**Giri’s TechHub Pvt Ltd. Pune**

**Python Full Stack August -24**

Date: 06-03-2025

**Duration: 3hrs**

**Solve any one**

1. Creating a **simple doctor appointment booking system** in Python using a console-based interface.

**Requirements:**

1. **View available doctors and their schedules.**
2. **Book an appointment.**
3. **View booked appointments.**
4. **Cancel an appointment.**

**Use the following tables**

**Table for storing doctor details**

CREATE TABLE doctors (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR (255) NOT NULL

);

**Table for storing available slots for doctors**

CREATE TABLE slots (

id INT AUTO\_INCREMENT PRIMARY KEY,

doctor\_id INT,

slot\_time VARCHAR (50),

is\_booked BOOLEAN DEFAULT FALSE,

FOREIGN KEY (doctor\_id) REFERENCES doctors(id) ON DELETE CASCADE

);

**Table for storing appointments**

CREATE TABLE appointments (

id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_name VARCHAR(255) NOT NULL,

doctor\_id INT,

slot\_id INT,

FOREIGN KEY (doctor\_id) REFERENCES doctors(id) ON DELETE CASCADE,

FOREIGN KEY (slot\_id) REFERENCES slots(id) ON DELETE CASCADE

);

1. Create a simple **Hospital Patient Management System** using Python with a console-based interface.

**Requirements:**

1. **Add a new patient** (name, age, gender, contact details).
2. **View patient details**.
3. **Assign a doctor to a patient**.
4. **View assigned doctors for a patient**.
5. **Delete a patient record**.

#### ****MySQL Tables:****

**Table for storing patient details**

CREATE TABLE patients (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR (255) NOT NULL,

age INT NOT NULL,

gender ENUM ('Male', 'Female', 'Other') NOT NULL,

contact VARCHAR (15) NOT NULL

);

#### Table for storing doctor details

CREATE TABLE doctors (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR (255) NOT NULL,

specialization VARCHAR (255) NOT NULL

);

#### Table for assigning doctors to patients

#### CREATE TABLE patient\_doctor (

#### id INT AUTO\_INCREMENT PRIMARY KEY,

#### patient\_id INT,

#### doctor\_id INT,

#### FOREIGN KEY (patient\_id) REFERENCES patients(id) ON DELETE CASCADE,

#### FOREIGN KEY (doctor\_id) REFERENCES doctors(id) ON DELETE CASCADE

#### );