Inteleyzer

Smart AI Agent to optimize the medical data collection tool

Novartis AI Hackathon

Our deployed Link; https://inteleyzer.onrender.com/

Github link

TEAM :

Members:

- **Mehak Dogra (Team Lead)**

mail: mehakdogra2005@gmail.com

- **Khushi Sharma**

mail: khushi.khushisharma1210@gmail.com

- **Suhani Goyal**

mail: goyalsuhani04@gmail.com

1. Product Context & Vision

1.1 Problem Statement

What real-world pain does this solve?

In the pharmaceutical industry, patient safety depends on complete, high-quality adverse event (AE) data. However, in real-world settings, initial AE reports are frequently incomplete, missing critical information such as seriousness criteria, timelines, outcomes, causality indicators, or concomitant medications.
To close these gaps, pharmacovigilance (PV) teams must initiate follow-up requests with patients or healthcare professionals (HCPs). Unfortunately, a large percentage of these follow-ups fail, leading to:

- Delayed or inconclusive safety assessments
- Increased regulatory risk due to incomplete submissions
- Missed early warning signals for patient harm
- Higher operational burden on PV teams

The current follow-up process is manual, repetitive, and human-dependent, resulting in low response rates (often only 30–40% completion). As safety data volumes grow globally, this approach is no longer scalable or reliable.
Core pain:
The industry relies on human goodwill to complete a process that is complex, fragmented, and exhausting for reporters.

Who experiences this problem?

Patients

- Receive multiple, confusing follow-up requests
- Fear scams, data misuse, or privacy violations
- Lack medical knowledge to answer technical questions
- Lose motivation due to repetitive or unclear questionnaires

Healthcare Professionals (Doctors, Nurses, Pharmacists)

- Overburdened with clinical duties
- Limited time to respond to lengthy PV follow-ups
- Repetitive questions across different companies
- Low perceived value of follow-ups compared to patient care

Pharmacovigilance Teams

- Spend significant time chasing responses
- Handle inconsistent or partially filled data
- Face pressure to meet strict regulatory timelines
- Struggle to prioritize high-risk cases effectively

Regulatory Authorities

- Receive aggregate reports with data gaps
- Face difficulty in risk–benefit evaluation
- Must rely on incomplete real-world evidence

1.2 Product Vision

Our Slogan

“Inteleyzer — fewer questions, better answers, safer patients.”

3. User Experience & Functional Flow

3.1 User Journeys

Step-by-step flow per persona

Primary Users

1) **Data Contributors**

Who: Doctors, hospitals, clinics, local pharma companies

Role: Create initial AE reports and respond to follow-ups

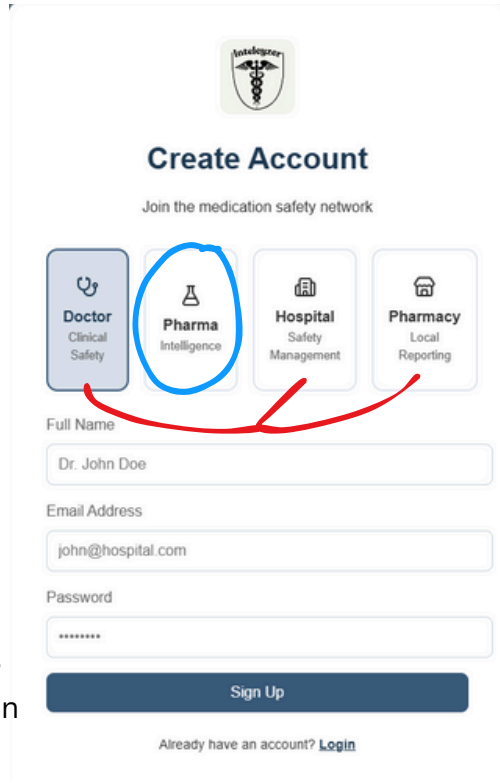
Need: Simple, secure, minimal data entry with clear questions

2) **Safety Analysts & Regulators**

Who: Pharma companies, pharmacovigilance teams, compliance stakeholders

Role: Analyze AE data, ensure PV compliance, identify safety risks

Need: Complete, validated, audit-ready data with risk prioritization



Data Contributors

Doctors

Create Account / Login
Secure sign-up and authentication

Dashboard View
View emergency AE alerts and number of affected patients

Select Report Method
New AE form
Upload Excel
Select existing patient record

Enter Patient & Clinical Details
Patient demographics
Drug(s) prescribed
Symptoms and severity

Patient Consent for Follow-Up?
Yes: Add contact details → follow-up link generated
No: Continue without contact info

Validate & Submit Report
System checks missing data → report submitted

Hospitals / Pharmacy

Create Account / Login
Secure authentication

Dashboard View
View emergency reports and number of adversely affected patients

Select Report Method
New entry, Upload Excel, Select existing patient record

Enter Purchase & Patient Details
Drug purchased / dispensed, Symptoms observed and Basic patient demographics

Consent for Contact Sharing?
Yes: Add email and WhatsApp number
Submit with contact details
No: Submit report in anonymous mode

Data Routing (System Action)
Information sent to Follow-Up Agent and Scoring Agent

Safety Analysts & Regulators

Create Account / Login
Secure regulator credentials created and authenticated

Dashboard View
Overview of: High-risk AE cases, Emergency safety alerts and Affected patient counts

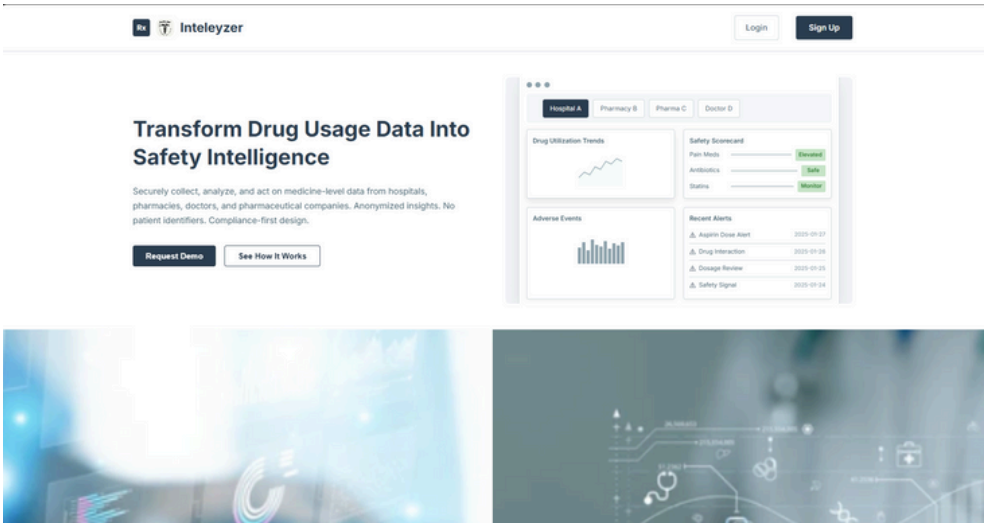
Regulatory Actions
Download high-risk patient / case data
Initiate patient recall
Access AE case details and traceability information

Data Analysis
Analyze AE trends using charts and dashboards
Review signals, severity distribution, and timelines

Admin Controls
User access management
Audit logs and compliance oversight
Basic configuration and permissions

3.2 Screens / Modules Description

Landing Page



Login Page

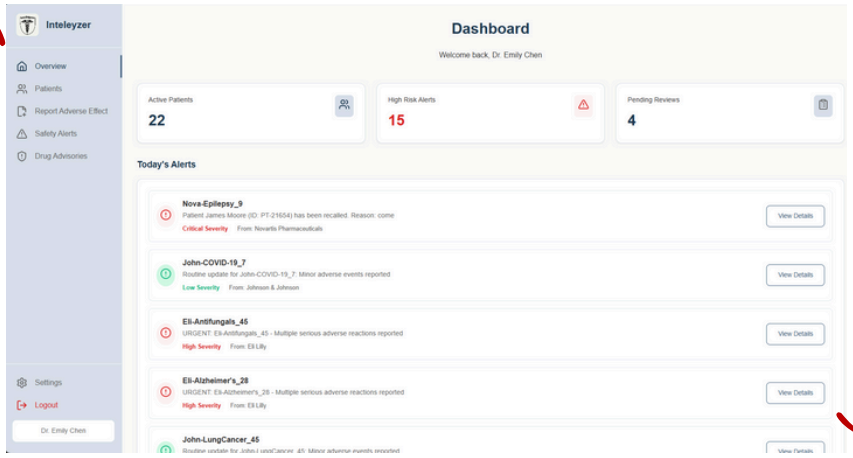
Please Use these dummy Credential

Role	Example Email	Password
Pharmaceutical Companies	admin@novartis.com	admin2024
Doctors	emily.chen@hospital.com	doctor123
Local Pharmacies	downtown@cvs-pharmacy.com	pharmacy123

Doctor's View

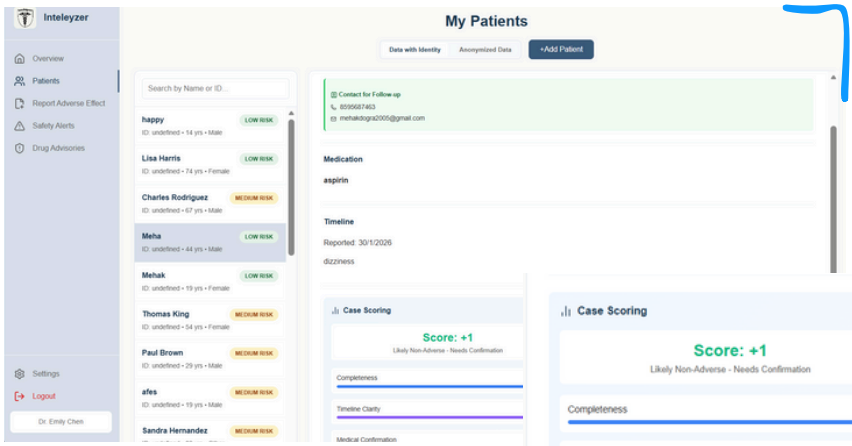
1

Dashboard (Overview Screen)
Provides a real-time summary of active patients, high-risk alerts, and pending reviews. Helps doctors quickly identify urgent adverse event cases requiring immediate attention.

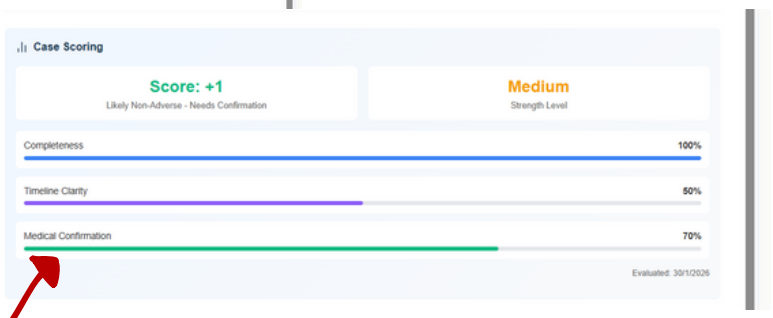


Safety Alerts / Today's Alerts Panel
Displays drug-specific safety alerts issued by pharmaceutical companies, categorized by severity.
Enables rapid review of critical, high, and low-risk adverse event notifications.

2



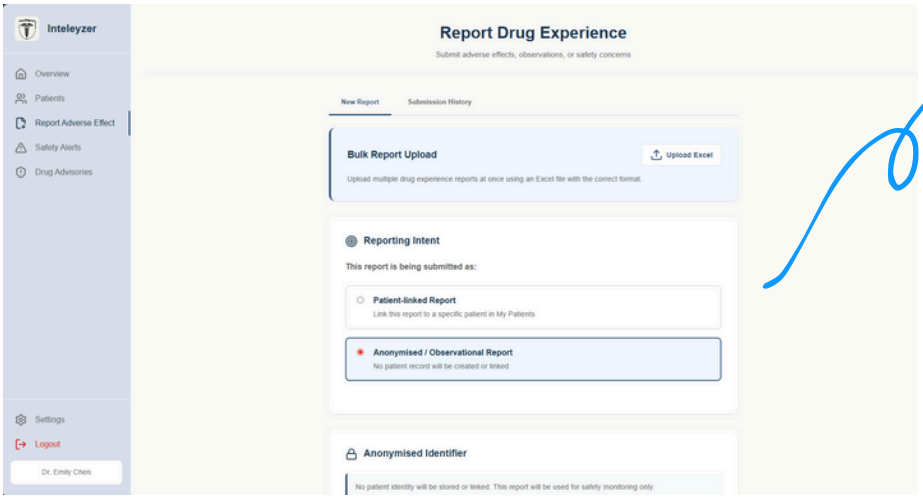
My Patients – Patient List View
Shows all patients linked to the doctor, along with demographic details and risk classification.
Allows quick selection of a patient to review medication history, symptoms, and case status.



Case Scoring Panel
Automatically scores the case based on data completeness, timeline clarity, and medical confirmation.
Score 1 indicates basic completeness, while 2 reflects a fully detailed, high-quality report.

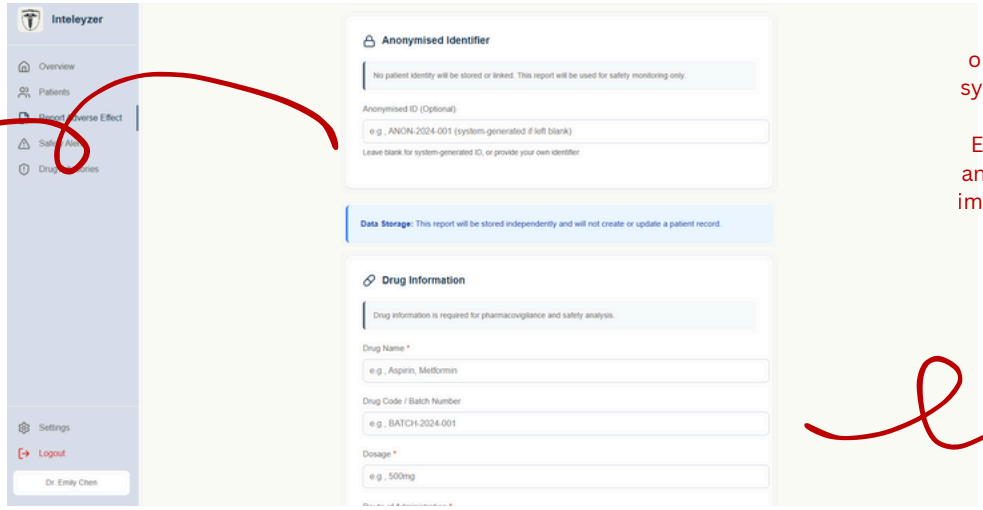
3

Drug Information Section
Captures essential drug details such as name, dosage, batch number, and route of administration.
Forms the foundation for pharmacovigilance analysis and causality assessment.



Reporting Intent Selection
Allows the doctor to choose between a patient-linked or anonymized observational report.
Ensures patient privacy while maintaining regulatory-compliant safety data collection.

4

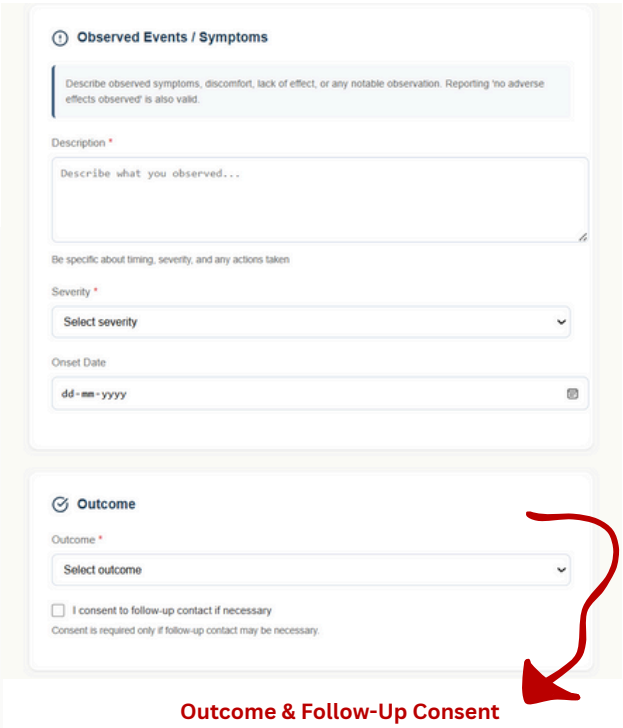


Anonymized Identifier Section
Generates or accepts an anonymized ID when patient identity is not shared.
Ensures the report is usable for safety monitoring without storing personal identifiers.

Observed Events / Symptoms Section
Records clinical observations including symptoms, severity, and onset date.
Encourages structured and precise reporting to improve signal detection accuracy.

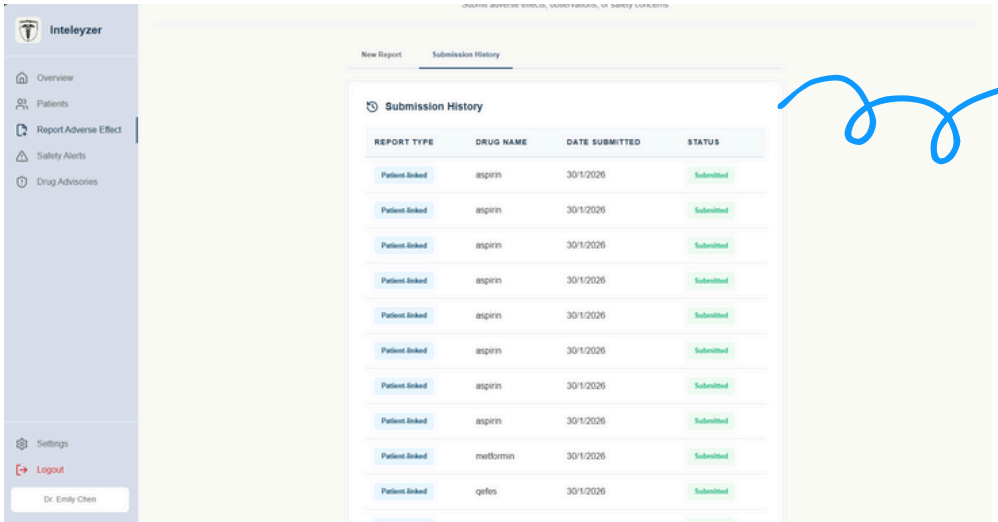
5

Bulk Report Upload (Excel Upload)
Enables doctors to submit multiple adverse event reports simultaneously using a standardized Excel format.
Reduces manual effort and accelerates reporting for high-volume cases.



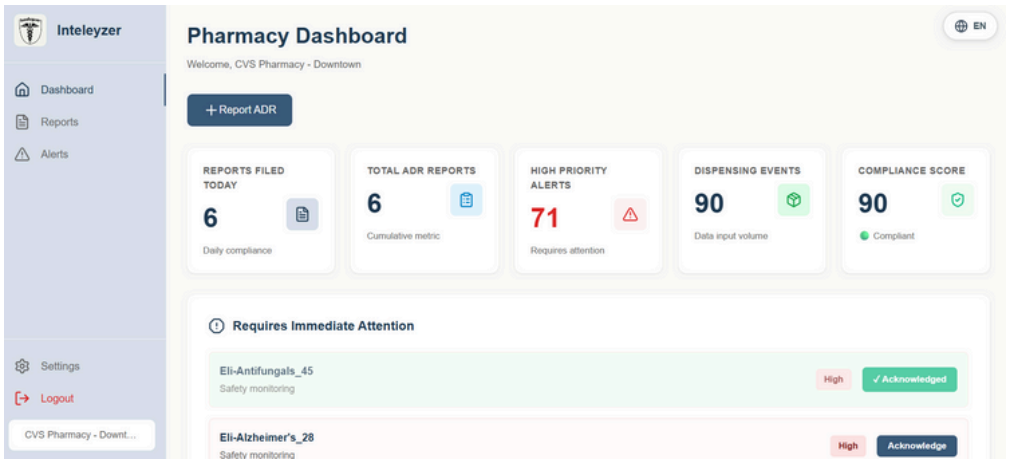
Outcome & Follow-Up Consent
Documents patient outcomes and consent for future follow-up if required.
Supports ethical engagement and enables automated follow-up workflows when permitted.

6



Submission History
Lists all previously submitted adverse event reports with drug name, date, and submission status.
Allows doctors to track reporting activity and verify successful regulatory submissions.

Pharmasisit's View

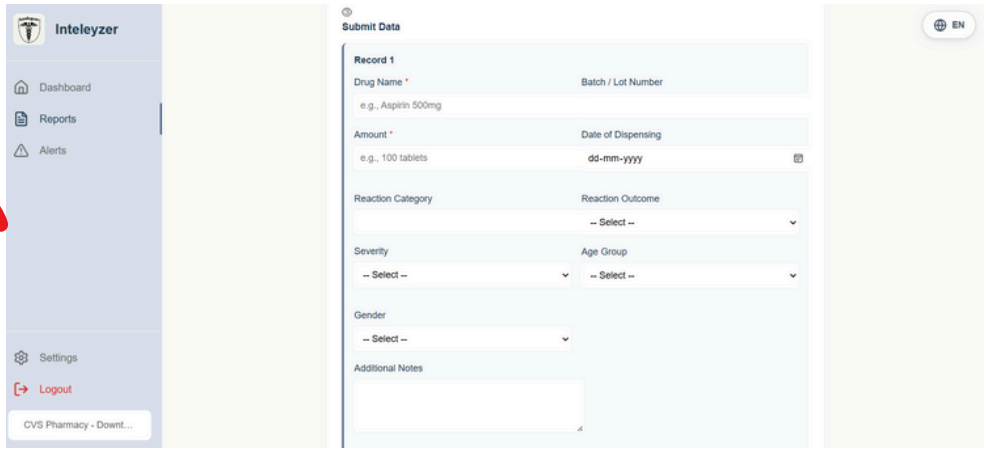


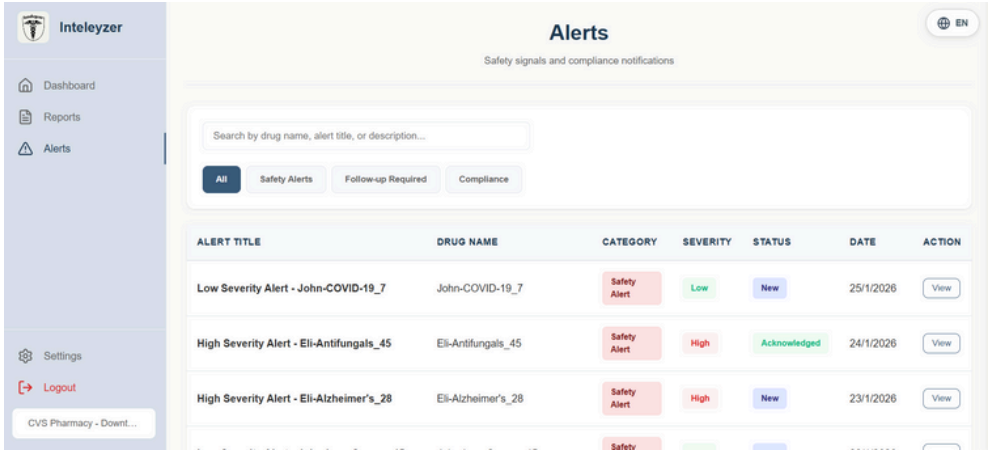
Pharmacy Dashboard (Overview)
"A real-time operational dashboard for local pharmacies to track ADR submissions, dispensing activity, compliance status, and high-priority safety alerts."

Why it matters:
Pharmacies get instant visibility into their reporting performance while being alerted to drugs requiring immediate attention.

Pharmacy Data Submission (ADR Reporting Form)
"A structured data entry form that allows pharmacies to report drug dispensing details and observed adverse reactions, including cases not captured during doctor visits."

Why it matters:
This enables pharmacies to contribute real-world safety data for walk-in patients or over-the-counter purchases, ensuring no safety signal is missed.



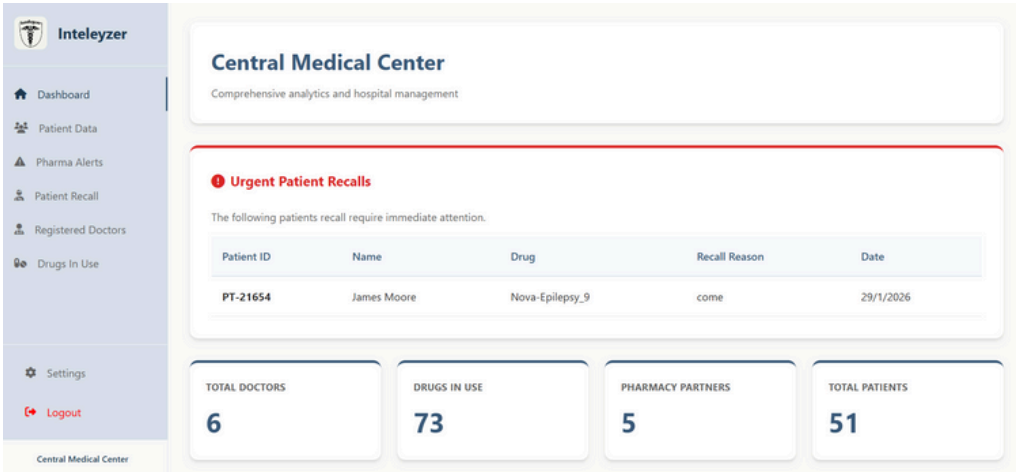


Alerts – Pharmacy View
"A centralized alert system displaying safety warnings, follow-up requirements, and compliance notifications relevant to drugs dispensed by the pharmacy."

Why it matters:
This ensures pharmacies stay informed about evolving drug risks and can quickly acknowledge or act on critical safety updates.

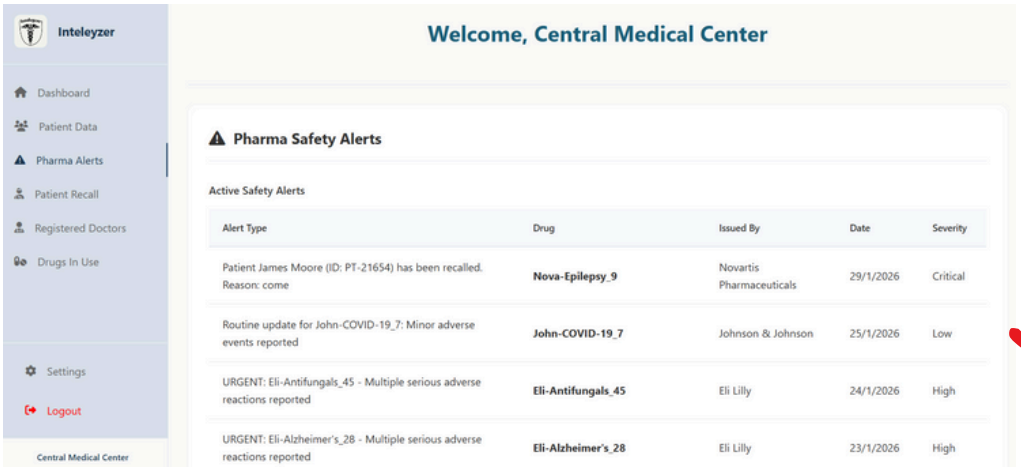
Hospital's View

Hospital Dashboard (Central Medical Center – Overview)
"A real-time hospital safety dashboard highlighting urgent patient recalls, key metrics, and overall pharmacovigilance status at a glance."

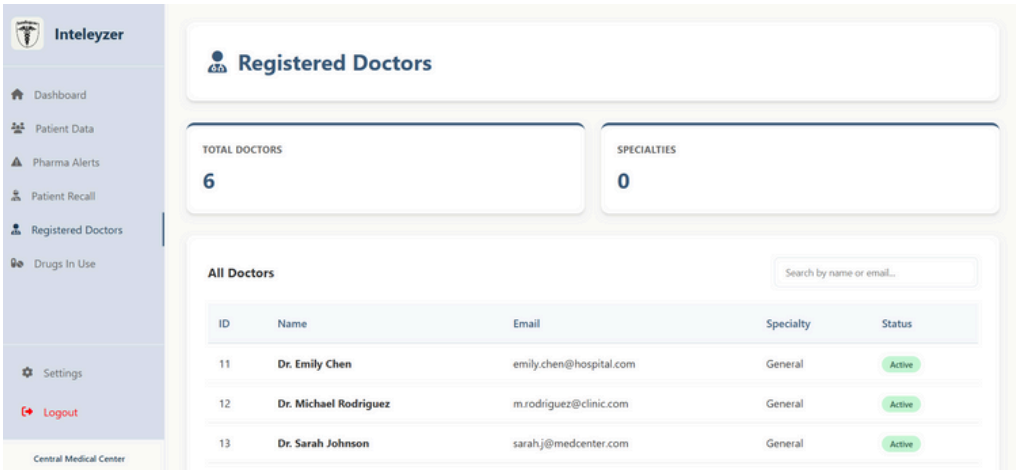


Pharma Safety Alerts
"A live alert feed displaying critical, high, and routine drug safety notifications issued by pharmaceutical companies and regulators.

Patient Data – With Identity
"Centralized patient registry enabling hospitals to securely manage identifiable patient data, reported symptoms, and associated drug risk levels."



Registered Doctors
"A consolidated view of all registered doctors, their specialties, and activity status to ensure accountable and traceable clinical reporting."



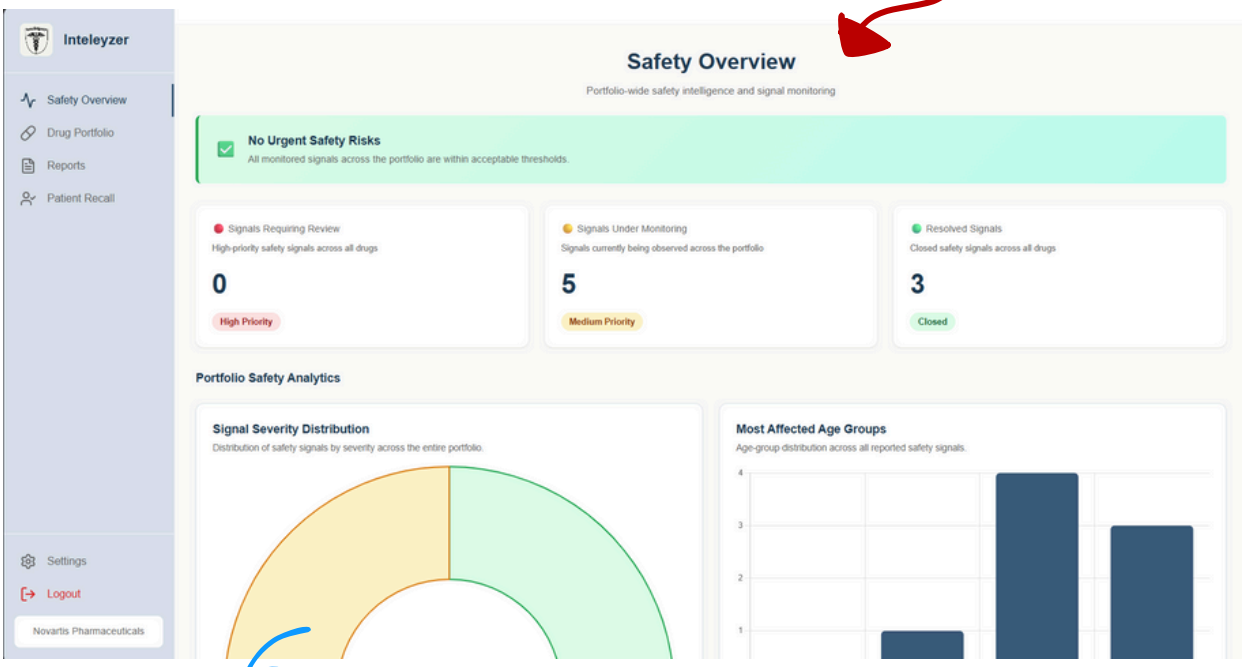
Drugs in Use
"An active drug portfolio view showing all medicines currently prescribed within the hospital, mapped to manufacturers and usage status

Pharmasutical Comapniies's View

1

Safety Overview Dashboard

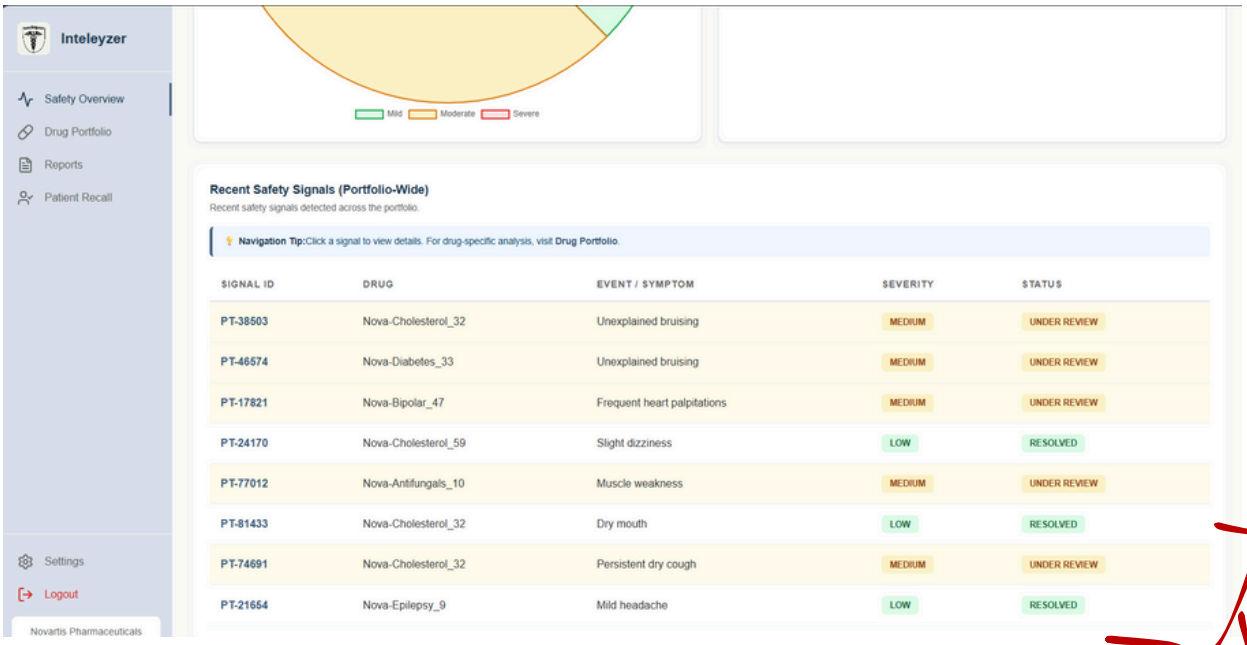
Provides a portfolio-wide snapshot of current safety status across all monitored drugs. Highlights urgent risks, signals under review, and resolved cases to support rapid safety decision-making.



Portfolio Safety Analytics (Charts & Trends)

Visualizes safety signals by severity, age group, and frequency using interactive charts. Helps hospitals and pharma teams identify emerging patterns and high-impact risk clusters.

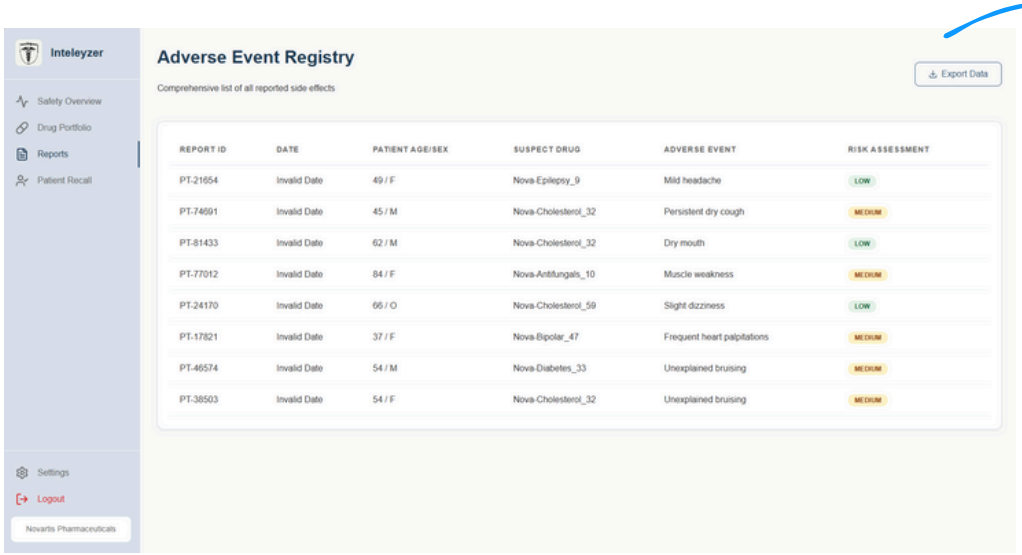
2



Recent Safety Signals Table

Displays the latest detected safety signals with drug name, symptom, severity, and review status. Enables quick drill-down into specific cases for further investigation or escalation.

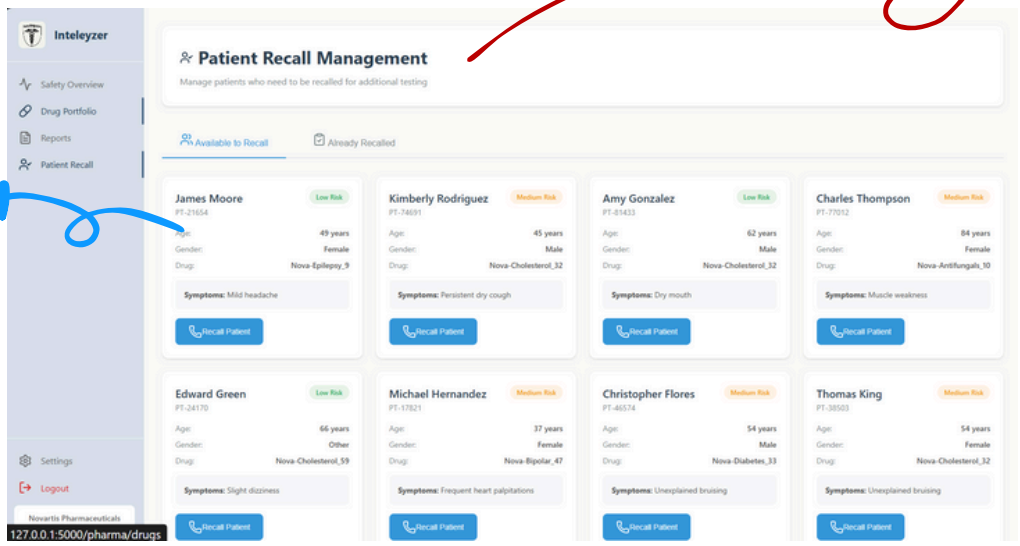
3



Adverse Event Registry
Acts as a centralized log of all reported adverse events across drugs and patients.
Supports audit readiness, traceability, and export for regulatory submissions or internal analysis.

4

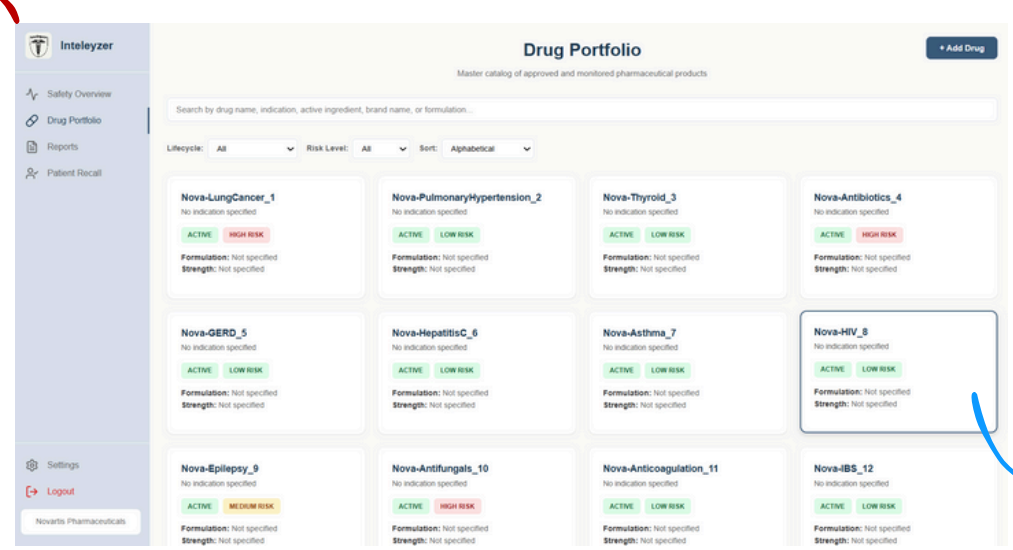
Follow-Up & Recall Enablement
Enables hospitals and pharma teams to send follow-up requests or trial re-invitations to high-risk patients.
Ensures targeted engagement while maintaining consent and regulatory compliance.



Patient Recall Management
Lists patients eligible for recall based on high or medium-risk adverse event signals.
Allows hospitals to initiate recalls or follow-ups for further testing, monitoring, or re-evaluation.

5

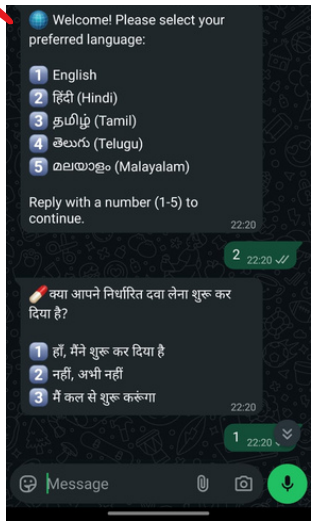
Drug Portfolio – Risk & Performance Insight
Allows users to compare drugs based on safety performance and reported adverse effects.
Supports informed decisions on monitoring intensity, labeling updates, or further investigation.

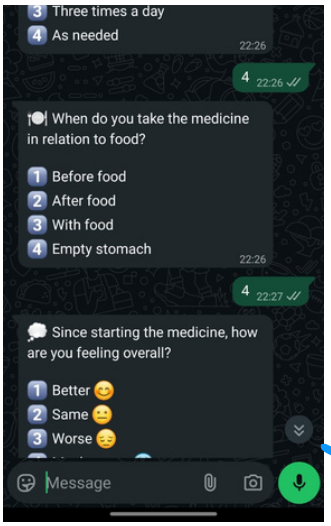


Drug Portfolio Overview
Displays the complete catalog of monitored drugs along with their current risk classification.
Risk levels are dynamically derived from aggregated adverse event reporting and case severity.

Patient's view of the whatsapp agent

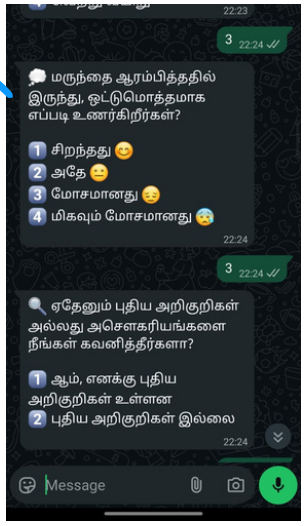
Adaptive Logic Based on Patient Responses
The conversation flow adapts based on answers:
If symptoms worsen → severity-focused questions
If medication not started → exposure clarification
If outcome unclear → timeline refinement
This ensures medically meaningful data collection.

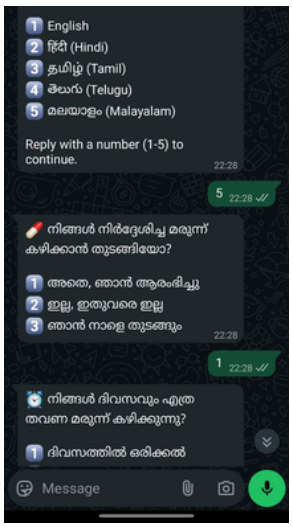


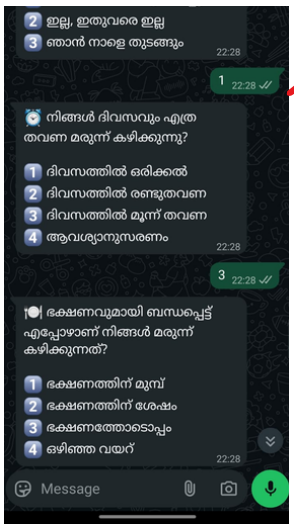


Personalized, Step-by-Step Questionnaires
The agent asks context-aware questions based on:
Drug reported
Patient demographics
Previously submitted information
This prevents repetitive or irrelevant questions and reduces patient fatigue.

Multilingual & Demographic-Aware Interaction
The agent begins by identifying the patient's preferred language (e.g., English, Hindi, Tamil, Telugu, Malayalam). Questions are dynamically localized to improve understanding, trust, and response accuracy.

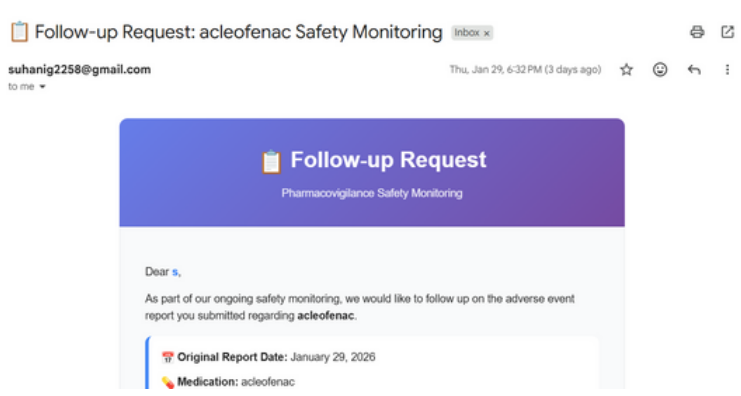


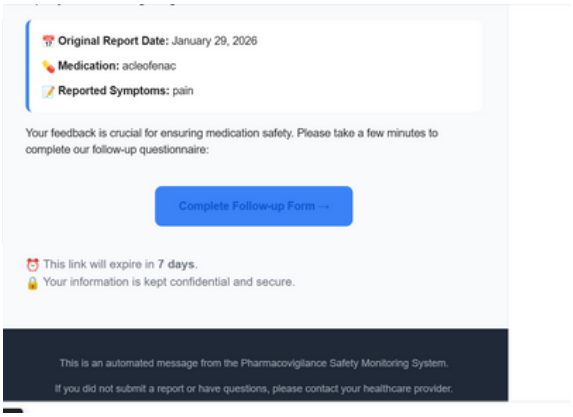




Completeness Scoring & Trigger Logic
If the case completeness score falls below the required threshold, the AI agent is automatically triggered.
Once sufficient information is collected, the case is re-scored and routed back to the safety system.

Patient's view of the Email Form





Step 1: Extract the Zip File

- Extract the zip to a folder like `C:\Projects\Inteleyzer`
- Open that folder

Step 2: Install Required Packages

```bash
pip install -r requirements.txt
```

Wait for all packages to install (takes 2-3 minutes)

Step 3: Create .env File

Create a file named .env (no extension) in the project root with:

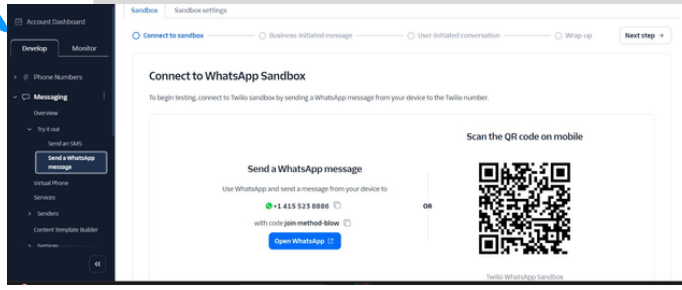
`

Where to get these:

- Gmail: Google Account → Security → App Passwords
- Twilio: console.twilio.com
- Google API: console.cloud.google.com
- ngrok: ngrok.com/signup

Step 4: Connect to Sandbox

`



Step 5: Connect to Engirok

` Currently, our WhatsApp chatbot works only for users joined to the Twilio Sandbox. Since Twilio cannot directly send incoming WhatsApp messages to a local system, we expose our localhost using ngrok, map the generated public URL to the Twilio Sandbox webhook, and receive messages via a POST API. This enables the agent to automatically trigger and respond to patients present in the Twilio Sandbox when they are added to the database.

Step 6: Run the Application

```bash
python app.py
```

Step 7: Open in Browser

- Go to: `http://127.0.0.1:5000`
- Login with:

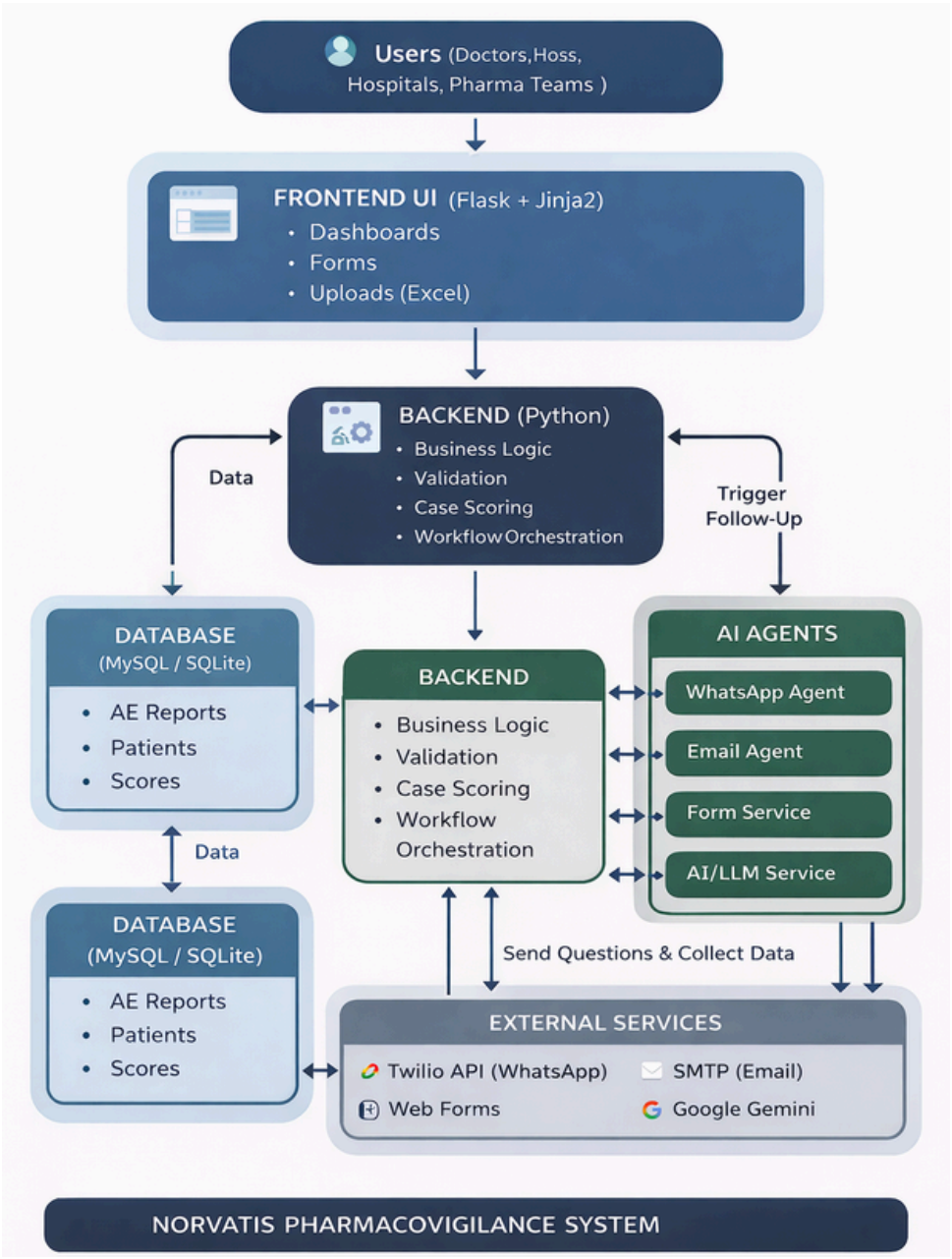
Role	Example Email	Password
Pharmaceutical Companies	admin@novartis.com	admin2024
Doctors	emily.chen@hospital.com	doctor123
Local Pharmacies	downtown@cvs-pharmacy.com	pharmacy123

What We'll Get:

- ✅ Full website with all features
- ✅ Translation system (30 languages)
- ✅ Working flowcharts
- ✅ All dashboards (doctor/pharma/pharmacy/hospital)
- ✅ Database with 43 test users
- ✅ WhatsApp integration (if we set up Twilio)

5. Technical Architecture

5.1 Architecture Diagram



Architecture Explanation

1) Frontend Layer (Flask + Jinja2): Renders HTML templates for different user roles (doctors, hospitals, pharma companies, pharmacies). Each role has dedicated dashboard views with role-specific functionality.

2) Backend Layer (Python Flask): The core application logic resides here, handling HTTP requests, business logic, authentication, and orchestrating communication between different components.

3) Database Layer (SQLite/MySQL): Stores all persistent data including users, patients, visits, prescriptions, follow-ups, and adverse event reports. SQLAlchemy ORM provides database abstraction.

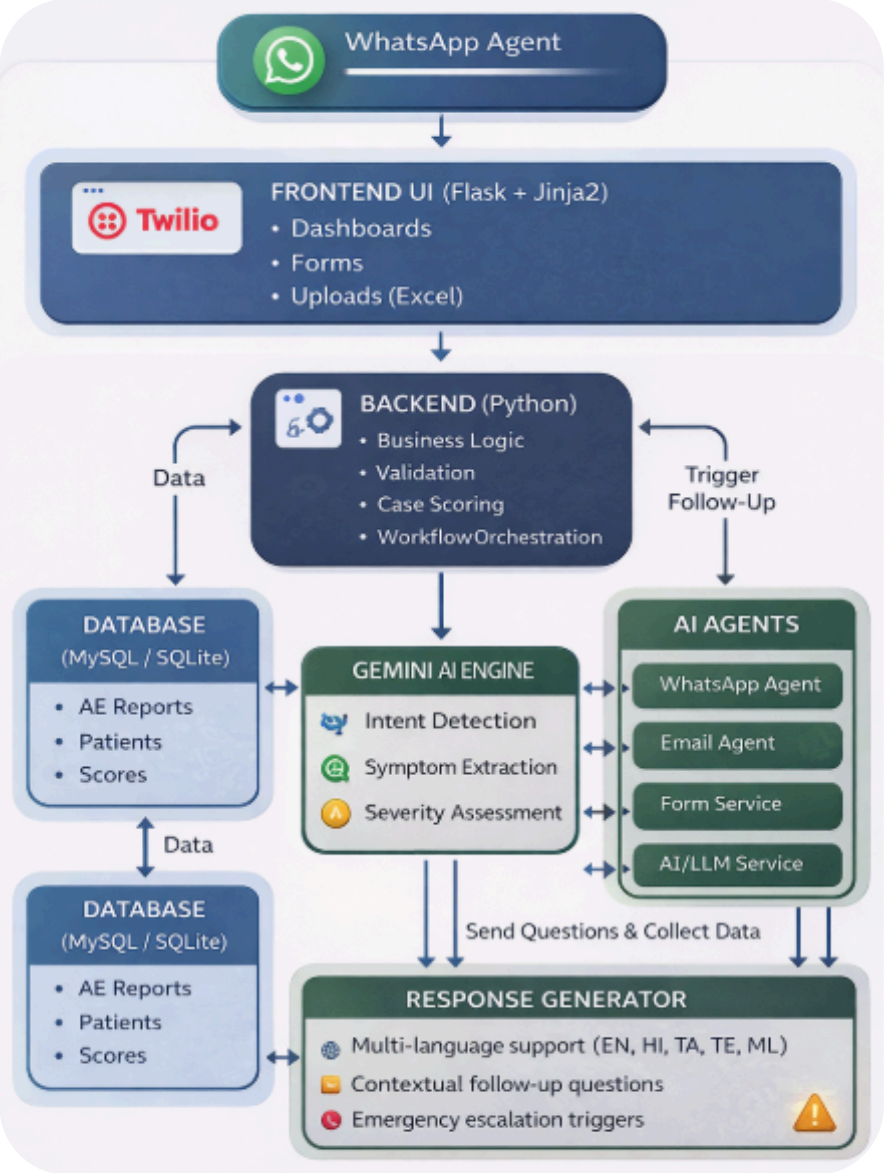
4) External Services: Third-party integrations including:

- Twilio API: Powers WhatsApp Business messaging for patient communication
- Gmail SMTP: Sends email notifications with secure form links
- Google Gemini: Provides AI/LLM capabilities for natural language processing and symptom analysis

5) AI Agents: Four specialized agents handle different aspects of patient communication and analysis, working together to provide comprehensive pharmacovigilance coverage.

5.2 AI Agent Architecture

1, Whatsapp Agent



The WhatsApp Communication Agent is the primary patient interaction channel, leveraging the familiarity and accessibility of WhatsApp for healthcare communication.

Twilio API Integration: The system uses Twilio's WhatsApp Business API to send and receive messages. Twilio provides enterprise-grade reliability, message delivery tracking, and compliance with healthcare communication standards. Messages are sent from a verified business number, ensuring patients trust the communication.

Webhook Handler: When a patient responds to a WhatsApp message, Twilio sends a webhook request to our server. The webhook handler parses the incoming message, extracts the sender's phone number, message content, and any media attachments (like voice notes). This handler runs asynchronously to ensure quick response times.

Question Handler: Manages the conversation flow, tracking which questions have been asked and answered. It maintains a state machine for each conversation.

Agent Router: The router determines how to process the incoming message based on:

- Conversation State: Is this a new conversation or continuation?
- Message Type: Text, voice note, or button response?
- Patient Context: Which visit/prescription is this related to?
- Language Detection: The system supports 5 Indian languages (English, Hindi, Tamil, Telugu, Malayalam). The language detector uses a combination of: Character set analysis (different scripts for each language), Previous language preference stored in patient profile and Explicit language selection by the patient

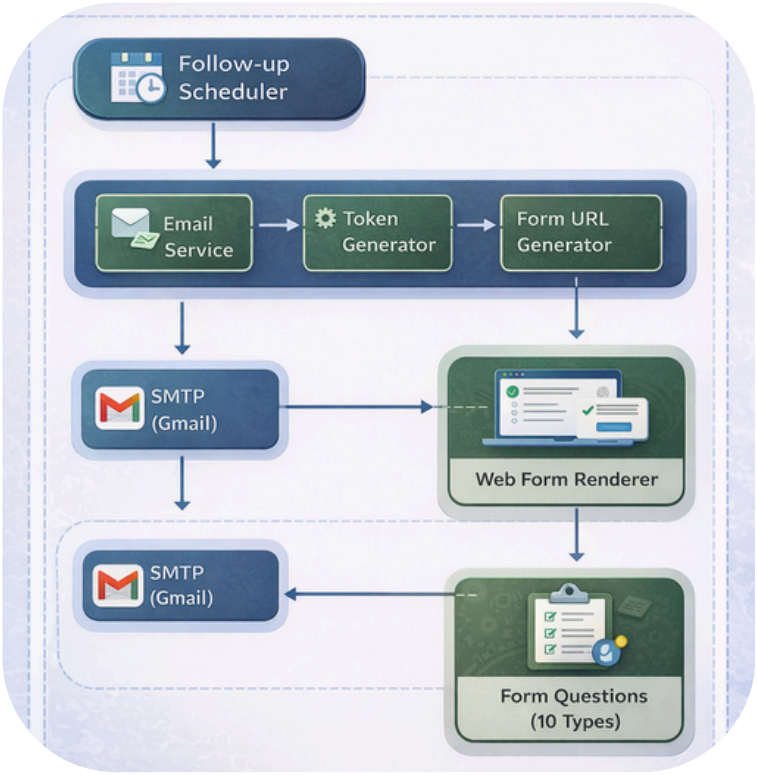
Response Parser: Extracts structured data from free-text patient responses. For example:
"I take it twice daily with food" → {frequency: 'twice', food_relation: 'with'}
"मुझे सिरदर्द हो रहा है" → {symptom: 'headache', language: 'hi'}

Gemini AI Engine: The core intelligence of the WhatsApp agent with three key functions:

- Intent Detection: Understands what the patient is trying to communicate
- Symptom Extraction: Identifies medical symptoms from natural language
- Severity Assessment: Rates the urgency of reported symptoms (1-10 scale)

Response Generator: Crafts contextually appropriate responses in the patient's preferred language. It considers: Medical appropriateness of responses, Empathetic tone for distressing symptoms, Clear escalation instructions for severe cases and Quick reply buttons for easy interaction

2. Email & Form Service Agent



Email/Form Agent Explanation
The Email & Form Service Agent provides a fallback communication channel and structured data collection mechanism for patients who prefer forms over chat.

Follow-up Scheduler:
A background process that runs periodically (every hour) checking for:
- Visits that need Day 3 follow-up
- Visits that need Day 7 follow-up
- Visits that need Day 14 follow-up

Unanswered previous follow-ups requiring reminders

Email Service (SMTP): Sends professionally formatted HTML emails through Gmail's SMTP server. Each email includes:
Patient's name and prescribed medication Clear call-to-action button for the form Doctor's contact information for emergencies
Multi-language greeting based on patient preference

Token Generator: Creates cryptographically secure, unique tokens for each form:

Token Security Features: 32-byte URL-safe token (practically impossible to guess)
- Single-use: token is invalidated after form submission
- Expiration: tokens expire after 7 days
- Association: each token is tied to specific visit/patient

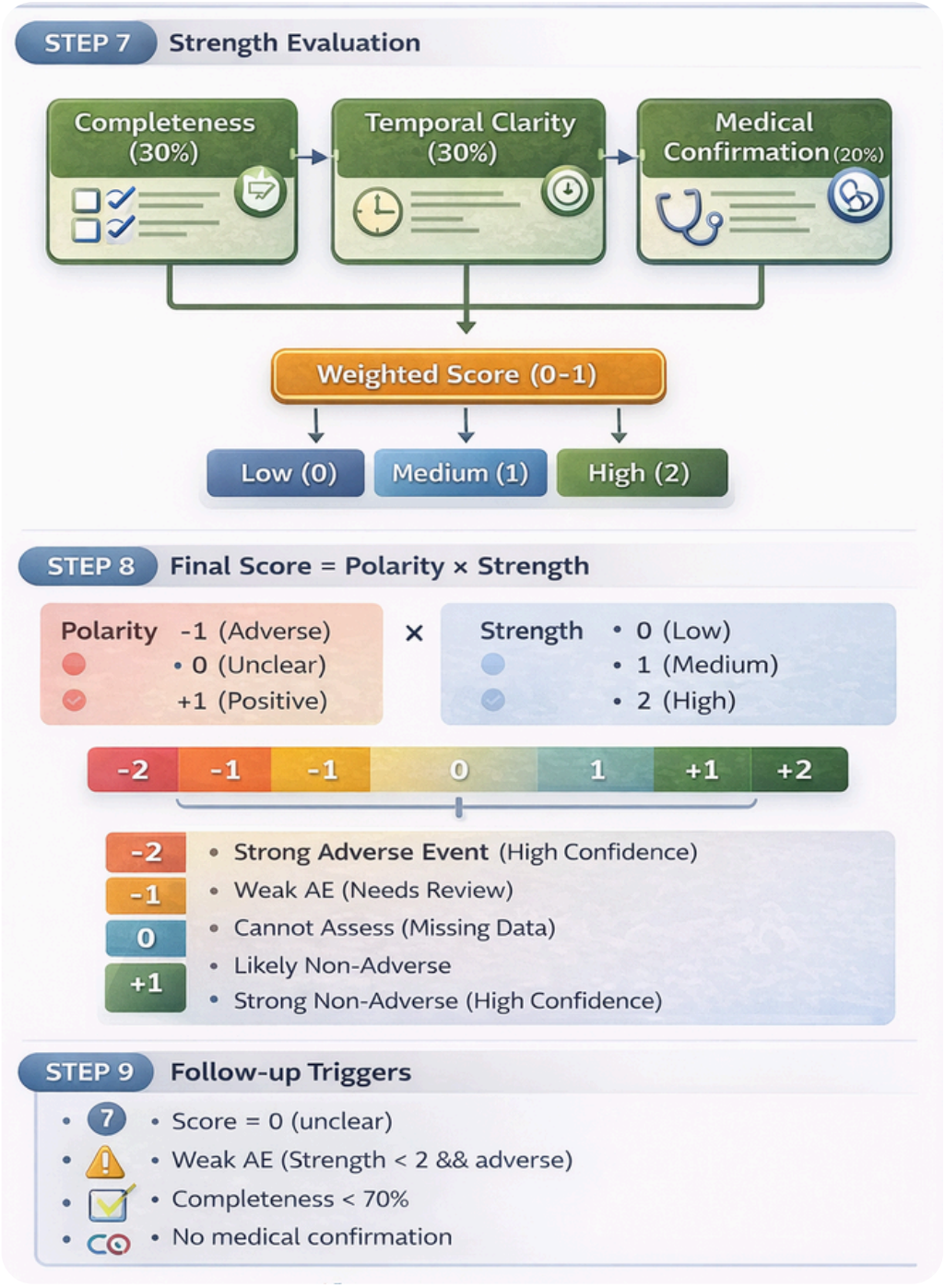
Form URL Generator: Constructs complete form URLs with embedded parameters:

Web Form Renderer: Dynamically generates HTML forms based on:
Patient's language preference
Question conditional logic (e.g., show symptom details only if "new symptoms" = Yes)
Previous responses (for clarification forms)

5.3 Bussiness Logic

Case Scoring Service

Purpose: Quality assessment & strength evaluation of adverse event cases



Function	Purpose
evaluate_case_strength()	Calculate strength (0-2) based on 4 factors
calculate_final_score()	Final Score = Polarity × Strength
check_followup_triggers()	Determine if follow-up needed

2. Follow-up Agent (followup_agent.py)

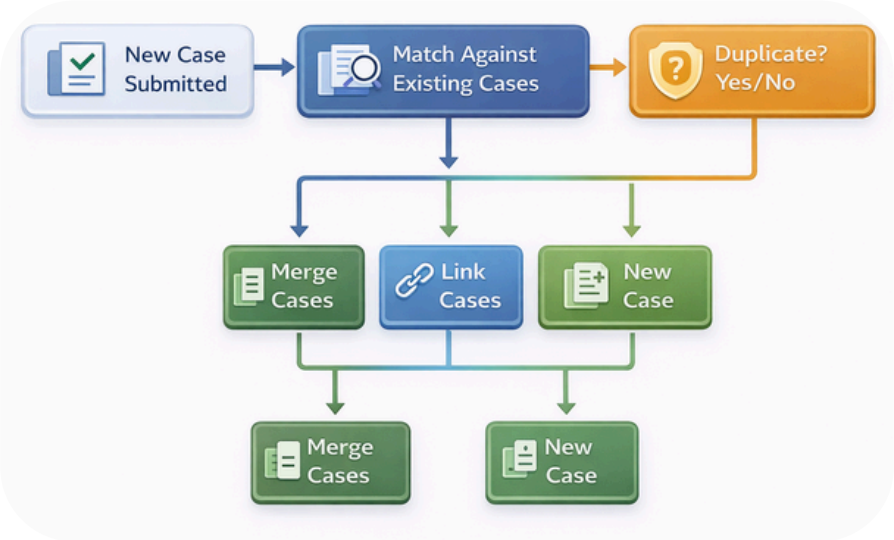
Purpose: AI-powered follow-up conversation management

Likely Functions:

- **generate_followup_questions()** - Create targeted questions
- **process_patient_response()** - Parse and validate answers
- **determine_next_action()** - Decide if more follow-up needed

3. Case Matching Service (case_matching.py)

Purpose: Match/deduplicate patient cases to avoid duplicates

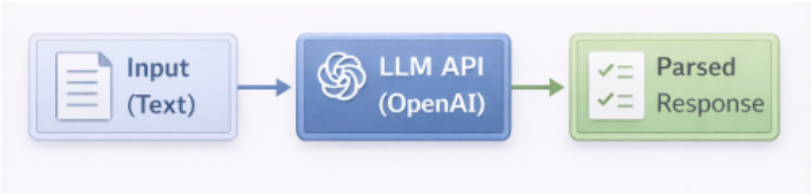


Likely Functions:

- **find_matching_cases()** - Search for similar cases
- **calculate_similarity_score()** - Compare case attributes
- **merge_cases()** - Combine duplicate cases

4.LLM Service (llm_service.py)

Purpose: Interface with Large Language Models (GPT/Claude)

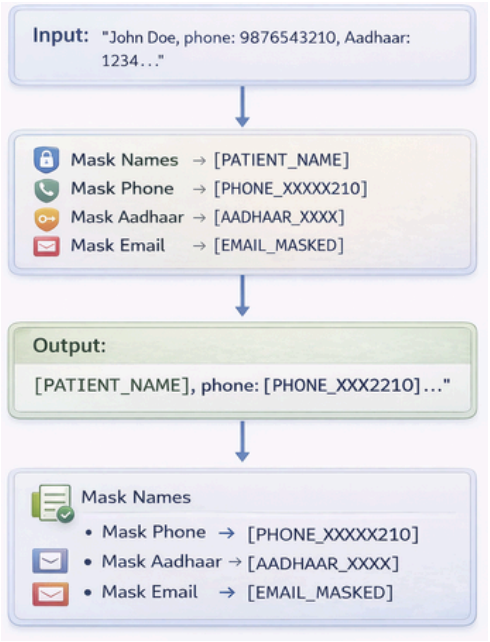


Likely Functions:

- **analyze_symptoms()** - Extract symptoms from text
- **generate_response()** - Create patient-friendly messages
- **classify_severity()** - AI-based severity assessment

Privacy Utils (privacy_utils.py)

Purpose: PII masking and data protection



Likely Functions:

- **mask_pii()** - Remove/mask personal information
- **anonymize_case()** - Create anonymized case copy
- **detect_pii()** - Find PII in text

5. Quality Agent (quality_agent.py)

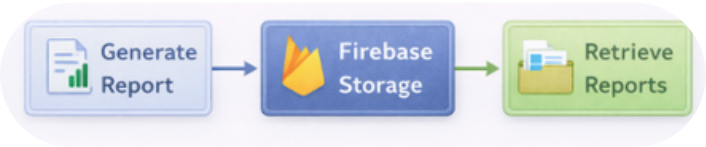
Purpose: Automated quality assessment of cases

Likely Functions:

- **assess_quality()** - Overall quality evaluation
- **check_data_consistency()** - Validate data coherence
- **generate_quality_report()** - Create quality summary

Firebase Reports Service (firebase_reports_service.py)

Purpose: Store and retrieve reports from Firebase

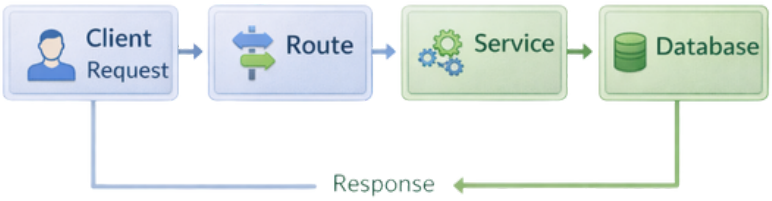


Likely Functions:

- **save_report() -** Upload report to Firebase
- **get_report() -** Retrieve report by ID
- **list_reports() -** List all reports for a patient/visit

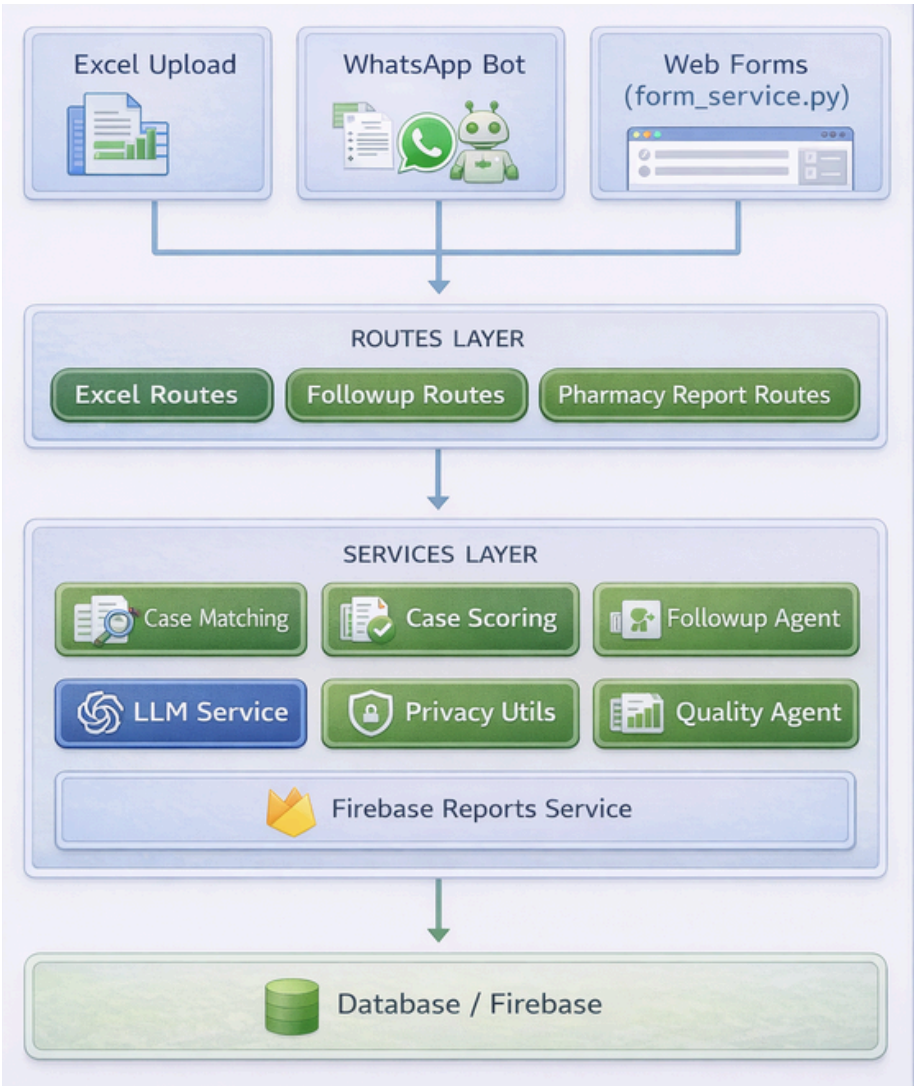
ROUTES (API Endpoints)

Routes are API endpoints that define URLs your application responds to. They connect incoming HTTP requests to the appropriate service/function.



File	Purpose	Likely Endpoints
excel_routes.py	Excel file upload/download	/api/excel/upload, /api/excel/download
followup_routes.py	Patient follow-up management	/api/followup/send, /api/followup/status
pharmacy_report_routes.py	Pharmacy reporting	/api/reports/pharmacy, /api/reports/generate

How the Routes and Services works together



5.4 Norvatis Repository Structure

High-Level Folder Overview

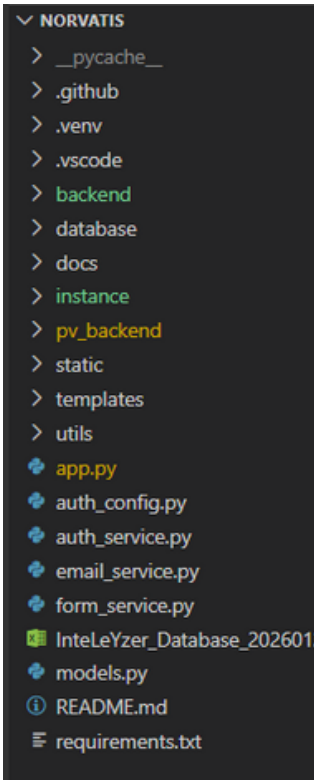
```
Norvatis/
├── 📄 Root Files          → Main app entry points & configs
├── 📁 pv_backend/         → Core Pharmacovigilance Backend (API + Services)
├── 📁 backend/            → Legacy/Secondary Backend
├── 📁 templates/          → HTML Frontend (Jinja2 templates)
├── 📁 static/             → CSS, JavaScript, Images
├── 📁 database/           → Database Schema
├── 📁 docs/               → Documentation
├── 📁 utils/              → Utility Scripts
└── 📁 instance/           → SQLite Database Files
```

Folder	Purpose	Key Contents
/ (root)	Entry points & configuration	app.py, models.py, auth_service.py
pv_backend	Main PV business logic	Routes, Services, Models
templates	User interfaces	Doctor, Hospital, Pharma, Pharmacy dashboards
static	Frontend assets	CSS styles, JavaScript logic, images
database	DB definitions	SQL schema
docs	Project documentation	Feature guides, architecture docs
utils	Helper scripts	Data population scripts

Detailed Structure

📁 Root Directory (/)

File	Purpose
app.py	Main Flask application entry point
models.py	SQLAlchemy database models
auth_service.py	Authentication logic (JWT)
auth_config.py	Auth configuration settings
form_service.py	Patient follow-up form handling
email_service.py	Email notification service
requirements.txt	Python dependencies
.env	Environment variables



Detailed view of each folder

1

pv_backend - Core Pharmacovigilance Backend

```
pv_backend/
├── app.py                    → Backend Flask app
├── routes/                   → API Endpoints
│   ├── excel_routes.py       → Excel import/export APIs
│   ├── followup_routes.py    → Patient follow-up APIs
│   └── pharmacy_report_routes.py → Pharmacy reporting APIs
├── services/                 → Business Logic
│   ├── case_matching.py      → Duplicate case detection
│   ├── case_scoring.py       → Case quality scoring
│   ├── followup_agent.py     → AI follow-up management
│   ├── llm_service.py        → LLM integration (GPT/Claude)
│   ├── quality_agent.py      → Quality assessment
│   ├── privacy_utils.py      → PII masking
│   ├── whatsapp_chatbot.py   → WhatsApp integration
│   └── firebase_reports_service.py → Firebase storage
└── models/                   → Data Models
```

2

templates - Frontend HTML

```
templates/
├── index.html                → Landing page
├── login.html                → Login page
├── signup.html               → Registration page
├── doctor/                   → Doctor Dashboard
│   ├── dashboard.html        → Main dashboard
│   ├── patients.html         → Patient list
│   ├── alerts.html           → Alert notifications
│   ├── warnings.html         → Warning notifications
│   └── report.html           → Report view
├── hospital/                 → Hospital Dashboard
│   ├── dashboard.html        → Hospital overview
│   ├── doctors.html          → Doctor management
│   ├── drugs.html            → Drug inventory
│   ├── patient-data.html     → Patient data query
│   └── pharma-alerts.html    → Pharma alerts
├── pharma/                   → Pharma Company Dashboard
│   ├── dashboard.html        → Pharma overview
│   ├── analysis.html         → Data analysis
│   ├── drugs.html            → Drug management
│   └── reports.html          → Reports view
├── pharmacy/                 → Pharmacy Dashboard
│   ├── dashboard.html        → Pharmacy overview
│   ├── alerts.html           → Alert management
│   └── reports.html          → Report generation
```

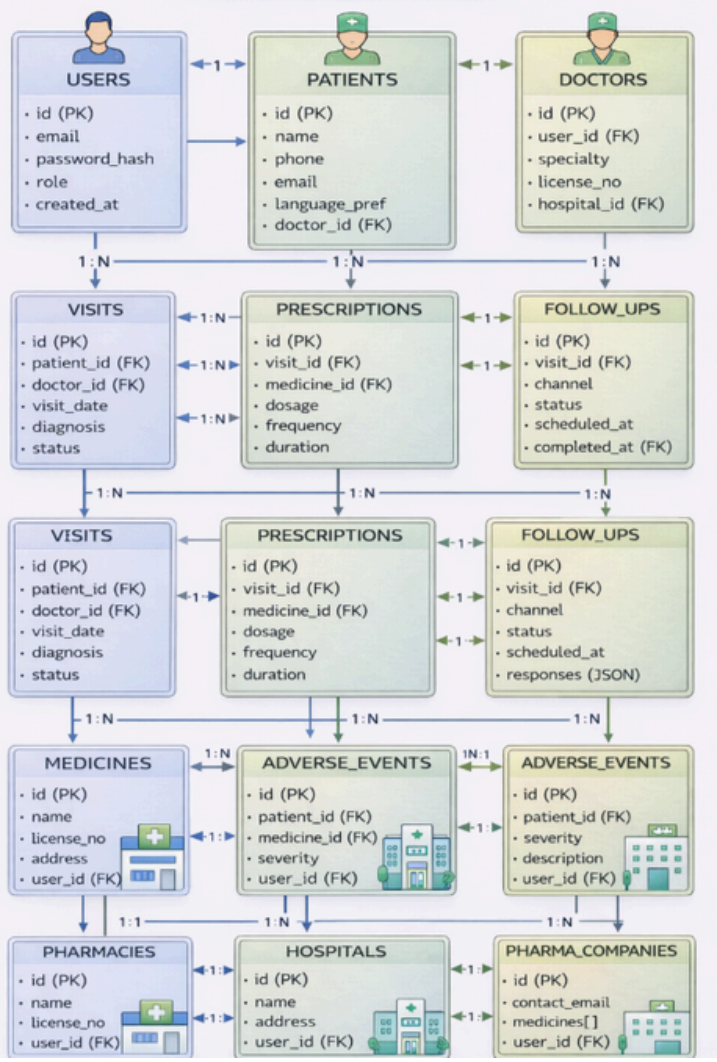
3

static - Frontend Assets

```
static/
├── css/
│   └── style.css             → Main stylesheet
├── js/
│   ├── auth.js               → Authentication logic
│   ├── doctor.js             → Doctor dashboard logic
│   ├── pharma.js             → Pharma dashboard logic
│   ├── alerts.js             → Alert handling
│   ├── reports.js            → Report generation
│   ├── pharmacy-reports.js   → Pharmacy reports
│   ├── sidebar.js            → Navigation sidebar
│   └── warnings.js           → Warning system
└── images/                   → Image assets
```

5.5 Database Structure

Database Schema



"This schema ensures traceable, role-aware, and regulation-ready pharmacovigilance data flow across clinical, follow-up, and safety analysis layers."

At the core, the **USERS** table provides a unified authentication layer, while domain-specific entities (doctors, hospitals, pharmacies, pharma companies) extend user roles.

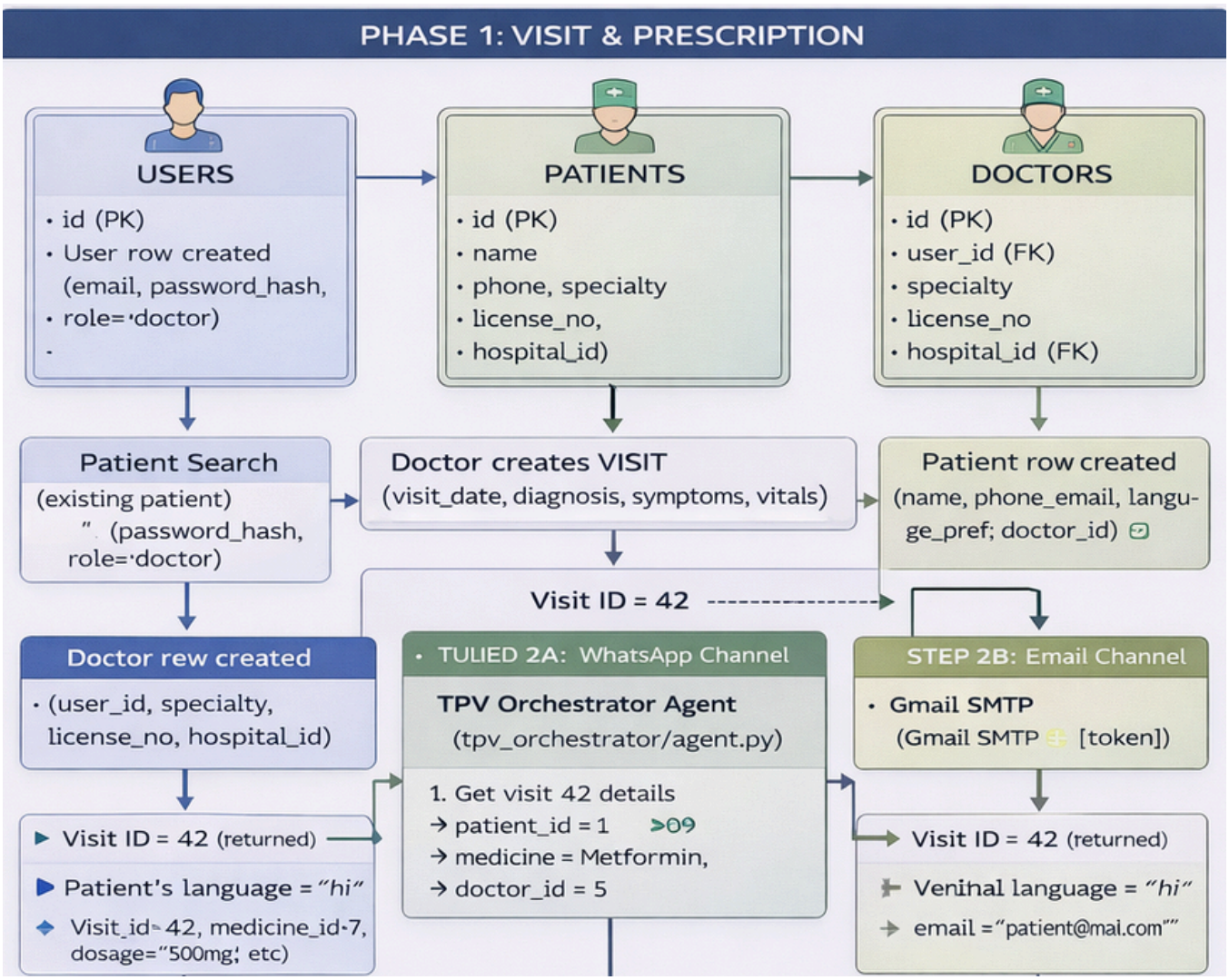
Clinical activity is captured through **VISITS**, **PRESCRIPTIONS**, and **FOLLOW_UPS**, with **ADVERSE_EVENTS** acting as the central safety intelligence entity linking patients, medicines, and follow-ups.

Core Relationship Logic

- **Users** define identity and access across all system roles
- **Doctors** manage patients and create visits
- **Visits** anchor prescriptions and follow-ups
- **Follow-ups** capture structured patient feedback
- **Adverse Events** consolidate safety signals across medicines and patients
- **Medicines** link prescriptions, ADRs, and pharma ownership

Table Name	Purpose	Key Relationships
USERS	Authentication and role management	1→1 with Doctors, Hospitals, Pharmacies, Pharma_Companies
DOCTORS	Medical professionals in the system	N→1 Users, N→1 Hospitals, 1→N Patients, 1→N Visits
HOSPITALS	Clinical institutions	1→1 Users, 1→N Doctors
PHARMACIES	Drug dispensing entities	1→1 Users
PHARMA_COMPANIES	Drug manufacturers and sponsors	1→1 Users, 1→N Medicines
PATIENTS	Individuals receiving treatment	N→1 Doctors, 1→N Visits, 1→N Adverse_Events
VISITS	Clinical encounters	N→1 Patients, N→1 Doctors, 1→N Prescriptions, 1→N Follow_Ups
PRESCRIPTIONS	Medicines prescribed during visits	N→1 Visits, N→1 Medicines
FOLLOW_UPS	Post-visit data collection	N→1 Visits, 1→N Adverse_Events
MEDICINES	Drug master data	1→N Prescriptions, 1→N Adverse_Events, N→1 Pharma_Companies
ADVERSE_EVENTS	Safety signals and reactions	N→1 Patients, N→1 Medicines, N→1 Follow_Ups

Data Flow



PHASE 1: Visit & Prescription
This phase captures clinical data at the point of care.

Doctor Authentication
- A doctor logs in using credentials stored in the USERS table.
- A corresponding DOCTORS record links the user to medical credentials and hospital affiliation.

Patient Identification
- The doctor searches for an existing patient or creates a new patient record.
- Patient data includes contact details, preferred language, and assigned doctor.

Visit Creation
- A VISIT is created containing diagnosis, symptoms, vitals, and visit date.
- The system generates a unique Visit ID (e.g., 42), which becomes the anchor for all future actions.

Prescription Entry
- Medicines are prescribed during the visit.
- Each prescription is linked to:
  - the visit
  - the patient
- the specific medicine and dosage

This phase ensures traceability from **doctor → patient → visit → medicine.**

PHASE 2: Follow-Up Trigger & Channel Selection
This phase initiates post-treatment monitoring.

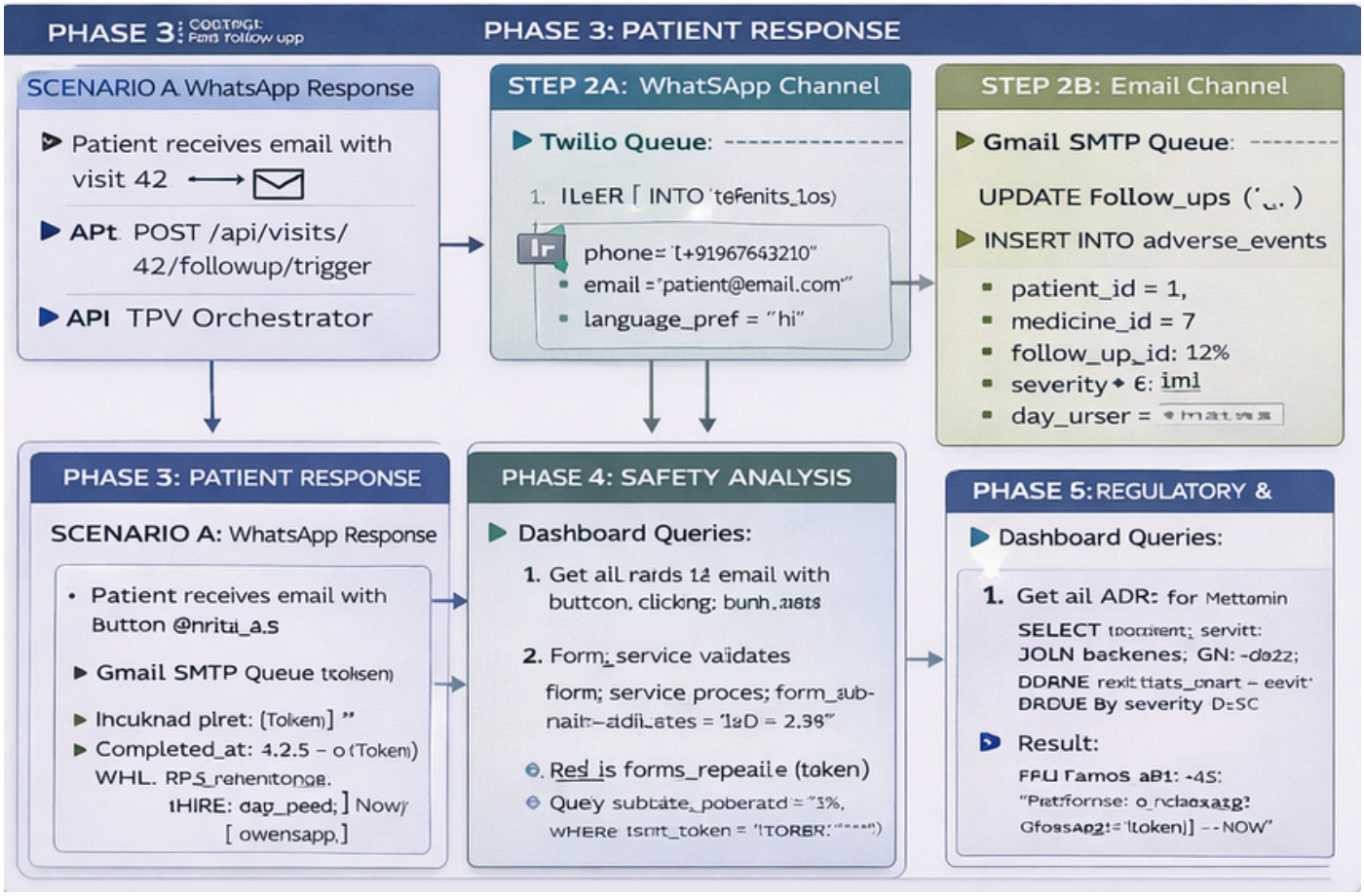
Follow-Up Trigger
- The doctor clicks "Send Follow-up" for a specific visit.
- An API request triggers the TPV Orchestrator Agent.

Orchestrator Decision Logic
- The agent fetches visit, patient, and medicine details.
- Based on patient preferences and availability:
  - WhatsApp is selected as the primary channel
  - Email + form acts as a fallback

Channel Execution
- WhatsApp messages are sent using Twilio.
- Email messages contain a secure, tokenized form link.

Each attempt is recorded in the FOLLOW_UPS table with status and timestamps.

Data Flow



PHASE 3: Patient Response
Patients can respond through two supported paths.

Scenario A: WhatsApp Response
- Patient receives questions in their preferred language.
- Replies are collected via Twilio webhooks.
- The WhatsApp agent: Maintains conversation state and asks conditional questions (e.g., symptom follow-ups)
- Responses are saved as structured JSON in the follow-up record.

Scenario B: Email / Form Response
- Patient opens a secure web form using a token.
- The system validates: Token authenticity, Expiry, One-time usage
- Conditional questions appear based on earlier answers.
- On submission, responses are stored and the follow-up is marked complete.

Both paths ensure data completeness without overwhelming the patient.

PHASE 4: Safety Analysis
This phase converts raw responses into clinical safety intelligence
.
Response Aggregation
- Data from WhatsApp or forms is normalized into a single structure.
- AI Safety Agent Analysis
- Uses an LLM (Gemini) to: Extract symptoms, Match known adverse drug reactions Assess causality (temporal + known ADRs)
- Calculate severity score

Adverse Event Creation
- A new ADVERSE_EVENTS record is created containing: patient, medicine, symptoms, severity, AI reasoning, escalation status and Doctor Notification
- Moderate or severe cases trigger alerts on the dashboard.
- Doctors can view full AI explanations for transparency.

PHASE 5: Regulatory & Analytics
This phase supports population-level safety monitoring and compliance.

- Dashboards aggregate adverse events by:
  - drug
  -severity
  -symptom frequency
- Incidence rates are calculated and compared against known baselines.
- Signal detection flags:
  -unexpected symptoms
  -rising frequency trends
- Pharma companies and regulators receive automated reports for high-risk signals.

This enables early risk detection and regulatory readiness.

6.Future Scope: Organization Verification & Trust Framework - for authorization

Currently, the platform allows hospitals, doctors, pharmacies, and pharmaceutical companies to sign up without full institutional verification. This approach was intentionally chosen during the initial phase to focus on core system architecture and data flow design.
However, due to the highly sensitive nature of healthcare and pharmaceutical data, future versions of the platform will introduce a multi-layer organization verification mechanism.

6.1 Need for Verification
Even when data is anonymized, improper access or manipulation can lead to:

- Data misuse
- Incorrect drug analysis
- Ethical and legal risks

Therefore, it is critical to ensure that only legitimate and regulated organizations are allowed to contribute and access data.

6.2 Proposed Verification Strategy
In the future, every organization signing up on the platform will undergo mandatory verification based on government-recognized credentials.

Hospitals and Clinics

- Clinical Establishment Registration Number
- State Health Authority License
- GST Registration
- Official institutional email domain verification

Local Pharmacies

- Drug License Number issued by the Drug Control Authority
- GST Registration
- Registered Pharmacist License (RPh ID)

Pharmaceutical Companies

- Corporate Identification Number (CIN)
- GST Registration
- CDSCO / Manufacturing License
- Official company email domain

6.3 Individual Identity Verification
The individual representing an organization may be required to submit:
PAN Card for identity validation

Aadhaar verification (optional and consent-based)

This ensures accountability while maintaining privacy compliance.

6.4 Approval Workflow
Automated document validation

Manual admin review

Role-based access granted only after approval

This approach balances security, scalability, and ethical data handling.