

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

«Радиотехнический»

Выполнил:

КАФЕДРА

ИУ-5 «Системы обработки информации и управления»

Отчет по домашнему заданию по курсу Разработка интернет-приложений

Группа:	РТ5-51Б
Дата выполнения:	«20» января 2021 г.
П	одпись:
Проверил:	Гапанюк Ю. Е
Дата проверки:	«20» января 2021 г.

Подпись:

Лисин. А. В.

Содержание

Описание задания	3
Ход работы	3
Результат работы	9

Цель лабораторной работы: изучение возможностей создания прототипа вебприложения на основе базы данных с использованием фреймворка Django.

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

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

```
Ход работы:
Текст models.py:
from diango.db import models
  Create your models here.
 class Author(models.Model):
  name = models.CharField(max_length=256, verbose_name='Имя')
birthday = models.DateField(verbose_name='Дата рождения')
  avatar = models.ImageField(verbose name='Аватар', upload to='static/img/')
       str (self):
    return self.name
   class Meta:
     verbose name = 'Автор'
     verbose name plural = 'Авторы'
     ordering = ['id']
 class Article(models.Model):
  title = models.CharField(max_length=1024, verbose_name='Заголовок') text = models.TextField(verbose_name='Текст')
  likes = models.IntegerField(verbose_name='Лайки', default=0)
  date_create = models.DateField(auto_now_add=True, verbos
  author = models.ForeignKey('Author', on de
                                                lete=models.CASCADE)
  def str (self):
     return self.title
  class Meta:
     verbose name = 'Вопрос
     verbose name plural = 'Вопросы'
class Answer(models.Model):
  text = models.TextField(ve
  likes = models.IntegerField(verbose name='Лайки', default=0)
  date create = models.DateField(auto now add=True, verbose
  author = models.ForeignKey('Author', on delete=models.CASCADE)
  correct = models.BooleanField(verbose_name='Правильно', d
  article = models.ForeignKey('Article', on delete=models.CASCADE
```

```
related guery name = "answer")
  def __str__(self):
   return self.text
  class Meta:
    verbose name = 'Ответ'
    verbose name plural = 'Ответы'
Текст admin.py:
from django.contrib import admin
from main import models
# Register your models here.
admin.site.register(models.Author)
admin.site.register(models.Article)
admin.site.register(models.Answer)
Teкст views.py:
from django.shortcuts import render
from main.models import Article
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
# Create your views here.
def paginate(objects list, request, per page=10):
  paginator = Paginator(objects list, per page)
  page number = request.GET.get('page')
  return paginator.get page(page number)
def index(request):
  questions = Article.objects.all()
  page = paginate(questions, request, 5)
  return render(request, 'index.html', {
     'page': page,
     'page_end_diff': page.paginator.num_pages - page.number,
     'user': request.user,
def settings(request):
 return render(request, 'settings.html')
def login(request):
 return render(request, 'login.html')
def question(request, question id):
  question = Article.objects.get(id=question id)
  answers = question.answers.all()
  return render(request, 'question.html', {'question': question,
                            'answers': answers
def ask(request):
 return render(request, 'ask,html')
def tag(request):
 return render(request, 'tag.html')
```

```
def registration(request):
    return render(request, 'registration.html')
```

Teкcт index.html:

Teкcт base.html:

```
{% load static %}
```

```
<main role="main" class="flex-shrink-0">
 <div class="container">
  <div class="row">
   <div class="col-9 center">
     {% block content %}
      {% endblock content %}
   </div>
   <div class="col-3 side">
    <div class="row side-menu-title">
     <div class="col-12"> Popular Tags
      <div>
         <span class="popular1">perl</span>
         <span class="popular3">python</span>
         <span class="popular1">TechnoPark</span>
         <span class="popular3">MySQL</span>
         <span class="popular3">InySQE/span/
<span class="popular2">django</span>
<span class="popular1">Mail.ru</span>
<span class="popular1">Voloshin</span>
         <span class="popular4">FireFox</span>
      </div>
     </div>
    </div>
    <div class="row side-menu-title">
     <div class="col-12"> Best Members </div>
     ul class="list">
        <a href="#">Mr. Freeman</a>
        <a href="#">Dr. House</a>
        <a href="#">Bender</a>
        <a href="#">Qeen Victoria</a>
        <|i class="list-item"><a href="#">V. Pupkin</a>>
```

```
</div>
      </div>
     </div>
      <nav aria-label="Page navigation example">
       {% if page.has previous %}
          class="page-item"><a class="page-link" href="?page=1"</li>
role="button">1</a>
          {% if page.previous page number == 2 %}
          <a class="page-link" href="?page=2" role="button">2/
a>
          {% elif page.previous page number > 2 %}
           <a class="page-link" href="?</pre>
page={{ page.previous page number }}" role="button">&lt</a>
          {% endif %}
         {% endif %}
         <a class="page-link" href="?page={{ page.number }}"</pre>
role="button">{{ page.number }}</a>
         {% if page.has next %}
          \{\% \text{ if page end diff} == 2 \%\}
           class="page-item"><a class="page-link" href="?</li>
page={{ page.next_page_number }}"
            role="button">{{ page.next_page_number }}</a>
          {% elif page_end_diff > 2 %}
           <a class="page-link" href="?</pre>
page={{ page.next page number }}" role="button">&gt</a>
          {% endif %}
          <a class="page-link" href="?</pre>
page={{ page.paginator.num pages }}"
          role="button">{{ page.paginator.num pages }}</a>
         {% endif %}
       </nav>
</div>
   <script src="../static/js/jquery.slim.min.js"></script>
   <script src="../static/js/popper.min.js"></script>
   <script src="../static/css/bootstrap.min.js"></script>
  </body>
</html>
Teкcт one_question.html:
<div class="question-preview">
  <div class="row">
  <div class="col-2 avatar-and-likes">
      <img class="avatar" src="{{ question.author.avatar.url }}" />
    </div>
    <div class="col-6">
      <a href = "{% url 'question' question.id %}" class = "navbar-brand,"
title">{{question.title}}</a>
    </div>
  </div>
  <div class="row">
    <div class="col-2 avatar-and-likes">
      <div class="list-group like-button">
```

```
<a href="#" class="list-group-item list-group-item-action text-center likes">▲</a>
       <div class="list-group-item text-center text-truncate likes">10</div>
       <a href="#" class="list-group-item list-group-item-action text-center likes">▼</a>
      </div>
    </div>
    <div class="col-6">
      {{question.text}}
    </div>
 </div>
 <div class="row">
    <div class="col-2 answer-bottom">
      <a href="#" class = "text">{{question.author.name}}</a>
   </div>
   <div class="col-7 answer-bottom">
      Tags: 
      <a href="#" class = "text">Black-jack</a>
      <a href="#" class = "text">Bender</a>
    </div>
 </div>
</div>
```

Текст question.html:

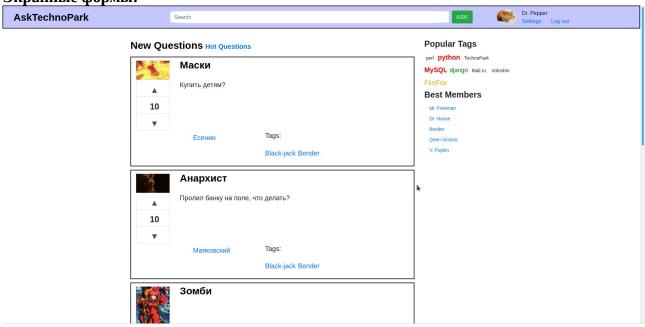
```
{% extends 'base.html' %}
{% block content %}
        <div class="question-preview">
           <div class="row">
             <div class="col-2 avatar-and-likes">
               <img class="avatar" src="{{ question.author.avatar.url }}" />
             </div>
             <div class="col-6">
               <h3 class = "navbar-brand, title">{{question.title}}</h3>
             </div>
           </div>
           <div class="row">
             <div class="col-2 avatar-and-likes">
                <div class="list-group like-button">
                <a href="#" class="list-group-item list-group-item-action text-center"
likes">▲</a>
                <div class="list-group-item text-center text-truncate likes">10</div>
                <a href="#" class="list-group-item list-group-item-action text-center"
likes">▼</a>
               </div>
             </div>
             <div class="col-6">
                { {question.text} } 
             </div>
           </div>
           <div class="row">
             <div class="col-2 answer-bottom">
               <a href="#" class = "text">{{question.author.name}}</a>
             </div>
             <div class="col-7 answer-bottom">
               Tags: 
               <a href="#" class = "text">Black-jack</a>
               <a href="#" class = "text">Bender</a>
             </div>
           </div>
        </div>
   <hr class = "separator">
```

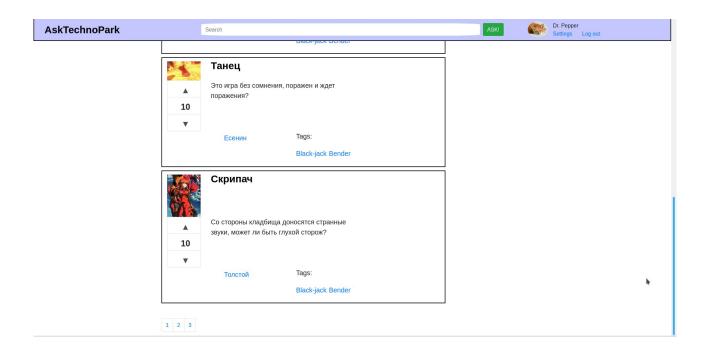
```
{% for answer in answers %}
         <div class = "col-12">
         <div class="question-preview">
           <img class="avatar" src="{{ answer.author.avatar.url }}" />
             </div>
           </div>
           <div class="row">
             <div class="col-2 avatar-and-likes">
                <div class="list-group like-button">
                <a href="#" class="list-group-item list-group-item-action text-center"
likes">▲</a>
                 <div class="list-group-item text-center text-truncate likes">10</div>
                <a href="#" class="list-group-item list-group-item-action text-center"
likes">▼</a>
                </div>
             </div>
             <div class="col-6">
                {{answer.text}}
             </div>
           </div>
           <div class="row">
             <div class="col-2 answer-bottom">
                <a href="#" class = "text">{{answer.author.name}}</a>
             </div>
             <div class="form-check check-bottom">
                  <input class="form-check-input" type="checkbox" value="" required>
                   Correct!
             </div>
           </div>
         </div>
         </div>
         {% endfor %}
{% endblock %}
Текст urls.py:
from django.urls import path
from django.conf.urls import url
from , import views
urlpatterns = [
  path(", views.index),
  path('settings', views.settings),
  path('login', views.login),
  path('question/<int:question_id>/', views.question, name='question'),
  path('ask', views.ask),
  path('tag', views.tag),
```

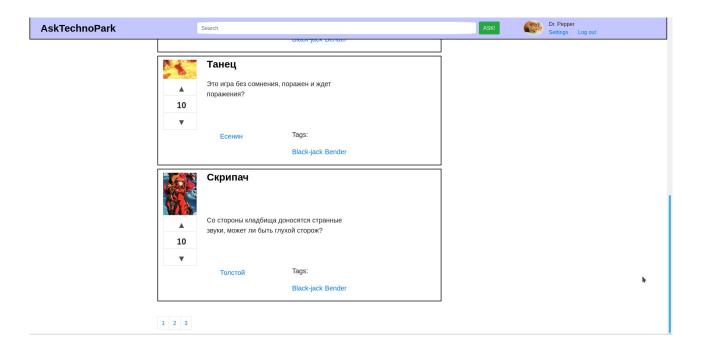
path('registration', views.registration)

Результат работы:

Экранные формы:







Django administration

Site administration

