Application Building
Flask App (Step - 2)

Date	03 October 2022
Team ID	PNT2022TMID34894
Project Name	Project - Web Phishing Detection

Creating The Application Directory:

To begin, you need to create the directory that will host the example code, which is available in a GitHub repository.

As discussed in “How to Work with the Example Code” on page xiii, the most convenient way to do this is by checking out the code directly from GitHub using a Git client. T

he following commands download the example code from GitHub and initialize the application to version 1a, which is the initial version you will work with:

$ git clone https://github.com/miguelgrinberg/flasky.git

$ cd flasky

$ git checkout 1a

If you prefer not to use Git and instead manually type or copy the code, you can sim- ply create an empty application directory as follows:

$ mkdir flasky

$ cd flasky

Creating A Virtual Environment With Python 3:

The creation of virtual environments is an area where Python 3 and Python 2 inter- preters differ. With Python 3, virtual environments are supported natively by the venv package that is part of the Python standard library.

If you are using the stock Python 3 interpreter on an Ubuntu Linux system, the standard venv package is not installed by default. To add it to your system, install the python3-venv package as follows:

$ sudo apt-get install python3-venv

The command that creates a virtual environment has the following structure:

$ python3 -m venv virtual-environment-name

The -m venv option runs the venv package from the standard library as a standalone script, passing the desired name as an argument.

You are now going to create a virtual environment inside the flasky directory.

A com- monly used convention for virtual environments is to call them venv, but you can use a different name if you prefer.

Make sure your current directory is set to flasky, and then run this command:

$ python3 -m venv venv

After the command completes, you will have a subdirectory with the name venv inside flasky, with a brand-new virtual environment that contains a Python inter- preter for exclusive use by this project.

Creating A Virtual Environment With Python 2:

Python 2 does not have a venv package.

In this version of the Python interpreter, vir- tual environments are created with the third-party utility virtualenv.

Creating a Virtual Environment with Python 3 | 3 Make sure your current directory is set to flasky, and then use one of the following two commands, depending on your operating system. If you are using Linux or macOS, the command is:

$ sudo pip install virtualenv

If you are using Microsoft Windows, make sure you open a command prompt win- dow using the "Run as Administrator" option, and then run this command:

$ pip install virtualenv

The virtualenv command takes the name of the virtual environment as its argu- ment. Make sure your current directory is set to flasky, and then run the following command to create a virtual environment called venv:

$ virtualenv venv
New python executable in venv/bin/python2.7 Also creating executable in venv/bin/python Installing setuptools, pip, wheel...done.

A subdirectory with the venv name will be created in the current directory, and all files associated with the virtual environment will be inside it.