HEALTH ASSISTANT COMMAND LINE APPLICATION

1)Title & Objective

Title: Getting started with a C++ health assistant API and PostgreSQL

Objective

-This guide demonstrates how to build a basic health assistant application using C++ postgreSQL database. The project serves as a minimum working example of a backend system showcasing c++ database connectivity and data management including CRUD Operations. This stack is chosen for its high performance and robust data handling Capabilities, ideal for development of efficient backend services

2)Quick Summary of the Technology

C++: A powerful high performance, general purpose programming language. It has been used to create a fast and efficient console application that manages patient records

PostgreSQL: An advanced open-source relational database system. It is known for its reliability and rich future set, including support for JSONB data types which are utilized To store structured health data

Libpqxx: A C++ library that provides a type-safe interface for interacting with PostgreSQL

nlohmann/json: A lightweight header only C++ library for handling json data, which is used for serialization and deserialization of data

3) System Requirements

Operating System: : Windows 10/11, Linux Distributions, macOs

Tools:
C++ compiler(e.g MinGW-w64 on Windows, g++ on Linux/MacOs)
PostgreSQL database server
A Package Manager for C++ libraries(vcpkg)

Libraries:
Libpqxx
Nlohmann-json

4) Installation or Setup Instructions

Install PostgreSQL
Windows: from official website:
https://www.enterprisedb.com/downloads/postgres-postgresql-downloads
Linux: sudo apt install postgresql postgresql-contrib
MacOS：brew install postgresql

Setup the Database for PostgreSQL
Connect the PostgreSQL server and create the database

Install C++ libraries - To install the required libraries:
vcpkg install pqxx:x64-windows
vcpkg install nlohmann-json:x64-windows

Compilation: Compile the project using C++ compiler:

5) Minimal Working Example

- **The Project provides a command-line interface to manage patient data. It demonstrates a complete CRUD flow**
- **Run an executable i.e (/.main)**
- **Expected Output: An interactive menu appears, prompting one to choose an operation to perform**

6)AI Prompt Journal
Prompt 1: Help me troubleshoot a C++ compilation error related to a missing symbol for 'isFitnessGoal'. 'The error is an undefined reference to…' Here is my 'fitness.h' and 'fitness.cpp'
AI Response: The AI accurately identified the problem as a missing function definition in fitness.cpp and provided the correct code to fix the linker error

Prompt 2: Explain how the provided C++ codebase for a PostgreSQL health assistant handles JSON data for structs like 'Nutrition' and 'Fitness'. Explain the role of 'Nlohmann::json' and JSONB in the database schema
AI Response: The AI provided a detailed description of the serialization and deserialization process, clarifying how C++ structs are converted into JSON And stored in the Databases JSONB columns

Prompt3: Generate a comprehensive C++ makefile for a multi- file project using 'libpqxx'

And ‘nlohmann’/json.hpp . The main file should handle compilation, linking, and cleaning and correctly specify include and library paths
AI Response: The AI created a fully functional Makefile correctly, setting up the necessary compiler and linker flags for the specified library

7)Common Issues and Fixes
. LinkerError(undefined reference to…) - Ensure all source files(.cpp) are included in the compilation command
.Database Connection Error - Verify the connection string in main.cpp matches your PostgreSQL server’s host, port, user and password
.Missing Headers(no such file or directory) - Check that your compiler’s include path (-I) is correctly pointing to the installed library header files

8. References
Libpqxx Official Docs: https://libpqxx.readthedocs.io/en/latest/
nlohmann/json Github: https://github.com/nlohmann/json
PostgreSQL Official site: https://www.postgresql.org/