Medigen Sprint Planning

Main Objectives (Sequence Overview)

- ❖ Project Setup & Architecture
- ❖ Database Design & Connection (MongoDB / PostgreSQL)
- ❖ Core API Endpoints (User, Auth, Data Models)
- ❖ Integration with Frontend
- ❖ Testing, Validation & Error Handling
- ❖ Deployment (Backend + Environment Config)

30-Day Backend Development Plan

Week 1 — Setup & Base Structure
Goal: Create a solid foundation for backend development.

Day 1–2:

- Initialize backend project (npm init -y)
- Install dependencies: express, cors, dotenv, mongoose (or prisma/pg), nodemon, bcrypt, jsonwebtoken
- Setup folder structure: backend/ ├── src/ │ ├── config/ │ ├── controllers/ │ ├── middleware/ │ ├── models/ │ ├── routes/ │ ├── utils/ │ └── server.ts ├── package.json ├── .env ├── tsconfig.json

Day 3–4:

- Setup Express server (basic /api/health route)
- Add dotenv config and connect it with environment variables.
- Add CORS setup to allow frontend calls.

Day 5–7:

- Design Database schema (e.g., User, Posts, Articles, Doctors, Forum Threads).
- Connect database (MongoDB via Mongoose or PostgreSQL via Prisma).
- Test connection with dummy data.

Week 2 — Core APIs (Auth + User)
Goal: Create and test authentication + user management system.

Day 8–10:

- Implement User model (fields: name, email, password, role, etc.)
- Add Auth controller:
- /register
- /login
- /logout (optional)
- Encrypt passwords with bcrypt.
- Generate JWT tokens.

Day 11–12:

- Add middleware for authentication (JWT verify).
- Create protected routes (like /profile).

Day 13–14:

- Test with Postman.
- Handle all error responses properly.
- Add reusable response format (e.g., { success, message, data }).

Week 3 — Feature APIs
Goal: Build endpoints for all main features your frontend needs.

Day 15–17:

- Create Articles API:
- GET all articles
- POST new article
- PUT/DELETE article (admin only)

Day 18–19:

- Create Forum API:
- Create new post/thread
- Add comment
- Fetch discussions

Day 20–21:

- Create Doctors API:
- List doctors
- Fetch single doctor details
- Add or update (admin)

Week 4 — Integration, Testing & Deployment

Goal: Connect backend to frontend, handle validation, and deploy.

Day 22–23:

- Use frontend pages (e.g., LoginPage.tsx, ProfilePage.tsx, etc.)
→ integrate backend APIs via Axios.
- Setup .env for frontend to store backend base URL.

Day 24–25:

- Add data validation (Joi, Zod, or express-validator).
- Add global error handler.
- Write simple unit tests (Jest or Supertest).

Day 26–27:

- Finalize API documentation (Swagger or simple README.md).
- Test all endpoints again with real data.
- Clean code, refactor redundant parts.

Day 28–30:

- Deploy backend (Render / Railway / Vercel / AWS).
- Connect frontend to deployed backend.
- Final full system testing.
- Celebrate 🎉 (then maybe optimize performance/logging).

Optional Enhancements

- Add Socket.io for live chat/forum.
- Add AI Assistant endpoint (use Gemini API integration).
- Add role-based access control (admin/user).
- Use Multer for file uploads (e.g., profile images).

Medigen Sprint Planning Objectives

- ❖ Project Setup & Architecture
  - ➢ (Omar Faruk Fahim -22235103132)
- ❖ Full stack developer
- ❖ (Omar Faruk Fahim -22235103132)
- ❖ Database Design & Connection
  - ➢ (Minhajur Rahman-22235103129)
- ❖ Core API Endpoints (User, Auth, Data Models)
  - ➢ (Minhajur Rahman-22235103129)
- ❖ Testing, Validation & Error Handling
  - ➢ (Muhammad Nazmus Sakib -22235103139)
- ❖ Deployment
  - ➢ (Shariar Soton-22235103105)
- ❖ UI/UX Designer
- ❖ (Fazla Rabbi- 22235103093)