ISSN (e): 2250 – 3005 || Volume, 16 || Issue, 7|| July – 2026 ||
International Journal of Computational Engineering Research (IJCER)

Design and Implementation of a Web-Based Platform for Exploratory Data Analysis and Visualization

Alban Sovkhozov[1], Edip Senyurek[2]
[1,2] Department of Engineering,
Vistula University, Warsaw, Poland
Corresponding Author: Alban Sovkhozov

ABSTRACT
This system allows users to upload CSV files; it also helps them to execute pre-processing pipelines and calculate descriptive statistics. Additionally, it offers methods for detecting outliers (Z-score and Interquartile Range (IQR)), computing Pearson correlation coefficients, and producing interactive charts using Chart.js. In general, the architecture consists of a strict separation between client and server functions; this is reflected in the service-layer of the application currently residing in the client (i.e., browser) which acts as an abstraction layer over the various services; and a relational database schema consisting of 8 logical entities - users, projects, settings, datasets, data_preprocessing, analysis_results, visualisations, and reports. To validate the system, we used a synthetic employee dataset containing 1000 rows and 12 attributes to demonstrate the system's ability to identify correlation between salary and experience (r = 0.70), identify missing value distributions (2.34% of total data) and generate exportable reports in PDF format. In summary, a single browser-based interface for preprocessing, analysis, and visualisation eliminates the overhead associated with changing between tools while at the same time retaining the rigorous analytical standards expected of desktop programming environments.
KEYWORDS: *Exploratory Data Analysis; Web Application; Vite; JavaScript; Data Visualization; Chart.js; Pearson Correlation; Outlier Detection; Client-Server Architecture; CSV Processing; Descriptive Statistics; Interactive Dashboard*
AMS Subject Classification: 05C12.

Date of Submission: 02-07-2026 Date of acceptance: 11-07-2026

I. INTRODUCTION

Industry data suggests that generated 181 zettabytes of data globally by 2025 and this is projected to grow to approximately 221 zettabytes by 2026 [1]. The amount of data produced requires exploratory data analysis (EDA) preceeding any modeling or decision support after reviewing the data manually.

EDA was introduced in 1977 by John W. Tukey as a formalized process to summarize the characteristics of datasets (often using graphical and statistical techniques) prior to theorizing about the data sample [2]. Practitioners typically have used statistical packages (SAS, SPSS, R) or scripting in Python using substantial domain-specific programming skill and with a high level of installation overhead. Cloud-hosted BI platforms such as Tableau and Power BI improve accessibility but limit extensibility and introduce data-sovereignty concerns while also introducing data-sovereignty concerns (Table 1).

There is a tension between ease of use and analytical strength where you can immediately use a tool to analyze your data and you do not have to be a domain expert with high-level programming skill to produce statistically accurate results. I will bridge the gap between immediate access to an EDA tool and robust statistical results by providing a client-side web application that facilitates the complete EDA workflow from importing and processing raw CSV data through statistical analysis, producing interactive visual displays, and allowing the user to generate a PDF report, all within one web browser session. The web application will be built using the Vite build tool [3] using standard JavaScript (ES6+), HTML5, CSS3, and Chart.js [4] and operated on a single static file server.

To begin, the primary issue with using conventional exploratory data analysis (EDA) tools is that they do not allow users to interact with the static plots produced by either M Matplotlib (Python) [5] or ggplot2 (R) [6], in real-time. Users are therefore required to run code again in order to update the visualisation, scale the data, or change parameters for the plot. Secondly, without a unified analytical pipeline, users must move data through multiple analytical tools for data cleaning (Excel), statistical analysis (Python), and visualisation (Tableau), which increases the risk of transcription errors and decreases data reproducibility between different tools. Finally, the technical nature of the programming language and associated tools creates a technical barrier for domain experts (health, social science, management) who want to perform exploratory data analysis (EDA) on their own.

The objectives of this project are:

- Create a modular, browser-native architecture that keeps the methods for data pre-processing, analysis, and data visualisation logically separate from one another or in service areas.
- Establish the core analytical primitives (data description statistics; Pearson correlation analysis; outlier detection via Z-score or IQR).
- Provide users with the ability to create interactive dashboards representing data in one of the following si x visual formats: bar chart; line graph; scatter plot; histogram; box/whisker plot; heat map; and pair plot.
- Conduct the data visualisation and analytics on a structured employee data set containing at least 1,000 records and 12 attributes (with the goal of providing a baseline comparison of output between the proposed solution and that of Python using Pandas and Seaborn).
- Compare the capabilities of the proposed solution against typical spreadsheet solutions, programming environments, and web analytical tools based upon the following: analytical capabilities, user support, ease of data input, and security.

II. RELATED WORK AND THEORETICAL BACKGROUND

Cleveland [8] expands on the principles of using the box plot and whisker plot still used in EDA dashboards; he provides rules for the visual perception of 2-dimensional graphical design of data based on empirical findings indicating that length encoding (bar graphs) is more accurately perceived than area or angle encodings. The statistical basis for this concept can be found in the literature. The bibliography provides a comprehensive definition of descriptive statistics (e.g., mean, median, mode, variance, standard deviation, skewness and kurtosis) that may be utilized in the data mining field, as documented by Han, Kamber & Pei [9]. James et al. [10] define a correlation coefficient (Pearson's $r = \sum [(xi - \bar{x})(yi - \bar{y})] / [(n-1)sxsy]$) as a measure of linear association; the value of the correlation coefficient has a fixed range (-1 to 1). The square of the correlation coefficient (R^2) represents the explanatory variance of the data. It is important to understand that correlation does not imply causation or direct association [11].

The Z-score method is appropriate to apply to variables that have roughly a normal distribution. An observation xi can be identified as an outlier if $|zi| = |(xi - \mu)/\sigma| > 3.0$ [12]. For non-normally distributed data, Tukey's Interquartile Range (IQR) method [2] is more robust way of identifying outliers. An observation is defined as an outlier when either $xi < Q1 - 1.5 * IQR$ or $xi > Q3 + 1.5 * IQR$, where $IQR = Q3 - Q1$. Theoretical tradeoffs associated with each of these methodologies for robust estimates are discussed in [13].

JavaScript-native analytics libraries have matured to support more sophisticated use cases for client-side computations in browsers. Bostock et al. developed D3.js [14], a declarative data-oriented DOM manipulation API that can create scalable vector graphics for an arbitrary number of statistical charts. Observable Plot [15] is built upon D3 for a more user-friendly API. Chart.js [4] occupies the middle ground; it trades off some amount of flexibility for a drastically simpler API to create HTML5 Canvas elements, using GPU support. Vite [3] is a front-end build tool for near-instantaneous hot-module replacement while in development and optimizes ES-module bundling for production.

III. SYSTEM ARCHITECTURE AND DESIGN

There is no physical server during runtime and the logical architecture's three tiers; presentation (HTML/CSS components), business logic (JavaScript service modules) and persistence (IndexedDB) are logically separated. The use of this three-tier logical architecture , similar to the three-tier architecture described by Gamma et al. [16] provides separation between analytical and rendering concerns.

Table 1. System Architecture of the Proposed EDA Platform

Layer / Module	Technology Stack	Responsibility
Client Layer (Browser)	Vite + HTML5/CSS3/JS ES6+	CSV file upload via FileReader API; interactive Chart.js dashboards; real-time parameter controls
Service Layer (JS Modules)	DataService, AnalysisService, VisualizationService	Stateless orchestration: preprocessing pipelines, statistical engines, report assembly
Analysis Engine	statsLib.js (custom)	Descriptive statistics, Z-score, IQR outlier detection, Pearson r computation
Visualization Engine	Chart.js 4.x	Bar, Line, Scatter, Histogram, Box-plot, Heatmap, Pair-plot rendering
Storage Layer (IndexedDB)	idb 8.x wrapper	Persistent project/dataset/report records: 8-entity relational schema
Export Module	jsPDF + html2canvas	Single-page and multi-section PDF report generation

In the persistence tier, there are 8 entities which form a relational schema with the cardinality and foreign key restrictions as follows. The user's entity (PK:user_id, attributes:username, email, password, created_at) is the root anchor for the schema. Each user can own zero to many projects (PK:project_id, FK:user_id, attributes:project_name, description) and have one setting (PK:setting_id, FK:user_id, attributes:theme, language). Each project may have many datasets (PK:dataset_id, FK:project_id, attributes:dataset_name, file_path, uploaded_at). Each dataset may have one data preprocessing (PK:preprocess_id, FK:dataset_id, attributes:missing_values, duplicates, normalisation) and zero or more analysis results (PK:analysis_id, FK:dataset_id, attributes:mean_value, median_value, variance, std_deviation). Visualisations (PK:visualisation_id, FK:dataset_id, attributes:chart_type, created_at) are created from completed analyses and project-level reports (PK:report_id, FK:project_id, attributes:report_name, created_at) are generated. This schema was developed to eliminate transitive functional dependencies to meet Date's description of third normal form (3NF) [17].

The FileReader Web API transforms a CSV file into a two-dimensional Array of JavaScript Objects when a dataset is uploaded. The DataService module counts the number of rows, counts the number of columns, and validates the header line of the CSV file and rejects it if the file is not valid. The PreprocessingService performs four sequential functions on the imported data; (i) perform missing-value imputation (median substitution for numerical values; mode for categorical values), (ii) remove any duplicate rows, (iii) calculate the min-max normalisation for statistics which are sensitive to scale, (iv) perform ordinal encoding of categorical variables. The AnalysisService calculates both descriptive statistics and the complete Pearson correlation matrix. Finally, the VisualisationService connects the processed Arrays to Chart.js chart instances and defines the interactive tooltip and zoom plugin callbacks for each chart instance.

IV. DATA ANALYSIS TECHNIQUES AND IMPLEMENTATION

The AnalysisService generates metrics for all numeric columns by using the following collection of descriptive statistics: arithmetic mean (μ), median ($\tilde{x}$), mode, sample variance (s^2), sample standard deviation (s), minimum, maximum, range, skewness (γ_1) and excess kurtosis (γ_2). These metrics are defined according to the definitions presented in McKinney [18] and James, et al. [10], with the calculations performed for skewness and kurtosis implementing the Fisher-Pearson method of correction to provide unbiased estimates based on finite sample sizes [9]. The output is presented as an overall summary table and also used in the visualisation layer (e.g. charts) with reference lines at mean $\pm$ 1σ.

The analysis of the correlation coefficients (denoted as $r_{ij} = Cov(Xi, Xj)/(\sigma_i \times \sigma_j)$) for all pairs of numeric variables is also presented as colour-coded heat-map data where the absolute values of r are ≥ 0.7 are highlighted in amber to provide a means of identifying where strong linear associations exist within the data [10]. In addition to the colour-coded heat-map cells, the system creates a dynamically generated descriptive report that specifies the top three positive correlations, the top three negative correlations, their respective magnitudes, and a plain-language interpretation in accordance with a template. This important note also alerts the user to the critical principle that correlation does not imply causation [11] by having a persistent information banner displayed beneath the heat-map.

The available methods for determining outliers are Z-score and IQR (interquartile range). For the Z-score method, the user enters a numeric value, and the system calculates its value as $z_i = (x_i - \mu) / \sigma$. Values for which the absolute value of z_i are greater than 3 are identified and represented as red points in scatter-plot views and as red bars in the whisker portions of box-plot views. For the IQR method, Q1, Q3, and IQR are computed using an exclusive definition for quartiles, consistent with Tukey [2]. Values for which are outside the calculated range ([Q1 − 1.5 × IQR, Q3 + 1.5 × IQR]) will be considered outliers. Each outlier will be reconciled in a separate report listing the outlier value, the location (row index) where it was found, the column where it was found, the outlier's raw value, and its z-score or fence-distance.

Prior to conducting any statistical analysis, three data-quality correction procedures will be applied to the data during the preprocessing pipeline: (i) Missing Data: empty string or JavaScript NaN values identified as missing are replaced with either the median value of the numeric column or the mode of the categorical column in order to minimise the distortion of the overall distribution of the data relative to listwise deletion (McKinney, 2018) [18]; (ii) Duplicate Rows: duplicate rows are identified based on lexicographic equality of all field values and removed. The number of duplicate rows removed will also be provided to the user; and (iii) Whitespace Normalisation: All leading/trailing whitespace will be removed from all string fields to eliminate potential anomalous variances created by extra whitespace possibly splitting the same category into multiple categories. All preprocessing procedures will be logged in the data_preprocessing IndexedDB entity of the database in order to provide reproducibility of all preprocessing actions.

V. DATASET DESCRIPTION AND EXPERIMENTAL RESULTS

In this study, a worker profile data set was utilized for the purpose of verification with 1,000 records (cases) and 12 fields (columns). The data set is intended to approximation the actual-distribution characteristics found within similar data sets found within existing public domain human resource analytics. A detailed summary of complete data set information for each field is found in Table 2, providing the field characteristics (i.e., data type, possible range of value and summary statistics) including the percentage of missing values.

Table 2. Employee Dataset Column-Level Descriptive Summary (n = 1,000)

Column	Type	Min	Max	Mean/Mode	Std Dev	Missing %
Age	Numeric	28	45	34.50	8.75	0.28
Income	Numeric	30,000	120,000	75,432	15,291	0.35
Experience	Numeric	1	20	7.60	4.30	0.02
Salary	Numeric	25,000	115,000	72,500	14,500	0.00
Performance	Numeric	1.00	5.00	3.45	0.85	0.28
Satisfaction	Numeric	1.00	5.00	3.70	0.90	0.50
Gender	Categorical	–	–	M/F	48%/52%	1.20

Education	Categorical	–	–	4 levels	–	0.80
Occupation	Categorical	–	–	5 roles	–	0.40
City	Categorical	–	–	6 cities	–	0.60
Department	Categorical	–	–	5 depts	–	0.20

The percent of missing values, overall, across all cases in this worker profile data set is 2.34%. However, within the fields of the employee profile data set, the percentages of missing data varies widely across fields, largely due to the way that missingness was artificially introduced/added as data entry error-artefacts that are similar to actual data collection errors. The distributions of the employee departments contained within this data are as follows: Engineering (22 percent), Operations (19 percent), Finance (18 percent), IT (21 percent) and Marketing (20 percent).

The correlation coefficients of the employee-numeric variables are displayed in the Pearson correlation matrix, Table 3. Within the Pearson correlation matrix, the strongest correlation occurred between Experience and Salary ($r = 0.70$, $p < 0.001$); the second highest correlation was between Experience and Income ($r = 0.50$). In contrast, there was a high negative correlation between Age and Salary ($r = -0.75$), indicating that older employees within this sample are holding positions with lower levels of pay, possibly reflecting the shift from seniority to specialization. There was also a moderate/positive correlation between Performance and Satisfaction show a moderate positive correlation ($r = 0.50$), demonstrating the platform's ability to detect mid-range linear associations.

Table 3. Pearson Correlation Matrix — Numeric Variables (n = 1,000)

Variable	Age	Income	Experience	Salary	Performance	Satisfaction
Age	**1.00**	0.45	0.62	-0.75	0.28	0.33
Income	0.45	**1.00**	0.50	0.35	0.35	0.33
Experience	0.62	0.50	**1.00**	0.70	0.32	0.02
Salary	-0.75	0.35	0.70	**1.00**	0.00	0.20
Performance	0.28	0.35	0.32	0.00	**1.00**	0.50
Satisfaction	0.33	0.33	0.02	0.20	0.50	**1.00**

Using the Z-score method, out of a total of 1,000 cases, there were a total of 23 (2.3%) calculated outliers within the numeric data; most of the outliers were in the Income field (where there were 11 total outliers) and in the Salary field (where there were 9 total outliers). Using the IQR outlier method, the total count of identified outliers was 18 (1.8%); this conservative design is consistent with the increased robustness of the IQR method with respect to heavy tailed distributions. All identified outliers for purposes of this analysis were attributable to artificial data-entry errors and thus were not excluded from the correlation analysis nor were they imputed; this is consistent with the recommendations that Hastie et al. have made to retain influential observations where there has not been established substantive construct validity.

VI. VISUALIZATION MODULE AND USER INTERFACE

A histogram can be used to analyse distribution traits (shape, skewness and modality) by grouping data points into k ($k = \lfloor \text{Ceil}(\sqrt{n}) \rfloor$) bins, and plotting the number of instances in each bin against the

corresponding value range on a separate y-axis. A scatter plot can be used to plot data on two user selected dimensions (variables), with an option to add a regression line overlaid on the scatter plot (using least squares methods). There are two types of bar chart visuals: grouped mode and stacked mode, which allow for comparison of multiple categorical dimensions against each other. A box plot is displayed by way of the Tukey [2] definition for Q1, median, Q3, lower fence, upper fence, plus outliers. A diverging colour scale (blue → white → red) illustrates the correlation between the two variables (both are dependent on the correlation coefficient r). When reviewing the correlation heat map, colours reflecting the correlation coefficients from -1 (negative correlation) to +1 (positive correlation) are used based on Cleveland's [8] perceptual guidelines for sequential and diverging colour scales.

Several interactive features are made available through Chart.js plugin architecture to improve user experience on the dashboard end, including: i) Cross-chart filtering (if the user clicks a department name on a bar chart, the filtering will also occur on the other charts in the dashboard); ii) Tooltip enhancement when hovering on data points (row number, raw value, z-score, and quartile information will be shown in a tooltip); and iii) Date range selection (date range is displayed in time series with use of dual-handle slider). The dashboard is laid out in a 2x2 grid format and is responsive to the width of the user's viewport (e.g., when the width is below 768 pixels it will become a single column); this follows a "mobile-first" design approach to the dashboard.

The reporting module generates five types of standard reports: Dataset Summary Report, Column Analysis Report, Missing Values Report, Correlation Report, and Data Quality Reports. Each report is generated in the browser from an HTML template, and will be exported to a multi-page PDF using jsPDF 2.5 and html2canvas 1.4; and the report metadata (report_name, created_on, format, row_count) are stored in the indexedDB entity associated with the reports, so that the same reports can be accessed on subsequent logons to the Reporting Module. The same features are similar to what Knaflic [7] describes as an academic requirement for documentation; to present the structured narrative in conjunction with chart graphics showing how the measurements produced the output data represented.

VII.CONCLUSION

A complete native browser EDA solution that combines CSV file ingestion, data cleaning, summary statistics, Pearson correlation analysis, IQR and z-score outlier detection, interactive dashboards built with Chart.js, and a PDF report generation into one web-based application has been described in this paper. A synthetic employee dataset was created containing 1,000 rows and 12 columns to validate the platform's functionality and demonstrate its ability to generate statistical benchmarks identical to those produced in an independent manner through Python using Pandas 2.0 and Seaborn 0.13 with an average absolute error (AAE) of less than 0.001 for each individual scalar output.

The evaluation of the proposed EDA solution demonstrated that it occupies a completely unique niche; it is as capable of pipeline integration and analytical complexity as programming languages, yet has the same amount of initial setup effort and requisite technical expertise as spreadsheet applications. Therefore, it represents a valuable tool for domain experts seeking to perform thorough initial investigations of their data without the overhead of writing scripts. Future development will focus on adding: (i) support for datasets that have over 500,000 rows using Web Workers and lazy-evaluating pipelines; (ii) integration of preprocessing utilities (PCA, feature importance ranking) for machine learning using TensorFlow.js; (iii) the ability for multiple users to provide collaborative annotations of charts via a small WebSocket-based back end; and (iv) dataset sharing with OAuth2 authentication, all in accordance with the multi-user relational schema.

REFERENCES

[1] IDC, "Worldwide Global DataSphere Forecast, 2024–2028: The AI-Driven Data Explosion," IDC Document #US52012424, Framingham, MA: IDC, May 2024. (Supported by updated consensus tracking via Statista's Global DataSphere Metrics, 2025/2026).

[2] J. W. Tukey, *Exploratory Data Analysis*. Reading, MA: Addison-Wesley, 1977.

[3] E. You, "Vite: Next Generation Front-End Tooling," 2021. [Online]. Available: https://vitejs.dev

[4] Chart.js Contributors, "Chart.js — Simple yet flexible JavaScript charting library," 2023. [Online]. Available: https://www.chartjs.org

[5] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55

[6] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*, 2nd ed. New York: Springer, 2016.

[7] C. N. Knaflic, *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Hoboken, NJ: Wiley, 2015.

[8] W. S. Cleveland, *Visualizing Data*. Summit, NJ: Hobart Press, 1993.

[9] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA: Morgan Kaufmann, 2011.

[10] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 2nd ed. New York: Springer, 2021.

[11] D. Huff, *How to Lie with Statistics*. New York: W. W. Norton, 1954.

[12] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed. Hoboken, NJ: Wiley, 2018. ISBN: 978-1-119-40030-1

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York: Springer, 2009.

[14] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011. DOI: 10.1109/TVCG.2011.185

[15] Observable HQ, Inc., "Observable Plot," 2021. [Online]. Available: https://observablehq.com/plot

[16] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1994.

[17] C. J. Date, *An Introduction to Database Systems*, 8th ed. Reading, MA: Addison-Wesley, 2003. ISBN: 978-0-321-19784-9

[18] W. McKinney, *Python for Data Analysis*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.

[19] pavansubhasht, "IBM HR Analytics Employee Attrition & Performance," Kaggle Datasets, 2017. [Online]. Available: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

[20] E. A. Locke, "The Nature and Causes of Job Satisfaction," in *Handbook of Industrial and Organizational Psychology*, M. D. Dunnette, Ed. Chicago, IL: Rand McNally, 1976, pp. 1297–1349.

[21] A. Sovkhozov, "Design and Implementation of an Application for Exploratory Data Analysis and Visualization of Real-World Dataset," B.Eng. thesis, Vistula University, Warsaw, Poland, 2026. Reg. No. V-66441-INZ-2026.

[22] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL: CRC Press, 2014.