Telecom Call Records Analysis Using GroupBy and Aggregation

NAME: TRUPTHI HP

EMAIL ID: hptrupthi@gmail.com

DATE: 01/09/2025

Problem statement - Telecom companies generate huge volumes of call records daily, but raw data alone lacks insights. To improve services and understand user behavior, it is essential to analyze call durations, frequent callers, and network usage patterns. This project focuses on processing and analyzing telecom call records to extract meaningful insights that can support better decision-making.

Index

INTRODUCTION

The Telecom Call Records Analysis Project is designed to analyse call log data and extract meaningful insights about communication patterns. It uses **Python (Pandas)** for data analysis and **Flask** as the backend framework to handle file uploads and processing. Users can upload call records in CSV format containing details such as caller, receiver, duration, call type, and date. The system then generates **summary reports, detailed tables, and trend visualizations**, making the data easier to understand. Additionally, reports can be exported in **CSV, Excel, and PDF formats** for further use. This project provides an efficient way for telecom operators, researchers, and learners to study **call usage behaviour and patterns** effectively.

Dataset Information

- **Call ID** – Unique identifier for each call
- **Customer ID** – ID of the customer making the call
- **Call Type** – Type of call (Local, STD, ISD)
- **Call Start time** – Start timestamp of the call
- **Call end time** – End timestamp of the call
- **Call Duration** – Duration of the call in minutes

2. **Description (Detailed)**

Project Description

This project, “Telecom Call Records Analysis Using Group By and Aggregation”, focuses on analysing telecom call data using Python’s Pandas library. The dataset consists of customer call records, including call type, duration, start and end times. The project involves step-by-step data processing such as cleaning, creating derived columns, filtering records, grouping, and generating summaries.

Purpose

The main purpose of this project is to practice data wrangling and aggregation techniques in Pandas while extracting meaningful insights from telecom data. It aims to identify call duration trends, frequent callers, and usage patterns without using any visualizations.

Outcome

□ Successfully analysed telecom call records using **Python (Pandas)** and **Flask**.

□ Generated **summary reports** showing total calls, total duration, and average call duration per caller.

□ Created **detailed transaction tables** with sorting, searching, and filtering options for better record management.

□ Implemented **visualizations** (daily and monthly trends) to understand call patterns over time.

□ Identified **top callers** based on frequency and duration of calls.

□ Enabled users to **export reports** in CSV, Excel, and PDF formats for offline use.

□ Automated the analysis process, reducing the need for manual calculations.

□ Provided a **user-friendly interface** for uploading, analyzing, and viewing call records.

□ Demonstrated how telecom data can be transformed into **actionable insights** for decision-making and research.

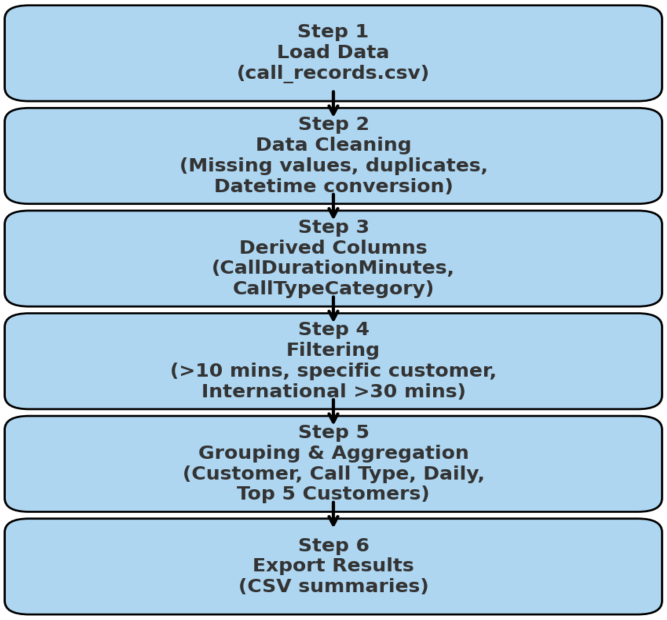
3. Plan

Step	Task	Explanation	Expected Outcome

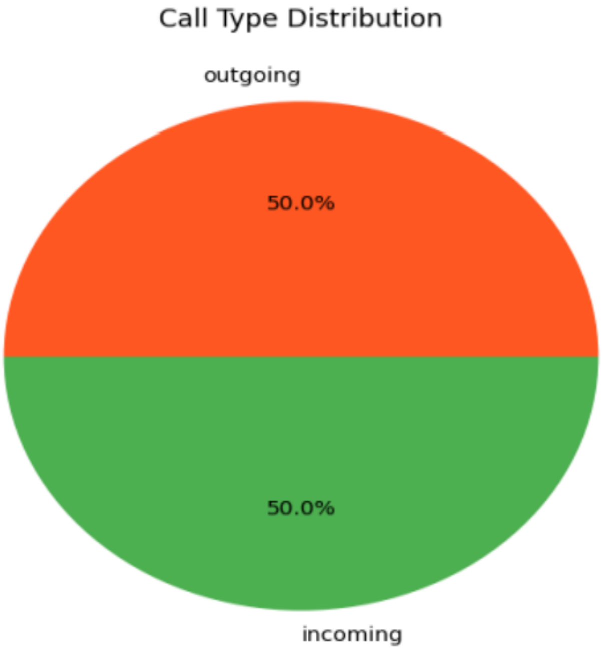
Step 1	Load Data	Load call_records.csv into a Pandas Data Frame. Inspect rows, columns, and data types.	Dataset is ready for processing with clear structure.
Step 2	Data Cleaning	Handle missing values (fill Call Duration with 0). Drop duplicate Call ID entries. Convert Call Start Time/Call End Time to date time.	Clean dataset with no duplicates and correct data types.
Step 3	Derived Columns	Create Call Duration Minutes (End Time – Start Time). Create Call Type Category (Domestic/International).	Enriched dataset with new fields for deeper analysis.
Step 4	Filtering	Filter calls > 10 minutes. Filter calls by specific Customer ID (e.g., C001). Find International calls > 30min	Subsets of data for analysis.
Step 5	Grouping & Aggregation	Group by Customer ID → total , average, number of calls. Group by → total & average duration. Group by Call Date → daily total duration. Identify top 5 customers by total duration.	Summarized insights: per customer, call type, and date.
Step 6	Export Results	Export customer summary → customer_call_summary.csv. Export call type summary → calltype_summary.csv. Export daily summary → daily_call_summary.csv.	Final reports for usage and sharing in CSV format.

4. Design (Diagrams)

Architecture Flow:

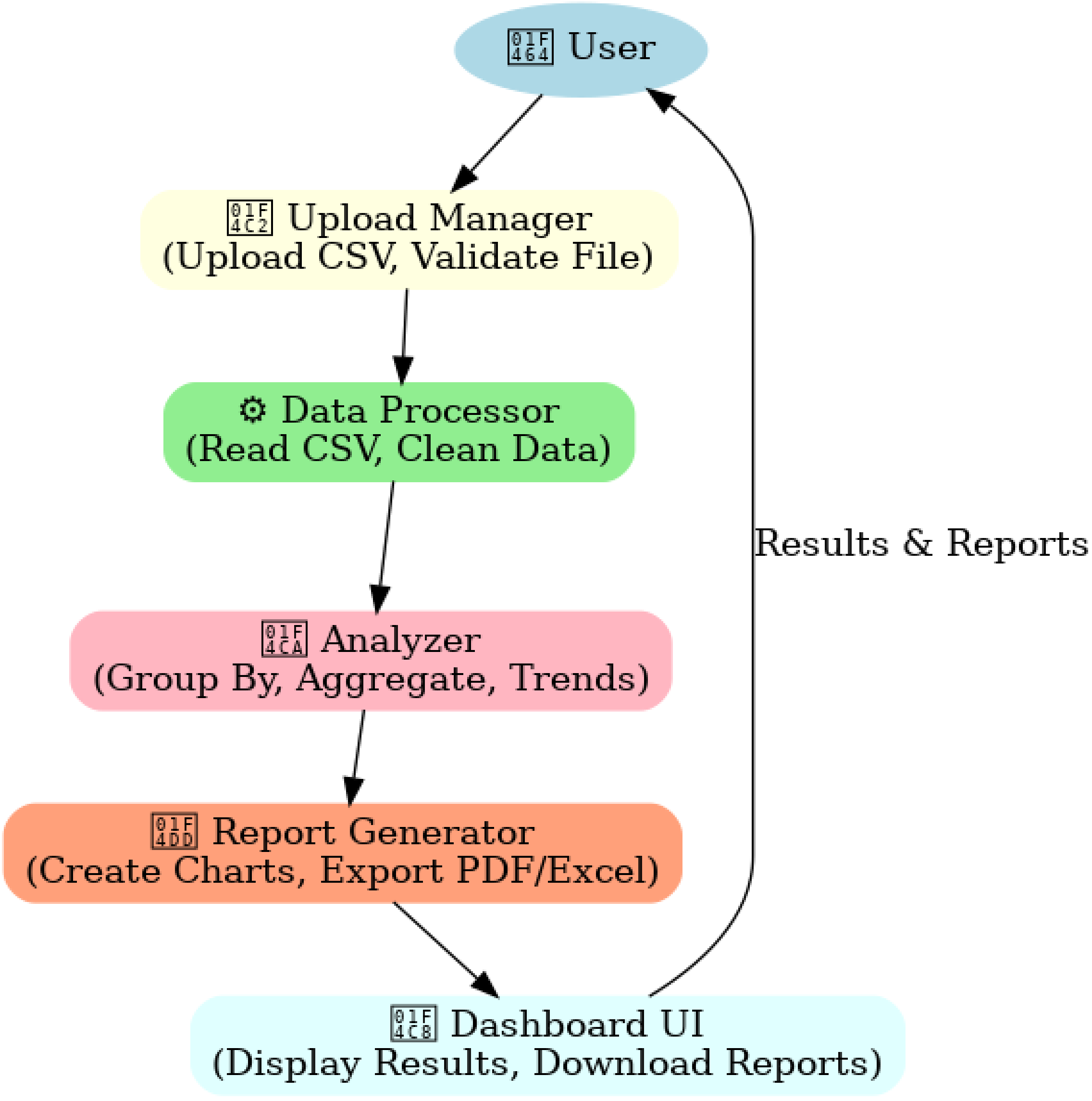


Pie Graph



UML DIAGRAM

The UML diagram represents the overall workflow of the Telecom Call Records Analysis project. It begins with the **user uploading a CSV file** containing call records, which is then processed by the **Flask backend** using **Pandas** for data analysis. The system generates **summary reports, detailed call tables, and visualizations** that are displayed on the **frontend interface**. Additionally, users can **export reports** in multiple formats (CSV, Excel, PDF).


User
Upload Manager
(Upload CSV, Validate File)
⚙ Data Processor
(Read CSV, Clean Data)
Analyzer
(Group By, Aggregate, Trends)
Report Generator
(Create Charts, Export PDF/Excel)
Dashboard UI
(Display Results, Download Reports)
Results & Reports

5. Implementation

The dataset was loaded into Pandas, cleaned (missing values, duplicates, date time conversion), and enriched with new columns. Filtering and grouping operations were applied to analyze call durations, frequent callers, and usage patterns. Finally, the results were exported into CSV summaries.

Analyses Conducted

- The dataset was cleaned and transformed to remove duplicates, handle missing values, and standardize date formats.

- New fields such as **Call Duration Minutes** and **Call Type Category** were derived to add more meaning to the data.

- **Customer-wise analysis** revealed total, average, and number of calls per customer, identifying the top 5 customers with the highest call durations.

- **Call type analysis** compared Domestic (Local) vs. International (STD/ISD) usage patterns, showing differences in frequency and call length.

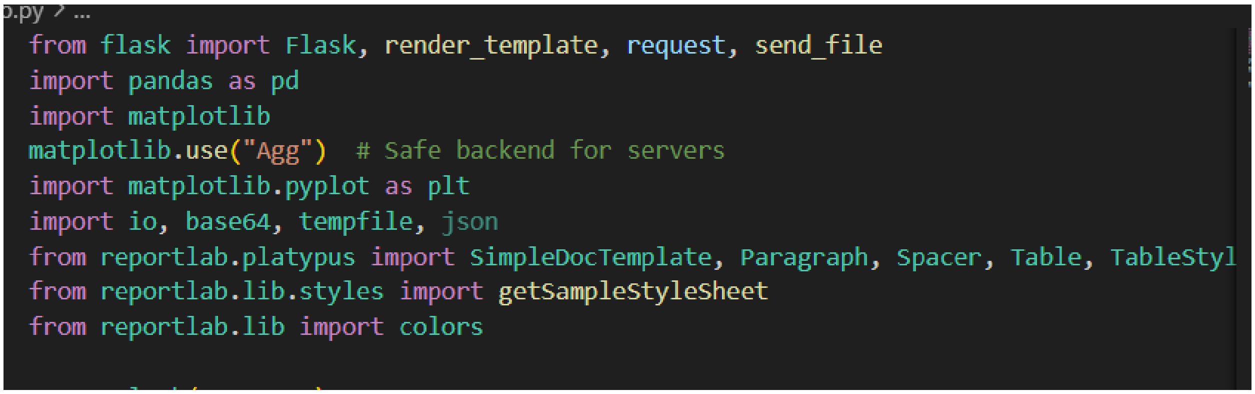
- **Daily analysis** highlighted total call durations per day, reflecting variations in network usage over time.

- **Filtering operations** helped in extracting specific insights such as calls longer than 10 minutes, customer-specific calls, and long-duration international calls.

- Overall, the analysis proved that raw telecom call records can be systematically processed into **actionable insights** about user behaviour and network activity.

Code

APP.PY

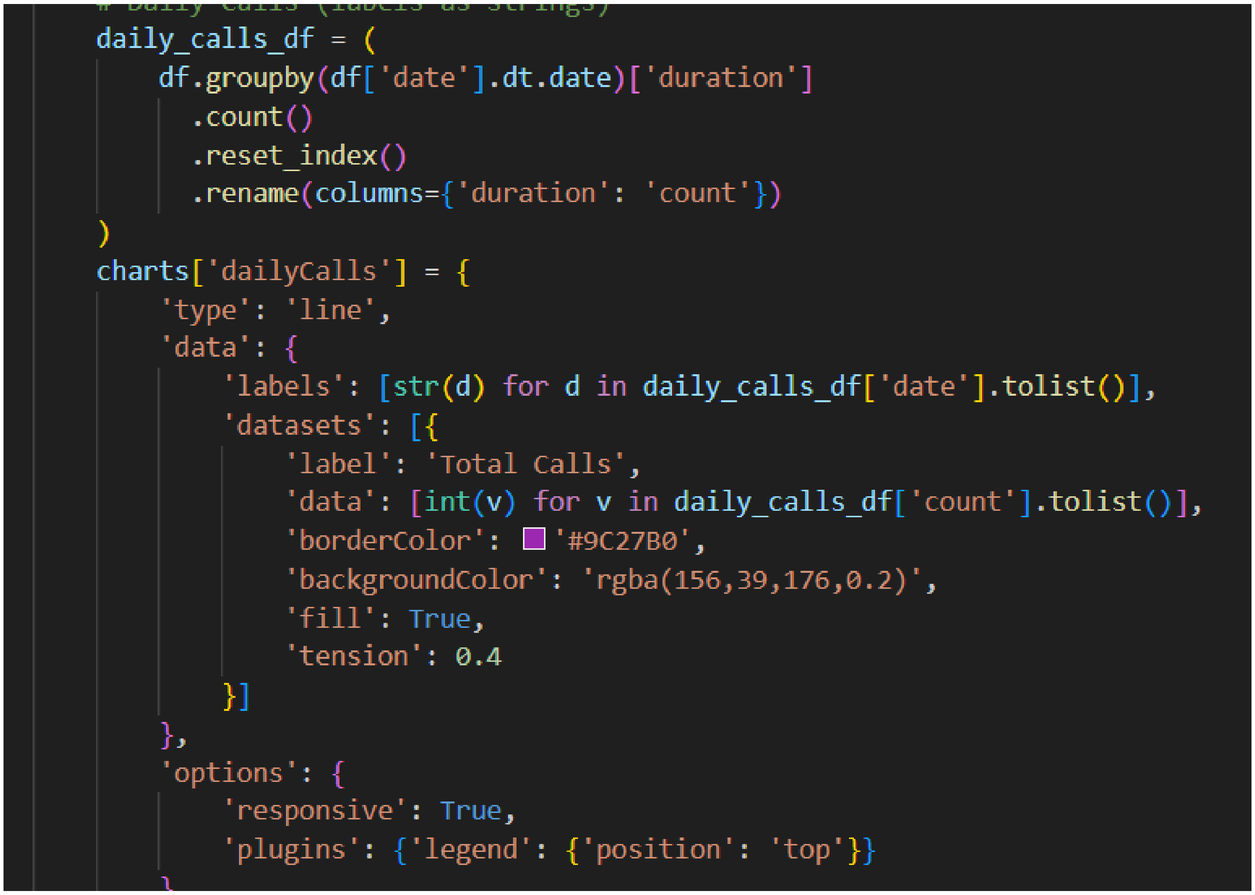


```
from flask import Flask, render_template, request, send_file
import pandas as pd
import matplotlib
matplotlib.use("Agg")  # Safe backend for servers
import matplotlib.pyplot as plt
import io, base64, tempfile, json
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer, Table, TableStyl
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib import colors
```

- **Flask** → Provides the web framework.
  - `render_template`: Displays HTML pages.
  - `request`: Handles CSV file uploads.
  - `send_file`: Allows report downloads.
- **Pandas** → Reads, cleans, and analyzes telecom call records from CSV files.
- **Matplotlib** → Generates charts and visualizations.
  - `matplotlib.use("Agg")`: Ensures safe plotting on servers without GUI.
  - `pyplot (plt)`: Simplifies chart creation.
- **Utility Modules** → Support file and data handling.
  - `io`: In-memory file handling.
  - `base64`: Encodes images (charts) for HTML.
  - `tempfile`: Creates temporary files.
  - `json`: Manages structured data.

•**Report Lab** → Used for PDF report generation.

- `SimpleDocTemplate`: Defines the structure of the PDF.
- `Paragraph`, `Spacer`: Add text and spacing.
- `Table`, `TableStyle`: Create formatted tables.
- `getSampleStyleSheet`: Provides default text styles.
- `colors`: Adds styling and highlights.



```
# Daily calls (labels as strings)
daily_calls_df = (
    df.groupby(df['date'].dt.date)['duration']
      .count()
      .reset_index()
      .rename(columns={'duration': 'count'})
)
charts['dailyCalls'] = {
    'type': 'line',
    'data': {
        'labels': [str(d) for d in daily_calls_df['date'].tolist()],
        'datasets': [{
            'label': 'Total Calls',
            'data': [int(v) for v in daily_calls_df['count'].tolist()],
            'borderColor': ■ '#9C27B0',
            'backgroundColor': 'rgba(156,39,176,0.2)',
            'fill': True,
            'tension': 0.4
        }]
    },
    'options': {
        'responsive': True,
        'plugins': {'legend': {'position': 'top'}}
    }
```

- The code groups call records by **date** and counts how many calls happened each day.

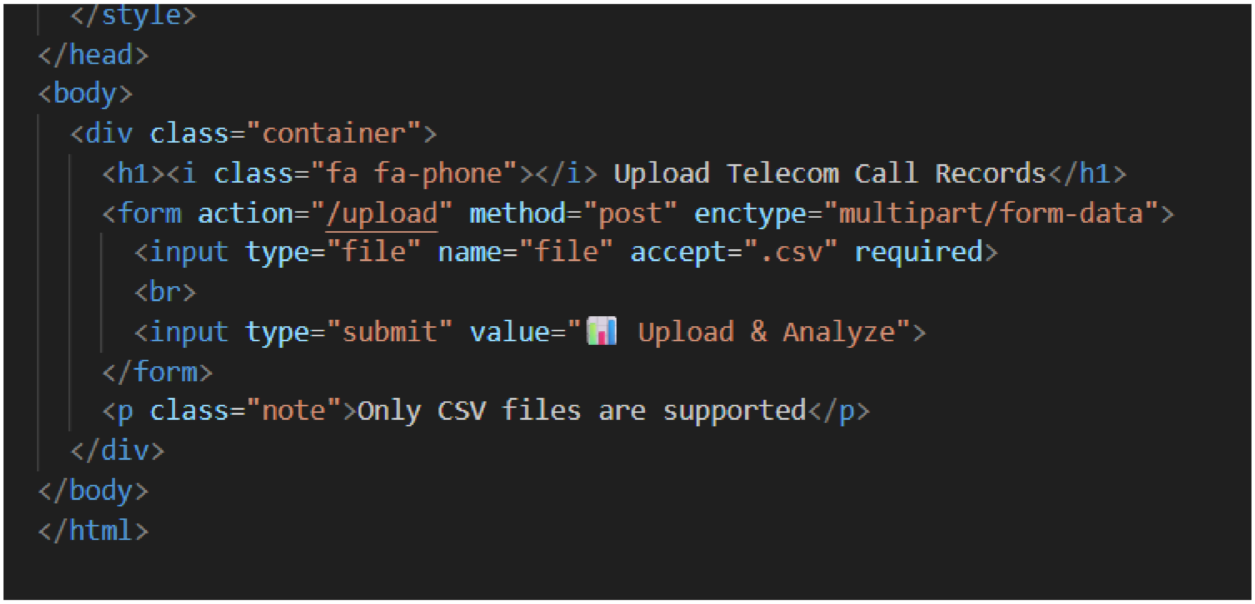
- It creates a new DataFrame (`daily_calls_df`) with two columns: `date` and `count`.

- It prepares chart data for **Chart.js** in the form of a **line chart**.

- The X-axis shows the dates, and the Y-axis shows the **total calls per day**.

- The chart is styled with a **purple line**, smooth curves (`tension=0.4`), filled background, and a legend at the top.

INDEX.HTML



```html
  </style>
</head>
<body>
  <div class="container">
    <h1><i class="fa fa-phone"></i> Upload Telecom Call Records</h1>
    <form action="/upload" method="post" enctype="multipart/form-data">
      <input type="file" name="file" accept=".csv" required>
      <br>
      <input type="submit" value="📊 Upload & Analyze">
    </form>
    <p class="note">Only CSV files are supported</p>
  </div>
</body>
</html>
```

- This code creates a simple **HTML form** for uploading telecom call record files.

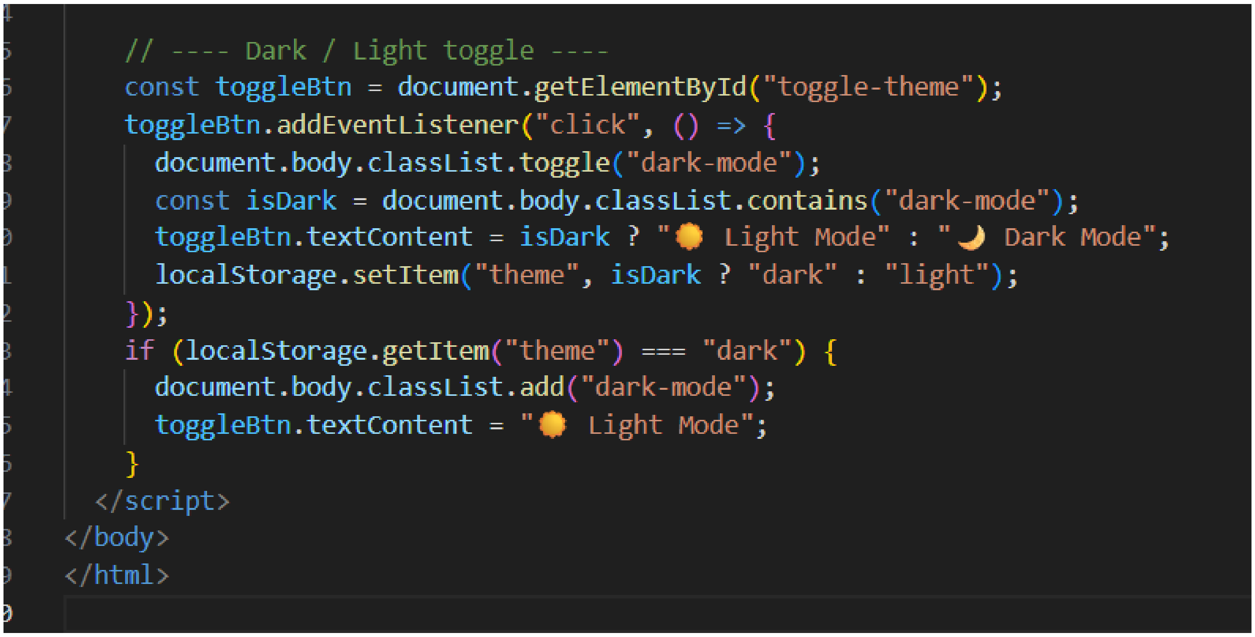
- The form uses the **POST method** and sends files to the `/upload` route in the backend.

- It allows only **CSV files** (`accept=".csv"`) and makes file upload **mandatory** (`required`).

- A **submit button** is provided with the label “📊 Upload & Analyze”.

- A note is displayed below, informing users that **only CSV files are supported**.

RESULT.HTML



```
    // ---- Dark / Light toggle ----
    const toggleBtn = document.getElementById("toggle-theme");
    toggleBtn.addEventListener("click", () => {
      document.body.classList.toggle("dark-mode");
      const isDark = document.body.classList.contains("dark-mode");
      toggleBtn.textContent = isDark ? "🌞 Light Mode" : "🌙 Dark Mode";
      localStorage.setItem("theme", isDark ? "dark" : "light");
    });
    if (localStorage.getItem("theme") === "dark") {
      document.body.classList.add("dark-mode");
      toggleBtn.textContent = "🌞 Light Mode";
    }
  </script>
</body>
</html>
```

• This JavaScript code adds a **Dark/Light mode toggle** feature.

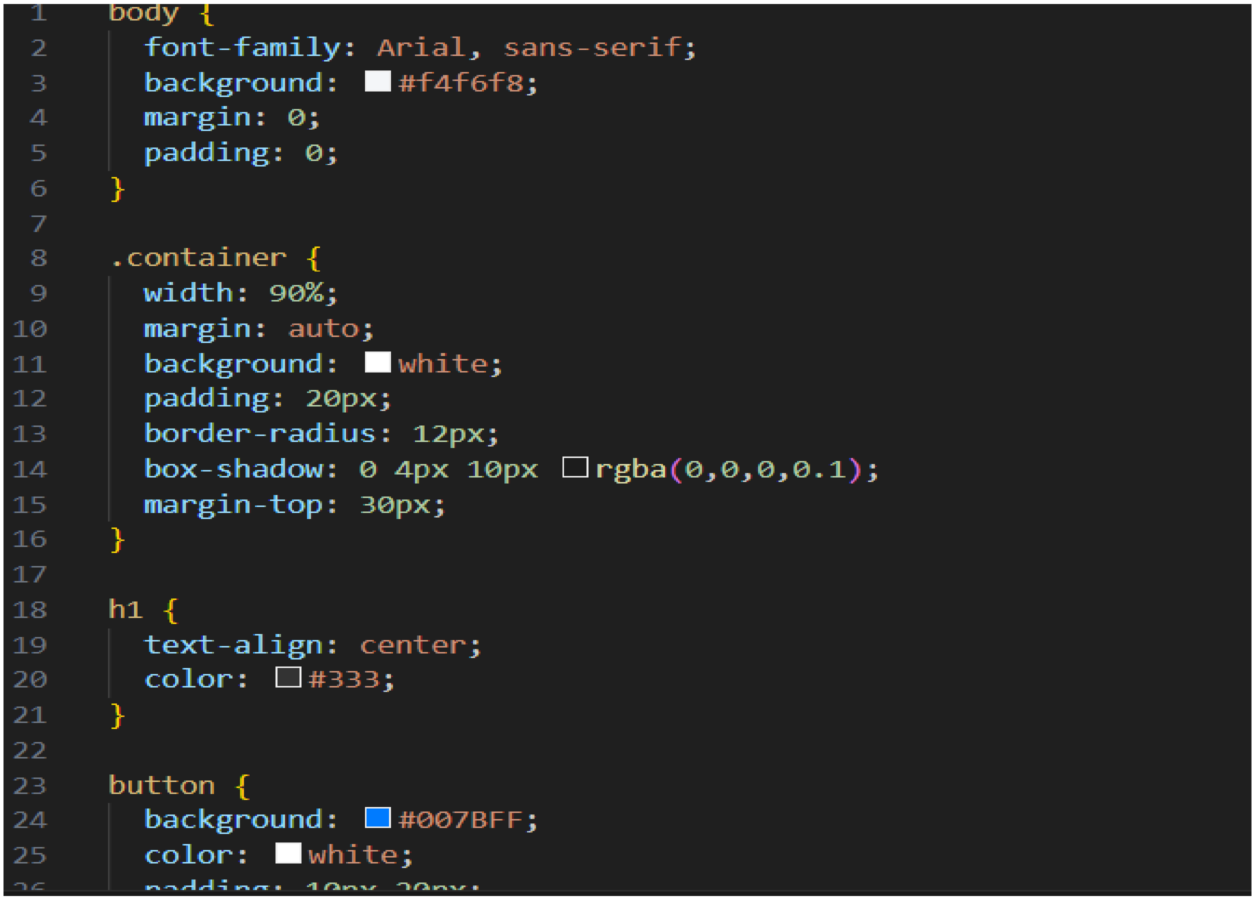
•It listens for a **button click** (`toggle-theme`) and switches the page style by adding/removing a `dark-mode` class.

•The button text changes dynamically between 🌞 Light Mode and 🌙 Dark Mode.

•The chosen mode is **saved in localStorage**, so the preference remains after page reload.

•On page load, if the saved theme is dark, it automatically applies **dark mode**.

STYLE.CSS



```css
body {
  font-family: Arial, sans-serif;
  background: #f4f6f8;
  margin: 0;
  padding: 0;
}

.container {
  width: 90%;
  margin: auto;
  background: white;
  padding: 20px;
  border-radius: 12px;
  box-shadow: 0 4px 10px rgba(0,0,0,0.1);
  margin-top: 30px;
}

h1 {
  text-align: center;
  color: #333;
}

button {
  background: #007BFF;
  color: white;
  padding: 10px 20px;
```

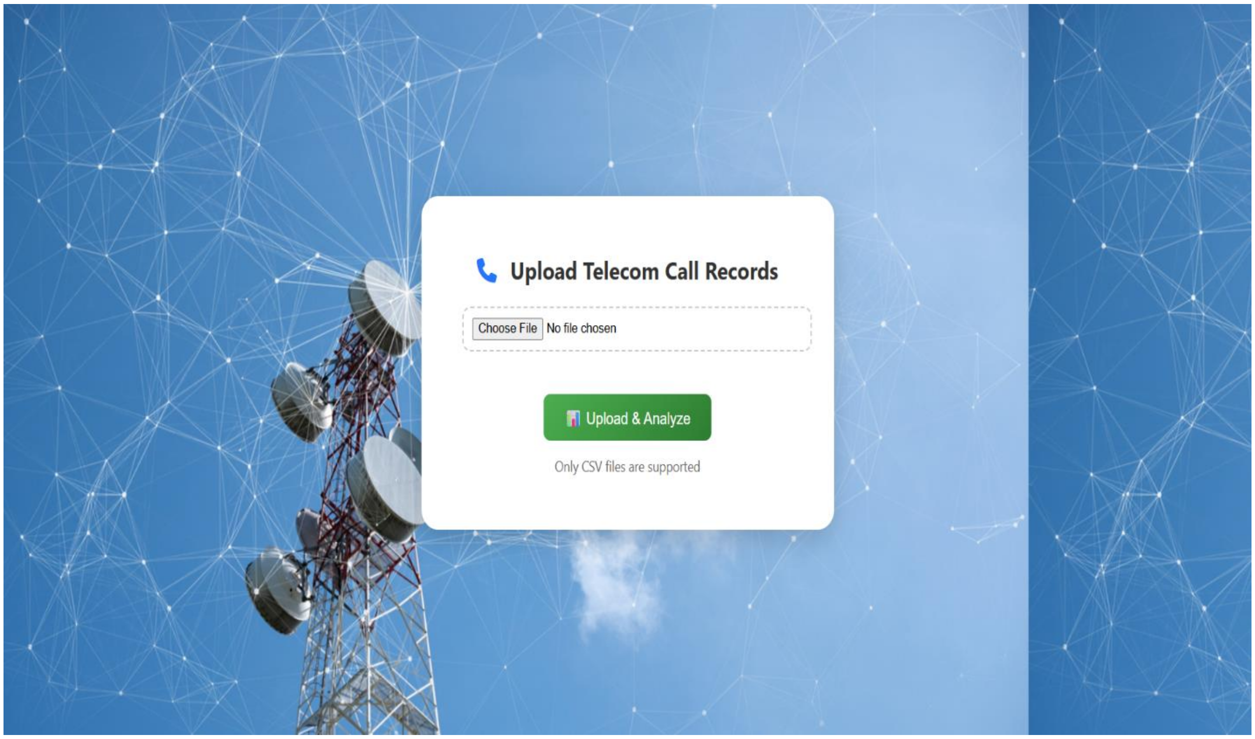
•The **body** section sets the font to Arial, background color to light gray (`#f4f6f8`), and removes margins/padding.

•The **.container** class styles the main box with 90% width, centered alignment, white background, padding, rounded corners, shadow, and top margin.

•The **h1** heading is centered with dark gray text color.

•The **button** is styled with a blue background (`#007BFF`), white text, and padding for better appearance.

7.OUTPUT AND SCREENSHOTS

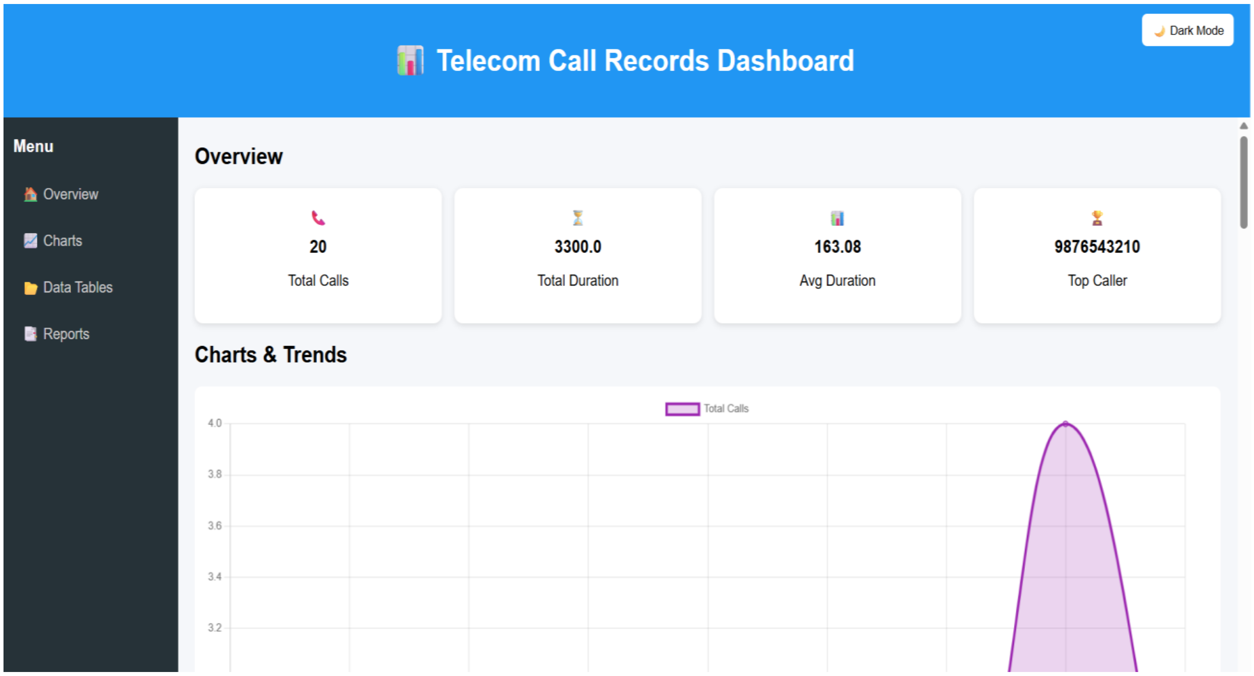


☎ Upload Telecom Call Records

No file chosen

Only CSV files are supported

- **File Upload Section** – Allows users to upload telecom call records in CSV format.
- **Upload & Analyze Button** – Processes the uploaded file and redirects to the analysis dashboard.
- **User Guidance** – Mentions that only CSV files are supported for compatibility.



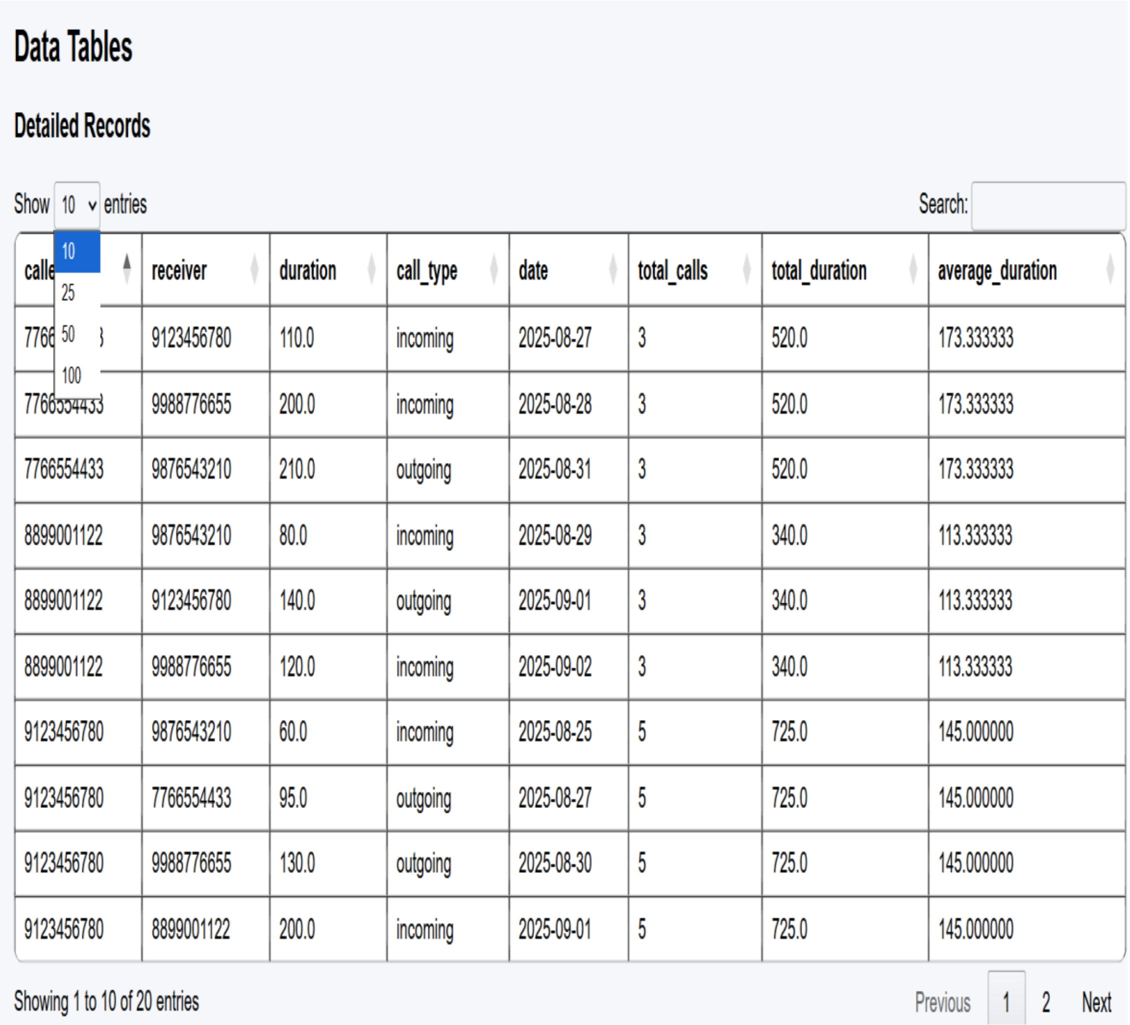
📊 Telecom Call Records Dashboard

Overview

📞	⏳	📊	🏆
20	3300.0	163.08	9876543210
Total Calls	Total Duration	Avg Duration	Top Caller

Charts & Trends

- **Overview Metrics** – Displays total calls, total duration, average call duration, and top caller in summary cards.
- **Charts & Trends** – Visual representation of call data over time for better insights.
- **Navigation Menu** – Sidebar with options like Overview, Charts, Data Tables, and Reports.
- **Dark Mode Option** – Provides a toggle for switching between light and dark themes.



Data Tables

Detailed Records

Show 10 entries

Search:

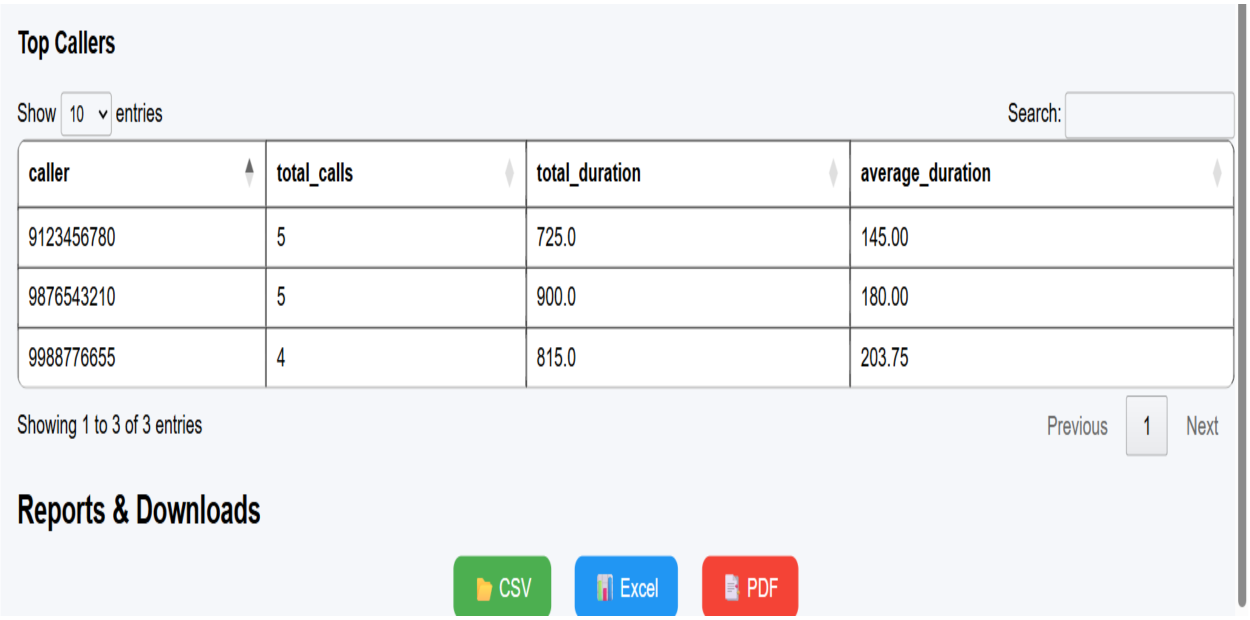
calle	receiver	duration	call_type	date	total_calls	total_duration	average_duration
776[illegible]3	9123456780	110.0	incoming	2025-08-27	3	520.0	173.333333
7766554433	9988776655	200.0	incoming	2025-08-28	3	520.0	173.333333
7766554433	9876543210	210.0	outgoing	2025-08-31	3	520.0	173.333333
8899001122	9876543210	80.0	incoming	2025-08-29	3	340.0	113.333333
8899001122	9123456780	140.0	outgoing	2025-09-01	3	340.0	113.333333
8899001122	9988776655	120.0	incoming	2025-09-02	3	340.0	113.333333
9123456780	9876543210	60.0	incoming	2025-08-25	5	725.0	145.000000
9123456780	7766554433	95.0	outgoing	2025-08-27	5	725.0	145.000000
9123456780	9988776655	130.0	outgoing	2025-08-30	5	725.0	145.000000
9123456780	8899001122	200.0	incoming	2025-09-01	5	725.0	145.000000

Showing 1 to 10 of 20 entries

- **Interactive Table** – The table displays telecom call records with columns like caller, receiver, duration, call type, date, total calls, total duration, and average duration.

- **User Controls** – It has a dropdown to select how many entries to show (10, 25, 50, 100) and a search box to quickly filter results.

- **Pagination & Sorting** – The table supports page navigation (Previous/Next) and sorting by clicking column headers for better data analysis.



Top Callers

caller	total_calls	total_duration	average_duration
9123456780	5	725.0	145.00
9876543210	5	900.0	180.00
9988776655	4	815.0	203.75

Showing 1 to 3 of 3 entries

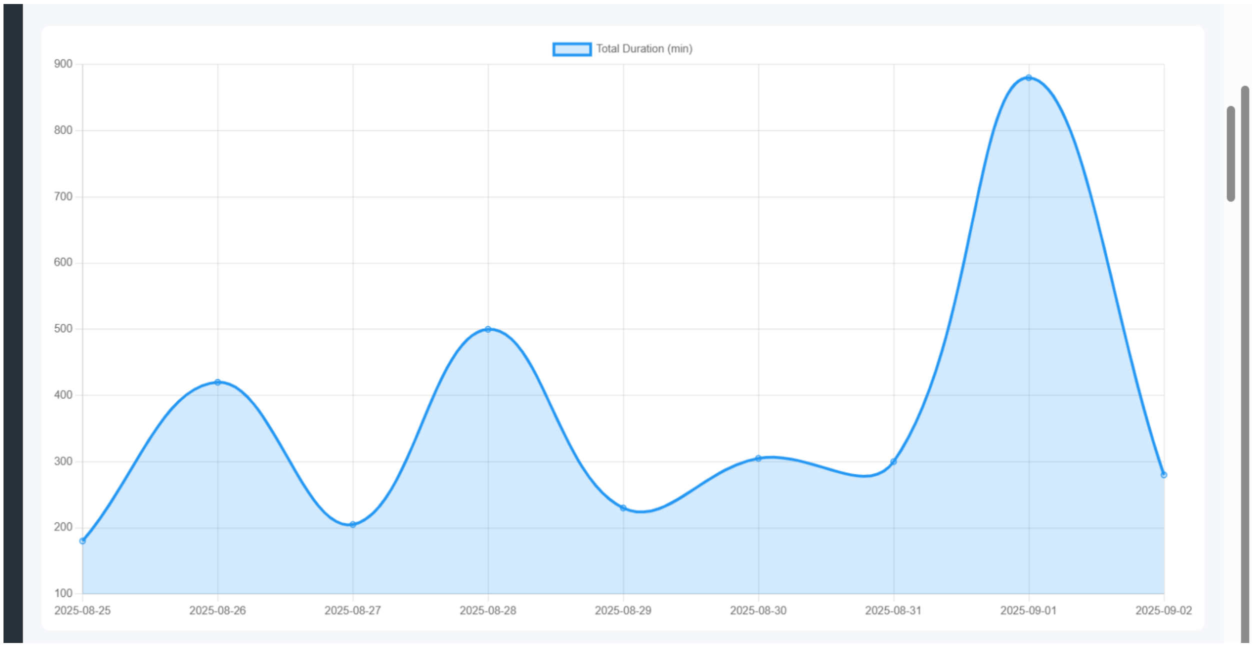
Reports & Downloads

- **Top Callers Summary** – The table highlights the top callers with details like total calls made, total duration of calls, and average duration per call.

- **Interactive Table** – Users can control entries shown, search specific callers, and sort data by columns.

- **Reports & Downloads** – Provides options to download reports in **CSV, Excel, and PDF formats**, making it easy for users to export and analyze data.

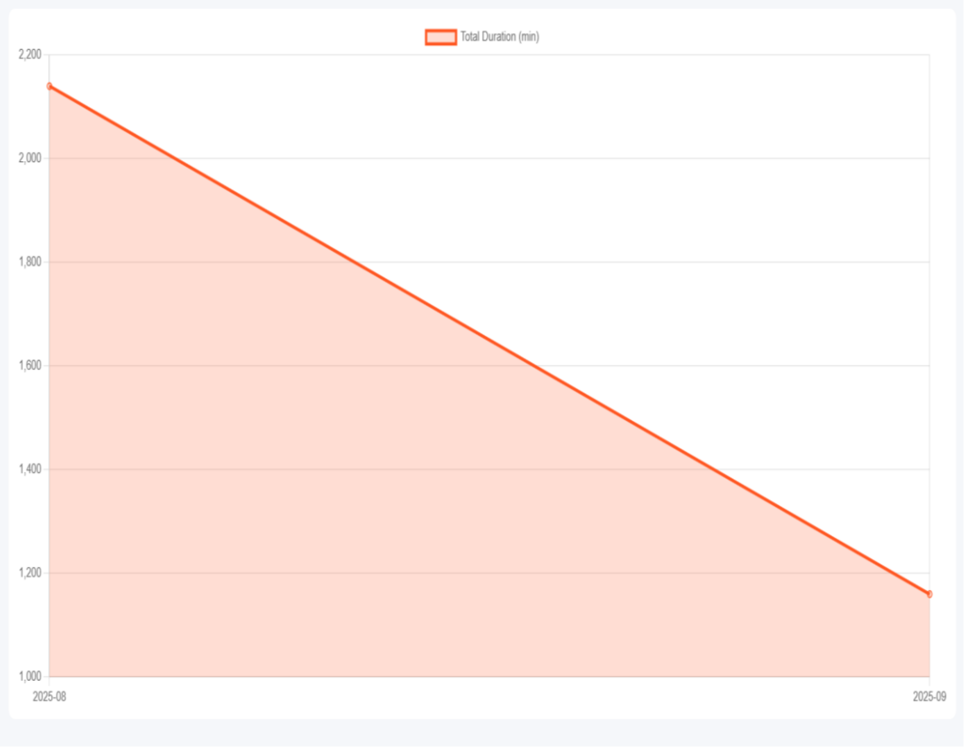
GRAPH



- Shows **daily total call duration (minutes)** from *25 Aug to 2 Sep 2025*.

- The graph fluctuates with peaks on **26 Aug (~420 mins), 28 Aug (~500 mins), and 1 Sep (~880 mins)**.

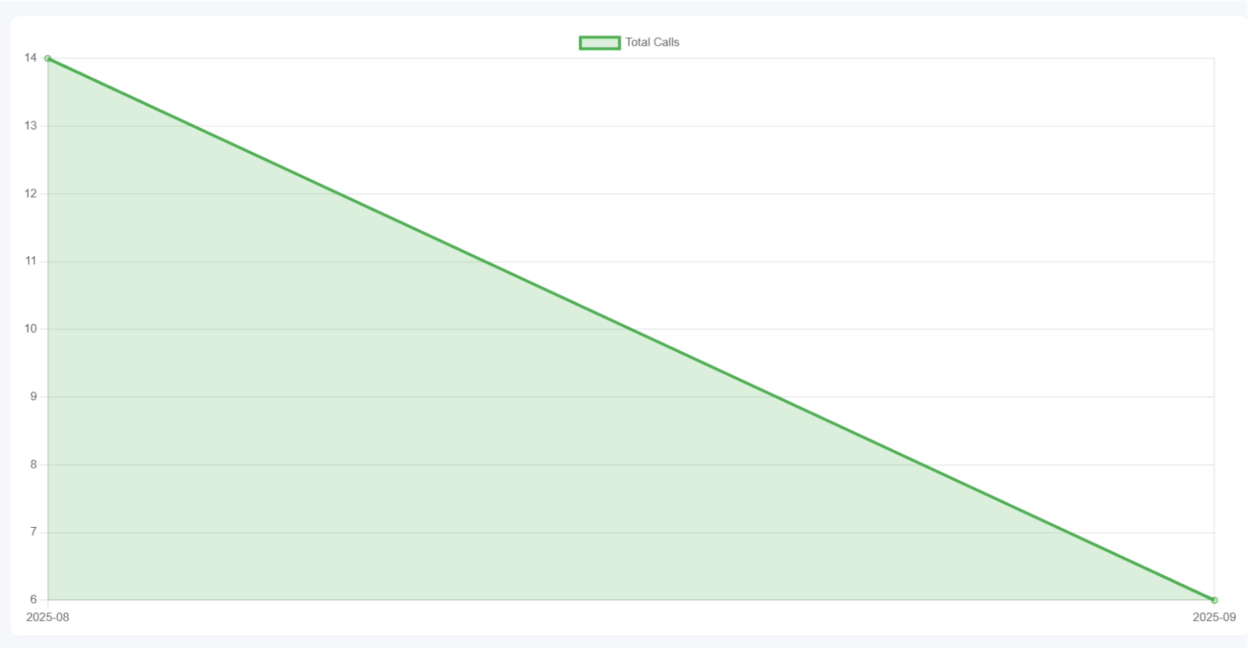
•Feature: Highlights **daily variations** and helps identify **high usage days**.



•Displays the **overall call duration trend by month (Aug–Sep 2025)**.

•Duration **drops from ~2200 mins in Aug to ~1150 mins in Sep**.

• Feature: Useful for **long-term trend analysis**.



8. Closure

The **Telecom Call Records Analysis Project** successfully demonstrated how raw call log data can be **processed, summarized, and visualized** using **Python (Pandas for data analysis)** and **Flask (for backend web framework)**. By uploading and analysing a CSV file containing fields like **caller, receiver, duration, call type, and date**, the system automatically generates insights in both **tabular and graphical formats**.

Key features included are:

- **Summary Report:** Highlights the **total call duration, total calls made, and average call duration** for each caller.
- **Detailed Records Table:** Displays **all individual call transactions** for reference and verification with sorting, searching, and pagination options.
- **Top Callers Section:** Identifies the **most active callers** with their total calls, total duration, and average call length.

- **Daily & Monthly Analysis Charts:** Visualizes trends of call duration and call counts across days and months for better usage insights.
- **Interactive Data Table Features:** Includes **filters, search box, and entries per page selection** for easy navigation of records.
- **Export Options:** Users can **download reports in CSV, Excel, or PDF formats** for offline use and sharing.
- **Automation of Analysis:** Eliminates manual calculations by automatically aggregating data once uploaded.
- **Scalability:** The framework can be extended to include **real-time call record monitoring, user authentication, and cloud integration** for larger datasets.

.

9. Bibliography

1. Pandas Documentation – https://pandas.pydata.org/docs/
2. Flask Documentation – https://flask.palletsprojects.com/
3. Python Official Docs – https://docs.python.org/3/
4. Stack Overflow – https://stackoverflow.com/