**Technical Document**

**Project Name:** SehatAI – AI-powered Healthcare Support System for Rural Pakistan  
**Version:** 1.0  
**Date:** September 2025

**1. Title Page & Abstract**

**Abstract:**  
SehatAI is an AI-powered solution designed to assist rural and public health institutions in Pakistan. It addresses the lack of diagnostic tools, prescription digitization, and predictive health monitoring. Using open datasets and open-source AI tools, SehatAI provides:

* **X-ray disease detection** (TB, pneumonia).
* **Prescription digitization** (Urdu & English).
* **Risk scoring models** for diabetes and heart disease.

*(Refer to SRS Section 1 & 2 for extended details.)*

**2. Introduction**

**Problem Statement:**  
Millions in rural Pakistan lack access to quality diagnostics and digital health systems. X-rays go unread, prescriptions are handwritten and often unclear, and health risks remain unmonitored.

**Motivation / Why AI?**  
Traditional methods (manual diagnosis, handwritten notes) are slow, error-prone, and not scalable. AI enables **fast, accurate, low-cost, and automated health support** even on simple computers.

**Objectives (MVP):**

* Detect TB/pneumonia from X-rays (≥85% accuracy).
* Digitize Urdu/English prescriptions (≥80% OCR accuracy).
* Provide health risk scoring with AUC ≥ 0.8.

*(Refer to SRS Section 3.)*

**3. System Overview**

**Workflow Diagram (High-level):**

User (Doctor/Clinic Staff)

↓

Upload Data (X-ray / Prescription / Patient Info)

↓

Preprocessing (Image cleaning, text extraction, data formatting)

↓

AI Models

(CNN for X-rays, OCR for prescriptions, ML for risk scoring)

↓

Results Generated

(Diagnosis, Digital Prescription, Risk Score)

↓

Dashboard (Streamlit/Flask)

↓

User sees results, downloads report

**Interaction with End Users:**  
Doctors upload files (X-ray or prescriptions) or enter patient data. AI models process them and show results in a simple dashboard.

**4. MVP Scope**

**In-Scope:**

* X-ray classification (TB/pneumonia).
* Prescription OCR (Urdu & English).
* Risk scoring model.
* Dashboard prototype.

**Out-of-Scope:**

* Integration with NADRA e-Health ID.
* GPU-heavy training.
* Nationwide deployment.

**Constraints:**

* CPU-only training (Phase-1 requirement).
* Use of only open datasets.
* Simplified dashboard (Streamlit).

*(Refer to SRS Section 8.)*

**5. User Types & Use Cases**

**User Types:**

* Doctors/paramedics (primary users).
* Patients (secondary, for reports).
* Government officers (secondary, for monitoring).

**Use Cases:**

* Doctor uploads X-ray → AI detects TB/pneumonia.
* Staff uploads prescription → OCR converts it.
* Doctor enters patient details → AI gives risk score.

**6. Functional Requirements**

* Input:
  + Images (JPG/PNG for X-rays).
  + Images (JPG/PNG for handwritten prescriptions).
  + Tabular input (CSV/JSON for patient demographics).
* Core AI Functions:
  + Classification (CNN for X-rays).
  + OCR (prescriptions).
  + Prediction (risk scoring).
* Output:
  + JSON results (for APIs).
  + Human-readable report in dashboard.
* Interfaces/APIs:
  + REST API for AI inference.
  + Web dashboard (Streamlit).

**7. Non-Functional Requirements**

* **Performance:** Models should return results in <10s.
* **Scalability:** Prototype supports small dataset; scalable later.
* **Security:** Basic login/authentication.
* **Usability:** Bilingual, simple UI.

*(Refer to SRS Section 7.)*

**8. Data Requirements**

* **X-rays:** NIH Chest X-ray, Kaggle TB dataset.
* **Prescriptions:** UCOM Urdu Handwritten dataset + synthetic data.
* **Risk Scoring:** Pakistan Demographic and Health Survey (PDHS).

**Preprocessing Needs:**

* Resize/normalize X-rays.
* Augment prescription images.
* Clean tabular survey data.

**MVP Dataset Size:**

* ~5,000–10,000 X-rays (sample subset).
* ~2,000–3,000 prescription images.
* ~1,000–2,000 records for risk scoring.

*(Refer to SRS Section 10.)*

**9. Model Design (AI Core)**

* **X-ray Classification:** CNN (ResNet/DenseNet).
* **Prescription OCR:** Tesseract OCR (Urdu + English).
* **Risk Scoring:** Logistic Regression (Scikit-learn).

**Evaluation Metrics:**

* X-ray accuracy ≥85%.
* OCR accuracy ≥80%.
* Risk scoring AUC ≥0.8.

**10. System Architecture**

**MVP Architecture:**

Frontend (Streamlit Dashboard)

↓

Backend (Flask REST API)

↓

AI Models (PyTorch CNN, Tesseract OCR, Scikit-learn ML)

↓

Database (SQLite/PostgreSQL for storing results)

**Tech Stack:**

* **Best:** PyTorch, Streamlit, Tesseract, Scikit-learn.
* **Alternative:** TensorFlow, Flask/Django, EasyOCR, LightGBM.

*(Refer to SRS Section 9.)*

**11. Integration & APIs**

* **REST APIs:**
  + /xray → Input: Image → Output: Diagnosis JSON.
  + /ocr → Input: Image → Output: Text JSON.
  + /risk → Input: JSON patient data → Output: Risk Score.
* **Formats:** JSON, CSV, PNG/JPG.

**12. MVP Testing & Validation**

* **Testing Methods:**
  + Unit tests for each AI model.
  + Validation split on datasets.
  + Integration test with dashboard.
* **Minimal Evaluation Goals:**
  + 70–75% acceptable for MVP.
  + Aim for ≥80–85% where possible.
* **Feedback Loop:**
  + Collect incorrect predictions.
  + Retrain with corrected samples.

**13. Deployment & Infrastructure**

* **Environment:** Local CPU (Phase-1).
* **Containerization:** Docker for reproducibility.
* **CI/CD:** GitHub Actions for auto-testing.

**14. Risks & Limitations**

* **Accuracy Risks:** OCR on messy Urdu handwriting may struggle.
* **Ethical Risks:** Misdiagnosis if AI is used without doctor oversight.
* **Technical Risks:** Limited compute power (no GPU in Phase-1).

**15. Project Timeline (14 Days)**

* **Day 1–2:** Dataset collection, team roles.
* **Day 3–5:** X-ray model training.
* **Day 6–7:** OCR setup + testing.
* **Day 8–9:** Risk scoring model.
* **Day 10:** Dashboard development.
* **Day 11–12:** Model integration.
* **Day 13:** Testing, report writing.
* **Day 14:** Final submission (demo + docs).

*(Refer also to SRS Section 11.)*

**16. Future Enhancements**

* Integration with NADRA e-Health ID.
* Mobile app for rural doctors.
* Cloud deployment with GPU acceleration.
* Multi-disease detection (expand beyond TB/pneumonia).
* Analytics dashboard for government health monitoring.

**17. Summary / Conclusion**

SehatAI is a **feasible, impactful MVP** built with **open datasets and open-source AI tools**, aligned with Techathon guidelines.  
It solves **real healthcare challenges in rural Pakistan** by:

* Detecting diseases via X-rays.
* Digitizing Urdu/English prescriptions.
* Predicting patient risks.

This MVP validates the **concept’s feasibility** and sets the foundation for **Phase-2 scaling and Phase-3 commercialization**.