

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ
УНИВЕРСИТЕТ

Кафедра информационных систем
и технологий в строительстве

КУРСОВАЯ РАБОТА

По дисциплине «Методы и средства проектирования информационных систем»

«Проектирование интернет-системы записи и учета скидок клиентов салона
красоты»

Выполнил: студент 3 курса 1ИС02

Егорова Д.А.

Проверил: доцент, кандидат технических наук

Галимов Р.А.

Казань, 2024

Содержание

Введение	3
1. Формулировка задачи	5
2. Техническое задание	6
2.1 Описание системы	6
2.2 Регистрация и авторизация	6
2.3 Требования к разграничению доступа.....	7
3. Эскизный проект	7
3.1. Диаграмма прецедентов	7
3.2 Диаграмма классов	8
4. Программная реализация	10
4.1 Приложение Users	10
4.2 Приложение Salon	14
4.3 Приложение Appointments	17
4.4 Шаблоны приложения	23
5. Визуальное представление	36
5.1 Сайт	36
5.2 База данных	41

Введение

Django - это веб-фреймворк, основанный на принципах проектирования, способствующих созданию шаблонов и повторному использованию кода. Он способствует группированию функциональностей и повторному использованию приложений. Разработанные приложения на Django кросс-платформенны и могут быть запущены на различных версиях Linux, Windows и MacOS.

В Django веб-фреймворки можно выделить два типа:

- 1) Прямое включение в решение конкретных задач. Такие фреймворки обычно предлагают быстрое развертывание в определенной области, но ограничивают выбор компонентов.
- 2) Гибкие фреймворки, которые дают разработчику больше свободы в выборе компонентов и инструментов для выполнения задач. Разработчику приходится самому подбирать компоненты для сборки.

Когда приложение получает запрос, оно обрабатывает его и выполняет заданные действия, после чего возвращает ответ пользовательскому браузеру. Этот ответ может быть как статическим, так и динамическим, в зависимости от логики приложения.

По адаптации к уже имеющимся запросам по URL каждой отдельной причине отдельной функции, удобнее написать отдельную функцию для каждого ресурса для обработки запросов. Диспетчер URL-адресов используется для направления HTTP-запросов к соответствующим представлениям и может извлекать данные из URL-адреса, соответствуя шаблонам, и передавать их в соответствующую функцию в качестве аргумента.

Модели предоставляют способы управления данными (добавление, изменение, удаление) и выполнения запросов в базе данных.

Шаблоны - это текстовые файлы, которые определяют структуру (HTML-страницу) с полями, используемыми для вставки актуального содержимого из базы данных или параметров, введенных пользователями.

Представление играет ключевую роль в архитектуре. Это ресурс, который при получении запроса обрабатывает его определенным образом. В ходе обработки представление может обращаться к моделям в базе данных для извлечения или сохранения данных.

1. Формулировка задачи

Формулировка задачи для создания веб-приложения для системы учета клиентской записи салона красоты с использованием фреймворка Django включает следующие пункты:

Анализ требований: Проведение детального анализа для определения ключевых функциональных и нефункциональных требований к приложению.

Проектирование базы данных: Разработка схемы базы данных для хранения информации о пользователях, их записях и других данных. В проектировании базы данных особое внимание будет уделено требованиям к безопасности и целостности данных.

Разработка моделей Django: Создание моделей в Django, которые будут соответствовать структуре базы данных. Определение и реализация связей между моделями для обеспечения эффективного хранения и быстрого доступа к данным.

Разработка пользовательского интерфейса: Создание веб-интерфейса с использованием HTML, CSS и JavaScript. Интерфейс должен быть интуитивно понятным и удобным, обеспечивающим доступ к основным функциям приложения, включая формы регистрации и входа, записи на услуги, их отслеживание и обновление.

2. Техническое задание

2.1 Описание системы

Система состоит из следующих основных функциональных блоков:

1. Регистрация, аутентификация и авторизация
2. Функционал для клиента
3. Функционал для администратора

Система предусматривает два типа пользователей: клиент и администратор. Клиент может зарегистрироваться, авторизоваться, записаться за услугу, просмотреть список услуг в личном кабинете, просмотреть и изменить профиль. Администратор может просматривать информацию о всех записях, добавлять новые услуги и записи, а также их удалять, просмотреть применение скидок клиентами, а также добавлять новые бонусы для зарегистрированных пользователей.

2.2 Регистрация и авторизация

Процесс регистрации клиента должен быть реализован в интерфейсах системы. При регистрации клиента должны быть запрошены следующие данные:

1. Имя и фамилия
2. Имя пользователя (login)
3. Адрес электронной почты
4. Номер телефона
5. Пароль

Аутентификация клиента осуществляется по имени пользователя и паролю.

2.3 Требования к разграничению доступа

Пользователей сайта можно разделить на 2 группы в соответствии с правами доступа:

1. Клиенты
2. Администратор

Клиент может зарегистрироваться, авторизоваться, записаться на услугу, просмотреть список услуг в личном кабинете, просмотреть и изменить профиль.

Администратор может просматривать информацию о всех записях, добавлять новые услуги и записи, а также их удалять, просмотреть применение скидок клиентами, а также добавлять новые бонусы для зарегистрированных пользователей. Управление данными происходит через панель администратора.

3. Эскизный проект

Диаграмма в языке моделирования UML, представляет собой графическое представление элементов модели системы, где связи между элементами отображают отношения между ними. Различные виды диаграмм UML, такие как диаграммы классов, компонентов, объектов и др., используются для визуализации различных аспектов устройства или поведения моделируемой системы. Каждый вид диаграммы UML имеет свою специфику и предназначен для отображения определенных аспектов системы.

3.1. Диаграмма прецедентов

Диаграмма UML (Unified Modeling Language) представляет собой графическое отображение элементов системы, где связи между элементами показывают их взаимоотношения. Существуют различные типы UML-

диаграмм, такие как диаграммы классов, компонентов, объектов и другие, которые используются для визуализации различных аспектов структуры и поведения системы. Каждая из этих диаграмм имеет свою уникальную специфику и предназначена для иллюстрации конкретных аспектов модели системы.

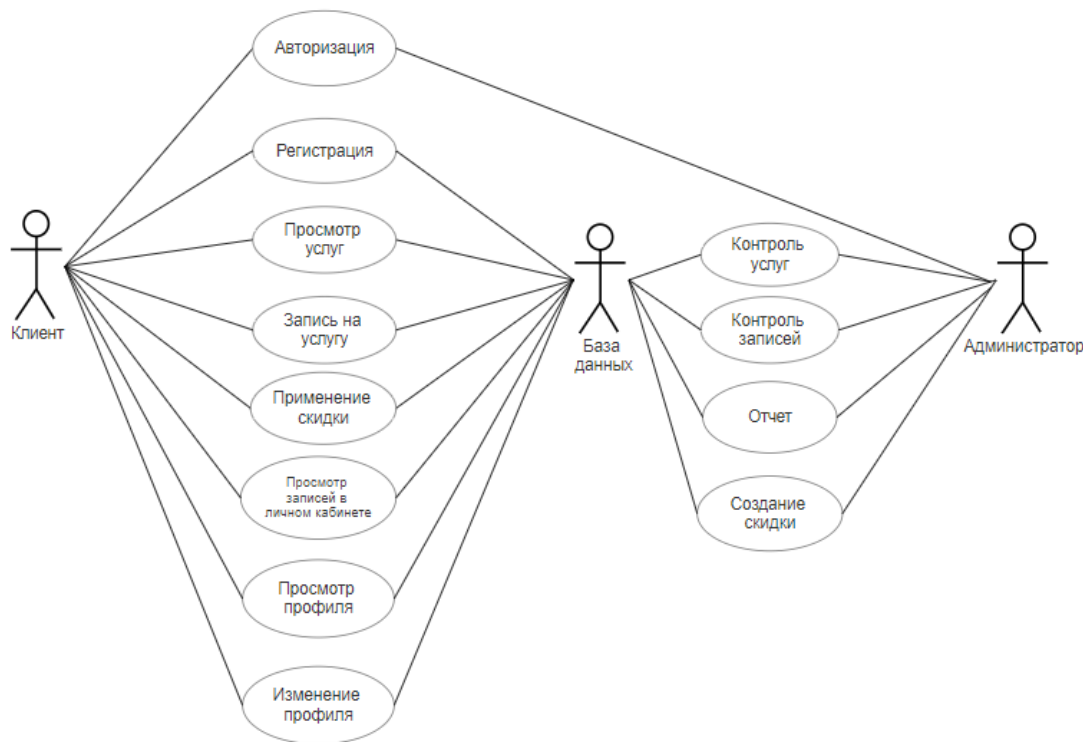


Рис. 1. Диаграмма прецендентов

3.2 Диаграмма классов

Диаграммы классов используются при моделировании наиболее часто. являются одной из форм статического описания системы с точки зрения её проектирования, показывая её структуру. Диаграмма классов не отображает динамическое поведение объектов изображенных на ней классов. На диаграмме классов показываются классы, интерфейс и отношения между ними.

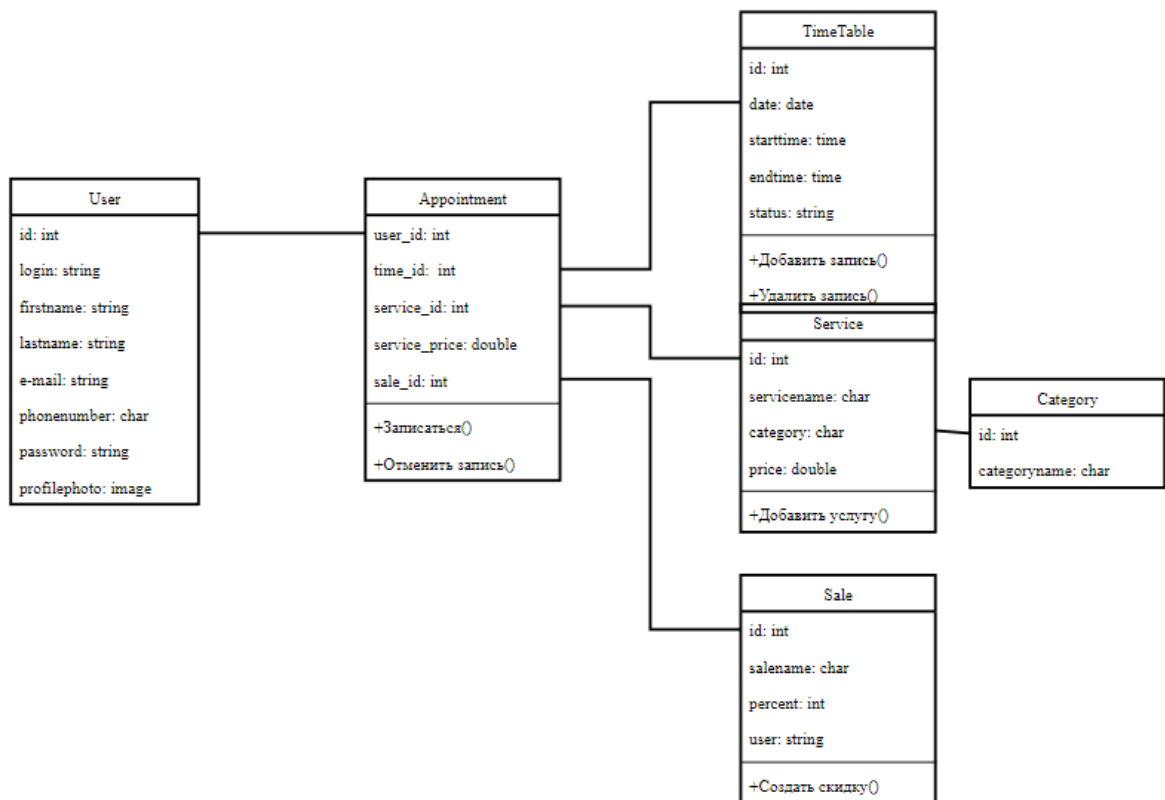


Рис. 2. Диаграмма классов

4. Программная реализация

4.1 Приложение Users

users\views.py

#Авторизация

```
def login(request):
    if request.method == 'POST':
        form = UserLoginForm(data=request.POST)
        if form.is_valid():
            username = request.POST['username']
            password = request.POST['password']
            user = auth.authenticate(username=username, password=password)
            if user:
                auth.login(request, user)
                return HttpResponseRedirect(reverse('index'))
        else:
            form = UserLoginForm()
    context = {
        'title': 'Авторизация',
        'form': form,
    }
    return render(request, 'users/login.html', context)
```

#Регистрация

```
def registration(request):
    if request.method == 'POST':
        form = UserRegistrationForm(data=request.POST)
        if form.is_valid():
            user = form.save()
            sale = Sale()
            sale.name = 'Скидка на первый заказ 20%'
            sale.user = user
            sale.percent = 20
            sale.save()
            messages.success(request, 'Регистрация прошла успешно!')
            return HttpResponseRedirect(reverse('users:login'))
        else:
            form = UserRegistrationForm
    context = {
        'title': 'Регистрация',
        'form': form,
    }
    return render(request, 'users/registration.html', context)
```

```

        #Просмотр профиля
@login_required
def profile(request):
    if request.method == 'POST':
        form = UserProfileForm(instance=request.user, data=request.POST, files=request.FILES)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect(reverse('users:profile'))
    else:
        form = UserProfileForm(instance=request.user)
    context = {
        'title': 'Профиль',
        'form': form,
        'appointment': Appointment.objects.filter(user=request.user),
        'sales': Sale.objects.filter(user=request.user),
    }
    return render(request, 'users/profile.html', context)

```

#Выход из профиля

```

def logout(request):
    auth.logout(request)
    return HttpResponseRedirect(reverse('index'))

```

#Панель администратора

```

def admin_panel(request):
    context = {
        'title': 'Панель администратора'
    }
    return render(request, 'users/admin-panel.html', context)

```

users\urls.py

```

from django.urls import path
from users.views import login, registration, profile, logout, admin_panel
app_name = 'users'
urlpatterns = [
    path('login/', login, name='login'),
    path('registration/', registration, name='registration'),
    path('profile/', profile, name='profile'),
    path('logout/', logout, name='logout'),
    path('admin_panel/', admin_panel, name='admin_panel'),
]

```

users\models.py

#Модель пользователя

```
from django.contrib.auth.models import AbstractUser
from django.db import models
class User(AbstractUser):
    image = models.ImageField(upload_to='users_images', blank=True, null=True)
    phone = models.CharField(max_length=20, blank=True, null=True)
```

users\forms.py

```
from django import forms
from django.contrib.auth.forms import AuthenticationForm, UserCreationForm,
UserChangeForm
from users.models import User

class UserLoginForm(AuthenticationForm):
    username = forms.CharField(widget=forms.TextInput(
        attrs={
            'placeholder': 'Введите имя пользователя',
            'class': 'form-input'
        }
    ))
    password = forms.CharField(widget=forms.PasswordInput(
        attrs={
            'placeholder': 'Введите пароль',
            'class': 'form-input'
        }
    )))
    class Meta:
        model = User
        fields = ('username', 'password')
class UserRegistrationForm(UserCreationForm):
    first_name = forms.CharField(widget=forms.TextInput(
        attrs={
            'placeholder': 'Введите имя',
            'class': 'form-input'
        }
    ))
    last_name = forms.CharField(widget=forms.TextInput(
        attrs={
            'placeholder': 'Введите фамилию',
            'class': 'form-input'
        }
    ))
    username = forms.CharField(widget=forms.TextInput(
        attrs={
            'placeholder': 'Введите имя пользователя',
```

```

        'class': 'form-input'
    }
))
email = forms.CharField(widget=forms.EmailInput(
    attrs={
        'placeholder': 'Введите адрес эл. почты',
        'class': 'form-input'
    }
))
phone = forms.CharField(widget=forms.TextInput(
    attrs={
        'class': 'form-input',
        'placeholder': 'Введите номер телефона'
    }
))
password1 = forms.CharField(widget=forms.PasswordInput(
    attrs={
        'placeholder': 'Введите пароль',
        'class': 'form-input'
    }
))
password2 = forms.CharField(widget=forms.PasswordInput(
    attrs={
        'placeholder': 'Подтвердите пароль',
        'class': 'form-input'
    }
))

class Meta:
    model = User
    fields = ('first_name', 'last_name', 'username', 'email', 'phone', 'password1', 'password2')
class UserProfileForm(UserChangeForm):
    first_name = forms.CharField(widget=forms.TextInput(
        attrs={
            'class': 'form-input'
        }
    ))
    last_name = forms.CharField(widget=forms.TextInput(
        attrs={
            'class': 'form-input'
        }
    ))
    image = forms.ImageField(widget=forms.FileInput(
        attrs={
            'class': 'custom-file-input',
        }
    ), required=False)
    username = forms.CharField(widget=forms.TextInput(
        attrs={

```

```

        'class': 'form-input',
        'readonly': True
    }
))
email = forms.CharField(widget=forms.EmailInput(
    attrs={
        'class': 'form-input',
    }
))
phone = forms.CharField(widget=forms.TextInput(
    attrs={
        'class': 'form-input',
    }
))
class Meta:
    model = User
    fields = ('first_name', 'last_name', 'image', 'username', 'email', 'phone')
users\admin.py

```

```

from django.contrib import admin
from users.models import User
admin.site.register(User)

```

4.2 Приложение Salon

salon\views.py

Добавление товара (услуги) администратором

```

@login_required
def service_add(request):
    if request.method == 'POST':
        form = ServiceForm(data=request.POST)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
    else:
        form = ServiceForm()
        context = {
            'title': 'Добавить товар',
            'form': form,
        }
    return render(request, 'salon/service-add.html', context)

```

Удаление товара (услуги) администратором

```

@login_required
def service_delete(request, service_id=None):
    service = Service.objects.get(id=service_id)
    service.delete()
    return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
    Отображение главное страницы

def index(request):
    context = {
        'title': 'Главная'
    }
    return render(request, 'salon/index.html', context)

```

Отображение страницы с услугами, фильтруя их по категории, если передан соответствующий идентификатор категории.

```

def service(request, category_id=None):
    if category_id:
        category = Category.objects.get(id=category_id)
        services = Service.objects.filter(category=category)
    else:
        services = Service.objects.all()

    context = {
        'title': 'Услуги',
        'services': services,
        'category': Category.objects.all()
    }
    return render(request, 'salon/service.html', context)

```

salon\urls.py

```

from django.urls import path
from salon.views import service_add, service_delete
app_name = 'salon'
urlpatterns = [
    path('service_add', service_add, name='service_add'),
    path('service_delete/<int:service_id>', service_delete, name='service_delete'),]

```

salon\models.py

Модель категорий услуг

```

class Category(models.Model):

```

```

name = models.CharField(max_length=256)
def __str__(self):
    return self.name

```

Модель услуг

```

class Service(models.Model):
    name = models.CharField(max_length=256)
    category = models.ForeignKey(Category, on_delete=models.CASCADE)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    def __str__(self):
        return f'{self.name} | Категория: {self.category.name}'

```

salon\forms.py

```

from django import forms
from salon.models import Service, Category
class ServiceForm(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(
        attrs={
            'class': 'form-input',
            'placeholder': 'Введите наименование услуги'
        }))
    category = forms.ModelChoiceField(widget=forms.Select(
        attrs={
            'class': 'form-input',
            'placeholder': 'Выберите категорию услуги'
        }), queryset=Category.objects.all(), required=False)
    price = forms.DecimalField(widget=forms.TextInput(
        attrs={
            'class': 'form-input',
            'placeholder': 'Введите цену услуги'
        }))
    class Meta:
        model = Service
        fields = ('name', 'category', 'price')

```

salon\apps.py

```

from django.apps import AppConfig
class SalonConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'salon'

```

salon\admin.py

```

from django.contrib import admin
from salon.models import Category, Service
admin.site.register(Category)

```



```
admin.site.register(Service)
```

4.3 Приложение Appointments

appointments\views.py

Отображение страницы выбора даты и времени для записи на услугу.

```
@login_required
def date_and_time(request, service_id):
    time = TimeTable.objects.all()
    date = DateTable.objects.all()
    service = Service.objects.get(id=service_id)
    context = {
        'time': time,
        'title': 'Выбор времени',
        'date': date,
        'service': service,
    }
    return render(request, 'appointments/date-and-time.html', context)
```

Отображение страницы редактирования расписания, где пользователь может добавлять и редактировать даты и времена записей на услугу.

```
@login_required
def date_and_time_edit(request):
    time = TimeTable.objects.all()
    date = DateTable.objects.all()
    if request.method == 'POST':
        form_date = DateForm(data=request.POST, files=request.FILES)
        if form_date.is_valid():
            form_date.save()
            return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
    else:
        form_date = DateForm()
    if request.method == 'POST':
        form_time = TimeForm(data=request.POST, files=request.FILES)
        if form_time.is_valid():
            form_time.save()
            return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
    else:
        form_time = TimeForm()
    context = {
        'time': time,
        'title': 'Редактор расписания',
    }
```

```

        'date': date,
        'form_date': form_date,
        'form_time': form_time
    }
    return render(request, 'appointments/date-and-time-edit.html', context)

```

Удаление времени и даты записи из расписания

```

@login_required
def date_and_time_delete(request, time_id):
    time = TimeTable.objects.get(id=time_id)
    time.delete()
    return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
def date_delete(request, date_id):
    date = DateTable.objects.get(id=date_id)
    date.delete()
    return HttpResponseRedirect(request.META.get('HTTP_REFERER'))

```

Запись на услугу

```

@login_required
def appointments(request, time_id, service_id):
    time = TimeTable.objects.get(id=time_id)
    service = Service.objects.get(id=service_id)
    sales = Sale.objects.filter(user=request.user)
    context = {
        'time': time,
        'service': service,
        'title': 'Запись',
        'sales': sales,
    }
    return render(request, 'appointments/appointment.html', context)

```

Запись на услугу с применением скидки

```

@login_required
def appointment_sale(request, time_id, service_id, sale_id):
    time = TimeTable.objects.get(id=time_id)
    service = Service.objects.get(id=service_id)
    sale = Sale.objects.get(id=sale_id)
    sale_percent = sale.percent
    service_price = service.price
    price = int(service_price) - sale_percent / 100 * int(service_price)
    context = {
        'time': time,

```

```

        'service': service,
        'title': 'Запись',
        'price': price,
        'sale': sale,
    }
    return render(request, 'appointments/appointments_sale.html', context)

```

Создание записи администратором

```

@login_required
def appointments_create(request, time_id, service_id, sale_id=None):
    time = TimeTable.objects.get(id=time_id)
    service = Service.objects.get(id=service_id)
    appointment = Appointment()
    appointment.time = time
    appointment.service = service
    appointment.user = request.user
    if sale_id:
        sale = Sale.objects.get(id=sale_id)
        sale_percent = sale.percent
        service_price = service.price
        price = int(service_price) - sale_percent / 100 * int(service_price)
        appointment.price = price
        appointment.save()
        sale.delete()
    else:
        appointment.price = service.price
        appointment.save()
    time.status = 1
    time.save()
    return HttpResponseRedirect(reverse('users:profile'))

```

Удаление записи администратором

```

@login_required
def appointments_delete(request, app_id):
    appointment = Appointment.objects.get(id=app_id)
    time = TimeTable.objects.get(id=appointment.time.id)
    time.status = 0
    time.save()
    appointment.delete()
    return HttpResponseRedirect(request.META.get('HTTP_REFERER'))

```

Добавление скидки администратором

```

def sale_add(request):
    if request.method == 'POST':
        form = SaleForm(data=request.POST, files=request.FILES)
        if form.is_valid():

```

```

        form.save()
        return HttpResponseRedirect(request.META.get('HTTP_REFERER'))
    else:
        form = SaleForm()
        sales = Sale.objects.all()
        context = {
            'title': 'Скидки',
            'sales': sales,
            'form': form
        }
        return render(request, 'appointments/sale_add.html', context)

```

appointments\urls.py

```

from django.urls import path

from appointments.views import (date_and_time, appointments_create, appointments,
appointments_delete,
                                appointment_sale, date_and_time_edit, date_and_time_delete, date_delete,
                                sale_add)

app_name = 'appointments'

urlpatterns = [
    path('date_and_time/<int:service_id>', date_and_time, name='date_and_time'),
    path('appointments_create/<int:time_id>/<int:service_id>', appointments_create,
name='appointments_create'),
    path('appointments_create_sale/<int:time_id>/<int:service_id>/<int:sale_id>',
appointments_create, name='appointments_create_sale'),
    path('appointments/<int:time_id>/<int:service_id>', appointments, name='appointments'),
    path('appointments_sale/<int:time_id>/<int:service_id>/<int:sale_id>', appointment_sale,
name='appointments_sale'),
    path('appointments_delete/<int:app_id>', appointments_delete, name='appointments_delete'),
    path('date_and_time_edit/', date_and_time_edit, name='date_and_time_edit'),
    path('date_and_time_delete/<int:time_id>', date_and_time_delete,
name='date_and_time_delete'),
    path('date_delete/<int:date_id>', date_delete, name='date_delete'),
    path('sale_add/', sale_add, name='sale_add'),
]

```

appointments\models.py

Модель для хранения и работы с датами

```

class DateTable(models.Model):
    date = models.DateField()
    def __str__(self):

```

```
return str(self.date)
```

Модель записей на сайте

```
class TimeTable(models.Model):
    FREE = 0
    OCCUPIED = 1
    STATUSES = (
        (FREE, 'Свободно'),
        (OCCUPIED, 'Занято')
    )
    date = models.ForeignKey(DateTable, on_delete=models.CASCADE)
    time_start = models.TimeField()
    time_end = models.TimeField()
    status = models.SmallIntegerField(choices=STATUSES, default=FREE)
    def __str__(self):
        return f'Дата: {self.date.date} Время: {self.time_start} - {self.time_end} | Статус: {self.status}'
```

Модель скидки

```
class Sale(models.Model):
    name = models.CharField(max_length=256)
    percent = models.IntegerField(default=0)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
```

Модель записи клиента

```
class Appointment(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    time = models.ForeignKey(TimeTable, on_delete=models.CASCADE)
    service = models.ForeignKey(Service, on_delete=models.CASCADE)
    price = models.IntegerField(default=0, blank=True, null=True)
```

appointments\forms.py

```
from django import forms
from appointments.models import DateTable, TimeTable, Sale
from users.models import User
class DateForm(forms.ModelForm):
    date = forms.DateField(widget=forms.DateInput(
        attrs={
            'class': 'input-form',
            'placeholder': 'Введите дату'
        }
    ))
    class Meta:
        model = DateTable
```

```

        fields = ('date', )
class TimeForm(forms.ModelForm):
    date = forms.ModelChoiceField(widget=forms.Select(
        attrs={
            'class': 'input-form',
            'placeholder': 'Выберите дату'
        }
    ), queryset=DataTable.objects.all())
    time_start = forms.TimeField(widget=forms.TimeInput(
        attrs={
            'class': 'input-form',
            'placeholder': 'Введите начало периода'
        }
    ))
    time_end = forms.TimeField(widget=forms.TimeInput(
        attrs={
            'class': 'input-form',
            'placeholder': 'Введите конец периода'
        }
    ))
    class Meta:
        model = TimeTable
        fields = ('date', 'time_start', 'time_end')
class SaleForm(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(
        attrs={
            'class': 'input-form',
            'placeholder': 'Введите название скидки'
        }
    ))
    percent = forms.IntegerField(widget=forms.NumberInput(
        attrs={
            'class': 'input-form',
            'placeholder': 'Введите процент скидки (в процентах)'
        }
    ))
    user = forms.ModelChoiceField(widget=forms.Select(
        attrs={
            'class': 'input-form',
            'placeholder': 'Выберите дату'
        }
    ), queryset=User.objects.filter(is_superuser=False))
    class Meta:
        model = Sale
        fields = ('name', 'percent', 'user')

```

appointments\admin.py

```
from django.contrib import admin
```

```

from appointments.models import Appointment, DateTable, TimeTable, Sale
admin.site.register(Appointment)
admin.site.register(DateTable)
admin.site.register(TimeTable)

admin.site.register(Sale)

```

4.4 Шаблоны приложения

Users

admin-panel.html

```

{% extends 'salon/base.html' %}
{% load static %}
{% block css %}
    <link rel="stylesheet" href="{% static 'css/admin-panel.css' %}">
{% endblock %}
{% block main %}
    <section class="admin-panel-container">
        <a href="{% url 'salon:service_add' %}" class="link-admin">Добавить услуги</a>
        <a href="{% url 'appointments:date_and_time_edit' %}" class="link-admin">Редактировать расписание</a>
        <a href="{% url 'appointments:sale_add' %}" class="link-admin">Добавить скидку</a>
        <a href="{% url 'admin:index' %}" class="link-admin">Панель Django</a>
    </section>
{% endblock %}

```

login.html

```

{% extends 'salon/base.html' %}
{% load static %}

{% block css %}
    <link rel="stylesheet" href="{% static 'css/login.css' %}">
{% endblock %}

{% block main %}
    <section class="login-container">
        <div class="auth-title">Авторизация</div>
        <form action="{% url 'users:login' %}" method="post" class="auth-form">
            {% csrf_token %}
            <div class="form-group">
                <label class="form-label" for="{% form.username.id_for_label %}">Имя
пользователя</label>

```

```

        {{ form.username }}
    </div>
    <div class="form-group">
        <label class="form-label" for="{{ form.password.id_for_label }}">Пароль</label>
        {{ form.password }}
    </div>
    <div class="form-nav">
        <input class="auth-button" type="submit" value="Авторизоваться">
    </div>
</form>
<div class="registration-container">
    <a href="{% url 'users:registration' %}" class="reg-button">Нужен аккаунт?
Зарегистрируйся!</a>
</div>
</section>
{% if messages or form.non_field_errors %}
    <div class="{% if messages %}container-successful{% else %}container-errors{% endif
%}">
        {% if messages %}
            {% for message in messages %}
                {{ message }}
            {% endfor %}
        {% else %}
            {{ form.non_field_errors }}
        {% endif %}
    </div>
{% endif %}

{% endblock %}

```

profile.html

```

{% extends 'salon/base.html' %}
{% load static %}
{% block css %}
    <link rel="stylesheet" href="{% static 'css/profile.css' %}">
{% endblock %}
{% block main %}
    <section class="profile-container">
        <form action="{% url 'users:profile' %}" method="post" class="profile-form"
enctype="multipart/form-data">
            {% csrf_token %}
            <div class="profile-avatar-container">
                <img src="{% if user.image %}
                    {{ user.image.url }}
                {% else %}
                    {% static 'img/avatar.jpg' %}

```



```

        { % endif % }" alt="avatar" class="profile-avatar">
        <div class="form-change-image">
            { { form.image } }
            <label class="custom-file-label" for="{ { form.image.id_for_label } }">Изменить
изображение</label>
        </div>
    </div>
    <div class="profile-info">
        <div class="form-group">
            <label class="form-label" for="{ { form.first_name.id_for_label } }">Имя</label>
            { { form.first_name } }
        </div>
        <div class="form-group">
            <label class="form-label" for="{ { form.last_name.id_for_label
} }">Фамилия</label>
            { { form.last_name } }
        </div>
        <div class="form-group">
            <label class="form-label" for="{ { form.phone.id_for_label } }">Номер
телефона</label>
            { { form.phone } }
        </div>
        <div class="form-group">
            <label class="form-label" for="{ { form.email.id_for_label } }">Адрес электронной
почты</label>
            { { form.email } }
        </div>
        <div class="form-group">
            <label class="form-label" for="{ { form.username.id_for_label } }">Имя
пользователя</label>
            { { form.username } }
        </div>
        <div class="form-save">
            <input class="form-save-btn" type="submit" value="Сохранить">
        </div>
    </div>
</form>
</section>
<section class="sale-container">
    <h2 class="sale-title">Мои скидки</h2>
    { % if sales % }
        { % for sale in sales % }
            <div class="sale-item">
                <div class="sale-name">{ { sale.name } }</div>
                <div class="sale-percent">Скидка: { { sale.percent } } %</div>
                <a href="{ % url 'service' % }" class="link-sale">Применит скидку</a>
            </div>
        { % endfor % }
    { % else % }

```

```

    <p class="message">У вас нет доступных скидок</p>
  {% endif %}
</section>
<section class="appointments-container">
  <h2 class="appointments-title">Мои записи</h2>
  {% if appointment %}
    {% for app in appointment %}
      <div class="appointment-item">
        <h3 class="appointment-name">{{ app.service.name }}</h3>
        <p class="appointment-date">Дата: {{ app.time.date.date }}: {{ app.time.time_start
}} - {{ app.time.time_end }}</p>
        <p class="appointment-price">Цена: {{ app.price }} руб.</p>
        <a href="{% url 'appointments:appointments_delete' app.id %}" class="link-
reverse">Отменить запись</a>
      </div>
    {% endfor %}
  {% else %}
    <p class="message">У вас еще нет записей</p>
  {% endif %}
</section>
{% endblock %}

```

registration.html

```

{% extends 'salon/base.html' %}
{% load static %}

{% block css %}
  <link rel="stylesheet" href="{% static 'css/registration.css' %}">
{% endblock %}

{% block main %}
  <section class="reg-container">
    <div class="reg-title">Регистрация</div>
    <form action="{% url 'users:registration' %}" method="post" class="reg-form">
      {% csrf_token %}
      <div class="form-group">
        <label class="form-label" for="{{ form.first_name.id_for_label }}">Имя</label>
        {{ form.first_name }}
      </div>
      <div class="form-group">
        <label class="form-label" for="{{ form.last_name.id_for_label }}">Фамилия</label>
        {{ form.last_name }}
      </div>
      <div class="form-group">
        <label class="form-label" for="{{ form.username.id_for_label }}">Имя
пользователя</label>

```

```

        {{ form.username }}
    </div>
    <div class="form-group">
        <label class="form-label" for="{{ form.email.id_for_label }}">Адрес электронной
почты</label>
        {{ form.email }}
    </div>
    <div class="form-group">
        <label class="form-label" for="{{ form.phone.id_for_label }}">Номер
телефона</label>
        {{ form.phone }}
    </div>
    <div class="form-group">
        <label class="form-label" for="{{ form.password1.id_for_label }}">Пароль</label>
        {{ form.password1 }}
    </div>
    <div class="form-group">
        <label class="form-label" for="{{ form.password2.id_for_label }}">Подтверждение
пароля</label>
        {{ form.password2 }}
    </div>
    <div class="form-nav">
        <input class="reg-button" type="submit" value="Создать аккаунт">
    </div>
</form>
<div class="registration-container">
    <a href="{% url 'users:login' %}" class="auth-button">Уже есть аккаунт?
Авторизоваться</a>
</div>
</section>
{% if form.errors %}
<div class="container-errors">
    {{ form.errors }}
</div>
{% endif %}
{% endblock %}

```

Salon

base.html

```

{% load static %}

<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"

```

```

        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="{ % static 'css/reset.css' % }">
    <link rel="stylesheet" href="{ % static 'css/base.css' % }">
    { % block css % } { % endblock % }
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,7
00;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
    <title>{{ title }}</title>
</head>
<body>
    <header>
        <nav>
            <div class="logo-container">
                <a href="{ % url 'index' % }" class="logo">INNAN</a>
            </div>
            <div class="nav-list">
                <a href="{ % url 'index' % }" class="nav-link">Главная</a>
                <a href="{ % url 'service' % }" class="nav-link">Услуги</a>
                { % if user.is_authenticated % }
                { % if user.is_superuser or user.is_staff % }
                <a href="{ % url 'users:admin_panel' % }" class="nav-link">Админ-панель</a>
                <a href="{ % url 'users:logout' % }" class="nav-link">Выход</a>
                { % else % }
                <a href="{ % url 'users:profile' % }" class="nav-link">Профиль</a>
                <a href="{ % url 'users:logout' % }" class="nav-link">Выход</a>
                { % endif % }
                { % else % }
                <a href="{ % url 'users:login' % }" class="nav-link">Вход</a>
                { % endif % }
            </div>
        </nav>
    </header>
    <main>
        { % block main % }

        { % endblock % }
    </main>
</body>
</html>

```

index.html

```

{ % extends 'salon/base.html' % }
{ % load static % }

```

```
{% block css %}
    <link rel="stylesheet" href="{% static 'css/index.css' %}" %}>
{% endblock %}

{% block main %}
    <section class="desc-container">
        <div class="desc-title-container">
            <h1 class="desc-title">INNAN — новая студия<br>красоты в сердце Казани</h1>
            <a href="{% url 'service' %}" class="desc-link">К выбору услуг</a>
        </div>
        <div class="desc-img-container">
            
        </div>
    </section>
{% endblock %}
```

service.html

```
{% extends 'salon/base.html' %}
{% load static %}

{% block css %}
    <link rel="stylesheet" href="{% static 'css/service.css' %}" %}>
{% endblock %}

{% block main %}
    <section class="services-container">
        <div class="category-container">
            <div class="category-list">
                {% for cat in category %}
                    <div class="category-item">
                        <a href="{% url 'category' cat.id %}" class="category-link">{{ cat.name }}</a>
                    </div>
                {% endfor %}
            <div class="category-item">
                <a href="{% url 'service' %}" class="category-link">Все услуги</a>
            </div>
        </div>
        <div class="service-container">
            <div class="service-list">
                {% for service in services %}
                    <div class="service-item">
                        <h2 class="service-name">{{ service.name }}</h2>
                        <span class="service-price">от {{ service.price }} руб.</span>
                    <div class="link-service-container">
```

```

        { % if user.is_superuser or user.is_staff % }
        <a href="{ % url 'salon:service_delete' service.id % }" class="link-
service">Удалить</a>
        { % else % }
        <a href="{ % url 'appointments:date_and_time' service.id % }" class="link-
service">Записаться</a>
        { % endif % }
    </div>
</div>
{ % endfor % }
</div>
</div>
</section>
{ % endblock % }

```

service-add.html

```

{ % extends 'salon/base.html' % }
{ % load static % }

{ % block css % }
    <link rel="stylesheet" href="{ % static 'css/service-add.css' % }">
{ % endblock % }

{ % block main % }
    <section class="service-add-container">
        <form action="{ % url 'salon:service_add' % }" method="post" class="service-form">
            { % csrf_token % }
            <div class="service-add">
                <div class="form-group">
                    <label class="form-label" for="{ { form.name.id_for_label } }">Наименование
услуги</label>
                    { { form.name } }
                </div>
                <div class="form-group">
                    <label class="form-label" for="{ { form.category.id_for_label } }">Категория
услуги</label>
                    { { form.category } }
                </div>
                <div class="form-group">
                    <label class="form-label" for="{ { form.price.id_for_label } }">Цена за
услугу</label>
                    { { form.price } }
                </div>
            </div>
            <div class="form-save">
                <input class="form-save-btn" type="submit" value="Добавить услугу">
            </div>
        </form>
    </section>
{ % endblock % }

```

```

    </div>
  </form>
</section>
{% endblock %}

```

Appointments

appointment.html

```

{% extends 'salon/base.html' %}
{% load static %}
{% block css %}
  <link rel="stylesheet" href="{% static 'css/appointment.css' %}">
{% endblock %}
{% block main %}
  <section class="appointment-create-container">
    <h1 class="appointment-name">Услуга: {{ service.name }}</h1>
    <p class="appointment-date">Дата записи: {{ time.date.date }} : {{ time.time_start }} - {{
time.time_end }}</p>
    <p class="appointment-price">Цена: {{ service.price }}</p>
    <a href="{% url 'appointments:appointments_create' time.id service.id %}"
class="appointment-create-link">Записаться</a>
  </section>
  {% if sales %}
    <div class="sale-container">
      <div class="sale-title">Доступные скидки:</div>
      {% for sale in sales %}
        <div class="sale-item">
          <div class="sale-name">{{ sale.name }}</div>
          <div class="sale-percent">Скидка: {{ sale.percent }} %</div>
          <a href="{% url 'appointments:appointments_sale' time.id service.id sale.id %}"
class="link-sale">Применит скидку</a>
        </div>
      {% endfor %}
    </div>
  {% endif %}
{% endblock %}

```

appointments_sale.html

```

{% extends 'salon/base.html' %}
{% load static %}
{% block css %}
  <link rel="stylesheet" href="{% static 'css/appointment.css' %}">
{% endblock %}
{% block main %}
  <section class="appointment-create-container">

```

```

    <h1 class="appointment-name">Услуга: {{ service.name }}</h1>
    <p class="appointment-date">Дата записи: {{ time.date.date }} : {{ time.time_start }} - {{
time.time_end }}</p>
    <p class="appointment-price">Цена (с учетом скидки): {{ price }}</p>
    <a href="{% url 'appointments:appointments_create_sale' time.id service.id sale.id %}"
class="appointment-create-link">Записаться</a>
</section>

{% endblock %}

```

date-and-time.html

```

{% extends 'salon/base.html' %}
{% load static %}
{% block css %}
    <link rel="stylesheet" href="{% static 'css/date-and-time.css' %}">
{% endblock %}
{% block main %}
    <section class="timetable-container">
        <h1 class="timetable-title">Выберите свободный промежуток времени</h1>
        <div class="timetable-list">
            {% if time %}
                {% for date in date %}
                    <div class="timetable-item">
                        <h2 class="timetable-date">Дата: {{ date }}</h2>
                        <div class="timetable-time-list">
                            <p class="timetable-time-title">Доступное время:</p>
                            {% for time in time %}
                                {% if time.date == date %}
                                    {% if time.status == 0 %}
                                        <div class="timetable-time-item">
                                            <span class="timetable-time">{{ time.time_start }} - {{ time.time_end
}}</span>
                                            <a href="{% url 'appointments:appointments' time.id service.id %}"
class="advertisement-link">Записаться</a>
                                        </div>
                                    {% endif %}
                                {% endif %}
                            {% endfor %}
                        </div>
                    </div>
                {% else %}
                    <span>Нет доступного времени для записи</span>
                {% endif %}
            {% endif %}
        </div>
    </section>

```



```
{% endblock %}
```

date-and-time-edit.html

```
{% extends 'salon/base.html' %}
```

```
{% load static %}
```

```
{% block css %}
```

```
<link rel="stylesheet" href="{% static 'css/date-and-time.css' %}">
```

```
{% endblock %}
```

```
{% block main %}
```

```
<section class="timetable-container">
```

```
<h1 class="timetable-title">Доступные промежутки времени</h1>
```

```
<div class="timetable-list">
```

```
{% if time %}
```

```
{% for date in date %}
```

```
<div class="timetable-item">
```

```
<h2 class="timetable-date">Дата: {{ date }}</h2>
```

```
<div class="timetable-time-list">
```

```
<p class="timetable-time-title">Доступное время:</p>
```

```
{% for time in time %}
```

```
{% if time.date == date %}
```

```
{% if time.status == 0 %}
```

```
<div class="timetable-time-item">
```

```
<span class="timetable-time">{{ time.time_start }} - {{ time.time_end
```

```
}}</span>
```

```
<a href="{% url 'appointments:date_and_time_delete' time.id %}"
```

```
class="link-delete">Удалить</a>
```

```
</div>
```

```
{% endif %}
```

```
{% endif %}
```

```
{% endfor %}
```

```
</div>
```

```
<a href="{% url 'appointments:date_delete' date.id %}" class="link-
```

```
delete">Удалить дату</a>
```

```
</div>
```

```
{% endfor %}
```

```
{% else %}
```

```
<span>Нет доступного времени для записи</span>
```

```
{% endif %}
```

```
</div>
```

```
</section>
```

```
<section class="date-and-time-add">
```

```
<div class="date-add-container">
```

```
<h2 class="date-add-title">Добавить новую дату в расписание</h2>
```

```
<form action="{% url 'appointments:date_and_time_edit' %}" method="post">
```

```
{% csrf_token %}
```

```
<div class="form-block">
```

```

        <label for="{{ form_date.date.id_for_label }}" class="label-form">Выберите дату
(дата.месяц.год)</label>
        {{ form_date.date }}
    </div>
    <div class="form-block">
        <button type="submit" class="btn-form">Добавить</button>
    </div>
</form>
</div>
<div class="time-add-container">
    <h2 class="time-add-title">Добавить новое время в расписание</h2>
    <form action="{% url 'appointments:date_and_time_edit' %}" method="post">
        {% csrf_token %}
        <div class="form-block">
            <label for="{{ form_time.date.id_for_label }}" class="label-form">Выберите
дату</label>
            {{ form_time.date }}
        </div>
        <div class="form-block">
            <label for="{{ form_time.time_start.id_for_label }}" class="label-form">Выберите
начало</label>
            {{ form_time.time_start }}
        </div>
        <div class="form-block">
            <label for="{{ form_time.time_end.id_for_label }}" class="label-form">Выберите
конец</label>
            {{ form_time.time_end }}
        </div>
        <div class="form-block">
            <button type="submit" class="btn-form">Добавить</button>
        </div>
    </form>
</div>
</section>
{% endblock %}

```

sale_add.html

```

{% extends 'salon/base.html' %}
{% load static %}

{% block css %}
    <link rel="stylesheet" href="{% static 'css/sale_add.css' %}">
{% endblock %}

{% block main %}
    <section class="sale-container">

```

```

<h1 class="sale-title">Все скидки в системе</h1>
<div class="sale-list">
  {% if sales %}
    {% for sale in sales %}
      <div class="sale-item">
        <p class="sale-name">{{ sale.name }}</p>
        <p class="sale-percent">Скидка: {{ sale.percent }} %</p>
        <p class="sale-user">Доступна для пользователя: {{ sale.user.first_name }} {{
sale.user.second_name }}</p>
        <a href="" class="link-delete">Удалить</a>
      </div>
    {% endfor %}
  {% else %}
    <p class="message">В системе нет скидок</p>
  {% endif %}
</div>
</section>
<section class="sale-add-container">
  <form action="{% url 'appointments:sale_add' %}" method="post" class="form">
    <h2 class="form-title">Добавить скидку</h2>
    {% csrf_token %}
    <div class="form-block">
      <label for="{{ form.name.id_for_label }}" class="label-form">Название
скидочной акции</label>
      {{ form.name }}
    </div>
    <div class="form-block">
      <label for="{{ form.percent.id_for_label }}" class="label-form">Процент
скидки</label>
      {{ form.percent }}
    </div>
    <div class="form-block">
      <label for="{{ form.user.id_for_label }}" class="label-form">Получатель
скидки</label>
      {{ form.user }}
    </div>
    <div class="form-block-btn">
      <button type="submit" class="btn-form">Добавить</button>
    </div>
  </form>
</section>
{% endblock %}

```

5. Визуальное представление

5.1 Сайт

The screenshot displays the INNAN website's authorization page. The header includes the INNAN logo on the left and navigation links for 'Главная', 'Услуги', and 'Вход' on the right. The main content area is titled 'Авторизация' and contains a form with two input fields: 'Имя пользователя' (Username) and 'Пароль' (Password). Below these fields is an 'Авторизоваться' (Log In) button. At the bottom of the form, there is a link that says 'Нужен аккаунт? Зарегистрируйся!' (Need an account? Register!).

Рис. 1 Авторизация

The screenshot displays the INNAN website's registration page. The header is identical to the previous page. The main content area is titled 'Регистрация' and contains a form with six input fields: 'Имя' (Name), 'Фамилия' (Surname), 'Имя пользователя' (Username), 'Адрес электронной почты' (Email address), 'Номер телефона' (Phone number), and 'Пароль' (Password). Each field has a corresponding label and a 'Введите...' (Enter...) placeholder text.

Рис. 2 Регистрация

The screenshot displays the INNAN website's successful registration confirmation page. The header is identical to the previous pages. The main content area is titled 'Авторизация' and contains the same form as in the previous screenshots. However, the input fields for 'Имя пользователя' and 'Пароль' now contain the values 'kiza' and '*****' respectively. Below the form, there is a green banner with the text 'Регистрация прошла успешно!' (Registration was successful!).

Рис. 3 Успешная регистрация

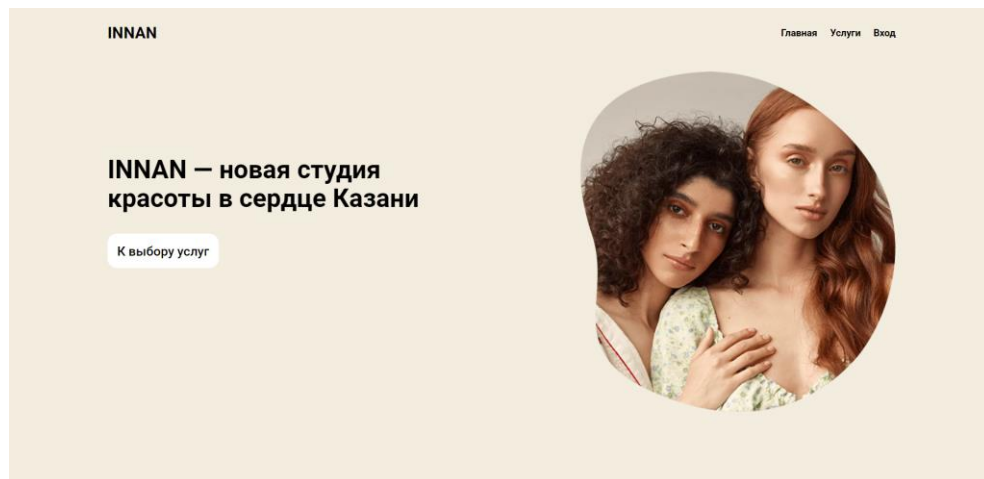


Рис. 4 Главная страница

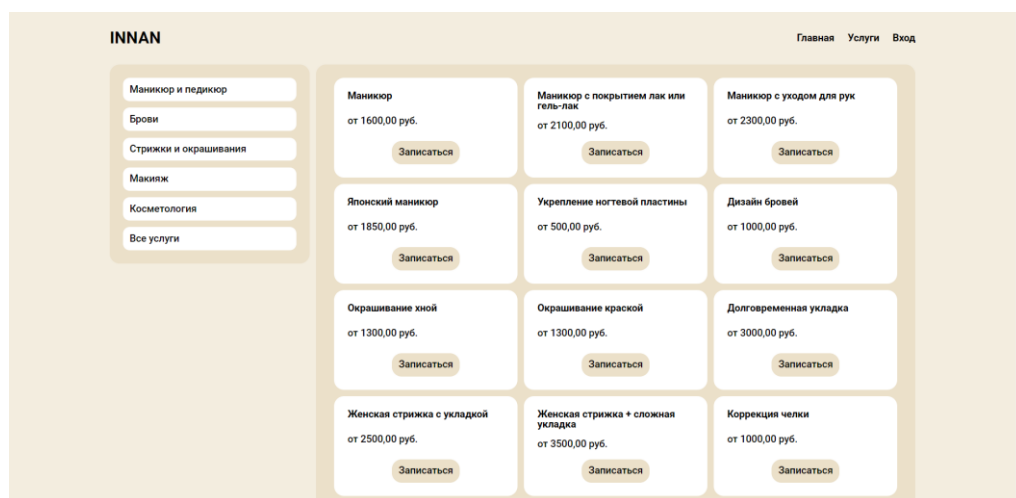


Рис. 5 Услуги

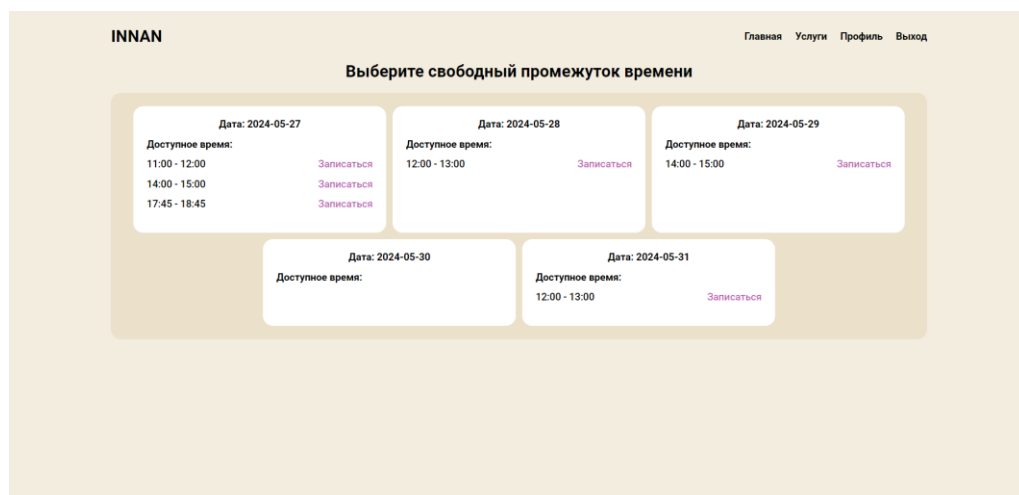


Рис. 6 Время записи на услугу

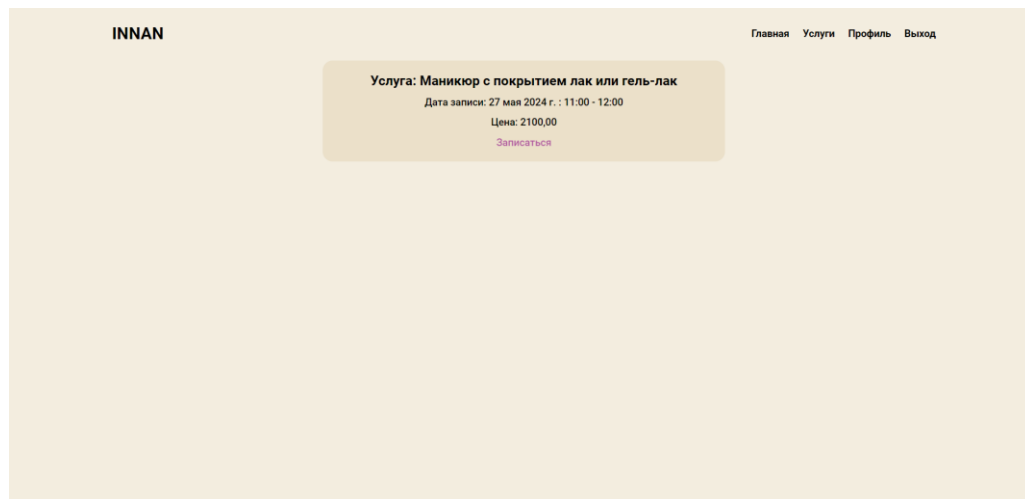


Рис. 7 Выбор записи

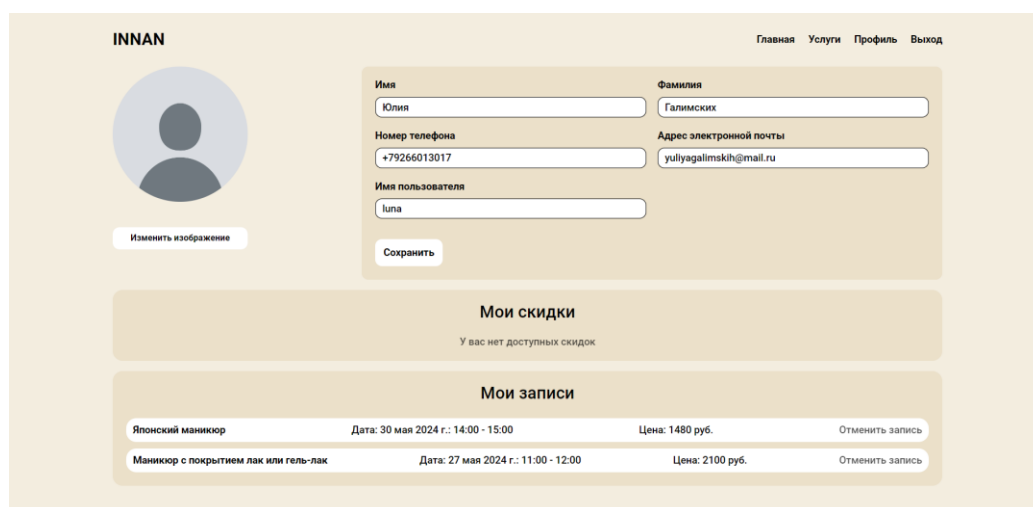


Рис. 8 Профиль клиента с историей записей

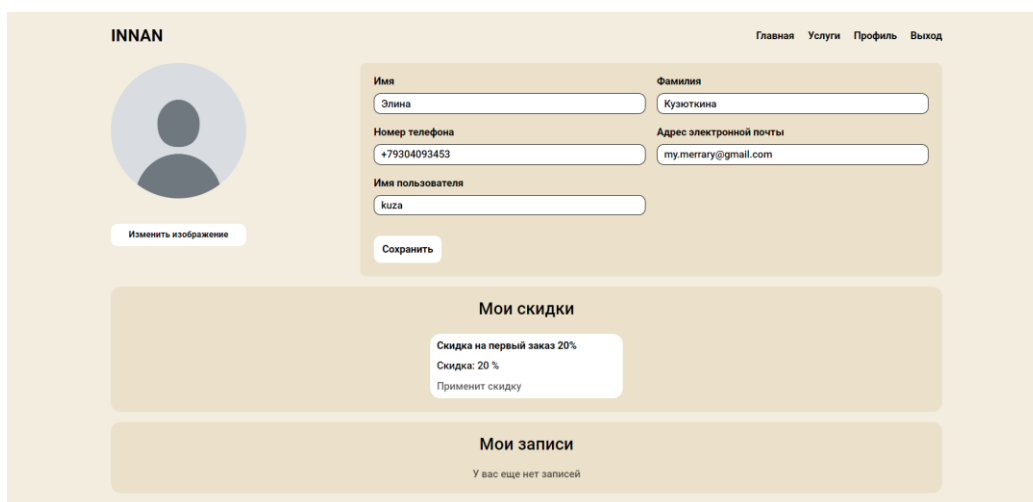


Рис.9 Отображение скидки нового пользователя

ИННАН

Главная Услуги Профиль Выход

Имя: Ирина

Фамилия: Кузоткина

Номер телефона: +79304093453

Адрес электронной почты: my.merrary@gmail.com

Имя пользователя: kiza

Изменить изображение

Сохранить

Мои скидки

Скидка на первый заказ 20%

Скидка: 20 %

Применит скидку

Мои записи

У вас еще нет записей

Рис.10 Изменение фото профиля

ИННАН

Главная Услуги Профиль Выход

Услуга: Окрашивание краской

Дата записи: 28 мая 2024 г. : 12:00 - 13:00

Цена: 1300,00

Записаться

Доступные скидки:

Скидка на первый заказ 20%

Скидка: 20 %

Применит скидку

Рис. 11 Возможность применения скидки

ИННАН

Главная Услуги Админ-панель Выход

Добавить услугу

Редактировать расписание

Добавить скидку

Панель Django

Рис. 12 Админ-панель

ИННАН

Главная Услуги Админ-панель Выход

Наименование услуги

Категория услуги

Цена за услугу

Добавить услугу

Рис. 13 Добавление услуги

ИННАН

Главная Услуги Админ-панель Выход

Доступные промежутки времени

Дата: 2024-05-27
Доступное время:
14:00 - 15:00
17:45 - 18:45
Удалить дату

Дата: 2024-05-28
Доступное время:
12:00 - 13:00
Удалить дату

Дата: 2024-05-29
Доступное время:
14:00 - 15:00
Удалить дату

Дата: 2024-05-30
Доступное время:
Удалить дату

Дата: 2024-05-31
Доступное время:
12:00 - 13:00
Удалить дату

Добавить новую дату в расписание
Выберите дату (дата.месяц.год)

Добавить

Добавить новое время в расписание
Выберите дату

Выберите начало

Рис. 14 Редактор расписания

ИННАН

Главная Услуги Админ-панель Выход

Все скидки в системе

Скидка на первый заказ 20%
Скидка: 20 %
Доступна для пользователя: Тимур
Удалить

Скидка на первый заказ 20%
Скидка: 20 %
Доступна для пользователя: Злина
Удалить

1
Скидка: 13 %
Доступна для пользователя: Ксения
Удалить

Добавить скидку
Название скидочной акции

Процент скидки

Получатель скидки

Добавить

Рис. 15 Добавить скидку

5.2 База данных

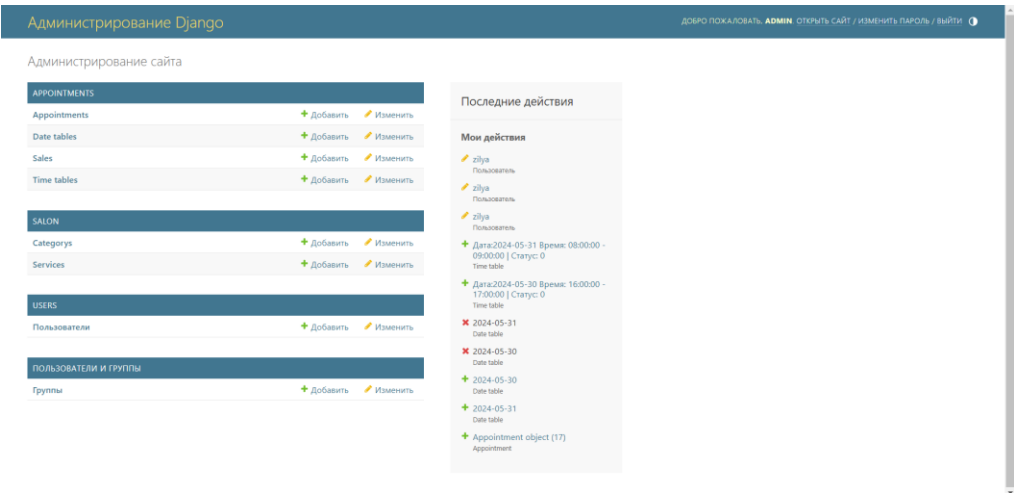


Рис 16. Панель администратора

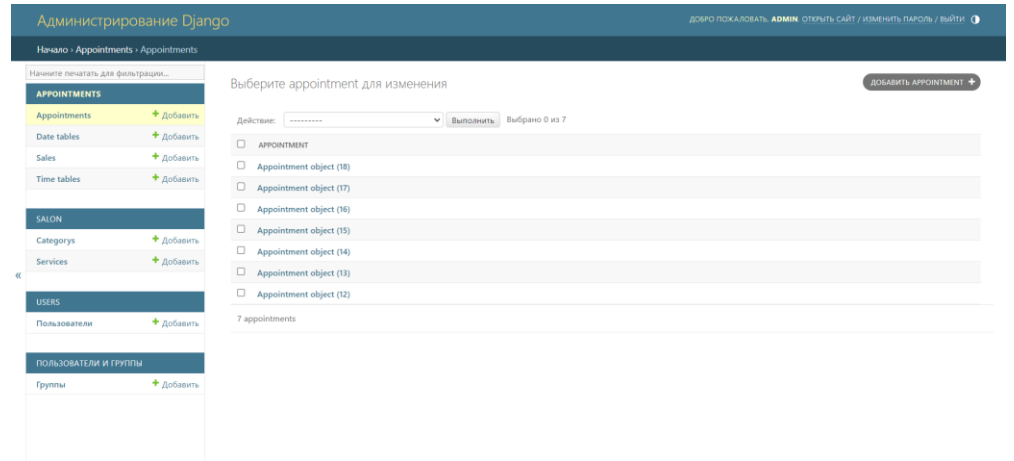


Рис. 17 Appointments

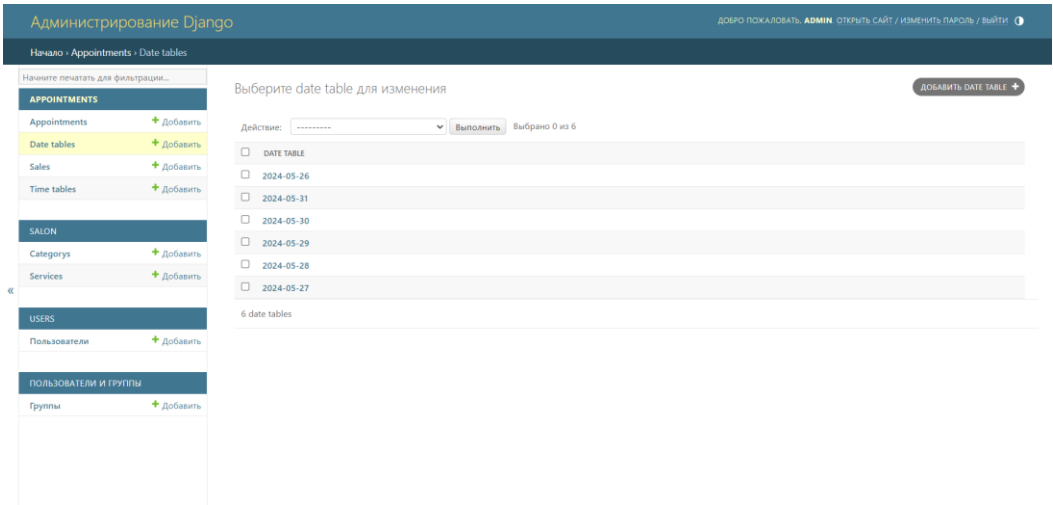


Рис. 18 Date tables

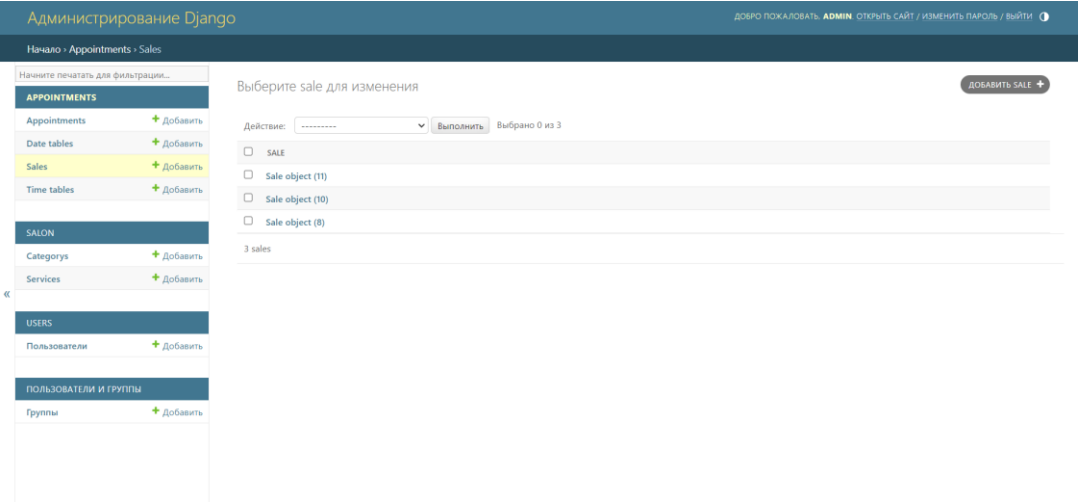


Рис. 19 Sales

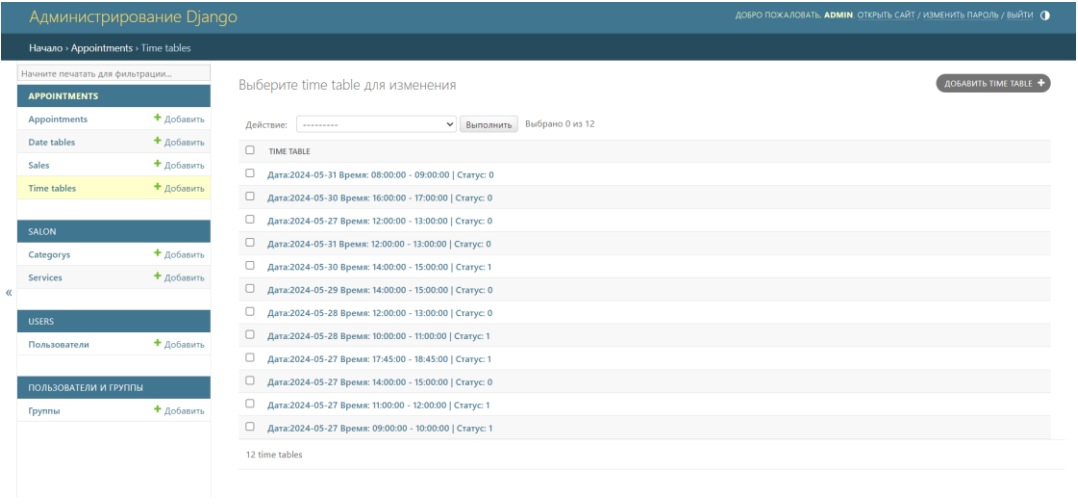


Рис. 20 Time tables

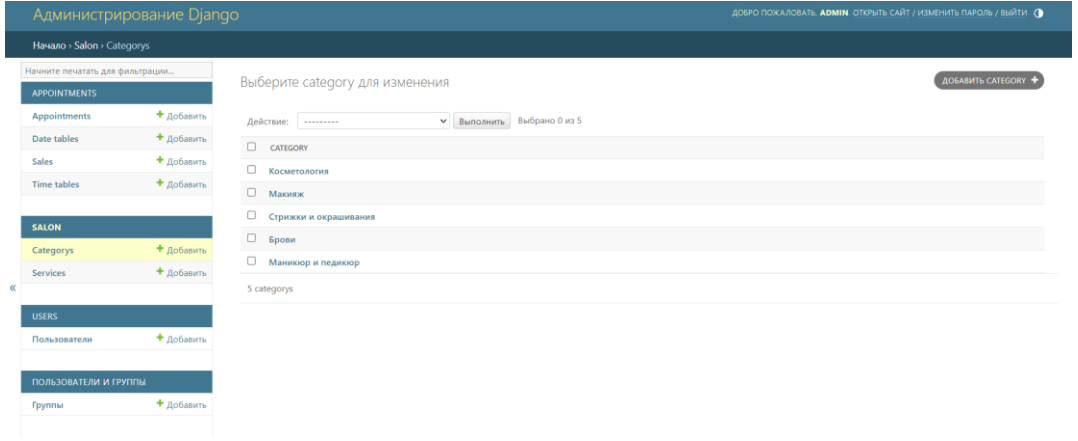


Рис. 21 Categories

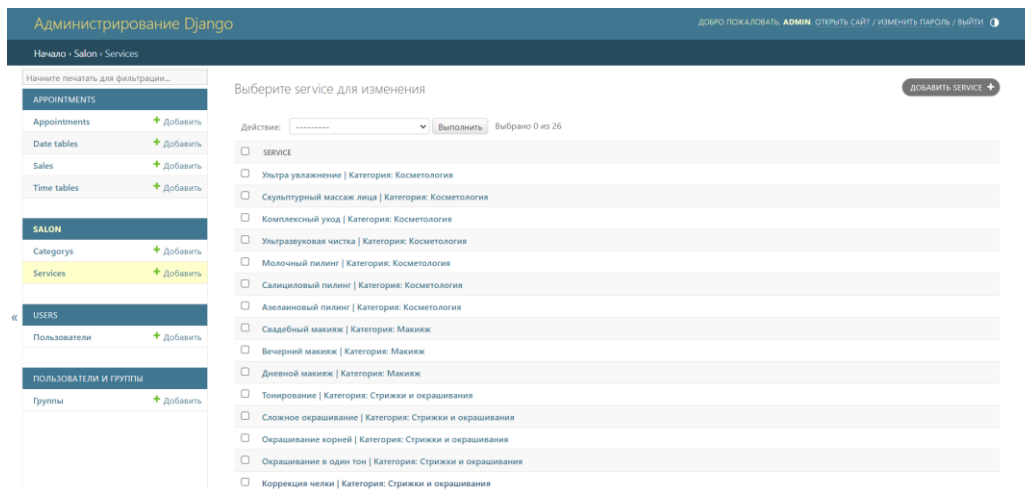


Рис. 22 Services

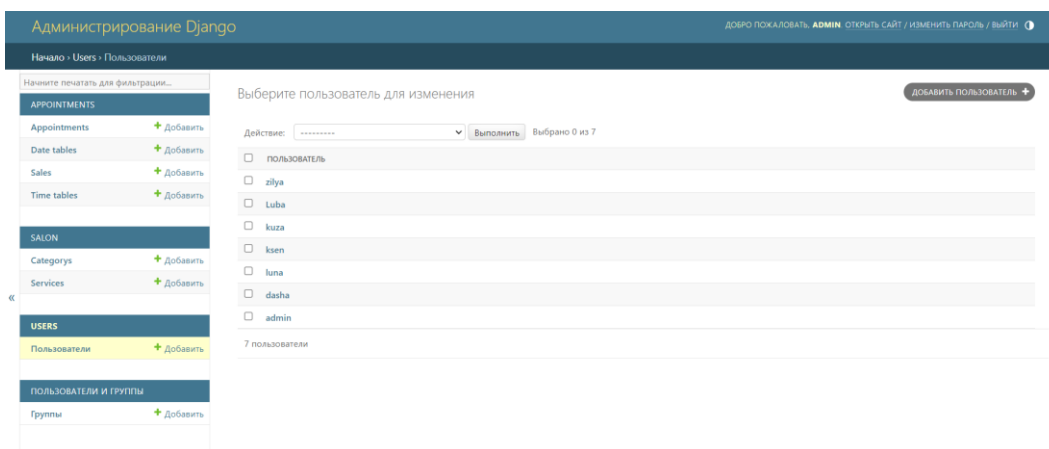


Рис. 23 Пользователи