|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Разработка интернет-приложений»

Отчет по домашнему заданию

Выполнила:

студент группы ИУ5-51Б

Павловская А.А.

15.01.2020

Проверил:

|  |
| --- |
| преподаватель каф. ИУ5 |
| Гапанюк Ю.Е. |

Москва, 2020 г.

**Цель лабораторной работы:** изучение возможностей создания прототипа веб-приложения на основе базы данных с использованием фреймворка Django.

**Задание:**

Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

* Две модели, связанные отношением один-ко-многим.
* Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
* Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
* Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фрейворк по желанию студента.

**Расширенное задание:**

* Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, https://c3js.org/) и развертывание приложения на облачном сервисе (например, heroku).

**Текст программы:**

**Airport\_app/models.py**

from django.db import models

from datetime import date

from django.urls import reverse

class Plane(models.Model):

    """Самолеты"""

    name = models.CharField("Модель", max\_length=50)

    seats = models.IntegerField("Количество мест",default = 0)

    cruising\_speed = models.IntegerField("Крейсерская скорость",default = 0)

    flight\_range = models.IntegerField("Дальность полета",default = 0)

    description = models.TextField("Описание")

    def \_\_str\_\_(self):

        return self.name

    class Meta:

        verbose\_name = "Самолет"

        verbose\_name\_plural = "Самолеты"

class Flight(models.Model):

    """Рейсы"""

    departure\_airport = models.CharField("Аэропорт вылета", max\_length=100)

    data = models.DateField("Дата")

    time = models.TimeField("Время")

    status = models.TextField("Статус")

    planes = models.ManyToManyField(Plane, verbose\_name="Самолет", related\_name="plane\_flight")

    draft = models.BooleanField("Черновик", default=False)

    def \_\_str\_\_(self):

        return self.departure\_airport

    def get\_absolute\_url(self):

        return reverse("detail", kwargs={"slug": self.url})

    class Meta:

        verbose\_name = "Рейс"

        verbose\_name\_plural = "Рейсы"

**Airport\_app/urls.py**

from django.urls import path

from . import views

urlpatterns = [

    path("", views.FlightsView.as\_view()),

    path("<int:pk>/", views.FlightDetailView.as\_view(), name="flight"),

]

**Airport\_app/views.py**

from django.shortcuts import render

# Create your views here.

from django.views.generic.base import View

from .models import Flight, Plane

class FlightsView(View):

    def get(self, request):

        flights = Flight.objects.all()

        flight1 = Flight.objects.get(id=1)

        flight2 = Flight.objects.get(id=2)

        flight3 = Flight.objects.get(id=3)

        flight4 = Flight.objects.get(id=4)

        plane1 = Plane.objects.get(id=1)

        plane2 = Plane.objects.get(id=2)

        plane3 = Plane.objects.get(id=3)

        plane4 = Plane.objects.get(id=4)

        return render(request, "flights/master.html", {"master": flights, "flight1": flight1,

        "flight2": flight2,"flight3": flight3,"flight4": flight4,

         "plane1": plane1,"plane2": plane2,"plane3": plane3, "plane4": plane4

        })

class FlightDetailView(View):

    def get(self, request, pk):

        flight = Flight.objects.get(id=pk)

        return render(request, "flights/detail.html", {"flight": flight})

**Airport\_app/admin.py**

from django.contrib import admin

# Register your models here.

from .models import Flight, Plane

admin.site.register(Flight)

admin.site.register(Plane)

**Airport/urls.py**

"""Airport URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

    https://docs.djangoproject.com/en/3.1/topics/http/urls/

Examples:

Function views

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

Class-based views

    1. Add an import:  from other\_app.views import Home

    2. Add a URL to urlpatterns:  path('', Home.as\_view(), name='home')

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path("", include("Airport\_app.urls"))

]

**detail.html**

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <title>Детали рейса</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

    <style type="text/css">

   BODY {

    background: url(/static/images/Фон.png);

    background-size: cover;

   }

    table th {

        width: 30px;

        height: 30px;

    }

    table td {

        width: 30px;

        height: 30px;

    }

   </style>

</head>

<body>

<button type="button" class="btn btn-light"><a href="/">Назад</a></button>

<p>

<table class="table table-light table-hover table-bordered">

    <thead>

      <tr class = "table-success">

        <th ><h1 align="center">Аэропорт вылета:</h1></th>

        <th><h1 align="center">{{ flight.departure\_airport }}</h1></th>

      </tr>

    </thead>

    <tbody>

        <tr>

            <td align="center">Дата:</td>

            <td>{{ flight.data }}</td>

        </tr>

        <tr>

            <td align="center">Время:</td>

            <td>{{ flight.time }}</td>

        </tr>

        <tr>

            <td align="center">Статус:</td>

            <td>{{ flight.status }}</td>

        </tr>

        {% for plane in flight.planes.all %}

        <tr>

        <td class = "table-success" align="center">Самолет:</td>

        <td class = "table-success">

                {{ plane.name }}

        </td>

        </tr>

        <tr>

            <td align="center">Количество мест:</td>

            <td>

                    {{ plane.seats}}

            </td>

        </tr>

        <tr>

            <td align="center">Крейсерская скорость:</td>

            <td>

                    {{ plane.cruising\_speed }}

            </td>

        </tr>

        <tr>

            <td align="center">Дальность полета:</td>

            <td>

                    {{ plane.flight\_range }}

            </td>

        </tr>

        <tr>

        <td align="center">Описание:</td>

        <td>

                {{ plane.description }}

        </td>

        </tr>

        {% endfor %}

    </tbody>

  </table>

</p>

</body>

</html>

**master.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

    <meta charset="UTF-8">

    <title>Аэропорт</title>

    <link href="/static/css/c3.min.css" rel="stylesheet">

    <style type="text/css">

        BODY {

         background: url(/static/images/Фон2.png);

         background-size: cover;

        }

    </style>

</head>

<body>

<div class="alert alert-success" role="alert">

  <h1 align="center">Аэропорт</h1>

</div>

<p>

    <button type="button" class="btn btn-light"><a href="/admin">Администрирование</a></button>

    </p>

<div class="alert alert-warning" role="alert">

    <p align="center">Табло прилета:</p>

  </div>

{% for flight in master %}

<div class="list-group">

    <a href="{% url "flight" pk=flight.pk %}" class="list-group-item list-group-item-action">

    <p align="center">

        {{ flight.departure\_airport }}

        |

        {{ flight.data }}

        |

        {{ flight.time }}

    {% for  planes in flight.planes.all %}

        | {{ planes.name }}

    {% endfor %}

    </p>

    </a>

</div>

{% endfor %}

<p></p>

<p></p>

<p align="center">Дальность полета</p>

<div id="chart"></div>

<p align="center">Количество мест</p>

<div id="chart2"></div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>

<script src="https://d3js.org/d3.v5.min.js"></script>

<script src="/static/js/c3.min.js"></script>

<script>

var chart2 = c3.generate({

   bindto: '#chart',

    data: {

        columns: [

            ['Airbus 320',"{{ plane1.flight\_range}}" ],

            ['Boeing 737-800', "{{ plane2.flight\_range}}"],

            ['Airbus 321', "{{ plane3.flight\_range}}"],

            ['Airbus 320neo', "{{ plane4.flight\_range}}"]

        ],

        type: 'pie'

    },

    bar: {

        width: {

            ratio: 0.2

        }

    }

});

</script>

<script>

var chart2 = c3.generate({

   bindto: '#chart2',

    data: {

        columns: [

            ['Airbus 320',"{{ plane1.seats}}" ],

            ['Boeing 737-800', "{{ plane2.seats}}"],

            ['Airbus 321', "{{ plane3.seats}}"],

            ['Airbus 320neo', "{{ plane4.seats}}"]

        ],

        type: 'bar'

    },

    bar: {

        width: {

            ratio: 0.2

        }

    }

});

  </script>

</body>

</html>

**Ссылка на репозиторий Github:**

<https://github.com/PavlAA79/LabsRIP>

**Экранные формы с примерами выполнения программы.:**















 

