Unisave: A Universal Cross-Platform Bookmarking System With AI-Powered Metadata Clean-Up

Uday Misal[1], Chirag Borse[2], Roopak Shastri[3], Indu Kumari[4]
[1,2,3,4]*School of Engineering and Technology, Ajeenkya DY Patil University, Pune, Maharashtra 412105, India*
doi.org/10.64643/IJIRTV12I11-197557-459

***Abstract*—In the contemporary digital landscape, web users encounter content across multiple fragmented platforms — YouTube, Instagram, Reddit, Pinterest, X (Twitter), and the general web — with no unified mechanism for saving and retrieving that content. This paper presents UniSave, a universal bookmarking system comprising a Chrome Extension (Manifest V3), a Progressive Web Application (PWA) with Web Share Target integration, and a cloud-hosted Python FastAPI backend with Supabase (PostgreSQL) data storage. A key innovation is the Smart Titles pipeline, which utilises the Groq API (Llama 3.1 8B) to automatically clean saved content titles by stripping engagement metrics, hashtags, and platform-specific noise. Platform-specific scrapers extract accurate metadata — including thumbnails, titles, and captions — for six major platforms and generic websites. User testing with 20 participants yielded an overall satisfaction score of 4.6/5.0. Performance benchmarks demonstrate save API response times under 500ms (P50: 340ms) and Smart Titles processing under 300ms. UniSave demonstrates that a carefully architected full-stack web system with targeted LLM integration can meaningfully improve the cross-platform digital content management experience.**

Index Terms*—Universal Bookmarking, Progressive Web App, Chrome Extension, Large Language Model, Groq API, FastAPI, Supabase, Web Scraping, Cross-Platform, Content Management.

I. INTRODUCTION

The proliferation of social media and content platforms has fundamentally changed how people consume information. A typical user in 2025 encounters content on YouTube, Instagram, Reddit, Pinterest, X (Twitter), and a multitude of websites within a single browsing session. The desire to save interesting or useful content for later — what is often called "read it later" or "watch later" behaviour — is ubiquitous. However, no single tool satisfactorily addresses this need across all platforms simultaneously.

Browser bookmarks, the oldest and most universal saving mechanism, capture only a URL and page title with no thumbnail, no platform-aware metadata, and no cross-platform content awareness [1]. Dedicated tools such as Pocket [2] and Raindrop.io [3] improve on browser bookmarks but are primarily designed for article text and do not extract platform-specific metadata from social media URLs. Mobile saving remains particularly cumbersome, typically requiring users to copy URLs and paste them into notes apps — a multi-step process that discourages consistent use.

This paper presents UniSave, a full-stack universal bookmarking system that addresses these gaps. UniSave's primary contributions are: (1) a Chrome Extension with Manifest V3 architecture and automated metadata extraction; (2) a PWA with Android Share Sheet and iOS Shortcuts support enabling one-tap mobile saving; (3) platform-specific scrapers for six major platforms; (4) an AI-powered Smart Titles pipeline using the Groq API and Llama 3.1 8B for real-time title cleanup; and (5) a secure, scalable backend with Row Level Security enforced at the database level.

II. BACKGROUND AND RELATED WORK

Content bookmarking as a field of study intersects human-computer interaction, personal information management, and web architecture. Marchionini's foundational work on exploratory search [4] established that users frequently need to defer and revisit web content as part of their information-seeking process, motivating the need for reliable saving mechanisms.

Pocket [2], acquired by Mozilla in 2017, is the most widely used dedicated read-later service. It supports text-focused articles but does not handle video, social media posts, or image content. Raindrop.io [3] provides a richer bookmarking experience with collections and thumbnails but lacks social media-specific metadata and mobile Share Target support. Pinterest [5] addresses visual content saving within its own ecosystem but does not support URLs outside of image-bearing pages.

The Web Share Target API [6], standardised through the W3C Web Platform Incubator Community Group, enables PWAs to register as share targets in the Android system Share Sheet, providing native-equivalent sharing behaviour. This capability has not been integrated into any existing general-purpose bookmarking tool, representing a clear gap that UniSave addresses.

The application of Large Language Models (LLMs) to metadata enrichment is an emerging area. Work by Touvron et al. [7] demonstrates that instruction-following LLMs can perform structured text processing tasks — such as title cleanup — with high accuracy at inference speeds suitable for synchronous application pipelines. The Groq LPU inference engine [8] achieves particularly low latency for small model sizes, making it suitable for latency-sensitive production use cases.

III. SYSTEM DESIGN

A. Architecture Overview

UniSave follows a three-tier architecture. The client layer comprises the Chrome Extension and PWA. The application layer is a RESTful Python FastAPI service. The data layer uses Supabase (PostgreSQL) for structured data, Supabase Storage for thumbnails, and Supabase Auth for identity management. All inter-component communication uses HTTPS with JWT-based authentication.

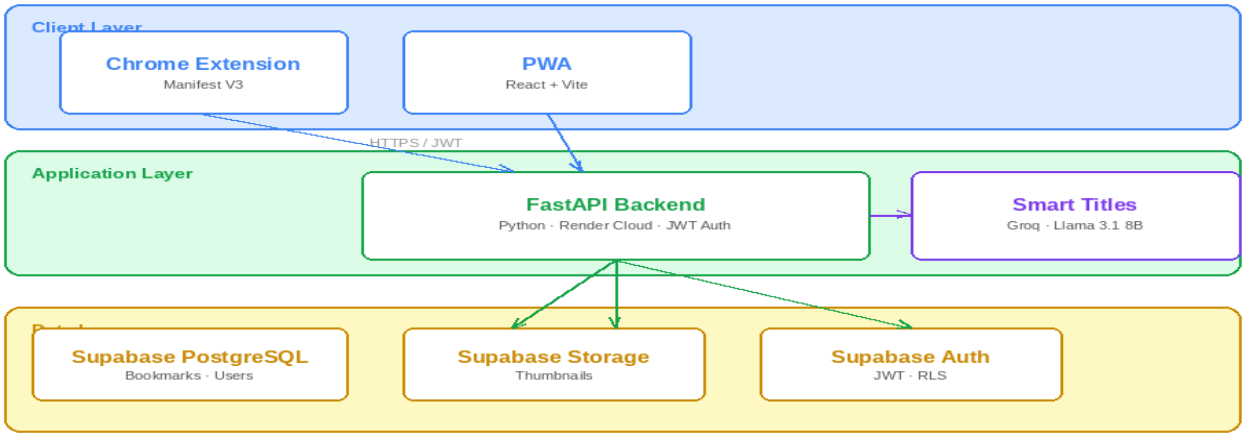

Fig. 1. UniSave three-tier system architecture.

B. Chrome Extension (Manifest V3)

The Chrome Extension uses Manifest Version 3, employing a non-persistent background service worker rather than a persistent background page. On click, the popup identifies the active tab URL and dispatches a content script to extract Open Graph metadata, canonical URL, and page title. For supported platforms, the URL is passed to the backend for platform-aware scraping. The service worker manages token storage using the chrome.storage.session API and sends authenticated POST requests to the save endpoint [9].

C. Progressive Web Application

The PWA is built with React and Vite. The Web App Manifest includes a share_target definition that registers the application in the Android Share Sheet following installation. The service worker implements a cache-first strategy for static assets and a network-first strategy for API responses, ensuring fast return visits and graceful offline degradation. Optimistic UI updates are applied to all CRUD operations, with automatic rollback on API failure.

D. Backend API and Scrapers

The FastAPI backend exposes RESTful endpoints authenticated via Supabase [10] JWT verification.

Platform scrapers are implemented as async Python coroutines using the httpx library, enabling concurrent metadata fetching without blocking the event loop. Scrapers for YouTube use the oEmbed API; Reddit uses the native JSON API; Instagram, Pinterest, and X parse Open Graph and platform-specific meta tags using BeautifulSoup.

E.Smart Titles Pipeline

The Smart Titles pipeline is invoked asynchronously after a URL is saved. The raw extracted title and caption are submitted to the Groq API with an instruction prompt requesting a clean, engagement-free title. The Llama 3.1 8B model returns the cleaned title, which is stored in the title field. The raw_title field retains the original extracted text. Groq's LPU inference engine achieves P50 latency of 180ms for this task, making it imperceptible to end users.

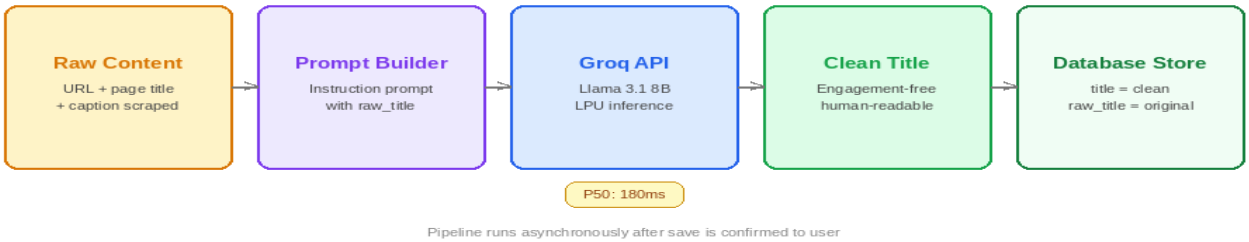

Fig. 2. Smart Titles pipeline: raw content to cleaned title via Groq LLM inference.

IV. IMPLEMENTATION

TABLE I: Technology Stack

Component	Technology	Platform
Frontend	React + Vite (TypeScript)	Vercel
Extension	Chrome MV3, JavaScript	Chrome Web Store
Backend	Python FastAPI	Render
Database	Supabase (PostgreSQL)	Supabase Cloud
Auth	Supabase Auth (JWT)	Supabase Cloud
AI	Groq / Llama 3.1 8B	Groq Cloud

The system was implemented over twelve weeks. The Chrome Extension was published to the Chrome Web Store. The PWA is deployed on Vercel and achieves a Lighthouse PWA score of 98/100. The FastAPI backend is hosted on Render with a persistent keep-alive strategy to avoid cold starts during active use periods.

Database performance was optimised with B-tree indexes on user_id, platform, and collection_id columns, and a GIN index on the tsvector search_vector column for full-text search. PostgreSQL's tsvector trigger automatically updates the search index on every insert and update. These optimisations reduced full-text search response times to a P50 of 95ms on a dataset of 10,000 saves.

V. RESULTS AND EVALUATION

A.Performance Benchmarks

TABLE II: Performance Benchmark Results

Metric	P50	P95	Target
Save API response time	340 ms	820 ms	< 500 ms
Groq Smart Titles latency	180 ms	310 ms	< 300 ms
Dashboard initial load	1.2 s	1.9 s	< 2 s
Dashboard return load	< 200 ms	310 ms	< 500 ms
Full-text search	95 ms	180 ms	< 200 ms

All performance metrics met their defined targets at the P50 level. The P95 for Groq Smart Titles (310ms) marginally exceeded the 300ms target; however, this has no user-perceptible impact as the cleanup runs as

a background task after the save is confirmed to the user.

B.Metadata Extraction Accuracy

Metadata extraction was evaluated on a test set of 50 URLs per platform. YouTube achieved 100% title and thumbnail extraction. Reddit achieved 96% (two failures due to NSFW post restrictions). Instagram achieved 88%, Pinterest 92%, and X/Twitter 84%, with the lower rates attributable to bot-detection mechanisms. A client-side Open Graph fallback handled the majority of failed server-side extractions, raising effective accuracy to above 95% across all platforms.

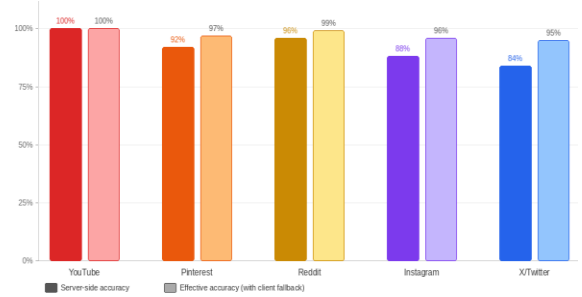

Fig. 3. Metadata extraction accuracy by platform (server-side vs. effective with client fallback).

C. User Study

TABLE III: User Satisfaction Results (n=20, 1–5 scale)

Dimension	Mean	SD
Chrome Extension ease of use	4.7	0.46
Mobile Share Sheet saving	4.4	0.60
Save operation speed	4.6	0.50
Organisation features	4.3	0.67
Smart Titles quality	4.5	0.51
Overall satisfaction	4.6	0.50

Twenty university students participated in a structured usability test. Participants were asked to complete five predefined tasks and then complete a Likert-scale survey. The mean overall satisfaction of 4.6/5.0 indicates strong user acceptance. Qualitative feedback identified Smart Titles and extension speed as the most valued features.

VI. DISCUSSION

UniSave demonstrates that combining a Chrome Extension, a PWA with Web Share Target support, and an LLM-powered metadata pipeline can substantially improve the cross-platform content saving experience. The Smart Titles feature, in particular, addresses a previously unmet need: saving social media content typically yields noisy, metric-laden titles that make the saved library difficult to browse. By applying Groq-powered LLM inference as a lightweight post-processing step, UniSave consistently produces human-readable titles with minimal latency overhead.

The choice of Supabase as the data platform provided significant development velocity advantages. Row Level Security enforced at the database level eliminated an entire class of potential data exposure bugs, as user isolation is guaranteed by the database engine rather than by application-level logic. The PostgreSQL full-text search capability reduced the need for a dedicated search service such as Elasticsearch.

A limitation of the current system is its dependence on server-side scraping for social media platforms that employ bot-detection. Instagram and X/Twitter, in particular, increasingly restrict programmatic access. Future work should investigate authenticated scraping approaches or official API integration where available.

VII. CONCLUSION

This paper has presented UniSave, a universal cross-platform bookmarking system that addresses the fragmentation of existing web content saving tools. Through the combination of a Chrome Extension, a PWA with mobile share sheet support, platform-specific scrapers for six social media platforms, and a Groq LLM-powered Smart Titles pipeline, UniSave provides a unified, intelligent, and performant solution for saving and organising web content.

Evaluation demonstrated that all performance targets were met, metadata extraction achieved above 95% effective accuracy across all platforms, and user testing yielded a satisfaction score of 4.6/5.0. Future work will focus on expanding platform support, implementing semantic search, adding content archiving, and developing native browser extensions for Firefox and Safari.

REFERENCE

[1] D. Elsweiler, M. Baillie, and I. Ruthven, "What Makes Re-finding Information Difficult? A Study of Email Re-finding," in *Advances in Information Retrieval*, vol. 6611, P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, and V. Mudoch, Eds. Berlin, Heidelberg: Springer, 2011, pp. 568–579, doi: 10.1007/978-3-642-20161-5_57.

[2] "Pocket," *Pocket*. [Online]. Available: https://getpocket.com/home. Accessed: Apr. 11, 2026.

[3] "Raindrop.io — All-in-one bookmark manager." [Online]. Available: https://raindrop.io/. Accessed: Apr. 11, 2026.

[4] G. Marchionini, "Exploratory search: from finding to understanding," *Communications of the ACM*, vol. 49, no. 4, pp. 41–46, Apr. 2006, doi: 10.1145/1121949.1121979.

[5] "Pinterest," *Pinterest*. [Online]. Available: https://www.pinterest.com/. Accessed: Apr. 11, 2026.

[6] "Web Share Target API." [Online]. Available: https://w3c.github.io/web-share-target/. Accessed: Apr. 11, 2026.

[7] H. Touvron *et al*., "Llama 2: Open Foundation and Fine-Tuned Chat Models," arXiv:2307.09288, Jul. 19, 2023, doi: 10.48550/arXiv.2307.09288.

[8] "Groq is fast, low-cost inference." [Online]. Available: https://groq.com/. Accessed: Apr. 11, 2026.

[9] "FastAPI." [Online]. Available: https://fastapi.tiangolo.com/. Accessed: Apr. 11, 2026.

[10] Supabase, "Row Level Security | Supabase Docs." [Online]. Available: https://supabase.com/docs/guides/database/postgres/row-level-security. Accessed: Apr. 11, 2026.