МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

дисциплина «Средства и технологии разработки информационных систем»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

«Методы моделей и их валидация»

ВАРИАНТ № 22

Выполнил:

студент группы ИТП-31, Пронуза М.Ю.

Принял:

преподаватель Карась О.В.

Гомель 2024

**1. Цель работы:** освоить механизм валидации данных приходящих из запросов клиентов.

**2. Ход работы и результаты.**

Задание: Доработать веб приложение, выполненное в предыдущей лабораторной работе. Реализовать функцию добавления данных в существующие таблицы. Перед добавлением данные необходимо проверять с помощью валидатора модели.

Задание варианта 22 представлено на рисунке 1.



Рисунок 1 - вариант задания

Страница добавления машины представлена на рисунке 2.

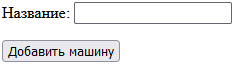
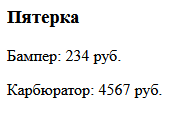


Рисунок 2 - добавление машины

Страница добавления детали представлена на рисунке 3.

Рисунок 3 - добавление детали

Код *djangoapp/urls.py* представлен в листинге A.

Код *djangoapp/settings.py* представлен в листинге Б.

Код *magazine/urls.py* представлен в листинге В.

Код *magazine/views.py* представлен в листинге Г.

Код *templates/car\_list.html* представлен в листинге Д.

Код *magazine/models.py* представлен в листинге Е.

Код *magazine/spare\_list.html* представлен в листинге Ё.

Код *magazine/forms.py* представлен в листинге Ж.

Код *magazine/add\_car.html* представлен в листинге З.

Код *magazine/add\_spare.html* представлен в листинге И.

**3. Выводы:** были освоены механизмы организации связей между моделями.

**Листинг А**

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

path('admin/', admin.site.urls),

path('', include("magazine.urls"))

]

**Листинг Б**

STATIC\_URL = 'static/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, "static"),

]

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': ['templates'],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'magazine'

]

**Листинг В**

from django.contrib import admin

from django.urls import include, path

urlpatterns = [

path('admin/', admin.site.urls),

path('', include("magazine.urls"))

]

**Листинг Г**

from django.shortcuts import render, redirect

from magazine.forms import SpareForm, CarForm

from magazine.models import Spare, Car

def spare\_list(request, car\_id):

car = Car.objects.get(pk=car\_id)

spare\_list = Spare.objects.filter(car=car)

return render(request, 'spare\_list.html', {'car\_name': car.name,'spare\_list': spare\_list})

def car\_list(request):

car\_list = Car.objects.all()

return render(request, 'car\_list.html', {'car\_list': car\_list})

def add\_spare(request):

if request.method == 'POST':

form = SpareForm(request.POST)

if form.is\_valid():

new\_spare: Spare = form.save()

return redirect('spare\_list', car\_id=new\_spare.car.id)

else:

form = SpareForm()

return render(request, 'add\_spare.html', {'form': form})

def add\_car(request):

if request.method == 'POST':

form = CarForm(request.POST)

if form.is\_valid():

form.save()

return redirect('car\_list')

else:

form = CarForm()

return render(request, 'add\_car.html', {'form': form})

**Листинг Д**

<!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">

<title>Список машин</title>

</head>

<body>

<p>

<a href="{% url 'add\_spare' %}">Добавить деталь</a>

</p>

<p>

<a href="{% url 'add\_car' %}">Добавить машину</a>

</p>

<ul>

{% for i in car\_list %}

<li><a href="{% url 'spare\_list' car\_id=i.id %}">{{i.name}}</a></li>

{% endfor %}

</ul>

</body>

</html>

**Листинг Е**

from django.db import models

class Car(models.Model):

name = models.CharField('название', max\_length=100, unique=True)

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

return str(self.name)

class Spare(models.Model):

car = models.ForeignKey(Car, on\_delete=models.CASCADE)

name = models.CharField('название', max\_length=100)

price = models.IntegerField('цена')

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

return str(self.name)

**Листинг Ё**

<!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">

<title>{{car\_name}}</title>

</head>

<body>

<h3>{{car\_name}}</h3>

{% for i in spare\_list %}

<p data-car-name="{{i.car.name}}">{{i.name}}: {{i.price}} руб.</p>

{% endfor %}

</body>

</html>

**Листинг Ж**

from django import forms

from .models import Spare, Car

class SpareForm(forms.ModelForm):

class Meta:

model = Spare

fields = '\_\_all\_\_'

class CarForm(forms.ModelForm):

class Meta:

model = Car

fields = '\_\_all\_\_'

**Листинг З**

<!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">

<title>Добавление машины</title>

</head>

<body>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Добавить машину</button>

</form>

</body>

</html>

**Листинг И**

<!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">

<title>Добавление детали</title>

</head>

<body>

<form method="post">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Добавить деталь</button>

</form>

</body>

</html>