Open-Source Report (HTTP Parsing)

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/
License Type	BSD 3-Clause
License Description	• Redistributions are permitted with or without modification • Including Commercial uses • Any redistribution must maintain the same license
License Restrictions	• Contributors are not liable for any damages caused by the software • Names of contributors shall not be used to promote derived software w/o permission

Magic

Dispel the magic of this technology. Replace this text with some that answers the following questions for the above tech:

- How does this technology do what it does? Please explain this in detail, starting from after the TCP socket is created
    - After a TCP socket is created, a user can send an HTTP request to a route created in flask by using “@app.route()” to define the route (while taking arguments for the route path itself and types of requests it can receive) and defining a function inside of it to handle the request. Once inside the route function, we will have access to the “request” object, which contains a dictionary with all the parsed headers (and forms and cookie headers) that can be accessed by calling .get after calling another function on the .request object depending on the type of information you want from the HTTP headers (i.e. request.headers.get(“headername”) would get the value of an http header named “headername”, request.cookies.get(“cookiename”) would get the cookie of “cookiename” in the cookie header.

- Where is the specific code that does what you use the tech for? You ***must*** provide a link to the specific file in the repository for your tech with a line number or number range.
    - Initialized Flask app at NN-CSE312/app.py: line 21
    - Handled routes with app.route function at NN/CSE-312/app.py lines 70-222
    - Got specific cookies from cookie header at NN/CSE-312/app.py lines 77-178
    - HTTP request split into header keys and values by Flask library in flask/src/flask/typing.py lines 9-38