Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

Flask

General Information & Licensing

Code Repository	https://github.com/pallets/flask
License Type	BSD 3-Clause "New" or "Revised" License

License Description	• A permissive license similar to the BSD 2-Clause License, but with a 3rd clause that prohibits others from using the name of the project or its contributors to promote derived products without written consent.
	• Permissions:
	  ○ Commercial use
	  ○ Modification
	  ○ Distribution
	  ○ Private use

License Restrictions	• Liability • Warranty
Who worked with this?	Eric

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.

class Flask

Purpose

- What does this tech do for you in your project?
  - This tech handles all the TCP connections as well as HTTP request handling
- 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.
  - This tech is used in the __init__.py file that is ran when main.py is ran
  - Flask's functionality will be used in almost every python file that makes up our project

Magic

Flask is the main class involved with the creation of the web application. It is defined at https://github.com/pallets/flask/blob/main/src/flask/app.py line 97
In our project it is used in the __init__.py file at the root of the repository where the application is initialized on line 4.

```
3   def create_app():
4       app = Flask(__name__)
```

It is a simple line of code that assigns the Flask object to the variable app with the parameter __name__. According to the source code:

> It is passed the name of the module or package of the application. Once it is created it will act as a central registry for the view functions, the URL rules, template configuration and much more. The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an :file:`__init__.py` file inside) or a standard module (just a ``.py`` file).

The create_app() function is called from main.py line 3 at the repo's root

```
1   from website import create_app
2
3   app = create_app()
4
5   if __name__ == '__main__':
6       app.run(debug=True)
```

When main.py gets ran, the run() method is called which is a method of the Flask class and is defined at https://github.com/pallets/flask/blob/main/src/flask/app.py line 804
This run() method is responsible for running the application on a local development server. Since the Flask framework is based on Werkzeug Web Server Gateway Interface library, it imports run_simple from werkzeug.serving and calls that method with the appropriate arguments for a local server creation (host = 127.0.0.1, port = 5000)
run_simple() is defined at https://github.com/pallets/werkzeug/blob/main/src/werkzeug/serving.py line 790.
Within this method, inner() is defined and ran which calls werkzeug's make_server() method defined at https://github.com/pallets/werkzeug/blob/9af52dbbfaa289ccd2b3034d5e3e51fc1e2bad97/src/werkzeug/serving.py#L744 line 744
If a threading or forking server isn't required, the BaseWSGIServer is returned which is a server object defined at https://github.com/pallets/werkzeug/blob/9af52dbbfaa289ccd2b3034d5e3e51fc1e2bad97/src/werkzeug/serving.py#L633 line 633
This server object is based on python's HTTPServer object which is defined athttps://github.com/python/cpython/blob/3.10/Lib/http/server.py line 132
This HTTPServer object is based on socketserver.TCPServer which is exactly where the TCP connections for the Flask framework are handled
The HTTPServer object is also responsible for handling and sending all HTTP requests, it has its own parsers and response builders similar to what is done on the homeworks.