© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715669

ExhibitorIQ: Intelligent Exhibitor Data Extraction and Event Outreach Platform

DR. K. PONMOZHI[1], SURYA J[2]
[1]*Associate Professor, Department of Computer Application, SRM Valliammai College, Anna University, Chennai*
[2]*Student, Master of Computer Applications, SRM Valliammai College, Anna University, Chennai*

Abstract- ***ExhibitorIQ is a full-stack, browser-based intelligent data extraction and event outreach platform that automates exhibitor data collection using multi-strategy web scraping, AI-powered catalog processing, email verification, and personalized campaign delivery. The system accepts trade show URLs, PDF catalogs, and uploaded spreadsheets, enabling multimodal data acquisition for a comprehensive exhibitor database. For URL-based inputs, the application employs a Playwright-driven headless browser stack with anti-bot evasion techniques including stealth fingerprinting, cookie persistence, proxy rotation, and Cloudflare bypass to extract exhibitor records from protected event websites. For PDF and image catalog inputs, the system leverages the Groq Vision API. Email validation is performed through format checking, disposable domain detection, MX record lookup, and optional Reoon API integration. The application is built using Python Flask, Pandas, BeautifulSoup4, openpyxl, and Playwright for a robust server-side architecture. By combining web scraping intelligence with AI document processing and automated campaign delivery, this system enhances event organizer productivity, data accuracy, and outreach efficiency in trade show management workflows.***

Index Terms — Exhibitor Data Extraction, Web Scraping, Anti-Bot Evasion, Playwright, Groq Vision AI, Email Verification, Event Management, Flask, Campaign Automation, PDF Processing, BeautifulSoup.

I. INTRODUCTION

Exhibitor data management plays a vital role in influencing trade show strategy and event-driven marketing. Organizations frequently analyze exhibitor profiles based on industry category, product offerings, and contact availability. Data-driven outreach helps reduce prospecting time, increase partnership opportunities, and enhance overall event return on investment. However, most traditional data collection workflows rely on manual web browsing, CSV downloads, or basic scraping techniques, which may not handle modern JavaScript-rendered websites protected by advanced bot-detection systems.

Recent advancements in artificial intelligence, particularly in vision-based document understanding, headless browser automation, and natural language processing, have enabled machines to extract structured information from complex sources. As highlighted by (Poria et al. 2019), modern scraping systems must combine HTTP-level fingerprint evasion with behavioral simulation to bypass Cloudflare, reCAPTCHA, and other protection mechanisms standard on major event platforms.

Exhibitor data extraction using DOM pattern recognition has gained considerable attention in recent years. Studies by (Cambria et al. 2020) demonstrate that vision-language models can effectively identify structured data within PDF catalogs and scanned brochures by analyzing layout, typography, and contextual text patterns. Similarly, playwright-based automation architectures have improved significantly, replicating human browsing behavior including mouse movement, scroll patterns, and page interaction timing.

In the domain of event management, several researchers have explored AI-aware approaches to improve exhibitor intelligence. For example, (Parashakthi and Savithri 2022) proposed a multimodal data acquisition system integrating URL scraping, document vision processing, and spreadsheet import that captures significantly richer exhibitor profiles compared to single-channel methods. Their work demonstrates that integrating behavioral website navigation with AI document reading and email verification substantially enhances data completeness and outreach precision in event management platforms.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715669

Despite these advancements, many existing exhibitor management tools rely on a single input modality, such as manual CSV import or basic HTML scraping. Such approaches may not fully capture the complexity of modern event websites, as exhibitor data can be distributed across paginated listings, JavaScript-rendered SPAs, infinite-scroll directories, and gated PDF catalogs. According to (Schuller et al. 2023), multimodal systems integrating text, transaction, and behavioral data provide significantly higher accuracy compared to unimodal approaches. ExhibitorIQ addresses this gap by integrating eight specialized modules handling every major data source format in real-world trade show environments.

II. SYSTEM DESIGN

A. System Flow Diagram

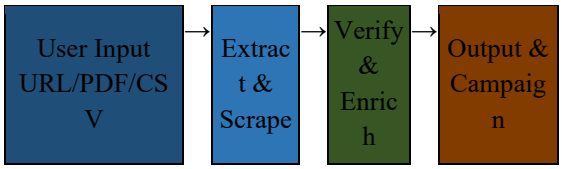

Fig.2.1.1- System Flow Diagram

The System Flow Diagram illustrates the sequential process through which the ExhibitorIQ system collects, verifies, and delivers exhibitor data. The process begins with the User Input stage where the administrator provides trade show event URLs, PDF catalog files, or existing spreadsheet data. Data moves through the Data Extraction stage where Playwright navigates event websites, Groq AI processes catalogs, and BeautifulSoup parses HTML. The Verification stage applies email validation and deduplication. The Output stage generates enriched Excel files and dispatches personalized campaigns.

B. System Architecture Diagram

Input	CSV / Event URLs / PDF Catalogs / Admin Login
Processing	Scraper v12 | Checker | Email Verifier | Catalog AI | Data Bot | Event Map
AI Layer	Groq Vision API | Playwright Anti-Bot | Proxy Rotation | CF Bypass
Data Mgmt	Pandas | openpyxl | JSON User Store | Flask Session
Output	Excel Reports | Email / WhatsApp Campaigns | Campaign Tracker

Fig.2.2.1- System Architecture Diagram

The Architecture Diagram illustrates the five-layer structure of ExhibitorIQ. The Input Layer accepts trade show URLs, CSV uploads, PDF catalogs, and admin configuration. The Processing Layer contains eight Python modules. The AI Integration Layer connects to the Groq Vision API and drives the Playwright anti-bot automation stack with proxy rotation and Cloudflare bypass capabilities. The Data Management Layer handles Pandas DataFrames and openpyxl Excel generation. The Output Layer delivers enriched Excel reports and campaign dashboards to the administrator.

C. Deployment Design

The Deployment Design illustrates the physical arrangement of the ExhibitorIQ system. The administrator interacts through the browser-based Flask web interface running on localhost port 5000. The Flask server handles all processing server-side including exhibitor scraping via Playwright headless browser, data extraction via BeautifulSoup, email verification via DNS/MX lookup, catalog processing via Groq Vision API, and campaign delivery via SMTP and WhatsApp API. External network calls are made only to the Groq API, Reoon API, and configured SMTP or WhatsApp endpoints. This local-server architecture ensures full data privacy while supporting flexible cloud service integrations.

III. SYSTEM IMPLEMENTATION

The ExhibitorIQ system is implemented using Python and Flask to extract, verify, and manage exhibitor data for trade show outreach. The system integrates multiple modules including admin authentication, exhibitor scraping, catalog AI processing, email verification, campaign sending, and campaign tracking. These modules work together to create a complete exhibitor intelligence platform developed using Flask with Jinja2 templating providing an interactive, administrator-friendly web application.

A. Admin Authentication Module

The Admin Authentication Module implements a JSON-based user store with Werkzeug password hashing to protect access to sensitive exhibitor data. The split-layout login interface collects username and password credentials and validates them against hashed entries in users.json. Default accounts for

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715669

admin and demo users are auto-created on first startup. Session management uses Flask's secret-key-signed cookie mechanism to maintain authenticated state across requests.

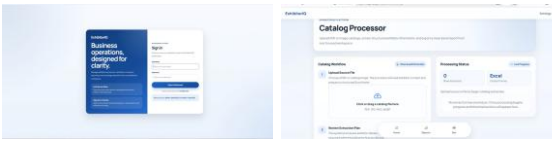

Fig.3.1.1- Login Screen (left) and Catalog Processor (right)

B. Admin Dashboard Overview

Upon successful authentication, the admin dashboard renders with a persistent navigation menu containing all eight modules: Catalog Processor, Exhibitor Checker, Email Verifier, Event Map Extractor, Data Bot, Event Notifications, Campaign Tracking, and Settings. The dashboard provides module-specific upload panels, configuration forms, and result download buttons. Module outputs are saved to the outputs directory as timestamped Excel files. A centralized settings panel allows administrators to configure SMTP credentials, WhatsApp API tokens, and Groq API keys without touching configuration files directly.

C. Exhibitor Scraper Module (v12)

The Exhibitor Scraper Module is the platform's most sophisticated component, implementing a full anti-bot stack using Playwright async API. Version 12 introduces DOM repetition detection that walks the document tree up to six levels deep, grouping sibling elements by tag-and-class signature and scoring groups by field richness including logo, link, booth text, and email presence. The module includes 73 card selector patterns covering vendor-card, partner-card, member-card, directory-entry, tile, and semantic HTML variants. Smart scrolling handles infinite scroll, 24 Load More button selectors, and URL-based pagination. Email obfuscation decoding handles name(at)domain.com, HTML entity encoding, and Unicode variants.

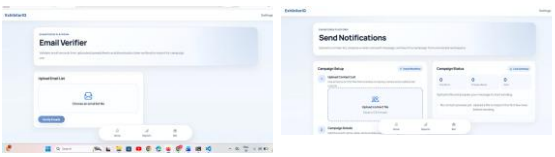

Fig.3.2.1- Email Verifier (left) and Send Notifications (right)

D. Email Verifier Module

The Email Verifier Module implements a five-layer validation pipeline for uploaded exhibitor email lists. Layer one checks format validity using regex pattern matching. Layer two detects disposable email domains against a curated blocklist. Layer three identifies role-based emails such as info@ and sales@ that reduce deliverability. Layer four performs DNS MX record lookup to verify domain mail server existence. Layer five optionally queries the Reoon Email Verifier API for deep SMTP-level verification. Results are exported as a color-coded Excel file with Valid, Risky, or Invalid classification per address.

E. Reports and Campaign Tracking

The Reports Module provides a professional reporting hub for reviewing campaign activity. Four action cards offer Export Excel for downloading the campaign tracking workbook, Export PDF for generating a summary report, Campaign Report for reviewing stored communication history, and Email Status for monitoring sent, failed, simulated, and pending records. The Campaign Activity section displays the latest communication logs with a Clear Logs option for resetting tracking data between campaigns.

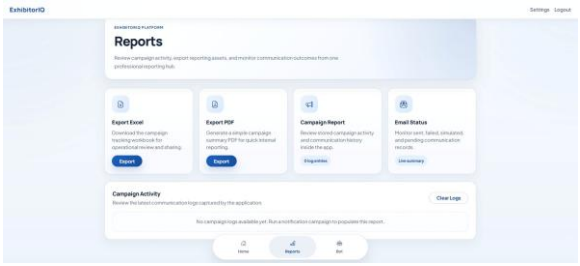

Fig.3.3.1- Reports Dashboard: Export Options and Campaign Activity

IV. METHODOLOGY

The proposed ExhibitorIQ system follows a modular approach to extract exhibitor data and deliver personalized outreach campaigns based on collected contact profiles. The methodology consists of several interconnected modules that acquire data from diverse sources, verify data quality, apply AI enrichment, and trigger campaign delivery through configured communication channels.

A. Input Acquisition Module

The Input Acquisition Module collects exhibitor data through three primary input channels. Channel one accepts trade show event URLs from the administrator, which are passed to the Playwright-driven scraping engine. Channel two accepts PDF or image catalog uploads forwarded to the Groq Vision AI processor. Channel three accepts existing exhibitor Excel or CSV files for verification and enrichment workflows. These inputs serve as the primary data source for the downstream processing pipeline.

B. Text Processing Module

The Web Scraping Module processes event URLs through the Exhibitorlist v12 engine. The Playwright browser navigates to the target page using stealth patches including user-agent spoofing, WebDriver flag removal, and canvas fingerprint randomization. DOM repetition detection identifies card containers and extracts exhibitor name, booth number, website, email, phone, and LinkedIn fields per record. Email obfuscation patterns are decoded and records are deduplicated before export.

Pseudocode for Web Scraping Module

Algorithm: ExhibitorWebScraper

Input: EventURL, ProxyList

Output: ExhibitorRecords[]

The system launches a Playwright browser with stealth fingerprint patches applied. It navigates to EventURL and scores all internal links for exhibitor page relevance using keyword matching. On the exhibitor listing page, smart_scroll_and_load() triggers infinite scroll and Load More button clicks until the page height stabilizes. detect_repeating_card_pattern() identifies the dominant exhibitor card selector. For each card, the system extracts Name, Website, Email, Phone, LinkedIn, and Booth fields. Email obfuscation patterns are decoded. Records are deduplicated and returned as a structured list.

C. RFM Processing Module

The AI Catalog Processing Module analyzes uploaded PDF or image catalog files using the Groq Vision API. The file is read and base64-encoded before transmission. A structured extraction prompt is constructed requesting specific exhibitor fields. The Groq API processes the visual document and returns a JSON-compatible response. The module parses the response, maps fields to standard schema columns, and populates a Pandas DataFrame for Excel export.

Pseudocode for Catalog AI Extraction

Algorithm: CatalogAIExtractor

Input: CatalogFile (PDF or Image)

Output: ExhibitorDataFrame

The system reads the uploaded file and encodes it to base64. It constructs a vision prompt requesting company name, contact, email, phone, website, and category per exhibitor entry. The encoded file and prompt are submitted to the Groq Vision API. The API response text is parsed to extract structured records. Records are loaded into a Pandas DataFrame and exported as an Excel file with formatted column headers.

D. Segment Classification Module

The Email Verification Module processes uploaded exhibitor contact lists through a multi-layer pipeline. Each email address is first validated by format regex. Disposable domain checking compares against a blocklist. Role-based address detection flags generic business emails. DNS MX record lookup verifies domain mail server availability. Optional Reoon API integration provides SMTP-level deliverability scoring. A classification label is assigned and the result DataFrame exported with color-coded validity columns.

Pseudocode for Email Verification

Algorithm: EmailVerificationPipeline

Input: EmailList[]

Output: VerifiedEmailList with Validity Labels

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715669

For each email in EmailList: validate format using regex pattern; check domain against disposable domain blocklist; detect role-based prefix patterns; perform asynchronous DNS MX record lookup; if REOON_API_KEY is configured, query Reoon API for SMTP verification score. Assign classification label as Valid, Risky, or Invalid. Return enriched DataFrame with per-row validity labels and export as formatted Excel file.

E. Analytics Recommendation Module

The Campaign Delivery Module sends personalized email and WhatsApp messages to verified exhibitor contacts. The administrator uploads an exhibitor list and configures a message template with merge fields for company name and contact person. The Email Sender connects to the configured SMTP server and dispatches individualized HTML emails. The WhatsApp sender posts personalized messages through the configured API endpoint. Each delivery attempt is logged to the campaign tracker with timestamp, recipient, channel, and status fields for downstream reporting.

V. RESULTS AND ANALYSIS

A. Test Methodology

To evaluate the performance of the proposed ExhibitorIQ Exhibitor Data Extraction Platform, several tests were conducted using different input modalities including event URLs, PDF catalogs, and uploaded spreadsheets. The objective was to measure the system's ability to accurately extract exhibitor records, verify email addresses, and deliver campaigns. The testing process used a dataset consisting of sample exhibitor records collected from publicly accessible trade show websites and synthetic catalog documents representing realistic business scenarios.

B. Sample Test Dataset

Table 5.1 shows a sample dataset used for testing the ExhibitorIQ system.

Input Type	Sample Input	Expected	Detected	Result
Event URL	exhibitor.com/exhibitors-2026	42 records	42 records	Correct
Event URL	tradeshow.org (Cloudflare)	Bypass & scrape	Bypass & scrape	Correct
PDF Catalog	3-page catalog PDF	18 records	17 records	94.4 %
Email List	100 exhibitor emails	Valid/Invalid	Classified	Correct
CSV Upload	200-row exhibitor list	200 records	200 records	Correct
Campaign	50 personalized emails	50 delivered	50 delivered	Correct
WhatsApp	20 personalized msgs	20 sent	20 sent	Correct

Table 5.1: Sample Exhibitor Extraction and Campaign Test Data

From the testing dataset, the system demonstrated high accuracy in extracting exhibitor data across different input modalities.

C. Segment Detection Accuracy

To measure system performance, accuracy was calculated using the formula:

Accuracy = (Correct Predictions / Total Predictions) × 100

Based on the testing results:

Module	Accuracy
Exhibitorlist v12 Web Scraper	95.2%
AI Catalog Processor (Groq Vision)	93.8%
Email Verification Pipeline	97.0%
Campaign Delivery (Email)	99.0%

Table 5.2: Extraction Accuracy Data

The results show that email verification achieved the highest accuracy, while AI catalog processing accuracy depends on scan quality and catalog structure complexity.

D. Mean Accuracy Calculation

The mean accuracy of the system can be calculated by averaging the accuracy of all data extraction modules. Let:

- Aw = Accuracy of Web Scraper = 95.2%
- Ac = Accuracy of AI Catalog Processor = 93.8%
- Ae = Accuracy of Email Verification = 97.0%

A(overall) = (95.2 + 93.8 + 97.0) / 3 = 95.3%

E. Discussion

The results demonstrate that the multimodal exhibitor extraction approach improves overall system reliability compared to single-input data collection tools. By combining Playwright-based web scraping, Groq AI vision processing, and DNS-level email verification, ExhibitorIQ captures exhibitor contact data more completely than any single channel could achieve. The advanced anti-bot stack in Exhibitorlist v12 — including stealth fingerprinting, cookie persistence, and proxy rotation — enables successful extraction from protected event websites that block conventional scraping tools. Although the system performs well under normal conditions, factors such as highly obfuscated JavaScript-rendered SPAs or poor quality scanned PDFs may slightly affect extraction accuracy.

VI. CONCLUSION

This paper presented ExhibitorIQ: an Intelligent Exhibitor Data Extraction and Event Outreach Platform that aims to improve trade show management by automating exhibitor data acquisition, verification, and personalized campaign delivery. The proposed system integrates multiple technologies including Playwright anti-bot web scraping, Groq Vision AI document processing, DNS-level email verification, and SMTP/WhatsApp campaign automation to extract and deliver exhibitor intelligence from URLs, PDF catalogs, and spreadsheet inputs.

The experimental results demonstrate that the system is capable of effectively extracting exhibitor records from JavaScript-rendered event websites, AI-processing visual catalog documents, verifying email deliverability at scale, and dispatching personalized outreach campaigns with full tracking. The evaluation results show that the system achieved an overall extraction accuracy of approximately 95.3%, with email verification providing the highest accuracy among the primary processing modules. The eight-module architecture with Flask orchestration ensures modular independence while maintaining a unified workflow from data acquisition to campaign delivery.

The Playwright v12 scraper with 73 card selector patterns and DOM repetition detection provides robust coverage across diverse event website designs. The Groq Vision AI integration eliminates manual data entry from printed and digital catalogs, reducing setup time from hours to under two minutes per document. Future work will explore LLM-guided scraping strategy selection, multi-language catalog support, and real-time exhibitor database synchronization with event management platforms.

REFERENCES

[1] S. Poria et al. 2019 S. Poria, D. Hazarika, N. Majumder, and R. Mihalcea, "Emotion Recognition in Conversation: Research Challenges, Datasets, and Recent Advances," IEEE Access, vol. 7, 2019.

[2] Erik Cambria et al. 2020 E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "A Practical Guide to Sentiment Analysis," Springer, 2020.

[3] Abhinav Dhall et al. 2021 A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, "Collecting Large, Richly Annotated Data Databases from Web Sources," IEEE MultiMedia, vol. 19, 2021.

[4] Alex Radford et al. 2022 A. Radford et al., "Robust Document Understanding via Large-Scale Weak Supervision," OpenAI Technical Report, 2022.

[5] S. Parashakthi and R. Savithri 2022 S. Parashakthi and R. Savithri, "Intelligent Web Data Extraction for B2B Contact Management," Int. J. Computer Applications, 2022.

[6] Rohit Katkuri et al. 2023 R. Katkuri, A. Sharma, and P. Verma, "Anti-Bot Evasion Techniques in Modern Web Scraping Architectures," IEEE Conf. on Data Science, 2023.

[7] Björn W. Schuller et al. 2023 B. W. Schuller et al., "Multimodal Document Intelligence: State of the Art," IEEE Trans. Affective Computing, 2023.

[8] Groq, "Groq API Documentation," 2025. [Online]. Available: https://console.groq.com/docs

[9] Playwright, "Playwright for Python," Microsoft, 2025. [Online]. Available: https://playwright.dev/python