LOG FILE ANALYSIS

Shaik Suraj

24B0907

April 2025

Contents

1 Introduction

This project aims for the conversion of Apache log file to csv file and also generates plots for better visualization, analysis and interpretation of data. In addition, the project features a table filter that allow users to filter data based on log level, event id, time. The project also includes a plot filter with options to filter by time and to generate plots based either on the original log data or on the filtered table data. Also the users can download the generated csv files and plots, which makes it easier to interpret the log file.

2 Running Instructions

Follow the steps below to run the Flask-based web application locally:

1. **Installing Python and its Libraries(Flask, Numpy, Matplotlib):**
   Ensure Python 3.x is installed on your system. You can download it from https://www.python.org/downloads/. Open a terminal or command prompt and install Flask by running:

   ```
   pip install Flask
   ```

2. **Navigate to the Project Directory:**
   Move into the project folder where `main.py` is located.

3. **Run the Flask Server:**
   Start the server using the following command:

   ```
   python3 main.py
   ```

   Or you could use below based on whatever python version you have, Just run main.py file

   ```
   python main.py
   ```

4. **Accessing the Application:**
   After running, the terminal will display a local URL, typically:

   ```
   http://127.0.0.1:5000/
   ```

   Open this URL on your browser to access the application.

3 Website Layout

3.1 Home Page

On running the application, you will see a page that has option for choosing the file (refer 1). After that click the upload button (which is in the form of image) (refer 2) to upload the file. You will be redirected to download page if the file is of correct format otherwise it will show error and redirects to the Home Page.

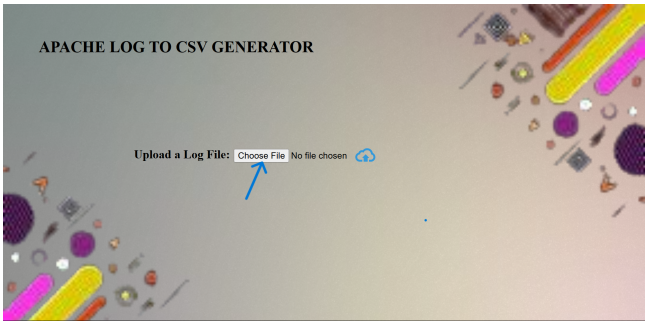

Figure 1: Selecting file

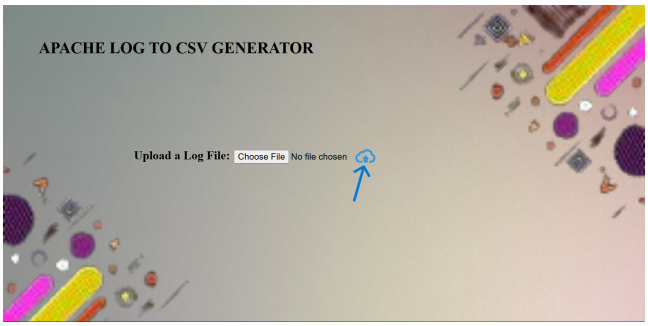

Figure 2: Uploading file

3.2 Download Page

3.2.1 Downloading Files

In this download page, you can click the respective buttons for downloading the files (refer 3). They are

1. Download CSV File: The csv formatted version of the given log file.

2. Download Table Filter CSV File: The csv file of the present table filter (default is original file if no filter is given).

3. Download Notice CSV File: The csv file consisting of only notice lines.

4. Download Error CSV File: The csv file consisting of only error lines.

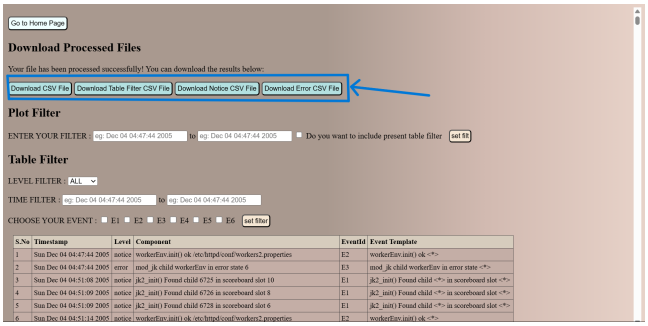

Figure 3: Download Files

It also has an option to go back to Home Page. (refer 4)

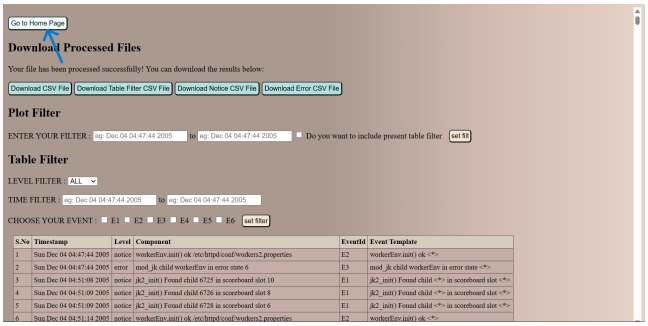

Figure 4: Home Page Button

3.2.2 Plot Filter

Enter the Time Filter in the respective boxes in the format shown on the place holder. (refer 5)

1) If you want to consider overall data to generate plots, then just click setfilt

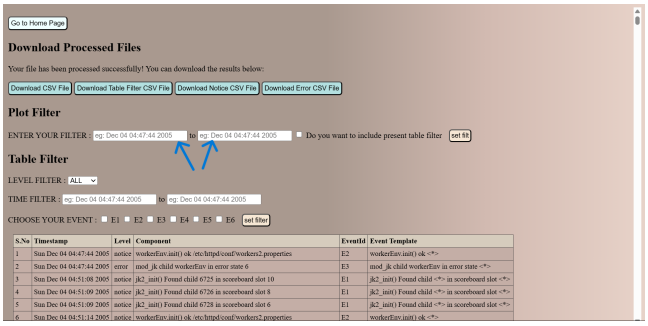

Figure 5: Time Filter

button leaving the boxes empty and not clicking the option of including present table filter.

2) If you want to consider from a date to end, then just enter the date on the left box and click setfilt.

3)If you want to procees the time filter from the present table filter, then click on the option just before setfilt button. (refer 6)

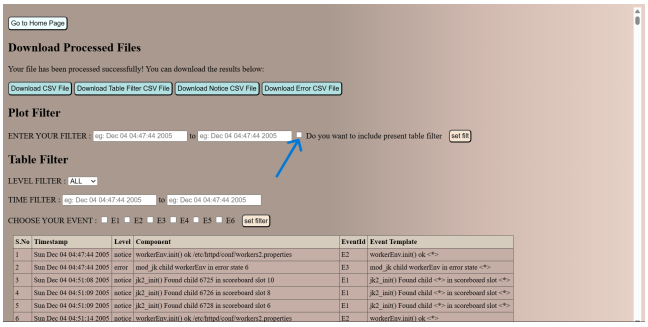

Figure 6: Including Table Filter

Any Time Format mismatch will show you the error message.

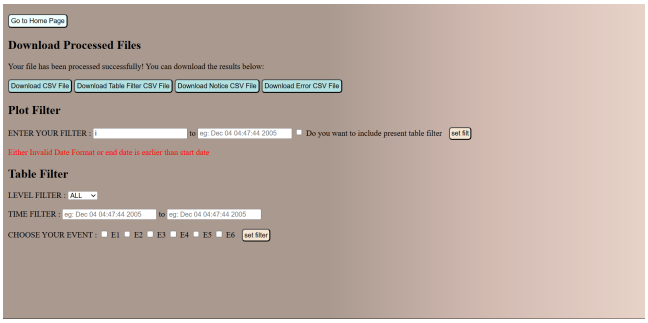

Figure 7: Error Message

After Clicking setfilt Option, if there is no error then you will be redirected to plot page , which we will discuss in next subsection.

3.2.3 Table Filter

Select the options you want to filter. If you did not select any option, it will be selecting all options of that particular filter by default.
The Time Filter is similar to that of in plot filter and shows error message if the time format is incorrect or the end time is earlier than start time.
After Clicking the set filter button, you will be displayed the table of current filter. (refer 8)

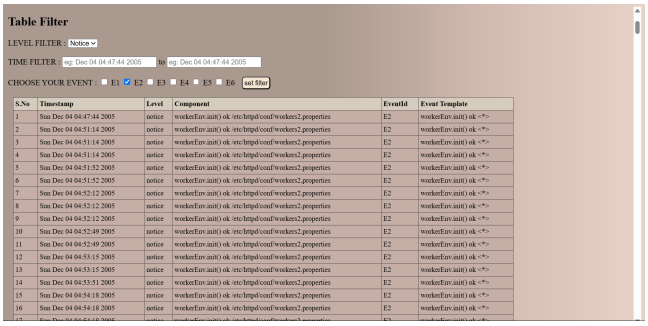

Figure 8: Table Filter

3.3 Plot Page

3.3.1 navigation bar

The navigation bar has two buttons.

1) Go To Home Page : To go back to Home Page
2) Go To Table Page : To go back to Table Page

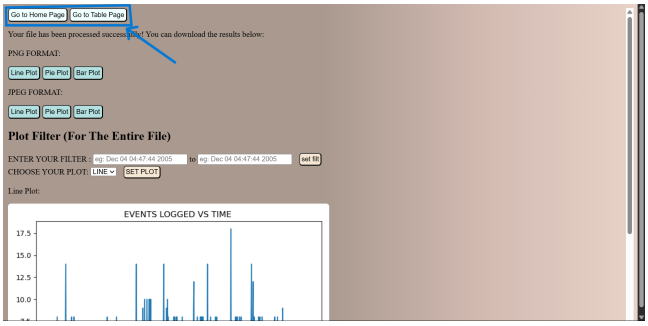

Figure 9: navigation Bar

3.3.2 Download Plots

You can download the png format of the plots in the png section (refer 10) and jpeg section (refer 11)

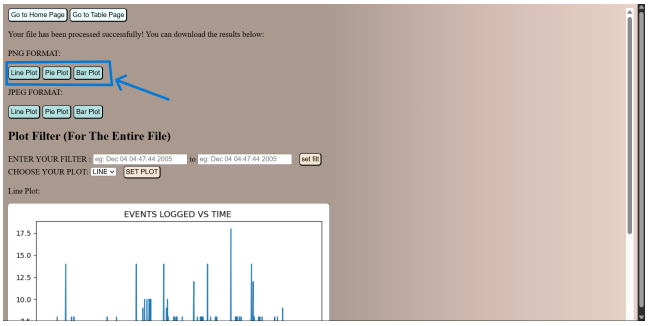

Figure 10: navigation Bar

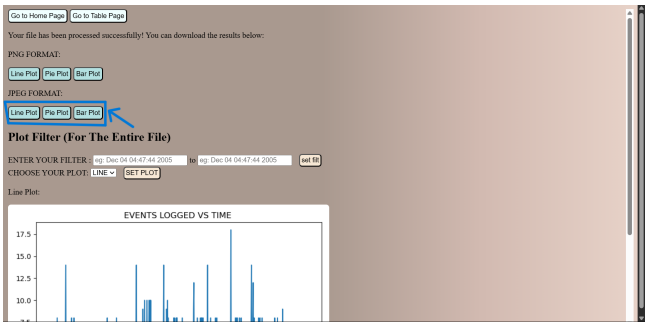

Figure 11: navigation Bar

3.3.3 Time Filter

It is similar to Time Filter in Download Page. Set time filter and click set filt button to set filter and generate the plots. (refer 12)

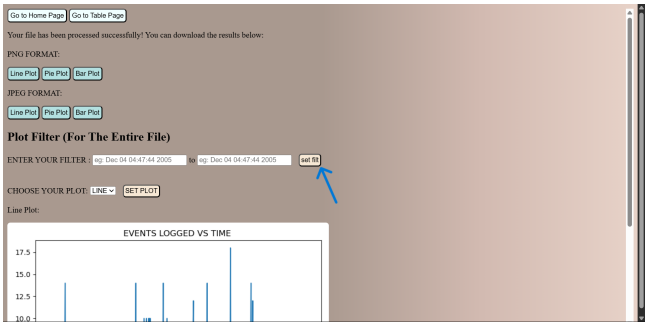

Figure 12: set filter

3.3.4 Plots

You can choose the plot you want to see and select it on the dropdown menu (refer 13) and click the SET PLOT button (refer 14)

Figure 13: choosing plot

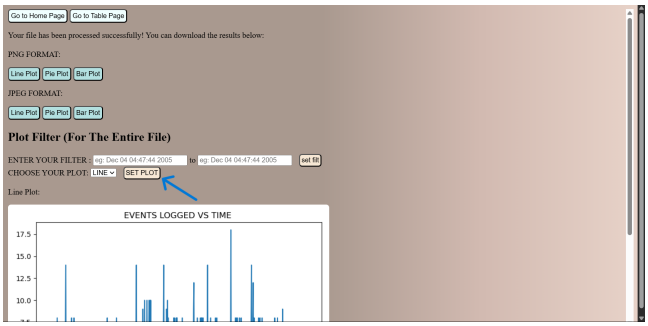

Figure 14: set plot

4 Modules

4.1 Flask.py

4.1.1 Flask

- **Flask**: Builds the web application, handles routing, file uploads, and form processing.
- **render_template**: Renders dynamic HTML pages with backend data.[4]
- **request**: Processes incoming form data and file uploads.
- **send_from_directory**: Serves files from server directories for download.[1]

- **redirect**: Redirect to a different route after actions like file upload.[5]
- **url_for**: Dynamically generates URLs for different routes and functions.[6]
- **subprocess**: Runs external Bash scripts and captures their output.[3]

4.2 plot.py

- **matplotlib**: To generate Plot. [2]
- **numpy**: To get X-axis data points for line plot

5 Directory Structure

5.1 HTML

- **Home.html**: It is the Home Page, which is for the File Upload.
- **download.html**: It is the download Page, which is for the CSV File Download, Table Filter and Plot Filter.
- **Plot.html**: It is the Plot Page, which is for the Plot Download, Plot Filter and Viewing the Plot.

5.2 static

- **style.css**: The css file used for decorating html files.
- **background1.png**: It is for background image of Home.html
- **background2.png**: It is for background image of download.html and Plot.html.
- **uploadbtn.png**: It is upload image for Home.html

5.3 UPLOADS_FOLDER

- **output.csv**: The CSV file with it's data from given logfile.
- **notice.csv**: The CSV file consisting only Notice lines.
- **error.csv**: The CSV file consisting only Error lines.
- **table_filter.csv**: The CSV file generated when you keep the table filter.
- **time_filter.csv**: It is when you select the time filter under table filter. it consists of time filter from output.csv file. It is further processed to meet other filter and generates table_filter which is final end product of table filter.

5.4 On Present Directory

- **csv.sh**: This bash script converts the given log file to csv file (output.csv) and also generates notice.csv and error.csv.
- **filter_time.sh**: For Time filter in table filter, it generates time_filter.csv which contains the timestamp between the ranges given. For Plot filter, it passes the lists (of lines which has it's data between the time range given) for the line,bar,pie plot to the plot.py and runs plot.py.
- **plot.py**: It takes the argument given by filter_time.sh and generate plots and save them in static folder.
- **main.py**: It is flask file which take care of all the process and links all together and maintain the webpage and all background process.

6 Basic And Advanced Features

6.1 Basic Features

6.1.1 Plot Filter

Implimented time filter for plot.

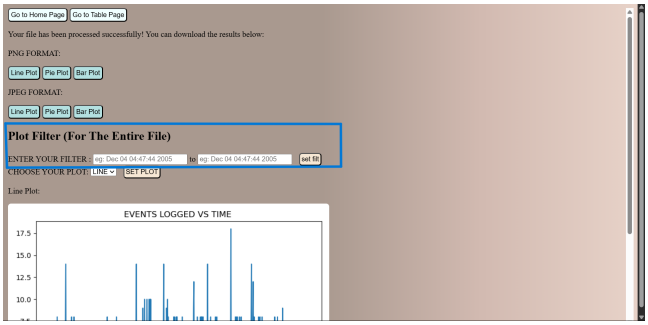

Figure 15: time filter

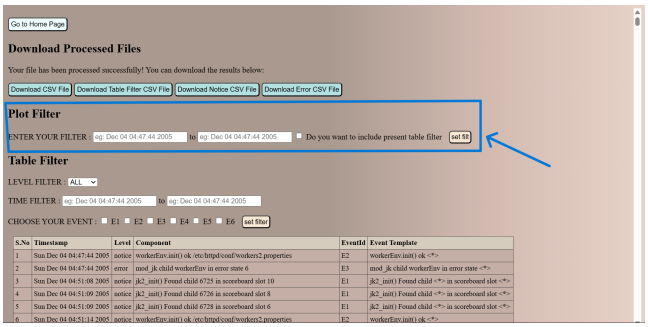

Figure 16: time filter

6.1.2 Download Option For Files

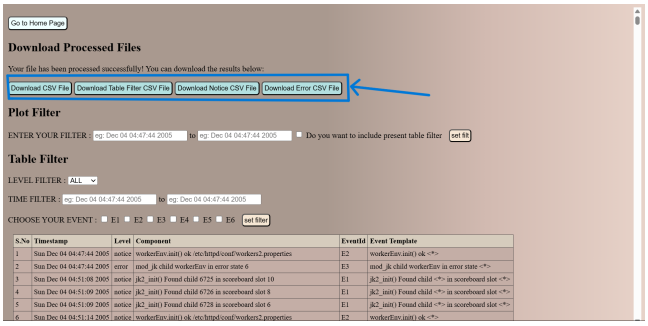

Figure 17: Download Option

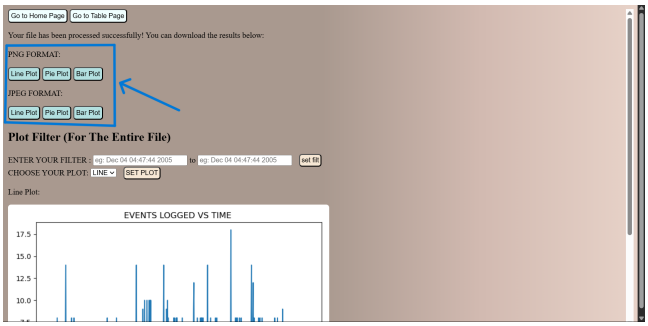

Figure 18: Download Option

6.1.3 viewing the plots

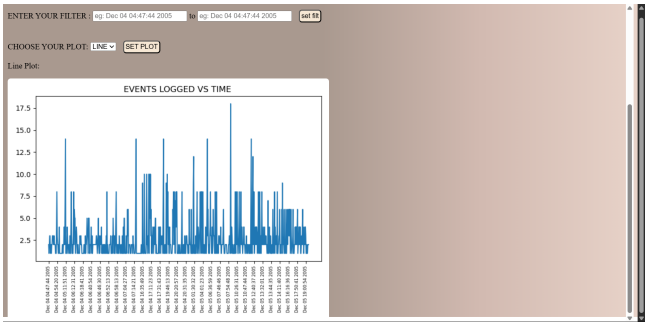

Figure 19: Plot viewing

6.1.4 Navigation Bar

Navigation bar for easy navigation.

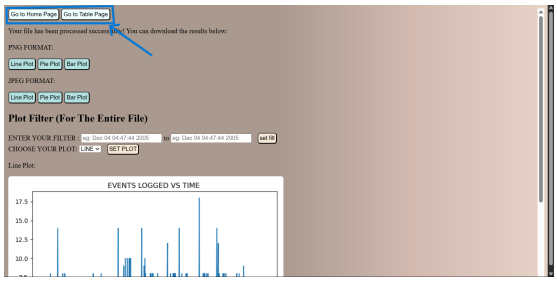

Figure 20: navigation Bar

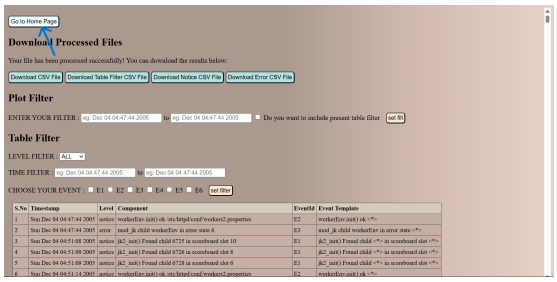

Figure 21: Home Page Button

6.2 Advanced Features

Table Filter

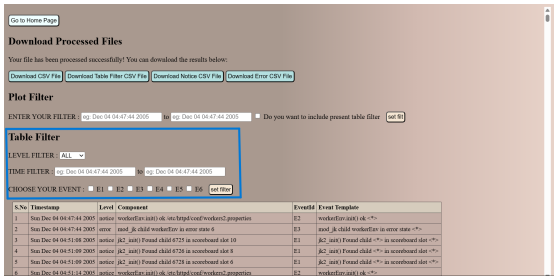

Figure 22: Table Filter

Downloading Table Filtered CSV File

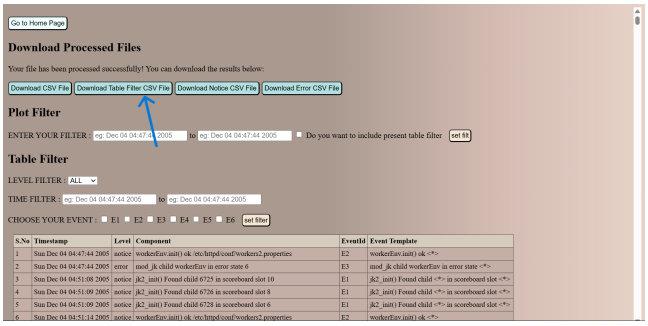

Figure 23: Table Filter

Including Table Filter for plot

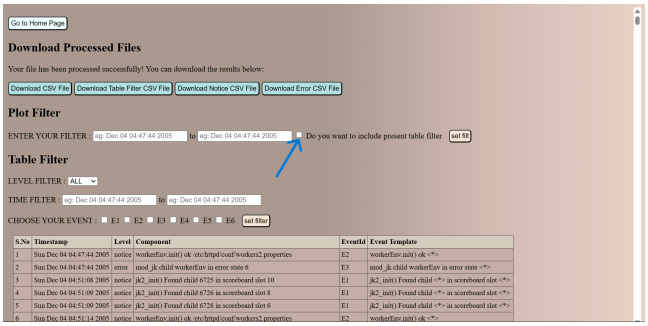

Figure 24: Including Table Filter

7 Project Journey

7.1 Learnings

1. Learned How to use Flask and an idea about backend.
2. Learned about unfamiliar modules like **subprocess**.
3. Learned how to pass a variable in flask to html.
4. Learned how URL's work.

7.2 Challenges Adressed

1. Starting again for **time filtering** once done in python and to redo it in bash.

2. Thinking in another way to address after noticing that if i passed around 8 arguments with 8th is very large giving some argument error..

3. Handling incorrect input entered input by the user.

4. Thinking how to print the table(in html page) with its data from another file.

References

[1] Flask Documentation. Flask. https://flask.palletsprojects.com/en/latest/api/#flask.send_from_directory.

[2] Matplotlib documentation. X-ticks. https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xticks.html.

[3] Python Documentation. Subprocess. https://docs.python.org/3/library/subprocess.html.

[4] Tech With TIm. Flask tutorial. https://www.youtube.com/watch?v=xIgPMguqyws.

[5] Tech With TIm. Flask tutorial. https://www.youtube.com/watch?v=mqhxxeeTbu0.

[6] Tech With TIm. Flask tutorial. https://www.youtube.com/watch?v=mqhxxeeTbu0.