

Election Mitra - Backend Development Roadmap

Phase 0: Pre-development Planning

Define:

- User Roles: Voter, Candidate, Volunteer, Admin
- Core Features: Auth, Dashboards, Campaigns, Posts, Reports, Chat
- Entity Relationships: Who can interact with whom (e.g., candidate <-> volunteer)

Deliverables:

- Feature list & MVP scope
- ERD (Entity Relationship Diagram)
- REST API route sketch
- Folder structure convention

Phase 1: Project Setup

Tasks:

1. Initialize Project:
 - Node.js + Express.js
 - TypeScript (optional)
 - Suggested Folder Structure
2. Environment Setup:
 - .env setup
 - Logging, CORS, Helmet
3. Database Setup:
 - MongoDB (Mongoose)
 - Collections: Users, Posts, Reports, Campaigns, Volunteers, Tasks, Chats

Phase 2: Authentication & Authorization

Features:

- Role-based signup/login
- OTP/email or password auth
- JWT tokens and role middleware

Routes:

POST /auth/signup
POST /auth/login
GET /auth/verify
POST /auth/logout

Phase 3: User Role Management

Tasks:

- Profile creation per role
- Extra fields by role
- Admin access levels

Routes:

GET /users/me

Election Mitra - Backend Development Roadmap

PUT /users/profile
GET /users/:id

Phase 4: Dashboards & Core Logic

Voter Dashboard:

- Feed, bookmark, complaint, view candidates

Candidate Dashboard:

- Manifesto, posts, provisions, tasks

Volunteer Panel:

- Apply, check status, certificate

Admin Panel:

- Approvals, duty assignment, report review

Phase 5: Posts, Campaigns & Reports

Posts:

- CRUD, likes, comments, tags, media

Campaigns:

- Browse/filter/follow

Reports:

- File reports (anonymously or logged)
- Admin review

Phase 6: Chat System

Tasks:

- Socket.IO for real-time chat
- Candidate-volunteer rooms

Sockets:

```
socket.on('join-room')  
socket.on('send-message')
```

Phase 7: File Uploads

Use:

- Multer (uploads)
- Cloudinary or AWS S3

For:

- Profile pics, ID proofs, media, reports

Phase 8: Mobile App Integration API

Tasks:

Election Mitra - Backend Development Roadmap

- Enable CORS
- Optimize for mobile use
- QR-based download support

Phase 9: Security & Optimization

Tasks:

- Rate limiting
- Input sanitization
- HTTPS
- Audit logs

Phase 10: Analytics & Reporting

For Admin:

- Users, reports, tasks analytics
- Heatmaps for elections

Phase 11: Testing & Documentation

Tests:

- Unit and integration tests
- API docs (Swagger/Postman)

Final Phase: Deployment

Tasks:

- Deploy: AWS, Vercel, Render, etc.
- Setup CI/CD
- Load tests and health checks

Technologies Summary

Server: Node.js + Express

Auth: JWT, bcrypt, OTP API

Database: MongoDB + Mongoose

Real-time: Socket.IO

Uploads: Multer + Cloudinary/S3

Admin: Role-based middleware

Chat: Socket.IO

Testing: Jest, Supertest

Docs: Swagger / Postman

Final Advice

- Separate user roles and secure APIs
- Modularize features
- Log sensitive actions
- Start MVP with core auth + campaign + reports