

EXPERIMENT 05

SAST TOOL : SYNK

CODE :

```
import os

import urllib

from flask import Flask, request

from django.db import connection, models

from django.db.models.expressions import RawSQL


app = Flask(__name__)


@app.route("/code-execution")
def code_execution():
    code1 = request.args.get("code1")
    exec("setname('%s')" % code1)
    return a

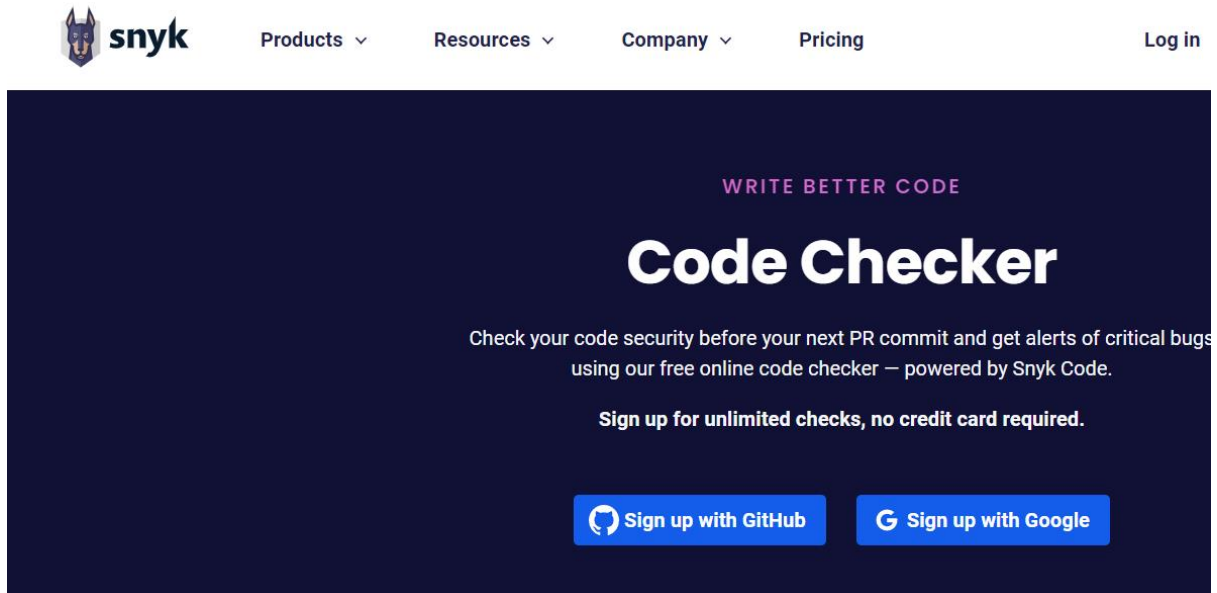

@app.route("/open-redirect")
def open_redirect():
    redirect_loc = request.args.get('redirect')
    return redirect(redirect_loc)


@app.route("/sqli/<username>")
def show_user(username):
    with connection.cursor() as cursor:
        cursor.execute("SELECT * FROM users WHERE username = '%s'" % username)


if __name__ == '__main__':
```

```
app.run(host='0.0.0.0', port=9000)
```

SOURCE CODE REVIEW USING SYNK



The image shows the Snyk Code Checker landing page. At the top, there is a navigation bar with the Snyk logo, links for Products, Resources, Company, Pricing, and a Log in button. The main section has a dark blue background with the text "WRITE BETTER CODE" in purple, followed by "Code Checker" in large white letters. Below this, a paragraph states: "Check your code security before your next PR commit and get alerts of critical bugs using our free online code checker — powered by Snyk Code." A sub-headline reads: "Sign up for unlimited checks, no credit card required." At the bottom, there are two blue buttons: "Sign up with GitHub" and "Sign up with Google".

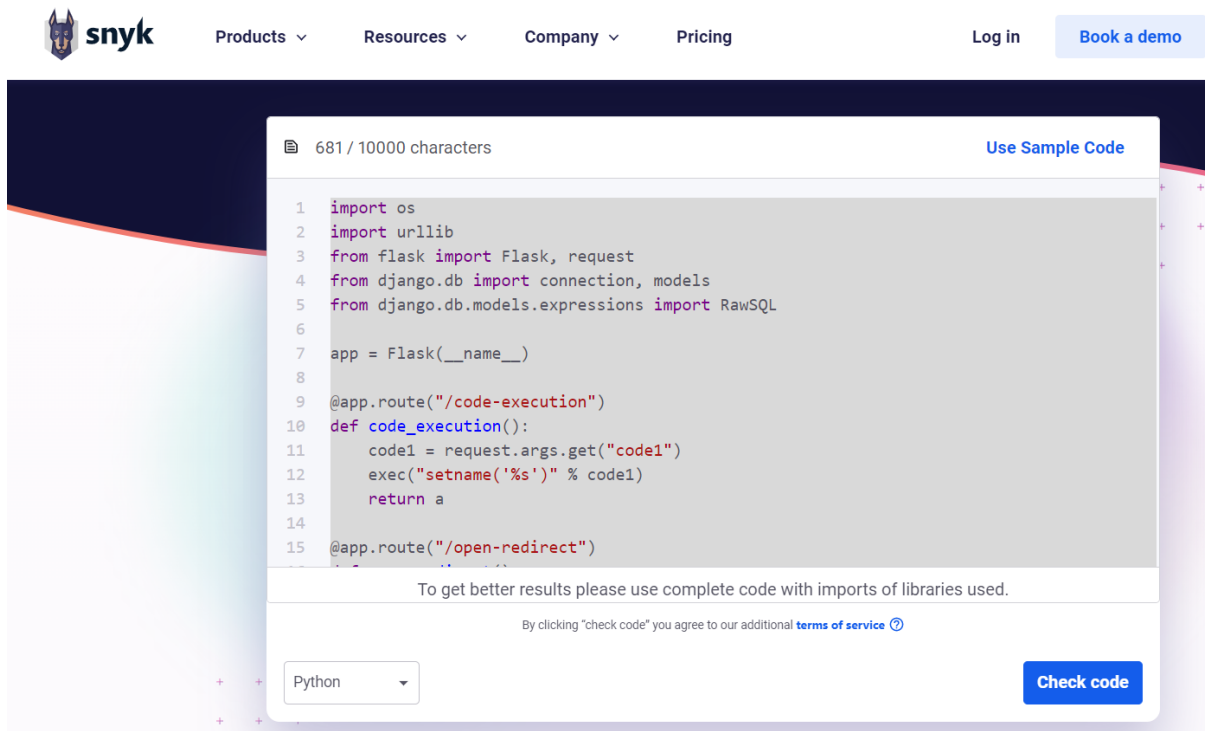
WRITE BETTER CODE

Code Checker

Check your code security before your next PR commit and get alerts of critical bugs using our free online code checker — powered by Snyk Code.

Sign up for unlimited checks, no credit card required.

[Sign up with GitHub](#) [Sign up with Google](#)



The image shows the Snyk Code Checker interface. At the top, there is a navigation bar with the Snyk logo, links for Products, Resources, Company, Pricing, a Log in button, and a "Book a demo" button. The main section features a code editor with a light gray background. The editor shows a Python Flask application with a vulnerable endpoint. Below the code editor, there is a message: "To get better results please use complete code with imports of libraries used." A note states: "By clicking 'check code' you agree to our additional [terms of service](#)". At the bottom, there is a dropdown menu set to "Python" and a blue "Check code" button.

681 / 10000 characters [Use Sample Code](#)

```
1 import os
2 import urllib
3 from flask import Flask, request
4 from django.db import connection, models
5 from django.db.models.expressions import RawSQL
6
7 app = Flask(__name__)
8
9 @app.route("/code-execution")
10 def code_execution():
11     code1 = request.args.get("code1")
12     exec("setname('%s')" % code1)
13     return a
14
15 @app.route("/open-redirect")
```

To get better results please use complete code with imports of libraries used.

By clicking "check code" you agree to our additional [terms of service](#)

Python [Check code](#)

We found 3 issues in your code

H 2 high severity **M** 1 medium severity **L** 0 low severity

H SQL Injection

VULNERABILITY | [CWE-89](#)

```
22 def show_user(username):
23     with connection.cursor() as cursor:
24         cursor.execute("SELECT * FROM users WHERE
25             username = '%s'" % username)
```

Unsanitized input from an HTTP parameter flows into execute, where it is used in an SQL query. This may result in an SQL Injection vulnerability.

H Code Injection

VULNERABILITY | [CWE-94](#)

```
10 def code_execution():
11     code1 = request.args.get("code1")
12     exec("Setname('%s')" % code1)
13     return a
14
```

Unsanitized input from an HTTP parameter flows into exec, where it is executed as Python code. This may result in a Code Injection vulnerability.

M Open Redirect

VULNERABILITY | [CWE-601](#)

```
16 def open_redirect():
17     redirect_loc = request.args.get('redirect')
18     return redirect(redirect_loc)
19
20
```