🏗️ CitizenConnect - Backend Development Progress

📊 Overall Status

- **Phase 0**: ✅ Completed
- **Phase 1**: ✅ Completed
- **Phase 2**: 🔄 In Progress (90% done)
- **Phase 3-8**: ⏳ Upcoming

✅ Phase 0 - Planning & Architecture Setup (COMPLETED)

Step	Task	Status
0.1	Defined architecture (Mobile + Web + Shared Backend)	✅ Done
0.2	Created GitHub repo and folder structure	✅ Done
0.3	Installed dependencies (Node, Express, Prisma, TypeScript)	✅ Done
0.4	Planned database models & multi-hierarchy admin	✅ Done

✅ Phase 1 - Backend Core & Authentication (COMPLETED)

Step	Task	Status
1.1	Initialize project & setup folders	✅ Done
1.2	Initialize TypeScript (tsconfig.json, scripts)	✅ Done
1.3	Database modeling with Prisma (PostgreSQL setup, schema design)	✅ Done
1.4	Express Server Setup (app.ts, server.ts)	✅ Done
1.5	Folder structure (src/routes, src/controllers, src/middlewares, src/utils)	✅ Done
1.6	Prisma formatting & User model creation	✅ Done
1.7	Authentication routes (/register, /login)	✅ Done
1.8	JWT & bcrypt integration (secure auth system)	✅ Done
1.9	Role-based authorization (Admin & Citizen access control)	✅ Done

---

🔄 Phase 2 - Complaint Management Module (IN PROGRESS - 90%)

✅ Completed in Phase 2:

Step	Task	Status
2.1	Complaint model setup in Prisma	✅ Done
2.2	Seed file for domain/category dropdowns	✅ Done
2.3	Complaint filtering & tracking by roles	✅ Done
2.4	Admin complaint dashboard routes	✅ Done
2.5	Role-restricted complaint updates	✅ Done
2.6	Real-time notifications (Socket.IO setup)	✅ Done
2.6a	Socket authentication middleware	✅ Done
2.6b	Socket event handlers & room management	✅ Done
2.6c	Emit notifications on complaint create/update	✅ Done

🔜 Remaining in Phase 2:

Step	Task	Status	Priority
2.7	**Admin Assignment Flow** - Assign complaints to departments/employees	⏳ Next	HIGH
2.8	**Department-wise Complaint Routing** - Auto-assign based on domain/category	⏳ Pending	HIGH
2.9	**Complaint Media Upload** - Image/video upload (Multer/Cloudinary)	⏳ Pending	MEDIUM
2.10	**Complaint Edit/Delete** - Citizens can edit before admin review	⏳ Pending	LOW

⏳ Phase 3 - Live Complaint Tracking & Analytics (UPCOMING)

Step	Task	Dependencies	Priority
3.1	**5-Stage Tracking Pipeline** (Raised → Acknowledged → In Progress → Resolved → Closed)	Phase 2 complete	HIGH
3.2	**Status Update History** - Track all status changes with timestamps	Step 3.1	HIGH
3.3	**Notification System Enhancement** - Push notifications for each status change	Socket.IO done	MEDIUM
3.4	**User Feedback & Rating** - Citizens rate resolved complaints	Step 3.1	MEDIUM
3.5	**Complaint Analytics API** - Stats for admins (total, pending, resolved, by category)	Step 3.1	HIGH
3.6	**Resolved Complaints History** - Archive and retrieve old complaints	Step 3.1	LOW

⏳ Phase 4 - Heatmap & Data Visualization (UPCOMING)

Step	Task	Dependencies	Priority
4.1	**Geo-location Integration** - Add lat/long to complaint model	Phase 3	HIGH
4.2	**Severity Zones API** - Calculate complaint density (Green → Red)	Step 4.1	HIGH
4.3	**Area-wise Filters** - Filter by ward, zone, district	Step 4.1	MEDIUM
4.4	**Issue-wise Filters** - Filter by domain/category	Phase 2 done	MEDIUM
4.5	**Real-time Map Data Endpoint** - API for heatmap rendering	Step 4.2	HIGH

⏳ Phase 5 - Multi-Hierarchy Admin Dashboards (UPCOMING)

Step	Task	Dependencies	Priority
5.1	**Employee Dashboard API** - View assigned complaints, update status	Phase 3	HIGH
5.2	**Ward Officer Dashboard API** - View ward-specific complaints	Step 4.1	HIGH
5.3	**Department Admin Dashboard API** - Manage department employees & complaints	Phase 2.7	HIGH
5.4	**City Admin Dashboard API** - View all city complaints, analytics	Phase 3.5	HIGH
5.5	**Mayor Dashboard API** - High-level city-wide analytics	Phase 3.5	MEDIUM
5.6	**Role-based Data Access Control** - Ensure proper permissions	Phase 1 done	HIGH

⏳ Phase 6 - Mobile Application (FRONTEND - NOT BACKEND)

This phase will be handled after backend completion

⏳ Phase 7 - Cloud Deployment & Integration (UPCOMING)

Step	Task	Dependencies	Priority
7.1	**Environment Configuration** - Production .env setup	All phases	HIGH
7.2	**Database Migration to Cloud** - Supabase/Neon PostgreSQL	Phase 1	HIGH
7.3	**Backend Deployment** - Render/Railway/AWS	Step 7.1	HIGH
7.4	**Socket.IO in Production** - Configure CORS, WSS	Phase 2.6	HIGH
7.5	**Media Storage Setup** - Cloudinary/S3 integration	Step 2.9	MEDIUM
7.6	**API Performance Optimization** - Caching, rate limiting	All phases	MEDIUM

⏳ Phase 8 - Testing, Optimization & Documentation (UPCOMING)

Step	Task	Dependencies	Priority
8.1	**Postman Test Collection** - Complete API test suite	All phases	HIGH
8.2	**API Documentation** - Swagger/Postman docs	All phases	HIGH
8.3	**Error Handling Review** - Standardize error responses	All phases	MEDIUM
8.4	**Security Audit** - JWT expiry, input validation, SQL injection prevention	All phases	HIGH
8.5	**Logging & Monitoring** - Winston/Morgan setup	All phases	MEDIUM
8.6	**README & Documentation** - Setup guide, architecture docs	All phases	MEDIUM

🎯 Immediate Next Steps (Priority Order)

🔥 CRITICAL - Complete Phase 2 (This Week)

1. **Step 2.7 - Admin Assignment Flow** ⭐ TOP PRIORITY
   - Create endpoint: `PUT /api/complaints/:id/assign`
   - Assign complaint to specific department/employee
   - Add `assignedTo` field to Complaint model
   - Emit socket notification to assignee
2. **Step 2.8 - Department-wise Routing**
   - Auto-assign based on domain/category mapping (DomainCategory model)
   - Create assignment logic in controller
3. **Step 2.9 - Media Upload** (Optional for now)
   - Can be skipped for MVP, implement later

📅 Week 2 - Start Phase 3

4. **Step 3.1 - 5-Stage Tracking Pipeline**
   - Update status enum: `Raised → Acknowledged → In Progress → Resolved → Closed`
   - Modify status update logic
5. **Step 3.5 - Analytics API**
   - Create `/api/admin/analytics` endpoint
   - Stats: total complaints, by status, by category, response time

---

📈 Progress Metrics

Metric	Count	Percentage
Total Backend Steps	45	100%
Completed	25	55%
In Progress	3	7%
Remaining	17	38%

🗂️ Current File Structure Status

```
citizenconnect-backend/
├── src/
│   ├── controllers/
│   │   ├── authController.ts          ✅ Done
│   │   ├── complaintController.ts     ✅ Done (90%)
│   │   ├── domainController.ts        ✅ Done
│   │   └── adminController.ts         ⏳ TODO (Step 2.7)
│   ├── middlewares/
│   │   ├── authMiddleware.ts          ✅ Done
│   │   ├── roleMiddleware.ts          ✅ Done
│   │   ├── errorHandler.ts            ✅ Done
│   │   ├── logger.ts                  ✅ Done
│   │   └── socketAuthMiddleware.ts    ✅ Done (NEW)
│   ├── routes/
│   │   ├── authRoutes.ts              ✅ Done
│   │   ├── complaintRoutes.ts         ✅ Done
│   │   ├── adminRoutes.ts             ✅ Done
│   │   ├── domainRoutes.ts            ✅ Done
│   │   └── protectedRoutes.ts         ✅ Done
│   ├── sockets/
│   │   └── socketHandlers.ts          ✅ Done (NEW)
│   ├── prisma/
│   │   ├── schema.prisma              ✅ Done
│   │   ├── seed.ts                    ✅ Done
│   │   └── client.ts                  ✅ Done
│   ├── utils/
│   │   └── auth.ts                    ✅ Done
│   ├── app.ts                         ✅ Done
│   └── server.ts                      ✅ Done
```

🎓 Key Achievements So Far

✅ Complete authentication system with JWT

✅ Role-based access control (5 roles)
✅ Complaint CRUD operations
✅ Real-time notifications with Socket.IO
✅ Status tracking system
✅ Domain/Category management
✅ Database schema with relationships
✅ Middleware architecture (auth, logging, error handling)

📝 Notes

- **Focus**: Complete Phase 2 before moving to Phase 3
- **Testing**: Test each endpoint in Postman as you build
- **Socket.IO**: Already working, just needs client testing
- **Database**: Keep running migrations as models change
- **Next Big Feature**: Admin assignment workflow (Step 2.7)

🚀 Recommended Development Order

```
WEEK 1:  Complete Phase 2 (Steps 2.7, 2.8)
WEEK 2:  Phase 3 - Tracking Pipeline & Analytics
WEEK 3:  Phase 4 - Geo-location & Heatmap APIs
WEEK 4:  Phase 5 - Admin Dashboard APIs
WEEK 5:  Phase 7 - Deployment preparation
WEEK 6:  Phase 8 - Testing & Documentation
```

This plan assumes ~2-3 hours of work per day. Adjust timeline based on your availability.

Ready to continue? Let's tackle Step 2.7 - Admin Assignment Flow next! 🎯