# Role Based Authentication

When I setup my Laravel project I created 2 roles, Admin and User. Users can read and write to the app, and admins have full crud functionality. Each user is given a user role which is stored in my database in the ‘user\_role’ table. The User Seeder acts as a template for the user. The user model will check the users role and see if they are authorized to do an action.

When a user registers an account, they are automatically assigned the regular user role. Users are given a user Id, which is linked to a role id in the role table. A user with a role id 1 will have admin permissions and a user with an Id of 2 will have only user permissions.

When a user goes to a page, the pages controller and middleware will check if the user is authorized to visit the page.

# Crud functionality

For my project, regular users will only have create and read functionality. Admins will have full crud functionality. Users will be allowed to visit a create page with a forum where they can enter data to the database. They can also view their posts all on one page, or view each post by Id.

Admins will be able to delete posts from the database or update them. They will have an update page with a forum that they can enter data to. The admins page controller will call the relevant edit, store, update and destroy functions

# Routes

If a user logs into the webapp they will be directed to the ‘user/home’ page. This is done with a route. When the user clicks on the link to go to the home page, the route in web.php gets the ‘user/home’ page, it then goes to the home controller and calls the index function. The index function then returns the users correct view.