**Практическая работа 4. Создание страниц проекта**

**Задание:**

К проекту ИС для торговой компании (созданному в предыдущем задании) добавить следующее:

Придумать дизайн страниц сайта. В соответствии с ним создать и заполнить html страницы:

* главная страница с информацией о компании: должна содержать текстовый блок и картинку.
* страница со списком товаров (содержит заголовок страницы и таблицу, в которой пока нет данных)
* страница склада (содержит заголовок страницы и таблицу, в которой пока нет данных)
* страница контакты (текстовые блоки: адрес, телефон компании)

**Пример.** То есть требуется создать html файлы для страниц, соответствующих заданию.

Возможно добавление стилей к элементам страниц. На главной странице также требуется добавить рисунок. Стиль и рисунки относятся к статическим файлам, для них в папке приложения нужно создать папку static

**1. Добавление стилей.** Веб-приложение, как правило, использует различные статические файлы - изображения, файлы стилей css, скриптов javascript и так далее. Рассмотрим, как мы можем использовать подобные файлы. При создании проекта Django он уже имеет некоторую базовую настройку для работы со статическими файлами. В частности, в файле **settings.py** определена переменная **STATIC\_URL**, которая хранит путь к каталогу со статическими файлами:

|  |  |
| --- | --- |
| 1 | STATIC\_URL = 'static/' |

Переменная **STATIC\_URL** имеет значение "static/", а это значит, что достаточно создать в папке приложения каталог с именем "static" и добавить в него необходимые нам статические файлы. Но при необходимости через данную настройку мы можем изменим расположение каталога статических файлов.

Добавим в **папку приложения** новый каталог **static**. Определим для каждого типа файлов отдельные папки. В частности, создадим в папке static для изображений каталог **images**, а для стилей - каталог **css**. Подобным образом можно создавать папки и для других типов файлов.

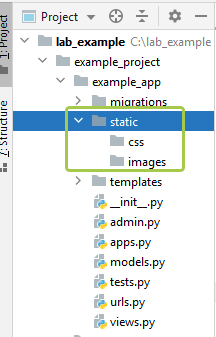


Рисунок. Структура статических файлов приложения

В папку **static/images** добавим изображение – например, файл с названием **figure1.jpg**.

В папке **static/css** создадим новый файл **styles.css**, который будет иметь какие-нибудь стили, например:

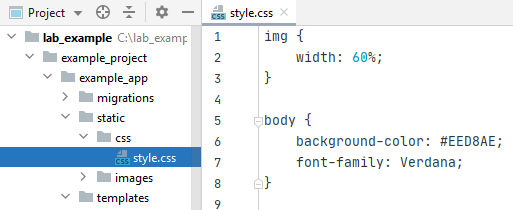


Рисунок. Пример файла со стилем для страницы

В результате структура статических файлов будет следующая:

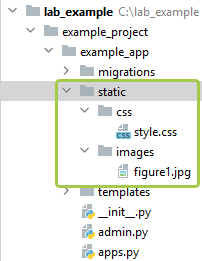


Рисунок. Структура статических файлов приложения

**2. Создать html страницы.**

Теперь можно использовать статические файлы в шаблоне (html файле). Для этого в начале файла шаблона необходимо определить инструкцию {% load static %}

При этом данный код должен идти после тега DOCTYPE.

Для определения пути к статическим файлам используются выражения типа

{% static "путь к файлу внутри папки static" %}

В приложении в папке **templates** ранее был создан шаблон **index.html для описания главной страницы**, который после подключения стилей будет иметь следующий код (цветом выделены новые строки):

<!DOCTYPE html>  
{% load static %}  
<html lang**="en"**>  
<head>  
 <meta charset**="UTF-8"**>  
 <title>Приложение для работников магазина</title>  
 <link rel**="stylesheet"** href**="{% static 'css/style.css' %}"** />  
</head>  
<body>  
<img src**="{% static 'images/figure1.jpg' %}"** alt**="Рисунок"** >  
<h1>Главная страница приложения</h1>  
<nav>  
 <a href**=""**>Главная страница</a>  
 <a href**="contacts"**>Контакты</a>  
 <a href**="vacancy"**>Список вакансий</a>  
 <a href**="products"**>Товары компании</a>  
 <a href**="store"**>Наличие на складе</a>  
</nav>  
</body>  
</html>

После запуска приложения шаблон index.html будет генерироваться в следующую веб-страницу, которая будет использовать изображение и применять стили, пример отображения главной страницы в браузере:

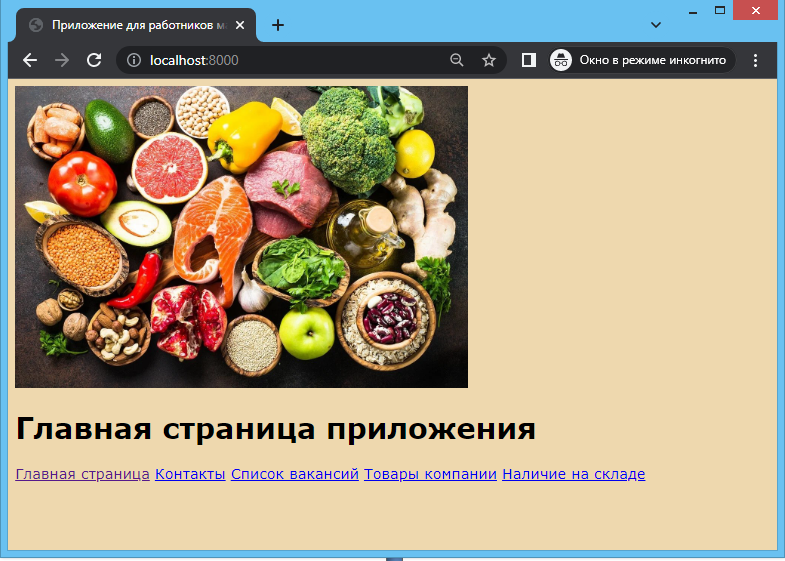


Рисунок. Пример отображения главной страницы приложения в браузере

(Следует обратить внимание, что при изменении стилей проект может не перезапуститься сразу после внесения изменений. Его нужно перезапустить. Кроме того, из-за того, что используется кеширование, изменения можно увидеть через некоторое время - или зайти в браузер в режиме инкогнито)

В задании требуется также добавить html файлы с описанием остальных страниц. Примеры.

* **страница со списком товаров (содержит заголовок страницы и таблицу, в которой пока нет данных)**

Добавим в папку templates файл products.html и опишем там следующую разметку.

<!DOCTYPE html>  
{% load static %}  
<html lang**="en"**>  
<head>  
 <meta charset**="UTF-8"**>  
 <title>Приложение для работников магазина</title>  
 <link rel**="stylesheet"** href**="{% static 'css/style.css' %}"** />  
</head>  
<body>  
<h1>Список товаров магазина</h1>  
<table>  
 <tr>  
 <th>Номер</th>  
 <th>Название продукта</th>  
 <th>Описание продукта</th>  
 </tr>  
 <tr>  
 <td>1</td>  
 <td>Тестовый продукт 1</td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>2</td>  
 <td>Тестовый продукт 2</td>  
 <td></td>  
 </tr>  
</table>  
</body>  
</html>

* **страница склада (содержит заголовок страницы и таблицу, в которой пока нет данных)** Добавим в папку templates файл store.html и напишем там разметку, похожую на файл products.html (похожую таблицу и заголовок страницы).
* **страница контакты (текстовые блоки: адрес, телефон компании)**

Добавим в папку templates файл contact.html и напишем там следующую разметку.

<!DOCTYPE html>  
{% load static %}  
<html lang**="en"**>  
<head>  
 <meta charset**="UTF-8"**>  
 <title>Приложение для работников магазина</title>  
 <link rel**="stylesheet"** href**="{% static 'css/style.css' %}"** />  
</head>  
<body>  
<h1>Контактные данные магазина</h1>  
<div>  
 <p>Адрес: г. Красноярск, ул. Киренского</p>  
 <p>Телефон: 84874873 </p>  
</div>  
</body>  
</html>

* **страница вакансии (текстовые блоки)**

Добавим в папку templates файл vacancy.html и напишем там следующую разметку.

<!DOCTYPE html>  
{% load static %}  
<html lang**="en"**>  
<head>  
 <meta charset**="UTF-8"**>  
 <title>Приложение для работников магазина</title>  
 <link rel**="stylesheet"** href**="{% static 'css/style.css' %}"** />  
</head>  
<body>  
<h1>Вакансии</h1>  
<div class**="vacancy"**>  
 <p>Должность: кассир</p>  
 <p>Требования: высшее образование </p>  
</div>  
<div class**="vacancy"**>  
 <p>Должность: старший кассир</p>  
 <p>Требования: высшее образование, опыт работы </p>  
</div>  
</body>  
</html>

Добавим в файл **style.css** описание стилей для таблиц и других элементов, используемых на страницах. Пример содержимого файла со стилями:

body {  
 background-color: #EED8AE;  
 font-family: Verdana;  
}  
  
table, th, td {  
 border: 1px solid black;  
 border-collapse: collapse;  
 padding: 5px;  
}  
  
.vacancy {  
 border: 1px solid black;  
 border-radius: 5px;  
 padding: 10px;  
 margin: 5px;  
}

**3. Создать функции представления.**

В файле views.py в функции изменим код таким образом, чтобы добавить отображения страниц html. Пример:

from django.shortcuts import render  
  
  
def index(request):  
 return render(request, **'index.html'**)  
  
  
def contacts(request):  
 return render(request, **'contact.html'**)  
  
  
def vacancy(request):  
 return render(request, **'vacancy.html'**)  
  
  
def products(request):  
 return render(request, **'products.html'**)  
  
  
def store(request):  
 return render(request, **'store.html'**)

Далее на рисунках приведены примеры внешнего вида созданных страниц в браузере.

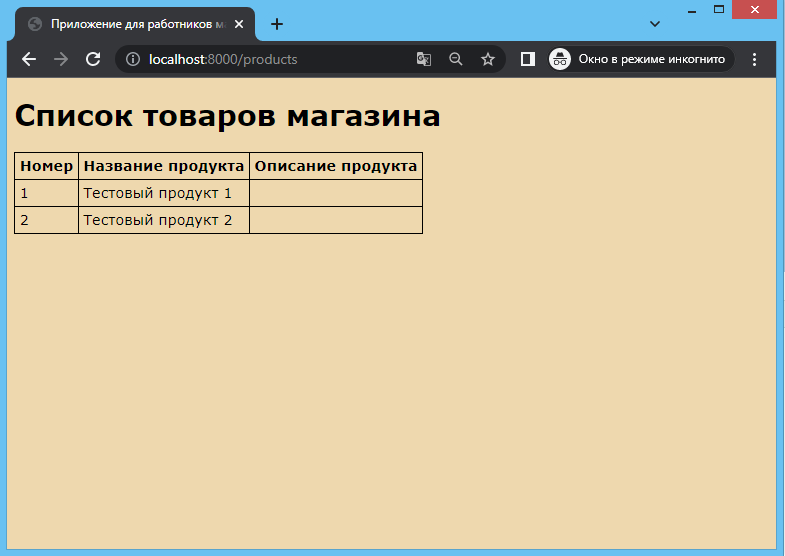


Рисунок. Пример страницы товаров

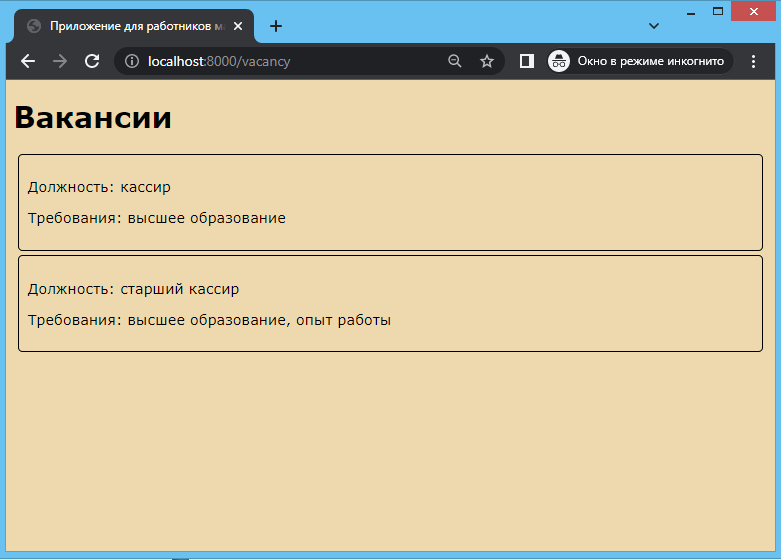


Рисунок. Пример страницы вакансий

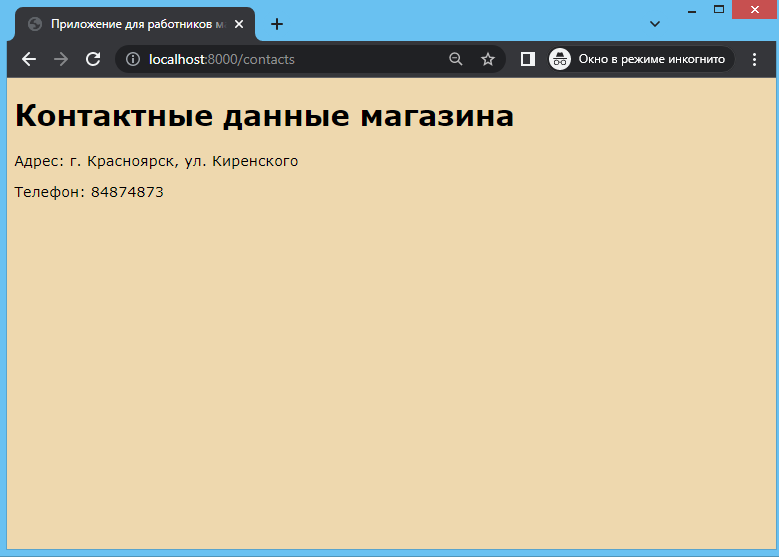


Рисунок. Пример страницы контактов