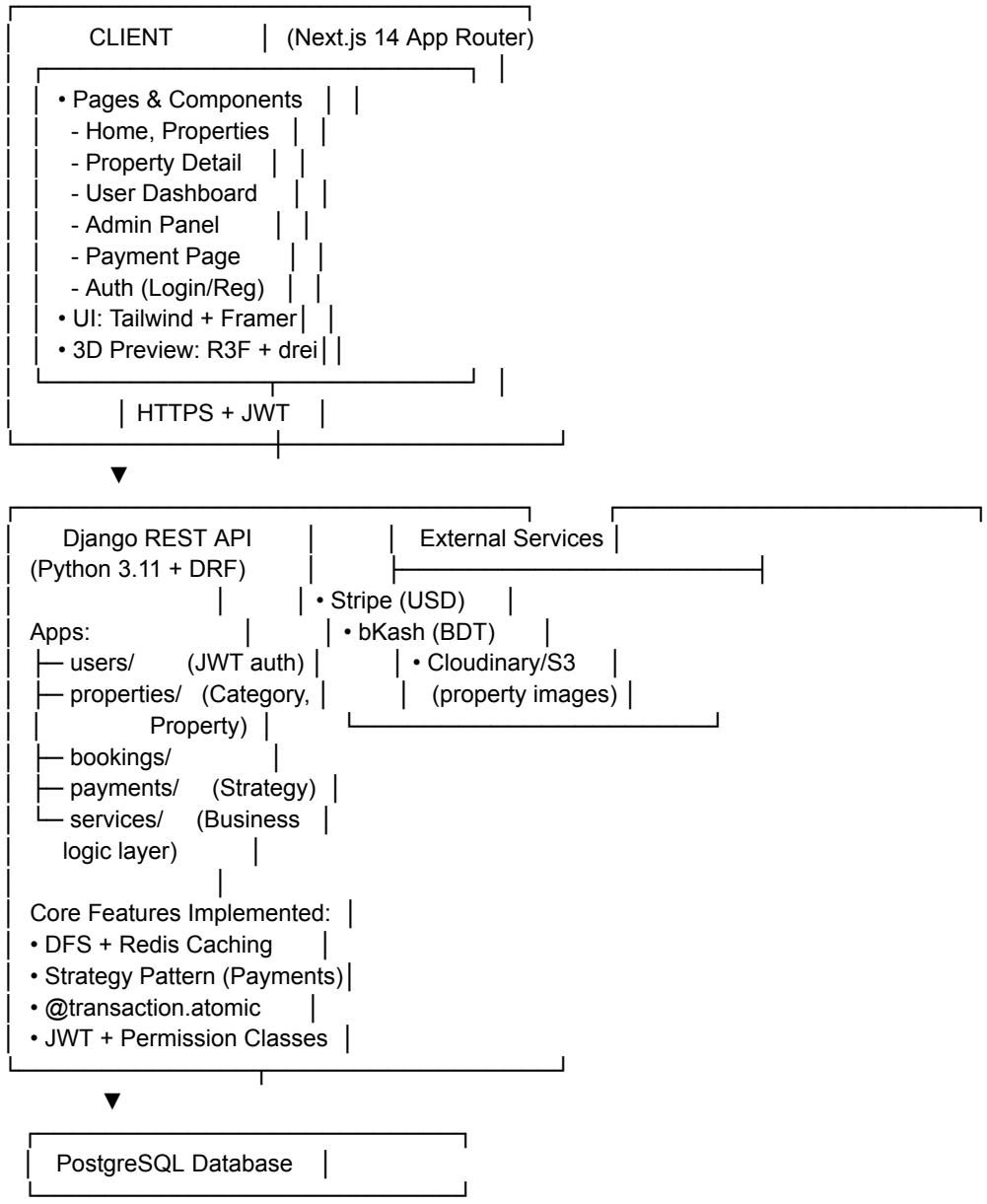
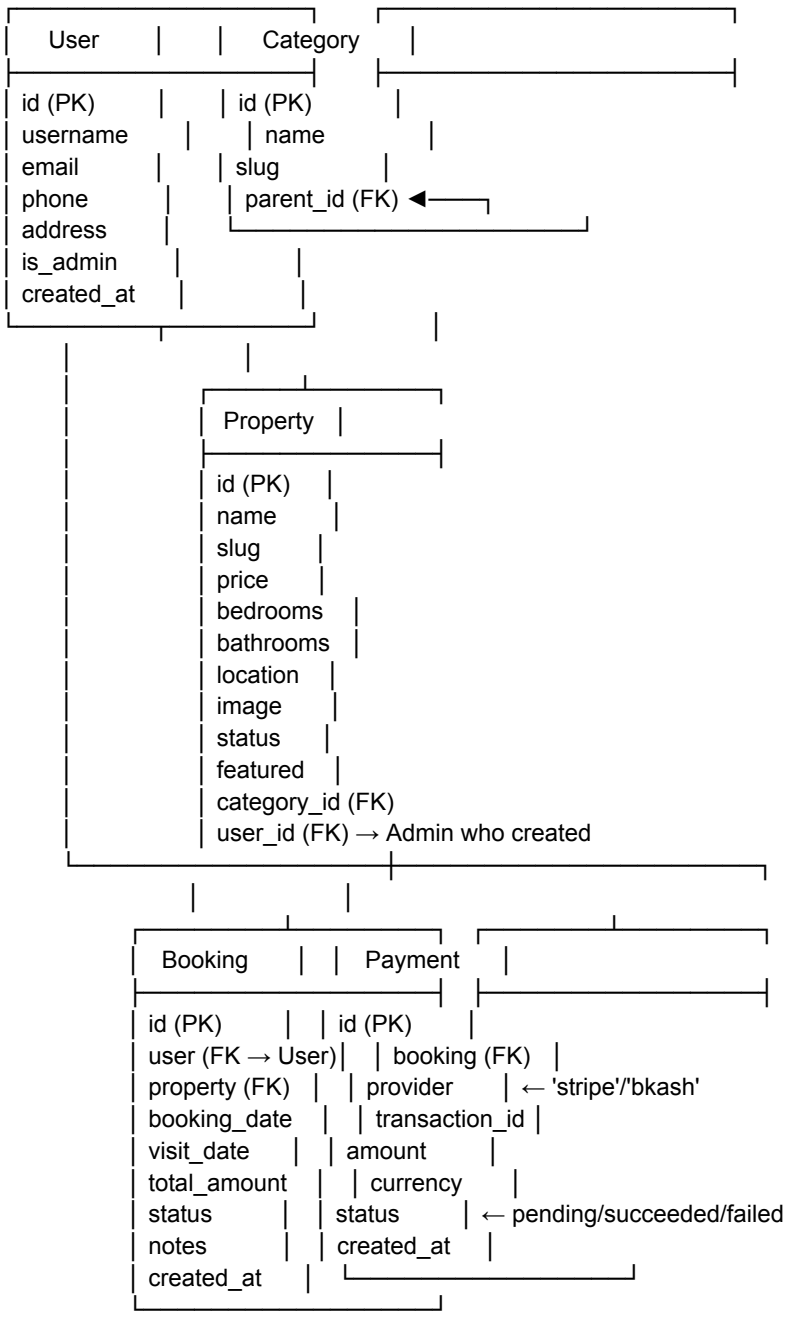
System Architecture




Deployment Flow

```
├── Frontend  → Vercel / Netlify
├── Backend   → Railway / Render / Heroku
├── Media     → Cloudinary or AWS S3
└── Domain    → yoursite.com (frontend) + api.yoursite.com (backend)
```

ERD


User
id (PK)
username
email
phone
address
is_admin
created_at
Category
id (PK)
name
slug
parent_id (FK)
Property
id (PK)
name
slug
price
bedrooms
bathrooms
location
image
status
featured
category_id (FK)
user_id (FK) → Admin who created
Booking
id (PK)
user (FK → User)
property (FK)
booking_date
visit_date
total_amount
status
notes
created_at
Payment
id (PK)
booking (FK)
provider
← 'stripe'/'bkash'
transaction_id
amount
currency
status
← pending/succeeded/failed
created_at

API Documentation – Swagger / Postman Ready (Main Endpoints)

| Method | Endpoint | Auth | Description |
|---|---|---|---|
| POST | `/api/users/` | Public | Register |
| POST | `/api/users/login/` | Public | Login → JWT |
| GET | `/api/users/me/` | JWT | Profile |
| GET | `/api/properties/` | Public | List + filter |
| GET | `/api/properties/{slug}/` | Public | Property detail |
| GET | `/api/properties/{slug}/recommendations/` | Public | DFS-based recommendations |
| POST | `/api/properties/` | Admin | Create property |
| PATCH/DELETE | `/api/properties/{slug}/` | Admin | Update/Delete |
| GET | `/api/bookings/` | JWT | My bookings |
| POST | `/api/bookings/` | JWT | Create booking |
| PATCH | `/api/bookings/{id}/` | JWT/Admin | Cancel / Admin update status |
| POST | `/api/payments/initiate/` | JWT | Start payment (stripe/bkash) |
| POST | `/api/payments/webhook/stripe/` | Public | Stripe webhook |
| POST | `/api/payments/webhook/bkash/` | Public | bKash webhook |

Stripe Payment Flow

.

Payment Flow Diagrams (Both Stripe & bKash)

User → Frontend → /api/payments/initiate/ (provider=stripe)
↓
Create PaymentIntent (Stripe)
↓
Return client_secret
↓
Frontend → Stripe Elements → confirmCardPayment()
↓
Success/Fail
↓
Webhook → Django → Update Payment & Booking status = paid

bKash Payment Flow

User → Frontend → /api/payments/initiate/ (provider=bkash)
↓
createPayment API → bKash returns paymentID + redirect URL
↓
Frontend redirect to bKash payment page
↓
User pays on bKash
↓
bKash redirects back → executePayment
↓
Webhook or polling → Update Payment & Booking status = paid