# Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

## Guidelines

Provided below is a template you must use to write your report for each of the technologies you use in 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'd like 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.
- **Who worked with this?**: It's not necessary for the entire team to work with every technology used, but we'd like to know who worked with what.

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.

## websockets

### General Information & Licensing

| | |
|---|---|
| Code Repository | https://github.com/aaugustin/websockets/ |
| License Type | BSD 3-Clause "New" or "Revised" License |
| License Description | • Private/Commercial Use Permitted<br>• Modification Permitted<br>• Distribution Permitted |
| License Restrictions | • Creators are not liable for damages code may produce<br>• No warranty is provided<br>• License/Copyright notice must be provided in licensed material |
| Who worked with this? | Colin |

*Use as many of the sections below as needed, or create more, to explain every function, method, class, or object type you used from this library/framework.*

# websocket.serve

## Purpose

- Creates the websocket server at a given IP address/port, and sends incoming messages to a customizable callback function
- This will be used to initialize the websocket server within websocket.py of our repo (line 34)

- an instance of serve is created by calling websockets.serve.
- Serve creates a WebSocketServer, which creates, holds, and destroys WebSocketServerProtocols.
- Serve also creates a WebSocketServerProtocol, which receives, processes, and sends out data to the server
- The handler for WebSocketServerProtocol waits for a handshake to commence, holding the entire handshake process in the handshake function
- The protocol created waits for an attempt at a valid handshake request, which is determined by reading the raw bytes and attempting to convert into a HTTP request using the read_http_request function.
- The proper HTTP handshake response is validated and processed by handshake, which after reading the HTTP request builds the response using build_response and writes it back to the client through write_http_response.

# websocket.broadcast

## Purpose

On a pre-existing websocket, send takes a string or bytes, formats it into a websocket frame, and then sends it to the client.
Broadcast is used to send messages to a variable amount of active users across multiple channels. Regardless of how they are chosen to be delivered, all usages of broadcast (lines 26 and 27 of websocket.py in our repo) echo messages sent over the channels back to other users listening into that channel, or to other channels (like the user's notification channel).