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.

python-dotenv

General Information & Licensing

Code Repository	https://github.com/theskumar/python-dotenv
License Type	https://github.com/theskumar/python-dotenv/blob/master/LICENSE -
License Description	• Redistribution and use in source and binary forms allowed ○ With or without modification •
License Restrictions	• Neither the name of python-dotenv nor its contributors can be used to endorse or promote products derived from python-dotenv unless written permission is given
Who worked with this?	Zaki

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.

load_dotenv

Purpose

The load_dotenv() function is used to load environment variables from the .env file. This file contains the information necessary for our image uploading functionality. Once the environment variables are loaded, we use os.getenv() to get their values and send them to cloudinary to configure with.

Magic

- We call load_dotenv() in line 14 of postHandlers.py, at the start of the uploadImage function. This function is defined in this file: https://github.com/theskumar/python-dotenv/blob/45848bb780c26ef0adf7898656f7d3d3f4e2d8ae/src/dotenv/main.py in line 300. It takes multiple parameters that we do not send, so it sets them itself. The conditional at line 321 will evaluate to true because we did not provide a dotenv_path and stream. This means that find_dotenv() will be called, which will return the path to the .env file. Once the path is known, a dotenv object is created using the DotEnv class' init function. After this object is created, the function will return the result of dotenv.set_as_environment_variables(). We do not necessarily care about the return of this function; what matters the most is that it runs, because this function sets the key value pairs stored in the dotenv object's dict as environment variables.

- find_dotenv() is defined in the same file as load_dotenv(), at line 259. The conditional at line 275 determines if the function is running in a REPL environment. It will evaluate to false, so the code from line 279 to 287 will run. This code will use the current file and directory name to find the .env file. Once it is found it will return the path to .env.

- DotEnv.__init__() creates an instance of the DotEnv class. It is sent the path to the .env file we need, as well as other parameters. A

- Set_as_environment_variables is defined at this line: https://github.com/theskumar/python-dotenv/blob/45848bb780c26ef0adf7898656f7d3d3f4e2d8ae/src/dotenv/main.py#L86. This function goes through the dictionary named dict that every DotEnv object has. This dictionary holds the key-value pairs in the .env file that we want to add to the set of environment variables. If a key-value pair is not in the current set of environment variables (os.environ), then that pair is added. This is the step that finally loads the environment variables from .env into the current environment variable space.