```
!pip install pyngrok flask
!pip install flask-ngrok pyngrok
from pyngrok import ngrok
```

```
Collecting pyngrok
  Downloading pyngrok-7.3.0-py3-none-any.whl.metadata (8.1 kB)
Requirement already satisfied: flask in /usr/local/lib/python3.11/dist-packages (3.1.1)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.11/dist-packages (from pyngrok) (6.0.2)
Requirement already satisfied: blinker>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from flask) (8.2.1)
Requirement already satisfied: itsdangerous>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from flask) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from flask) (3.0.2)
Requirement already satisfied: werkzeug>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.3)
Downloading pyngrok-7.3.0-py3-none-any.whl (25 kB)
Installing collected packages: pyngrok
Successfully installed pyngrok-7.3.0
Collecting flask-ngrok
  Downloading flask_ngrok-0.0.25-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyngrok in /usr/local/lib/python3.11/dist-packages (7.3.0)
Requirement already satisfied: Flask>=0.8 in /usr/local/lib/python3.11/dist-packages (from flask-ngrok) (3.1.1)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from flask-ngrok) (2.32.3)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.11/dist-packages (from pyngrok) (6.0.2)
Requirement already satisfied: blinker>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from Flask>=0.8->flask-ngrok) (1.9.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from Flask>=0.8->flask-ngrok) (8.2.1)
Requirement already satisfied: itsdangerous>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from Flask>=0.8->flask-ngrok) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from Flask>=0.8->flask-ngrok) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from Flask>=0.8->flask-ngrok) (3.0.2)
Requirement already satisfied: werkzeug>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from Flask>=0.8->flask-ngrok) (3.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->flask-ngrok) (2025.8.3)
Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Installing collected packages: flask-ngrok
Successfully installed flask-ngrok-0.0.25
```

```
from flask_ngrok import run_with_ngrok
from flask import Flask, request, jsonify
from pyngrok import ngrok

app = Flask(__name__)
run_with_ngrok(app)

users = {
    1: {"id": 1, "username": "bharadwajivaturi", "email": "bharadwajivaturi@gmail.com"},
    2: {"id": 2, "username": "sahasraivaturi", "email": "sahasraivaturi@gmail.com"}
}
user_id_counter = 3

@app.route('/users', methods=['GET'])
```

```
def get_users():
    return jsonify({
        "users": list(users.values()),
        "count": len(users)
    })

@app.route('/users/<int:user_id>', methods=['GET'])
def get_user(user_id):
    user = users.get(user_id)

    if not user:
        return jsonify({"error": "User not found"}), 404
    return jsonify(user)

@app.route('/users', methods=['POST'])
def create_user():
    global user_id_counter
    data = request.json

    if not data or 'username' not in data:
        return jsonify({"error": "Username is required"}), 400

    if any(u['username'] == data['username'] for u in users.values()):
        return jsonify({"error": "Username already exists"}), 409

    new_user = {
        "id": user_id_counter,
        "username": data['username'],
        "email": data.get('email', "")
    }
    users[user_id_counter] = new_user
    user_id_counter += 1
    return jsonify(new_user), 201

@app.route('/users/<int:user_id>', methods=['PUT'])
def update_user(user_id):
    user = users.get(user_id)

    if not user:
        return jsonify({"error": "User not found"}), 404

    data = request.json

    if not data:
        return jsonify({"error": "No data provided"}), 400

    if 'username' in data:
        if any(u['username'] == data['username'] for u in users.values() if u['id'] != user_id):
            return jsonify({"error": "Username already taken"}), 409
        user['username'] = data['username']
    if 'email' in data:
        user['email'] = data['email']
    return jsonify(user)

@app.route('/users/<int:user_id>', methods=['DELETE'])
def delete_user(user_id):
    if user_id not in users:
        return jsonify({"error": "User not found"}), 404

    del users[user_id]
    return jsonify({"message": "User deleted successfully"}), 200

@app.route('/')
def home():
    """Home endpoint with API information"""
    return """
    <!DOCTYPE html>
    <html>
    <head>
        <title>User Management API</title>
        <style>
            body { font-family: Arial, sans-serif; margin: 40px; }
            h1 { color: #333; }
            ul { list-style-type: none; padding: 0; }
            li { margin: 10px 0; }
            a {
                display: inline-block;
                padding: 10px 15px;
                background-color: #000000;
                color: white;
                text-decoration: none;
                border-radius: 5px;
            }
            a:hover { background-color: #787878; }
            #usersList { margin-top: 20px; padding: 10px; border: 1px solid #ddd; }
        </style>
        <script>
            function fetchUsers() {
                fetch('/users')
                    .then(response => response.json())
                    .then(data => {
                        const usersList = document.getElementById('usersList');
                        usersList.innerHTML = '<h3>Users:</h3>';

                        if (data.users.length === 0) {
                            usersList.innerHTML += '<p>No users found</p>';
                            return;
                        }

                        const ul = document.createElement('ul');
                        data.users.forEach(user => {
                                const li = document.createElement('li');
                                li.innerHTML = `<strong>ID:</strong> ${user.id},
                                              <strong>Username:</strong> ${user.username},
                                              <strong>Email:</strong> ${user.email}`;
                                ul.appendChild(li);
                            });
                            usersList.appendChild(ul);
                        })
                        .catch(error => {
                            console.error('Error:', error);
                        });
                }
            </script>
        </head>
        <body>
            <h1>User Management API</h1>
            <p>Available endpoints:</p>
            <ul>
                <li><a href="javascript:void(0);" onclick="fetchUsers()">GET /users</a> - List all users</li>
                <li>GET /users/&lt;id&gt; - Get specific user</li>
                <li>POST /users - Create new user (requires username)</li>
                <li>PUT /users/&lt;id&gt; - Update user</li>
                <li>DELETE /users/&lt;id&gt; - Delete user</li>
            </ul>
            <div id="usersList"></div>
        </body>
        </html>
        """

ngrok.set_auth_token("30zyBQBi3tMl2gl7XtkNGhdCcxN_6w3p3X1dKe42WBKvygc3C")
public_url = ngrok.connect(5000).public_url
print("Public URL:", public_url)

if __name__ == '__main__':
    app.run()
```

```
Public URL: https://d5233ae31edc.ngrok-free.app
 * Serving Flask app '__main__'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
 * Running on http://d5233ae31edc.ngrok-free.app
 * Traffic stats available on http://127.0.0.1:4040
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:19:50] "GET /users HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:20:52] "GET /users/1 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:21:42] "POST /users HTTP/1.1" 415 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:22:54] "POST /users HTTP/1.1" 201 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:23:28] "PUT /users/1 HTTP/1.1" 409 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:24:06] "PUT /users/1 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:24:32] "DELETE /users/2 HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [08/Aug/2025 15:24:49] "GET /users HTTP/1.1" 200 -
```