GRAM SAHAY: COMPLETE HACKATHON GUIDE

Technical + Theoretical Foundation

PART A: THEORETICAL PROJECT FLOW & ESCALATION SYSTEM

FLOW 1: CITIZEN FILES COMPLAINT (Complete Journey)

```
CITIZEN SUBMITS COMPLAINT
        ↓
[Day 0, Hour 0]
Citizen takes photo of broken streetlight, fills form in Gujarati
- Category: Street Light
- Description: Light in front of primary school not working 5 days
- Photo with location (geo-tagged)
- Submits → System generates Complaint ID #12345
        ↓
ALERT #1 → TALATI (Gram Panchayat Secretary)
Talati receives INSTANT notification:
- New complaint received
- SMS: "          . ID #12345. [Link]"
- Talati opens admin dashboard → Sees "1 NEW COMPLAINT"
- Status shows: SUBMITTED (Yellow badge)
        ↓
TALATI ACTIONS (Next 24 hours)
[Day 0-1] Talati reviews:
   Photo is clear?  Location valid?  Category correct?

  Option A: APPROVE & ASSIGN
  - Talati clicks "Approve"
  - Assigns to "Electrician Rajesh"
  - Adds note: "              "
  - Status changes: IN-PROGRESS (Blue badge)
        ↓
  ALERT #2 → CITIZEN (Phone SMS)
  "            !        : #12345
      :      |           "
        ↓
  ELECTRICIAN WORKS (Days 1-7)
  - Electrician Rajesh fixes the light
```

```
- Takes photo of fixed light
- Submits: "     ,          "
      ↓
ALERT #3 → TALATI
"Complaint #12345 marked as RESOLVED by Rajesh"
      ↓
TALATI FINAL APPROVAL (Day 7)
- Talati reviews photo of fixed light
- Clicks "CLOSE COMPLAINT"
- Status: RESOLVED (Green badge)
      ↓
ALERT #4 → CITIZEN (Final SMS)
"  !              !
               .          ?"
      ↓
CITIZEN RECEIVES FEEDBACK SURVEY
- "Rate our service:    "
- System stores rating for panchayat transparency
      ↓
============================================
TOTAL TIME: 7 DAYS | STATUS: CLOSED
============================================


Option B: REJECT (If photo unclear, or not in panchayat scope)
- Talati clicks "REJECT"
- Adds reason: "                    ,       "
- Status: REJECTED (Red badge)
      ↓
ALERT #2B → CITIZEN (Immediate SMS)
"              .   : [reason]
         : [Link to appeal]"
      ↓
COMPLAINT CLOSED
      ↓
============================================
TOTAL TIME: Same day | STATUS: REJECTED
============================================
```

FLOW 2: ESCALATION IF TALATI DOESN'T RESPOND (AUTO-ESCALATION)

```
CITIZEN FILES COMPLAINT (Day 0)
       ↓
```

```
        ↓
        ↓
[DAY 3] NO RESPONSE FROM TALATI
System auto-checks:
- Has talati taken action? NO
- Timeline remaining: 4 days left
        ↓
ALERT #E1 → TALATI (Reminder SMS)
"       #12345          .
                     TALUKA    ."
        ↓
        ↓
        ↓
[DAY 7] STILL NO RESPONSE FROM TALATI
System checks:
- Timeline exceeded: 7 days passed
- No action taken
        ↓
AUTO-ESCALATION TRIGGERED
Status changes: PENDING → ESCALATED TO TALUKA
        ↓
ALERT #E2 → TALUKA OFFICER (Taluka Panchayat / Block Officer)
"Critical: Complaint #12345 auto-escalated from Gram Panchayat.
 Talati [Name] did not respond in 7 days.
 Citizen status: UNSATISFIED
 Your time: 7 days to respond"
        ↓
ALERT #E3 → CITIZEN
"
             .          :     "
        ↓
TALUKA OFFICER REVIEWS
- Reads complaint
- Sees: Talati didn't respond
- Can either:
  A) Direct talati to work immediately
  B) Assign to another worker
  C) Take action at taluka level
        ↓
ALERT #E4 → TALATI (Authority warning)
"                          .
                        ."
        ↓
        ↓
[DAY 8-14] TALUKA LEVEL RESPONSE
Officer either resolves or escalates to DISTRICT
```

```
        ↓
        ↓
IF STILL UNRESOLVED AFTER 14 DAYS:
AUTO-ESCALATION TO DISTRICT PANCHAYAT
        ↓
[DAY 15+] DISTRICT LEVEL INVOLVEMENT
        ↓
==========================================
ESCALATION LADDER:
Gram Panchayat (7 days) →
Taluka/Block (7 days) →
District (7 days) →
[Can't escalate further: Public complaint]
==========================================
```

FLOW 3: PANCHAYAT ADMIN DASHBOARD (What Talati Sees)

```
TALATI LOGS IN TO ADMIN DASHBOARD
        ↓
        ↓
HOME SCREEN Shows:

  PANCHAYAT DASHBOARD - TODAY

    THIS MONTH STATISTICS
  • Total Complaints: 23
  • Resolved: 18
  • Pending: 4
  • Rejected: 1
  • Avg Resolution Time: 4.2 days

    URGENT (Auto-escalate in 2 days)
     - ID #12340: Water supply [BLUE]
     - ID #12341: Garbage [YELLOW]

    COMPLAINTS BY CATEGORY
     Street Light:      (8)
     Garbage:     (6)
     Water:    (4)
     Road:   (3)
     Toilet:   (2)

        ↓
```

```
CLICK "ALL COMPLAINTS" TAB
        ↓
        ↓
TABLE VIEW:

  ID     Category   Status     Citizen  Submitted

 12345   Light      BLUE       Raj      2 days
 12346   Water      YELLOW     Priya    5 days
 12347   Garbage    GREEN      Vikram   CLOSED
 12348   Road       RED        Neha     REJECTED

        ↓
CLICK ON COMPLAINT #12345
        ↓
        ↓
COMPLAINT DETAIL VIEW:

  COMPLAINT #12345 - STREET LIGHT
  Status: IN-PROGRESS [BLUE]

  Citizen: Rajkumar Singh
  Phone: 94XXXX9090
  Village: Vasad, Kheda
  Submitted: Jan 5, 2026 @ 2:30 PM

  COMPLAINT DETAILS
  Category: Street Light
  Description: "Light near school is dark"
  Photo: [IMG] (Geo-tagged: 22.74°N)
  Location: Primary School Road

  TIMELINE (7 Days)
         (Day 2 of 7)
  [AUTO-ESCALATE in 5 days if no action]

  TALATI ACTIONS
  [Dropdown] Status:
     SUBMITTED (Current)
     APPROVED → IN-PROGRESS
     REJECTED
     CLOSED (After Work Done)

  [Dropdown] Assign to:
     Select Worker (Electrician Rajesh)
```

```
  [Text Box] Remarks:
    "           ..."

  [BUTTON] APPROVE & ASSIGN
  [BUTTON] REJECT
  [BUTTON] CLOSE COMPLAINT

        ↓
TALATI CLICKS "APPROVE & ASSIGN"
        ↓
SYSTEM:
- Updates status to IN-PROGRESS
- Sends SMS to electrician: "      "
- Sends SMS to citizen: "          "
- Timeline resets (7 days from today for completion)
        ↓
==========================================
TALATI'S RESPONSIBILITY:  DONE
Now waiting for worker to complete work
==========================================
```

FLOW 4: WORKER COMPLETES WORK (Electrician Perspective)

```
ELECTRICIAN RAJESH RECEIVES NOTIFICATION
SMS: "      . ID #12345
                .   :     "
        ↓
RAJESH OPENS APP
        ↓
SEES: "ASSIGNED WORK" TAB

  MY ASSIGNED WORK
  Complaint #12345
  Street Light Near School
  Deadline: Jan 12 (6 days)
  [MARK AS DONE] button

        ↓
RAJESH GOES TO LOCATION
Uses Google Maps to navigate
        ↓
FIXES THE LIGHT
Changes bulb, tests it
```

```
        ↓
RAJESH CLICKS "MARK AS DONE"
        ↓
APP ASKS:
- Take photo of fixed light? [Camera]
- Write completion remark? [Text: "         ,     "]
- Submit
        ↓
SYSTEM NOTIFIES:
        ↓
ALERT #W1 → TALATI
"Complaint #12345 marked DONE by Rajesh.
 Review & approve to close."
        ↓
TALATI REVIEWS PHOTO
"Light looks working  "
        ↓
TALATI CLICKS "CLOSE COMPLAINT"
        ↓
SYSTEM:
- Changes status: IN-PROGRESS → RESOLVED (Green)
- Triggers final SMS to citizen
- Requests citizen feedback
        ↓
==========================================
COMPLAINT FULLY RESOLVED
Time taken: 3 days (from submission to closure)
==========================================
```

FLOW 5: TRANSPARENCY & ACCOUNTABILITY (Budget + Analytics)

```
PANCHAYAT BUDGET PAGE (Everyone can see)
        ↓
BUDGET BREAKDOWN FOR THIS MONTH:

 Total Allocated:  50,000

 Street Lights:  15,000 spent
   - 8 repairs @  1,500 each
   - Parts cost:  2,000
 Garbage:  12,000 spent
   - Weekly cleaning
 Water:  8,000 spent
```

```
   - Pipe repair + maintenance
 Roads:  10,000 spent
   - Pothole filling
 Toilet Cleaning:  5,000 spent

        ↓
ANALYTICS DASHBOARD (Citizen views on website)
        ↓

 OUR PANCHAYAT'S PERFORMANCE
 This Month: January 2026

   COMPLAINTS CHART
    Street Light: 8 (8/8 resolved)
    Garbage: 6 (6/6 resolved)
    Water: 4 (3/4 resolved, 1 pending)
    Road: 3 (2/3 resolved, 1 pending)
    Toilet: 2 (2/2 resolved)

   TOTAL RESOLVED: 21/23 (91%)
   PENDING: 2
   AVG RATING: 4.2/5
   AVG TIME: 4.1 days

   TREND (Last 30 days)
    Week 1: 5 complaints
    Week 2: 6 complaints ↑
    Week 3: 7 complaints ↑
    Week 4: 5 complaints ↓ (ongoing)

 → More people trusting system

        ↓
SARPANCH REVIEWS EVERY WEEK IN GRAM SABHA
"Last month: 91% complaints resolved on time.
 Average 4.1 days. People rating us 4.2/5 stars.
 This means we're doing well! But 2 are pending.
 We must focus on water supply this month."
        ↓
==========================================
TRANSPARENCY BUILDS TRUST
Villagers see real numbers, not fake promises
==========================================
```

COMPLETE ESCALATION HIERARCHY

```
GRAM SAHAY COMPLAINT ESCALATION HIERARCHY:

DAY 0: Citizen Files Complaint
       ↓
       Status: SUBMITTED (Yellow)
       Responsible: Talati/Gram Panchayat Secretary
       Time Limit: 7 days

DAY 1-3: No Response?
         System sends reminder to Talati
         "Please respond within 4 days"

DAY 7: Talati Still Silent?
       System triggers AUTO-ESCALATION
       ↓
       Complaint sent to Taluka/Block Officer
       Status: ESCALATED (Orange)
       New Responsible: Taluka Panchayat Officer
       New Time Limit: 7 days

DAY 8-10: Taluka reviews
          "Talati ignored this. I'll direct action."

DAY 14: Taluka Still Silent?
        System triggers AUTO-ESCALATION
        ↓
        Complaint sent to District Panchayat
        Status: DISTRICT LEVEL (Red)
        New Responsible: District Panchayat Officer
        New Time Limit: 7 days

DAY 21: District involvement
        Public complaint log (transparency)
        Media may pick up

DAY 30: If STILL unresolved
        Complaint becomes PUBLIC RECORD
        Panchayat must file written response
        RTI can be filed


KEY STATS FOR JUDGES:
- Meri Panchayat: 15% usage (users give up)
- Gram Sahay: Auto-escalation forces action
```

```
  → Expected 60-70% resolution rate (pilot scale)
- Panchayat staff forced to respond or escalated
- Citizen gets SMS at each step (confidence)
```

PART B: TECHNICAL DEVELOPMENT ROADMAP

ROADMAP: 24-48 HOUR HACKATHON BUILD

Phase 1: Planning & Setup (0-2 hours)

Tasks: - [ ] Set up GitHub repo with proper folder structure - [ ] Initialize Node.js + Express backend - [ ] Initialize React frontend (Vite for faster builds) - [ ] Configure Firebase/Google Cloud (for Google tech requirement) - [ ] Set up environment variables (.env file) - [ ] Create database schema (MongoDB or Firestore)

Tools to Use (Google-approved for GDG): - **Frontend**: React 18 + Vite + Tailwind CSS - **Backend**: Node.js + Express.js - **Database**: Firebase Firestore (Google's database) - **Authentication**: Firebase Authentication (OTP) - **File Storage**: Google Cloud Storage - **Maps**: Google Maps API (geo-tagging) - **Hosting**: Firebase Hosting (free for hackathon)

Phase 2: Backend Setup (2-6 hours)

Build these API endpoints:

```
CITIZEN SIDE:
POST   /api/auth/register          → Register with phone + OTP
POST   /api/auth/verify-otp        → Verify OTP
GET    /api/profile                → Get citizen profile
GET    /api/complaints             → Get my complaints
POST   /api/complaints             → Submit new complaint
GET    /api/complaints/:id         → Get complaint details
PUT    /api/complaints/:id         → Update complaint with photo

PANCHAYAT ADMIN SIDE:
POST   /api/admin/login            → Login with credentials
GET    /api/admin/dashboard        → Dashboard stats
GET    /api/admin/complaints       → All complaints (paginated)
PUT    /api/admin/complaints/:id   → Update complaint status
GET    /api/admin/analytics        → Charts & metrics

SHARED FEATURES:
```

```
GET    /api/panchayat/info         → Panchayat details
GET    /api/panchayat/budget       → Budget info
GET    /api/services               → Available services
POST   /api/services/apply         → Apply for service
```

Database Schema (Firestore):

```
Collections:
  citizens
      uid
      phone
      name
      createdAt
      address
  complaints
      complaintId
      citizenId
      category (street light, garbage, water, etc.)
      description
      latitude / longitude (for geo-tagging)
      photo_url (stored in Cloud Storage)
      status (submitted, in-progress, resolved, closed)
      createdAt
      updatedAt
      remarks (for admin)
  panchayat
      name
      district
      budget
      phone
      email
  services
       serviceId
       name (certificate, license, etc.)
       description
       documents_required
```

Phase 3: Frontend - Citizen Side (6-14 hours)

Pages/Components to Build:

1. **Landing Page**
   - Hero section with "Gram Sahay" title
   - Brief intro in Gujarati + English
   - "Register Now" / "Login" buttons
   - Features list (3-4 bullet points)

2. **Auth Pages**
   - Phone number input with OTP verification (Firebase Auth)
   - Set profile info (name, village)
3. **Citizen Dashboard**
   - Welcome message
   - Quick stats: "Your complaints: X, Resolved: Y"
   - My complaints list (cards showing status)
   - Action buttons: "File Complaint", "Request Service"
4. **File Complaint Page**
   - Category dropdown (select from: , , , , etc.)
   - Description text area
   - **Photo upload** (capture from camera or upload, geo-tagged via Google Maps API)
   - Location picker (Google Maps)
   - Submit button
   - Success message with complaint ID
5. **Complaint Details Page**
   - Show: Status (color-coded: submitted=yellow, in-progress=blue, resolved=green)
   - Timeline of updates
   - Admin remarks
   - Option to add photo update
6. **Services Page**
   - List available services (birth certificate, income certificate, etc.)
   - Click to apply → collect required documents → submit
   - Track application status
7. **Panchayat Info Page**
   - Basic info (name, phone, address)
   - Budget overview (simple bar chart)
   - Contact details

Phase 4: Frontend - Panchayat Admin Side (14-18 hours)

Pages/Components:

1. **Admin Login Page**
   - Simple username/password login (hardcoded for demo: username=panchayat, password=admin123)
2. **Admin Dashboard**
   - **Key metrics**:
     - Total complaints received (this month)
     - Complaints resolved (this month)
     - Pending complaints
     - Avg resolution time
   - **Charts**:

  - Bar chart: Complaints by category
  - Line chart: Complaints over 7 days
  - Pie chart: Status breakdown (submitted, in-progress, resolved)
3. **All Complaints Page**
   - Table with columns: Complaint ID | Category | Status | Citizen Name | Submitted Date | Action
   - Filters: By category, by status, by date range
   - Search bar
   - Pagination (10 per page)
4. **Complaint Details Page (Admin)**
   - Full complaint info + photo
   - Update status dropdown (submitted → in-progress → resolved)
   - Add remarks text area
   - Assign to worker (worker name field)
   - Save/Submit button
5. **Analytics Dashboard**
   - Simple metrics (total, resolved, pending)
   - Charts (bar, line, pie)
   - Filter by date range

Phase 5: Mobile Responsiveness (18-20 hours)

- Make all pages mobile-friendly (Tailwind CSS handles this)
- Test on mobile browsers
- Ensure touch-friendly buttons/inputs

Phase 6: Testing, Refinement & Deployment (20-24 hours)

☐ Test all flows end-to-end
☐ Fix bugs
☐ Deploy to Firebase Hosting
☐ Get live URL
☐ Prepare demo script

PART C: SUPER TECHNICAL BUILD PROMPT

COMPREHENSIVE BUILD PROMPT FOR AI CODE GENERATOR

```
You are an expert full-stack web developer building a hyperlocal panchayat complaint and ser
```

TECH STACK (Google-approved):
- Frontend: React 18 + Vite + Tailwind CSS
- Backend: Node.js + Express.js
- Database: Firebase Firestore
- Authentication: Firebase Authentication (OTP)
- File Storage: Google Cloud Storage
- Maps: Google Maps API (geo-tagging)
- Hosting: Firebase Hosting

PROJECT CONTEXT:
Gram Sahay is designed as a hyperlocal pilot system for ONE village. It is inspired by Meri
it's a learning system to understand adoption barriers and test features that work at grassr

KEY FEATURES (Based on Meri Panchayat's core offerings):
1. Citizen complaint registration with photo + geo-tagging (categories: street lights, garba
2. Complaint tracking with status updates (submitted → in-progress → resolved)
3. SMS/email notifications when complaint status changes
4. Panchayat admin dashboard to manage complaints
5. Simple panchayat info page (budget, contact, profile)
6. Services page (citizens can request certificates, licenses)
7. Analytics dashboard (simple charts, complaint trends)
8. Multi-language support (Gujarati primary, English fallback)

DATABASE SCHEMA (Firestore):
Collections:
- citizens: {uid, phone, name, address, createdAt}
- complaints: {complaintId, citizenId, category, description, photo_url, latitude, longitude
- panchayat: {name, district, phone, email, budget_amount}
- services: {serviceId, name, description, documents_required}

REQUIRED PAGES:

CITIZEN SIDE:
1. Landing Page: Hero section, 3-4 feature bullets, "Register" / "Login" buttons. Include Gr
2. Register Page: Phone input + OTP verification (Firebase). Then collect name, village addr
3. Citizen Dashboard: Welcome msg, "Your complaints: X, Resolved: Y" stats, list of my compl
4. File Complaint Page:
   - Category dropdown (, , , ,)
   - Description text area (in Gujarati or English)
   - Photo upload (capture from phone camera, geo-tagged via Google Maps API)
   - Location picker on map
   - Submit → generate complaint ID, show success message with tracking number
5. Complaint Details Page: Show status (color: yellow=submitted, blue=in-progress, green=res
6. Services Page: List of available services (birth certificate, income certificate, etc.),
7. Panchayat Info Page: Name, phone, address, basic budget overview (simple bar chart showin

```
PANCHAYAT ADMIN SIDE:
1. Admin Login: Simple username/password (demo: username="panchayat", password="admin123").
2. Admin Dashboard:
   - KPIs: Total complaints (this month), Resolved (this month), Pending, Avg resolution day
   - Charts: Bar chart (complaints by category), Line chart (complaints over 7 days), Pie ch
3. All Complaints: Table view with columns: ID | Category | Status | Citizen Name | Submitte
4. Complaint Detail (Admin View): Show full info + photo, dropdown to update status, text ar
5. Analytics Page: Metrics + charts (simple bar, line, pie). Filter by date range.

DESIGN & UX REQUIREMENTS:
- Color theme: Green (#22c55e or #10b981) + Gold (#f59e0b or #fbbf24) + White background
- Icons: Use Heroicons or Lucide React (simple, clean)
- Mobile-first responsive design (Tailwind CSS breakpoints)
- Forms: Clear labels in Gujarati + English, input validation, error messages
- Buttons: Primary (green), secondary (outline), danger (red for delete/cancel)
- Complaint cards: Show category with icon, status with color badge, date, action button
- Dashboard charts: Use a lightweight library like Recharts or Chart.js (lightweight for hac

CRITICAL FEATURES:
1. Geo-tagging: Use Google Maps API to capture lat/long when citizen files complaint
2. Photo upload: Store in Google Cloud Storage, return URL, display in complaint details
3. Status tracking: Complaint color changes based on status (yellow → blue → green)
4. Notifications: Log that SMS/email would be sent (don't actually send; just show "Notifica
5. Validation: Phone number format, OTP, required fields, photo format
6. Error handling: Try-catch blocks, user-friendly error messages

IMPLEMENTATION PRIORITIES (for 24-48 hour hackathon):
MUST-HAVE (Core, hackathon demo):
- Citizen registration + login (OTP)
- File complaint with photo + geo-tagging
- Complaint tracking (my complaints list, details, status updates)
- Admin login + view all complaints
- Update complaint status (admin can change: submitted → in-progress → resolved)
- Dashboard KPIs + 1 simple chart

NICE-TO-HAVE (Time permitting):
- SMS notifications (log/demo only)
- Services page
- Analytics dashboard with multiple charts
- Panchayat budget page
- Social audit features
- Gram Sabha meeting tracker

NOT IN SCOPE (Skip for hackathon):
- Payment integration
- Advanced ML/AI
```

```
- Voice input (IVR)
- Offline mode
- Integration with external government APIs

CODE STRUCTURE:
Backend:
- /routes: auth.js, complaints.js, admin.js, panchayat.js, services.js
- /controllers: handle business logic
- /middleware: auth verification
- /utils: database queries, helpers
- .env: Firebase config, API keys

Frontend:
- /pages: Landing, Register, Dashboard, FileComplaint, ComplaintDetails, AdminDash, etc.
- /components: Header, Footer, ComplaintCard, StatusBadge, Chart, etc.
- /hooks: useAuth, useComplaints, useAdmin (custom hooks)
- /utils: api.js (Axios calls), formatters.js
- App.jsx: Routes
- index.css: Tailwind config

DEPLOYMENT:
- Backend: Deploy Node.js + Express to Firebase Functions or Render.com (free tier)
- Frontend: Deploy React to Firebase Hosting (zero-config)
- Database: Firebase Firestore (built-in)
- Live URL example: gram-sahay-demo.web.app

TESTING BEFORE DEMO:
1. Citizen: Register → File complaint with photo → See complaint in dashboard → See status u
2. Admin: Login → See all complaints → Update one complaint status to "in-progress" → See up
3. Mobile: Test on phone browser (iPad, Android phone if available)
4. Error cases: Submit empty form, upload wrong file type, test validation

DEMO SCRIPT (1-2 min):
"Gram Sahay is a hyperlocal panchayat complaint system. First, citizen registers with phone

IMPORTANT NOTES:
- Keep scope SMALL: Don't try to build 50 features. 5-6 core features done well > 20 feature
- Use FREE Google services: Firebase (free tier), Google Maps (free tier for hackathon), Goo
- Code quality: Clean, commented code. Beginners should understand. Not production-level; ha
- No hardcoding: Use .env for config. Use database for data, not JSON files.
- Error handling: All API calls should have try-catch. Show friendly error messages to users
- Responsive: Test on phone. All pages should work on mobile.
- UI: Clean, simple. Gram Sahay branding (green + gold). Gujarati text where applicable.

BUILD INCREMENTALLY:
Hour 1-4: Backend setup + APIs
```

```
Hour 4-8: Frontend auth + dashboard
Hour 8-12: File complaint feature
Hour 12-16: Admin panel
Hour 16-20: Polish UI, mobile responsive
Hour 20-24: Testing, bugs, prepare demo

GO BUILD. NO MISTAKES. SHIP IT.
```

PART D: KEY PRESENTATION POINTS

QUICK EXPLANATION (2 MINUTES)

"GRAM SAHAY COMPLAINT FLOW - STEP BY STEP:

1 **CITIZEN PHASE (Day 0)** - Citizen takes photo of problem with phone - Fills complaint form in Gujarati (name, category, description) - System automatically captures location (GPS) - Submits → Gets Complaint ID (example: #12345) - Receives SMS confirmation

2 **TALATI PHASE (Day 1-7)** - Talati (village secretary) gets INSTANT SMS alert - Logs into admin panel - Reviews: Photo clear? Location correct? Legitimate issue? - Decision: APPROVE (assign to worker) or REJECT (send reason) - If approved: Worker gets SMS assignment - Updates status: SUBMITTED → IN-PROGRESS - Citizen gets status update SMS

3 **WORKER PHASE (Day 1-7)** - Worker receives assignment SMS - Opens app, navigates to location (Google Maps) - Fixes the problem (light repair, garbage cleanup, etc.) - Takes photo of fixed problem - Marks as DONE in app - System notifies Talati

4 **TALATI APPROVAL (Day 1-7)** - Talati reviews worker's photo - If satisfied: Clicks CLOSE COMPLAINT - Status: SUBMITTED → IN-PROGRESS → RESOLVED (Green) - Citizen gets FINAL SMS: 'Problem solved! Rate our service'

5 **ESCALATION IF STUCK (Auto-triggered)** - Day 3: No response? SMS reminder to Talati - Day 7: Still silent? AUTO-ESCALATE to Block Officer - Day 14: Block silent? AUTO-ESCALATE to District Officer - Day 21: Still unresolved? GOES TO PUBLIC RECORD

6 **TRANSPARENCY & ACCOUNTABILITY** - Panchayat info page shows: Budget spent, complaints resolved, avg time - Citizens see: 'This month: 23 complaints, 21 resolved, 4.1 day avg' - Sarpanch presents data in monthly gram sabha meeting - No hiding, no lies, numbers speak

RESULT: - Citizens use system because they get RESPONSE - Talati works on time to avoid escalation - Transparent data builds TRUST - Panchayat can

focus on REAL problems instead of politics"

KEY DIFFERENTIATORS FROM MERI PANCHAYAT

Feature	Meri Panchayat (National)	Gram Sahay (Village Pilot)
Scale	2.65 lakh panchayats	1 village
Response Rate	~15% usage, low response	Expected 60-70% (because of escalation)
Escalation	Manual, slow	Automatic at Day 3, 7, 14
Talati Accountability	Low, can ignore	HIGH (escalates to higher authority)
Language	12 languages (generic)	Gujarati FIRST (village-specific)
UI Complexity	Complex (national features)	Ultra-simple (only 5 issue types)
Training	Online docs	College students visit every week
Feedback Loop	Exists but rarely used	Mandatory citizen rating
Timeline	Not strict	Auto-escalates if exceeded
Transparency	Budget info somewhere	CLEAR public dashboard

QUICK CHECKLIST FOR TOMORROW

Before you start coding:

- ☐ GitHub repo created + link shared
- ☐ Firebase project created + config ready
- ☐ Google Cloud Storage bucket created
- ☐ Google Maps API key generated
- ☐ Figma/design mockup (even simple sketches)
- ☐ Task board set up (Trello/GitHub Projects) with tasks
- ☐ Slack/Discord for team communication
- ☐ Assign roles: Frontend lead, Backend lead, DevOps/Deployment lead

Team structure (if 3-4 people): - **Person 1**: Backend APIs + Database - **Person 2**: Citizen UI (Register, Dashboard, File Complaint) - **Person 3**: Admin UI (Admin Panel, Complaint Management) - **Person 4** (if available): Testing + Deployment

By hour 12: - Backend APIs should be 80% done - Frontend skeleton done (pages structure, routing) - Database connected

By hour 20: - All features coded - Mobile responsive - Core bugs fixed

By hour 24: - Demo ready - PPT/presentation updated - Practice pitch (2-3 times)

YOU NOW HAVE YOUR COMPLETE ROADMAP + THEORY + TECH + PRESENTATION GUIDE.

GO BUILD. NO MISTAKES. SHIP IT TOMORROW.