Flask application documentation

-Dev Nandan P(22MID0227)

#activity done on VS code and all necessary packages have been installed which include flask and creating of a virtual environment to run flask

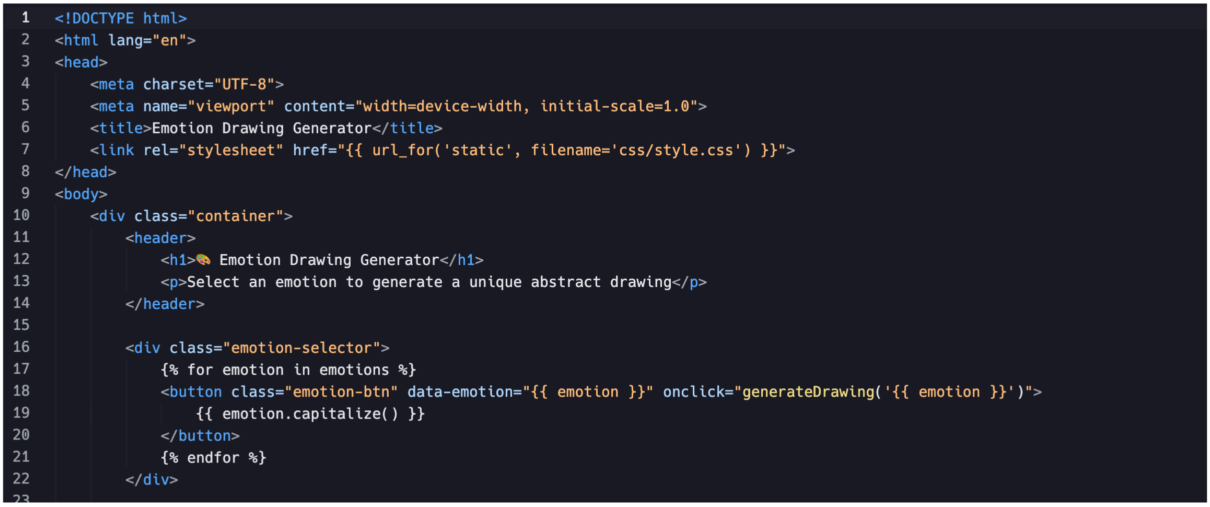
Application brief:

Using flask packages I have created and application that generates an image based on the selected emotion

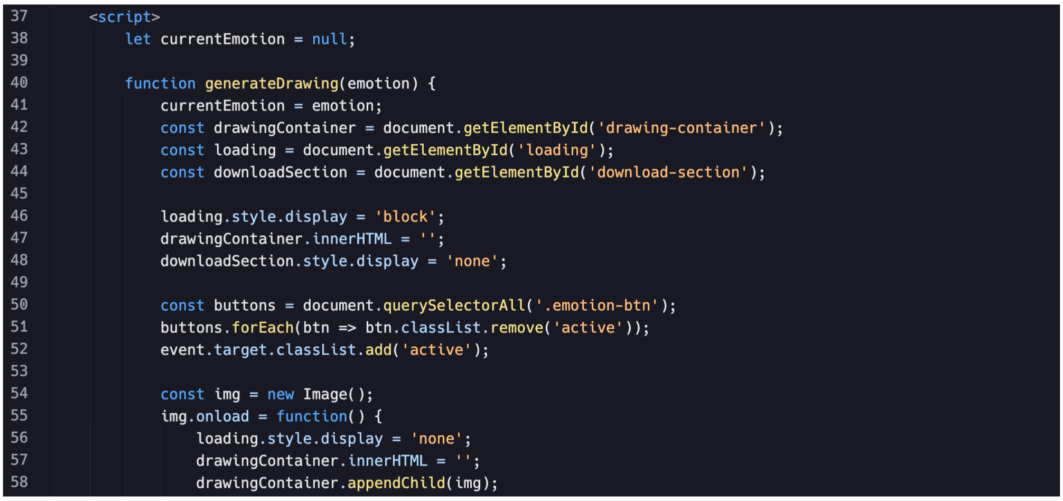
The generation of image happens based on the set of constraints that have been given example excitement shows spread of lot of stars and anger shows red triangles scattered around

Finally the application also lets you download the image in png format

Html code preview(index.html):

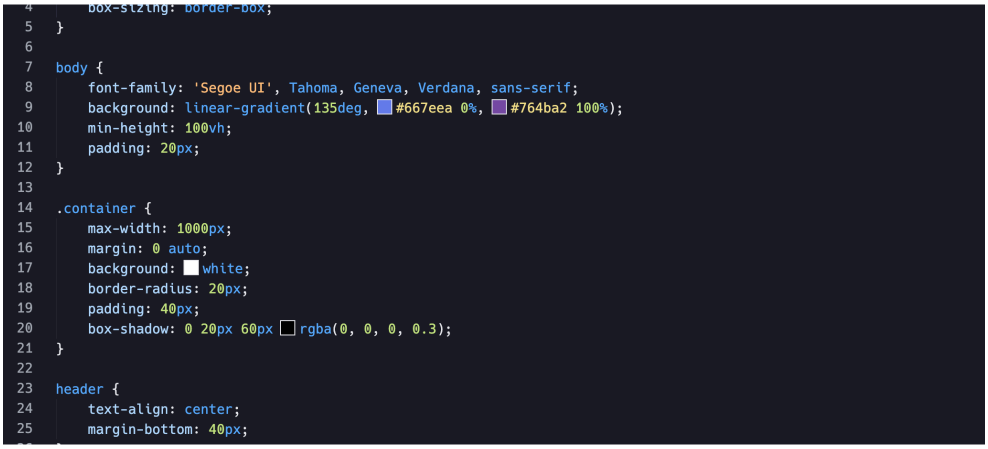


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Emotion Drawing Generator</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <div class="container">
        <header>
            <h1>🎨 Emotion Drawing Generator</h1>
            <p>Select an emotion to generate a unique abstract drawing</p>
        </header>

        <div class="emotion-selector">
            {% for emotion in emotions %}
            <button class="emotion-btn" data-emotion="{{ emotion }}" onclick="generateDrawing('{{ emotion }}')">
                {{ emotion.capitalize() }}
            </button>
            {% endfor %}
        </div>
```



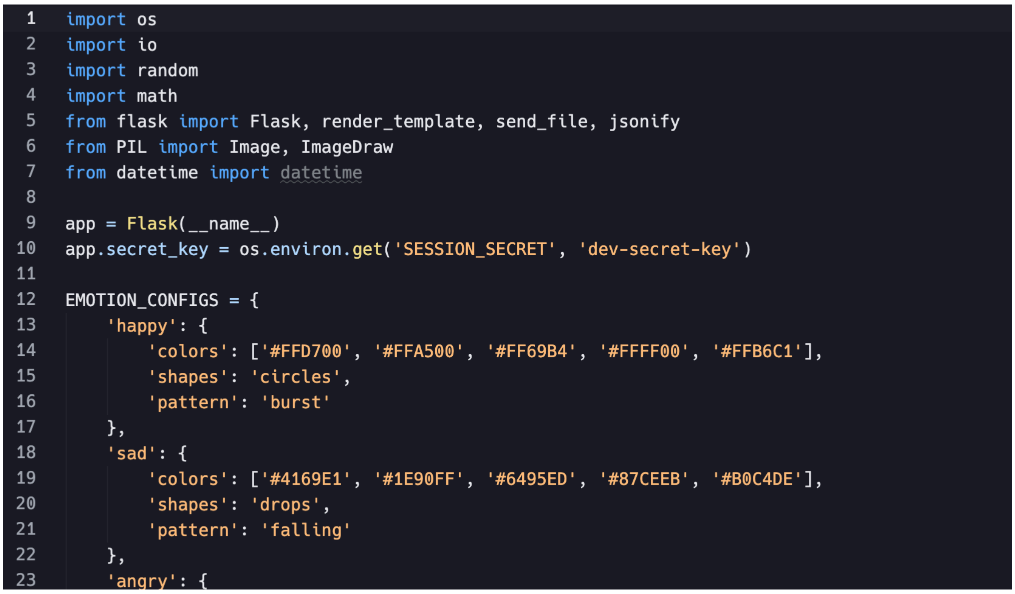
```html
    <script>
        let currentEmotion = null;

        function generateDrawing(emotion) {
            currentEmotion = emotion;
            const drawingContainer = document.getElementById('drawing-container');
            const loading = document.getElementById('loading');
            const downloadSection = document.getElementById('download-section');

            loading.style.display = 'block';
            drawingContainer.innerHTML = '';
            downloadSection.style.display = 'none';

            const buttons = document.querySelectorAll('.emotion-btn');
            buttons.forEach(btn => btn.classList.remove('active'));
            event.target.classList.add('active');

            const img = new Image();
            img.onload = function() {
                loading.style.display = 'none';
                drawingContainer.innerHTML = '';
                drawingContainer.appendChild(img);
```

CSS code preview(styles.css):

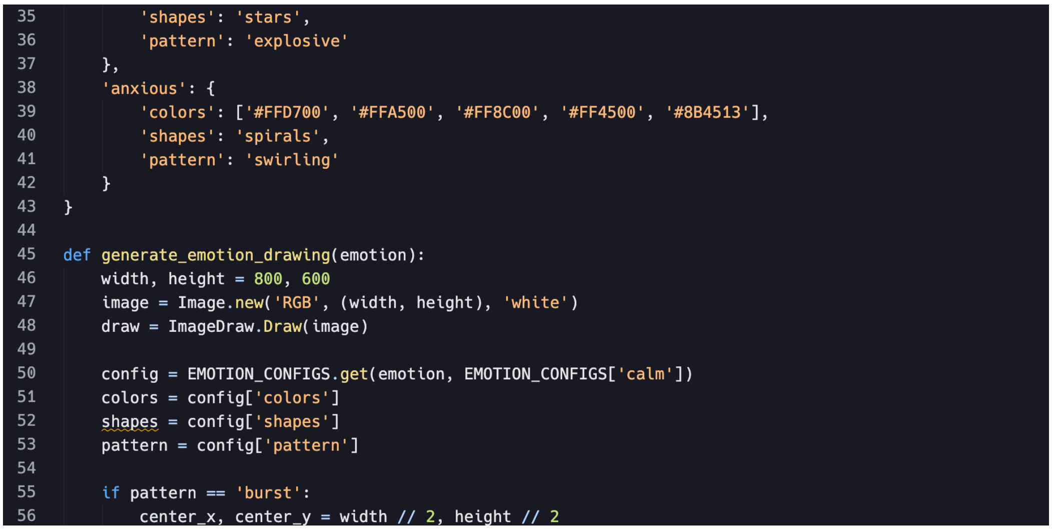


```css
    box-sizing: border-box;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    min-height: 100vh;
    padding: 20px;
}

.container {
    max-width: 1000px;
    margin: 0 auto;
    background: white;
    border-radius: 20px;
    padding: 40px;
    box-shadow: 0 20px 60px rgba(0, 0, 0, 0.3);
}

header {
    text-align: center;
    margin-bottom: 40px;
```

Python code preview(app.py):

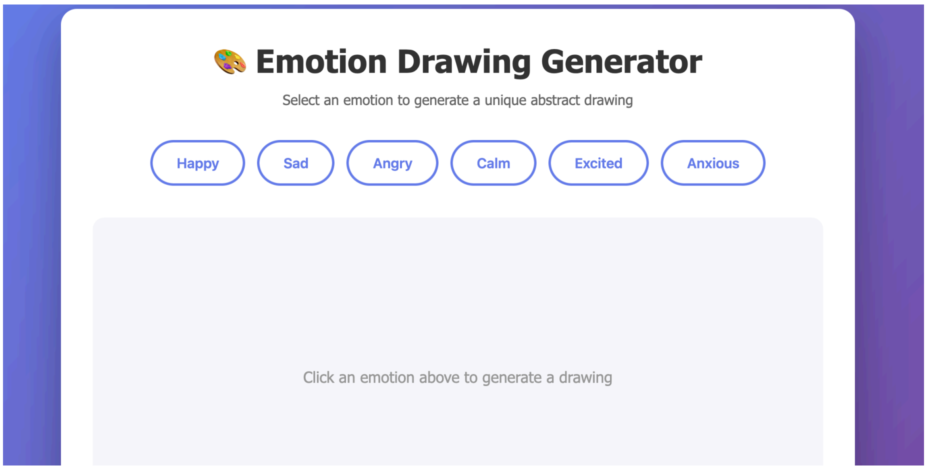


```python
import os
import io
import random
import math
from flask import Flask, render_template, send_file, jsonify
from PIL import Image, ImageDraw
from datetime import datetime

app = Flask(__name__)
app.secret_key = os.environ.get('SESSION_SECRET', 'dev-secret-key')

EMOTION_CONFIGS = {
    'happy': {
        'colors': ['#FFD700', '#FFA500', '#FF69B4', '#FFFF00', '#FFB6C1'],
        'shapes': 'circles',
        'pattern': 'burst'
    },
    'sad': {
        'colors': ['#4169E1', '#1E90FF', '#6495ED', '#87CEEB', '#B0C4DE'],
        'shapes': 'drops',
        'pattern': 'falling'
    },
    'angry': {
```



```python
        'shapes': 'stars',
        'pattern': 'explosive'
    },
    'anxious': {
        'colors': ['#FFD700', '#FFA500', '#FF8C00', '#FF4500', '#8B4513'],
        'shapes': 'spirals',
        'pattern': 'swirling'
    }
}

def generate_emotion_drawing(emotion):
    width, height = 800, 600
    image = Image.new('RGB', (width, height), 'white')
    draw = ImageDraw.Draw(image)

    config = EMOTION_CONFIGS.get(emotion, EMOTION_CONFIGS['calm'])
    colors = config['colors']
    shapes = config['shapes']
    pattern = config['pattern']

    if pattern == 'burst':
        center_x, center_y = width // 2, height // 2
```

Generated application:

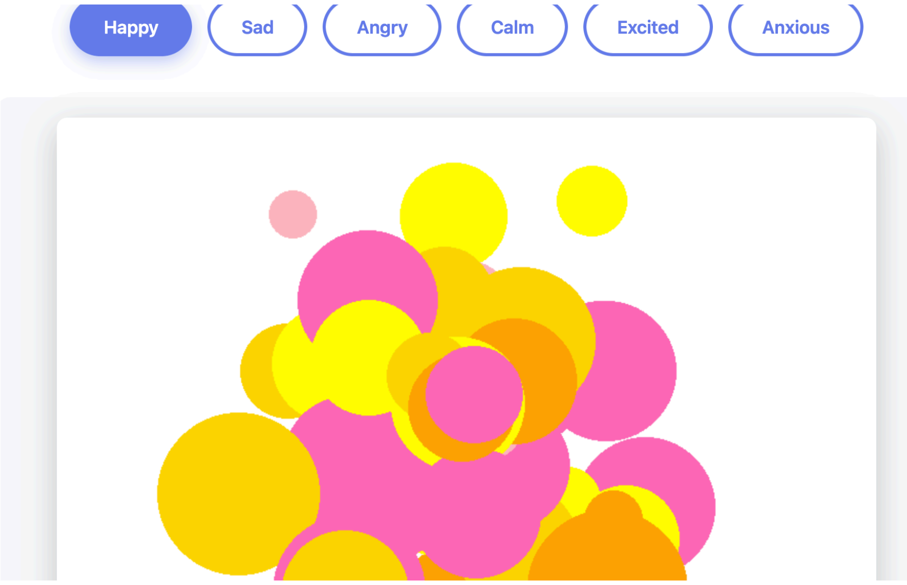


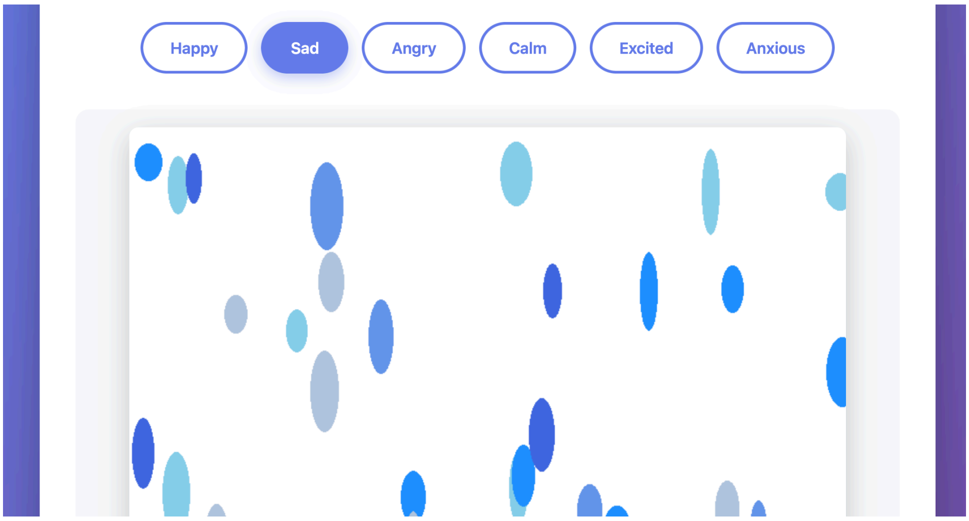
🎨 Emotion Drawing Generator

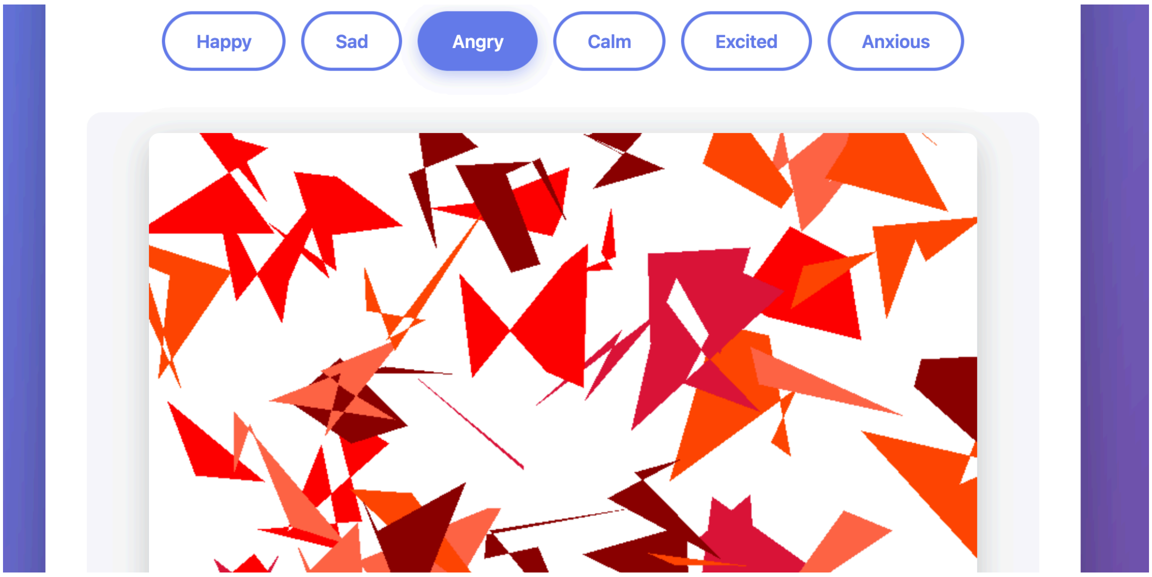
Select an emotion to generate a unique abstract drawing

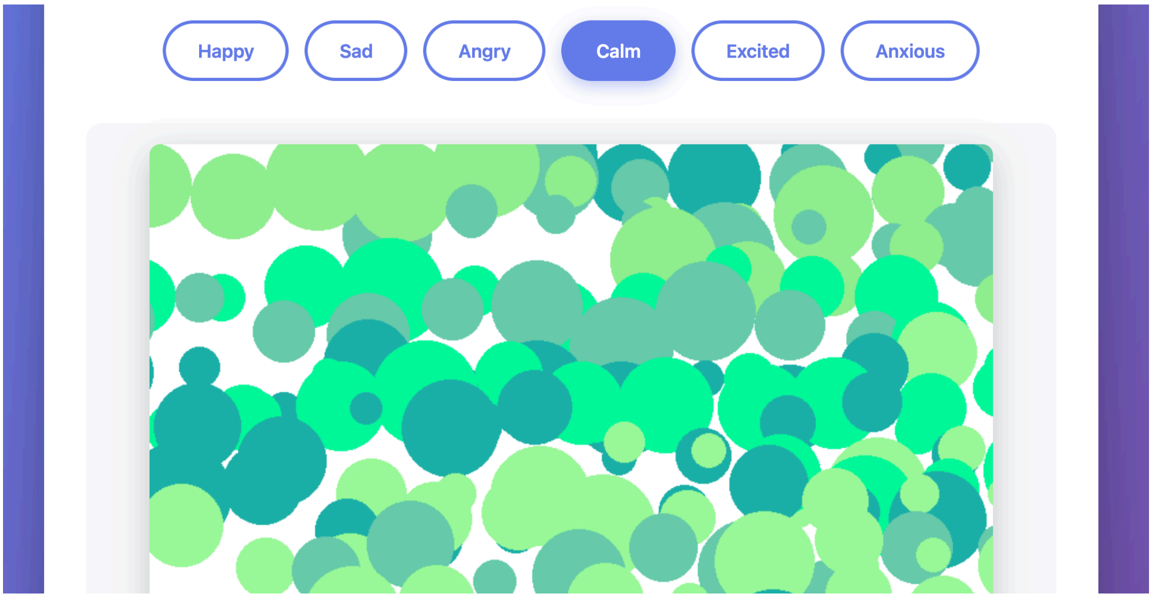
Click an emotion above to generate a drawing

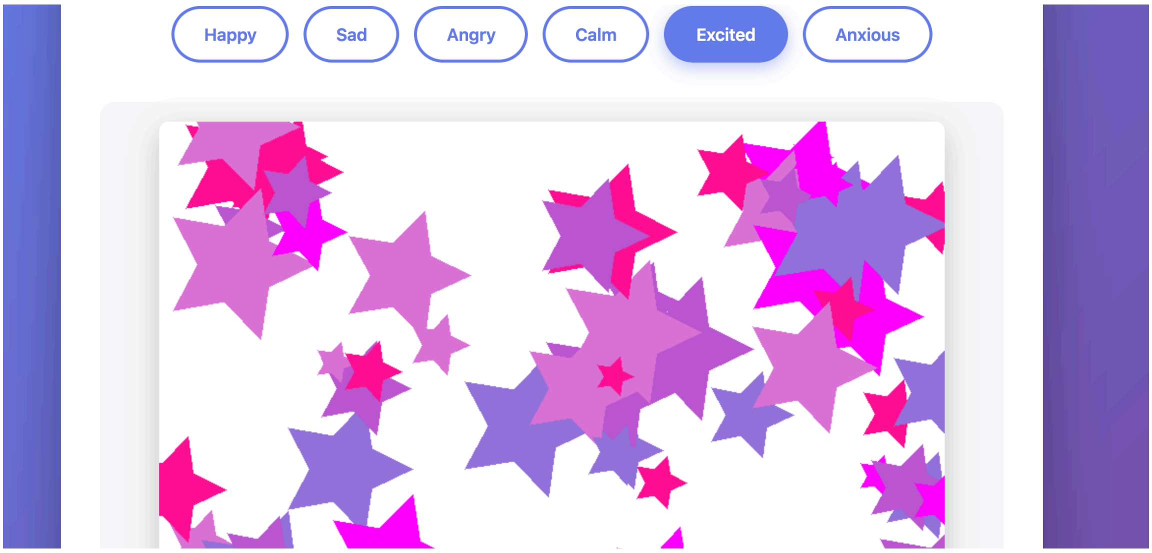
Diagram based on emotions

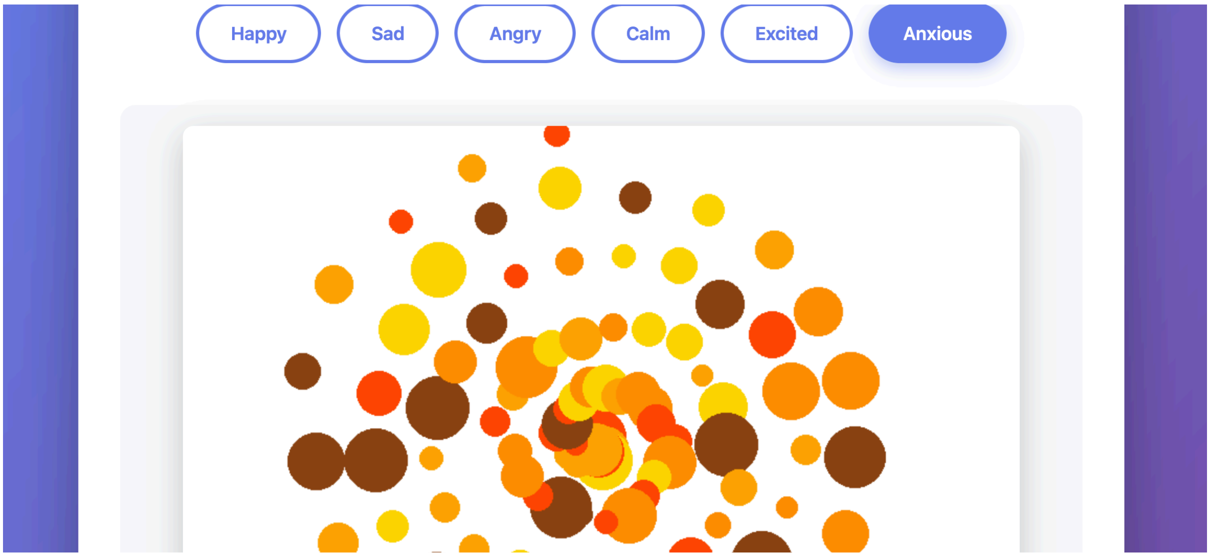




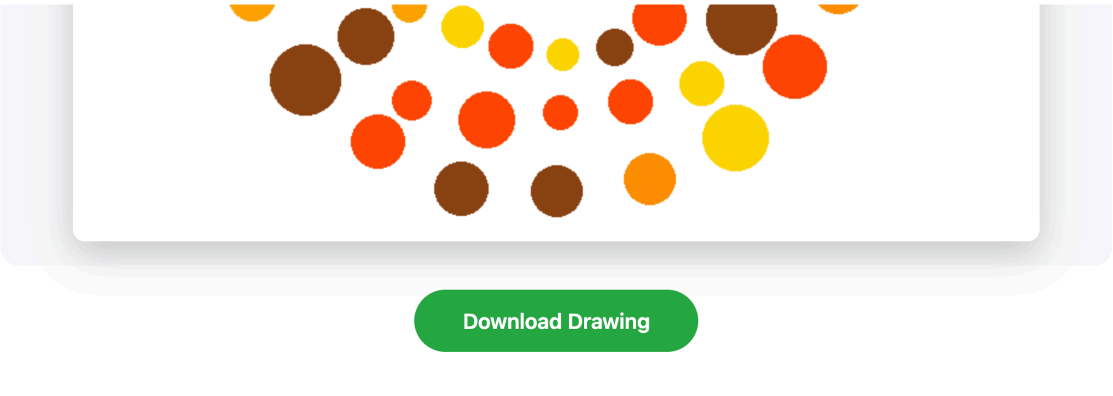

Happy
Sad
Angry
Calm
Excited
Anxious


Happy
Sad
Angry
Calm
Excited
Anxious

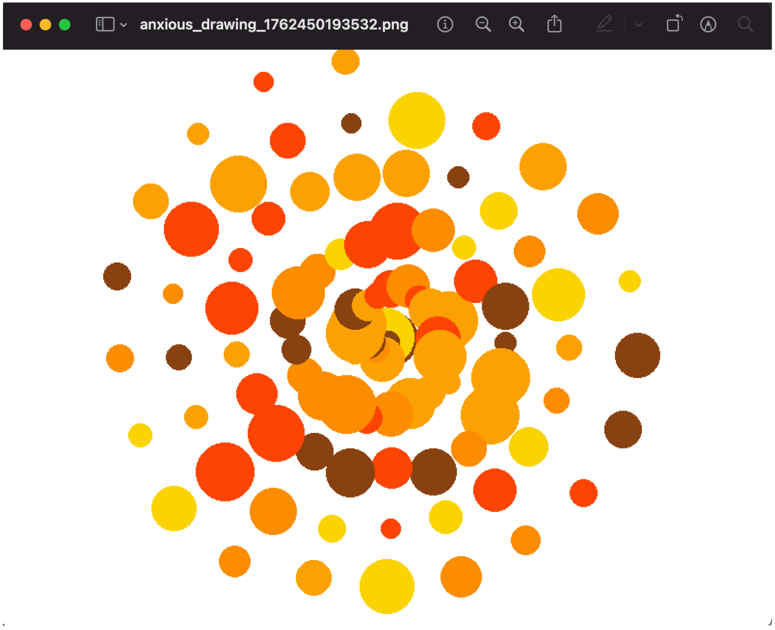

Happy
Sad
Angry
Calm
Excited
Anxious



Click to download



Downloaded image


anxious_drawing_1762450193532.png