Log File Analysis Project Report

Jaspreet Brar, 24B1082

April 29, 2025

1 Introduction

This project is a web-based log file analysis tool that allows users to upload logs and analyze them for better readability by conversion to CSV, visualize the data using various plots, and download filtered logs or visualizations. The theme of the project was inspired by the **Harry Potter** series, featuring a magical vibe.

2 Requirements

The following modules and software are needed to run the application:

- Python 3.x
- Gawk (GNU-Awk)
- GNU Sed
- Flask (Web framework)
- Numpy (For data processing)
- Matplotlib (For generating plots)
- HTML, CSS, JavaScript
- Apache, Syslog, or Android log files (for input data)

This application has been developed on a MacOS system, and hence is definitely best suited for a MacOS. Running it on other systems MAY cause errors.

Installation commands:

```
# Install necessary Python modules
pip install flask numpy matplotlib
```

3 Running Instructions

To run the application, follow the following steps:

1. Clone the repository from github:

   ```
   git clone ‘link’ path/to/destination
   ```

2. Run the Flask server in the directory:

```
python run.py
```

- MacOS may require activation of the venv in the directory via

```
source venv/bin/activate
```

before running the flask server

3. Open your browser and go to `http://127.0.0.1:5000`

```
If that doesn't work, try using incognito mode or
clearing your browser's cache.
Additionally, you can try changing port to
some other number (XXXX) in  run.py and go to
```

`http://127.0.0.1:XXXX`

4 Website Layout

The website is designed to provide an intuitive interface for users to interact with log data. The layout includes the following key pages:

- **Intro Page**: "The Daily Prophet" styled intro page (see if you can find clues!) having a portkey that takes you to the upload page.
- **Upload Page**: Upload logs and choose your Hogwarts House (or not?!).
- **Results Page**: Choose your next action : Viewing CSV, Downloading CSV or Viewing Plots.
- **CSV Display Page**: Displays a structured CSV for the entire log file. Also provides options to filter the CSV.
- **Plots Display Page**: Displays generated plots for the entire duration, also allowing user to select a time range and view plots for that duration only.

5 Modules

The following external libraries are used in the project:

- **Flask**: Used for creating the web application and handling HTTP requests.
- **Numpy**: Used for efficient data handling and manipulation of large log files.
- **Matplotlib**: Used for generating static, animated, and interactive visualizations.
- **Subprocess**: Used for running shell commands to process log files in different formats.
- **OS**: Making directories and output file destinations.

6 Directory Structure

The project directory structure is as follows:

```
project_directory/
 run.py                    # Main Flask application file
 app/
    __pycache__/
    plots.py         # contains plot generator function
    routes.py        # handles the entire application
 bash_scripts/
    android.sh             # parses android log files
    apache.awk             # event template parsing for apache logs
    apache.sh              # parses apache log files
    epoch.sh                # appends "seconds since epoch" column & creates times.csv
    filter_csv.sh           # filters logs by time
    syslog.sh               # filters syslogs
    validate.sh             # validates format
 static/
    cursors/
    fonts/
    media/
    plots/                 # stores generated plots
    style.css
 templates/
    display.html             # csv display page
    error.html               # incorrect log file error page
    index.html               # daily prophet style intro page
    plots.html               # Plot options page
    results.html             # view or download csv, view plots
    upload.html              # upload page for logs
 static/
    css/                   # Contains CSS files for styling
    js/                    # Contains JavaScript files
    images/                # Contains images used on the site
 tmp/
    logtype.txt            # Stores type of logfile
 uploads/                   # Folder containing uploaded log files
 visualizations/            # Folder where generated CSVs are saved
 venv/                      # Needed to run on MacOS
```

7 Basic & Advanced Features

7.1 Basic Features

- **Log File Upload**: Users can upload .log files. System automatically parses, validates and creates a structured CSV using bash.

- **Log File Validation**: If the format of the log file is incorrect, the website shows an error, otherwise it continues with filtering. item
- **Log Display**: The converted log file can be viewed in a tabular format and also the entire log (in csv format) can be downloaded
- **Visualization Generation**: The system generates visualizations based on log data, including line plots, bar charts, and pie charts.
- **Download Processed Data**: Users can download filtered plots as PNG or JPEG files. Filtered CSV can also be downloaded.
- **Customized Interface**: Users can select their Hogwarts house, the theme gets changed accordingly. There are a few references to *Harry Potter* that users may find amusing.

7.2 Advanced Features

- **Support for Multiple log formats**: Apache Logs, android logs and syslogs (Linux) are supported.
- **Extended Visualization**: Visualizations for all the mentioned type of log files are available.
- **Drag-and-Drop Interface**: Files can be uploaded via the glowing button on the upload page or by dragging and dropping.
- **Log Filtering**: Users can filter logs by date range to narrow down their analysis. Further, users can select a specific column (such as "Level") and further filter it's elements. Users can also download filtered CSVs.
- **Customized CSS**: The entire website is harry potter themed, allowing colors to be chosen according to user's preference.

8 Website Snapshots

8.1 Intro Page

I have recreated The Daily prophet, which is the newspaper in the Harry Potter series, containing CS related news in the wizarding world. The portkey at the bottom (upon holding for 3 seconds), takes you to the upload page.

Figure 1: Idle Portkey

Figure 2: Portkey upon holding

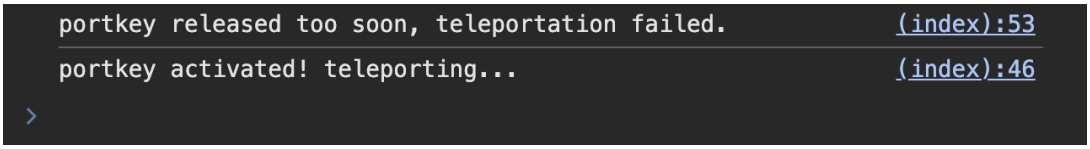

Figure 3: Console output

8.2 Upload Page

This page is title Weasley's Wizarding Logs. The user can upload only .log files via the glowing button or by dragging-and-dropping their file onto the browser window. The cursor has also been replaced by a wand. Additionally, the user selects their Hogwarts House. This decides the theme of the remaining pages.

```
Wand (cursor) is not visible in the snapshots.
```

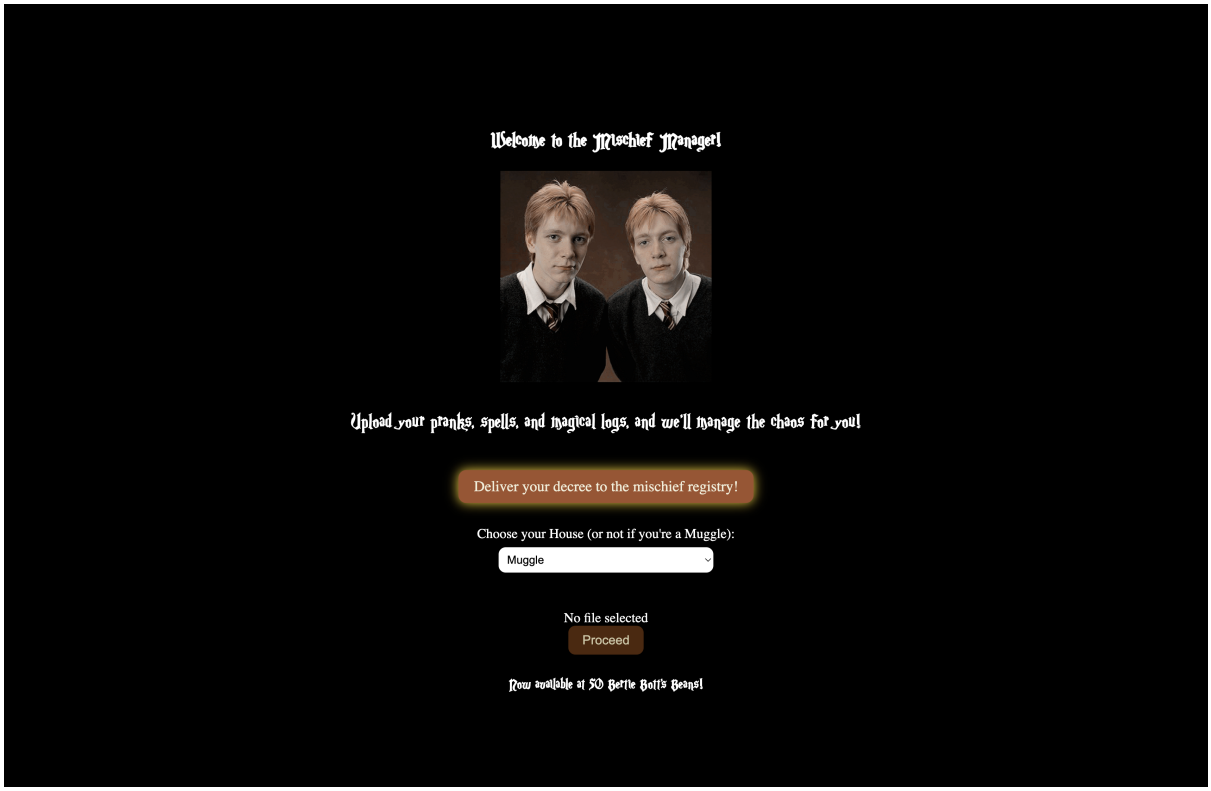

Figure 4: Uploading interface

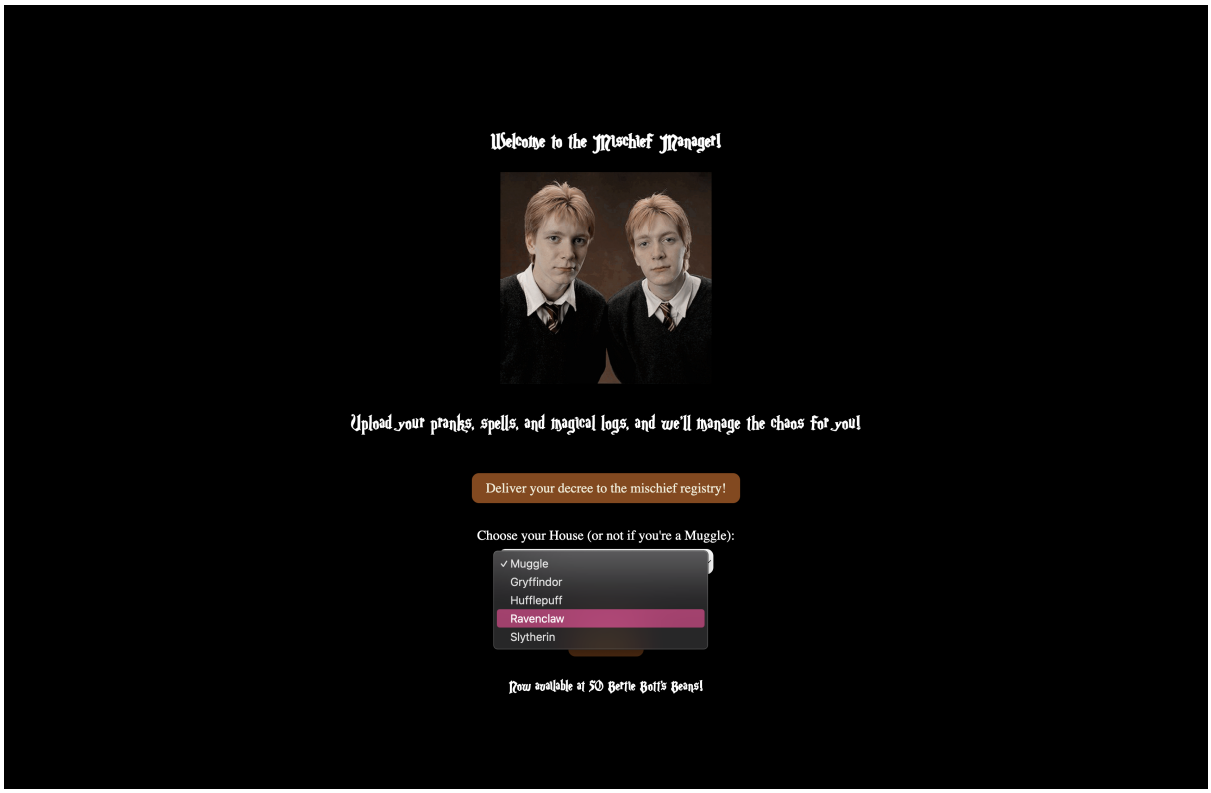

Figure 5: House selector (Muggle by default)

8.3 Results Page

This is the intermediate page that gives the users options about what they can proceed with. It is user's house themed and the buttons glow when hovered upon. It also mentions the name and the type of the uploaded log file.

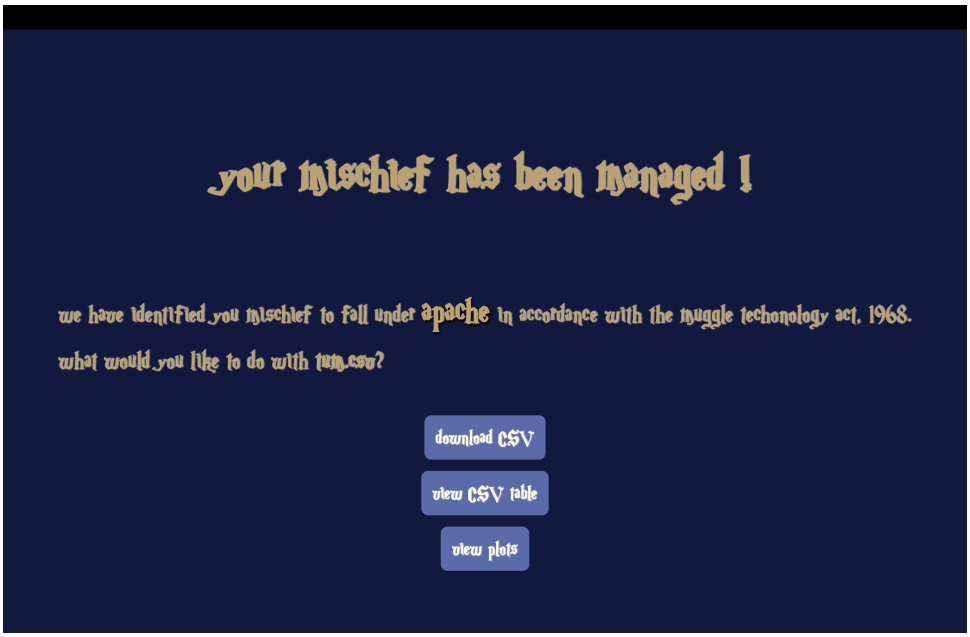

Figure 6: Ravenclaw Theme

8.4 CSV Display Page

Displays entire log file by default. Can be filtered by a certain time period or column wise. Each filtered CSV can be downloaded by the given button. Can go back to entire log using "back to original button".

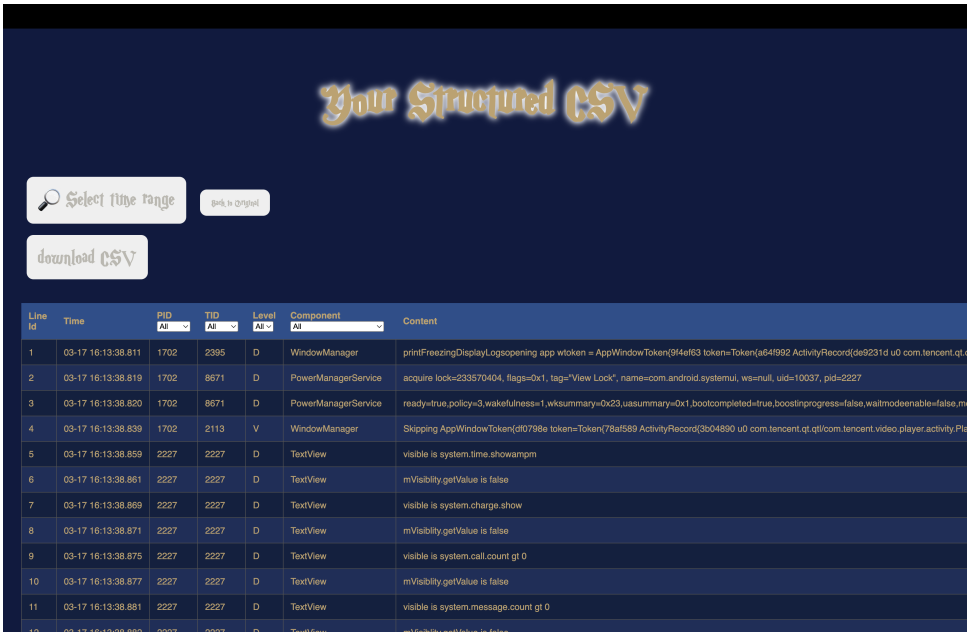

Figure 7: Entire Log File as CSV

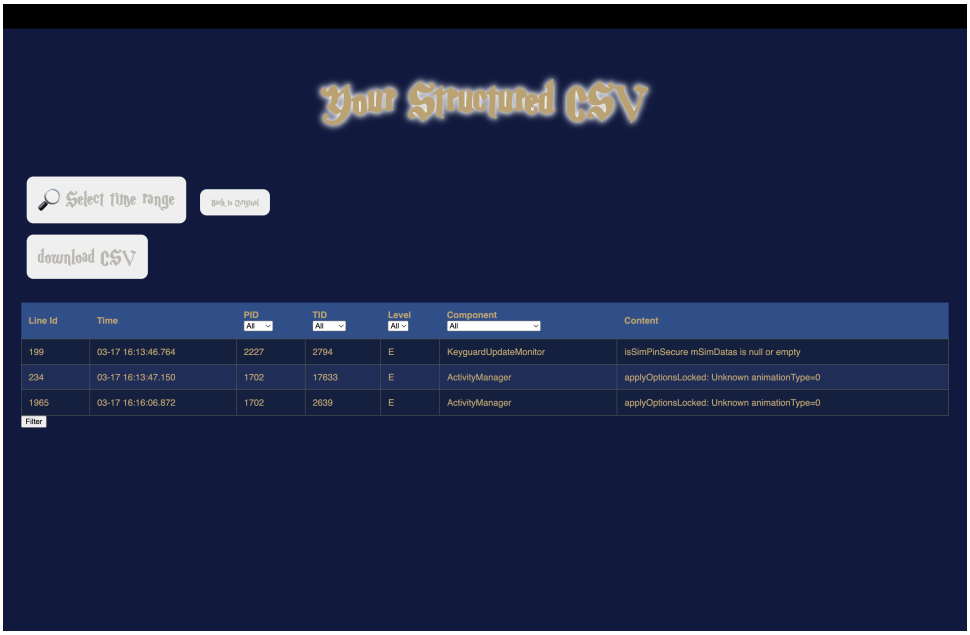

Figure 8: CSV Filtered by column

8.5 Plots Page

Shows entire duaration's plots by default. Can be adjusted to show for a specific time duration only. Different kinds of plots for different log file formats. Plots can be downloaded in PNG or JPEG format.

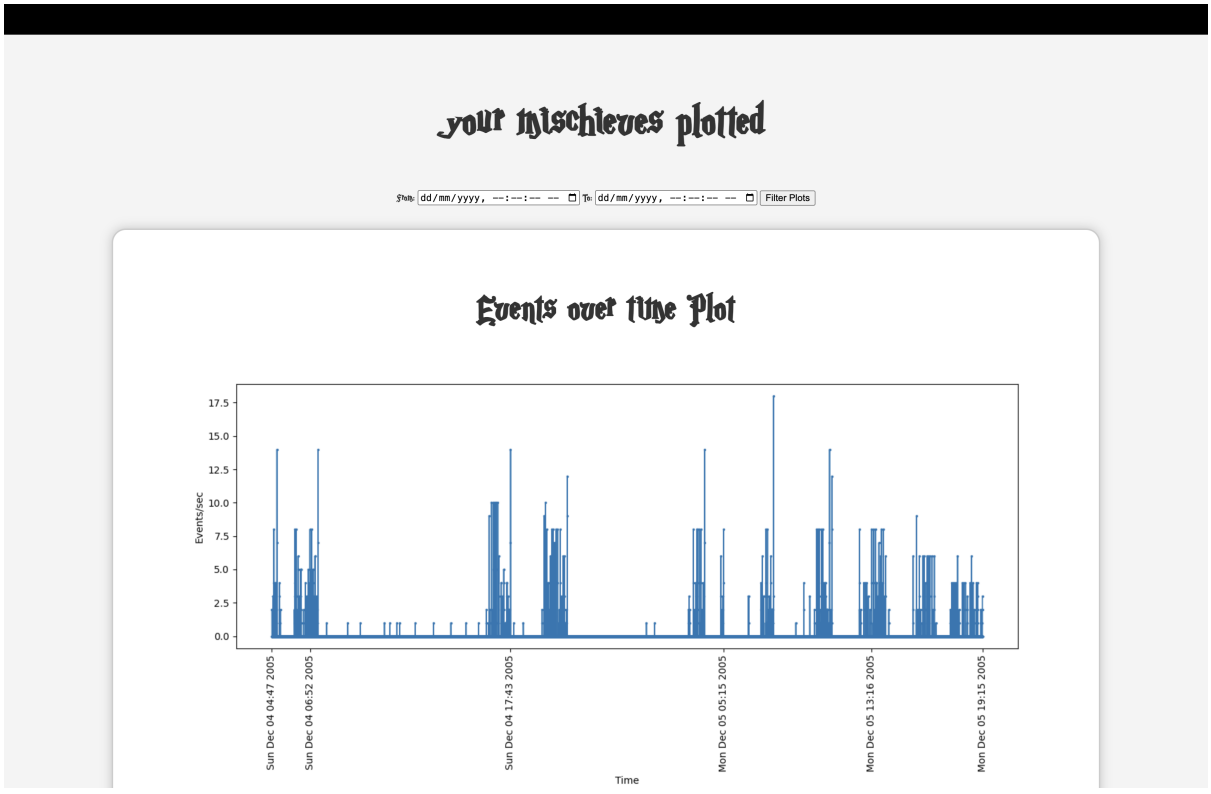

Figure 9: Plot for entire log file

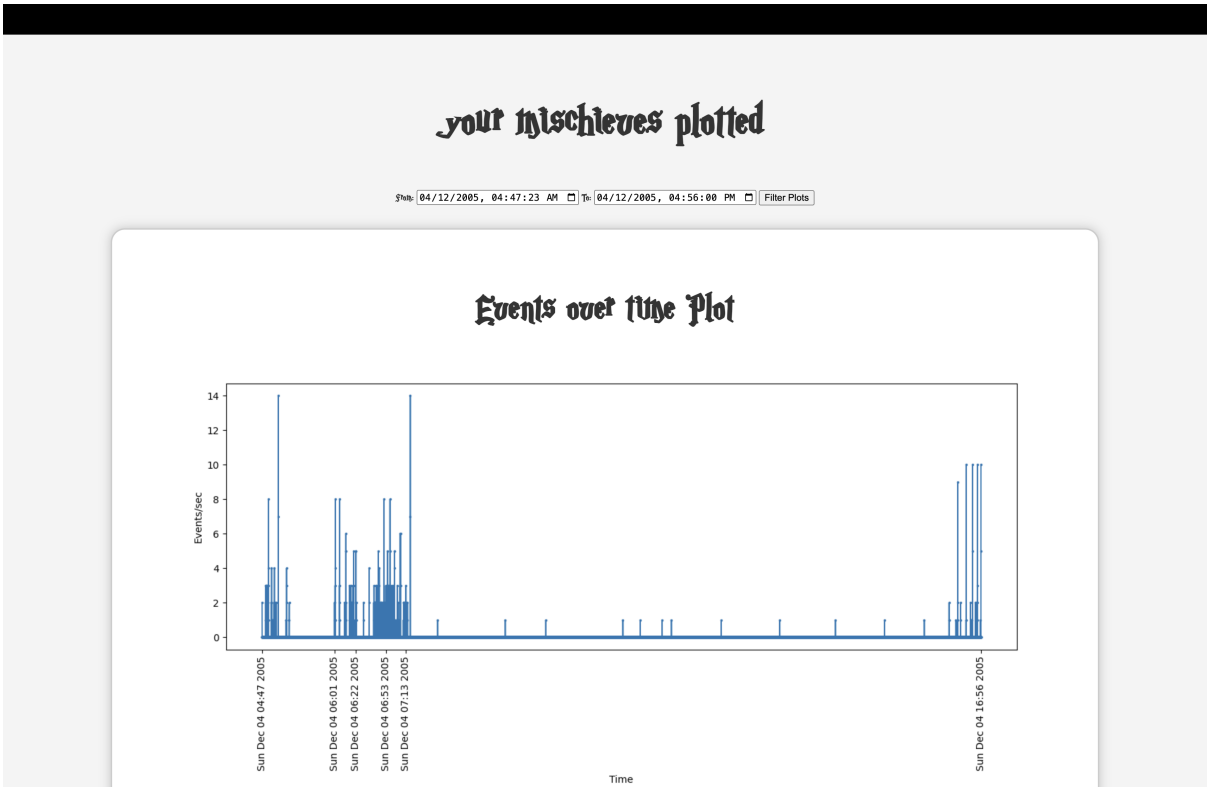

Figure 10: Plot Filtered by time period

9 Project Journey

Throughout the development of this project, several key challenges were encountered and addressed:

- **Multi-Format Log Parsing**: Supporting Android, Syslog, and Apache log formats required writing distinct parsing and filtering scripts tailored to each format. Each log type had unique delimiters, timestamp styles, and field layouts, demanding dynamic field separator handling and conditional logic based on log type.
- **Timestamp Extraction & Comparison**: One of the major challenges was normalizing inconsistent date-time formats across logs. Apache and Syslog entries lacked full timestamps (e.g., missing the year), while Android logs used millisecond precision with `"MM-DD HH:MM:SS.sss"` format. I used a combination of shell functions and `mktime()` in `gawk` to convert these into comparable epoch values for accurate time-based filtering.
- **Delimiters and Field Handling**: Default field separators often conflicted with the actual log format (e.g., commas vs. exclamation marks), leading to initially empty or misparsed files. This was resolved by dynamically setting `FS` and `OFS` inside `awk` based on the log type, instead of relying on command-line flags which override in-script definitions.
- **Filtering Logic and Edge Cases**: Implementing accurate time-range filtering required safeguarding against partial timestamps, missing values, and malformed lines. Logs outside the specified range had to be excluded without breaking the

CSV structure or plot expectations. I had to write multiple sed-awk expressions to fully bring everything together such as Event Templates and Syslog non-PID lines.

- **Visualization Customization**: To support consistent analysis across log types, I designed three core visualizations (event frequency over time, level distribution, event code distribution). Each had to adapt to filtered data ranges while maintaining legibility and thematic consistency across user-selected house color schemes.

10 Bibliography

- **Flask Documentation**: `https://flask.palletsprojects.com/`
- **Matplotlib Documentation**: `https://matplotlib.org/`
- **Numpy Documentation**: `https://numpy.org/`
- **Python Official Documentation**: `https://docs.python.org/3/`
- **GAWK Effective Programming**:`https://www.gnu.org/software/gawk/manual/`
- *CS104 Course Material - Lecture and Tutorial slides*