iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80633

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Excel Cleaner: A Web-Based Application for Automated Data Cleaning

Prof. Shital Gujar[1],Dhruv Shikhare[2], Harshal Dalvi[3], Mayur Sanap[4], Abhishek Jadhav[5]
Department of Computer Science and Engineering, Bharat College of Engineering, Badlapur, Maharashtra

Abstract: ***Data cleaning is a crucial stage in data analysis because the quality of input data directly affects the accuracy of results and decision-making. Real-world Excel datasets often contain missing values, duplicate records, inconsistent formatting, and incomplete entries, which make preprocessing difficult and time-consuming when performed manually. Traditional cleaning methods using spreadsheets or scripts are often repetitive, error-prone, and unsuitable for non-technical users. This paper presents Excel Cleaner, a web-based application designed to automate the preprocessing of Excel datasets in an efficient and user-friendly manner. The system follows a client–server architecture in which users upload Excel files through a web interface and receive cleaned outputs in real time. The application performs key cleaning operations such as removal of empty rows, duplicate elimination, and column normalization. It is implemented using React for the frontend, Node.js and Express for backend processing, and the XLSX (SheetJS) library for Excel file handling. Experimental results show that the system provides high accuracy and fast processing for small to medium-sized datasets. The proposed solution reduces manual effort, improves data quality, and offers a scalable foundation for future enhancements such as intelligent data cleaning and visualization.***

Keywords: Data Cleaning, Data Preprocessing, Excel File Processing, Web Application, Automated Data Cleaning, React, Node.js, Express, XLSX.

I. INTRODUCTION

In the field of data science and analytics, data preprocessing is one of the most important stages in the data analysis pipeline. The quality of any analysis largely depends on the quality of the input data. In practice, raw datasets—especially those stored in Excel spreadsheets—often contain problems such as missing values, duplicate records, inconsistent formatting, extra spaces, and incomplete entries. These issues reduce data reliability and may lead to inaccurate analysis, incorrect insights, and poor decision-making.

Traditionally, such data cleaning tasks are performed manually using spreadsheet software or simple scripts. Although these approaches are widely used, they are repetitive, time-consuming, and highly prone to human error, particularly when working with large datasets. Manual cleaning also requires significant effort from users and may not always produce consistent results. At the same time, many existing automated tools demand programming knowledge, which limits their usability for non-technical users such as students, researchers, and small organizations.

To address these limitations, this paper presents Excel Cleaner, a web-based application designed to automate the process of cleaning Excel datasets in a simple and efficient manner. The system enables users to upload Excel files through an interactive interface and receive cleaned, structured data in real time. It performs essential preprocessing operations such as removal of empty rows, elimination of duplicate records, and normalization of column names, thereby reducing manual effort and improving data quality.

The proposed system is developed using modern web technologies, including React for the frontend, Node.js and Express for backend processing, and the XLSX (SheetJS) library for Excel file handling. By combining automation with ease of use, the system provides a practical solution for users who frequently work with raw spreadsheet data. In addition, the project establishes a foundation for future improvements such as intelligent data cleaning, enhanced validation, and advanced visualization features.

A. Problem Statement

In In data analysis, the quality of input data directly affects the accuracy and reliability of results. However, most real-world datasets, especially those stored in Excel spreadsheets, often contain missing values, duplicate records, inconsistent formatting, and incomplete or unstructured entries. Such issues make data preprocessing an essential but time-consuming step before analysis can be performed.

©IJRASET: All Rights are Reserved

Traditional data cleaning methods usually rely on manual operations using spreadsheet tools or simple scripts. These approaches are repetitive, inefficient, and highly prone to human error, particularly when handling large datasets. In addition, many existing automated solutions require programming knowledge, which limits their accessibility for non-technical users such as students, researchers, and small organizations.

Therefore, there is a need for a simple, efficient, and automated system that can clean Excel data accurately without requiring technical expertise. The proposed solution should be capable of handling common data inconsistencies, reducing manual effort, and providing a clean and structured output in minimal time.

B. Research Objectives

The main objective of this research is to design and develop a web-based application that automates the process of cleaning Excel datasets efficiently, accurately, and in a user-friendly manner. The proposed system aims to simplify data preprocessing and make it accessible to users without programming knowledge.

The specific objectives of this research are as follows:

1) To develop an automated system for cleaning Excel data by removing inconsistencies such as empty rows, duplicate records, and irregular column structures.
2) To reduce the manual effort and human error involved in traditional spreadsheet data cleaning methods.
3) To design a user-friendly web interface that allows users to upload, process, and manage Excel files easily without requiring technical expertise.
4) To ensure fast and efficient data processing for small to medium-sized datasets in real time.
5) To standardize and structure raw spreadsheet data so that it becomes more suitable for analysis, reporting, and decision-making.
6) To implement the system using modern web technologies in a scalable architecture that can support future enhancements and additional preprocessing features.
7) To provide a reliable output mechanism through which users can preview and download cleaned Excel files conveniently.

II. LITERATURE REVIEW

Data cleaning and preprocessing have been widely recognized as essential steps in the data analysis process. Various researchers and tools have addressed the challenges associated with handling unstructured and inconsistent data, particularly in spreadsheet formats such as Excel.

Several studies highlight that real-world datasets often contain issues like missing values, duplicate entries, and inconsistent formatting, which can significantly impact the accuracy of analysis. Traditional tools such as Microsoft Excel provide basic functionalities like filtering, sorting, and conditional formatting; however, these features are largely manual and require continuous user intervention. This makes the process time-consuming and prone to errors, especially when dealing with large datasets.

In recent years, programming-based solutions using languages like Python and libraries such as Pandas have been introduced to automate data cleaning tasks. These tools are powerful and efficient, allowing users to handle large volumes of data with minimal effort. However, they require programming knowledge and technical expertise, which creates a barrier for non-technical users such as students, business professionals, and researchers from non-technical backgrounds.

There are also advanced data preprocessing platforms and business intelligence tools available in the market. While these tools offer sophisticated features, they are often complex, expensive, and not specifically designed for simple Excel-based data cleaning tasks. Additionally, many of these solutions focus more on data visualization and analytics rather than preprocessing and cleaning.

Based on the existing research and available tools, it is evident that there is a gap between powerful technical solutions and easy-to-use applications for general users. This gap highlights the need for a system that combines automation with simplicity, enabling users to clean and structure their data efficiently without requiring advanced technical skills.

The proposed system, Excel Cleaner, aims to address this gap by providing a web-based platform that automates essential data cleaning tasks while maintaining a simple and user-friendly interface.

III. METHODOLOGY

The proposed system follows a structured and systematic approach to automate the process of Excel data cleaning using a web-based architecture. The methodology is designed to ensure efficiency, accuracy, and ease of use, especially for users who may not have technical expertise. The entire workflow is divided into multiple stages, each responsible for a specific function in the data processing pipeline.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

A. Data Input&File Handling

The process begins with the user interacting with the web-based interface of the application. The system provides a simple and intuitive file upload feature, allowing users to select and upload Excel files directly from their local system. Once the file is selected, it is transmitted securely to the backend server using an HTTP request. This step ensures a seamless connection between the frontend and backend components of the system.

Upon After receiving the uploaded file, the backend server—developed using Node.js and Express—handles the file efficiently. A file-handling middleware is used to temporarily store the uploaded file in a designated directory on the server. This approach ensures proper file management, prevents data loss, and enables organized processing. Additionally, temporary storage supports scalability and allows the system to handle multiple file uploads if required in the future.

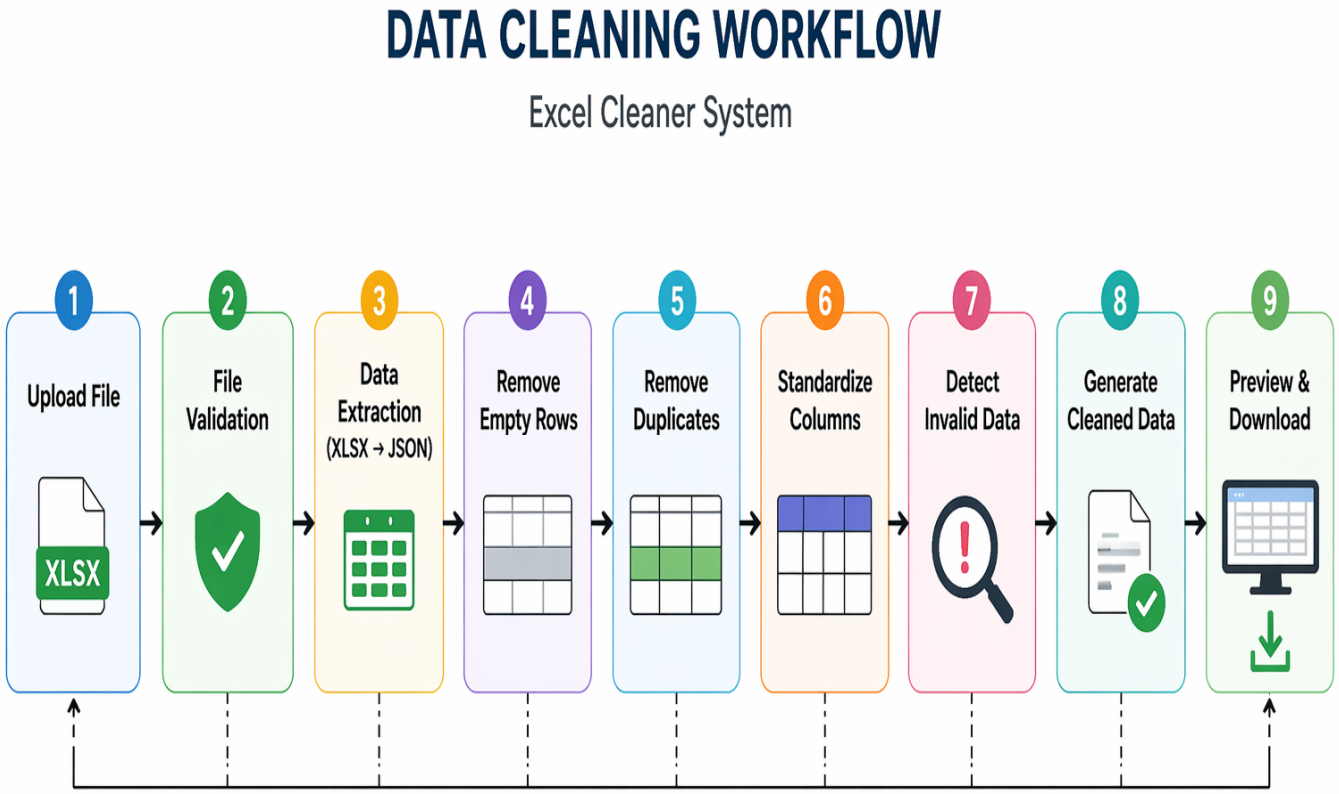


Fig 2: Data Cleaning Workflow of Excel Cleaner System

Fig 2: Data Cleaning Workflow of System

B. Data Extraction&Data Cleaning Process

In this stage, the system processes the uploaded Excel file using the XLSX (SheetJS) library. The library reads the Excel file and extracts its contents, converting the data into a structured JSON format. This transformation is crucial, as JSON provides a flexible and easy-to-manipulate structure for performing various data operations. By converting the data into JSON, the system can efficiently analyze and modify the dataset during the cleaning process.

This stage represents the core functionality of the system, where multiple data preprocessing operations are applied to improve overall data quality. The system analyzes the dataset and performs the following operations:

1) Removal of empty or null rows: Rows that do not contain meaningful data are identified and removed to reduce redundancy.

2) Elimination of duplicate records: Duplicate entries are detected and removed to ensure data accuracy and consistency.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

3) Standardization of column names: Column headers are cleaned by removing extra spaces, converting text into a consistent format (such as lowercase), and applying uniform naming conventions.

These operations collectively ensure that the dataset becomes clean, structured, and ready for further analysis.

Operation	Description
Remove Empty Rows	Deletes rows with no data
Remove Duplicates	Eliminates repeated entries
Standardize Columns	Formats column names uniformly
Invalid Data Detection	Identifies incorrect values

Table 1: Cleaning operation table

C. Data Transformation&Output Generation

Once the cleaning process is complete, the refined data—currently in JSON format—is converted back into Excel format. This step is crucial to maintain compatibility with widely used tools, as Excel remains a standard format for data storage and analysis. The system ensures that the output file preserves a clear and organized structure.

In the final stage, the cleaned Excel file is returned to the frontend interface. The user is provided with options to preview the cleaned data within the application or download the processed file for further use. This completes the data cleaning cycle, offering users a fast, efficient, and user-friendly experience.

IV. SYSTEM ARCHITECTURE

The Excel Cleaner system follows a client–server architecture in which the frontend interacts with the backend server to process data efficiently. The architecture consists of three major components: the frontend interface, the backend server, and the data processing module. Frontend provides a simple and interactive interface that allows users to upload Excel files, initiate the cleaning process, and view or download results. The backend acts as the core processing unit, responsible for handling requests, managing files, and coordinating data processing tasks. The data processing module performs all cleaning operations such as removing empty rows, eliminating duplicate entries, and standardizing column names.

After processing, the cleaned data is returned to the frontend, ensuring efficient communication between system components and providing a scalable structure for future enhancements.

A. Architecture Overview

The overall architecture follows a structured workflow where the user interacts with the frontend to upload Excel files. These files are transmitted to the backend server, where data processing operations are performed. Once processing is completed, the cleaned data is returned to the user through the frontend interface. This separation of concerns ensures efficient processing, better system organization, and an improved user experience.

1) Frontend Layer (Client Side)

The frontend of the Excel Cleaner system provides an interactive and user-friendly interface that enables users to upload Excel files and visualize the results of data cleaning operations. The interface is designed to be simple and intuitive, making it accessible to both technical and non-technical users.

The frontend handles user inputs and communicates with the backend using HTTP requests.The file upload interface allows users to either click to upload or drag and drop Excel files into the system. It supports multiple formats such as .xlsx, .xls, and .csv, improving usability and flexibility.

After uploading, the system displays a preview of the dataset along with a summary of detected issues such as missing values, duplicate records, and formatting inconsistencies. This enables users to quickly understand data quality and download the cleaned dataset efficiently.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

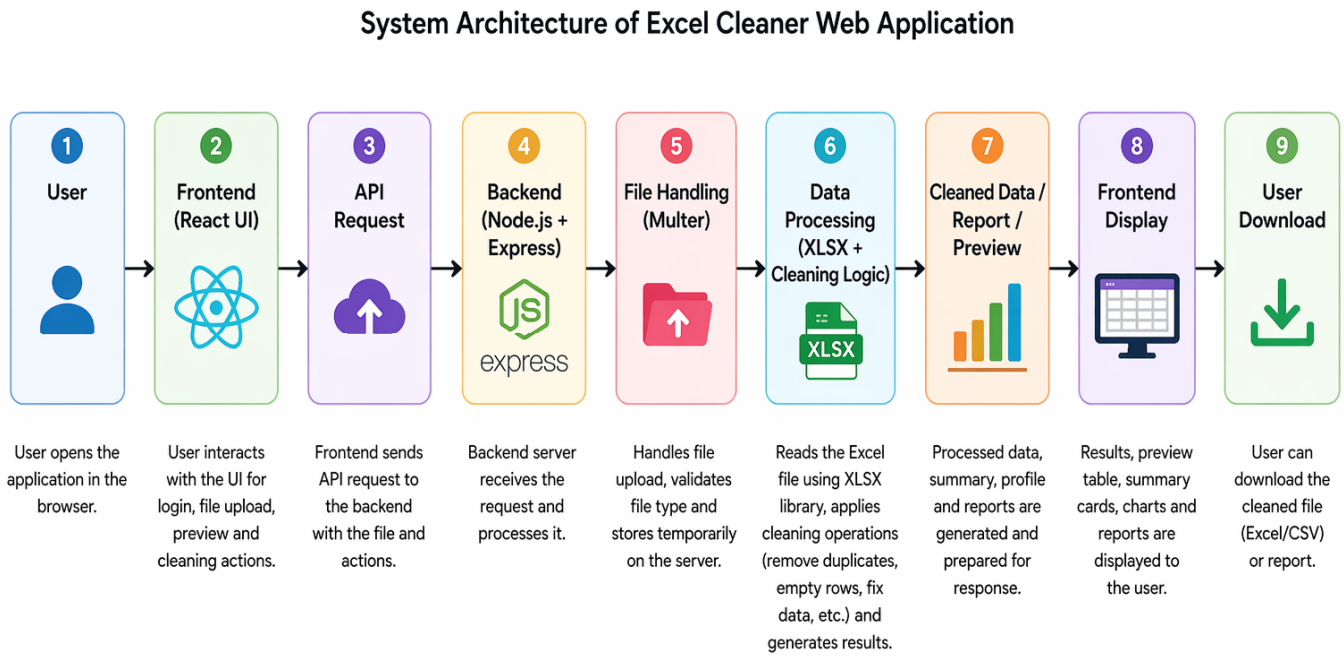


Fig 1: System architecture of Excel Cleaner System

2) Backend Layer (Server Side)

The backend layer serves as the core component of the system, responsible for handling all server-side operations and managing the overall workflow. It receives uploaded files from the frontend through HTTP requests and ensures secure data transfer and storage.

The backend validates uploaded files, manages processing tasks, and coordinates with the data processing module. It converts Excel files into structured formats and applies necessary cleaning operations. Additionally, it includes error-handling mechanisms to detect invalid file formats or corrupted data, ensuring system reliability.

After processing, the backend returns the cleaned dataset along with any invalid records in a structured format. This ensures efficient communication and enhances user experience. The modular backend design also supports future scalability, allowing integration of advanced features such as cloud storage and multi-user support.

B. Data pipeline

The file handling module manages Excel files uploaded by users. When a file is uploaded, it is temporarily stored in a designated directory on the server. This ensures safe access during processing and prevents data loss or corruption.

The module validates file types (such as .xlsx, .xls, and .csv) to maintain system reliability and security. It also organizes and removes temporary files after processing to optimize storage and improve performance.This design supports efficient handling of multiple uploads and contributes to faster processing and better resource management.

The data processing module is the core component where all data cleaning operations are performed. The uploaded Excel file is first converted into a structured format (such as JSON), allowing efficient manipulation.

The system performs the following cleaning operations:

- Removal of empty or null rows
- Elimination of duplicate records
- Standardization of column names

After processing, the cleaned data is converted back into Excel format to ensure compatibility with commonly used tools.

To enhance usability and ensure transparency in data processing, the system generates two types of outputs after the cleaning operation:’

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

a) Cleaned Data Output

The cleaned dataset consists only of valid, consistent, and structured records obtained after removing errors and inconsistencies. This output ensures that the data is accurate, well-organized, and ready for further analysis, reporting, or decision-making.

b) Invalid Data Detection Output

In addition to the cleaned dataset, the system also generates a separate dataset containing invalid or problematic records. This dataset includes an additional column named "Error Type", which identifies the specific issue associated with each record, such as Invalid Phone, Invalid Age, or Duplicate Row.

This approach improves transparency by allowing users to review, understand, and correct errors rather than permanently discarding the data. It provides better control over data quality and supports informed decision-making.

C. Data Flow &Communication and Advantages of the Architecture

The system follows a step-by-step data flow:

1) The user uploads an Excel file through the frontend
2) The file is sent to the backend server
3) The backend stores and processes the file
4) Data is extracted and converted into a structured format
5) Cleaning operations are applied
6) The cleaned data is converted back into Excel format
7) The final output is sent to the frontend
8) The user can view or download the cleaned file

This structured workflow ensures efficient processing and accurate results.

Communication between system components is handled through HTTP requests and APIs. The frontend sends user data to the backend, which processes it and returns structured responses. This ensures real-time interaction and smooth system performance.

The architecture offers several advantages:

- Clear separation of frontend and backend components
- Improved scalability and maintainability
- Faster and reliable data processing
- User-friendly and flexible design
- Easy integration of future enhancements

V. RESULTS AND DISCUSSION

The proposed Excel Cleaner system was successfully developed and tested using multiple real-world Excel datasets containing common data quality issues such as missing values, duplicate records, and inconsistent formatting. The system demonstrated effective performance in identifying and cleaning these issues while maintaining data integrity.

The system produced accurate and efficient results across different datasets. After processing, a significant reduction in duplicate records and empty rows was observed, which helped in improving overall data quality. The system also ensured better consistency in column names and data formatting by standardizing values across the dataset. As a result, the processed data became well-structured and ready for further analysis. In addition, the system was able to automatically detect and separate invalid or problematic records, thereby enhancing transparency and allowing users to review errors effectively.

The application also provides a visual cleaning summary dashboard, where users can view key metrics such as:

- Total rows processed
- Number of missing values filled
- Duplicate records removed
- Invalid entries detected
- Overall data quality score (%)

These results confirm that the system effectively automates the data cleaning process and reduces manual effort.

Metric	Before Cleaning	After Cleaning
Total Rows	1000	850

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Missing Values	200	20
Duplicate Records	150	0
Invalid Entries	100	10

Table 2:Before vs After table

D. Performance Analysis

The system was tested on small to medium-sized datasets, and the processing time was found to be fast and efficient. Most datasets were cleaned within a few seconds, depending on file size and complexity.

Key performance observations include:

- Quick file upload and processing time
- Smooth interaction between frontend and backend
- Real-time display of cleaned data and reports
- Minimal system lag during operations

This demonstrates that the system is suitable for practical use in academic and small-scale professional environments.

E. Discussion

The resultsindicate that the proposed system successfully addresses the limitations of traditional data cleaning methods. Unlike manual cleaning, which is time-consuming and error-prone, the Excel Cleaner provides an automated and reliable solution.

One of the major advantages of the system is its user-friendly interface, which allows non-technical users to clean datasets without requiring programming knowledge. The integration of data preview, profiling, and cleaning summary features enhances user understanding and improves overall experience.

Additionally, the system introduces an important feature of invalid data detection, where incorrect records are not simply removed but stored separately with error labels. This improves transparency and allows users to review and correct data if needed.

However, the system currently focuses on basic data cleaning operations and may have limitations when handling very large datasets or highly complex data structures. Future improvements can include advanced validation techniques, AI-based data cleaning, and cloud integration for better scalability.

F. Overall Analysis

Overall, the Excel Cleaner system provides a practical, efficient, and user-friendly solution for Excel data preprocessing. It significantly reduces manual effort, improves data quality, and enhances productivity. The results validate that the system can be effectively used in academic, business, and research environments.

VI. ADVANTAGES

The Excel Cleaner system offers several advantages that make it a valuable tool for data preprocessing:

1) Reduction in Manual Effort:The system automates repetitive data cleaning tasks, eliminating the need for manual intervention and saving significant time for users.

2) Improved Accuracy and Reliability:By minimizing human involvement, the system reduces the chances of errors and ensures consistent and accurate data cleaning results.

3) User-Friendly Interface:The application is designed with simplicity in mind, allowing even non-technical users to upload and process Excel files easily.

4) Fast Processing Speed:The system provides quick results, especially for small and medium-sized datasets, making it suitable for near real-time data processing.

5) Structured and Clean Output:The cleaned data is well-organized and formatted, making it ready for further analysis, reporting, and visualization.

6) Data Transparency:The system separates invalid or problematic data instead of deleting it, allowing users to review and understand errors effectively.

7) Support for Multiple File Formats:The application supports commonly used formats such as .xlsx, .xls, and .csv, increasing usability and flexibility.

8) Scalability and Flexibility:The modular architecture allows easy expansion and integration of additional features in the future

VII. LIMITATIONS

Although the Excel Cleaner system provides an efficient and user-friendly solution for automating data cleaning tasks, it has certain limitations that need to be considered. The current system primarily focuses on basic data preprocessing operations such as removal of empty rows, elimination of duplicate entries, and standardization of column names. While these features are effective for general use, the system does not yet support advanced data cleaning techniques such as handling missing values through imputation, detecting outliers, or applying complex data transformations.

Another limitation is that the system is optimized for small to medium-sized datasets, and its performance may decrease when processing very large Excel files due to memory and processing constraints on the server. Additionally, the system assumes that the uploaded Excel files follow a relatively consistent structure. If the dataset contains highly irregular formats, multiple sheets with different schemas, or complex relationships between columns, the system may not handle them accurately.

Furthermore, the application currently operates in a local or limited deployment environment and does not include features such as cloud storage, user authentication, or real-time collaboration. These limitations restrict its scalability and usability in enterprise-level applications. Despite these constraints, the system provides a strong foundation for further development and improvement.

VIII. CONCLUSION AND FUTURE WORK

A. *Conclusion*

In conclusion, the Excel Cleaner system successfully addresses the challenges associated with manual data cleaning by providing an automated, efficient, and user-friendly solution. It simplifies the process of handling raw and unstructured Excel datasets by performing essential preprocessing operations such as removing duplicates, handling missing values, and standardizing data formats.

The system significantly reduces the time and effort required for data preparation, which is often one of the most time-consuming tasks in data analysis. By minimizing human intervention, it also improves the accuracy and consistency of the cleaned data. The intuitive interface ensures that users from both technical and non-technical backgrounds can easily use the application without requiring programming knowledge.

Another key strength of the system is its ability to provide transparency through features such as data preview and invalid data detection. Instead of permanently removing incorrect records, the system separates them, allowing users to review and understand errors. This enhances trust and usability.

Overall, the Excel Cleaner system proves to be a practical and reliable tool for data preprocessing. It can be effectively used in academic projects, business environments, and research applications where clean and structured data is essential for accurate analysis and decision-making.

B. *Future work*

The Excel Cleaner system can be further improved by incorporating advanced features and technologies to enhance its functionality and scalability. One important area of improvement is the integration of machine learning techniques, which can enable intelligent data cleaning by automatically detecting complex patterns, anomalies, and inconsistencies in datasets.

Another potential enhancement is the ability to handle large-scale datasets through cloud-based processing and storage solutions. By leveraging cloud platforms, the system can improve performance, scalability, and accessibility, making it suitable for enterprise-level applications and big data environments.

The system can also be extended by adding customizable data cleaning options, allowing users to define their own rules and conditions based on specific requirements. Advanced data validation features and support for additional file formats can further increase flexibility and usability.

In addition, future versions of the system can include features such as real-time collaboration, user authentication, and interactive data visualization dashboards. These improvements would enhance user experience, promote teamwork, and provide deeper insights into data. With these advancements, the Excel Cleaner system can evolve into a more powerful and versatile data processing tool.

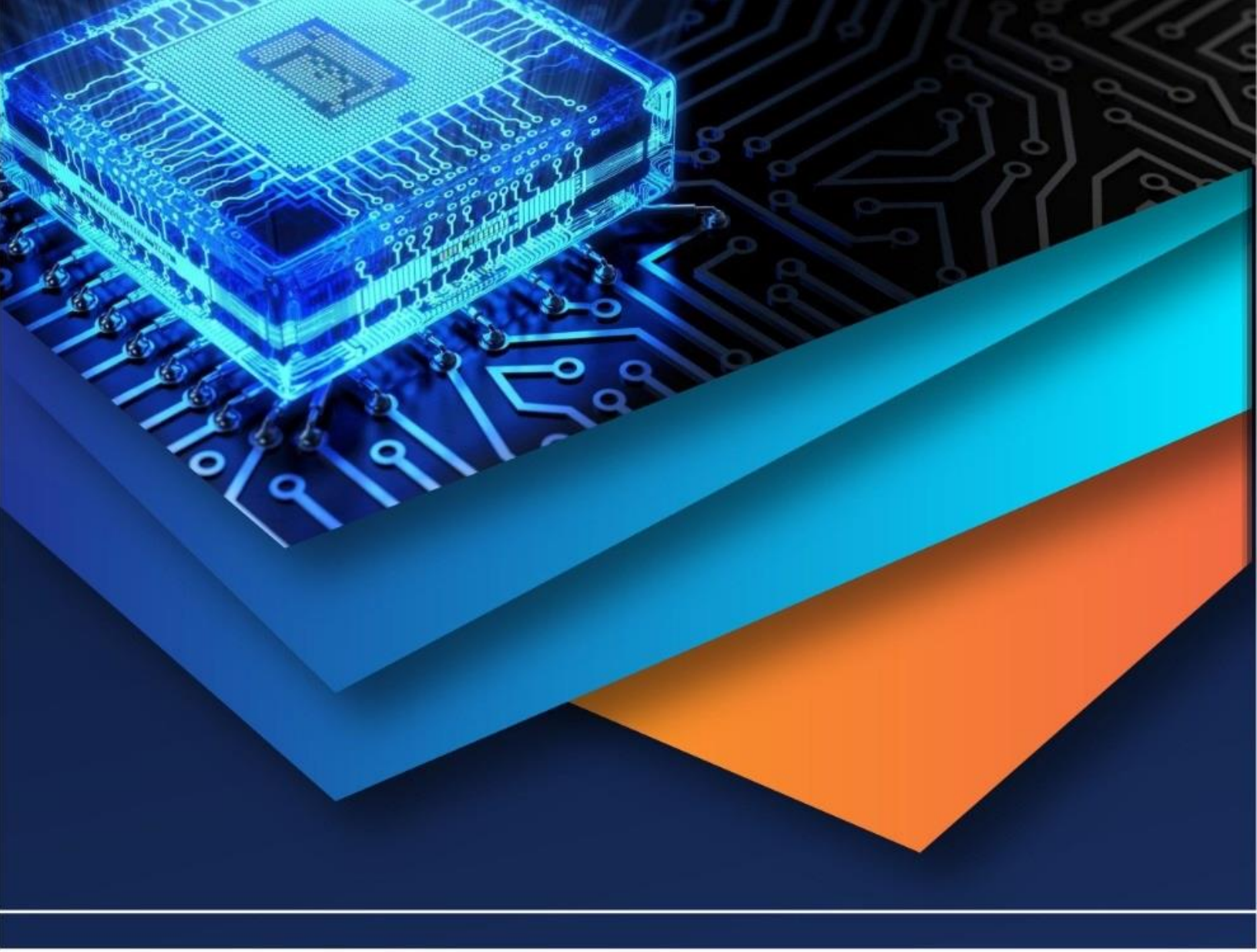
REFERENCES

[1] Microsoft Corporation, Microsoft Excel Documentation, Available at: https://support.microsoft.com/excel
[2] SheetJS, XLSX JavaScript Library Documentation, Available at: https://docs.sheetjs.com
[3] Node.js Foundation, Node.js Documentation, Available at: https://nodejs.org/en/docs
[4] Express.js, Express Framework Documentation, Available at: https://expressjs.com

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[5] React Documentation, React – A JavaScript Library for Building User Interfaces, Available at: https://react.dev
[6] W3Schools, HTML, CSS, and JavaScript Tutorials, Available at: https://www.w3schools.com
[7] Kelleher, J. D., & Tierney, B., Data Science Fundamentals for Python and MongoDB, O'Reilly Media, 2018
[8] Rahm, E., & Do, H. H., "Data Cleaning: Problems and Current Approaches," IEEE Data Engineering Bulletin, Vol. 23, No. 4, 2000
[9] Kimball, R., & Caserta, J., The Data Warehouse ETL Toolkit, Wiley Publishing, 2004
[10] Han, J., Kamber, M., & Pei, J., Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)