

# 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.

## Flask

### General Information & Licensing

Code Repository	<a href="https://github.com/pallets/flask">https://github.com/pallets/flask</a> , <a href="https://github.com/pallets/flask/blob/main/src/flask/templating.py">https://github.com/pallets/flask/blob/main/src/flask/templating.py</a>
License Type	BSD License
License Description	<ul style="list-style-type: none"><li>• A low restriction type license usually for open source, freeware, and shareware software.</li><li>• Other than the three restrictions listed below, there are no other requirements for redistribution.</li></ul>
License Restrictions	<ul style="list-style-type: none"><li>• "Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.</li><li>• Redistributions in binary form must reproduce the above</li></ul>

	<p>copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.</p> <ul style="list-style-type: none"> <li>• The names of the contributors may not be used to endorse or promote products derived from this software without specific prior written permission.” - Sourced from Flask Documentation; <a href="https://flask.palletsprojects.com/en/0.12.x/license/#flask-license">https://flask.palletsprojects.com/en/0.12.x/license/#flask-license</a></li> </ul>
Who worked with this?	Armin Ronacher and contributors

*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.*

`route (rule, **options)`

## Purpose

Replace this text with some that answers the following questions for the above tech:

- What does this tech do for you in your project?
  - Route essentially simplifies how http requests are sent. It allows the creator to input on a specific route, located in the parameter, there are also optional parameters where you can list that there will be GET and/or POST requests. It allows for easy processing, receiving and sending of data.
- Where specifically is this tech used in your project? Give us some details like file location and line number, if applicable. If too cumbersome, a general description of where it's used for a given purpose is fine as well.
  - Route is used in app.py which can be considered the main file that starts up the 'app' and website. It is being used on;
    - Line 9
    - Line 15
    - Line 41

\*This section may grow beyond the page for many features.

The relevant lines can be found in the app.py file. Flask works by creating a Flask class from the Scaffold class when a Flask application is first initialized (Line 7 app.py). The class takes a import\_name parameter so the flask application knows what files are relevant to the application. The server is started by invoking the run method on the initialized app from the Flask class (Line 79 App.py). The run method optional takes a HOST and PORT number. The run method in the Flask class then calls another library werkzeug.serving ([Line 917](#) in flask class library) and calls the method run\_simple (811 serving.py). Inside the run\_simple method an inner method is called which then calls make\_server (line 950 serving.py) passing our specified port and host number and then creates an object of class BaseWSGIServer (Line 652 serving.py) which is of class HTTPserver and uses libraries that we've seen on the homework such as the socket library. Once the server object is returned by make\_server, then the serve\_forever method is invoked on the returned object to start our server.

Then the route method can be called on our object of Class Flask (Line 16, 22, etc app.py). The route method calls the method add\_url\_rule ([line 439](#) Scaffold.py) which registers a rule for routing incoming requests and builds the url. The route decorator is shortcut for defining the view function. For example, in line 16 & 17 in app.py login is the view function. This is the same as making the call app.add\_url\_rule("/", view\_func = login). An endpoint is then set for the specified URL default named to the view\_func.

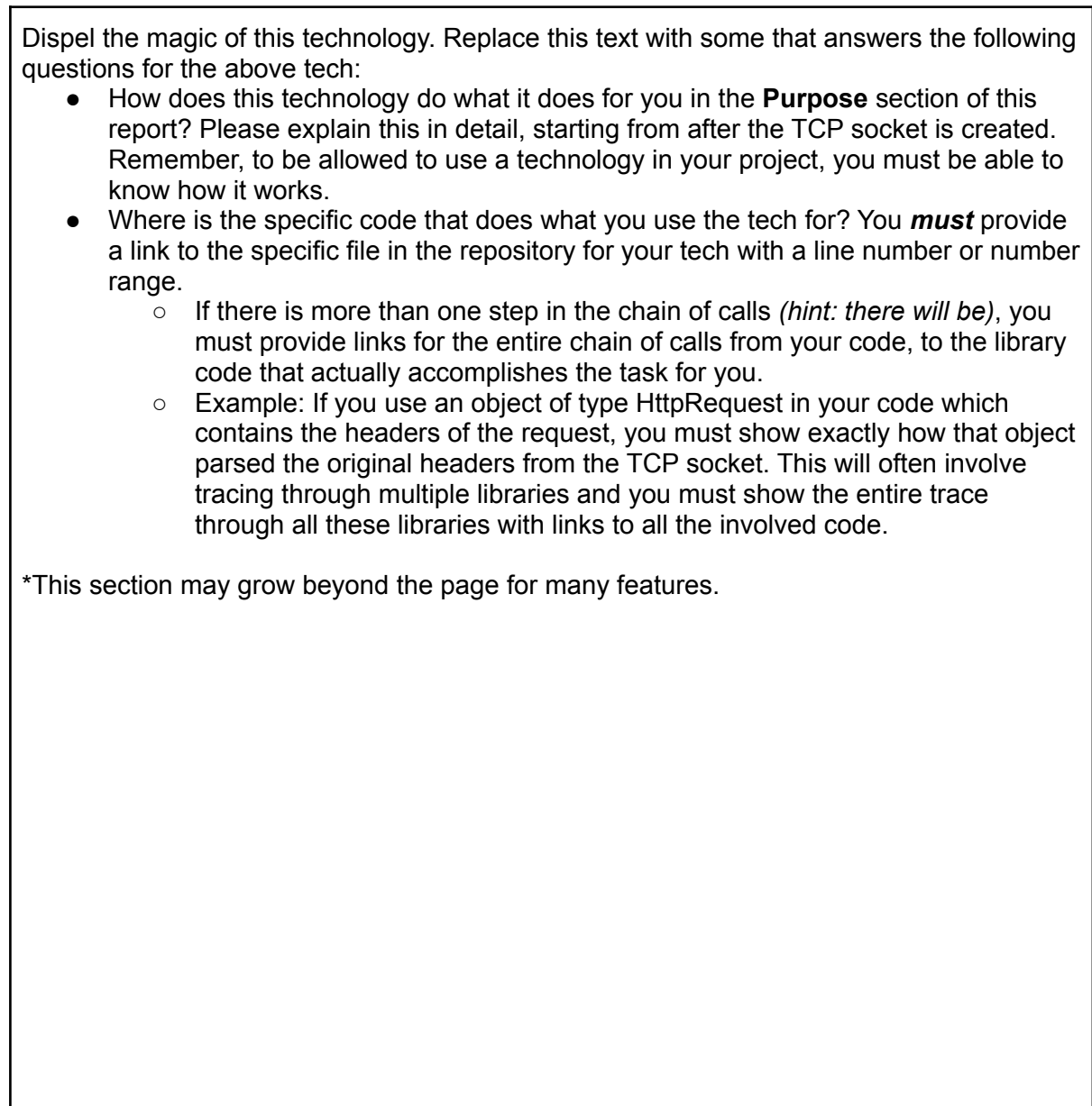
We also use render\_template which renders a template from a template folder with various parameters. This is the same as reading the file as bytes and sending it over when we did it on the homework. But now, they call the \_render method which takes the updated template with the listed parameters and if there is additional context, renders completely including that.

[insert method/function/class/object name here]

## Purpose

Replace this text with some that answers the following questions for the above tech:

- What does this tech do for you in your project?
- Where specifically is this tech used in your project? Give us some details like file location and line number, if applicable. If too cumbersome, a general description of where it's used for a given purpose is fine as well.



## Purpose

- What does this tech do for you in your project?
- Where specifically is this tech used in your project? Give us some details like file location and line number, if applicable. If too cumbersome, a general description of where it's used for a given purpose is fine as well.

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 for you in the **Purpose** section of this report? Please explain this in detail, starting from after the TCP socket is created. Remember, to be allowed to use a technology in your project, you must be able to know how it works.
- 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 may grow beyond the page for many features.

[insert method/function/class/object name here]

## Purpose

Replace this text with some that answers the following questions for the above tech:

- What does this tech do for you in your project?
- Where specifically is this tech used in your project? Give us some details like file location and line number, if applicable. If too cumbersome, a general description of where it's used for a given purpose is fine as well.

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 for you in the **Purpose** section of this report? Please explain this in detail, starting from after the TCP socket is created. Remember, to be allowed to use a technology in your project, you must be able to know how it works.
- 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 may grow beyond the page for many features.