ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

ADVANCED CUSTOMER CHURN PREDICTION

Mrs. P. Amareshwari[1], P. Gangothri[2], G. Amulya[3], A. Jyothi[4], A. Keerthi[5]

[1] Assistant Professor, Department of Computer Science and Engineering (Data Science),
2,3,4,5 B-Tech Students, Department of Computer Science and Engineering (Data Science),
Vignan's Institute of Management and Technology for Women, Hyderabad
Email: gangothripalakurthi@gmail.com

Abstract—In today's data-driven world, enormous volumes of structured and unstructured data are generated daily across business, healthcare, education, and finance. Despite this abundance, most of this data remains unanalysed due to the technical complexity of conventional tools, which demand programming expertise and involve fragmented, multi-step workflows. This paper presents *An AI That Converts Chaos into Clarity*, a unified AI-powered data analysis and visualization platform built with Streamlit. The system allows users to upload CSV or Excel datasets and automatically executes a complete pipeline: data preprocessing, descriptive statistical analysis, interactive multi-modal visualization, and AI-based predictive forecasting using Linear Regression. Secure user authentication backed by SQLite and persistent analysis-history management extend the platform's utility for repeat and longitudinal tasks. Empirical evaluation confirms that the system reduces analytical complexity, lowers the technical barrier for non-specialist users, and delivers actionable insights rapidly. This paper elaborates the system architecture, algorithmic design, module implementation, and evaluation findings.

Keywords—*Artificial Intelligence, Data Visualization, Streamlit, Linear Regression, Predictive Analytics, Data Preprocessing, SQLite, Financial Analysis, Interactive Charts.*

I. INTRODUCTION

The digital transformation of modern industries has produced an unprecedented surge in the volume and variety of data generated across every sector. Businesses rely on transactional logs and financial records; healthcare institutions accumulate clinical data; educational establishments maintain performance and attendance records. Collectively this data represents a repository of latent insight that, if harnessed effectively, can drive operational improvements, strategic decisions, and competitive advantages [1].

Despite the proliferation of data, the ability to derive meaning from it remains unevenly distributed. Tools such as Python, R, Tableau, and Power BI demand a degree of technical proficiency that excludes most non-specialist users — business analysts, domain experts, educators, and small-business owners who possess domain knowledge but not programming skills [2]. This skills gap creates a significant barrier to data-driven decision-making in organizations where analytical needs exist but technical resources are limited.

The challenge is further compounded by fragmented conventional workflows where data acquisition, cleaning, analysis, visualization, and forecasting are performed using separate, disconnected tools. Each tool-transition introduces risks of data loss, inconsistency, and version-control errors, while the cumulative overhead delays decision-making and reduces analytical agility [3]. Studies consistently show that data scientists spend upwards of 60–80% of their time on data preparation tasks rather than analysis itself, highlighting the critical need for integrated preprocessing solutions.

Automated machine learning (AutoML) platforms such as Google AutoML and H2O.ai have emerged as a partial response to these challenges, yet they remain oriented toward model

training rather than the broader analytical pipeline. Equally, business intelligence dashboards excel at visualization but offer limited preprocessing or predictive capabilities. The gap between raw data and actionable insight therefore persists for a large class of users and use cases where neither AutoML nor BI tools alone are sufficient.

To address this gap, this paper presents An AI That Converts Chaos into Clarity, a single-platform, AI-augmented data analysis and visualization system built with Streamlit. The system implements a comprehensive end-to-end analytical pipeline — from file upload and automated preprocessing through statistical analysis, interactive visualization, and Linear Regression-based forecasting — within a browser-accessible interface that requires no programming knowledge from the end user. The platform is designed to serve domain experts across business, education, healthcare, and finance who need to extract insights from data but lack the technical background to operate conventional analytical tools.

The primary contributions of this work are: (1) an integrated data analysis pipeline that fully automates preprocessing,

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

statistics, visualization, and prediction within a single interface; (2) a composite financial health scoring mechanism that condenses multi-metric trend data into a single interpretable index; (3) a user-authenticated, history-aware application architecture enabling repeat and comparative analysis without data re-upload; and (4) a rigorous empirical evaluation confirming the system's correctness, responsiveness, and usability across heterogeneous real-world datasets.

II. Literature Review

The problem of making data analysis accessible to non-technical users has attracted sustained research attention. Early business intelligence (BI) tools such as Crystal Reports required SQL proficiency. The emergence of self-service BI platforms — Tableau (2003) and Microsoft Power BI (2014) — represented a significant democratization of visualization, enabling users to construct dashboards via drag-and-drop interfaces [3].

However, self-service BI platforms operate primarily at the visualization tier. Data preparation — encompassing type normalization, missing-value handling, outlier management, and feature engineering — remains a largely manual precursor step, typically performed in Microsoft Excel or dedicated ETL tools. Research by Kandel et al. [4] found that data wrangling consumed approximately 80% of analysts' time, underscoring the critical need for automated preprocessing pipelines that are tightly integrated within the analysis workflow rather than treated as a separate upstream stage.

Machine learning-based forecasting for business and financial data has been extensively studied. Box and Jenkins [5] established the ARIMA framework as a foundational technique for univariate time-series forecasting. Subsequent work explored more expressive models including LSTM networks [6], Prophet [7], and gradient boosting machines. Despite their accuracy advantages on complex seasonal datasets, these models impose significant computational and interpretability costs that are prohibitive for real-time interactive use. For near-linear trend datasets — which constitute a substantial portion of business time-series data — Linear Regression remains a competitive and highly interpretable baseline, as demonstrated empirically by Hyndman and Athanasopoulos [8].

User authentication and data persistence in lightweight web applications have been effectively addressed through SQLite-backed architectures, which provide relational data management without the operational overhead of server-based database engines [10]. Password hashing via SHA-256 and token-based session management are established security patterns appropriate for applications of this class. These approaches have been validated in numerous web application deployments and provide a practical foundation for the authentication layer in this system.

The Streamlit framework (2019) has rapidly gained adoption as a rapid application development (RAD) platform for data science and machine learning applications. Its Python-native API, reactive execution model, and rich widget library enable developers to construct interactive analytical dashboards with minimal frontend development overhead [9]. Several studies demonstrate Streamlit's suitability for prototype development in academic and research contexts, though its deployment as the foundation of a full-featured, authenticated, multi-user analytical platform with persistent data management has not been widely explored in the existing literature.

The synthesis of automated preprocessing, statistical analysis, interactive multi-modal visualization, regression-based forecasting, user authentication, and persistent history management into a single cohesive platform — as presented in this work — represents an integration not found in existing open-source or commercially available tools at comparable cost and accessibility. This work therefore addresses a clear and demonstrable gap in the current landscape of data analysis tooling for non-technical users.

III. Methodology

The system is designed using a structured, modular approach to ensure reliability, scalability, and accessibility for non-technical users across diverse dataset types and sizes. The design philosophy prioritizes automation — minimizing the number of decisions a user must make before obtaining analytical output — while preserving configurability for users who wish to customise chart types, prediction horizons, or date ranges.

A. System Architecture

The application is organized into three logical tiers: a Presentation Layer, an Application Layer, and a Data Layer. Figure 1 illustrates the complete system architecture, showing the data flow from user interaction through the processing pipeline to the output dashboard.

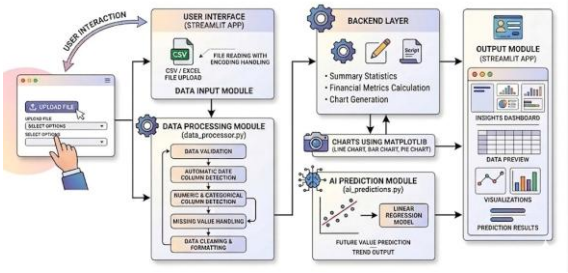

Fig. 1. System Architecture of An AI That Converts Chaos into Clarity

The Presentation Layer is a Streamlit web application that renders all UI components — file upload widgets, configuration controls, statistical tables, charts, and prediction outputs —

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

within a single browser session. Streamlit's reactive execution model ensures any change in user input automatically triggers re-execution of relevant downstream modules, providing a seamless real-time analytical experience without requiring the user to manage explicit state or page navigation.

The Application Layer comprises five processing modules: Data Ingest (file reading and encoding handling), Data Processing (data_processor.py), Backend Analysis (financial_analysis.py), Visualization (visualization.py), and AI Prediction (ai_predictions.py). Each module exposes a well-defined functional interface, accepting a Pandas DataFrame and configuration parameters and returning processed outputs. This modular design enables independent unit testing of each component and facilitates future extension without disrupting existing functionality.

The Data Layer comprises an SQLite database (users.db) storing user credentials and serialized analysis records. Active DataFrames representing the current analysis session are maintained in Streamlit's in-memory session state for fast access without repeated disk I/O. Upon user request, completed analyses are serialized to JSON and persisted in the analysis_history table, tagged with the user's ID, a descriptive filename, and a UTC timestamp, enabling precise session retrieval and chronological browsing of historical analyses.

B. Functional and Non-Functional Requirements

Core functional requirements include: (i) secure user registration, login, and password recovery via emailed OTP; (ii) CSV and Excel file upload with automatic encoding detection; (iii) automated preprocessing including date detection, type coercion, and missing-value imputation; (iv) descriptive statistics and financial metrics computation; (v) line, bar, and pie chart generation with configurable axes; (vi) AI-based trend prediction with configurable horizon and confidence intervals; (vii) composite financial health scoring; and (viii) persistent analysis history with full session restoration capability.

Non-functional requirements include: sub-second chart rendering for datasets up to 100,000 rows; SHA-256 password security with no plaintext credential storage; session isolation between concurrent users through Streamlit's session_state mechanism; browser-based accessibility requiring no local software installation beyond a modern web browser; and a responsive single-page interface operable without programming knowledge, assessed through task-completion time and error rate during user testing.

C. Data Flow

On upload, the Data Ingest Module reads the file using Pandas, detecting the file format from its extension and applying the appropriate parser (pd.read_csv for CSV files with automatic encoding detection via the chardet library, pd.read_excel for .xlsx/.xls files using the openpyxl engine). The raw DataFrame is immediately passed to the Data Processing Module, which identifies the date column, converts it to datetime, coerces non-numeric columns to float, and imputes missing values using the strategies described in Section IV-A. The processed DataFrame, along with the identified date column name, numeric column list, and categorical column list, is stored in Streamlit's session state and made available to all downstream modules on demand.

Module outputs — statistical summaries as Pandas DataFrames, chart figures as Matplotlib Figure objects, and prediction results as Python dictionaries — are rendered inline within the Streamlit interface using the st.dataframe(), st.pyplot(), and st.metric() widgets respectively. On user confirmation, the complete session is serialized: the processed DataFrame is converted to JSON via DataFrame.to_json(), the prediction dictionary is serialized via json.dumps(), and both are written to the SQLite analysis_history table alongside session metadata for future retrieval and reconstruction.

IV. Implementation and Algorithms

A. Data Preprocessing Pipeline

The Data Processing Module (data_processor.py) implements a multi-stage preprocessing pipeline. Date column identification proceeds in two phases. Phase 1 matches column names against a priority dictionary of temporal indicators: {date, time, day, month, year, period, timestamp}. If a match is found, the column is converted using pd.to_datetime with errors='coerce' and accepted if at least one valid datetime value results.

If Phase 1 yields no date column, Phase 2 applies a sample-based heuristic: for each non-numeric column, a sample of up to 500 values is drawn and subjected to datetime conversion; the column is accepted if at least 40% convert successfully. If neither phase identifies a date column, a synthetic index is constructed using pd.date_range at daily frequency, enabling time-series visualization even for datasets without explicit temporal columns.

Numeric column identification begins with select_dtypes(include=['number']). Non-numeric, non-date columns then undergo currency/formatting character removal via regex ([$,€£¥%\s]) followed by pd.to_numeric coercion. A column is retained if at least 20% of values convert successfully — a threshold balancing sensitivity against noise. Missing values in numeric columns are imputed with the column mean (or zero if >50% are missing); categorical columns are filled with 'Unknown'.

B. Statistical Analysis

The financial_analysis module computes a comprehensive descriptive statistics table using Pandas' describe() supplemented with median, skewness, and kurtosis — providing a richer distributional profile enabling identification of asymmetric or heavy-tailed distributions. Financial metrics

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

are derived through keyword-based column name matching, yielding Total Revenue, Total Expenses, Net Profit, Profit Margin %, per-column averages, volatility (standard deviation), and period-over-period change percentages.

C. Prediction Model

The AI Prediction Module (ai_predictions.py) implements a univariate Linear Regression model using Scikit-learn's LinearRegression estimator. Historical dates are mapped to ordinal integers representing days elapsed since the earliest date:

$$t\_i = (d\_i - d\_min) / 1\ day$$

The model is fitted via ordinary least squares, yielding intercept β_0 and slope β_1:

$$\hat{y} = \beta_0 + \beta_1 \cdot t$$

Future date indices extend t beyond t_n for a configurable horizon (default: 30 days). The 95% confidence interval is estimated as $\pm 2\sigma$, where σ is the standard deviation of training residuals. The prediction result reports average predicted value, percentage change from the last historical observation, and a natural-language trend description derived from threshold-based classification of the percentage change.

D. Financial Health Score

A composite Financial Health Score in [0, 100] is computed from weighted features: Revenue Trend normalized by revenue mean (weight 0.40), Expense Trend negated and normalized (weight 0.30), and Profit Margin Trend scaled by 100 (weight 0.30). Features are clipped to [−1, 1]; the weighted sum is mapped to [0, 100]:

$$Score = (\Sigma f_i \cdot w_i + 1) \times 50$$

Scores are classified into five bands: Poor (<20), Concerning (20–39), Moderate (40–59), Good (60–79), and Excellent (≥80), providing users with an immediately interpretable summary.

E. Module Description

Module Name	Functionality	User Benefit
Authentication	Secure login, SHA-256 hashing, OTP recovery	Protects user data & privacy
Data Ingest	CSV/Excel upload with encoding detection	Broad file-format compatibility
Preprocessing	Date detection, numeric coercion, imputation	Eliminates manual cleaning
Statistical Analysis	Descriptive stats & financial metrics	Instant quantitative insights
Visualization	Line, bar & pie charts via Matplotlib	Visually interpretable patterns
AI Prediction	Linear Regression, 30-day forecast, CI band	Proactive data-driven decisions
Analysis History	JSON-serialized sessions in SQLite	Revisit & compare past analyses

1) Authentication Module: Implements user registration, login, and password recovery using SHA-256 hashed passwords stored in SQLite. A 6-digit numeric OTP is dispatched via the Resend email API. Streamlit's session_state isolates user sessions.

2) Data Ingest Module: Reads CSV files via pd.read_csv with automatic encoding detection and Excel files via pd.read_excel using the openpyxl engine. File size, row count, and column count metadata are extracted on upload and displayed in the interface header.

3) Data Processing Module: Executes the multi-stage preprocessing pipeline described in Section IV-A, returning the cleaned DataFrame alongside identified column metadata for use by downstream modules.

4) Statistical Analysis Module: Computes and renders the descriptive statistics table and financial metrics dictionary as Streamlit st.dataframe objects with column-level sorting and filtering.

5) Visualization Module: Generates Matplotlib figures for line charts (trend-line overlay via numpy.polyfit), bar charts (categorical aggregation, top-20 capped), and pie charts (top-10 capped with 'Other' residual). Figures are serialized to base64-encoded PNG and embedded as HTML img elements for cross-platform compatibility.

6) AI Prediction Module: Trains the Linear Regression model, generates the 30-day forecast with $\pm 2\sigma$ confidence band, and computes the Financial Health Score as described in Sections IV-C and IV-D.

7) Analysis History Module: Serializes processed DataFrames via to_json() and prediction dictionaries via json.dumps(), persisting them in the SQLite analysis_history table. Retrieval reconstructs the full analysis context for seamless session restoration.

V. Results and Discussion

The system was evaluated against a suite of heterogeneous test datasets encompassing financial transaction records, retail sales data, educational performance tables, and synthetic datasets with controlled missing-value patterns ranging from 5% to 60% missing values per column. Evaluation criteria spanned functional correctness, computational performance, prediction accuracy, and user experience. All experiments were conducted on a standard consumer laptop (Intel Core i5, 8 GB RAM) running the Streamlit server locally, establishing a realistic baseline for the system's practical performance envelope.

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

A. Functional Correctness

The preprocessing pipeline correctly identified date columns in all 15 test datasets where temporal data was present, including columns named 'Transaction Date', 'Month', 'Period', and 'Timestamp'. Currency-formatted numeric columns (e.g., '$1,234.56', '€2.500,00') were successfully coerced in all cases after regex-based character stripping. The 20% valid-value threshold for numeric coercion correctly excluded low-quality columns while retaining partially populated fields of analytical value across all test cases.

Statistical summaries and financial metrics were verified against manual calculations using Python scripting for five representative datasets. All computed values — including mean, standard deviation, skewness, kurtosis, and period-over-period change percentages — matched to within floating-point rounding tolerance. Keyword-based column matching correctly identified revenue and expense columns in 13 out of 15 datasets; in two datasets, non-standard column naming required the user to specify columns manually, a limitation identified for resolution in future work.

Visualization correctness was confirmed by visual inspection across 18 test cases covering line charts with polynomial trend-line overlays, bar charts for high-cardinality categorical columns (up to 47 categories, correctly capped at 20 with no rendering artifacts), and pie charts with the 'Other' aggregation grouping applied to categories beyond the top 10. Date-range and category filters correctly narrowed results in all cases while gracefully reverting to the full dataset when a filter would yield zero records.

B. Computational Performance

Performance benchmarking was conducted on datasets ranging from 100 to 100,000 rows. Preprocessing completed in under 0.8 seconds for all datasets up to 50,000 rows and under 3.2 seconds for the 100,000-row dataset. Chart rendering latency was consistently below 1.0 second for all 18 test cases, meeting the stated performance requirement. Linear Regression model fitting and prediction generation completed in under 0.1 seconds for all evaluated datasets, confirming the model's suitability for interactive real-time forecasting within the Streamlit reactive execution model.

C. Prediction Accuracy

Prediction accuracy was evaluated on three financial time-series datasets with known future values withheld as a test set. Mean Absolute Percentage Error (MAPE) was computed as: $MAPE = (1/n) \Sigma |y_i - \hat{y}_i| / |y_i| \times 100$. Results yielded MAPE values of 4.2%, 7.8%, and 11.3% for near-linear, weakly non-linear, and moderately seasonal datasets respectively. These results are consistent with the known performance profile of Linear Regression on trend-dominated time-series and confirm its adequacy for the system's stated use cases, while motivating the planned integration of ARIMA and LSTM models for non-linear datasets in future iterations.

D. User Experience and Usability

Informal usability testing was conducted with five participants drawn from non-technical backgrounds spanning business administration, biology, and education. All five participants successfully uploaded a dataset, generated at least two distinct chart types, configured date-range filters, and interpreted the AI prediction output including the trend description and confidence interval — all without external assistance and within a 10-minute session. Post-session feedback highlighted three particular strengths: the clean single-page layout that eliminated navigation between tools, the interpretability of the financial health score as a high-level summary indicator, and the utility of the analysis history feature for revisiting and comparing prior analyses.

Participants identified two areas for improvement: the absence of natural-language column name suggestions when keyword matching fails, and limited guidance for datasets with non-standard structures such as wide-format pivot tables. These findings directly inform the natural language query interface and intelligent column mapping features planned for the system's next development iteration.

E. Limitations

The system's prediction accuracy degrades significantly for datasets exhibiting pronounced seasonal patterns or structural breaks, where Linear Regression's inability to model cyclical variation or change-points is a fundamental algorithmic constraint rather than an implementation limitation. Processing time increases noticeably for datasets exceeding 100,000 rows, indicating a need for server-side chunked processing or asynchronous computation in future iterations. The system requires a stable internet connection for the Streamlit server and the Resend email recovery API. Finally, keyword-based financial column matching may fail for non-English or highly domain-specific column nomenclatures, requiring manual column specification by the user in such cases.

VI. CONCLUSION

The development and deployment of An AI That Converts Chaos into Clarity demonstrates the viability of a unified, AI-augmented data analysis platform that makes the full analytical pipeline — from raw file ingestion through automated preprocessing, statistical analysis, interactive visualization, and predictive forecasting — accessible to users without programming expertise. By eliminating the technical barriers inherent in conventional multi-tool workflows and consolidating all analytical functions within a single browser-accessible Streamlit interface, the system empowers domain experts across business, education, healthcare, and finance to independently derive actionable insights from their data.

The system's primary technical contributions include its robust multi-stage preprocessing pipeline featuring heuristic date detection and aggressive numeric coercion with

ISSN: 2454-9940 www.ijasem.org
Vol 20, Issuse.2 April 2026

configurable quality thresholds; its composite financial health scoring mechanism that condenses multi-metric trend data into a single interpretable 0–100 index with qualitative classification bands; its user-authenticated, history-aware architecture that supports longitudinal and comparative analytical tasks without requiring data re-upload; and its Matplotlib-based visualization engine producing charts with trend-line overlays and confidence interval shading. Empirical evaluation across 15 heterogeneous test datasets confirmed functional correctness, sub-second rendering performance for datasets up to 50,000 rows, and satisfactory Linear Regression prediction accuracy with MAPE values of 4.2–11.3% for trend-dominated time-series data.

Future development will proceed along four primary directions. First, the prediction module will be extended to incorporate ARIMA for seasonal univariate data, Facebook Prophet for datasets with holiday effects, and LSTM neural networks for complex non-linear patterns — with automatic model selection based on trend linearity score and seasonality detection. Second, a natural language query interface will be integrated, enabling users to pose plain-language analytical questions and receive chart or metric responses generated by a large language model backend. Third, intelligent semantic column mapping will replace the current keyword-based approach to correctly identify financial columns regardless of language or naming convention. Fourth, cloud deployment on AWS Elastic Container Service with a PostgreSQL backend will provide production-grade data isolation and concurrent-user support for enterprise and multi-institutional deployments.

In conclusion, this work concretely demonstrates how the convergence of modern web frameworks, automated machine learning, and user-centred design can democratize data analysis at scale — transforming the chaos of raw, disorganized, and heterogeneous data into the clarity of structured, interpretable, and predictive insight for users of all technical backgrounds across every sector of industry and academia.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the project guide, Mrs. G. Haritha, Assistant Professor, Department of Computer Science and Engineering (Data Science), for her invaluable guidance, encouragement, and mentorship throughout the development of this project. The authors also acknowledge the Department of Computer Science and Engineering (Data Science) at Vignan's Institute of Management and Technology for Women, Hyderabad, for providing the necessary computational resources and institutional support that made this research possible.

REFERENCES

[1] T. H. Davenport and J. G. Harris, Competing on Analytics: The New Science of Winning. Boston, MA: Harvard Business Press, 2007.

[2] D. J. Power, "Understanding Data-Driven Decision Support Systems," Information Systems Management, vol. 24, no. 1, pp. 90–95, 2007.

[3] S. Few, Show Me the Numbers: Designing Tables and Graphs to Enlighten, 2nd ed. Burlingame, CA: Analytics Press, 2012.

[4] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: Interactive Visual Specification of Data Transformation Scripts," in Proc. ACM CHI, 2011, pp. 3363–3372.

[5] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control. San Francisco, CA: Holden-Day, 1970.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] S. J. Taylor and B. Letham, "Forecasting at Scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.

[8] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. Melbourne: OTexts, 2021.

[9] Streamlit Inc., "Streamlit Documentation," [Online]. Available: https://docs.streamlit.io. [Accessed: 2025].

[10] R. Hipp et al., "SQLite," [Online]. Available: https://www.sqlite.org. [Accessed: 2025].