

**hospital-hms\main.py**

```
1 """
2 Minimal FastAPI app for Hospital HMS – patient and appointment basics.
3 Run: `uvicorn main:app --reload --port 8000`
4 """
5 from fastapi import FastAPI, HTTPException
6 from pydantic import BaseModel
7 from typing import Optional
8 from datetime import date, datetime
9
10 from db import get_connection, rows_to_dicts
11
12 app = FastAPI(title="Hospital HMS (minimal)")
13
14
15 class PatientIn(BaseModel):
16     name: str
17     dob: Optional[date] = None
18     phone: Optional[str] = None
19
20
21 class PatientOut(PatientIn):
22     id: int
23
24
25 class AppointmentIn(BaseModel):
26     patient_id: int
27     doctor_id: int
28     scheduled_at: datetime
29     reason: Optional[str] = None
30
31
32 @app.get("/", summary="Health")
33 def health():
34     return {"status": "ok", "app": "hospital-hms"}
35
36
37 @app.get("/patients", response_model=list[PatientOut])
38 def list_patients():
39     conn = get_connection()
40     cur = conn.cursor()
41     cur.execute("SELECT id, name, dob, phone FROM patients ORDER BY id DESC FETCH FIRST 100 ROWS ONLY")
42     rows = rows_to_dicts(cur)
43     conn.close()
44     return rows
45
46
47 @app.get("/patients/{patient_id}", response_model=PatientOut)
48 def get_patient(patient_id: int):
49     conn = get_connection()
50     cur = conn.cursor()
51     cur.execute("SELECT id, name, dob, phone FROM patients WHERE id = :id", {"id": patient_id})
```

```
52     row = cur.fetchone()
53     conn.close()
54     if not row:
55         raise HTTPException(status_code=404, detail="Patient not found")
56     cols = [c[0].lower() for c in cur.description]
57     return dict(zip(cols, row))
58
59
60 @app.post("/patients", response_model=PatientOut, status_code=201)
61 def create_patient(p: PatientIn):
62     conn = get_connection()
63     cur = conn.cursor()
64     # Use Oracle RETURNING INTO to get generated ID
65     new_id = cur.var(int)
66     cur.execute(
67         "INSERT INTO patients (name, dob, phone) VALUES (:n, :d, :p) RETURNING id INTO :id",
68         n=p.name,
69         d=p.dob,
70         p=p.phone,
71         id=new_id,
72     )
73     conn.commit()
74     pid = new_id.getvalue()[0]
75     conn.close()
76     return {"id": pid, "name": p.name, "dob": p.dob, "phone": p.phone}
77
78
79 @app.post("/appointments", status_code=201)
80 def create_appointment(a: AppointmentIn):
81     conn = get_connection()
82     cur = conn.cursor()
83     cur.execute(
84         "INSERT INTO appointments (patient_id, doctor_id, scheduled_at, reason) VALUES (:p,:d,:s,:r)",
85         p=a.patient_id,
86         d=a.doctor_id,
87         s=a.scheduled_at,
88         r=a.reason,
89     )
90     conn.commit()
91     conn.close()
92     return {"ok": True, "patient_id": a.patient_id, "scheduled_at": a.scheduled_at}
93
```