Multimodal Lung Diagnosis System — Work Completed So Far

1. What We Are Building

We are developing a **Multimodal Lung Diagnosis System** that allows patients to submit medical information and doctors to review it through an AI-assisted platform.

Right now, we are in the **Structure Phase**, which means we are creating the full working flow of the system before adding real AI training or datasets.

The goal of this phase is to make sure:

- Frontend and backend communicate correctly
- Patient and doctor interfaces exist
- Data moves properly through the system

---

2. Main Components Created

👱 Patient Interface (Frontend – React)

We built two main patient features:

1) Symptom Checker

- Patient types symptoms
- Data is sent to backend using an API
- Backend receives and responds

2) Report Upload

- Patient uploads:
  - X-ray image
  - Clinical notes
  - Vitals
- Data is sent using FormData
- Backend receives files + text successfully

This proves that our multimodal inputs (image + text + vitals) are working.

👨‍⚕️ Doctor Interface (Frontend – React)

We created a Doctor Dashboard that:

- Displays patient submissions
- Shows symptoms, file names, and vitals
- Displays an AI result placeholder
- Fetches real data from backend instead of fake demo data

This is now a **live dashboard**, not static UI.

⚙ Backend System (FastAPI)

We built a working backend server with these APIs:

POST /symptoms

- Receives patient symptoms

POST /upload

- Receives:
  - X-ray file
  - Clinical notes
  - Vitals
- Stores data temporarily in memory

GET /reports

- Sends stored patient reports to Doctor Dashboard

The backend currently uses temporary storage instead of a database because we are still building the structure.

---

3. Navigation System Added

Instead of showing everything on one page, we implemented role-based navigation:

🏠 Home Screen
→ Patient Portal
→ Doctor Dashboard

Patient Portal contains:

- Symptom Checker
- Report Upload

Doctor Dashboard contains:

- Patient review interface

This makes the application behave like a real medical platform.

---

4. How Data Flows in the System

The working flow is:

Patient enters data
→ React frontend sends request
→ FastAPI backend receives it
→ Backend stores report
→ Doctor Dashboard fetches reports
→ Doctor sees patient information

This confirms that the full data pipeline is operational.

---

5. How We Used Claude AI

Claude AI was used to:

- Generate UI components (ReportUpload, DoctorDashboard)
- Speed up frontend design
- Provide clean React layouts

However, we:

- Integrated the code ourselves
- Connected frontend with backend manually
- Controlled architecture and logic

Claude helped with coding speed, but system design decisions were made by us.

---

6. What We Have NOT Done Yet

The following are intentionally postponed:

- No AI model training
- No datasets added
- No real prediction algorithms

- No database integration

Right now, AI results shown in the dashboard are placeholders.

---

7. Why We Built Structure First

Building structure first ensures that:

- When we train the AI model later,
  we only need to insert prediction logic into backend.
- UI and data pipeline are already ready.
- Integration becomes easier and faster.

This is the same approach used in real-world AI system development.

---

8. Team Work and Fair Contribution

During this phase, team members rotated roles:

- Leader – Generated components using AI tools
- Builder – Implemented logic and backend endpoints
- Integrator – Connected components and navigation

This ensured everyone learned:

- React
- FastAPI
- System integration

---

9. Current System Status (Achievement Summary)

At this stage, our project already includes:

✓ Patient symptom input
✓ X-ray and clinical report upload
✓ Backend data handling
✓ Doctor dashboard displaying real data
✓ Navigation between patient and doctor roles

The application is now a complete full-stack structure ready for AI integration.

---

10. What Will Happen Next (Future Phase)

The next phase will include:

- Adding smart AI logic inside backend
- Connecting real datasets
- Training multimodal models
- Replacing placeholder AI results with real predictions

Because the structure is finished, training can be added without redesigning the system.

End of Document