Hospital Management System (HMS)

A simple console-based Hospital Management System in ANSI C using binary files for persistent storage and basic role management.

Overview

- **Language**: ANSI C
- **Interface**: Console
- **Storage**: `.dat` binary files
- **Roles**: Admin, Doctor, Pharmacy Owner

The system runs locally and requires no external database.

Features

- **Admin**:
  - Manage patients: add, view, update, delete, discharge.
  - Manage doctors: add, view.
  - Manage appointments: book, view.
  - View bed occupancy report.
  - Generate billing records.
  - Manage medicines: add, view.
  - Manage lab results: add, view.
  - Trigger a daily report message.
- **Doctor**:
  - View patients.
  - View appointments.
  - View lab results.
- **Pharmacy Owner**:
  - Add medicines.
  - View medicines.

Data Structures

All entities are stored as C structs written to binary files.

Entity	Struct Name	Key Fields
Patient	`Patient`	`id`, `name`, `age`, `gender`, `disease`, `admitted`, `bedNo`
Doctor	`Doctor`	`id`, `name`, `specialization`
Bed	`Bed`	`bedNo`, `occupied`, `patientId`
Appointment	`Appointment`	`patientId`, `doctorId`, `date`
Bill	`Bill`	`patientId`, `consultation`, `medicine`, `room`, `total`
Medicine	`Medicine`	`id`, `name`, `quantity`, `expiry`
Lab	`Lab`	`patientId`, `testName`, `result`
User	`User`	`username`, `password`, `role`

Storage Files

Each struct type is persisted in its own binary file.

File	Description
`patients.dat`	Patient records
`doctors.dat`	Doctor records
`beds.dat`	Bed occupancy data
`appointments.dat`	Appointment records
`billing.dat`	Billing records
`medicines.dat`	Medicine inventory
`labs.dat`	Lab results
`users.dat`	User credentials and roles

Initialization

On first run, the system initializes core files if missing.

- **Beds**: `initBeds` creates 20 beds, all marked free.
- **Users**: `initUsers` creates three default users:
  - `admin` / `admin123` → Admin
  - `doctor` / `doc123` → Doctor
  - `pharmacy` / `pharma123` → Pharmacy Owner

Authentication and Menus

- `login()` reads `users.dat` and validates username and password.
- Based on the returned role, the program shows:
  - Admin main menu.
  - Doctor main menu.
  - Pharmacy main menu.
- Input is handled via `scanf` in a loop until the user selects `0` (Exit).

Core Modules

Patient Management

- **Functions**:
  - `addPatient()`
  - `viewPatients()`
  - `updatePatient()`
  - `deletePatient()`
  - `dischargePatient()`
- Patients are stored in `patients.dat`.
- Admission automatically calls `allocateBed()` to assign the first free bed.

Doctor Management

- **Functions**:
  - `addDoctor()`
  - `viewDoctors()`
- Data is stored in `doctors.dat`.

Appointment Management

- **Functions**:
  - `bookAppointment()`
  - `viewAppointments()`
- Uses patient and doctor IDs.
- Data is stored in `appointments.dat`.

Bed Management

- **Functions**:
  - `initBeds()`
  - `allocateBed(int patientId)`
  - `freeBed(int bedNo)`
  - `bedReport()`

- Beds are stored in `beds.dat`.
- `bedReport()` prints each bed status as occupied or free.

Billing

- **Function**:
  - `generateBill()`
- Stores consultation, medicine, and room charges.
- Computes `total` and saves to `billing.dat`.

Pharmacy

- **Functions**:
  - `addMedicine()`
  - `viewMedicines()`
- Medicines stored in `medicines.dat` with quantity and expiry.

Laboratory

- **Functions**:
  - `addLabResult()`
  - `viewLabResults()`
- Lab results stored in `labs.dat` and linked to a `patientId`.

Reports

- **Function**:
  - `dailyReport()`
- Currently prints a simple success message only.

Build and Run

Use a standard C compiler to build and run.

```
gcc -o hms "Hospital management system.c"
./hms
```

Quick Reference

Area	Key Functions
Setup	`initBeds()`, `initUsers()`, `login()`
Patients	`addPatient()`, `viewPatients()`, `updatePatient()`, `deletePatient()`, `dischargePatient()`
Doctors	`addDoctor()`, `viewDoctors()`
Appts	`bookAppointment()`, `viewAppointments()`

Beds	`allocateBed()`, `freeBed()`, `bedReport()`
Billing	`generateBill()`
Pharmacy	`addMedicine()`, `viewMedicines()`
Lab	`addLabResult()`, `viewLabResults()`
Reports	`dailyReport()`