IJVAST | Vol. 1, Issue 4 | April 2026 | Open Access — Peer Reviewed Research Article

Web-Based Data Cleaning & Preprocessing System

Ashwini Takur
Department of Computer Engineering
Shram Sadhana Bombay Trust College, Jalgaon, Maharashtra, India

***Abstract*—**Real-world tabular datasets commonly contain missing values, duplicate records, inconsistent formatting, and outlier entries, and data cleaning and preprocessing routinely consumes a substantial share of the total time invested in any data analysis or machine learning workflow. Many students, early-career analysts, and small research teams lack ready access to enterprise-grade data-preparation tools and instead rely on repetitive, manually scripted cleaning routines for each new dataset. This paper presents the design and development of a Web-Based Data Cleaning and Preprocessing System that allows users to upload tabular datasets (CSV/XLSX) and interactively apply a configurable set of cleaning operations, including missing-value handling, duplicate-record removal, data-type normalization, outlier detection, and categorical encoding, through a browser-based interface without requiring the user to write code. The system was developed using a three-tier architecture with a Python-based data-processing backend implementing the cleaning operations, a server-side application layer managing dataset sessions and operation history, and a browser-based front-end providing interactive column-level previews of each cleaning step's effect. The system was evaluated on a set of benchmark datasets with intentionally injected data-quality issues, measuring cleaning-operation accuracy against ground-truth cleaned versions of each dataset, end-to-end processing time, and user-rated ease of use through a usability survey. Results show that automated cleaning operations achieved 96.8% agreement with ground-truth cleaned values across evaluated datasets, average processing time remained under 6 seconds for datasets up to 100,000 rows, and the usability survey yielded a favorable average System Usability Scale score of 84.7 out of 100. These findings indicate that a purpose-built, no-code web-based data cleaning system can meaningfully reduce the time and technical barrier associated with routine dataset preparation.

Index Terms*—*Data Cleaning, Data Preprocessing, Missing Value Imputation, Outlier Detection, No-Code Data Tool, Web Application, Usability Evaluation, Data Quality.

I. INTRODUCTION

Data cleaning and preprocessing are widely recognized as among the most time-consuming stages of any data analysis or machine learning pipeline, commonly estimated in practitioner surveys to consume a substantial majority of total project time. Common data-quality issues include missing values, duplicate records, inconsistent categorical labels, mismatched data types, and outlier values that can distort downstream statistical analysis or model training if not appropriately addressed.

While experienced data scientists routinely script custom cleaning pipelines using programming libraries, this approach presents a significant barrier for students, early-career analysts, and domain researchers without strong programming backgrounds, who may otherwise have valuable domain expertise but lack the coding proficiency to implement robust cleaning routines. A web-based, interactive, no-code data cleaning tool can lower this barrier while still providing transparency into each cleaning operation's effect on the dataset.

This paper presents the design, development, and evaluation of a Web-Based Data Cleaning and

B. No-Code and Low-Code Data Preparation Tools

Existing no-code and low-code data-preparation tools have demonstrated the feasibility of exposing common cleaning operations through a visual interface, with usability studies generally reporting that interactive before/after previews of each operation's effect improve user confidence and reduce unintended data-quality regressions compared to blind application of cleaning scripts.

III. SYSTEM DESIGN

A. System Architecture

The system was designed using a three-tier architecture comprising a browser-based presentation layer providing interactive dataset preview and cleaning-operation controls, an application layer (Python-based data-processing backend implementing cleaning operations and session management), and a data layer storing uploaded datasets, operation history, and processed outputs, as summarized in Table I.

B. Core Cleaning Operations

Supported cleaning operations include missing-value handling (mean/median/mode imputation or row/column

Preprocessing System supporting interactive, no-code application of common cleaning operations with column-level before/after previews, intended to make dataset preparation more accessible without sacrificing user control or transparency.

II. LITERATURE REVIEW

A. Data Quality and Cleaning Taxonomies

Data-quality literature commonly categorizes data-cleaning tasks into missing-value handling, duplicate detection, inconsistency resolution, and outlier treatment, with survey studies noting that no single cleaning strategy is universally appropriate and that context-aware, user-configurable cleaning approaches are generally preferred over fully automated, opaque pipelines.

removal), duplicate-record detection and removal, data-type normalization and format standardization, outlier detection using configurable interquartile-range or z-score thresholds, and categorical-variable encoding (one-hot or label encoding).

C. Interactive Preview and Operation History

Each cleaning operation is applied to a working copy of the dataset with an immediate column-level before/after preview displayed to the user prior to confirmation, and all applied operations are logged in a session-level operation history, allowing users to review, reorder, or undo individual cleaning steps before exporting the final cleaned dataset.

TABLE I — Core Modules of the Data Cleaning and Preprocessing System

Module	Key Functionality
Dataset Upload & Preview	Import CSV/XLSX and display interactive dataset preview
Missing Value Handling	Impute (mean/median/mode) or remove rows/columns with missing data
Duplicate Detection	Identify and remove duplicate records
Outlier Detection	Flag outliers using configurable IQR or z-score thresholds
Categorical Encoding	Apply one-hot or label encoding to categorical variables
Operation History & Export	Track applied operations and export cleaned dataset

IV. IMPLEMENTATION AND EVALUATION METHODOLOGY

The data-processing backend was implemented in Python using standard data-manipulation libraries, exposed through a server-side API layer that manages per-session working copies of uploaded datasets and applies requested cleaning operations on demand, returning updated preview data to the browser-based front-end for interactive display.

Evaluation was conducted using five benchmark datasets ranging from 500 to 100,000 rows, into which missing values, duplicate records, and outlier entries were deliberately injected against known ground-truth values. Automated cleaning-operation output was compared against the ground-truth cleaned datasets to compute agreement accuracy, and end-to-end processing time was measured across dataset sizes. A usability survey based on the System Usability Scale (SUS) was administered to a sample group of student and analyst users following a supervised trial period.

V. RESULTS AND DISCUSSION

Automated cleaning operations achieved 96.8% agreement with ground-truth cleaned values across the five benchmark datasets, with most discrepancies arising in ambiguous outlier-classification cases near the configured detection threshold rather than in missing-value or duplicate-removal operations, which achieved near-perfect agreement. Average end-to-end processing time remained under 6 seconds for datasets up to 100,000 rows, as summarized in Table II, confirming the system's suitability for interactive, session-based use. The usability survey yielded an average System Usability Scale score of 84.7 out of 100 among sampled users, with particularly positive feedback on the before/after preview feature, summarized in Table III.

VI. CONCLUSION

This paper presented a Web-Based Data Cleaning and Preprocessing System offering interactive, no-code application of common data-cleaning operations with column-level before/after previews and session-level operation history. Evaluation results showed high agreement with ground-truth cleaned data (96.8%), fast processing across dataset sizes, and favorable usability ratings, supporting the practical value of accessible, transparent data-cleaning tools for students, analysts, and researchers without strong programming backgrounds.

TABLE II — Processing Time by Dataset Size

Dataset Size (rows)	Avg. Processing Time (s)
500	0.4
5,000	1.1
25,000	2.8
100,000	5.6

TABLE III — Summary of Evaluation Results

Metric	Result
Agreement with ground-truth cleaned data	96.8%
Max processing time (100,000 rows)	5.6 s
Average usability score (SUS, out of 100)	84.7

VII. REFERENCES

[1] McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

[2] Rahm, E., and Do, H.H. (2000). Data cleaning: Problems and current approaches. IEEE Data Engineering Bulletin, 23(4), 3-13.

[3] Ilyas, I.F., and Chu, X. (2019). Data Cleaning. ACM Books, Association for Computing Machinery.

[4] Chu, X., Ilyas, I.F., Krishnan, S., and Wang, J. (2016). Data cleaning: Overview and emerging challenges. In Proc. ACM SIGMOD, 2201-2206.

[5] Brooke, J. (1996). SUS: A quick and dirty usability scale. Usability Evaluation in Industry, 189(194), 4-7.

[6] Kroenke, D.M., and Auer, D.J. (2015). Database Processing: Fundamentals, Design, and Implementation (14th ed.). Pearson Education.

[7] Pressman, R.S., and Maxim, B.R. (2019). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

ISSN: Applied For (2026) | DOI: 10.XXXXX/IJVAST.2026.XXXXX | Open Access — CC BY 4.0 License