

**D.Y. PATIL COLLEGE OF ENGINEERING &
TECHNOLOGY, KOLHAPUR**

(An Autonomous Institute)



DEPARTMENT OF CSE (DATA SCIENCE)

A

Synopsis Report

on

“Axnos AI”

Submitted by

Name	Roll No.
Mr. Shreyash Suresh Patil	59
Mr. Aakash Raju Mohole	60
Mr. Shreyash Ravindra Shetty	67
Mr. Vinayak Dilip Vathare	68

Under the guidance of

Mrs. T. V. Deokar

Final Year B. Tech. CSE (Data Science)

Academic Year 2025-26

INDEX

Sr. No.	Topic	Page Number
1.	Abstract	1
2.	Introduction	2
3.	Literature Survey	3
4.	Motivation	5
5.	Innovative Contribution	6
5.	Problem Definition	7
	a. Problem Statement	7
	b. Objectives	8
	c. Proposed Architecture	9
	d. Experimental Setup	10
6.	References	11
7.	Approval Statement	12
8.	Plagiarism Report	13

ABSTRACT

Modern organizations increasingly depend on large amounts of data, often stored in common formats like CSV, Excel, JSON, and PDF files. However, getting useful insights from these raw datasets remains a challenge, especially for non-technical users without programming skills or knowledge of data analysis tools. "Axnos AI, Talk to Your Data" fills this gap by offering an AI chatbot platform that allows users to upload data files and ask questions in plain language, removing the need for coding or complex interfaces. By using advanced AI models, the proposed system turns user questions into executable Pandas operations that filter, group, sort, and summarize data as needed.

All queries, results, and file uploads are securely logged, letting users check their analysis history whenever they like. The proposed system uses a range of modern technologies, including asynchronous task handling, RESTful APIs and a user-friendly frontend made with Tailwind CSS and shadcn UI. By making data analysis accessible to everyone, Axnos AI helps business users, researchers, and anyone working with data make informed decisions quickly, promoting a data-driven culture across various fields.

INTRODUCTION

The growing importance of data in daily business decisions, research, and personal productivity highlights the need for tools that can uncover insights from commonly used file formats. Yet, many professionals and enthusiasts struggle with analyzing raw datasets stored as CSV, Excel, JSON, or PDF files due to technical barriers. Traditional spreadsheet software, SQL queries, and coding in Python or R can be overwhelming for those without specialized training. This often results in inefficiencies, misinterpretation of important information, or dependence on technical experts, limiting the full benefits of data-driven insights.

The project introduces an easy-to-use AI chatbot that allows users to explore and analyze their data in real-time through simple conversations. Users can upload their data files and ask questions or make requests naturally, such as "Show me the top five products by sales" or "Find all entries from last quarter." The AI engine converts these requests into Pandas-based operations, quickly delivering actionable insights without users needing to see any code.

To provide this smooth experience, Axnos AI uses a strong technology stack: Advanced language understanding and logical reasoning, asynchronous tasks guarantee quick responses even for large or complex datasets, and the RESTful API gateway and Neon ensure secure and scalable communication with the backend. The application also features ongoing user history and session management—every query, result, and uploaded file is logged for easy review, enhancing transparency and usability. The sleek, responsive interface built with Tailwind CSS makes the platform modern and simple to navigate, even on mobile devices.

Ultimately, Axnos AI is more than just a tool; it represents progress toward making data analytics accessible to everyone, regardless of technical ability. By lowering the entry barrier for meaningful analysis, it empowers users across industries to make smarter, faster, and more confident decisions using the wealth of data available to them.

LITERATURE SURVEY

Wang et al. proposed an NLI for interactive data analysis. The system recommends context-aware next-step queries to guide novice analysts during step-by-step exploration. It also organizes query histories and results into dashboards, showing improved effectiveness compared to a non-recommendation baseline. However, it focuses more on query suggestions than on multi-format ingestion or remembering analyses across sessions.

Quamar et al. surveyed NLID approaches, including rule-based systems, text-to-SQL, and hybrids. They outlined challenges in identifying entities, linking semantics, and generating structured queries. They also highlighted progress toward conversational multi-turn interfaces and evaluation benchmarks. The monograph maps out the design space and open issues but does not include heterogeneous file analysis or persistent interaction trails.

The IEEE work linked via 9699035 addresses conversational analytics for multi-turn natural language interaction in analytical workflows. It demonstrates accessibility gains for non-technical stakeholders but often relies on predefined schemas or visualization grammars. This creates gaps in generalized onboarding for various file types and in-session provenance unless such features are specifically designed.

Talk2Data allows for exploratory visual analysis through question decomposition. It breaks down complex intents into sub-questions that correspond to visual operations and uses multi-hop reasoning. However, its focus on decomposition and visual answerability does not cover broader uploads for PDF or JSON files or track longitudinal histories of user queries and insights beyond specific study settings.

The ACL N19-1423 paper presents techniques for multi-hop or context-aware question processing to support decomposition, disambiguation, and retrieval for conversational data analysis. However, these methods are tested on text-focused datasets and do not offer end-to-end interfaces for heterogeneous uploads, Pandas-backed code generation, or secure execution for non-technical users.

The literature highlights several approaches to natural language interfaces for data. These include query recommendation, text-to-SQL, hybrid parsing, conversational context, and visual exploration through question decomposition. However, there are still common gaps. These include challenges in onboarding multi-format data, maintaining cross-session histories, and providing secure execution that is accessible to non-technical users.

MOTIVATION

Today, many people and businesses keep their data in formats like CSV, Excel, PDF, and JSON. While saving this data is simple, understanding and using it can be difficult. A lot of individuals do not know how to use programming or advanced software to explore what their data means. Because of this, important information often goes unused. This challenge gave rise to Axnos AI.

Making Data Simple for Everyone

Most data tools are built for people who already know how to code or work with complex software. This makes it hard for non-technical users such as teachers, store managers, office workers, and students to use their data. Many of them want helpful insights but don't have the required skills. Axnos AI makes it possible for them to simply type a question in everyday English and get meaningful answers. No coding or special software is needed.

Talking Naturally with Data

People are now familiar with talking to devices using natural language. Axnos AI applies this idea to working with data. Instead of scrolling through spreadsheets or writing formulas, users can just ask something like, "What were the most sold items last month?" and get a clear answer. This way of asking makes data work feel easier and more human.

Helping People Use Data to Decide Better

When people can easily understand their data, they make better decisions. Axnos AI supports this by removing the need for technical knowledge. It helps users rely on real facts instead of guesses. This leads to clearer planning, fewer mistakes, and more confident choices in everyday work.

Axnos AI is made to be easy to understand and use. It works with many types of files and keeps a record of every question and answer. This means users can go back and check what they did anytime. Whether someone is preparing for a project or writing a report, the tool helps them find accurate information quickly and without hassle.

INNOVATIVE CONTRIBUTION

1. Natural Language to Data Science tasks:

The core innovation of Axnos AI is its ability to translate natural language prompts into executable data operations without requiring programming knowledge.

- User queries such as “*Show me sales by region*” are automatically mapped to Pandas operations.
- The system supports filtering, grouping, sorting, visualization, and statistical summaries directly from plain English.
- Unlike traditional BI tools, this removes the barrier of SQL queries, scripts, or complex UI navigation.

2. Multi-Format Data Handling:

Axnos AI is designed to support heterogeneous data sources.

- Users can upload CSV, Excel, JSON, and PDF datasets seamlessly.
- File validation and preprocessing ensure data consistency before analysis.
- This multi-format support distinguishes Axnos AI from existing NLID systems that are often schema-dependent or SQL-focused.

3. Secure and Transparent Code Generation:

To promote trust and reproducibility, Axnos AI provides not only the result but also the generated code.

- Each operation is sandboxed to prevent malicious execution.
- Users can review, copy, or reuse the generated code for further customization.
- This transparency builds confidence and bridges the gap between novice and technical users.

4. User History and Session Management:

A unique contribution of Axnos AI is its persistent query and file history:

- Every uploaded file, query, and result is securely logged.
- Users can revisit past analyses, export sessions, and track decision-making trails.

PROBLEM DEFINITION

a. Problem Statement:

“To develop a system for Data Science Tools that enables users to perform data analysis, preprocessing, and visualization using simple text or voice prompts.”

b. Objectives:

1. To design a user-friendly interface that allows data upload and interaction through natural language or voice commands.
2. To implement a natural language processing (NLP) engine that interprets user queries and maps them to appropriate data operations.
3. To automate common data science tasks such as data visualization, cleaning, and summary statistics generation through prompt-based interaction.
4. To provide real-time feedback and responses to user queries by integrating backend data processing with the front-end interface.
5. To support multiple data formats (e.g., CSV, Excel, JSON, PDF) and abstract away technical complexities for users with minimal data science knowledge.

c. Proposed Architecture:

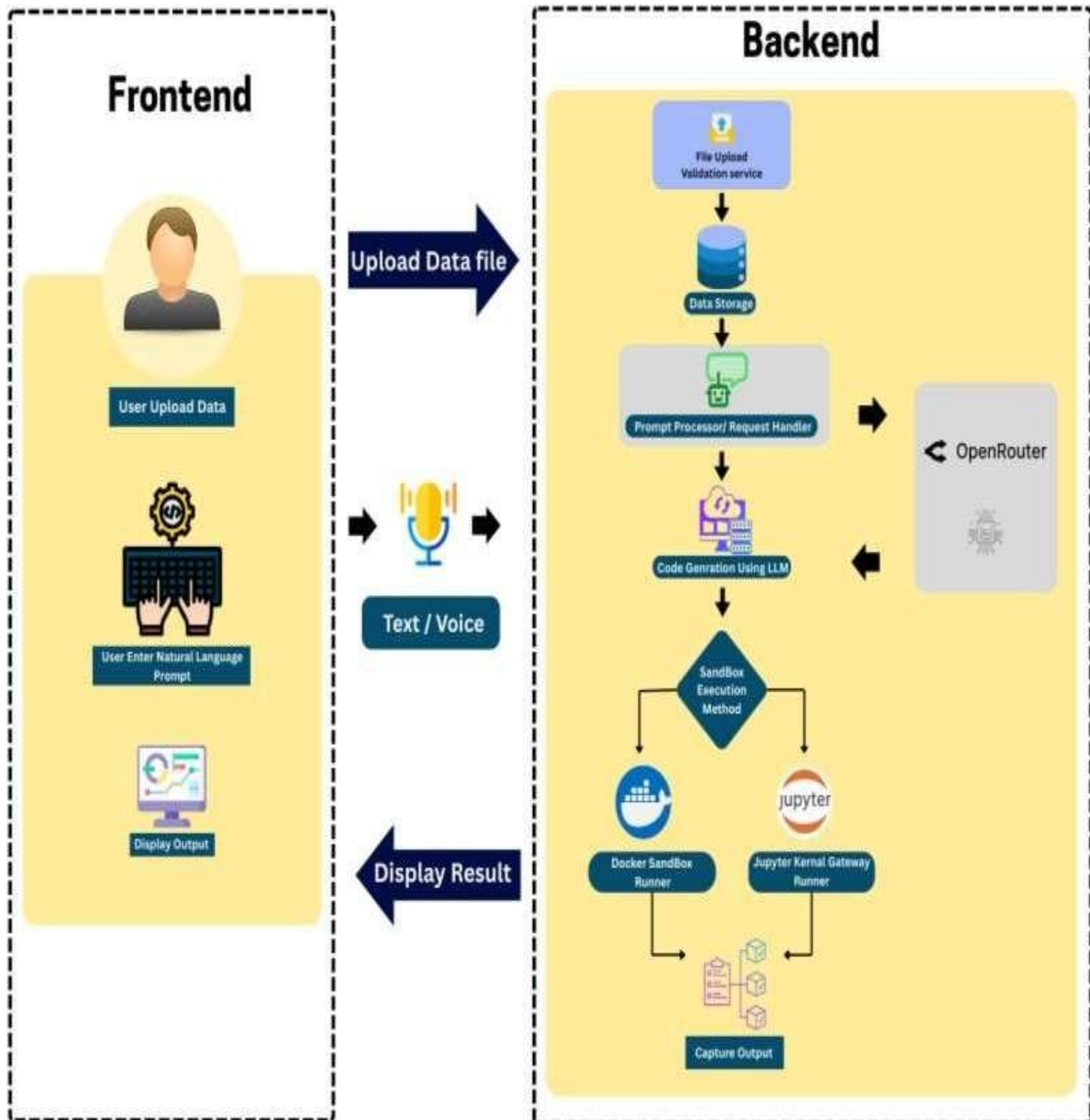


Fig.1 Proposed Architecture Diagram

❖ User Uploads Data:

Users select and upload their data files via a simple web interface. The frontend uses React with file input components for smooth user interaction. An upload API on the backend receives the file for processing.

❖ **File Validates & Saves:**

Uploaded files are checked to ensure they match supported formats like CSV, JSON or XLSX. Backend validation is performed using Python to verify file integrity. Files are stored securely on cloud storage environments.

❖ **User Inputs Prompt:**

Users can provide queries via natural language prompts either by entering or using voice to describe their desired data tasks. The frontend chat or user interface captures these input requests. This interaction initiates the data analysis workflow.

❖ **Prompt Sent to AI:**

The user's prompt is sent from the backend to an AI language model API. The AI interprets the request, determines analysis goals, and drafts a plan. Python backend manages communication with external LLM APIs like OpenAI.

❖ **Code Generated**

The AI generates executable Python code tailored for analysis or visualization. Code can include data processing steps, statistical calculations, or charting commands. LLMs such as GPT-4 drive the automated code authoring process.

❖ **Code Execution**

The generated code is executed within a secure, sandboxed environment. Common sandbox options include Docker containers, cloud VMs, or notebook converters. This ensures safe handling of arbitrary code without compromising system integrity.

❖ **Return Results & Code**

The system compiles analysis output, charts, and the executed code into a user-friendly format. Results are delivered back to the frontend for easy display and review. Users see both visualizations and the underlying code for transparency.

d. Experimental Setup:

Software requirements:

- Operating System: Linux/Windows.
- AI Tools : OpenRouter , Gemini.
- Programming Languages: Python, JavaScript, Typescript, HTML, CSS.
- Frameworks: React, Node.js, FastAPI, Nest.js, Django.
- Databases: Neon, Supabase, MongoDB, PostgreSQL, Redis.
- Tools: Docker, Git & Github, Postman, Transformer, Vercel, Render.
- Styling: Tailwind CSS, Shadcn UI.

Hardware requirements:

- Processor: Intel i3 or higher/ Ryzen 3 or higher
- Memory: Min 4GB RAM

Libraries

- Python Libraries:
 1. **Pandas** - A Python library that provides high-performance, easy- to- use data structures like DataFrame for drawing, transubstantiating, and assaying labeled, irregular data.
 2. **NumPy** - The main scientific computing package in Python, offering fast n- dimensional arrays and vectorized operations for calculation, direct algebra, and arbitrary slice.
 3. **Seaborn** - A statistical visualization library erected on Matplotlib. It provides pandas-friendly, dataset- acquainted APIs with seductive defaults for exploratory plots.
 4. **Matplotlib** - A plotting library for creating static, animated, and interactive visualizations with both py-plot and object-oriented interfaces.
- JavaScript Libraries:
 1. **React** - An element- grounded JavaScript library for erecting dynamic stoner interfaces with JSX and a virtual DOM. It uses one- way data inflow and hooks- grounded state operation.
 2. **Redux** - A state vessel that keeps app state consolidated in a single store.

REFERENCES

[1]	<u>Interactive Data Analysis with Next-step Natural Language Query Recommendation</u> Authors: <u>Xingbo Wang, et al.</u>
[2]	<u>Natural Language Interfaces to Data</u> Authors: <u>Abdul Quamar, Vasilis Efthymiou,</u> <u>Chuan Lei, Fatma Özcan</u>
[3]	<u>Towards Natural Language Interfaces for Data Visualization: A Survey</u> Authors: <u>Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu,</u> <u>Xiongshuai</u> <u>Zhang, Zhiwei Tai, Jianmin Wang</u>
[4]	<u>Talk2Data: A Natural Language Interface for Exploratory Visual Analysis via</u> <u>Question Decomposition</u> <u>Yi Guo, Danqing Shi, Mingjuan Guo, Yanqiu Wu,</u> <u>Qing Chen, Nan Cao</u>
[5]	<u>Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., “BERT: Pre-training of Deep</u> <u>Bidirectional Transformers for Language Understanding”</u> , <u>Proceedings of</u> <u>NAACL-</u> <u>HLT, Vol. 1, No. 1, pp. 4171–4186, 2019.</u>

APPROVAL STATEMENT

This is to certify that the synopsis report entitled "**Axnos AI - Talk to your Data**" has been thoroughly reviewed by the expert panel members of CSE (Data Science) Department at D. Y. Patil College of Engineering & Technology, Kolhapur. The content of the synopsis report has been verified for the accuracy, and the proposed work has been approved for further development.

Roll No.	Unique ID	Student's Name	Signature
59	DSE23111026	Mr. Shreyash Suresh Patil	
60	DSE23116496	Mr. Aakash Raju Mohole	
67	DSE23111633	Mr. Shreyash Ravindra Shetty	
68	DSE23120234	Mr. Vinayak Dilip Vathare	

Date:

Place: Kolhapur

Mrs. T. V. Deokar

Mr. S. K. Patil

Mrs. R. A. Kothiwale

DR. G. V. Patil

Project Guide

Project Co-ordinators

HOD CSE(Data Science)

PLAGIARISM REPORT

iThenticate Page 2 of 16 - Integrity Overview

Submission ID trn:oid::3117490871651





3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography

Match Groups

-  **5 Not Cited or Quoted 3%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%**  Internet sources
- 0%**  Publications
- 0%**  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Fig. 2. Snapshot of Plagiarism

