



Системы обработки изображений в медицине



Климочкина
Ксения Б19-302

Постановка задачи

Основной задачей моего проекта является отправка изображений на отладочный веб-сервер Django с графическими примитивами

Использованные технологии

- HTML - интерфейс сайта
- JavaScript - отрисовка графических примитивов
- Django - развертывание веб-сервера

Интерфейс

Основные компоненты:

- Кисть для нанесения графических примитивов
- Кнопка очистки нанесенных примитивов
- Кнопка загрузки изображения
- Кнопка загрузки изображения на сервер



Работа программы

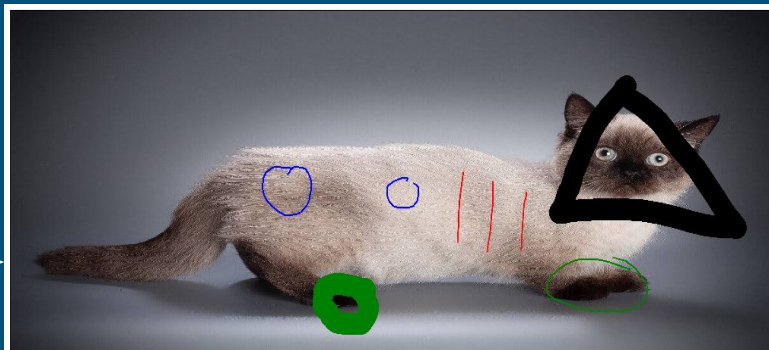
Нанесение графических примитивов различных цветов и размеров



Размер кисти:

Цвет: ☒ Черный ☐ Белый ☐ Красный ☐ Зеленый ☐ Синий

d1fd5d6e...6a7fc.jpeg



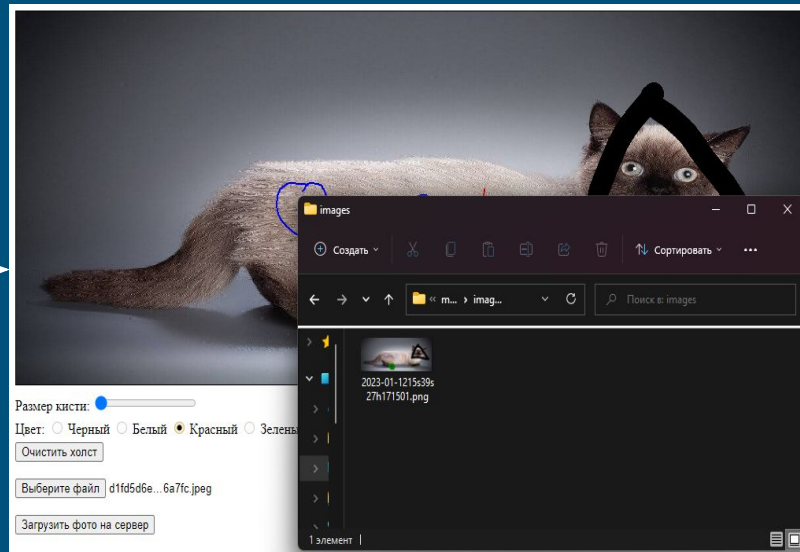
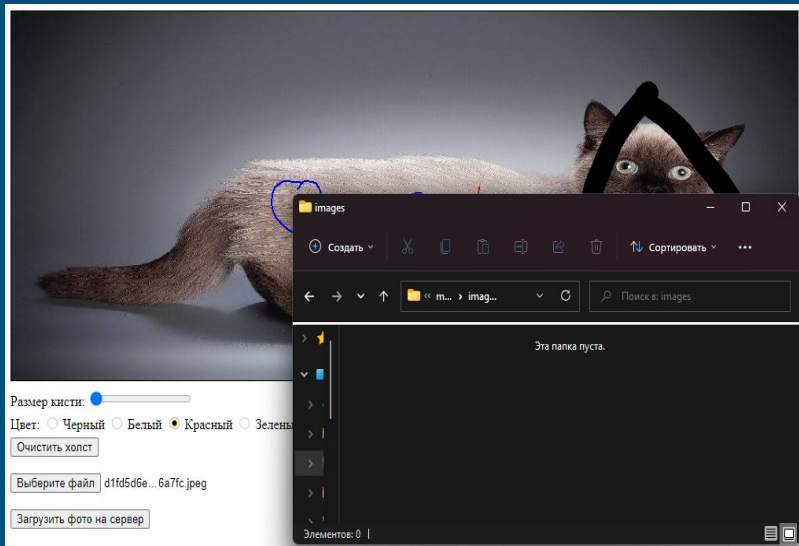
Размер кисти:

Цвет: ☐ Черный ☐ Белый ☒ Красный ☐ Зеленый ☐ Синий

d1fd5d6e...6a7fc.jpeg

Работа программы

Сохранение нового изображения



Настройки settings.py

Добавлено приложение 'page' для работы решения

Подключена папка 'static' для использования JavaScript

Подключена папка 'media' для сохранения изображений

```
INSTALLED_APPS = [
```

```
...
```

```
    'page',
```

```
]
```

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR ,  
"static"),  
]
```

```
STATIC_URL = 'static/'
```

```
MEDIA_URL = "/media/"
```

```
MEDIA_ROOT =
```

```
os.path.join(BASE_DIR , 'media')
```

Настройка urls.py

Настроена
маршрутизация по
сайту

```
from django.conf import settings
from django.conf.urls.static
import static
from django.contrib import admin
from django.urls import path,
include
```

```
urlpatterns = [
    path('admin/',
admin.site.urls),
    path('', include('page.urls'))
] + static(settings.MEDIA_URL ,
document_root=settings.MEDIA_ROO
T)
```


Маршрутизация

Настроена маршрутизация в самом приложении

Используются два адреса:

‘page’ используется для отображения интерфейса приложения

‘save pic’ используется для сохранения изображения на сервере

```
from django.urls import path
from .views import save_pic, page
```

```
urlpatterns = [
    path('page/', page,
name="images"),
    path('save pic/', save_pic,
name='save_pic'),
]
```

Файл views.py

Функция 'save pic' принимает изображение через POST запрос в формате dataURL и декодирует его в изображение и сохраняет в отдельную папку с индивидуальным названием .

Функция 'page' отображает page.html

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.views.decorators.csrf import csrf_exempt
from base64 import b64decode
import datetime

@csrf_exempt
def save_pic(request):
    if request.method == 'POST':
        data_uri = request.POST['p']
        header, encoded = data_uri.split(",", 1)
        data = b64decode(encoded)
        name =
"C:/h3/media/images/"+str(datetime.datetime.now()).replace(" ", "").replace(".", "h").replace(":", "s")+".png"
        with open(name, "wb") as f:
            f.write(data)
        return HttpResponseRedirect("GOOD!", status=200)

def page(request):
    return render(request, 'files/page.html')
```

Программа создает полотно, на котором прорисовываются графические примитивы и на которое загружается изображение.

Далее представлено описание настройки кисти и добавление функциональных кнопок.

Файл draw.js

В этой файле находятся функции отрисовки и загрузки изображения.

На слайд вынесена часть кода, который отвечает за отправку изображения с графическими примитивами на веб - сервер.

```
function save() {  
  const   
  canvas=document.getElementById('canvas');  
  const imageUri = canvas.toDataURL();  
  console.log(imageUri);  
  $.ajax({  
    type: 'POST',  
    url: '/save_pic/',  
    data: {  
      'p': imageUri  
    },  
    success: function(response) {  
      console.log(response)  
    },  
    error: function(response) {  
      console.log(response)  
    }  
  })  
}
```

Ссылка на - > [github](#)

СПАСИБО ЗА ВНИМАНИЕ !!

