



**Министерство науки и высшего образования Российской
Федерации Федеральное государственное бюджетное
образовательное учреждение высшего образования «Московский
государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)» (МГТУ им.
Н.Э. Баумана)**

**Лабораторная работа №8
по курсу «Разработка интернет-приложений»**

**Выполнил
студент группы ИУ5-54Б
Сысойкин Е.М.**

Москва, 2020

1 Описание задания

1. Создайте прототип веб-приложения с использованием фреймворка Django:

- Создайте виртуальное окружение.
- Установите в него Django.
- Создайте проект и приложение Django.

2. Создайте представления и шаблоны (по желанию можно использовать модели), реализующие концепцию master/detail со следующей функциональностью:

- На странице master в виде списка HTML выводится информация о трех объектах (например, о трех сортах мороженого). Каждая строка списка представляет собой гиперссылку, при нажатии на которую происходит переход к странице detail.
- Страница detail содержит детальное описание объекта (сорта мороженого), фотографию, гиперссылку на master-страницу.
- Фотография относится к статическому содержимому сайта.
- Страница detail должна выводить данные с использованием таблицы HTML.
- Шаблон страницы detail получает от представления данные о детальном объекте с использованием контекста.
- НЕОБЯЗАТЕЛЬНЫЙ ПУНКТ. По желанию можно использовать верстку с применением Bootstrap (или аналогичного фреймворка), а также представления на основе классов (class-based views).

2 Текст программы

myapp/fill_data.sql

```
1 INSERT INTO myapp_detail(id, name, quantity, price, img) VALUES
2 (0, 'Труба A', 50, 200.1, '1.png'),
3 (1, 'Труба B', 64, 400.1, '2.png'),
4 (2, 'Труба C', 10, 500.1, '3.png'),
5 (3, 'Труба D', 90, 600.1, '4.png'),
6 (4, 'Труба E', 84, 700.1, '5.png'),
7 (5, 'Труба F', 18, 800.1, '6.png'),
8 (6, 'Труба G', 6, 900.1, '7.png'),
9 (7, 'Труба K', 64, 100.1, '8.png'),
10 (8, 'Труба L', 52, 110.1, '9.png'),
11 (9, 'Труба M', 76, 120.1, '10.png'),
12 (10, 'Труба N', 39, 130.1, '11.png'),
```

```

13 (11, 'Трыба O', 102, 140.1, '12.png'),
14 (12, 'Трыба P', 11, 150.1, '13.png'),
15 (13, 'Трыба R', 47, 160.1, '14.png'),
16 (14, 'Трыба S', 29, 170.1, '15.png'),
17 (15, 'Трыба T', 58, 180.1, '16.png'),
18 (16, 'Трыба Q', 78, 190.1, '17.png'),
19 (17, 'Трыба W', 110, 200.1, '18.png'),
20 (18, 'Трыба Y', 140, 210.1, '19.png'),
21 (19, 'Трыба U', 13, 220.1, '20.png'),
22 (20, 'Трыба I', 18, 230.1, '21.png'),
23 (21, 'Трыба H', 28, 240.1, '22.png'),
24 (22, 'Трыба J', 44, 250.1, '23.png'),
25 (23, 'Трыба Z', 87, 260.1, '24.png'),
26 (24, 'Трыба X', 67, 270.1, '25.png'),
27 (25, 'Трыба V', 34, 280.1, '26.png');

```

myapp/myapp/urls.py

```

1 """myapp URL Configuration
2
3 The `urlpatterns` list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/3.1/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from myapp import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', views.master, name='master'),
23     path('detail?<id>', views.detail, name='detail')
24 ]

```

myapp/myapp/apps.py

```

1 from django.apps import AppConfig
2

```

```

3 class MyAppConfig(AppConfig):
4     name = 'myapp'

```

myapp/myapp/views.py

```

1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from django.template import loader
4 from .models import Detail
5
6 def master(request):
7     template = loader.get_template("master.html")
8     context = {
9         "details": Detail.objects.all(),
10    }
11    return HttpResponse(template.render(context, request))
12
13 def detail(request, id):
14     template = loader.get_template("detail.html")
15     context = {
16         "detail": Detail.objects.get(id=id),
17    }
18    return HttpResponse(template.render(context, request))

```

myapp/myapp/models.py

```

1 from django.db import models
2
3 class Detail(models.Model):
4     id = models.AutoField(primary_key=True)
5     name = models.CharField(max_length=100)
6     quantity = models.IntegerField()
7     price = models.FloatField()
8     img = models.CharField(max_length=100)

```

myapp/myapp/templates/master.html

```

1 <!--DOCTYPE html-->
2 <html>
3     <head>
4         <meta charset="UTF-8">
5         <title>Master</title>
6         <meta name="viewport" content="width=device-width, initial-scale=1">
7         <link href="static/css/bootstrap.min.css" rel="stylesheet">
8         <script src="static/js/bootstrap.bundle.min.js"></script>
9     </head>
10    <body>
11        <div class="list-group m-5">

```

```

12         {% for item in details %}
13         <a href="{% url 'detail' item.id %}" class="list-group-item
        ↳ list-group-item-action">{{ item.name }}</a>
14         {% endfor %}
15     </div>
16 </body>
17 </html>

```

myapp/myapp/templates/detail.html

```

1 <!DOCTYPE html>
2 <meta charset="UTF-8">
3 <html>
4     <head>
5         <title>Detail</title>
6         <meta name="viewport" content="width=device-width, initial-scale=1">
7         <link href="static/css/bootstrap.min.css" rel="stylesheet">
8         <script src="static/js/bootstrap.bundle.min.js"></script>
9         <style>
10             .limited-keepaspect {
11                 max-width: 256px;
12                 max-height: 256px;
13                 width:auto;
14                 height:auto;
15             }
16         </style>
17     </head>
18     <body>
19         <div class="m-5">
20             <a class="btn btn-primary" role="button" href="{% url 'master' %}">Ha
                ↳ главную</a>
21             <h1>{{ detail.name }}</h1>
22             <table class="table table-bordered">
23                 <thead>
24                     <tr>
25                         <th scope="col">Номер</th>
26                         <th scope="col">Количество</th>
27                         <th scope="col">Цена</th>
28                         <th scope="col">Картинка</th>
29                     </tr>
30                 </thead>
31                 <tbody>
32                     <tr>
33                         <td>{{ detail.id }}</td>
34                         <td>{{ detail.quantity }}</td>
35                         <td>{{ detail.price }}</td>

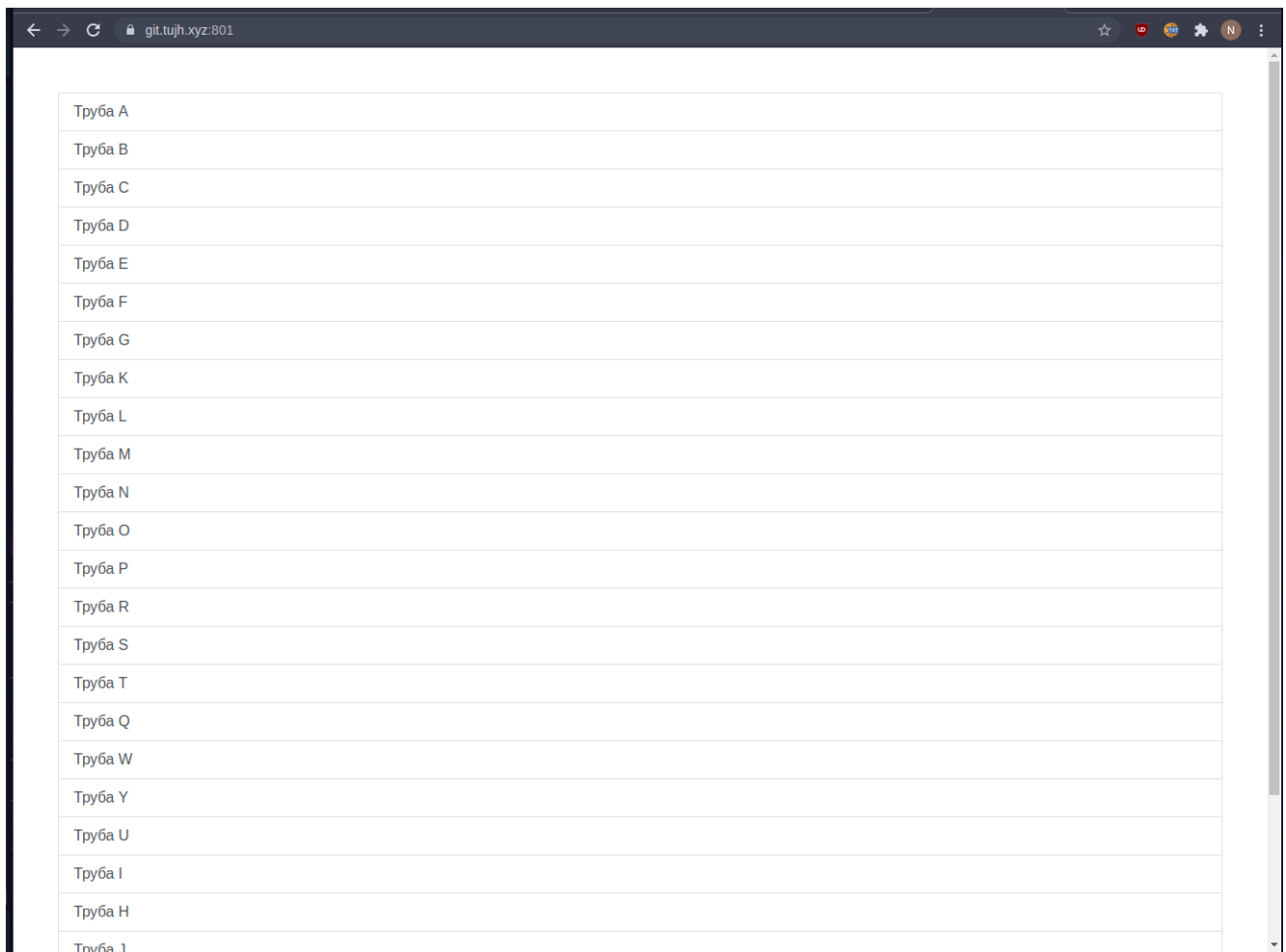
```

```

36         {% load static %}
37         <td></img></td>
38     </tr>
39 </tbody>
40 </table>
41 </div>
42 </body>
43 </html>

```

3 Экранные формы



The screenshot shows a web browser window with the address bar displaying "git.tujh.xyz:801". The main content area contains a table with 20 rows, each containing a single text entry. The entries are pipe names in Russian, starting from "Труба А" and ending with "Труба Л".

Труба А
Труба В
Труба С
Труба D
Труба E
Труба F
Труба G
Труба K
Труба L
Труба M
Труба N
Труба O
Труба P
Труба R
Труба S
Труба T
Труба Q
Труба W
Труба Y
Труба U
Труба I
Труба H
Труба J

[На главную](#)

Труба А

Номер	Количество	Цена	Картинка
0	50	200.1	