**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по Домашнему заданию

## «Создание сайта, используя фреймворк django»

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил: |  | Проверил: | |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 | |
| Паншин М.В. |  | Гапанюк Ю.Е. | |
|  |  | |  |

Москва, 2024 г.

### **Постановка задачи**

В качестве домашнего задания предлагается выполнить проект «Вопросы и Ответы». Этот сервис позволит пользователям Интернета задавать вопросы и получать на них ответы. Возможности комментирования и голосования формируют сообщество и позволяет пользователям.

**Требования к проекту**

1. Структура проекта должна быть понятна пользователям. Переходы по страницам осуществляются по ссылкам. Обработка форм должна осуществляться с редиректом.
2. Код проекта должен быть аккуратным и без дублирования.
3. Верстка проекта должна быть выполнена с помощью css фреймворка Twitter Bootstrap.
4. Страницы проекты не должны отдаваться более 1 секунды.

**Текст программы**

**Файл Views.py**

from django.http import HttpResponse

from django.shortcuts import render

import copy

from django.core.paginator import Paginator

from app import models

def pagination(list, request, per\_page = 5):

    page\_num = request.GET.get('page',1)

    try:

        page\_num = int(page\_num)

    except ValueError:

        page\_num = 1

    paginator = Paginator(list,per\_page)

    page = paginator.page(page\_num)

    elided\_page\_range = page.paginator.get\_elided\_page\_range(number = page.number,on\_each\_side = 1, on\_ends = 1)

    return {'list': page.object\_list,

                   'page\_obj' : page,

                   'elided\_page\_range' : elided\_page\_range,

                   'ELLIPSIS' : paginator.ELLIPSIS}

def index (request):

    QUESTIONS = models.Question.objects.create\_list()

    return render(request, template\_name = 'index.html',

        context = pagination(QUESTIONS,request,5)

        )

def hot(request):

    HOT\_QUESTIONS = models.Question.objects.create\_list(sort\_by = "likes")

    return render(request,template\_name = 'hot.html',

                  context = pagination(HOT\_QUESTIONS,request,5))

def one\_question(request, question\_id):

    ANSWERS = models.Answer.objects.create\_list(question\_id = question\_id)

    context = pagination(ANSWERS,request,per\_page = 5)

    context['question'] = models.Answer.objects.get\_question(question\_id = question\_id)

    return render(request, template\_name = 'one\_question.html',

                  context = context )

def ask(request):

    return render(request, template\_name= 'ask.html'

                  )

def tag(request,tag):

    QUESTIONS = models.Question.objects.create\_list(tag = tag)

    context = pagination(QUESTIONS,request,10)

    context['tag'] = str(tag)

    return render(request,  template\_name='tag.html',

                  context = context)

def profile(request):

    return render(request,template\_name="profile.html",

                  )

def login(request):

    return render(request,template\_name="login.html",

                  )

def sigup(request):

    return render(request,template\_name="signup.html",)

**Файл Models.py**

from django.db import models

from django.contrib.auth.models import User

from django.db.models import Count

class Profile(models.Model):

    user = models.OneToOneField(User,on\_delete=models.CASCADE)

    nickname = models.CharField(max\_length = 255)

    avatar = models.ImageField(blank = True, null = True)

    email = models.EmailField(max\_length=255)

    created\_at = models.DateTimeField(auto\_now\_add = True)

    updated\_at = models.DateTimeField(auto\_now = True)

    def \_\_str\_\_(self):

        return self.nickname

class TagManager(models.Manager):

    def get\_popular\_tags(self, number = 8):

        tags = Tag.objects.all().annotate(question\_count = Count("question")).order\_by('-question\_count')

        return [{'name':tag.name } for tag in tags[:number]]

class Tag(models.Model):

    name = models.CharField(max\_length = 50)

    created\_at = models.DateTimeField(auto\_now\_add = True)

    updated\_at = models.DateTimeField(auto\_now = True)

    objects = TagManager()

    def \_\_str\_\_(self):

        return self.name

class QuestionManager(models.Manager):

    def create\_list(self, sort\_by = 'created\_at',tag = "none"):

        if sort\_by == 'likes':

            ordering = '-like\_count'

        else:

            ordering = '-created\_at'

        if tag == 'none':

            questions = (Question.objects.all().annotate(like\_count = Count('questionlike')).order\_by(ordering))

        else:

            questions = (Question.objects.filter(tags\_\_name = tag).annotate(like\_count = Count('questionlike')).order\_by(ordering))

        return [

        {

           'author' : question.author.nickname,

            'likes' :question.like\_count,

            "title": question.title,

            "id": question.id,

            "text": question.text,

            "tags": list(question.tags.all().values\_list('name',flat = True)),

        } for question in questions.filter(author\_\_nickname\_\_startswith = "User")

  ]

class Question(models.Model):

    title = models.CharField(max\_length = 255)

    text = models.CharField(max\_length = 1024)

    author = models.ForeignKey(Profile, on\_delete=models.CASCADE)

    tags = models.ManyToManyField(Tag)

    created\_at = models.DateTimeField(auto\_now\_add = True)

    updated\_at = models.DateTimeField(auto\_now = True)

    objects = QuestionManager()

    def \_\_str\_\_(self):

        return self.title

class AnswerManager(models.Manager):

    def create\_list(self,question\_id):

        question = Question.objects.get(id=question\_id)

        answers = (Answer.objects.filter(question = question).annotate(like\_count = Count('answerlike')).order\_by('-like\_count'))

        return [

        {

           'author' : answer.author.nickname,

            'likes' :answer.like\_count,

            "id": answer.id,

            "text": answer.text,

            "status" : answer.status,

            'question': question,

        } for answer in answers.filter(author\_\_nickname\_\_startswith = "User")

        ]

    def get\_question(self,question\_id):

        question = (Question.objects.all().annotate(like\_count = Count('questionlike'))).get(id = question\_id)

        return {

           'author' : question.author.nickname,

            'likes' :question.like\_count,

            "title": question.title,

            "id": question.id,

            "text": question.text,

            "tags": list(question.tags.all().values\_list('name',flat = True)),

        }

class Answer(models.Model):

    STATUS\_CHOISES = [

        ('c','Corrrect'),

        ('w', 'Wrong'),

    ]

    text = models.TextField(max\_length = 1024)

    author = models.ForeignKey(Profile, on\_delete=models.CASCADE)

    question = models.ForeignKey(Question, on\_delete=models.CASCADE)

    status = models.CharField(max\_length = 10, choices = STATUS\_CHOISES)

    created\_at = models.DateTimeField(auto\_now\_add = True)

    updated\_at = models.DateTimeField(auto\_now = True)

    objects = AnswerManager()

    def \_\_str\_\_(self):

        return f"Answer by {self.author.nickname} to {self.question.title}"

class AnswerLike(models.Model):

    answer = models.ForeignKey(Answer, on\_delete=models.CASCADE)

    user = models.ForeignKey(Profile, on\_delete=models.CASCADE)

    created\_at = models.DateTimeField(auto\_now\_add = True)

    updated\_at = models.DateTimeField(auto\_now = True)

    class Meta:

        unique\_together = ('user','answer')

class QuestionLike(models.Model):

    question = models.ForeignKey(Question, on\_delete=models.CASCADE)

    user = models.ForeignKey(Profile, on\_delete=models.CASCADE)

    created\_at = models.DateTimeField(auto\_now\_add = True)

    updated\_at = models.DateTimeField(auto\_now = True)

    class Meta:

        unique\_together = ('user','question')

**Файл fill\_db.py**

from app import models

from django.core.management.base import BaseCommand

from django.contrib.auth.models import User

import random

class Command(BaseCommand):

    def add\_arguments(self, parser):

        parser.add\_argument('--ratio', type=int)

        super().add\_arguments(parser)

    def handle(self, \*args, \*\*options):

        ratio = options['ratio']

        users = [User(username=f'User {i}', email=f'user{i}@gmail.com') for i in range(ratio)]

        User.objects.bulk\_create(users)

        users = User.objects.filter(username\_\_startswith='User')

        profiles = [models.Profile(user=user, nickname=user.username, email=user.email) for user in users]

        models.Profile.objects.bulk\_create(profiles)

        profiles = list(models.Profile.objects.all())

        tags = [models.Tag(name=f'Tag #{i}') for i in range(ratio)]

        models.Tag.objects.bulk\_create(tags)

        tags = list(models.Tag.objects.all())

        questions = [

            models.Question(

                title=f'Title for question {i}',

                text=f'This is text for question {i}',

                author=profiles[i % ratio],

            ) for i in range(ratio \* 10)

        ]

        models.Question.objects.bulk\_create(questions)

        questions = list(models.Question.objects.select\_related('author').prefetch\_related('tags').all())

        tags\_len = len(tags)

        for item in range(len(questions)) :

            selected\_tags = [tags[item % tags\_len],tags[(item + 1) % tags\_len],tags[(item + 1) % tags\_len]]

            questions[item].tags.set(selected\_tags)

        for j in range (10):

            answers = [

                models.Answer(

                    text=f'This is text for answer {i}',

                    author=profiles[i % ratio],

                    question=questions[i % (ratio \* 10)],

                    status='c' if i % 10 == 4 else 'w',

                ) for i in range(j \* ratio \* 10,(j+1) \* ratio \* 10)

            ]

            models.Answer.objects.bulk\_create(answers)

            print(f"{j} part of answers has been uploaded")

        answers = list(models.Answer.objects.select\_related('author', 'question').all())

        question\_likes = [

            models.QuestionLike(

                user=profiles[i // (ratio \* 10)],

                question=questions[i % (ratio \* 10)],

            ) for i in range(ratio \* 9 \* 10)

        ]

        models.QuestionLike.objects.bulk\_create(question\_likes,batch\_size = 1000)

        answer\_likes = [

            models.AnswerLike(

                user=profiles[i // (ratio \* 100)],

                answer=answers[i % (ratio \* 100)],

            ) for i in range(ratio \* 10 \* 13)

        ]

        models.AnswerLike.objects.bulk\_create(answer\_likes,batch\_size = 1000)

        question\_like\_ids = list(models.QuestionLike.objects.values\_list('id', flat=True))

        random.shuffle(question\_like\_ids)

        random\_likes\_ids = question\_like\_ids[:ratio \* 10]

        models.QuestionLike.objects.filter(id\_\_in=random\_likes\_ids).delete()

**Файл answer.html**

{% load static %}

<div class="row mt-5 mb-5">

    <div class="card mb" style="border: 2px solid black;">

      <div class="row">

        <div class="col-2">

            <div class = "d-flex flex-column">

            <div style="width: 120px; height: auto;" class = "m-1 mt-3"  >

                <img src="{% static 'img/i.jpg' %}" alt="Profile pic" class="img-thumbnail" >

            </div>

          <div class = "d-flex justify-content-center m-2">

            <h5>{{answer.author}}</h5>

        </div>

          <input class="form-control mb-2" type="number" value = {{answer.likes}} placeholder="Likes" aria-label="default input example">

        </div>

    </div>

          <div class="col-9">

            <div class="d-flex flex-column justify-content-between" style = "height: 100%;">

              <p class="card-text mt-2">{{answer.text}}  </p>

              <div class="form-check mb-3">

                <input class="form-check-input" {% if answer.status == 'c' %} checked {%endif%} type="checkbox" value="" id="flexCheckDefault">

                <label class="form-check-label" for="flexCheckDefault">

                  Correct !

                </label>

            </div>

          </div>

        </div>

      </div>

    </div>

  </div>

**Файл base.html**

{% load static %}

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>base</title>

    <link rel="stylesheet" href={% static "css/bootstrap.min.css" %}>

    <style>

    </style>

  </head>

  <body>

    <header>

        <div class="d-flex align-items-center justify-content-between mb-5 p-3" style = "border-bottom: 1px solid black;">

          <a href="{% url "index" %}" class="badge rounded-pill text-bg-success m-3" style="font-size: 30px; text-decoration: none;">

            AskPupkin

          </a>

              <form class="d-flex" role="search">

                <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

                <a class="btn btn-outline-success" type="submit" href = "{% url "ask" %}" >Ask</a>

              </form>

            <div class="d-flex gap-2">

               {% block profile %}

               {% endblock profile %}

            </div>

        </div>

    </header>

    <main class="container mt-3">

        <div class="row">

              <div class="col-9 px-5">

                   {% block content %}

                   {% endblock  %}

              </div>

              <div class="col-3">

                  <section class="mb-2">

                      <h2>Popular Tags</h2>

                      <div class= "d-flex flex-wrap gap-2">

                        {% for tag in popular\_tags %}

                          <a href = "{% url "tag" tag=tag.name %}"type="button" class="btn btn-outline-success">{{tag.name}}</a>

                        {% endfor %}

                      </div>

                  </section>

                  <section>

                      <h2>Best Members</h2>

                      <ul class="list-group list-group-flush">

                          <li class="list-group-item"><a href="#">Member 1</a></li>

                          <li class="list-group-item"><a href="#">Member 2</a> </li>

                          <li class="list-group-item"><a href="#">Member 3</a></li>

                          <li class="list-group-item"><a href="#">Member 4</a> </li>

                          <li class="list-group-item"><a href="#">Member 5</a></li>

                        </ul>

                  </section>

              </div>

        </div>

    </main>

    <script src= {% static "js/bootstrap.min.js" %}>

    </script>

  </body>

</html>

**Файл pagination.py**

{% load static %}

<nav aria-label="Page navigation ">

    <ul class="pagination justify-content-center">

      <li class="page-item {% if not page\_obj.has\_previous %} disabled {% endif %}">

        <a class="page-link" href="{% if page\_obj.has\_previous %} ?page={{page\_obj.previous\_page\_number}} {% endif %}" aria-label="Previous">

          <span aria-hidden="true">&laquo;</span>

        </a>

      </li>

      {% for p in elided\_page\_range %}

            {% if p == ELLIPSIS %}

                <li class="page-item   disabled ">

                    <a class="page-link" href="?page={{ p }}">{{ p }}</a>

                </li>

            {% else %}

            <li class="page-item {% if page\_obj.number == p %} active {% endif %}">

                <a class="page-link" href="?page={{ p }}">{{ p }}</a>

            </li>

            {% endif %}

        {% endfor %}

      <li class="page-item {% if not page\_obj.has\_next %}disabled{% endif %}">

        <a class="page-link" href="{% if page\_obj.has\_next %} ?page={{page\_obj.next\_page\_number}} {% endif %}" aria-label="Next">

          <span aria-hidden="true">&raquo;</span>

        </a>

      </li>

    </ul>

  </nav>

**Файл question.html**

{% load static %}

<div class="card mb-3">

    <div class="row">

      <div class="col-2">

        <div class = "d-flex flex-column" style = "height: 100%;">

            <div style="height: auto; width: 120px;" class="m-1 mt-3">

              <img src="{% static 'img/i2.jpg' %}" class="img-thumbnail" alt="Avatar">

            </div>

            <div class = "m-2" style="text-align: center;">

                <h5>{{question.author}}</h5>

            </div>

                <input class="form-control  my-3" type="number" value = {{question.likes}} placeholder="Likes" aria-label="default input example">

        </div>

      </div>

      <div class="col-9">

        <div class="card-body d-flex flex-column" style="padding-top: 0; height: 100%;">

          <a class="card-title" href="{% url "one\_question" question.id %}" style="font-size: 30px;">{{ question.title }}</a>

          <p class="card-text ">{{ question.text}} </p>

          <div class="d-flex gap-5 mt-auto">

            <a href="{% url "one\_question" question.id %}" class="card-link"> Answers (10) </a>

            <div>

              Tags:{% for tag in question.tags  %}

                <a role="button" class="btn btn-sm btn-success" style = "background-color: #" href = "{% url "tag" tag=tag %}">{{tag}}</a>

                {% endfor %}

            </div>

          </div>

        </div>

      </div>

    </div>

  </div>

**Файл ask.html**

 {% extends "layouts/base.html" %}

 {% block profile %}

  {% include "layouts/authorized.html" %}

{% endblock profile %}

{% block content %}

<div >

    <h2>New Question</h2>

</div>

<div class="container mt-4">

    <form class="d-flex flex-column gap-3">

      <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Title</label>

        <div class="col-sm-10">

          <input type="text" class="form-control" id="questionTitle" placeholder="Enter question title">

        </div>

      </div>

      <div class="row mb-3">

        <label for="questionText" class="col-sm-2 col-form-label">Text</label>

        <div class="col-sm-10">

          <textarea class="form-control" id="questionText" rows="5" placeholder="Write your question here"></textarea>

        </div>

      </div>

      <div class="row mb-3 d-flex align-items-center">

        <label for="tags" class="col-sm-2 col-form-label">Tags</label>

        <div class="col-sm-10">

          <input type="text" class="form-control" id="tags" placeholder="Add tags here, using a comma ">

        </div>

      </div>

      <div class="row">

        <div class="col-sm-2"></div>

        <div class="col-sm-10">

            <a class="btn btn-outline-success" type="submit" href="#" >Ask</a>

        </div>

      </div>

    </form>

  </div>

{% endblock content %}

**Файл hot.html**

{% extends "layouts/base.html" %}

{% load static %}

{% block profile %}

  {% include "layouts/not\_authorized.html" %}

{% endblock profile %}

{% block content %}

    <div class="d-flex gap-3 align-items-center mb-4 ">

        <div class = "d-flex align-items-center gap-3">

        <h2> Hot questions</h2>

        <a href ="{% url "index" %}"> New Questions</a>

        </div>

    </div>

    {% for question in list %}

        {% include "layouts/question.html" %}

    {% endfor %}

    {% include "layouts/pagination.html" %}

{% endblock content %}

**Файл index.html**

{% extends "layouts/base.html" %}

{% load static %}

{% block profile %}

  {% include "layouts/not\_authorized.html" %}

{% endblock profile %}

{% block content %}

<div class="d-flex gap-3 align-items-center mb-4 " >

    <div class = "d-flex align-items-center gap-3">

      <h2> New Questions</h2>

      <a href ="{% url "hot" %}"> Hot questions</a>

    </div>

</div>

{% for question in list %}

  {% include "layouts/question.html" %}

{% endfor %}

{% include "layouts/pagination.html" %}

{% endblock content %}

**Файл login.html**

{% extends "layouts/base.html" %}

{% block profile %}

  {% include "layouts/not\_authorized.html" %}

{% endblock profile %}

{% block content %}

<div >

    <h2>Log In</h2>

</div>

  <div class="container mt-4">

    <form class="d-flex flex-column gap-3">

      <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Login</label>

        <div class="col-sm-4">

            <input type="text" class="form-control" required placeholder = "Enter login">

        </div>

      </div>

     <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Password</label>

        <div class="col-sm-4">

            <input type="password" class="form-control" id="password" minlength="8" required>

        </div>

      </div>

      <div class="row">

        <div class="col-sm-2"></div>

        <div class="col-sm-4">

            <div class = "d-flex justify-content-center">

            <a class="btn btn-success" type="submit" href="#" >Log In</a>

            </div>

        </div>

      </div>

      <div class="row">

        <div class="col-sm-2"></div>

        <div class="col-sm-4">

            <div class = "d-flex justify-content-center">

            <a class="btn btn-sm btn-outline-success" type="submit" href="#" >Create new account</a>

            </div>

        </div>

      </div>

    </form>

  </div>

{% endblock content %}

**Файл one\_question.html**

{% extends "layouts/base.html" %}

{% block profile %}

  {% include "layouts/authorized.html" %}

{% endblock profile %}

{% block content %}

<div class="row" style="border-bottom: 1px solid black;">

    {% include 'layouts/question.html' %}

</div>

{% for answer in list %}

  {% include "layouts/answer.html" %}

{% endfor %}

<div class="row d-flex align-items-center pt-5 mb-4" style="border-top: 1px solid black;">

          <textarea class="form-control" id="questionText" rows="5" style="resize: none;" placeholder="Enter your answer here.."></textarea>

</div>

{% include "layouts/pagination.html" %}

 {% endblock content %}

**Файл profile.html**

{% extends "layouts/base.html" %}

{% block profile %}

  {% include "layouts/authorized.html" %}

{% endblock profile %}

{% block content %}

<div >

    <h2>Settings: Dr.Pepper</h2>

</div>

<div class="container mt-4">

    <form class="d-flex flex-column gap-3">

      <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Login</label>

        <div class="col-sm-4">

          <input type="text" class="form-control" id="questionTitle" placeholder="Enter Login">

        </div>

      </div>

     <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Email</label>

        <div class="col-sm-4">

          <input type="text" class="form-control" id="questionTitle" placeholder="Enter Email">

        </div>

      </div>

      <div class="row mb-3 d-flex align-items-center">

        <label for="tags" class="col-sm-2 col-form-label">NickName</label>

        <div class="col-sm-4">

          <input type="text" class="form-control" id="tags" placeholder="Write youre nickname here">

        </div>

      </div>

      <div class="row mb-3 d-flex ">

        <label for="questionTitle" class="col-sm-2 col-form-label">Upload Avatar</label>

        <div class="col-sm-6 d-flex flex-column" >

            <div style="height: 150px; width: 150px;" class="m-2 border border-2 rounded-3">

                <img src="" class="img-fluid" alt="Avatar">

              </div>

              <div class="input-group d-flex gap-3">

                <input type="file" class="form-control" id="inputGroupFile04" aria-describedby="inputGroupFileAddon04" aria-label="Upload">

                <button class="btn btn-outline-secondary" type="button" id="inputGroupFileAddon04">Choose</button>

              </div>

        </div>

      </div>

      <div class="row">

        <div class="col-sm-2"></div>

        <div class="col-sm-10">

            <a class="btn btn-outline-success" type="submit" href="#" >Save</a>

        </div>

      </div>

    </form>

  </div>

{% endblock content %}

**Файл signup.html**

{% extends "layouts/base.html" %}

{% block profile %}

  {% include "layouts/not\_authorized.html" %}

{% endblock profile %}

{% block content %}

<div >

    <h2>Settings: Dr.Pepper</h2>

</div>

<div class="container mt-4">

    <form class="d-flex flex-column gap-3">

      <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Login</label>

        <div class="col-sm-4">

          <input type="text" class="form-control" required minlength = 5 placeholder="Enter Login">

        </div>

      </div>

     <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Email</label>

        <div class="col-sm-4">

          <input type="text" class="form-control" required minlength = 10 placeholder="Enter Email">

        </div>

      </div>

      <div class="row mb-3 d-flex align-items-center">

        <label for="tags" class="col-sm-2 col-form-label">NickName</label>

        <div class="col-sm-4">

          <input type="text" class="form-control" required minlength = 5 placeholder="Write youre nickname here">

        </div>

      </div>

      <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Password</label>

        <div class="col-sm-4">

            <input type="password" class="form-control"  minlength="8" required>

            <div class="invalid-feedback">

              Your password must be at least 8 characters long.

            </div>

        </div>

      </div>

      <div class="row mb-3 d-flex align-items-center">

        <label for="questionTitle" class="col-sm-2 col-form-label">Repeat password</label>

        <div class="col-sm-4">

            <input type="password" class="form-control"  minlength="8" required>

        </div>

      </div>

      <div class="row mb-3 d-flex ">

        <label for="questionTitle" class="col-sm-2 col-form-label">Upload Avatar</label>

        <div class="col-sm-6 d-flex flex-column" >

              <div class="input-group d-flex gap-3">

                <input type="file" class="form-control"  aria-label="Upload">

                <button class="btn btn-outline-secondary" type="button" id="inputGroupFileAddon04">Choose</button>

              </div>

        </div>

      </div>

      <div class="row">

        <div class="col-sm-2"></div>

        <div class="col-sm-4">

            <div class = "d-flex justify-content-center">

            <button class="btn btn-success" type="submit" >Register</button>

            </div>

        </div>

      </div>

      <div class="row">

        <div class="col-sm-2"></div>

        <div class="col-sm-4">

            <div class = "d-flex justify-content-center">

            <a class="btn btn-sm btn-outline-success"  href="{% url "login" %}" >Already have an account? Log in </a>

            </div>

        </div>

      </div>

    </form>

  </div>

{% endblock content %}

**Файл tag.html**

{% extends "layouts/base.html" %}

{% load static %}

{% block profile %}

  {% include "layouts/authorized.html" %}

{% endblock profile %}

{% block content %}

<div class="d-flex gap-3 align-items-center mb-4 ">

    <h2>Tag: {{tag}}<h2>

</div>

{% for question in list %}

  {% include "layouts/question.html" %}

{% endfor %}

{% include "layouts/pagination.html" %}

{% endblock content %}

**Результаты программы**



















