

Project Title: Modular AI Content Platform

Vision: A highly modular and extensible AI platform that supports: - Custom-trained AI plugins/tools (LLM fine-tuned or RAG-based) - Multi-user chat with different AI personas - Plugin marketplace interface for tool selection - Dynamic prompt engineering and context-based responses

Technology Stack: Frontend: - React.js (component-driven UI) - TailwindCSS (modern styling) - Zustand or Redux (state management) - WebSocket (for real-time chat)

Backend: - Spring Boot 3 (Java) - Spring Security + JWT (Authentication) - WebSocket (Real-time communication) - PostgreSQL (User data, chat logs) - Redis (Optional caching)

AI/LLM Core: - OpenAI GPT API or Ollama LLaMA 3 for local - Vector DB: ChromaDB or Pinecone - Embeddings via Sentence Transformers - Prompt templates per plugin - Fine-tuning via OpenAI CLI or Hugging Face

Features: 1. User Management: - Register/Login - Role-based access (User/Admin) - Session persistence

1. Chat System:

2. WebSocket-based chat

3. Multiple AI plugins per user

4. Threaded chat sessions

5. Plugin-specific context management

6. Plugin Framework:

7. Each plugin = AI tool

8. Types: Prompt-Only, Fine-Tuned, RAG, Multi-modal

9. Registered via PluginManager interface

10. Dynamically route requests based on user selection

11. Admin Dashboard:

12. Upload datasets (.jsonl)

13. Monitor usage

14. Enable/disable plugins per user

15. Plugin Examples:

16. BlogBot (SEO content generator)

17. FinanceRAG (Query Indian finance articles)

18. LegalBot (Fine-tuned for Indian law)

19. ResumeBuilder (Prompt-only)

20. ImageCaptioner (Multi-modal plugin)

21. RAG Flow (Retrieval Augmented Generation):

22. Store data embeddings in ChromaDB

23. Query vector DB to fetch relevant documents

24. Inject results into prompt context for final generation

25. Optional Enhancements:

26. TTS/STT integration (Whisper)

27. Chat export (PDF, Markdown)

28. Plugin SDK for developers

29. Email verification, subscription tiering

Backend Project Structure: /src /main /java /com.aihub /controller /service /model /plugin /config /chat /
llm /resources application.yml /test

Development Timeline: Phase 1: Project Skeleton + Auth (1 week) Phase 2: Chat + Plugin Routing (1 week)
Phase 3: AI Integration (1–2 weeks) Phase 4: Admin Panel + Dataset Handling (1 week) Phase 5: RAG
Implementation (1 week) Phase 6: UI Polish + Testing + Deployment (1 week)

Deployment: Frontend: Vercel / Netlify Backend: Railway / Render / Fly.io VectorDB: Chroma (Docker) or
Pinecone (hosted) Database: Supabase / Neon (PostgreSQL hosted)

Let me know if you'd like: - Sample .jsonl fine-tuning data - Plugin interface code (Java) - RAG pipeline
example (Python) - UI wireframes for the chat interface