Technology Advice Based on Existing Systems Table

System	Category	Tech Components	Technology Advice
DHIS2	Storage	Relational database, PostgreSQL. Cloud storage for scalability, local data caching for offline access	- Replace existing storage methods with SQL databases optimized for Online Transaction Processing (OLTP) to handle large-scale transactional data. - Leverage PostGIS for geospatial extensions, enabling TB hotspot mapping - Implement encryption for data protection, addressing privacy concerns
	Communications	APIs for data integration, SMS/USSD for low-bandwidth communication, Internet for synchronization	- Integrate HL7 and HTTPS protocols for secure, standardized data exchange. Implement APIs that adhere to REST standards for better system interoperability. - Add SMS gateways and USSD workflows for low-bandwidth communication in rural areas
	User Interface	Web-based interface for centralized access, responsive design for mobile use, customizable modular design	- Use responsive design frameworks for mobile and desktop compatibility - Include role-based customizations to streamline user-specific workflows - Add visualization libraries
	Reporting	Customizable report generation, real-time data visualization dashboards, analytics for health indicators, export options (CSV, Excel, PDF)	- Optimize SQL queries for Online Analytical Processing (OLAP) systems to support complex analytics and dashboards - Automate report generation using Python libraries (e.g., Pandas) for real-time insights - Integrate GIS tools (e.g., QGIS or PostGIS) for geospatial reporting on TB hotspots.
	Data Entry	Web forms, mobile apps with offline data entry options, validation rules to prevent data entry errors	- Use offline-ready data entry tools with automatic synchronization to the main server. - Incorporate real-time validation rules using client-side scripting (e.g., JavaScript) to prevent errors - Integrate barcode scanners for rapid patient identification
HIV Management System	Storage	SQL databases for structured data storage, encrypted local storage for sensitive patient data, cloud backup options	- Replace existing storage methods with SQL databases optimized for OLTP to handle large-scale transactional data. - Leverage PostGIS for geospatial extensions, enabling TB hotspot mapping - Implement encryption for data protection, addressing privacy concerns

	Communications	Secure data exchange protocols (HTTPS, HL7), APIs for cross-system integration, SMS/email notification for patient follow-ups	- Integrate HL7 and HTTPS protocols for secure, standardized data exchange - Implement APIs that adhere to REST standards for better system interoperability - Add SMS gateways and USSD workflows for low-bandwidth communication in rural areas
	User Interface	Desktop and mobile interfaces with customizable workflows, simplified screens for rapid data entry, role-based access	- Use responsive design frameworks (e.g., Bootstrap) for mobile and desktop compatibility - Include role-based customizations to streamline user-specific workflows - Add visualization libraries like Chart.js for intuitive
	Reporting	Predefined and customizable patient and aggregate data reports, HIV indicator tracking dashboards, data export tools	- Optimize SQL queries for OLAP systems to support complex analytics and dashboards - Automate report generation using Python libraries (e.g., Pandas) for real-time insights - Integrate GIS tools (e.g., QGIS or PostGIS) for geospatial reporting on TB hotspots
	Data Entry	Structured data entry forms, real-time validation, batch data entry options, integration with barcode scanners for patient identification	- Use offline-ready data entry tools with automatic synchronization to the main server - Incorporate real-time validation rules using client-side scripting (e.g., JavaScript) to prevent errors - Integrate barcode scanners for rapid patient identification
EMR (Electronic Medical Record)	Storage	Relational databases (MySQL, SQL Server), encrypted storage for patient data, local storage with sync for offline scenarios	- Replace existing storage methods with SQL databases optimized for OLTP to handle large-scale transactional data. - Leverage PostGIS for geospatial extensions, enabling TB hotspot mapping - Implement encryption for data protection, addressing privacy concerns
	Communications	Interoperability standards (e.g., HL7, FHIR) for external system integration, secure	- Integrate HL7 and HTTPS protocols for secure, standardized data exchange - Implement APIs that adhere to REST standards for better system interoperability

		messaging protocols for patient data, notification systems	- Add SMS gateways and USSD workflows for low-bandwidth communication in rural areas
	User Interface	Web and desktop interfaces, customizable for different medical workflows, user-friendly design for clinicians	- Use responsive design frameworks (e.g., Bootstrap) for mobile and desktop compatibility - Include role-based customizations to streamline user-specific workflows - Add visualization libraries like Chart.js for intuitive
	Reporting	Standardized and customizable reporting tools for patient outcomes, interactive analytics dashboards, automated reporting for regulatory compliance	- Optimize SQL queries for OLAP systems to support complex analytics and dashboards - Automate report generation using Python libraries (e.g., Pandas) for real-time insights - Integrate GIS tools (e.g., QGIS or PostGIS) for geospatial reporting on TB hotspots
	Data Entry	Electronic forms with standardized fields, options for structured and unstructured data entry, error-checking and validation mechanisms	- Use offline-ready data entry tools with automatic synchronization to the main server. - Incorporate real-time validation rules using client-side scripting (e.g., JavaScript) to prevent errors. - Integrate barcode scanners for rapid patient