WildFire Incident Logger

By: Mahir Sadique and Yohanse Nurga

Goal of the Project

The Wildfire Data Logger is a software program that automates the logging, administration, and analysis of wildfire incidents. It allows users to register incidents interactively or using mock data generation, making it suitable for both real-world and demonstration purposes. The program calculates critical data such as average severity, response time, and total affected areas, providing useful information about wildfire trends. Detailed reports, including risk evaluations based on incident severity and affected areas, are created to aid decision-making. The user-friendly interface, which is enhanced by the Rich library, provides straightforward interaction and visually beautiful results, making the tool a crucial resource for analysts and decision-makers involved in wildfire management and resource allocation.

Significance of the Project

Wildfires pose a major threat to ecosystems, communities, and resources, so it's critical to track incidents immediately and correctly. This project creates an easy-to-use platform for tracking and evaluating wildfire incidents, allowing emergency responders, environmentalists, and lawmakers to make educated decisions. Its emphasis on clear, interactive features and visually stunning outputs using the stunning library sets it apart. Unlike other tools, it is lightweight, easy to update, and scaleable to meet a variety of requirements, making it a practical and effective wildfire management solution.

# Installation and Usage Instructions

Installation:
- Ensure Python 3.8 or later is installed
- Install the latest Pycharm version(2024) and open and run "Hello World" to check if the Pycharm runs the program
- Install the required library and in our case install the [rich] library
- Next easily copy and paste the source code from the GitHub repository of the main file to Pycharm

Usage Instructions:
- Run the program
- After running 4 options are given

- Select from the menu options and interact with the program
- When done interacting press 4 to save and exit the program
- The program will ask if you are sure to exit to confirm
- Press y to exit and press n to stay
- After saving the logged incidents to a JSON file, it can be found out in the saved file of your machine and will be updated everytime you log a new incident

Structure of the code
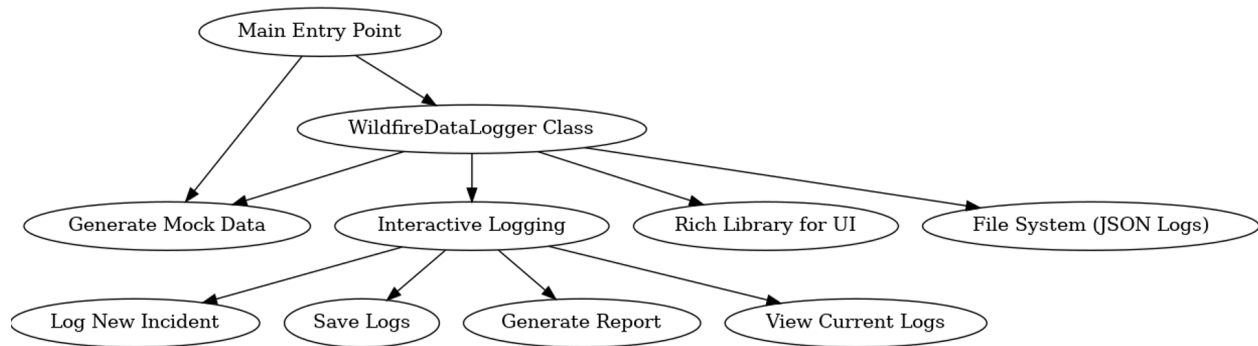
Main Program: Initializes WildfireDataLogger, generates mock data, and starts interactive mode.
WildfireDataLogger Class:

- Logs incidents, saves data to JSON, generates reports, and assesses risk.

Core Functions:

- log_incident: Adds a new incident.
- save_log: Saves logs to a timestamped JSON file.
- generate_report: Analyzes data and provides insights.
- _log_new_incident: Guides user to log incidents interactively.
- generate_mock_data: Creates mock incidents for testing.

Dependencies:

- **Rich Library**: For enhanced UI (tables, panels, error handling).
- **File System**: Manages JSON log storage.

Core Functionalities

Interactive Logging: Users can input wildfire incident data, including location, severity, response time, area affected, and resources deployed.
Mock Data Generation: Generates realistic incident data for testing or demonstration purposes.
Data Visualization: Displays incidents and reports in well-structured tables with statistical summaries.
Report Generation: Computes and displays metrics like average severity, highest severity, total area affected, and risk level.
Risk Assessment: Provides a clear risk evaluation based on incident severity and affected area.
Data Export: Saves logged incidents to a JSON file for future reference.

List of functionalities and verification results

Below we have screenshots of the output of the program.
- This shows that the user entered 2 to view the current logs and the program shows the current logs of wildfire reported previously

This result shows the overall wildfire report in summary

The following demonstrates logging new incident

```
 [1]  Log New Incident
 [2]  View Current Logs
 [3]  Generate Report
 [4]  Save and Exit
Select an option [1/2/3/4] (1): 1
Select incident location [Yosemite National Park, CA/Sequoia National Forest,
CA/Mount Hood National Forest, OR/Crater Lake National Park, OR/Coconino
National Forest, AZ/Saguaro National Park, AZ/Olympic National Park, WA/Mount
Rainier National Park, WA/Rocky Mountain National Park, CO/Mesa Verde National
Park, CO/Yellowstone National Park, WY/Grand Teton National Park, WY/Zion
National Park, UT/Bryce Canyon National Park, UT/Sawtooth National Forest,
ID/Craters of the Moon, ID/Everglades National Park, FL/Ocala National Forest,
FL]: Olympic National Park, WA
Enter severity (1-5): 2
Enter response time (minutes): 30
Enter area affected (square kilometers): 2
Enter resources deployed: 6
───────────────────────── Incident Logged Successfully ─────────────────────────
```

```
───────────────────────── Incident Logged Successfully ─────────────────────────
 {
   "timestamp": "2024-12-06T10:27:11.996758",
   "location": "Olympic National Park, WA",
   "severity": 2,
   "response_time": 30.0,
   "area_affected": 2.0,
   "resources_deployed": 6
 }
```

The following confirms that the incident is logged in saved in the JSON file for future reference

```
🔥 WILDFIRE INCIDENT LOGGER 🔥
 [1]  Log New Incident
 [2]  View Current Logs
 [3]  Generate Report
 [4]  Save and Exit
Select an option [1/2/3/4] (1): 4
Are you sure you want to save and exit? [y/n]: y
✓ Incidents logged to wildfire_logs\wildfire_log_20241206_102603.json
```

Showcasing the achievement of project goals

Results:

1. **Interactive Incident Logging**:
   ○ Users can log wildfire incidents with real-time validation for severity, response time, area affected, and resources deployed.
   ○ Mock data generation provides pre-filled incidents for testing and demonstration.
2. **Detailed Reporting**:
   ○ Reports include total incidents, average severity, response times, total area affected, and a risk assessment.
   ○ Rich Library visuals (tables and panels) enhance readability and engagement.
3. **Data Storage**:
   ○ All incidents are stored in organized JSON files with timestamped filenames for traceability.
4. **Risk Assessment**:
   ○ Provides a risk level (Low, Moderate, High) based on average severity and total area affected, aiding decision-making.

Implementation Highlights:

- **Goal Achievement**:

- - Automates wildfire incident management by integrating data logging, analysis, and reporting into a single tool JSON.
    - Ensures usability with a user-friendly interface and visually appealing outputs.
  - **Efficiency**:
    - Simplifies incident tracking with interactive prompts and validation.
    - Enables real-time insights for emergency responders and decision-makers.

Discussion and Conclusions

Our project effectively applies software design principles to create a practical tool for wildfire management. It provides a streamlined, user-friendly way to log, analyze, and report wildfire incidents, meeting its objectives with a solid foundation for real-world use.

Issues and Limitations:

While the tool is functional, there are areas for improvement. It handles moderate data well but may struggle with larger datasets, which limits scalability. The risk assessment uses simple thresholds and could benefit from more advanced modeling. Additionally, the program does not integrate with external data sources, such as APIs for real-time wildfire tracking, and lacks advanced visualizations like charts or maps.

Operating System Concepts:

The project leverages the **file system** to organize and store log data efficiently. It also showcases practical **input/output operations**, such as capturing user inputs and saving files, demonstrating real-world use of operating system functionality.

In conclusion, the project is a valuable step toward efficient wildfire data management. Although it has limitations, it provides a solid base for future enhancements, including better scalability, integration with external systems, and richer visualizations.