# Open-Source Report:
# Parsing HTTP Headers

Proof of knowing your stuff in CSE312

## Guidelines

Provided below is a template you must use to write your reports for your project.

Here are some things to note when working on your report, specifically about the **General Information & Licensing** section for each technology.

- **Code Repository**: Please link the code and not the documentation. If you'd like to refer to the documentation in the **Magic** section, you're more than welcome to, but we need to see the code you're referring to as well.
- **License Type**: Three letter acronym is fine.
- **License Description**: No need for the entire license here, just what separates it from the rest.
- **License Restrictions**: What can you *not* do as a result of using this technology in your project? Some licenses prevent you from using the project for commercial use, for example.

Also, feel free to extend the cell of any section if you feel you need more room.

If there's anything we can clarify, please don't hesitate to reach out! You can reach us using the methods outlined on the course website or see us during our office hours.

## Flask

### General Information & Licensing

| | |
|---|---|
| Code Repository | https://github.com/pallets/flask<br>https://github.com/eventlet/eventlet |
| License Type | BSD-3-Clause, MIT license |
| License Description | ● Open-source license and allows free use<br>● Permits modification and distribution<br>● provides a patent grant copies of the Software |

| | |
|---|---|
| License Restrictions | ● Redistributions of source code must retain the above copyright notice. |

## *Magic* ⋆★⁺₊ °.° ☽ ˚⁀➷。˚★彡⁺★ ༄

self.handle() calls an object from class HttpProtocol, which can handle connections by self.handle. Line 350
https://github.com/eventlet/eventlet/blob/master/eventlet/wsgi.py

Function “handle” uses a while loop to call self.handle_one_request(), until the server closes. Line 379
https://github.com/eventlet/eventlet/blob/master/eventlet/wsgi.py

Function “handle_one_request” can read the header, by creating FileObjectForHeaders to store the self.rfile data. Line 413
https://github.com/eventlet/eventlet/blob/master/eventlet/wsgi.py

Class SecureCookieSession used to add a ``Vary: Cookie` to the header, which allows caching proxies to cache different pages for different users. Line 47
https://github.com/pallets/flask/blob/main/src/flask/sessions.py

The class SessionInterface can implement and save the session. Line 106
https://github.com/pallets/flask/blob/main/src/flask/sessions.py

Function “get_cookie_name” can be used to get the name of the cookie, which can verify the username in our server, then the user does not need to login every time. Line 177.
https://github.com/pallets/flask/blob/main/src/flask/sessions.py

Function “get_cookie_path” can return the path for which the cookie should be valid, then the cookie is only valid in this path. Line 194
https://github.com/pallets/flask/blob/main/src/flask/sessions.py

Function “open session” called at the beginning of each request, after pushing the request context, before matching the URL. Line 250
https://github.com/pallets/flask/blob/main/src/flask/sessions.py

Class SecureCookieSessionInterface interface that stores sessions in signed cookies through the :mod:`itsdangerous` module. In the class, the salt is applied on the secret key for the signing of cookie based sessions. Line 284
https://github.com/pallets/flask/blob/main/src/flask/sessions.py

In function “add_url_rule”, if the methods are not given and the view_func object knows its methods we can use that instead.  If neither exists, we go with a tuple of only ``GET`` as default. Line 1020.
https://github.com/pallets/flask/blob/main/src/flask/app.py

In the function handle_http_exception, it can handle an HTTP exception, by default this will invoke the registered error handlers and fall back to returning the exception as response. Line 1261
https://github.com/pallets/flask/blob/main/src/flask/app.py