# Making Astronomical Data User-Friendly: An Interface for the Sloan Digital Sky Survey

CS4991 Capstone Report, 2025

Celestine Nguyen
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
haw3wk@virginia.edu

## ABSTRACT

Accessing the Sloan Digital Sky Survey (SDSS) data is essential for astronomers and researchers, but the reliance on complex SQL queries presents a significant barrier to non-technical users. To address this issue, I propose utilizing human-computer interaction (HCI) design principles to develop a user-friendly platform that simplifies data retrieval from the SDSS database. The platform will integrate frontend design principles with backend SQL handling to create an easy and streamlined experience for users, allowing them to interact with SDSS data without needing to understand SQL. The development process will follow Agile/Scrum methodologies to ensure iterative improvements and flexibility. Anticipated outcomes include a more accessible interface, increased engagement with SDSS data, and a user-friendly approach to complex astronomical research. Future work involves refining the platform's features, gathering user feedback, and expanding its capabilities to support advanced queries and broader applications.

## 1. INTRODUCTION

The Sloan Digital Sky Survey (SDSS) is one of the most prominent astronomical databases used by astronomers and researchers worldwide. It is responsible for fundamental work done across various astronomical disciplines. The main method for accessing this data is through utilizing Structured Query Language (SQL). This requires researchers to be somewhat proficient in SQL, which can be difficult for new and non-technical users. This limitation restricts access to important astronomical data, potentially slowing scientific discovery.

To address this issue, I propose development of a user-friendly platform that allows researchers to retrieve and analyze SDSS data without requiring direct SQL knowledge. By prioritizing human-computer interaction (HCI) design principles, the platform will provide an intuitive user interface that allows users to translate their queries into SQL commands. This solution is meant to lower the barrier to entry, making astronomical research more accessible to a wider audience, including students and interdisciplinary researchers.

The proposed platform will integrate frontend design best practices with backend database management, allowing for an easy and efficient experience. The project will be developed using Agile/Scrum methodologies, which will allow for iterative enhancements and user-driven improvements. By simplifying the data retrieval process, this project will make access to SDSS data more approachable and help make new discoveries in the field of astronomy.

## 2. RELATED WORKS

There have been many other projects with the goal of improving database accessibility for non-technical users. One example is CasJobs, a web-based SQL workbench designed for large-scale astronomical data analysis (Szalay, et al., 2008). It provides researchers with batch query capabilities and personal database storage to manage and manipulate datasets efficiently. CasJobs is an essential tool for working with large astronomical datasets, such as those from the SDSS. However, CasJobs still requires users to understand SQL syntax, which limits accessibility for researchers without prior programming experience.

Google BigQuery is another relevant system. It provides a graphical user interface for querying large datasets using structured tables, allowing users to execute SQL queries without managing underlying infrastructure (Borra, 2024). While BigQuery simplifies query execution, it is optimized for cloud-based data analysis rather than astronomical research, making it less suitable for SDSS users.

My solution will overcome these limitations by focusing specifically on SDSS data accessibility through HCI driven design. By studying these existing solutions, I can design a platform that will bridge the gap between complex SQL queries and user-friendly interaction to introduce a more intuitive, astronomy-focused interface.

## 3. PROPOSED DESIGN

The proposed platform is designed to provide an intuitive and user-friendly interface for accessing SDSS data without requiring direct SQL knowledge. This section outlines the system architecture, key design considerations, and implementation approach. The design focuses on HCI principles to ensure accessibility for both technical and non-technical users.

### 3.1 System Architecture

The platform consists of three main components: a frontend user interface, a backend processing unit and an API integration with a query history database. These components work together to provide an efficient, accessible and user-friendly approach to retrieving SDSS data.

The frontend interface will be developed using HTML, CSS, and JavaScript, with a framework such as React.js to ensure a dynamic and responsive user experience. This interface will allow users to construct queries through an interactive graphical query builder, using dropdown menus, filters, and guided options. A key feature of the frontend is the Query History section, which enables users to view, modify, and re-run past searches without having to manually reconstruct them. Additionally, real-time query previews will allow users to understand how their selections translate into SQL commands, making the process more transparent and educational.

The backend processing unit will be built using Python and Django, leveraging the Django REST Framework (DRF) to handle communication between the frontend and the SDSS API. When a user submits a query, the backend will first check whether a similar query has been executed recently. If the query exists in the cache, the stored results will be retrieved and displayed immediately, reducing redundant API calls. If the query is new, the backend will convert the user’s selections into an optimized SQL query and send it to the SDSS API for execution. Once the results are retrieved, they will be stored for future reference. To support query history and caching, the platform will integrate a lightweight PostgreSQL or SQLite database to store structured query logs. This feature allows users to track past queries and retrieve previous results efficiently.

By combining a user-friendly frontend, backend, and an efficient data storage approach, the system architecture ensures that SDSS data is accessible to a broader range of researchers. The integration of caching and query history enhances performance while providing users with a seamless way to interact with astronomical datasets.

### 3.2 System Requirements

The system must meet a set of functional and non-functional requirements to ensure usability, security, and efficiency. The functional requirements define the core capabilities of the platform. First, the system must provide an intuitive graphical user interface (GUI) that allows users to construct and execute queries without writing SQL manually. This interface should include dropdown menus, filters, and visual query-building tools to facilitate interaction with the SDSS database. Additionally, the platform must support a query history feature, enabling users to save, review, and re-execute previous queries. The backend should be able to convert user selections into optimized SQL queries and communicate with the SDSS API to retrieve relevant data. Finally, the system must provide data visualization tools, such as tables and charts, to help users interpret query results effectively.

Beyond functional requirements, the platform must also meet several non-functional requirements to ensure reliability and performance. The system must be efficient and scalable, handling multiple simultaneous user queries. Additionally, the platform should be cross-platform compatible, functioning seamlessly on different web browsers and devices without requiring additional software installations.

By following these requirements, the platform will provide a user-friendly, secure, and high-performing environment for researchers accessing SDSS data. These specifications serve as a blueprint for development, ensuring that the final product aligns with both user needs and technical best practices.

### 3.3 Potential Challenges and Solutions

Developing a user-friendly platform for querying the Sloan Digital Sky Survey (SDSS) database presents several challenges, ranging from technical limitations to usability concerns. Identifying these challenges early in the development process allows for strategic solutions to be implemented, ensuring a smooth and efficient user experience.

One of the primary challenges is translating user-friendly inputs into optimized SQL queries. Since the platform is designed for users who may not have prior SQL knowledge, the system must accurately interpret their selections and generate well-structured SQL queries. A potential issue arises when complex queries require joins, aggregations or nested subqueries, which may not be straightforward to construct automatically. To address this, the backend will implement a query translation engine that uses predefined query templates and an intelligent query parser. By structuring queries dynamically based on user inputs, the system can generate SQL queries that are both efficient and accurate.

Another challenge is performance optimization when handling large astronomical datasets. The SDSS database contains vast amounts of data and inefficient query execution could lead to long response times or server timeouts. To mitigate this, I could try implementing caching mechanisms using Redis or Memcached to store frequently-accessed query results. Additionally, query optimization techniques such as indexing strategies and pagination will be incorporated to improve database efficiency and reduce unnecessary data retrieval.

A usability-related challenge is ensuring that the graphical user interface (GUI) remains intuitive yet powerful. While simplifying the query-building process is essential, it is also important to allow advanced users to construct more complex queries if needed. To achieve this balance, the interface will offer both a basic mode, which provides guided selections and dropdowns, and an advanced mode, where users can directly modify the generated SQL before execution. Additionally, a real-time query preview feature will help users understand how their selections translate into SQL, bridging the gap between simplicity and flexibility.

## 4. ANTICIPATED RESULTS

The anticipated results of this platform focus on improving accessibility, efficiency, and usability for researchers querying SDSS data. By eliminating the need for direct SQL knowledge, the system will significantly lower the barrier to entry for users unfamiliar with database query languages. The graphical query builder, coupled with real-time query previews, is expected to streamline the search process, reducing the time required to construct and execute queries. Researchers will be able to retrieve and analyze SDSS data more intuitively, making astronomical research more inclusive for both experienced data analysts and those new to database interactions.

## 5. CONCLUSION

From a broader perspective, the proposed platform is expected to increase productivity for researchers and institutions utilizing SDSS data. The automation of query generation and execution will reduce errors, improve data retrieval accuracy, and allow researchers to focus on analysis rather than query syntax. In the long term, this system has the potential to serve as a model for other large-scale astronomical databases, demonstrating how intuitive interfaces and optimized query execution can enhance research accessibility and efficiency.

Beyond its immediate applications, this project focuses on the importance of HCI principles in scientific data retrieval, bridging the gap between complex database management and user-friendly design. By integrating an interactive graphical query builder, caching mechanisms, and real-time query previews, the platform not only improves accessibility but also optimizes performance for large-scale astronomical datasets. Additionally, the development process has reinforced the value of iterative design and Agile methodologies in building adaptable, user-driven software solutions. Ultimately, this project represents a step toward making astronomical data more accessible and actionable, empowering a wider audience to engage with scientific discovery.

## 6. FUTURE WORK

While the proposed platform lays the foundation for a more accessible and user-friendly approach to querying SDSS data, there are several opportunities for future enhancements and expansions. One key area for improvement is refining the query-building interface to support more complex and customizable queries. While the current design simplifies standard query construction, advanced users may benefit from additional features such as custom parameterization, nested queries, and broader filtering options. Integrating machine learning techniques for query optimization could further enhance performance by predicting user intent and suggesting the most efficient query structures.

Another important area of development involves expanding the platform to accommodate additional astronomical databases beyond SDSS. By designing a more flexible backend architecture, the system could be adapted to work with datasets from other major astronomical surveys, such as the Gaia mission or the Large Synoptic Survey Telescope (LSST). This expansion would make the platform a more comprehensive tool for researchers working with large-scale astronomical data.

User feedback and usability testing will also be important in the next phase of development. Conducting trials with astronomers, students, and interdisciplinary researchers will help refine the interface, optimize query generation, and ensure the system effectively meets user needs. Additionally, exploring integration with data visualization tools and interactive analysis features could further streamline research workflows by allowing users to manipulate and interpret query results in real time. By addressing these future challenges, the platform can continue evolving into a powerful tool for the astronomical community, increasing access to complex datasets and aiding in new discoveries in the field.

## REFERENCES

Borra, P. (2024). An overview of cloud data warehouses: Amazon Redshift (AWS), Azure Synapse (Azure), and Google BigQuery (GCP). *International Journal of Advanced Research in Computer Science, 15*(3), 23-27.

Szalay, A., Thakar, A., Fekete, G., Gray, J., & Kunszt, P. (2008). CasJobs and MyDB: A batch query workbench. *Computing in Science & Engineering, 10*(1), 30–37. https://doi.org/10.1109/MCSE.2008.10