

hospital-hms\README.md

# Hospital HMS (minimal) — FastAPI + Oracle

A tiny, resume-friendly backend template for a Cloud-Based Hospital Management System using Python (FastAPI) and Oracle Database.

This project is intentionally compact and beginner-friendly:

- No ORM, no Docker, no frontend — minimal backend only
- Beginner-friendly DB helper (`db.py`) using `oracledb` and env variables
- Compact SQL schema (`schema.sql`) for use in Oracle Cloud (Autonomous DB)

## Quick setup

1. Install Python 3.10+ and Oracle Instant Client (if needed) or use `oracledb` thin mode with network access.
2. Create a Python virtualenv and install requirements:

```
python -m venv .venv; .\.venv\Scripts\Activate; pip install -r requirements.txt
```

3. Configure environment variables for database connectivity:

```
$env:ORACLE_USER = "admin"  
$env:ORACLE_PASSWORD = "yourpassword"  
$env:ORACLE_DSN = "yourdb_high" # or host:port/service
```

4. Load `schema.sql` via SQL\*Plus, SQLcl, or a client in Oracle Cloud. For Autonomous DB, use the SQL workshop or SQLcl.
5. Run the API server:

```
uvicorn main:app --reload --port 8000
```

## API Endpoints

- GET / — health
- GET /patients — list patients (up to 100)
- GET /patients/{id} — get a patient by id

- POST /patients — create a patient (JSON: name, dob, phone)
- POST /appointments — create an appointment (JSON: patient\_id, doctor\_id, scheduled\_at, reason)

### Why this project is resume-ready

- Clean, minimal backend illustrating REST API, DB integrations, and environment-driven config
- Uses modern FastAPI and Oracle DB driver (oracledb) — relevant for enterprise roles
- Easy to extend with authentication, auditing, and advanced features

### Notes & Tips

- For production, use connection pooling and proper secret management (Vault / OCI Secrets). This template favours clarity and size.
- To connect to Oracle Autonomous DB use the correct wallet/connection options; set ORACLE\_DSN to the appropriate connect string.

License: MIT