

AI for Bharat Hackathon

Powered by **aws**



Team Name : PleaseOfferMeAJob

Team Leader Name : Shashank Dubey

Problem Statement : India's rural healthcare system relies on unstructured, multilingual paper records, leading to fragmented patient histories, delayed insurance claims, and increased fraud.

Brief About the Idea:

The Problem:

- Rural clinics rely on handwritten, multilingual medical records
- Insurance TPAs manually verify inconsistent claim documents
- No structured patient history → delays, errors, fraud

Our solution is an AI system that can:

- Converts English & Telugu medical documents into structured clinical events
- Builds a chronological patient health timeline
- Detects suspicious insurance claim patterns
- Runs on low-cost Docker infrastructure (cloud-scalable later)

Vision:

Create a structured, fraud-aware digital health intelligence layer for rural India.

Differentiation, Problem Solving & USP

How Are We Different?

- Not just OCR or document storage
- Converts documents into structured clinical events
- Builds a longitudinal patient timeline
- Integrates fraud detection within the same pipeline
- Designed specifically for rural + multilingual India

How We Solve the Problem

1. Extract medical entities from English & Telugu documents
2. Normalize and structure data into clinical events
3. Generate chronological patient timelines
4. Run fraud detection rules during claim validation

USP (Unique Strengths)

- Event-based patient memory model
- Healthcare + fraud in one unified system
- Low-cost Docker deployment
- Cloud-ready, scalable architecture

Core Capabilities

1. Multilingual Document Processing

- Supports English & Telugu prescriptions, lab reports, bills

2. Clinical Event Extraction

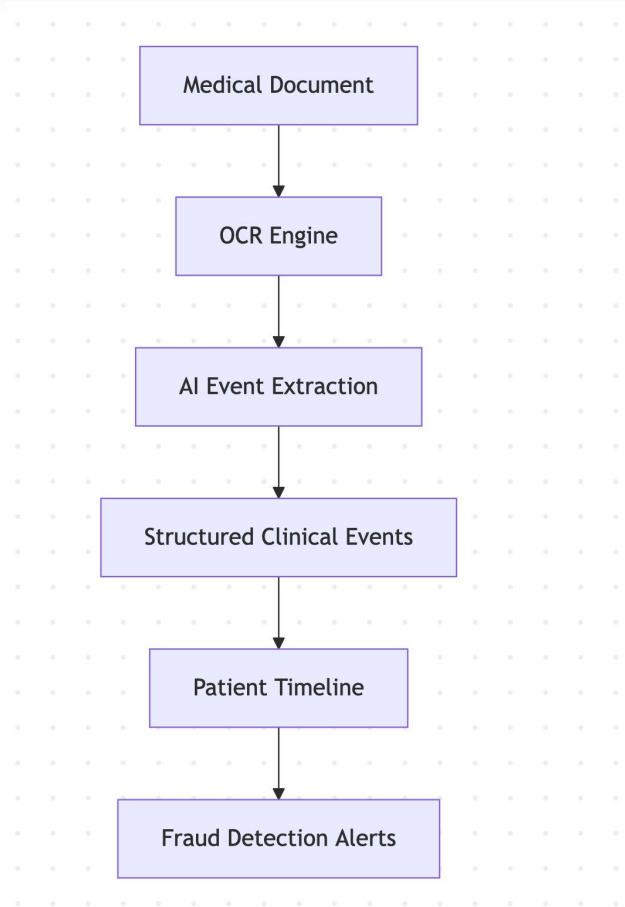
- Diagnosis
- Medication
- Lab Results
- Hospital Visits

3. Automated Patient Timeline

- Chronological health history
- Queryable by condition or date

4. Insurance Fraud Detection

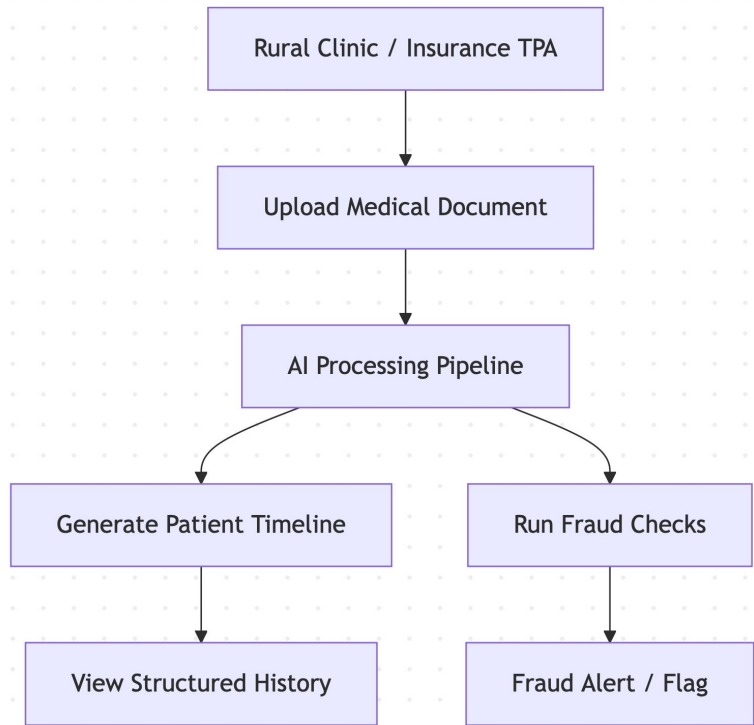
- Duplicate billing detection
- Suspicious claim pattern alerts
- Missing document flags



Process Flow Diagram

End-to-End Workflow:

1. Clinic / TPA uploads medical document
2. OCR extracts multilingual text (English / Telugu)
3. AI engine converts text into structured clinical events
4. Events stored in patient event database
5. Timeline generated automatically
6. Fraud rules executed during claim validation
7. Structured output + alerts displayed






 **ClaimFlow**
Medical Claims AI


Quick Actions


 New Claim

 Filter


-  Claims Processing
-  New Claim


Start Here
-  Processing Queue


3
-  Verification

5
-  Completed

12

 Analytics

 **Dashboard**

 Financial Reports

 Medical Insights

 Management

 All Claims

 Favorites

 Quick Actions

Claim #CF-2024-001

Dashboard Overview

 Quick Actions

 %K




 New Claim

Dashboard

Overview of your medical claims processing


Total Claims

1,247

 +12% from last month **↑ 12%**


Approval Rate

87.3%

 +2.1% from last month **↑ 2.1%**


Avg Processing Time

2.4 min

 -15s from last month **↑ 8%**

Total Savings

\$2.4M

 +18% from last month **↑ 18%**

Processing Status

Upload Queue15%


Processing45%

Verification75%

Completed90%


Recent Claims

CF-2024-001

 John Smith - Appendectomy


\$10,075

CF-2024-002

 Sarah Johnson - MRI Scan

\$2,500



CF-2024-003


 Mike Davis - Surgery


\$15,000

Claim Processing Timeline

0% Complete

  Document Upload **completed**

 View >




System Performance

Real-time processing metrics

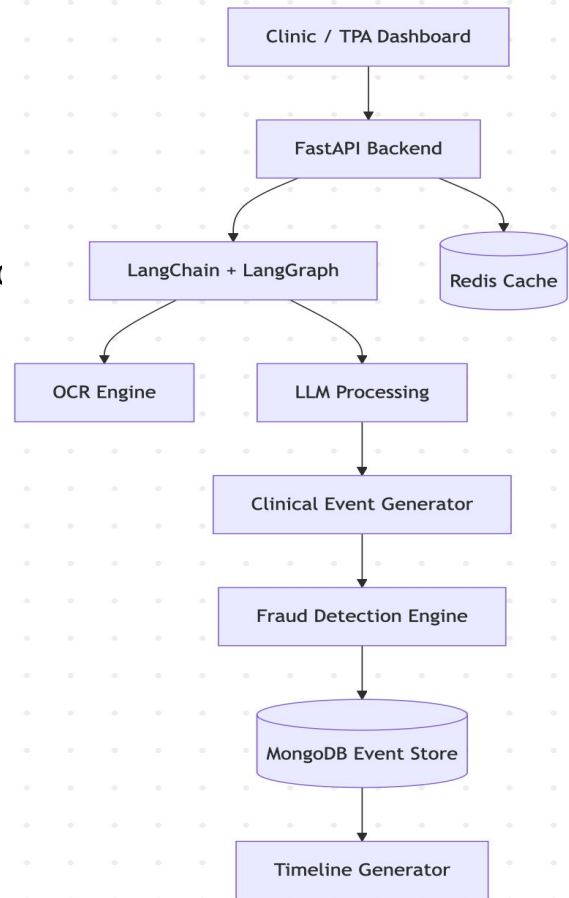
AI Accuracy

98.5%

 Built with V0

High-Level Architecture Overview

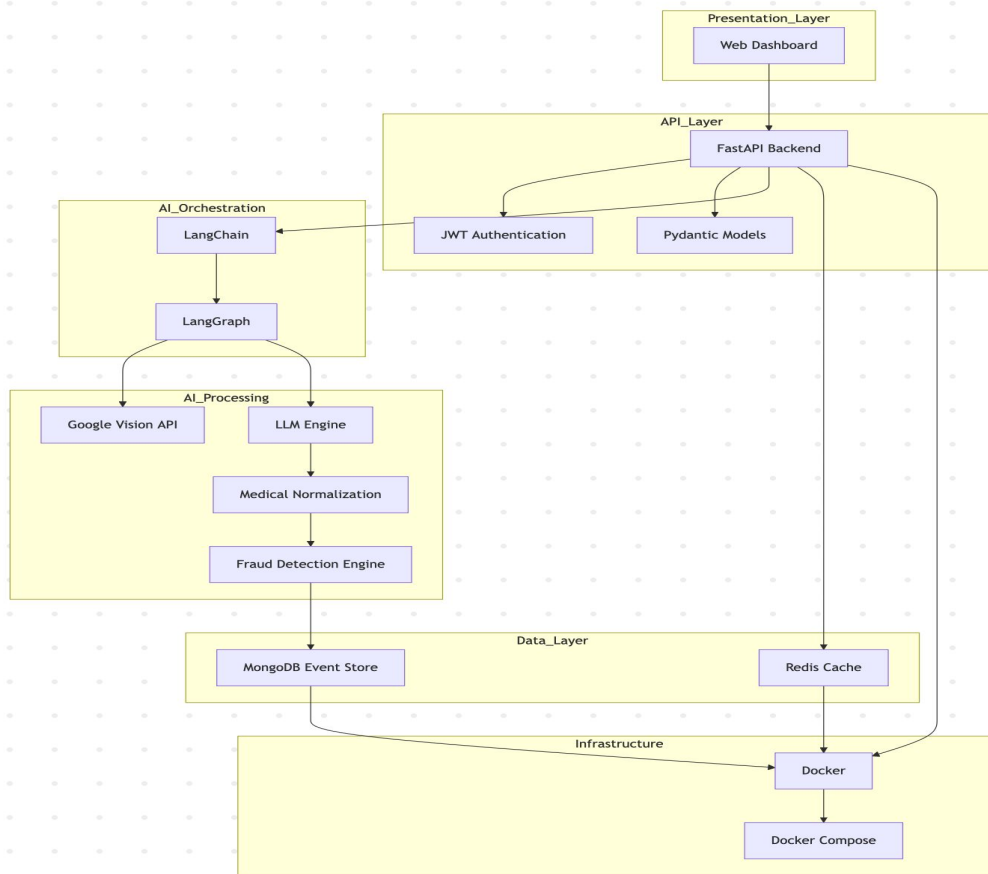
- Frontend Dashboard (Clinic / TPA Interface)
- FastAPI Backend (API Layer)
- LangChain + LangGraph (AI Orchestration)
- OCR Engine (Google Vision – Replaceable with AWS Textract)
- LLM Engine (OpenAI / Bedrock / Local LLM)
- MongoDB (Event Store)
- Redis (Caching / Task Handling)
- Docker Compose Deployment



Technologies Used

Why This Stack?

- Fast to build within 48 hours
- Cost-efficient for MVP
- Modular and cloud-replaceable
- Scalable to AWS when required



Innovation partner **I12S**
HACKATHON

Media partner **YOURSTORY**

AI for Bharat Hackathon

Powered by **aws**

Thank You

