

Admin Login:

Use database seeds to create the first user with email admin@admin.com and password "password".

Create and Manage Posts:

Allow admin to create, edit, and delete blog posts. Implement a WYSIWYG editor for creating rich content posts. Fields: Name, Date, Author, Content, Image. Use AJAX for CRUD operations.

Comments

CRUD: Users should be able to view and add comments to the blogs. Comments should support Create, Read, Update, and Delete (CRUD) operations.

Comment Features:

- Users can only edit or delete their own comments.
- Comments should support nested comments, and each comment should display the total count of its sub-comments.
- Display the timestamp of each comment's creation.
- Blog View: Display the total count of all comments for each blog.

Validation:

Use Laravel's validation function, utilizing Request classes for form validation. 5. Resource: Use basic Laravel resource controllers with default methods – index, create, store, etc.

API

Implement API endpoints for authentication (login API) and blog listing.

Additional Notes:

Organize the codebase into clear directories and files with meaningful names. Use Eloquent relationships for defining associations between models. Include information on how to install dependencies, configure the database, and run migrations and seeders. Create a public repository on Git for collaborative development.