***B.P.H.E Society Ahmednagar***

**College Ahmednagar**



***SYBBA(CA) SEMESTER IV***

***SUBJECT: - PROJECT EVALUATION SHEET COURSE***

***Project Name-***

***Smart Data Mediator***

***NAME OF STUDENT: -***

***1: RITU BHRADWAJ 2: ANJALI SHINDE   
ROLL NO: 86, 113***

***TEACHER NAME: -   
Madhvi Chavan Ma’am***

******

***Department Computer Science***

***CERTIFICATE***

***This is to certify that the project,***

***Smart Data Mediator***

***Has been completed successfully by-***

***Ritu Bhardwaj, Anjali Shinde***

***For the practical fulfilment of Department of computer Science as laid down by the Pune University during Academic Year 2025 -2026***

***Head of Department Project in charge Python***

***Internal Examiner’s External Examiner’s***

***Place: Ahmednagar Date:***

**Acknowledgement:**

**I would like to extend my heartful appreciation to everyone who played a role in the successful completion of the Smart Data Mediator project.**

**First and foremost ,my deepest thanks go to the “HOD[Head of the Department],Dr.Sayyed Razzak”,and my mentor “Mrs.Madhavi Chavan ”,for their invaluable guidance ,unwavering support,and constructive feedback throughout the entire project development journey.Their expertise and encouragement were crucial in transforming the project from its initial idea into a fully realized system.**

**I am making this project not only for the good marks but also to increase my knowledge.**

**This project would not have been achievable without the collaboration and support of everyone involved,and for that,I am truly thankful.**

|  |  |  |
| --- | --- | --- |
| ***Sr. No*** | ***Chapter*** | ***Page. No*** |
| **1** | **Abstract** | **4** |
| **2** | **Chapter 1.Introduction** | **5-10** |
| **3** | **Chapter 2.Requirement Analysis**  **2.1 Feasibility Study**  **2.2 Fact Finding Techniques**  **2.3 Project Requirement**  **2.3.1 Software/Hardware Specification** | **11-21** |
| **4** | **Chapter 3.System Design**  **3.1. E-R Diagram**  **3.2 Data Flow Diagram** | **22-26** |
| **5** | **Chapter 4.Data Dictionary** | **27-28** |
| **6** | **Chapter 5.Screens**  **5.1 Input Screens**  **5.2 Output Screens** | **29-36** |
| **7** | **Chapter 6.Testing**  **Test Steps, Data Validation Test Cases and result** | **37-39** |
| **8** | **Chapter 7.Limitations**  **7.1 Limitations**  **7.2 Future Enhancement** | **40-41** |
| **9** | **Chapter 8.Bibliography** | **42** |

***INDEX***

***SMART DATA MEDIATOR***

# *Abstract*

# The Smart Data Mediator is a Flask-based intelligent data analysis and reporting system designed to bridge the gap between technical and non-technical users. The project provides an integrated environment where datasets can be uploaded, processed, queried, and visualized with ease. Unlike traditional BI tools that demand SQL knowledge or advanced Excel skills, this system leverages natural language processing (NLP) to let users ask questions in plain English.

# The system supports multiple features such as dataset upload, preview, query processing, visualization generation, project management, and exportable reports (PDF/Excel). This enables organizations and students to explore data quickly, make decisions faster, and reduce dependency on technical experts

# *INTRODUCTION*

# *Objective*

The objective of the Smart Data Mediator is to simplify the process of querying and analyzing datasets. It allows users to interact with data through natural language, generate visual insights, and save reports for future reference.

* To provide a **user-friendly data analysis system** where queries can be written in natural language.
* To integrate **visualizations** (Bar, Line, Pie, Scatter) automatically from queries.
* To enable **project management** by saving previous queries and reports.
* To support **export options** like Excel and PDF for easy sharing.
* To bridge the gap between **non-technical users** and **complex data systems**.

# *Existing System*

Traditional data analysis systems require technical expertise in SQL, Excel functions, or BI tools. Non-technical users face difficulties in extracting meaningful insights without specialized knowledge.

* Traditional tools like **SQL, MS Excel, Power BI** require expertise.
* Non-technical users often depend on developers or analysts.
* Limited real-time data exploration and poor accessibility.

# *Proposed System*

The Smart Data Mediator provides an easy-to-use platform where users can upload datasets, ask questions in plain English, and receive instant results with visualizations. Reports are automatically stored, and users can export them in multiple formats {excel, pdf }

The Smart Data Mediator addresses these limitations by:

* Allowing **dataset upload and preview**.
* Supporting **natural language queries** using NLP (SpaCy).
* Generating **instant data insights with charts**.
* Storing projects and queries in a **database** for reuse.
* Exporting results in multiple formats.

# *Requirement Analysis*

# *Feasibility Study*

The system is technically feasible as it is built on widely used technologies such as Flask, Pandas, and SpaCy. Operational feasibility is ensured since it simplifies data analysis tasks. Economically, it is cost-effective as it uses open-source libraries.

* **Technical Feasibility:** Uses open-source libraries (Flask, Pandas, SpaCy, Matplotlib). Runs on any OS (Windows/Linux/Mac).
* **Economic Feasibility:** Cost-effective as no paid licenses are required.
* **Operational Feasibility:** Easy for non-technical users since the interface is simple and intuitive.
* **Schedule Feasibility:** Can be developed and deployed within a semester project timeframe.

# *Fact Finding Techniques*

* **Observation:** Collected problems faced by students and teachers using Excel.
* **Interviews:** Discussed with faculty about difficulties in data analysis.
* **Record Review:** Reviewed past student reports to identify gaps.

# *Project Requirements*

# *Software Requirements*

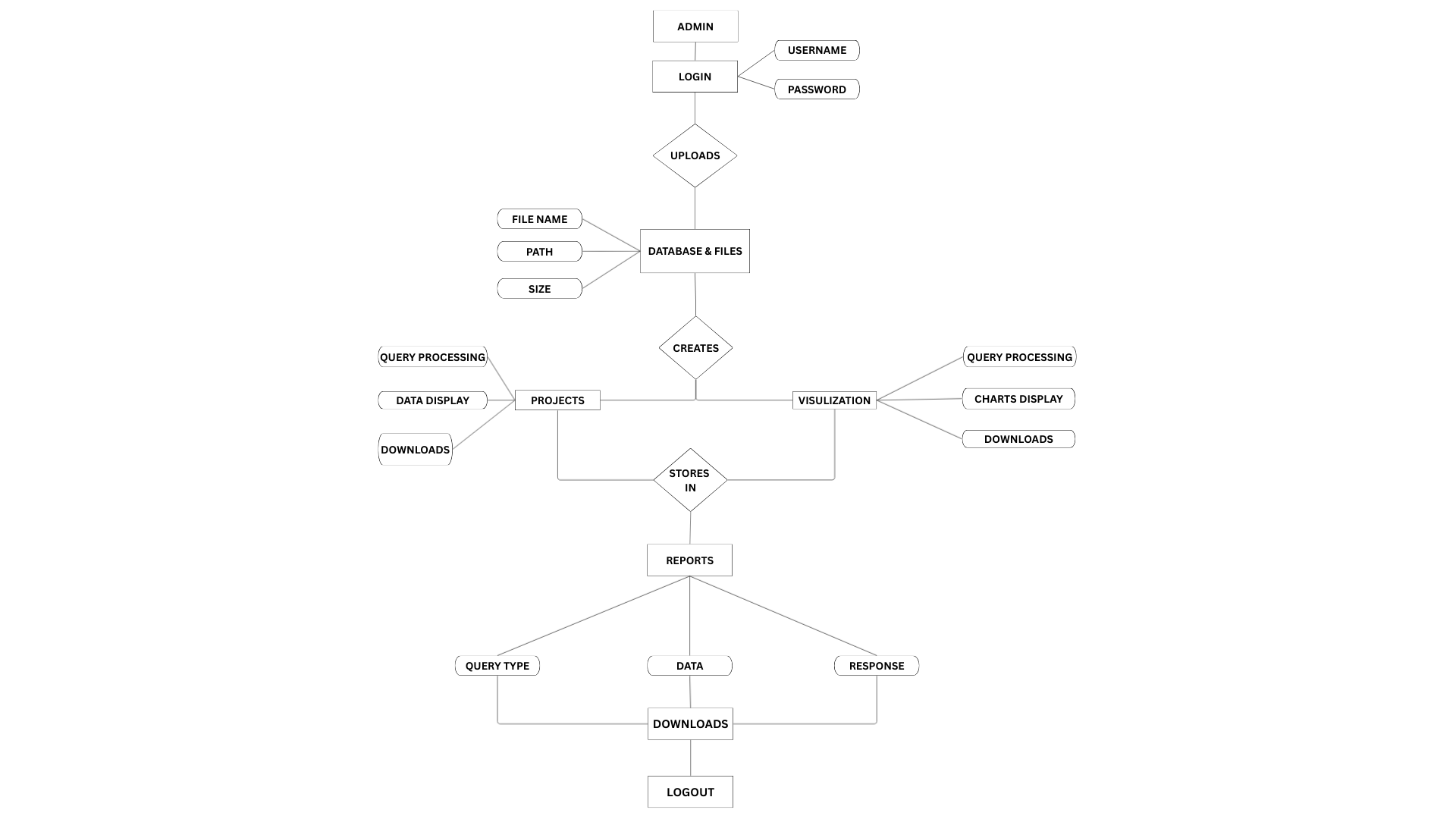
- Python 3.10+  
- Flask 3.0+  
- Pandas, OpenPyXL, ReportLab  
- SpaCy (for NLP)  
- HTML, CSS, JavaScript (Frontend)

# *Hardware Requirements*

- Processor: Intel i5 or higher  
- RAM: 8 GB minimum  
- Storage: 1 GB free disk space  
- OS: Windows/Linux/MacOS

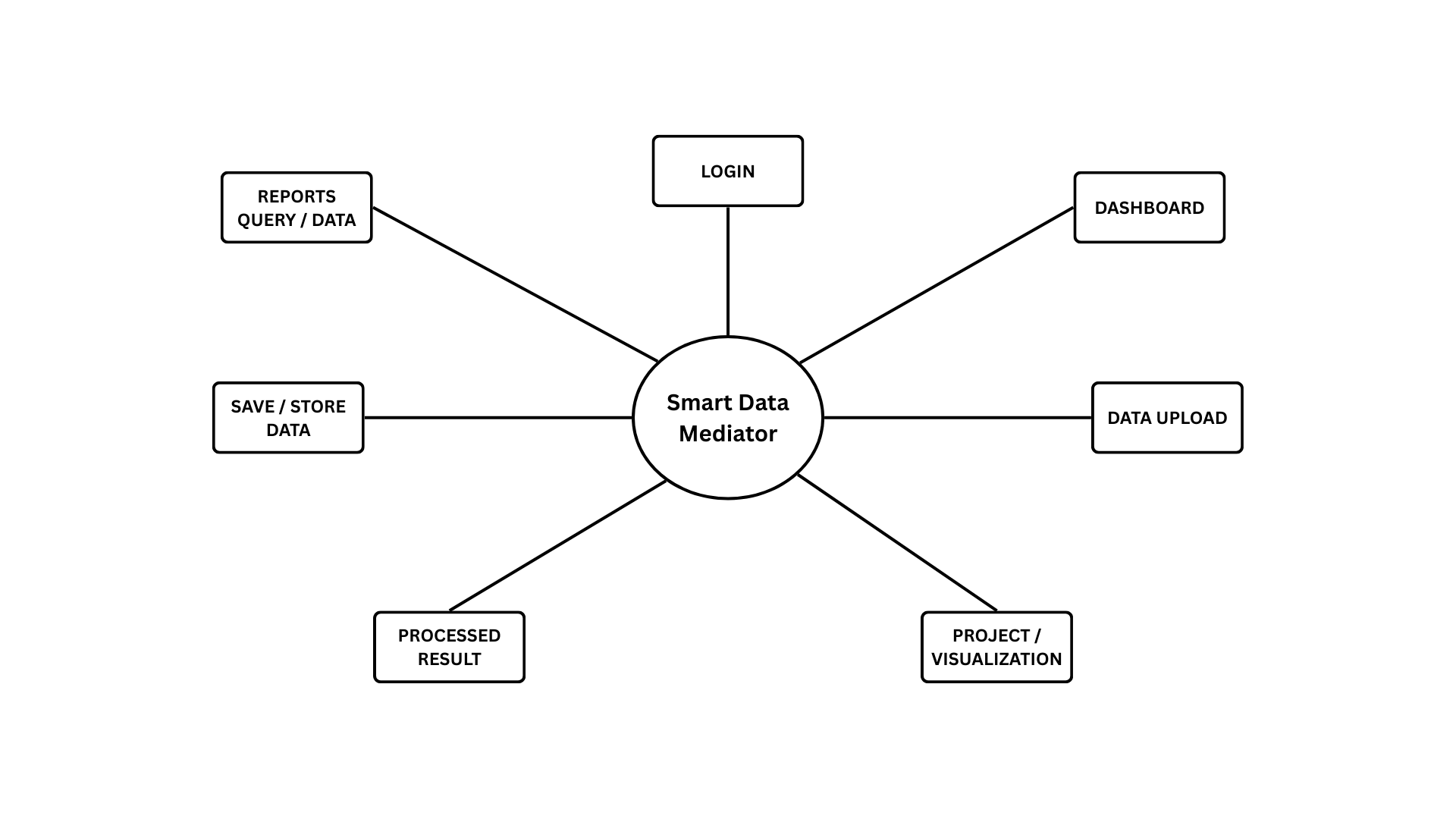
# *System Design*

**The system is designed using ERD and Flowchart diagrams.**

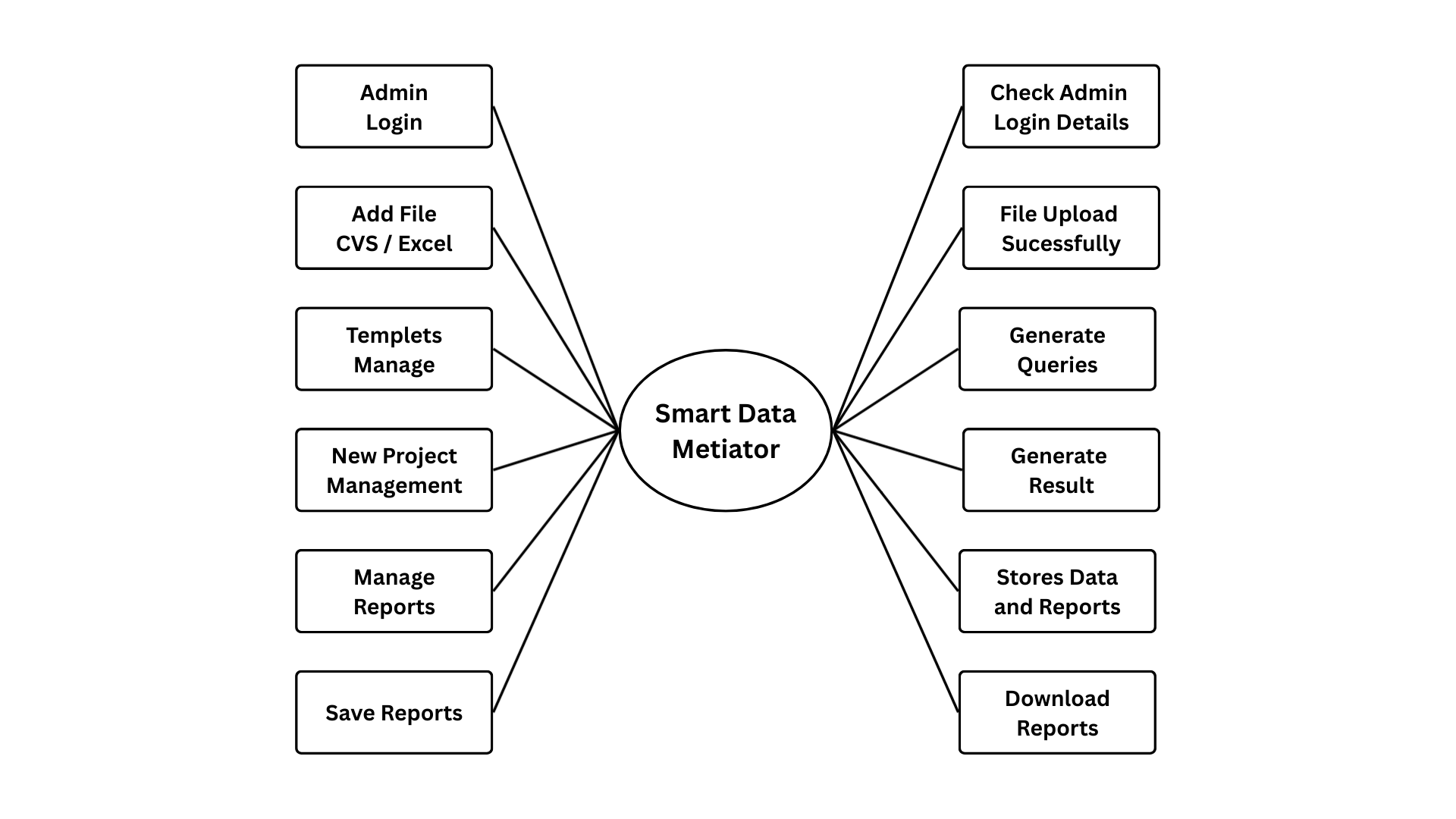
***Entity Relationship Diagram (ERD)***

## *Flowchart:*

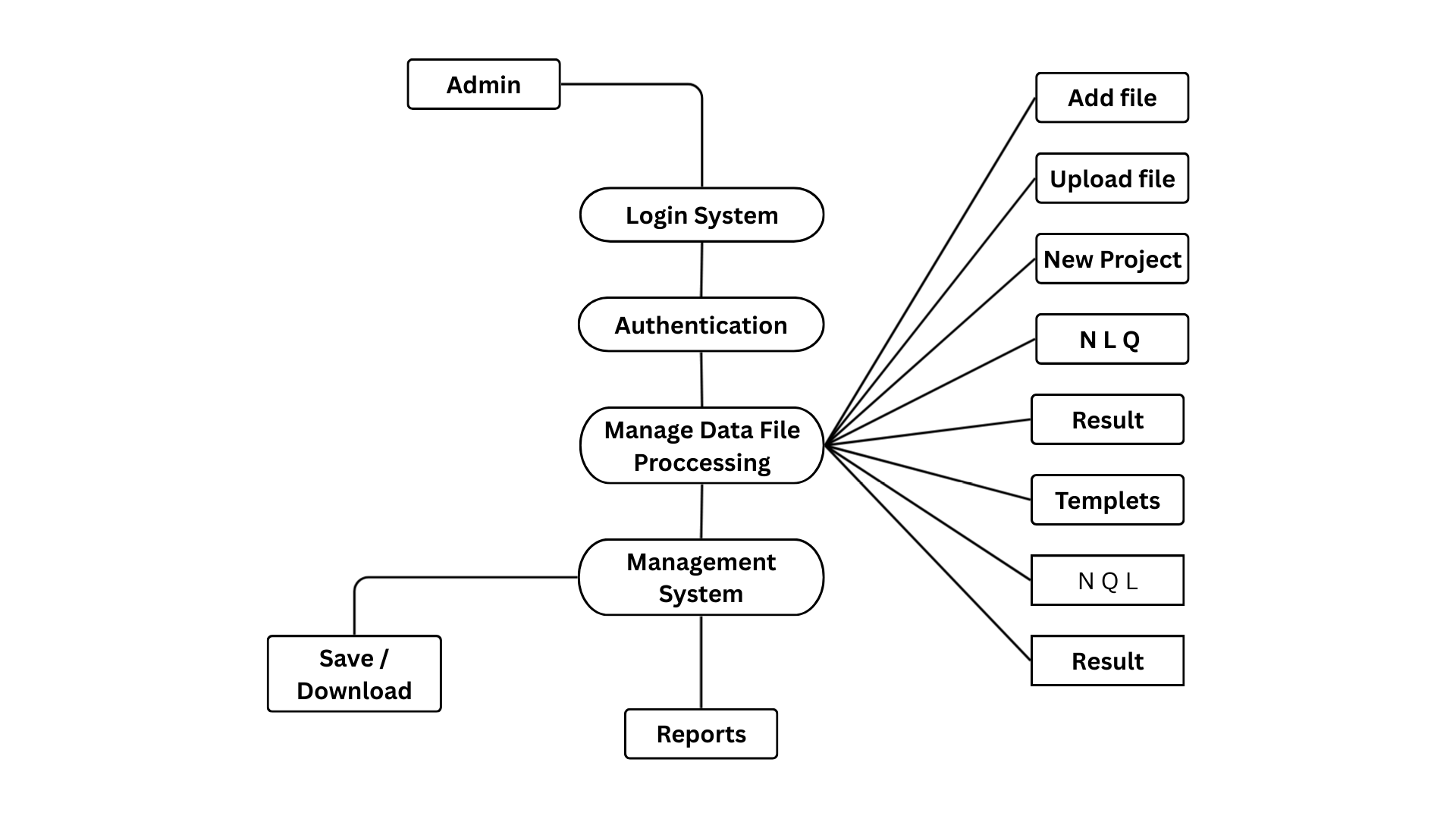
***DFD LEVEL 0***



***DFD LEVEL 1***



***DFD LEVEL 2***



# Data Dictionary

**User Table**

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Data type** | **constraints** |
| user\_id | INT | NOT NULL |
| username | VARCHAR(50) | NOT NULL |
| password | VARCHAR(255) | NOT NULL |
| role | VARCHAR(20) | NOT NULL |
| create\_at | DATETIME | NOT NULL |

**Upload Table**

|  |  |  |
| --- | --- | --- |
| **Attribut name** | **Data type** | **Constraints** |
| upload-id | INT | NOT NULL |
| user-id | INT | NOT NULL |
| filename | VARCHAR(255) | NOT NULL |
| path | VARCHAR(255) | NOT NULL |
| size | INT | NOT NULL |
| data\_uploaded | DATETIME | NOT NULL |

**Project Table**

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Data type** | **constraints** |
| project\_id | INT | NOT NULL |
| user\_id | INT | NOT NULL |
| name | VARCHAR(100) | NOT NULL |
| Path | VARCHAR(255) | NOT NULL |
| **Attribute name** | **Data type** | **constraints** |
| created\_at | DATETIME | NOTNULL |

**Visualization Table**

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Data type** | **constraints** |
| version | INT | NOT NULL |
| Project\_id | INT | NOT NULL |
| version\_number | VARCHAR(10) | NOT NULL |
| date | JSON | NULLABLE |
| created\_at | DATETIME | NOTNULL |

**Report Table**

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Data type** | **constraints** |
| rport\_id | INT | NOT NULL |
| User\_id | INT | NOT NULL |
| query | TEXT | NOTABLE |
| response | TEXT | NOTABLE |
| type | VARCHAR(50) | NOT NULL |
| timestamp | DATETIME | NOTNULL |
| file | VARCHAR(255) | NULLABLE |

***Test cases***

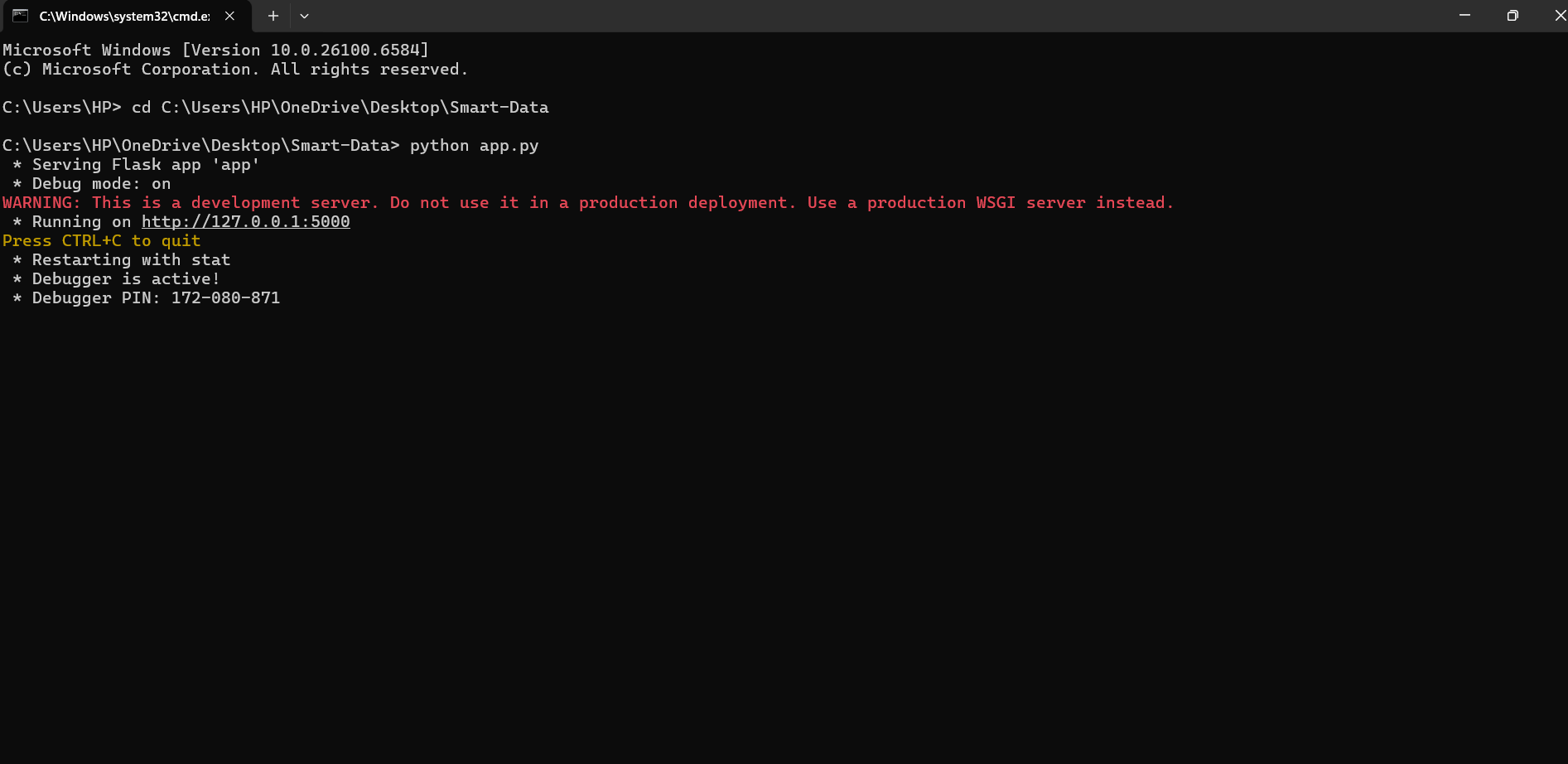
***Data Testing is checking the tables data of the database. The Stored Procedure provides the end-user with the Expected result. Adding, Updating, Deleting of Data is Performed. Functional Testing verifies that the software performs its stated functions in a way that the users expect.***

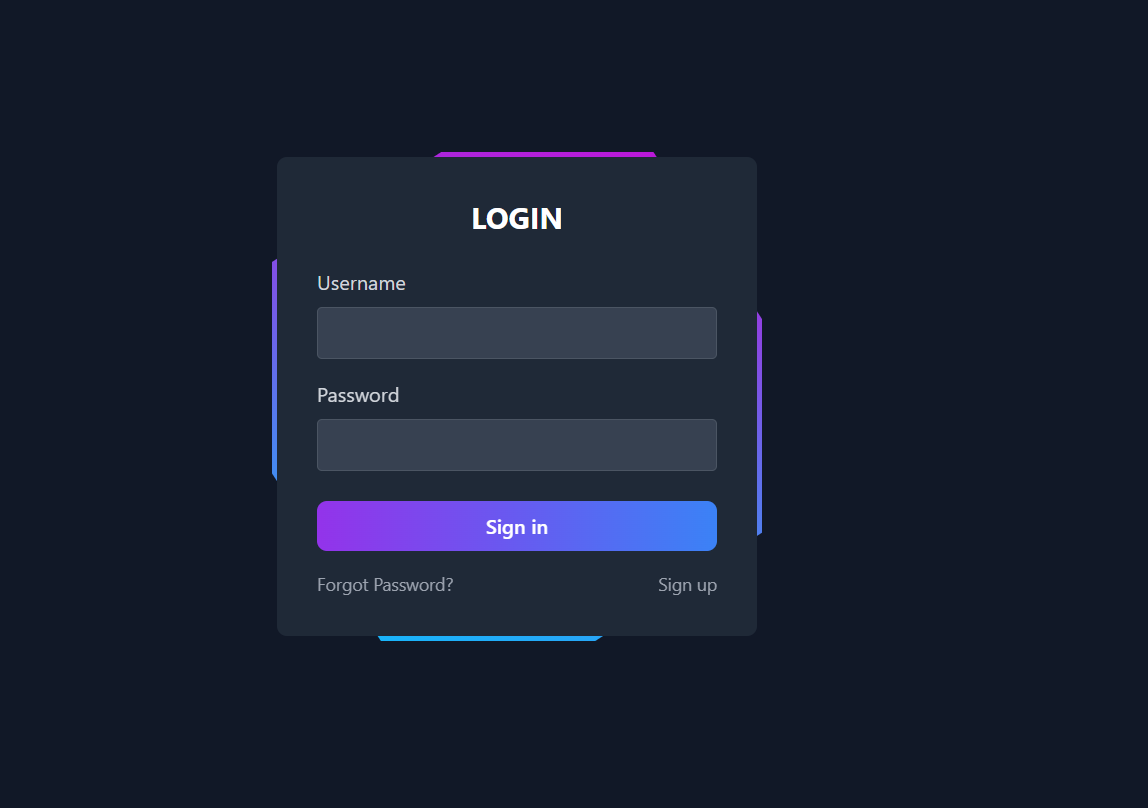
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Case Name | Description | Steps | Input | Expected Result | Actual Result | Status |
| TCDM01 | Login\_Valid\_Credentials | User should be able to log in with valid credentials | 1. Enter valid username & password. 2. Click 'Login'. | Username: ritu Password: 12345 | Redirect to dashboard. | Redirected to dashboard. | Pass |
| TCDM02 | Login\_Invalid\_Credentials | User should not be able to log in with invalid credentials | 1. Enter wrong username or password. 2. Click 'Login'. | Username: abc Password: wrong | Error message should appear. | Error message displayed. | Pass |
| TCDM03 | File\_Upload\_Valid | System should accept valid Excel/CSV upload | 1. Click 'Upload'. 2. Select a valid Excel/CSV file. | sales\_data.csv | File should be uploaded and preview displayed. | Preview displayed. | Pass |
| TCDM04 | File\_Upload\_Invalid | System should reject unsupported file formats | 1. Click 'Upload'. 2. Select a .txt file. | notes.txt | Error message: Invalid file format. | Error message displayed. | Pass |
| TCDM05 | Query\_Processing | System should process simple query correctly | 1. Enter query: 'show first 5 rows'. 2. Click 'Submit'. | Query text | First 5 rows should be displayed. | First 5 rows displayed. | Pass |
| TCDM06 | Visualization\_BarChart | User should be able to generate a bar chart | 1. Upload dataset. 2. Select 'Bar Chart'. 3. Provide fields. | X: Category Y: Sales | Bar chart should be displayed. | Bar chart displayed. | Pass |
| TCDM07 | Visualization\_Invalid | Chart should not generate without required fields | 1. Upload dataset. 2. Select 'Bar Chart' without fields. | Empty fields | Error message: Please select fields. | Error message displayed. | Pass |
| TCDM08 | Save\_Project | User should be able to save a project | 1. Run query & visualization. 2. Click 'Save Project'. | Project Name: Sales Analysis | Project should be saved. | Project saved. | Pass |
| TCDM09 | Export\_Report | User should be able to export | 1. Run analysis. | Click Export | PDF report should download. | PDF downloaded. | Pass |

# *System Screens*

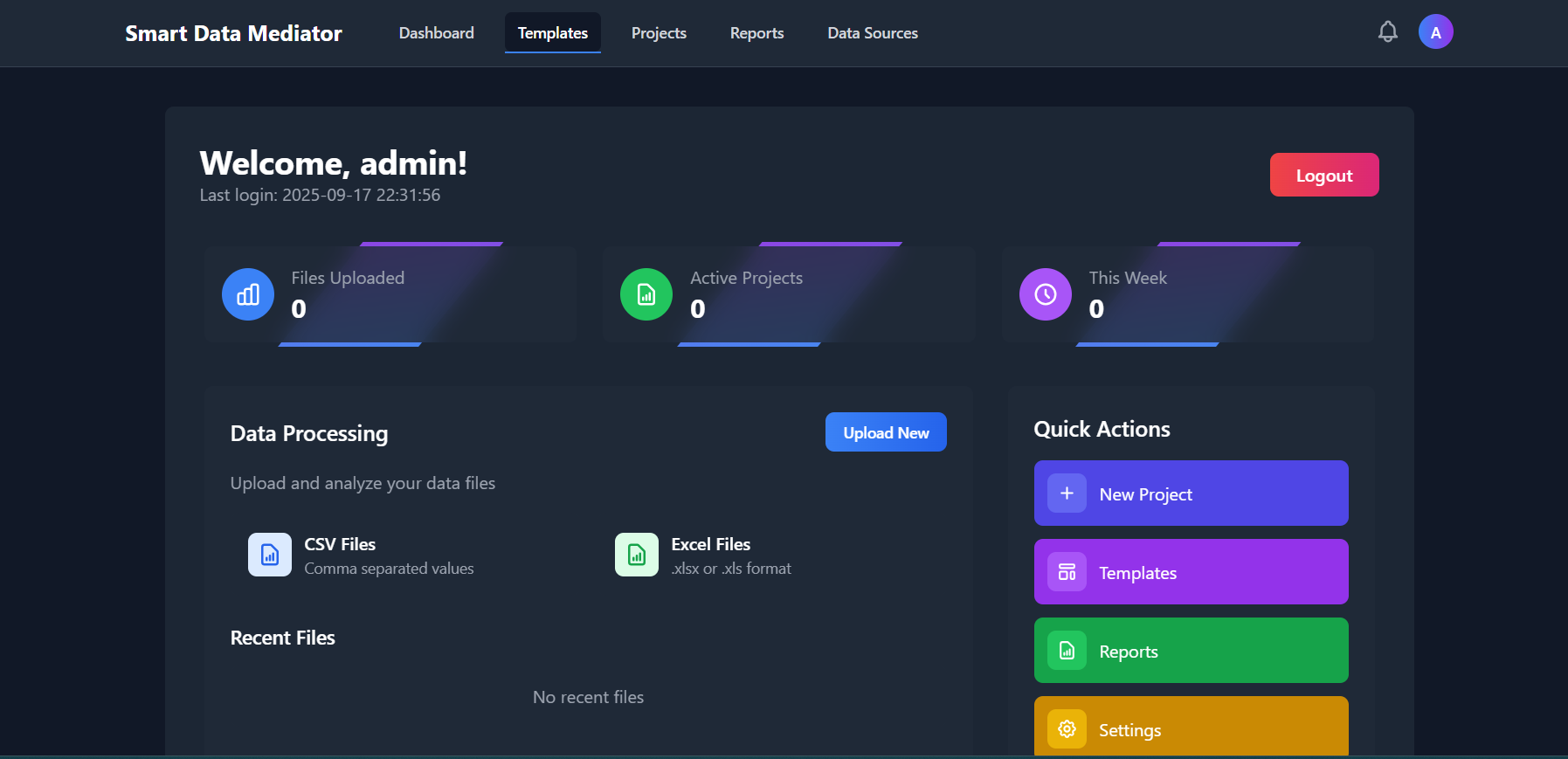
The following screenshots illustrate the Smart Data Mediator interface:

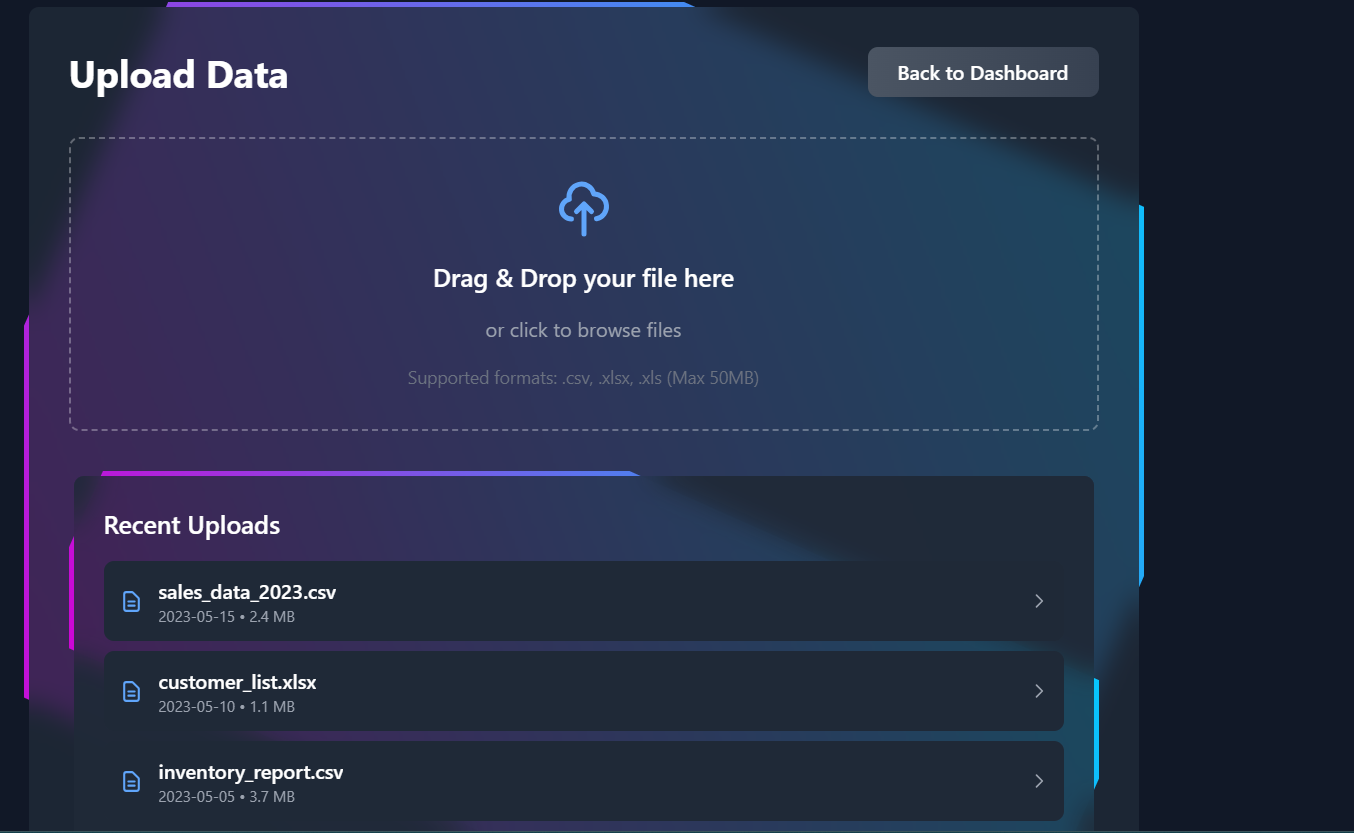
**Input screens:**

**Starting of the sever:**

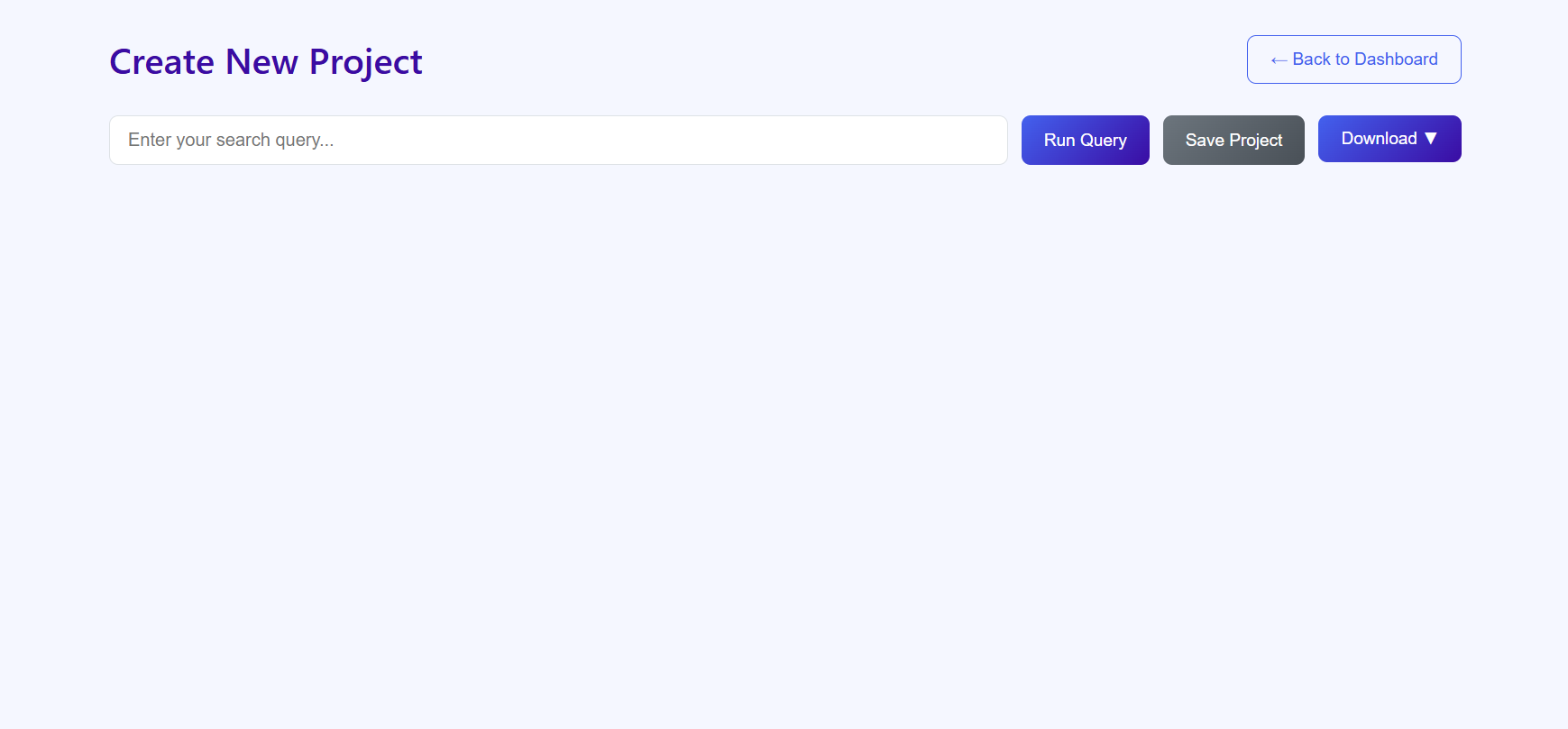
**Login Page:**

**Dashboard:**

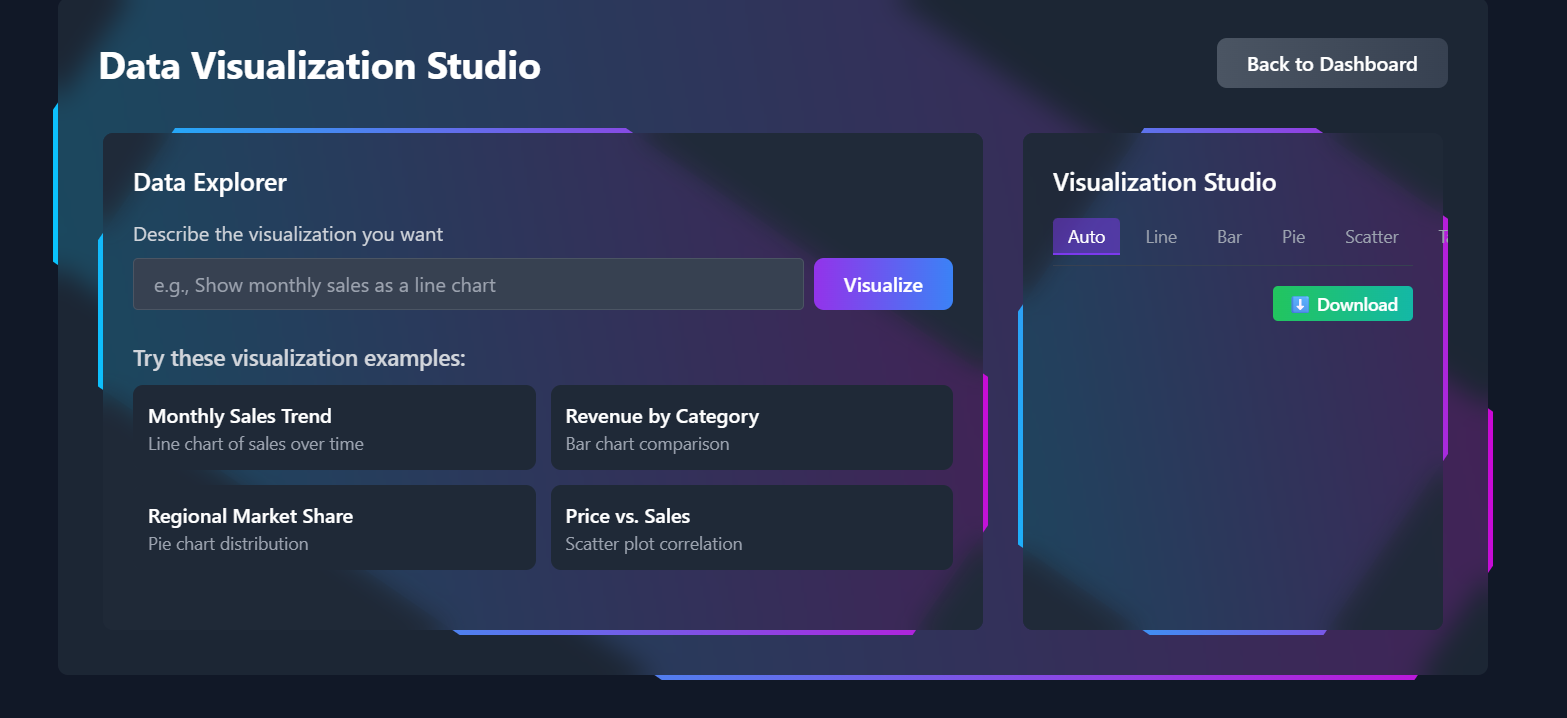
****

**Upload File:**

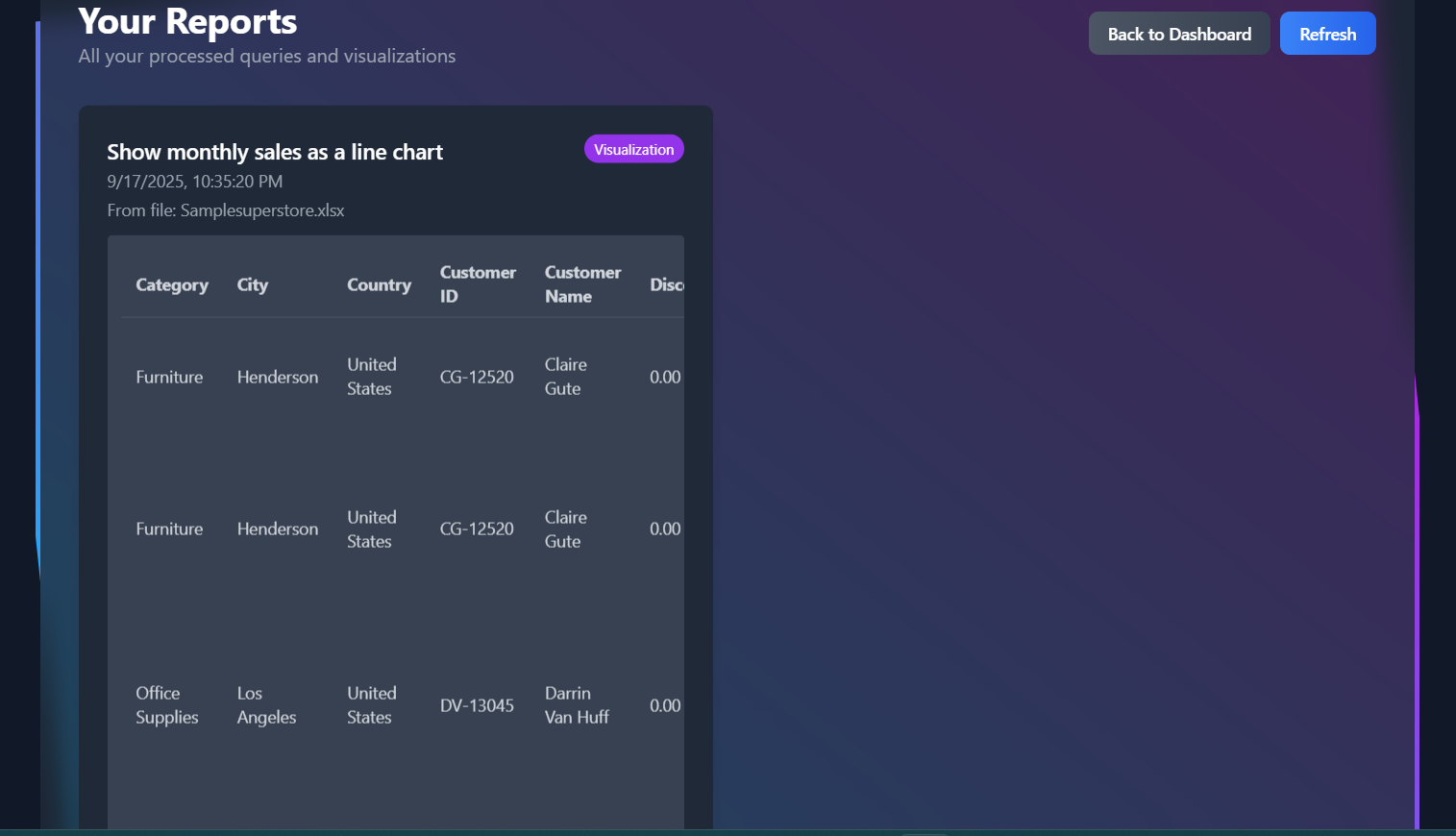
**New Project Query Processing:**



**Visualization Process:**

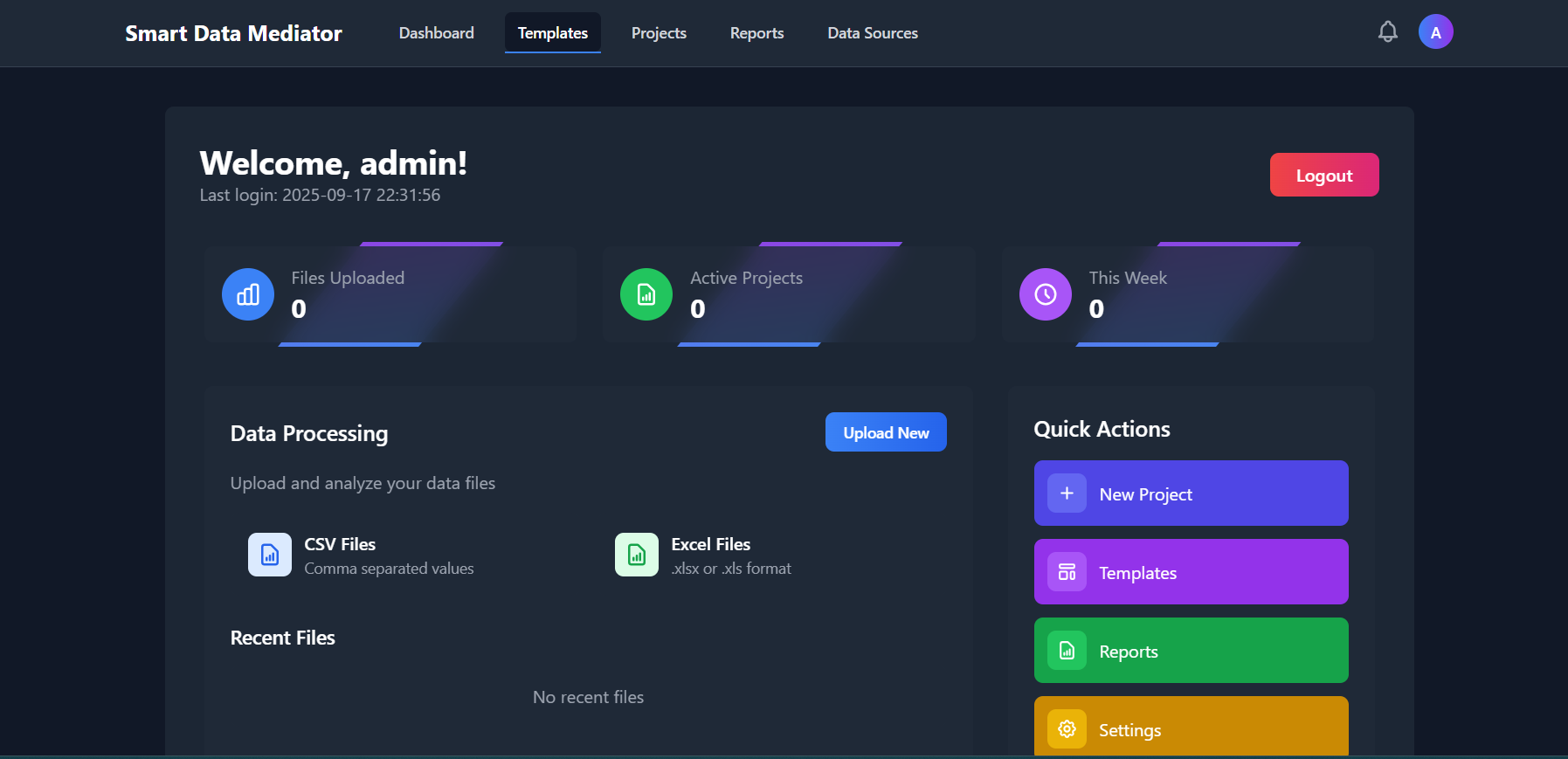
****

**Reports Saving:**

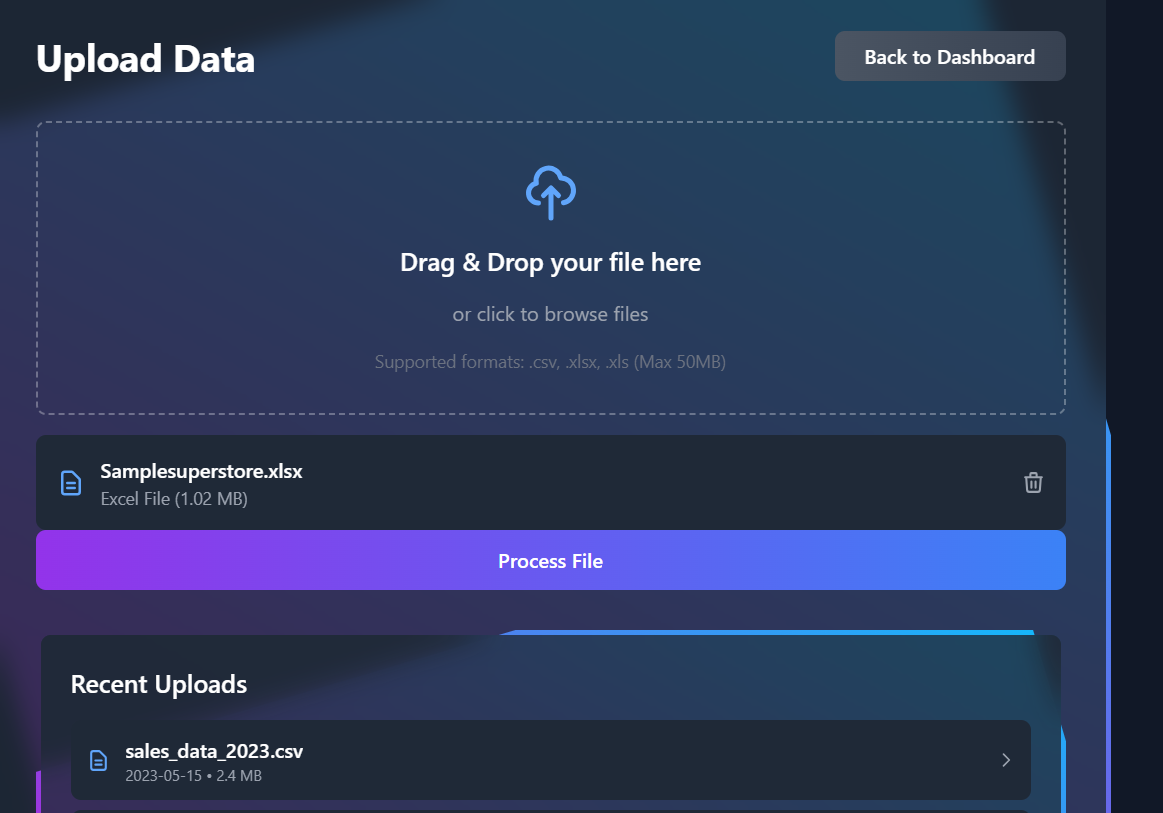


**Output screens:**

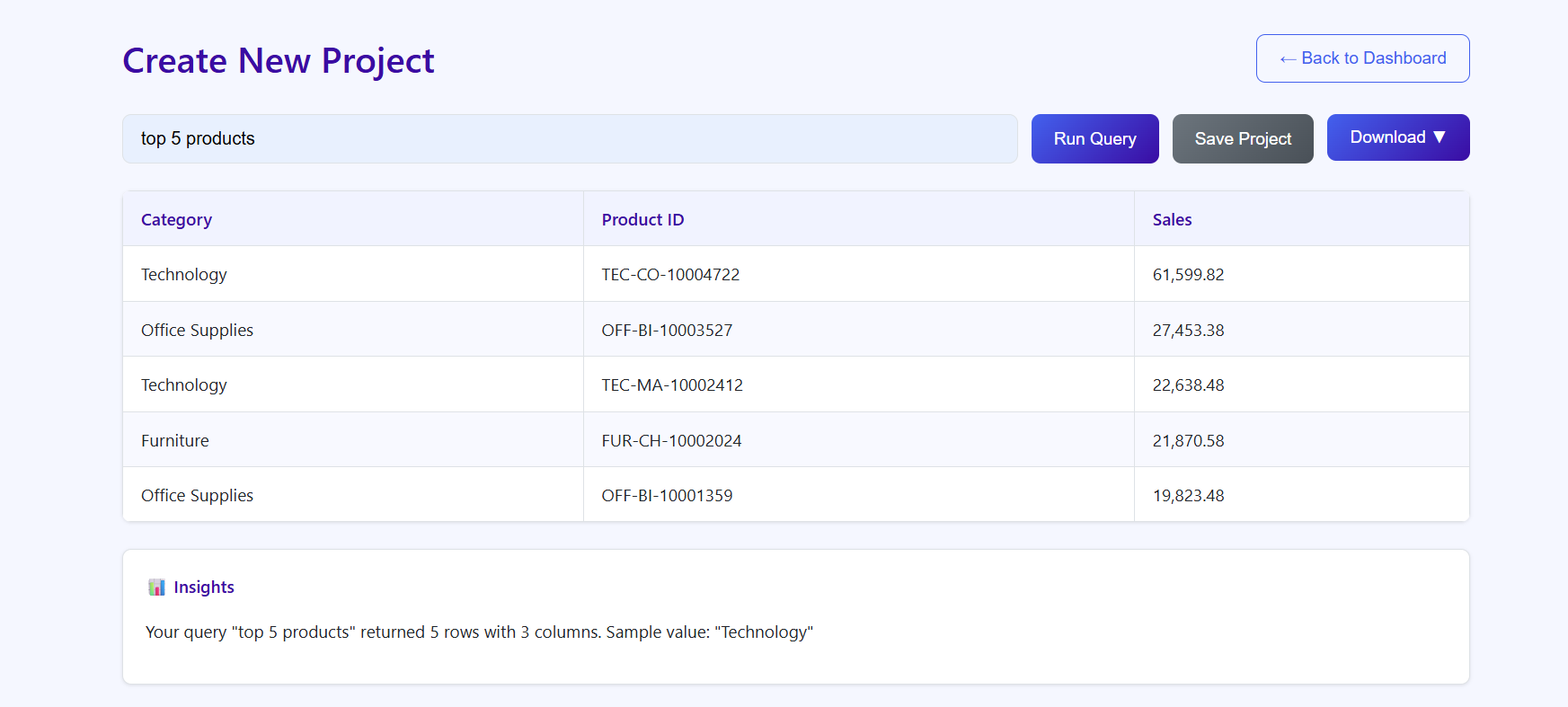
**Dashboard:**

****

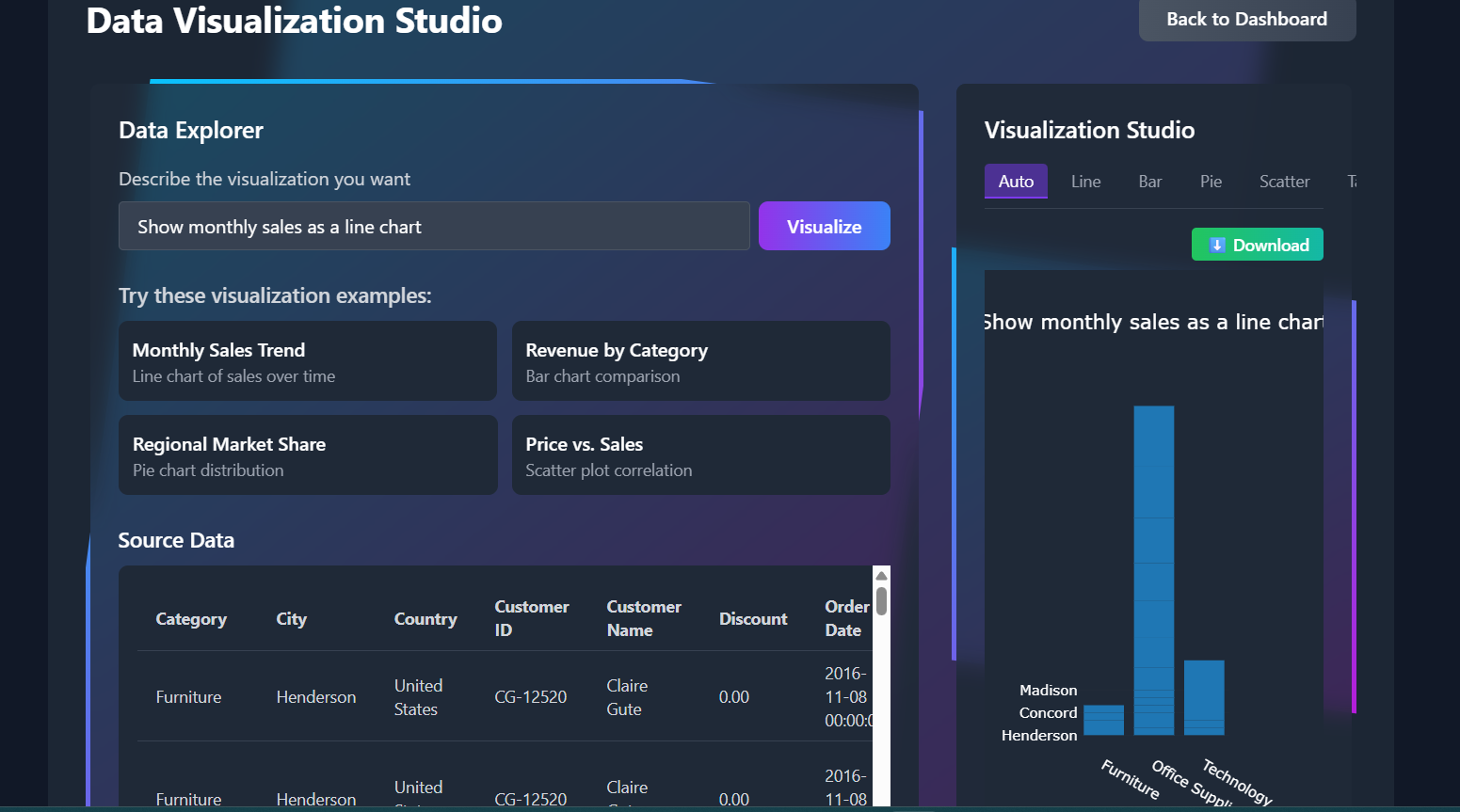
**File Uploaded:**



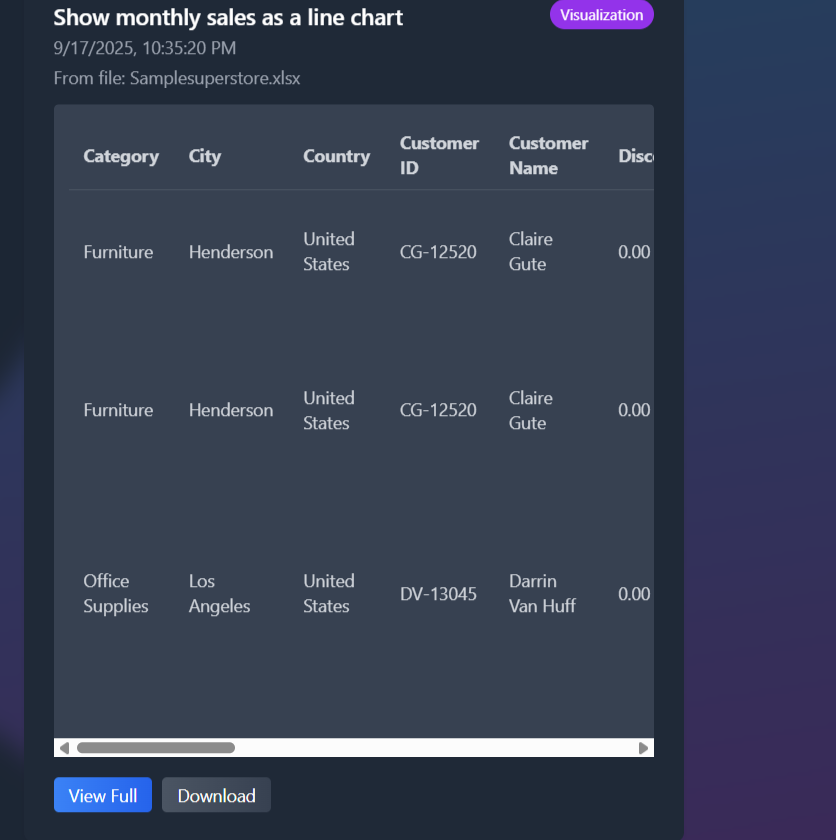
**Query Result:**

****

**Visualization Result:**

****

Reports Download {CVS, Excel, PDF}:



# System Testing

The system has been tested using functional and validation test cases:  
1. Login validation  
2. File upload testing (CSV/Excel)  
3. Query processing validation  
4. Report saving and retrieval  
5. Export functionality (CSV, Excel, PDF, HTML)

# Limitations and Future Enhancements

**Limitations:**  
- NLP understanding is limited to predefined patterns.  
- Reports are stored locally, not on cloud.  
  
**Future Enhancements:**  
- Cloud storage integration  
- Advanced AI query understanding  
- Real-time collaboration  
- Role-based access control  
- Multiple dataset merging and linking

# Bibliography

- Flask Documentation  
- Pandas Documentation  
- SpaCy NLP Toolkit  
- ReportLab PDF Toolkit  
- OpenPyXL Excel Toolkit

***THANK TOU…***