Election Mitra – Backend Work Summary

## 1. Project Initialization

- Project setup using Node.js + Express.  
- Directory structure defined for routes, controllers, models, middlewares, utils.  
- MongoDB setup with Mongoose and basic environment configuration.  
- Git repository initialized, `.env` file created, and development dependencies installed.

## 2. Database Models Designed

- \*\*User Model\*\*: Includes role, region, supportingParty, verification, preferences.  
- \*\*Candidate Model\*\*: Includes userId, manifesto, party, provisions, volunteers.  
- \*\*Post Model\*\*: Media, tags, engagement metrics, region-based visibility.  
- \*\*Volunteer Application Model\*\*: Tracks application docs, types, status, certificate.  
- \*\*Complaint Model\*\*: Allows anonymous/identified complaint filing, media evidence.  
- \*\*Campaign Model\*\*: Tracks visibility, filters by party/region, engagement.

## 3. Authentication and Authorization

- \*\*Signup/Login APIs\*\*: Role-based, with OTP/Password logic.  
- \*\*Session/Token Handling\*\*: JWT tokens issued and validated per role.  
- \*\*Guest Access\*\*: Visitors can fetch trending posts without full login.  
- \*\*Middleware\*\*: Role-based access guards, error handling, region checker.

## 4. API Implementation

- \*\*User Routes\*\*: Feed, dashboard view, preferences update.  
- \*\*Candidate Routes\*\*: Profile edit, view volunteers, upload provisions, post campaigns.  
- \*\*Post Routes\*\*: Create post, view post, like/comment/report functionality.  
- \*\*Volunteer Routes\*\*: Apply, verify status, assigned task list, certificate download.  
- \*\*Complaint Routes\*\*: Submit complaint, track status, admin resolve.  
- \*\*Analytics Routes\*\*: Party-wise support per region, winning probability estimation.

## 5. Dashboard Functional Logic

- \*\*Voter Dashboard\*\*: Region-specific posts and candidates, support stats, volunteer options.  
- \*\*Candidate Dashboard\*\*: Post reach, provision listing, volunteer management.  
- \*\*Volunteer Panel\*\*: Application status, tasks, verification badge.  
- \*\*Admin Panel\*\*: User moderation, analytics insights, complaint management.

## 6. Analytics & Regional Features

- Users tagged with region and supporting party.  
- Data used to compute party-wise support and probable winner.  
- Guest users shown trending posts by engagement (likes/comments).

## 7. Current Progress Summary

- ✅ Database models finalized and implemented.  
- ✅ Core authentication system working (login/signup/guest).  
- ✅ Regional logic integrated into user profiles and post filters.  
- ✅ API endpoints developed and documented for major features.  
- ⏳ Dashboard UI/backend sync in progress.  
- ⏳ Admin features for moderation and analytics are next in queue.

### 📁 election-mitra-backend/

#### ┣ 📁 src/ – Core backend logic

| **Folder/File** | **Purpose** |
| --- | --- |
| **┣ 📁 config/** |  |
| ┃ ┗ db.js – Database connection |  |
| ┃ ┗ jwt.js – JWT config and helpers |  |

**┣ 📁 controllers/**  
┃ ┣ authController.js – Login/signup/OTP  
┃ ┣ userController.js – Feed, region/party selection, bookmarks  
┃ ┣ candidateController.js – Manifesto, posts, volunteers  
┃ ┣ postController.js – Create/view posts, like/comment/report  
┃ ┣ volunteerController.js – Application, status, certificate  
┃ ┣ adminController.js – Approvals, moderation, analytics  
┃ ┗ reportController.js – Complaint logic

**┣ 📁 middlewares/**  
┃ ┣ authMiddleware.js – Role/region protection  
┃ ┗ errorHandler.js – Error formatting

**┣ 📁 models/**  
┃ ┣ User.js – Voter/Candidate/Admin  
┃ ┣ Candidate.js – Manifesto, volunteers, provisions  
┃ ┣ Post.js – Media/news/posts  
┃ ┣ VolunteerApplication.js – Application form and docs  
┃ ┣ Complaint.js – User/guest filed issues  
┃ ┗ Campaign.js – Candidate campaigns and filters

**┣ 📁 routes/**  
┃ ┣ authRoutes.js  
┃ ┣ userRoutes.js  
┃ ┣ candidateRoutes.js  
┃ ┣ postRoutes.js  
┃ ┣ volunteerRoutes.js  
┃ ┣ adminRoutes.js  
┃ ┗ reportRoutes.js

**┣ 📁 services/**  
┃ ┣ otpService.js – OTP logic  
┃ ┣ analyticsService.js – Support stats, win probability  
┃ ┗ notificationService.js – Email/SMS/push alerts

**┣ 📁 utils/**  
┃ ┣ validator.js – Validation logic  
┃ ┣ constants.js – Enums like roles, status  
┃ ┗ fileUpload.js – File/media upload helper

**┣ server.js** – Entrypoint  
**┣ app.js** – App setup and routing

### 📁 public/

* Static files (images, uploads if not using cloud storage)

### 📁 tests/

* Unit/integration tests

### Root Level Files:

| **File** | **Purpose** |
| --- | --- |
| package.json – Dependencies and scripts |  |
| .env – Environment variables (DB URI, JWT\_SECRET, etc.) |  |
| .gitignore – Ignored files/folders |  |
| README.md – Project overview and setup |  |