Harvard Artifacts Explorer

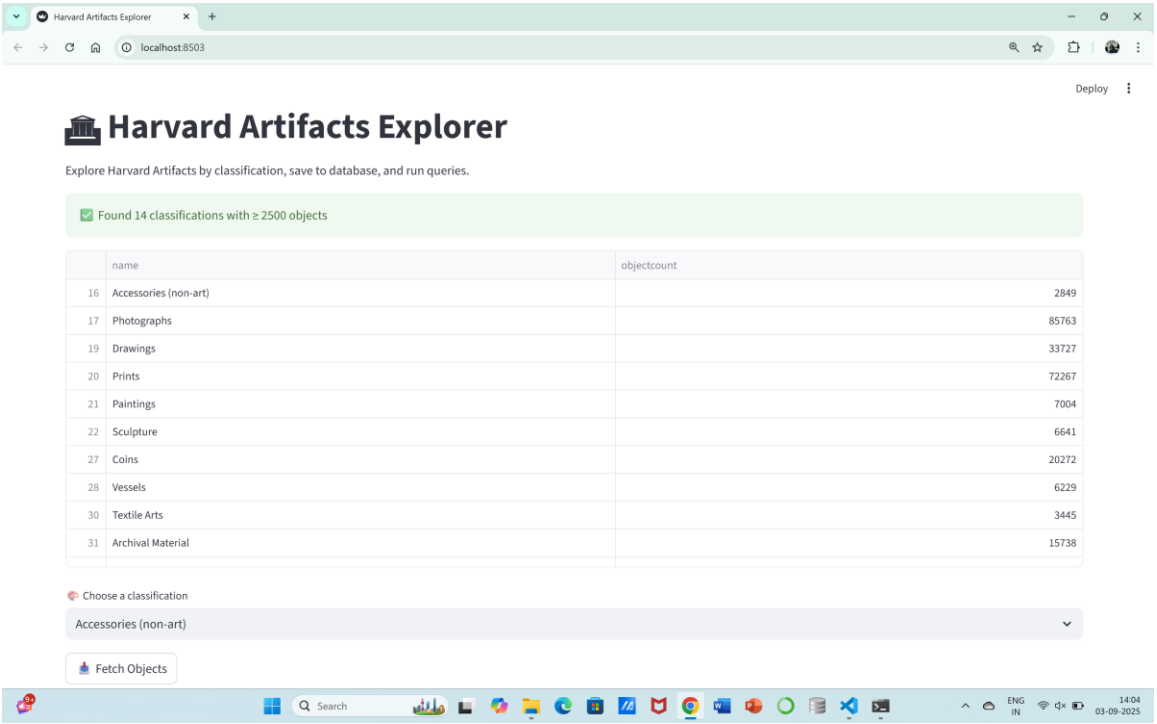
Author: Anubriya Baskaran

Project: Data Exploration and SQL Queries using Streamlit and SQLite

Date: September 2025

1. Introduction

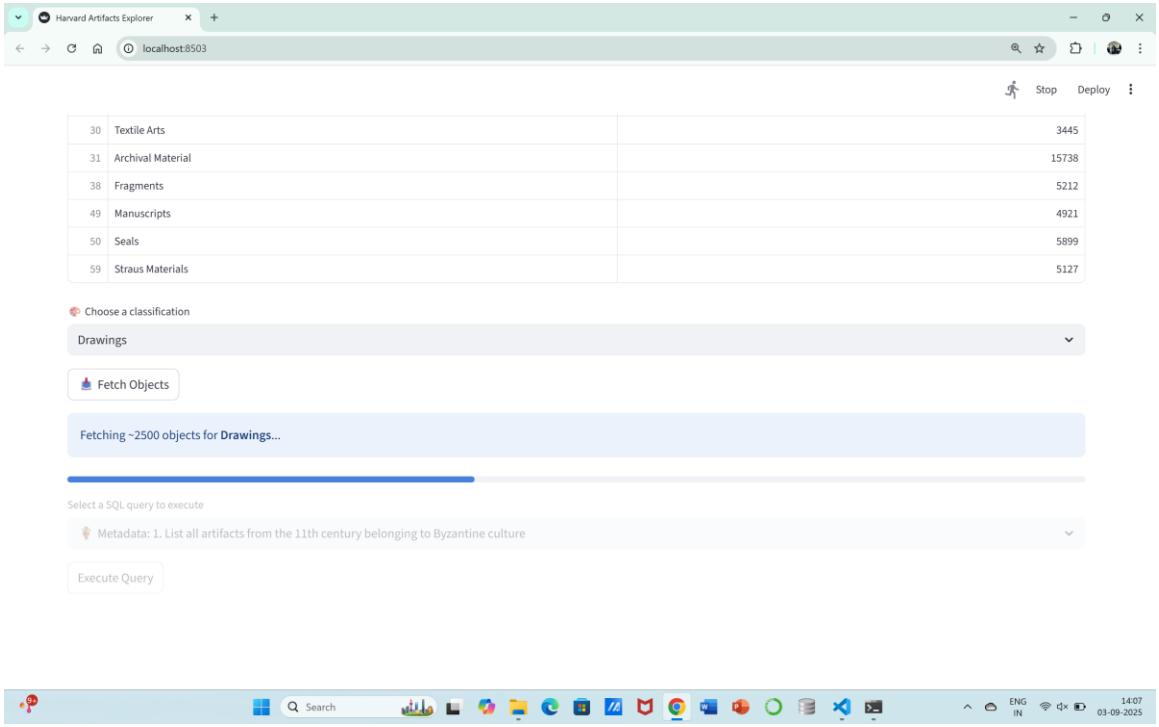
The Harvard Artifacts Explorer project is a data-driven application built using Python, Streamlit, SQLite, and the Harvard Art Museums API. The goal of the project is to fetch, store, and explore artifact data from the Harvard Art Museums. Users can interactively choose artifact classifications, fetch corresponding objects, save the data into a database, and run SQL queries to analyze the collection.



1.1. Classifications Greater than or Equal to 2500 objects

2. Objectives

- To integrate and fetch real-time artifact data from the Harvard Art Museums API.
- To filter and store artifact data into structured SQLite tables.
- To enable interactive exploration of metadata, media, and color information.
- To design and execute SQL queries for gaining insights into the dataset.
- To provide a simple, beginner-friendly, and interactive data exploration tool.



2.1. Fetching the desired classification

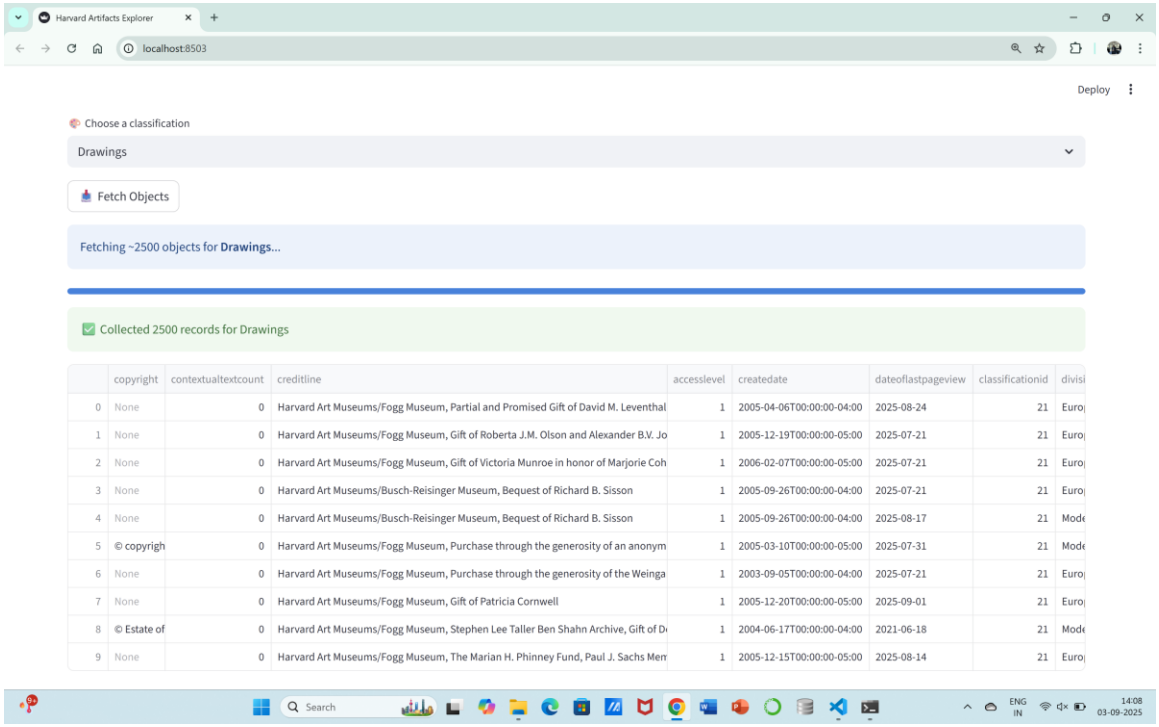
3. Tools and Technologies Used

- Programming Language: Python
- Framework: Streamlit
- Database: SQLite
- API: Harvard Art Museums API
- Libraries: pandas, requests, sqlite3, time
- Documentation Tools: MS Word, MS PowerPoint

4. System Architecture

1. Data is fetched from the Harvard Art Museums API.
2. Data is filtered to select only classifications with more than 2500 objects.
3. Selected classifications are explored by fetching artifacts (metadata, media, colors).

4. Data is stored in SQLite tables (artifact_metadata, artifact_media, artifact_colors).
5. Users interact via Streamlit interface to insert, query, and visualize data.



4.1. Fetched data with progress bar

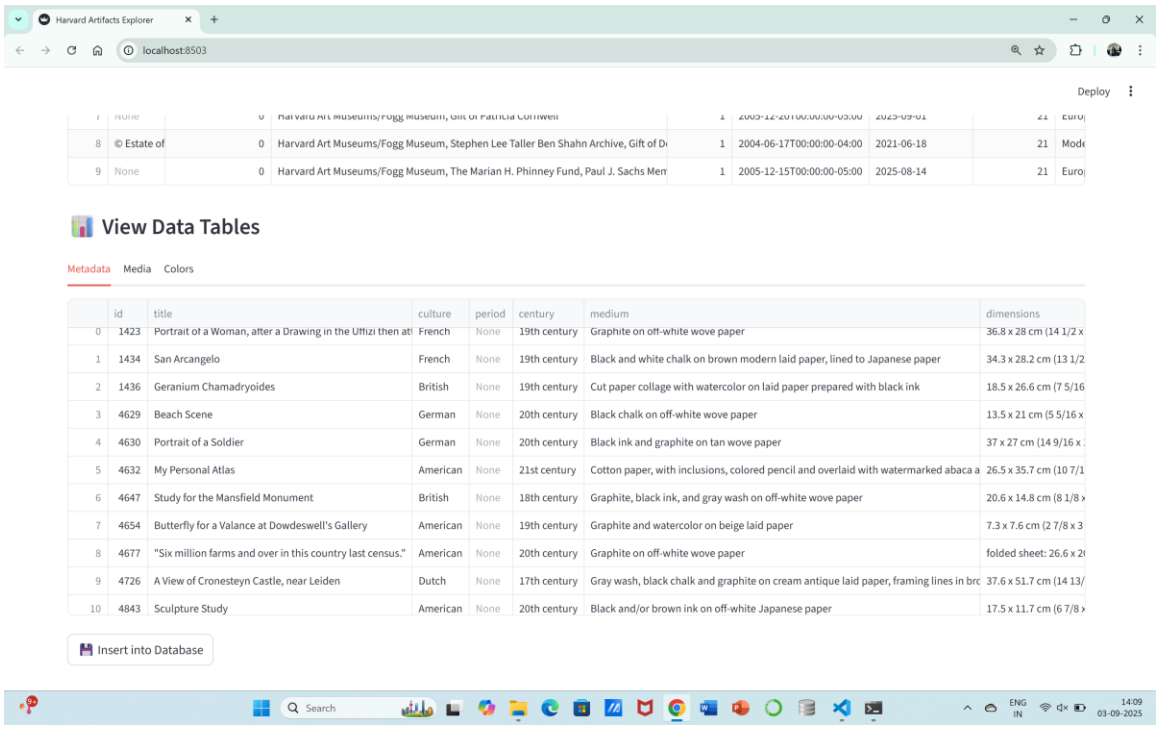
5. Database Design

The project uses three tables for structured storage:

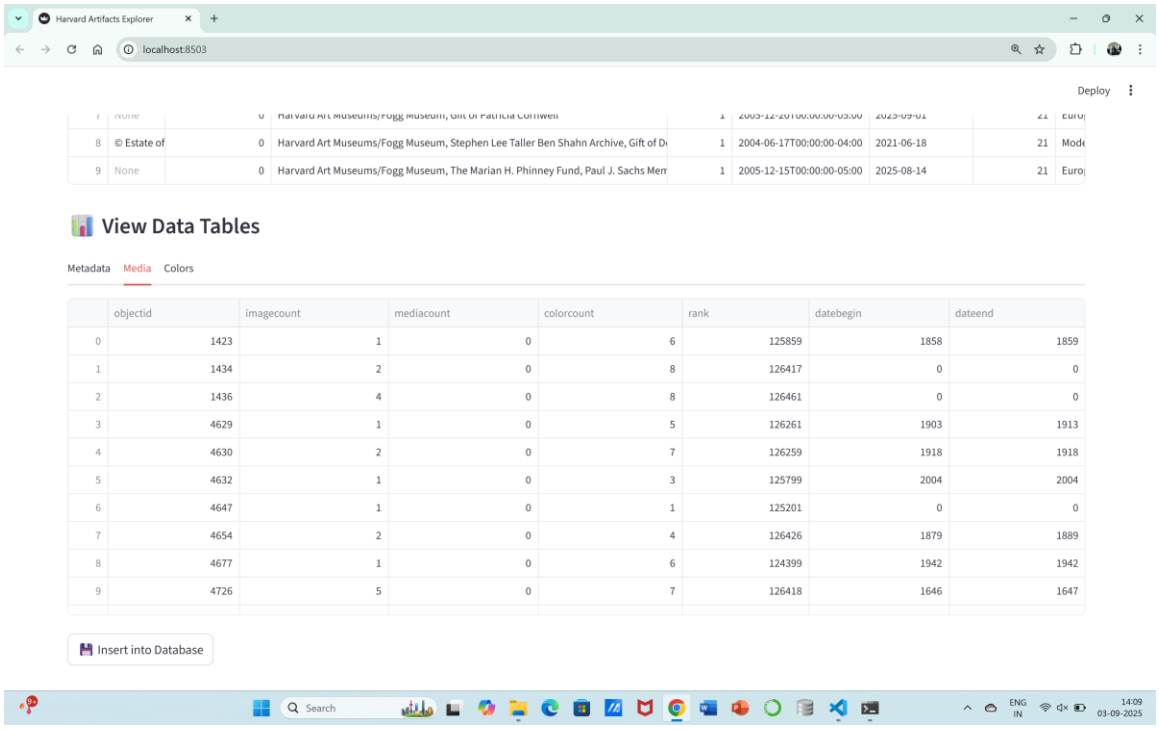
• artifact_metadata: Stores general metadata such as title, culture, period, century, medium, department, classification, etc.

• artifact_media: Stores media-related data such as imagecount, mediacount, colorcount, rank, and date ranges.

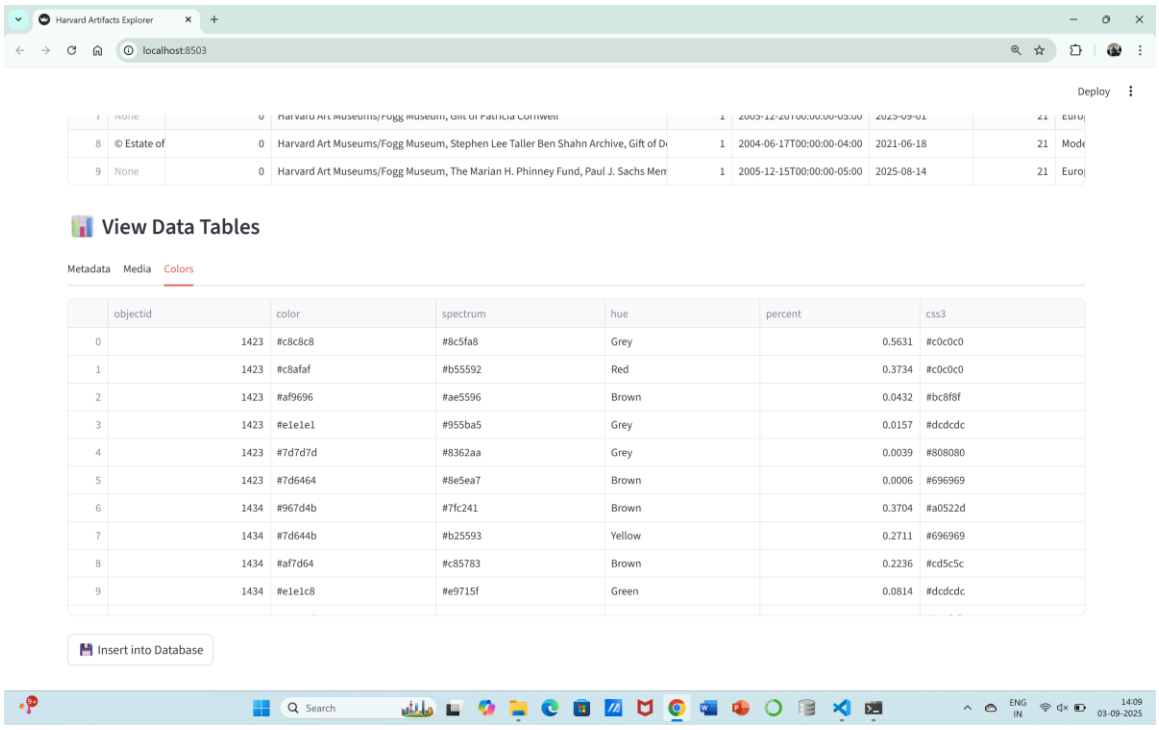
• artifact_colors: Stores extracted colors, hues, coverage percentages, and CSS3 representations.



5.1. Metadata Table



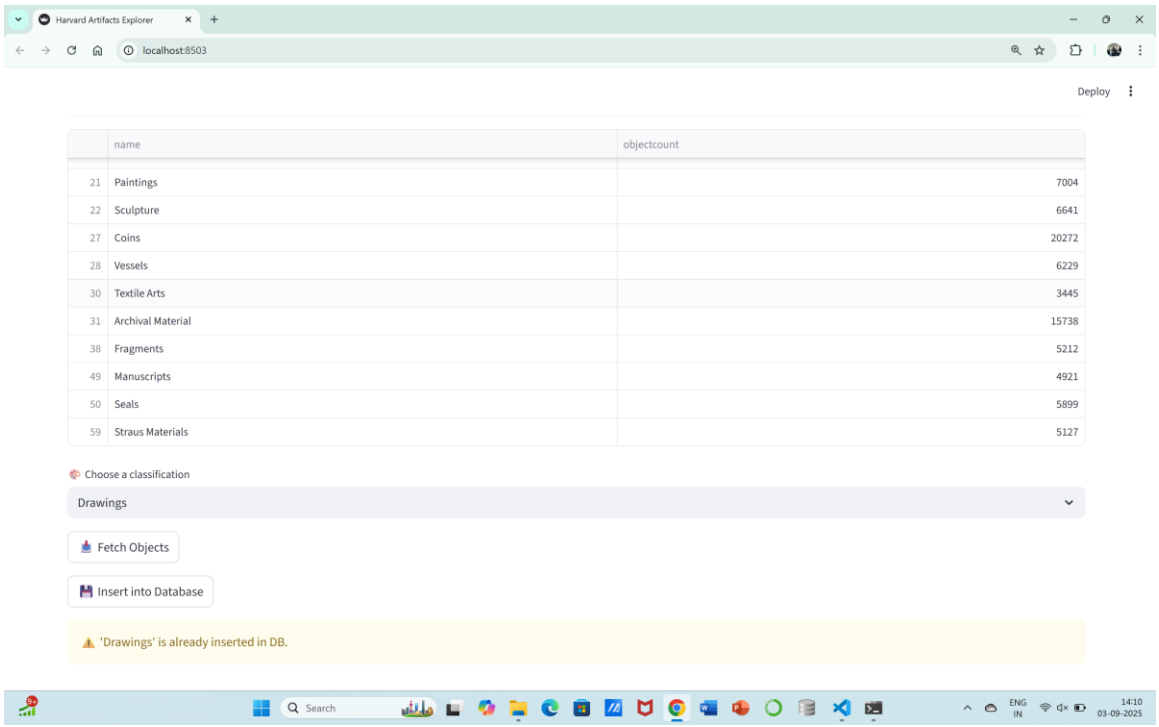
5.2. Media Table



5.3. Colors Table

6. Implementation

The application is implemented in Python with Streamlit providing an interactive frontend. The workflow includes fetching data, splitting it into structured DataFrames, inserting into SQLite, and providing query execution options. A dropdown-based query explorer allows users to select one of 30 SQL queries (20 predefined + 10 DIY) and view results instantly.

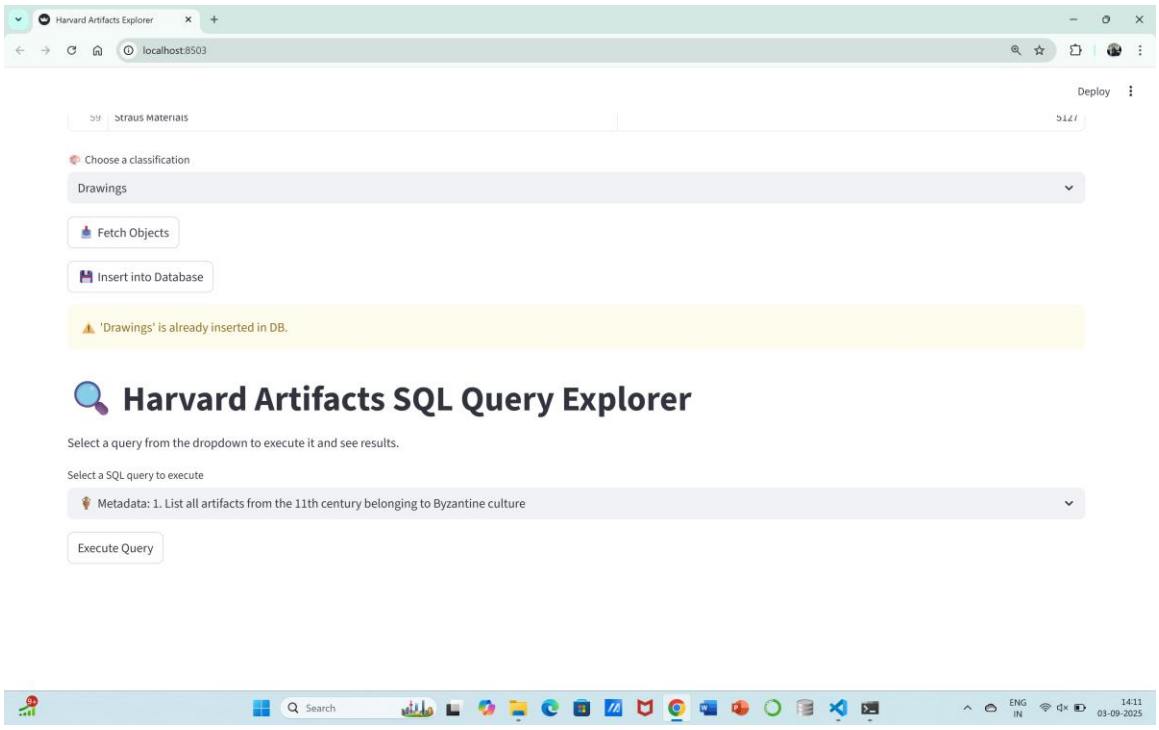


6.1. Inserting into the Database

7. Sample SQL Queries

Examples of SQL queries implemented in this project include:

- List all artifacts from the 11th century belonging to Byzantine culture.
- Show unique cultures represented in the dataset.
- Find the average rank of all artifacts.
- Show top 5 most frequently used colors.
- Count artifacts per department.
- List artifact titles and hues for Byzantine culture (join query).

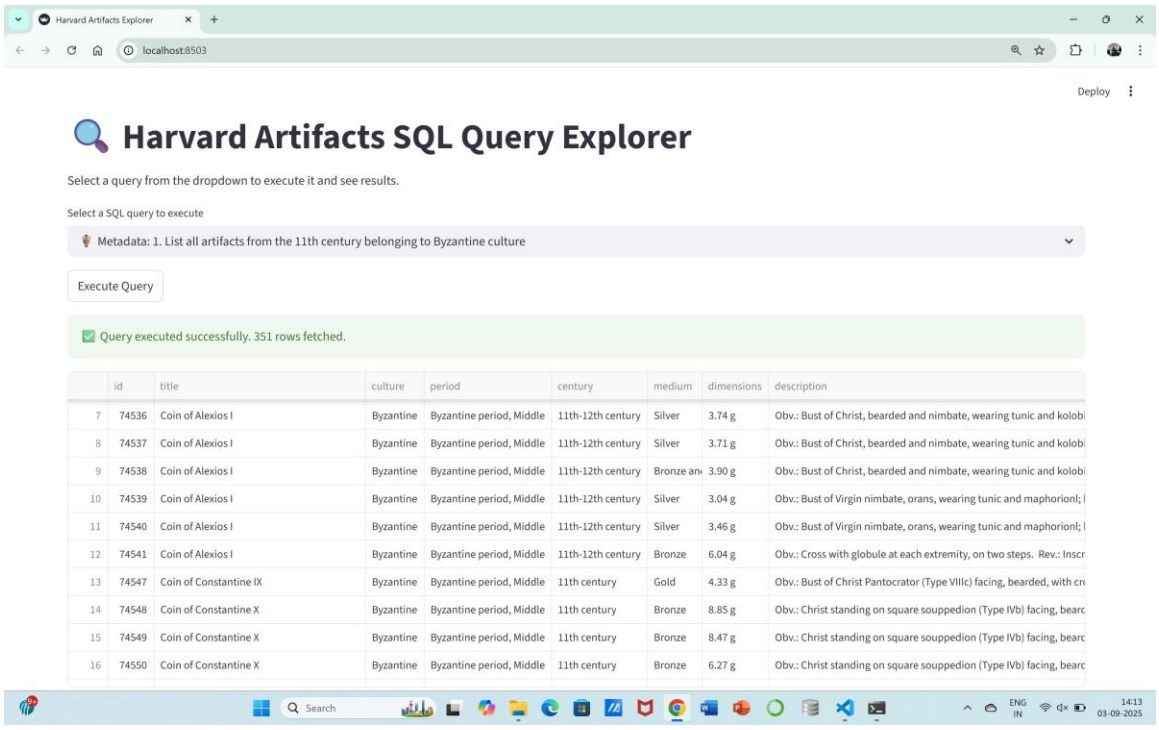


7.1. SQL Query Explorer to choose desired query

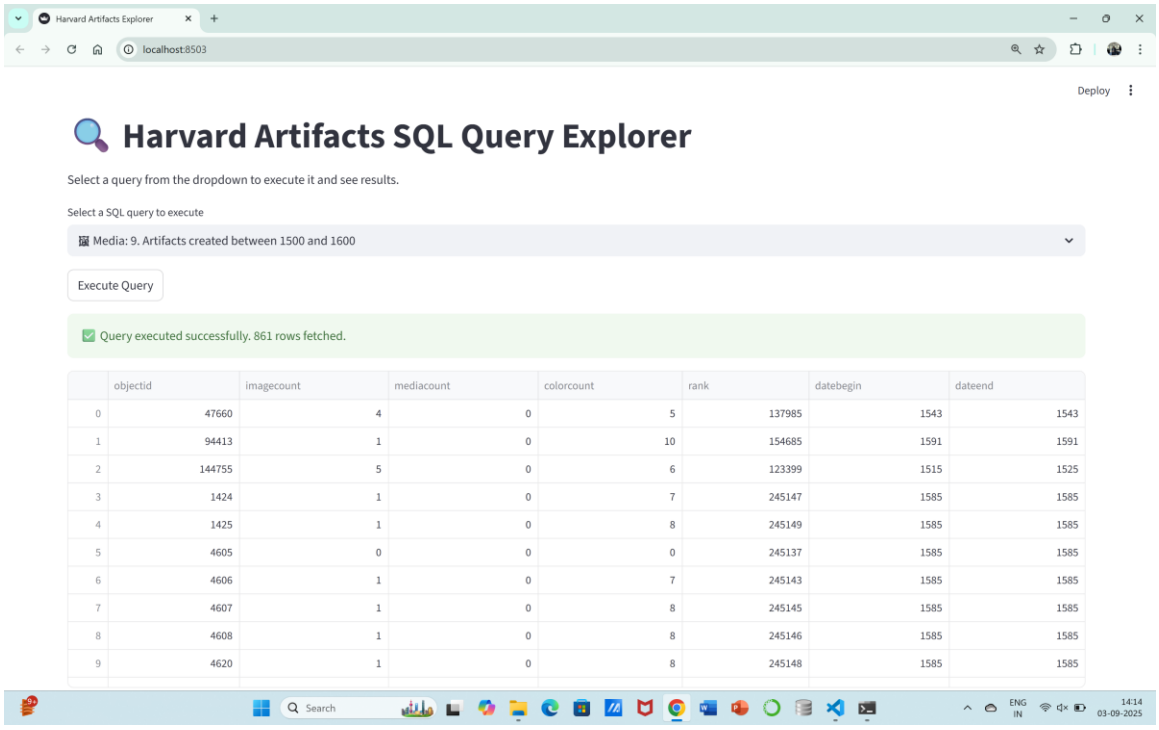
8. Results and Analysis

The project successfully demonstrates how real-world datasets can be fetched, stored, and analyzed interactively.

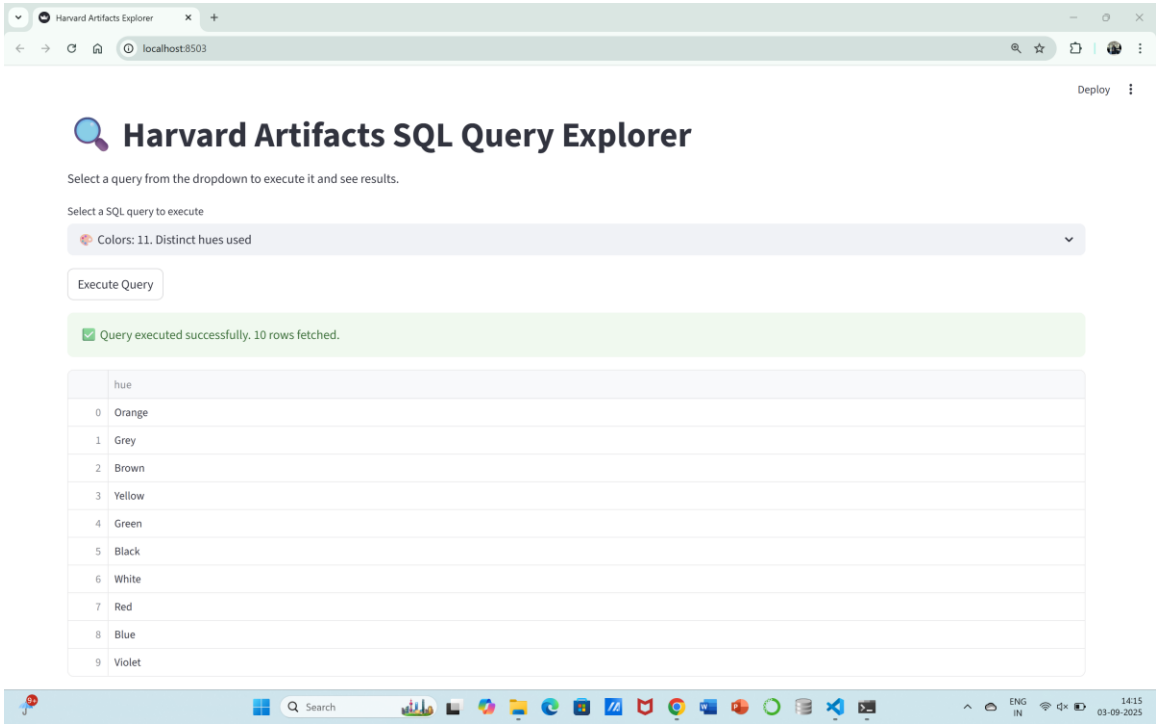
Key findings include:
- The database contains thousands of artifacts across multiple cultures, centuries, and classifications.
- Byzantine and Egyptian artifacts are richly represented.
- Colors like Grey, Red, and Blue dominate in artifact imagery.
- Join-based queries reveal detailed relationships between metadata, media, and color usage.



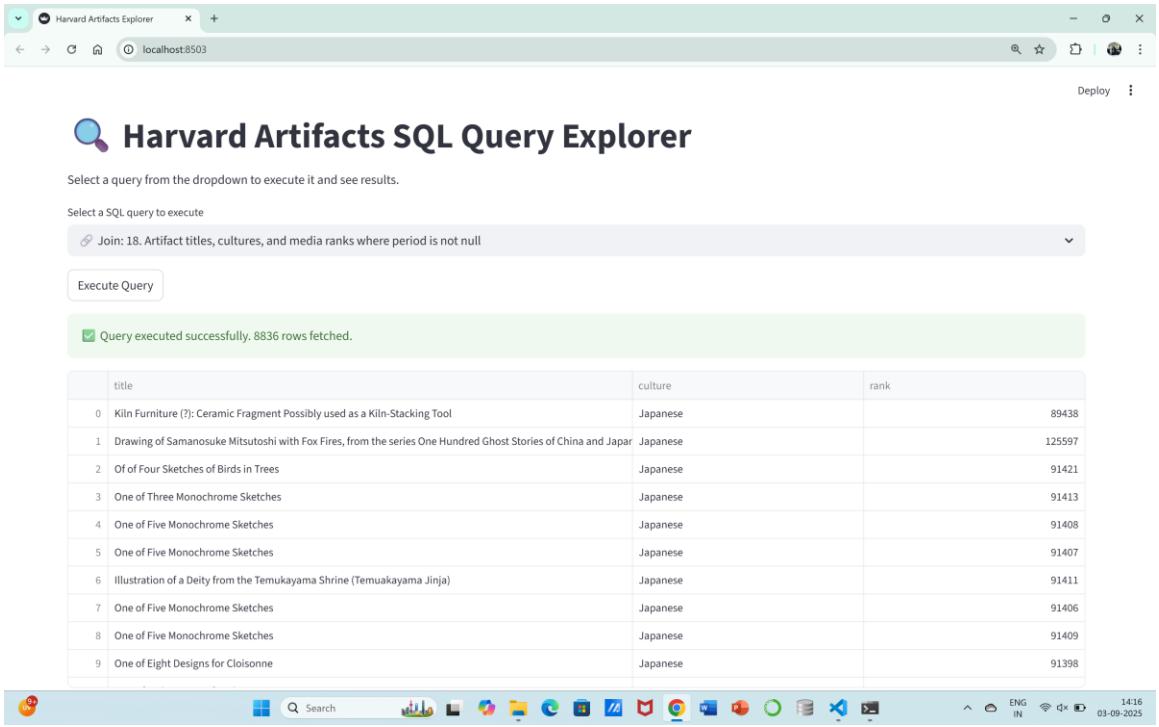
8.1. Example query from Metadata



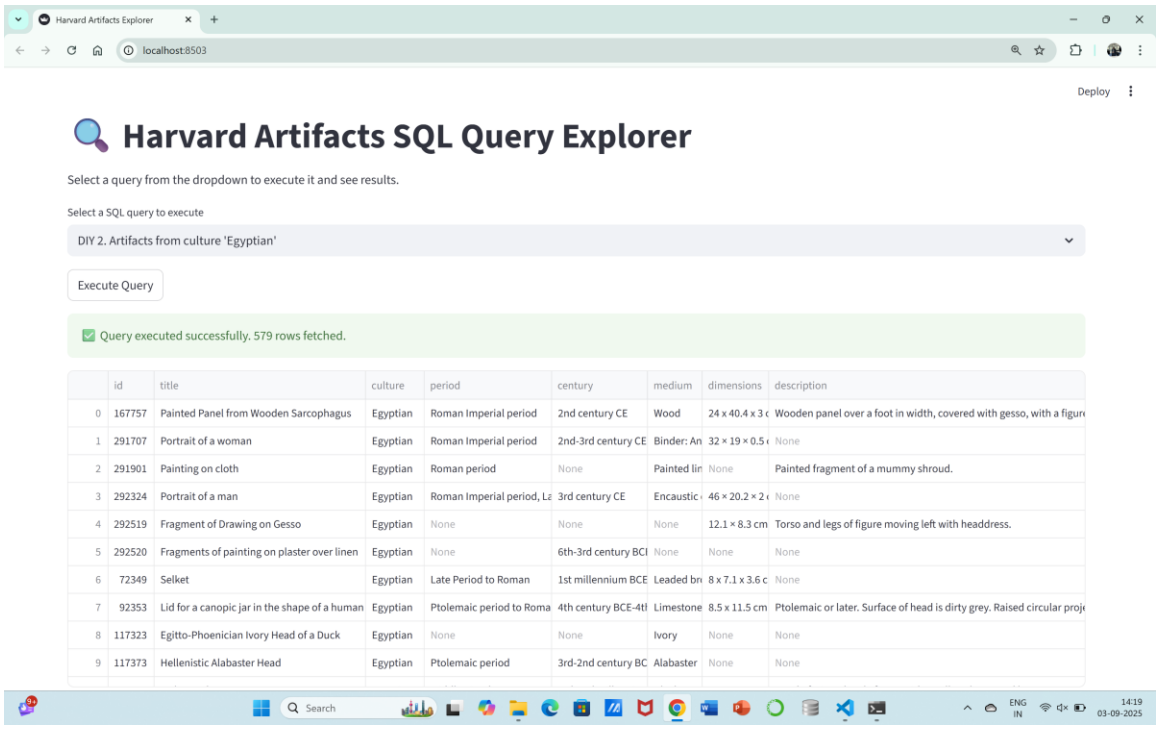
8.2. Example query from Media



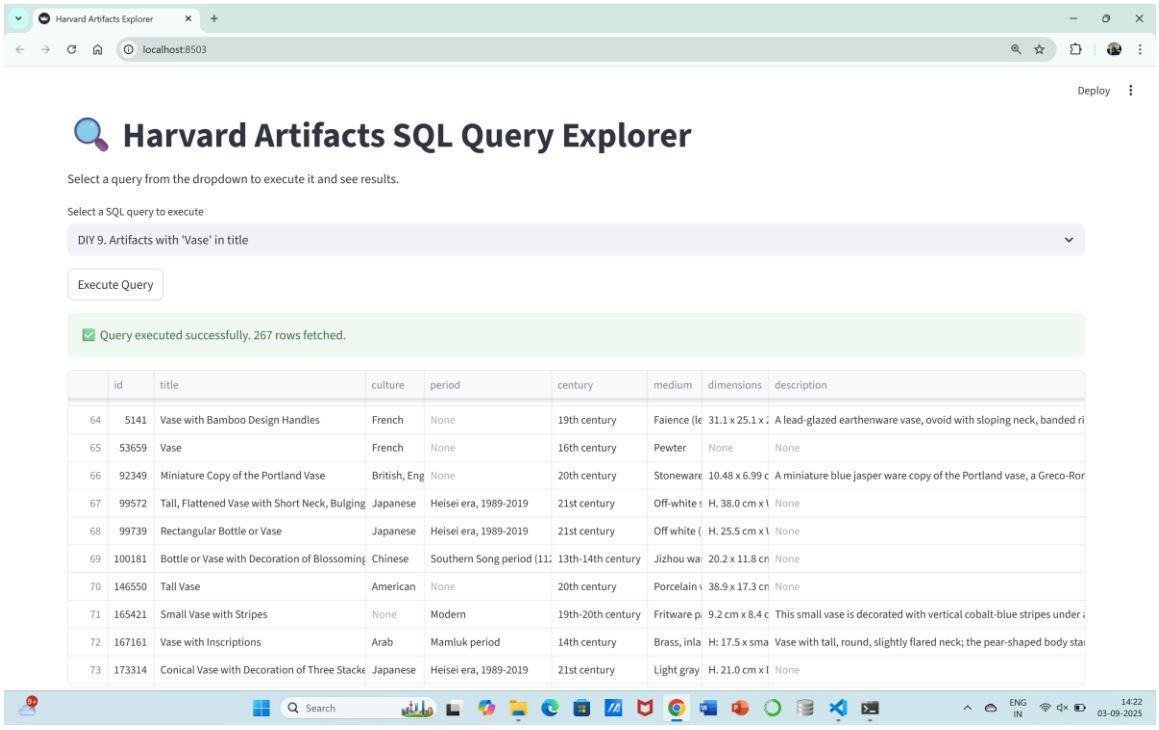
8.3. Example query from Colors



8.4. Example Join query



8.5. DIY Example query



8.6. DIY Example query

9. Conclusion

The Harvard Artifacts Explorer provides an end-to-end solution for exploring museum collections with modern data science tools. It highlights the importance of structured storage, interactive visualization, and query-based insights in cultural data analysis. This project can be extended further with advanced visualizations, deployment, and predictive analytics.

10. Future Work

- Deploying the Streamlit app on the cloud for public access.
- Enhancing visualizations using charts and graphs.
- Adding Natural Language Processing (NLP) for querying in plain English.
- Expanding dataset coverage beyond 2500-object classifications.
- Integrating AI/ML for pattern detection and predictive insights.