HIPAA Compliance and Security Plan

This document outlines how the AI on FHIR application I designed ensures HIPAA compliance and secure handling of healthcare data. My goal was to build a system that respects patient privacy while demonstrating sound technical and ethical principles in healthcare data management.

1. Authentication & Authorization (OAuth 2.0 & SMART on FHIR)

The system would use OAuth 2.0 as the authentication standard, ensuring users securely log in through a trusted identity provider. For healthcare-specific access, the SMART on FHIR protocol extends OAuth 2.0, enabling granular permissions so apps only access the FHIR resources they're authorized for. This prevents data overreach and enforces user accountability.

2. Data Encryption (At Rest & In Transit)

All communication between the frontend and backend would be encrypted definitely using HTTPS (TLS 1.3). Sensitive information stored temporarily for mock purposes follows the principle of encryption-at-rest using AES-256 standards. This would ensure even in the unlikely event of a breach, data remains unreadable and protected.

3. Role-Based Access Control (RBAC)

Users would be assigned roles such as healthcare providers, analysts, and even administrators with defined access scopes. RBAC ensures users can only view or manipulate data relevant to their respective function. This prevents unauthorized access and minimizes exposure of sensitive patient data records, aligning with the "minimum necessary" rule under HIPAA.

4. Audit Logging & Monitoring

Every interaction with the FHIR data layer will be logged, I would be capturing the user's identity, timestamp, and the type of resource accessed. These logs would be stored securely and reviewed periodically to detect anomalies or potential breaches. Having a traceable audit trail reinforces accountability and transparency across the system.

5. Data Minimization & Privacy by Design

I would approach the architecture with a “privacy by design” mindset. Only essential data attributes are handled, and the mock FHIR API simulates responses without exposing real patient identifiers. This practice reduces the risk of accidental data disclosure and maintains compliance even during testing or demos.

6. Infrastructure & Deployment Security

When deployed, the system would rely on HIPAA-compliant hosting providers (such as AWS or Google Cloud’s healthcare APIs). All environment secrets like API keys and tokens would be stored securely using environment variables or managed secrets storage. Regular patches and vulnerability scans would also be part of the maintenance routine for sure.

In essence, the system would demonstrates how AI-powered FHIR applications can balance innovation and compliance. Beyond technical safeguards, I designed this solution to reflect a real-world awareness of patient privacy and healthcare ethics.