

📌 صورت مسئله پروژه: پیاده سازی وبلاگ با Spring Boot

شما باید یک وبلاگ ساده با استفاده از **Spring Boot + Thymeleaf + Spring Security + PostgreSQL** پیاده سازی کنید. پروژه باید تمام بخش های زیر را شامل شود:

♦ بخش ۱: احراز هویت و کاربرها

- کاربران می توانند ثبت نام کنند و وارد سایت شوند.
- نقش ها:
 - **ROLE_USER**: کاربر عادی که می تواند پست ها را ببیند و نظر بدهد.
 - **ROLE_ADMIN**: مدیر که می تواند همه چیز را مدیریت کند (کاربران، پست ها، دسته ها و نظرات).
- رمز عبور باید با **BCrypt** ذخیره شود.
- لاگین و ثبت نام باید با صفحات **Thymeleaf** انجام شود.

♦ بخش ۲: مدل ها (Entities)

شما باید جداول زیر را در دیتابیس PostgreSQL طراحی کنید:

- User**
 - فیلدها: id, username, email, password, fullName, roles, createdAt
- Role**
 - فیلدها: id, name (ROLE_USER, ROLE_ADMIN)
- Post**
 - فیلدها: id, title, slug, content, excerpt, status (DRAFT/PUBLISHED), author, createdAt, updatedAt, publishedAt, views
 - هر پست متعلق به یک نویسنده (User) است.
- Category**
 - فیلدها: id, name, slug, description
 - هر پست می تواند یک دسته داشته باشد.
- Tag**
 - فیلدها: id, name, slug
 - هر پست می تواند چند تگ داشته باشد.
- Comment**
 - فیلدها: id, content, authorName, authorEmail, approved, createdAt

- هر کامنت متعلق به یک پست است.

◆ بخش ۳: امکانات کاربر عادی

- صفحه اصلی (/) باید لیست پست‌های منتشر شده را نمایش دهد.
- امکان مشاهده جزئیات هر پست (/posts/{slug}) همراه با نظرات آن وجود داشته باشد.
- کاربر می‌تواند نظر (Comment) برای پست‌ها ثبت کند (بعد از تأیید ادمین نمایش داده می‌شود).
- امکان جستجو در عنوان و محتوای پست‌ها فراهم باشد.
- لیست دسته‌ها و تگ‌ها باید در سایت نمایش داده شود و با کلیک روی هر کدام، پست‌های مرتبط فیلتر شوند.

◆ بخش ۴: امکانات مدیر (Admin Panel)

- آدرس admin/ برای مدیر در دسترس است.
- مدیر باید بتواند:
- 1. پست‌ها را **CRUD** کند (ایجاد، خواندن، ویرایش، حذف).
- 2. دسته‌ها و تگ‌ها را مدیریت کند.
- 3. کاربران را مدیریت کند (فعال/غیرفعال).
- 4. نظرات را تأیید یا رد کند.

◆ بخش ۵: امنیت (Spring Security)

- صفحات عمومی مثل /auth/**, /posts/**, / برای همه آزاد است.
- مسیر admin/** فقط برای ROLE_ADMIN در دسترس است.
- از CSRF در فرم‌ها استفاده کنید.
- بعد از لاگین موفق، کاربر به صفحه اصلی منتقل شود.

◆ بخش ۶: صفحات Thymeleaf

- صفحه اصلی: نمایش لیست پست‌ها با pagination.
- صفحه جزئیات پست: نمایش متن کامل، نویسنده، تاریخ انتشار و نظرات.

- صفحه لاگین و ثبت نام.

- پنل ادمین:

- لیست پست‌ها

- فرم ساخت/ویرایش پست

- مدیریت دسته‌ها و تگ‌ها

- مدیریت نظرات

♦ خروجی مورد انتظار

در پایان، پروژه شما باید یک وبلاگ ساده اما کامل باشد که:

- کاربران بتوانند ثبت نام و ورود کنند.

- مدیر بتواند پست‌ها را منتشر یا مدیریت کند.

- کاربران بتوانند پست‌ها را ببینند، جستجو کنند و نظر ثبت کنند.

- تمام صفحات با Thymeleaf ساخته شوند.

- دیتابیس PostgreSQL به درستی مدیریت شود.

👉 شما باید کد خود را در یک مخزن Git قرار دهید و هر بخش (Auth, Post, Comment, Admin) را در قالب کامیت‌های جدا پیاده‌سازی کنید.