



DEVOPS

Automated Testing Assignment

Submitted By-

Priyanshi

SAP ID – 500107299

Roll no - R2142220603

Batch – B7 (CCVT NH)

Submitted To -

Dr. Mitali Chugh

Step 1: Install Python and Pip

First, ensure that **Python** and **pip** are installed.

1. Open **Command Prompt (CMD)** and check if Python is installed:

```
python --version
```

If Python is not installed, download it from [Python's official website](#) and install it.

- ✓ Make sure to **check the "Add Python to PATH" option** during installation.

2. Verify if **pip** (Python's package manager) is installed:

```
pip --version
```

```
C:\Users\hp>python --version
Python 3.11.1

C:\Users\hp>pip --version
pip 22.3.1 from C:\Users\hp\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)
```

✓ Step 2: Install FastAPI and Required Libraries

Run the following command in **CMD or PowerShell** to install FastAPI, Uvicorn (server), and Requests (for API testing):

```
pip install fastapi uvicorn requests pytest
```

```
C:\Users\hp>pip install fastapi uvicorn requests pytest
Collecting fastapi
  Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
    95.2/95.2 kB 993.4 kB/s eta 0:00:00
Collecting uvicorn
  Downloading uvicorn-0.34.0-py3-none-any.whl (62 kB)
    62.3/62.3 kB 3.3 MB/s eta 0:00:00
Requirement already satisfied: requests in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (2.32.3)
Collecting pytest
  Downloading pytest-8.3.5-py3-none-any.whl (343 kB)
    343.6/343.6 kB 2.4 MB/s eta 0:00:00
Collecting starlette<0.47.0,>=0.46.0
  Downloading starlette-0.46.1-py3-none-any.whl (71 kB)
    72.0/72.0 kB 3.9 MB/s eta 0:00:00
Collecting pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4
  Downloading pydantic-2.10.6-py3-none-any.whl (431 kB)
    431.7/431.7 kB 3.0 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions=>=3.8.0 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from fastapi) (4.12.2)
Collecting click<8.1.0,>=8.0.0
  Downloading click-8.1.8-py3-none-any.whl (98 kB)
    98.2/98.2 kB 2.8 MB/s eta 0:00:00
Requirement already satisfied: h11>=0.8 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from uvicorn) (0.14.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from requests) (3.4.1)
Requirement already satisfied: idna=4,>=2.5 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from requests) (3.18)
Requirement already satisfied: urllib3<3,>=2.1.1 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from requests) (2.3.0)
Requirement already satisfied: certifi=>2017.4.17 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from requests) (2024.12.14)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest) (0.4.6)
Collecting iniconf
  Downloading iniconf-2.1.0-py3-none-any.whl (6.9 kB)
Requirement already satisfied: packaging in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest) (24.2)
Collecting pluggy<2,>=1.5
  Downloading pluggy-1.5.0-py3-none-any.whl (20 kB)
Collecting annotated-types<=0.6.0
  Downloading annotated-types-0.7.0-py3-none-any.whl (13 kB)
Collecting pydantic-core==2.27.2
  Downloading pydantic_core-2.27.2-cp311-cp311-win_amd64.whl (2.0 MB)
    2.0/2.0 MB 800.0 kB/s eta 0:00:00
Requirement already satisfied: aioio<5,>=3.6.2 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from starlette<0.47.0,>=0.46.0->fastapi) (4.8.0)
Requirement already satisfied: sniffio>=1.1 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from aioio<5,>=3.6.2->starlette<0.47.0,>=0.46.0->fastapi) (1.3.1)
Installing collected packages: pydantic-core, pluggy, iniconf, click, annotated-types, uvicorn, starlette, pytest, pydantic, fastapi
Successfully installed annotated-types-0.7.0 click-8.1.8 fastapi-0.115.12 iniconf-2.1.0 pluggy-1.5.0 pydantic-core-2.27.2 pytest-8.3.5 starlette-0.46.1 uvicorn-0.34.0
[notice] A new release of pip available: 22.3.1 => 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Step 3: Create the API Server (FastAPI)

1. **Open Notepad, VS Code, or any text editor** and create a new file:
Save it as apiserver.py
2. **Copy and paste the following FastAPI code:**

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def home():
    return {"message": "Welcome to FastAPI Calculator API!"}

@app.get("/add/{num1}/{num2}")
def add(num1: int, num2: int):
    return {"result": num1 + num2}

@app.get("/subtract/{num1}/{num2}")
def subtract(num1: int, num2: int):
    return {"result": num1 - num2}

@app.get("/multiply/{num1}/{num2}")
def multiply(num1: int, num2: int):
    return {"result": num1 * num2}

if __name__ == "__main__":
    import uvicorn
    uvicorn.run("apiserver:app", host="127.0.0.1", port=8000, reload=True)
```

Step 4: Run the API Server

1. Open **CMD** and navigate to the folder where apiserver.py is saved:

```
cd path\to\your\file
```

(Replace path\to\your\file with the actual folder path where your Python file is stored.)

2. Start the API server using:

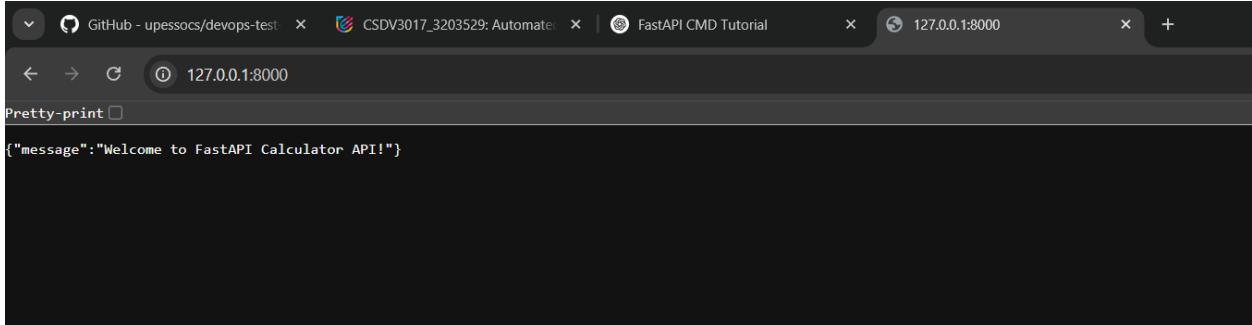
```
python apiserver.py
```

OR

```
uvicorn apiserver:app --host 127.0.0.1 --port 8000 --reload
```

```
C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt>python apiserver.py
INFO:     Will watch for changes in these directories: ['C:\\\\Users\\\\hp\\\\OneDrive\\\\Desktop\\\\Sem_6\\\\Devops\\\\Expt']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [11808] using StatReload
INFO:     Started server process [13720]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO: 127.0.0.1:51033 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:51033 - "GET /favicon.ico HTTP/1.1" 404 Not Found
```

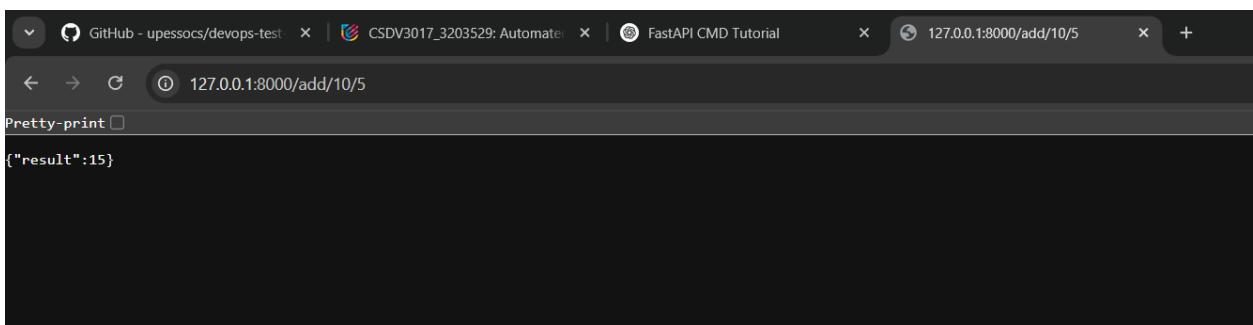
✓ Now your API is running on <http://127.0.0.1:8000>



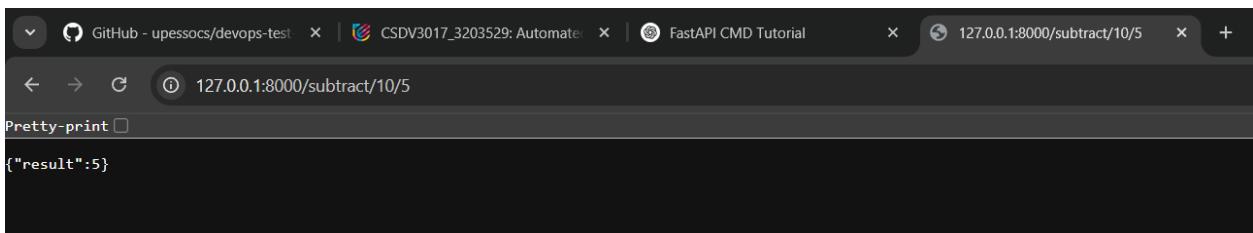
✓ Step 5: Test the API in a Web Browser

Open your browser and enter the following URLs to test:

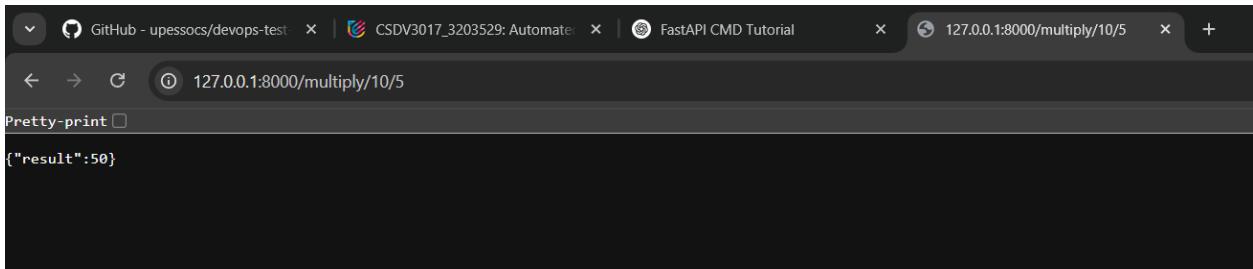
- **Addition:** <http://127.0.0.1:8000/add/10/5> → Should return {"result": 15}



- **Subtraction:** <http://127.0.0.1:8000/subtract/10/5> → Should return {"result": 5}



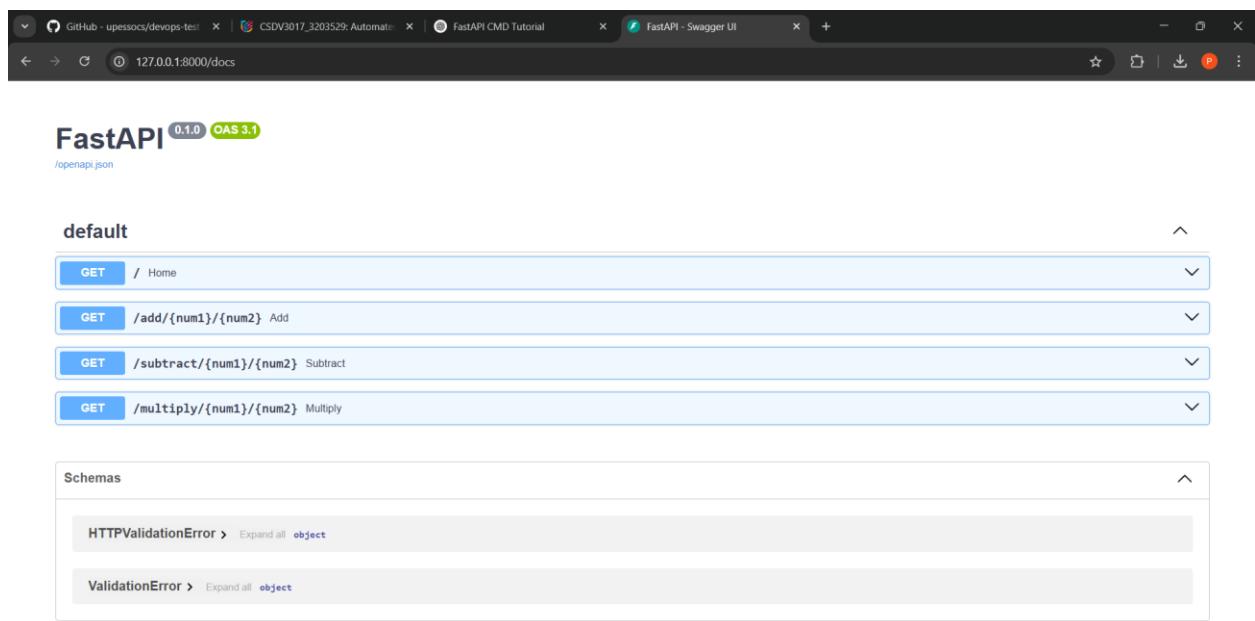
- **Multiplication:** `http://127.0.0.1:8000/multiply/10/5` → Should return {"result": 50}



```
127.0.0.1:8000/multiply/10/5
Pretty-print □
{"result":50}
```

📌 FastAPI also provides interactive documentation:

Open `http://127.0.0.1:8000/docs` in your browser to test the API using Swagger UI.



FastAPI 0.1.0 OAS 3.1
[/openapi.json](#)

default

GET / Home

GET /add/{num1}/{num2} Add

GET /subtract/{num1}/{num2} Subtract

GET /multiply/{num1}/{num2} Multiply

Schemas

HTTPValidationError > Expand all [object](#)

ValidationError > Expand all [object](#)

✓ Step 6: Automate API Testing

1. **Create another file** and save it as `testAutomation.py`.
2. **Copy and paste the following test script:**

✓ Step 6: Automate API Testing

Now that our FastAPI server is running, let's write a test script to **automate API testing**.

Steps to Follow

1. Create a new Python file and name it **testAutomation.py**.
2. Copy and paste the following test script:

```
python
CopyEdit
import requests

# Define test cases
testcases = [
    ("http://127.0.0.1:8000/add/10/5", 15, "Test addition of 10 and 5"),
    ("http://127.0.0.1:8000/subtract/10/5", 5, "Test subtraction of 10 and 5"),
    ("http://127.0.0.1:8000/multiply/10/5", 50, "Test multiplication of 10 and 5"),
    ("http://127.0.0.1:8000/add/-3/3", 0, "Test addition of -3 and 3"),
    ("http://127.0.0.1:8000/multiply/0/5", 0, "Test multiplication by zero"),
]

def test_api():
    """
    Runs automated tests on API endpoints.
    Asserts that the API response matches the expected result.
    """
    for url, expected, description in testcases:
        response = requests.get(url)
        result = response.json()["result"]

        assert result == expected, f"{description} FAILED! Expected {expected}, got {result}"
        print(f"{description} PASSED ✅")

    print("\n✅ All tests passed successfully!")

# Run the test function
if __name__ == "__main__":
    test_api()
```

Step 7: Run the Automated Tests

1. **Make sure your FastAPI server is running** (python apiserver.py).
2. Open a new **Command Prompt (CMD)**.
3. Navigate to the folder where testAutomation.py is saved:

```
cd path\to\your\file
```

4. Run the test script:

```
python testAutomation.py
```

```
C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt>python testAutomation.py
Test addition of 10 and 5 PASSED ✓
Test subtraction of 10 and 5 PASSED ✓
Test multiplication of 10 and 5 PASSED ✓
Test addition of -3 and 3 PASSED ✓
Test multiplication by zero PASSED ✓

✓ All tests passed successfully!
```

✓ Expected Output (if all tests pass):

```
Test addition of 10 and 5 PASSED ✓
Test subtraction of 10 and 5 PASSED ✓
Test multiplication of 10 and 5 PASSED ✓
Test addition of -3 and 3 PASSED ✓
Test multiplication by zero PASSED ✓
```

✓ All tests passed successfully!

✓ Step 8: Enhance Testing with Pytest

Instead of manually running the script, we can use **pytest** for better test automation.

Install pytest:

```
pip install pytest
C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt>pip install pytest
Requirement already satisfied: pytest in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (8.3.5)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest) (0.4.6)
Requirement already satisfied: iniconfig in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest) (2.1.0)
Requirement already satisfied: packaging in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest) (24.2)
Requirement already satisfied: pluggy<2,>=1.5 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest) (1.5.0)

[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Modify testAutomation.py for pytest:

Replace the previous code with this **pytest** version:

```
import pytest
import requests

# Define test cases for parameterized testing
testcases = [
    ("http://127.0.0.1:8000/add/10/5", 15, "Addition of 10 and 5"),
    ("http://127.0.0.1:8000/subtract/10/5", 5, "Subtraction of 10 and 5"),
```

```

        ("http://127.0.0.1:8000/multiply/10/5", 50, "Multiplication of 10 and 5"),
        ("http://127.0.0.1:8000/add/-3/3", 0, "Addition of -3 and 3"),
        ("http://127.0.0.1:8000/multiply/0/5", 0, "Multiplication by zero"),
    ]
}

@pytest.mark.parametrize("url, expected, description", testcases)
def test_api(url, expected, description):
    """
    Parameterized test for API endpoints.

    Parameters:
        url (str): The API endpoint URL.
        expected (int): The expected result.
        description (str): A descriptive message for the test.

    Returns:
        None
    """

    response = requests.get(url)
    result = response.json()["result"]
    assert result == expected, f"{description} FAILED! Expected {expected}, got {result}"
    print(f"{description} PASSED ✅")

if __name__ == "__main__":
    pytest.main()

```

Run pytest:

```

pytest testAutomation.py -v
C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt>pytest testAutomation.py -v
===== test session starts =====
platform win32 -- Python 3.11.1, pytest-8.3.5, pluggy-1.5.0 -- C:\Users\hp\AppData\Local\Programs\Python\Python311\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt
plugins: anyio-4.8.0
collected 5 items

testAutomation.py::test_api[http://127.0.0.1:8000/add/10/5-15-Addition of 10 and 5] PASSED [ 20%]
testAutomation.py::test_api[http://127.0.0.1:8000/subtract/10/5-5-Subtraction of 10 and 5] PASSED [ 40%]
testAutomation.py::test_api[http://127.0.0.1:8000/multiply/10/5-50-Multiplication of 10 and 5] PASSED [ 60%]
testAutomation.py::test_api[http://127.0.0.1:8000/add/-3/3-0-Addition of -3 and 3] PASSED [ 80%]
testAutomation.py::test_api[http://127.0.0.1:8000/multiply/0/5-0-Multiplication by zero] PASSED [100%]

===== 5 passed in 2.58s =====

```

✅ This makes testing easier and provides detailed reports.

✓ Step 9: Expanding for Real-World Use

How to Improve This Further?

- **Add Edge Cases:** Test negative numbers, non-integer inputs, and large values.
- **Use CI/CD:** Integrate with **GitHub Actions** or **Jenkins** for automated testing.
- **Mock Dependencies:** Use **responses** or **unittest.mock** to simulate external API calls.
- **Generate Reports:** Use **pytest-html** for detailed reports:

```
pip install pytest-html
```

```
C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt>pip install pytest-html
Collecting pytest-html
  Downloading pytest_html-4.1.1-py3-none-any.whl (23 kB)
Requirement already satisfied: jinja2>=3.0.0 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest-html) (3.1.5)
Collecting pytest-metadata>=2.0.0
  Downloading pytest_metadata-3.1.1-py3-none-any.whl (11 kB)
Requirement already satisfied: pytest>=7.0.0 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest-html) (8.3.5)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from jinja2>=3.0.0->pytest-html) (3.0.2)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest>=7.0.0->pytest-html) (0.4.6)
Requirement already satisfied: iniconfig in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest>=7.0.0->pytest-html) (2.1.0)
Requirement already satisfied: packaging in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest>=7.0.0->pytest-html) (24.2)
Requirement already satisfied: pluggy<2,>=1.5 in c:\users\hp\appdata\local\programs\python\python311\lib\site-packages (from pytest>=7.0.0->pytest-html) (1.5.0)
Installing collected packages: pytest-metadata, pytest-html
Successfully installed pytest-html-4.1.1 pytest-metadata-3.1.1

[notice] A new release of pip available: 22.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
pytest testAutomation.py --html=report.html
```

```
C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt>pytest testAutomation.py --html=report.html
=====
===== test session starts =====
platform win32 -- Python 3.11.1, pytest-8.3.5, pluggy-1.5.0
rootdir: C:\Users\hp\OneDrive\Desktop\Sem_6\Devops\Expt
plugins: anyio-4.8.0, html-4.1.1, metadata-3.1.1
collected 5 items

testAutomation.py .....
```

[100%]

```
===== Generated html report: file:///C:/Users/hp/OneDrive/Desktop/Sem_6/Devops/Expt/report.html =====
===== 5 passed in 0.41s =====
```

🎯 Final Summary

What We Achieved

- ✓ Set up a **FastAPI** server with endpoints (add, subtract, multiply).
- ✓ Created an automated test script using requests.
- ✓ Enhanced testing with pytest for better automation.
- ✓ Expanded the project for real-world usage with CI/CD, reporting, and mocking.

🚀 Next Steps

Now that you have **Backend API Test Automation** working on Windows, try:

- Adding more endpoints (e.g., division, square root).
- Implementing input validation (handle non-integer values).
- Scaling the project to integrate with databases.