Державний університет інтелектуальних технологій і зв’язку

Факультет: Інформаційних технологій та кібербезпеки

Кафедра: Комп’ютерних наук

**Протокол лабораторної роботи №7**

**з дисципліни**

**«Веб-технології»**

Виконав студент групи ІПЗ-3.02

Жигальов Володимир Олександрович

Перевірив ст. викл. каф. КН

Северин М.В.

Одеса - 2025

**Лабораторна робота № 7**

**Тема:** Шаблони та статичні файли в структурі фреймворку Django.

**Мета роботи:** Набуття практичного досвіду створення шаблонів, маршрутів та обробників маршрутів, робота зі статичними файлами.

**Завдання**

1. Створити скрипт, що створює каталог шаблонів `templates/shop/` у застосунку ЛБ №6.

*Рішення:*

Було вручну створено структуру каталогів:

shop/

└── templates/

└── shop/

Структура призначена для збереження шаблонів застосунку згідно з Django-конвенцією.

2. Реалізувати HTML-шаблони для відображення щонайменше двох таблиць БД.

*Рішення:*

`templates/shop/product\_list.html`

{% extends "shop/layout.html" %}

{% block title %}Список товарів{% endblock %}

{% block content %}

<h2>Список товарів</h2>

<ul>

{% for product in products %}

<li>{{ product.name }} - {{ product.price }} грн</li>

{% endfor %}

</ul>

{% endblock %}

`templates/shop/category\_list.html`

{% extends "shop/layout.html" %}

{% block title %}Список категорій{% endblock %}

{% block content %}

<h2>Категорії</h2>

<ul>

{% for category in categories %}

<li>{{ category.name }}</li>

{% endfor %}

</ul>

{% endblock %}

3. Додати базовий шаблон `layout.html` та винести повторюваний код.

*Рішення:*

`templates/shop/layout.html`

{% load static %}

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<title>{% block title %}Магазин{% endblock %}</title>

<link href="{% static 'shop/css/style.css' %}" rel="stylesheet">

</head>

<body>

<header>

<h1>Мій магазин</h1>

<nav>

<a href="/shop/products/">Товари</a>

<a href="/shop/categories/">Категорії</a>

</nav>

</header>

<main>

{% block content %}{% endblock %}

</main>

</body>

</html>

4. У `views.py` створити обробники запитів і передати дані у шаблони.

*Рішення:*

from django.shortcuts import render

from .models import Product, Category

def product\_list(request):

products = Product.objects.all()

return render(request, 'shop/product\_list.html', {'products': products})

def category\_list(request):

categories = Category.objects.all()

return render(request, 'shop/category\_list.html', {'categories': categories})

5. У `urls.py` додати маршрути для перегляду таблиць БД.

*Рішення:*

from django.urls import path

from . import views

urlpatterns = [

path('products/', views.product\_list, name='product\_list'),

path('categories/', views.category\_list, name='category\_list'),

]

6. Створити `shop/static/shop/` і додати каталоги css, images, js.

*Рішення:*

shop/

└── static/

└── shop/

├── css/

│ └── style.css

├── images/

└── js/

7. Створити файл стилів і підключити його у `layout.html`.

*Рішення:*

`static/shop/css/style.css`

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

header {

background-color: #333;

color: white;

padding: 10px;

}

nav a {

color: white;

margin-right: 10px;

text-decoration: none;

}

ul {

background: white;

padding: 15px;

border-radius: 5px;

}

8. Застосувати стилі до шаблонів для покращення вигляду таблиць.

*Рішення:*

Стилі з `style.css` автоматично застосовуються до всіх шаблонів через `layout.html`. Інтерфейс став більш охайним і зручним для читання.

**Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.**

**Відповідь на контрольні питання:**

1. Як забезпечити використання різних шаблонів, що мають однакове ім’я для різних програмних застосунків?

Відповідь:

У Django шаблони розміщуються у підкаталозі з іменем застосунку, наприклад:

`shop/templates/shop/template.html`

Під час використання шаблону вказується повний шлях:

return render(request, 'shop/template.html')

Це дозволяє уникнути конфліктів між шаблонами з однаковими назвами у різних застосунках.

2. Як створити блок динамічного виведення контенту шаблону?

Відповідь:

У базовому шаблоні використовується тег `{% block %}` для визначення динамічного блоку:

{% block content %}{% endblock %}

3. Як підключити базовий шаблон до шаблону з формування контенту?

Відповідь:

У шаблоні контенту вказується директива розширення базового шаблону:

{% extends "shop/layout.html" %}

Це дозволяє використовувати загальну структуру з базового шаблону.

4. Як вивести дані з шаблону в блок динамічного контенту?

Відповідь:

Після оголошення блоку в шаблоні контенту, всередині нього можна виводити передані з `views.py` дані, наприклад:

{% block content %}

<ul>

{% for item in items %}

<li>{{ item.name }}</li>

{% endfor %}

</ul>

{% endblock %}

5. Як підключити стилі в проєкті Django?

Відповідь:

1. Створити каталог: `shop/static/shop/css/style.css`

2. У шаблоні підключити стилі через теги `{% load static %}` і `<link>`:

{% load static %}

<link rel="stylesheet" href="{% static 'shop/css/style.css' %}">

**Висновок**

У ході виконання завдання було створено шаблони для відображення даних із бази даних, базовий шаблон з динамічними блоками, налаштовано маршрутизацію та стилізацію через статичні файли. Реалізовано розділення структури шаблонів між застосунками, що забезпечує масштабованість і повторне використання коду.