A

PROJECT REPORT

ON

“Data Cleaner- Automated Data Cleaning Web Application”

SUBMITTED TO

SHIVAJI UNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF DEGREE
BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR. SHREYAS SHIVAJI MANE 23UAD303

UNDER THE GUIDANCE OF

Mr. S. P. Pise

DKTE
AN AUTONOMOUS INSTITUTE
NAAC ACCREDITED WITH 'A +' GRADE

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE ENGINEERING

DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(AN EMPOWERED AUTONOUMOUS INSTITUTE)

2024-2025

D.K.T.E. SOCIETY’S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN EMPOWERED AUTONOUMOUS INSTITUTE)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE ENGINEERING

DKTE
AN AUTONOMOUS INSTITUTE
NAAC ACCREDITED WITH 'A +' GRADE

CERTIFICATE

This is to certify that, project work entitled

“Data Cleaner-
Automated Data Cleaning Web Application”

is a bonafide record of project work carried out in this college by

MR. SHREYAS SHIVAJI MANE 23UAD303

is in the partial fulfillment of award of degree Bachelor of Technology in Artificial Intelligence and Data Science Engineering prescribed by Shivaji University, Kolhapur for the academic year 2024-2025.

MR. S. P. PISE
(PROJECT GUIDE)

PROF. (DR.) T. I. BAGBAN	**PROF.(DR.) L.S.ADMUTHE**
(HOD AI & DS DEPT.)	**(DIRECTOR)**

EXAMINER: ________________

DECLARATION

We hereby declare that, the project work report entitled "Data Cleaner- Automated Data Cleaning Web Application" which is being submitted to D.K.T.E. Society's Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.Tech.(AI & DS). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

Title	**Name of the Student**	**PRN**	**Signature**
MR.	Shreyas Shivaji Mane	23UAD303	

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Mr. S. P. Pise for his valuable guidance, support, and encouragement in the completion of this project report.

Also, we would like to take the opportunity to thank our head of department Dr. T. I. Bagban for his cooperation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Artificial Intelligence and Data Science Department for their support, help, and assistance which they extended as and when required.

Thank you,

Title	Name of the Student	PRN
MR.	Shreyas Shivaji Mane	23UAD303

ABSTRACT

In the era of data-driven decision-making, the quality of data plays a crucial role in deriving accurate and meaningful insights. However, real-world datasets often contain inconsistencies, missing values, and duplicate entries that hinder effective analysis. This project introduces a web-based data cleaning application built using **Streamlit** for the frontend and integrated with **Firebase Authentication** for secure user management. The system enables users to upload raw CSV or Excel datasets, automatically detects and removes duplicate records, and handles missing values by filling numeric columns with mean values or dropping incomplete non-numeric rows. The cleaned and duplicate datasets are made available for download, providing users with a transparent and streamlined preprocessing experience. The application is designed with simplicity, security, and usability in mind, making it an ideal solution for students, analysts, and researchers seeking quick data sanitization without coding. This project aims to enhance data reliability and efficiency in data science workflows by automating essential cleaning operations.

INDEX

1. Introduction

a. Problem definition

In today's data-driven landscape, vast amounts of information are collected daily from various sources. However, these datasets often suffer from issues such as missing values, duplicate records, and inconsistent formatting. These problems can lead to inaccurate analysis, misleading insights, and inefficient decision-making. Manual data cleaning is time-consuming, error-prone, and typically requires technical expertise. There is a clear need for an automated, user-friendly, and secure platform that enables users to clean their data effectively, without writing code. This project addresses this need by building a web-based application for smart and automated data cleaning.

b. Aim and objective of the project

To develop a secure and user-friendly web application that automates the process of cleaning datasets by removing duplicates and handling missing values.

Objectives:

- To allow users to securely log in or sign up using Firebase Authentication.
- To enable users to upload raw datasets in CSV or Excel format.
- To automatically detect and remove duplicate rows.
- To fill missing numeric values with column mean and drop rows with missing non-numeric values.
- To allow users to preview and download cleaned datasets and duplicate records separately.

c. Scope and limitation of the project

This project offers an end-to-end automated data cleaning solution through a web interface. It supports common formats like CSV and Excel and applies cleaning operations suited for exploratory data analysis. It is intended for data science learners, analysts, and researchers who require fast, code-free data preparation.

Limitations:

- The system does not currently support advanced cleaning like outlier detection, data type corrections, or categorical encoding.
- Uploaded files are not persistently stored, the platform does not include historical tracking.
- The application performs cleaning steps automatically, users cannot choose which operations.

2. Background study and literature overview

a. Literature overview

In the field of data science, data preprocessing is a critical initial step that significantly impacts the outcomes of analysis and model performance. Numerous studies have emphasized the importance of clean, consistent, and structured data for deriving reliable insights. According to Rahm and Do (2000), data cleaning can consume up to 80% of the time in data analysis projects, highlighting its critical role in the data pipeline.

Recent advancements have led to the development of automated data wrangling tools such as Trifacta, OpenRefine, and cloud-based platforms like Google DataPrep. These tools offer GUI-based operations for cleaning data, but they often come with limitations such as cost, complexity, or steep learning curves for beginners. In educational and exploratory contexts, simpler and more focused tools can offer better accessibility and usability.

Several academic and industry projects have attempted to address common data quality problems, including missing values and duplicate entries, using automated scripts or machine learning algorithms. However, these are often implemented in code-heavy environments that may not be suitable for non-technical users. The need for a secure, accessible, and lightweight tool for data cleaning especially one that integrates user authentication and downloadable output remains largely unaddressed.

This project aims to contribute to this space by delivering a streamlined, user-friendly web application that automates fundamental data cleaning tasks, including duplicate detection and null value handling, all while offering a secure user experience through Firebase Authentication.

Link for Research Paper:

https://www.researchgate.net/publication/220282831_Data_Cleaning_Problems_and_Current_Approaches

b. Investigation of current project and related work

This project focuses on providing a secure, no-code solution for cleaning datasets by addressing two of the most common issues in raw data: duplicate entries and missing values. The approach is implemented using a **Streamlit web interface**, backed by **Python (Pandas)** for data processing and **Firebase** for authentication.

Unlike enterprise-level tools that aim to solve all aspects of data preparation, this system narrows its scope to basic but essential cleaning operations, enabling faster processing and minimal user input. It is particularly beneficial for students, junior analysts, and researchers who work with relatively small datasets and need quick preprocessing support without coding.

While platforms like Talend and DataRobot offer comprehensive data preparation capabilities, they require considerable technical setup and cost. On the other hand, educational tools such as Excel provide limited automation, often requiring manual filtering or formulas. This project bridges the gap by providing an automated, lightweight, and accessible alternative with authentication security built in.

applicability. By targeting high-impact crops like potato, tomato, and corn, the system ensures relevance to major agricultural economies.

3. Requirement analysis

a. Requirement Gathering

1. Functional Requirements

These are the key features the system must support:

- **User Authentication**
  The application should allow users to securely sign up, log in, and reset passwords using Firebase Authentication.
- **File Upload**
  Users must be able to upload datasets in .csv or .xlsx formats through a user-friendly file uploader.
- **Data Cleaning Logic**
  The backend must:
  - Detect and remove duplicate rows from the uploaded dataset
  - Identify and handle missing values:
    - Fill numeric columns with mean values
    - Drop rows with missing non-numeric values
- **Output Delivery**
  The system should:
  - Display a preview of raw, duplicate, and cleaned data
  - Allow users to download both cleaned and duplicate datasets as .csv files
- **Session Management**
  Restrict upload access to logged-in users only and include logout functionality.

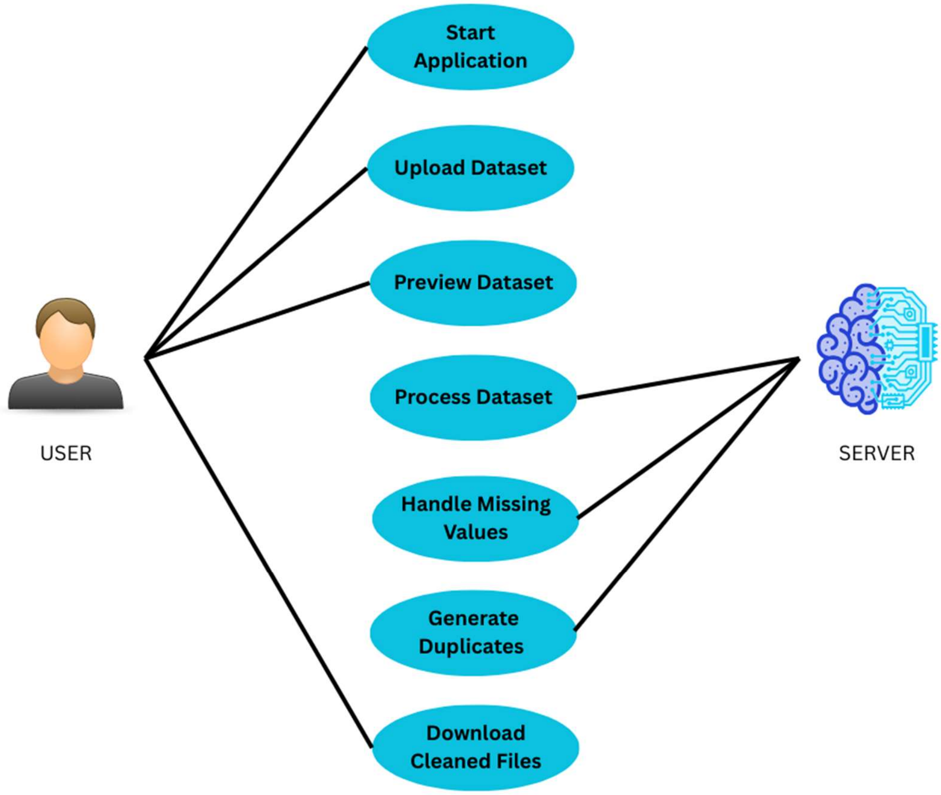
2. Non-Functional Requirements

- **Usability**
  The UI must be simple and intuitive, suitable for students, analysts, and non-technical users.
- **Security**
  Secure login, password reset, and logout should be integrated using Firebase Authentication.
- **Performance**
  The application should process moderate-sized files quickly (within seconds) and operate smoothly on standard hardware.
- **Portability**
  The app should be deployable via Streamlit Cloud or hosted locally, and accessible from web browsers on desktops and tablets.
- **Maintainability & Scalability**
  Code should be modular to allow future enhancements like advanced cleaning options (e.g., outlier detection, datatype normalization).
- **Reliability**
  The system should handle malformed files or unexpected data gracefully and provide clear feedback to the user.

3. Technical Requirements

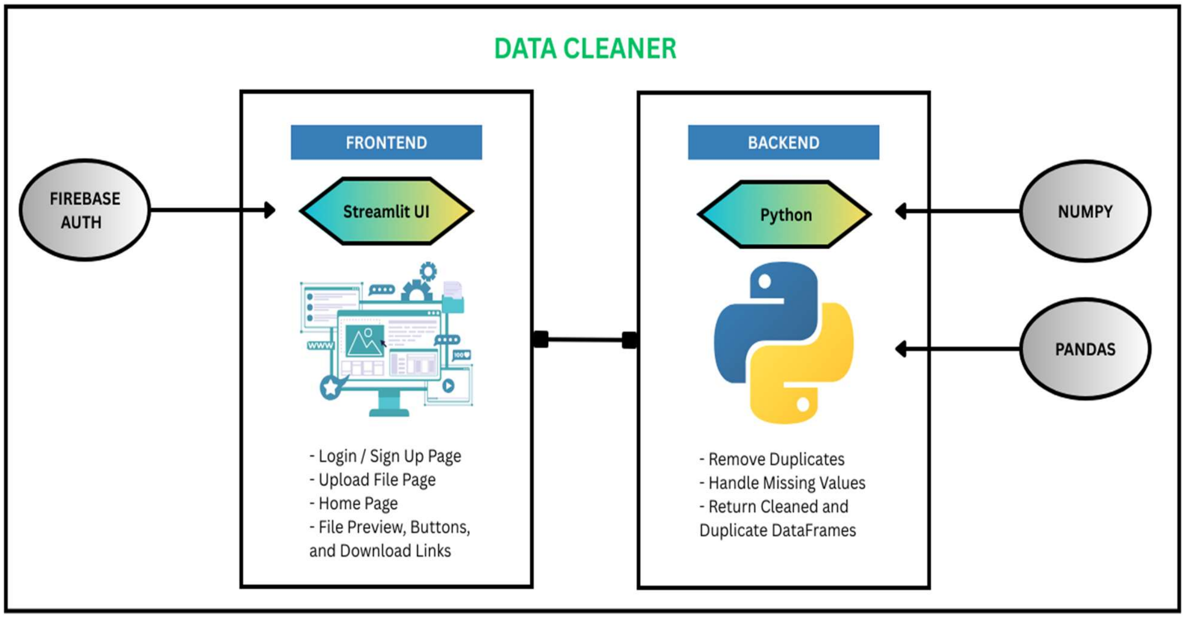
- **Programming Language**
  - Python
- **Libraries/Frameworks**
  - Streamlit (for UI and frontend interaction)
  - Firebase Admin SDK + REST API (for authentication)
  - Pandas, NumPy (for data handling and cleaning)
  - io, os (for in-memory file generation and upload handling)
- **Architecture**
  - Modular backend (backend.py) to perform cleaning logic
  - Multi-page structure using Streamlit's pages/ folder
- **Deployment Environment**
  - Streamlit Cloud
  - Alternatively, run locally using streamlit run app.py

a. Use case Diagram

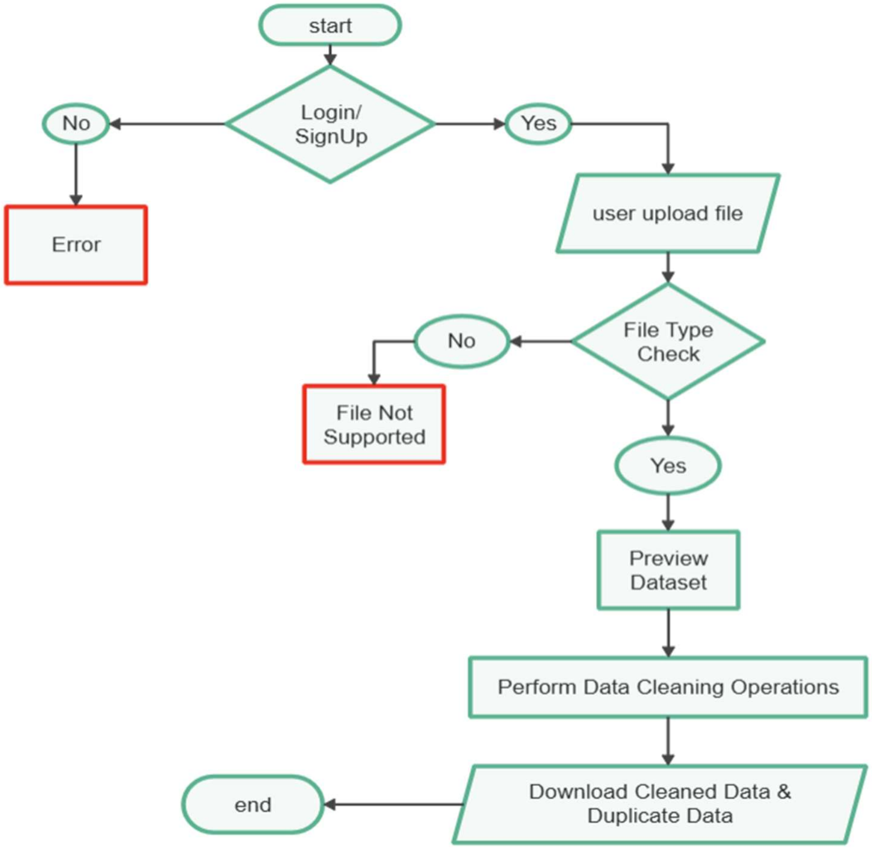


4. System design

A. Architectural Design

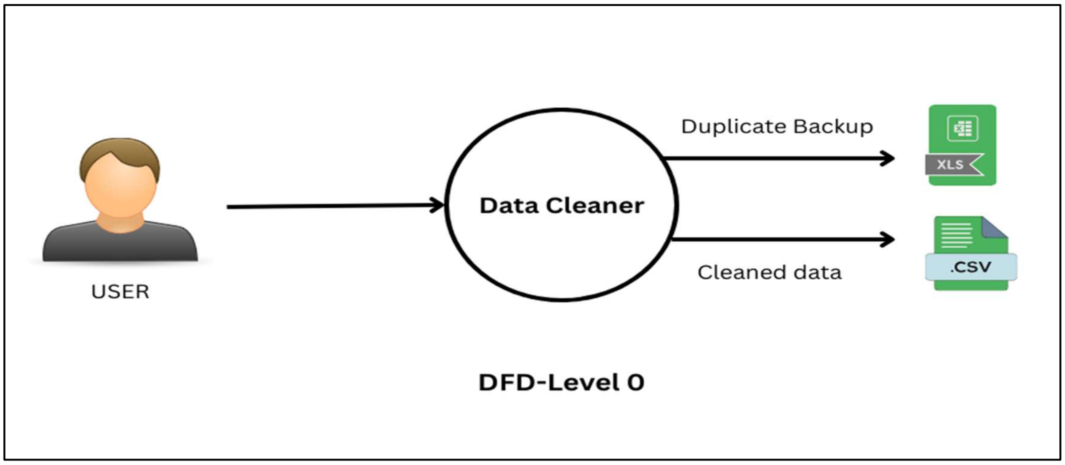


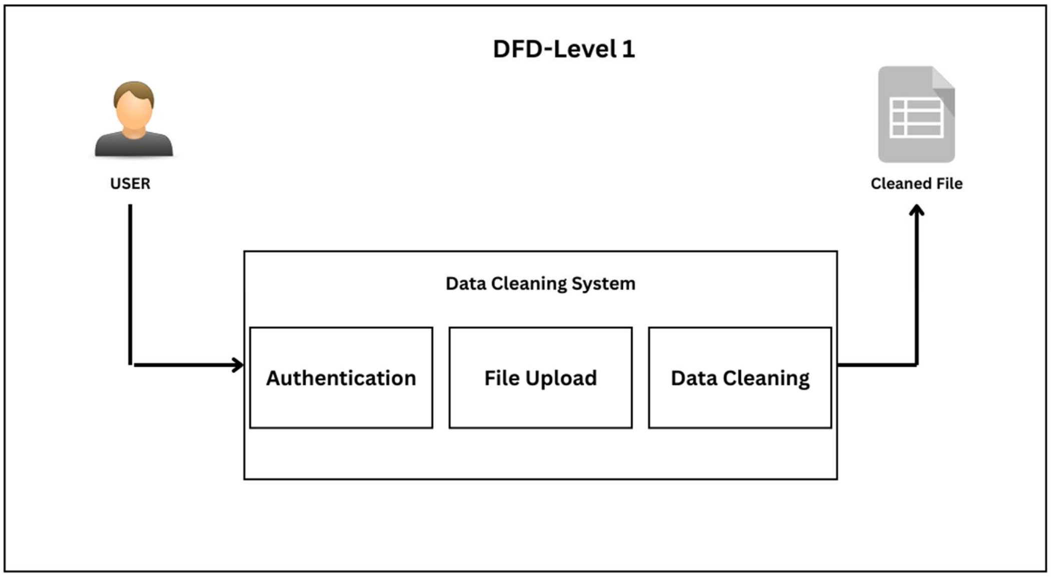
B. Flow Chart



C. System Modeling

Dataflow Diagram





5. Implementation

a. Agile Methodologies

The development of the Data Cleaning Web Application follows Agile principles to ensure flexibility, iterative delivery, and continuous feedback. Agile methodologies support collaborative development, allowing for rapid adjustments based on evolving user needs and functional enhancements. Short development cycles (sprints) were used to incrementally build the system, test features, and deploy usable components. Regular reviews helped integrate feedback from testers and improve the user experience. This approach ensured that essential modules—like file upload, cleaning logic, and Firebase authentication—were tested and improved in stages rather than waiting for final delivery.

b. Development Model

Spiral Model

The **Spiral Model** was chosen for this project due to its **iterative structure**, strong focus on **risk mitigation**, and its ability to incorporate feedback at every stage. This model is particularly suitable for this system, where usability, security, and reliability are essential, and where refinements based on user interaction are crucial.

Spiral consists of four phases: **Planning**, **Risk Analysis**, **Engineering**, and **Evaluation**.

1. Planning Phase:

Project requirements and functionality were identified, including:

-User login/signup and secure authentication

-Uploading CSV/Excel datasets

-Removing duplicate entries

-Handling missing values (filling numeric, dropping non-numeric)

-Downloading cleaned and duplicate files

This phase defined the scope and set deliverables for each spiral.

2. Risk Analysis Phase:

Possible risks were identified, such as:

-Handling unexpected file formats or corrupted data

-Failing to clean large datasets in memory

-Firebase authentication failures or misconfiguration

-UI usability concerns for non-technical users

Mitigation strategies were applied, such as file type checks, in-memory processing optimizations, and frontend validations. These risks were reviewed after each spiral.

3. Engineering Phase:

The actual implementation occurred in multiple spirals:

-Spiral 1: Developed the backend logic for cleaning and duplicate detection

-Spiral 2: Built and tested the Streamlit UI with sample files

-Spiral 3: Integrated Firebase for user authentication and session handling

-Spiral 4: Finalized download logic, logout flow, and visual enhancements

Each component was tested individually and integrated with others in a modular.

4. Evaluation Phase:
After each spiral, the system was evaluated to ensure:
-Files were cleaned correctly
-Duplicate and cleaned data were downloadable
-Authentication and session states worked as expected
-The interface was user-friendly
Feedback from trial users (peers and project reviewers) was used to guide the next development cycle, improving both functionality and UI.

Advantages of the Spiral Model:

- Flexibility and Adaptability: Allows incremental updates and easy adjustments based on testing and user feedback.
- Risk Management: Each phase includes identifying and addressing potential technical and usability risks early.
- Continuous Improvement: Features are enhanced step-by-step through repeated cycles, improving reliability and performance.
- Stakeholder Engagement: User feedback is incorporated continuously, aligning the tool with actual needs and expectations.

6. Future Scope

The current system serves as a foundational tool for automated data cleaning through a web-based interface. While it effectively handles basic data issues such as duplicates and missing values, several enhancements can be introduced to extend its usability and functionality in the future:

- **Advanced Data Cleaning Techniques**
  Future versions can include outlier detection, inconsistent data correction and handling categorical encoding.
- **File Preview Before Cleaning**
  Add functionality to visually inspect and select which columns to clean or exclude before processing.
- **User Data History and Storage**
  Enable logged-in users to view past uploaded and cleaned datasets, stored either locally or in Firebase Storage.
- **Multi-file and Batch Processing**
  Allow users to upload and clean multiple files in a single session, making the tool more efficient for data-heavy users.
- **Integrated Data Visualization**
  Provide visual summaries (charts, null-value heatmaps, column distributions) to help users understand their data before and after cleaning.
- **Custom Cleaning Options**
  Give users control over how to treat missing values (e.g., fill with median, mode, or a fixed value) and how to define duplicates.
- **Mobile Support or App Version**
  A mobile-friendly version or dedicated app could help users clean data on the go, especially useful for students or quick insights.
- **Deployment to Cloud with Persistent Database**
  Host the application on a secure cloud platform with database integration for storing user profiles, files, and history permanently.

7. References (public repository GitHub source code links)

https://github.com/shreyas2004/Data_Cleaner.git