Project Overview & High-Level Architecture

1. Project Vision & Objective

GlycoSight AI is a full-stack, AI-powered web application designed to provide preliminary risk assessment for Type-2 Diabetes. The project's core objective is to revolutionize early detection by empowering users with a secure, intuitive, and highly intelligent tool. By accepting a wide range of multimodal medical documents (lab reports, clinical notes, retinal scans), the application leverages a stateful, agentic AI backend to deliver a comprehensive, evolving, and verifiable diagnostic picture, moving beyond simple, one-off analyses.

2. Live Application & Repositories

The complete application has been deployed and is publicly accessible:

- **Live Frontend:** https://glycosight-ai.vercel.app/
- **Live Backend API:** https://glycosightapi.vercel.app/docs
- **GitHub Repository:** https://github.com/ishan-kshirsagar0-7/GlycoSight-AI

AI Core, Technical Deep Dive & Roadmap

3. The AI Core: An Agentic, Multimodal Workflow

The backend is architected as an advanced agentic workflow using **LangGraph**, enabling a multi-step reasoning process that intelligently handles diverse and unstructured data.

- **Intelligent Routing:** The workflow begins by identifying the input file type (pdf, image, dicom) and routing it to a specialized agent. For images, a VLM agent first classifies the content as either a text-based report or a medical scan, branching the logic accordingly.
- **Stateful Analysis:** The core diagnostic agents are stateful. They fetch the user's existing profile from Supabase, **merge new data with historical data**, and pass the complete, cumulative context to the LLM. This allows the system to build a progressively more accurate understanding of the user's health over time.

- **High-Fidelity In-Context RAG:** To ensure maximum accuracy and verifiability, the system provides the LLM with the full text of established medical guidelines (e.g., ADA's "Standards of Care") directly in the prompt. This leverages Gemini's large context window, allowing it to reason with the entire source document and provide precise, citable analysis without the risk of poor retrieval from a vector database.

---

4. Model Selection: A Data-Driven Decision

While deploying a specialized, open-source medical model like MedGemma-14b-it was explored and successfully tested on GCP, a data-driven decision was made to use the **Google Gemini API** for the final application.

Comparative analysis revealed that the Gemini API delivered **on-par or superior performance** in accurately interpreting medical data, particularly for complex multimodal tasks like analyzing retinal scans. This strategic choice prioritizes output quality, user safety, and lower latency over the complexity of self-hosting, ensuring the best possible result for the end-user.

---

5. Key Differentiators

- **Verifiable & Explainable AI (XAI):** All LLM-generated explanations are backed by source-linked citations and confidence justifications, allowing users to trace the AI's reasoning.
- **Secure & Private by Design:** The content of user-uploaded files is **never stored**. It is processed in-memory to extract parameters and then immediately discarded.
- **Resilient & Iterative:** The system is designed to handle fragmented medical records. Context is retained across multiple uploads, empowering real-world clinical workflows where data arrives over time.

---

6. Future Roadmap

The current architecture provides a robust foundation for several powerful future enhancements:

- **Multi-File Upload:** Implement functionality to allow users to upload a batch of documents at once, enabling the system to build a comprehensive initial profile from multiple sources in a single session.

- **Contextual Editing & Removal:** Develop a UI for users to review their extracted data points and manually correct or exclude specific documents/findings from their analysis history, providing greater control and accuracy.
- **Conversational Doctor-Bot:** Introduce an interactive chat interface where users can provide additional symptoms or context conversationally (e.g., "I've also been feeling dizzy lately"). The bot would process this new information and update the diagnostic results on the page in real-time.