Implementing form validation using CodeIgniter's validation library

Topics : Codeigniter
Written on March 01, 2024

Implementing form validation using CodeIgniter's validation library allows you to ensure that the data submitted through your forms meets certain criteria before processing it further. This helps maintain data integrity and security. Here's how you can implement form validation using CodeIgniter's validation library:

1. Load the Form Validation Library:

- The Form Validation Library is loaded automatically when you load the `form` helper. Ensure that the helper is loaded in your controller or autoloaded.

```
$this->load->helper('form');
```

2. Set Validation Rules:

- Define validation rules for each form field using the `set_rules()` method. You can do this in the controller method handling the form submission.

```
$this->form_validation->set_rules('username', 'Username',
'required|min_length[5]|max_length[12]');
$this->form_validation->set_rules('email', 'Email', 'required|valid_email');
```

3. Run Form Validation:

- Use the `run()` method to run the validation rules. This method returns a boolean value indicating whether the validation was successful or not.

```
if ($this->form_validation->run() == FALSE) {
    // Form validation failed, reload the form view
    $this->load->view('form_view');
} else {
    // Form validation passed, process the form data
    // Handle form submission
}
```

4. Display Validation Errors:

- If form validation fails, you can display validation errors next to the form fields using the `validation_errors()` function.

```
echo validation_errors();
```

5. Best Practices:

- Define clear and specific validation rules for each form field to ensure data integrity and security.
- Sanitize user input before displaying or processing it to prevent security risks like XSS attacks.
- Use server-side validation as the primary line of defense, but also consider implementing client-side validation for a better user experience.

Example Controller Method:

Here's an example of how the controller method handling the form submission might look:

```
public function handle_submission() {
    // Load form validation library
    $this->load->library('form_validation');

    // Set validation rules
    $this->form_validation->set_rules('username', 'Username',
'required|min_length[5]|max_length[12]');
    $this->form_validation->set_rules('email', 'Email',
'required|valid_email');

    if ($this->form_validation->run() == FALSE) {
        // Form validation failed, reload the form view
        $this->load->view('form_view');
    } else {
        // Form validation passed, process the form data
        $username = $this->input->post('username');
        $email = $this->input->post('email');

        // Perform actions with form data (e.g., save to database, send
email)

        // Redirect or show success message
    }
}
```