template login.html

```

```

Kod samego programu

```
from flask import Flask, request, render_template
import paramiko
import time
import re
# MikroTik API credentials
ROUTER_IP = '192.168.1.1'
USERNAME = 'admin'
PASSWORD = 'PASS'

app = Flask(__name__)

# Configure Flask to trust X-Forwarded-For header
app.config['TRUSTED_PROXIES'] = '127.0.0.1'

def remove_port(ip_address_with_port):
    return re.split(r'[;,|:]',ip_address_with_port)

# Function to add IP address to the specified list with a timeout
def add_to_list(ip_address_arg):
    ip_address = remove_port(ip_address_arg)[0]
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    ssh.connect(ROUTER_IP,port=22, username=USERNAME, password=PASSWORD)

    # Send command to add IP address to address list
    command = f"/ip firewall address-list add list=port_knocking_stage1
address={ip_address} timeout=12h"
    stdin, stdout, stderr = ssh.exec_command(command)

    # Wait for the command to execute
    time.sleep(1)

    # Check for any errors
    if stderr.read().decode():
        print("Error:", stderr.read().decode())
    else:
        print("IP address added successfully. "+ip_address)

    ssh.close()
# Dummy database for demonstration (replace with your own authentication
mechanism)
users = {
    'admin': 'pass',
```

```
}

# Authentication route
@app.route('/', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        if username in users and users[username] == password:
            user_ip = request.headers.get('X-Forwarded-For',
request.remote_addr)
            add_to_list(user_ip)
            add_to_list(user_ip)
            success_message = 'Authenticated successfully! Your IP address
<strong>{}</strong> has been added to the whitelist for 12
hours.'.format(remove_port(user_ip)[0])
            return '<div style="font-family: Arial, sans-serif; text-align:
center; margin-top: 50px;"><h2 style="color:
#4CAF50;">Success!</h2><p>{}</p></div>'.format(success_message)
        else:
            error = 'Invalid credentials. Please try again.'
    return render_template('login.html', error=error)

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