PREREQUISTES

ANACONDA:

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012.As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

JUPYTER NOTEBOOK:

Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension.

Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

JupyterLab is a newer user interface for Project Jupyter, offering a flexible user interface and more features than the classic notebook UI. The first stable release was announced on February 20, 2018. In 2015, a joint $6 million grant from The Leona M. and Harry B. Helmsley Charitable Trust, The Gordon and Betty Moore Foundation, and The Alfred P. Sloan Foundation funded work that led to expanded capabilities of the core Jupyter tools, as well as to the creation of JupyterLab.

JupyterHub is a multi-user server for Jupyter Notebooks. It is designed to support many users by spawning, managing, and proxying many singular Jupyter Notebook servers.

SPYDER:

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software.It is released under the MIT license.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Spyder is extensible with first-party and third-party plugins,includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings.QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend.

VISUAL STUDIO CODE:

Visual Studio Code, also commonly referred to as VS Code,is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that they use it.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran.It is based on the Electron framework,which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

LIBRARIES USED:

FLASK-Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

render_template is a Flask function from the flask. templating package. render_template is used to generate output from a template file based on the Jinja2 engine that is found in the application's templates folder. Note that render_template is typically imported directly from the flask package instead of from flask.

REQUEST-The data from a client's web page is sent to the server as a global request object. In order to process the request data, it should be imported from the Flask module. Form − It is a dictionary object containing key and value pairs of form parameters and their values.

NumPy-NumPy (pronounced /ˈnʌmpaɪ/ (NUM-py) or sometimes /ˈnʌmpi/ (NUM-pee)) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.[6] The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy is a NumFOCUS fiscally sponsored project.

OpenCV-OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

Matplotlib-Visualization with Python.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Create publication quality plots.

Make interactive figures that can zoom, pan, update.

Customize visual style and layout.

Export to many file formats.

Embed in JupyterLab and Graphical User Interfaces.

Use a rich array of third-party packages built on Matplotlib.

Try Matplotlib (on Binder)

Tenserflow- TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

A **Keras model** consists of multiple components:

The architecture, or configuration, which specifies what layers the model contain, and how they're connected.

A set of weights values (the "state of the model").

An optimizer (defined by compiling the model).

A set of losses and metrics (defined by compiling the model or calling add_loss() or add_metric()).

The Keras API makes it possible to save all of these pieces to disk at once, or to only selectively save some of them:

Saving everything into a single archive in the TensorFlow SavedModel format (or in the older Keras H5 format). This is the standard practice.

Saving the architecture / configuration only, typically as a JSON file.Saving the weights values only. This is generally used when training the model.

Werkzeug.utils-Werkzeug is a comprehensive WSGI web application library. It began as a simple collection of various utilities for WSGI applications and has become one of the most advanced WSGI utility libraries.

It includes:

An interactive debugger that allows inspecting stack traces and source code in the browser with an interactive interpreter for any frame in the stack.

A full-featured request object with objects to interact with headers, query args, form data, files, and cookies.

A response object that can wrap other WSGI applications and handle streaming data.

A routing system for matching URLs to endpoints and generating URLs for endpoints, with an extensible system for capturing variables from URLs.

HTTP utilities to handle entity tags, cache control, dates, user agents, cookies, files, and more.

A threaded WSGI server for use while developing applications locally.

A test client for simulating HTTP requests during testing without requiring running a server.

Werkzeug doesn’t enforce any dependencies. It is up to the developer to choose a template engine, database adapter, and even how to handle requests. It can be used to build all sorts of end user applications such as blogs, wikis, or bulletin boards.

Flask wraps Werkzeug, using it to handle the details of WSGI while providing more structure and patterns for defining powerful applications.