**একটি পূর্ণাঙ্গ Real Estate Management System-এ যা যা থাকা উচিত:**

**🔹 ১. প্রপার্টি ম্যানেজমেন্ট (Property Management)**

* নতুন প্রপার্টি যুক্ত/এডিট/ডিলিট করা
* প্রপার্টির ধরন: জমি, ফ্ল্যাট, অফিস, বাণিজ্যিক স্পেস
* অবস্থান (Location), সাইজ, ফ্লোর প্ল্যান, ছবি, দাম
* প্রপার্টির স্ট্যাটাস: Available, Booked, Sold, Under Construction

**🔹 ২. ক্লায়েন্ট ম্যানেজমেন্ট (Client Management)**

* গ্রাহকের নাম, মোবাইল, ইমেইল, NID
* কোন ফ্ল্যাট বুক করেছে, কবে করেছে
* ডকুমেন্ট আপলোড (NID, Booking Form)
* ক্লায়েন্টের লগইন অপশন (স্ট্যাটাস দেখার জন্য)

**🔹 ৩. বিক্রয় ও বুকিং মডিউল (Sales & Booking)**

* বুকিং ফর্ম (অনলাইন বা অফিস থেকে)
* বুকিং ফি ইনপুট
* ইনভয়েস জেনারেট
* বুকিং কনফার্ম/ক্যানসেল
* বুকিং সময়সীমা ও কিস্তির শিডিউল তৈরি

**🔹 ৪. পেমেন্ট ও কিস্তি ট্র্যাকিং (Installment & Payment System)**

* কিস্তির সময়সূচি (Monthly/Quarterly)
* পেমেন্ট ট্র্যাকিং (তারিখ, পরিমাণ, মোড)
* কিস্তি বাকি থাকলে অ্যালার্ট
* পেমেন্ট রিসিট ডাউনলোড/প্রিন্ট

**🔹 ৫. প্রজেক্ট ম্যানেজমেন্ট (Project Management)**

* নতুন প্রজেক্ট যুক্ত ও অগ্রগতি ট্র্যাক
* Estimated Budget, Actual Cost
* Timeline ও Deadline
* কনস্ট্রাকশন ফেজ আপডেট

**🔹 ৬. একাউন্টিং ও রিপোর্টিং (Accounting & Reports)**

* আয়-ব্যয় রিপোর্ট (Income/Expense)
* কাস্টমার রিপোর্ট (কে কত টাকা জমা দিয়েছে/বাকি আছে)
* বিক্রয় রিপোর্ট (Daily/Monthly/Project Wise)
* Downloadable PDF/Excel রিপোর্ট

**🔹 ৭. ইউজার রোল ও পারমিশন (User Roles & Access Control)**

* Admin (সব দেখতে ও কন্ট্রোল)
* Sales Manager (বিক্রয় ও ক্লায়েন্ট হ্যান্ডলিং)
* Accountant (ফিন্যান্স সেকশন)
* Site Engineer (প্রজেক্ট আপডেট)

**🔹 ৮. ডকুমেন্ট ম্যানেজমেন্ট**

* জমি দলিল, অনুমোদনপত্র, RAJUK কাগজ স্ক্যান করে সংরক্ষণ
* ক্লায়েন্ট ডকুমেন্ট (NID, Photo, Agreement)
* সার্চ সিস্টেমসহ ডেটাবেস

**🔹 ৯. SMS/Email Integration**

* বুকিং কনফার্ম হলে SMS পাঠানো
* কিস্তির ডেট মনে করিয়ে দেওয়া
* প্রমোশনাল অফার পাঠানো

**🔹 ১০. অনলাইন পেমেন্ট গেটওয়ে (Optional)**

* SSLCommerz, bKash, Nagad ইন্টিগ্রেশন
* অনলাইনে বুকিং মানি/কিস্তি পরিশোধ
* ট্রান্সাকশন হিস্টোরি সংরক্ষণ

**🔹 ১১. ওয়েবসাইট/ফ্রন্টএন্ড ইন্টিগ্রেশন (Client View)**

* ফ্ল্যাট বা প্রপার্টি ব্রাউজ করা
* বুকিং ফর্ম পূরণ
* লগইন করে কিস্তির হিসাব দেখার ব্যবস্থা
* ব্লগ/নিউজ সেকশন

**🎯 পরিকল্পনা ধাপসমূহ:**

1. ✅ ইউজার ফ্লো (User Flow)
2. ✅ ড্যাশবোর্ড লে-আউট (Dashboard Layout)
3. ✅ ডেটাবেজ স্ট্রাকচার (Database Schema Idea)
4. ✅ ফিচার ওয়্যারফ্রেম (Feature Breakdown)
5. ✅ কোড স্ট্রাকচার প্ল্যান (Frontend + Backend)

**🧭 ১. ইউজার ফ্লো (User Flow)**

mermaid

CopyEdit

graph TD

A[Login Page] --> B{User Role?}

B --> |Admin| C[Admin Dashboard]

B --> |Sales Manager| D[Sales Dashboard]

B --> |Accountant| E[Accounting Dashboard]

B --> |Client| F[Client Portal]

C --> C1[Manage Projects]

C --> C2[Manage Users & Roles]

C --> C3[Property Entry]

C --> C4[Reports & Analytics]

D --> D1[New Booking]

D --> D2[Client Profiles]

D --> D3[Booking Reports]

E --> E1[Installments]

E --> E2[Payments]

E --> E3[Revenue Reports]

F --> F1[View Booked Flat Info]

F --> F2[Check Installment Status]

F --> F3[Download Invoice]

**🧩 ২. ড্যাশবোর্ড লে-আউট (Admin Dashboard UI Layout)**

plaintext

CopyEdit

-----------------------------------------------

| Logo | Dashboard | Projects | Properties |

| | Clients | Payments | Settings |

|---------------------------------------------|

| TOP BAR: Search | Notifications |

|---------------------------------------------|

| Stats Cards: |

| - Total Flats Sold - Ongoing Projects |

| - Monthly Revenue - Booked Clients |

|---------------------------------------------|

| Graphs (Sales Trend, Payment Flow) |

|---------------------------------------------|

| Recent Bookings | Recent Payments |

-----------------------------------------------

**🗃️ ৩. ডেটাবেজ স্কিমা আইডিয়া (Simplified)**

(RDBMS: MySQL / PostgreSQL)

**🔸 users**

| id | name | email | role | password | created\_at |

**🔸 projects**

| id | name | location | status | total\_flats | start\_date |

**🔸 properties**

| id | project\_id | flat\_no | size\_sqft | price | status (Available/Sold/Booked) |

**🔸 clients**

| id | name | phone | email | nid | booked\_property\_id |

**🔸 bookings**

| id | client\_id | property\_id | booking\_date | booking\_amount |

**🔸 installments**

| id | booking\_id | due\_date | amount | paid\_date | status |

**🔸 payments**

| id | client\_id | amount | date | method (bkash, cash) | note |

**🧰 ৪. ফিচার ব্রেকডাউন (Feature Modules)**

**🔹 Admin Panel:**

* Dashboard Overview
* Project CRUD
* Property CRUD
* User/Role Management
* Reports (PDF/CSV export)

**🔹 Sales Module:**

* New Booking Form
* Client Profile Management
* Flat Availability Checker

**🔹 Accounts Module:**

* Add/Edit Installments
* Mark Payment as Paid
* Generate Invoice
* Automated Due Alerts

**🔹 Client Portal:**

* Login System
* See Flat Info
* View Payment History
* Download Invoice PDF

**🧱 ৫. কোড স্ট্রাকচার প্ল্যান**

**✅ Frontend (React.js Structure):**

bash

CopyEdit

src/

│

├── components/

│ ├── Sidebar.js

│ ├── Header.js

│ ├── PropertyCard.js

│

├── pages/

│ ├── Dashboard.js

│ ├── Projects.js

│ ├── Bookings.js

│ ├── Payments.js

│ ├── Clients.js

│

├── services/

│ └── api.js # Axios calls to backend

│

├── App.js

└── index.js

**✅ Backend (Node.js + Express or Laravel):**

bash

CopyEdit

src/

├── controllers/

│ ├── projectController.js

│ ├── bookingController.js

│ ├── paymentController.js

│

├── models/

│ ├── User.js

│ ├── Property.js

│ ├── Booking.js

│

├── routes/

│ ├── projectRoutes.js

│ ├── bookingRoutes.js

│

├── middleware/

│ ├── authMiddleware.js

│

└── server.js

-- 🧑‍💼 Users Table (Admin, Sales, Accounts)

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

password VARCHAR(255),

role ENUM('admin', 'sales', 'accountant', 'client') DEFAULT 'client',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- 🏗️ Projects Table

CREATE TABLE projects (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

location VARCHAR(150),

status ENUM('ongoing', 'completed', 'upcoming') DEFAULT 'ongoing',

start\_date DATE,

end\_date DATE,

total\_units INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- 🏘️ Properties Table (Flats, Units, etc.)

CREATE TABLE properties (

id INT AUTO\_INCREMENT PRIMARY KEY,

project\_id INT,

flat\_number VARCHAR(20),

size\_sqft FLOAT,

price DECIMAL(15,2),

status ENUM('available', 'booked', 'sold') DEFAULT 'available',

floor INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (project\_id) REFERENCES projects(id)

);

-- 👨‍💼 Clients Table

CREATE TABLE clients (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

phone VARCHAR(20),

email VARCHAR(100),

nid VARCHAR(30),

address TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- 📄 Bookings Table

CREATE TABLE bookings (

id INT AUTO\_INCREMENT PRIMARY KEY,

client\_id INT,

property\_id INT,

booking\_date DATE,

booking\_amount DECIMAL(12,2),

status ENUM('booked', 'cancelled', 'confirmed') DEFAULT 'booked',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (client\_id) REFERENCES clients(id),

FOREIGN KEY (property\_id) REFERENCES properties(id)

);

-- 💳 Installments Table

CREATE TABLE installments (

id INT AUTO\_INCREMENT PRIMARY KEY,

booking\_id INT,

due\_date DATE,

amount DECIMAL(12,2),

paid\_date DATE,

status ENUM('due', 'paid', 'partial') DEFAULT 'due',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (booking\_id) REFERENCES bookings(id)

);

-- 💰 Payments Table

CREATE TABLE payments (

id INT AUTO\_INCREMENT PRIMARY KEY,

client\_id INT,

booking\_id INT,

amount DECIMAL(12,2),

payment\_date DATE,

method ENUM('cash', 'bkash', 'nagad', 'bank'),

reference\_number VARCHAR(50),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (client\_id) REFERENCES clients(id),

FOREIGN KEY (booking\_id) REFERENCES bookings(id)

);

-- 🧾 Documents Table (Optional - For storing file links)

CREATE TABLE documents (

id INT AUTO\_INCREMENT PRIMARY KEY,

client\_id INT,

doc\_type VARCHAR(50), -- e.g., NID, Booking Form, Agreement

file\_path VARCHAR(255),

uploaded\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (client\_id) REFERENCES clients(id)

);