Project Overview

This is a simple healthcare management system built using Python and SQLite that allows managing patient records and appointments through a console-based interface.

Code Approach

1. Database Design

The system uses SQLite database with two main tables:

patients: Stores patient information (ID, name, age, phone)

appointments: Manages appointment details linked to patients

2. Object-Oriented Structure

The system is implemented as a class SimpleHealthcare that encapsulates all database operations and user interactions.

Functions and Their Purpose

1. __init__(self)

Purpose: Initializes the database connection

Action: Creates connection to 'healthcare.db' and calls table creation

2. create_tables(self)

Purpose: Sets up database schema

Tables Created:

patients table with fields: id, name, age, phone

appointments table with fields: id, patient_id, doctor, date, reason

3. add_patient(self)

Purpose: Adds new patient to the database

Flow:

Takes user input for name, age, phone

Inserts record into patients table

Provides confirmation

4. view_patients(self)

Purpose: Displays all registered patients

Output: Lists all patients with their complete details

5. add_appointment(self)

Purpose: Schedules new appointment

Flow:

Shows existing patients

Takes patient ID, doctor name, date, and reason

Creates appointment record

6. view_appointments(self)

Purpose: Shows all scheduled appointments

Output: Displays appointment details including patient information

7. menu(self)
Purpose: Main user interface

Features:

Continuous loop until user exits

Option menu with 5 choices

Error handling for invalid inputs

Conclusion

This Healthcare Management System demonstrates a practical application of Python with SQLite for building simple database-driven applications. The project successfully implements:

Core CRUD Operations: Create, Read for patients and appointments

Database Relationships: Proper linking between patients and appointments

Modular Design: Well-organized class structure with separate methods for each functionality

User Interaction: Intuitive menu system for easy navigation

While basic in its current form, this system provides a solid foundation that can be extended with additional features like editing records, appointment conflicts checking, reporting, and a graphical user interface. The use of SQLite makes it lightweight and portable, suitable for small clinics or educational purposes.

The project effectively showcases fundamental programming concepts including object-oriented design, database operations, and user interface development, making it an excellent learning tool for understanding how to build practical database applications.