1. Project Structure
- Always generate a full MERN project structure with separate backend and frontend folders.
- Backend must include models, routes, controllers, server.js.
- Frontend must include components, pages, App.js, index.js.
- Include a .env.example file for environment variables.

2. Coding Standards
- Use ES6+ syntax consistently.
- Use arrow functions, template literals, and destructuring.
- Name variables in camelCase, components in PascalCase, collections in plural lowercase.
- Always include comments explaining logic for functions, APIs, and components.

3. Backend Rules
- Use Express.js for routing.
- Use Mongoose for MongoDB models and ensure schema validation.
- Include async/await with proper try/catch error handling.
- Return JSON responses for all APIs with consistent structure:
  { "success": true/false, "data": {}, "message": "..." }
- Include basic security middleware: cors, helmet, dotenv.

4. Frontend Rules
- Use React functional components.
- Use React Router for navigation between pages.
- Manage state using hooks (useState, useEffect) or context API.
- Style components consistently (CSS modules, Tailwind, or inline styles).
- Components should be reusable and modular.

5. Database Rules
- Use MongoDB with clear collection names and field names.
- Include example documents for testing.
- Validate incoming data before saving.
- Include sample queries for CRUD operations.

6. Error Handling
- Backend: return proper HTTP status codes (200, 400, 404, 500).
- Frontend: show user-friendly error messages on API failures.
- Always log errors for debugging.

7. AI Behavior Rules

- Generate full files, never snippets unless asked.
- Include dry-run explanation for complex logic.
- Suggest best practices, optimizations, and scalability improvements.
- Avoid hardcoded secrets, dummy data should be realistic.
- Ensure code is directly runnable on Replit without extra manual fixes.

8. Documentation Rules

- Include README.md with:
  - Project description
  - Setup instructions
  - API documentation
  - Component structure explanation
- Inline comments for every function, model, and component.

9. Testing & Validation

- Include example API tests (Postman/Insomnia).
- Include frontend example interactions for demonstration.
- Validate all data flows: form input -> backend -> DB -> frontend.