🩺 Project Proposal: Medeo plus ✨

👥 Team Members

Name	UTORid	Email
Mianli Wang	`wangm246`	`mianli.wang@mail.utoronto.ca`
Steve Nguyen	`nguy3671`	`st.nguyen@mail.utoronto.ca`

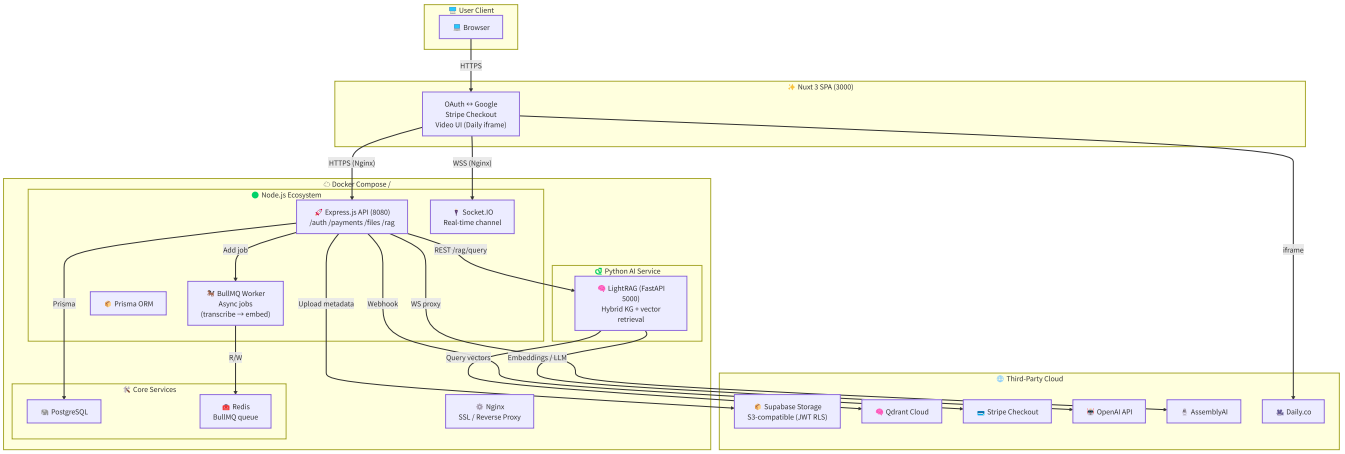
🚀 Project Overview

Medeo plus is a modern web application that streamlines patient–provider communication and mental-health support. Users can exchange **Messages**, and go on video calls with their clinicians in a single, secure workspace.

The platform adopts a **micro-service architecture**, separating core business logic from compute-heavy AI workloads. A **Light Retrieval-Augmented Generation (LightRAG)** server—running in its own Python container—provides hybrid **knowledge-graph + vector** retrieval, source citation, and an Ollama-compatible REST API. This enables truly personalized, evidence-grounded responses based on each user's history and uploads.

🧬 System Architecture

Paste the Mermaid snippet below into the Mermaid Live Editor for a visual diagram.



🛠️ Tech Stack

Layer	Technology
🖼️ **Frontend**	Nuxt 3 (Vue 3) • TypeScript • TailwindCSS
🎨 **UI Kit**	DaisyUI

Layer	Technology
🚪 **Backend API Gateway**	Express.js (TypeScript)
💾 **ORM**	Prisma (PostgreSQL adapter)
⚡ **Real-time**	Socket.IO
🔐 **Auth (OAuth 2.0)**	Supabase OAuth (Google)
💳 **Payments**	Stripe Checkout (test)
🧠 **LLM / RAG**	LightRAG (FastAPI) • OpenAI GPT-4o • LangChain.js *(optional post-processing)*
🌌 **Vector DB**	Qdrant Cloud *(optional optimization)*
🔄 **Queue**	BullMQ + Redis *(optional optimization)*
📹 **Video**	Daily.co
🎤 **Speech-to-Text**	AssemblyAI Streaming API
🚰 **Deployment**	DigitalOcean VM • Docker Compose • Nginx
🚀 **DevOps**	GitHub Actions (lint / CI)

✨ Core Features & Implementation (Killer features)

a. Messages with Hybrid LightRAG Integration

1. Patient and providers can message one-to-one.
2. We will have an AI provider. When the user messages them, **Express API** receives a user prompt and forwards it via **REST** to `light-rag:5000`.
3. **LightRAG** performs hybrid KG + vector retrieval from **Qdrant**, assembles citations and context, calls **OpenAI** for generation, and streams tokens back to Express.
4. Express emits the answer to the browser over **Socket.IO** (with streaming).

b. Live Video + Transcription

- **Daily.co** iframe handles video.
- Browser captures audio → **WebSocket** to Express → proxied to **AssemblyAI** for real-time captions/translation.
- Captions are pushed back through the Socket.IO overlay.

🗓️ Project Milestones

Alpha Version

- Architecture Validation: Achieve a stable local launch of all services using Docker Compose.
- Debug the inter-service communication between Express and FastAPI.
- Basic Features: Build the foundational UI and APIs for the Messages, Appointments, and Documents modules.

Beta Version

- LightRAG Implementation: Complete the RAG data indexing and retrieval pipeline, enabling personalized AI conversations.
- Feature Completion: Integrate the real-time video transcription/translation feature and deploy the full application to a DigitalOcean VM.
- Core Workflow: Implement the complete user flow from OAuth registration to a successful Stripe subscription payment.

Final Version

- Optimization & Bug Fixes: Resolve all identified bugs based on beta testing feedback. Optimize RAG retrieval efficiency and front-end performance.
- Security Hardening: Conduct a thorough review of all authentication, payment, and data-handling processes.
- Documentation & Submission: Finalize all code and documentation for submission to Gradescope.

⚙️ Local Development

```
# 1 Clone
git clone https://github.com/UTSC-CSCC09-Programming-on-the-Web/project-medeoplus
cd medeo-plus

# 2 Env
cp backend/.env.example backend/.env
cp rag-service/.env.example rag-service/.env
# — fill POSTGRES_URL, SUPABASE_JWT_SECRET, STRIPE_KEY, OPENAI_KEY, etc.

# 3 Run
docker compose up --build  # new Docker CLI

# URLs
# • Frontend  http://localhost:3000
# • API       http://localhost:8080
# • LightRAG  http://localhost:5000
```

⚖️ Legal & Ethical

*This academic prototype is **not** a certified medical device. All AI output is informational only and must not replace professional advice.*

- No real PHI or live payment credentials should be used.
- Secrets are injected **only** via GitHub Secrets or Docker secrets—no keys are committed to the repo.