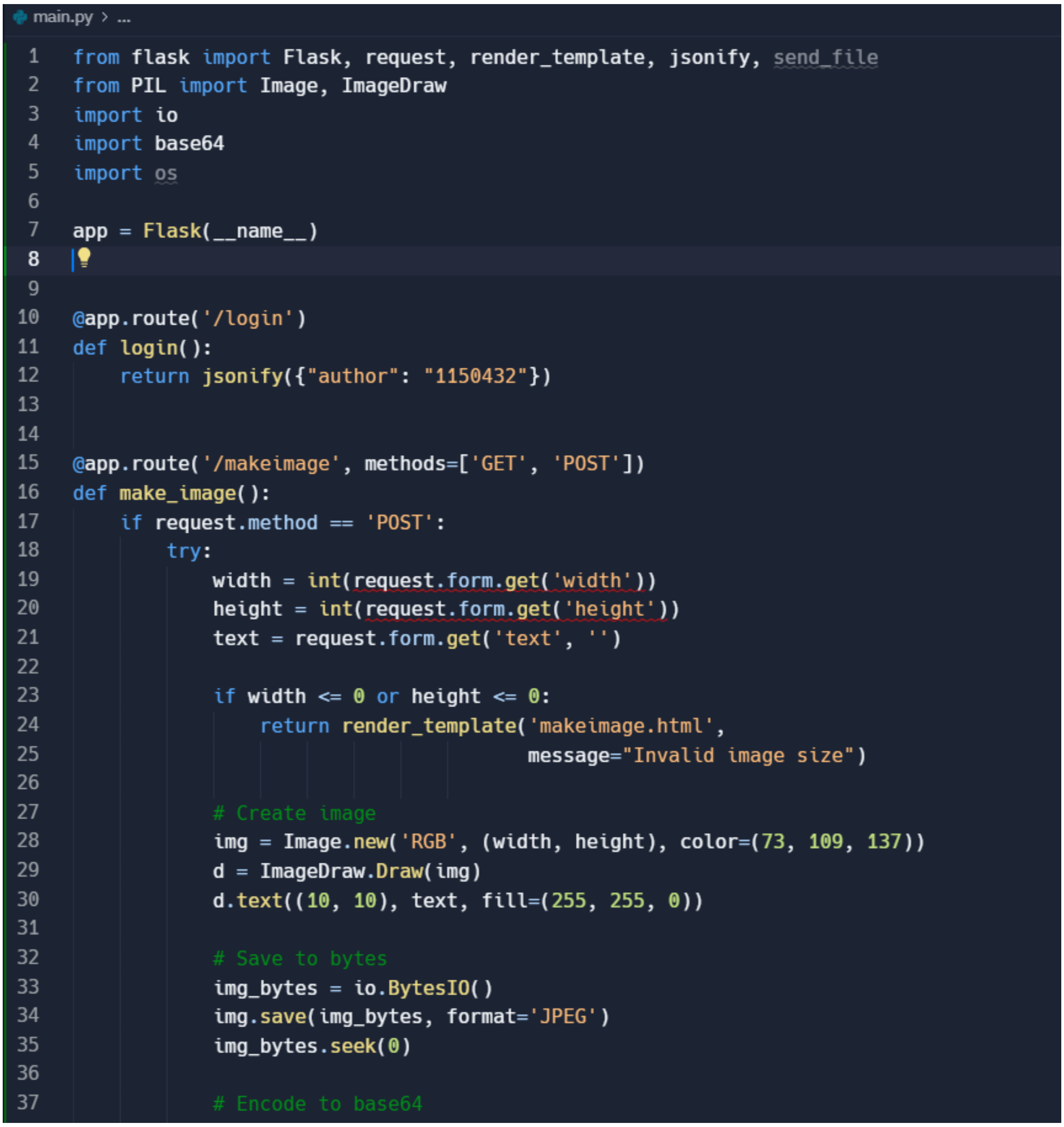
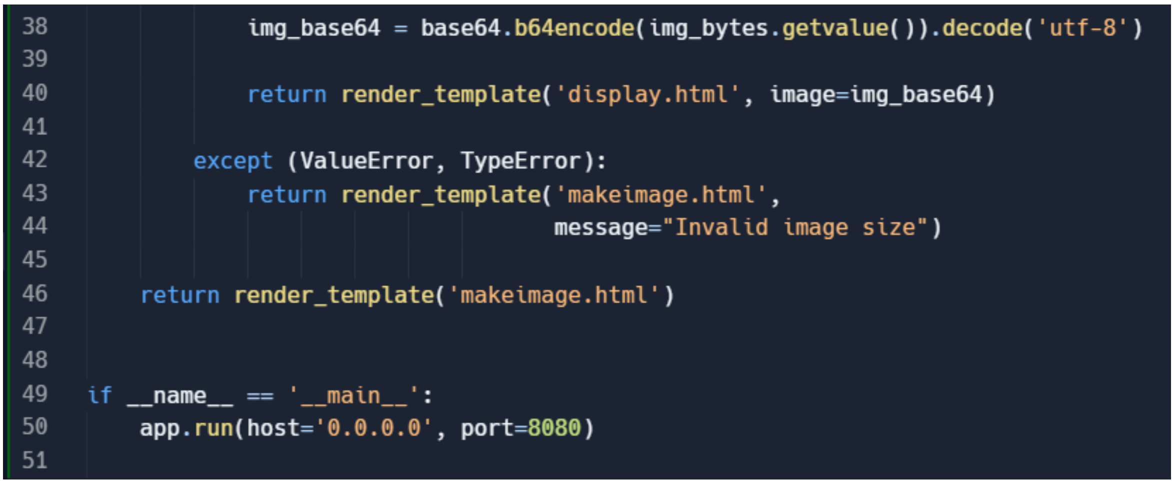
Лабораторная работа №10

Код программы:

main.py:

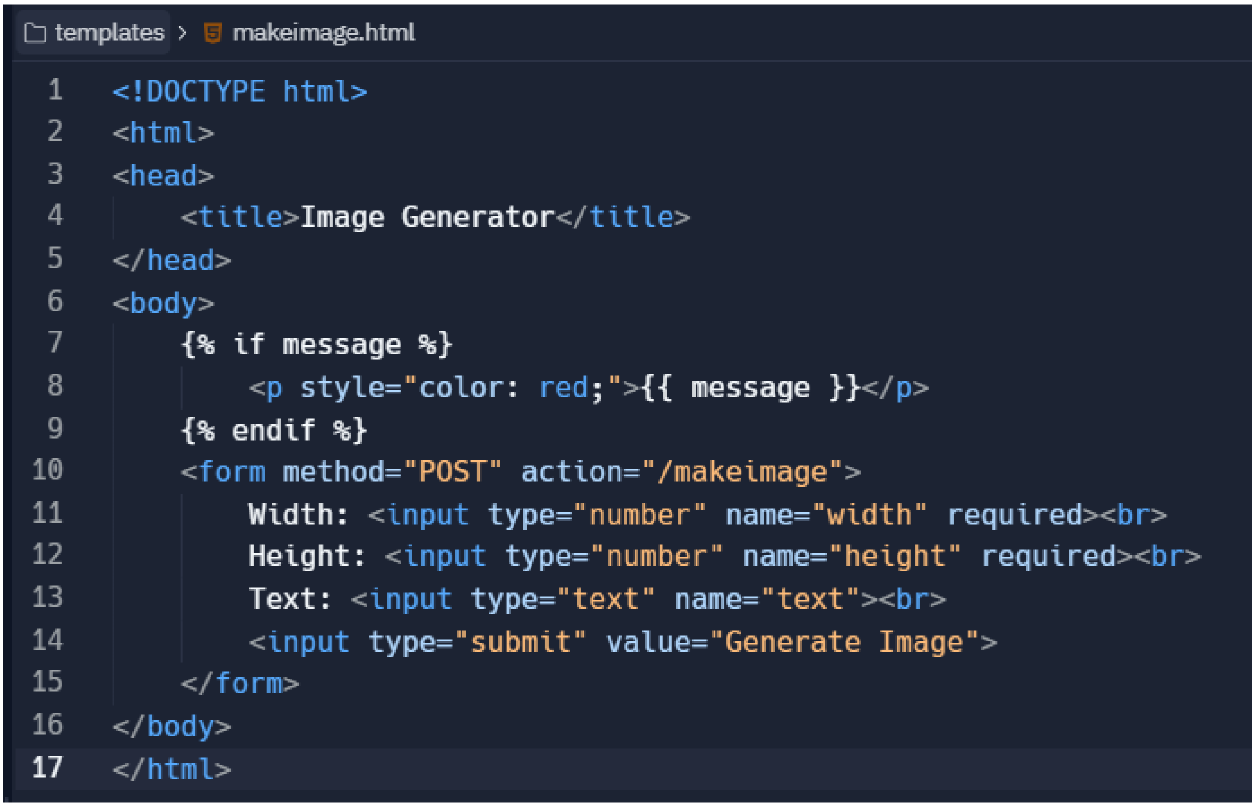


```python
from flask import Flask, request, render_template, jsonify, send_file
from PIL import Image, ImageDraw
import io
import base64
import os

app = Flask(__name__)


@app.route('/login')
def login():
    return jsonify({"author": "1150432"})


@app.route('/makeimage', methods=['GET', 'POST'])
def make_image():
    if request.method == 'POST':
        try:
            width = int(request.form.get('width'))
            height = int(request.form.get('height'))
            text = request.form.get('text', '')

            if width <= 0 or height <= 0:
                return render_template('makeimage.html',
                                       message="Invalid image size")

            # Create image
            img = Image.new('RGB', (width, height), color=(73, 109, 137))
            d = ImageDraw.Draw(img)
            d.text((10, 10), text, fill=(255, 255, 0))

            # Save to bytes
            img_bytes = io.BytesIO()
            img.save(img_bytes, format='JPEG')
            img_bytes.seek(0)

            # Encode to base64
```



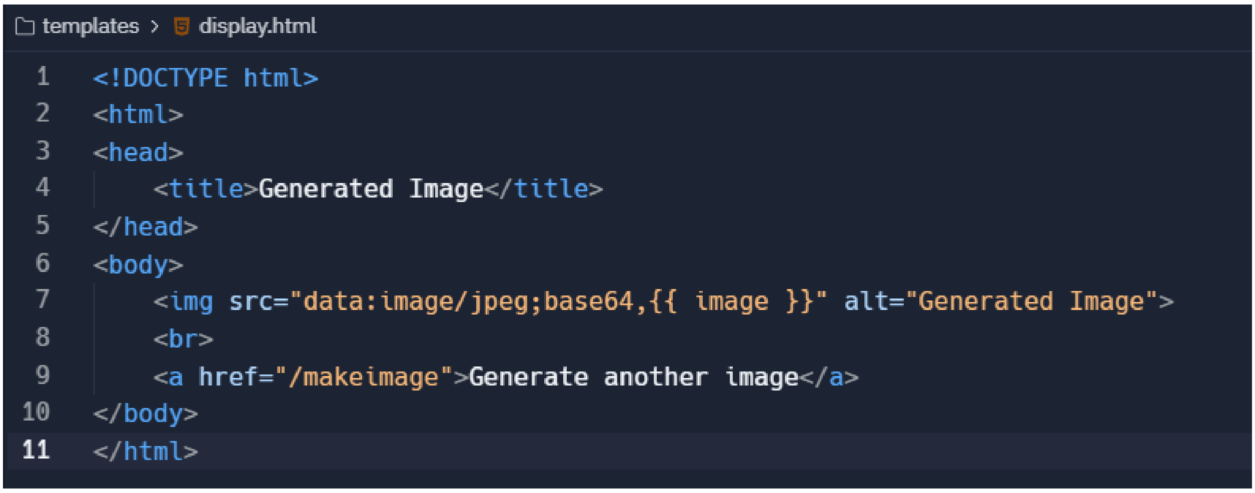
```python
            img_base64 = base64.b64encode(img_bytes.getvalue()).decode('utf-8')

            return render_template('display.html', image=img_base64)

        except (ValueError, TypeError):
            return render_template('makeimage.html',
                                   message="Invalid image size")

    return render_template('makeimage.html')


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)

```

makeimage.html:



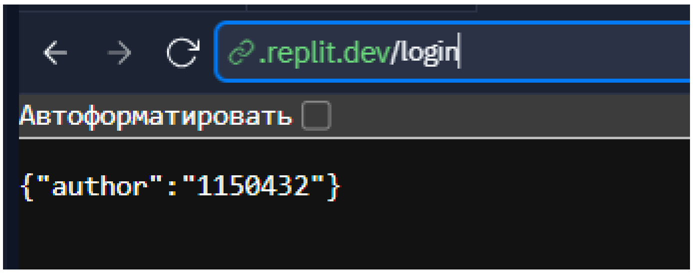
```html
<!DOCTYPE html>
<html>
<head>
    <title>Image Generator</title>
</head>
<body>
    {% if message %}
        <p style="color: red;">{{ message }}</p>
    {% endif %}
    <form method="POST" action="/makeimage">
        Width: <input type="number" name="width" required><br>
        Height: <input type="number" name="height" required><br>
        Text: <input type="text" name="text"><br>
        <input type="submit" value="Generate Image">
    </form>
</body>
</html>
```

display.html:

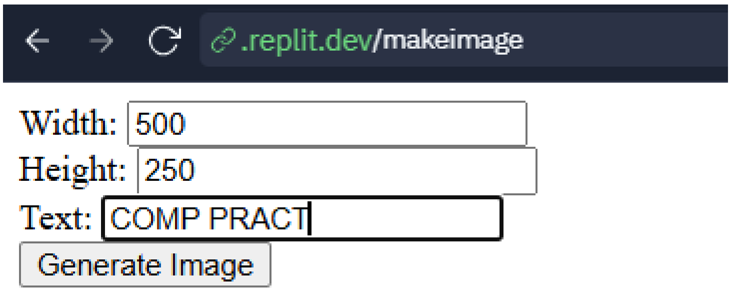


```html
<!DOCTYPE html>
<html>
<head>
    <title>Generated Image</title>
</head>
<body>
    <img src="data:image/jpeg;base64,{{ image }}" alt="Generated Image">
    <br>
    <a href="/makeimage">Generate another image</a>
</body>
</html>
```

Результат:



```
{"author":"1150432"}
```

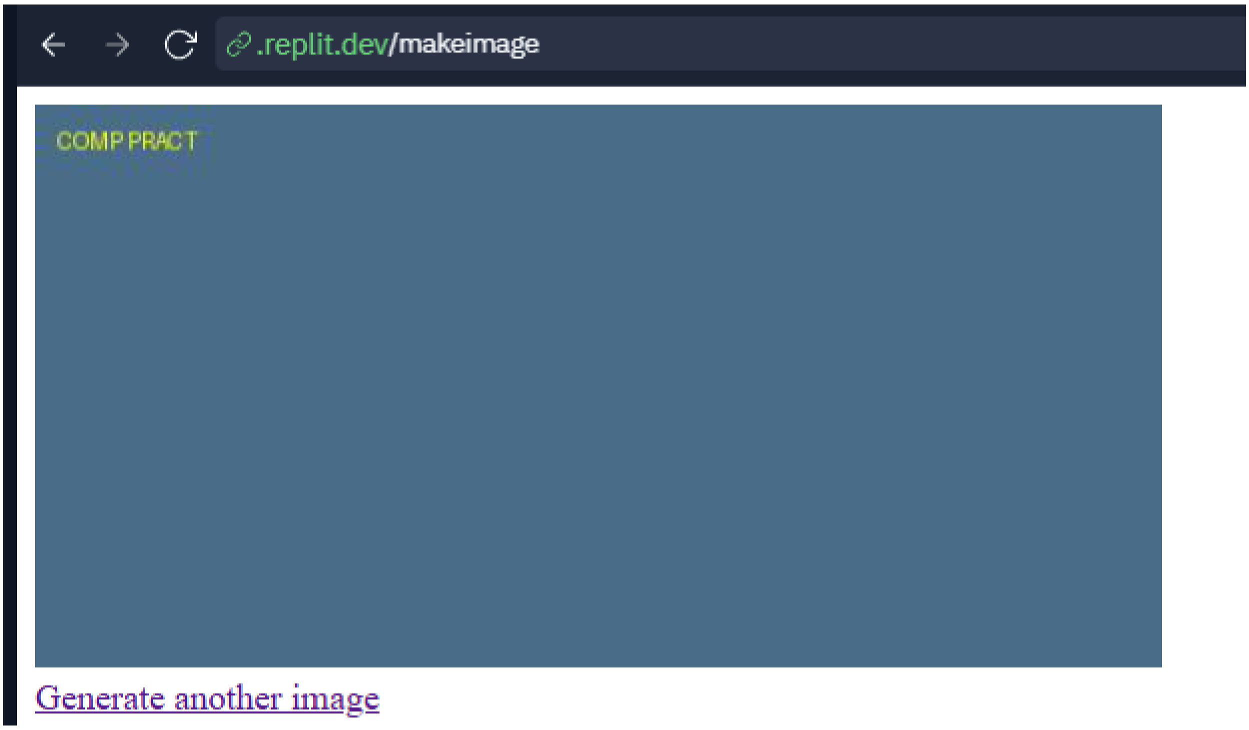


.replit.dev/makeimage

Width: 500

Height: 250

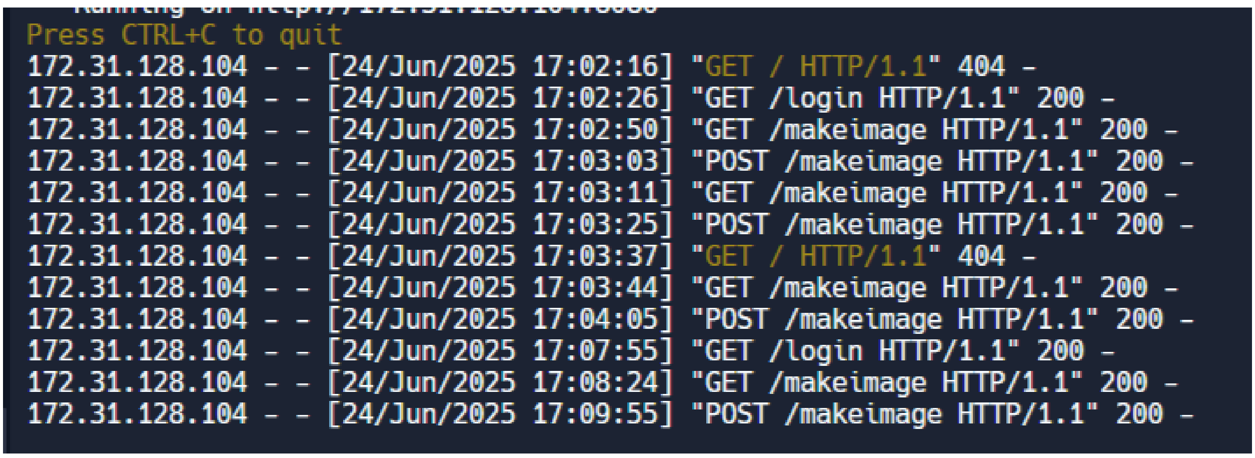
Text: COMP PRACT



.replit.dev/makeimage

Generate another image

Консоль:



```
 * Running on http://172.31.128.104:8080
Press CTRL+C to quit
172.31.128.104 - - [24/Jun/2025 17:02:16] "GET / HTTP/1.1" 404 -
172.31.128.104 - - [24/Jun/2025 17:02:26] "GET /login HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:02:50] "GET /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:03:03] "POST /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:03:11] "GET /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:03:25] "POST /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:03:37] "GET / HTTP/1.1" 404 -
172.31.128.104 - - [24/Jun/2025 17:03:44] "GET /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:04:05] "POST /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:07:55] "GET /login HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:08:24] "GET /makeimage HTTP/1.1" 200 -
172.31.128.104 - - [24/Jun/2025 17:09:55] "POST /makeimage HTTP/1.1" 200 -
```