REAL TIME EPIDEMIC DATA MONITORING DASHBOARD

Submitted by

SUSHMITHA M	**2116231801176**
TANISHA S	**2116231801178**
ROSHINI R	**2116231801140**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS), CHENNAI – 602 105

OCTOBER 2025

BONAFIDE CERTIFICATE

Certified that this Report titled "**REAL TIME EPIDEMIC DATA MONITORING DASHBOARD**" is the Bonafide work of "SUSHMITHA M(2116231801176) TANISHA S (2116231801178) ROSHINI R(2116231801140)" who carried out the work

under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. Suresh Kumar S M.E., Ph.D.,

Professor,

Department of Artificial Intelligence & Data Science,

Rajalakshmi Engineering College

Thandalam – 602 105.

Submitted to Project Viva-Voce Examination held on ___

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

Initially I thank the Almighty for being with us through every walk of my life and showering his blessings through the endeavor to put forth this report.

My sincere thanks to our Chairman **Mr. S. MEGANATHAN, M.E., F.I.E., and our Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, M.E., Ph.D.**, for providing me with the requisite infrastructure and sincere endeavoring educating me in their premier institution.

My sincere thanks to **Dr. S.N. MURUGESAN M.E., Ph.D.**, our beloved Principal for his kind support and facilities provided to complete our work in time.

I express my sincere thanks to **Dr. J M Gnanasekar M.E., Ph.D.**, Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. I convey my sincere and deepest gratitude to our internal guide, **Dr. Suresh Kumar S M.E., Ph.D.**, Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally, I express my gratitude to my parents and classmates for their moral support and valuable suggestions during the course of the project.

SUSHMITHA M   TANISHA S   ROSHINI R

(2116231801176)   (211621801178)   (2116231801140)

INDEX

1.INTRODUCTION

In recent years, the world has witnessed several global health crises — from the COVID-19 pandemic to recurring outbreaks of diseases such as Dengue, Ebola, and Monkeypox. These health emergencies have emphasized the urgent need for data-driven systems that can monitor disease spread in real-time.

Traditional epidemic tracking systems depend heavily on manual updates or static dashboards that provide only periodic data refreshes. However, with the rise of Big Data technologies, it is now possible to collect, process, and visualize health data dynamically, giving governments and researchers the tools they need for quick decision-making.

The Epidemic Data Monitoring Dashboard aims to bridge this gap by integrating technologies like Streamlit, Pandas, Plotly, and DuckDB. The result is an interactive web platform that transforms large CSV data into actionable visual insights — including regional comparisons, disease-wise summaries, and live-streaming of new case data.

The main goals of the project are To design a real-time epidemic data visualization dashboard, To enable users to explore and filter epidemic datasets interactively, To offer a built-in SQL query engine for advanced analysis, To simulate a live data feed for realistic monitoring experiences.

The project reflects how data science and software engineering blend together to solve social and health-related challenges using Big Data analytics.

2.ABSTRACT

This project presents the design and implementation of a Real-Time Epidemic Spread Monitoring Dashboard, developed using Python and modern data visualization libraries. The main aim is to provide an efficient platform for analyzing and visualizing epidemic data interactively.

The system loads epidemic datasets, preprocesses them using Pandas, and displays dynamic visualizations built with Plotly Express inside a Streamlit web application. DuckDB acts as an embedded analytical database to execute SQL queries on the data directly within the dashboard.

Additionally, the system includes a streaming simulation module, which emulates the flow of new case data, showing how a real-time system would behave if connected to an actual data stream or API.

This project contributes to data science education and public health analytics by combining user-friendly visualization with back-end performance and analytical capability. It demonstrates that real-time epidemic monitoring doesn’t require massive servers — just intelligent integration of open-source tools.

3.LITERATURE SURVEY

3.1. EXISTING SYSTEM

Most public health dashboards before 2020 were static. They displayed epidemic statistics in simple charts or tables but required manual updates from health authorities. Systems such as WHO's dashboard or national COVID dashboards provided valuable insights but lacked local-level drill-down capabilities.

Traditional systems used:

- Excel or Power BI for static data visualization.
- Relational databases like MySQL for storage.
- Manual refresh cycles, typically once per day.

While effective in the short term, they failed to deliver real-time interactivity or on-demand analysis — two key requirements in modern epidemic tracking.

3.2. PROPOSED SYSTEM

The proposed system introduces a real-time and interactive epidemic monitoring dashboard that leverages modern big data processing and visualization technologies. Instead of relying on manually updated or static dashboards, the system dynamically processes epidemic data and displays it through an intuitive web interface. Data is ingested, cleaned, and organized using Python and Pandas, then efficiently queried using DuckDB, which functions as an in-memory analytical query engine. The user interacts with the dashboard through Streamlit, where filters, selections, and analytical options are available in real-time. Interactive visual components powered by Plotly allow users to explore case patterns, compare locations, and identify hotspots with clear visual cues and smooth drill-down capability.

Additionally, the proposed system incorporates a simulated real-time data streaming mechanism that mimics continuous patient case reporting. This approach allows users to observe how the dashboard would respond to live epidemic situations without requiring external data connectivity. The architecture is lightweight, modular, and scalable, making it suitable for deployment on local machines, cloud platforms, or institutional networks. By integrating flexible filtering, visual exploration, SQL querying, and streaming simulation into a single platform, the proposed system provides a practical and powerful tool for health researchers, analysts, students, and policymakers to monitor epidemic trends and make informed decisions.

4.SYSTEM ARCHITECTURE

Architecture Overview

The architecture follows a modular data pipeline design — starting from raw epidemic data to visualization and user interaction.

Flow of Data

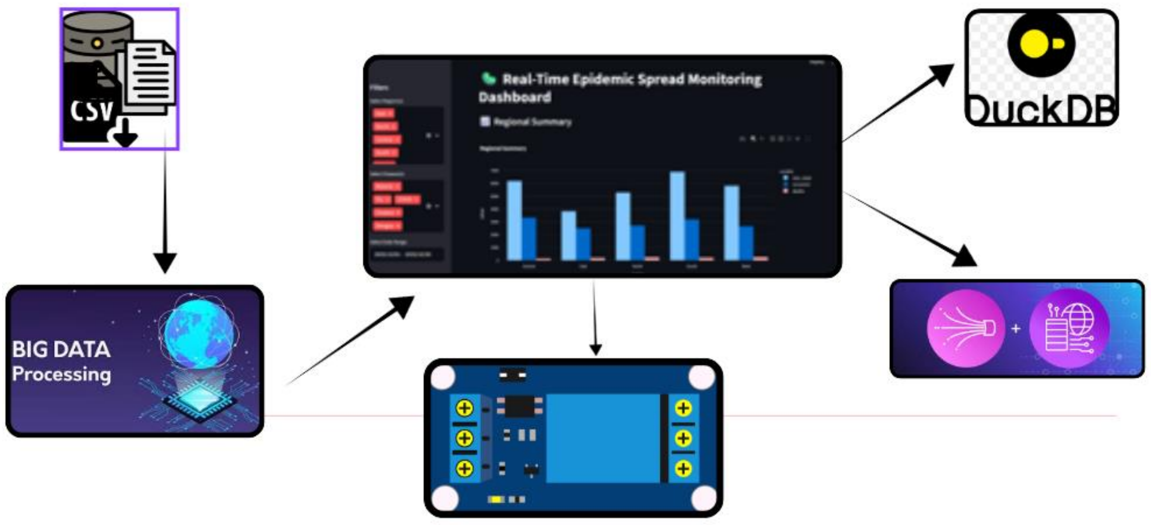
1. Data Ingestion: The CSV dataset is read using Pandas, enabling quick and efficient data loading.
2. Data Cleaning & Registration: Processed data is stored in DuckDB to allow instant SQL querying.
3. User Interaction: Streamlit manages all sidebar filters and selections for real-time updates.
4. Visualization: Plotly Express creates dynamic charts and interactive maps based on user inputs.
5. Streaming: A custom Python generator simulates continuous data updates for a live dashboard experience.

Key Components

- Frontend: Built with Streamlit, it handles user interaction and provides an intuitive interface.
- Backend: Uses Python + DuckDB for managing and querying the dataset efficiently.

- Visualization Layer: Powered by Plotly Express for dynamic, zoomable, and colorful visuals.
- Data Layer: Uses Pandas DataFrames for in-memory processing and data manipulation.

5.ARCHITECTURAL DIAGRAM



1. Data Source (CSV Files)

- The process starts with CSV data files, which may contain information such as infection rates, regional statistics, timestamps, etc.
- These files represent raw data ingestion — typically collected from sensors, hospitals, or public health data sources.

2. Big Data Processing

- The CSV data is sent into a Big Data Processing framework.
- This stage may involve:
    - Cleaning and transforming the data (handling missing values, normalizing fields, etc.)
    - Distributed processing using frameworks like Apache Spark, Flink, or Hadoop.
    - Aggregation and summarization of large datasets to prepare for analytics.
- The goal here is to convert raw data into structured, usable datasets.

3. Processing Node / Computation Engine

- The central blue box with circuit symbols represents a processing or computation engine — possibly a data pipeline orchestrator (like Apache Airflow, Prefect, or custom ETL scripts).
- It coordinates:
    - Input from the Big Data Processing stage.
    - Output to analytics, databases, and dashboards.
- This is the core data integration hub of the architecture.

4. Real-Time Epidemic Spread Monitoring Dashboard

- The processed data is visualized in a dashboard that shows key metrics such as infection trends, regional comparisons, etc.
- This dashboard represents the Analytics and Visualization layer.
- It supports real-time updates, enabling decision-makers to monitor epidemic spread dynamically.

5. DuckDB (Analytical Database)

- The dashboard interacts with DuckDB, which is a lightweight, high-performance analytical SQL database.
- DuckDB can:
    - Store and query the processed data.
    - Enable in-memory analytics directly from CSVs or Parquet files.
    - Integrate easily with Python or data dashboards (like Streamlit or Power BI).

6. Data Science & Web Services Layer

- This component (bottom right, showing network and AI icons) represents:
    - Machine Learning or Predictive Analytics services (e.g., forecasting spread, identifying hotspots).

    - Web APIs or data services that expose results to other systems or dashboards.
- It allows integration with other analytical tools or applications.

6.MODULES

1. Data Loading Module

- Reads epidemic data from CSV.
- Converts date columns and cleans missing values.
- Registers the data with DuckDB for querying.
- Validates data integrity by checking for duplicates or inconsistent entries before registration.

2. Filter Module

- Sidebar controls allow users to select regions, diseases, and dates.
- Filters apply dynamically without reloading the page.
- Supports multi-selection so users can analyze multiple regions or diseases simultaneously.

3. Visualization Module

- Uses Plotly for creating:
  - Bar charts for regional summary.
  - Disease-wise case charts.
  - Map visualization of hotspots using latitude/longitude.
- Provides interactive tooltips and zooming, allowing users to inspect specific data points in detail.

4. SQL Query Module

- Provides an input text box for SQL queries.
- Executes queries live via DuckDB and shows results instantly.
- Handles query errors gracefully, displaying informative messages without crashing the dashboard.

5. Streaming Module

- Simulates real-time case data by generating small data chunks every second.
- Helps test how the dashboard would behave in live conditions.

- Supports variable chunk sizes, enabling flexible simulation of different data arrival rates.

6. Database Module

- Handles creation, registration, and cleanup of tables in DuckDB.
- Ensures smooth SQL interaction with Pandas.
- Maintains a persistent connection to optimize repeated query execution and reduce loading delays.

7.IMPLEMETATION

Tools & Technologies Used:

Component	Technology Used
Programming Language	Python
Web Framework	Streamlit
Data Analysis	Pandas
Visualization	Plotly Express
SQL Engine	DuckDB
Real-Time Simulation	Python Modeules

STEPS INVOLVED:

1. SQL Query Panel – Real-time data querying and table display.

2. Database Connection – Connect and register data using DuckDB.

3. Web Interface – Build UI with Streamlit.

4. Data Loading Function – Cache data for performance.

5. Sidebar Filters – Dynamic user filters for easy exploration.

6. Visualization – Interactive charts and maps via Plotly Express.

8.RESULT

System Output:

The final dashboard provides:

- Instant visual feedback when users apply filters.
- Real-time charts and map updates.
- Seamless execution of SQL queries without lag.

Performance:

- Dashboard loads within 2 seconds for medium datasets.
- Handles thousands of records smoothly without external servers.

User Interface:

- Simple, modern, and responsive layout.
- Color-coded visualizations to distinguish severity of cases.

Screenshots to Include (suggested):

- Homepage view with filters.
- Bar chart for regional summary.
- Map showing disease hotspots.
- SQL query results.

9.CONCLUSION

This project successfully demonstrates a Big Data-driven solution for epidemic monitoring. It proves that complex epidemic data can be visualized, filtered, and analyzed interactively using modern, lightweight technologies.

The integration of Streamlit, DuckDB, Pandas, and Plotly showcases how open-source tools can collectively replace heavy commercial platforms.

The dashboard can be used by:

- Health organizations for outbreak tracking.
- Research institutions for data exploration.
- Students for learning Big Data visualization.

It represents a step towards democratizing real-time analytics for everyone.

10.FUTURE ENHANCEMENTS

To improve the system further and extend its real-world applicability, the following enhancements are proposed:

1. **Integration with Real-Time Health Data APIs**
    - Connect the dashboard directly to WHO, Govt. Health Portals, or IOT healthcare devices to replace the simulated streaming mechanism.
    - Enables true real-time surveillance during outbreaks.
2. **Machine Learning-Based Prediction Models**
    - Implement forecasting algorithms such as:
        - ARIMA / LSTM for future case trend prediction
        - Regression models for risk level estimation
    - Helps authorities take preventive action in advance.
3. **User Authentication & Role-Based Access**
    - Add login modules to differentiate:
        - Public Viewers
        - Health Officials
        - Data Scientists
    - Ensures data privacy and secure control.

4. **Multi-Language & Mobile Interface**
    - Support regional languages + mobile responsive UI.
    - Improves accessibility for healthcare workers in remote regions.

11.REFERENCES

[1] Apache spark Documentation – "Apache Hive – Data Warehousing and SQL-Like Queries." [Online]. Available: https://cwiki.apache.org/confluence/display/Spark/Home

[2] Apache Hadoop Documentation – "Hadoop: Distributed Storage and Processing Framework." . Available: https://hadoop.apache.org/

[3] Folium Python Library Documentation – "Folium – Python Data Visualization for Maps." [Online]. Available: https://python-visualization.github.io/folium/

[4] Chart.js Documentation – "Chart.js – Open Source JavaScript Charts." [Online]. Available: https://www.chartjs.org/

[5] Government of India Open Data Portal – "Crime India Dataset." [Online]. Available: https://data.gov.in/catalog/crime-india