





**Graphic Era**  
**Hill University**  
 DEHRADUN • BHIMTAL • HALDWANI

## PROJECT AND TEAM INFORMATION

### Project Title

Smart Data Pre-processor: A Web-Based Tool for Cleaning Datasets

### Student/Team Information

Team Name:	Sapphire
Team member 1 (Team Lead)	Atharv Gangwar – 230112386 <a href="mailto:atharvgangwar8@gmail.com">atharvgangwar8@gmail.com</a> 
Team member 2	Dhruv Negi – 23011451 <a href="mailto:negi67291@gmail.com">negi67291@gmail.com</a> 

Team member 3

Abhishek Negi – 23011732

[abhisheknegi75054@gmail.com](mailto:abhisheknegi75054@gmail.com)



## PROJECT PROGRESS DESCRIPTION

### Project Abstract

The Smart Preprocessor and Visualizer is a web application developed in Java that enhances data preprocessing and visualization workflows. It integrates a MongoDB database to efficiently store and manage preprocessed data, ensuring users can handle large datasets effectively.

**Data Preprocessing:** The application enables users to clean and structure their data, improving the quality of insights derived from subsequent analyses.

**MongoDB Integration:** Utilizing MongoDB allows for robust and scalable data storage, accommodating various user-defined requirements.

**User-Centric Design:** The application is designed with a focus on user experience, offering a seamless interface for both preprocessing and visualization tasks.

### Updated Project Approach and Architecture

**File Upload and Parsing:** Users can upload their CSV files through the Java interface. Upon upload, the application parses the CSV data to prepare it for preprocessing.

**Data Processing Options:** After parsing, users are directed to a dedicated route (/process) where they can select various preprocessing operations to apply to their data. This flexibility allows users to tailor the preprocessing steps according to their specific needs.

**Data Storage:** Once the preprocessing is complete, the cleaned and structured data is saved to a MongoDB Atlas database. This ensures efficient storage and management of the data for future use.

## Tasks Completed

Task Completed	Team Member
1. Codebase Design and Workflow	Atharv Gangwar and Abhishek Negi
2. MongoDB Database Setup	Atharv Gangwar
3. Route and MVC Architecture	Atharv Gangwar
4. HTML Template Development	Dhruv Negi
5. CSV File Parsing Implementation	Dhruv Negi
6. Model Layer Code	Abhishek Negi
7. Exception Handling Mechanisms	Abhishek Negi

## Challenges/Roadblocks

One of the primary challenges encountered during the development of the Smart Preprocessor and Visualizer was the lack of prior knowledge among team members regarding Java, the core programming language used for the project. This presented a significant hurdle, as understanding Java's syntax, libraries, and frameworks was essential for successful implementation.

Another challenge arose from managing the numerous dependencies required for the project. With a variety of libraries and frameworks needed for both the Java backend and the React frontend, ensuring compatibility and proper configuration became a complex task. Additionally, the creation of mathematical functions for data preprocessing posed another significant challenge. Developing accurate and efficient algorithms to handle various preprocessing tasks, such as normalization, scaling, and data transformation, required a deep understanding of both the mathematical concepts involved and their practical implementation in code.

Despite these challenges, the team's commitment to learning and problem-solving ultimately led to a successful project outcome, demonstrating resilience and adaptability in the face of obstacles.

## Tasks Pending

Task Pending	Team Member (to complete the task)
All Done	--

## Project Outcome/Deliverables

**User -Friendly Interface:** A Java-based interface that allows users to easily upload CSV files and select preprocessing operations, enhancing user experience.

**Efficient Data Processing:** Implementation of robust data parsing and preprocessing functionalities, ensuring that users can clean and structure their data effectively.

**MongoDB Integration:** A fully functional MongoDB database that securely stores preprocessed data, enabling efficient data management and retrieval.

**Dynamic Data Visualization:** A separate React.js application utilizing Chart.js for interactive data visualization, allowing users to gain insights from their processed data seamlessly.

**Learning Experience:** The project fostered significant learning among team members, particularly in Java programming, dependency management, and mathematical function development.

## Progress Overview

The Smart Preprocessor and Visualizer project has made significant strides since its inception. Initially, the team focused on understanding Java, which was a new language for all members. Through collaborative learning and hands-on practice, we quickly gained proficiency, enabling us to build the core functionalities of the application.

We successfully implemented the file upload feature, allowing users to upload CSV files for preprocessing. The parsing and data cleaning processes were developed, ensuring that users can effectively prepare their data for analysis.

The integration of MongoDB was completed, providing a reliable database solution for storing preprocessed data. We also designed the application's routes and established a Model-View-Controller (MVC) architecture, promoting organized code management.

As of now, we are finalizing the documentation and conducting thorough testing to ensure the application functions as intended. Overall, the project is on track, with all major components in place and a clear path toward completion.

## Codebase Information

(Repository link, branch, and information about important commits.)

Repository Link: [https://github.com/AtharvGangwar48/Smart\\_Preprocessor\\_and\\_Visualizer](https://github.com/AtharvGangwar48/Smart_Preprocessor_and_Visualizer)

Branch: Origin

Commits: "Initial commit and Project Done" (20th May 2025)

(Pushed in a single commit to origin via IntelliJ IDE)

## Testing and Validation Status

Test Type	Status (Pass/Fail)	Notes
Unit Testing	Pass	Individual components of the application, such as the CSV parsing functions and data preprocessing algorithms.
Integration Testing	Pass	Integration tests were conducted to verify that different components of the application, like Java backend worked seamlessly together. This included testing the data flow from file upload to MongoDB storage.
Route Testing	Pass	Each route in the application was thoroughly tested to ensure proper functionality. This included verifying that users could successfully navigate from the file upload page to the processing page and then to the visualization page.
Responsive Testing	Pass	All pages of the application were tested for responsiveness across different devices and screen sizes. The layout and functionality were verified to ensure a consistent and user-friendly experience on desktops, tablets, and mobile devices.
Error Handling Validation	Pass	

## Deliverables Progress

The Smart\_Preprocessor project successfully implements a complete structured data analysis. Users can upload CSV datasets, apply preprocessing steps such as cleaning, encoding, and scaling the data. The system supports user-uploaded models, allowing flexibility in analysis. The backend is built using Spring Boot with MongoDB, providing scalable API endpoints, while the frontend is a Java-based GUI using JavaFX. Major deliverables such as data upload interface, preprocessing modules, model integration, and real-time visualization are fully developed. Remaining tasks focus on extended model repository support and cloud deployment optimization.