International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)
ISSN: 2456-236X
Vol. **11** Special Issue **03** | **2026**

AI-Powered Personal Life Logger with Multilingual Support Digital Memory Jar

Mr. Sagar Parab[1], Mr. Shubham Jadhav[2], Mr. Sahil Kambli[3], Mr. Raghu Ramchandra *Bhogate[4],* Professor Poonam Kadam[5]

[1,2,3,4] *Student, Computer Engineering Department, Metropolitan Institute of Technology and Management Sindhudurg, Maharashtra, India*

[5] *Vice Principal, Computer Engineering Department, Metropolitan Institute of Technology and Management Sindhudurg, Maharashtra, India*

DOI: 10.5281/zenodo.20631221

ABSTRACT

The rapid adoption of digital wellness applications has created a growing demand for intelligent personal journaling systems that go beyond simple text storage. Conventional digital journals fail to extract meaningful emotional and contextual insights from user entries, leaving users without actionable feedback on their mental well-being and personal growth trends. In response to this gap, this paper presents Digital Memory Jar (DMJ) — an AI-powered personal life logger that integrates a multi-stage Natural Language Processing (NLP) pipeline, transformer-based emotion detection, and mood-adaptive music recommendation within a Progressive Web Application (PWA) architecture. The system applies a sequential NLP pipeline to each journal entry: text normalization and lemmatization via spaCy, fine-grained emotion scoring using the multilingual GoEmotions transformer model (AnasAlokla/multilingual_go_emotions), keyword extraction using KeyBERT with paraphrase-multilingual-MiniLM-L12-v2, and zero-shot topic classification using XLM-RoBERTa-large-XNLI. This design enables the platform to support journal entries written in English, Hindi, and Marathi within a single unified backend without language-specific preprocessing. A novel mood-adaptive Spotify music recommendation subsystem constructs contextual queries from the derived emotion and topic signals to deliver affectively congruent musical suggestions. The backend is implemented using FastAPI with an asynchronous APScheduler-based NLP processing queue, ensuring low-latency user experience independent of transformer inference time. The frontend is built with Next.js 15 and Tailwind CSS, delivering a glassmorphic PWA interface with real-time analytics dashboards powered by Recharts. User authentication and data isolation are enforced through Firebase Authentication. The system is cloud-deployed on Vercel (frontend) and Render (backend) with MongoDB Atlas as the persistent data store. Experimental evaluation demonstrates that the transformer-based pipeline accurately classifies emotional content across all three supported languages, while the PWA architecture provides seamless cross-device usability.

Keywords: Digital Journaling, NLP Pipeline, Emotion Detection, Affective Computing, Multilingual NLP, Mood-Adaptive Recommendation, Progressive Web App, Transformer Models, GoEmotions, Mental Wellness Technology

1. INTRODUCTION

1.1 Background of Digital Journaling and Mental Wellness

Personal journaling has long been recognized as a powerful tool for improving mental well-being, emotional regulation, and self-awareness. Research consistently demonstrates that reflective writing helps individuals process experiences, manage stress, and identify behavioral patterns over time. With the proliferation of smartphones and cloud computing, digital journaling applications have seen rapid growth in adoption globally.

However, the digital transformation of journaling has remained largely superficial. Most modern journaling applications including Penzu, Day One, and Journey — function as cloud-synchronized text editors, offering no intelligent analysis of the emotional or thematic content within entries. Users record their thoughts and emotions but receive no structured feedback, no emotional trend visualization, and no personalized support derived from their own writing. This represents a significant missed opportunity, particularly given the advances in transformer-based NLP and affective computing that now make sophisticated text analysis accessible even in consumer-grade applications.

1.2 Need for Intelligent Life Logging Systems

A truly intelligent life logger must do more than store text — it must understand the emotional content of entries, identify recurring themes, track mood patterns over time, and provide contextually relevant feedback to support the user's well-being. The challenge lies in deploying such capabilities in a practical, privacy-respecting, and

multilingual application that works seamlessly across devices.
India presents a particularly relevant context for such a system. With over 500 million internet users and a linguistic diversity spanning Hindi, Marathi, and dozens of regional languages, conventional English-only NLP tools are insufficient for meaningful emotional analysis of the Indian user base. A system capable of processing journal content in English, Hindi, and Marathi within a single unified pipeline addresses a real and underserved need in the digital wellness landscape.
The aim of this research is to design, implement, and evaluate Digital Memory Jar — an end-to-end AI-powered personal life logger that applies state-of-the-art NLP techniques to journal entries in a production-ready Progressive Web Application, with explicit multilingual support for the Indian context.

2.LITERATURE REVIEW

Digital journaling and affective computing have been active areas of research across psychology, human-computer interaction, and natural language processing. Early work by Pennebaker and Smyth established the therapeutic value of expressive writing, demonstrating measurable reductions in anxiety and improved emotional regulation through regular reflective journaling. These findings motivated subsequent research into computational systems that could support and enhance the journaling practice. In the domain of sentiment and emotion analysis, rule-based approaches such as VADER provided interpretable sentiment scores but were limited to English and could not distinguish between fine-grained emotional states. The introduction of transformer architectures — particularly BERT by Devlin et al. and its multilingual variants fundamentally changed the landscape of text-based emotion analysis. The GoEmotions dataset released by Google Research provided 28-class fine-grained emotion labels derived from Reddit comments, enabling supervised training of models capable of nuanced emotional classification far beyond positive and negative polarity.
Keyword extraction has evolved from TF-IDF statistical methods to neural approaches. KeyBERT, proposed by Grootendorst, leverages sentence-transformer embeddings to identify semantically central keywords through cosine similarity, providing more contextually relevant extractions than frequency-based methods. The paraphrase-multilingual-MiniLM-L12-v2 sentence transformer extends this capability to over 50 languages, making it directly applicable to multilingual journal content.Zero-shot topic classification using natural language inference models, particularly XLM-RoBERTa-large trained on the XNLI dataset, has demonstrated strong performance on cross-lingual topic assignment without the need for labeled training data specific to each domain. This approach is especially valuable in personal journaling contexts where predefined topic taxonomies may not match available labeled datasets. Prior work on mood-adaptive music recommendation has explored the relationship between emotional valence and musical features, with platforms such as Spotify incorporating audio valence and arousal dimensions into their recommendation APIs. However, the integration of NLP-derived journaling context — mood labels, keywords, and topics — as direct drivers of a music recommendation query represent a novel cross-modal application that has not been explored in the existing literature.
Progressive Web Application architecture has gained attention as a deployment strategy for wellness applications due to its ability to provide native app-like experiences on both mobile and desktop without the distribution overhead of native application stores. Studies on PWA adoption in health applications have shown improvements in user engagement and accessibility compared to mobile-web alternatives.

3.PROBLEM STATEMENT

The widespread adoption of digital journaling platforms has not been accompanied by a corresponding advancement in the intelligence of those platforms. Users increasingly generate rich emotional and experiential data through their journal entries, yet this data remains unanalyzed and unexploited by existing applications. The core problem addressed by this research is threefold. First, existing digital journaling tools treat entries as opaque text archives with no semantic understanding of their content. Users who wish to understand their own emotional patterns, recurring themes, or mood trends over time have no structured mechanism to do so within current platforms. This leaves a significant gap between the data users generate and the insights they could potentially derive from it. Second, the majority of NLP-powered wellness applications are designed exclusively for English language content, rendering them ineffective for the substantial and growing user population in India whose primary languages of expression are Hindi, Marathi, or other regional languages. The absence of multilingual NLP support in this domain represents a systemic barrier to equitable access to AI-powered wellness technology. Third, the deployment of transformer-based NLP models in consumer-facing applications introduces significant latency challenges. Naive synchronous integration of heavy inference models into web API endpoints would produce unacceptable response times, deterring user adoption. A practical system must decouple NLP processing from user-facing interactions to maintain responsiveness without sacrificing analytical depth. The objective of this study is therefore to develop a full-stack AI-powered personal life logger that: (1) applies a multi-stage transformer-based NLP pipeline to derive structured emotional and thematic insights from journal entries; (2) supports journal content in English, Hindi, and Marathi within a unified processing

architecture; (3) maintains low user-facing latency through asynchronous NLP scheduling; (4) provides real-time emotional trend visualization; and (5) delivers mood-adaptive music recommendations based on the derived emotional context of each entry.

4.PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture of Digital Memory Jar is organized as a three-tier decoupled client-server architecture comprising a Next.js PWA frontend, a FastAPI Python backend, and a MongoDB Atlas cloud database. The separation of concerns across these three tiers enables independent scaling and facilitates asynchronous NLP processing without degrading the user experience.The end-to-end data flow through the system proceeds as follows: the user composes a journal entry via the PWA interface and submits it through an authenticated REST API call. The FastAPI backend immediately persists the raw entry to MongoDB and returns a 201 Created response, ensuring sub-100ms user-perceived latency. A background APScheduler job continuously polls MongoDB for entries that have been preprocessed but lack NLP insights, and processes them in configurable batches through the full NLP pipeline. The enriched data — including emotion scores, mood label, keywords, topics, and AI summary — is written back to the corresponding MongoDB document. The frontend dashboard and analytics pages then consume this enriched data through dedicated REST endpoints, rendering real-time visualizations of the user's emotional and thematic patterns.
When a user requests music suggestions, the Spotify integration subsystem constructs a composite search query by concatenating the derived mood label, the top three extracted keywords, and the top two topic labels, submitting this query to the Spotify Web API to return affectively congruent track recommendations.

5.METHODOLOGY

The methodology employed for the development and evaluation of Digital Memory Jar encompasses system design, NLP pipeline construction, frontend engineering, backend API development, database schema design, and deployment configuration. Each stage is described in detail below.

5.1 NLP Pipeline Design

The NLP pipeline is the intellectual core of the DMJ system, implemented as a sequential six-stage processor in the backend NLP processing module (nlp_processor.py). The pipeline operates on persisted journal entries asynchronously, ensuring that transformer inference does not block user-facing API responses.
Stage 1 — Text Preprocessing: The TextPreprocessor class (text_preprocessor.py) implements a dual-mode architecture. When spaCy is available, the pipeline performs full linguistic analysis including Part-of-Speech tagging and lemmatization using the en_core_web_sm model. When unavailable, the system gracefully degrades to lightweight regex-based normalization. Processing steps include URL removal, special character stripping, Unicode normalization, case folding, and stopword removal.
Stage 2 — Emotion Scoring: Fine-grained emotion classification is performed by calling the AnasAlokla/multilingual_go_emotions model via the Hugging Face Inference API. This model produces probability scores across 28 emotion classes derived from the GoEmotions taxonomy. The system maps these 28 labels to six canonical buckets — joy, sadness, anger, fear, surprise, and disgust — through a curated alias dictionary. Bucket scores are normalized to sum to 1.0, producing a probability distribution over emotional states. The dominant emotion determines the entry's primary mood label.
Stage 3 — Keyword Extraction: Keywords are extracted using KeyBERT with the paraphrase-multilingual-MiniLM-L12-v2 sentence transformer model. The configurable top-N parameter (default 8, set via KEYBERT_TOP_N environment variable) controls extraction granularity. The multilingual sentence transformer enables accurate keyword extraction from Hindi and Marathi entries without any language-specific preprocessing logic.
Stage 4 — Topic Categorization: Zero-shot topic classification is performed using joeddav/xlm-roberta-large-xnli. Eight candidate topic labels are defined covering the primary domains of personal journaling: Work and Productivity, Health and Wellness, Emotions and Mental Health, Relationships and Family, Learning and Growth, Finance, Travel and Leisure, and Daily Life. The input text is augmented with extracted keywords before classification to enrich the semantic signal. A minimum confidence threshold of 0.2 and a maximum of two topic labels per entry prevent low-confidence over-tagging.
Stage 5 — Named Entity Recognition: The pipeline architecture includes a NER stage designed for the multilingual Hugging Face NER model (xx_ent_wiki_sm) with spaCy fallback. This stage is currently maintained as a placeholder to preserve pipeline architecture integrity while avoiding memory overhead in the current deployment environment.
Stage 6 — Embedding Generation: Semantic embeddings are generated using paraphrase-multilingual-MiniLM-L12-v2, with FAISS designated as the vector store for similarity-based memory retrieval. The architecture is designed for full activation in future iterations to enable semantic search over past journal entries.

5.2 Dataset and Feature Description

The system operates on user-generated journal entries rather than a fixed training dataset. Each persisted memory document in MongoDB is structured according to the Memory DB Pydantic schema, capturing both raw and derived attributes for each entry.

Table-1: Memory Document Feature Schema

Field Name	Description	Data Type
content	Raw journal entry text submitted by user	String
content_clean	Normalized and lemmatized version of entry	String
mood	Dominant emotion label (joy / sadness / anger / fear / surprise / disgust)	String
ai_summary	Lightweight AI-generated summary of entry	String
tags	Combined keyword and topic tags (max 5)	Array<String>
emotion_scores	Normalized probability scores across 6 emotion buckets	Object
keywords	Top-N semantically central keywords extracted by KeyBERT	Array<String>
topics	Zero-shot classified topic labels (maximum 2 per entry)	Array<String>
embedding_id	Reference ID for FAISS vector store	Integer
uid	Firebase Authentication user identifier	String
created_at	ISO 8601 timestamp of entry creation	DateTime

5.3 Data Preprocessing

Data preprocessing is handled by the Text Preprocessor class which performs URL removal, punctuation stripping, Unicode normalization, case folding, tokenization, stop word removal, and lemmatization. The preprocessing module implements a dual-mode architecture — full spaCy-based processing when the NLP model is available, and lightweight regex-based fallback otherwise. This resilience ensures deployment viability across resource-constrained cloud environments without compromising pipeline integrity.

5.4 Model Configuration

The DMJ system employs pre-trained transformer models accessed through the Hugging Face Inference API rather than training custom models from scratch. Configuration of model behavior is managed entirely through environment variables, enabling operators to adjust inference timeout, keyword extraction depth, topic confidence thresholds, and maximum topic labels without code modifications.

Table-2: NLP Model Configuration

Pipeline Stage	Model / Tool	Configuration Parameter
Preprocessing	spaCy en_core_web_sm (+ regex fallback)	Auto-detect on startup
Emotion Scoring	AnasAlokla/multilingual_go_emotions	HF_INFERENCE_TIMEOUT_SECONDS
Keyword Extraction	KeyBERT + paraphrase-multilingual-MiniLM-L12-v2	KEYBERT_TOP_N (default: 8)
Topic Classification	joeddav/xlm-roberta-large-xnli	TOPIC_MIN_SCORE / TOPIC_MAX_LABELS
Embeddings	paraphrase-multilingual-MiniLM-L12-v2 + FAISS	EMBEDDING_MODEL env var

5.5 Technology Stack

The complete technology stack employed in Digital Memory Jar spans frontend, backend, database, AI infrastructure, and cloud deployment layers.

Table-3: System Technology Stack

Layer	Technology	Version / Notes
Frontend Framework	Next.js with App Router, React, TypeScript	Next.js 15, React 18
UI & Styling	Tailwind CSS, shadcn/ui, Radix UI Primitives	Glassmorphic design system
Data Visualization	Recharts — line charts, mood distribution	v2.15.4
PWA Support	Service Workers, Web App Manifest	Mobile + desktop install
Authentication	Firebase Authentication (Google OAuth + Email)	firebase v12.6
Backend Framework	FastAPI (Python), Uvicorn ASGI server	FastAPI 0.133, Uvicorn 0.41

NLP Processing	spaCy, KeyBERT, Hugging Face Inference API	Python-based pipeline
Task Scheduling	APScheduler BackgroundScheduler	v3.11.2
Database	MongoDB Atlas (cloud), PyMongo	PyMongo v4.16
External APIs	Spotify Web API, Hugging Face Inference API	Music recommendation
Deployment	Vercel (frontend), Render (backend), Docker	Cloud-native, containerized

5.6 Implementation

The DMJ system is implemented using Python 3.11 for the backend and TypeScript for the frontend. The backend employs FastAPI with Pydantic v2 models for request and response validation, and PyMongo for database operations. All protected API endpoints enforce Firebase Authentication token verification through a dependency injection pattern (verify_firebase_token), ensuring that data access is strictly scoped to the authenticated user's Firebase UID. The frontend is structured as a Next.js 15 App Router application with React Server Components for optimized rendering performance. Protected routes are enforced through a React Context-based authentication provider wrapping Firebase's onAuthStateChanged observable. The glassmorphism visual design — characterized by frosted glass card effects, backdrop blur, and gradient backgrounds — is implemented through Tailwind CSS utility classes, providing a calm and focused aesthetic appropriate for emotional reflection contexts. The Spotify music recommendation subsystem (routers/spotify.py) constructs a composite natural language search query by concatenating the mood label, the top three keywords, and the top two topic labels derived from the NLP pipeline, then submits this query to the Spotify Web API search endpoint. The system returns one primary track recommendation and up to two alternatives, each with album artwork URL and audio preview URL. The web application is deployed as a Progressive Web App with a service worker enabling offline access to previously loaded content and a Web App Manifest enabling installation on Android, iOS, Windows, and macOS devices without requiring submission to any application store.

6. EXPERIMENTAL RESULTS

The experimental evaluation of Digital Memory Jar focused on four key dimensions: NLP pipeline performance across supported languages, system latency, PWA usability, and music recommendation relevance. Results were gathered through functional testing of the deployed system using sample journal entries in English, Hindi, and Marathi.

Table-4: NLP Pipeline Performance Evaluation

Pipeline Stage	Metric	English	Hindi	Marathi
Emotion Scoring	Mood label accuracy	High	High	Moderate-High
Keyword Extraction	Semantic relevance	High	High	High
Topic Classification	Zero-shot accuracy	High	Moderate-High	Moderate
Memory Save API	Response latency	< 100 ms	< 100 ms	< 100 ms
Async NLP Pipeline	Per-entry processing	2–5 sec	2–5 sec	2–6 sec

The decoupled asynchronous architecture achieves its primary performance objective: the POST /memories/ endpoint returns a 201 response within a single database write operation, typically under 100 milliseconds on the MongoDB Atlas free tier, entirely independent of NLP pipeline complexity. This ensures that transformer inference overhead — which varies between 2 and 6 seconds per entry depending on Hugging Face Inference API load — does not degrade the user-perceived experience of saving a journal entry.

The emotion scoring pipeline using the multilingual GoEmotions model demonstrated high accuracy for English and Hindi entries, reflecting the model's multilingual training corpus. Marathi entries showed moderate-to-high accuracy, consistent with the relatively lower representation of Marathi in the model's pretraining data compared to Hindi and English. Zero-shot topic classification via XLM-RoBERTa was most accurate for English entries, with performance gracefully degrading for Hindi and Marathi as expected given the cross-lingual transfer characteristics of the model.The real-time analytics dashboard renders emotion distribution charts, time-series mood trend lines (switchable between weekly and monthly views), a keyword frequency tag cloud, and summary statistics including total memory count, most common mood, and top topics. These visualizations collectively support both immediate emotional awareness and longitudinal pattern recognition across different temporal scales of self-reflection. The mood-adaptive Spotify music recommendation system successfully constructed contextually relevant search queries from NLP-derived journal context in all three supported languages. The composite query strategy — combining mood label, keywords, and topic labels — consistently produced thematically and emotionally relevant track recommendations, with primary recommendations

receiving satisfactory subjective relevance ratings in informal evaluations.

Table-5: System Performance Summary

Performance Dimension	Metric	Achieved Value
Memory Save Latency	API Response Time	< 100 ms
Real-time NLP Analysis (sync)	Analyze Endpoint Latency	2 – 5 seconds
Async NLP Pipeline (batch)	Per-entry Processing Time	2 – 6 seconds
Supported Languages	Active Languages	English, Hindi, Marathi
Emotion Classes	Canonical Buckets	6 (joy / sadness / anger / fear / surprise / disgust)
PWA Installability	Platform Coverage	Android, iOS, Windows, macOS
Music Recommendation	Tracks per Request	1 primary + 2 alternatives

7. CONCLUSION

This research paper presented Digital Memory Jar — an AI-powered personal life logger that integrates a production-deployed transformer-based NLP pipeline, multilingual emotion detection, zero-shot topic classification, and mood-adaptive music recommendation within a Progressive Web Application architecture. The system demonstrates that state-of-the-art affective computing techniques can be practically deployed in consumer-facing wellness applications without requiring specialized hardware or sacrificing user experience.The multilingual NLP pipeline — supporting English, Hindi, and Marathi through deliberate selection of multilingual transformer models within a single unified architecture — addresses a meaningful gap in AI-powered wellness technology for South Asian language communities, where the vast majority of existing tools are English-only. The asynchronous APScheduler-based NLP processing architecture effectively decouples transformer inference from user-facing API latency, achieving sub-100ms response times for journal entry persistence independent of NLP pipeline complexity. This architectural pattern is broadly applicable to any consumer application requiring heavy backend AI processing without degrading user experience.The mood-adaptive Spotify music recommendation feature, which constructs contextual search queries from NLP-derived mood, keyword, and topic signals, represents a novel cross-modal application of journaling-derived affective context to music discovery — a contribution with implications for both the digital journaling and music information retrieval domains. Based on the experimental results, Digital Memory Jar is concluded to be a technically sound and practically deployable intelligent life logging platform. The open-source codebase at github.com/isgr9801/Digital-Memory Jar provides a foundation for further research, community contributions, and extension into clinical or enterprise wellness contexts.

8. FUTURE SCOPE

Even though Digital Memory Jar demonstrates strong foundational capabilities, several directions for future enhancement have been identified. One priority area is the activation of FAISS-based semantic memory retrieval, which would enable users to discover past journal entries that are semantically similar to their current writing — a potentially powerful tool for longitudinal self-reflection and pattern recognition across time. Another important direction is the implementation of the AI Companion conversational feature, which would provide users with empathetic, contextually grounded conversational support based on their personal journaling history. This feature could leverage the user's accumulated emotional context to provide responses tailored to their specific emotional trajectory over time. Anomaly detection over emotional time-series data represents a further enhancement — identifying unusual mood patterns that deviate significantly from a user's established emotional baseline and surfacing these proactively. Similarly, automated weekly reflection summaries could surface longitudinal themes, growth patterns, and notable emotional shifts to support deeper self-awareness.On the infrastructure side, deploying quantized transformer models locally via ONNX runtime or llama.cpp would eliminate the dependency on the Hugging Face Inference API, enabling fully offline operation and eliminating API latency variability. Integration of voice-to-text input through the Web Speech API would also enable hands-free journaling, expanding accessibility for users who prefer spoken expression over typed entries. Finally, connecting the system with complementary wellness data sources — such as wearable device physiological signals, calendar context, or sleep tracking data — could enable multi-modal emotional inference that combines written expression with objective physiological indicators, significantly enriching the accuracy and depth of the emotional insights provided to users.

9. REFERENCES

[1]. Pennebaker, J. W., & Smyth, J. M. (2016). Opening Up by Writing It Down: How Expressive Writing Improves Health and Eases Emotional Pain (3rd ed.). Guilford Press.
[2]. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, pp. 4171-4186.
[3]. Demszky, D., Movshovitz-Attias, D., Ko, J., et al. (2020). GoEmotions: A Dataset of Fine-Grained Emotions. Proceedings of ACL 2020, pp. 4040-4054.
[4]. Grootendorst, M. (2020). KeyBERT: Minimal Keyword Extraction with BERT. Zenodo. https://doi.org/10.5281/zenodo.4461265
[5]. Conneau, A., Khandelwal, K., Goyal, N., et al. (2020). Unsupervised Cross-Lingual Representation Learning at Scale. Proceedings of ACL 2020, pp. 8440-8451.
[6]. Picard, R. W. (1997). Affective Computing. MIT Press, Cambridge, MA.
[7]. Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1.
[8]. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, vol. 30.
[9]. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP 2019, pp. 3982-3992.
[10]. Johnson, J., Douze, M., & Jegou, H. (2019). Billion-Scale Similarity Search with GPUs. IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535-547.
[11]. Juslin, P. N., & Sloboda, J. A. (Eds.). (2010). Handbook of Music and Emotion: Theory, Research, Applications. Oxford University Press.
[12]. Spotify for Developers. (2024). Web API Reference. https://developer.spotify.com/documentation/web-api
[13]. Calvo, R. A., Milne, D. N., Hussain, M. S., & Christensen, H. (2017). Natural Language Processing in Mental Health Applications Using Non-Clinical Texts. Natural Language Engineering, vol. 23, no. 5, pp. 649-685.
[14]. Smyth, J. M. (1998). Written Emotional Expression: Effect Sizes, Outcome Types, and Moderating Variables. Journal of Consulting and Clinical Psychology, vol. 66, no. 1, pp. 174-184.
[15]. Patel, V., Saxena, S., Lund, C., et al. (2018). The Lancet Commission on Global Mental Health and Sustainable Development. The Lancet, vol. 392, no. 10157, pp. 1553-1598.
[16]. Birnbaum, M. L., Rizvi, A. F., Correll, C. U., Kane, J. M., & Confino, J. (2017). Role of Social Media and the Internet in Pathways to Care for Adolescents and Young Adults with Psychotic Disorders. Psychiatric Services, vol. 68, no. 7, pp. 680-686.