TCP Connections 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 / Python

General Information & Licensing

Code Repository	https://github.com/emilydesantis/cse312-because-the-internet
License Type	MIT
License Description	• Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. • M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.
License Restrictions	• Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. • 2. Redistributions in binary form must reproduce the

	above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. ● 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. ● THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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
- 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

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask began as a wrapper around Werkzeug and Jinja and has become now one of the most popular Python web application frameworks.

Our code doesn't explicitly establish a TCP connection but it is handled by the Flask and Flask-SocketIO libraries.

Flask-SocketIO enables seamless integration with Socket.IO, a powerful protocol that facilitates real-time communication between clients and servers. The integration is established through a standard HTTP handshake, which upgrades to WebSocket protocol if available. If not, it falls back to HTTP long polling to ensure uninterrupted communication. WebSocket, in turn, is a widely used protocol built on top of TCP, designed for efficient and secure communication.

In line 15 once the SQLAlchemy object is initialized, that will be used to interact with our database. Once we attempt to interact with our database to perform operations such as in line 41: (users = User.query.all()) a TCP connection is established.

Throughout the code the function emit() is called which is a part of the Flask-SocketIO Library that allows the server to send messages to the client.
Lines: 141, 142, 143, 155, 158, 179, 191, 205, 208, 209, 211, 215, 217, 218, 226, 227, 237, 247, 248, 256, 277, 280, 286, 292
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L401

In line 135 once a client sends a ("create_room") socketio.on event, the function handle_create_room(data) is called to create the room for the two users.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 148 once a client sends a ("join_room") socketio.on event the function handle_join_room(data) is called so that a user can join a created room.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 164 once a client sends a (“join_lobby”) socketio.on event the function join_lobby(data) is called so that a user can join the lobby.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 182 once a client sends a (“submit_word”) socketio.on event the function handle_submit(data) is called once a user makes a guess .
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 194 once a client sends a (“submit_question_or_guess”) socketio.on event the function handle_submit_question_or_guess(data) is called when a user submits a guess or a question.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 222 once a client sends a (“submit_answer”) socketio.on event the function handle_submit_answer(data) is called once a user makes a final guess.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 230 once a client sends a (“start_game”) socketio.on event the function handle_start_game(data) is called once the host user starts the game.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 244 once a client sends a (“start_question_round”) socketio.on event the function handle_start_question_round(data) is called so that once the game begins and there hasn't been an item selected yet for a guess.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 253 once a client sends a (“correct_guess”) socketio.on event the function handle_correct_guess(data) is called so that once a user makes a correct guess.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 265 once a client sends a (“select_role”) socketio.on event the function handle_select_role(data) is called once two users have to pick their roles.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 283 once a client sends a (“hide_ask”) socketio.on event the function handle_hide_ask(data) is called so once the host user hasn’t selected an item so that the user guessing isn’t able to answer yes or no to the questions.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L258

In line 301 a web server is started when socketio.run() runs. This server listens for incoming HTTP and Websocket connections on port 8080, operating on top of TCP.
https://github.com/miguelgrinberg/Flask-SocketIO/blob/main/src/flask_socketio/__init__.py#L553