

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 Анисимов А.И. (Фамилия И. О.) Группа № У2335

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 распределено (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 за- дание Дневник практики
3.	Моделирование базы данных и реализация	10.03.2020 – 31.03.2020	Описание предметной области. Со- здание диаграммы классов Создание таблиц Заполнение таблиц данными (ко- манды)	Отчет по практике: индивидуальное за- дание Дневник практики
4.	Реализация модели данных средствами Django ORM	01.04.2020 – 07.04.2020	Создание модели Django в соответ- ствии с моделью данных и настройка связи между таблицами	Отчет по практике: индивидуальное за- дание Дневник практики
5.		82.04.2020 – 21.04.2020	Реализация элементов инфраструк- туры Django, в соответствии с архи- тектурным паттерном Model-View- Template или сокращенно MVT. Ре- ализация интерфейсов к системе средствами Django Templates или сторонними средствами.	Отчет по практике: индивидуальное за- дание Дневник практики
6.	Подготовка отчет- ных материалов	17.06.2020 – 23.06.2020	Формирование отчета о практике	Отчет по практике: индивидуальное за- дание Дневник практики
7.	Защита результатов практики	24.06.2020 – 02.07.2020	Защита результатов практики в форме устного собеседования и представления результатов прак- тики	

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 Анисимов А.И. Группа У2335
(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

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
3.1 Используемые технологии	8
3.2 Программные средства	8
3.3 Шаблон проектирования MVT	8
4 Программная реализация	10
Заключение	17
Список литературы	18
Приложение А	20
Приложение Б	24

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

«Морской круиз». Управление морского пароходства регулярно выделяет определенное количество теплоходов для выполнения круизных рейсов. Каждый теплоход имеет свою команду. За выполнение рейса команда теплохода получает 5 % от стоимости рейса. Маршрут круиза содержит несколько пунктов остановки, но обычно везет одну и ту же группу туристов из начального пункта в конечный. Билеты делятся на три категории: первого, второго и третьего классов.

2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

В процессе проектирования базы данных была использована методология ERD (Entity Relationship Diagrams – диаграмма «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:

- теплоходы;
- моряки;
- команды;
- туры;
- билеты;
- круизы;
- маршруты;
- категории билетов.

Логическая модель соответствует первой нормальной форме, так как все атрибуты атомарные. Описание логической модели в приложении А в таблицах А.1-А.8.

3 ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА

3.1 Используемые технологии

При разработке системы были использованы следующие технологии:

- технология баз данных – SQLite;
- Django

3.2 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 2019.2.4 x64 – IDE для профессиональной разработки на Python
- Docker – открытая платформа для разработки, доставки и запуска приложений.

3.3 Шаблон проектирования MVT

Фреймворк Django реализует архитектурный паттерн Model-View-Template или сокращенно MVT, который является модификацией распространенного в веб-программировании паттерна MVC (Model-View-Controller).

На рисунке 1 представлена архитектура MVT в Django.

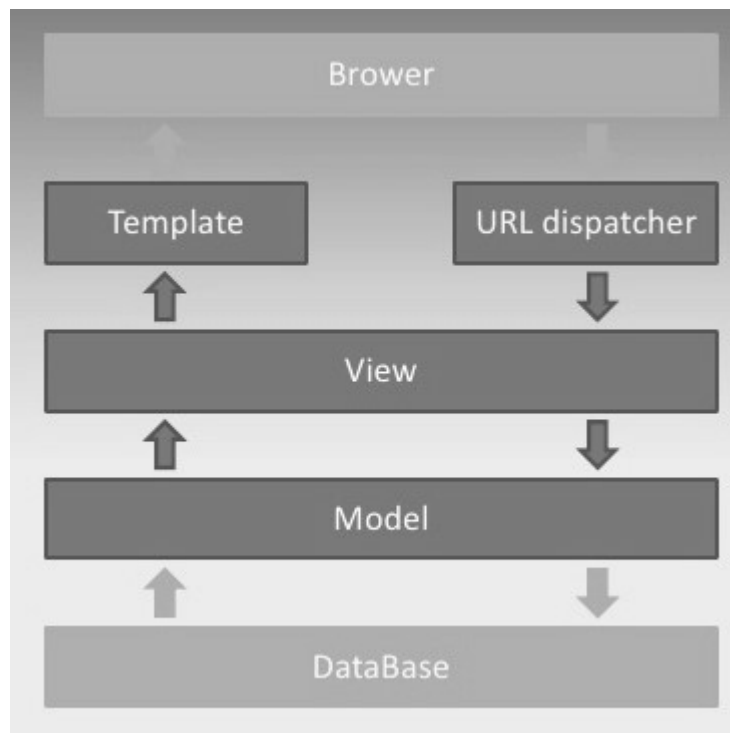


Рисунок 1 – Архитектура MVT

URL dispatcher на основании запрошенного адреса URL определяет, какой ресурс должен обрабатывать данный запрос.

View получает запрос, обрабатывает его и отправляет пользователю некоторый ответ. Если для обработки запроса необходимо обращение к модели и базе данных, то View взаимодействует с ними. Для создания ответа может применять Template или шаблоны.

Model: описывает данные, используемые в приложении. Отдельные классы, как правило, соответствуют таблицам в базе данных.

Template: представляет логику представления в виде сгенерированной разметки html.

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.

На рисунке 2 представлен интерфейс покупки билета на тур.

Вы собираетесь купить билет на тур.

Идентификатор: 1

Время начала: July 3, 2020, 2:46 p.m.

Ваше имя:

Category: 

Рисунок 2 – Покупка билета

Главная страница для пользователей представлена на рисунке 3.

Предстоящие туры:

Тур #1

Время начала: July 3, 2020, 2:46 p.m.

Название теплохода: Винни-Пух

[Купить билет](#)

Тур #2

Время начала: Sept. 23, 2022, 3:12 p.m.

Название теплохода: Пятачок

[Купить билет](#)

Тур #3

Время начала: Oct. 15, 2020, 9:03 a.m.

Название теплохода: Винни-Пух

[Купить билет](#)

Не знаете, какой тур выбрать?

[Помочь выбрать](#)



Рисунок 3 – Вид главной страницы

Для пользователей доступна услуга поиска подходящего тура. На рисунке 4 представлена система поиска подходящего тура для пользователя.

Найдём подходящий для вас тур.

На каком теплоходе вы хотите совершить путешествие?

Motorship: 

[Найти](#)

Взгляните на наши теплоходы, на которых вам предстоит путешествовать:

[Посмотреть](#)

Рисунок 4 – Поиск подходящего тура

Пользователь может ознакомиться со всеми теплоходами, чтобы найти подходящий. Страница с теплоходами представлена на рисунке 5.

Наши теплоходы

Теплоход#1

Название: Винни-Пух

Дата изготовления: May 14, 2019

Конец срока годности: April 18, 2023

Теплоход#2

Название: Пятачок

Дата изготовления: Sept. 25, 2019

Конец срока годности: March 18, 2021

[Вернуться на главную](#)

Рисунок 5 – Страница с теплоходами

После регистрации на тур, пользователю выводится информация о билете, его номер и категория. Страница окончания регистрации представлена на рисунке 6.

Спасибо!

Вы зарегистрированы на тур.

Категория вашего билета: Половина включена

Номер билета для предоставления на очной регистрации: #18

[Вернуться на главную](#)

Рисунок 6 – Страница окончания регистрации

Интерфейсы пользователя (регистрация, авторизация, список билетов) представлен на рисунках 7-9.

Ваши билеты:

#21

Категория: Половина включена

Дата: July 3, 2020, 2:46 p.m.

Сдать -

#22

Категория: Половина включена

Дата: July 3, 2020, 2:46 p.m.

Сдать -

Вернуться на главную

Рисунок 7 – Список билетов пользователя

Username:

Password:

login

Нет аккаунта?

Рисунок 8 – Интерфейс авторизации

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 9 – Интерфейс регистрации

ЗАКЛЮЧЕНИЕ

В ходе изучения учебной практики по профессиональному модулю ПМ.11 «Разработка, администрирование и защита баз данных» были углублены знания и получен начальный опыт по основным видам деятельности «Разработка, администрирование и защита баз данных». Был разработан прототип веб-приложения по заданной предметной области, использующего реляционную базу данных.

В ходе разработки была спроектирована база данных, которая описана в модели данных приложения, описаны методы взаимодействия с базой данных, описана внешняя оболочка веб-приложения. Готовый прототип был упакован в Docker.

СПИСОК ЛИТЕРАТУРЫ

1. Методология проектирования баз данных [Электронный ресурс] // Библиотека Карагандинского Государственного Технического Университета URL: http://lib.kstu.kz:8300/tb/books/Baz@i_dann@ih_Rad@mko/Theory/10_2.htm (дата обращения: 25.06.2020)
2. MySQL Workbench: Visual Database Design [Электронный ресурс] // MySQL URL: <https://www.mysql.com/products/workbench/design/> (дата обращения: 25.06.2020)
3. Информатика – курс лекций [Электронный ресурс] // Учебно-методический центр кафедры информационных технологий URL: http://dit.isuct.ru/IVT/sitanov/Literatura/InformLes_4.html (дата обращения: 25.06.2020)
4. PyCharm [Электронный ресурс] // JetBrains URL: <https://www.jetbrains.com/ru-ru/pycharm/> (дата обращения: 25.06.2020)
5. Get Docker [Электронный ресурс] // docker docs URL: <https://docs.docker.com/get-docker/> (дата обращения: 25.06.2020)
6. Введение в Django [Электронный ресурс] // METANIT.COM сайт о программировании URL: <https://metanit.com/python/django/1.1.php> (дата обращения: 25.06.2020)
7. Django rest framework [Электронный ресурс] // YouTube URL: <https://www.youtube.com/playlist?list=PLF-NY6ldwAWqP9PqPU3LA7mX2KJVyLhC> (дата обращения: 07.06.2020)
8. Создание Django API используя Django Rest Framework часть 1 [Электронный ресурс] // Еще один блог веб разработчика URL: <https://webdevblog.ru/sozdanie-django-api-ispolzuya-django-rest-framework-apiview/> (дата обращения: 11.06.2020)
9. Создание Django API используя Django Rest Framework часть 2 [Электронный ресурс] // Еще один блог веб разработчика URL:

<https://webdevblog.ru/sozдание-django-api-ispolzuya-djangorestframework-chast-2/> (дата обращения: 11.06.2020)

10. Muse-UI [Электронный ресурс] Muse-UI URL: <https://muse-ui.org/#/en-US> (дата обращения: 11.06.2020)

ПРИЛОЖЕНИЕ А

На рисунке А.1 представлена инфологическая схема спроектированной базы данных.

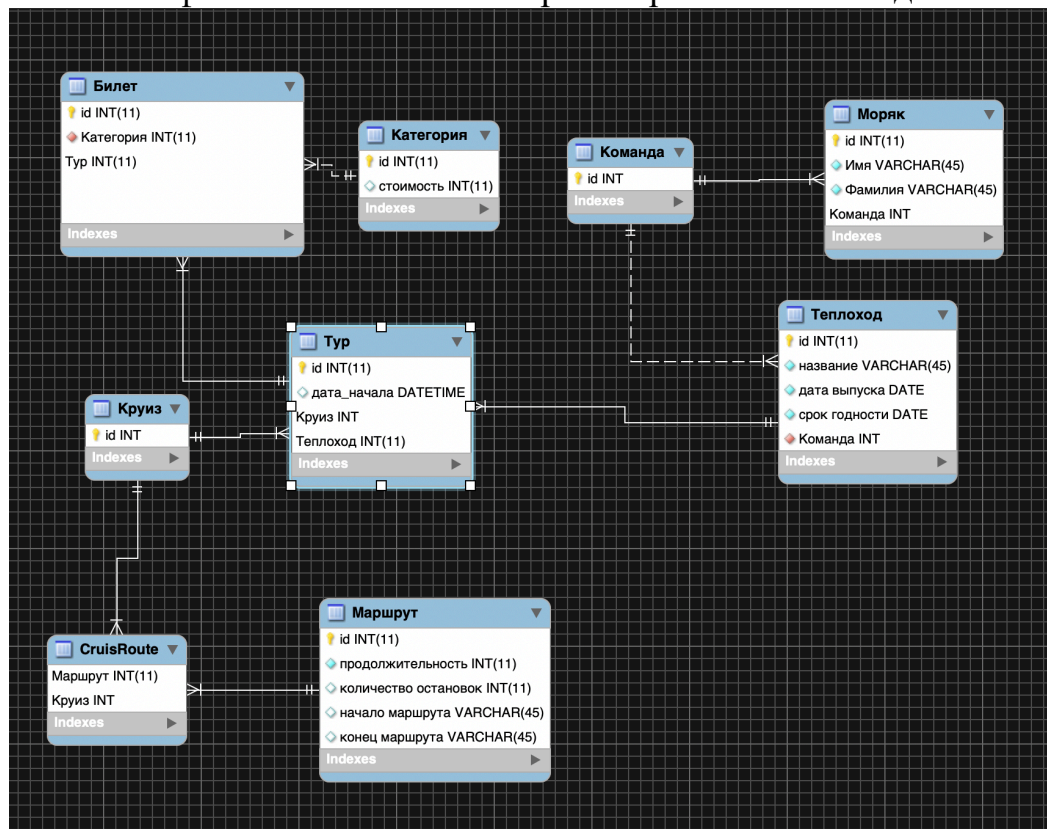


Рисунок А.1 – Инфологическая схема

В таблицах А.1-А.8 представлены описания всех полей таблиц.

Таблица А.1 – Билет

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Идентификатор	1
category	INT	+	-	+	Цифры	Категория билета	ВНЕШНИЙ КЛЮЧ
tour	INT	+	-	+	Цифры	На какой тур билет	ВНЕШНИЙ КЛЮЧ

Таблица А.2 – Категория билета

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Идентификатор	1
cost	FLOAT	+	-	-	Уникален, целое положительное число	Стоимость билета по этой категории	500

Таблица А.3 – Теплоход

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Идентификатор	1
name	VARCHAR(45)	+	-	-	Строковое значение	Название	Винни-Пух
Date_start	DATE	+	-	-	Формат YYYY-MM-DD	Дата производства	2020.05.10
Date_end	DATE	+	-	-	Формат YYYY-MM-DD	Дата конца срока годности	2020.05.10
team	INT	-	-	+	Цифры	Команда теплохода	2

Таблица А.4 – Моряк

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Табельный номер	1
name	Varchar(45)	+	-	-	Строка	Имя	Петров
lastName	Varchar(45)	+	+	-	Строка	Фамилия	Пётр

Продолжение таблицы А.4

1	2	3	4	5	6	7	8
team	INT	+	-	+	Цифры	Номер команды	2

Таблица А.5 – Тур

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Идентификатор	1
Date_start	DATE	+	-	-	Формат YYYY-MM-DD	Дата начала	2020.05.10
cruis	INT	+	-	+	Целое положительное число	Номер круиза	To client
motorship	INT	+	-	+	Целое положительное число	Номер теплохода	1

Таблица А.6 – Круиз

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Идентификатор	1

Таблица А.7 – Маршрут круиза

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
cruis	INT	+	-	+	Целое положительное число	Идентификатор круиза	1
route	INT	+	-	+	Целое положительное число	Идентификатор маршрута	1

Таблица А.8 – Маршрут

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения	Пример заполнения поля
id	INT	+	+	-	Уникален, целое положительное число	Идентификатор	1
duration	INT	+	-	-	Целое положительное число	Продолжительность маршрута	2
stops	INT	+	-	-	Целое положительное число	Количество остановок	2
start	VAR-CHAR(45)	+	-	-	Строковое значение	Название товара	Санкт-Петербург
end	VAR-CHAR(45)	+	-	-	Строковое значение	Место конца маршрута	Москва

ПРИЛОЖЕНИЕ Б

В листингах Б.1-Б.10 представлен программный код разработки.

Листинг Б.1 – Модель данных

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

# Create your models here.

class MotorshipTeam(models.Model):
    team_id = models.IntegerField()

class Motorship(models.Model):
    motorship_id = models.IntegerField()
    motorship_name = models.CharField(max_length=45)
    motorship_date_start = models.DateField()
    motorship_date_end = models.DateField()
    team = models.ManyToManyField(MotorshipTeam)

    def __str__(self):
        return "%s" % self.motorship_name

class Sailor(models.Model):
    sailor_id = models.IntegerField()
    sailor_name = models.CharField(max_length=45)
    sailor_lastName = models.CharField(max_length=45)
    team = models.ManyToManyField(MotorshipTeam)

class Cruis(models.Model):
    cruise_id = models.IntegerField()

class Tour(models.Model):
    tour_id = models.IntegerField()
    tour_datetime = models.DateTimeField()
    cruise = models.ManyToManyField(Cruis)
    motorship = models.ForeignKey(Motorship,
on_delete=models.DO_NOTHING)

class Route(models.Model):
    route_id = models.IntegerField()
    route_days = models.IntegerField()
    route_stops = models.IntegerField()
    route_begin = models.CharField(max_length=45)
    route_end = models.CharField(max_length=45)
```



```

class CruisRoute(models.Model):
    route = models.ManyToManyField(Route)
    cruises = models.ManyToManyField(Cruises)

class TicketCategory(models.Model):
    category_name = models.CharField(default="Category",
max_length=45)
    category_id = models.IntegerField()
    category_cost = models.IntegerField()

    def __str__(self):
        return "%s" % self.category_name

class Ticket(models.Model):
    ticket_id = models.IntegerField()
    user_id = models.IntegerField(default=0)
    ownername = models.CharField("Ваше имя",
max_length=45, default=" ")
    category = models.ForeignKey(TicketCategory,
on_delete=models.DO_NOTHING)
    tour = models.ForeignKey(Tour,
on_delete=models.DO_NOTHING)

```

Листинг Б.2 – Файл views.py

```

from django.shortcuts import render,
get_object_or_404, redirect
from cruises.models import Motorship, Tour,
Ticket, Sailor
from cruises.forms import TicketForm,
FindForm
from django.views.generic.edit import
FormView
from django.contrib.auth.forms import
UserCreationForm
from django.http import HttpResponseRedirect
from django.views.generic.base import View
from django.contrib.auth import logout

```

```

class LogoutView(View):

```

```

def get(self, request):
    # Выполняем выход для пользователя,
запросившего данное представление.
    logout(request)

    # После чего, перенаправляем
пользователя на главную страницу.
    return HttpResponseRedirect("/")

class RegisterView(FormView):
    form_class = UserCreationForm

    # Ссылка, на которую будет
перенаправляться пользователь в случае
успешной регистрации.
    # В данном случае указана ссылка на
страницу входа для зарегистрированных
пользователей.
    success_url = "/accounts/login"

    # Шаблон, который будет использоваться
при отображении представления.
    template_name =
"registration/register.html"

    def form_valid(self, form):
        # Создаём пользователя, если данные
в форму были введены корректно.
        form.save()

        # Вызываем метод базового класса
return super(RegisterView,

```

```

self).form_valid(form)

# Create your views here.
def motorship_list(request):
    motorships = Motorship.objects.all()
    return render(request,
"crises/motorship_list.html",
{"motorships": motorships})

# def login(request):
#     return render(login,
"registration/login.html")

def find(request):
    if request.method == "POST":
        form = FindForm(request.POST)
        if form.is_valid():
            form = form.save(commit=False)
            return redirect("/tour/" +
str(form.motorship.id))
        else:
            form = FindForm()
            return render(request,
"crises/find.html", {"form" : form})

def sailor_list(request):
    sailors = Sailor.objects.all()
    return render(request,
"crises/sailor_list.html", {"sailors":
sailors})

def thanks(request, pk):

```

```

        ticket = get_object_or_404(Ticket,
id=pk)
        return render(request,
"cruises/thanks.html", {"ticket" : ticket})

def tickets(request):
    tickets =
Ticket.objects.all().filter(user_id =
request.user.id)
    return render(request,
"cruises/tickets.html", {"tickets" :
tickets})

def tour_list(request):
    tours = Tour.objects.all()
    return render(request,
"cruises/tours.html", {"tours" : tours })

def tour(request, pk):
    tours =
Tour.objects.all().filter(motorship_id = pk)
    return render(request,
"cruises/tour.html", {"tours" : tours })

def remove(request, pk):
    ticket = get_object_or_404(Ticket,
id=pk)
    ticket.delete()
    return render(request,
"cruises/remove.html", {"ticket" : ticket})

def tour_buy(request, pk):

```

```

    tour = get_object_or_404(Tour, id=pk)

    if request.method == "POST":
        form = TicketForm(request.POST)
        if form.is_valid():
            form = form.save(commit=False)
            form.ticket_id = 0
            form.user_id = request.user.id
            form.category = form.category
            form.tour = tour
            form.save()
            return redirect("/thanks/" +
str(form.id))
        else:
            form = TicketForm()
            return render(request,
"cruises/buy.html", {"tour" : tour, "form" :
form})

```

Листинг Б.3 – urls.py приложения

```

# coding=utf-8
from django.urls import path, include
from . import views

urlpatterns = [
    path('', views.tour_list, name = "tour_list"),
    path('thanks/<int:pk>', views.thanks, name = "thanks"),
    path('remove/<int:pk>', views.remove, name="remove"),
    path('tour/<int:pk>', views.tour, name = "tour"),
    path('motorship_list/', views.motorship_list, name =
"motorships"),
    path('sailor_list/', views.sailor_list, name =
"sailors"),
    path("buy/<int:pk>", views.tour_buy, name = "tour_buy"),
    path("tickets/", views.tickets, name="tickets"),
    path('accounts/', include('django.contrib.auth.urls')),
    path('register/', views.RegisterView.as_view()),

```

```
path('logout/', views.LogoutView.as_view()),  
path("find/", views.find, name = "find")  
]
```

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>Анисимов А.И.</u> (Фамилия, И.О.)
Факультет	<u>СПО</u>
Группа	<u>У2335</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</u> <u>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 уни-
верситета)

(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 Анисимов А.И.

(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 _____ Анисимов А.И. _____

Группа _Y233_

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.	
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;	

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
	— 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: _____