BRAINSTER

Laravel / Form Requests, Cookies and Sessions

Brainster Web Development Academy

Table of contents

01 Exercise 01

02 Exercise 02

03 Exercise 03

04 Exercise 04

Exercise 01

Make a simple page that has a form with 5 inputs - Title, Short Description, Photo (url), Post (textarea) and Author. After the form is submitted, redirect the user to a page that shows the following info:

- The title of the post is: $title
- The short description of the post is: $desc
- The url of the photo is: $photo
- The post is: $post
- The author of the post is: $author
- The date of the post is: $date

Exercise 02

Make a login page. It should have inputs for username, password, date and email. After you send the request for logging in, in the controller create different methods for validation.

- The username should be equal or bigger than 6 characters and contain at least one number
- Password should be at least 8 characters and contain at least one number and one uppercase letter.

If the validation fails, the user should be redirected to an error page that says “Validation error! Try again!”. If the validation succeeds, the user should be redirected to a page that says: Validation is ok!

Exercise 03

Modify the previous exercise. Instead of having all the validations as separate methods, create a ValidationRequest, and validate the request in that way, adding validation for date (required and needs to be a date) and email (required and it needs to be a valid email address).

If there’s an error, the error should appear on top of the login form, and the form should remember the old values the inputs had.

If the validation was successful, save the username in session and redirect the user to a new page, home, that has a header that says “Hi $username!”.

Exercise 04

Modify the exercise again.
Add three links on the top of the page: **Home** that brings you to the home page(the one we just created in the previous exercise), **Comment** and **Log out**.

If the user is logged in and tries to visit the login page, he should be redirected to home.
If the user is not logged in and tries to visit the homepage or the comment page, an error should appear saying “You need to be logged in to get to this page”.

The comment page is another form that has 3 inputs: title, description and image url.
When the user submits the comment, it should be validated.
The title is required and it can have a maximum of 40 characters. The description is required and can be a maximum of 100 characters. The url is required and has to be a valid url.

If the validation fails, the errors should appear on top of the form, and the old values should remain in the inputs. If the validation is successful, save the values in session and redirect the user to the homepage, where underneath the username and the links, there should be a header that reads “Your latest comment:” and display their last comment (title, then the picture, not just the url, and the description). The log out link should delete everything from the session, and redirect the user to the login page.