

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 Платонов Е.Н. _____ Группа № У2335
(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-ViewTemplate 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 Платонов Е.Н. 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 | 5 |
| 1.1 Формулировка поставленной задачи | 5 |
| 1.2 Описание предметной области | 6 |
| 2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ | 7 |
| 3 ТЕХНОЛОГИИ И ПРОГРАММНЫЕ СРЕДСТВА | 7 |
| 2.1 Используемые технологии | 7 |
| 2.2 Программные средства | 8 |
| 4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ | 8 |
| ЗАКЛЮЧЕНИЕ..... | 12 |
| СПИСОК ЛИТЕРАТУРЫ..... | 13 |
| ПРИЛОЖЕНИЕ А..... | 14 |
| ПРИЛОЖЕНИЕ Б | 19 |

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 «Автомобильный дилер». Описание предметной области: Дилер занимается продажей различных б/у автомобилей. Клиенты могут выставлять свои авто на продажу.

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:

- автомобили;
- марка;
- модель;
- привод;
- коробка;
- клиент.

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-crispy-forms;
- Bootstrap4;
- Django;
- Django-allauth.

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.

Страница с объявлениями представлена на рис. 1.

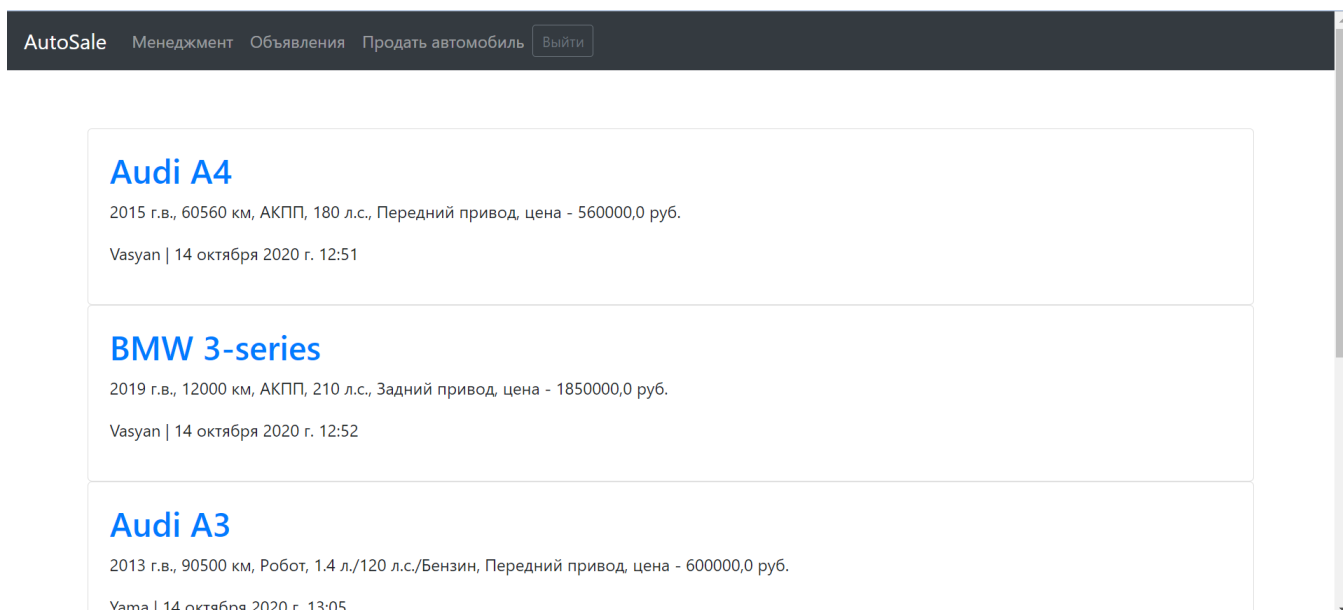


Рисунок 1 – Главная страница

Страница добавления авто представлена на рис. 2.

AutoSale Менеджмент Объявления Продать автомобиль Выйти

Марка*
----- ▾

Модель*
----- ▾

Коробка передач*
----- ▾

Год выпуска*

Пробег*

Цвет*

Рисунок 2 – Страница добавления авто

Страница «менеджмент» представлена на рис. 3.

- Добавить марку
- Добавить модель
- Добавить коробку передач
- Добавить привод
- Одобрение объявлений

Рисунок 3 – «Менеджмент»

Страница модерирования (модератор решает стоит ли объявлять пост о продаже) объявлений представлена на рис.4.

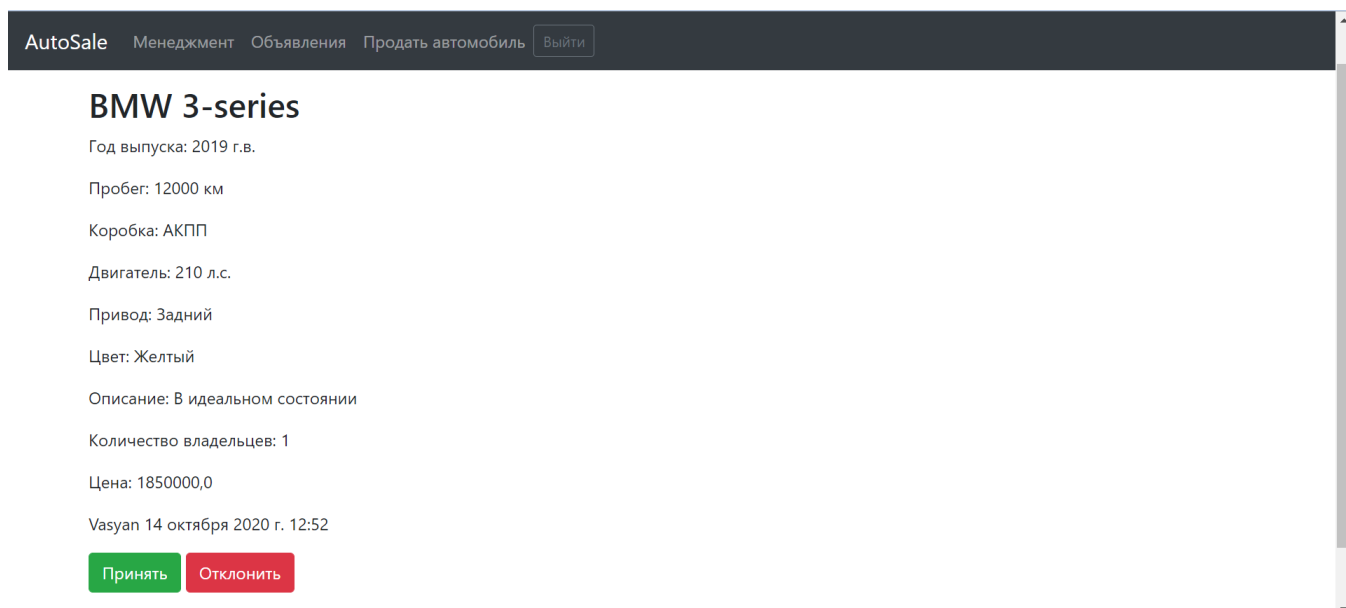


Рисунок 4 – Страница «Модерация»

На странице модерирования марками, моделями и т.п. менеджер может добавлять новые записи(рис. 5).

Название марки*

Добавить

Рисунок 5 – Страница модерации

При чем для того, чтобы выставить объявление на продажу пользователь должен быть зарегистрирован на сайте, а также, чтобы пользоваться возможностями модератора, пользователь должен иметь статус “manager”.

ЗАКЛЮЧЕНИЕ

В ходе выполнения индивидуального задания были углублены знания и получен начальный опыт по разработке, администрированию и защиты 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. Всё, что вы хотели знать о фреймворке Django и его библиотеках [Электронный ресурс] // Django.Fun URL: <https://django.fun/> (дата обращения: 24.06.2020).
2. Django documentation [Электронный ресурс] // Django URL: <https://docs.djangoproject.com/en/3.0/> (дата обращения: 23.06.2020).
3. Introduction Bootstrap [Электронный ресурс] // Bootstrap URL: <https://getbootstrap.com/docs/4.0/> (дата обращения: 23.06.2020).
4. Get Docker [Электронный ресурс] // docker docs URL: <https://docs.docker.com/get-docker/> (дата обращения: 20.06.2020).
5. Django rest framework [Электронный ресурс] // YouTube URL: <https://www.youtube.com/playlist?list=PLF-NY6ldwAWqP4S95brtPHZ5fTCxilgei> (дата обращения: 12.06.2020).

ПРИЛОЖЕНИЕ А

На рисунке А.1 представлена инфологическая схема базы данных.

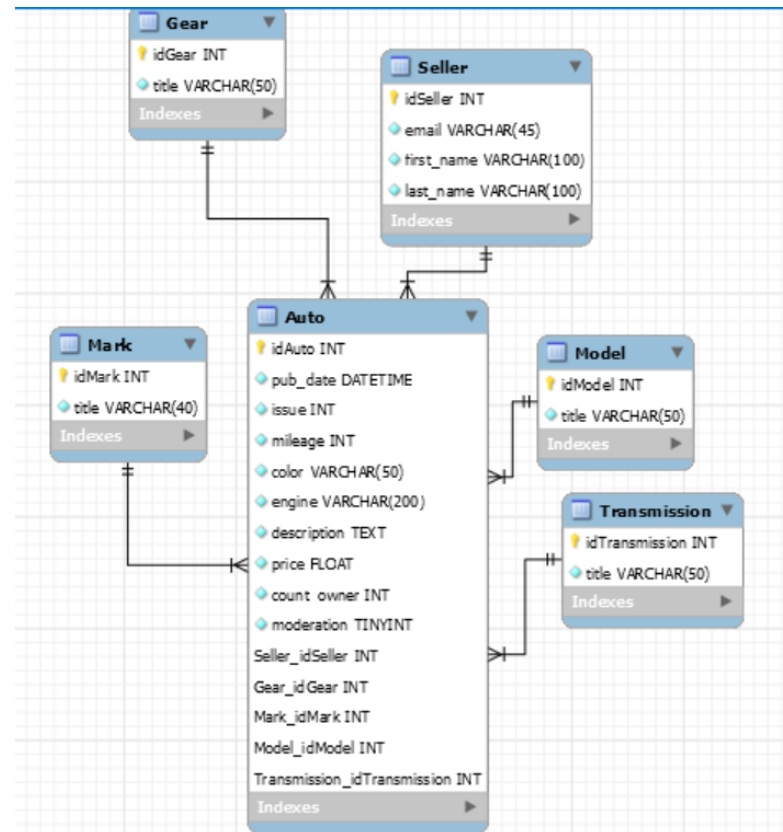


Рисунок А.1 - Инфологическая схема базы данных

В таблицах А.1-А.9 представлены описания всех полей таблиц.

Таблица А.1 – Привод

| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
|----------|-------------|----------------|----------------|--------------|--------------------|---------------|
| idGear | INT | + | + | - | Уникален, число | Идентификатор |
| title | VARCHAR(50) | + | - | - | Строковое значение | Название |

2 Марка

| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
|----------|-------------|----------------|----------------|--------------|------------------------------|---------------|
| idMark | INT | + | + | - | Уникален, число | Идентификатор |
| book | VARCHAR(13) | + | - | - | Уникален, строковое значение | Книга |

Таблица А.3 – Автомобиль

| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
|----------|------------|----------------|----------------|--------------|-------------|-----------|
|----------|------------|----------------|----------------|--------------|-------------|-----------|

| | | | | | | |
|-----------------------------|--------------|---|---|---|--------------------|-------------------------|
| idAuto | INT | + | + | - | Уникален, число | Идентификатор |
| pud_date | DATETIME | + | - | - | Дата и время | Дата и время публикации |
| issue | INT | + | - | - | Число | Год выпуска |
| mileage | INT | + | - | - | Число | Пробег |
| color | VARCHAR(50) | + | - | - | Строковое значение | Цвет |
| engine | VARCHAR(200) | + | - | - | Строковое значение | Двигатель |
| description | TEXT | + | - | - | Текст | Описание |
| price | FLOAT | + | - | - | Дробное число | Цена |
| count_owner | INT | + | - | - | Число | Количество владельцев |
| moderation | TINYINT | + | - | - | Число | Модерация |
| Seller_idSeller | INT | + | - | + | Число | Продавец |
| Gear_idGear | INT | + | - | + | Число | Привод |
| Mark_idMark | INT | + | - | + | Число | Марка |
| Model_idModel | INT | + | - | + | Число | Модель |
| Transmission_idTransmission | INT | + | - | + | Число | Коробка |

Таблица А. –
4 Трансмиссия

| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
|----------------|-------------|----------------|-------------------|-----------------|--------------------|----------------------|
| idTransmission | INT | + | + | - | Уникален, число | Идентификатор |
| title | VARCHAR(50) | + | - | - | Строковое значение | Название трансмиссии |

Таблица А.5 – Продавец

| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
|------------|--------------|----------------|-------------------|--------------|--------------------|---------------|
| idSeller | INT | + | + | - | Уникален, число | Идентификатор |
| email | VARCHAR(45) | + | - | - | Строковое значение | Почта |
| first_name | VARCHAR(100) | + | - | - | Строковое значение | Имя |
| last_name | VARCHAR(100) | + | - | - | Строковое значение | Фамилия |

Таблица А. –
6 Модель

| Имя поля | Тип данных | Обязательность | Первичный ключ | Внешний ключ | Ограничения | Пояснения |
|----------|-------------|----------------|-------------------|--------------|--------------------|---------------|
| idModel | INT | + | + | - | Уникален, число | Идентификатор |
| title | VARCHAR(50) | + | - | - | Строковое значение | Название |

ПРИЛОЖЕНИЕ Б

В листинге Б.1-Б.4 представлен программный код разработки.

Листинг Б.1 – Модель данных

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

User = get_user_model()

class Mark(models.Model):
    title = models.CharField("Название марки", max_length=40)

    class Meta:
        verbose_name = "Марка"
        verbose_name_plural = "Марки"

    def __str__(self):
        return self.title

class Model(models.Model):
    title = models.CharField("Наименование модели", max_length=50)

    class Meta:
        verbose_name = "Модель"
        verbose_name_plural = "Модели"

    def __str__(self):
        return self.title

class Transmission(models.Model):
    title = models.CharField("Тип коробки", max_length=50)

    class Meta:
        verbose_name = "Коробка передач"
        verbose_name_plural = "Коробки передач"

    def __str__(self):
        return self.title

class Gear(models.Model):
    title = models.CharField("Привод", max_length=50)

    class Meta:
        verbose_name = "Привод"
        verbose_name_plural = "Приводы"

    def __str__(self):
```

```

        return self.title

class Auto(models.Model):
    user = models.ForeignKey(
        User,
        verbose_name="Продавец",
        on_delete=models.CASCADE)
    model = models.ForeignKey(Model,
                               verbose_name="Модель",
                               on_delete=models.SET_NULL,
                               null=True)
    mark = models.ForeignKey(Mark,
                              verbose_name="Марка",
                              on_delete=models.SET_NULL,
                              null=True)
    pub_date = models.DateTimeField("Дата создания объявления",
auto_now_add=True)
    transmission = models.ForeignKey(Transmission,
                                       verbose_name="Коробка передач",
                                       on_delete=models.SET_NULL,
                                       null=True)
    issue = models.IntegerField("Год выпуска")
    mileage = models.IntegerField("Пробег")
    color = models.CharField("Цвет", max_length=50)
    gear = models.ForeignKey(Gear,
                              verbose_name="Привод",
                              on_delete=models.SET_NULL,
                              null=True)
    engine = models.CharField("Двигатель", max_length=200)
    description = models.TextField("Описание")
    price = models.FloatField("Цена", null=True)
    count_owners = models.IntegerField("Количество владельцев")
    moderation = models.BooleanField("Модерация", default=False)

    class Meta:
        verbose_name = "Автомобиль"
        verbose_name_plural = "Автомобили"

    def __str__(self):
        return "{} {} {}".format(self.user, self.mark, self.model)

```

Листинг Б.2 – Файл views.py

```

from django.shortcuts import render, get_object_or_404, redirect
from autoProbeg.forms import *
from django.http import Http404

def auto_list(request):
    """Список продающихся авто"""
    autos = Auto.objects.filter(moderation=True)
    return render(request, 'probeg/auto_list.html', {"autos": autos})

```

```

def auto_single(request, pk):
    """Конкретное продающееся авто"""
    auto = get_object_or_404(Auto, id=pk)
    return render(request, 'probeg/auto_single.html', {'auto': auto})

def createAuto(request):
    """Добавить авто"""
    if request.method == 'POST':
        form = AutoForm(request.POST)
        if form.is_valid():
            form = form.save(commit=False)
            form.user = request.user
            form.save()
            return redirect(auto_list)
    else:
        form = AutoForm()
    return render(request,
                  "probeg/createAuto.html",
                  {'form': form})

def manageTool(request):
    if request.method == 'POST':
        if 'mark' in request.POST:
            return redirect(createMark)
        elif 'model' in request.POST:
            return redirect(createModel)
        elif 'transmission' in request.POST:
            return redirect(createTransmission)
        elif 'gear' in request.POST:
            return redirect(createGear)
        elif 'manage_autos' in request.POST:
            return redirect(manageAutoAnoun)
    if not request.user.groups.filter(name="manager").exists():
        raise Http404
    return render(request,
                  'probeg/manager.html')

def createMark(request):
    if request.method == 'POST':
        form = MarkForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect(manageTool)
    else:
        form = MarkForm()
    if not request.user.groups.filter(name="manager").exists():
        raise Http404
    return render(request,
                  "probeg/manage_mark.html",
                  {'form': form})

```

```

def createModel(request):
    if request.method == 'POST':
        form = ModelForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect(manageTool)
    else:
        form = ModelForm()
    if not request.user.groups.filter(name="manager").exists():
        raise Http404
    return render(request,
                  "probeg/manage_model.html",
                  {'form': form})

def createTransmission(request):
    if request.method == 'POST':
        form = TransmissionForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect(manageTool)
    else:
        form = TransmissionForm()
    if not request.user.groups.filter(name="manager").exists():
        raise Http404
    return render(request,
                  "probeg/manage_transmission.html",
                  {'form': form})

def createGear(request):
    if request.method == 'POST':
        form = GearForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect(manageTool)
    else:
        form = GearForm()
    if not request.user.groups.filter(name="manager").exists():
        raise Http404
    return render(request,
                  "probeg/manage_gear.html",
                  {'form': form})

def manageAutoAnoun(request):
    autos = Auto.objects.filter(moderation=False)
    if request.method == 'POST':
        if 'accept' in request.POST:
            auto = Auto.objects.get(pk=int(request.POST['accept']))
            auto.moderation = True
            auto.save()

```

```

        elif 'deni' in request.POST:
            auto = Auto.objects.get(pk=int(request.POST['deni']))
            auto.delete()
            return redirect(manageAutoAnoun)
    if not request.user.groups.filter(name="manager").exists():
        raise Http404
    return render(request,
                  'probeg/manage_announcement.html',
                  {'autos': autos})

```

Листинг Б.3 – Файл urls.py

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.auto_list,
         name="auto_list"),
    path('auto/<int:pk>',
         views.auto_single,
         name="auto_single"),
    path('sellauto/',
         views.createAuto, name="createAuto"),
    path('manager/',
         views.manageTool, name="manageTool"),
    path('manager/mark',
         views.createMark, name="createMark"),
    path('manager/model',
         views.createModel,
         name="createModel"),
    path('manager/transmission',
         views.createTransmission,
         name="createTransmission"),
    path('manager/gear',
         views.createGear, name="createGear"),
    path('manager/announcement',
         views.manageAutoAnoun,
         name="manageAutoAnoun"),
]

```

Листинг Б.4 – Файл forms.py

```

from django.forms import ModelForm
from .models import *

class AutoForm(ModelForm):
    """Форма ввода авто"""

    class Meta:
        model = Auto
        fields = (
            'mark', 'model', 'transmission', 'issue',
            'mileage', 'color', 'count_owners', 'gear',
            'description', 'engine', 'price'
        )

```

```

class MarkForm(ModelForm):
    """Форма ввода марки"""

    class Meta:
        model = Mark
        fields = (
            'title',
        )

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        if self.save:
            self.fields['title'].required = True

class ModelForm(ModelForm):
    """Форма ввода марки"""

    class Meta:
        model = Model
        fields = (
            'title',
        )

class TransmissionForm(ModelForm):
    """Форма ввода марки"""

    class Meta:
        model = Transmission
        fields = (
            'title',
        )

class GearForm(ModelForm):
    """Форма ввода марки"""

    class Meta:
        model = Gear
        fields = (
            'title',
        )

class SignupForm(ModelForm):
    class Meta:
        model = User
        fields = ['first_name', 'last_name']

    def signup(self, request, user):
        user.first_name = self.cleaned_data['first_name']
        user.last_name = self.cleaned_data['last_name']

```



```
user.save()
```

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 Платонов Е.Н.
(Фамилия, И.О.)

Факультет СПО
Группа У2335

Направление (специальность) 09.02.07 Информационные системы и
программирование

Место прохождения практики Факультет СПО

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

Ответственный за проведение практики от университета Королев В.В., зам. директора факультета

С
П
О

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 Платонов Е.Н.
(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 __ Платонов Е.Н. _____

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 выполнении¹ |
|---|--|---|
| 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.

| | | |
|--|--|---|
| 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; | |
| Результаты (освоенные профессиональные компетенции) | Основные показатели оценки результата | Отметка о выполнении¹ |
| | 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: _____