To help you and your partner, I have compiled this comprehensive **Project Execution Report**. You can use this to keep your development on track, and then copy the "Final Report" section into a document to save as a PDF for your portfolio.

📑 PART 1: Project Execution & Workload (The "Plan")

Since you are both second-year students, the goal is to balance learning with delivery. Here is how you should split the work from **January to March**.

Workload Division

Phase	Partner A (AI Engine & Logic)	Partner B (Systems & Interface)
Jan: Foundation	Setup PDF parser (PyMuPDF) and text chunking logic.	Setup **ChromaDB** and write scripts to store/retrieve vectors.
Feb: Intelligence	Build the **LangGraph** "Reasoning Loop" (Search -> Verify).	Build the **FastAPI** backend to handle requests between AI and UI.
Mar: Multimodal	Integrate **Whisper** (Voice) and **Llama Vision** (Images).	Design the **Streamlit** dashboard and **Dockerize** the app.

🛠️ PART 2: Amateur-to-Pro Skill Matrix

Don't worry about being an amateur; these are the specific "mini-skills" you need to pick up.

The "Survival" Skills (Learning Path)

1. **Python Mastery:** Focus on Dictionaries, List Comprehensions, and JSON (AI speaks in JSON).
2. **RAG Logic:** Learn why we "chunk" data (LLMs have a memory limit).

3. **Prompt Engineering:** Instead of "Tell me about X," learn to write: *"You are a medical expert. Using ONLY the provided context, answer X. If unsure, say you don't know."*
4. **Version Control (Git):** Essential for a team of two so you don't overwrite each other's code.

📝 PART 3: The Final Project Report (PDF Content)

Copy the section below into a Word/Doc file and "Export to PDF" to create your formal report.

PROJECT REPORT: MediSight Multimodal AI Agent

Objective: To develop an autonomous AI agent capable of synthesizing medical research from text, diagrams, and voice queries.

1. Problem Statement

Medical researchers spend hours manually cross-referencing PDFs, charts, and audio notes. MediSight AI automates this by providing a unified, multimodal RAG interface that ensures 100% citation accuracy.

2. Technical Architecture

- **Orchestration:** LangGraph for stateful reasoning.
- **Vector Database:** ChromaDB for semantic retrieval.
- **Models:** Llama 3.2 (Text), Llama 3.2 Vision (Images), OpenAI Whisper (Audio).
- **Infrastructure:** Dockerized FastAPI backend with a Streamlit frontend.

3. Key Features

- **Multimodal Input:** Supports .pdf, .jpg/png, and .mp3/wav.
- **Agentic Reasoning:** Self-correcting retrieval loops to reduce hallucinations.
- **Verification:** Automatic source-citation for every claim made by the AI.

4. System Evaluation (Amateur Benchmarking)

We utilized the **RAGAS framework** to score our system on:

- **Faithfulness:** Is the answer derived solely from the documents?
- **Relevancy:** Does the answer actually address the user's question?

🚀 PART 4: Internship Targets (Summer 2026)

Since you are in India, apply to these specific programs between **February and March 2026**:

1. **Google STEP Internship:** Specifically for 2nd-year students. Your "Multimodal" project will put you in the top 5% of applicants.
2. **Microsoft Research (MSR) India:** They hire "Research Fellows." Your focus on LangGraph and citations is exactly what they look for.
3. **IBM SkillsBuild (Virtual):** A 6-week AIML internship officially listed on the **AICTE Portal** starting in January/February.
4. **Startups (Bangalore/Remote):**
   - **Sarvam AI:** Focuses on Indian-language LLMs.
   - **Yellow.ai:** Specializes in AI Agents (like your LangGraph project).
   - **Skit.ai:** Perfect if you emphasize the **Whisper/Voice** part of your project.

Would you like me to draft a custom "Project Summary" for your LinkedIn profile so you can start building a reputation before you finish the project?

Resources

1. Building Agentic RAG with Llamaindex - DeepLearning.AI
2. Foundation: Introduction to LangChain - Python - LangChain Academy
3. Fundamentals of AI Agents Using RAG and LangChain | Coursera
4. Building Applications with Vector Databases - DeepLearning.AI
5. https://youtu.be/-JLUXRcqPnY
6. Introduction to Generative AI | Coursera

3. Prerequisite Checklist

Before you start the project on January 1st, make sure you both have these "Basics" ready:

- Python: You should know how to use `pip install`, handle `JSON` files, and write `try-except` blocks.
- GitHub: Create a shared repository. Learn how to `push`, `pull`, and handle

`branches`.

- **API Access: Sign up for a Google AI Studio account (for free Gemini API keys) or Groq (for super-fast Llama 3 access).**

Week 1 (Jan 1-7) Action Plan:

1. **Both: Watch the first 30 minutes of a LangGraph Tutorial to understand the "Graph" concept.**
2. **Member A: Take the DeepLearning.AI "Agentic RAG" short course.**
3. **Member B: Set up the GitHub repo and a basic "Hello World" Python environment.**