Great—main file mil gaya. Neechay **asaan alfaaz** me ek **step-by-step roadmap** de raha hoon: kahan se start karein, kya banayen, kis order me, aur har phase me clear deliverables. Roadmap SRS ke features ko follow karta hai (user/staff roles, rooms, bookings, billing, housekeeping, reports, security, docs, etc.).

**Roadmap (MERN Hotel Management System)**

**0) Kick-off (½–1 din)**

* **Goal set karein (MVP):** Pehla version me yeh features fix karein: **Login/roles**, **Rooms inventory + status**, **Reservations + Check-in/out**, **Billing/Invoice**. (Yahi SRS ke core use-cases hain.)
* **Tech setup (MERN):** Node + Express, MongoDB + Mongoose, React + Vite/CRA, JWT auth, RBAC middleware. SRS bhi **role-based access** demand karta hai.
* **Repo structure:** /client (React), /server (Express), .env, README.

**Phase 1) Auth & Roles (Admin/Staff/Guest)**

**Why first?** Baqi sab features role-based hain. SRS me **Admin panel + roles/permissions** clear mention hai; security me **secure login + RBAC** bhi.   
**Deliverables:**

* Sign-in (JWT), Logout, Password hash.
* Roles: admin, manager, receptionist, housekeeping, guest. (SRS: staff profiles/guest profiles + role-based access. )
* Admin dashboard (basic shell).

**Acceptance test:** Admin staff create/disable kar sakta hai; non-admin restricted areas nahi dekh sakta.

**Phase 2) Rooms Inventory & Status**

**SRS match:** **Room types, availability, status (cleaning/occupied/available), pricing**. Real-time status updates required.   
**Deliverables:**

* Collections/Models: RoomType, Room (no., type, price, status).
* UI: Room list + filters; detail page; Admin me create/edit.
* API: GET /rooms, PATCH /rooms/:id (status), POST /room-types.

**Acceptance test:** Receptionist room list dekh kar status update kar sakta hai; price/type admin set karta hai.

**Phase 3) Reservations + Check-in/Check-out**

**SRS match:** **Online/staff bookings**, **check-in/out flow**, **allocation**, **confirmation**.   
**Deliverables:**

* Model: Booking (guest, room, dates, status: reserved/checked-in/checked-out).
* Availability check (date range overlap).
* Flows:
  + **Reservation:** select dates → suggest available rooms → hold/confirm.
  + **Check-in:** assign room, issue key (logical), status occupied.
  + **Check-out:** free room, generate bill trigger.
* Notifications (basic): booking created/changed (SRS: system notifications).

**Acceptance test:** Overlapping booking reject hoti hai; check-out par room vacant/cleaning me chala jata hai.

**Phase 4) Billing & Invoicing**

**SRS match:** **Room rate + add-on services (food, laundry…)**, **print/email invoice**.   
**Deliverables:**

* Model: Invoice + InvoiceItem (room nights, services, taxes).
* Auto-calc: stay duration × rate, plus items.
* UI: invoice view → PDF/print; email stub.
* Admin settings: taxes/policies from System Settings.

**Acceptance test:** Correct totals & tax; invoice link printable.

**Phase 5) Housekeeping & Maintenance**

**SRS match:** **Housekeeping tasks**, mark **completed**, **maintenance requests** from guest/staff + track resolution.   
**Deliverables:**

* Models: HKTask (room, task, due, status), MaintenanceTicket (issue, room, priority, status).
* UI: Housekeeping dashboard: today’s tasks; mark done → room status auto-update.
* Guest/staff: create maintenance tickets; staff resolve + notes.

**Acceptance test:** Check-out pe auto “cleaning” task banta hai; resolve par room “available” hota hai.

**Phase 6) Guest Services & Feedback**

**SRS match:** **Feedback & ratings**, **additional services** (room service, wake-up call, transport).   
**Deliverables:**

* Models: Feedback (rating, comments, booking), ServiceRequest (type, status, charges?).
* UI: Guest portal: request services, leave feedback after check-out.

**Acceptance test:** Feedback submit after stay; service requests appear in staff queue.

**Phase 7) Reporting & Analytics**

**SRS match:** **Occupancy, revenue, feedback summaries**, basic **analytics** (demand forecast/pricing ideas). Start with reports; analytics later.   
**Deliverables:**

* Reports: Occupancy by date range, Revenue by source, Popular room types, Avg rating.
* Export CSV/PDF; simple charts on admin dashboard.

**Acceptance test:** Date filters work; totals match invoices/bookings.

**Phase 8) Non-Functional: Security, Privacy, Performance, Reliability**

**SRS match:**

* **Security:** encryption (at rest+in transit), robust auth, strict authorization.
* **Privacy:** GDPR-style consent & data handling.
* **Performance:** ~1–2s response, hundreds of concurrent users, horizontal scaling.
* **Reliability:** ~99% uptime target, automated backups, logging/monitoring.

**Practical steps (short):**

* HTTPS; strong CORS; helmet; rate-limit; input validation.
* RBAC guards on every route; Mongoose row-level checks.
* Indexes on Room.status, Booking date ranges.
* Winston/Morgan logs; error tracking; health checks; nightly Mongo backups.

**Phase 9) QA, Testing & Docs (+ Demo Video)**

**SRS match:** **Unit/integration/E2E tests**, **security testing**, **user & developer docs**, **working video**.   
**Deliverables:**

* Tests: Jest (backend), React Testing Library (frontend), Cypress (E2E).
* Security check: auth bypass, IDOR, SQLi-style (Mongo) injection, weak JWT.
* Docs: User guide (how to book, check-in/out, invoice), Dev guide (setup, env, scripts).
* **Short demo video** showing complete flow.

**Simple Data Model (start small)**

* **User**: name, email, role, status (active/disabled), passwordHash. (SRS: staff & guest profiles.)
* **RoomType**: name, basePrice, capacity, amenities.
* **Room**: number, typeId, status (available/occupied/cleaning/maintenance), currentPrice.
* **Booking**: guestId, roomId, checkIn, checkOut, status, totalNights.
* **Invoice**: bookingId, items [], taxes, total, issuedAt, emailedTo.
* **HKTask**: roomId, task, due Date, status. **MaintenanceTicket**: roomId, issue, priority, status.
* **ServiceRequest**: bookingId, type (room-service/wake-up/transport), charge, status. **Feedback**: bookingId, rating, comment.

**Minimal Screens & APIs (MVP)**

**Screens (React):**

* Login • Admin Dashboard • Staff Dashboard (Reception/Housekeeping) • Rooms • Bookings • Check-in/out • Invoices • Settings (taxes/policies).

**Key APIs (Express):**

* /auth/login, /auth/logout
* /Users (admin only)
* /Rooms, /room-types
* /Bookings (create/availability/update status)
* /Invoices (generate/print)
* /housekeeping/tasks, /maintenance/tickets
* /reports/occupancy revenue  
  (Guard all routes per **RBAC**.)

**Kasie shore karein? (1-2 din ka starter plan)**

1. **Project skeleton banal:** MERN folders, Slant/Prettier, ends.
2. **Auth first:** /auth/login + JWT + require Role () middleware. (SRS security)
3. **Rooms CRUD:** seed 10–20 rooms + statuses; React list page.
4. **Booking flow:** date-picker → availability API → confirm → check-in/out buttons.
5. **Invoice generate:** on check-out create printable invoice.

Is point take **MVP workable** ho jata hai; phi housekeeping/maintenance, feedback/services, reports add karate Jayne—exact SRS order me.

**Submission-friendly checklist**

* ✅ Features map SRS se: Users/Roles, Rooms, Booking + Check-in/out, Billing, Housekeeping/Maintenance, Feedback/Services, Reports, Notifications, Settings.
* ✅ Non-functional: Security (auth/RBAC/encryption), Privacy/Consent, Performance, Uptime/Backups, Logging/Monitoring.
* ✅ QA: Unit/Integration/E2E + Security testing.
* ✅ Docs + **working demo video**.