

PyTest pass:

```
(venv) PS C:\Users\antho\OneDrive\Desktop\Adv Web Dev Labs\devops_flask_activity> pytest -q
.....
5 passed in 1.34s
(venv) PS C:\Users\antho\OneDrive\Desktop\Adv Web Dev Labs\devops_flask_activity> |
```

Git Commits working:

```
(venv) PS C:\Users\antho\OneDrive\Desktop\Adv Web Dev Labs\devops_flask_activity> git commit -m "Lint issue fix"
[main bfd6530] Lint issue fix
1 file changed, 4 insertions(+), 2 deletions(-)
(venv) PS C:\Users\antho\OneDrive\Desktop\Adv Web Dev Labs\devops_flask_activity> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 617 bytes | 154.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/AnthonyCiceuOTU/devops_flask_activity.git
35e62f6..bfd6530 main -> main
(venv) PS C:\Users\antho\OneDrive\Desktop\Adv Web Dev Labs\devops_flask_activity> pytest -q
```

Actions on GitHub also pass:

14 workflow runs				Event ▾	Status ▾	Branch ▾	Actor ▾
✔	Lint issue fix	DevOps Flask activity - CodeQL #4: Commit bfd6530 pushed by AnthonyCiceuOTU	main		Today at 2:42 PM 56s	...	
✔	Lint issue fix	DevOps Flask activity - Tests #4: Commit bfd6530 pushed by AnthonyCiceuOTU	main		Today at 2:42 PM 16s	...	
✔	Lint issue fix	DevOps Flask activity - Lint #4: Commit bfd6530 pushed by AnthonyCiceuOTU	main		Today at 2:42 PM 11s	...	

This is optional but I decided to test in POSTMAN too to make sure everything was working fine:

GET:

HTTP **http://127.0.0.1:5000/hello** Save Share

GET **http://127.0.0.1:5000/hello** Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK • 38 ms • 198 B • 🌐 ...

{} JSON Preview Visualize 🔍 📄 🔗

```
1 {
2   "message": "Hello, World!"
3 }
```

POST:

HTTP **http://127.0.0.1:5000/echo** Save Share

POST **http://127.0.0.1:5000/echo** Send

Params Authorization Headers (9) **Body** Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "msg": "ping",
3   "user": "Anthony"
4 }
```

Body Cookies Headers (5) Test Results 201 CREATED • 56 ms • 211 B • 🌐 ...

{} JSON Preview Visualize 🔍 📄 🔗

```
1 {
2   "msg": "ping",
3   "user": "Anthony"
4 }
```

Postbot Runner Start Proxy Cookies Vault Trash

PUT:

The screenshot shows a REST client interface with the following details:

- URL:** `http://127.0.0.1:5000/items/1`
- Method:** `PUT`
- Body (raw):**

```
1 {
2   "name": "example item",
3   "value": 42
4 }
```
- Response Status:** `200 OK` (10 ms, 240 B)
- Response Body (JSON):**

```
1 {
2   "data": {
3     "name": "example item",
4     "value": 42
5   },
6   "id": 1
7 }
```

The interface includes tabs for Params, Authorization, Headers (9), Body, Scripts, and Settings. The Body tab is active, showing the raw request and the JSON response. The response is also displayed in a preview view.

DELETE:

The screenshot shows a REST client interface with a dark theme. At the top, the URL bar displays `http://127.0.0.1:5000/items/1`. Below it, the HTTP method is set to `DELETE` and the same URL is entered in the request field. A `Send` button is visible on the right. The interface has tabs for `Params`, `Authorization`, `Headers (7)`, `Body` (selected), `Scripts`, and `Settings`. Under the `Body` tab, the content type is set to `raw`. The main editor area contains a single line of text: `1 Ctrl+Alt+P to Ask AI`. Below the editor, the response section shows a status of `200 OK` with a response time of `8 ms` and a size of `243 B`. The response body is displayed in `JSON` format, showing a nested object:

```
1 {
2   "deleted": {
3     "name": "example item",
4     "value": 42
5   },
6   "id": 1
7 }
```

 The bottom of the interface features a toolbar with icons for `Postbot`, `Runner`, `Start Proxy`, `Cookies`, `Vault`, `Trash`, and a help icon.

Trying to DELETE again but with no item there:

The screenshot shows a REST client interface with the following details:

- URL:** `http://127.0.0.1:5000/items/1`
- Method:** `DELETE`
- Body:** A text area containing the comment "Ctrl+Alt+P to Ask AI".
- Response:** A `404 NOT FOUND` status with a response time of 16 ms and a size of 204 B.
- Response Body (JSON):**

```
1 {
2   "error": "Item not found"
3 }
```

The interface includes tabs for Params, Authorization, Headers (7), Body, Scripts, and Settings. The Body tab is currently selected. The response body is displayed in JSON format, with options for Preview and Debug with AI. The bottom status bar shows various tool icons like Postbot, Runner, Start Proxy, Cookies, Vault, and Trash.