

# Real-Time Digital Doppelgänger - Backend Setup + Feature Ideas

## BACKEND SETUP (MongoDB + Express):

1. Initialize backend folder using npm.
2. Create folder structure:
  - /models/Activity.js
  - /controllers/activityController.js
  - /routes/activityRoutes.js
  - server.js
  - .env (Mongo URI and Port)
3. Implement Express server, connect MongoDB using Mongoose.
4. Create Activity schema (activity, mood, timestamp, simulation flag).
5. Create POST and GET routes for logging and retrieving activity data.

Test using Postman or Thunder Client.

## Suggested Features for the Real-Time Digital Doppelgänger Project:

1. User Authentication (Register/Login with JWT).
2. Log daily activities and mood ratings.
3. Simulate alternate actions and visualize the difference.
4. Timeline view showing real vs simulated choices (Chart.js/D3.js).
5. Insights: Mood trends, most common regretful activities, best times of day.
6. Mood heatmap or calendar view.
7. Monthly summary export (PDF or shareable report).

8. AI/Rule-based engine for predicting outcome of different choices.
9. Voice input for logging ('I studied for 1 hour').
10. Gamification (streaks, productivity score, daily challenges).
11. Light/dark mode toggle.
12. Deployment using Render (backend) + Netlify or Vercel (frontend).