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.2026.79618

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

VisoryBI: A Template-Driven Offline Business Intelligence Platform with Integrated Statistical Analytics and Natural Language Querying

H.Mohamed Hashim, S.Mohamed Thasleem, S.Abdul Thowfik, M. Naveen, Mrs. J. Rosalind Geetha
Department of Artificial Intelligence and Data Science M.I.E.T. Engineering College, Trichy, Tamil Nadu, India

Abstract: ***Business Intelligence platforms have become indispensable in modern organizational workflows, yet most production-grade tools demand persistent internet access, vendor-managed servers, and considerable technical proficiency. These constraints effectively exclude a wide range of users, from small-business analysts to data teams operating in air-gapped environments. VisoryBI was built specifically to remove those barriers. It is an offline-capable, browser-native dashboard builder implemented as a Progressive Web Application that performs all computation, storage, and rendering on the client device without any backend infrastructure; after the initial load, the application operates without any network connection. The platform ships with more than seventeen chart types, sixteen domain-specific dashboard templates equipped with intelligent field-mapping, an integrated statistical engine performing Z-Score and Interquartile Range anomaly detection alongside linear-regression forecasting, and a client-side Natural Language Processing interface that lets users query their datasets in plain English. Persistence is handled through a two-tier storage model pairing Zustand-managed localStorage for lightweight state with IndexedDB for large datasets. Export pipelines support PDF, PNG, PowerPoint, and CSV output. A three-tier role-based access control system governs who may create, modify, or view content. Independent end-to-end testing across all modules recorded an overall working efficiency of approximately ninety-seven percent.***

Keywords: Business Intelligence, Dashboard Builder, Progressive Web Application, Natural Language Processing, Anomaly Detection, Time Series Forecasting, IndexedDB, Offline Analytics, React, Template Visualization.

I. INTRODUCTION

The volume of structured and semi-structured operational data generated by modern organizations has grown to a point where raw tabular access is no longer sufficient for timely decision-making. Dashboards that aggregate, visualize, and contextualize this data have consequently moved from a specialized reporting function to a core business process. Yet the platforms that dominate this space, Tableau, Microsoft Power BI, and Google Looker Studio among them, share a common architecture: they depend on cloud infrastructure, require organizational accounts or desktop installations, and assume a degree of data modeling literacy that most business users do not possess.

These assumptions create three distinct exclusion patterns. First, users in bandwidth-limited environments or organizations with strict data-sovereignty policies cannot reliably operate cloud-dependent tools. Second, the dashboard creation workflow in conventional platforms requires understanding of data schemas, dimension-versus-measure classification, and join logic, competencies that belong to data engineers rather than the domain experts who most need insight. Third, advanced analytics capabilities such as anomaly detection, trend forecasting, and conversational data querying either are absent from free tiers or demand premium licensing.

VisoryBI was designed to address all three patterns within a single browser-based application. The platform is implemented as a Progressive Web Application that, after its initial load, requires no network connection for any part of the normal workflow. Data never leaves the user's device. No server is installed, no database is configured, and no internet access is needed during data ingestion, analysis, or export. Despite this zero-infrastructure posture, the platform delivers capabilities that compete with paid BI tools: seventeen visualization types, sixteen domain-specific dashboard templates with automated field mapping, client-side anomaly detection, linear-regression forecasting, a natural language query interface, and export to four file formats including PowerPoint.

Research in dashboard design has advanced considerably in recent years. Setlur et al. [1] articulated a cooperative conversation framework comprising thirty-nine heuristics across five interaction states, demonstrating that existing platforms handle data initiation and grounding adequately but fall short in the turn-taking and repair states that characterize iterative exploration. Lin et al. [2] mined eight hundred and fifty-four Tableau dashboards to derive data-driven layout and coordination rules, achieving seventy-one to seventy-three percent classifier accuracy. Yang et al. [3] conducted the first eye-tracking study specific to multi-view dashboards, establishing that numeric KPI cards attract the highest saliency coverage and that stratified layouts encourage broader visual exploration. Zeng et al. [4] addressed the problem of adapting multiple-view compositions to small displays, showing that preserving view topology and aspect ratios is essential for analytical task completion.

The contributions of this paper are as follows:

- An offline-capable Progressive Web Application with seventeen visualization types and sixteen domain-specific dashboard templates, requiring no network connection after the initial load.
- A smart field-mapping engine that automatically assigns dataset columns to chart axes, KPI bindings, and filter controls based on inferred data types and naming heuristics.
- An integrated statistical analytics engine supporting Z-Score and IQR anomaly detection, ordinary least-squares forecasting with moving-average smoothing, and ten distributional summary statistics.
- A client-side NLP query interface supporting aggregation, comparison, temporal, and top-N intent categories without external API dependencies.
- Dashboard versioning, a three-tier role-based access control system, and multi-format export to PDF, PNG, PowerPoint, and CSV.
- Empirical validation across all modules recording approximately ninety-seven percent working efficiency.

II. RELATED WORK

A. *Dashboard Design Heuristics and Cooperative Interaction*

Setlur et al. [1] approached dashboard design as a cooperative conversation between the analyst and the artifact, deriving thirty-nine heuristics organized across five analytical states: initiation, grounding, turn-taking, repair and refinement, and closure. Their evaluation involved fifty-two graduate students and revealed a consistent deficiency in the turn-taking and repair states, precisely the moments when an analyst needs to reformulate a question or correct a misinterpretation. VisoryBI's NLP Ask Data interface and drill-down navigation are intended to give users a responsive turn-taking channel, while the undo/redo stack and dashboard versioning address the repair-and-refinement gap.

B. *Data-Driven Dashboard Layout and Coordination*

Lin et al. [2] presented DMiner, a framework that applies decision-tree classifiers to layout and coordination patterns extracted from 854 real-world Tableau dashboards. Their core findings, that text views belong in compact header regions, that comparison views should sit adjacent to one another, and that charts sharing common fields benefit from brushing coordination, are embedded in VisoryBI's template architecture: KPI cards occupy the top row, related trend charts are grouped in proximity, and cross-widget filtering propagates selections automatically.

C. *Eye-Tracking and Visual Attention in Dashboards*

Yang et al. [3] collected 2,133 eye-movement trajectories from sixty participants across 1,216 dashboard configurations, building a Dashboard Vision Saliency Model that surpasses single-view saliency baselines. Their empirical result, that numeric KPI cards attract the highest saliency coverage and that stratified layouts drive broader exploration, directly informs VisoryBI's default widget arrangement. KPI cards are placed at the top of every template, and the default grid follows a stratified structure that guides the viewer's eye through summary metrics before reaching detailed charts.

D. *Responsive Adaptation of Multi-View Visualizations*

Zeng et al. [4] proposed AdaMV, a two-stage framework that uses simulated annealing to minimize a composite cost over space utilization, aspect-ratio fidelity, relative sizing, and view topology when adapting desktop multi-view compositions to smaller displays. VisoryBI's PowerPoint export pipeline applies an analogous principle: each widget occupies a dedicated slide at its original aspect ratio, preserving the analytical integrity of the layout when dashboards transition from the browser to presentation contexts.

E. Research Gap

Prior work has established theoretical frameworks for heuristic-guided design [1], automated layout recommendation [2], attention-aware arrangement [3], and responsive adaptation [4]. What remains absent is a system that operationalizes these principles in an offline-capable, zero-installation platform accessible to non-technical users. VisoryBI fills this gap by unifying template intelligence, statistical analytics, and natural language querying within a browser application that requires no backend and stores all data locally.

III. PROBLEM STATEMENT

Three interrelated challenges impede the broad adoption of Business Intelligence tooling in organizations of all sizes.

The first is connectivity and deployment dependency. Enterprise BI platforms assume persistent internet access and managed server infrastructure. This disqualifies them for use in air-gapped networks, low-bandwidth field settings, and organizations operating under data-residency regulations that prohibit cloud transmission of sensitive records.

The second is the technical expertise barrier. Creating a meaningful dashboard in a conventional BI tool requires familiarity with relational data modeling, dimension-measure classification, and visualization grammar. Business analysts, department managers, and domain specialists, the people who most need timely insight, routinely lack this background, creating a bottleneck in which insight generation depends on scarce data engineering resources.

The third is the analytics capability gap in accessible tools. Free tiers of mainstream BI platforms omit advanced analytics such as anomaly detection, trend forecasting, and conversational querying. Users who need these capabilities are directed to premium subscriptions or forced to maintain separate statistical software, fragmenting the analytical workflow. VisoryBI addresses all three challenges within a single browser application that stores all data locally and operates without any network connection.

IV. PROPOSED SYSTEM

A. System Architecture

VisoryBI is a single-page application built with React 18.3.1 and TypeScript, bundled by Vite with Progressive Web App support provided through vite-plugin-pwa. The application operates entirely client-side. All data persistence is handled through a two-tier storage model: Zustand-managed localStorage for application state and idb-keyval-managed IndexedDB for large datasets. After the initial page load, no network requests are made during normal operation. The system decomposes into four functional modules corresponding to the phases of the dashboard creation lifecycle.

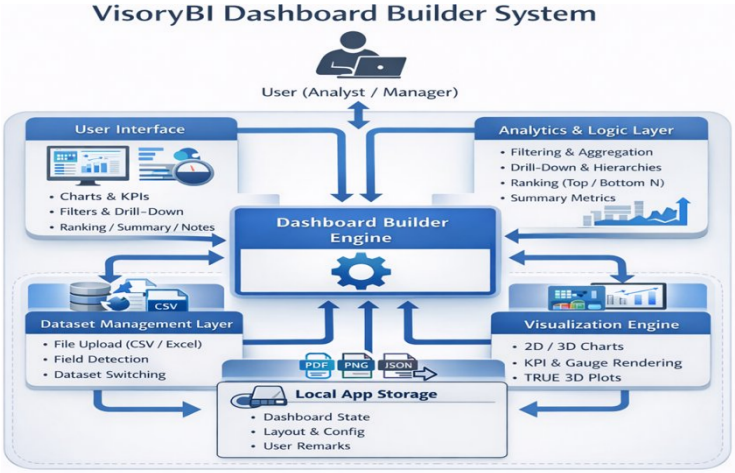


Fig. 1. VisoryBI Dashboard Builder System Architecture

1) Module 1 — Data Ingestion and Dataset Management

The ingestion module accepts CSV files processed through PapaParse and Excel workbooks processed through SheetJS. On upload, the system performs automatic column type inference, distinguishing categorical dimensions from numeric measures, and constructs a structured dataset schema. Datasets under two megabytes are retained in localStorage; datasets at or above that threshold are transparently routed to IndexedDB, eliminating the size constraints of localStorage.

The smart field-mapping engine operates at template selection time. It reads the inferred schema and applies naming heuristics and type rules to assign columns to chart axes, KPI metric bindings, and filter controls automatically.
A column named revenue or sales with a numeric type is bound to a value axis; a column named date, month, or period with a string or date type is assigned to the time axis of trend charts. This automation compresses dashboard setup from a multi-step manual process into a single user action.

2) Module 2 — Dashboard and Visualization Builder

The dashboard builder presents a drag-and-drop canvas powered by @hello-pangea/dnd. Users position widgets on a responsive grid and configure each widget's chart type, data bindings, and formatting through an edit dialog. All seventeen visualization types are rendered using Recharts 2.15, a composable charting library built on React and D3. Table I lists the complete chart inventory by category.

TABLE I — VisoryBI Visualization Types

Category	Chart Types
Standard	Bar, Line, Pie, Area, Scatter, Data Table
Metrics	KPI Card, Gauge, Sparkline
Advanced	Donut, H-Bar, Funnel, Treemap, Radar, Combo, Waterfall, Stacked Bar

The builder supports real-time data-field reassignment, cross-chart filter propagation, and drill-down navigation, addressing the turn-taking and repair interaction states identified as deficient by Setlur et al. [1]. Dashboard state is persisted continuously by Zustand, enabling session recovery after browser closure and supporting a full undo/redo operation history.

3) Module 3 — Advanced Analytics Engine

The analytics engine operates entirely in the browser. Anomaly detection is implemented using two methods: the Z-Score method marks observations whose standardized deviation from the sample mean exceeds two, while the IQR method marks values outside Q1 − 1.5×IQR and Q3 + 1.5×IQR, providing robustness for skewed distributions. Detected anomalies are highlighted directly on the associated chart.
Time series forecasting applies ordinary least-squares linear regression to historical data, computes slope, intercept, and R-squared as a goodness-of-fit metric, and projects N future periods. Moving-average smoothing with a configurable window reduces high-frequency noise before regression is applied. Summary statistics for any selected numeric column include mean, median, standard deviation, minimum, maximum, range, Q1, Q3, IQR, and skewness.

4) Module 4 — Export, Versioning, and Presentation

VisoryBI supports four export formats: PDF via jsPDF combined with html2canvas, PNG for individual widgets, PowerPoint via pptxgenjs 4.0 with one widget per slide at its native aspect ratio, and CSV for filtered or aggregated data. Presentation Mode activates a fullscreen auto-cycling display with configurable transition intervals and keyboard navigation. Dashboard versioning stores named snapshots through the versionStore, giving analysts an audit trail and a recovery mechanism for experimental changes.

V. TECHNOLOGY STACK

Table II summarizes the complete set of libraries and tools constituting the VisoryBI implementation. The selection prioritizes zero-dependency, browser-native libraries to maintain the offline-first design principle throughout the stack.

TABLE II — VisoryBI Technology Stack

Layer	Technology	Version
Framework	React	18.3.1

Language	TypeScript	5.x
Build Tool	Vite + vite-plugin-pwa	5.x / 1.2
Styling	Tailwind CSS + shadcn/ui	3.x
State Mgmt	Zustand (persist)	5.0
Routing	React Router DOM	6.30
Charts	Recharts	2.15
Data Parsing	PapaParse + SheetJS	5.5 / 0.18
Drag & Drop	@hello-pangea/dnd	18.0
PDF Export	jsPDF + html2canvas	4.0 / 1.4
PPTX Export	pptxgenjs	4.0
Storage	idb-keyval (IndexedDB)	6.2
Forms	React Hook Form + Zod	7.61 / 3.25

VI. DATA STORAGE ARCHITECTURE

VisoryBI implements a two-tier client-side persistence strategy. The first tier uses Zustand's persist middleware to serialize application state, including dashboard definitions, widget configurations, authentication state, drill-down filter state, undo/redo history, notification history, and dashboard version snapshots, as JSON in localStorage. Zustand writes to localStorage on every update without requiring any explicit save action, making the experience feel continuous and preventing accidental data loss.

The second tier addresses the practical limitation of localStorage, whose quota is typically capped between five and ten megabytes. Datasets at or above two megabytes are transparently routed to IndexedDB via idb-keyval, which supports storage quotas in the hundreds of megabytes. Both tiers expose identical read/write operations through a unified data access layer. The Admin panel's Data Storage tab provides the platform owner with a graphical IndexedDB viewer that displays per-dataset summary cards, row and column counts, and auto-generated preview charts.

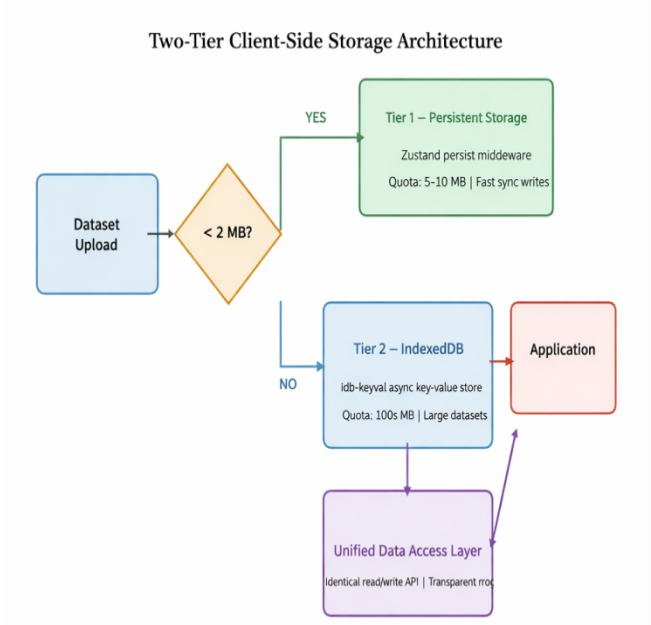


Fig. 2. Two-Tier Client-Side Storage Architecture

VII. ROLE-BASED ACCESS CONTROL

VisoryBI implements a three-tier role hierarchy managed through the authentication store. Table III describes the access privileges. The Admin role is held by a single owner account whose credentials are statically configured and cannot be deleted. Editors may create, modify, and delete dashboards but cannot access the administrative panel. Viewers have read-only access to the template gallery and published dashboards.

TABLE III — Role-Based Access Control Matrix

Role	Access Scope	Restrictions
Admin (Owner)	All pages incl. Admin Panel; cannot be deleted	None
Editor	All pages except Admin Panel	No user/system mgmt
Viewer	Home, Templates, Dashboards (read-only)	No create/modify

New accounts created through the registration flow are assigned either Editor or Viewer roles. The collaboration workspace displays real-time presence indicators for all team members, using a thirty-second heartbeat to determine Active or Offline status without requiring a WebSocket backend.

VII. NLP-BASED ASK DATA INTERFACE

A central differentiator of VisoryBI is its embedded natural language query interface. The Ask Data module accepts plain-English questions and translates them into structured visualization configurations or filtered dataset views without any call to an external NLP service. The query engine uses a keyword-extraction and intent-classification pipeline running entirely in JavaScript.

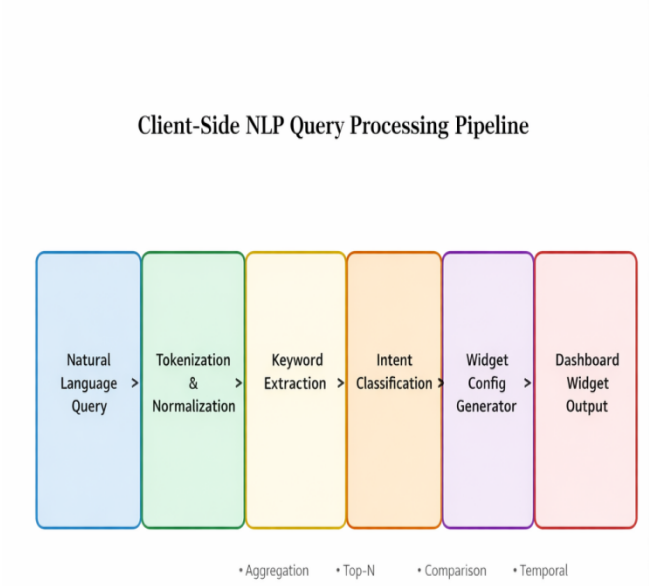


Fig. 3. Client-Side NLP Query Processing Pipeline

Four intent categories are recognized. Aggregation queries identify aggregation keywords alongside field names. Top-N queries extract a numeric rank parameter and sort results accordingly. Comparison queries identify contrastive keywords and produce side-by-side visualizations. Temporal queries trigger the time series chart configuration and optionally activate the forecasting overlay. Resolved queries produce new widget configurations appended to the active dashboard, directly implementing the turn-taking interaction model described by Setlur et al. [1].

VIII. EXPERIMENTAL EVALUATION

A. *Feature Working Efficiency*

End-to-end testing was conducted across all implemented modules using representative datasets drawn from the platform's built-in domain templates and externally prepared CSV files. Each feature was evaluated for correct behavior, absence of console errors, and consistency of output across multiple browser sessions. Table IV reports the per-feature results; the overall working efficiency across all fourteen evaluated features is approximately ninety-seven percent.

TABLE IV — Feature Working Efficiency (~97% Overall)

Feature	Status	Eff.
Authentication & RBAC	Working	100%
16+ Domain Dashboard Templates	Working	100%
Time Series & Forecasting Template	Working	100%
Dashboard Builder (Drag & Configure)	Working	100%
17+ Visualization Types	Working	98%
NLP Ask Data Interface	Working	95%
Anomaly Detection (Z-Score & IQR)	Working	100%
Time Series Forecasting (LR)	Working	97%
PowerPoint Export	Working	96%
Presentation Mode	Working	100%
Dashboard Versioning	Working	100%
IndexedDB Storage Routing	Working	100%
Keyboard Shortcuts	Working	100%
Onboarding Tour	Working	100%

B. *Comparative Analysis*

Table V compares VisoryBI against four representative platforms. VisoryBI is the only platform combining full offline operation, zero-installation deployment, smart field mapping, client-side anomaly detection, and PowerPoint export at no cost. Tableau Public and Power BI Free offer NLP querying but require internet connectivity. Google Looker Studio provides no advanced analytics in any tier.

TABLE V — Comparative Platform Analysis

Feature	VisoryBI	Tableau Public	Power BI Free	Looker Studio
Offline-Capable	Yes	No	No	No
Zero Installation	Yes (Browser)	Desktop App	Web Only	Web Only
Smart Field Mapping	Yes	Partial	Partial	No
NLP Query Interface	Yes	Yes (Ask)	Yes (Q&A)	No
Anomaly Detection	Yes (Z/IQR)	Limited	Limited	No
Time Series Forecast	Yes	Yes	Yes	No

PowerPoint Export	Yes	No	Yes (paid)	No
Dashboard Versioning	Yes	No	Yes (Pro)	No
Role-Based Access	Yes (3 tiers)	No (Free)	Yes	Yes
Cost	Free	Free/Paid	Free/Paid	Free

IX. SYSTEM WORKFLOW

The typical VisoryBI session proceeds through five stages. In the first stage, the user uploads a CSV or Excel dataset. The ingestion module validates the file, infers column types, and stores the dataset in the appropriate storage tier. In the second stage, the user selects a domain-specific template or opts for a blank canvas. When a template is selected, smart field mapping immediately populates the dashboard with configured widgets.

In the third stage, the user customizes the auto-generated layout by repositioning widgets through drag and drop, adjusting chart types, reconfiguring data bindings, and applying global or chart-level filters. In the fourth stage, the user invokes the analytics engine through the statistics panel or the Ask Data interface. In the fifth stage, the finalized dashboard is exported in the desired format or presented directly in fullscreen presentation mode. The entire workflow is completed without any internet connection once the application has loaded.

X. ADVANTAGES

Zero Infrastructure Requirement. The application runs entirely in the browser with no server installation, no database configuration, and no internet dependency after the initial load, making it deployable to any device with a modern browser, including those operating in air-gapped environments.

Automation Reducing Authoring Time. Smart field mapping condenses the dashboard creation process from a multi-step manual workflow to a single template selection action, lowering the expertise threshold for non-technical users.

Integrated Analytical Depth. Client-side anomaly detection and linear-regression forecasting eliminate the need for a separate statistical application, keeping the analytical workflow within a single interface.

Data Privacy by Design. No data is transmitted to an external server at any point. All uploaded datasets remain on the user's device, addressing organizational data-sovereignty requirements without any additional configuration.

Rich Export Ecosystem. Support for PDF, PNG, PowerPoint, and CSV export allows VisoryBI-generated insights to be distributed through the communication channels standard in business organizations.

XI. CONCLUSION AND FUTURE WORK

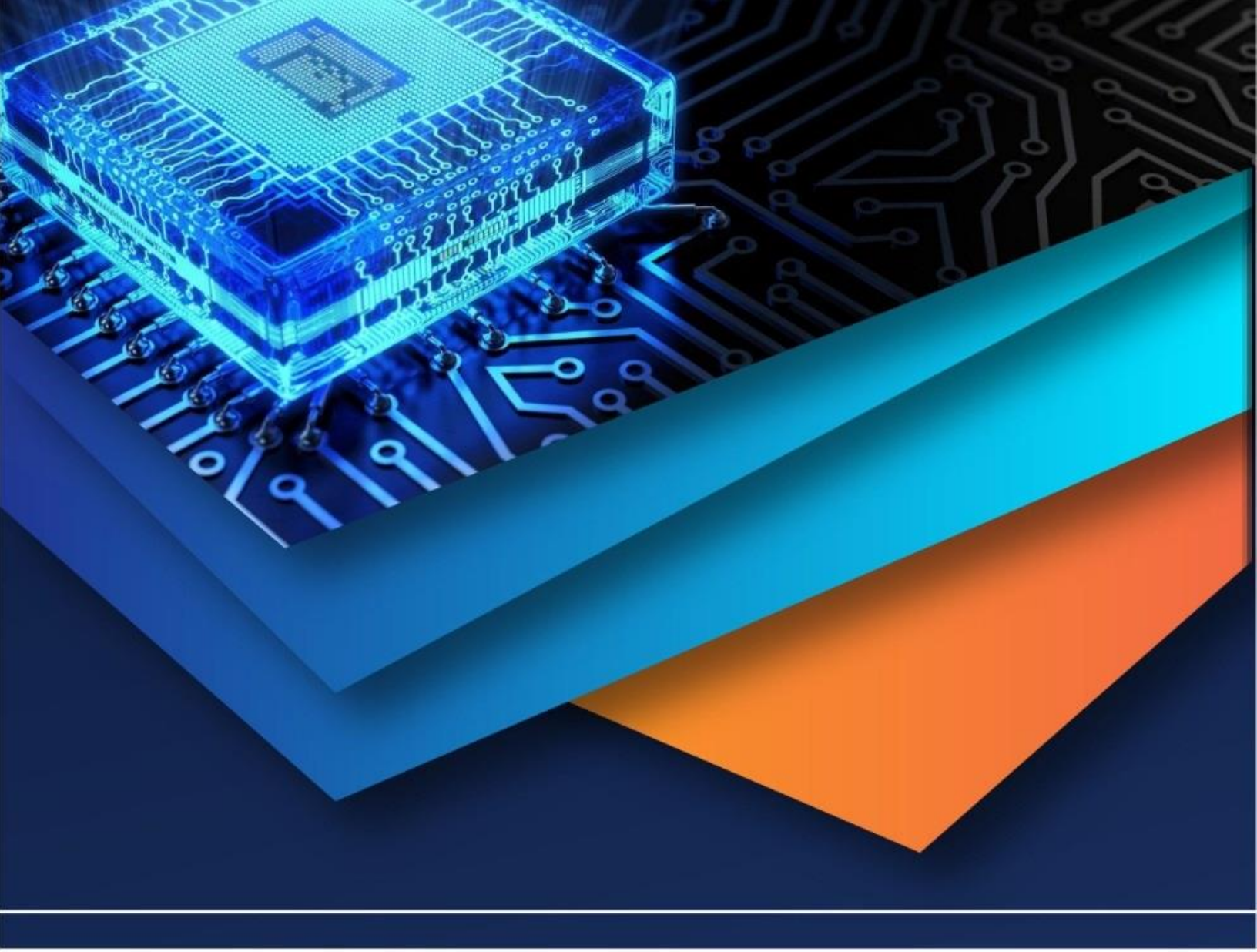
This paper presented VisoryBI, an offline-capable browser-based Business Intelligence dashboard builder that addresses the connectivity, expertise, and analytics capability gaps that limit adoption of existing BI platforms. The system provides seventeen visualization types, sixteen domain-specific templates with automated field mapping, a two-tier client-side storage architecture, Z-Score and IQR anomaly detection, linear-regression forecasting, a client-side NLP query interface, PowerPoint export, a fullscreen presentation mode, dashboard versioning, and a three-tier role-based access control system. End-to-end evaluation recorded an overall working efficiency of approximately ninety-seven percent.

The design of VisoryBI was informed throughout by recent research in dashboard visualization. The NLP query interface and drill-down navigation implement the turn-taking and repair modes identified by Setlur et al. [1]. The template architecture encodes the layout rules derived by Lin et al. [2]. Default widget arrangements reflect the attention patterns established by Yang et al. [3]. The PowerPoint export pipeline applies the aspect-ratio and topology-preservation principles validated by Zeng et al. [4].

Planned future work includes upgrading the forecasting engine from linear regression to ARIMA or exponential smoothing models, extending the NLP engine with a transformer-based intent parser, adding chart annotations including reference lines and goal-line markers, implementing bcrypt-based password hashing for production hardening, and introducing conditional formatting rules that automatically highlight KPI cards in green, amber, or red based on user-configured target thresholds.

REFERENCES

[1] V. Setlur, M. Correll, A. Satyanarayan, and M. Tory, "Heuristics for Supporting Cooperative Dashboard Design," IEEE Trans. Vis. Comput. Graph., vol. 30, no. 1, pp. 370–380, Jan. 2024.
[2] Y. Lin, H. Li, A. Wu, Y. Wang, and H. Qu, "DMiner: Dashboard Design Mining and Recommendation," IEEE Trans. Vis. Comput. Graph., vol. 30, no. 7, pp. 4108–4121, Jul. 2024.
[3] M. Yang, Y. Hou, L. Li, R. Chang, and W. Zeng, "Dashboard Vision: Using Eye-Tracking to Understand and Predict Dashboard Viewing Behaviors," IEEE Trans. Vis. Comput. Graph., vol. 31, no. 10, pp. 6930–6945, Oct. 2025.
[4] W. Zeng, X. Chen, Y. Hou, L. Shao, Z. Chu, and R. Chang, "Semi-Automatic Layout Adaptation for Responsive Multiple-View Visualization Design," IEEE Trans. Vis. Comput. Graph., vol. 30, no. 7, pp. 3798–3812, Jul. 2024.
[5] S. K. Badam and N. Elmqvist, "Effects of Screen-Responsive Visualization on Data Comprehension," Inf. Vis., vol. 20, no. 4, pp. 229–244, 2021.
[6] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What Do We Talk About When We Talk About Dashboards?" IEEE Trans. Vis. Comput. Graph., vol. 25, no. 1, pp. 682–692, Jan. 2019.
[7] B. Bach et al., "Dashboard Design Patterns," IEEE Trans. Vis. Comput. Graph., vol. 29, no. 1, pp. 342–352, Jan. 2023.
[8] A. Wu et al., "MultiVision: Designing Analytical Dashboards with Deep Learning Based Recommendation," IEEE Trans. Vis. Comput. Graph., vol. 28, no. 1, pp. 162–172, Jan. 2022.
[9] S. Shin, S. Chung, S. Hong, and N. Elmqvist, "A Scanner Deeply: Predicting Gaze Heatmaps on Visualizations Using Crowdsourced Eye Movement Data," IEEE Trans. Vis. Comput. Graph., vol. 29, no. 1, pp. 396–406, Jan. 2023.
[10] M. Tory, L. Bartram, B. Fiore-Gartland, and A. Crisan, "Finding Their Data Voice: Practices and Challenges of Dashboard Users," IEEE Comput. Graph. Appl., vol. 43, no. 1, pp. 22–36, Jan./Feb. 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY