

# **MediTrack — Clinic & Appointment Management System**

## **Project Summary**

MediTrack is a modular, object-oriented Clinic & Appointment Management System implemented in Core Java. The system models patients, doctors, appointments, and billing; demonstrates strong OOP design, SOLID principles, standard Java features (collections, exceptions, I/O, serialization), and optional advanced topics (concurrency, design patterns, streams, cloning). The project is ideal for mentees to practice Java fundamentals, architecture, collaboration, and documentation.

### **Expected Project Structure :**

```
src/main/java/com/airtribe/meditrack/
├── Main.java
├── constants/
│   └── Constants.java
└── entity/
    ├── Person.java
    ├── Doctor.java
    ├── Patient.java
    ├── Appointment.java
    ├── Bill.java
    └── BillSummary.java    // immutable
├── service/
    ├── DoctorService.java
    ├── PatientService.java
    └── AppointmentService.java
└── util/
    ├── Validator.java
    ├── DateUtil.java
    ├── CSVUtil.java
    ├── IdGenerator.java
    ├── DataStore.java    // generic DataStore<T>
    └── AIHelper.java    // optional
├── exception/
    ├── AppointmentNotFoundException.java
    └── InvalidDataException.java
└── interface/
    ├── Searchable.java
    └── Payable.java
└── test/
```

```
└── TestRunner.java    // manual tests (no JUnit)  
  
docs/  
├── JVM_Report.md  
├── Setup_Instructions.md  
└── Design_Decisions.md
```

## Deliverables

- Complete Java source code in GitHub repo with readable commits & branches.
- `README.md` with setup, usage, and demo instructions; include sample run outputs.
- `docs/` with `JVM_Report.md`, `Setup_Instructions.md`,  
`Design_Decisions.md`.
- `TestRunner.java` to run manual tests.
- UML class diagram (optional).
- Short 2–3 minute walkthrough video (optional but recommended).
- JavaDoc-generated HTML (bonus).