

# Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

Instagram 2.0 CSE 312 Project

## 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>
License Type	BSD
License Description	<ul style="list-style-type: none"><li>• It gives permission for anyone else to use it for commercial use, modification, distribution, and private use.</li><li>• The copyright holder or contributors are not liable for any damages</li></ul>
License Restrictions	<ul style="list-style-type: none"><li>• The name of the copyright holder nor the names of the contributors can be used to endorse or promote products derived from Flask without written permission.</li><li>• Redistributions in binary form must reproduce the</li></ul>

	copyright notice <ul style="list-style-type: none"> <li>• Redistribution of source code needs the copyright notice</li> </ul>
Who worked with this?	Tirth, Vedant, Aditya, Abi

*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.*

## @app.route("")

### Purpose

What does this tech do for you in your project?

Flask allows us to route our requests to the appropriate files in our public directory. More specifically, we have the `@app.route("/")` to route to path "/", and so on for all the other paths. It also allows us to choose methods that we can have access so it limits it to that. It allows you to return many different things as we can do with our headers. We can return templates, scripts, or various other things that we normally do in our headers.

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.

The `@app.route()` function is being used in our `route.py` file to route requests via their appropriate paths. We have multiple routes, such as `account`, `home`, `logout`, and more which has different functionality respectively.



Flask.make\_response() -> Flask.render\_template()

## Purpose

Used Extensively in Routes.py

Render\_template(): Replaces the index.html pages' placeholder with the appropriate strings. This is used in routes.py, after each call to the path in order to render the page.

Render\_template() also calls a function named make\_response(), which is also used in our code when we need to set cookies as in the response as well.

Template inheritance allowed us to import templates from a base file called base.html,

Magic ★★🌙🍀🌟🌀

<https://www.geeksforgeeks.org/template-inheritance-in-flask/>

# Flask.set\_cookie

## 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?  
This allows us to set\_cookie to our headers when we send out the data back to the client.
- 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.  
We use this feature in our routes.py file where we set the cookie to the number of visits that have been to this page.

## Magic

Dispel the magic of this technology. Replace this text with some that answer 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 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.

### 1. Purpose

This function takes in the least 2 arguments that are key and value, and more cookies arguments that you might want to pass. It then adds these values to the headers in the response object. It calls a dump\_cookie() function which creates a 'Set-Cookie' header without the 'set-cookie' prefix.' which then returns the headers in bytes.

### 2. Links

<https://github.com/pallets/werkzeug/blob/main/src/werkzeug/http.py>

line~ 1218 onwards

<https://github.com/pallets/werkzeug/blob/main/src/werkzeug/sansio/response.py>

line ~ 232 onwards as well as line 127

<https://github.com/hahshtrit/solid-server/blob/master/routes.py>

line ~ 51 onwards

# Flask Form management

## 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?

This tech allows us to receive the data that the clients submit so that we can host it on our webpage. It collects the headers from the form and parses the relevant data, and creates a file storage object that holds all our data.

- Where specifically is this tech used in your project?

This global variable is used in our route.py file, and used to receive forms that users upload to their account.

## 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 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.

This technology takes in the request that the TCP sends into its request class, and then that class contains multiple methods for parsing forms and different type of data. The specific method that we use are form and files. The form method calls `_load_form_data()`, and this function parses all the data that is received from the client and stores all the relevant information. Then the form function returns that and it is a dictionary. For the files function, it is similar to form, but it instead creates a `ImmutableMultiDict[str, FileStorage]`, and that stores the name of the input you had on the form, and creates a file storage class, which stores all relevant details of the files. The function files calls another function `_load_form_data`, which creates a `FormDataParser` object, which stores the filename, filesize, file type and all relevant information related to the file. We then call `file read`, which essentially reads the data in the file.

- 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.

<https://github.com/hahshtrit/solid-server/blob/master/routes.py> ~

<https://github.com/hahshtrit/solid-server/blob/master/templates/signup.html> ~ 6 onwards

<https://github.com/pallets/werkzeug/blob/main/src/werkzeug/datastructures.py>

~848 onwards

<https://github.com/pallets/flask/blob/main/src/flask/wrappers.py> ~15

<https://github.com/pallets/flask/blob/main/src/flask/globals.py> line 55~

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

## Flask.redirect()

### Purpose

- This allows us to redirect the file, by sending a response that elicits a 302 response code, which is better than entirely rendering the template from scratch. Basically the same thing as we did in the redirects for HW 1



