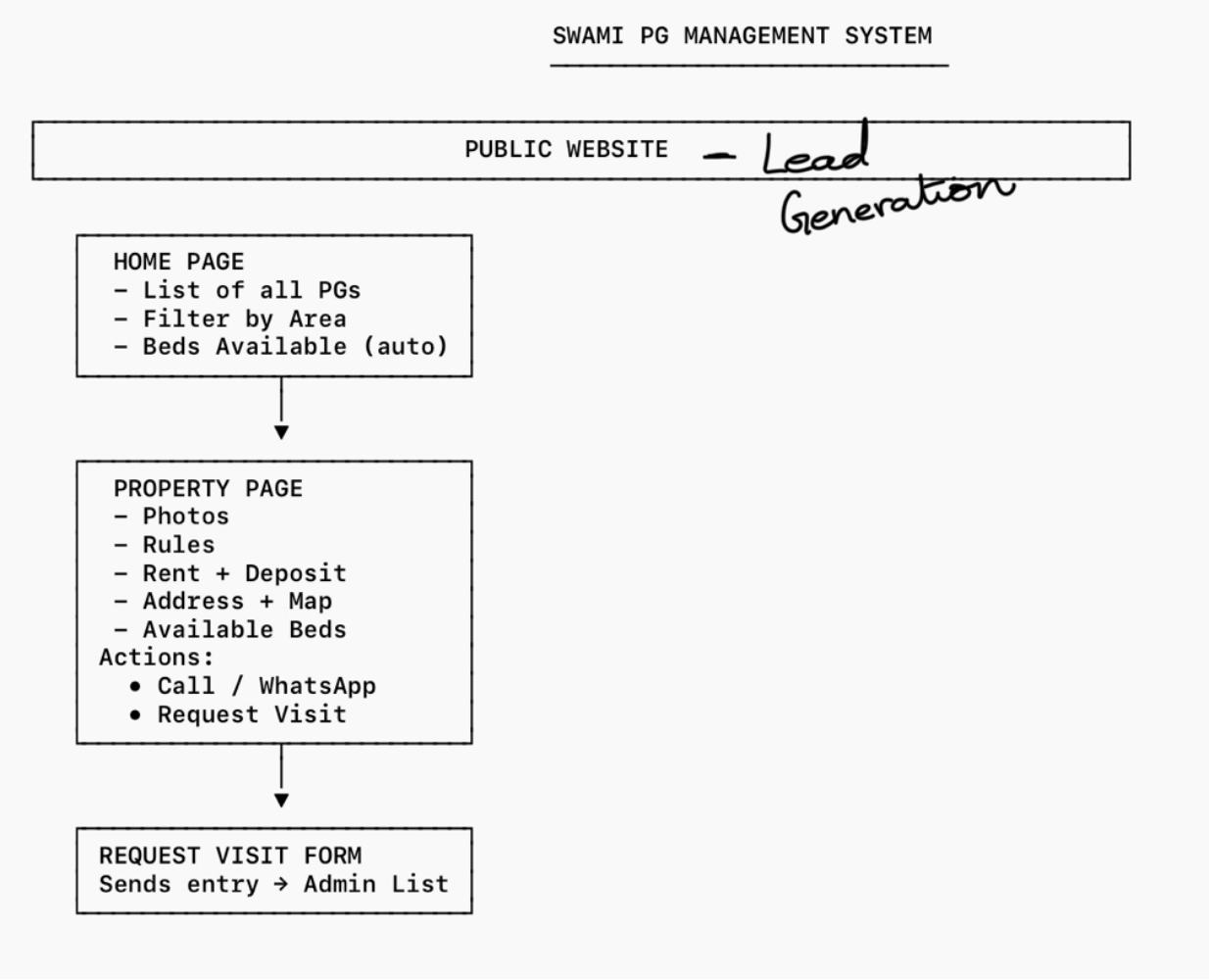
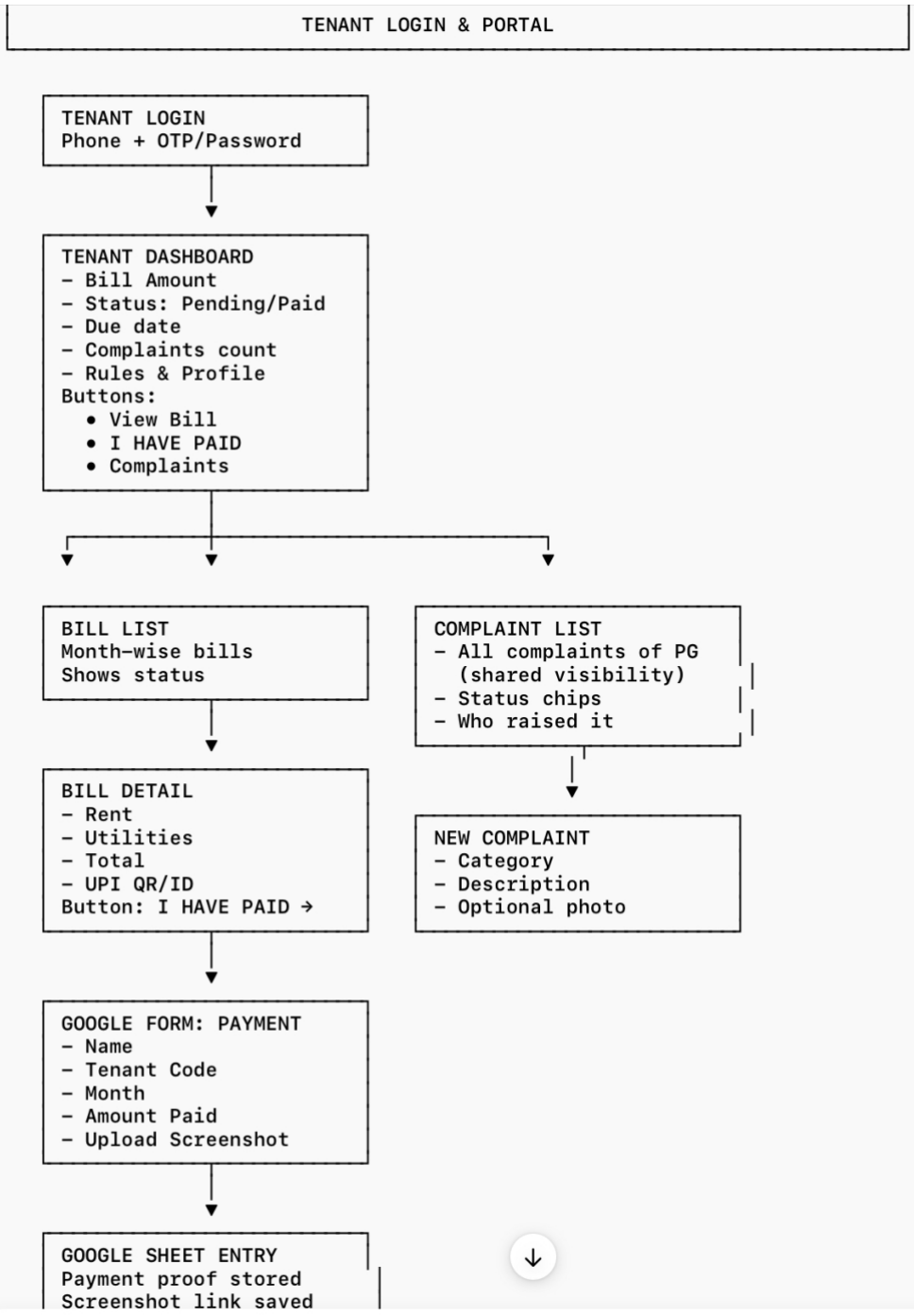
PHASE 1

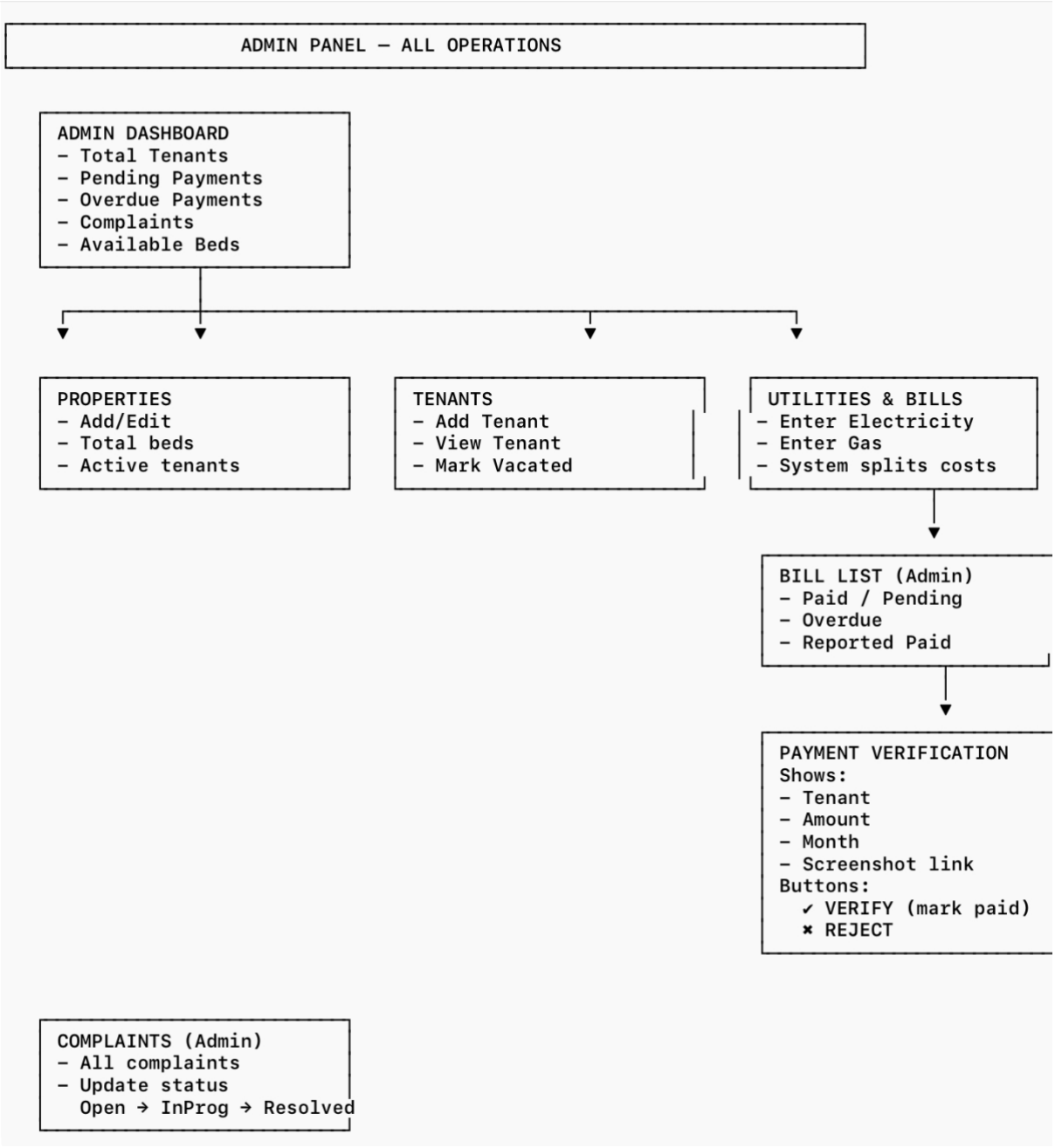

SWAMI PG MANAGEMENT SYSTEM
PUBLIC WEBSITE – Lead Generation
HOME PAGE
- List of all PGs
- Filter by Area
- Beds Available (auto)
PROPERTY PAGE
- Photos
- Rules
- Rent + Deposit
- Address + Map
- Available Beds
Actions:
• Call / WhatsApp
• Request Visit
REQUEST VISIT FORM
Sends entry → Admin List

PHASE 2


NEW TENANT ONBOARDING (GOOGLE FORM FLOW)
Tenant visits PG in person → decides to stay
SEND GOOGLE FORM LINK
- Name
- Phone
- Property (dropdown)
- Start date
- Aadhaar upload
- Other docs
GOOGLE SHEET ENTRY
Stores all tenant data
+ Document link
ADMIN → ADD TENANT
(From Google Sheet)
- Name
- Phone
- Property
- Rent / Deposit
- Start Date
- Docs link
Status = Active
UPDATE AVAILABLE BEDS
available = total − active tenants


TENANT LOGIN & PORTAL
TENANT LOGIN
Phone + OTP/Password
TENANT DASHBOARD
- Bill Amount
- Status: Pending/Paid
- Due date
- Complaints count
- Rules & Profile
Buttons:
• View Bill
• I HAVE PAID
• Complaints
BILL LIST
Month-wise bills
Shows status
COMPLAINT LIST
- All complaints of PG
(shared visibility)
- Status chips
- Who raised it
BILL DETAIL
- Rent
- Utilities
- Total
- UPI QR/ID
Button: I HAVE PAID →
NEW COMPLAINT
- Category
- Description
- Optional photo
GOOGLE FORM: PAYMENT
- Name
- Tenant Code
- Month
- Amount Paid
- Upload Screenshot
GOOGLE SHEET ENTRY
Payment proof stored
Screenshot link saved

PHASE 3


ADMIN PANEL — ALL OPERATIONS
ADMIN DASHBOARD
- Total Tenants
- Pending Payments
- Overdue Payments
- Complaints
- Available Beds
PROPERTIES
- Add/Edit
- Total beds
- Active tenants
TENANTS
- Add Tenant
- View Tenant
- Mark Vacated
UTILITIES & BILLS
- Enter Electricity
- Enter Gas
- System splits costs
BILL LIST (Admin)
- Paid / Pending
- Overdue
- Reported Paid
PAYMENT VERIFICATION
Shows:
- Tenant
- Amount
- Month
- Screenshot link
Buttons:
✔ VERIFY (mark paid)
✖ REJECT
COMPLAINTS (Admin)
- All complaints
- Update status
Open → InProg → Resolved

PHASE 4

```
TENANT DECIDES TO LEAVE
        |
        ▼
In real life: tenant informs owner (call / in-person)
        |
        ▼
ADMIN PANEL → OPEN TENANT PROFILE
        |
        ▼
Click: "INITIATE VACATION / LEAVING"
        |
        ▼
STEP 1: SET LEAVING DATE
        |
        ├─ Option A: Immediate (today)
        └─ Option B: Future date (end of month / notice period)
        |
        ▼
STEP 2: FINAL BILL CALCULATION
        |
        ├─ Auto-generate last bill:
        |     • Rent (full or pro-rata)
        |     • Utilities share till leaving date
        |     • Any penalties/dues
        |
        └─ Show: "Final Amount Due"
        |
        ▼
STEP 3: PAYMENT SETTLEMENT (OFFLINE)
        |
        ├─ Tenant pays remaining dues
        ├─ Admin updates status of final bill → PAID / PARTIAL
        └─ Admin decides deposit refund amount (offline)
        |
        ▼
STEP 4: MARK TENANT AS VACATED
        |
        ├─ tenant.status = "Vacated"
        ├─ tenant.vacated_date = [date]
        ├─ EXCLUDE from:
        |     • future bill generation
        |     • utilities splitting
        |
        └─ REVOKE LOGIN ACCESS (optional but ideal)
        |
        ▼
STEP 5: UPDATE AVAILABILITY
        |
        ├─ ACTIVE_TENANTS(property) -= 1
        └─ AVAILABLE_BEDS = TOTAL_BEDS - ACTIVE_TENANTS
        |
        ▼
SYSTEM NOW:
- Property shows 1 more available bed on website
- Tenant no longer appears in "Active Tenants" / billing
```

Alright, let's package this like you're handing Priyanshi a mini product Bible.

---

1. Project Overview

Project: Swami PG Management System

Location: Vadodara

Business: Swami PG – 15 PG properties, ~100+ tenants (aim: 1500)

Goal of this system:

To simplify daily operations of Swami PG and make it possible to scale from 15 PGs to 150+ by digitizing:

- Property listing & availability
- Tenant onboarding (via Google Forms)
- Monthly rent + utility billing
- Payment tracking using payment screenshots (via Google Form)
- Complaints management (visible to all tenants in same PG)

---

2. V1 Scope – Only Build This First (ALL the systems to be interconnected so beds payment all change accordingly)

Do NOT build beyond this list in V1.

2.1 Public Website

1. Home Page
    - List all PGs.
    - Filter by area.
    - Show "Beds Available".
2. Property Detail Page
    - Photos.
    - Rules.
    - Rent & deposit.
    - Address + map link.
    - Buttons: Call / Request Visit.
3. Request Visit Page

- Form to collect visit leads.

2.2 Tenant Portal

4. Tenant Login. For old and for new tenant login to be created.
    - Phone + password (or OTP if she's comfortable with Firebase Auth).
5. Tenant Dashboard
    - Current month bill.
    - Status: Pending / Paid / Overdue / Reported Paid.
    - Buttons: View Bill, I Have Paid (opens payment Google Form), Complaints, Rules/Profile.
6. Bills
    - Bills list per month.
    - Bill detail screen with breakdown.
7. Complaints
    - Complaints list (all complaints for that PG).
    - New complaint form.
8. Profile & Rules
    - Basic tenant info.
    - Rules for the PG.

2.3 Admin Panel

9. Admin Dashboard
    - Total tenants.
    - Pending & overdue payments count.
    - Open complaints count.
    - Available beds summary. (Update as per vacant or occupied)
10. Properties Management
    - Add/Edit PG.
    - Total beds.
    - Rules.
    - Auto-calculated available beds.
11. Tenants Management
    - Add tenant (from Google Sheet).
    - View tenants.
    - Mark as Vacated.
12. Utilities & Bills
    - Enter monthly electricity & gas bill per property.
    - Generate bills (rent + utilities split).
    - See bill statuses.
13. Payment Verification
    - Show all "Reported Paid" bills.
    - Link to corresponding Google Sheet row (screenshot).
    - Buttons: Verify / Reject.
14. Complaints Admin
    - View all complaints by property.

    - Change status: Open → In Progress → Resolved.
15. Visit Requests
    - List of visit forms from public site.

That's V1. Nothing more.

---

3. Tech Stack Recommendation (Keep it Simple)

Frontend:

- React (or Next.js) OR even basic HTML+Bootstrap if she's more comfortable.

Backend / DB / Auth (strongly recommended):

- Firebase Auth (phone/password or email/password)
- Firebase Firestore (for properties, tenants, bills, complaints)
- Firebase Hosting for the website + portal

Forms & Files:

- Google Forms + Google Sheets + Google Drive for:
    - New tenant onboarding
    - Payment screenshot upload

This avoids writing file upload code and heavy backend.

---

4. Database Schema (FireStore / SQL Friendly)

She can map this to Firebase collections or tables in SQL.

4.1

properties

- id (string)
- name (string) – e.g., “Swami PG – Gotri”
- area (string) – e.g., “Gotri”
- address (string)
- total_beds (number)
- default_rent (number)
- default_deposit (number)
- rules_text (string / long text)
- landmark (string, optional)

Derived in UI (not stored):

available_beds = total_beds - active_tenants_count

Where active_tenants_count = number of tenants with status = "Active" in that property.

4.2

tenants

- id (string)
- name (string)
- phone (string)
- property_id (string → reference to properties.id)
- tenant_code (string, e.g., SPG101)
- start_date (date)
- rent (number)
- deposit (number)
- status (string: "Active" / "Vacated")
- docs_link (string → Google Drive / Sheet URL)
- vacated_date (date, nullable)

4.3

bills

- id (string)
- tenant_id (string)
- month (number 1–12)
- year (number)
- rent_amount (number)

- electricity_share (number)
- gas_share (number)
- late_fee (number)
- total_amount (number)
- status (string: "Pending" | "ReportedPaid" | "Paid" | "Overdue")
- is_final (boolean, default false)
- created_at (timestamp)
- paid_at (timestamp, nullable)

4.4

complaints

- id (string)
- tenant_id (string)
- property_id (string)
- title (string)
- description (string)
- category (string: “Electrical”, “Water”, etc.)
- status (string: "Open" | "InProgress" | "Resolved")
- image_url (string, optional)
- created_at (timestamp)
- updated_at (timestamp)

When tenant opens complaints list, query by property_id (so all tenants in same PG see all complaints).

4.5

visit_requests

- id (string)
- name (string)
- phone (string)
- property_id (string)
- preferred_date (date)
- preferred_time (string)
- status (string: "New" | "Contacted" | "Completed")
- created_at (timestamp)

---

5. Google Form Templates (Text for You to Create)

5.1 Google Form 1 –

New Tenant Onboarding

Form title: “Swami PG – New Tenant Details”

Fields:

1. Full Name – Short answer
2. Mobile Number – Short answer (validation: numeric, 10 digits)
3. Which PG property are you staying at? – Dropdown
    - Swami PG – Gotri
    - Swami PG – Akota
    - … (all 15)
4. Start Date of Stay – Date
5. Upload Aadhaar (front/back or PDF) – File upload
6. Upload any other ID / documents (optional) – File upload
7. Any additional notes – Paragraph (optional)

Responses go into Sheet: “New Tenants – Responses”.

5.2 Google Form 2 –

Payment Screenshot Submission

Form title: “Swami PG – Rent Payment Proof”

Fields:

1. Full Name – Short answer
2. Mobile Number – Short answer
3. Tenant Code (given by owner, e.g., SPG101) – Short answer
4. PG Property – Dropdown
5. Month & Year of Rent – Short answer or dropdown (e.g., “Jan 2026”)
6. Amount Paid (₹) – Short answer (numeric)

7. Last 4 digits of UPI Transaction / UTR – Short answer (optional but useful)
8. Upload Payment Screenshot – File upload

Responses go into Sheet: “Rent Payments – Responses”.

Admin uses this + UPI app to verify.

---

6. UI Style Guide (Mini)

Colors (Example):

- Primary: #1E88E5 (blue)
- Success (Paid): #2e7d32 (green)
- Warning (Pending / ReportedPaid): #f9a825 (yellow/orange)
- Danger (Overdue): #c62828 (red)
- Background: #f5f5f5

Status chips:

- Paid → Green chip, text: “Paid”
- Pending → Yellow chip, text: “Pending”
- Reported Paid → Orange chip, text: “Reported Paid”
- Overdue → Red chip, text: “Overdue”

Buttons:

- Border radius: ~6–8px
- Primary button color: primary blue
- Full width on mobile where possible.

Typography:

- Font: Roboto / system sans.
- Title size ~20–24px.
- Card titles ~16–18px.

- Body ~14–16px.

---

7. Simple ASCII Wireframes (She can convert to Figma/UI)

7.1 Home Page

```
+-----------------------------------------------------+
| Swami PG (logo)                [Tenant Login btn]   |
+-----------------------------------------------------+

Find Your PG in Vadodara
Strict, safe and affordable accommodation.

[ Area: (Dropdown v) ] [ Apply ]

-------------------------------------------------------
| Swami PG – Gotri              Area: Gotri           |
| Starting from ₹6,500 / month                       |
| Beds Available: 3                                   |
| Rules: No smoking • No alcohol • No guests          |
| [View Details] [Call]                               |
-------------------------------------------------------
| Swami PG – Akota              Area: Akota           |
| ...                                                 |
-------------------------------------------------------
```

7.2 Property Detail Page

```
[Back]      Swami PG – Gotri

Area: Gotri, Vadodara
Landmark: Near XYZ College

Total beds: 30   Occupied: 27   Available: 3

[ Photo 1 ]
[ Photo 2 ]
[ Photo 3 ]

[Card] House Rules
- No smoking
- No alcohol or drugs
- No guests staying overnight
- Maintain cleanliness
...

[Card] Rent & Deposit
Rent: ₹6,500 – ₹7,500 / month
Deposit: ₹3,000 (refundable*)
```

```
[Card] Location
Address: ...
[View on Google Maps]

[ Call / WhatsApp ]   [ Request a Visit ]
```

7.3 Tenant Dashboard

```
Hi, Vivek
You are staying at: Swami PG – Gotri

[Card] This Month
Total Due: ₹7,200
Status: [Pending]   Due by: 7 Jan 2026
[View Bill]  [I HAVE PAID]

[Card] Complaints
Open: 1   Resolved this month: 3
[View Complaints]

[Card] Info
[View House Rules]
[View Profile]

Bottom Nav (mobile):
[Dashboard] [Bills] [Complaints] [Profile]
```

7.4 Admin Dashboard

```
Swami PG – Admin Panel

Top cards:
[ Total Tenants: 96 ]
[ Total PGs: 15 ]
[ Pending Payments: 18 ]
[ Open Complaints: 5 ]

Sections:

[ Properties ]  [ Tenants ]  [ Bills & Utilities ]
[ Complaints ]  [ Visit Requests ]  [ Payment Verification ]
```

---

8. Week-by-Week Development Plan for Her

Week 1

- Set up project (React/Firebase or stack she chooses).
- Implement:
  - Home Page
  - Property Detail Page

    - Request Visit form (saving into database).

Week 2

- Implement:
    - Tenant Login
    - Tenant Dashboard UI
    - Bills List & Bill Detail (static/dummy data first).

Week 3

- Implement:
    - Complaints (tenant side + admin list + status change).
    - Complaint visibility by property.

Week 4

- Implement:
    - Admin: Properties & Tenants (add/edit).
    - Availability calculation (total beds – active tenants).

Week 5

- Implement:
    - Utilities entry per property.
    - Bill generation logic (rent + equal split).
    - Payment flow status fields (Pending, ReportedPaid, Paid, Overdue).

Week 6

- Implement:
    - Payment Verification admin screen.
    - Visit Requests admin screen.
    - Tenant exit (mark Vacated, stop future bills).
- Test full flow end-to-end & deploy.

---

9. Developer Checklist (Give this directly to her)

1. Read this full document end-to-end.
2. Decide tech stack (React + Firebase recommended).
3. Create collections/tables exactly as defined.
4. Build website pages: Home → Property Detail → Request Visit.
5. Build Tenant Login + Dashboard.
6. Build Bills & Complaints (tenant).
7. Build Admin Dashboard, Properties, Tenants.
8. Build Utilities + Bill generation.
9. Implement Payment statuses & Verification page.
10. Hook Google Forms (onboarding + payment screenshot) via links.
11. Test:
    - New tenant added → shows in app.
    - Bill generated → visible to tenant.
    - Payment screenshot submitted → mark as ReportedPaid → admin verifies → Paid.
    - Complaint by one tenant visible to others in same PG.
    - Tenant vacated → availability updates & no future bills.

---

10. About UI "Images"

I can't attach real graphic mockups in this reply, but the ASCII wireframes above are structured so she can directly recreate them in Figma/Canva or any design tool in 30–60 minutes:

- One Figma frame per screen:
    - Home
    - Property Detail
    - Tenant Dashboard
    - Bills List & Detail
    - Complaints List & New Complaint
    - Admin Dashboard
    - Payment Verification page

If you want, I can next:

- Go screen-by-screen and write exact text and labels for every button and error message, so she literally just copies it into UI.