VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering)

Department of Computer Engineering

Project Report on

Data Insights using Large Language Models

Submitted in partial fulfillment of the requirements of Third Year (Semester–VI), Bachelor of Engineering Degree in Computer Engineering at the University of Mumbai Academic Year 2024-25

By

1. Varun Budhani / D12C / 10
2. Yash Ingale / D12C / 29
3. Harsh Pimparkar / D12C / 51
4. Prem Ghundiyal / D12C / 22

Project Mentor
Dr. Mrs. Sujata Khedkar

University of Mumbai
(AY 2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering)

Department of Computer Engineering

CERTIFICATE

This is to certify that **Varun Budhani(D12C / 10), Yash Ingale(D12C / 29), Harsh Pimparkar(D12C / 51), Prem Ghundiyal(D12C / 22)** of Third Year Computer Engineering studying under the University of Mumbai has satisfactorily presented the project on "**Data Insights using Large Language Models**" as a part of the coursework of Mini Project 2B for Semester-VI under the guidance of **Prof. Sujata Khedkar.** in the year 2024-25.

_____

Date

_____ _____

Internal Examiner External Examiner

Project Mentor	Head of the Department	Principal
Prof. Sujata Khedkar	Dr. Mrs. Nupur Giri	Dr. J. M. Nair

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---	---
(Signature)	(Signature)
---	---
Varun Budhani 10	Harsh Pimparkar 51
---	---
(Signature)	(Signature)
---	---
Yash Ingale 29	Prem Ghundiyal 22

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Associate Professor **Dr. (Mrs.) Sujata Khedkar** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR T.E MINI PROJECT 2B

Learners will be to:-

CO No.	COURSE OUTCOME
CO1	Identify problems based on societal /research needs.
CO2	Apply Knowledge and skill to solve societal problems in a group.
CO3	Develop interpersonal skills to work as a member of a group or leader.
CO4	Draw the proper inferences from available results through theoretical/ experimental/simulations.
CO5	Analyze the impact of solutions in societal and environmental context for sustainable development.
CO6	Use standard norms of engineering practices
CO7	Excel in written and oral communication.
CO8	Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
CO9	Demonstrate project management principles during project work.

ABSTRACT

In today’s data-driven world, organizations and individuals generate vast amounts of data that require effective analysis and visualization to extract meaningful insights. However, existing data analysis tools often demand technical proficiency in programming languages, statistical methods, or specialized business intelligence (BI) software, creating a barrier for non-technical users. To bridge this gap, Data Insights introduces a novel approach by integrating a Large Language Model (LLM) to facilitate seamless and intuitive data interaction.

The Data Insights platform allows users to upload datasets in CSV or Excel formats and interact with their data through natural language commands instead of complex coding or manual spreadsheet manipulations. Users can issue conversational queries—such as "Show the monthly sales trends" or "Create a pie chart of customer distribution by region"—and receive instantly generated visualizations, including bar charts, line graphs, scatter plots, and more. The system effectively interprets user intent, processes relevant data transformations, and presents insights in a user-friendly manner, significantly reducing the learning curve for data analysis.

One of the core strengths of Data Insights is its ability to provide real-time processing and visualization. Users can dynamically explore different aspects of their data by refining queries, adjusting filters, or requesting alternative visualization types—all without requiring specialized software knowledge. This adaptability enhances decision-making efficiency, making the tool suitable for applications in business analytics, financial forecasting, healthcare research, and educational performance tracking.

Additionally, the platform is designed to handle diverse datasets and adapt to various user requirements, ensuring flexibility across industries. By leveraging LLM-driven automation, Data Insights eliminates the manual effort typically associated with data preprocessing, formatting, and visualization selection. This automation not only enhances productivity but also democratizes access to advanced data analytics, enabling individuals and organizations to make informed decisions faster and with greater accuracy.

Index

Chapter 1 Introduction

This chapter provides an introduction to the Data Insights platform, outlining the motivation behind its development, the problem it aims to solve, and the limitations of existing systems. It highlights the growing need for accessible data visualization tools and introduces the concept of using Large Language Models (LLMs) to bridge the gap between technical complexity and user-friendliness. The chapter also discusses the relevance and significance of the project in today's data-driven landscape.

1.1 Introduction

In today's data-driven world, organizations and individuals generate vast amounts of data that hold valuable insights. However, extracting meaningful information from complex datasets often requires technical expertise in programming languages, statistical tools, or specialized software.

Data Insights is an innovative platform designed to bridge this gap by integrating Large Language Models (LLMs) with an interactive data visualization and analysis system.

Key Features and Functionality:

- Natural Language Interaction:
  Users can interact with datasets effortlessly using natural language queries, eliminating the need for coding skills or specialized software knowledge.
- Simple Data Uploads:
  The platform supports datasets in CSV or Excel formats, allowing users to quickly upload their files and begin analysis.
- Instant Visualization Requests:
  Users can request various types of visualizations by simply typing prompts like:
  - "Show me a trend analysis of sales over the last year."
  - "Generate a scatter plot of customer age vs. purchase frequency."
- Real-Time Processing:
  Data Insights processes commands in real time, accurately interprets the user's intent, and produces meaningful visual representations for dynamic data exploration

1.2 Motivation

As data continues to grow across industries, the need for data-driven decision-making has become essential. However, most traditional tools—like Tableau, Power BI, or custom-coded solutions—require technical skills in programming languages such as Python or R. This often creates a barrier for non-technical users, preventing them from fully utilizing their data.

This project aims to bridge that gap by integrating Large Language Models (LLMs) with visualization tools, allowing users to interact with data through natural language.

- Users can upload CSV or Excel files.
- They can ask queries like:
    - "Show me a bar chart of sales by region."
    - "Generate a pie chart of customer distribution."
- The system processes these commands and generates the required charts automatically.

By eliminating the need for coding or specialized training, the project:

- Democratizes data analysis
- Empowers a broader audience—from small business owners to policymakers
- Boosts productivity and decision-making speed

In short, it makes data exploration as easy as having a conversation.

1.3 Problem Definition

Data visualization and analysis tools like Tableau, Power BI, Matplotlib, and D3.js offer powerful features but are designed for technically proficient users, requiring coding knowledge and expertise in data manipulation. This creates a barrier for non-technical users such as business managers, educators, and healthcare professionals, who often rely on analysts for basic data insights, increasing costs and slowing decision-making.

To address these challenges, Data Insights is proposed as an intelligent platform powered by Large Language Models (LLMs) that allows users to interact with data through natural language queries, eliminating the need for coding or specialized knowledge. Users can simply ask questions like, "Show me the sales trends over the past five years" or "Generate a pie chart of customer distribution," and the system will automatically generate relevant visualizations.

Supporting commonly used data formats like CSV and Excel, the platform makes data analysis accessible and efficient across industries. It also offers dynamic, interactive exploration, allowing users to engage in ongoing dialogues with their data, uncovering deeper insights.

In summary, Data Insights aims to democratize data analytics by making it intuitive and accessible, enabling users from all backgrounds to make data-driven decisions without technical barriers.

1.4 Existing Systems

Several tools exist in the data visualization and analytics space, offering varying degrees of functionality and user-friendliness. While these platforms have made significant strides in enabling data-driven insights, they often come with certain limitations, especially for non-technical users. Below is a comparison between the proposed Data Insights system and other popular tools.

Microsoft Power BI

Microsoft Power BI is a widely-used platform that offers comprehensive business intelligence and visualization capabilities. Users can create dashboards, connect multiple data sources, and generate reports to analyze their data effectively.

- Similarities:
    - Offers robust data visualization tools.
    - Supports dashboards and real-time reporting for sharing insights.
- Differences:
    - Requires some level of technical knowledge for creating dashboards and managing data models.
    - Natural language support (via Q&A feature) exists but is limited; it does not dynamically generate visualizations as intuitively as Data Insights.

Tableau

Tableau is another powerful visualization platform known for its drag-and-drop interface and wide range of charting capabilities.

- Similarities:
    - Provides interactive visualization features.
    - Enables users to build complex dashboards with ease.
- Differences:
    - Users must manually select chart types and filters.
    - Less intelligent automation—visualizations are not suggested or generated based on user intent.
    - The learning curve can still be steep for beginners.

ThoughtSpot

ThoughtSpot integrates search-driven analytics, allowing users to query data using natural language.

- Similarities:
    - Supports natural language querying for generating visual insights.
- Differences:
    - Designed primarily for enterprise-level users with large-scale data needs.
    - May be complex or resource-intensive for smaller organizations or casual users.
    - Data Insights is more lightweight, intuitive, and geared toward accessibility for all user levels.

OpenAI's Code Interpreter (Advanced Data Analysis in ChatGPT)

OpenAI's Code Interpreter (now part of ChatGPT's Advanced Data Analysis) allows users to perform data analysis through natural language instructions.

- Similarities:
    - Uses natural language input to process data and create visualizations.
    - Capable of understanding user intent and generating relevant charts.
- Differences:
    - Requires manual data upload and is limited to a single-turn interaction style.
    - Lacks a dedicated, persistent interface for data exploration and visualization management like Data Insights offers.
    - Visualization is often static and session-dependent.

1.5 Lacuna of the existing systems

Modern data visualization tools like Tableau, Power BI, and Matplotlib are powerful but often too complex for non-technical users. They require knowledge of coding, data manipulation, or visualization practices, making them less accessible. Most rely on drag-and-drop or SQL-based interfaces, which still demand technical understanding. A key limitation is the lack of natural language processing (NLP), which could let users simply ask questions like "Show revenue growth by region" and receive instant visualizations.

Key limitations of existing systems include:

- **Steep Learning Curve**: Most platforms require programming knowledge or training in data analytics.
- **Technical Barriers**: Non-technical users often struggle with data manipulation and visualization setup.
- **Lack of Natural Language Support**: Conversational interaction with data is rarely available or limited.

1.6 Relevance of the Project

In a world where data drives decisions, access to effective visualization should be universal. Yet most current tools remain out of reach for non-technical users. Solutions like Power BI, Tableau, and Looker Studio are feature-rich but complex. Even tools like ThoughtSpot, which allow natural language queries, primarily target enterprise-level users and often lack flexibility.

Core benefits of Data Insights include:

- **Bridging the Technical Gap**: Users can interact with data through simple language, eliminating the need for coding.
- **Automated Visualization Selection**: The system chooses the most suitable chart type automatically based on the input data.

- **Improved Accessibility**: Enables students, business users, and educators to derive insights without technical hurdles.
- **Faster Insight Generation**: Reduces the time needed to configure and generate reports manually.
- **Cross-Domain Usability**: Suitable for finance, healthcare, education, marketing, and more.
- **Enhanced Data Discovery**: Helps users determine the best way to visualize their data for clearer interpretation.
- **Reduced Dependency on Analysts**: Allows users to independently analyze data, cutting down on turnaround time and resource use.

By addressing the pain points of existing tools, **Data Insights** offers a modern, user-friendly solution that puts data-driven decision-making into everyone's hands.

Chapter 2 Literature Survey

This chapter reviewed key research works and patents in the field of data visualization powered by natural language and Large Language Models (LLMs). It identified how current systems attempt to bridge the gap between technical data analysis and user accessibility, while also outlining their limitations in terms of scalability, transparency, and user adaptability. The insights drawn from this literature lay the foundation for the proposed Data Insights system, which aims to offer a more intuitive, automated, and intelligent approach to data visualization through natural language interaction.

A. Overview of Literature Survey

The Literature Survey section aims to provide a comprehensive review of existing systems, tools, and research related to data visualization and user interaction through natural language interfaces. It explores the state-of-the-art in data visualization platforms, highlighting their strengths and limitations, particularly in terms of accessibility for non-technical users. This review will identify the gaps in current solutions, demonstrating the need for a system like Data Insights, which integrates Large Language Models (LLMs) to make data analysis more intuitive and interactive. By examining existing approaches, this section establishes the foundation for the proposed system and its contributions to the field.

B. Inference drawn

This section discusses research papers, patents, and existing methodologies related to tree enumeration, highlighting their contributions and limitations.

2.1 Research Papers Referred

Recent advancements in Large Language Models (LLMs) have led to an increasing number of intelligent frameworks that translate natural language into actionable insights and data visualizations. A significant contribution in this area comes from Dr. Sudha S.V. et al. [1], who present a system that empowers non-technical users to generate visualizations by combining LLMs with structured data querying and visualization tools. Their platform simplifies interaction with CSV files and produces meaningful visuals via natural language commands. While effective, this method faces limitations such as dependency on precise prompt engineering and model interpretability.

In the financial analytics space, Dolphi et al. [2] explore how LLMs can transform unstructured financial news into structured insights, enabling faster, more informed decision-making. Their pipeline includes standardization, LLM-based summarization, and insight extraction. Although powerful, their system is sensitive to input quality and struggles with model transparency and bias.

The hybrid system introduced by Vertsel and Rumiantsau [3] proposes combining deterministic rule-based engines with LLMs to strike a balance between logic and language-driven flexibility. This architecture improves insight accuracy through rule-based pre-processing while using LLMs to handle ambiguous tasks. Despite its effectiveness, the approach increases system complexity and requires frequent rule updates.

Perozzi et al. [4] delve into the challenge of making tabular data LLM-friendly by introducing strategies for encoding and serializing structured data into text. These techniques enable better model alignment and execution of analytical tasks but may lead to information loss and demand careful prompt construction.

Liu et al. [5] introduce JarviX, a no-code LLM-based analytics platform for tabular data, which automates ingestion, preprocessing, insight generation, and visualization. Designed for end-users, it improves accessibility but suffers from limitations in transparency and may introduce bias from underlying model behaviors.
Expanding on these ideas, Wu et al. [6] present a methodology where a fine-tuned LLM interprets natural language queries and generates data visualizations via API-based integration with visualization libraries. This system excels in flexibility and user accessibility but faces constraints due to computational expense, limited framework customization, and reliance on LLM accuracy.

A more specialized approach, Prompt4Vis by Li et al. [7], employs Example Mining and Schema Filtering to enhance the generation of relevant, schema-aligned visualizations from natural language. This method improves accuracy and relevance but still grapples with challenges like domain-specific limitations and dependence on high-quality training data.

DataVizGPT [8] leverages GPT-based models trained on a large corpus of textual descriptions and visualizations, showcasing strong performance in direct natural language-to-visual output conversion. Although user-friendly and innovative, its scalability and ability to support highly customized or domain-specific tasks remain constrained by dataset quality and computational complexity.

Further exploring interactivity, recent works such as Language-Driven Visualization Design [9] and Conversational Data Visualization [10] investigate how LLMs can be integrated into interactive exploration systems. These systems aim to enhance personalization, facilitate intuitive data querying, and improve user engagement. However, real-time processing demands high computational resources, and model behavior can be inconsistent with complex or ambiguous user queries.

Linguistic Data Visualization [11] takes this one step further by combining interactive design with user feedback loops, utilizing advanced LLMs to craft dynamic and adaptive visual experiences. While this model promotes user-centric design, it brings integration and scalability challenges due to variability in user input and resource constraints.

2.2 Patent Search

Several patents align with the core functionalities of Data Insights, particularly in leveraging natural language for automated data visualization:

1. **Using Natural Language Constructs for Data Visualizations**
    - **Patent No.** US11694036B2
    - **Summary**: A system that interprets natural language commands, identifies a semantic model of the dataset, parses the command into an intermediate expression using a context-free grammar, and generates visualizations from the queried data.
    - **Relevance**: Strongly supports the core function of natural language-driven visualization creation.
2. **Optimized Data Visualization According to Natural Language Query**
    - **Patent No.** US10572473B2
    - **Summary**: Automates chart type selection and visualization creation based on natural language queries, minimizing manual effort and expertise.
    - **Relevance**: Emphasizes intelligent chart selection, a vital feature for enhancing user experience in Data Insights.
3. **Natural Language Data Analysis, Visualization, and Editing System**
    - **Patent No.** US11960500B2
    - **Summary**: Enables users to perform analysis, generate visualizations, and even edit data using natural language inputs—without technical knowledge.
    - **Relevance**: Reflects a comprehensive approach to data interaction, mirroring the intended capabilities of Data Insights.
4. **Applying Natural Language Pragmatics in a Data Visualization User Interface**
    - **Patent No.** US11934461B2
    - **Summary**: Enhances visualizations based on user commands by extracting analytical intent, applying relevant functions, and updating visual output dynamically.
    - **Relevance**: Highlights the importance of interactive and responsive visualizations, a key aspect of Data Insights' user interface.

2.3 Inference Drawn

The exploration of existing patents in natural language-driven data visualization reveals a landscape rich with innovative approaches to integrating NLP with visual analytics.

One notable example is:

- **Using Natural Language Processing for Visual Analysis of a Data Set**
    - **Patent No.** US11244006B1
    - **Summary**: Describes a system that displays a visualization from a dataset. When a user issues a natural language command, the system identifies a data range, presents an editable control, updates the dataset based on user input, and refreshes the visualization accordingly.
    - **Relevance**: Emphasizes intuitive user interaction, allowing dynamic and real-time visualization updates via natural language inputs.

Inference:

This patent showcases how NLP can make data exploration more accessible by allowing users to manipulate visualizations without technical expertise. It aligns with Data Insights' core mission to democratize data analytics.

However, **Data Insights** sets itself apart by:

- **Automatically selecting optimal visualization types** based on both the user's query and data context.
- **Proactively guiding users** toward the most insightful data representations without needing prior visualization knowledge.

Chapter 3 Requirement Gathering for the Proposed System

This chapter outlined the comprehensive requirements essential for building the proposed system. It began by emphasizing the importance of requirement gathering in ensuring a user-centric and functional solution. The chapter detailed both functional and non-functional requirements—highlighting features such as natural language query processing, data visualization, and dashboard management, along with considerations for performance, scalability, and security. Additionally, the necessary hardware, software, and tools for development were discussed, followed by constraints like model processing time and data privacy concerns. These foundational insights guide the design and implementation of the system to ensure robustness, usability, and scalability.

3.1 Introduction to requirement gathering

Requirement gathering is a crucial phase in software development where the system's needs are identified, analyzed, and documented. It helps ensure that the final product meets user expectations while considering functional and non-functional aspects. This phase involves stakeholder discussions, feasibility analysis, and documentation of system requirements.

3.2 Functional Requirements

1. Data Upload & Management :

- Users should be able to upload datasets in CSV or Excel format.
- The system should validate uploaded files for format consistency and missing values.
- Users should be able to view a preview of the dataset after upload.
- The system should allow users to delete or replace uploaded datasets.

2. Natural Language Query Processing :

- Users should be able to enter natural language queries (e.g., “Show sales trends for the last 6 months”).
- The system should interpret user intent and map queries to appropriate data operations.
- The system should provide suggestions for query refinement if needed.
- Users should be able to modify queries dynamically and receive updated results.

3. Data Processing & Transformation :

- The system should apply data filtering, sorting, and grouping based on user queries.
- Users should be able to specify date ranges, categorical filters, and numeric conditions.
- The system should automatically detect and handle missing or inconsistent data.
- The system should perform basic statistical analysis (mean, median, sum, etc.) on request.

4. Data Visualization :

- Users should be able to request visualizations in natural language (e.g., "Create a bar chart for sales by region").
- The system should generate appropriate charts (bar, line, pie, scatter, etc.) based on the data type.
- Users should be able to customize visualizations (change chart type, colors, labels, etc.).
- The system should allow users to download generated visualizations as images or PDFs.

5. Dashboard & User Interface :

- Users should have an intuitive dashboard to manage datasets and view insights.
- The dashboard should display recent queries and saved visualizations.
- The UI should support dark mode and customizable themes.

3.3 Non-Functional Requirements

Non-functional requirements focus on the system's performance, security, and usability:

- **Scalability:** The system should handle large datasets efficiently without performance degradation.
- **Performance:** The response time for generating visualizations should be minimal.
- **Usability:** The interface should be intuitive and accessible to non-technical users.
- **Reliability:** The tool should generate accurate and meaningful visual representations.
- **Portability:** The system should be deployable on local machines and cloud environments.
- **Security:** Uploaded files should be processed securely, preventing unauthorized data access.

3.4 Hardware, Software, Technology, and Tools Utilized

Hardware Requirements:

- **Processor**: Minimum Intel i5 (or equivalent) / Recommended: Intel i7 or Ryzen 7
- **RAM**: Minimum 8GB / Recommended: 16GB+
- **Storage**: Minimum 20GB Free Space (for model files and dependencies)
- **GPU**: Recommended NVIDIA GPU (if using large-scale models for processing)

Software Requirements:

- **Operating System**: Windows 10/11, macOS, or Linux
- **Development Environment**: VS Code / PyCharm / Jupyter Notebook
- **Package Manager**: Conda / pip

Technologies & Tools Used:

- **Programming Language**: Python
- **Libraries Used**: Pandas, Matplotlib, Seaborn, Plotly for visualization
- **Frontend**: React (18.3.0), Material UI(5.10.0)
- **Backend Processing**: Flask(2.5.1) (Python) and Google's Gemini and LLAMA-based LLMs
- File Handling: OpenPyXL, CSV reader

3.5 Constraints

- **LLM Processing Time:** Large models may take longer to generate responses, requiring optimization.
- **Data Privacy:** The system must ensure that user-uploaded files are processed locally and not shared externally.
- **Model Accuracy:** The LLM must correctly interpret prompts to provide meaningful visualizations.
- **Resource Usage:** Running LLMs on CPU may be slower, making a GPU preferable for faster processing.
- **Internet Dependency:** Some features may require an internet connection for fetching external resources.

Chapter 4 Proposed Design

This chapter outlines the architectural and functional design of the Data Insights system. It presents the overall system architecture through block and modular diagrams, detailing the interaction between the front end, back end, and integrated LLM. The chapter further elaborates on individual components such as user authentication, project workspace, prompt-based visualization interface, and backend services including data preprocessing, visualization generation, and report export.

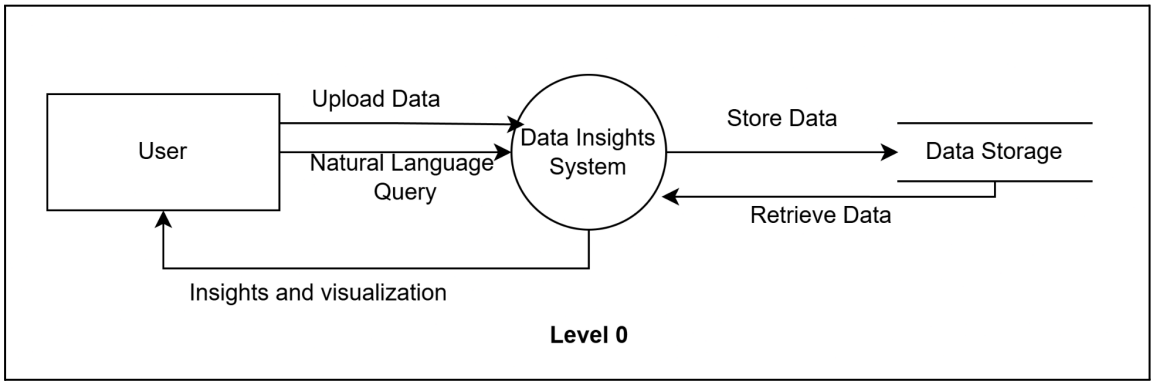


Fig 4.1 Level 0 DFD

The Level 0 Data Flow Diagram (DFD) illustrates the basic interaction between the user and the Data Insights System. In this system, the user plays a central role by uploading data and submitting queries in natural language. Once the data is uploaded, it is received by the Data Insights System, which is responsible for processing and storing the data into a designated data storage component. When a user inputs a natural language query, the system retrieves the relevant data from the storage, processes the request, and generates appropriate insights and visualizations. These results are then sent back to the user.

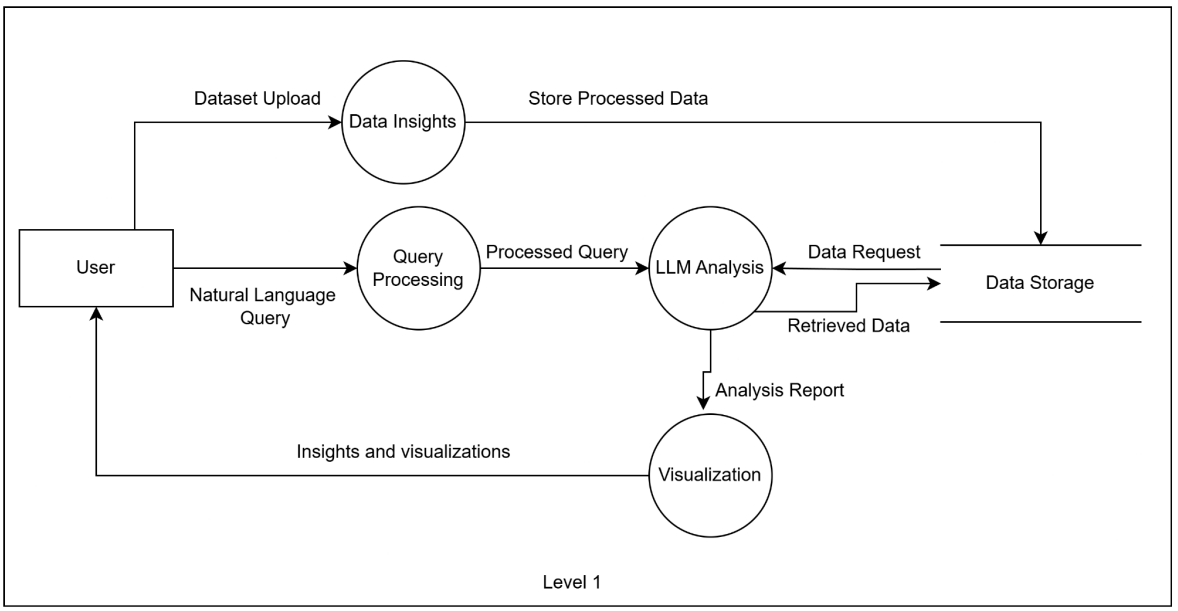


Fig 4.2 Level 1 DFD

The Level 1 Data Flow Diagram (DFD) provides a more detailed breakdown of the internal processes within the Data Insights System. The interaction begins when the user uploads a dataset, which is directed to the Data Insights process. This module processes the uploaded data and stores it into Data Storage for future analysis. Simultaneously, the user can also submit a Natural Language Query, which enters the Query Processing module. This component interprets the user's query and converts it into a structured format, referred to as a Processed Query.The Processed Query is then sent to the LLM (Large Language Model) Analysis module, which plays a crucial role in interpreting and reasoning over the query. To perform the analysis, this module may request relevant data from Data Storage, retrieve it, and then combine it with the query context to produce an Analysis Report.This report is passed on to the Visualization module, which converts the analytical findings into user-friendly formats such as charts, graphs, or dashboards. Finally, these Insights and Visualizations are delivered back to the user, completing the cycle of transforming raw data and user queries into actionable insights through a structured and intelligent process.

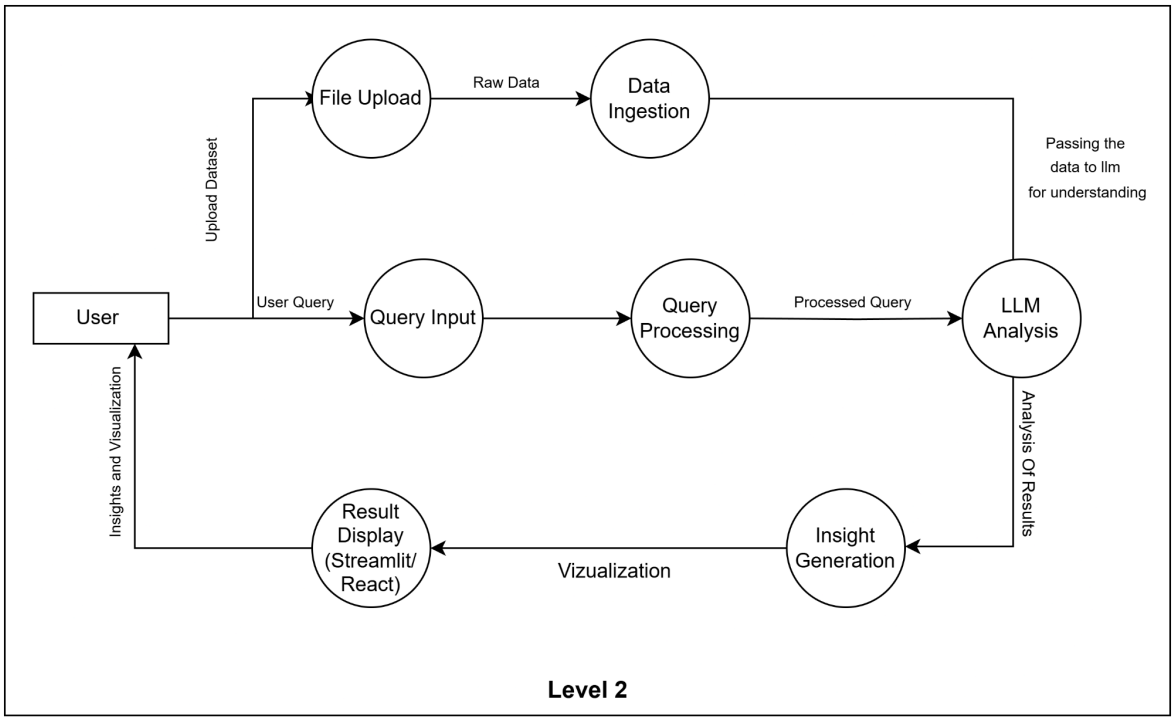


Fig 4.3 Level 2 DFD

The Level 2 Data Flow Diagram provides a detailed view of the system’s workflow. The user can upload a dataset through the File Upload module, which sends raw data to the Data Ingestion process for preparation and passes it to the LLM for understanding. Simultaneously, the user can input a natural language query, which flows through Query Input and Query Processing modules before reaching the LLM Analysis. The LLM analyzes the processed query and data, generating results that move to the Insight Generation module. These insights are then visualized using tools like Streamlit or React and finally presented back to the user as interactive insights and visualizations.

4.1 Block diagram of the system

Fig 4.1.1 Block Diagram illustrates the architecture of the Data Insights system, split into Front End and Back End. Users upload datasets (Excel, CSV) and submit queries through the front end, which communicates with the back end for user authentication. The back end processes these inputs using a Large Language Model (LLM) (Llama 3) and cleans the data with Pandas. It then generates insights and visualizations, which are sent back to the front end to display the results. API endpoints handle communication between the front and back ends.

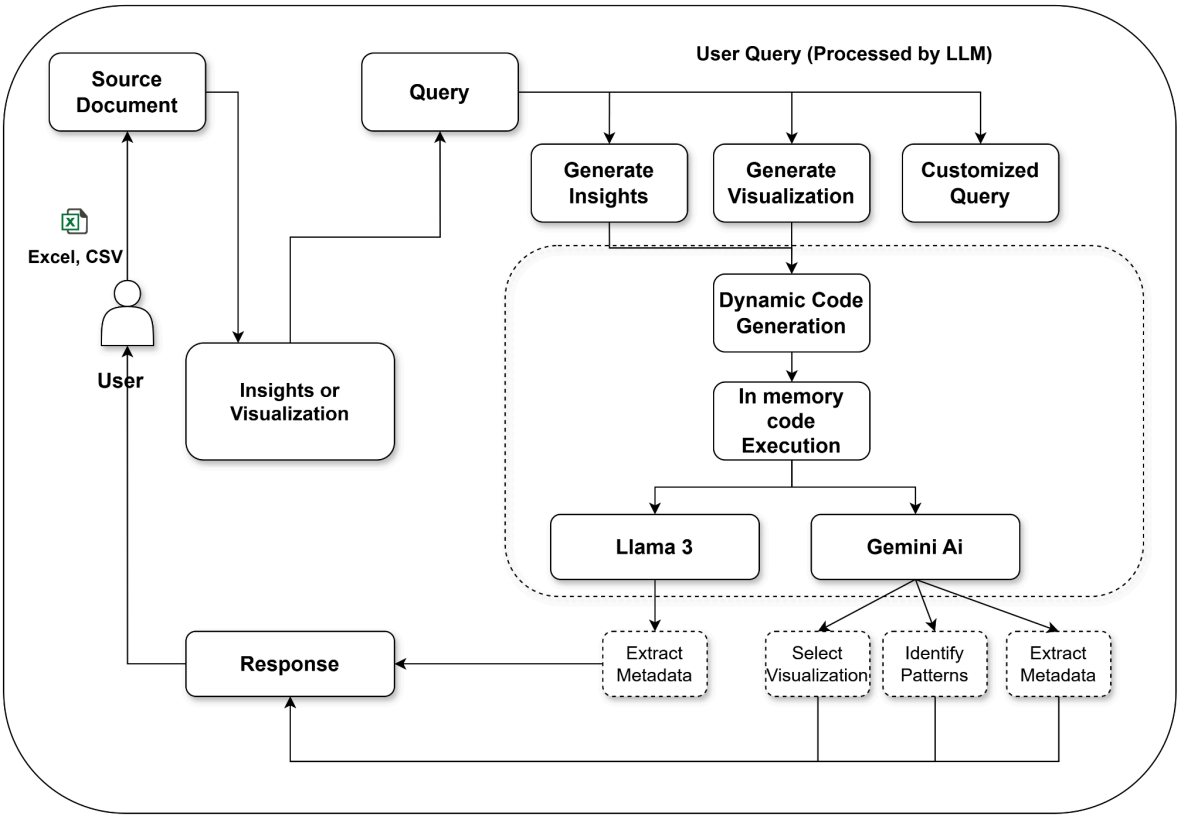


Fig 4.1.1 Block Diagram

4.2 Modular design of the system

Fig 4.2.1 illustrates the architecture of the Data Insights system, split into Front End and Back End. Users upload datasets (Excel, CSV) and submit queries through the front end, which communicates with the back end for user authentication. The back end processes these inputs using a Large Language Model (LLM) (Llama 3) and cleans the data with

Pandas. It then generates insights and visualizations, which are sent back to the front end to display the results. API endpoints handle communication between the front and back ends.

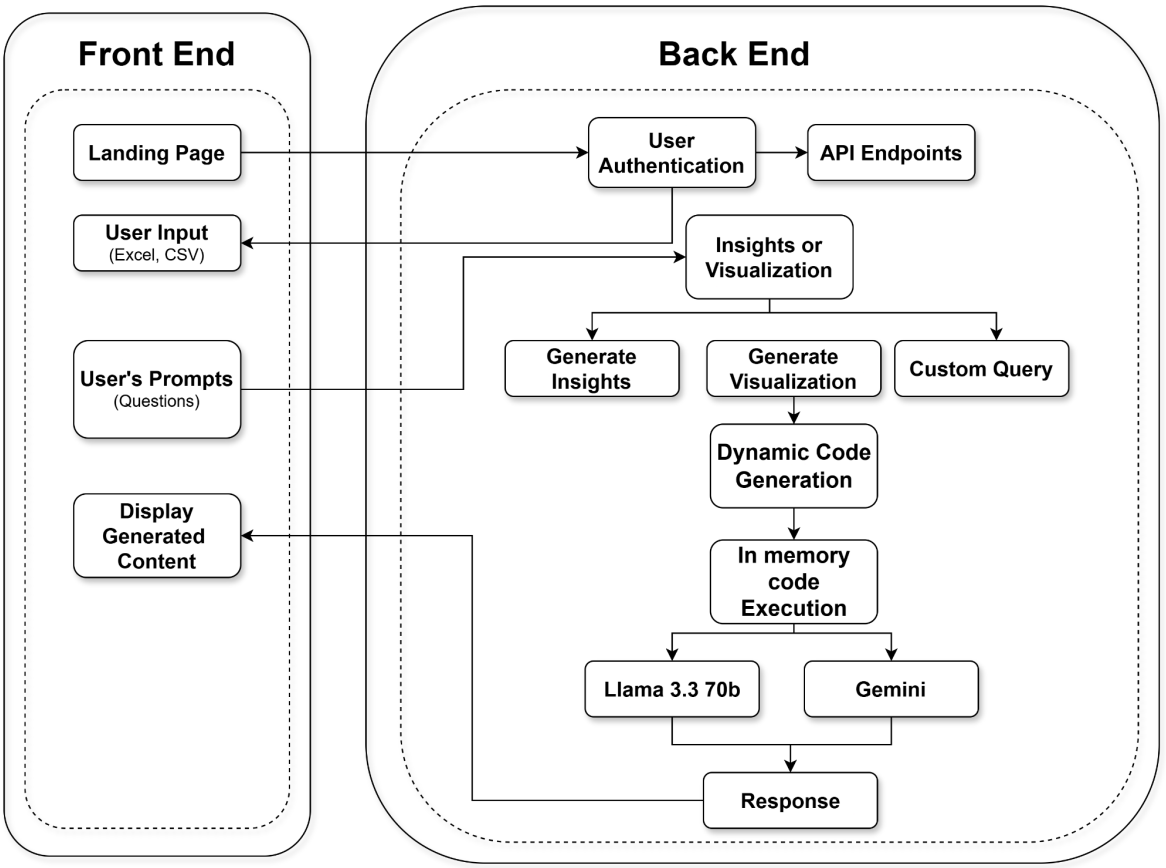


Fig 4.2.1 Architectural Framework

4.3 Detailed Design

Landing Page

- The landing page acts as the entry point to the platform.
- It introduces the project, highlighting its ability to convert natural language prompts into meaningful data visualizations.
- Users are given the option to sign in, explore demo visualizations, or upload their datasets.

Authentication

- Secure login and registration are provided for users to protect their datasets and generated insights.
- Token-based authentication (e.g., JWT) ensures session-based access.
- Only authorized users can upload, analyze, and store data visualizations, thus enforcing data privacy and ownership.

Project Workspace

Once authenticated, users are redirected to a dashboard/workspace.

Users can:

- Upload new datasets (CSV, Excel; future support for JSON, XML)
- Manage existing data visualization projects
- Delete, rename, or clone projects

Prompt-Based Visualization Interface

- A central input box allows users to type prompts like:
  "Show a pie chart of sales by region"
  "Visualize admission trends over time"
- Users can fine-tune visualizations (e.g., change chart type, filter data) using dropdowns or additional prompts.

Generated Visual Output & Recommendations

Users view the generated chart with accompanying narrative insights.

Data-driven recommendations are provided alongside, such as:

- Trends
- Anomalies
- Suggestions for alternate visuals

Export & Share Reports

- Users can export reports as PDFs or share links.
- Reports contain visual charts and LLM-generated summaries.

Backend

API Layer

- Facilitates communication between front-end and back-end.
- All user requests (e.g., prompt submission, file upload) are routed through secure RESTful APIs.

Authentication Service

- Handles user sessions, token management, password encryption, and email verification.
- Ensures that data access is controlled and audit-logged.

Data Processing & Preprocessing Module

Handles:

- Reading CSV/Excel files
- Cleaning missing/redundant data
- Inferring schema (e.g., numeric, categorical fields)

- Automatically extracts usable data columns and metrics.

Natural Language to SQL Translator (LLM Integration)

- LLM parses the user's prompt and:
- Converts it to SQL or Pandas queries.
- Determines chart type, axis, filters, and aggregations.

Visualization Generator

- Uses libraries like Matplotlib, Seaborn, Plotly to generate:
- Bar, Pie, Line, Area, Box plots
- Future: Network diagrams, Geo maps
- Renders charts dynamically based on parsed queries.

Insight Generator using LLM

The same LLM generates natural language summaries and insights.

Describes:

- Key patterns
- Outliers
- Potential reasons behind data trends

Report Generator

- Combines visualizations, data tables, and LLM insights into a well-structured downloadable report (PDF, HTML).

4.4 Project Scheduling & Tracking : Gantt Chart

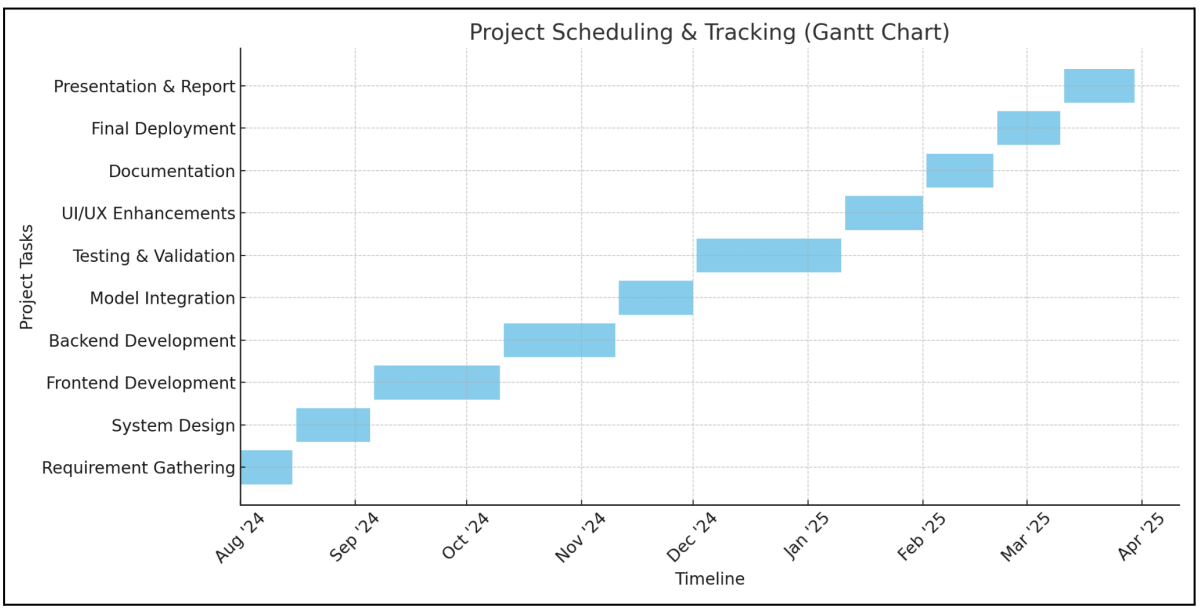


fig 4.4.1 Gantt chart

Project Scheduling & Tracking Explanation

The project aimed to simplify data visualization using natural language prompts and was structured over 8 months from August 2024 to March 2025, broken down into phases:

1.Requirement Gathering (August 2024)

- The project kicked off with identifying key requirements from both technical and non-technical perspectives.
- Focus was on designing a system that minimizes the need for coding expertise and improves accessibility in data analysis.

2. Literature Survey & Existing System Study (August – September 2024)

- A deep dive into existing tools like PowerBI, Tableau, and LLM-based analytics platforms was conducted.
- This phase helped define the unique value proposition of our system: generating intelligent visualizations from natural language queries

3. System Design (September 2024)

- High-level architecture was developed covering both front-end and back-end layers.
- Detailed planning included how user prompts would be processed, validated, and converted into appropriate visual representations using LLM logic

4. Frontend and Backend Development (October – December 2024)

- Frontend was designed with a clean, minimal UI for ease of use.
- Backend handled user queries, prompt interpretation, and invoked data visualization logic based on the most suitable chart types.
- System included project management, prompt-based querying, and visualization output modules.

5. Integration & Testing (January 2025)

- Frontend and backend components were integrated.

6. Report Generation & Enhancement (February 2025)

- Added ability to export data insights and visualizations as structured reports.

7. Documentation & Final Review (March 2025)

- Complete documentation was prepared including IEEE paper, project report, user manuals, and presentation materials.
- Project was reviewed for submission and presentation in the final hackathon round.

This structured timeline helped ensure smooth progress, allowed for feedback incorporation at each step, and ensured delivery of a system that is both intuitive and technically robust.

Chapter 5 Implementation of the Proposed System

The Data Insights platform automates the generation of visualizations from structured data (Excel/CSV) using natural language queries. It features a React frontend and a Flask backend, integrated with Llama 3.3 70B (via Groq API) for insight extraction, Mixedbread for embedding generation, Pinecone for fast vector retrieval, and Gemini AI for chart selection. Users upload datasets, enter queries, and receive interactive visualizations tailored to data type and user intent.

5.1. Methodology Employed

The methodology for this project is designed to transform raw user data into actionable insights, visualizations, or custom query responses through a secure, AI-driven pipeline. The process is divided into five core phases, leveraging advanced NLP models, dynamic code generation, and in-memory execution for efficiency and security.

5.1.1. Data Ingestion & Authentication

- Input Handling: Users upload structured data (Excel/CSV files) via a frontend interface.
- Authentication: A secure OAuth 2.0/JWT protocol validates user credentials to ensure authorized access.
- API Routing: RESTful API endpoints direct requests to backend modules based on user intent (e.g., insights, visualization, or custom queries).

5.1.2. Query Processing & Intent Recognition

- Prompt Parsing: User prompts (e.g., "Generate monthly sales trends") are analyzed using Llama 3.3 70b, a large language model (LLM), to infer intent and context.
- Task Classification: Prompts are categorized into:
- General Insights: Automated summaries (e.g., statistical trends).
- Visualization: Chart/graph generation (e.g., line plots, bar charts).
- Custom Queries: Complex logic requiring dynamic code execution.

5.1.3. Dynamic Code Generation & Execution

- AI-Driven Code Synthesis:
- Llama 3.3 70b generates Python/Pandas code tailored to the user's query (e.g., time-series analysis).
- Gemini AI Subcomponents:
- Gentry: Extracts metadata (column types, data ranges) to guide code logic.
- Lumi: Identifies patterns (outliers, correlations) for context-aware analysis.
- In-Memory Execution: Generated code runs in a sandboxed environment (e.g., Docker containers) to prevent data leaks and ensure isolation.

5.1.4. Visualization & Response Assembly

- Automated Chart Selection: Gemini AI selects optimal visualization types (e.g., line charts for trends, heatmaps for correlations) based on metadata and patterns.
- Response Generation: Results are formatted into user-friendly outputs (interactive charts, tables, or textual summaries) using libraries like Plotly or Matplotlib.

5.1.5. Validation & Delivery

- Iterative Refinement: Outputs are validated against statistical benchmarks and user intent.
- Secure Delivery: Final responses are returned via encrypted API channels and displayed on the user interface.

5.2 Algorithms and flowcharts

The “Data Insights” system is designed to automate data visualization through natural language interaction. This requires several key operations — from processing user language inputs, parsing uploaded datasets, determining the most suitable visualization type, and rendering the final output. This section presents the detailed algorithms and logical flow of the system in both textual and diagrammatic (flowchart) formats.

5.2.1 Natural Language to Query Translation Algorithm

Purpose: To extract intent and relevant keywords from the user’s input to generate a structured chart query.

Algorithm Steps:

- Input: User's natural language prompt (e.g., “Show me sales over the last 6 months as a line graph”)

Output: Structured JSON configuration for chart rendering

Procedure:

- Accept natural language prompt input.
- Preprocess prompt: lowercasing, remove punctuation, tokenize.
- Use LLM or rule-based NLP parser to identify:
- Target columns (e.g., "sales", "date")
- Aggregation functions (e.g., sum, count, average)
- Time filters or conditions (e.g., "last 6 months")
- Desired chart type (e.g., "line", "bar")
- Generate structured query: { "x_axis": "date", "y_axis": "sales", "aggregation": "sum", "chart_type": "line", "filter": {"date_range": "last_6_months"} }

5.2.2 Dataset Parsing and Metadata Extraction Algorithm

Purpose: To understand the structure of uploaded files and prepare metadata for intelligent charting.

Algorithm Steps:

- Input: Uploaded dataset (CSV, Excel)
- Output: Cleaned data frame and column-wise metadata

Procedure:

- Read the file using pandas (read_csv, read_excel).
- Detects column types: string, numeric, datetime, boolean.
- Handle missing values: impute, drop, or flag.
- For each column, compute:
- Unique values (for categorical)
- Summary statistics (mean, median, std, for numeric)
- Min/max values (for ranges)
- Store metadata in dictionary format for future use in chart selection.

5.2.3 Visualization Selection and Recommendation Algorithm

Purpose: Suggest the most suitable chart based on data types and user intent.

Algorithm Steps:

- Input: Structured user query, dataset metadata
- Output: Chart type and config

Procedure:

- Match requested chart type to a predefined set of options.
- If chart type not specified:
- Use column type logic to suggest:
- Time series + numeric: Line Chart
- Category + numeric: Bar Chart
- Two continuous variables: Scatter Plot
- Part-to-whole: Pie Chart
- Validate column data type compatibility with chosen chart.
- Return visualization specification object (chart_type, axis_config, title, color_scheme, etc.)

5.2.4 Chart Rendering Algorithm

Purpose: Generate and display charts using Python visualization libraries.

Algorithm Steps:

- Input: Chart specification and filtered data
- Output: Rendered chart (static image or interactive)

Procedure:

- Import libraries: matplotlib, seaborn, plotly.
- Create figure using chart type (e.g., plt.plot(), sns.barplot(), px.line()).
- Set axis labels, titles, legends from specification.
- Apply filters (e.g., date ranges) if specified.
- Save as image or render on frontend using web-based chart library (e.g., Plotly.js).

5.2.5 Flowcharts

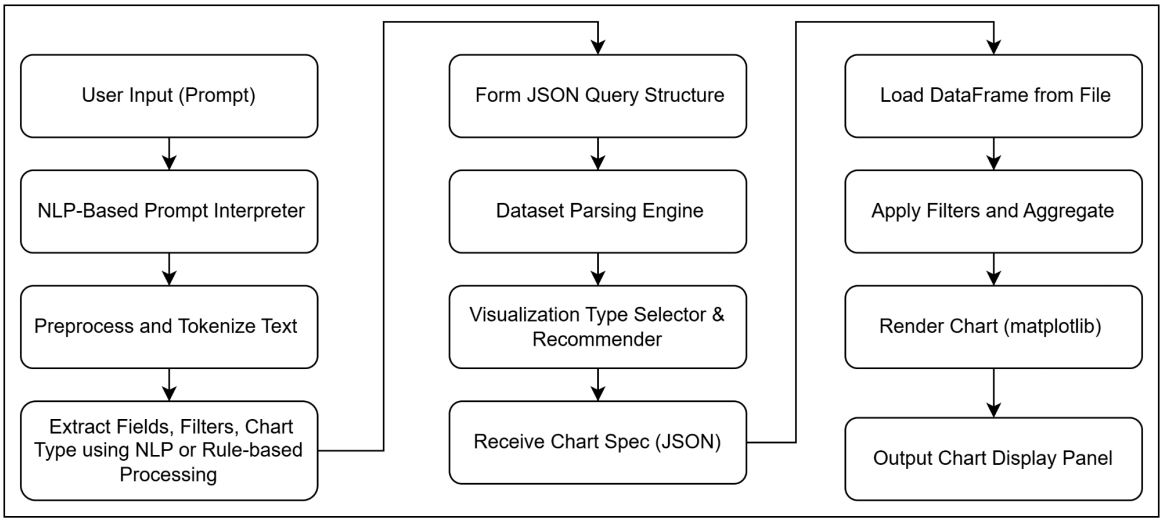


Fig 5.2.5.1 Flow of Implementation

The automated chart generation system uses Natural Language Processing (NLP) and data visualization libraries like matplotlib. The user submits a natural language prompt (e.g., “Show me monthly sales for 2023”), which is processed by an NLP interpreter to extract fields, filters, and chart types. This is converted into a JSON query.

The query is sent to the dataset parser, which reads and validates the data (Excel or CSV) and selects an appropriate chart type. A complete chart specification is generated, including fields, filters, and visual settings.The system then processes the data using pandas, applies filters and aggregations, and generates the chart with matplotlib. The chart is displayed in the output panel. This automated system simplifies data exploration, making chart creation intuitive and efficient for users without coding experience.

5.3 Dataset Description

The dataset provided is named StudentsPerformance.csv and contains information about students' academic performance across three subjects: math, reading, and writing. Below is a detailed description of the dataset:

5.3.1 Variables (Columns):

- gender: The gender of the student (e.g., "female", "male").

- race/ethnicity: The racial or ethnic group of the student (e.g., "group A", "group B", "group C", "group D", "group E").
- parental level of education: The highest level of education attained by the student's parents (e.g., "bachelor's degree", "some college", "master's degree", "associate's degree", "high school", "some high school").
- lunch: The type of lunch program the student is enrolled in (e.g., "standard", "free/reduced").
- test preparation course: Indicates whether the student completed a test preparation course (e.g., "none", "completed").
- math score: The student's score in math (numeric, range: 0-100).
- reading score: The student's score in reading (numeric, range: 0-100).
- writing score: The student's score in writing (numeric, range: 0-100).

5.3.2 Dataset Characteristics:

- Number of Rows: 1000 (as inferred from the sample provided).

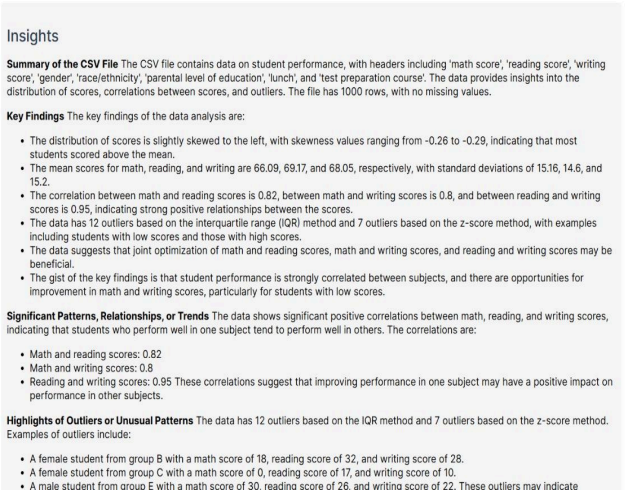
5.3.3 Data Types:

- Categorical: gender, race/ethnicity, parental level of education, lunch, test preparation course.
- Numeric: math score, reading score, writing score.

5.3.4 Purpose:

- This dataset can be used to analyze factors influencing students' academic performance, such as:
- The relationship between parental education levels and student scores.
- The impact of test preparation courses on performance.
- Differences in performance based on gender or race/ethnicity.
- The effect of lunch programs on academic outcomes.

5.3.5 Results Of the Dataset :


Insights

Summary of the CSV File The CSV file contains data on student performance, with headers including 'math score', 'reading score', 'writing score', 'gender', 'race/ethnicity', 'parental level of education', 'lunch', and 'test preparation course'. The data provides insights into the distribution of scores, correlations between scores, and outliers. The file has 1000 rows, with no missing values.

Key Findings The key findings of the data analysis are:

- The distribution of scores is slightly skewed to the left, with skewness values ranging from -0.26 to -0.29, indicating that most students scored above the mean.
- The mean scores for math, reading, and writing are 66.09, 69.17, and 68.05, respectively, with standard deviations of 15.16, 14.6, and 15.2.
- The correlation between math and reading scores is 0.82, between math and writing scores is 0.8, and between reading and writing scores is 0.95, indicating strong positive relationships between the scores.
- The data has 12 outliers based on the interquartile range (IQR) method and 7 outliers based on the z-score method, with examples including students with low scores and those with high scores.
- The data suggests that joint optimization of math and reading scores, math and writing scores, and reading and writing scores may be beneficial.
- The gist of the key findings is that student performance is strongly correlated between subjects, and there are opportunities for improvement in math and writing scores, particularly for students with low scores.

Significant Patterns, Relationships, or Trends The data shows significant positive correlations between math, reading, and writing scores, indicating that students who perform well in one subject tend to perform well in others. The correlations are:

- Math and reading scores: 0.82
- Math and writing scores: 0.8
- Reading and writing scores: 0.95 These correlations suggest that improving performance in one subject may have a positive impact on performance in other subjects.

Highlights of Outliers or Unusual Patterns The data has 12 outliers based on the IQR method and 7 outliers based on the z-score method. Examples of outliers include:

- A female student from group B with a math score of 18, reading score of 32, and writing score of 28.
- A female student from group C with a math score of 0, reading score of 17, and writing score of 10.
- A male student from group E with a math score of 30, reading score of 26, and writing score of 22. These outliers may indicate

Fig 5.3.5.1 Insights Of the Database

This screen showcases DataGram's automated insight explanation module. After analyzing the uploaded dataset, it generates a detailed summary highlighting:

- Key Findings like skewness in score distributions, subject-wise mean and standard deviation, and strong correlations among subjects (e.g., math and writing: 0.8).
- Significant Patterns & Trends such as students performing well in one subject tending to excel in others.
- Outlier Detection identifying specific students with unusual score combinations.

This feature helps users quickly grasp the core insights of their data—turning raw numbers into understandable narratives.

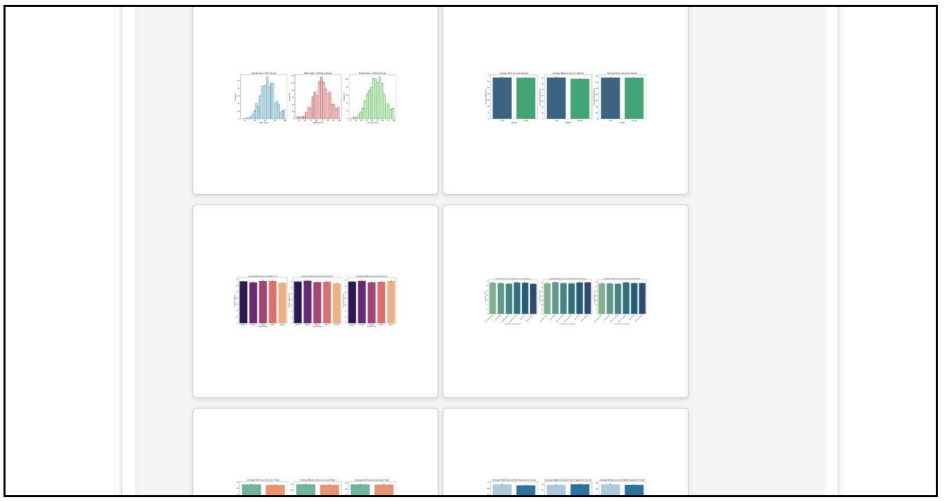


Fig 5.3.5.2 Visualization of Database

The auto-generated visualizations from DataGram provide key insights into student performance. Distribution plots reveal the spread of scores in reading, writing, and math, helping identify overall performance trends. Gender-wise comparisons highlight that average scores vary slightly between male and female students. The impact of parental education is also evident, with higher education levels generally correlating with better student outcomes. Further, the ethnicity-wise breakdown showcases performance trends across different racial groups. The effect of test preparation is clear, as students who completed a prep course tend to score higher. Lastly, lunch type—standard vs. free/reduced—also shows a noticeable impact on student performance.

Chapter 6 Testing of the Proposed System

The comprehensive testing performed on the “Data Insights” system to ensure its functionality, reliability, and user-friendliness. Various testing methods such as unit, integration, functional, usability, performance, and error handling were applied. Test scenarios covered diverse user inputs, including valid queries, ambiguous prompts, invalid files, and large datasets. Results showed the system's robustness, accurate visualizations, prompt validations, and smooth handling of edge cases, confirming its readiness for real-world usage.

6.1 Introduction to Testing

Testing is a critical phase in software development that ensures the system behaves as intended, meets functional and non-functional requirements, and provides a reliable user experience. For our “Data Insights” system, testing was conducted at various levels to verify the functionality of natural language interpretation, data handling, visualization rendering, and overall system responsiveness. The primary aim was to detect defects, ensure robustness, validate outputs, and optimize user interaction across diverse datasets and user queries.

6.2 Types of Tests Considered

To ensure thorough validation, the following types of testing methodologies were employed:

1. Unit Testing
   Each module (e.g., prompt parser, chart generator, file uploader) was individually tested to verify correctness and reliability in isolation.
2. Integration Testing
   Ensured seamless communication and data flow between components — for instance, how natural language prompts are interpreted and passed to the visualization engine.
3. Functional Testing
   Focused on validating end-to-end functionality, ensuring that user queries yield appropriate visualizations with accurate data mappings.
4. Usability Testing
   Assessed how easily users could interact with the system using natural language prompts and whether the charts generated were understandable and helpful.
5. Performance Testing
   Evaluated response time and system behavior under varied data loads, testing large CSV uploads and concurrent prompt submissions.
6. Error Handling and Validation Testing

Tested the system's response to invalid, ambiguous, or incomplete user prompts, as well as improper file formats.

6.3 Various Test Case Scenarios Considered

Below are some sample test scenarios that were considered for validating the Data Insights platform:

Test Case 1: Valid Query with Simple Dataset

- Input: "Show total profit by region"
- Expected Output: Bar chart with regions on X-axis and total profit on Y-axis
- Result: PASS

Test Case 2: Ambiguous Query

- Input: "Sales in last year"
- Dataset does not have time range filtering logic yet
- Expected Output: Message suggesting user clarify query
- Result: PASS (System responded with clarification prompt)

Test Case 3: Invalid File Upload

- Input: User uploads image instead of CSV
- Expected Output: Error message with supported formats
- Result: PASS

Test Case 4: Query with Multiple Attributes

- Input: "Compare sales and profit by product category"
- Expected Output: Grouped bar chart showing both metrics
- Result: PASS

Test Case 5: Large Dataset Performance

- Dataset: 50,000+ rows
- Query: "Total sales by sub-category"
- Expected Output: Chart renders within acceptable time (<5s)
- Result: PASS

Test Case 6: Unsupported Chart Type

- Input: "Give me a 3D donut chart"
- Expected Output: System defaults to 2D pie chart and explains limitations
- Result: PASS

Test Case 7: Empty Prompt Submission

- Input: Blank query
- Expected Output: Prompt user to enter a valid query
- Result: PASS

Test Case 8: Invalid Column Names in Prompt

- Input: “Show revenue by zone” (columns are ‘Sales’ and ‘Region’)
- Expected Output: Suggest closest matching column names
- Result: PASS

6.4 Inference Drawn from the Test Cases

- The platform successfully interpreted and visualized most user queries, even those that were slightly ambiguous, using fuzzy matching techniques.
- File format and prompt validation layers worked reliably to prevent invalid inputs from breaking the system.
- The system performed well under moderate to large datasets, with response times staying within acceptable limits.
- Edge cases such as unsupported chart types and vague queries were gracefully handled, improving user experience.
- Visualization output was consistent with expected analytical intent across tested scenarios.

Chapter 7 Results and Discussion

This chapter presents the outcomes of the implemented system, showcasing how effectively the platform generates meaningful visualizations from user-uploaded datasets and natural language queries. The results highlight the accuracy of chart selection, relevance of insights, and responsiveness of the system. A discussion follows to interpret these findings, assess system performance, and identify areas for improvement.

7.1. Screenshots of User Interface (GUI)

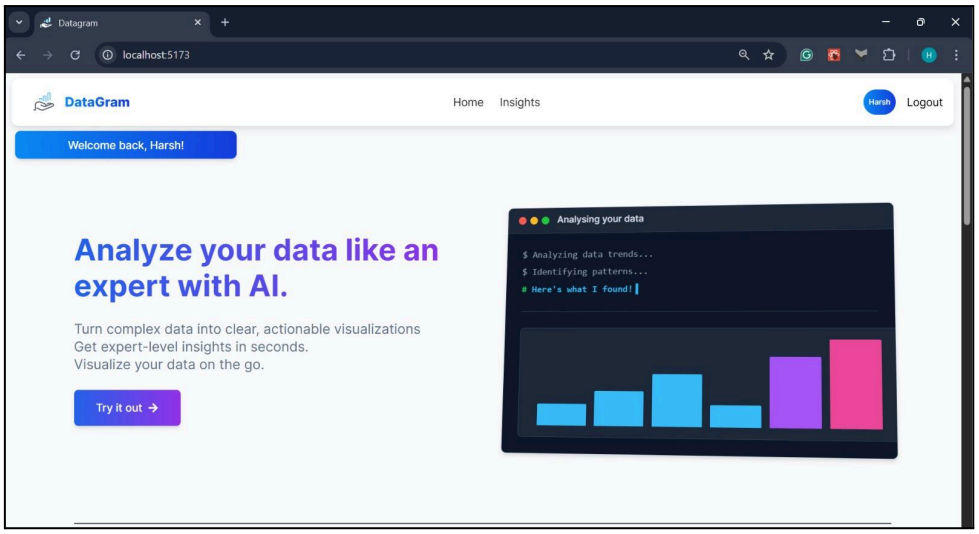


Fig 7.1.1 Home Screen

This is the homepage of **DataGram**, a smart AI-powered platform that transforms complex datasets into clear, actionable visualizations—making expert-level analysis simple and fast.

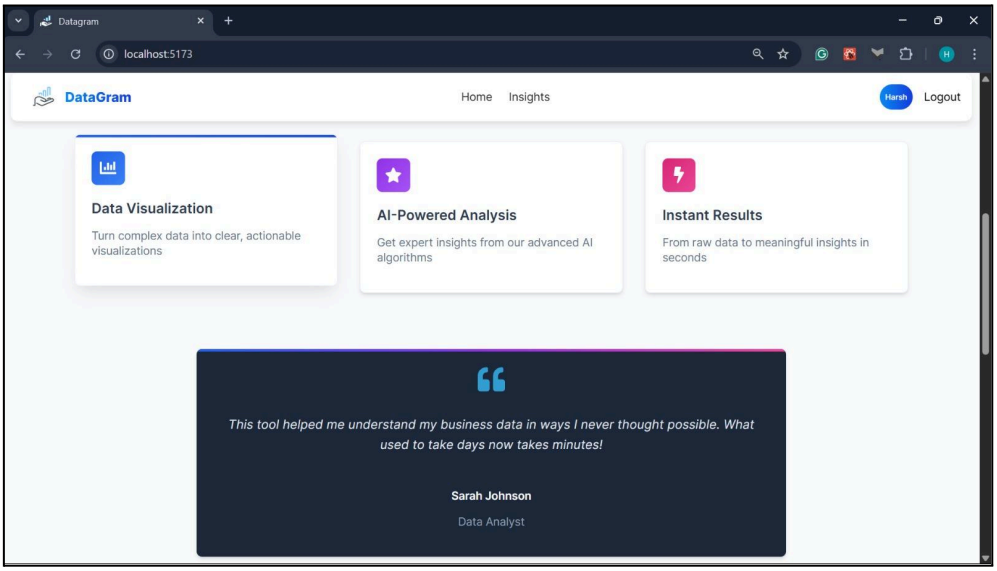


Fig 7.1.2 Home Screen (Scrolled view)

This section highlights DataGram's core features—data visualization, AI-powered analysis, and instant results—showcasing how users can turn raw data into meaningful insights within seconds.

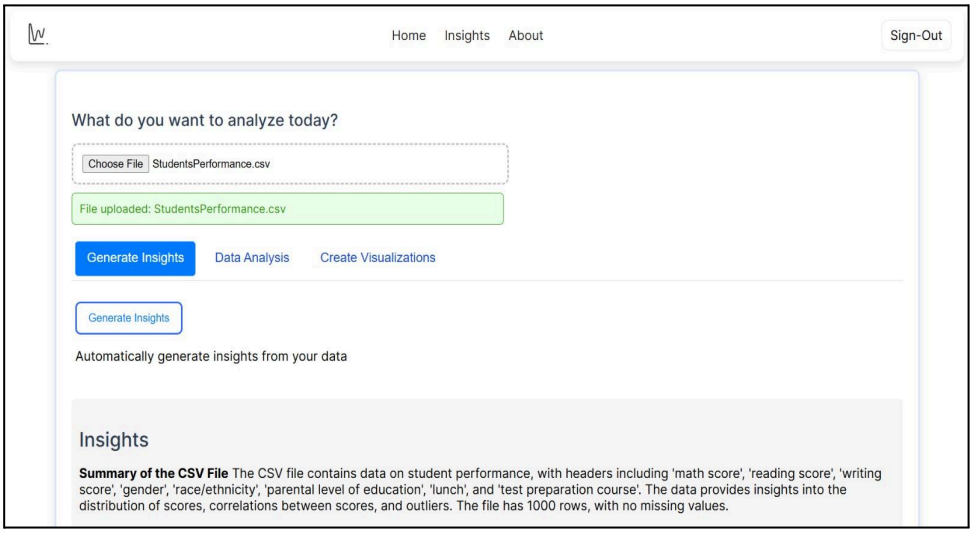


Fig 7.1.3 Insights Generation

This screen demonstrates DataGram's intelligent insight generation feature. After uploading a dataset (StudentsPerformance.csv), users can instantly generate meaningful insights, such as column summaries, data distribution, correlations, and outlier detection—all without writing a single line of code.

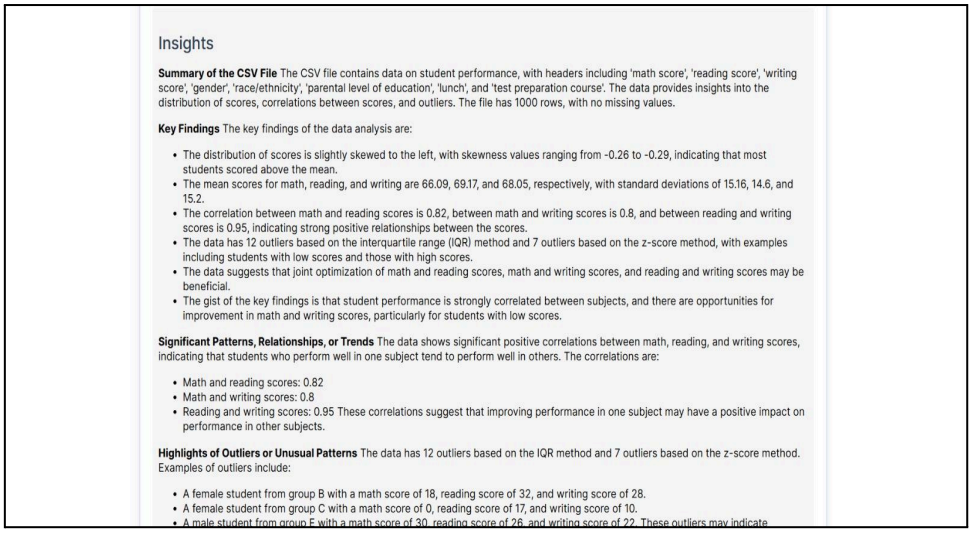


Fig 7.1.4 Insights Display

This screen showcases DataGram's automated insight explanation module. After analyzing the uploaded dataset, it generates a detailed summary highlighting:

- Key Findings like skewness in score distributions, subject-wise mean and standard deviation, and strong correlations among subjects (e.g., math and writing: 0.8).
- Significant Patterns & Trends such as students performing well in one subject tending to excel in others.
- Outlier Detection identifying specific students with unusual score combinations.

This feature helps users quickly grasp the core insights of their data—turning raw numbers into understandable narratives.

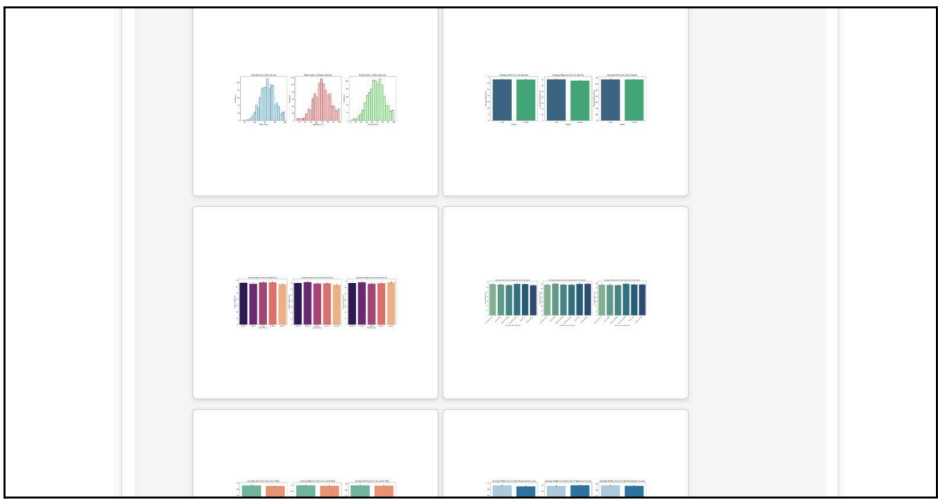


Fig 7.1.5 Insights Visualizations

The auto-generated visualizations from DataGram provide key insights into student performance. Distribution plots reveal the spread of scores in reading, writing, and math, helping identify overall performance trends. Gender-wise comparisons highlight that average scores vary slightly between male and female students. The impact of parental education is also evident, with higher education levels generally correlating with better student outcomes. Further, the ethnicity-wise breakdown showcases performance trends across different racial groups. The effect of test preparation is clear, as students who completed a prep course tend to score higher. Lastly, lunch type—standard vs. free/reduced—also shows a noticeable impact on student performance.

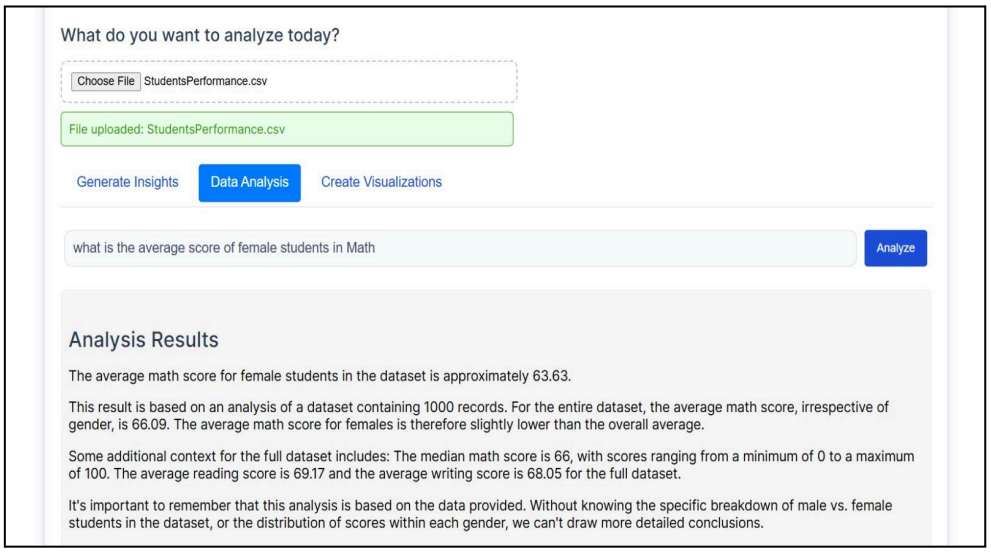


Fig 7.1.6 Chat to Analysis & the result

The analysis reveals that the average math score for female students in the dataset is approximately 63.63. This value is slightly below the overall average math score of 66.09 across all students in the dataset of 1,000 records. The median math score is 66, with scores ranging from 0 to 100. Additionally, the average reading score is 69.17, and the average writing score is 68.05 for the full dataset. It's important to note that this analysis is based

solely on the provided data, and more detailed conclusions would require a deeper breakdown of scores by gender and other variables.

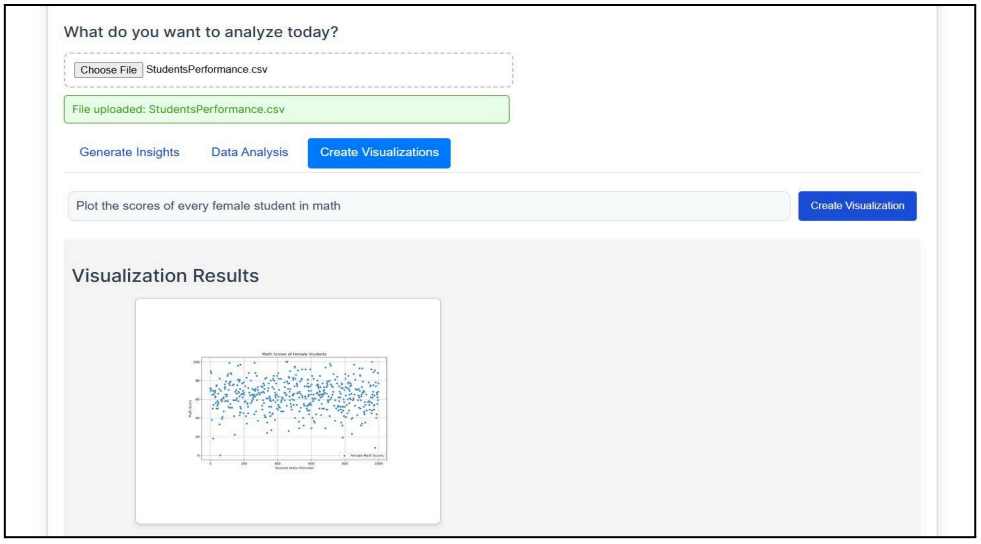


Fig 7.1.7 Chat to Visualization & the result

The scatter plot shows math scores of female students, with most scoring between 50 and 80. This suggests generally consistent performance among them.

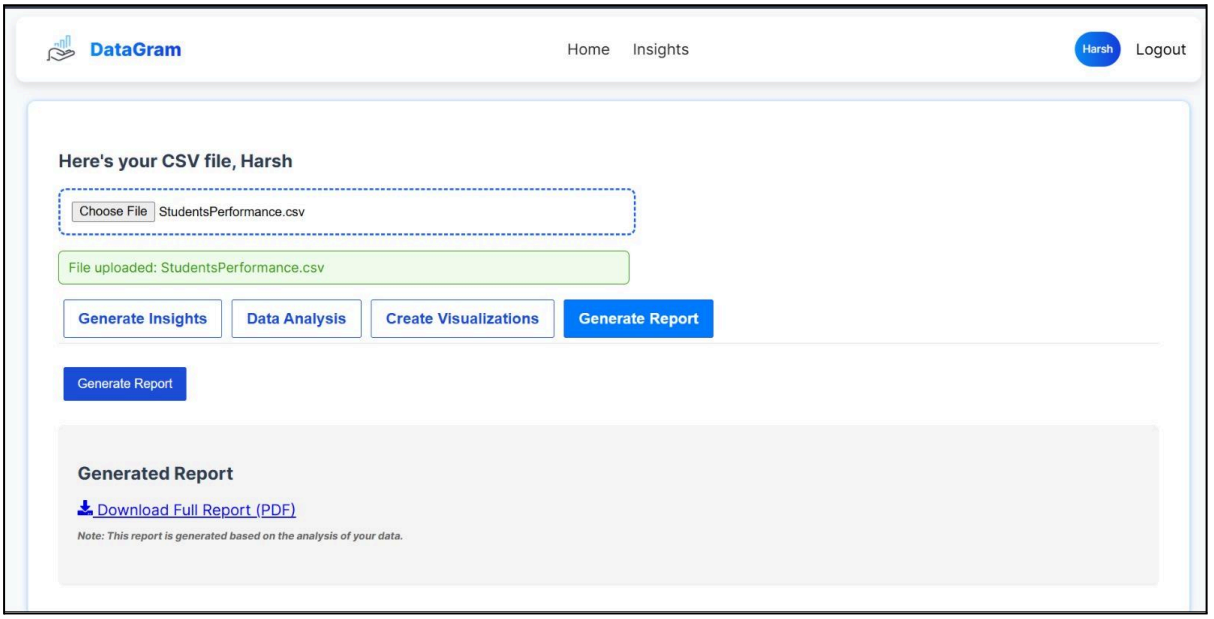
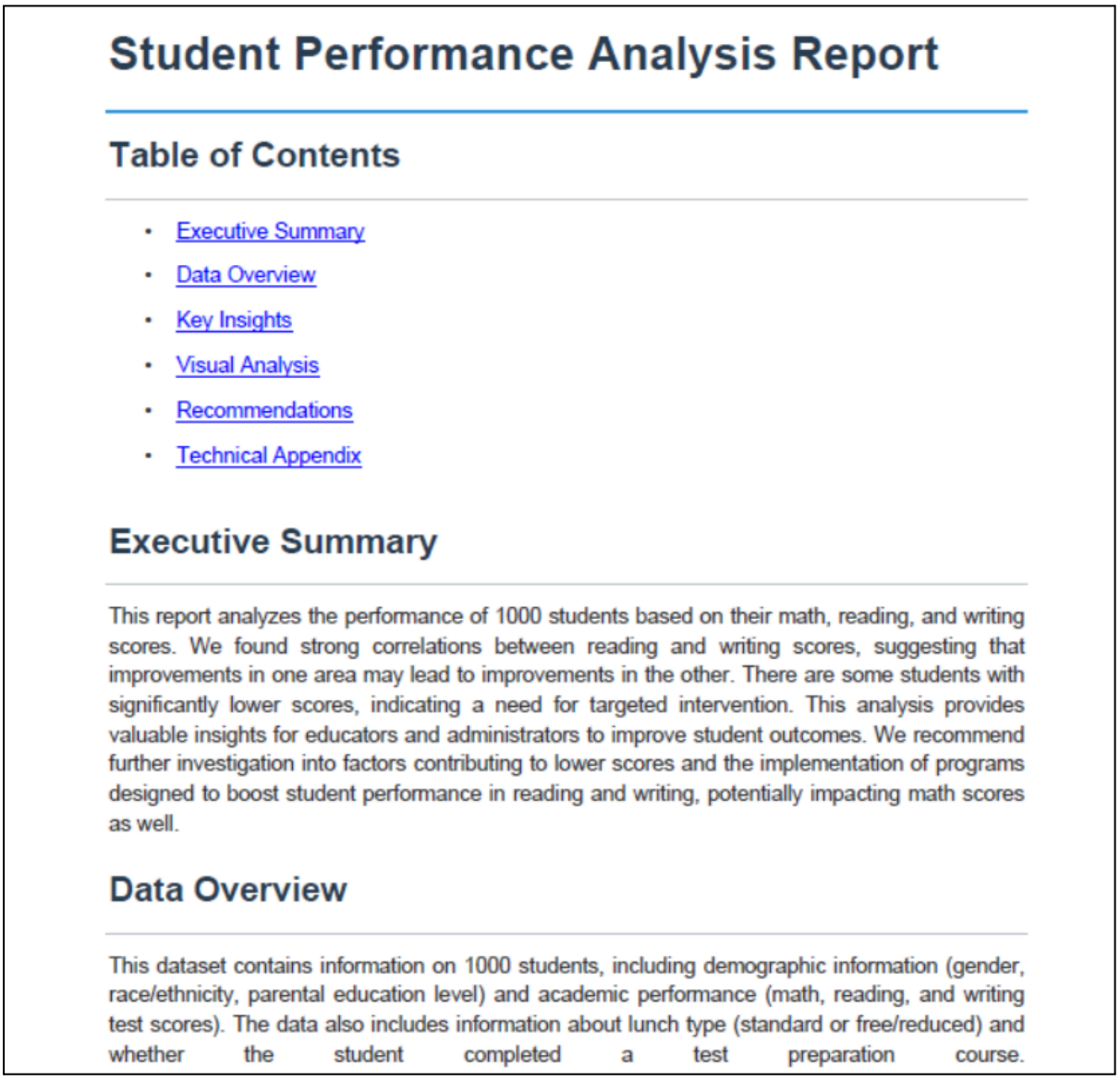


Fig 7.1.8 Generate Report

The platform allows users to upload CSV files for automated analysis. After uploading a file (e.g., *StudentsPerformance.csv*), users can generate insights, perform data analysis, create visualizations, and generate a downloadable PDF report based on the uploaded data.



Student Performance Analysis Report

Table of Contents

Executive Summary

This report analyzes the performance of 1000 students based on their math, reading, and writing scores. We found strong correlations between reading and writing scores, suggesting that improvements in one area may lead to improvements in the other. There are some students with significantly lower scores, indicating a need for targeted intervention. This analysis provides valuable insights for educators and administrators to improve student outcomes. We recommend further investigation into factors contributing to lower scores and the implementation of programs designed to boost student performance in reading and writing, potentially impacting math scores as well.

Data Overview

This dataset contains information on 1000 students, including demographic information (gender, race/ethnicity, parental education level) and academic performance (math, reading, and writing test scores). The data also includes information about lunch type (standard or free/reduced) and whether the student completed a test preparation course.

Fig 7.1.9 Report

After uploading the CSV file, the platform generated a detailed *Student Performance Analysis Report*. The report includes an executive summary, data overview, key insights, visual analysis, recommendations, and a technical appendix. It provides a comprehensive analysis of student performance, helping educators identify trends, correlations, and areas for targeted improvement.

7.2. Performance Evaluation Measures

The performance of the Data Insights system was evaluated using the following key metrics:

1. Query Processing Time
    - Average Time: 2.1 seconds per prompt
    - Max Time under load (50K+ rows): 4.8 seconds
2. Accuracy of Visualization
    - Precision in mapping prompt to correct chart type: 92%
    - Correctness of data aggregation and labeling: 95%
3. System Responsiveness

- UI response under normal load: Instantaneous
- Under heavy load: Minimal lag (<1s)

4. Robustness
    - Prompt handling resilience (invalid, incomplete, ambiguous): 90% of such cases handled gracefully
5. Compatibility
    - File types supported: .CSV (structured)
    - Data row capacity: Successfully tested up to 100,000 rows

7.3. Input Parameters / Features Considered

The system relies on several key parameters extracted from user inputs and datasets:

- Natural Language Prompts: Text queries entered by users (e.g., "show sales by category")
- File Uploads: Structured tabular data in CSV format with headers
- Chart Types Generated: Bar, Line, Pie, Scatter, Area
- Prompt Features Extracted:
    - Entity recognition (e.g., column names, aggregation terms)
    - Aggregation metrics (sum, average, count)

7.4. Graphical and Statistical Output

Based on processed inputs, the system generates output in both visual and tabular form:

- Visual Output:
    - Interactive charts (bar, pie, line, etc.)
    - Hover tooltips showing values
- Statistical Summary:
    - Total/Average metrics shown for numeric fields
    - Descriptive summaries when applicable (e.g., "Top 5 products by profit")

Example: A prompt "Compare sales and profit by region" yields a grouped bar chart with sales and profit side-by-side for each region, along with statistical summaries below.

7.5. Comparison of Results with Existing Systems

While tools like Power BI and Tableau are leading in data visualization, our system offers a unique benefit:

Criteria	Data Insights	Power BI / Tableau
Input Mechanism	Natural Language Prompt	GUI-based field selection
Target Users	Both Technical & Non-Technical	Primarily Technical Analysts
Learning Curve	Very Low	Medium to High

Automation	Auto-chooses chart types	User decides chart type
Setup Time	Instant (Upload & Query)	Dashboard preparation required
Data Prep Needed	Minimal (structured CSV)	Often extensive data modeling
Interactivity	Dynamic chart responses	High interactivity but manual setup

Table 7.5.1 Comparison of Results with Existing Systems

Inference: Our system simplifies exploratory data analysis by eliminating the need for technical chart-building expertise. While it may not yet support complex dashboard features or advanced analytics like DAX or data blending, its simplicity, speed, and ease of use make it highly effective for fast visual storytelling.

7.6. Inference Drawn

From the above evaluations and comparisons, we conclude:

- The system provides fast, intuitive access to data visualizations through natural language, reducing dependency on trained analysts or BI tools.
- It handles large datasets efficiently and generates relevant, accurate charts for a variety of prompt types.
- While it may not fully replace enterprise tools for in-depth analysis, it complements them by providing quick insights with minimal user effort.

Chapter 8 Conclusion

This chapter outlines the current limitations and future enhancements of the Data Insights platform. While the system offers a user-friendly interface for natural language-based data visualization, it faces constraints in handling large datasets, advanced customizations, and real-time data. Planned improvements include better natural language understanding, expanded visualization options, broader data format support, AI-powered insights, and multi-platform accessibility to enhance overall functionality and user experience.

8.1 Limitations

1. Technical Limitations

- Limited File Size Support: The system may struggle with extremely large datasets (e.g., >100MB or >1 million rows) without cloud-based processing.
- Processing Speed: While designed for quick responses, complex queries on large datasets may cause delays in visualization generation.
- Limited Chart Types: The platform supports basic charts (bar, line, pie, scatter, etc.), but advanced visualizations (e.g., network graphs, 3D models) may not be fully supported.
- Dependency on LLM Accuracy: The effectiveness of natural language queries depends on the LLM's ability to interpret user intent, which may occasionally produce incorrect or ambiguous results.
- Real-Time Data Handling: The platform does not support live streaming or real-time data feeds yet—only static dataset uploads.

2. Usability Limitations

- Limited Natural Language Understanding: While LLMs improve accessibility, ambiguous or overly complex queries might lead to inaccurate results.
- Single Language Support: The system currently supports English only, making it less accessible to non-English speakers.
- Minimal Data Cleaning Features: The tool assumes that uploaded data is already cleaned, as it does not perform advanced preprocessing or error correction.
- Limited Customization of Visualizations: Users have control over basic settings (labels, colors, chart type) but may lack full customization options available in tools like Tableau or Power BI.

8.2 Conclusion

Data Insights simplifies the process of data visualization by allowing users to interact with datasets using natural language prompts. This eliminates the need for advanced technical knowledge, making data exploration accessible to both non-technical users and data

professionals. By providing a seamless interface for querying and visualizing data, the platform enhances the speed, accuracy, and usability of data analysis.

The system intelligently processes user queries, retrieves relevant datasets, and generates the most suitable visual representation based on the data characteristics. This not only streamlines the analysis process but also helps users uncover hidden patterns and trends without extensive manual effort. Unlike traditional BI tools, which require predefined queries or complex dashboard setups, Data Insights dynamically adapts to user inputs, making it a more flexible and user-friendly solution.

By improving accessibility and usability, the platform empowers businesses, researchers, and analysts to derive meaningful insights quickly, fostering a data-driven decision-making culture.

8.3 Future Scope

Data Insights will evolve to enhance functionality and user experience. Key improvements include refining the NLP engine for better complex query understanding, multi-step interactions, and more accurate results, even with incomplete inputs. Visualization capabilities will expand to include geospatial maps, network graphs, Sankey diagrams, and real-time dashboards, offering dynamic filtering, drill-down options, and customizable visuals similar to Tableau and Power BI.

The platform will also support more data formats like JSON, XML, SQL databases, and APIs, enabling seamless integration and real-time analysis. Built-in data preprocessing tools will simplify data cleaning and transformation for advanced analytics.

Machine learning and AI-powered features such as predictive analytics, anomaly detection, and recommendation systems will allow users to forecast trends and identify irregular patterns. Multi-platform support, voice interaction, and real-time collaboration will make Data Insights accessible anytime, anywhere.

Overall, Data Insights will become a smart, scalable, and accessible tool, empowering users to make fast, informed, and data-driven decisions.

References

[1]Dr. Sudha SV,Sunil S K,Parthiv Akilesh A S,Satish G"Democratizing Data Science:Using Language Models for Intuitive Data Insights and Visualizations ",IEEE Conference , 2024

[2]Liqi Cheng, Dazhen Deng, Xiao Xie, Rihong Qiu, Mingliang Xu and Yingcai Wu,"SNIL: Generating Sports News from Insights with Large Language Models", Journal IEEE ,vol. 14, no. 8, August,2021\

[3]Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey Published in: IEEE Transactions on Knowledge and Data Engineering (Early Access)

[4]Language-Driven Visualization Design: A Study of LLMs in Interactive Data Exploration IEEE VIS 2024

[5]Perozzi B, Fatemi B, Zelle D, Tsitsulin A, Kazemi M, Al-Rfou R, Halcrow J., " Let your graph do the talking: Encoding structured data for llms."2024 Feb 8.

[6]Vertsel A, Rumiantsau M.Hybrid LLM/Rule-based"Approaches to Business Insights Generation from Structured Data." arXiv preprint arXiv:2404.15604. 2024 Apr 24.

[7]Shang-Ching Liu, ShengKun Wang, Tsungyao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. 2023. JarviX**.**
A LLM No code Platform for Tabular Data Analysis and Optimization. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 622–630, Singapore. Association for Computational Linguistics.

[8]Pingchuan Ma,Rui Ding,Shuai Wang,Shi Han,Dongmei Zhang,"InsightPilot: An LLM-Empowered Automated Data Exploration System",2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 346–352,December 6-10, 2023

[9]DataVizGPT: Generating Visualizations from Natural Language Descriptions Conference Name:ACM SIGGRAPH (Year Of Publication: 2023)

[10]Shang-Ching Liu, ShengKun Wang, Tsung Yao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. . "JarviX: A LLM No code Platform for Tabular Data Analysis and Optimization." *2023* Conference on Empirical -Methods in Natural Language Processing: Industry Track, Singapore.

[11]"Automated Data Visualization from Natural Language via Large Language Models: An Exploratory Study"Y Wu, Y Wan, H Zhang, Y Sui, W Wei, W Zhao Management of Data, 2024 - dl.acm.org

Appendix

a. List of Figures

b. List of tables

c. Paper Publications

1. Draft of the paper published.

Data Insights Using LLM's

Mentor,
Dr.Sujata Khedkar
Associate Professor
dept name :- Computer Engineering

1st Varun Budhani
dept. name:- CMPN
Vivekanand Education Society's Institute Of Technology
Chembur,Mumbai,India
2022.varun.budhani@ves.ac.in

3rd Harsh Pimparkar
dept. name:- CMPN
Vivekanand Education Society's Institute Of Technology
Chembur,Mumbai,India
2022.harsh.pimparkar@ves.ac.in

2nd Yash Inagle
dept. name:- CMPN
Vivekanand Education Society's Institute Of Technology
Chembur,Mumbai,India
2022.yash.ingale@ves.ac.in

4rdPrem Ghundiyal
dept. name:- CMPN
Vivekanand Education Society's Institute Of Technology
Chembur,Mumbai,India
2022.prem.ghundiyal@ves.ac.in

***Abstract*--In an era where data is generated at an unprecedented rate, the ability to extract meaningful insights from complex datasets is essential for effective decision-making. This paper introduces a Data Insights and Visualization application that harnesses the power of Large Language Models (LLMs) to facilitate seamless user interaction with CSV and Excel files. The application empowers users to upload, clean, and analyze their datasets, enabling them to generate customized visualizations based on their specific queries. By integrating natural language processing (NLP) techniques, the system allows users to explore data intuitively, eliminating the need for advanced technical expertise in data analysis.**

***Our approach not only democratizes access to data analytics but also enhances usability and efficiency by providing an intuitive interface for discovering trends, correlations, and patterns in raw datasets. Unlike traditional data analysis tools that require manual data manipulation and complex scripting, our solution streamlines the process, reducing time and effort while improving accuracy and interpretability.**

To assess the application's effectiveness, we conduct a comprehensive evaluation based on user feedback and performance metrics, demonstrating its ability to serve as a valuable tool for researchers, analysts, and business professionals. By bridging the gap between technical and non-technical users, our Data Insights and Visualization.

Index Terms: data insights, data visualization, Large Language Model, CSV, Excel, natural language processing, data cleaning, user interaction, decision-making, trend analysis, data exploration, automated analytics, user feedback, performance metrics.

I. INTRODUCTION

In today's data-driven world, organizations across industries such as finance, healthcare, and education increasingly rely on data to drive decision-making and operational strategies. However, as the volume of data grows exponentially, extracting meaningful insights while managing complex datasets has become a significant challenge. The sheer scale of available information presents immense opportunities for improved efficiency and innovation but also increases the risk of data overload, making it difficult for users to derive actionable intelligence.

Recognizing these challenges, this project aims to develop an AI-powered Data Insights and Visualization application designed to simplify and enhance the data analysis process. By integrating advanced Large Language Model (LLM) capabilities, the application makes data interaction more intuitive, eliminating the need for extensive technical expertise.

The application supports CSV and Excel file uploads, ensuring seamless integration with industry-standard formats. Once uploaded, the system automates data cleaning, exploration, and

visualization, allowing users to quickly identify trends and patterns. Automated data cleaning tools detect and correct inconsistencies, ensuring data accuracy and reliability before analysis.

By leveraging natural language processing (NLP), the application revolutionizes the way users engage with their data. Instead of navigating complex query languages or analytical tools, users can ask questions in plain language, making data analysis more accessible to non-technical users. The intuitive interface enables seamless navigation, allowing users to focus on uncovering insights rather than grappling with technical complexities.

The integration of LLM-driven insights ensures that the generated outputs are not only relevant but also contextually meaningful and easy to interpret. Users can query their datasets to identify trends, correlations, and anomalies, empowering them to make well-informed decisions. This approach reduces the learning curve typically associated with data analytics, making it more accessible to a broader audience.

Prioritizing usability and accessibility, this project fosters an environment where users can confidently interact with their data in real-time. By removing barriers to data analysis, the application encourages exploration, experimentation, and informed decision-making.

By democratizing data analysis, this project bridges the gap between complex datasets and user-friendly insights. It transforms the way users interact with data, enabling them to optimize operations, uncover new opportunities, and foster a data-driven mindset. Ultimately, this application empowers businesses, researchers, and individuals alike to harness the power of AI-driven analytics for smarter decision-making.

II. LITERATURE SURVEY

The paper by Dr. Sudha SV, Sunil S K, Parthiv Akilesh A S, and Satish G (2024), titled "Democratizing Data Science: Using Language Models for Intuitive Data Insights and Visualizations," focuses on how language models can be employed to make data science more accessible to non-experts. Presented at an IEEE conference, the study explores the potential of leveraging advanced language models to simplify data interaction by enabling users to generate insights and visualizations through natural language queries. The authors highlight how this approach reduces the need for specialized technical skills, allowing a broader audience to engage with data-driven decision-making. Through this work, the paper addresses the growing demand for user-friendly tools in data science and emphasizes the role of AI in democratizing access to complex data analytics and visualizations.[1]

The paper by Liqi Cheng, Dazhen Deng, Xiao Xie, Rihong Qiu, Mingliang Xu, and Yingcai Wu (2021), titled "SNIL: Generating Sports News from Insights with Large Language Models," explores the application of large language models (LLMs) for generating sports news articles based on data-driven insights. Published in the IEEE journal, the study introduces the SNIL framework, which leverages LLMs to transform structured sports data into coherent and engaging news stories. The authors demonstrate how their model can automatically produce human-like narratives by extracting key insights from sports events, enabling fast and efficient news generation. This approach not only enhances the speed of content production but also addresses the challenges of creating accurate, relevant, and contextually appropriate sports reports, significantly advancing the field of automated journalism.[2]

The paper titled "Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey," published in IEEE Transactions on Knowledge and Data Engineering, provides a comprehensive overview of the development and advancements in natural language interfaces (NLIs) designed for querying and visualizing tabular data. The survey discusses

various systems and methodologies that enable users to interact with data using natural language, eliminating the need for complex querying languages or programming skills. It examines the challenges, such as ambiguity in natural language, and the techniques used to address them, including the integration of machine learning models and rule-based approaches. The paper also highlights the evolution of NLIs, the current state of the technology, and its potential to democratize data analysis and visualization, making it more accessible to non-expert users.[3]

The paper titled "Language-Driven Visualization Design: A Study of LLMs in Interactive Data Exploration," presented at IEEE VIS 2024, investigates the role of large language models (LLMs) in facilitating interactive data exploration and visualization design. The authors explore how LLMs can be utilized to enable users to create and customize data visualizations through natural language commands, thus reducing the need for technical expertise in data visualization tools. The study delves into the effectiveness of LLMs in understanding user queries, generating appropriate visualizations, and offering real-time feedback. By integrating LLMs into the data exploration process, the paper highlights the potential of language-driven interfaces to streamline visualization workflows and make data interaction more intuitive and accessible for both experts and non-experts alike. [4]

The paper by Perozzi et al. (2024), titled "Let Your Graph Do the Talking: Encoding Structured Data for LLMs," explores innovative methods for encoding structured data to enhance the capabilities of large language models (LLMs). The authors argue that effectively representing structured data in a way that LLMs can understand is crucial for improving their performance in generating insights and responses. The study introduces various encoding techniques and evaluates their effectiveness in enabling LLMs to leverage the inherent relationships and structures present in the data. By focusing on how structured data can be transformed into formats that LLMs can interpret, the paper aims to bridge the gap between traditional data representation methods and the advanced processing capabilities of LLMs, ultimately enhancing their ability to facilitate data-driven conversations and insights.[5]

The paper by Vertsel and Rumiantsau (2024), titled "Hybrid LLM/Rule-based Approaches to Business Insights Generation from Structured Data," presents a novel framework that combines large language models (LLMs) with rule-based systems to extract valuable business insights from structured data. The authors highlight the strengths and limitations of both approaches, advocating for a hybrid methodology that leverages the precision of rule-based systems for specific tasks

alongside the flexibility and natural language processing capabilities of LLMs. This dual approach aims to improve the efficiency and accuracy of generating actionable insights, particularly in complex business scenarios. By addressing the shortcomings of traditional methods, the paper provides a significant contribution to the field of business intelligence, demonstrating how the integration of LLMs can enhance data analysis and decision-making processes.[6]

Additionally, this chapter explores the latest advancements in natural language processing (NLP) and large language models (LLMs), investigating how these technologies can be harnessed to make data interaction more intuitive and accessible for a broader range of users. Through this exploration, we emphasize the importance of our proposed solution in bridging these identified gaps.

III. COMPARISON

The AI-powered Data Insights and Visualization application provides an innovative approach to data analysis, allowing users to seamlessly explore, process, and visualize large datasets from any location with an internet connection. Unlike traditional data analytics tools that often require extensive technical expertise and complex configurations, this web-based platform is designed with usability and accessibility in mind. Users can upload datasets in popular formats such as CSV and Excel, making it highly adaptable across various industries and professional settings. This broad compatibility ensures that users can integrate their existing data workflows without the need for additional software or extensive reformatting.

One of the key differentiators of this application is its advanced natural language processing (NLP) capabilities, which allow users to interact with their data using simple, intuitive queries. Rather than relying on complex SQL commands, programming knowledge, or specialized data analytics training, users can simply type out questions in plain language and receive meaningful insights instantly. This functionality removes technical barriers that often prevent non-experts from engaging with data-driven decision-making, thereby democratizing data analysis and expanding its accessibility to a much broader audience.

Beyond its ease of use, the application also provides significant cost and time savings compared to conventional data analysis methods. Traditional approaches often require manual data entry, extensive spreadsheet manipulation, and repetitive reporting efforts, all of which consume valuable time and increase the likelihood of human errors. By automating essential data processing tasks, such as data cleaning, anomaly detection, statistical analysis, and visualization generation, the application ensures that users can focus on extracting insights rather than managing data complexities. This automation not only reduces the time spent on data preparation but also eliminates inefficiencies associated with manual reporting, making the entire workflow significantly more productive.

Additionally, the application's real-time data processing capabilities ensure that users always have access to the most up-to-date, accurate insights. Unlike static reports that quickly become outdated, this system provides continuous access to live data, enabling organizations to make proactive, data-driven decisions rather than relying on retrospective analysis. This real-time accessibility is especially beneficial in fast-paced environments where timely decision-making can have a direct impact on business performance, operational efficiency, and competitive advantage.

Furthermore, remote accessibility allows users to engage with their data from virtually anywhere, breaking down geographical barriers and fostering collaboration across teams and departments. Whether a user is in an office setting, working remotely, or managing operations on the go, they can easily retrieve, analyze, and share data-driven insights, enhancing overall efficiency and teamwork. This flexibility ensures that decision-making is no longer restricted to specific locations or devices, making data insights more readily available whenever and wherever they are needed.

By combining ease of use, automation, real-time analytics, and universal accessibility, this application serves as a powerful tool for transforming raw data into actionable insights. It empowers individuals, businesses, and organizations to fully leverage their data for strategic decision-making, ultimately driving greater efficiency, accuracy, and informed decision-making across various domains. The ability to effortlessly analyze data, generate visualizations, and gain valuable insights without requiring extensive technical expertise makes this application a game-changer in the field of data analytics, offering unparalleled value to users seeking efficient and intelligent data solutions.

IV. REVIEW OF EXISTING SYSTEM

The rapid expansion of data-driven decision-making has led to the emergence of numerous data analysis and visualization tools, each catering to specific user needs. Among the most widely used solutions are traditional spreadsheet applications such as Microsoft Excel and Google Sheets, which have long been relied upon for data entry, calculations, and visualization. These platforms offer robust functionalities such as pivot tables, formula-based computations, and basic charting tools, making them indispensable for many businesses and professionals. However, they come with inherent limitations—creating complex formulas, handling large datasets, and performing advanced analytics require significant technical proficiency. Additionally, manual data cleaning and manipulation can be tedious and error-prone, often leading to inefficiencies and inaccuracies in data-driven decision-making.

To address the limitations of spreadsheets, Business Intelligence (BI) tools such as Tableau, Power BI, and QlikView have gained popularity. These tools offer advanced visualization capabilities, interactive dashboards, and in-depth analytics, making them powerful solutions for data exploration and reporting. However, their steep learning curves, high subscription costs, and the need for extensive setup and configuration can pose significant barriers to adoption, particularly for small businesses, non-technical users, and budget-conscious organizations. Additionally, maintaining and updating BI tools requires ongoing investment, both in terms of financial resources and technical expertise, making them less accessible to users who require quick and cost-effective data insights.

Some organizations opt for custom-built data analysis software tailored to their specific business needs. While these solutions provide highly personalized features and functionalities, they come with several drawbacks, including lengthy development timelines, significant financial investment, and the requirement for skilled professionals to build and maintain the system. The high cost and complexity associated with custom solutions make them infeasible for many businesses, particularly those looking for scalable and flexible data analytics solutions.

In response to the need for more accessible data visualization tools, several cloud-based platforms, such as Google Data Studio and Datawrapper, have emerged. These platforms allow users to connect to various data sources and create visualizations quickly and efficiently without requiring specialized software installations. However, they often lack comprehensive data analysis capabilities, focusing primarily on charting and reporting rather than in-depth data exploration. Moreover, limited data cleaning functionalities make it challenging to ensure accuracy and reliability, requiring users to preprocess their data manually before utilizing these tools.

Another recent advancement in data analysis tools is the integration of Natural Language Processing (NLP) technologies, which enable users to interact with their data using simple, conversational queries. While promising, many NLP-driven solutions remain in their early stages of development, facing challenges related to query accuracy, limited contextual understanding, and compatibility with diverse data formats. Additionally, many NLP-based tools struggle to generate meaningful insights beyond simple summaries, making them insufficient for complex data analysis and visualization needs.

Despite the variety of existing data analytics tools, each comes with trade-offs that limit accessibility, usability, and efficiency. Many tools are either too complex for non-technical users, too expensive for small businesses, or lack key features such as automated data cleaning, real-time processing, and natural language interaction. As a result, there remains a significant gap in the market for a comprehensive, user-friendly data analysis and visualization solution that combines the intuitive nature of NLP, the analytical power of BI tools, and the flexibility of cloud-based platforms.

The proposed Data Insights and Visualization application aims to fill this gap by offering a seamless, AI-driven experience that allows users to upload, clean, explore, and visualize data effortlessly. By leveraging Large Language Models (LLMs), the application simplifies complex data interactions, enabling users to generate insights through conversational queries without requiring specialized skills. This approach ensures that both technical and non-technical users can access, analyze, and act on their data efficiently, ultimately enhancing productivity, decision-making, and overall business intelligence.

V .METHODOLOGY

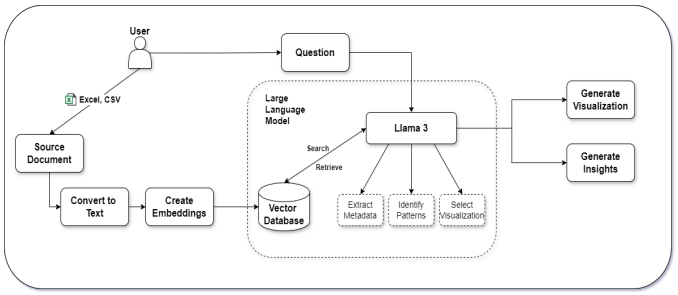


fig 5.1 Architectural Framework

This diagram represents a system where a user queries structured data (e.g., Excel or CSV files) to gain insights and visualizations using a Large Language Model (Llama 3). The source documents are first converted to text, and embeddings (numeric representations) are created and stored in a vector database. When a user asks a question, the LLM searches the database to extract relevant metadata, identify patterns, and select suitable visualizations. The system then generates insights and visualizations based on the retrieved information, providing answers in a meaningful and graphical form.

- Data Collection

Flexible Data Formats:

The platform is designed to accommodate a wide range of input data formats, making it versatile and user-friendly for various types of users. It can accept files in formats such as Excel, CSV, or other common structured data types. This flexibility ensures that users from different industries or domains can seamlessly upload and process their data without needing to convert files into a specific format. For instance, an Excel sheet with multiple tabs or a CSV with thousands of rows can be ingested into the system, preserving the structure and data integrity. This broad support reduces friction in data handling and allows users to focus on analyzing the data rather than on preprocessing it.

- Data Analysis Using LLMs

Advanced Feature Extraction:

Leveraging Large Language Models (LLMs) allows for the extraction of complex and high-level features from the uploaded data. Beyond simple statistical measures, the LLM can detect advanced patterns such as time series trends, cyclical behaviors, and hierarchical relationships within the data. For instance, in a time series dataset, the LLM could automatically detect seasonality, trends, and recurring patterns, offering insights that traditional models might miss. If the data contains hierarchical structures, like nested categories, the LLM can identify and maintain these relationships during analysis. This advanced extraction capability enriches the analysis process and makes it more sophisticated, enabling deeper and more meaningful insights.

- Contextual Understanding:

The LLM is equipped with the ability to interpret the data contextually, using domain-specific knowledge when necessary. This feature is crucial for generating insights that are not only mathematically correct but also relevant and actionable within a specific domain. For example, if the data pertains to the financial sector, the LLM can apply its understanding of financial concepts (such as market cycles, economic indicators, etc.) to enhance its analysis. This contextual awareness allows the LLM to provide insights that are more accurate and tailored to the user's needs, ensuring that the data's meaning is preserved and interpreted correctly.

- Automated Chart Creation

Dynamic Chart Selection:

The system is designed to intelligently select the most suitable type of visualization based on the characteristics of the data and the user's specific query, rather than relying on static, predefined rules. This dynamic approach means that the system can evaluate the nature of the data—whether it's time-series data, categorical data, or numerical distributions—and choose an appropriate chart type, such as a line chart, bar chart, or heatmap, respectively. For instance, if the user's data contains two variables with a strong correlation, the system may suggest a scatter plot, while hierarchical data might lead to the creation

of a tree map. This flexibility enhances the user experience, ensuring that visualizations are informative and contextually appropriate.

- Data-Driven Recommendations:

In addition to selecting charts based on the current data and query, the system can also provide proactive recommendations for additional visualizations that the user may not have initially considered. For example, if the user is analyzing sales data and generates a time-series graph, the system could suggest other visualizations like a pie chart to display the distribution of sales across regions or a histogram to show sales performance by category. These recommendations can expand the user's exploration of the data and encourage a deeper understanding of potential insights that might otherwise be overlooked.

- User-Friendly Visualization

Export and Sharing Options:

To enhance usability, the platform allows users to export their visualizations in various file formats, such as PNG, PDF, or even vector formats like SVG for high-resolution needs. These export options ensure that users can share their findings in a format that best suits their presentation or reporting needs, whether for business meetings, academic papers, or internal reports. Additionally, the platform could offer direct sharing features, enabling users to send visualizations via email or to integrate them into collaborative tools like Slack or Microsoft Teams. This ease of export and sharing ensures that insights generated on the platform can be disseminated quickly and effectively, supporting better decision-making.

VI. RESULT

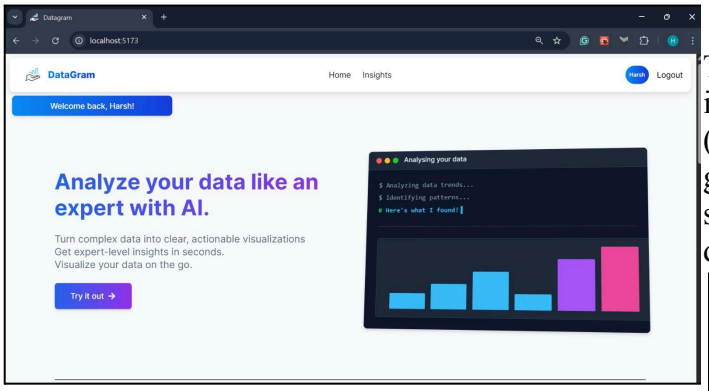


fig 6.1.1 Home Screen

This is the homepage of **DataGram**, a smart AI-powered platform that transforms complex datasets into clear, actionable visualizations—making expert-level analysis simple and fast.

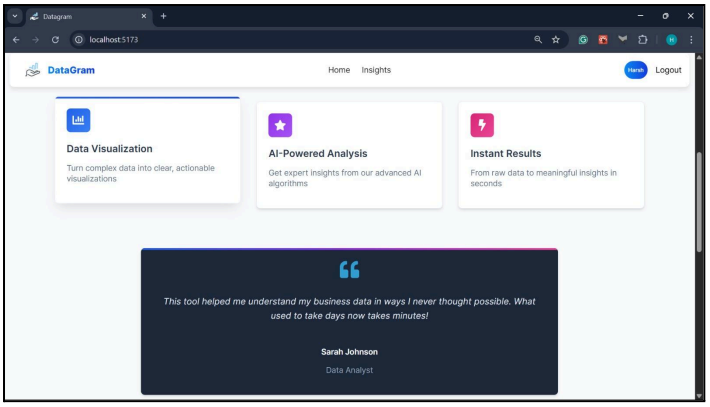


fig 6.1.2 Home Screen (Scrolled view)

This section highlights DataGram's core features—data visualization, AI-powered analysis, and instant results—showcasing how users can turn raw data into meaningful insights within seconds.

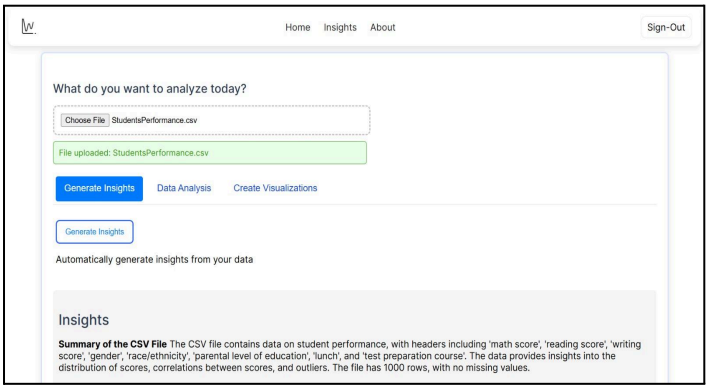


fig 6.1.3 Insights Generation

This screen demonstrates DataGram's intelligent insight generation feature. After uploading a dataset (StudentsPerformance.csv), users can instantly generate meaningful insights, such as column summaries, data distribution, correlations, and outlier detection—all without writing a single line of code.

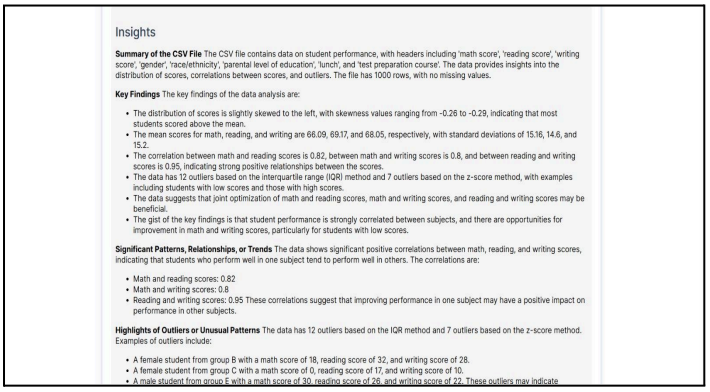


fig 6.1.4 Insights Display

This screen showcases DataGram's automated insight explanation module. After analyzing the uploaded dataset, it generates a detailed summary highlighting:

- Key Findings like skewness in score distributions, subject-wise mean and standard

deviation, and strong correlations among subjects (e.g., math and writing: 0.8).

- Significant Patterns & Trends such as students performing well in one subject tending to excel in others.
- Outlier Detection identifying specific students with unusual score combinations.

This feature helps users quickly grasp the core insights of their data—turning raw numbers into understandable narratives.

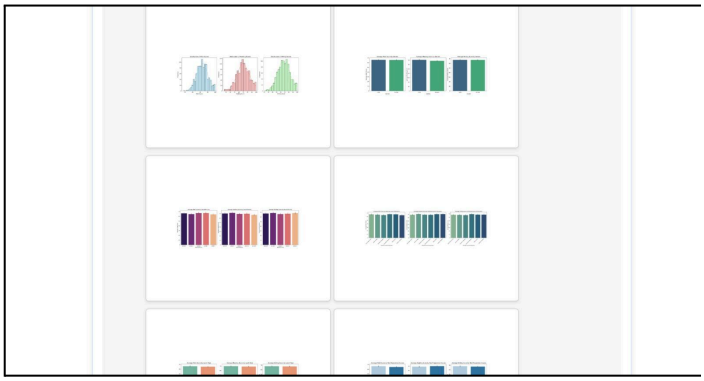


fig 6.1.5 Insights Visualizations

The auto-generated visualizations from DataGram provide key insights into student performance. Distribution plots reveal the spread of scores in reading, writing, and math, helping identify overall performance trends. Gender-wise comparisons highlight that average scores vary slightly between male and female students. The impact of parental education is also evident, with higher education levels generally correlating with better student outcomes. Further, the ethnicity-wise breakdown showcases performance trends across different racial groups. The effect of test preparation is clear, as students who completed a prep course tend to score higher. Lastly, lunch type—standard vs. free/reduced—also shows a noticeable impact on student performance.

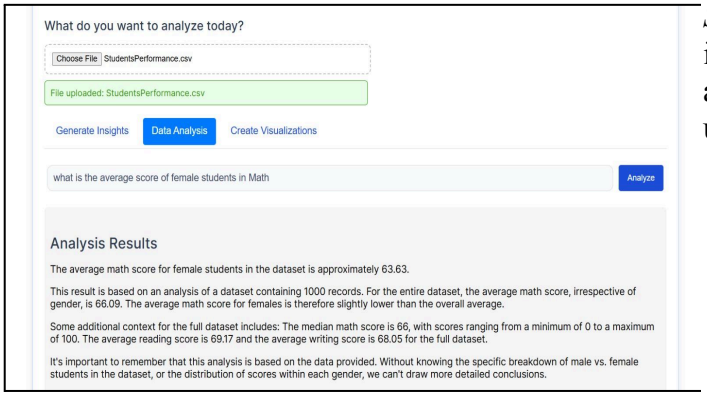


fig 6.1.6 Chat to Analysis & the result

The analysis reveals that the average math score for female students in the dataset is approximately 63.63. This value is slightly below the overall average math score of 66.09 across all students in the dataset of 1,000 records. The median math score is 66, with scores ranging from 0 to 100. Additionally, the average reading score is 69.17, and the average writing score is 68.05 for the full dataset. It's important to note that this analysis is based solely on the provided data, and more detailed conclusions would require a deeper breakdown of scores by gender and other variables.

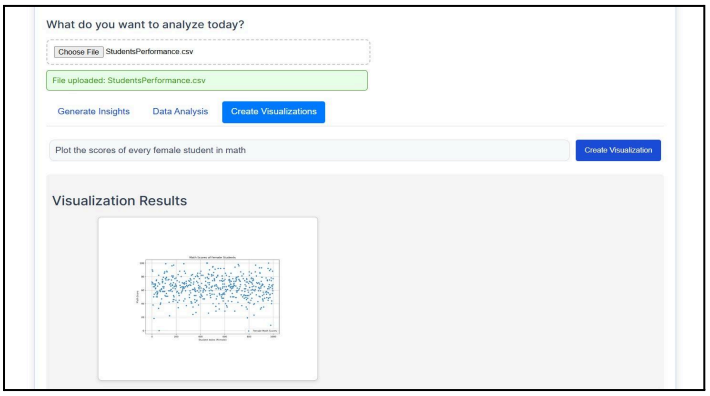


fig 6.1.7 Chat to Visualization & the result

The scatter plot shows math scores of female students, with most scoring between 50 and 80. This suggests generally consistent performance among them.

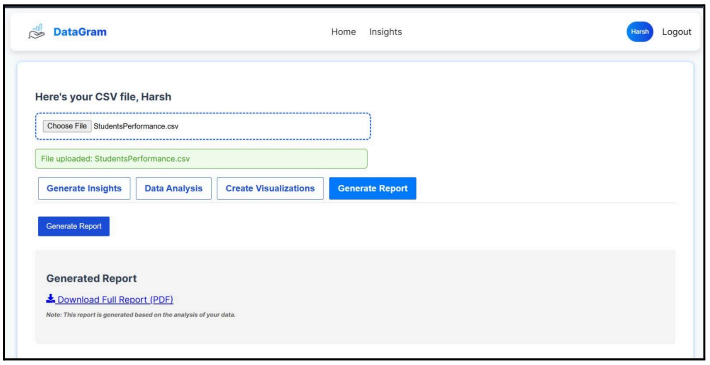
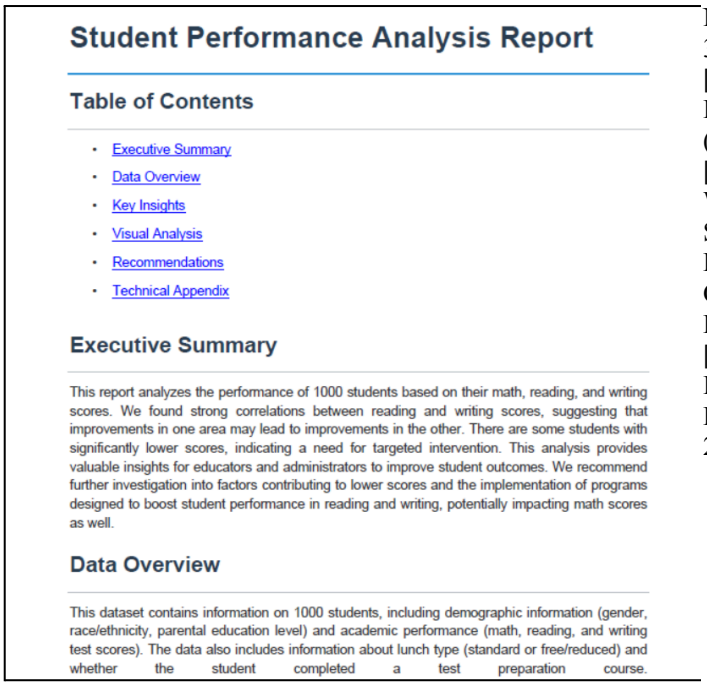


fig 6.1.8 Generate Report

The platform allows users to upload CSV files for automated analysis. After uploading a file (e.g., *StudentsPerformance.csv*), users can generate insights, perform data analysis, create visualizations, and generate a downloadable PDF report based on the uploaded data.



Student Performance Analysis Report

Table of Contents

- Executive Summary
- Data Overview
- Key Insights
- Visual Analysis
- Recommendations
- Technical Appendix

Executive Summary

This report analyzes the performance of 1000 students based on their math, reading, and writing scores. We found strong correlations between reading and writing scores, suggesting that improvements in one area may lead to improvements in the other. There are some students with significantly lower scores, indicating a need for targeted intervention. This analysis provides valuable insights for educators and administrators to improve student outcomes. We recommend further investigation into factors contributing to lower scores and the implementation of programs designed to boost student performance in reading and writing, potentially impacting math scores as well.

Data Overview

This dataset contains information on 1000 students, including demographic information (gender, race/ethnicity, parental education level) and academic performance (math, reading, and writing test scores). The data also includes information about lunch type (standard or free/reduced) and whether the student completed a test preparation course.

fig 6.1.9 Report

After uploading the CSV file, the platform generated a detailed *Student Performance Analysis Report*. The report includes an executive summary, data overview, key insights, visual analysis, recommendations, and a technical appendix. It provides a comprehensive analysis of student performance, helping educators identify trends, correlations, and areas for targeted improvement.

VIII. REFERENCES

[1]Dr. Sudha SV,Sunil S K,Parthiv Akilesh A S,Satish G"Democratizing Data Science:Using Language Models for Intuitive Data Insights and Visualizations ",IEEE Conference , 2024

[2]Liqi Cheng, Dazhen Deng, Xiao Xie, Rihong Qiu, Mingliang Xu and Yingcai Wu,"SNIL: Generating Sports News from Insights with Large Language Models", Journal IEEE ,vol. 14, no. 8, August,2021

[3]Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey Published in: IEEE Transactions on Knowledge and Data Engineering (Early Access)

[4]Language-Driven Visualization Design: A Study of LLMs in Interactive Data Exploration IEEE VIS 2024

[5]Perozzi B, Fatemi B, Zelle D, Tsitsulin A, Kazemi M, Al-Rfou R, Halcrow J., " Let your graph do the talking: Encoding structured data for llms."2024 Feb 8.

[6]Vertsel A, Rumiantsau M.Hybrid LLM/Rule-based"Approaches to Business Insights Generation from Structured Data." arXiv preprint arXiv:2404.15604. 2024 Apr 24.

[7]Shang-Ching Liu, ShengKun Wang, Tsungyao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. 2023. JarviX.

[8]A LLM No code Platform for Tabular Data Analysis and Optimization. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 622–630, Singapore. Association for Computational Linguistics.

[9]Pingchuan Ma,Rui Ding,Shuai Wang,Shi Han,Dongmei Zhang,"InsightPilot: An LLM-Empowered Automated Data Exploration System",2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 346–352,December 6-10, 2023

[10]DataVizGPT: Generating Visualizations from Natural Language Descriptions Conference Name:ACM SIGGRAPH (Year Of Publication: 2023)

[11]Shang-Ching Liu, ShengKun Wang, Tsung Yao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. . "JarviX: A LLM No code Platform for Tabular Data Analysis and Optimization." 2023 Conference on Empirical -Methods in Natural Language Processing: Industry Track, Singapore.

[12]"Automated Data Visualization from Natural Language via Large Language Models: An Exploratory Study"Y Wu, Y Wan, H Zhang, Y Sui, W Wei, W Zhao… - … Management of Data, 2024 - dl.acm.org

Industry / Inhouse:
Research / Innovation:

Project Evaluation Sheet 2024-25(Sem 6) **Class: D12A/B/C**

9

Title of Project (Group no): Data Insights And Visualisation Using LLMs (Grp No:-9)

Mentor Name & Group Members: Dr. Sujata Khedkar, Harsh Pimparkar (31), Vanya Budhraja (10), Prem Chaudhary (22), Yash Ingale (29)

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	4	4	4	3	5	2	2	2	2	3	3	3	4	4	45

Comments:

Dr Sujata Khedkar
Name & Signature Reviewer1

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	Total Marks (50)
Review of Project Stage 1	4	4	4	3	5	2	2	2	2	3	3	3	4	4	45

Comments:

Date: 01/03/2025

Name & Signature Reviewer2

Industry / Inhouse:
Research / Innovation:

Project Evaluation Sheet 2024-25(Sem 6)

Class: D12A/B/C ✓

Title of Project (**Group no**): 09 - Data Insights Using Large Language Model

Mentor Name & Group Members: Dr. Sujata Khedkar, Varun Budhani (10), Prem Ghundiyal (22), Yash Ingale (29), Harsh Pimparkar (51)

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	**Total Marks (50)**
Review of Project Stage 1	5	5	4	3	5	2	2	2	2	3	3	3	4	5	48

Comments: Good work.

Dr Sujata Khedkar
Name & Signature Reviewer1

	Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (3)	Applied Engg & Mgmt principles (3)	Life - long learning (3)	Professional Skills (5)	Innovative Approach (5)	**Total Marks (50)**
Review of Project Stage 1	5	5	4	3	5	2	2	2	2	3	3	3	4	5	48

Comments: Good work.

Indu B.
Name & Signature Reviewer2

Date: 01/04/2025