**Installation and Usage Guide**

**System Requirements**

1. **Operating System**:
   * Windows, macOS, or Linux with support for Python 3.8+.
2. **Software Requirements**:
   * Python 3.8+ installed on your system.
   * PyCharm IDE (recommended for editing and running Python files).
3. **Dependencies**:
   * Python libraries: pandas, matplotlib, seaborn, dash, sqlite3, fpdf, plotly, BeautifulSoup4, requests,json,Counter and other libraries.
   * Ensure internet access for API requests and external resource downloads.

**Installation Steps**

1. **Clone the Repository**:
   * Clone the repository from GitHub or Simply download the zip and extract it from the link:

git clone <repository-link>

cd <repository-folder>

* Link -: [https://github.com/Prathameshmane1710/Digital\_Forensic\_project\_repository](%20https:/github.com/Prathameshmane1710/Digital_Forensic_project_repository)

1. **Set Up Python Environment**:
   * Create and activate a virtual environment (optional but recommended):

python -m venv venv

source venv/bin/activate # For macOS/Linux

venv\Scripts\activate # For Windows

1. **Install Required Libraries**:
   * Install the dependencies listed in the requirements.txt file:

pip install -r requirements.txt

1. **Set Up Environment Variables**:
   * For geolocation API integration, ensure valid API keys are set in your environment variables or directly in the script configuration files.
2. **Open the Project in PyCharm**:
   * Open PyCharm and load the project directory. Ensure the Python interpreter is configured to use your virtual environment.

**Usage Instructions**

**1. Data Collection**

* **Script**: API\_data.py
* **Directory**: scraper and extractor
* **Purpose**: Fetch phishing website URLs from the PhishTank API.
* **Command**: (Run the commands or simply run from IDE)

python scraper and extractor/API\_data.py

**2. Scraping Metadata**

* **Script**: scraper\_extractor.py
* **Directory**: scraper and extractor
* **Purpose**: Extract metadata (IP addresses, geolocation, links, forms, etc.) from phishing websites.
* **Command**:

python scraper and extractor/scraper\_extractor.py

**3. Analyze Data**

* **Script**: pattern\_analyzer.py
* **Directory**: analyzer
* **Purpose**: Perform data analysis to extract insights (e.g., most targeted regions, suspicious links).
* **Command**:

python analyzer/pattern\_analyzer.py

**4. Data Visualization**

* **Scripts**:
  + data\_cleaning.py
  + visualizer.py
  + dashboard.py
* **Directory**: visualizer
* **Purpose**: Generate visualizations and interactive dashboards for data exploration.
* **Command**:

python visualizer/data\_cleaning.py

python visualizer/visualizer.py

python visualizer/dashboard.py

**5. Repository Setup**

* **Script**: repository.py
* **Directory**: repository
* **Purpose**: Set up and manage an SQLite database for forensic queries.
* **Command**:

python repository/repository.py

**6. Generate Comprehensive Report**

* **Script**: Forensic\_Report.py
* **Directory**: reports
* **Purpose**: Generate a PDF report summarizing the project's findings and insights.
* **Command**:

python reports/Forensic\_Report.py

**Troubleshooting**

1. **Missing Dependencies**:
   * Run the following to install any missing library:

pip install <library-name>

1. **API Configuration Issues**:
   * Ensure that valid API keys (e.g., ipinfo.io) are properly configured in the scripts.
2. **Internet Access**:
   * Verify your internet connection for data scraping and API requests.
3. **PyCharm Issues**:
   * Ensure the Python interpreter is set to the correct virtual environment under **File > Settings > Project: <Project-Name> > Python Interpreter**.
4. **Database Issues**:
   * Ensure SQLite is properly installed, and no permission errors occur when accessing database files.
5. **Invalid URLs**:
   * If URLs fail during scraping, the script is designed to skip invalid entries. Check Output logs for detailed errors.