Suganya Purushothaman

AI Product Manager | Governance, Security & Responsible AI

PRD for Kids Book Recommendation System(Kittylit)

CONTENTS

APPROVALS

ROLE	TEAMMATE	REVIEWED	STATUS
Product	@xxxxxxxx	MM/DD/YY	Approved
Engineering	@xxxxxxxxx	MM/DD/YY	See comments
UX	@xxxxxxxxx	MM/DD/YY	[Changes Required] Approved
Legal	@xxxxxxxxxx	MM/DD/YY	Not yet reviewed

ABSTRACT

KittyLit is a friendly AI-powered assistant designed to make reading joyful for children and easier for parents. It helps parents discover the right books based on a child's age, interests, and learning needs, while also offering a supportive **Help Chatbot** that answers parenting and book-related questions in simple, empathetic language. Behind the scenes, KittyLit demonstrates AI Agent–driven decision-making (Cache → DB → API) to deliver faster, cost-effective, and safe recommendations — built with strong governance and care.

MVP Scope:
- Dropdown-based recommender (age/genre/year/language).
- Cache-first AI Agent (Cache → DB → API).
- Basic HelpBot with RAG grounding.

Deferred Scope:
- Multilingual support in Future.
- Advanced memory for HelpBot.
- Analytics dashboards for parents.
- Recommendation explainability features.

BUSINESS OBJECTIVES

1. *Simplify book discovery for parents*:
   Provide an easy and engaging way for parents to discover age-appropriate, interest-based, and learning-focused children's books, ensuring every child gets stories and resources that nurture growth.

2. *Offer personalized parenting support through HelpBot*:
   Integrate a fine-tuned chatbot assistant that can respond to parenting and book-related queries in clear, empathetic language, helping parents feel supported in everyday decisions, and takes feedback from the parents for better improved performance.

3. *Demonstrate safe AI Agent–based decision-making*:
   Implement automated flows (Cache → DB → API) that not only improve efficiency but also serve as a showcase for responsible AI Agent orchestration — balancing performance with safety.

4. *Increase engagement and build trust in AI-assisted parenting tools*:
   Position KittyLit as more than a recommendation system — as a trusted parenting companion. By combining useful recommendations with reliable HelpBot guidance, we aim to drive higher user engagement and foster long-term trust.

5. *Embed AI governance and compliance from day one*:
   Ensure fairness, bias checks, safety filters, and compliance with emerging data protection laws (e.g., GDPR, India DPDP Act). By treating governance as a core product pillar, KittyLit differentiates itself as a responsible AI parenting solution.

KPI

Goal	Metric	Question
New User Growth	# New Signups	How many parents sign up?
Engagement	Avg. session length	Are parents actively browsing?
Recommendation Accuracy	% positive feedback	Are suggestions relevant?
HelpBot Satisfaction	CSAT/NPS	Do parents find HelpBot useful?
AI Agent Efficiency	% correct cache/DB/API decision	Is the agent making cost-efficient choices?
RAG Effectiveness	Reduction in hallucinations	Is retrieved content factual?
LLM Accuracy	% factual responses	Are generated answers reliable?

SUCCESS CRITERIA

- ≥ 80% parents report useful recommendations
- ≥ 70% HelpBot positive feedback
- Retention rate > 60% after 3 months
- ≥ 20% reduction in API costs due to cache-first design
- Hallucination rate reduced significantly via RAG + re-ranking

USER JOURNEYS

1. Parent browses book categories via dropdowns → gets recommendations
2. Parent asks HelpBot about suitable books → gets personalized suggestions
3. Parent interacts with HelpBot for general parenting guidance, giving feedback.
4. AI Agent automatically decides whether to use cache, DB, or API for retrieval

🕮 Scenarios

Scenario 1: Parent selects age group 5-7 & genre → receives book suggestions.
Scenario 2: Parent asks HelpBot: 'Recommend bedtime stories for my daughter.'
Scenario 3: Parent continues conversation with HelpBot, which remembers prior queries.
Scenario 4: Parent request triggers AI Agent → system decides to fetch from cache, DB, or API.

Scenario 5: Parents can give their feedback to the Chatbot→ stored in DB and shared with the product/Dev team.

USER FLOW

Parent → UI Dropdowns/HelpBot → Flask Backend → AI Agent → (Cache/DB/API) → RAG Pipeline (FAISS + Re-ranking) → Book Dataset/HelpBot → Response → Parent UI.

FUNCTIONAL REQUIREMENTS

Section	Sub-section	User Story & Expected Behaviors	Screens
Recommendation	Dropdown	As a parent, I want to select age/genre/year to see relevant books. Expected: System fetches recommendations via RAG.	[Dropdown Screens]

Chatbot	HelpBot	As a parent, I want to ask questions and receive personalized suggestions. Expected: LLM responds with memory + RAG grounding.	[HelpBot Modal]
AI Agent	Decision Layer	System should decide whether to fetch from Cache, DB, or API. Expected: Cache preferred, fallback to DB, then API.	[System Flow]
API	Batch Requests	As a system, I want to restrict API calls to 600 and batch them. Expected: Reduced cost, improved efficiency.	[System Logs]
Governance	Firewall & Filters	As a system, I must filter PII and unsafe content before LLM. Expected: Secure, safe responses.	[Governance Filters]

MODEL REQUIREMENTS

Specification	Requirement	Rationale
Model	LoRA + Q-LoRA fine-tuned MiniLM	Lightweight, cost-efficient, accurate
Context Window	Up to 8K tokens	Sufficient for parenting/book Q&A
Latency	<2 seconds	Smooth parent experience
Re-ranking	Enabled	Improves factual grounding
API Strategy	Max 600 calls, batch requests	Cost control
Governance	Retrieval Firewall, Explainability, Guardrails	Ensures safety and compliance

DATA REQUIREMENTS

- Dataset: books_dataset.json (JSON → embeddings for FAISS)
- Metadata-only API queries (age, genre, language, year)
- Cache: most searched data, refreshed every 24 hrs (TTS)
- DB + Cache sync for performance
- Iterative updates: new books added regularly

- Feedbacks added to the DB for the statistical analysis by product team.

PROMPT REQUIREMENTS

- HelpBot refuses harmful requests
- Maintains polite, supportive tone and acceptance tone for feedbacks
- Retrieval Firewall filters unsafe content
- PII filtering before API requests
- Personalization via memory

TESTING & MEASUREMENT

- Unit testing: dropdowns, retrieval, cache
- Component tests: API calls, firewall, DB sync, memory
- Metrics by layer:
  * AI Agent → Decision accuracy, cost savings
  * RAG → Precision/recall, hallucination reduction
  * LLM → Accuracy, factual grounding
- Human evaluation: 100 parent queries scored
- End-to-end: latency, satisfaction, accuracy

RISKS & MITIGATIONS

Risk	Mitigation
Hallucinations	RAG + re-ranking
API overuse	Cache-first + restricted calls
Data leakage	PII filtering, metadata-only API
Prompt injection	Retrieval Firewall, validation
Bias in dataset	Regular dataset reviews

COSTS

- Development: dataset prep, fine-tuning (GPU), RAG setup
- Operational: Flask hosting, FAISS search, inference, caching infra
- Cost control: capped 600 API calls, Q-LoRA optimization

ASSUMPTIONS & DEPENDENCIES

- Parents willing to interact with AI
- Dataset maintained regularly
- Cache refresh acceptable at 24 hr cycle
- Engineering support for observability
- Governance reviews in place

COMPLIANCE/PRIVACY/LEGAL

- GDPR-compliant data handling
- PII filtering before APIs
- Governance: explainability, logs, observability
- Security-first design against prompt injection, leakage
- Ethical AI: bias mitigation, fairness checks

GTM/ROLLOUT PLAN

Phase 1: Internal testing with parent focus group
Phase 2: Beta launch with 100 parents
Phase 3: Public launch via parent communities
Phase 4: Governance audit + scaling to larger market