Git Setup

HMS

GitHub & Git Complete Step-by-Step Guide (Windows + Antigravity IDE)

---

1. Is Guide ka Maqsad

Is document ka maqsad students ko **zero se** ye sikhana hai:

- Git kya hai
- GitHub repository kaise use hoti hai
- Commands **kahan aur kaise** chalani hain
- Project ko laptop par kaise lana hai
- Apna kaam GitHub par kaise bhejna hai

---

2. Required Cheezen (Pehle Check Karein)

2.1 Operating System

- Windows 10 ya Windows 11

2.2 GitHub Account

1. Browser open karein (Chrome / Edge)
2. Address bar mein likhein:

1. https://github.com

3. “Sign up” par click karein
4. Email, username, password enter karein
5. Account verify karein

---

3. Git Install Karna (Step by Step)

3.1 Git Download

1. Browser open karein
2. Address bar mein likhein:
2. https://git-scm.com/downloads

3. “Download for Windows” par click karein
4. File download hone dein

3.2 Git Install

1. Downloaded file par double click karein
2. “Next” par click karte jayein
3. Default settings change na karein
4. “Install” par click karein
5. Installation complete hone ka wait karein

---

4. Git Install Hua Ya Nahi? (IMPORTANT)

4.1 Command Prompt Open Karna

1. Keyboard se **Windows key** dabayein
2. Search box mein likhein:
3. cmd

3. “Command Prompt” par click karein

4.2 Command Run Karna

Command Prompt ke andar **ye likhein** aur Enter dabayein:

4. git --version

4.3 Expected Result

Agar output aisa aaye:

5. git version 2.xx.x

toh Git sahi install ho chuka hai.

Agar error aaye:

- Git sahi install nahi hua
- Instructor ko inform karein

---

5. HMS Main Repository Open Karna

5.1 Repository Page Open Karein

1. Browser open karein
2. Address bar mein **ye exact link paste karein**:
6. https://github.com/idaraalkhair4130-ux/hospital-management-system

3. Enter dabayein

---

6. Repository Fork Karna (Sab se Important Step)

Fork ka matlab hai:

Instructor wali repository ki **apni copy banana**

Steps:

1. HMS repository page open ho
2. Page ke **top right corner** par “Fork” button hoga
3. “Fork” par click karein
4. Wait karein jab tak fork complete ho

Result:

- Ab repository **student ke GitHub account** mein aa chuki hogi

---

7. Repository Link Copy Karna (Proper Way)

7.1 Code Button Se Link Copy Karna

1. Apni forked repository open karein
2. Green **“Code”** button par click karein
3. “HTTPS” option selected ho
4. **Copy icon** par click karein

Link example:

7. https://github.com/your-username/hospital-management-system.git

---

8. Laptop Par Project Folder Banana

1. File Explorer open karein
2. Desktop ya Documents par jayein
3. Right click → New Folder
4. Folder ka naam rakhein:

8. hms

---

9. Command Prompt Folder Ke Andar Open Karna

1. hms folder open karein
2. Folder ke address bar par click karein
3. cmd likhein aur Enter dabayein

Ab Command Prompt **isi folder ke andar** open hoga.

---

10. Repository Clone Karna (Laptop Par Lana)

Command Prompt mein **ye command paste karein**:

9. git clone https://github.com/your-username/hospital-management-system.git

Enter dabayein.

Result:

- hospital-management-system naam ka folder ban jayega

---

11. Project Antigravity IDE Mein Open Karna

1. Antigravity IDE open karein
2. “Open Folder” option select karein
3. hms/hospital-management-system folder select karein
4. Project load hone ka wait karein

---

12. Git Basic Commands (Daily Use)

Command Prompt kahan se open hoga?

- Project folder ke andar (Step 9 wala tareeqa)

Commands ka order:

10. git status
11. git add .
12. git commit -m "Short message"
13. git push origin main

---

13. Simple Example (Practice)

1. Antigravity IDE mein koi file change karein
2. File save karein
3. Command Prompt mein likhein:

14. git status
15. git add .
16. git commit -m "My first update"
17. git push origin main

---

14. Important Rules

- Students **main repository** par direct push nahi karenge
- Sirf apni forked repository use karein
- Sirf assigned module par kaam karein
- Final submission Pull Request ke zariye hogi

---

15. HMS Core Modules (Reference)

1. Authentication & Roles
2. Patient Management
3. Doctor Management
4. Appointment Management
5. Billing
6. Pharmacy
7. Lab Reports
8. Admin Dashboard

Hospital managment system HMS

Hospital Management System ki Tafseeli Wazahat

Hospital Management System (HMS) aik web-based application hai jo hospital ke kaam ko asan aur munazzam banata hai. Is mein mukhtalif roles ke users ke liye alag alag modules honge – jese **Admin**, **Doctor**, **Receptionist**, **Ward Boy/Staff**, aur **Pharmacist**. Har role ka apna **login**, **profile**, **dashboard** aur **settings** honge. System ki bunyaadi jhalak yeh hogi ke admin doctoron, receptionists, ward boys aur pharmacists ko add/remove karega; receptionist mariz (patient) ko doctor ke sath assign karke token number generate karega; doctor apne schedule aur token queue dekhega; aur pharmacist dawaon ka hisaab kitaab manage karega.

Poore system ka maqsad patient ki dekh bhaal, scheduling, billing aur pharmacy operations ko digital tor par manage karna hai takay hospital ke operations streamlined hon. Project requirements ke mutabiq **patient attendance system** ya **general inventory management** shamil nahin hai. Sirf in hisaon par tawajjo di jayegi jo neeche wazeh ki gayi hain.

System ke Ahem Features (Functional Requirements)

- **User & Role Management:** System me **3 khaas roles** honge – Admin, Receptionist, Pharmacist (aur agar ward boy ko role samjhen to 4th role). Har user apna account aur profile rakhay ga, jise wo edit kar sakta hai. Login ke zariye har user apni dashboard par pohanchta hai. Admin user roles create/add aur delete kar sakta hai.

- **Admin Module:**

  - **Doctors** ko add/remove karna (un ki basic maloomat, specialization, aur fee set karna). Admin har doctor ki **fees** set karega jo har consultation (slot/token) par apply hogi.

  - **Receptionists, Pharmacists, Ward Staff** ko add/remove karna. Admin un sab ke accounts create karega.

  - System ki settings aur configurations manage karna (jese hospital ki general settings).

  - Departments organize karna (agar hospital me departments hon tou un ki tasdeeq, jese Cardiology etc.). (Yeh optional hai agar requirement ho.)

  - **Reports Generation:** (Zaroori samjha to) simple reports jese total income ya token count nikalna.

- **Receptionist Module:**

  - **Patient Assignment:** Agar mariz pehli dafa aa raha ho tou uski **basic details** daakhil kare (naam, contact, mm?), varna pehle se registered mariz ko search kar kay chunay. Phir receptionist **doctor assign** karke token number generate karti hai. Har token ke sath mariz ke liye room number ya queue number fix ho sakta hai.

  - **Payment Collection:** Receptionist patient se doctor ki fee collect kar ke system me record karegi. Har doctor ki alag fee hoti hai jo Admin ne set ki hai.

  - **Appointment Scheduling:** Mariz ke liye appointment date/time select karna (ya token queue me daalna). Zaroorat par appointment ko reschedule ya cancel bhi kiya ja sakta hai.

  - **Token Management:** Receptionist digital **token display system** istemal karegi takay queue easily manage ho. (Jis tarah clinics me token systems hote hain).

  - **Doctor Assignment:** Receptionist kisi bhi doctor ko mariz ke saath assign kar sakti hai.

- **Doctor Module:**

  - **Schedule aur Appointments:** Doctor apne appointments ka schedule dekhega – kis din kin tokens (patients) ka intezaar hai. Apna daily timetable dekh sakta hai.

  - **Patient Records:** Jab doctor ke paas mariz aaye, tou wo unka case dekh kar zaroori notes ya prescription de sakta hai (system me update ho sakta hai). (Note: Ham log extended EHR nahin bana rahe, magar doctor ke liye patient ke treatment record rakhna madadgar ho ga.)

  - **Prescriptions:** Agar doctor dawa likhta hai tou pharmacist us prescription ko dekh ke dawaain nikal sakta hai. Yeh process pharmacy module se jurega.

  - **Status Updates:** Zaroori samjha tou doctor mariz ki status (inpatient, discharged etc.) update kar sakta hai. (Agar Inpatient module bohat important ho.)

- **Pharmacy Module:**

  - **Medicine Inventory:** Pharmacist naya **dawa** (medicines) system me add karega, aur jo purani expired hain unko remove karega. Har dawa ki brand, batch number, expiry date, aur price record rakhi jayegi.

  - **Prescription Handling:** Doctor ki prescription pharmacist dekhega aur uske mutabiq medicine patient ko dega. Agar doctor ke dwara kisi medicine ka substitute dena ho, pharmacist system me update kar ke record kar sakta hai.

○ **Billing aur Sales:** Pharmacist dawaon ki **cash** ya **credit** billing generate karega. Outpatient ya in-hospital patient dono ki receipts yahan banengi.

○ **Stock Alerts:** Pharmacy module khud expire ho rahi medicines ke liye alerts nikal sakta hai (out-of-stock ya expiry alerts). Yeh ghair-munasib ghaltiyan rokne me madadgar hota hai.

● **Ward Boy/Staff Module (Agar shamil hai):**

○ **Tasks and Assignments:** Ward boy ko marizon ki dekhbhaal aur hospital ki safai waghera ki zimmedariyan di ja sakti hain. System me wo apne assigned tasks dekhega.

○ **Profile and Dashboard:** Ward boy apni profile update karega, apna schedule ya duty roster dekh sakta hai. (Agar zaroorat ho toh.)

● **Common Features:**

○ **Profile Editing:** Har user (doctor/receptionist/etc.) apni profile (naam, password, contact waghera) update kar sakta hai. Profile settings me shayad multiple security options hon.

○ **Dashboards:** Har role ka dashboard hoga jahan relevant information hogi – jaise admin ke liye overall stats, doctor ke liye aaj ke tokens, receptionist ke liye pending payments etc.

○ **Authentication & Authorization:** Sab users secure login/password ke zariye system me aayenge. Role-based access control honi chahiye takay har user sirf apne role ke hisaab se hi modules use kar sake. (For example, receptionist doctor ka personal file nahin dekhegi.)

Ye saari functional requirements milke system ke primary kaam define karti hain. Har feature ke peeche dono sides (user interaction aur backend logic) consider ki jayegi.

Ghair-Functional Requirements (Non-Functional Requirements)

● **Security (Tahaffuz):** Patient aur hospital ke data ki hifazat sab se ahem hai. Role-based access control hona chahiye – sirf sahi users ko mutaliq information ki ijazat. Sensitive data encryption me rahna chahiye (jaise passwords, patient records). Regular security audits aur vulnerability testing honi chahiye. Healthcare regulations (HIPAA ya local data privacy kanoon) ki pabandi zaruri hai.

● **Scalability (Takaat):** System ko hospital ke barahtay huay istemal aur data load jhelna chahiye. Shuruat me chhota hospital ho sakta hai, magar baad me users (doctors, patients) barhein to performance compromise na ho. Database aur servers ko scale karke zyada load (zaida simultaneous logins, appointments) ko bardasht karna chahiye. Docker containerization se environment consistency aur

scalability asaan hoti hai.

- **Reliability (Bharosa mandi):** System ko 24/7 chalna chahiye aur downtime bohat kam hona chahiye. Regular backups liye jayenge aur failure recovery plan hoga takay data loss na ho. Koi component fail bhi ho to failover mechanism se operations chalti rahengi. Monitoring tools use karke system health check kiya jayega.

- **Usability (Istemaal mein Asaani):** Interface user-friendly honi chahiye takay kam training se log use kar saken. Saaf navigation, samajh mein aanay walay buttons, aur coherent workflow hona chahiye. Urdu ya local zubaan ka istemal agar users comfortable hon to support kiya ja sakta hai. Accessibility features (visually impaired ke liye fonts, colors) shamil hon. System multi-device (PC/mobile) friendly hoga takay mobile phones par bhi dashboard ya token display dekhne aasaan ho.

- **Performance (Raftar):** System ko jaldi response dena chahiye – jaise token generate karte hi receptionist turant dekh le, doctor apna schedule turant laa sake. Database queries optimize hongi. Server-side caching ya efficient coding se response time kam rakhna hai.

- **Maintainability (Iztimam-e-Nizaam):** Code ko modular banana hoga (jaise har module apna code) takay aasani se update ho sakay. Docker use karne se naya features add karna aur deploy karna asaan ho jata hai. Documentation aur comments se future me team k liye samajhna aasan ho.

- **Interoperability (Maahiri Tabadla):** System dusre healthcare systems ke saath data share kar sakay (agar zaroorat ho). Standard formats (HL7 jese) ko support karna faidemand hai jisse labs ya imaging departments se data integration hogi.

Yeh non-functional requirements aise aspects hain jo system ki core functionality ko supplement karte hain, jese ki **protection, stability, aur user experience** ko behtar banana.

Technology Stack aur System Architecture

Is Hospital Management System ko banane ke liye modern web tech stack use kiya ja sakta hai. Misal ke taur par:

- **Backend:** Node.js/Express ya Java/Spring Boot jese frameworks. Ye REST APIs provide karenge jin se frontend data fetch karega. Docker container me backend server chalaya jayega takay environment har jaga same ho. Backend me user authentication (JWT tokens) aur business logic (doctor scheduling, token algorithms) handle hogi. Database ke liye MySQL/PostgreSQL jese relational DB sahi rahenge, jahan user, doctor, token, medicine tables ban sakti hain.

- **Frontend:** React.js framework istemal kiya ja sakta hai (jaise GitHub example me React mention hai). React Router ke zariye multiple pages (Login, Dashboard, Profile, etc.) manage hongi. Axios jese libraries se backend APIs call karenge. Bootstrap ya Material UI jese UI libraries se attractive forms aur tables ban sakte hain. Frontend me har role ke views separate honge (Admin Panel,

Receptionist Panel, Doctor Panel, Pharmacist Panel).

- **Containerization:** Poora backend aur database ek ya zyada Docker containers me rakhe jayenge. Is se deployment asaan hota hai (kis bhi server par ek hi command se chalana). Microservices architecture bhi consider ho sakti hai: misal front-end container, backend container, aur DB container alag alag hote hain.

- **Authentication:** Secure login ke liye **JWT (JSON Web Tokens)** ya session-based auth use kiya ja sakta hai. Is se endpoints secure hongi aur sirf valid users hi data tak pahunch sakenge.

- **Deployment:** System ko initially local/server par Docker ke zariye deploy kiya ja sakta hai. Baad me scale karte hue AWS/Heroku jese cloud platforms use kiye ja sakte hain.

- **Other Tools:** Version control ke liye Git/GitHub, testing ke liye Postman (API testing) istamal hogi.

Ye tech choices industry standards ki misal hain. Achi architecture se system stable, secure aur scaleable bana rahega.

Modules aur Taqseem-e-Karkardagi

Aage chal kar poori system ki tafseeli documentation 8 modules me banai jayegi. Alag modules banane se development organized hota hai. In mein se kuch mumkin modules yeh ho sakte hain:

1. **Admin Module:** Doctors/Receptions/Staff management, fee settings, reports.

2. **Doctor Module:** Schedule management, patient list, prescriptions.

3. **Receptionist Module:** Patient assignment, token generation, payment processing.

4. **Pharmacy Module:** Medicine inventory, prescriptions fulfillment, billing.

5. **Ward Staff Module:** Duty roster, assigned tasks (agar isko separate consider karen).

6. **User Profile/Settings Module:** Login, profile edit, password change, access control.

7. **Dashboard & Reporting Module:** Role-specific dashboards, summary reports (optional).

8. **Authentication/Authorization Module:** Secure login, token validation (could be integrated with each).

Har module apne document ki demand karega jisme uske use cases, data schema, aur flow detail honge. In modules par development ke steps pehley plan kiye jayenge, phir code likha jayega.

Natija aur Taqreer

Is detailed document me Hospital Management System ke **functional aur non-functional requirements** wazeh taur par bayan kiye gaye hain. Ye guidance aam taur par kisi bhi modern HMS ke liye banai gayi sites aur best practices pe mabni hai. Hamare system me user roles, scheduling, token management aur pharmacy operations shamil hain, jabke sirf un components ko hataya gaya hai jo extra complexity barhate (jese patient attendance/inventory). Aglay marahil me is document ko 8 modules me divide karke har ek ke liye mukammal development documentation taiyar ki jayegi, taki implementation asaan ho.

Basic instructions

PART-1: HMS ko 8 Proper Modules mein Divide Karna

Main industry-style divide kar raha hoon taa-ke:

- modules **loosely coupled** hon
- integration baad mein asaan ho
- Antigravity / AI tools ko clear boundaries milen

🔹 MODULE-1: Authentication & Authorization Module

(System ki foundation)

Ye module kya karega?

- Login / Logout
- Role-based access control
- JWT / Session handling
- Password change / reset
- User identity verify karna

Kon use karega?

👉 **Sab users**

- Admin
- Doctor
- Receptionist

- Pharmacist
- Ward staff

📌 **Important note:**
Is module ke baghair koi aur module kaam nahi karega.

🔹 MODULE-2: Admin & System Configuration Module

Ye module kya karega?

- Doctors add / remove
- Receptionist add / remove
- Pharmacist add / remove
- Ward staff add / remove
- Doctor ki **fee set** karna
- Hospital basic settings
- System level control

Ye module kis ke paas hoga?

👉 **Sirf Admin**

📌 Admin ka kaam:

- System banana ❌
- **System control karna** ✅

🔹 MODULE-3: Doctor Management Module

Ye module kya karega?

- Doctor profile
- Doctor dashboard
- Doctor schedule (days / time / room)
- Doctor ke assigned tokens
- Doctor prescriptions likhna

Doctor kya NAHI karega?

❌ Fee set
❌ Users add/remove
❌ Pharmacy control

👉 sirf **medical flow**

🔹 MODULE-4: Receptionist & Token Management Module

Ye module kya karega?

- Patient basic record (lightweight)
- Doctor ke sath patient assign
- Token generate karna
- Room-based scheduling
- Payment entry (doctor fee)
- Token queue handle karna

📌 **Receptionist = system ka traffic controller**

🔹 MODULE-5: Patient Interaction (Light Module)

⚠️ tum ne kaha:

attendance system nahi chahiye

Is liye ye **light module** hoga.

Ye module kya karega?

- Patient basic profile
- Patient token history
- Doctor visit history
- Prescription view

❌ Attendance
❌ Full medical records

👉 sirf **supporting data**

🔹 MODULE-6: Pharmacy Management Module

Ye module kya karega?

- Medicines add/remove
- Batch number
- Expiry tracking
- Prescription based medicine issue
- Pharmacy billing
- Stock alerts

📌 Ye **pure tarah se separate system** hai
lekin doctor & patient se linked hai.

🔹 MODULE-7: Ward / Staff Management Module

Ye module kya karega?

- Ward staff profile
- Duty assignment
- Task list
- Staff dashboard

📌 Ye module optional lagta hai
lekin architecture mein rakhna **professional decision** hai.

🔹 MODULE-8: Dashboard, Reports & Integration Module

Ye module kya karega?

- Role-based dashboards
- Daily stats
- Token reports
- Income summary
- System overview

📌 Ye module **data consume karta hai**, create nahi.

PART-2: Frontend + Backend Folder Structure (VERY IMPORTANT)

Ye structure tum **Antigravity ko directly bol sakte ho**
aur students ko **strictly follow** karwa sakte ho.

📁 ROOT PROJECT STRUCTURE

```
hospital-management-system/
│
├── backend/
│   ├── auth/
│   ├── admin/
│   ├── doctor/
│   ├── receptionist/
│   ├── patient/
│   ├── pharmacy/
│   ├── ward/
│   ├── dashboard/
│   │
│   ├── common/
│   ├── config/
│   ├── database/
│   ├── docker/
│   └── main-server/
│
├── frontend/
│   ├── auth/
│   ├── admin/
│   ├── doctor/
│   ├── receptionist/
│   ├── patient/
│   ├── pharmacy/
│   ├── ward/
│   ├── dashboard/
│   │
│   ├── shared/
│   ├── assets/
│   └── main-app/
│
```

```
├── docs/
│   ├── 01-auth-module.md
│   ├── 02-admin-module.md
│   ├── 03-doctor-module.md
│   ├── 04-receptionist-module.md
│   ├── 05-patient-module.md
│   ├── 06-pharmacy-module.md
│   ├── 07-ward-module.md
│   └── 08-dashboard-module.md
│
└── README.md
```

PART-3: Student Rule (VERY IMPORTANT – ye tum doc mein daalna)

📌 **Golden Rule for Students:**

❗ Har group sirf **apne module ke folder** mein kaam karegi
❌ kisi aur module ko touch karna strictly forbidden

Example:

- Doctor group 👉 frontend/doctor + backend/doctor
- Pharmacy group 👉 frontend/pharmacy + backend/pharmacy
- Auth group 👉 frontend/auth + backend/auth

👉 Integration **sirf lead/instructor karega**

MODULE-1

MODULE-1

Authentication & Authorization Module

(HMS – Hospital Management System)

---

1. Module Overview (Simple Samajh)

Authentication module ka matlab hota hai:

> **“System ye confirm kare ke user kon hai**
> **aur usay kya karne ki ijazat hai.”**

Is HMS system mein **koi bhi user** bina login ke:

- dashboard nahi dekh sakta
- data access nahi kar sakta
- kisi module ko use nahi kar sakta

Is liye ye module **poore system ki foundation** hai.

---

2. Roles Jo Is Module Mein Handle Honge

System mein ye **roles** defined honge:

1. Admin
2. Doctor
3. Receptionist
4. Pharmacist

5. Ward Staff

📌 Har role ka:

- alag dashboard
- alag permissions
- alag access hoga

---

3. Authentication Kya Karega? (High Level)

Authentication module ye kaam karega:

- User login kar sake
- User logout kar sake
- User ka role verify ho
- Secure token generate ho
- Har request mein user verify ho

---

4. Authorization Kya Karega?

Authorization ka matlab hota hai:

> **“Is user ko is kaam ki ijazat hai ya nahi?”**

Example:

- Doctor → Admin panel access ❌
- Receptionist → Doctor add kare ❌

- Admin → Sab kuch access ✅

---

5. Functional Requirements

(System kya kya karega)

5.1 User Login

- User email/username + password enter kare
- System credentials verify kare
- Agar sahi hon:
    - token generate ho
    - user dashboard open ho

5.2 User Logout

- User manually logout kar sake
- Token invalidate ho jaye
- Session close ho jaye

5.3 Role-Based Access Control (RBAC)

- Har API request ke sath token check ho
- Token se role identify ho
- Agar role allowed hai → request allow

- Warna → access denied

5.4 Password Management

- First time login par password change option
- Password update kar sakta ho
- Password securely store ho (hashed)

5.5 Token Handling

- Login par token generate ho
- Token expiry defined ho
- Expired token par auto logout

---

6. Non-Functional Requirements

(System kaisa hona chahiye)

- Password plain text mein store nahi hoga
- Token secure hoga (JWT recommended)
- System scalable ho (multiple users)
- Response fast ho
- Unauthorized access prevent ho

---

7. Backend Scope (Auth Module)

Backend folder:

backend/auth/

Backend responsibilities:

- Login API
- Logout API
- Token generation
- Token verification middleware
- Role checking middleware

📌 Auth module **kisi aur module ka data change nahi karega**

---

8. Frontend Scope (Auth Module)

Frontend folder:

frontend/auth/

Frontend responsibilities:

- Login screen
- Logout button
- Forgot password UI
- Token store (local/session storage)
- Redirect based on role

---

9. Database Design (Auth Related)

Auth module ke liye **minimum tables**:

Users Table (Basic)

Field	Description
id	unique user id
name	full name
email	login email
password_hash	encrypted password
role	admin / doctor / receptionist / pharmacist
is_active	true / false
created_at	record creation

📌 Doctor / Receptionist details **alag modules mein** honge
Auth sirf identity handle karega.

---

10. Authentication Flow (Step by Step)

1. User login page open kare
2. Credentials enter kare
3. Backend verify kare
4. Token generate kare
5. Frontend token save kare
6. User role ke hisaab se dashboard open ho

---

11. Module Boundaries (IMPORTANT)

Auth module:

✅ Login
✅ Token
✅ Role verification

❌ Doctor profile edit
❌ Payments
❌ Pharmacy data
❌ Scheduling

---

12. Student Development Rules (Auth Module)

- Sirf auth folders touch karein
- Kisi aur module ke API modify na karein
- Role logic hardcode na karein
- Secure practices follow karein

---

13. Output of This Module

Is module ke complete hone ke baad:

- Secure login system ready hoga
- Roles defined honge
- Baqi modules is par depend kar sakte hain

---

14. Integration Note (For Instructor)

- Auth middleware har module ke backend mein reuse hoga

- Frontend mein central auth guard use hoga
- Token expiry global handle hogi

MODULE-2

MODULE-2

Admin & System Configuration Module

(HMS – Hospital Management System)

---

1. Module Overview (Simple Samajh)

Admin module HMS ka **control center** hota hai.

Admin system chalata nahi
Admin system ko control karta hai.

Is module ke zariye:

- Users add/remove honge
- Doctors ki fees set hongi
- Hospital ki basic settings manage hongi

📌 Ye module **sirf Admin role** ke liye hai.

---

2. Admin Role ki Authority

Admin ke paas ye powers hongi:

- Doctor add / remove
- Receptionist add / remove
- Pharmacist add / remove
- Ward staff add / remove

- Doctor ki consultation fee set karna
- System level settings manage karna

❌ Admin:

- Token generate nahi karega
- Prescription nahi likhega
- Pharmacy billing nahi karega

---

3. Functional Requirements

(Admin kya kya kar sakta hai)

3.1 User Management

Admin ye users manage karega:

- Doctor
- Receptionist
- Pharmacist
- Ward Staff

Admin kya karega:

- New user create
- User deactivate / remove
- User details update

📌 Password generation:

- First time password auto generate hoga
- User first login par password change karega

3.2 Doctor Fee Management

- Har doctor ki fee alag ho sakti hai
- Fee admin define karega
- Receptionist sirf **use** karegi, change nahi

Example:

- Dr. Ali → 1500 PKR
- Dr. Ahmad → 2500 PKR

3.3 Hospital Configuration

Admin ye cheezen set karega:

- Hospital name
- Working hours
- Token limit per doctor
- Room numbers
- System-wide rules

3.4 Role Assignment

- User create karte waqt role assign hoga
- Role change sirf Admin kar sakta hai

---

4. Non-Functional Requirements

- Sirf authenticated admin access
- Strong validation
- Audit-friendly changes
- Fast response
- Secure APIs

---

5. Backend Scope (Admin Module)

Backend folder:

backend/admin/

Backend responsibilities:

- Create user APIs
- Update user APIs
- Deactivate user APIs
- Doctor fee APIs
- Hospital settings APIs

📌 Admin module **auth middleware** use karega
📌 Direct database access allowed nahi

---

6. Frontend Scope (Admin Module)

Frontend folder:

frontend/admin/

Frontend responsibilities:

- Admin dashboard
- User management screens
- Doctor fee management UI
- System settings screens

---

7. Database Design (Admin Related)

Users Table (Extended)

Field	Description
id	unique user id
role	admin / doctor / receptionist / pharmacist
is_active	user active status
created_by	admin id

Doctor Fee Table

Field	Description
doctor_id	reference to doctor
fee_amount	consultation fee

updated_at	last update

Hospital Settings Table

Field	Description
key	setting name
value	setting value

---

8. Admin Dashboard Sections

Admin dashboard mein ye sections honge:

- Total users
- Active doctors
- System alerts
- Quick actions

---

9. Admin Flow (Step by Step)

1. Admin login kare
2. Admin dashboard open ho
3. Admin user create kare
4. Role assign kare
5. Doctor fee set kare
6. System settings save kare

---

10. Module Boundaries (IMPORTANT)

Admin module:

✅ User management
✅ Fee configuration
✅ System rules

❌ Token generation
❌ Appointment handling
❌ Pharmacy stock

---

11. Student Development Rules (Admin Module)

- Sirf admin folders touch karein
- Role validation zaroori
- Fee logic sirf yahin ho
- No hardcoded users

---

12. Output of This Module

Is module ke complete hone ke baad:

- System controllable ho jayega
- Users manageable honge
- Doctor fees centralized hongi

---

13. Integration Note (For Instructor)

- Admin APIs sab modules se linked hongi
- Doctor fee receptionist module use karega
- Role-based guard strictly enforced hoga

MODULE-3

MODULE-3

Doctor Management & Clinical Workflow

(HMS – Hospital Management System)

---

1. Module Overview (Simple Samajh)

Doctor module HMS ka **clinical core** hota hai.

Doctor ka kaam:
patients ko consult karna,
prescription likhna,
aur apni schedule follow karna.

Doctor:

- system control nahi karta
- users add/remove nahi karta
- fees set nahi karta

📌 Ye module **sirf Doctor role** ke liye hota hai.

---

2. Doctor Role Responsibilities

Doctor ye kaam karega:

- Apni profile dekhna / edit karna
- Apni daily schedule dekhna
- Assigned tokens dekhna

- Patient consult karna
- Prescription likhna
- Visit complete mark karna

---

3. Functional Requirements

(Doctor kya kya kar sakta hai)

3.1 Doctor Profile Management

Doctor apni profile mein ye update kar sakta hai:

- Name
- Contact info
- Specialization
- Profile photo
- Password

❌ Fee change nahi kar sakta
❌ Role change nahi kar sakta

3.2 Doctor Dashboard

Doctor dashboard mein ye dikhna chahiye:

- Aaj ke total tokens
- Current token number
- Waiting patients

- Completed visits

3.3 Schedule Viewing

Doctor:

- Apni working days dekh sakta hai
- Apni time slots dekh sakta hai
- Room number dekh sakta hai

📌 Schedule admin define karta hai.

3.4 Token & Patient Queue

- Doctor sirf **assigned tokens** dekh sakta hai
- Token order change nahi kar sakta
- Queue read-only hogi

3.5 Patient Consultation

Doctor consultation ke dauran:

- Patient basic info dekhe
- Symptoms likhe
- Diagnosis likhe
- Notes add kare

3.6 Prescription Management

Doctor:

- Medicines prescribe kare
- Dosage likhe
- Duration likhe

📌 Prescription pharmacy module ke sath linked hogi.

3.7 Visit Completion

- Consultation complete mark kare
- Token close ho jaye
- Record history mein save ho

---

4. Non-Functional Requirements

- Fast dashboard load
- No data leakage
- Patient data confidentiality
- Secure prescription handling
- Role-based access strictly enforced

---

5. Backend Scope (Doctor Module)

Backend folder:

backend/doctor/

Backend responsibilities:

- Doctor profile APIs
- Schedule fetch APIs
- Token list APIs
- Consultation APIs
- Prescription APIs

📌 Doctor module:

- Auth middleware use karega
- Pharmacy & receptionist modules se linked hoga

---

6. Frontend Scope (Doctor Module)

Frontend folder:

frontend/doctor/

Frontend responsibilities:

- Doctor dashboard UI
- Profile screen
- Token list view
- Consultation form
- Prescription UI

---

7. Database Design (Doctor Related)

Doctor Profile Table

Field	Description
doctor_id	reference to user
specialization	doctor field
room_number	assigned room
created_at	record time

Consultation Table

Field	Description
consultation_id	unique id
doctor_id	doctor
patient_id	patient
token_no	token
notes	consultation notes
created_at	time

Prescription Table

Field	Description
prescription_id	unique id
consultation_id	linked visit
medicine_name	medicine
dosage	how to take
duration	days

---

8. Doctor Workflow (Step by Step)

1. Doctor login kare
2. Dashboard open ho
3. Token queue dekhe
4. Patient call kare
5. Consultation kare
6. Prescription likhe
7. Visit complete kare

---

9. Module Boundaries (IMPORTANT)

Doctor module:

✅ Consultation
✅ Prescription
✅ Profile management

❌ Token creation
❌ Fee handling
❌ User management

---

10. Student Development Rules (Doctor Module)

- Sirf doctor folders edit karein
- Prescription format consistent rakhein
- No hardcoded data

- Patient data privacy ka khayal rakhein

---

11. Output of This Module

Is module ke complete hone ke baad:

- Doctors system par consult kar sakenge
- Prescription flow complete ho jayega
- Pharmacy integration possible ho jayegi

---

12. Integration Note (For Instructor)

- Tokens receptionist module se aayenge
- Prescriptions pharmacy module consume karega
- Admin fee config yahan reflect nahi hogi

MODULE-4

MODULE-4

Receptionist & Token Management Module

(HMS – Hospital Management System)

---

1. Module Overview (Simple Samajh)

Receptionist module HMS ka **front-desk brain** hota hai.

> **Receptionist patient ko system ke andar laati hai,**
> **doctor ke sath connect karti hai,**
> **aur token + payment manage karti hai.**

Receptionist:

- system control nahi karti
- doctor fee set nahi karti
- pharmacy manage nahi karti

📌 Ye module **sirf Receptionist role** ke liye hai.

---

2. Receptionist Role Responsibilities

Receptionist ye kaam karegi:

- Patient ka basic record banana
- Doctor ke sath patient assign karna
- Token generate karna
- Room & schedule ke hisaab se token dena

- Doctor fee collect karna
- Token queue manage karna

---

3. Functional Requirements

(Receptionist kya kya kar sakti hai)

3.1 Patient Registration (Basic)

Receptionist:

- New patient ka basic data enter karegi
- Existing patient ko search karegi

Patient data limited hoga:

- Name
- Contact
- Age / Gender

📌 Full medical history nahi banegi.

3.2 Doctor Assignment

- Receptionist kisi bhi available doctor ko select kar sakti hai
- Doctor ki schedule & room check hoga
- Doctor ki fee auto fetch hogi (Admin set ki hui)

3.3 Token Generation System

Token system ke rules:

- Har doctor ke tokens alag honge
- Token number auto generate hoga
- Token schedule ke mutabiq hoga
- Token room number ke sath linked hoga

Example:

- Dr. Ali – Room 3 – Token #12

3.4 Payment Entry

Receptionist:

- Doctor ki consultation fee enter karegi
- Payment mode select karegi
- Receipt generate karegi

📌 Fee edit karna ❌
📌 Discount dena ❌ (unless admin allow kare)

3.5 Token Queue Management

Receptionist:

- Current token status dekh sakti hai
- Completed tokens mark nahi karegi
- Token order change nahi karegi

---

4. Non-Functional Requirements

- Fast token generation
- No duplicate tokens
- Fee tampering prevent ho
- Data consistency
- Role-based access enforced

---

5. Backend Scope (Receptionist Module)

Backend folder:

backend/receptionist/

Backend responsibilities:

- Patient create/search APIs
- Doctor availability APIs
- Token generation APIs
- Payment record APIs

📌 Doctor schedule & fee APIs read-only hongi.

---

6. Frontend Scope (Receptionist Module)

Frontend folder:

frontend/receptionist/

Frontend responsibilities:

- Patient registration UI
- Doctor selection UI
- Token generation screen
- Payment entry screen
- Token queue view

---

7. Database Design (Receptionist Related)

Patient Table (Basic)

Field	Description
patient_id	unique id
name	patient name
contact	phone
gender	male/female
created_at	time

Token Table

Field	Description
token_id	unique token
doctor_id	assigned doctor
patient_id	patient
room_number	doctor room

status	waiting / completed
created_at	time

Payment Table

Field	Description
payment_id	unique id
token_id	related token
amount	fee
payment_mode	cash/card
created_at	time

---

8. Receptionist Workflow (Step by Step)

1. Receptionist login kare
2. Patient search ya register kare
3. Doctor select kare
4. Token generate kare
5. Fee collect kare
6. Token queue update ho

---

9. Module Boundaries (IMPORTANT)

Receptionist module:

✅ Token generation
✅ Patient assignment
✅ Payment entry

❌ Doctor consultation
❌ Fee configuration
❌ Pharmacy handling

---

10. Student Development Rules (Receptionist Module)

- Sirf receptionist folders edit karein
- Token logic yahin rahe
- Fee value backend se fetch ho
- No manual overrides

---

11. Output of This Module

Is module ke complete hone ke baad:

- Patient flow smooth ho jayega
- Doctor queue properly manage hogi
- Payment records maintain honge

---

12. Integration Note (For Instructor)

- Doctor module token consume karega
- Admin module fee set karega
- Dashboard module reports banayega

MODULE-5

MODULE-5

Patient Interaction & History Module

(HMS – Hospital Management System)

---

1. Module Overview (Simple Samajh)

Patient module HMS ka **supporting module** hota hai.

Patient system ka center nahi
lekin system ka important part hota hai.

Is HMS mein patient:

- system control nahi karega
- login heavy features nahi honge
- sirf apni information aur history dekhega

📌 Ye module **lightweight** rakha gaya hai by design.

---

2. Patient Scope (Kya Included Hai / Kya Nahi)

✅ Included

- Patient basic profile
- Token history
- Doctor visit history
- Prescription view

❌ Not Included

- Attendance system
- Full medical record system
- Lab test management
- Appointment booking by patient

---

3. Functional Requirements

(Patient kya kya dekh sakta hai)

3.1 Patient Profile

Patient profile mein ye cheezen hongi:

- Name
- Contact number
- Age / Gender

📌 Patient:

- Profile edit nahi karega
- Data receptionist ke zariye update hoga

3.2 Token History

Patient apni:

- Past tokens
- Token dates
- Assigned doctors

dekh sakta hai.

3.3 Visit History

Patient:

- Doctor visit history dekhega
- Consultation notes read-only hongi

3.4 Prescription View

Patient:

- Apni prescriptions dekh sakta hai
- Medicines ka naam
- Dosage
- Duration

📌 Prescription edit ❌

---

4. Non-Functional Requirements

- Patient data read-only
- Privacy maintained
- Fast loading views
- Unauthorized access blocked

---

5. Backend Scope (Patient Module)

Backend folder:

backend/patient/

Backend responsibilities:

- Patient profile fetch APIs
- Token history APIs
- Visit history APIs
- Prescription fetch APIs

📌 Is module mein:

- Create/update logic nahi hoga
- Sirf read operations honge

---

6. Frontend Scope (Patient Module)

Frontend folder:

frontend/patient/

Frontend responsibilities:

- Patient dashboard
- History screens
- Prescription viewer

---

7. Database Design (Patient Related)

Patient ka data **multiple modules se aata hai**
is module ka koi heavy table nahi hoga.

Patient Reference Table

Field	Description
patient_id	unique id
name	patient name
contact	phone
created_at	time

History Views (Derived Data)

- Tokens → Receptionist module
- Visits → Doctor module
- Prescriptions → Doctor module

📌 Patient module **sirf read karta hai**.

---

8. Patient Workflow (Step by Step)

1. Patient profile open hota hai
2. Token history dikhti hai
3. Visit history available hoti hai

4. Prescription view ho jati hai

---

9. Module Boundaries (IMPORTANT)

Patient module:

✅ History view
✅ Prescription view

❌ Token creation
❌ Consultation
❌ Pharmacy billing

---

10. Student Development Rules (Patient Module)

- Sirf patient folders edit karein
- No write APIs
- No business logic duplication
- Security checks enforce karein

---

11. Output of This Module

Is module ke complete hone ke baad:

- Patients apna record dekh sakenge
- Transparency improve hogi
- Doctor–patient linkage clear hogi

---

12. Integration Note (For Instructor)

- Data receptionist & doctor modules se aata hai
- Auth module role validation karega
- Dashboard module stats banayega

MODULE-6

MODULE-6

Pharmacy Management Module

(HMS – Hospital Management System)

---

1. Module Overview (Simple Words)

Pharmacy module ka kaam hota hai:

Doctor ki prescription ko medicines mein convert karna
aur medicine issue + record maintain karna

Ye module:

- doctor aur patient ke darmiyan bridge hai
- billing se linked hota hai
- inventory ko basic level par manage karta hai

📌 Ye **hospital pharmacy** ke liye hai,
koi full commercial medical store system nahi.

---

2. Pharmacy Scope (Included / Not Included)

✅ Included

- Prescription receive karna
- Medicines issue karna
- Basic medicine stock

- Pharmacy billing (simple)

❌ Not Included

- Supplier management
- Advanced inventory forecasting
- Insurance claims
- GST / tax system

---

3. Functional Requirements

3.1 Prescription Receiving

Pharmacist:

- Doctor ki prescription dekh sakta hai
- Patient name & token number verify karta hai

Prescription:

- Read-only hoti hai
- Doctor ke module se aati hai

3.2 Medicine Issuing

Pharmacist:

- Prescription ke mutabiq medicines issue karta hai

- Quantity confirm karta hai
- Stock check karta hai

📌 Agar stock available na ho:

- Partial issue allowed
- Record maintain hota hai

3.3 Pharmacy Billing

Billing features:

- Medicine name
- Quantity
- Price per unit
- Total amount

📌 Simple billing:

- Cash based
- No insurance logic

3.4 Stock Management (Basic)

Pharmacy:

- Medicine list dekh sakta hai
- Available quantity check hoti hai

Stock update:

- Medicine issue hone par quantity kam hoti hai

📌 Manual stock add admin ya pharmacist karega.

---

4. Non-Functional Requirements

- Accurate stock calculation
- Prescription tampering prohibited
- Fast billing process
- Secure pharmacist access

---

5. Backend Scope (Pharmacy Module)

Backend folder:

backend/pharmacy/

Backend responsibilities:

- Fetch prescriptions
- Issue medicines
- Update stock
- Generate pharmacy bill

📌 Doctor prescription **modify nahi hogi**.

---

6. Frontend Scope (Pharmacy Module)

Frontend folder:

frontend/pharmacy/

Frontend screens:

- Prescription list
- Issue medicine screen
- Billing screen
- Medicine stock view

---

7. Database Design (Pharmacy Related)

Medicine Table

Field	Description
medicine_id	unique id
name	medicine name
quantity	available stock
price	per unit
updated_at	last update

Pharmacy Bill Table

Field	Description
bill_id	unique id
patient_id	reference
total_amount	bill total

issued_at	date

Issued Medicines Table

Field	Description
bill_id	reference
medicine_id	medicine
quantity	issued

---

8. Pharmacy Workflow (Step by Step)

1. Pharmacist login karta hai
2. Prescription open karta hai
3. Medicines verify karta hai
4. Stock check hota hai
5. Medicines issue hoti hain
6. Bill generate hota hai

---

9. Module Boundaries (IMPORTANT)

Pharmacy module:

✅ Medicine issuing
✅ Stock deduction
✅ Pharmacy billing

❌ Doctor consultation
❌ Prescription writing
❌ Patient registration

---

10. Student Development Rules

- Prescription read-only rakhein
- Negative stock allow na karein
- Billing calculation backend se ho
- Role-based access enforce karein

---

11. Output of This Module

Is module ke complete hone ke baad:

- Pharmacy workflow smooth hoga
- Prescription errors kam honge
- Medicine stock track hoga
- Patient billing clear hogi

---

12. Integration Notes

- Prescription → Doctor Module
- Patient info → Patient Module
- Payments → Billing Module
- Auth → Role validation

MODULE-7

MODULE-7

Ward & Room Management Module

(HMS – Hospital Management System)

---

1. Module Overview

Ward & Room Management module ka kaam hota hai:

Hospital ke rooms / wards ko manage karna
aur patients ko rooms assign karna

Ye module:

- indoor patients (admitted patients) ke liye hota hai
- bed availability track karta hai
- billing module ke sath linked hota hai

📌 OPD patients (sirf consultation wale) is module mein **nahi aate**.

---

2. Ward / Room Scope

✅ Included

- Wards & rooms define karna
- Beds availability track karna
- Patient ko room/bed assign karna
- Discharge par bed free karna

❌ Not Included

- Nurse duty scheduling
- ICU medical equipment tracking
- Housekeeping workflow

---

3. Functional Requirements

3.1 Ward Management

Admin ya authorized staff:

- New ward add kar sakta hai
- Ward type set kar sakta hai

Ward types example:

- General Ward
- Semi-Private
- Private Room

3.2 Room & Bed Management

System mein:

- Har ward ke andar multiple rooms
- Har room ke andar multiple beds

For each bed:

- Bed number
- Status (Available / Occupied)

📌 Ek bed par aik time par sirf **1 patient** assign ho sakta hai.

3.3 Patient Admission

Admission process:

- Patient already registered hona chahiye
- Doctor admission recommend kare

During admission:

- Ward select hota hai
- Room select hota hai
- Bed assign hota hai

📌 Bed availability auto-check hogi.

3.4 Patient Discharge

Discharge par:

- Bed status → Available
- Stay duration calculate hoti hai
- Billing module ko stay data send hota hai

3.5 Bed Availability View

Staff:

- Real-time bed availability dekh sakta hai
- Ward-wise filter laga sakta hai

---

4. Non-Functional Requirements

- Real-time updates
- Data consistency (no double booking)
- Fast room assignment
- Role-based access

---

5. Backend Scope (Ward Module)

Backend folder:

backend/ward/

Backend responsibilities:

- Ward CRUD
- Room & bed management
- Admission & discharge handling
- Billing integration data

---

6. Frontend Scope (Ward Module)

Frontend folder:

frontend/ward/

Frontend screens:

- Ward list
- Room & bed layout view
- Patient admission screen
- Discharge screen

---

7. Database Design (Ward Related)

Ward Table

Field	Description
ward_id	unique id
name	ward name
type	general/private
created_at	date

Room Table

Field	Description
room_id	unique id
ward_id	reference
room_number	room no

Bed Table

Field	Description
bed_id	unique id
room_id	reference
bed_number	bed no
status	available / occupied

Admission Table

Field	Description
admission_id	unique id
patient_id	reference
bed_id	assigned bed
admitted_at	date
discharged_at	date

---

8. Ward Workflow (Step by Step)

1. Patient OPD se admission recommend hota hai
2. Staff ward select karta hai
3. Available bed assign hota hai
4. Patient admitted status mein jata hai
5. Discharge par bed free hota hai

---

9. Module Boundaries

Ward module:

✅ Admission & discharge
✅ Bed availability
✅ Stay tracking

❌ Doctor treatment
❌ Pharmacy handling
❌ Payment collection

---

10. Student Development Rules

- Bed assignment atomic rakhein
- Discharge ke bina bed free na ho
- Billing ko sirf stay duration send karein
- Hard delete avoid karein

---

11. Output of This Module

Is module ke baad:

- Hospital occupancy clear hogi
- Overbooking issues nahi honge
- Billing accurate hogi
- Patient admission smooth hogi

---

12. Integration Notes

- Patient → Patient Module

- Admission order → Doctor Module
- Stay cost → Billing Module
- Auth → Role management

MODULE-8

MODULE-8

Billing & Payment Management Module

(Hospital Management System – HMS)

---

1. Module Overview

Billing & Payment module ka kaam hai:

> **Hospital ki saari services ka paisa calculate karna**
> **aur patient se payment receive karna**

Ye module:

- Doctor consultation fee
- Pharmacy medicines
- Ward / bed charges

sab ko **combine karke final bill** banata hai.

📌 **Ye module kisi service ko create nahi karta**,
sirf **baaki modules se data leta hai**.

---

2. Billing Scope

✅ Included

- Bill generation
- Item-wise charges
- Partial payments

- Payment history

❌ Not Included

- Insurance claims
- Online payment gateways
- Refund automation

---

3. Billing Sources (Integration)

Billing module data leta hai:

Module	Data
Doctor Module	Consultation fee
Reception Module	Token & visit
Pharmacy Module	Medicines
Ward Module	Bed stay charges

---

4. Functional Requirements

4.1 Bill Creation

System automatically bill create kare jab:

- Consultation complete ho
- Medicine issue ho
- Patient discharge ho

Bill mein include:

- Date
- Patient info
- Service list
- Total amount

4.2 Bill Items (Line Items)

Har bill ke andar multiple items hotay hain:

Item Type	Example
Consultation	Dr. fee
Medicine	Paracetamol
Room	Bed charges

Har item ke sath:

- Quantity
- Unit price
- Sub-total

4.3 Doctor Fee Handling

- Har doctor ki fee **admin set karega**
- Reception token generate karte waqt fee attach hogi
- Fee bill mein auto add hogi

4.4 Ward Charges Calculation

Ward module se:

- Admit date
- Discharge date

Billing:

- Stay duration calculate kare
- Per-day charges apply kare

4.5 Pharmacy Billing

Pharmacy se:

- Issued medicines
- Quantity

Billing:

- Medicine-wise cost calculate kare

4.6 Payments Handling

Receptionist:

- Cash payment receive kar sakti hai
- Partial payment accept kar sakti hai

Payment types:

- Cash
- Card (manual entry)

📌 Online gateway future scope mein hai.

4.7 Bill Status

Bill statuses:

- Unpaid
- Partially Paid
- Paid

4.8 Receipt Generation

System:

- Printable receipt generate kare
- Payment history show kare

---

5. Non-Functional Requirements

- Accurate calculations
- No data duplication
- Transaction consistency
- Fast response time
- Role-based access

---

6. Backend Scope (Billing)

Backend folder:

backend/billing/

Backend responsibilities:

- Bill generation
- Line item management
- Payment records
- Integration APIs

---

7. Frontend Scope (Billing)

Frontend folder:

frontend/billing/

Frontend screens:

- Patient billing view
- Payment entry screen
- Receipt preview
- Billing history

---

8. Database Design (Billing)

Bill Table

Field	Description
bill_id	unique id
patient_id	reference
total_amount	total
status	paid/unpaid
created_at	date

Bill Items Table

Field	Description
item_id	unique id
bill_id	reference
type	consultation / medicine
amount	price

Payment Table

Field	Description
payment_id	unique id
bill_id	reference
amount	paid
method	cash/card
paid_at	date

---

9. Billing Workflow (End-to-End)

1. Patient visit complete hoti hai

2. Doctor fee add hoti hai
3. Pharmacy medicines add hoti hain
4. Ward stay charges add hotay hain
5. Final bill generate hota hai
6. Payment receive hoti hai
7. Receipt generate hoti hai

---

10. Module Boundaries

Billing module:

✅ Calculate & collect payment
✅ Maintain payment history
✅ Generate receipts

❌ Medical decisions
❌ Inventory control
❌ Appointment scheduling

---

11. Student Development Rules

- Direct amount edit allowed na ho
- Data sirf modules se aaye
- Partial payment logic correct ho
- Bill delete nahi, sirf cancel ho

---

12. Output of This Module

Is module ke baad:

- Hospital ka complete financial flow clear
- Errors kam honge
- Patient trust improve hoga
- System fully integrated ho jayega

---

13. Final System Integration Summary

Module	Status
Auth & Roles	Integrated
Patient	Integrated
Doctor	Integrated
Reception	Integrated
Pharmacy	Integrated
Ward	Integrated
Billing	FINAL

---

14. Project Completion Note

> **Billing module ke baad HMS functional complete mana jayega.**

Future improvements:

- Online payments
- Insurance system
- Reports & analytics