# Ensuring HIPAA Compliance in the NLP FHIR Query Tool

The current NLP FHIR Query Tool allows users to run natural language queries over FHIR compliant patient data. Given the sensitivity of healthcare data, ensuring the system is HIPAA compliant is essential. Below is a breakdown of what the system already includes and what improvements can enhance its security and compliance posture.

## Authentication & Authorization:

Currently:

- The backend is hosted securely on Render and exposes a /query endpoint via Flask.
- No authentication mechanism is currently in place, so the API is publicly accessible.

To Implement:

- Integrate OAuth 2.0 with SMART on FHIR to ensure only authorized users can query FHIR data.
- Use access tokens scoped by role (e.g., user/*.read) to restrict permissions dynamically.

## Data Privacy & Audit Logging

Currently:

- The tool processes patient queries in-memory using spaCy, and no PHI is stored at rest.
- Requests are handled over HTTPS when deployed, ensuring data is encrypted in transit.
- Errors (e.g., no diagnosis found) are returned in a user-friendly but informative way.

To Implement:

- Add structured audit logging: capture user ID, query timestamp, endpoint hit, and success/failure status. Logs should be encrypted and access-controlled.
- Ensure all dependencies and containers are updated and scanned for vulnerabilities regularly.

## Role-Based Access Control (RBAC)

Currently:

- The tool doesn’t yet implement RBAC — any frontend request can access backend functionality.

To Implement:

- Define user roles such as Clinician, Researcher, and Admin.
- Use role-based filters in both the frontend and backend to control what data is returned (e.g., researchers see only de-identified results).
- Incorporate claims-based logic in Flask using JWT or session-based access to enforce RBAC.