

CPSC 449 - Web Back-End Engineering

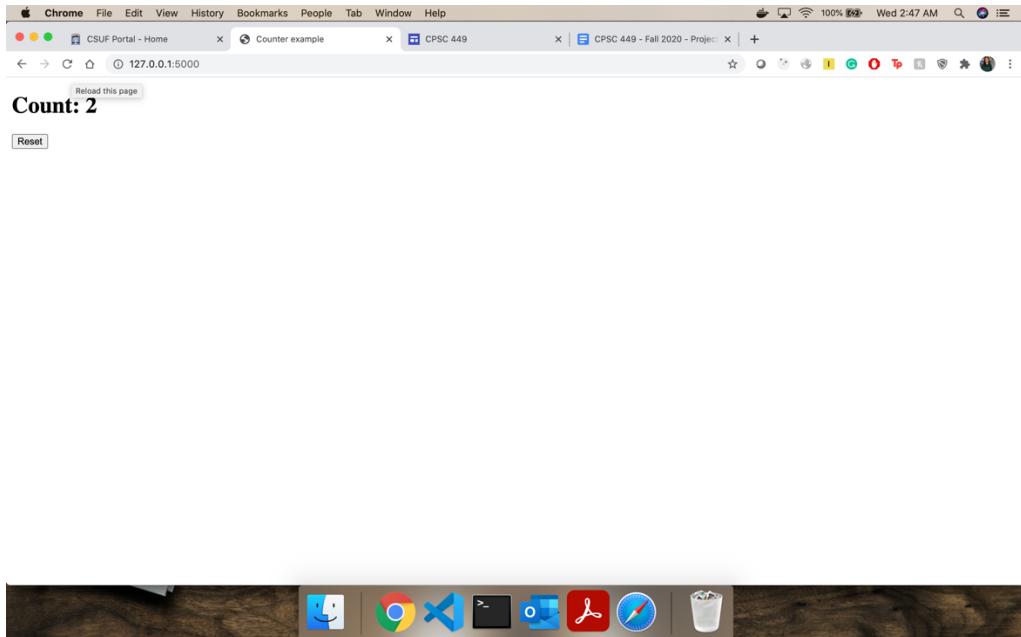
Project 3, Fall 2020

Contributors:

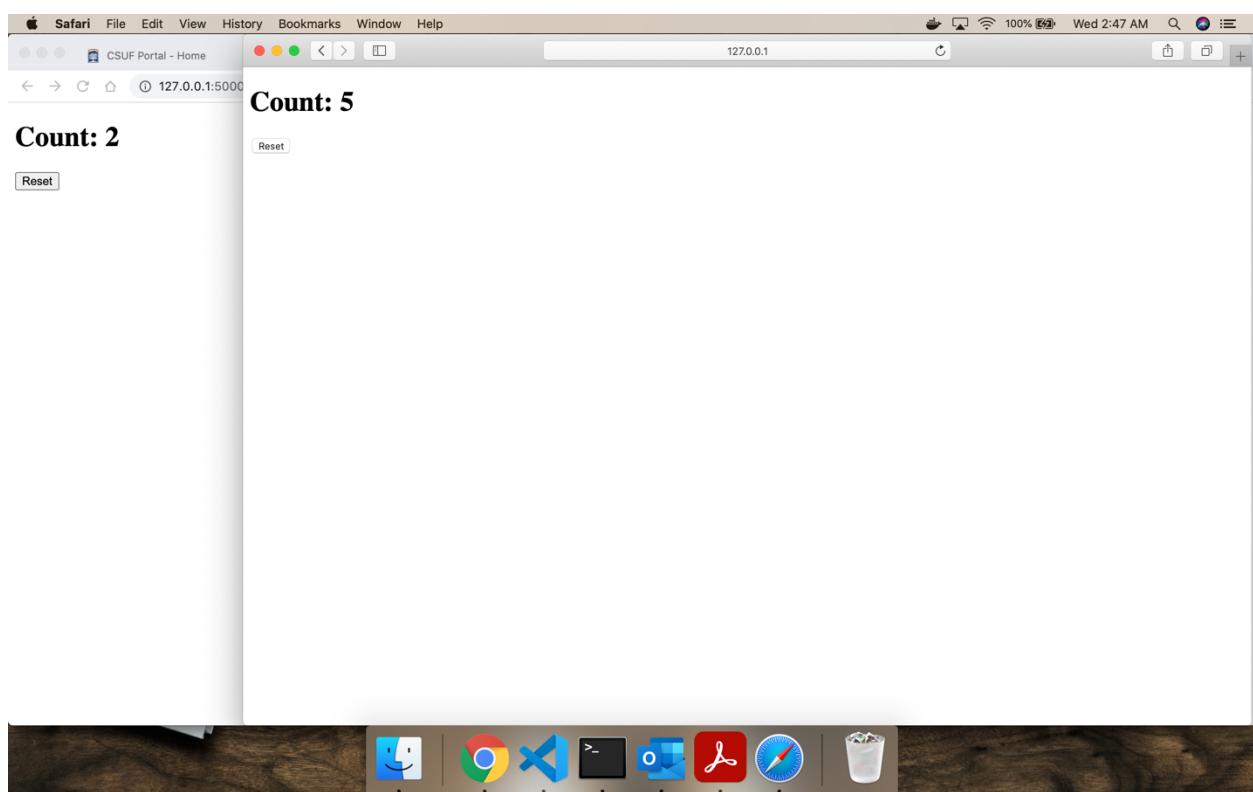
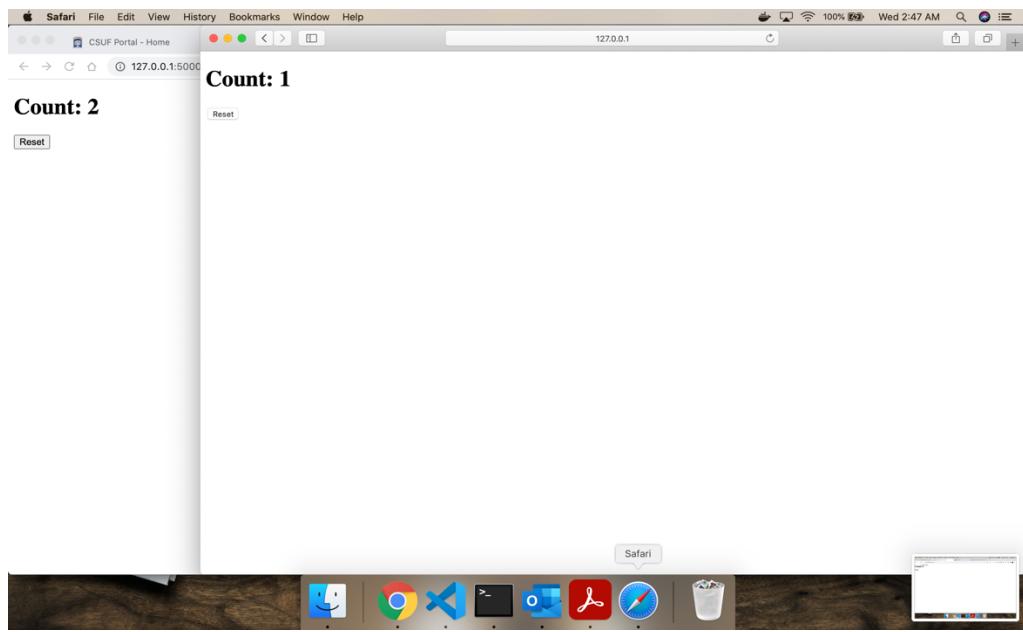
1. Sai Pavani Nagisetti (CWID: 888160793, Email: nagisettipavani@csu.fullerton.edu)
2. Rushabh Shah (CWID: 887456218, Email: rushabhshah@csu.fullerton.edu)

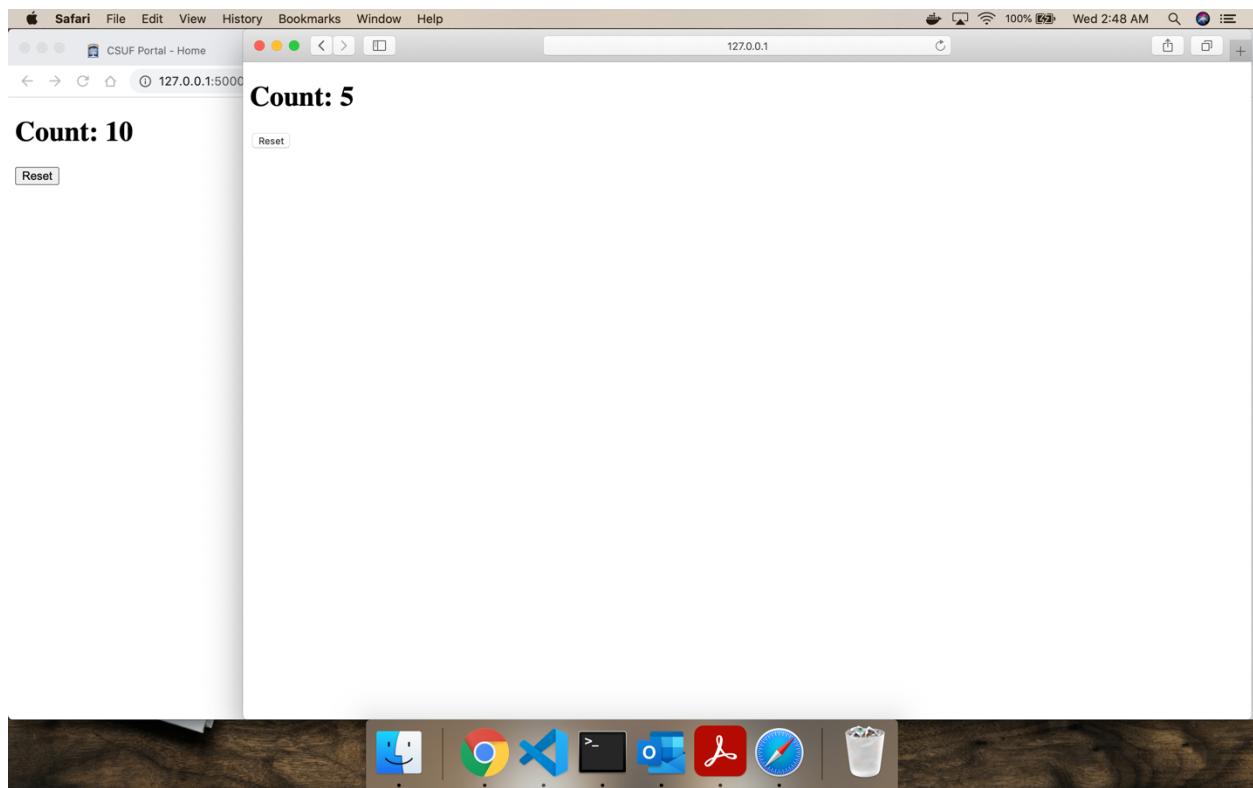
Cookie-based sessions:

2. Opened Google chrome browser and after running the application and refreshing it the count changes to 2.



Opened Safari browser in parallel and they maintain a different count as in the following screenshots





3. Each time we try to refresh the page we see different Set-Cookie: response header as in the following screenshots.

A screenshot of the Network tab in the Chrome DevTools developer console. The tab is selected, showing a timeline of network requests. A single request to "127.0.0.1" is highlighted, showing its details. The "Headers" section of the request details shows the following response headers:

```
Content-Length: 262
Content-Type: text/html; charset=utf-8
Date: Wed, 14 Oct 2020 09:52:24 GMT
Server: Werkzeug/1.0.1 Python/3.7.4
Set-Cookie: session=eyJjb3VudCI6Mn0.X4bKWA.tgY8E7yotZNN2fvWu3b442PE4Go; HttpOnly; Path=/
Vary: Cookie
```

The "Request Headers" section shows:

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.9
Accept-Encoding: gzip, deflate, br
```

At the bottom of the screenshot, the developer console navigation bar is visible with "Console", "What's New", and "Issues".

Chrome File Edit View History Bookmarks People Tab Window Help

CSUF Portal - Home Counter example CPSC 449 CPSC 449 - Fall 2020 - Project

127.0.0.1:5000

Count: 3

Network Performance Memory Application Security Lighthouse AdBlock Redux

Preserve log Disable cache Online

Filter Hide data URLs XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies Blocked Requests

5 ms 10 ms 15 ms 20 ms 25 ms 30 ms 35 ms 40 ms 45 ms 50 ms 55 ms 60 ms 65 ms 70 ms 75 ms 80 ms 85 ms 90 ms 95 ms 100 ms 105 ms 11

Name Headers Preview Response Initiator Timing Cookies

127.0.0.1

Remote Address: 127.0.0.1:5000
Referrer Policy: strict-origin-when-cross-origin

Response Headers Content-Length: 262 Content-Type: text/html; charset=utf-8 Date: Wed, 14 Oct 2020 09:52:58 GMT Server: Werkzeug/1.0.1 Python/3.7.4 Set-Cookie: session=eyJhbGciOiJIUzI1NiJ9.pq19wOHxql7Fsy9EM; HttpOnly; Path=/ Vary: Cookie

1 requests 520 B transferred Request Headers

Console What's New Issues



The screenshot shows a Chrome browser window with three tabs open: 'CSUF Portal - Home', 'Counter example', and 'CPSC 449'. The 'Counter example' tab displays the text 'Count: 4'. Below the browser is the Chrome DevTools Network tab, which lists a single request to '127.0.0.1'. The request details show headers like Content-Length: 262, Content-Type: text/html; charset=utf-8, and Date: Wed, 14 Oct 2020 09:52:58 GNT. The system tray at the bottom of the screen shows icons for Mail, Calendar, Photos, and other applications.

Key-value store service

5. Populated store with

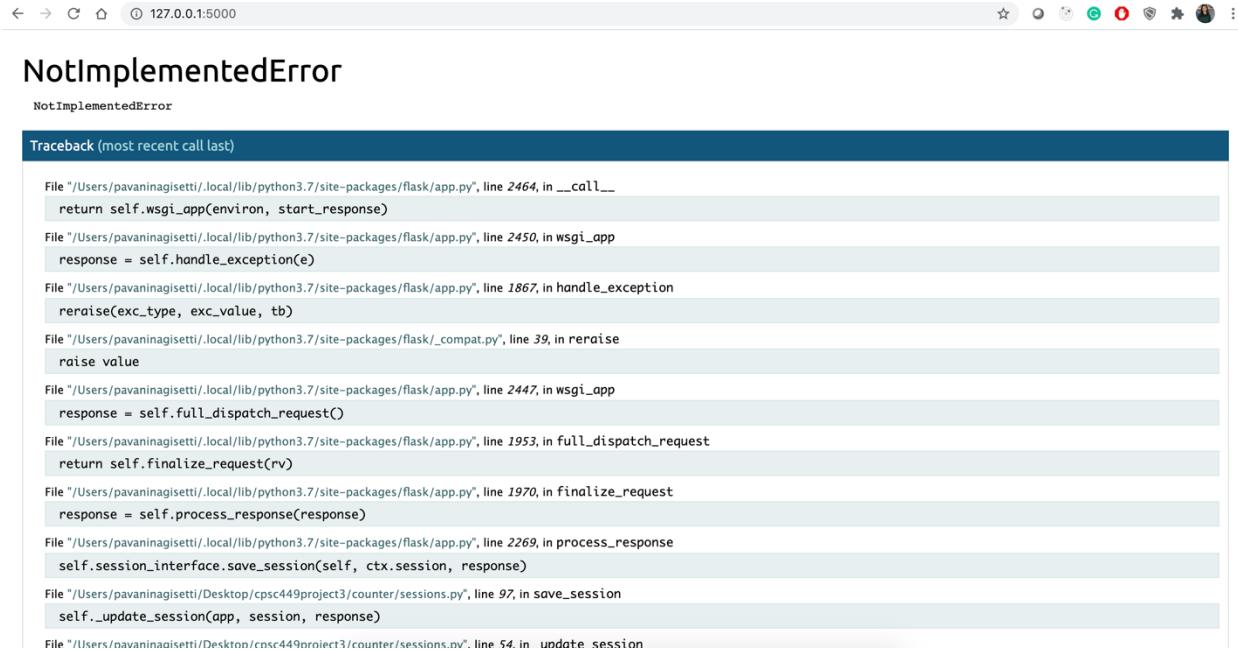
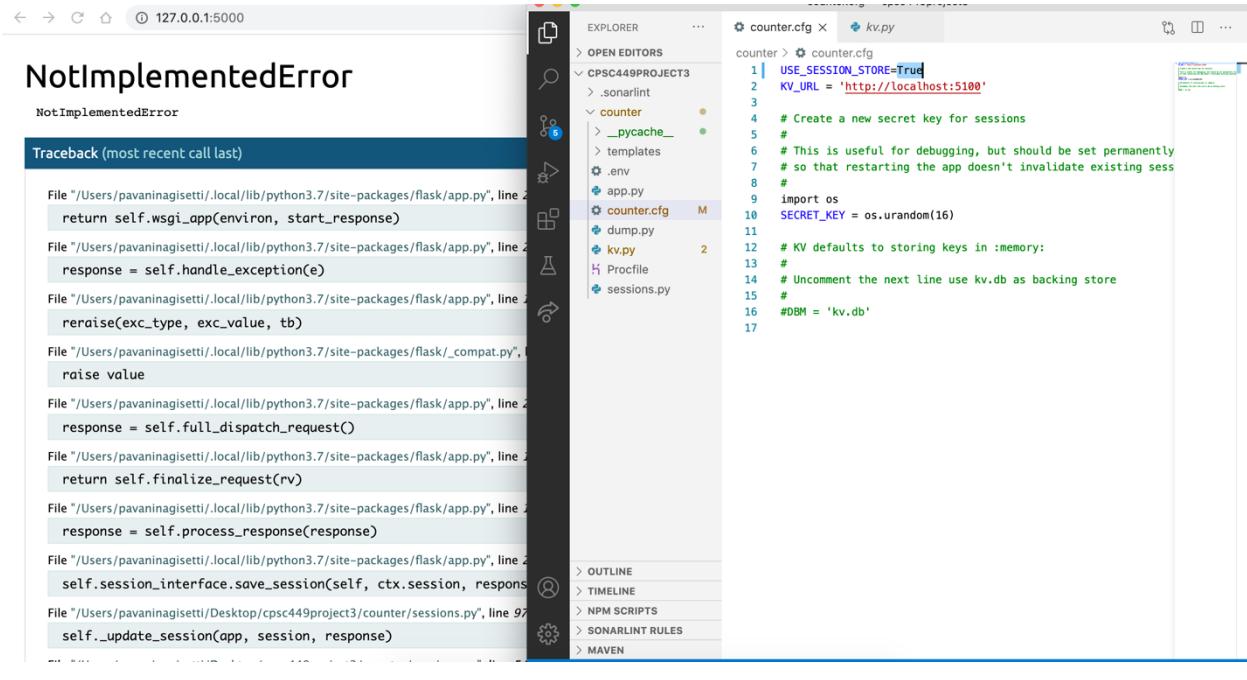
```
$ http localhost:5000 twitter=https://twitter.com/ProfAvery
$ http localhost:5000 github=https://github.com/ProfAvery
$ http localhost:5000 cpsc449=https://sites.google.com/view/cpsc449
```

and then running the dump.py script gives the following expected result

```
(base) sais-MBP:counter pavaninagisetti$ python dump.py http://localhost:5000
[
  {
    "twitter": "https://twitter.com/ProfAvery"
  },
  {
    "github": "https://github.com/ProfAvery"
  },
  {
    "cpsc449": "https://sites.google.com/view/cpsc449"
  }
]
```

Server-side sessions

`USE_SESSION_STORE` is modified to True, restarted counter app and the key-value store service with foreman start. On loading the counter page, we can see the “`NotImplementedError`”. The methods of `KeyValueSessionStore` have not yet been implemented.



```

file /Users/pavaninagisetti/.local/lib/python3.7/site-packages/flask/app.py , line 2447, in wsgi_app
    response = self.full_dispatch_request()
File "/Users/pavaninagisetti/.local/lib/python3.7/site-packages/flask/app.py", line 1953, in full_dispatch_request
    return self.finalize_request(rv)
File "/Users/pavaninagisetti/.local/lib/python3.7/site-packages/flask/app.py", line 1970, in finalize_request
    response = self.process_response(response)
File "/Users/pavaninagisetti/.local/lib/python3.7/site-packages/flask/app.py", line 2269, in process_response
    self.session_interface.save_session(self, ctx.session, response)
File "/Users/pavaninagisetti/Desktop/cpsc449project3/counter/sessions.py", line 97, in save_session
    self._update_session(app, session, response)
File "/Users/pavaninagisetti/Desktop/cpsc449project3/counter/sessions.py", line 54, in _update_session
    self.session_store.set_key(session.sid, dict(session))
File "/Users/pavaninagisetti/Desktop/cpsc449project3/counter/sessions.py", line 19, in set_key
    raise NotImplementedError()
NotImplementedError

```

NotImplementedError

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

- `dump()` shows all variables in the frame
- `dump(obj)` dumps all that's known about the object

Brought to you by DON'T PANIC, your friendly Werkzeug powered traceback interpreter.

9. Initially on load of page we can see that the Set-cookie has session id as in the below screenshot.

Now when trying to refresh the page, we see that the count increments to 2 and the session id remains the same.

The screenshot shows the Network tab in Chrome DevTools. A single request to 'localhost' is listed. The response headers include 'Content-Length: 262', 'Content-Type: text/html; charset=utf-8', 'Date: Wed, 21 Oct 2020 01:46:26 GMT', 'Server: Werkzeug/1.0.1 Python/3.7.4', and a 'Set-Cookie' header with the session ID. The request headers show standard browser settings like 'Accept', 'Accept-Encoding', 'Accept-Language', and 'Cache-Control'. The timeline at the top shows a single blue bar representing the duration of the request.

Now trying to refresh page two more times, count gets incremented to 3 and finally 4 but the session id remains the same as in the below screenshot.

The screenshot shows the Network tab in Chrome DevTools after multiple refreshes. Four requests to 'localhost' are listed, each with a different count value (1, 2, 3, 4) in the URL. The session ID remains constant across all requests. The response headers are identical to the first request, including the 'Set-Cookie' header with the same session ID. The request headers are also consistent. The timeline at the top shows four distinct blue bars corresponding to each request.

10. Now on clicking reset, we can see that the session id is reset in the POST request and subsequent GET request generates a new session id as in the below screenshots.

The screenshot shows the Network tab in Chrome DevTools. A single request is listed: a POST to `http://localhost:5000/reset`. The response status is 302 FOUND. The response headers include `Content-Length: 209`, `Content-Type: text/html; charset=utf-8`, `Date: Wed, 21 Oct 2020 01:51:16 GMT`, `Location: http://localhost:5000/`, `Server: Werkzeug/1.0.1 Python/3.7.4`, and `Set-Cookie: session=; Expires=Thu, 01-Jan-1970 00:00:00 GMT; Max-Age=0; Path=/`.

The screenshot shows the Network tab in Chrome DevTools. A single request is listed: a GET to `http://localhost:5000/`. The response status is 200 OK. The response headers include `Content-Length: 262`, `Content-Type: text/html; charset=utf-8`, `Date: Wed, 21 Oct 2020 01:51:16 GMT`, `Server: Werkzeug/1.0.1 Python/3.7.4`, and `Set-Cookie: session=46bbc93-5680-4de9-9a63-21754adbd91a; HttpOnly; Path=/`.

11. Running dump.py outputs two active session ids and count values for different browsers (Chrome & Safari) as in below screenshot

