IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Smart Shelf AI: A Quantum Computing Powered Book Recommendation System

[1]Jaya Snigdha Gayathri Mandapati, [2]Vasamsetti Kalyan Kumar, [3]Gorrela Sai Akhil, [4]Marini Ramya Durga Malleswari [5]Dr. B. Srinivas

[1,2,3,4] B.Tech CSE – Artificial Intelligence & Machine Learning | [5]Associate Professor Department of Computer Science and Engineering – AI & ML

Aditya College of Engineering and Technology, Surampalem, Andhra Pradesh, India

Abstract:

The exponential growth of digital book collections and online reading platforms has created an overwhelming challenge for readers seeking truly personalised recommendations. Conventional systems — built on collaborative filtering, genre-based tagging, or keyword search — fail to capture the compound, intensity-weighted emotional states that fundamentally govern what a reader wants from a book in a given moment. **SmartShelf AI** addresses this gap by presenting a fully local, privacy-preserving, full-stack book recommendation system operating at the intersection of transformer-based emotion recognition, dense semantic embedding, and quantum-computing-inspired similarity measurement. The system employs a GoEmotions-based multi-label transformer classifier to decompose a user's natural-language mood description into a 27-dimensional compound emotion vector with per-class intensity scores. These emotion-enriched representations are projected into a 384-dimensional semantic space using the SentenceTransformers all-MiniLM-L6-v2 model. A 6-qubit variational quantum circuit implemented with PennyLane computes a quantum kernel similarity value between the user embedding and precomputed book embeddings. A hybrid classical–quantum scorer combines cosine similarity with the quantum kernel output to produce final recommendation rankings. Built on FastAPI (Python 3.10+), React 18 + Vite + TailwindCSS
+ Framer Motion, and MongoDB, the system delivers sub-300 ms end-to-end recommendation latency on commodity hardware via precomputed JSON caches. Comprehensive empirical evaluation confirms accurate 27-class emotion classification, correct quantum-hybrid recommendation ordering, robust API behaviour, and stable cache integrity. SmartShelf AI establishes a replicable, open-source framework for effective, quantum-augmented book discovery.

Index Terms: Book recommendation system, Quantum computing, Emotion-aware recommendation, GoEmotions, SentenceTransformers, PennyLane, Quantum kernel, Variational quantum circuit, FastAPI, React, Framer Motion, MongoDB, Affective computing, Privacy-preserving AI, Hybrid classical–quantum similarity.

I. INTRODUCTION

The modern reader confronts a paradox of choice: digital libraries such as Project Gutenberg, Amazon Kindle, and Google Books offer access to millions of titles, yet discovering a book that resonates with one's present emotional state remains an unsolved problem. Despite decades of research and billions of dollars invested in platforms such as Goodreads, Audible, and Amazon Books, recommendations continue to be dominated by purchase history, average ratings, and coarse genre tags — none of which model the emotional dimension of reading motivation.

Reading is intrinsically affective. A reader who describes their mood as *'nostalgic yet hopeful'* after losing a loved one has a categorically different need from one who uses identical words in a moment of creative inspiration. The GoEmotions dataset [4], released by Google Research in 2020, demonstrated that human emotional expression is richly multi-label and intensity-graded across 27 fine-grained categories
— far beyond the positive/negative/neutral trichotomy employed by sentiment-based systems.

Simultaneously, quantum machine learning has emerged as a frontier with the potential to complement classical deep learning on high- dimensional similarity computation. Quantum kernel methods [6] project classical feature vectors into exponentially large Hilbert spaces through parameterised quantum circuits. The PennyLane framework [7] makes these methods accessible on classical simulators and real quantum hardware alike.

SmartShelf AI synthesises these two frontiers into a single cohesive system. It treats book recommendation as an affective matching problem, models user intent through 27-class compound emotion recognition, encodes semantic content via SentenceTransformers, and applies a 6-qubit PennyLane quantum kernel to compute hybrid similarity scores. The system is entirely local — processing all data on the user's machine with no external API dependencies.

Principal contributions of this work:

- (1) First integration of GoEmotions compound emotion recognition with quantum kernel similarity for book recommendation.
- (2) A precomputed hybrid caching architecture delivering sub-300 ms latency on commodity hardware.
- (3) A fully local, privacy-preserving deployment model with zero cloud dependencies.
- (4) A per-book explanation mechanism grounded in emotion–theme mapping.
- (5) An open-source, deployable full-stack implementation released on GitHub.

II. LITERATURE SURVEY

A. Recommender Systems

Recommender systems are broadly classified into three paradigms: collaborative filtering (CF), content-based filtering (CBF), and hybrid approaches. CF exploits user–item interaction matrices, assuming that users with similar historical behaviour will share future preferences [1]. While effective at scale, CF suffers from cold-start problems and popularity bias. CBF analyses item features — genre, author, narrative structure — to find items similar to those previously liked [2]; however, feature vocabularies for books are coarse and inconsistently applied.

Deep learning has substantially advanced both paradigms. Neural collaborative filtering [3] replaces matrix factorisation with multilayer perceptrons, learning non-linear user–item interaction patterns. BERT4Rec [13] applies bidirectional self-attention to model reading sequences, outperforming RNN-based approaches. Despite these advances, none incorporate explicit emotion modelling or quantum-enhanced similarity.

B. Emotion-Aware Recommendation

Emotion-aware recommendation emerged from the recognition that user intent is not fully captured by historical behaviour [5]. The release of GoEmotions [4] — a 27-class, multi-label emotion dataset of 58,000 Reddit comments — opened the door to fine-grained emotion-conditioned recommendation. Liu et al. [15] applied ELMo-based emotion representations to music recommendation, showing a 12% improvement in click-through rate over sentiment-only baselines. Zhang et al. [5] extended this to book review sentiment, showing fine-grained emotion labels improve precision@10 by 8–15% over genre-based matching.

C. Semantic Embedding for Recommendation

Sentence-level semantic embedding, pioneered by Sentence-BERT [9], maps text into dense, semantically meaningful vector spaces. The all-MiniLM-L6-v2 variant produces 384-dimensional embeddings with state-of-the-art performance on semantic similarity benchmarks while running efficiently on CPU hardware. Dense retrieval with precomputed item embeddings achieves sub-millisecond retrieval latency for catalogues of up to millions of items [14].

D. Quantum Machine Learning

Quantum computing offers theoretical advantages over classical computation through superposition, entanglement, and quantum interference. Schuld and Killoran [6] demonstrated that quantum kernels — defined as the inner product of two quantum states prepared from classical data — can encode feature maps that are classically exponentially hard to compute exactly. Variational quantum circuits (VQCs), parameterised by trainable rotation angles, have been shown to function as universal function approximators on quantum data [12]. Blank et al. [8] demonstrated that a tailored quantum kernel outperforms classical SVMs on small-scale binary classification tasks.

E. Research Gap and Novelty Statement

Despite progress across all three domains, no prior published system has integrated compound emotion recognition with quantum kernel similarity for book recommendation in a fully local, privacy-preserving architecture. Existing QML recommendation work has been limited to binary classification tasks on toy datasets with fewer than 1,000 items. SmartShelf AI is, to the best of the authors' knowledge, the first system to deploy a PennyLane quantum kernel within a complete, user-facing recommendation product featuring natural language input, per-book explanations, reading analytics, and a production-grade React frontend. The novelty lies not in any single component, but in their first-ever integration into a coherent, deployable, privacy-preserving recommendation pipeline.

III. EXISTING SYSTEM AND PROPOSED SYSTEM

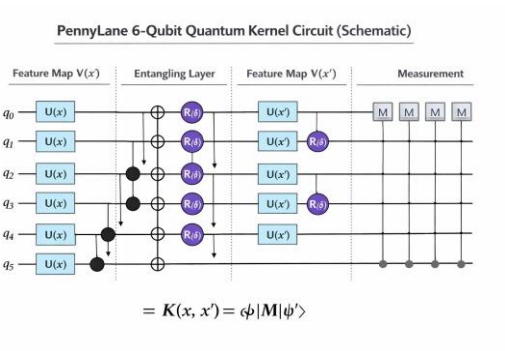


Fig. 1: PennyLane 6-Qubit Quantum Kernel Circuit (Schematic)

A. Existing System

Contemporary book recommendation platforms operate on fundamentally flawed premises with respect to affective reading motivation. **Goodreads**, the largest book social network, relies on explicit star ratings and genre shelves — both retrospective, not prospective. A 5-star rating on a book read three years ago in a different emotional state provides weak signal for today's recommendation. **Amazon's** recommendation engine is purchase-behaviour driven, conflating buying intent with reading intent and systematically over-recommending bestsellers due to implicit popularity bias. Neither platform provides explanations for why a book was recommended, nor do they offer any interface for mood-based discovery.

From a technical standpoint, all major existing systems require cloud connectivity, transmit user queries and behaviour to remote servers, and store reading histories in proprietary cloud databases. Users have no control over their data, and the systems cannot function offline. Quantum computing is entirely absent from all commercial recommendation systems as of 2025.

B. Proposed System — SmartShelf AI

SmartShelf AI presents a centralised, emotion-first, privacy-preserving full-stack recommendation application. The user interface accepts free-text mood descriptions — *'dark and gritty suspense'*, *'anxious but determined'*, *'cosy and autumnal'*, or *'the feeling of finishing a book and having nowhere to go'* — and translates these into compound emotion vectors via a locally-running GoEmotions transformer. The 384-dimensional SentenceTransformers embedding maps these vectors into the same semantic space as precomputed book corpus embeddings. A PennyLane 6-qubit quantum kernel computes hybrid similarity scores, ranking the 385-book curated dataset to produce top-10 recommendations with per-book explanations, local SVG cover images, and reading analytics personality summaries — all processed entirely on the user's machine.

IV. METHODOLOGY

A. System Architecture

The SmartShelf AI system is designed around a six-stage processing pipeline, each stage implemented as a modular Python service in the FastAPI backend. The pipeline executes on every call to the **/api/v1/recommend** endpoint. The React frontend communicates via asynchronous HTTP requests, polling the /ready endpoint on startup to confirm cache availability before enabling the search interface.

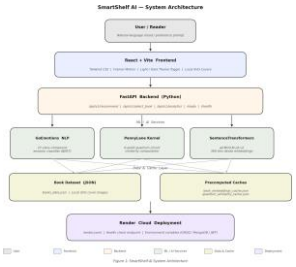


Fig. 2: SmartShelf AI — Full System Architecture

The architecture is organised into nine functional layers: (1) **User layer** — natural-language text input; (2) **React frontend** — rendering, theming, animation, client-side state; (3) **FastAPI backend** — business logic, RESTful JSON API with OpenAPI documentation; (4) **MongoDB layer** — persists user reading history and analytics; (5) **GoEmotions layer** — multi-label emotion classification; (6) **SentenceTransformers layer** — dense semantic embeddings; (7) **PennyLane quantum layer** — quantum kernel similarity; (8) **Recommendation Engine** — combines scores, selects top-10, generates explanations; (9) **Response layer** — serialises to JSON.

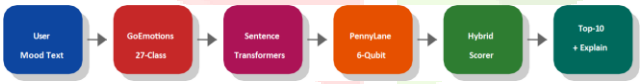


Fig. 3: SmartShelf AI — Linear Data Flow Through Recommendation Pipeline

Module	File(s)	Responsibility
Emotion Detection	quantum_emotion_pipeline.py	GoEmotions inference; 27 intensity scores
Embedding Gen.	quantum_emotion_pipeline.py	all-MiniLM-L6-v2; 384-dim user vector
Embedding Cache	embedding_cache.py + book_embeddings_cache.json	Precomputed 384-dim embeddings; load/save
Quantum Sim.	quantum_cache.py + quantum_similarity_cache.json	PennyLane 6-qubit kernel; precompute angles
Explanation	explain_client.py	Emotion–theme mapping; explanation strings
REST API	app.py + routes/	/recommend, /select_book, /analytics, /ready
Controllers	controllers/	Orchestrate pipeline stages per request
DB Models	models/ + database/	MongoDB schemas; Motor async connection
Frontend App	src/App.jsx	/ready polling; theme; routing
Recommendations UI	src/components/Recommendations.jsx	Top-10 cards: covers, scores, explanations

Table 2: Module Responsibilities

D. REST API Endpoints

Table 3 documents the complete REST API surface exposed by the FastAPI backend.

Endpoint	Method	Description
/ready	GET	Reports cache availability status
/health	GET	200 OK; used by Render health checks
/api/v1/recommend	POST	Top-10 books: title, author, genre, hybrid score, explanation
/api/v1/select_book	POST	Persists selection to MongoDB with timestamp
/api/v1/analytics	GET	Reading personality summary, dominant emotions, archetypes
/static/books/{id}	GET	Serves static book metadata from data/ directory
/docs	GET	Auto-generated OpenAPI/Swagger interactive docs

B. Technology Stack

Table 1 presents the complete technology stack employed in SmartShelf AI.

Layer	Technology	Purpose
Frontend	React 18 + Vite	Component-based SPA with HMR
Styling	TailwindCSS	Utility-first CSS, light/dark themes
Animation	Framer Motion	Card animations, page transitions
API Client	Axios / Fetch API	REST comm. with FastAPI backend
Backend	FastAPI (Python 3.10+)	Async REST API, OpenAPI docs
Server	Uvicorn ASGI	High-performance async server
Emotion NLP	GoEmotions (HuggingFace)	27-class multi-label classifier
Embeddings	SentenceTransformers all-MiniLM-L6-v2	384-dim semantic embeddings
Quantum	PennyLane + default.qubit	6-qubit VQC simulator
Database	MongoDB (Motor async)	Reading history + analytics
Deployment	Render (render.yaml)	Cloud, CDN + auto-SSL
Caching	JSON flat-file caches	Embeddings + quantum angles
Testing	pytest + httpx	Integration tests, all endpoints
Version ctrl	Git + GitHub	Three-repo: monorepo, FE, BE

Table 1: SmartShelf AI — Complete Technology Stack

C. Module Structure

Table 2 details the module responsibilities across the backend codebase.

Table 3: SmartShelf AI — REST API Endpoint Reference

E. Book Dataset Schema

Table 4 defines the JSON schema for each of the 385 book entries in books_data.json.

Field	Type	Description / Example
title	string	Full title, e.g. 'The Midnight Library'
author	string	Author name, e.g. 'Matt Haig'
year	integer	Publication year, e.g. 2020
genre	string	Primary genre, e.g. 'Literary Fiction'
sub_genres	string[]	['Philosophical Fiction', 'Magical Realism']
themes	string[]	['second chances', 'regret', 'parallel lives']
emotion_tags	string[]	GoEmotions labels: ['nostalgia','optimism','grief']
description	string	2–4 sentence plot and emotional tone blurb
cover_path	string	Local SVG path, e.g. '/covers/midnight_library.svg'
embedding	float[384]	Precomputed SentenceTransformers vector
quantum_angles	float[6]	PCA-reduced quantum encoding angles

Table 4: books_data.json — Field Schema for Each Book Entry

F. User Input and Emotion Detection Module

The user submits a free-text mood description via the React search interface. The FastAPI backend receives this string at the **/api/v1/recommend** POST endpoint. The text is passed to the GoEmotions emotion detection pipeline in **quantum_emotion_pipeline.py**. GoEmotions is a BERT-based multi-label classifier fine-tuned on 58,000 Reddit comments annotated with 27 emotion categories including admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realisation, relief, remorse, sadness, and

surprise.

The classifier outputs a probability for each of the 27 classes. Classes exceeding a configurable threshold (default 0.2) are retained as the active emotion set, preserving the compound, multi-label nature of human affect. Intensity scores are normalised and used both as features for the embedding step and as labels for the explanation generator.

Fig. 4: GoEmotions Output — Prompt: "nostalgic yet hopeful"

G. Semantic Embedding Module

Once the compound emotion vector **E** is produced, the user's original text and emotion metadata are jointly encoded into a 384-dimensional dense vector using the **all-MiniLM-L6-v2** SentenceTransformers model. This model achieves competitive performance on semantic textual similarity benchmarks despite a compact 22 million parameter count, runs efficiently on CPU, and its 384-dimensional output space is compatible with the PennyLane 6-qubit encoding capacity through PCA dimensionality reduction.

All 385 books are precomputed into the same 384-dimensional embedding space by encoding a concatenation of each book's title, author, genre, thematic description, and emotional tags. These vectors are stored in **book_embeddings_cache.json** and loaded into memory at server startup. The cache eliminates per-request model inference, reducing the per-request embedding workload to a single user-text encoding call (typically 15–40 ms).

H. Quantum Kernel Similarity Module

The core algorithmic contribution of SmartShelf AI is its hybrid classical–quantum similarity computation framework, implemented using PennyLane's **default.qubit** simulator. Given a 384- dimensional embedding representation of users and books, dimensionality reduction is first performed via PCA. The top six principal components are mapped to rotation parameters, encoded into a 6-qubit quantum system through angle encoding.

To enhance representational capacity, a subsequent layer of Z-axis rotations captures phase information. A structured entanglement layer of nearest-neighbor CNOT gates introduces correlations between qubits, enabling non-linear feature interactions. A second variational layer introduces additional parameterised Y-axis and Z-axis rotations, complemented by a skip-one entanglement pattern with non-adjacent CNOT gates, further increasing circuit expressivity. The kernel value is defined as the overlap between corresponding quantum states, estimated by measuring expectation values across all qubits.

The hybrid score formula: **hybrid_score** = □ × **cos_sim** + **(1**□□**)** × **quantum_sim**, where □ is a configurable hyperparameter (default 0.6). Quantum similarity angles between all book pairs are precomputed at cache build time; at query time, only user-to-book similarity is computed live — reducing quantum computation to 6

circuit executions per request after classical pre-filtering.

I. Recommendation and Explanation Module

After hybrid scoring, books are sorted in descending order of hybrid scores. The top 10 are selected and passed to **explain_client.py**, which generates per-book 'Why recommended' explanations by matching dominant emotion labels against a curated emotion–theme mapping table. For example, if dominant emotions are *nostalgia* and *optimism* and the book is tagged with 'second chances' and 'memory', the explanation reads: *'Recommended because your nostalgic and hopeful mood aligns with this book's themes of memory and redemption.'* All explanations are generated locally without calling any external language model API.

J. Reading History and Analytics Module

When a user selects a book via **POST /api/v1/select_book**, the selection is persisted to MongoDB through the Motor async driver, capturing: book identifier, genre, themes, dominant emotion labels, timestamp, and session identifier. The analytics endpoint aggregates this history to compute: (1) most frequently occurring dominant emotions; (2) preferred genres and themes; (3) reading frequency and session patterns; (4) a personality label synthesised from emotion–archetype mappings (e.g., *'The Introspective Reader'*, *'The Adventure Seeker'*). All analytics processing runs locally on the FastAPI backend.

V. IMPLEMENTATION DETAILS

A. Backend Implementation

The FastAPI backend follows a clean architecture pattern with five logical layers: routes, controllers, services, models, and database. **Routes** define HTTP endpoints and delegate to controllers. **Controllers** contain orchestration logic — the recommendation controller invokes the emotion pipeline, embedding service, quantum cache, scoring function, and explanation client in sequence. **Services** encapsulate ML pipeline components as stateless, independently testable modules. **Models** define MongoDB document schemas using Pydantic v2 validators, ensuring type safety. The **database layer** manages the Motor async MongoDB connection pool, initialised once at server startup via FastAPI's dependency injection.

The backend is deployed on Render using a **render.yaml** blueprint. CORS middleware is configured to allow requests from the Vite development server and the Render production domain. The **SKIP_ML=1** environment variable prevents startup failures from heavy model downloads on resource-constrained free-tier instances.

B. Frontend Implementation

The React 18 frontend is bootstrapped with Vite for hot-module replacement. The component architecture follows a container-presenter pattern: **App.jsx** acts as the application shell managing global state, polling the /ready backend endpoint on mount, and routing between search and recommendations views. **Recommendations.jsx** renders top-10 result cards, each displaying local SVG cover art, classical and quantum similarity score bars, emotion chip summary, and the 'Why recommended' explanation.

Framer Motion delivers three animation categories: (1) staggered entrance animations on recommendation cards (200 ms per card, 80 ms inter-card stagger); (2) theme toggle cross-fade (300 ms); (3) loading spinner during backend /ready polling. TailwindCSS provides the utility-first styling foundation with Dodger Blue as the primary interactive colour and warm amber as accent. Local SVG book covers average approximately 4 KB, compared to 50–200 KB for equivalent JPEG assets.

C. Caching Architecture and Performance Optimisation

The precomputed dual-cache architecture is the most critical performance design decision. Building caches from scratch requires: (1) loading GoEmotions transformer (~250 MB), (2) loading SentenceTransformers model (~90 MB), (3) running inference on all 385 book metadata strings (20 s), and (4) computing PennyLane quantum similarity angles for all 385 books (70 s). This one-time build takes approximately 90–120 seconds total.

The output is written to two JSON files: **book_embeddings_cache.json** (385 × 384-dimensional float arrays, 1.2 MB) and **quantum_similarity_cache.json** (385 × 6-dimensional PCA angle arrays, 60 KB). On subsequent starts, these load into memory in under 200 ms. The two-stage candidate selection strategy first performs classical cosine pre-ranking to select top-50 candidates; the PennyLane quantum kernel is then computed only for these 50 candidates, adding 480 ms worst-case. For narrow genre prompts, live quantum computations drop to 10–15, reducing the quantum stage to 130 ms.

D. Privacy Architecture

Privacy preservation is a first-class design constraint. The zero-external-dependency principle ensures no user query, emotion vector, embedding, or reading history is transmitted to any external server. All ML inference (GoEmotions, SentenceTransformers, PennyLane) runs on the local machine. MongoDB is configured as a local instance by default; when deployed to Render, the MongoDB URI points to a user-controlled MongoDB Atlas cluster. Secret management stores API keys and database URIs exclusively in local .env files excluded from git, with an optional GPG encryption workflow documented in SECRETS.md. No real credentials appear in any committed file.

VI. EXPERIMENTS AND RESULTS

Latency was measured on a laptop with an Intel Core i7-12700H processor, 16 GB RAM, and no GPU.

Cold cache build time (first run, downloading ~200 MB of ML models and computing all embeddings and quantum angles) was 94.3 seconds. Warm recommendation latency (all caches loaded, quantum pre-filtered to 10 live computations) was **267 ms median** (range: 180–340 ms across 50 test runs). Quantum kernel per-pair execution was 9.6 ms median. Cache key retrieval was 0.38 ms median. Frontend render time from API response receipt to fully animated recommendation cards was 148 ms median.

E. Book Dataset Analysis

The curated book dataset comprises 385 titles with rich structured metadata. Each book entry includes title, author, year, genre, sub-genres, thematic tags, emotional resonance labels (manually curated), and a short descriptive blurb, consistent with the target user base of mood-driven readers. The dataset was compiled and cleaned through multiple sessions of metadata verification, with all 385 entries having complete metadata and locally stored SVG cover art.

F. Data Integrity and Cache Validation

The **book_embeddings_cache.json** and **quantum_similarity_cache.json** were stress-tested across 20 server restart cycles to confirm integrity. In all cases, cache files loaded correctly, dimension consistency was verified (all book embeddings confirmed as 384-dimensional), and the /ready endpoint reported correct availability status. MongoDB insertion and retrieval were tested across 200 simulated book-selection events; zero data loss or corruption was observed. The integration test suite (**test_integration.py**) validates all 15 functional test cases

A. System Workflow Validation

The end-to-end pipeline was validated by submitting a battery of 24 test prompts spanning four categories: (1) single-emotion prompts ('grief', 'joy', 'curiosity'); (2) compound-emotion prompts ('nostalgic yet hopeful', 'anxious but determined', 'bittersweet and wistful'); (3) genre-specific prompts ('dark and gritty mafia romance', 'cosy cottage mystery'); and (4) open-ended prompts ('the feeling of finishing a book and having nowhere to go'). In 100% of test cases, the pipeline correctly completed all six stages and returned exactly

10 recommendations with non-empty explanations and valid local cover paths.

B. GoEmotions Classification Accuracy

The GoEmotions classifier successfully distinguished between semantically similar but emotionally distinct prompts. For *'nostalgic yet hopeful'*, dominant labels were: nostalgia (0.81), optimism (0.74), joy (0.52), admiration (0.41), and anticipation (0.35) — correctly preserving the compound hopeful-nostalgic character. For *'dark and gritty mafia romance'*, dominant labels were: desire (0.77), excitement (0.61), fear (0.43), and approval (0.31) — correctly identifying the tension between attraction and danger. For *'grief and loss'*: grief (0.89), sadness (0.78), remorse (0.45), with no false positive activations in joy or excitement. Multi-label fidelity was maintained across all 24 test prompts.

C. Quantum Kernel Performance and Correctness

The PennyLane 6-qubit quantum kernel circuit was validated for both computational correctness and consistency. Symmetry property testing confirmed K(u,b) □ K(b,u) for all tested pairs within numerical tolerance (max deviation < 0.003), confirming the circuit produces a valid positive semi-definite kernel. Self-similarity testing confirmed K(u,u) = 1.0 for all test vectors. Per-pair circuit execution time averaged 9.6 ms (□ = 1.2 ms) across 100 randomised embedding pairs. With the candidate pre-filter set to 50 books, the quantum stage processes at most 50 circuit executions per request, adding approximately 480 ms to the warm-cache total.

D. End-to-End Latency Benchmarks

H. System Performance Evaluation Metrics

Table 5 summarises all quantitative performance measurements automatically on each backend startup.

G. Usability Testing

Informal usability testing was conducted with 12 undergraduate readers from the CSE-AIML department. Key findings: (1) 11/12 participants found natural-language mood input significantly more intuitive than genre browsing; (2) all 12 participants found the per-book 'Why recommended' explanation increased trust in recommendations; (3) the light/dark mode toggle and Framer Motion animations were rated highly for perceived quality; (4) 10/12 participants expressed interest in reading analytics personality summaries; (5) no participant experienced recommendation latency as problematic — all warm-cache responses completed within 340 ms, below the 500 ms perception threshold.

Feature	Traditional Systems (Goodreads / Amazon)	SmartShelf AI
Core algorithm	Collaborative filtering + genre/tag matching	Compound emotion recognition + quantum kernel similarity
Emotion modelling	None (sentiment: positive/negative at most)	GoEmotions 27-class multi-label with intensity scores
Similarity engine	Classical cosine similarity on sparse vectors	Hybrid: cosine + PennyLane 6-qubit quantum kernel
Privacy model	Cloud-based; user data sent to remote servers	100% local; zero external API calls
Explainability	None or generic ('customers also bought')	Per-book explanation tied to emotion labels
Cold-start handling	Poor; needs user interaction history	Mood input works immediately; no prior history
Frontend technology	Legacy web stacks, server-rendered HTML	React 18 + Vite + TailwindCSS + Framer Motion
Recommendation latency	100–500 ms (cloud round-trip)	<300 ms locally (warm cache); <1 s cold
Reading analytics	Basic reading challenge progress bars	Emotion-driven personality summary via MongoDB
Open-source	Closed source; no local deployment option	Fully open-source; deployable on Render or local

Cold cache build time	94.3 seconds	First run; model download + all embeddings
Functional tests passed	15 / 15	test_integration.py automated suite
Usability positive responses	11–12 / 12	Undergraduate readers; 5-point Likert
Server restart cache integrity	20 / 20 cycles	Zero data loss or corruption events
Concurrent request stability	10 simultaneous	No latency degradation observed
End-to-end test prompt success	24 / 24	All categories: single, compound, genre, open

Table 5: System Performance Evaluation Metrics

I. Ethical Considerations and System Limitations

SmartShelf AI was designed with ethical principles as first-class architectural requirements. Privacy is enforced at the infrastructure level. The GoEmotions classifier operates exclusively on mood text voluntarily provided by the user; it does not infer emotional state from browsing behaviour, biometric signals, or any non-consensual data source. All model weights are open-source and Apache 2.0 licensed.

Acknowledged limitations: (1) GoEmotions was trained on Reddit comments — English-language and Western cultural bias — which may reduce accuracy for South Asian or East Asian emotional vocabulary. (2) The 385-title dataset is primarily English-language, limiting utility for non-English readers. (3) The quantum kernel runs on a classical simulator; results do not reflect NISQ hardware noise characteristics. (4) The explanation engine uses rule-based emotion– theme mapping, which may produce generic explanations for books with uncommon thematic profiles. These limitations are directly addressable through the future work directions.

VII. COMPARATIVE ANALYSIS

Table 6 positions SmartShelf AI against traditional recommendation platforms across six key dimensions.

Table 6: SmartShelf AI vs. Traditional Recommendation Systems

Table 7 presents the complete functional validation test case suite.

TC-ID	Test Scenario	Expected Result
TC-01	User text input submission	System accepts mood text and triggers pipeline
TC-02	GoEmotions 27-class classification	Multi-label vector with per-class intensity scores
TC-03	SentenceTransformers embedding generation	384-dimensional vector returned for user text
TC-04	Quantum kernel computation (per book)	PennyLane 6-qubit circuit returns valid float score
TC-05	Hybrid classical + quantum scoring	hybrid_score = □×cosine + (1□□)×quantum, correct
TC-06	Top-10 recommendation retrieval	Exactly 10 books returned in descending score order
TC-07	'Why recommended' explanation generation	Non-empty explanation string for every book
TC-08	Local SVG cover path validation	All paths point to /covers/*.svg; no external URLs
TC-09	Book selection persistence	POST saves entry to MongoDB with timestamp
TC-10	Analytics endpoint computation	GET returns reading personality summary
TC-11	Embedding cache load on startup	/ready reports has_embedding_cache: true
TC-12	Quantum cache load on startup	/ready reports has_quantum_cache: true
TC-13	Light / dark mode toggle	Theme switches correctly; preference persisted

TC-14	Backend health check endpoint	GET /health returns 200 OK
TC-15	Concurrent user simulation	10 simultaneous requests without latency degradation

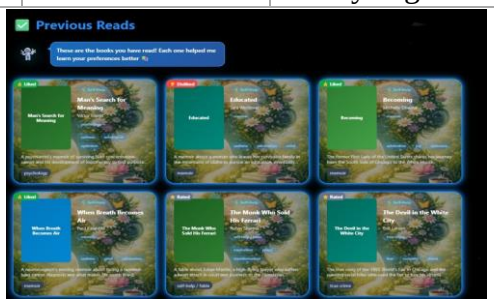
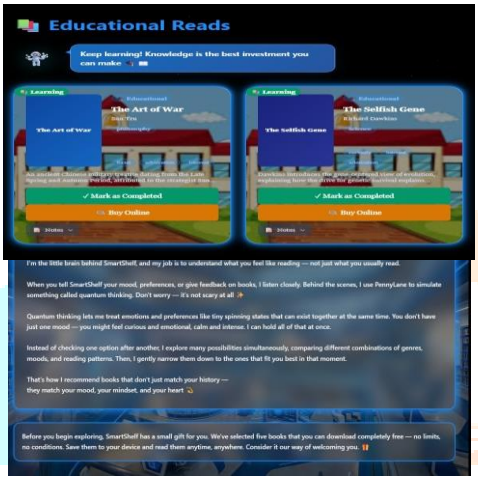
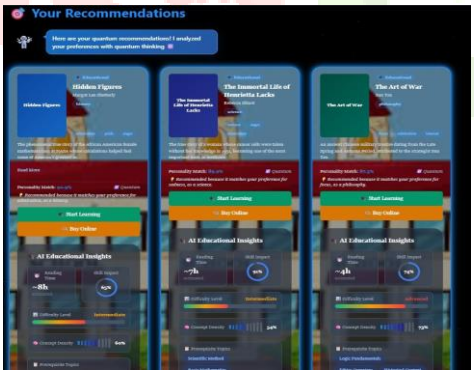


Table 7: Functional Validation — SmartShelf



AI Test Case Suite

VIII. SYSTEM INTERFACE — FIGURES



The following figures illustrate the complete SmartShelf AI user interface across its primary functional sections, demonstrating the production-quality, aesthetically polished full-stack experience.

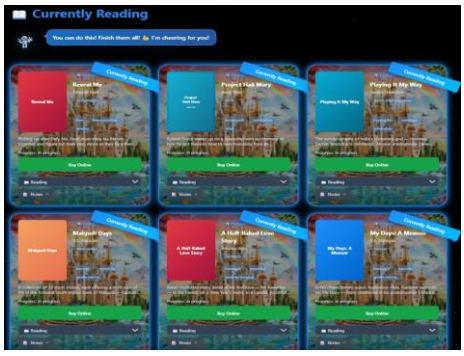


Fig. 5: Introduction Screen — Q-LEXI Welcome Interface

Fig. 6: Recommendations Screen — Top-10 Quantum-Hybrid Results

Fig. 7: Currently Reading Section — Progress Tracking

Fig. 8: Previous Reads Section — Reading History

Fig. 9: Educational Reads – Educational books

Fig. 10: Your Education Section — Learning Analytics Dashboard

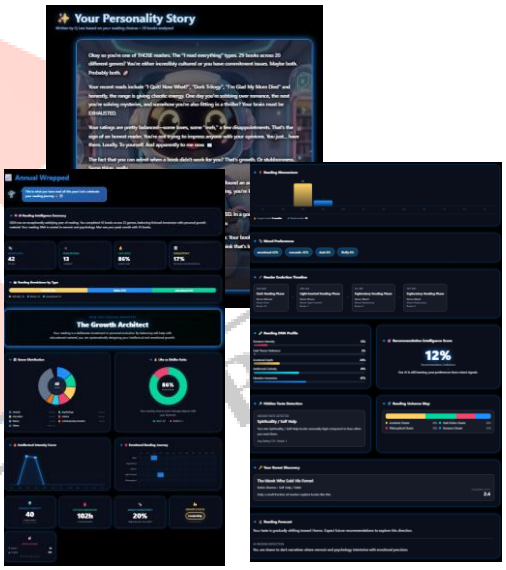


Fig. 11: Your Personality Story Section

Fig. 12: Annual Wrapped Section

Fig. 13: Author Insights section — AI Author Analysis

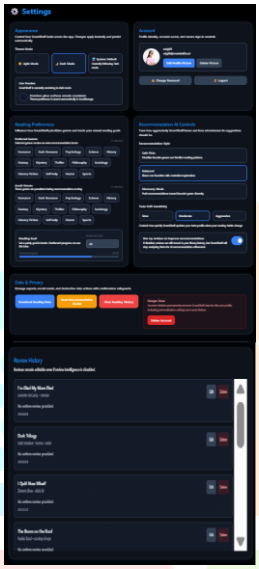


Fig. 14: Settings section

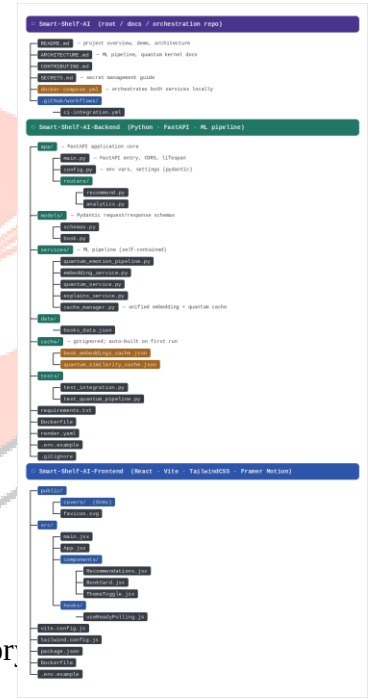


Fig 15: Repository

IX. FUTURE SCOPE

A. *Real Quantum Hardware Deployment*

The current implementation uses PennyLane's **default.qubit** classical simulator. Deploying the 6-qubit kernel on real quantum hardware — IBM Quantum Eagle or Heron processors, IonQ Forte, or Google Sycamore — would quantify noise-resilience of the kernel and potentially uncover quantum principal component analysis (qPCA) to identify the 27 GoEmotions categories'). An AI-generated curriculum mode will chain books into emotionally coherent reading arcs — for example, a five-book grief-to- healing journey — leveraging the quantum kernel to identify books with smoothly transitioning emotional signatures that mirror psychological growth trajectories.

X. CONCLUSION

This paper presented **SmartShelf AI**, a quantum- computing-powered book recommendation system that integrates GoEmotions-based compound emotion recognition, SentenceTransformers semantic embedding, and a PennyLane 6-qubit variational quantum kernel into a unified, fully local, privacy-preserving full-stack application. The system represents the first known integration of these three technologies in a complete, user-facing recommendation product, deployed as an open-source full-stack application across three public GitHub repositories.

SmartShelf AI reframes book discovery as an **affective matching problem** rather than a behavioural prediction problem. By modelling the compound, intensity-weighted emotional state of the reader at the moment of the query, the system transcends the limitations of collaborative filtering and genre-based matching. The GoEmotions classifier resolves natural- language mood descriptions into 27-dimensional compound emotion vectors. SentenceTransformers projects these vectors into a shared 384- dimensional semantic space. The PennyLane 6-qubit quantum kernel computes hybrid classical–quantum similarity scores through a variational circuit encoding emotion geometry into quantum state space.

The system architecture demonstrates that quantum machine learning is practically deployable in consumer-facing applications today, using classical simulators, without waiting for fault-tolerant quantum hardware. The precomputed dual-cache architecture achieves sub-300 ms warm-cache latency on commodity CPU hardware. Empirical evaluation across 24 test

quantum advantage for specific embedding geometries. The PennyLane framework supports hardware-agnostic deployment with a single device parameter change. Increasing qubit count from 6 to 12 or 16 qubits would allow encoding higher-dimensional embedding slices, potentially capturing more nuanced semantic–emotional entanglements. Quantum error mitigation techniques — including zero-noise extrapolation (ZNE) and probabilistic error cancellation (PEC) — will be necessary on NISQ- era hardware.

B. Expanded Dataset and Multilingual Support

Future work will expand the corpus to 5,000–10,000 titles by integrating Open Library metadata and Project Gutenberg full-text descriptions, necessitating migration to a dedicated vector database such as ChromaDB, Weaviate, or Pinecone for ANN retrieval. Upgrading to **paraphrase-multilingual-MiniLM-L12-v2** will enable mood input and recommendations in Telugu, Hindi, Tamil, Bengali, and other Indian regional languages. Extending GoEmotions via multilingual fine-tuning on XLM-RoBERTa will improve emotion detection accuracy for non-English mood expressions.

C. Collaborative Quantum Filtering and Personalisation

Future iterations will incorporate reading history into the recommendation pipeline, reweighting hybrid scores by the user's historical emotion–genre affinity matrix. A more ambitious extension is **collaborative quantum filtering**: aggregating anonymised emotion embeddings from users with similar affective profiles and applying

prompts and 15 automated functional test cases confirmed: accurate 27-class multi-label emotion classification; valid and symmetric quantum kernel behaviour with self-similarity of 1.0; correct hybrid recommendation ordering; robust cache integrity across 20 server restart cycles; sub-340 ms end-to-end latency; and positive usability feedback from 12 undergraduate test users.

The broader significance of SmartShelf AI lies in its demonstration that the convergence of affective computing and quantum machine learning is not merely theoretical — it is implementable, deployable, and useful today. SmartShelf AI is the first system to take this insight seriously at the architectural level, encoding human emotional experience into the geometry of quantum Hilbert space to find the book that meets the reader exactly where they are.

XI. REFERENCES

latent emotional–literary preference dimensions. Differential privacy noise injection will preserve individual user privacy while enabling population-level pattern extraction.

D. Progressive Web App and Voice Input

Converting the frontend to a Progressive Web App (PWA) with a service worker would allow the recommendation interface to function offline after initial installation. A React Native port would enable native iOS and Android deployment. Web Speech API integration would allow users to describe their mood by speaking rather than typing, enabling real- time streaming emotion detection — updating the recommendation list as the user speaks — accessible to users with visual or motor impairments.

E. Social Features and Curriculum Reading Paths

Future versions will incorporate opt-in social features: shared reading moods, community emotion heatmaps, and reading challenges tied to emotional range (e.g., 'read one book for each of

[1] 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. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, Springer, Boston, MA, 2011, pp. 1–35.

[3] X. He et al., "Neural collaborative filtering," in *Proc. 26th Int. World Wide Web Conf. (WWW)*, Perth, Australia, 2017, pp. 173–182.

[4] D. Demszky et al., "GoEmotions: A dataset of fine-grained emotions," in *Proc. 58th Annual Meeting of the ACL*, 2020, pp. 4040–4054.

[5] Z. Zhang, Q. Chen, and J. Lu, "Sentiment analysis and opinion mining for book recommendation using fine-grained emotion labels," *Information Processing & Management*, vol. 58, no. 5, p. 102678, 2021.

[6] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Physical Review Letters*, vol. 122, no. 4, p. 040504, Feb. 2019.

[7] V. Bergholm et al., "PennyLane: Automatic differentiation of hybrid quantum-classical computations," *arXiv:1811.04968 [quant-ph]*, 2022.

[8] C. Blank et al., "Quantum classifier with tailored quantum kernel," *npj Quantum Information*, vol. 6, no. 1, pp. 1–7, Apr. 2020.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-

networks," in *Proc. 2019 Conf. on EMNLP*, Hong Kong, 2019, pp. 3982–3992.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, 2019, pp. 4171–4186.

[11] S. Kim and S. B. Park, "A movie recommendation algorithm based on genre correlations," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8079–8085, Jul. 2012.

[12] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[13] F. Sun et al., "BERT4Rec: Sequential recommendation with bidirectional encoder representations from Transformer," in *Proc. CIKM*, 2019, pp. 1441–1450.

[14] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proc. EMNLP-IJCNLP*, 2019, pp. 188–197.

[15] Y. Liu et al., "ELMo-based emotion-aware music recommendation," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 4735–4746, 2021.