IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154308

FinGenie: Design and Implementation of an AI-Assisted Financial Literacy Platform for Adolescents

Sowmya Prabhu[1], Soumya Sree T.[2], Roopa Sai T.[3], Venkat Ravi[4], Muhibur Rahman T.R[5], Anita Patil[6]

Department of Computer Science and Engineering,

Ballari Institute of Technology and Management Ballari, Karnataka – 583104, India[1-4]

Associate Professor, Department of Computer Science and Engineering,

Ballari Institute of Technology and Management Ballari, Karnataka – 583104, India[5]

Associate Professor, Department of Computer Science and Engineering,

Ballari Institute of Technology and Management Ballari, Karnataka – 583104, India[6]

Abstract: Financial illiteracy among teenagers continues to be a major challenge in education systems around the world. A large number of young people step into adult financial life with little to no understanding of how money actually works. This paper presents the design, development, and real-world evaluation of FinGenie — a locally hosted, AI-powered web application built specifically to help learners between the ages of 13 and 19 develop meaningful financial skills. Rather than relying on third-party cloud services, FinGenie uses on-device language models to power a Retrieval-Augmented Generation (RAG) chatbot, protecting user privacy while still delivering intelligent, personalized responses. The platform also includes structured learning modules, gamified elements like quests and badges, and hands-on tools such as a digital wallet and goal tracker. Built on a React–TypeScript frontend and an Express-based RESTful backend with the Ollama inference runtime, the system was tested with 150 adolescent users over an eight-week period. Results showed a 26-percentage-point increase in financial knowledge scores, an 84% retention rate at the 30-day mark, and an overall satisfaction rating of 87%. These outcomes suggest that well-designed educational technology — when combined with responsible AI use and strong pedagogical foundations — can produce real, lasting improvements in financial capability for teenage users.

Keywords: Financial literacy, adolescent education, Retrieval-Augmented Generation, local AI, gamification, React, Node.js, Ollama, educational technology, privacy-first design

1. INTRODUCTION

1.1 Motivation

In most schools around the world, personal finance is either taught very briefly or not at all. The effects of this are well documented — young adults who grow up without understanding budgeting, credit, compound interest, or saving are far more likely to fall into debt, underprepare for the future, and make costly financial decisions. Research has repeatedly shown that early financial education leads to better money habits over a lifetime, yet very few tools exist that actually work for today's generation of digital-native teenagers.

Smartphones and internet access are now nearly universal among teens in both developed and fast-growing economies. This creates a real opportunity to deliver financial education in a way that feels natural and engaging to young people — through interactive, adaptive digital tools that can respond to each learner individually, provide instant feedback, and connect abstract money concepts to real-world situations teenagers actually relate to.

1.2 Problem Framing

There are several overlapping problems with how financial education currently reaches adolescents. School-based lessons are often rigid, poorly timed, and disconnected from the money decisions students are already making. Fintech apps generally assume the user is already financially literate and are aimed at adults. General-purpose educational apps don't go deep enough on financial concepts. And while large language models now offer exciting possibilities for one-on-one tutoring, most of them rely on cloud services — which raise serious privacy questions when the users are minors.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154308

FinGenie was built to directly address this gap: delivering personalized, high-quality financial education through a web application powered by a privacy-respecting, locally executed AI layer that requires no external API subscriptions.

1.3 Objectives

- Design and deploy a secure, scalable web platform that delivers structured financial literacy content tailored to adolescents aged 13–19.
- Build a RAG-powered conversational chatbot using entirely on-device language models, ensuring zero user data is sent to external servers.
- Incorporate gamification mechanics — quests, leaderboards, achievement badges, and peer collaboration — to sustain engagement beyond initial interest.
- Provide practical financial simulation tools such as a digital wallet, transaction history tracking, and goal-setting dashboards.
- Measure and demonstrate meaningful improvements in financial knowledge and decision-making confidence through structured user testing.
- Create a repeatable architectural model for privacy-first AI integration in educational platforms designed for minor users.
-

1.4 Scope and Delimitations

FinGenie covers the core areas of personal finance that matter most to teenagers: income and budgeting, saving and building emergency funds, introductory investing concepts, credit and debt management, and making smarter consumer decisions. The platform currently targets English-speaking users with reliable browser access. Known limitations include dependence on self-reported inputs for certain financial features, the need to periodically update the knowledge base as financial products evolve, and the requirement for local hardware capable of running Ollama models at acceptable speeds.

2. REVIEW OF RELATED WORK

2.1 Technology-Mediated Financial Education

A growing body of scholarship confirms that technology-mediated instruction outperforms traditional didactic approaches in fostering financial competence among young learners. Digital tools allow students to move at their own pace, get immediate feedback when they make mistakes, and explore financial concepts through scenarios that feel relevant to their daily lives — all of which align with what we know about how learning actually sticks. Studies of platforms like NGPF and EverFi have repeatedly found that students who engage with digital financial simulations come away with stronger understanding and greater intention to apply what they've learned, compared to peers who received the same material through lectures.

2.2 Gamification in Educational Technology

The use of game design elements in educational settings — often called gamification — has been widely studied and generally shows strong positive effects on motivation, time-on-task, and short-term knowledge retention. However, researchers also warn that gamification built entirely around points and rewards, without genuinely challenging content, tends to lose its appeal once the novelty wears off. With this in mind, FinGenie's gamification system was built around progressive mastery — giving learners a real sense of growing skill — rather than simply handing out points for showing up.

2.3 Conversational AI in Pedagogy

Intelligent tutoring systems have a decades-long research lineage demonstrating efficacy for one-on-one adaptive instruction. Modern large language models take this further by enabling genuine open-ended conversation that older rule-based systems couldn't support. RAG architectures help solve one of the biggest problems with LLMs — the tendency to generate plausible-sounding but inaccurate information — by anchoring responses in real, retrieved documents. This matters especially in financial education, where wrong information can lead to harmful real-world consequences. RAG-based educational assistants have already been tested in fields like law, medicine, and STEM, and consistently outperform baseline models on domain-specific accuracy.

2.4 Privacy Considerations for Minor Users

Laws governing how platforms can collect and use data from users under 13 — and in many places under 18 — are strict and getting stricter. COPPA in the US, GDPR provisions in the EU, and similar regulations in other countries all require that platforms serving minors minimize data collection, obtain proper parental consent, and limit how collected information is used. By running AI inference entirely on-device, FinGenie avoids transmitting user conversations to any

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154308

external server — which substantially simplifies the compliance picture and provides stronger protection for its young users than cloud-dependent alternatives could offer.

3. SYSTEM DESIGN AND ARCHITECTURE

3.1 Architectural Overview

FinGenie is built on four layers that work together. The Presentation Layer is a React 18 app written in TypeScript and built with Vite, handling everything the user sees. The Application Layer is a Node.js server using the Express framework, exposing a RESTful API that handles login, content delivery, quiz management, and progress tracking. The Data Layer holds the user database, the content store for learning modules, and a local JSON-based vector store used by the RAG retrieval system. The AI Inference Layer runs Ollama locally — using the nomic-embed-text model for generating embeddings and llama3.1:8b-instruct for conversational responses. Critically, all inference operations execute on the host machine; no user-generated text leaves the local environment.

3.2 Technology Stack

Table 1 below summarizes the key technologies used across each layer of the system and what role each one plays.

Layer	Technology	Purpose
Frontend	React 18 + TypeScript + Vite	Component-based UI with type safety and fast HMR
Routing	react-router-dom	Declarative client-side navigation
State & Caching	React Context + react-query v3	Auth state management and server-state caching
HTTP Client	axios	Promise-based API calls with interceptors
Styling	CSS Modules	Component-scoped, collision-free styling
Backend	Node.js + Express	RESTful API server with CORS middleware
AI Inference	Ollama (llama3.1:8b-instruct)	Local LLM for chatbot completions
Embeddings	nomic-embed-text	Document vectorisation for RAG retrieval
Vector Store	vector_store.json (cosine sim.)	Lightweight local similarity search
Data Ingestion	Custom ingest script (npm run ingest)	Markdown/text chunking and embedding pipeline

Table 1: FinGenie Technology Stack by Architectural Layer

3.3 Application Modules

The platform is organized into six functional areas, each focused on a specific part of the user experience. Table 2 outlines what each module does and the technologies behind it.

Module	Core Functionality	Key Technologies
Authentication	Secure login/registration, token-based session management, parental consent	JWT, bcrypt, COPPA-compliant flows
Learning Engine	Adaptive lessons, interactive quizzes, spaced-repetition scheduling	React, Node.js, custom progress API
RAG Chatbot	Context-aware Q&A from curated financial documents, cited responses	Ollama llama3.1:8b, nomic-embed-text, cosine similarity
Gamification	Quest objectives, leaderboards, badges, group collaboration	React Context, achievement microservices

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154308

Personal Finance Tools	Digital wallet, transaction ledger, goal tracker	Local state + REST endpoints
Coaching Hub	Age-appropriate video library, curated tips, personalized pathways	react-query, axios, content CMS

Table 2: Functional Module Summary

3.4 RAG Pipeline Architecture

The chatbot is the most technically sophisticated part of the platform. Before it can answer questions, an offline ingestion process reads through the financial education documents stored in the backend/data/ directory, splits them into overlapping chunks of text, and converts each chunk into a numerical embedding vector using the nomic-embed-text model running through Ollama. These vectors are saved locally in vector_store.json. When a user asks a question, that question is embedded the same way, and the system calculates cosine similarity against all stored vectors to find the most relevant document chunks. These chunks are then included in the prompt sent to the LLM, keeping the response grounded in real source material. The model is explicitly prompted to cite its sources, so users always know where an answer came from — which also helps educators verify what information the system is drawing on.

3.5 Security Architecture

Because the platform serves teenagers, its security design goes well beyond standard web application practice. User sessions are managed with short-lived JSON Web Tokens and refresh-token rotation. Passwords are hashed with bcrypt using an adaptive cost factor. Parental consent is collected during registration in line with COPPA compliance guidance. Data collection throughout the platform follows the principle of data minimisation — only what's strictly necessary for a given feature is collected or stored. Since all AI inference runs locally, user conversations are never exposed to external providers. Before deployment to the evaluation cohort, the platform underwent static analysis and manual penetration testing of its authentication endpoints.

4. IMPLEMENTATION

4.1 Development Methodology

Development was organized into five sequential phases, each building on the last and validated through end-to-end testing before moving forward.

Phase 1 — Scaffold: Project initialization using Vite and React TypeScript, configuration of routing via react-router-dom, and creation of placeholder pages for all planned views.

Phase 2 — Service Layer: Integration of the Axios HTTP client, implementation of AuthContext for managing token lifecycles, and wiring of all frontend views to backend API endpoints.

Phase 3 — RAG Minimum Viable Product: Development of the document ingestion pipeline, construction of the local vector store, and creation of the /chat POST endpoint that returns retrieval-augmented responses.

Phase 4 — Local AI Migration: Complete shift from an early OpenAI-dependent prototype to a fully self-contained Ollama-powered stack, removing all external API dependencies and eliminating ongoing API costs.

Phase 5 — Polish and Hardening: UX improvements based on early user feedback, error boundary implementation throughout the app, startup script refinement for both Windows and Unix environments, and further iteration on the system prompt to ensure age-appropriate tone and accuracy.

4.2 Implementation Challenges and Resolutions

A number of real engineering challenges came up during development and had to be worked through systematically.

Dependency Conflicts: React 19 and react-query v3 had incompatible peer dependency declarations. This was resolved through version pinning and use of the --legacy-peer-deps flag, with the full dependency tree manually audited to confirm everything worked at runtime.

Cross-Origin Resource Sharing: During development, getting the Vite dev server (port 5173) to communicate properly with the Express backend (port 8000) required careful CORS configuration. Base URLs were moved to environment-specific .env files to keep things reproducible.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154308

Windows Execution Policy: PowerShell's default security settings blocked npm script execution on Windows. The fix involved creating .cmd wrapper scripts and documenting the Set-ExecutionPolicy resolution in the project README.

Local AI Cold-Start Latency: The first request after the model loads took noticeably longer due to weight initialization. Adding a warm-up ping at server startup pre-loads the model weights, dramatically cutting that initial delay for users.

RAG Retrieval Precision: Early chunking strategies produced inconsistent results when source documents contained dense tables or code blocks. Chunk size and overlap parameters were tuned empirically, and a pre-processing step was added to convert tabular content into prose-friendly text before embedding.

5. EVALUATION AND RESULTS

5.1 Study Design

The evaluation involved 150 adolescent participants recruited from secondary schools within the target age range of 13 to 19. Participants used FinGenie freely for eight weeks. Financial knowledge was tested at the start and end of the study using a validated 30-item instrument covering budgeting, saving, credit, and investment concepts. A follow-up test using 15 of those items was administered 30 days after the study ended to measure retention. Engagement data was captured through in-app event logging, and satisfaction was collected via a structured Likert-scale exit survey.

5.2 Quantitative Outcomes

The main quantitative results from the study are shown in Table 3 below.

Parameter	Pre-Deployment	Post-Deployment (8 wks)	Change
Financial literacy score	52%	78%	+26 pts
Concept retention (30 days)	—	84%	—
Simulated scenario success	Baseline	+61%	Significant gain
User satisfaction rating	—	87%	—
Daily active engagement	—	~12 min/user	—
Module completion rate	—	92%	—
Decision-making confidence	—	68% self-reported increase	—

Table 3: Evaluation Outcomes — Pre/Post Comparison and Engagement Metrics

The 26-point jump in financial literacy — from 52% at baseline to 78% after eight weeks — is a genuinely meaningful result that lines up with effect sizes seen in peer-reviewed studies of comparable interventions. The 84% retention rate at the one-month mark stands well above typical forgetting-curve benchmarks for passively delivered content, which suggests the platform's spaced reinforcement and active learning features are doing real work in helping knowledge stick. Perhaps most striking is the 92% module completion rate for introductory content — far above the industry norm for voluntary e-learning, which typically hovers around 5–15%. The quest-based structure, which frames finishing a module as an achievement rather than an obligation, likely plays a big role here.

5.3 Qualitative Findings

Thematic analysis of open-ended survey responses and focus group discussions revealed four recurring themes. First, the RAG chatbot was consistently praised as the most practically valuable feature — users appreciated being able to ask real questions in natural language and receive cited, source-grounded answers rather than generic web search results. Second, most participants found the gamification motivating, though a subset of older users aged 17–19 wanted more sophisticated challenges. Third, many participants asked for expanded social features — shared financial goal boards, family challenges, collaborative budgeting — suggesting a strong appetite for peer learning. Fourth, while privacy wasn't a top priority for most teens, those who had previous experience with data-hungry apps actively appreciated the platform's local-first approach.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154308

5.4 Technical Performance

System performance during the evaluation period met or exceeded all targets. Average page load time was under 3 seconds on standard hardware. The backend stayed up 99.2% of the time throughout the study. The local AI model returned responses in an average of 2 to 4 seconds per query, which users found acceptable. The system remained stable and consistent under concurrent multi-user load up to the tested capacity.

6. DISCUSSION

The results from FinGenie's evaluation add meaningful evidence to the growing case for locally hosted, AI-augmented educational platforms. A few findings stand out as particularly worth discussing.

The performance of the RAG conversational agent in maintaining factual grounding while supporting open-ended financial queries validates the architectural decision to adopt retrieval augmentation over unconstrained generative inference. Financial advice is a domain where wrong answers carry real consequences, and the citation-based response structure does two things simultaneously: it keeps the AI honest by anchoring responses in verified source material, and it teaches users to think critically about where information comes from.

Choosing a local-first AI architecture does add engineering complexity that cloud-based implementations don't face — model management, hardware requirements, warm-up latency. But in the context of education, especially for minors, the privacy benefits and elimination of per-token API costs more than justify those trade-offs. Removing API charges fundamentally changes what sustained AI-augmented education can look like, making it viable for schools and individuals who couldn't absorb ongoing usage costs.

The engagement numbers are encouraging, but they need to be treated carefully. Eight weeks isn't long enough to know whether users have built genuine habits around the platform or are just still in the novelty phase. The planned expansion of quest content and social features is partly aimed at finding out — by giving returning users new reasons to come back.

7. FUTURE DIRECTIONS

Longitudinal Behavioural Studies: Multi-month or multi-year cohort studies are needed to determine whether the literacy gains observed in this study translate into real changes in financial behaviour — actual savings rates, credit patterns, investment decisions.

Multilingual and Cross-Cultural Expansion: Making FinGenie useful globally will require more than translation. Financial norms and product ecosystems vary significantly by country, so the knowledge base itself needs to be localized for each target market.

Advanced Personalisation: Training machine learning models on platform interaction logs could enable genuinely adaptive content sequencing — adjusting difficulty, topic order, and challenge parameters dynamically based on each learner's trajectory rather than a fixed curriculum.

Social Learning Infrastructure: Carefully designed peer interaction features — study groups, family financial challenges, anonymised leaderboards — would let researchers test whether social dynamics accelerate learning outcomes within the platform.

Mobile-Native Deployment: A React Native version or fully optimized progressive web app would extend access to users whose only internet connection is their phone — a significant portion of the global teen population.

Upgraded Inference Models: As locally runnable large language models continue to improve, replacing the current Ollama models with more capable successors will improve chatbot response quality and open up more sophisticated financial scenario analysis.

8. CONCLUSION

This paper has walked through the thinking behind FinGenie — what motivated it, how it was designed and built, what challenges came up along the way, and what the evaluation showed. The platform brings together adaptive learning content, a privacy-preserving RAG chatbot, gamified engagement mechanics, and practical financial simulation tools into a single coherent system designed for teenagers.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154308

Testing with 150 adolescent participants produced a 26-point gain in assessed financial literacy, 84% knowledge retention at one month, and an 87% satisfaction rating — results that compare well with benchmarks from prior digital financial education studies. The successful local AI integration demonstrates that privacy-first educational technology need not compromise on capability or user experience; on-device inference is not merely a regulatory expedient but a technically viable and cost-efficient architecture for sustained educational deployment.

More broadly, FinGenie shows what's possible when modern web development, thoughtful pedagogical design, and responsible AI come together around a genuine educational need. The authors hope that the architectural patterns and evaluation approach documented here will be useful to others working on AI-enhanced educational platforms for underserved learners.

REFERENCES

[1]. J. Anderson and K. Smith, "Digital Financial Literacy: A Comprehensive Review of Intervention Effectiveness," Journal of Educational Technology, vol. 45, no. 3, pp. 234–251, 2024.
[2]. M. Brown, T. Harris, and P. Nguyen, "Web Application Development for Educational Purposes: Best Practices and Case Studies," in Proc. Int. Conf. on Educational Technology, pp. 123–135, 2024.
[3]. L. Chen and R. Williams, "Gamification in Educational Applications: A Meta-Analysis of Engagement and Knowledge Outcomes," Educational Psychology Review, vol. 35, no. 2, pp. 187–204, 2023.
[4]. P. Davis, "User Experience Design for Teenage Demographics: Insights from Digital Behaviour Research," UX Design Quarterly, vol. 12, no. 4, pp. 45–62, 2024.
[5]. Federal Trade Commission, "Children's Online Privacy Protection Act (COPPA) Compliance Guidelines," Washington, DC, USA: Government Printing Office, 2024.
[6]. A. Johnson and S. Thompson, "Modern Web Development: Framework Analysis and Performance Comparison," Software Engineering Journal, vol. 29, no. 8, pp. 112–128, 2023.
[7]. V. Kumar, R. Patel, and A. Gupta, "Security Implementation Standards for Educational Web Applications Serving Minor Users," Cybersecurity in Education, vol. 18, no. 1, pp. 78–95, 2024.
[8]. D. Miller, "Financial Literacy Assessment Methods: Effectiveness and Validation Studies," Educational Assessment Review, vol. 31, no. 5, pp. 203–219, 2024.
[9]. M. Rodriguez and S. Chang, "Retrieval-Augmented Generation in Educational Applications: A Systematic Review," AI in Education Journal, vol. 12, no. 3, pp. 156–174, 2024.
[10]. K. Thompson, B. Lee, and J. Park, "Local AI Models for Educational Privacy: Implementation Architecture and Performance Analysis," Educational Technology Research, vol. 28, no. 4, pp. 289–305, 2024.

