Hospital Patient Record & Billing System

Design and implement a **Hospital Management System** using **Core Python (with OOP)** and a **Relational Database (MySQL / MS SQL Server)** to manage **patients**, **doctors**, **services**, and **billing** operations. The system will allow admins to track patient data, assign doctors, record treatments, and generate bills automatically.

Technology Stack:

- **Language**: Core Python (OOP, File Handling, Exception Handling)
- **Database**: MySQL or MS SQL Server
- **Libraries**: mysql-connector-python or pyodbc, pandas, datetime

Database Schema (Tables):

Patients

patient_id	name	age	gender	admission_date	contact_no
1001	John Doe	32	M	2025-05-10	9999999999

doctors

doctor_id	name	specialization	contact_no
D01	Dr. Smith	Cardiology	8888888888

services

service_id	service_name	cost
S01	Blood Test	500

appointments

appt_id	patient_id	doctor_id	date	diagnosis
A001	1001	D01	2025-05-10	Hypertension

billing

bill_id	patient_id	total_amount	billing_date
B001	1001	3500	2025-05-11

Tasks for Learners

No.	Task Description	Area
1	**Design the ER diagram and normalize** the database up to 3NF	DB Design
2	Create all 5 tables using SQL CREATE statements	SQL
3	Create a **Python class** for each entity (Patient, Doctor, Service, Billing)	OOP
4	Establish **Python DB connection** using mysql-connector-python or pyodbc	Python + DB
5	Build a **menu-driven CLI application** with options like Add, View, Update, Delete	CLI
6	Implement **Add Patient/Doctor/Service** features with data validation	CRUD
7	Allow **Assigning doctors to patients** and store the diagnosis in appointments	Relational Mapping
8	Implement **date-based filtering** (e.g., appointments today, last week)	SQL + Python
9	Track multiple services used by the patient and store in a temp table/dictionary	OOP
10	Compute **total billing** by summing service costs and consulting charges	Python Logic
11	Insert billing details to the billing table with current date	SQL

No.	Task Description	Area
12	Fetch **complete patient history** including doctor visits and services used	Joins
13	Generate a **detailed invoice** with breakdown of services and total charges	File I/O
14	Implement **exception handling** (e.g., duplicate IDs, DB connection issues)	Error Handling
15	Use datetime to calculate days admitted, days between appointments, etc.	Python
16	Add **reporting features**: daily visits, most consulted doctors	SQL Aggregates
17	Implement **search functionality** (search patients/doctors by name)	SQL + Python
18	Add optional export of billing or appointment summary to .csv	File Handling
19	Use OOP principles – inheritance (e.g., Person base class for Patient/Doctor)	Python OOP
20	Create a final PDF report with screenshots and documentation	Documentation

Suggested Folder Structure

hospital_mgmt/

```
├── db_config.py       # DB connection logic
├── patient.py         # Patient class
├── doctor.py          # Doctor class
├── service.py         # Services logic
├── billing.py         # Billing logic
├── hospital_main.py   # Main menu & CLI
├── requirements.txt   # List of required libraries
├── output/
│   └── invoices/
│       └── bill_1001.txt
└── README.md
```

Note: In each table, there should be a minimum of 250 records.

Optional Visualizations (via Python or Power BI):

- Number of appointments per doctor per month
- Revenue generated from services
- Daily patient inflow trends

Deliverables:

- SQL scripts to create and populate DB
- Python source code (.py files)
- CSV reports (optional)
- User Guide & Screenshots
- Final PDF of documentation