Ex2-1:

**1. Goals and Purpose of the Exercise**

The main goal of this exercise is to practice how to use the sqlite3 module in Python to create and manage a simple database.

Through this task, the student learns:

* How to create an SQLite database file (doctors.db)
* How to define a table and its columns using SQL commands.
* How to perform CRUD operations (Create, Read, Update, Delete) using Python.

**2. Algorithm and Approach Description**

Step 1 – Create Database

Connect to a local database file using



Use CREATE TABLE IF NOT EXISTS to create the table doctors only once, containing:

A black screen with white text

AI-generated content may be incorrect.

Step 2 – Insert Data

Use parameterized query to insert data safely:

A black screen with green text

AI-generated content may be incorrect.

Step 3 – Select Data

Retrieve all rows and display results:



Step 4 – Update and Delete

Update salary or name by doctor ID:



Delete a record:



**3. Results and Demonstration**

**A screenshot of a computer program

AI-generated content may be incorrect.**

Ex2-2:  
**1. Goals and Purpose of the Exercise**

The goal of Exercise 2-2 is to extend the database created in Exercise 2-1 into a web application using Flask and SQLite..

The purpose of this exercise is to help students understand:

* How to use Flask to build a simple dynamic website.
* How to collect data from HTML forms and store it in an SQLite database.
* How to retrieve and display data from the database through a web page.

**2. Algorithm and Approach Description**

Step 1 – Initialize Flask Application

Import Flask and sqlite3.

Create a Flask app instance:



Step 2 – Create the Database Table

When the app starts, a function create\_table() checks if the table doctors exists.

If not, it runs:

A computer screen shot of text

AI-generated content may be incorrect.

Step 3 – Define Flask Routes

'/' (Home Page)

Displays a simple HTML form for entering doctor information.

When the user submits the form (POST), data is inserted into the database.

A black background with white text

AI-generated content may be incorrect.

'/doctors' (List Page)

Displays all doctors from the database in a simple HTML table using:



Step 4 – HTML Templates

The project uses three small HTML files placed in the templates/ folder:

form.html: the input form for adding doctors.

success.html: shown after a successful submission.

list.html: shows all doctor records in a table.

render\_template() is used to render these HTML files dynamically:



Step 5 – Run the Server

Start the web app locally using:



**3. Results and Demonstration**

(a) Form Input Page

The home page / allows users to input details such as:

First name, Last name, Joining date, Salary, Address.

When “Submit” is clicked, data is stored in doctors.db.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

(b) Success Page

A close up of a text

AI-generated content may be incorrect.

(c) Doctors List Page

A white rectangular box with black text

AI-generated content may be incorrect.