Practice Project

# Project Overview

# **CMS (Content Management System)** is a self-practice project to improve backend API structure, database design, and frontend development using **Java Spring Boot**, **MySQL**, and **Docker**. The project also aims to explore web security and authentication, version control (Git), and containerized environments for consistent deployment.

# Technology Stack

|  |  |
| --- | --- |
|  |  |
| **Layer** | | **Technology** |
| **Backend** | | Java 21, Spring Boot 3.4.3 |
| **Frontend** | | HTML/CSS, (future: Thymeleaf or React?) |
| **Database** | | MySQL 8 |
| **Authentication** | | Spring Security (planned) |
| **Containerization** | | Docker, Docker Compose |
| **Build Tool** | | Maven |
| **Version Control** | | Git, GitHub |

# System Design

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

AI 產生的內容可能不正確。

# Project requirements

## Functional requirements

Allow users to create, edit, delete and publish blog posts.

Allow users to create, login/logout and manage their account.

Allow users to use tag, title and time to filter blog.

Allow admin to hide or delete blog and comment.

Provide 2-factor authentication for login

## Non-functional requirements

Support markdown and editor for easier content formatting.

Prevent SQL injection in both blog and comment.

Secure user passwords so that only the user would know his/her own password.

Ensure access control only the owner can edit the blog.

Provide a responsive layout for full screen, half screen and mobile.

# Database Design

CMS’s Database contain 7 tables:

1. post

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| id | Int | No | The id of the post. Primary key, auto\_increment |
| title | Varchar(255) | No | The title of the post, can be used as search/filter function |
| content | Test | No | The text content of the post |
| user\_id | Int | Yes | The owner of the post |
| Category\_id | Int | Yes | The category reference id |
| created\_at | Timestamp | Yes | The created time stamp |
| Updated\_at | timestamp | yes | The last update time stamp |

1. post\_images

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| Id | Int | No | The id of image. Primary key, auto\_increment |
| Post\_id | Int | Yes | The id of the post which have this image |
| Image\_url | Varchar(255) | No | The path to the image |

1. post\_tags

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| Post\_id | int | no | The id of the post |
| Tag\_id | Int | no | The id of the tag |

1. tags

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| id | int | No | The id of the tag |
| name | Varchar(50) | No | The name of the tag |

1. comments

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| Id | int | No | The id of the comment |
| Post\_id | Int | Yes | The post\_id of the post |
| User\_id | Int | Yes | The user\_id of the comment owner |
| Content | Text | No | The text content of the comment |
| Created\_at | Timestamp | Yes | The timestamp when create this comment |

1. categories

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| Id | int | No | The id of the categories |
| name | Varchar(100) | No | The name of the categories |

1. users

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Null | Description |
| id | int | no | The id of the user |
| username | Varchar(50) | no | The username of the user account |
| email | Varchar(100) | no | The email address of the user account |
| Password\_hash | Varchar(255) | no | The hashed password of the user account. It been salted and even admin can not view the password |
| role | Enum(‘admin’,’author’,’subscriber’) | Yes | The role of the user account.  Admin can delete post  Author can create post  Subscriber can read post and leave comment |
| Created\_at | timestamp | Yes | The timestamp of account creation |

# API Table