**MediTrack: Healthpost’s Database Management System**

Shobha Adhikari

Rajesh Sunar

Westcliff University

Fundamentals of C Programming

Professor Puri

January 21, 2025

**MediTrack: Healthpost Database Management System**

### **Introduction**

This Healthpost Database Management System is designed to simplify and improve the healthcare services at local health posts, particularly in rural areas in Nepal. The system addresses common inefficiencies such as the manual handling of patient records, health worker data, and other critical information. Using C programming, this project provides a practical solution to streamline data management processes, ensuring better accessibility, reliability, and service delivery.

**Problem Statement**

During community work, our friend discovered that most of Nepal's healthcare posts need better data filing systems in the village. They need to record all the required information about the medicines provided to patients or the diseases being treated. Patient records may be stored in various places, making it difficult to retrieve information efficiently. This can lead to errors in patient care, especially in emergencies.

Given the manual nature of record-keeping, there is a significant risk of records being lost or destroyed due to natural disasters, accidents, or poor storage conditions. This is particularly a problem in areas prone to floods, landslides, and other natural disasters. To resolve these issues, we would like to create a useful database management system for the Bhotang health post.

### **Objectives**

The project objectives are defined using the SMART framework:

* **Specific:** Create a C-program-based system to manage patient and health worker data.
* **Measurable:** Store data for at least 100 patients and 10 health workers for now.
* **Achievable:** Use fundamentals of C programming features, such as arrays, file handling, and functions.
* **Relevant:** Solve existing challenges in rural health post-data management.
* **Timebound:** Complete the project within four weeks.

### **Scope**

**Included:**

* Patient information management (Patient ID, Name, DOB, Gender, Contact).
* Health worker data management (Health worker ID, Name, specialization, contact).
* Viewing the recorded information.
* Deleting functionality.

**Excluded:**

* Appointment scheduling.
* Medication inventory management.
* Integration with external systems.
* Advanced analytics or reporting.

### **Methodology**

#### **Approach**

* Analyze requirements and design a basic database structure.
* Use modular programming techniques to implement Create, Read, and Delete operations for each entity.
* Incorporate file handling for data persistence.

#### **Tools and Technologies**

* **Compiler:** GCC, G++
* **Code Editor:** VSCode
* **Version Control:** GitHub

#### **Techniques**

* Arrays for temporary data storage.
* Functions for organizing operations like adding, deleting, and retrieving data.
* Loops and conditionals for logical operations.
* File handling to store and retrieve data persistently.

### **Process**

**Week 1: Planning and Setup**

* Finalized project requirements and defined the scope.
* Set up the development environment with the necessary tools.
* Created the base structure of the program.

**Week 2: Core Functionalities Development**

* Developed functions for managing patient and health worker data.
* Implemented Create, Read, and Delete operations.
* Began file handling for persistent storage.

**Week 3: Extended Features and Testing**

* Optimized and refined the code for better performance.
* Conducted module integration testing to ensure smooth functionality.
* Debugged and resolved issues in file handling.

**Week 4: Documentation, Presentation, and Finalization**

* Finalized the system by polishing the code.
* Prepared a presentation summarizing the project outcomes.
* Hosted the source code on GitHub.
* Conducted final testing to ensure all functionalities worked as intended.

### **Challenges**

### Ensuring seamless interaction between different entities, such as patients and healthcare workers, required extensive debugging and rigorous testing.

### Balancing the project timeline with academic commitments was challenging, especially during the testing and debugging phases.

### Starting the coding process was difficult due to the complexity of the database management system (DBMS) involved in the project.

### The initial phases were particularly challenging as linking various programs, functions, and files required significant effort and troubleshooting.

### The project required the implementation of several programs that were not covered in our coursework. So, it was also very challenging to research and understand new concepts.

### **Outcomes**

The project successfully achieved its objectives:

* Developed MediTrack: a functional Healthpost Database Management System in C.
* Simplified the process of managing patient and health worker data.
* Ensured persistent data storage through file handling.
* Demonstrated the applicability of programming concepts in solving real-world problems.

### 

### **Conclusion & Future Recommendation**

The Healthpost Database Management System provides a reliable and efficient solution for managing healthcare data in rural health posts. By reducing manual processes and ensuring data persistence, the system enhances service delivery and reduces errors. This project demonstrates how programming can address critical real-world challenges.

With further development, the system could include advanced features like appointment scheduling and medication inventory management, making it even more comprehensive and impactful.

### **Future Recommendations**

* Expand the system to include updating functionality, appointment scheduling and tracking, and stock medicine tracking.
* Develop a user-friendly interface for easier access.
* Integrate the system with hospital databases for better collaboration.
* Implement advanced reporting features for better decision-making.