

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет среднего профессионального образования

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на учебную практику
по ПМ.11 «Разработка, администрирование и защита баз данных»

Студент Дмитриев Н. К. Группа № Y2335
(Фамилия И. О.)

Руководитель Ефимова Т.Н., преподаватель факультета СПО
Говоров А.И., преподаватель факультета СПО

Тема задания: Проектирование и реализация базы данных.

Сроки прохождения практики: 02.02.2020 -02.07.2020

Место прохождения практики: Факультет СПО

1. Виды работ и требования к их выполнению:

Учебная практика проводится распределенно (понедельно в течение семестра) на базе факультета СПО в лаборатории разработки баз данных. В ходе прохождения практики выполняются следующие виды работ:

- I. Вводный инструктаж по технике безопасности и общим целям, и задачам практики.
- II. Анализ поставленной задачи
- III. Выполнение индивидуального задания: проектирование БД, разработка прототипа веб-приложения.
- IV. Формирование отчета по учебной практике.

2. Виды отчетных материалов и требования к их оформлению:

По результатам прохождения практики составляется отчет, в котором представляются индивидуальное задание, модель базы данных, перечень использованных технологий, программных средств, использованных паттернов (шаблонов) проектирования программ, программный код, описание результатов работы программы. Оформление отчета должно соответствовать Рекомендациям по оформлению технических документов факультета СПО Университета ИТМО.

3. ПЛАН-ГРАФИК

№ эта па	Наименование этапа	Срок завершения этапа	Виды работ	Форма отчетности
1	2	3	4	5
1.	Вводный инструктаж	02.02.2020 – 09.02.2020	Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами производственной практики	Журнал по технике безопасности
2.	Постановка задачи	09.02.2020 – 09.03.2020	Анализ индивидуального задания. Обследование предметной области согласно индивидуальной теме учебной практики.	Отчет по практике: индивидуальное задание Дневник практики
3.	Моделирование базы данных и реализация	10.03.2020 – 31.03.2020	Описание предметной области. Создание диаграммы классов Создание таблиц Заполнение таблиц данными (команды)	Отчет по практике: индивидуальное задание Дневник практики
4.	Реализация модели данных средствами Django ORM	01.04.2020 – 07.04.2020	Создание модели Django в соответствии с моделью данных и настройка связи между таблицами	Отчет по практике: индивидуальное задание Дневник практики
5.		82.04.2020 – 21.04.2020	Реализация элементов инфраструктуры Django, в соответствии с архитектурным паттерном Model-View-Template или сокращенно MVT. Реализация интерфейсов к системе средствами Django Templates или сторонними средствами.	Отчет по практике: индивидуальное задание Дневник практики
6.	Подготовка отчетных материалов	17.06.2020 – 23.06.2020	Формирование отчета о практике	Отчет по практике: индивидуальное задание Дневник практики
7.	Защита результатов практики	24.06.2020 – 02.07.2020	Защита результатов практики в форме устного собеседования и представления результатов практики	

Задание утверждено председателем выпускающей комиссии факультета СПО
 Председатель выпускающей комиссии факультета СПО _____ Королев В.В.
 «___» _____ 20__ г

Дата выдачи задания: _____

Руководитель от факультета _____
 (подпись руководителя)

Задание принял к
 исполнению _____
 (подпись студента)

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет среднего профессионального образования

Направление подготовки (специальность) 09.02.07 Информационные системы и программирование

О Т Ч Е Т

об учебной практике по профессиональному модулю
ПМ.11 «Разработка, администрирование и защита баз данных»

Тема задания: Разработка и реализация базы данных по предметной области

Обучающийся Дмитриев Н.К. Группа Y2335
(Фамилия И.О.) (номер группы)

Руководитель практики: Ефимова Т.Н., преподаватель факультета СПО
Говоров А.И., преподаватель факультета СПО

Ответственный за практику от университета: Королев В.В.зам. директора факультета СПО

Практика пройдена с оценкой _____

Подписи членов комиссии _____ (_____) (подпись)

_____ (А.И.Говоров) (подпись)

_____ (Т.Н.Ефимова) (подпись)

Дата _____

Санкт-Петербург
2020

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	4
ВВЕДЕНИЕ.....	5
1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ	6
1.1 Формулировка поставленной задачи.....	6
1.2 Описание предметной области	6
2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.....	7
3 ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА	8
2.1 Используемые технологии.....	8
2.2 Программные средства	8
4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	9
ЗАКЛЮЧЕНИЕ	13
СПИСОК ЛИТЕРАТУРЫ.....	14
ПРИЛОЖЕНИЕ А	15
ПРИЛОЖЕНИЕ Б.....	20

ВВЕДЕНИЕ

Целью учебной практики по профессиональному модулю ПМ.11 «Разработка, администрирование и защита баз данных» является углубление знаний и практических умений и получение начального практического опыта по основным видам деятельности «Разработка, администрирование и защита баз данных» и овладение соответствующими общими и профессиональными компетенциями: ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9, ОК 10, ОК

11, ПК 11.1., ПК 11.2., ПК 11.3, ПК 11.4., ПК 11.5., ПК 11.6. (см. рабочая программа и фонд оценочных средств по производственной практике).

Учебная практика проводится на базе факультета СПО Университета ИТМО.

Результатом практики является разработка прототипа веб-приложения по заданной предметной области, использующего реляционную базу данных.

Задачи:

1. Спроектировать базу данных.
2. Описать модель данных приложения.
3. Описать методы получения, вставки, редактирования и удаления данных.
4. Описать внешнюю оболочку приложения.
5. Упаковать приложение в Docker.

1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

1.1 Формулировка поставленной задачи

Разработать веб-приложение на Django в соответствии с вариантом.

1.2 Описание предметной области

БД “Фильмотека” Описание предметной области: Фильмотека занимается продажей кассет различных жанров. Покупатели приобретают кассеты на интернет ресурсе с помощью специальных форм. Когда в магазине заканчиваются кассеты, есть возможность дополнить товар.

2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

В процессе проектирования базы данных была использована методология «сущность-связь» и соответствующая ей нотация.

При анализе предметной области были выделены следующие сущности, обозначенные в инфологической модели на рисунке А.1 в приложении А:

- кассеты;
- продавцы;
- поставки;
- заказы;
- заказчики;
- кассеты в заказе.

Логическая модель соответствует первой нормальной форме, так как все атрибуты атомарные. Описание логической модели в приложении А в таблицах А.1-А.6.

3 ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА

2.1 Используемые технологии

При разработке системы были использованы следующие технологии:

- SQLite;
- Django.

2.2 Программные средства

При проектировании базы данных был использован MySQL Workbench 8.0 CE – унифицированный визуальный инструмент для разработки и администрирования баз данных.

Для реализации системы были использованы следующие программные средства:

- JetBrains PyCharm 2020.1.2 x64 – IDE для профессиональной разработки на Python;
- Docker – открытая платформа для разработки, доставки и запуска приложений.

4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Для реализации системы была описана модель данных на языке Python с использованием фреймворка Django, представленная в листинге Б.1 приложения Б.

После чего были разработаны пользовательские интерфейсы.

Программный код представлен в приложении Б.

Главная страница представлена на рис. 1, на странице представлен каталог товаров.



Рисунок 1 – Главная страница

Страница «корзина» представлена на рис. 2.



Рисунок 2 – «Корзина» покупателя

Страница «заказы» представляет собой список всех оформленных заказов покупателя с указанием купленных кассет, их количества, цену во время покупки за одну штуку, а также суммарное количество кассет и итоговую стоимость заказов. Страница представлена на рис. 3.

Каталог Корзина Мои заказы Выйти	
Мои заказы	
Дата заказа: 12-09-2020 21:21:12	
Название фильма: rjgjr Цена: 83 Добавлено: 1 Общая цена: 83 Общее количество: 1	
Дата заказа: 12-09-2020 21:21:12	
Название фильма: wewg Цена: 77 Добавлено: 23 Название фильма: rjgjr Цена: 83 Добавлено: 16 Общая цена: 3099 Общее количество: 39	
Дата заказа: 12-09-2020 21:21:08	
Название фильма: ddfd Цена: 99 Добавлено: 2 Название фильма: wewg Цена: 77 Добавлено: 3 Название фильма: fjfjfnrjf Цена: 88 Добавлено: 4	

Рисунок 4 – Страница «Заказы»

Если пользователь не авторизован на сайте, то ему будет предложено пройти регистрацию, либо произвести авторизацию в уже существующий аккаунт. Пользователь без аккаунта на сайте не может делать заказы. Страницы «каталог» неавторизованного пользователя, «регистрация», «авторизация» представлены на рис. 5, 6, 7.

Для приобретения товара требуется авторизация: Авторизоваться	
Каталог	
Название фильма: ddfd Жанр: dfd Продюсер: ttg Год выхода: 1996 Студия: too... Доступно: 1 Цена: 99	
Название фильма: wewg Жанр: bytvc Продюсер: l_mdn Год выхода: 1986 Студия: eee Доступно: 5 Цена: 77	
Название фильма: fjfjfnrjf Жанр: rfnrnfecow Продюсер: iwmfiw Год выхода: 1983 Студия: pfff Доступно: 4 Цена: 88	

Рисунок 6 – Страница «Каталог» неавторизованного пользователя

Username: Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: Enter the same password as before, for verification.

Рисунок 7 – Страница «Регистрация»

Username:

Password:

Рисунок 8 – Страница «Авторизация»

Также на сайте предусмотрена возможность добавления товаров на склад, просмотр истории пополнений. Страницы «пополнение товара», «история пополнений» представлены на рис. 9, 10.

Пополнение товара

Название кассеты:

1 шт.

Название	Количество на складе
dfdf	1
wewg	5
fjfnrjfnrif	4
rjrgjfr	0

Рисунок 9 – Страница «Пополнение товара»

История пополнений

Дата заказа: 15-09-2020 11:11:04

Название фильма: wewg
Добавлено: 5
Продавец: time Eire

Дата заказа: 12-09-2020 21:21:10

Название фильма: fjfnrjfnrif
Добавлено: 3
Продавец: time Eire

Дата заказа: 12-09-2020 21:21:10

Название фильма: rjrgjfr
Добавлено: 5
Продавец: kaiwue jdfndjnf

Дата заказа: 12-09-2020 21:21:07

Название фильма: wewg
Добавлено: 2
Продавец: time Eire

Дата заказа: 12-09-2020 18:18:36

Название фильма: rjrgjfr
Добавлено: 1
Продавец: Анисимов Артур

Рисунок 10 – Страница «История пополнений»

ЗАКЛЮЧЕНИЕ

В ходе выполнения индивидуального задания были углублены знания и получен начальный опыт по разработке, администрированию и защите баз данных, а также по созданию веб-приложения на языке Python с помощью фреймворка Django. Был разработано веб-приложение по заданной предметной области, использующего реляционную базу данных.

В ходе разработки была спроектирована база данных. Готовый прототип был упакован в Docker, успешно запущен и протестирован.

СПИСОК ЛИТЕРАТУРЫ

1. Документация по фреймворку Django [Электронный ресурс] // Django.Fun URL: <https://django.fun/> (дата обращения: 01.07.2020).
2. Django documentation [Электронный ресурс] // Django URL: <https://docs.djangoproject.com/en/3.0/> (дата обращения: 01.07.2020).
3. Get Docker [Электронный ресурс] // docker docs URL: <https://docs.docker.com/get-docker/> (дата обращения: 01.07.2020).
4. Docker документация на русском [Электронный ресурс] // dker.ru URL: <https://dker.ru/docs/> (дата обращения: 01.07.2020).
5. Перевод документации Python 3.x [Электронный ресурс] // Программирование на Python URL: <https://pythoner.name/documentation> (дата обращения: 01.07.2020).

ПРИЛОЖЕНИЕ А

На рисунке А.1 представлена инфологическая схема базы данных.

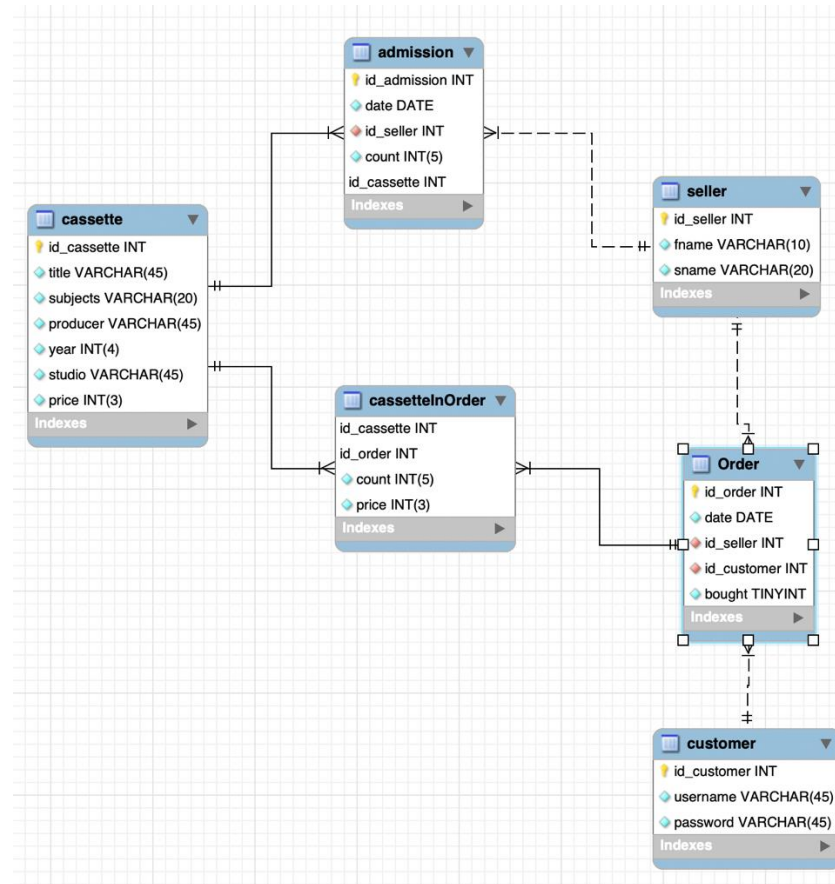


Рисунок А.1 - Инфологическая схема базы данных

В таблицах А.1-А.6 представлены описания всех полей таблиц.

Таблица А.1 – Кассеты

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id_cassette	INT	+	+	-	Уникален, число	Идентификатор
title	VARCHAR(45)	+	-	-	Строковое значение	Название
subjects	VARCHAR(20)	+	-	-	Строковое значение	Жанр
producer	VARCHAR(45)	+	-	-	Строковое значение	Отчество
year	INT(4)	+	-	-	Число	Год выпуска
studio	VARCHAR(45)	+	-	-	Строковое значение	Студия
price	INT(3)	+	-	-	Число	Цена

Таблица А.2 – Продавцы

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id_seller	INT	+	+	-	Уникален, число	Идентификатор
fname	VARCHAR(10)	+	-	-	Строковое значение	Имя
sname	VARCHAR(20)	+	-	-	Строковое значение	Фамилия

Таблица А.3 – Поставки

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id_admission	INT	+	+	-	Уникален, число	Идентификатор
date	DATE	+	-	-	Дата	Дата поставки
id_seller	INT	+	-	+	Число	Идентификатор продавца
count	INT(5)	+	-	-	Число	Кол-во кассет
id_cassette	INT	+	+	+	Число	Идентификатор кассеты

Таблица А.4 – Заказы

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	INT	+	+	-	Уникален, число	Идентификатор
date	DATE	-	-	-	Дата	Дата заказа
id_seller	INT	+	-	+	Число	Идентификатор продавца
id_customer	INT	+	-	+	Число	Идентификатор заказчика
bought	TINYINT	+	-	-	Число	Статус покупки

Таблица А.5 – Заказчики

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id_customer	INT	+	+	-	Уникален, число	Идентификатор
username	VARCHAR(45)	+	-	-	Строковое значение	Логин
password	VARCHAR(45)	+	-	-	Строковое значение	Пароль

Таблица А.6 – Кассеты в заказе

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id_cassette	INT	+	+	+	Уникален, число	Идентификатор кассеты
id_order	INT	+	+	+	Уникален, число	Идентификатор заказа
count	INT(5)	+	-	-	Число	Количество кассет
price	INT(3)	+	-	-	Число	Цена кассеты

ПРИЛОЖЕНИЕ Б

В листинге Б.1-Б.3 представлен программный код разработки.

Листинг Б.1 – Модель данных

```
from django.db import models
from django.contrib.auth.models import User

class Cassette(models.Model):
    title = models.CharField(max_length=45, null=False)
    subjects = models.CharField(max_length=20, null=False)
    producer = models.CharField(max_length=45, null=False)
    price = models.DecimalField(max_digits=3, decimal_places=0,
null=False)
    year_of_release = models.IntegerField(null=False)
    film_studio = models.CharField(max_length=45, null=False)
    count = models.DecimalField(max_digits=5, decimal_places=0,
null=False, default=0)
    in_order = models.ManyToManyField('Order',
through='CassetteInOrder')

class Seller(models.Model):
    first_name = models.CharField(max_length=10, null=False)
    last_name = models.CharField(max_length=20, null=False)

class Admission(models.Model):
    cassette = models.ForeignKey('Cassette', on_delete=models.CASCADE,
null=False)
    count = models.DecimalField(max_digits=5, decimal_places=0,
null=False, default=0)
    date = models.DateTimeField(auto_now=False, null=True)
    seller = models.ForeignKey('Seller', on_delete=models.CASCADE,
null=False)

class Order(models.Model):
    customer = models.IntegerField(default=0, null=False)
    seller = models.ForeignKey('Seller', on_delete=models.CASCADE,
null=False)
    date = models.DateTimeField(auto_now=False, null=True)
    bought = models.BooleanField(null=False, default=False)

class CassetteInOrder(models.Model):
    cassette = models.ForeignKey('Cassette', on_delete=models.CASCADE,
null=False)
    order = models.ForeignKey('Order', on_delete=models.CASCADE,
null=False)
    count = models.DecimalField(max_digits=5, decimal_places=0,
null=False, default=0)
    price = models.DecimalField(max_digits=3, decimal_places=0,
null=False)
```

Листинг Б.2 – Файл view.py

```
import datetime
import random

from django.contrib.auth import logout
from django.contrib.auth.forms import UserCreationForm
from django.db.models import Q
from django.http import HttpResponseRedirect
from django.shortcuts import render
from django.views.generic.base import View
from django.views.generic.edit import FormView

from .models import *

def toHomepage(request):
    if request.user.username == 'merchandiser':
        return HttpResponseRedirect("/admission/")
    else:
        return HttpResponseRedirect("/")

class RegisterView(FormView):
    form_class = UserCreationForm
    success_url = "/accounts/login"
    template_name = "registration/register.html"

    def form_valid(self, form):
        form.save()
        return super(RegisterView, self).form_valid(form)

class LogoutView(View):
    def get(self, request):
        logout(request)
        return HttpResponseRedirect("/")

def catalog(request):
    if request.POST.get('btn_add', None) is not None:
        add_cassette(request.POST['cassette_id'], 1, request)
        return HttpResponseRedirect(request.path_info)
    if request.user.is_authenticated:
        order = get_order_or_create(request)
        cassettes_in_order =
CassetteInOrder.objects.filter(order_id=order.id).values('cassette_id'
)
        cassettes =
Cassette.objects.filter(~Q(id__in=cassettes_in_order))
    else:
        cassettes = Cassette.objects.all()
    return render(request, "cassettes.html", {"cassettes": cassettes})

def find_seller():
    sellers = Seller.objects.all()
    arr = []
```

```

    for seller in sellers:
        arr.append(seller.id)
    choice = random.choice(arr)
    return choice

def get_order_or_create(request):
    try:
        order = Order.objects.get(customer=request.user.id,
bought=False)
    except:
        order = Order()
        order.customer = request.user.id
        order.seller = Seller.objects.get(pk=find_seller())
        order.save()
    return order

def add_cassette(cassette_id, count, request):
    order = get_order_or_create(request)
    try:
        cassette_in_order =
CassetteInOrder.objects.get(cassette_id=cassette_id,
order_id=order.id)
    except:
        cassette_in_order = CassetteInOrder()
        cassette_in_order.cassette =
Cassette.objects.get(pk=cassette_id)
        cassette_in_order.order = order
        cassette_in_order.price =
Cassette.objects.get(pk=cassette_id).price
        cassette_in_order.count = count
        cassette_in_order.save()
    return

def cart(request):
    if request.POST.get('btn_edit', None) is not None:
        add_cassette(request.POST['cassette_id'],
int(request.POST['count']), request)
        return HttpResponseRedirect(request.path_info)
    if request.POST.get('btn_remove', None) is not None:
        remove_cassette(request.POST['cassette_id'], request)
        return HttpResponseRedirect(request.path_info)
    order = get_order_or_create(request)
    cassettes_in_order =
CassetteInOrder.objects.filter(order_id=order.id).values('cassette_id'
)
    cassettes = Cassette.objects.filter(id__in=cassettes_in_order)
    sum_price = 0
    sum_count = 0
    for cassette in cassettes:
        cassette.selected_count =
CassetteInOrder.objects.get(cassette_id=cassette.id,
order_id=order.id).count
        sum_price += cassette.selected_count * cassette.price
        sum_count += cassette.selected_count
    if request.POST.get('btn_confirm', None) is not None:

```

```

        order.date = datetime.datetime.now()
        order.bought = True
        for cassette in cassettes:
            cassette.count = cassette.count - cassette.selected_count
            cassette.save()
        order.save()
        return HttpResponseRedirect(request.path_info)
    context = {
        'cassettes': cassettes,
        'sum_price': sum_price,
        'sum_count': sum_count,
    }
    return render(request, 'cart.html', context)

def remove_cassette(cassette_id, request):
    order = get_order_or_create(request)
    cassette_in_order =
    CassetteInOrder.objects.get(cassette_id=cassette_id,
    order_id=order.id)
    if cassette_in_order is not None:
        cassette_in_order.delete()
    return

def orders(request):
    c_orders = Order.objects.filter(customer=request.user.id,
    bought=True).order_by('-date')
    for order in c_orders:
        sum_price = 0
        sum_count = 0
        cassettes_in_order =
    CassetteInOrder.objects.filter(order=order).values('cassette_id')
        cassettes = Cassette.objects.filter(id__in=cassettes_in_order)
        order.cassettes = cassettes
        for cassette in cassettes:
            try:
                cassette_in_order =
    CassetteInOrder.objects.get(order_id=order, cassette_id=cassette.id)
                cassette.selected_count = cassette_in_order.count
                cassette.bought_price = cassette_in_order.price
                sum_price += cassette.selected_count *
    cassette.bought_price
                sum_count += cassette.selected_count
            except:
                cassette.selected_count = 0
                order.sum_price = sum_price
                order.sum_count = sum_count
    context = {
        'c_orders': c_orders,
    }
    return render(request, 'orders.html', context)

def admission(request):
    cassettes = Cassette.objects.all()
    context = {
        'cassettes': cassettes,

```

```

    }
    if request.POST.get('btn_add_cassette', None) is not None:
        try:
            add_admission = Admission()
            add_admission.cassette =
Cassette.objects.get(title=request.POST['cassette_title'])
            add_admission.count = int(request.POST['count'])
            add_admission.date = datetime.datetime.now()
            add_admission.seller =
Seller.objects.get(pk=find_seller())
            add_admission.save()

            cassette =
Cassette.objects.get(title=request.POST['cassette_title'])
            cassette.count = cassette.count +
int(request.POST['count'])
            cassette.save()
            return HttpResponseRedirect(request.path_info)
        except:
            return render(request, 'admission.html', context)
    return render(request, 'admission.html', context)

def getAdmissions(request):
    admissions = Admission.objects.all().order_by('-date')
    context = {
        'admissions': admissions,
    }
    return render(request, 'admissions.html', context)

```

Листинг Б.3 – Файл urls.py

```

from django.urls import path, include
from . import views

urlpatterns = [
    path('', views.catalog, name="catalog"),
    path('accounts/', include('django.contrib.auth.urls')),
    path('register/', views.RegisterView.as_view()),
    path('logout/', views.LogoutView.as_view()),
    path('accounts/profile/', views.toHomepage),
    path('cart/', views.cart, name='cart'),
    path('orders/', views.orders, name='orders'),
    path('admission/', views.admission, name='admission'),
    path('admissions/', views.getAdmissions, name='admissions')
]

```


Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет среднего профессионального образования

ДНЕВНИК ПРАКТИКИ

за период с 02.02.2020 по 02.07.2020

Студент	<u>Дмитриев Н.К.</u> (Фамилия, И.О.)
Факультет	<u>СПО</u>
Группа	<u>Y2335</u>
Направление (специальность)	<u>09.02.07 Информационные системы и программирование</u>
Место прохождения практики	<u>Факультет СПО</u>
Руководитель практики	<u>Ефимова Т.Н., факультет СПО, преподаватель Говоров А.И., факультет СПО, преподаватель</u>
Ответственный за проведение практики от университета	<u>Королев В.В., зам. директора факультета</u>

Индивидуальное задание
выполнено полностью

(подпись ответственного
за проведение практики от
университета)

(дата)

Санкт-Петербург

2020

Период	Краткое содержание работы	Отметка о выполнении
02.02.2020 – 09.02.2020	Вводный инструктаж. Ознакомление с инструкцией по технике безопасности. Ознакомление с целями и задачами практики.	
09.02.2020 – 09.03.2020	Анализ индивидуального задания. Обследование предметной области согласно индивидуальной теме учебной практики.	
10.03.2020 – 31.03.2020	Описание предметной области. Создание диаграммы классов. Создание таблиц.	
01.04.2020 – 07.04.2020	Создание модели Django в соответствии с моделью данных и настройка связи между таблицами	
08.04.2020 – 21.04.2020	Реализация элементов инфраструктуры Django, в соответствии с архитектурным паттерном Model-View-Template или сокращенно MVT. Реализация интерфейсов к системе средствами Django Templates или сторонними средствами.	
17.06.2020 – 23.06.2020	Подготовка отчетных материалов по результатам практики.	
24.06.2020 – 02.07.2020	Защита результатов практики.	

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет среднего профессионального образования

О Т З Ы В

**руководителя учебной практики
по профессиональному модулю
ПМ.11 «Разработка, администрирование и защита баз данных»**

Обучающийся Дмитриев Н.К.

(Фамилия, И.О.)

Факультет СПО

Группа У2335

Направление (специальность) 09.02.07 Информационные системы и программирование

Место прохождения практики Факультет СПО

Тема индивидуального задания

Оценка достигнутых результатов

№ п/п	Планируемые результаты обучения (показатели)	Оценка			
		5	4	3	2
1.	Корректность определения структуры базы данных				
2.	Качество реализации компонентов описания модели данных средствами Django ORM				
3.	Качество реализации контроллеров Django				
4.	Качество реализации интерфейсов к системе средствами Django Templates или сторонними средствами.				
Итоговая оценка					

Отмеченные достоинства:

.....

Отмеченные недостатки:

.....

Заключение: Считаю, все задачи, поставленные на практику, выполнены и по результатам практики студент(ка) заслуживает оценки «.....».

Руководитель практики _____

(подпись)

(ФИО)

« ____ » _____ 2020г.

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

АТТЕСТАЦИОННЫЙ ЛИСТ
Характеристика профессиональной деятельности студента во время
прохождения учебной практики

Студент ____Дмитриев Н.К.____

Группа __Y2335__

Специальность 09.02.07 Информационные системы и программирование

Место проведения практики _____

Сроки прохождения практики 20.01.2020 – 02.07.2020

Наименование профессиональных модулей (видов деятельности)

ПМ.11 «Разработка, администрирование и защита баз данных»

Виды выполняемых работ:

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Отметка о выполнении ¹
ПМ.11 Разработка, администрирование и защита баз данных		
ПК 11.1. Осуществлять сбор, обработку и анализ информации для проектирования баз данных	— осуществление корректного сбора, обработки и анализа информации для проектирования баз данных	
	— обоснование выявления объектов проектируемой БД и установки отношений между ними на основе анализа предметной области;	
ПК 11.2. Проектировать базу данных на основе анализа предметной области.	— использование CASE-средств автоматизированного проектирования при моделировании базы данных при построении концептуальной, даталогической и физической моделей БД;	
	— соответствие проекта структурной и манипуляционной частей БД заданным критериям функциональности.	

¹ Указывается «+» или «-». Считается, что программа практики выполнена, если студентом выполнено не менее 70% перечисленных видов работ.

Результаты (освоенные профессиональные компетенции)	Основные показатели оценки результата	Отметка о выполнении ¹
ПК 11.3. Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.	использование CASE-средств автоматизированного проектирования при моделировании базы данных при построении концептуальной, даталогической и физической моделей БД;	
	обоснование выбранных методов защиты объектов базы данных в соответствии с требованиями задачи.	
ПК 11.4. Реализовывать базу данных в конкретной системе управления базами данных.	соответствие реализации структурной части БД средствами СУБД даталогической и физической моделям данных;	
	— соответствие реализации манипуляционной части БД средствами СУБД заданным критериям;	
	— соответствие реализации приложения БД заданным критериям функциональности;	
ПК 11.5. Администрировать базы данных.	— обоснованность выбора архитектуры клиент-серверного взаимодействия в соответствии с технологией разработки базы данных;	
	соответствие заданным требованиям обеспечения целостности данных и контроля доступа к данным;	
	— соответствие заданным требованиям управления привилегиями пользователей базы данных программными средствами;	
	— соответствие конфигурирования сетевых устройств требованиям обеспечения доступа к данным.	
ПК 11.6. Защищать информацию в базе данных с использованием технологии защиты информации.	соответствие конфигурирования сетевых устройств требованиям защиты данных при передаче данных по сети	
	соответствие заданным требованиям программных средств защиты информации в базе данных средствами СУБД	
	— соответствие заданным требованиям управления привилегиями пользователей базы данных программными средствами;	

Руководитель практики от факультета СПО: _____

Дата: _____