказа')

sender\_address = models.CharField(max\_length=255, verbose\_name='Адрес отправителя')

recipient\_address = models.CharField(max\_length=255, verbose\_name='Адрес получателя')

postmat = models.BooleanField(default=False, verbose\_name='Доставка в постамат')

orders = models.ManyToManyField(MyUser, through='UserOrders')

def \_\_str\_\_(self):

return self.uniqueOrderNum

def clean(self):

# Check that only one delivery method is selected

if int(self.doorTodoor) + int(self.postmat) > 1:

raise ValidationError(

'Можно выбрать только один тип доставки.')

def save(self, \*args, \*\*kwargs):

# Рассчитываем начальную стоимость заказа

base\_cost = self.box.price

delivery\_cost = self.delivery.delivery\_cost

if self.doorTodoor:

base\_cost += 599

if self.delicate:

base\_cost += 399

if self.postmat:

base\_cost += 259

# 1% от valueOfContent к итоговой стоимости заказа

base\_cost += base\_cost \* float(self.box.valueOfContent) / 100

# Общая стоимость заказа

self.cost = base\_cost + delivery\_cost + 199

# Удаляем из kwargs аргумент 'through', чтобы избежать ошибки при сохранении

kwargs.pop('through', None)

super().save(\*args, \*\*kwargs)

class UserOrders(models.Model):

user = models.ForeignKey(MyUser, on\_delete=models.PROTECT)

order = models.ForeignKey(Order, on\_delete=models.PROTECT)

def \_\_int\_\_(self):

return self.pk

* + - 1. Serializers.py

from rest\_framework import serializers

from .models import \*

class PackageBoxSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = PackageBox

fields = ('name', 'description')

class TypeOfDeliverySerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = TypeOfDelivery

fields = ('name', 'delivery\_cost', 'description')

class TypeOfPaySerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = TypeOfPay

fields = ('name', 'description')

class SizePackageSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = SizePackage

fields = ('name', 'description')

class RoleSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = Role

fields = ('name', 'permission')

class StatusOrderSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = StatusOrder

fields = ('name', 'description')

class TrackingSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = Tracking

fields = ('uniqueOrderNum', 'status')

class MyUserSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = MyUser

fields = ('login', 'password', 'user\_surname', 'user\_name', 'user\_patronymic', 'telephoneNumber', 'role')

class MyUserMobileSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = MyUser

fields = ('login', 'password')

class BoxSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = Box

fields = ('user\_sender', 'valueOfContent', 'package', 'size\_package', 'price')

class OrderSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = Order

fields = ('uniqueOrderNum', 'box', 'delivery', 'doorTodoor', 'delicate', 'cost', 'pay', 'timeAdd', 'sender\_address', 'recipient\_address', 'postmat')

class UserOrdersSerializer(serializers.HyperlinkedModelSerializer):

class Meta:

model = UserOrders

fields = ('user', 'order')

* + - 1. Admin\_BoxCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for i in list\_box %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.user\_sender }}</h5>

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.price }}</h5>

<a href="{% url 'adminBoxDetailView' i.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminBoxCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_BoxDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ box.pk }}</li>

<li>{{ box.user\_sender }}</li>

<li>{{ box.price }}</li>

<li>{{ box.package }}</li>

<li>{{ box.valueOfContent }}</li>

<li>{{ box.size\_package }}</li>

</ul>

<a href="{% url 'adminBoxUpdateView' box.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить заказ</a>

<a href="{% url 'adminBoxDeleteView' box.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить заказ</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_OrderDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ order.pk }}</li>

<li>{{ order.uniqueOrderNum }}</li>

<li>{{ order.box }}</li>

<li>{{ order.cost }}</li>

<li>{{ order.timeAdd }}</li>

<li>{{ order.delicate }}</li>

<li>{{ order.delivery }}</li>

<li>{{ order.pay }}</li>

<li>{{ order.postmat }}</li>

<li>{{ order.sender\_address }}</li>

<li>{{ order.recipient\_address }}</li>

</ul>

<a href="{% url 'useraccountPage' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_OrderCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for i in list\_order %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.uniqueOrderNum }}</h5>

<a href="{% url 'adminOrderDetailView' i.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminOrderCreateView' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_RoleCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for role in list\_roles %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ role.name }}</h5>

<a href="{% url 'adminRoleDetailView' role.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminRoleCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_RoleDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ role.pk }}</li>

<li>{{ role.name }}</li>

<li>{{ role.permission }}</li>

</ul>

<a href="{% url 'adminRoleUpdateView' role.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить заказ</a>

<a href="{% url 'adminRoleDeleteView' role.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить заказ</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_SizePackCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for i in list\_sizepack %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.name }}</h5>

<a href="{% url 'adminSizePackDetailView' i.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminSizePackCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_SizePackDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ sizepackage.pk }}</li>

<li>{{ sizepackage.name }}</li>

<li>{{ sizepackage.description }}</li>

</ul>

<a href="{% url 'adminSizePackUpdateView' sizepackage.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить заказ</a>

<a href="{% url 'adminSizePackDeleteView' sizepackage.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить заказ</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_StatusOrderCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for i in list\_status %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.name }}</h5>

<a href="{% url 'adminStatusOrderDetailView' i.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminStatusOrderCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_StatusOrderDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ statusorder.pk }}</li>

<li>{{ statusorder.name }}</li>

<li>{{ statusorder.description }}</li>

</ul>

<a href="{% url 'adminStatusOrderUpdateView' statusorder.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить заказ</a>

<a href="{% url 'adminStatusOrderDeleteView' statusorder.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить заказ</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_TypeOfDeliveryCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for i in list\_typesofdeliv %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.name }}</h5>

<a href="{% url 'adminTypeofDeliveryDetailView' i.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminTypeofDeliveryCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_TypeOfDeliveryDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ typeofdelivery.pk }}</li>

<li>{{ typeofdelivery.name }}</li>

<li>{{ typeofdelivery.delivery\_cost }}</li>

<li>{{ typeofdelivery.description }}</li>

</ul>

<a href="{% url 'adminTypeofDeliveryUpdateView' typeofdelivery.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить заказ</a>

<a href="{% url 'adminTypeofDeliveryDeleteView' typeofdelivery.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить заказ</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_TypeOfPayCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for i in list\_typesofpay %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ i.name }}</h5>

<a href="{% url 'adminTypeofPayDetailView' i.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminTypeofPayCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_TypeOfPayDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ typeofpay.pk }}</li>

<li>{{ typeofpay.name }}</li>

<li>{{ typeofpay.description }}</li>

</ul>

<a href="{% url 'adminTypeofPayUpdateView' typeofpay.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить заказ</a>

<a href="{% url 'adminTypeofPayDeleteView' typeofpay.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить заказ</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_UserCRUD.html

{% extends 'admin\_base.html' %}

{% block content %}

<div style="display: flex; flex-wrap: wrap">

{% for us in list\_users %}

<div class="card" style="width: 18rem; margin: 10px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; background-color: rgba(217,217,217,0.49)">

<div class="card-body">

<h5 class="card-title" style="color: #333; font-size: 18px">{{ us.login }}</h5>

<a href="{% url 'adminUserDetailListView' us.pk %}" class="button">Подробнее</a>

</div>

</div>

{% endfor %}

<a href="{% url 'adminUserCreateView' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Добавить</a>

<a href="{% url 'adminIndexPade' %}" class="buttonadmin" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

{% endblock %}

* + - 1. Admin\_UserDetailView.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1 class="header">Детали</h1>

<div class="container">

<div class="product">

<ul>

<li>{{ myuser.pk }}</li>

<li>{{ myuser.login }}</li>

<li>{{ myuser.password }}</li>

<li>{{ myuser.user\_surname }}</li>

<li>{{ myuser.user\_name }}</li>

<li>{{ myuser.user\_patronymic }}</li>

<li>{{ myuser.telephoneNumber }}</li>

<li>{{ myuser.role }}</li>

</ul>

<a href="{% url 'adminUserUpdateView' myuser.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить</a>

<a href="{% url 'adminUserDeleteView' myuser.pk %}" class="button" style="background-color: #ff4040; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Удалить</a>

<a href="{% url 'adminIndexPade' %}" class="button" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 30px 5px 10px 10px;">Назад</a>

</div>

</div>

</body>

</html>

* + - 1. Admin\_index.html

{% extends 'admin\_base.html' %}

{% block content %}

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: auto;

overflow: hidden;

}

.btn {

display: inline-block;

color: #ffffff;

background: #333;

padding: 10px 20px;

margin: 10px;

border-radius: 5px;

text-decoration: none;

transition: background 0.3s;

}

.btn:hover {

background: #666;

}

h1 {

color: #333;

text-align: center;

padding: 20px 0;

}

</style>

<div style="margin: 40px">

<div class="container">

<h1>ПАНЕЛЬ АДМИНИСТРАТОРА</h1>

<p>Номер телефона: {{ request.session.telephoneNumber }}</p>

<a class="btn" style="font-size: small" href="{% url 'logoutPage' %}">Выйти из аккаунта</a>

<br><br>

<h2>Управление данными по таблицам:</h2>

<a class="btn" href="{% url 'adminUsersListView' %}">users</a>

<a class="btn" href="{% url 'adminRoleListView' %}">roles</a>

<a class="btn" href="{% url 'adminSizePackListView' %}">size package</a>

<a class="btn" href="{% url 'adminStatusOrderListView' %}">status order</a>

<a class="btn" href="{% url 'adminTypeofDeliveryListView' %}">type of delivery</a>

<a class="btn" href="{% url 'adminTypeofPayListView' %}">type of pay</a>

<a class="btn" href="{% url 'adminBoxListView' %}">box</a>

<a class="btn" href="{% url 'adminOrderListView' %}">order</a>

доделать отслеживание и упаковку (и много ко многим)

</div>

</div>

{% endblock %}

* + - 1. Index.html

{% extends 'base.html' %}

{% block content %}

<h1 style="font-size: 50px; text-align: center; margin-top: 50px; color: #333">DlBox - сервис лучших доставок</h1><br>

{# super user alinaryzhkova 123<br><br>#}

<div class="container px-4 py-5" id="custom-cards" style="margin-bottom: 70px">

<h2 class="pb-2 border-bottom">Оформить доставку легко!</h2>

<div class="row row-cols-1 row-cols-lg-3 align-items-stretch g-4 py-5">

<div class="col">

<a href="{% url 'boxRegPage' %}" style="text-decoration: none">

<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg" style="cursor: pointer; background-image: url('https://img.freepik.com/free-photo/young-adult-moving-into-new-home\_23-2149218974.jpg?w=740&t=st=1710747038~exp=1710747638~hmac=bc521524e2c1093a80174662ac4e706a33cbe07a7fa7119a58fb66fca05b82fc'); background-position: center center; background-repeat: repeat; background-size: cover; background-color: #ffffff;">

<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1"><br><br>

<h3 class="pt-5 mt-5 mb-4 display-7 lh-1 fw-bold">Оформить доставку</h3>

<ul class="d-flex list-unstyled mt-auto">

<li class="me-auto">

<img src="https://img.freepik.com/free-photo/high-angle-delivery-truck-boxes\_23-2148790067.jpg?t=st=1709050493~exp=1709054093~hmac=c415b2f3e573c73c4b08133e4d135a78a06fc17c9e4d8b0d597342b68429cbbc&w=1480" alt="сушки" width="32" height="32" class="rounded-circle border border-white">

</li>

<li class="d-flex align-items-center">

<svg class="bi me-2" width="1em" height="1em"><use xlink:href="#calendar3"></use></svg>

<small>DlBox</small>

</li>

</ul>

</div>

</div>

</a>

</div>

<div class="col">

<a href="{% url 'trackingPage' %}" style="text-decoration: none">

<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg" style="cursor: pointer; background-image: url('https://img.freepik.com/free-photo/empty-storage-space-filled-with-stacks-of-merchandise-and-goods\_482257-67569.jpg?w=2000&t=st=1710747108~exp=1710747708~hmac=79f370869f5f90464699348ad6d139db27f8e0a39d14fbbe1dc617c054a1596a'); background-position: center center; background-repeat: repeat; background-size: cover; background-color: #ffffff;">

<div class="d-flex flex-column h-100 p-5 pb-3 text-white text-shadow-1"><br><br>

<h3 class="pt-5 mt-5 mb-4 display-7 lh-1 fw-bold">Отследить посылку</h3>

<ul class="d-flex list-unstyled mt-auto">

<li class="me-auto">

<img src="https://img.freepik.com/free-photo/high-angle-delivery-truck-boxes\_23-2148790067.jpg?t=st=1709050493~exp=1709054093~hmac=c415b2f3e573c73c4b08133e4d135a78a06fc17c9e4d8b0d597342b68429cbbc&w=1480" alt="сушки" width="32" height="32" class="rounded-circle border border-white">

</li>

<li class="d-flex align-items-center">

<svg class="bi me-2" width="1em" height="1em"><use xlink:href="#calendar3"></use></svg>

<small>DlBox</small>

</li>

</ul>

</div>

</div>

</a>

</div>

<div class="col">

<a href="{% url 'infoPage' %}" style="text-decoration: none">

<div class="card card-cover h-100 overflow-hidden text-bg-dark rounded-4 shadow-lg" style="cursor: pointer; background-image: url('https://img.freepik.com/free-photo/deliver-man-holding-package\_23-2148890014.jpg?t=st=1710746628~exp=1710750228~hmac=06633e09237126b9634d82aff849b5144b64c2b4f1f4d62a1447c0ac2afd2337&w=740'); background-position: center center; background-repeat: repeat; background-size: cover; background-color: #ffffff;">

<div class="d-flex flex-column h-100 p-5 pb-3 text-shadow-1"><br><br>

<h3 class="pt-5 mt-5 mb-4 display-7 lh-1 fw-bold">Информация о нас</h3>

<ul class="d-flex list-unstyled mt-auto">

<li class="me-auto">

<img src="https://img.freepik.com/free-photo/high-angle-delivery-truck-boxes\_23-2148790067.jpg?t=st=1709050493~exp=1709054093~hmac=c415b2f3e573c73c4b08133e4d135a78a06fc17c9e4d8b0d597342b68429cbbc&w=1480" alt="сушки" width="32" height="32" class="rounded-circle border border-white">

</li>

<li class="d-flex align-items-center">

<svg class="bi me-2" width="1em" height="1em"><use xlink:href="#calendar3"></use></svg>

<small>DlBox</small>

</li>

</ul>

</div>

</div>

</a>

</div>

</div>

</div>

{% endblock %}

* + - 1. Auth.html

{% extends 'base.html' %}

{% block content %}

<form method="post" enctype="multipart/form-data" style="width: 550px; margin: 20px auto; padding: 10px; border: 0.6px solid #ccc; border-radius: 5px">

{% csrf\_token %}

<label style="display: block; margin-bottom: 10px">

{{ form.as\_p }}

</label>

<button type="submit" style="padding: 10px 20px; background-color: rgba(36,87,129,0.92); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgba(28,49,72,0.86)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Войти

</button>

</form>

<div style="margin-bottom: 10%">

<a href="{% url 'regPage' %}" style="text-decoration: none">

<button type="submit" style="padding: 10px 20px; background-color: rgba(58,85,110,0.92); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgb(41,60,79)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Нет аккаунта? Тык сюда

</button>

</a>

</div>

{% endblock %}

* + - 1. Box\_reg.html

{% extends 'base.html' %}

{% block content %}

<form method="post" enctype="multipart/form-data" style="width: 550px; padding: 10px; border: 0.6px solid #ccc; border-radius: 5px; margin: 20px auto 14%;">

{% csrf\_token %}

<label style="display: block; margin-bottom: 10px">

{{ form.as\_p }}

</label>

<button type="submit" style="padding: 10px 20px; background-color: rgba(58,85,110,0.92); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgb(41,60,79)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Готово</button>

</form>

{% endblock %}

* + - 1. Create.html

{% extends 'base.html' %}

{% block content %}

<form method="post" enctype="multipart/form-data" style="width: 550px; margin: 20px auto; padding: 10px; border: 0.6px solid #ccc; border-radius: 5px; background-color: #f8f9fa; color: #343a40;">

{% csrf\_token %}

<label style="display: block; margin-bottom: 10px">

{{ form.as\_table }}

</label>

<button type="submit" style="padding: 10px 20px; background-color: rgba(58,85,110,0.92); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgb(41,60,79)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Готово</button>

</form>

{% endblock %}

* + - 1. Delete.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1>ВЫ РЕАЛЬНО ХОТИТЕ УДАЛИТЬ {{ myuser.login }}</h1>

<form method="post">

{% csrf\_token %}

<button class="btn btn-danger" style="margin-left: 10px">Удалить</button>

</form>

<a class="button" href="{% url 'adminUsersListView' %}" style="background-color: #007bff; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Нет, я хочу домой</a>

</body>

</html>

* + - 1. Error.html

{% extends 'base.html' %}

{% block content %}

<h1>Error</h1>

<p style="margin-left: 50px">{{ error }}</p>

{% endblock %}

* + - 1. Info.html

{% extends 'base.html' %}

{% block content %}

<div class="mamainfo">

<div class="info\_container">

<div class="info\_content">

<h2>Новый ускоренный сервис доставки: получите посылку в день заказа!</h2>

<p>Мы рады представить наш новый ускоренный сервис доставки, который позволяет вам получить посылку в день заказа! Теперь ваши подарки и документы будут доставлены быстрее, чем когда-либо.</p>

</div>

<div class="info\_content">

<h2>Скоро новый сервис доставки: самовывоз посылок!</h2>

<p>Мы рады представить новый сервис самовывоза посылок. Теперь вы можете забрать свои посылки в удобное для вас место и время. Удобство и гибкость – вот наши приоритеты!</p>

</div>

<div class="info\_content">

<h2>Присоединяйтесь к нашей команде!</h2>

<p>Наша компания заботится о своих сотрудниках и ждет новых.</p>

</div>

</div>

<div class="info\_btn\_container">

<button class="info\_btn" style="background-color: rgba(75,75,75,0.61)" id="prevBtn"><</button>

<button class="info\_btn" style="background-color: rgba(75,75,75,0.61)" id="nextBtn">></button>

</div>

</div>

<script>

document.addEventListener("DOMContentLoaded", function() {

const content = document.querySelectorAll(".info\_content");

const prevBtn = document.getElementById("prevBtn");

const nextBtn = document.getElementById("nextBtn");

let currentIndex = 0;

// Показать текущий блок и скрыть остальные

function showCurrentContent() {

content.forEach((item, index) => {

if (index === currentIndex) {

item.classList.add("active");

} else {

item.classList.remove("active");

}

});

}

// Обработчик кнопки "Предыдущий"

prevBtn.addEventListener("click", function() {

currentIndex = (currentIndex - 1 + content.length) % content.length;

showCurrentContent();

});

// Обработчик кнопки "Следующий"

nextBtn.addEventListener("click", function() {

currentIndex = (currentIndex + 1) % content.length;

showCurrentContent();

});

// Показать первый блок при загрузке страницы

showCurrentContent();

});

</script>

<div class="faq">

<div class="question" onclick="toggleAnswer(this)">

<span>Как можно отслеживать статус доставки посылки?</span>

<div class="answer">Вы можете отслеживать статус доставки посылки, используя наш наш веб-сайт. Просто введите номер вашей посылки, и вы получите актуальную информацию о ее статусе доставки.</div>

</div>

<div class="question" onclick="toggleAnswer(this)">

<span>Можно ли забрать посылку самостоятельно?</span>

<div class="answer">Да, у нас есть новый сервис самовывоза посылок из специальных постаматов. Вы можете забрать свою посылку в удобное для вас место и время.</div>

</div>

<div class="question" onclick="toggleAnswer(this)">

<span>Какие документы необходимы для оформления доставки посылок?</span>

<div class="answer">Для оформления доставки посылок вам потребуется предоставить адрес отправления и адрес получения, а также контактные данные отправителя. Также вам нужно указать данные посылки и ее упаковки.</div>

</div>

<div class="question" onclick="toggleAnswer(this)">

<span>Есть ли какая-то бонусная программа?</span>

<div class="answer">К сожалению, пока что это в планах. Но скоро она обязательно появится!</div>

</div>

</div>

<script>

function toggleAnswer(question) {

var answer = question.querySelector('.answer');

answer.style.display = answer.style.display === 'block' ? 'none' : 'block';

question.classList.toggle('active');

}

</script>

{% endblock %}

* + - 1. Order\_detail.html

{% extends 'base.html' %}

{% block content %}

<div style="margin: 50px">

<h2>Детали заказа</h2>

<p><strong>Трек-номер:</strong> {{ order.uniqueOrderNum }}</p>

{# <p><strong>Посылка:</strong> {{ order.box }}</p>#}

<p><strong>Тип доставки:</strong> {{ order.delivery }}</p>

<p><strong>От двери до двери:</strong> {{ order.doorTodoor|yesno:"Да,Нет" }}</p>

<p><strong>Хрупкая посылка:</strong> {{ order.delicate|yesno:"Да,Нет" }}</p>

<p><strong>Доставка в постамат:</strong> {{ order.postmat|yesno:"Да,Нет" }}</p>

<p><strong>Цена:</strong> {{ order.cost }}руб.</p>

<p><strong>Тип оплаты:</strong> {{ order.pay }} <sub>\*кто получает подгон, тот и платит))</sub></p>

<p><strong>Дата заказа:</strong> {{ order.timeAdd }}</p>

<p><strong>Адрес отправителя:</strong> {{ order.sender\_address }}</p>

<p><strong>Адрес получателя:</strong> {{ order.recipient\_address }}</p>

</div>

{% endblock %}

* + - 1. Order\_reg.html

{% extends 'base.html' %}

{% block content %}

<form method="post" enctype="multipart/form-data" style="width: 550px; margin: 20px auto; padding: 10px; border: 0.6px solid #ccc; border-radius: 5px">

{% csrf\_token %}

<label style="display: block; margin-bottom: 10px">

{{ form.as\_p }}

</label>

<input id="addres\_sender" type="text" name="sender\_address" placeholder="Введите адрес отправителя">

<input id="address\_recipient" type="text" name="recipient\_address" placeholder="Введите адрес получателя">

<button type="submit" style="padding: 10px 20px; background-color: rgba(58,85,110,0.92); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgb(41,60,79)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Готово

</button>

</form>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<link href="http://cdn.jsdelivr.net/npm/suggestions-jquery@22.6.0/dist/css/suggestions.min.css" rel="stylesheet" />

<script src="http://cdn.jsdelivr.net/npm/suggestions-jquery@22.6.0/dist/js/jquery.suggestions.min.js"></script>

<script>

$("#addres\_sender").suggestions({

token: "5105eca38d394ea872e578e80d85d21599a91e40",

type: "ADDRESS",

/\* Вызывается, когда пользователь выбирает одну из подсказок \*/

onSelect: function(suggestion) {

console.log(suggestion);

}

});

</script>

<script>

$("#address\_recipient").suggestions({

token: "5105eca38d394ea872e578e80d85d21599a91e40",

type: "ADDRESS",

/\* Вызывается, когда пользователь выбирает одну из подсказок \*/

onSelect: function(suggestion) {

console.log(suggestion);

}

});

</script>

{% endblock %}

* + - 1. Status\_track\_order.html

{% extends 'base.html' %}

{% block content %}

<div class="container">

<h2>Информация о посылке</h2><br>

<h5>Трек номер: {{ package.uniqueOrderNum }}</h5>

<h5>Статус: {{ package.status }}</h5>

</div>

{% endblock %}

* + - 1. Tracking\_order.html

{% extends 'base.html' %}

{% block content %}

<style>

body {

margin: 0;

}

form {

padding: 80px;

border-radius: 5px;

}

button {

margin-top: 10px;

padding: 10px 20px;

border: none;

border-radius: 5px;

background-color: #595959;

color: #fff;

cursor: pointer;

}

button:hover {

background-color: #1e1e1e;

}

</style>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Отследить</button>

</form>

{% if error %}

<p>{{ error }}</p>

{% endif %}

{% endblock %}

* + - 1. User\_orders.html

{% extends 'base.html' %}

{% block content %}

<style>

.order-link:hover {

transition: color 0.2s ease; /\* добавляем плавный переход \*/

color: #245781; /\* изменяем цвет при наведении \*/

}

</style>

<h2>Мои заказы</h2>

<ul style="margin-bottom: 25%">

{% for order in orders %}

<li>

<a href="{% url 'adminOrderDetailView' order.pk %}" class="order-link">{{ order.uniqueOrderNum }} - {{ order.timeAdd }}</a>

</li>

{% empty %}

<li>У вас нет заказов.</li>

{% endfor %}

</ul>

{% endblock %}

{% block extra\_css %}

{% endblock %}

* + - 1. Useraccount.html

{% extends 'base.html' %}

{% block content %}

<div style="display: flex; flex-direction: column; justify-content: center; align-items: center;">

<h1 style="margin-bottom: 30px">Личный кабинет</h1>

{% if request.session.login %}

<div class="container" style="width: 600px; display: flex; flex-direction: column; justify-content: space-between; align-items: center;">

<div style="text-align: center;">

<h3 style="margin-bottom: 30px">{{ request.session.user\_surname }} {{ request.session.user\_name }} {{ request.session.user\_patronymic }}</h3>

<div class="info" style="font-size: 17px">

<label>Логин</label>

<p>{{ request.session.login }}</p>

<label>Номер телефона</label>

<p>{{ request.session.telephoneNumber }}</p>

</div>

<a class="btn" href="{% url 'userOrdersPage' %}">

<button type="submit" style="padding: 10px 20px; background-color: rgba(44,68,91,0.92); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgba(28,49,72,0.86)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Мои заказы

</button>

</a>

</div>

{# <a href="{% url 'UserUpdateSelfView' myuser.pk %}" class="button" style="background-color: #ff8c4f; border: none; color: white; padding: 8px 20px; text-align: center; text-decoration: none; display: inline-block; font-size: 15px; cursor: pointer; margin: 10px 5px 10px 10px;">Изменить данные</a>#}

</div>

<a class="btn" href="{% url 'logoutPage' %}">

<button type="submit" style=" padding: 7px 17px; background-color: rgba(24,34,42,0.94); color: #fff; border: none; border-radius: 3px; cursor: pointer; font-size: 14px; transition: all 0.3s ease; box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);" onmouseover="this.style.backgroundColor='rgb(31,45,58)'; this.style.boxShadow='0 6px 8px rgba(0, 0, 0, 0.2)';" onmouseout="this.style.backgroundColor='rgba(58,85,110,0.92)'; this.style.boxShadow='0 4px 6px rgba(0, 0, 0, 0.1)';">

Выйти

</button>

</a>

{% endif %}

</div>

{% endblock %}

* + - 1. \_adminHeader.html

<nav class="navbar navbar-expand-lg bg-body-tertiary" style="height: 60px">

<div class="container-fluid" style="margin-top: 10px; margin-bottom: 10px">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item" style="margin-top: 16px">

<a class="nav-link" aria-disabled="true" href="{% url 'adminIndexPade' %}"> Главная </a>

</li>

<li class="nav-item" style="margin-top: 16px">

<a class="nav-link" target="\_blank" href="http://127.0.0.1:8000/api/">API</a>

</li>

</ul>

</div>

</div>

</nav>

* + - 1. \_Header.html

<nav class="navbar navbar-expand-lg bg-body-tertiary" style="height: 60px; width: 100%">

<div class="container-fluid" style="margin-top: 10px; margin-bottom: 10px">

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li>

<a href="{% url 'indexPage' %}" style="text-decoration: none">

<svg width="40" height="60" style="margin-right: 40px" viewBox="0 0 577 661" fill="none" xmlns="http://www.w3.org/2000/svg">

<g opacity="0.5">

<path d="M281.138 659.127C281.138 659.897 280.304 660.378 279.637 659.992L0.499084 498.556C0.190063 498.377 -0.000244141 498.048 -0.000244141 497.691L-0.000213623 174.815C-0.000213623 174.045 0.833771 173.564 1.50043 173.949L280.638 335.386C280.947 335.564 281.138 335.894 281.138 336.251L281.138 659.127Z" fill="#428260"/>

<path d="M288.999 326.897C288.689 327.077 288.306 327.077 287.996 326.897L8.18109 164.567C7.51678 164.181 7.51685 163.222 8.18109 162.837L287.996 0.506409C288.306 0.326385 288.689 0.326385 288.999 0.506409L568.814 162.837C569.478 163.222 569.478 164.181 568.814 164.567L288.999 326.897Z" fill="#428260"/>

<path d="M577 497.691C577 498.048 576.81 498.377 576.501 498.556L297.363 659.992C296.696 660.378 295.862 659.897 295.862 659.127L295.862 336.251C295.862 335.894 296.052 335.564 296.361 335.386L575.499 173.949C576.166 173.564 577 174.045 577 174.815L577 497.691Z" fill="#383838"/>

<path d="M170.029 70.8396C169.719 70.6604 169.337 70.6608 169.028 70.8405L127.385 95.0222C126.721 95.4081 126.722 96.3691 127.388 96.7537L406.664 258.151C406.974 258.33 407.355 258.33 407.664 258.151L449.396 234.021C450.062 233.635 450.061 232.673 449.395 232.288L170.029 70.8396Z" fill="#DBB11B"/>

<path d="M414.391 267.476C414.391 267.119 414.582 266.788 414.892 266.61L456.633 242.603C457.3 242.219 458.132 242.701 458.132 243.47V566.264C458.132 566.622 457.941 566.952 457.632 567.13L415.891 591.24C415.224 591.625 414.391 591.144 414.391 590.374V267.476Z" fill="#DBB11B"/>

<g clip-path="url(#clip0\_405\_2442)">

<rect width="67.3554" height="67.3554" transform="matrix(0.865834 0.500331 -0.865834 0.500331 83.4062 129.437)" fill="white"/>

<path d="M39.6673 157.52L73.6865 137.862L78.5464 140.67L44.5272 160.328L39.6673 157.52ZM46.9572 161.732L80.9763 142.074L85.8362 144.882L51.817 164.541L46.9572 161.732ZM54.247 165.945L88.2662 146.287L90.6961 147.691L56.6769 167.349L54.247 165.945ZM61.5368 170.157L95.556 150.499L100.416 153.307L66.3967 172.966L61.5368 170.157ZM68.8266 174.37L102.846 154.712L110.136 158.924L76.1165 178.582L68.8266 174.37ZM78.5464 179.987L112.566 160.328L114.996 161.732L80.9763 181.391L78.5464 179.987ZM85.8362 184.199L119.855 164.541L127.145 168.753L93.1261 188.412L85.8362 184.199Z" fill="#383838"/>

</g>

</g>

<defs>

<clipPath id="clip0\_405\_2442">

<rect width="67.3554" height="67.3554" fill="white" transform="matrix(0.865834 0.500331 -0.865834 0.500331 83.4062 129.437)"/>

</clipPath>

</defs>

</svg>

</a>

</li>

<li class="nav-item" style="margin-top: 16px">

<a class="nav-link" href="{% url 'useraccountPage' %}">Аккаунт</a>

</li>

</ul>

</div>

</div>

</nav>

* + - 1. \_Footer.html

<div class="containerfoot" id="content">

<footer class="py-3 my-4">

<ul class="nav justify-content-center border-bottom pb-3 mb-3">

<li class="nav-item"><a href="{% url 'indexPage' %}" class="nav-link px-2 text-body-secondary">Главная</a></li>

<li class="nav-item"><a href="{% url 'trackingPage' %}" class="nav-link px-2 text-body-secondary">Отслеживание</a></li>

<li class="nav-item"><a href="{% url 'infoPage' %}" class="nav-link px-2 text-body-secondary">Акции</a></li>

<li class="nav-item"><a href="{% url 'infoPage' %}" class="nav-link px-2 text-body-secondary">FAQs</a></li>

</ul>

<p class="text-center text-body-secondary">© 2024 DlBox, Inc</p>

</footer>

</div>

* + - 1. Base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

{% block extra\_js %}

{% endblock %}

<script src="https://api-maps.yandex.ru/2.1/?apikey=79e73dca-3c9a-4620-81fb-da58c21167cb&lang=ru\_RU" type="text/javascript"></script>

<link rel="icon" href="https://c0.klipartz.com/pngpicture/174/493/gratis-png-camioneta-iconos-de-camioneta-carro.png">

{# <link rel="stylesheet" href="{% static 'DlBox/css/base.css' %}">#}

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<title>DlBox</title>

<style>

body, html {

margin: 0;

padding: 0;

height: 100%;

}

header {

background-color: #4B4B4B;

color: white;

padding: 10px 0;

}

.header-content {

display: flex;

justify-content: space-between;

align-items: center;

max-width: 1200px;

margin: 0 auto;

padding: 0 20px;

}

.navbar-brand {

font-size: 1.5em;

color: white;

text-decoration: none;

}

.navbar-toggler {

display: none;

}

.navbar-nav ul {

list-style-type: none;

padding: 0;

margin: 0;

display: flex;

}

.navbar-nav ul li {

margin: 0 10px;

}

.navbar-nav ul li a {

color: white;

text-decoration: none;

}

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

}

h1 {

text-align: center;

margin-top: 50px;

color: #333;

}

.container {

width: 600px;

margin: 40px auto;

background-color: #fff;

border-radius: 10px;

padding: 20px;

box-shadow: 0 2px 4px rgba(0,0,0,0.3);

display: flex;

flex-direction: column;

justify-content: space-between;

}

.avatar {

display: block;

margin: 0 auto;

width: 150px;

height: 150px;

border-radius: 50%;

object-fit: cover;

}

h2 {

text-align: center;

margin-top: 20px;

color: #333;

}

.info {

margin-top: 20px;

font-size: 18px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

}

p {

margin-bottom: 10px;

}

button {

display: block;

margin: 20px auto 0 auto;

padding: 10px 20px;

border-radius: 5px;

background-color: #333;

color: #fff;

font-size: 16px;

cursor: pointer;

}

button:hover {

background-color: #555;

}

ul li a {

text-decoration: none;

color: #000;

transition: all 0.3s ease-in-out;

font-size: 20px;

}

ul li a:hover {

color: #0051ff;

font-weight: bold;

}

input[type="text"],

input[type="email"],

input[type="password"] {

width: 100%;

padding: 10px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 3px;

}

button[type="submit"]:hover {

background-color: #0056b3;

}

.header {

text-align: center;

margin-top: 50px;

margin-bottom: 30px;

}

.container {

width: 80%;

margin: 0 auto;

display: flex;

flex-wrap: wrap;

justify-content: space-between;

align-items: center;

}

.product {

width: 30%;

background-color: #f2f2f2;

border-radius: 10px;

padding: 20px;

margin-bottom: 30px;

}

.pagination {

display: flex;

justify-content: center;

align-items: center;

margin-top: 20px;

}

.pagination a {

color: #000;

padding: 8px 16px;

text-decoration: none;

transition: background-color .3s;

border: 1px solid #ddd;

margin: 0 4px;

}

.pagination a.active {

background-color: #4CAF50;

color: white;

border: 1px solid #4CAF50;

}

.pagination a:hover:not(.active) {

background-color: #ddd;

}

.button {

display: inline-block;

padding: 5px 10px;

font-size: 14px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #007bff;

border: none;

border-radius: 5px;

}

.button:hover {background-color: #0069d9}

.button:active {

background-color: #0062cc;

box-shadow: 1px 2px #a6a6a6;

transform: translateY(1px);

}

{# INFO #}

.mamainfo {

margin: 0;

padding: 0;

height: 60%;

display: flex;

justify-content: center;

align-items: center;

}

.info\_container {

text-align: center;

}

.info\_content {

display: none; /\* Скрываем все блоки по умолчанию \*/

padding: 20px;

background-image: url("https://img.freepik.com/free-vector/customer-using-mobile-app-tracking-order-delivery\_74855-5229.jpg?t=st=1714656687~exp=1714660287~hmac=f5f3eab9c312cf0b05f91ce78ecec37f70f9be64e5ce1875aff95e8fce68fb94&w=1480");

background-position: top center;

background-repeat: repeat;

background-size: cover;

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

opacity: 0; /\* Изначально скрываем блоки \*/

transition: opacity 0.5s ease; /\* Добавляем плавный переход \*/

position: absolute; /\* Позиционируем блоки абсолютно \*/

top: 40%;

left: 50%;

transform: translate(-50%, -50%); /\* Центрируем блоки \*/

width: 850px;

height: 400px;

}

.info\_content.active {

display: block; /\* Показываем активный блок \*/

opacity: 1;

}

.info\_btn\_container {

position: absolute;

top: 50%;

left: 50%;

transform: translateX(-50%);

display: flex; /\* Выравниваем кнопки по горизонтали \*/

justify-content: center; /\* Выравниваем кнопки по центру \*/

}

.info\_btn {

margin: 0 10px;

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

}

h2 .mamainfo {

font-size: 25px;

color: #333;

font-weight: bold;

margin-bottom: 10px;

}

p {

font-size: 16px;

color: #666;

line-height: 1.6;

}

{# FFAAAAAQQQ#}

.faq {

width: 100%;

max-width: 600px;

margin: 0 auto;

}

.question {

position: relative;

padding: 10px 0;

border-bottom: 1px solid #ccc;

cursor: pointer;

font-size: x-large;

}

.question span {

font-weight: bold;

}

.answer {

display: none;

padding-top: 10px;

transition: display 0.5s ease-in-out;

font-size: large;

}

.question.active .answer {

display: block;

}

</style>

</head>

<body>

{% include 'inc/\_header.html' %}

{% block content %}

{% endblock %}

{% include 'inc/\_footer.html' %}

</body>

</html>

* + - 1. Admin\_base.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<link rel="icon" href="https://c0.klipartz.com/pngpicture/174/493/gratis-png-camioneta-iconos-de-camioneta-carro.png">

{# <link rel="stylesheet" href="{% static 'DlBox/css/base.css' %}">#}

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">

<title>DlBox</title>

<style>

header {

background-color: #4B4B4B;

color: white;

padding: 10px 0;

}

.header-content {

display: flex;

justify-content: space-between;

align-items: center;

max-width: 1200px;

margin: 0 auto;

padding: 0 20px;

}

.navbar-brand {

font-size: 1.5em;

color: white;

text-decoration: none;

}

.navbar-toggler {

display: none;

}

.navbar-nav ul {

list-style-type: none;

padding: 0;

margin: 0;

display: flex;

}

.navbar-nav ul li {

margin: 0 10px;

}

.navbar-nav ul li a {

color: white;

text-decoration: none;

}

body {

font-family: Arial, sans-serif;

background-color: #f5f5f5;

}

h1 {

text-align: center;

margin-top: 50px;

color: #333;

}

.container {

width: 600px;

margin: 40px auto;

background-color: #fff;

border-radius: 10px;

padding: 20px;

box-shadow: 0 2px 4px rgba(0,0,0,0.3);

display: flex;

flex-direction: column;

justify-content: space-between;

}

.avatar {

display: block;

margin: 0 auto;

width: 150px;

height: 150px;

border-radius: 50%;

object-fit: cover;

}

h2 {

text-align: center;

margin-top: 20px;

color: #333;

}

.info {

margin-top: 20px;

font-size: 18px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

}

p {

margin-bottom: 10px;

}

button {

display: block;

margin: 20px auto 0 auto;

padding: 10px 20px;

border-radius: 5px;

background-color: #333;

color: #fff;

font-size: 16px;

cursor: pointer;

}

button:hover {

background-color: #555;

}

ul li a {

text-decoration: none;

color: #000;

transition: all 0.3s ease-in-out;

font-size: 20px;

}

ul li a:hover {

color: #0051ff;

font-weight: bold;

}

input[type="text"],

input[type="email"],

input[type="password"] {

width: 100%;

padding: 10px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 3px;

}

button[type="submit"]:hover {

background-color: #0056b3;

}

.header {

text-align: center;

margin-top: 50px;

margin-bottom: 30px;

}

.container {

width: 80%;

margin: 0 auto;

display: flex;

flex-wrap: wrap;

justify-content: space-between;

align-items: center;

}

.product {

width: 30%;

background-color: #f2f2f2;

border-radius: 10px;

padding: 20px;

margin-bottom: 30px;

}

.pagination {

display: flex;

justify-content: center;

align-items: center;

margin-top: 20px;

}

.pagination a {

color: #000;

padding: 8px 16px;

text-decoration: none;

transition: background-color .3s;

border: 1px solid #ddd;

margin: 0 4px;

}

.pagination a.active {

background-color: #4CAF50;

color: white;

border: 1px solid #4CAF50;

}

.pagination a:hover:not(.active) {

background-color: #ddd;

}

.button {

display: inline-block;

padding: 5px 10px;

font-size: 14px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #494949;

border: none;

border-radius: 5px;

}

.button:hover {background-color: #212121

}

.button:active {

background-color: #0062cc;

box-shadow: 1px 2px #a6a6a6;

transform: translateY(1px);

}

.buttonadmin {

padding: 5px 10px;

font-size: 14px;

cursor: pointer;

text-align: center;

text-decoration: none;

outline: none;

color: #fff;

background-color: #494949;

border: none;

border-radius: 5px;

width: 100px; /\* Фиксированная ширина кнопки \*/

height: 40px; /\* Фиксированная высота кнопки \*/

margin: 10px 0; /\* Отступы между кнопками \*/

}

.buttonadmin:hover {background-color: #212121

}

.buttonadmin:active {

background-color: #232323;

box-shadow: 1px 2px #a6a6a6;

transform: translateY(1px);

}

</style>

</head>

<body>

{% include 'inc/\_adminHeader.html' %}

{% block content %}

{% endblock %}

</body>

</html>

* + 1. Код мобильного приложения
       1. OrderAdapter.java

package com.example.dlboxmobile;

import android.content.Context;

import android.content.Intent;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

/\*\*

\* Класс OrderAdapter отвечает за работу со списком заказов для вывода в RecycleView.

\*/

public class OrderAdapter extends RecyclerView.Adapter<OrderAdapter.ViewHolder> {

private Context context;

private ArrayList<Order> list;

public OrderAdapter(Context context, ArrayList<Order> list) {

this.context = context;

this.list = list;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(context).inflate(R.layout.item\_order\_card, parent, false);

return new ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

Order order = list.get(position);

holder.tv.setText(order.getUniqueOrderNum());

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

openOrderDetails(order);

}

});

}

@Override

public int getItemCount() {

return list.size();

}

public class ViewHolder extends RecyclerView.ViewHolder{

TextView tv;

public ViewHolder(@NonNull View itemView) {

super(itemView);

tv = itemView.findViewById(R.id.item\_tv);

}

}

private void openOrderDetails(Order order) {

Intent intent = new Intent(context, OrderDetailsActivity.class);

intent.putExtra("order", order);

context.startActivity(intent);

}

}

* + - 1. TrackingAdapter.java

package com.example.dlboxmobile;

import android.content.Context;

import android.content.Intent;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

/\*\*

\* Класс TrackingAdapter отвечает за работу со списком отслеживаний для вывода в RecycleView.

\*/

public class TrackingAdapter extends RecyclerView.Adapter<TrackingAdapter.ViewHolder> {

private Context context;

private ArrayList<Tracking> list;

public TrackingAdapter(Context context, ArrayList<Tracking> list) {

this.context = context;

this.list = list;

}

@NonNull

@Override

public TrackingAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(context).inflate(R.layout.item\_tracking\_card, parent, false);

return new TrackingAdapter.ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull TrackingAdapter.ViewHolder holder, int position) {

Tracking tracking = list.get(position);

holder.tv.setText(tracking.getUniqueOrderNum());

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

openOrderDetails(tracking);

}

});

}

@Override

public int getItemCount() {

return list.size();

}

public class ViewHolder extends RecyclerView.ViewHolder{

TextView tv;

public ViewHolder(@NonNull View itemView) {

super(itemView);

tv = itemView.findViewById(R.id.item\_tv);

}

}

private void openOrderDetails(Tracking tracking) {

Intent intent = new Intent(context, TrackingDetailsActivity.class);

intent.putExtra("tracking", tracking);

context.startActivity(intent);

}

}

* + - 1. ItemAdapter.java

package com.example.dlboxmobile;

import android.content.Context;

import android.content.Intent;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

/\*\*

\* Класс ItemAdapter отвечает за работу со списком пользователей для вывода в RecycleView.

\*/

public class ItemAdapter extends RecyclerView.Adapter<ItemAdapter.ViewHolder> {

private Context context;

private ArrayList<MyUser> list;

public ItemAdapter(Context context, ArrayList<MyUser> list) {

this.context = context;

this.list = list;

}

@NonNull

@Override

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(context).inflate(R.layout.item\_card, parent, false);

return new ViewHolder(view);

}

@Override

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {

MyUser myuser = list.get(position);

holder.tv.setText(myuser.getLogin());

// Установка обработчика нажатия на элемент списка

holder.itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// Открытие подробной информации о пользователе

openUserDetails(myuser);

}

});

}

@Override

public int getItemCount() {

return list.size();

}

public class ViewHolder extends RecyclerView.ViewHolder{

TextView tv;

public ViewHolder(@NonNull View itemView) {

super(itemView);

tv = itemView.findViewById(R.id.item\_tv);

}

}

private void openUserDetails(MyUser user) {

Intent intent = new Intent(context, UserDetailsActivity.class);

intent.putExtra("user", user);

context.startActivity(intent);

}

}

* + - 1. MyUser.java

package com.example.dlboxmobile;

import java.io.Serializable;

/\*\*

\* Класс MyUser отвечает за модель пользователя.

\* Он содержит переменные, конструктор, getter и setter.

\*/

public class MyUser implements Serializable{

private String login;

private String password;

private String user\_surname;

private String user\_name;

private String user\_patronymic;

private String telephoneNumber;

private String role;

public MyUser(String login, String password) {

this.login = login;

this.password = password;

}

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getUser\_surname() {

return user\_surname;

}

public void setUser\_surname(String user\_surname) {

this.user\_surname = user\_surname;

}

public String getUser\_name() {

return user\_name;

}

public void setUser\_name(String user\_name) {

this.user\_name = user\_name;

}

public String getUser\_patronymic() {

return user\_patronymic;

}

public void setUser\_patronymic(String user\_patronymic) {

this.user\_patronymic = user\_patronymic;

}

public String getTelephoneNumber() {

return telephoneNumber;

}

public void setTelephoneNumber(String telephoneNumber) {

this.telephoneNumber = telephoneNumber;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public MyUser(String login, String password, String user\_surname, String user\_name, String user\_patronymic, String telephoneNumber, String role) {

this.login = login;

this.password = password;

this.user\_surname = user\_surname;

this.user\_name = user\_name;

this.user\_patronymic = user\_patronymic;

this.telephoneNumber = telephoneNumber;

this.role = role;

}

public MyUser(String login, String password, String user\_surname, String user\_name, String user\_patronymic, String telephoneNumber) {

this.login = login;

this.password = password;

this.user\_surname = user\_surname;

this.user\_name = user\_name;

this.user\_patronymic = user\_patronymic;

this.telephoneNumber = telephoneNumber;

}

}

* + - 1. Order.java

package com.example.dlboxmobile;

import java.io.Serializable;

/\*\*

\* Класс Order отвечает за модель заказа.

\* Он содержит переменные, конструктор, getter и setter.

\*/

public class Order implements Serializable {

private String uniqueOrderNum;

private String box;

private String delivery;

private boolean doorTodoor;

private boolean delicate;

private float cost;

private String pay;

private String timeAdd;

private String sender\_address;

private String recipient\_address;

private boolean postmat;

public Order(String uniqueOrderNum, String box, String delivery, boolean doorTodoor, boolean delicate, float cost, String pay, String timeAdd, String sender\_address, String recipient\_address, boolean postmat) {

this.uniqueOrderNum = uniqueOrderNum;

this.box = box;

this.delivery = delivery;

this.doorTodoor = doorTodoor;

this.delicate = delicate;

this.cost = cost;

this.pay = pay;

this.timeAdd = timeAdd;

this.sender\_address = sender\_address;

this.recipient\_address = recipient\_address;

this.postmat = postmat;

}

public String getUniqueOrderNum() {

return uniqueOrderNum;

}

public void setUniqueOrderNum(String uniqueOrderNum) {

this.uniqueOrderNum = uniqueOrderNum;

}

public String getBox() {

return box;

}

public void setBox(String box) {

this.box = box;

}

public String getDelivery() {

return delivery;

}

public void setDelivery(String delivery) {

this.delivery = delivery;

}

public boolean isDoorTodoor() {

return doorTodoor;

}

public void setDoorTodoor(boolean doorTodoor) {

this.doorTodoor = doorTodoor;

}

public boolean isDelicate() {

return delicate;

}

public void setDelicate(boolean delicate) {

this.delicate = delicate;

}

public float getCost() {

return cost;

}

public void setCost(float cost) {

this.cost = cost;

}

public String getPay() {

return pay;

}

public void setPay(String pay) {

this.pay = pay;

}

public String getTimeAdd() {

return timeAdd;

}

public void setTimeAdd(String timeAdd) {

this.timeAdd = timeAdd;

}

public String getSender\_address() {

return sender\_address;

}

public void setSender\_address(String sender\_address) {

this.sender\_address = sender\_address;

}

public String getRecipient\_address() {

return recipient\_address;

}

public void setRecipient\_address(String recipient\_address) {

this.recipient\_address = recipient\_address;

}

public boolean isPostmat() {

return postmat;

}

public void setPostmat(boolean postmat) {

this.postmat = postmat;

}

}

* + - 1. Tracking.java

package com.example.dlboxmobile;

import java.io.Serializable;

/\*\*

\* Класс Tracking отвечает за модель отслеживания.

\* Он содержит переменные, конструктор, getter и setter.

\*/

public class Tracking implements Serializable {

private String uniqueOrderNum;

private String status;

public Tracking(String uniqueOrderNum, String status) {

this.uniqueOrderNum = uniqueOrderNum;

this.status = status;

}

public String getUniqueOrderNum() {

return uniqueOrderNum;

}

public void setUniqueOrderNum(String uniqueOrderNum) {

this.uniqueOrderNum = uniqueOrderNum;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

}

* + - 1. MainActivity.java

package com.example.dlboxmobile;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import org.mindrot.jbcrypt.BCrypt;

import java.io.IOException;

import java.util.ArrayList;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

import retrofit2.Retrofit;

import retrofit2.converter.gson.GsonConverterFactory;

/\*\*

\* Класс MainActivity отвечает за авторизацию пользователя в системе.

\* Он содержит методы для отправки запроса на авторизацию пользователя и обработки ответа от сервера.

\*/

public class MainActivity extends AppCompatActivity {

TextView txt;

Button loginButton;

EditText loginInput, passwordInput;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

txt = findViewById(R.id.txt);

loginInput = findViewById(R.id.login\_input);

passwordInput = findViewById(R.id.password\_input);

loginButton = findViewById(R.id.btn\_login);

loginButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String login = loginInput.getText().toString();

String password = passwordInput.getText().toString();

MyUser user = new MyUser(login, password);

loginUser(user, v);

}

});

}

private void loginUser(MyUser user, View view) {

api userService = ApiClient.getClient().create(api.class);

Call<MyUser> call = userService.loginUser(user);

call.enqueue(new Callback<MyUser>() {

@SuppressLint("SetTextI18n")

@Override

public void onResponse(Call<MyUser> call, Response<MyUser> response) {

if (response.isSuccessful()) {

Toast.makeText(MainActivity.this, "Данные " + response.body(), Toast.LENGTH\_SHORT).show();

// Сохранение данных пользователя

UserSaveInfo userSaveInfo = new UserSaveInfo(MainActivity.this);

MyUser user = response.body();

userSaveInfo.saveUserInfo(user);

Log.d("MainActivity", "пользователя: " + userSaveInfo.getUserInfo().getLogin());

startActivity(new Intent(MainActivity.this, HomeActivity.class));

} else { // Авторизация не удалась

switch (response.code()) {

case 400:

txt.setText("Неверный логин или пароль");

txt.setVisibility(View.VISIBLE);

break;

case 404:

txt.setText("Пользователь не найден");

txt.setVisibility(View.VISIBLE);

break;

default:

txt.setText("Ошибка авторизации: " + response.code());

txt.setVisibility(View.VISIBLE);

break;

}

}

}

@SuppressLint("SetTextI18n")

@Override

public void onFailure(Call<MyUser> call, Throwable t) {

txt.setText("Произошла сетевая ошибка");

txt.setVisibility(View.VISIBLE);

}

});

}

}

* + - 1. HomeActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import java.util.ArrayList;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

/\*\*

\* Класс HomeActivity отвечает за навигацию между окнами.

\* Он содержит метод для перехода по окнам.

\*/

public class HomeActivity extends AppCompatActivity {

Button btnUsers, btnOrders, btnTrackings, btnAccount;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_home);

btnUsers = findViewById(R.id.btnUsers);

btnOrders = findViewById(R.id.btnOrders);

btnTrackings = findViewById(R.id.btnTrackings);

btnAccount = findViewById(R.id.btnAccount);

btnUsers.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(HomeActivity.this, UsersActivity.class));

}

});

btnOrders.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(HomeActivity.this, OrdersActivity.class));

}

});

btnTrackings.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(HomeActivity.this, TrackingActivity.class));

}

});

btnAccount.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(HomeActivity.this, AccountActivity.class));

}

});

}

}

* + - 1. AccountActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Button;

import android.widget.TextView;

/\*\*

\* Класс AccountActivity отвечает за отображение авторизованного данных пользователя.

\* Он содержит методы для вывода данных.

\*/

public class AccountActivity extends AppCompatActivity {

TextView login, name, surname, patronymic, telephone;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_account);

login = findViewById(R.id.tvLogin);

name = findViewById(R.id.tvUserName);

surname = findViewById(R.id.tvUserSurname);

patronymic = findViewById(R.id.tvUserPatronymic);

telephone = findViewById(R.id.tvTelephoneNumber);

UserSaveInfo userSaveInfo = new UserSaveInfo(AccountActivity.this);

MyUser user = userSaveInfo.getUserInfo();

if (user != null) {

// Отображение информации о пользователе

login.setText(user.getLogin());

surname.setText(user.getUser\_surname());

name.setText(user.getUser\_name());

patronymic.setText(user.getUser\_patronymic());

telephone.setText(user.getTelephoneNumber());

}

Button btnBack = findViewById(R.id.btnBack);

btnBack.setOnClickListener(v -> onBackPressed());

}

}

* + - 1. UsersActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import android.util.Log;

import android.widget.Toast;

import java.util.ArrayList;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

/\*\*

\* Класс UsersActivity отвечает за вывод списка пользователей.

\* Он содержит методы для работы с API.

\*/

public class UsersActivity extends AppCompatActivity {

RecyclerView recyclerView;

com.example.dlboxmobile.api apiInterface;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_users);

recyclerView = findViewById(R.id.recycle\_view);

apiInterface = ApiClient.getClient().create(api.class);

Call<ArrayList<MyUser>> getList = apiInterface.getUsersList();

getList.enqueue(new Callback<ArrayList<MyUser>>() {

@Override

public void onResponse(Call<ArrayList<MyUser>> call, Response<ArrayList<MyUser>> response) {

if (response.isSuccessful()){

recyclerView.setLayoutManager(new LinearLayoutManager(UsersActivity.this));

recyclerView.setHasFixedSize(true);

ArrayList<MyUser> list = response.body();

ItemAdapter adapter = new ItemAdapter(UsersActivity.this, list);

recyclerView.setAdapter(adapter);

assert list != null;

for (MyUser user : list) {

// имена пользователей null, когда login имеет данные

Log.d("UsersActivity", "Имя пользователя: " + user.getUser\_name());

}

}

else {

Toast.makeText(UsersActivity.this, response.message(), Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onFailure(Call<ArrayList<MyUser>> call, Throwable t) {

Toast.makeText(UsersActivity.this, "", Toast.LENGTH\_SHORT).show();

}

});

}

}

* + - 1. OrdersActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import android.util.Log;

import android.widget.Toast;

import java.util.ArrayList;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

/\*\*

\* Класс OrdersActivity отвечает за вывод списка заказов.

\* Он содержит методы для работы с API.

\*/

public class OrdersActivity extends AppCompatActivity {

RecyclerView recyclerView;

com.example.dlboxmobile.api apiInterface;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_orders);

Log.d("OrdersActivity", "в классе");

recyclerView = findViewById(R.id.recycle\_view);

apiInterface = ApiClient.getClient().create(api.class);

Call<ArrayList<Order>> getList = apiInterface.getOrderList();

Log.d("OrdersActivity", getList.toString());

getList.enqueue(new Callback<ArrayList<Order>>() {

@Override

public void onResponse(Call<ArrayList<Order>> call, Response<ArrayList<Order>> response) {

if (response.isSuccessful()){

Log.d("OrdersActivity", "в респонсе");

recyclerView.setLayoutManager(new LinearLayoutManager(OrdersActivity.this));

recyclerView.setHasFixedSize(true);

ArrayList<Order> list = response.body();

OrderAdapter adapter = new OrderAdapter(OrdersActivity.this, list);

recyclerView.setAdapter(adapter);

for (Order order : list) {

Log.d("HomeActivity", "Номер заказа: " + order.getUniqueOrderNum());

}

} else {

Toast.makeText(OrdersActivity.this, response.message(), Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onFailure(Call<ArrayList<Order>> call, Throwable t) {

Toast.makeText(OrdersActivity.this, "fail", Toast.LENGTH\_SHORT).show();

}

});

}

}

* + - 1. TrackingActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import android.util.Log;

import android.widget.Toast;

import java.util.ArrayList;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

/\*\*

\* Класс TrackingActivity отвечает за вывод списка отслеживаний.

\* Он содержит методы для работы с API.

\*/

public class TrackingActivity extends AppCompatActivity {

RecyclerView recyclerView;

com.example.dlboxmobile.api apiInterface;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_tracking);

recyclerView = findViewById(R.id.recycle\_view);

apiInterface = ApiClient.getClient().create(api.class);

Call<ArrayList<Tracking>> getList = apiInterface.getTrackingList();

getList.enqueue(new Callback<ArrayList<Tracking>>() {

@Override

public void onResponse(Call<ArrayList<Tracking>> call, Response<ArrayList<Tracking>> response) {

if (response.isSuccessful()){

recyclerView.setLayoutManager(new LinearLayoutManager(TrackingActivity.this));

recyclerView.setHasFixedSize(true);

ArrayList<Tracking> list = response.body();

TrackingAdapter adapter = new TrackingAdapter(TrackingActivity.this, list);

recyclerView.setAdapter(adapter);

for (Tracking tracking : list) {

Log.d("TrackingActivity", "Номер заказа: " + tracking.getUniqueOrderNum());

}

} else {

Toast.makeText(TrackingActivity.this, response.message(), Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onFailure(Call<ArrayList<Tracking>> call, Throwable t) {

Toast.makeText(TrackingActivity.this, "fail", Toast.LENGTH\_SHORT).show();

}

});

}

}

* + - 1. TrackingDetailsActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import okhttp3.ResponseBody;

import retrofit2.Call;

import retrofit2.Callback;

import retrofit2.Response;

/\*\*

\* Класс TrackingDetailsActivity отвечает за вывод данных отслеживания.

\*/

public class TrackingDetailsActivity extends AppCompatActivity {

TextView trackNum, status, txt;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_tracking\_details);

trackNum = findViewById(R.id.tvtrackNum);

status = findViewById(R.id.tvstatus);

txt = findViewById(R.id.txt);

// Получение данных из предыдущей активности

Tracking tracking = (Tracking) getIntent().getSerializableExtra("tracking");

if (tracking != null) {

// Отображение информации

Log.d("TrackingDetailsActivity", "ИНФА: " + tracking.getStatus());

trackNum.setText(tracking.getUniqueOrderNum());

if (status.getText().toString().equals("3")){

status.setText("Доставлен");

}

else{

status.setText("Оформлен");

}

}

Button btnBack = findViewById(R.id.btnBack);

btnBack.setOnClickListener(v -> onBackPressed());

Button changeStatusBtn = findViewById(R.id.btnChangeStatus);

changeStatusBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String tracknum = trackNum.getText().toString();

Tracking tracking1 = new Tracking(tracknum, "Доставлен");

changestatus(tracking1, v);

// if (status.getText() == "3"){

//

// }

// else{

// Toast.makeText(TrackingDetailsActivity.this, "Посылка уже доставлена", Toast.LENGTH\_SHORT).show();

// }

}

});

}

private void changestatus(Tracking tracking, View view) {

api userService = ApiClient.getClient().create(api.class);

Call<Tracking> call = userService.changeStatus(tracking);

Log.d("TrackingDetailsActivity", "ИНФА ЗАПРОСА: " + tracking.getUniqueOrderNum() + tracking.getStatus());

call.enqueue(new Callback<Tracking>() {

@SuppressLint("SetTextI18n")

@Override

public void onResponse(Call<Tracking> call, Response<Tracking> response) {

if (response.isSuccessful()) {

Toast.makeText(TrackingDetailsActivity.this, "Данные " + response.body(), Toast.LENGTH\_SHORT).show();

Log.d("TrackingDetailsActivity", "ИНФА ТРЕКА: " + response.body());

} else {

switch (response.code()) {

case 400:

txt.setText("Неверный логин или пароль");

txt.setVisibility(View.VISIBLE);

break;

case 404:

txt.setText("Пользователь не найден");

txt.setVisibility(View.VISIBLE);

break;

default:

txt.setText("Ошибка авторизации: " + response.code());

txt.setVisibility(View.VISIBLE);

break;

}

}

}

@SuppressLint("SetTextI18n")

@Override

public void onFailure(Call<Tracking> call, Throwable t) { // Ошибка сетевого запроса

txt.setText("Произошла сетевая ошибка");

txt.setVisibility(View.VISIBLE);

}

});

}

}

* + - 1. OrderDetailsActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.ViewDebug;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.TextView;

/\*\*

\* Класс OrderDetailsActivity отвечает за вывод данных заказа.

\*/

public class OrderDetailsActivity extends AppCompatActivity {

TextView trackNum, cost, pay, sender, recipient;

CheckBox doortodoor, postmat;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_order\_details);

trackNum = findViewById(R.id.tvtrackNum);

cost = findViewById(R.id.tvcost);

pay = findViewById(R.id.tvPayType);

sender = findViewById(R.id.tvadSender);

recipient = findViewById(R.id.tvadRecipient);

doortodoor = findViewById(R.id.tvdoorToDoor);

postmat = findViewById(R.id.tvpostmat);

// Получение данных из предыдущей активности

Order order = (Order) getIntent().getSerializableExtra("order");

if (order != null) {

// Отображение информации

trackNum.setText(order.getUniqueOrderNum());

cost.setText(String.valueOf(order.getCost()));

if (pay.getText().toString().equals("2")){

pay.setText("Картой при получении");

}

else{

pay.setText("Наличными при получении");

}

sender.setText(order.getSender\_address());

recipient.setText(order.getRecipient\_address());

doortodoor.setChecked(order.isDoorTodoor());

postmat.setChecked(order.isPostmat());

}

Button btnBack = findViewById(R.id.btnBack);

btnBack.setOnClickListener(v -> onBackPressed());

}

}

* + - 1. UserDetailsActivity.java

package com.example.dlboxmobile;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Button;

import android.widget.TextView;

/\*\*

\* Класс UserDetailsActivity отвечает за вывод данных пользователя.

\*/

public class UserDetailsActivity extends AppCompatActivity {

TextView tvLogin, tvUserSurname, tvUserName, tvUserPatronymic, tvTelephoneNumber;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_user\_details);

tvLogin = findViewById(R.id.tvLogin);

tvUserSurname = findViewById(R.id.tvUserSurname);

tvUserName = findViewById(R.id.tvUserName);

tvUserPatronymic = findViewById(R.id.tvUserPatronymic);

tvTelephoneNumber = findViewById(R.id.tvTelephoneNumber);

// Получение данных о пользователе из предыдущей активности

MyUser user = (MyUser) getIntent().getSerializableExtra("user");

if (user != null) {

// Отображение информации о пользователе

tvLogin.setText(user.getLogin());

tvUserSurname.setText(user.getUser\_surname());

tvUserName.setText(user.getUser\_name());

tvUserPatronymic.setText(user.getUser\_patronymic());

tvTelephoneNumber.setText(user.getTelephoneNumber());

}

// Оптимизация

Button btnBack = findViewById(R.id.btnBack);

btnBack.setOnClickListener(v -> onBackPressed());

}

}

* + - 1. ApiClient.java

package com.example.dlboxmobile;

import okhttp3.OkHttpClient;

import okhttp3.logging.HttpLoggingInterceptor;

import retrofit2.Retrofit;

import retrofit2.converter.gson.GsonConverterFactory;

/\*\*

\* Класс ApiClient отвечает за подключение к базе данных.

\* Он содержит метод для работы с запросами.

\*/

public class ApiClient {

private static final String URL = "http://192.168.0.112:8000/api/";

private static Retrofit retrofit = null;

public static Retrofit getClient() {

HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor();

interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);

OkHttpClient client = new OkHttpClient.Builder().addInterceptor(interceptor).build();

if (retrofit == null) {

retrofit = new Retrofit.Builder()

.baseUrl(URL)

.client(client)

.addConverterFactory(GsonConverterFactory.create())

.build();

}

return retrofit;

}

}

* + - 1. api.java

package com.example.dlboxmobile;

import java.util.ArrayList;

import java.util.List;

import okhttp3.ResponseBody;

import retrofit2.Call;

import retrofit2.http.Body;

import retrofit2.http.DELETE;

import retrofit2.http.Field;

import retrofit2.http.FormUrlEncoded;

import retrofit2.http.GET;

import retrofit2.http.POST;

import retrofit2.http.PUT;

import retrofit2.http.Path;

/\*\*

\* Интерфейс api для подключения к базе данных.

\*/

public interface api {

@GET("users/")

Call<ArrayList<MyUser>> getUsersList();

@POST("login/")

Call<MyUser> loginUser(@Body MyUser user);

@POST("trackchange/")

Call<Tracking> changeStatus (@Body Tracking tracking);

@GET("order/")

Call<ArrayList<Order>> getOrderList();

@GET("tracking/")

Call<ArrayList<Tracking>> getTrackingList();

}

* + - 1. UserSaveInfo.java

package com.example.dlboxmobile;

import android.content.Context;

import android.content.SharedPreferences;

public class UserSaveInfo {

private final SharedPreferences sharedPreferences;

public UserSaveInfo(Context context) {

sharedPreferences = context.getSharedPreferences("UserData", Context.MODE\_PRIVATE);

}

public void saveUserInfo(MyUser user) {

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("login", user.getLogin());

editor.putString("password", user.getPassword());

editor.putString("name", user.getUser\_name());

editor.putString("sername", user.getUser\_surname());

editor.putString("patronymic", user.getUser\_patronymic());

editor.putString("telephoneNumber", user.getTelephoneNumber());

editor.apply();

}

public MyUser getUserInfo() {

String login = sharedPreferences.getString("login", null);

String hash\_password = sharedPreferences.getString("password", null);

String name = sharedPreferences.getString("name", null);

String sername = sharedPreferences.getString("sername", null);

String patronymic = sharedPreferences.getString("patronymic", null);

String telephoneNumber = sharedPreferences.getString("telephoneNumber", null);

return new MyUser(login, hash\_password, name, sername, patronymic, telephoneNumber);

}

}

* + - 1. Activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/txt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Авторизация"

android:textSize="25dp"

android:layout\_marginBottom="100dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

<EditText

android:id="@+id/login\_input"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toTopOf="@+id/txt"

android:layout\_margin="40dp"

android:hint="Login"

/>

<EditText

android:id="@+id/password\_input"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toTopOf="@+id/login\_input"

android:layout\_margin="40dp"

android:hint="Password"

android:inputType="textPassword"

/>

<Button

android:id="@+id/btn\_login"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@+id/password\_input"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:layout\_margin="40dp"

android:text="войти"

/>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_home.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".HomeActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<Button

android:id="@+id/btnUsers"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Перейти к пользователям"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

<Button

android:id="@+id/btnOrders"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Перейти к заказам"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

<Button

android:id="@+id/btnTrackings"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Перейти к отслеживаниям"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text=" "/>

<Button

android:id="@+id/btnAccount"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Аккаунт"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_users.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".UsersActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recycle\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

android:background="#C59292"/>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_orders.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".OrdersActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recycle\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

android:background="#C59292"/>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_tracking.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".TrackingActivity">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recycle\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

android:background="#C59292"/>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_account.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".AccountActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/txt"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Ваш аккаунт"

android:textStyle="bold"

android:textSize="30dp"

android:layout\_marginTop="70dp"/>

<TextView

android:id="@+id/Login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Логин: "

android:textStyle="bold"

android:layout\_marginTop="20dp"/>

<TextView

android:id="@+id/tvLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:text=" "

/>

<TextView

android:id="@+id/UserSurname"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Фамилия: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvUserSurname"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/UserName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Имя: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvUserName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/UserPatronymic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Отчество: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvUserPatronymic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/TelephoneNumber"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Номер телефона: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvTelephoneNumber"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<Button

android:id="@+id/btnBack"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Назад"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_order\_details.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".UserDetailsActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/trackNum"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Трек-номер: "

android:textStyle="bold"

android:layout\_marginTop="20dp"/>

<TextView

android:id="@+id/tvtrackNum"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:text=" "

/>

<TextView

android:id="@+id/doorToDoor"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="От двери до двери: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<CheckBox

android:id="@+id/tvdoorToDoor"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:clickable="false"

android:focusable="false"

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/postmat"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Доставка в постамат: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<CheckBox

android:id="@+id/tvpostmat"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:clickable="false"

android:focusable="false"

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/cost"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Стоимость: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvcost"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/PayType"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Оплата: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvPayType"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/adSender"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Адрес отправителя: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvadSender"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/adRecipient"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Адрес получателя: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvadRecipient"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<Button

android:id="@+id/btnBack"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Назад"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_user\_details.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".UserDetailsActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/Login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Логин: "

android:textStyle="bold"

android:layout\_marginTop="20dp"/>

<TextView

android:id="@+id/tvLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:text=" "

/>

<TextView

android:id="@+id/UserSurname"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Фамилия: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvUserSurname"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/UserName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Имя: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvUserName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/UserPatronymic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Отчество: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvUserPatronymic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<TextView

android:id="@+id/TelephoneNumber"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Номер телефона: "

android:textStyle="bold"

android:layout\_marginTop="20dp" />

<TextView

android:id="@+id/tvTelephoneNumber"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text=" "

android:layout\_marginStart="10dp"

android:layout\_marginTop="8dp" />

<Button

android:id="@+id/btnBack"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Назад"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Activity\_tracking\_details.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".TrackingDetailsActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/trackNum"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Трек-номер: "

android:textStyle="bold"

android:layout\_marginTop="20dp"/>

<TextView

android:id="@+id/tvtrackNum"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:text=" "

/>

<TextView

android:id="@+id/status"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Статус доставки: "

android:textStyle="bold"

android:layout\_marginTop="20dp"/>

<TextView

android:id="@+id/tvstatus"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:text=" "

/>

<Button

android:id="@+id/btnChangeStatus"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Отметить доставленым"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

<TextView

android:id="@+id/txt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

/>

<Button

android:id="@+id/btnBack"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Назад"

android:textStyle="bold"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

* + - 1. Item\_card.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="140dp"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_margin="16dp"

app:cardElevation="30dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/item\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

* + - 1. Item\_order\_card.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="140dp"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_margin="16dp"

app:cardElevation="30dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/item\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

* + - 1. Item\_tracking\_card.xml

<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="140dp"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_margin="16dp"

app:cardElevation="30dp">

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/item\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>