

File Integrity Monitor – Environment Setup

Responsibility : Aakash R – 20BCE1003

Purpose : PBL – Review 1

1. **Version Control System (VCS): Git** is the version control System used to develop this project. The branches are named '**Version.X**' to work concurrently on the product features. Higher numbers indicate subsequent versions of the product. The branches are then merged to the '**master**' branch. The '**master**' branch reflects the latest version of the product.
2. **Code Editor/IDE: Microsoft Visual Studio Code (VSCode)** is the code editor used to develop this project.
3. **Programming Language and Runtime Environment:** The product is developed using **Python** a versatile and powerful programming language. Python 3.9.6 is the interpreter version used to develop this project.
4. **Package Manager: Pip** is the package manager used. pip is the default package manager for Python, used for installing, managing, and updating Python packages.
5. **Collaboration Tool: GitHub** is used to collaborate with other developers. It's a powerful tool to facilitate communication and coordination among team members.
6. **Operating System:** The project is developed in the **Windows 10 & Windows 11** operating system.
7. **Packages & Libraries**
 - a. **os** - The os module provides a way to interact with the operating system. It allows you to perform various operations related to file and directory management, process management, and system-related functionalities.
 - b. **smtplib** - The smtplib module is used for sending emails using the Simple Mail Transfer Protocol (SMTP). It provides functions for connecting to an SMTP server, sending emails, and handling email-related operations.

- c. **imghdr** - The imghdr module provides functions for determining the type of image files. It can be used to validate and process image files in different formats, such as JPEG, PNG, GIF, etc
- d. **email.message** - The EmailMessage class from the email.message module is used to create and manipulate email messages in Python, allowing you to set various email properties such as sender, recipient, subject, body, attachments, etc.
- e. **pickle** - The pickle module provides functions for serializing and deserializing Python objects. Serialization is the process of converting Python objects into a binary format that can be stored in a file or transmitted over a network, and deserialization is the reverse process of recreating Python objects from binary data.
- f. **datetime** - The datetime module provides classes for working with dates, times, and time intervals in Python. It includes various useful functions for formatting and manipulating dates and times, such as getting the current date and time, calculating time differences, formatting dates and times, etc.
- g. **prettytable** - The PrettyTable class from the prettytable module is used for creating formatted and visually appealing ASCII tables in Python. It provides methods for adding rows, columns, setting table properties such as alignment and style, and rendering the table to a string for display or saving to a file.
- h. **hashlib** - The hashlib module provides functions for generating secure hash functions in Python. Hash functions are used to convert input data into a fixed-size string of characters, which is typically a hexadecimal representation of the hash value. The hashlib module supports various hash algorithms, such as MD5, SHA-1, SHA-256, and more, which can be used for tasks such as data integrity verification, password storage, and digital signatures.
- i. **pefile** - The pefile module is a library for parsing and analyzing PE (Portable Executable) files, which are the standard executable file format used in Windows operating systems. The pefile module provides functions for extracting information from PE files, such as header information, section information, import/export tables, resource information, and more. It can be used for tasks such as reverse engineering, malware analysis, and binary file analysis.
- j. **subprocess** - The subprocess module provides a way to create and interact with additional processes from within a Python script. It allows you to spawn new processes, connect to their input/output/error pipes, and communicate with them. It is commonly used for tasks such as running external programs, executing system commands, and managing child processes from a Python script.

- k. **re** - The re module stands for regular expressions. It provides functions for working with regular expressions, which are a powerful and flexible way to perform pattern matching and text manipulation in Python. The re module allows you to search, match, and replace strings using regular expressions, making it useful for tasks such as text processing, data validation, and pattern extraction.
- l. **requests** - The requests module is a popular library for making HTTP requests in Python. It provides a convenient and high-level interface for sending HTTP requests, handling responses, and managing cookies, headers, and authentication. The requests module is widely used for tasks such as web scraping, API integration, and web service testing.
- m. **argparse** - The argparse module provides a convenient way to parse command-line arguments and options in Python. It allows you to define the arguments and options that your script should accept, specify their types, default values, and help messages, and automatically generates usage messages and error handling. The argparse module is commonly used for creating command-line interfaces for Python scripts and applications, making them more user-friendly and interactive.
- n. **time** - The time module provides functions for working with time-related operations in Python. It includes functions for getting the current time, formatting time values, sleeping or delaying execution of code, and measuring elapsed time. The time module is commonly used for tasks such as timing events, benchmarking, and scheduling operations.
- o. **json** - The json module provides functions for encoding and decoding JSON (JavaScript Object Notation) data in Python. JSON is a lightweight data interchange format that is commonly used for exchanging data between a client and a server, or for storing configuration data in a human-readable format. The json module allows you to convert Python objects to JSON format (serialization) and vice versa (deserialization).