# TCP Open Source Report

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/fcap25/CSE312-Project |
|---|---|
| License Type | BSD |
| License Description | <ul><li>Redistributions of source code must retain copyright notice</li><li>The name of the copyright holder/contributors may not be used to endorse products created by this software</li></ul> |
| License Restrictions | <ul><li>Flask-SSO is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.</li><li>It's worth noting that while the BSD-3-Clause Source License is a permissive open-source license, it does not include any warranty or guarantee of fitness for a particular purpose.</li></ul> |

# *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

- Flask does not directly handle TCP Connections, rather it is designed to handle communication over HTTP protocol. However, under the hood, Flask does use the underlying TCP socket infrastructure to communicate with clients over the network. When a Flask application is started, it typically listens on a specific TCP port for incoming HTTP requests. This involves creating a TCP socket that is bound to the server's IP address and the specified port. When a client sends an HTTP request to the server, the request is received by the TCP socket created by the Flask application. The Flask application then reads the request data from the socket and processes it using the appropriate Python function or method. Once the response is generated, it is sent back to the client using the same TCP socket. Flask works hand in hand with outside libraries to handle TCP Sockets.

- 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.
    - If there is more than one step in the chain of calls *(hint: there will be)*, you must provide links for the entire chain of calls from your code, to the library code that actually accomplishes the task for you.
    - Example: If you use an object of type HttpRequest in your code which contains the headers of the request, you must show exactly how that object parsed the original headers from the TCP socket. This will often involve tracing through multiple libraries and you must show the entire trace through all these libraries with links to all the involved code.

*This section will likely grow beyond the page