# Technical Assessment Task

**Requirements**: Laravel, Docker, Nginx/Apache2, API

## Project Overview

Create a RESTful API for a simple Blog Management System with user roles and permissions.

### Technical Requirements

#### 1. Environment Setup

* Set up the project using Docker and Docker Compose
* Configure Nginx/Apache2 server
* Create a docker-compose.yml with:
  + PHP 8.1+
  + MySQL/PostgreSQL
  + Nginx/Apache2

#### 2. Core Features Implementation

##### Authentication System

* Implement Sanctum authentication
* Create registration and login endpoints
* Implement password reset functionality

##### Blog Management

* CRUD operations for blog posts
* Posts should have:
  + Title
  + Content
  + Categories (many-to-many relationship)
  + Tags (many-to-many relationship)
  + Featured image
  + Status (draft/published)
  + Author relationship

##### Role-Based Access Control

* Implement roles:
  + Admin
  + Editor
  + Author
  + Reader
* Create permissions:
  + manage\_posts
  + publish\_posts
  + edit\_posts
  + delete\_posts
  + manage\_categories
  + manage\_users

#### 3. Technical Implementation Requirements

##### Laravel Features Usage

* Implement at least one Trait (e.g., for image handling)
* Use Form Requests for validation
* Implement API Resources for response formatting
* Use proper Controller structure with SOLID principles
* Implement proper database migrations and seeders
* Use Laravel's built-in authentication system

##### API Documentation

* Implement Swagger/OpenAPI documentation
* Document all endpoints with:
  + Request parameters
  + Response format
  + Authentication requirements
  + Example requests and responses

##### ****Optional:**** Testing

* Write PHPUnit/Pest tests for:
  + Authentication flow
  + CRUD operations
  + Permission checks

## Evaluation Criteria

### 1. Code Quality

* Code organization and structure
* Error handling
* Input validation

### 2. Technical Requirements

* Docker setup functionality
* API implementation
* Database design
* Authentication and authorization implementation

### 3. Documentation

* API documentation quality
* Code comments
* Setup instructions

### 4. Extra Points

* Implementing tests

## Submission Requirements

1. Provide a GitHub repository with:

* Complete source code
* Docker configuration
* Database migrations and seeders
* API documentation
* README.md with setup instructions