**Project Documentation: Hospital Database Management System**

This project is a Hospital Database Management System using PostgreSQL and Streamlit. The system includes the following pages: Home, Patients, and Doctors. For these to run, they must be in the same directory and the PostgreSQL database should be created as well.

**Home Page**

Purpose:

The home page serves as the landing page of the application. It displays a welcome message, an image, and a Bible quote.

Key Elements:

Welcome Message and Image: Displays the hospital name and logo.

Bible Quote: Shows a biblical quote under the image.

Sidebar Navigation: Contains links to navigate to other pages (Home, Patients, Doctors, Pharmacy, About, Code Snippets).

Implementation:

Defined in home.py.

Uses st.markdown for the message and quote, and st.image for the hospital logo.

Sidebar navigation is implemented using st.sidebar.radio.

**Patients Page**

Purpose:

Allows patients to either sign up as new patients or log in as existing patients.

New patients can enter their details and get a unique patient ID.

Existing patients can enter their patient ID to view and download their data.

Key Elements:

New Patient Form: Collects first name, middle name (optional), last name, and BP reading.

Existing Patient Form: Collects patient ID and displays patient data if the ID is valid.

Download Data: Allows patients to download their data in CSV format.

Implementation:

Defined in patients.py.

Functions for database connection, adding new patients, retrieving patient data, and downloading data.

Uses st.form for new patient and existing patient forms.

st.session\_state to manage state for patient ID.

**Doctors Page**

Purpose:

Allows doctors to either sign up or log in.

After logging in, doctors can access patient records by entering the patient ID and update the doctor's report and drug recommendation.

Key Elements:

Doctor Sign Up Form: Collects first name, middle name (optional), last name, and specialization. Generates a unique doctor ID starting from 100.

Doctor Log In Form: Collects doctor ID and last name for authentication.

Update Patient Record Form: Allows doctors to update the doctor's report and drug recommendation for a patient by entering the patient ID.

Implementation:

Defined in doctors.py.

Functions for database connection, creating a new doctor, verifying doctor credentials, and updating patient records.

Uses st.form for sign up, log in, and update patient record forms.

st.session\_state to manage logged-in state of doctors.

**Main Application**

Purpose:

Acts as the central script to run the application and manage navigation between different pages.

Key Elements:

Sidebar Navigation: Uses st.sidebar.radio to navigate between Home, Patients, Doctors, Pharmacy, About, and Code Snippets pages.

Page Routing: Imports and displays the corresponding page content based on the selected sidebar option.

**Implementation:**

Defined in app.py.

Imports the home, patients, and doctors modules.

Uses conditional statements to display the appropriate page content.

**Database Setup**

**Tables:**

Patients Table:

Columns: patient\_id (SERIAL, Primary Key), first\_name (VARCHAR), middle\_name (VARCHAR), last\_name (VARCHAR), bp\_reading (VARCHAR), doctor\_report (TEXT), drug\_recommendation (TEXT).

Doctors Table:

Columns: doctor\_id (SERIAL starting from 100, Primary Key), first\_name (VARCHAR), middle\_name (VARCHAR), last\_name (VARCHAR), specialisation (VARCHAR).

Requirements:

Ensure the database is created and the tables are set up as described.

Update connection parameters (dbname, user, password, host, port) in the connection function to match your PostgreSQL setup.

**Running the Application**

Ensure all files (app.py, home.py, patients.py, doctors.py) are in the same directory.

Ensure the PostgreSQL database is created and the tables are set up.

Run the application using the command:

streamlit run app.py

This will start the Streamlit application, and you can navigate through the pages using the sidebar.