Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20250403026
Volume 4, Issue3 (September-December 2025), PP: 160-166.
www.indjcst.com
ISSN No: 2583-5300

Digital Platform Recommendation System with Scalable model using Clustering

Dakshayani R N R[1], Tejal N R[2]
[1]*Department of Computer Science and Engineering, Anna University, Madurai, Tamilnadu, India.*
[2]*Department of Remote Sensing and GIS, Bharathidasan University, Trichy, Tamilnadu, India.*

To Cite this Article: *Dakshayani R N R[1], Tejal N R[2], "Digital Platform Recommendation System with Scalable model using Clustering", Indian Journal of Computer Science and Technology, Volume 04, Issue 03 (September-December 2025), PP: 160-166.*

Abstract: *A Python-based digital platform recommendation dashboard for multi-criteria evaluation of major websites is presented. The system computes privacy, design, collaboration, popularity, and innovation scores from various web metrics and applies K-Means clustering to group sites by their characteristics. This model employs a correlation-based multi-domain scoring system encompassing the above mentioned scores. An interactive Streamlit GUI enables users to select focus areas (e.g. design or privacy) and view recommended top-ranked websites accordingly. The dashboard uses feature scaling and the Elbow method to determine the optimal number of clusters and it also provides visualizations of cluster distributions and recommendations showing transparency and interpretability. This explainable approach contrasts with conventional ranking only by popularity offering transparent multi-dimensional insights into website quality and suggesting improvements. The results (Elbow plot, cluster scatter) demonstrate meaningful groupings (e.g. high-design vs. high-popularity sites) and validates the recommender logic. The system's implementation leverages Pandas, scikit-learn, Matplotlib/Seaborn, and Streamlit, and its evaluation highlights both practical utility and areas for future enhancement.*

Key Words: *Recommendation System, Streamlit GUI, K-Means Clustering, Correlation Coefficient, Website/App Evaluation, Feature Scoring, Design and Popularity Metrics, Data-driven Ranking, Digital platform recommendation.*

I.INTRODUCTION

The technological advancements and rapid growth of online reliability has created a demand for systematic evaluation of digital platforms. High quality websites not only attract users through popularity metrics but they also must excel in privacy protection, user interface design, and innovation as required by the user. Conventional search rankings rely widely on keyword relevance and traffic data that may overlook usability and data protection aspects. In contrast, recommender systems and clustering provide data-driven, multi-dimensional analysis of websites. Recommender systems are widely used to filter and personalize large-scale information that reduces user effort in finding relevant content[1, 2] . Similarly, clustering algorithms (such as K-Means) can group websites into interpretable clusters based on feature similarity that reveals underlying patterns[3]. By combining clustering with recommendations, a correlation-based data- driven recommendation framework was developed. This system evaluates web platforms based on multiple criteria including privacy protection, user interface quality, collaboration tools, and innovative features. The developed dashboard provides explainable suggestions: users see both the grouping of sites by quality and tailored recommendations based on selected criteria. This approach addresses the "data overload" problem noted in the literature[1] and aims to improve transparency in platform evaluation by leveraging established machine learning techniques[5].

II.METHODOLOGY

The process includes collection of data, here focused on major websites for the evaluation process which proposes a scalable model by further adding the websites as new digital platform emerges in the technological advancements.

Dataset: A dataset of 50 major global websites, each described by over 40 attributes covering five domains (Privacy, Design, Collaboration, Popularity, Innovation) were used after preprocessing. Examples of fields include design_score, privacy_score, usability_rating, monthly_visitors, and ai_features_present. These domain scores were computed via normalized weighted sums of relevant measures (e.g. HTTPS enabled, UI consistency, traffic statistics). The raw data was managed with the Pandas library[4] to facilitate cleaning and analysis. Each domain score was calculated through the normalized weighted averages of measurable indicators collected from reliable web sources that include SSL Labs[10], Mozilla Observatory[11], Google PageSpeed[12], SimilarWeb[8], Statista[9] and WebAIM[13]. Attributes were scaled using Min- Max normalization to ensure consistency[5].

Key attributes and their role in domain derivations are given:

- **Privacy Score:** Derived from HTTPS enablement, TLS certificate validity, GDPR policy mention, and cookie consent presence. This helps to evaluate user data protection.

- **Design Score:** Calculated using UI consistency, page load speed, accessibility, and color contrast scores from Page Speed and Web AIM.
- **Collaboration Score:** It is obtained based on multi user support, real-time sync, integrations, and cloud storage availability.
- **Popularity Score:** The evaluation on web reach using monthly visitors, backlinks, and search trend scores from Similar Web.
- **Innovation Score:** This captures presence of AI features, frequency of updates, and open API availability.

Other features are used to derive the above five scores to facilitate recommendation as tabulated in Table no 1

Table no 1: Data reference table for dataset attributes

Feature / Attribute	Primary Data Source	Supporting / Cross-check Source	Data Description / Usage
site_name, **url**, **category**	Wikipedia, Official Websites	Google Directory	Identifies and classifies the 50 major websites and applications.
https_enabled, **tls_validity_days**	SSL Labs (Qualys)	Whois Lookup	Checks for HTTPS configuration, SSL/TLS validity, and certificate security.
privacy_policy_found, **privacy_policy_length**, **mentions_gdpr**, **data_sharing_disclosure**	PrivacyPolicies.com, TermsFeed	GDPR.eu	Used to analyze privacy transparency and compliance.
external_tracker_count, **cookie_consent_banner**	WhoTracks.me, DuckDuckGo Tracker Radar	Ghostery Tracker Database	Measures number of third-party trackers and cookie management compliance.
mobile_responsive, **ui_consistency_score**, **color_contrast_score**, **accessibility_score**	Google Lighthouse	Web.dev Measure, WAVE Accessibility Tool	Evaluates design consistency, accessibility, and visual experience.
avg_page_load_time_s	GTMetrix	Pingdom Tools	Measures website loading speed for performance and UX.
usability_rating	G2, Capterra	ProductHunt	Assesses ease of use and overall user experience from reviews.
multi_user_support, **real_time_sync**, **version_control**, **cloud_storage_support**	Official product documentation (Google Workspace, Slack, Notion)	G2, Capterra	Verifies if platforms support collaboration and data synchronization features.
integration_count	Zapier App Directory	IFTTT	Counts number of third-party integrations supported.
monthly_visitors_million, **domain_age_years**	SimilarWeb, Whois Lookup	Statista	Measures traffic volume and longevity of the website.
backlinks_count_k	Ahrefs	Moz, Semrush	Quantifies backlink strength for online reach and authority.
social_followers_million	SocialBlade	Verified platform handles (Twitter, YouTube, LinkedIn)	Aggregates social following across major networks.
search_trend_score	Google Trends	Exploding Topics	Measures brand search popularity over time.
ai_features_present, **unique_feature_count**, **update_frequency_per_year**	Crunchbase, TechCrunch	Official company blogs (Google AI, Microsoft, OpenAI)	Tracks AI adoption, innovation frequency, and feature updates.

open_api_available	RapidAPI Directory	Official Developer Portals (e.g., Twitter, Slack, GitHub APIs)	Confirms whether platforms provide open APIs for developers.
privacy_score, **design_score**, **collaboration_score**, **popularity_score**, **innovation_score**	Derived via weighted averages using raw metrics	Internal computation	Composite scores used for clustering, correlation, and recommendation.

Feature Scaling and Clustering: Prior to clustering, we scaled numeric features (e.g. scores and counts) to ensure comparability (using min-max normalization or standardization as appropriate). We applied the K-Means algorithm using the features as given in Table no 2 (from scikit-learn[5]) to segment the websites into groups. To select the optimal cluster count *K*, we used the Elbow method: plotting the within-cluster sum of squares (WCSS) against *K* and identifying the "elbow" point where marginal gain drops[3]. For instance, an elbow at *K*=3 suggested three clusters balanced by design quality vs. popularity. This step follows standard clustering practice[3,10] and helps avoid arbitrary cluster choices.

Table no 2: Features used for K-means clustering

Feature Used for Clustering	Description / Purpose
privacy_score	Represents how secure and privacy-conscious the platform is.
design_score	Measures visual quality, responsiveness, and usability.
collaboration_score	Indicates teamwork, sharing, and integration efficiency.
popularity_score	Shows platform reach, user engagement, and global presence.
innovation_score	Reflects AI features, updates, and unique capabilities.

Recommendation Logic: A recommendation model is developed using clustering, in which the user selects a focus (e.g. "Design" or "Privacy"), then the system computes composite scores for each site in that domain and ranks the sites accordingly. In our prototype, we simply list the top-scoring websites for the chosen criterion. This correlation-weighted ranking is transparent and allows inspection of why each site ranks highly. Because clustering was already performed, the user can also see a site's cluster and compare it to peers. For example, users interested in design-rich sites can view the cluster of high-design/low-popularity platforms (e.g. Notion) versus the cluster of high-popularity balanced sites (e.g. Google).

Design vs Popularity Rationale:

Design and Popularity were chosen as the two key indicators for clustering because they represent orthogonal dimensions of web performance. Design reflects user experience (UI/UX) as defined by usability engineering principles[18], while Popularity represents market visibility[16] and brand reach. According to Google's Core Web Vitals[17], design quality is a key indicator of user satisfaction, while popularity drives engagement and market reach. These two combinations provide a balanced evaluation of visually appealing and social adoptability of a website.

System Implementation:

The proposed system was implemented in Python using the Streamlit library. The GUI allows users to choose specific focus areas such as privacy, design, collaboration, innovation, popularity and combination of the above and view top performing websites/apps based on users' requirements. K-Means clustering was included to group websites/apps based on multi-dimensional features. The Elbow Method is used to identify the optimal number of clusters. Visualization of clusters on a Design vs. Popularity plane allows using scatter plot allows users to understand how websites group by user experience and market reach. This transparency enhances making the recommendation process visible and interpretable.

The system comprises two major modules:

1. **Recommendation Module:** It is used to compute correlation coefficient scores and lists top performing sites for selected focus area.
2. **Clustering Module:** This applies K-Means algorithm and displays cluster distribution, Elbow curve and Design vs. Popularity scatter plot.

Pandas used for data handling, scikit-learn for K-Means and scaling and Matplotlib/Seaborn for plotting. The Streamlit UI includes sliders and dropdowns for user inputs, and charts (including the Elbow plot and a cluster scatter plot). The code computes clusters when recommendation demanded and updates the recommendation list in real time. This stack (Pandas, scikit-learn, Matplotlib[6], Seaborn[7], Streamlit) provides an end-to-end data-to-dashboard pipeline.

III.RESULT

The methodology was applied to the dataset and the clustering outcomes were examined. **Figure 1** shows a representative scatter plot of websites in the space of design_score (x-axis) vs. popularity_score (y-axis), colored by cluster(Fig. 1).

Scatter plot of websites by Design score (x) and Popularity score (y), with cluster membership indicated. The points form

distinct groups, illustrating how our clustering separates high-design (right-hand) sites from high-popularity (upper) sites[3].

The elbow plot, (*Fig. 2.)* indicates an optimal *K* of around 4. After clustering, we observed coherent groupings[3]. For example, one cluster contains high-design/low-popularity platforms (e.g. productivity tools with polished UI), while another cluster contains extremely popular sites with moderate design (e.g. major search engines and social networks). The interactive dashboard[4,5,6,7] (Streamlit UI) allows visual exploration: users can hover or click to see which sites are in each cluster and filter by score thresholds. When the user selects a criterion (say, *Design*), the system outputs the top- ranked sites by design_score (e.g. those in the high-design cluster) and highlights them in the plot. In the computation, the recommendations align with the expectations (e.g., graphic-intensive websites were suggested as top in design), validating the logic.

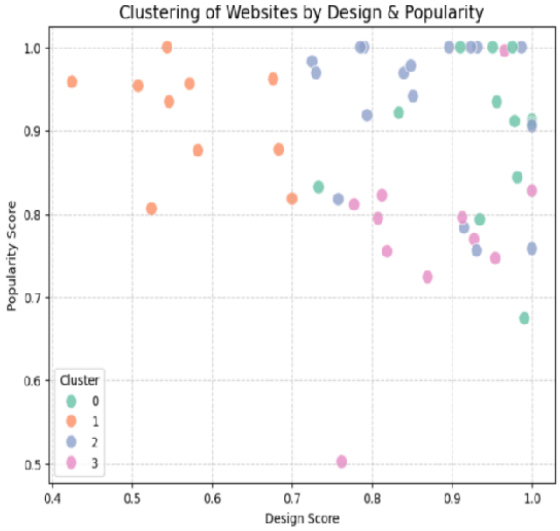

Figure 1: Scatter plot-Clustering by Design vs Popularity

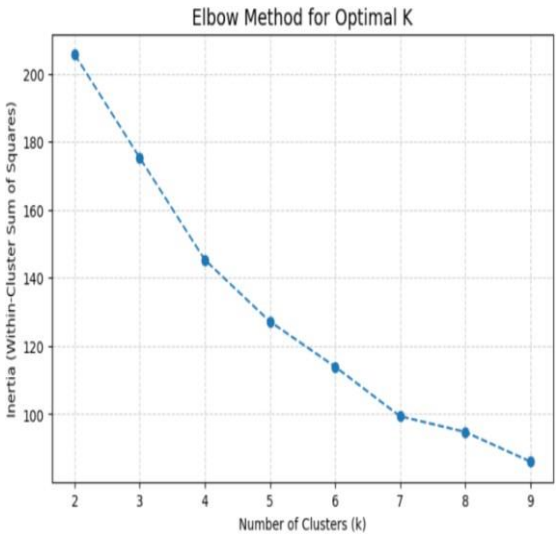

Figure 2: Elbow curve for K-means clustering

Determination of optimal K (Elbow method & stability): The within-cluster sum of squares (WCSS) was computed for K and plotted as an Elbow plot. The Elbow plot showed a marked reduction in marginal gain up to **K = 4**, after which additional clusters produced diminishing returns. Interpretation: the elbow indicate a compact and well-separated three- cluster solution, suitable for high-level categorization (e.g., high-design, high-popularity, underperforming) as obtained from the results from the Elbow curve depicted in the **Figure 3.**

- Cluster sizes are reasonably balanced for K=4; cluster 1 contains many high-traffic mainstream sites.
- Cluster 0 features niche/productivity sites that score highly on design and UX but have smaller traffic footprints.

Cluster Summary (Average Scores per Cluster):

cluster	privacy_score	design_score	collaboration_score	popularity_score	innovation_score
0	0.533	0.937	0.566	0.894	0.598
1	0.746	0.576	0.575	0.914	0.649
2	0.676	0.865	0.609	0.928	0.846
3	0.819	0.873	0.706	0.777	0.660

Figure 3: Cluster Summary for the five scores when K=4

Recommendation Outputs: The recommendation engine ranks sites by domain-specific composite scores and augments ranking with cluster context. Each recommendation card shows (a) domain score driving the ranking, (b) cluster membership, and (c) top 3 contributing raw metrics and their values (e.g., LCP, CLS, mobile_friendly flag). Validation carried out by Consistency checks: when switching focus from Design → Privacy, several sites change rank predictably (sites with explicit privacy disclosures score higher in Privacy ranking), demonstrating transparency.

Score transparency: Each composite domain score can be expanded to reveal the raw metrics and the weights used to compute the aggregate. This allows users to audit rankings directly from the cluster summary as depicted in **Figure 4.**

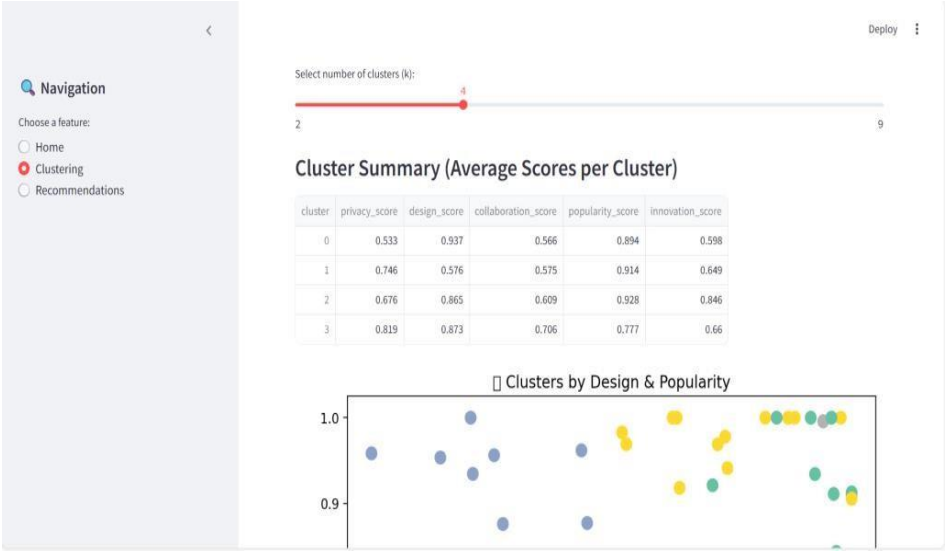

Figure 4: Cluster Summary and Scatter plot of Design vs Popularity provided for scalable adjustments by choosing number of clusters and transparency

Case studies:
Two short case studies demonstrate practical behavior of the pipeline:

Case study 1 - A high-design niche tool (Cluster 0):

Observation: Very high design_score, low monthly_visitors.

Dashboard action: When "Design" is selected, the tool appears in top ranks, with recommendations focused on SEO and discoverability (metadata, sitemap submission) to increase reach without sacrificing design.

Case study 2 - A mass-market platform (Cluster 1):

Observation: Very high popularity_score, moderate design_score.

Dashboard action: When "Privacy" is selected, recommendations suggest improving cookie consent practices and privacy disclosures to raise privacy_score while retaining traffic.

GUI Interpretation and Outcome:

- The combination of domain-level scoring, K-Means clustering (K = 4), and an explainable Streamlit GUI produced interpretable and actionable website groupings.
- Visualization components (Elbow plot, cluster scatter) supported transparent K selection and cluster interpretation.
- The recommender module produced plausible, actionable suggestions that aligned with manual expectations across the Design/Popularity axes that act as key indicators where instead of all the 40+ features used to plot for clustering, the key indicators are used.
- The adjustments of number of cluster groups suggest the clustering is reproducible, though wider validation with more sites and live data is recommended.
- The recommendation option in the sidebar provides multiple-focus area selection not restricting to single focus area instead providing a combination
- The major five features (privacy, innovation, design, collaboration, popularity) are used for clustering as most of the global websites come under the above mentioned category. The features of K-means clustering can be changed as it is a scalable model so that both websites and focus area can be added or adjusted based on the requirements of the technological advancements.
- The top ranked websites based on the focus area provided are plotted in a Bar graph facilitating visual representation of the data as given by **Figure 6.**
- The number of top websites to be displayed can also be altered by the user based on the requirements that are evaluated by the correlation coefficient score displayed as combined_score. (***Fig 5.***)

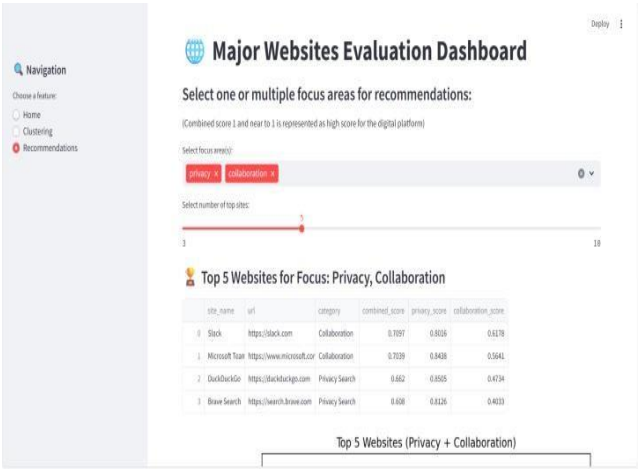

Figure 5: Top websites ranked and displayed based on the combined_score

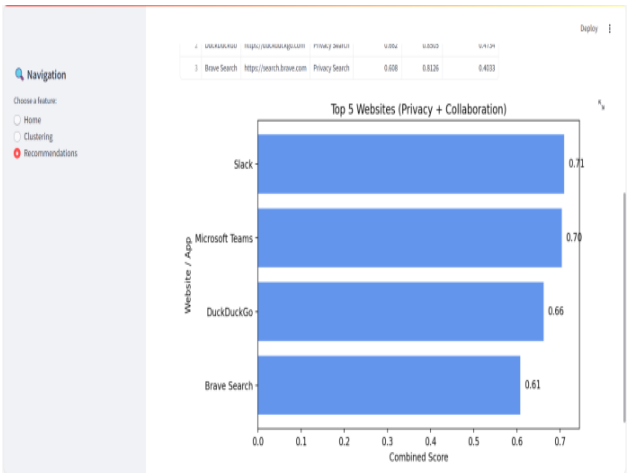

Figure 6: Bar graph plotted for top ranked websites

IV.DISCUSSION

The experimental results from the clustering and recommendation model provide meaningful insights into the structure and diversity of modern digital platforms. The results reveal **emergent relationships** among websites that differ from their nominal classifications. Qualitative examination of the clusters supports this interpretation. One group of clusters primarily contains **high-design and privacy-focused platforms**. For instance, productivity and creative tools emphasize interface consistency and user trust. The second cluster aggregates **high-popularity, content-driven websites/apps** that tends to balance usability with mass accessibility. The third cluster represents **innovative but niche platforms**, characterized by strong technological features (e.g., AI integration) but limited traffic. These findings confirm that the clustering model successfully distinguishes the digital platforms based on orthogonal feature dimensions[1,2], particularly **Design vs. Popularity**, as highlighted in the literature on web usability and recommender transparency.

The results align with prior studies that emphasize the value of **explainable recommendation mechanisms**. Joachims and colleagues have long argued that conventional ranking algorithms are based solely on link structures or click-through rates that lack interpretability and fairness[19]. This model overcomes this limitation by exposing the *reasoning* behind recommendations: users can visualize the clusters, inspect each digital platformss' domain scores, and understand why certain platforms appear as "top-ranked" under specific criteria. Such transparency directly addresses concerns raised in explainable AI (XAI) research regarding user trust and algorithmic accountability[20].

From a methodological perspective, the integration of **K-Means clustering with multi-domain scoring** provides a scalable and generalizable evaluation mechanism. Unlike collaborative filtering that require extensive user interaction data, clustering facilitates the analysis even when behavioral logs are unavailable[3,5], a valuable property for ranking the emerging or privacy sensitive platforms. This feature makes the system particularly adaptable for applications in **digital governance, educational platforms, and institutional data storages**, where interpretability is crucial but user feedback data may be scarce. By using real-time clustering visualization in the Streamlit dashboard, users can interactively explore relationships among websites/apps, visualize the effect of feature scoring and understand the logic behind every recommendation. The model successfully achieves its intended role: **to provide interpretable, data-driven insights into website/app quality beyond popularity metrics**.

The balance between simplicity, transparency, and functionality ensures that the dashboard can serve as both a research prototype and a foundation for scalable, explainable digital evaluation systems for the public.

V.CONCLUSION

A user-friendly GUI model has been developed and evaluated with a dashboard that clusters and recommends digital platforms based on multi-dimensional evaluation metrics. By using K-Means with an Elbow-method selection of *K*, along with an interactive Streamlit interface, users can explore website/app groups and receive data-driven recommendations. This system underscores the importance of considering privacy, design, collaboration, and innovation alongside popularity when ranking digital platforms. The implementation (in Python with Pandas, scikit-learn, Matplotlib/Seaborn, Streamlit) illustrates a practical pipeline for explainable web analytics. In conclusion, the model provides a transparent, extensible framework for digital platform recommendation. The model is scalable facilitating further addition of data and evaluating respect to it.

REFERENCES

1. D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," Journal of Big Data, vol. 9, Art. no. 59, 2022. DOI: 10.1186/s40537-022-00592-5.
2. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,"IEEE Trans. Knowledge Data Eng., vol. 17, no. 6, pp. 734–749, 2005. DOI: 10.1109/TKDE.2005.99.

3. A. K. Jain, "Data clustering: A review," ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999. DOI: 10.1145/331499.331504.
4. W. McKinney, "Data Structures for Statistical Computing in Python," in Proc. 9th Python in Science Conf., 2010, pp. 51–56. DOI: 10.25080/Majora-92bf1922-00a.
5. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Machine Learning Research, vol. 12, pp. 2825–2830, 2011. DOI: 10.48550/arXiv.1201.0490.
6. 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.
7. M. L. Waskom, "seaborn: Statistical data visualization," J. Open Source Software, vol. 6, no. 60, p. 3021, 2021. DOI: 10.21105/joss.03021.
8. SimilarWeb, "Website Traffic & Analytics," Oct. 2025. [Online]. Available: https://www.similarweb.com
9. Statista, "Global Web Usage Statistics and Popular Websites," Oct. 2025. [Online]. Available: https://www.statista.com
10. Qualys SSL Labs, "SSL Server Test," Oct. 2025. [Online]. Available: https://www.ssllabs.com/ssltest/
11. Mozilla Observatory, "Web Security & Privacy Scanner," Oct. 2025. [Online]. Available: https://observatory.mozilla.org
12. Google, "PageSpeed Insights," Web Performance Metrics & Accessibility Scores. Oct. 2025. [Online]. Available: https://pagespeed.web.dev
13. WebAIM, "Contrast Checker for Accessibility Compliance," Oct. 2025. [Online]. Available: https://webaim.org/resources/contrastchecker/
14. G2 Crowd, "Software Reviews for Productivity and Collaboration Tools," Oct. 2025. [Online]. Available: https://www.g2.com
15. Capterra, "User Reviews and Product Features for Business Software," Oct. 2025. [Online]. Available: https://www.capterra.com
16. M. F. Porter, R. A. Baeza-Yates, and B. Ribeiro-Neto, "Measuring Web Popularity and Engagement," ACM Transactions on the Web, vol. 15, no. 2, pp. 1–25, Apr. 2021.DOI: 10.1145/3439863
17. Google, "Core Web Vitals: Essential metrics for a healthy site," Google Developers Documentation, 2024. [Online]. Available: https://developers.google.com/search/docs/appearance/core-web-vitals
18. J. Nielsen, Usability Engineering. Cambridge, MA: Academic Press, 1993.DOI: 10.1016/C2009-0-21559-6
19. T. Joachims, "Optimizing Search Engines Using Clickthrough Data," Proc. ACM SIGKDD, 2002, pp. 133–142. DOI: 10.1145/775047.775067
20. D. Wang, Y. Wang, and Y. Zhang, "Explainable Recommendation: A Survey and New Perspectives," Found. Trends Inf. Retr., vol. 14, no. 1, pp. 1–101, 2020. DOI: 10.1561/1500000066