Machine Test Requirement Document

Duration: 8–10 Hours
Submission: GitHub/GitLab repo link + deployed project link

Project Overview

You are required to build a **Blog Application with User Management** using **Next.js**. The application must include both frontend and backend functionality. Backend can be implemented using **Next.js API routes** or a separate **Node.js/Express server**.

The goal of this machine test is to assess your ability to design a production-like application with **authentication, CRUD operations, and deployment**.

Technical Requirements

1. Tech Stack

- **Frontend**: Next.js (App Router or Pages Router – your choice)
- **Backend**:
  - Option 1: Next.js API Routes
  - Option 2: Express.js + MongoDB (separate backend)
- **Database**: MongoDB (preferred)
- **Authentication**: JWT / NextAuth.js
- **Styling**: Tailwind CSS / CSS Modules / Any UI Library (Mantine, Antd, Chakra, etc.)

---

2. Core Features

Authentication & User Management

- User Registration (name, email, password)
- User Login (JWT or NextAuth session handling)
- Password should be securely hashed (bcrypt or similar)
- Role-based access:
  - **Admin**: Can manage (create/update/delete) all posts and users
  - **User**: Can create, edit, delete only their own posts

Blog Management

- Create, Read, Update, Delete (CRUD) blog posts
- Blog post should have:
  - Title
  - Content (rich text or plain)
  - Author (linked to user)
  - Created/Updated date

- Public Blog Listing Page (anyone can view posts without login)
- Single Blog Detail Page

User Profile

- User can view and edit their profile (name, email, password change)
- User can see their own posts

Extra (Bonus if done)

- Search & Filter posts (by title/author)
- Pagination or infinite scroll for blog listing
- Commenting system (optional but adds weight)

---

3. Deployment & Submission

- **Code**: Push to GitHub/GitLab (include `README.md` with setup instructions)
- **Deployment**:
    - Frontend/Next.js: Deploy to Vercel/Netlify
    - Backend/Database: Deploy on Render/Heroku/Railway (or host API in same Next.js project)
    - MongoDB: Use MongoDB Atlas (free tier)
- **Submission**:
    - GitHub/GitLab Repository Link
    - Deployed Live URL

Deliverables Checklist

- User authentication (register/login/logout with JWT or NextAuth)
- Role-based access (Admin/User)
- Blog CRUD operations
- Public blog listing + detail view
- User profile management
- Database integration (MongoDB Atlas)
- Project deployed and live
- Clean, reusable code with proper folder structure

Evaluation Criteria

- **Code Quality & Structure** (reusable components, clean API design)
- **Correctness of Features** (auth, blog CRUD, role handling)
- **UI/UX** (doesn’t need to be fancy, but functional and clear)
- **Database Design** (schema relationships for users & posts)
- **Deployment** (must be live and working)
- **Git Hygiene** (meaningful commits, not a single dump commit)

***Note**: Bonus features are optional. Focus on **core requirements** first.*