CodeIgniter - Working with HTML, CSS, and JavaScript in Views

Topics : Codeigniter
Written on February 29, 2024

Working with HTML, CSS, and JavaScript in views in CodeIgniter is similar to working with them in any other web development environment. Views in CodeIgniter are typically HTML files with embedded PHP code that allow you to dynamically generate content. Here's how you can work with HTML, CSS, and JavaScript in views:

1. HTML Markup:

- Use HTML tags to structure the content of your views.
- CodeIgniter views support standard HTML elements such as `<div>`, `<p>`, `<h1>`, `<a>`, etc.
- You can use HTML attributes like `class`, `id`, `href`, `src`, etc., to add styling and behavior to elements.

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>My CodeIgniter App</title>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/style.css'); ?>">
</head>
<body>
    <div class="container">
        <h1>Welcome to My CodeIgniter App</h1>
        <p>This is a paragraph of text.</p>
        <a href="<?php echo site_url('page/about'); ?>">About</a>
    </div>
    <script src="<?php echo base_url('assets/js/script.js'); ?>"></script>
</body>
</html>
```

2. CSS Styling:

- Use CSS to style the HTML elements in your views.
- You can include CSS stylesheets using the `<link>` tag within the `<head>` section of your view.
- Alternatively, you can use inline CSS or style tags within the view itself.

Example:

```
<head>
    <style>
        .container {
            width: 80%;
            margin: 0 auto;
            padding: 20px;
            background-color: #f0f0f0;
            border: 1px solid #ccc;
        }
        h1 {
            color: blue;
        }
    </style>
</head>
```

3. JavaScript Behavior:

- Use JavaScript to add interactivity and dynamic behavior to your views.
- Include JavaScript files using the `<script>` tag at the bottom of your view or within the `<head>` section.
- You can also use inline JavaScript directly within the HTML markup.

Example:

```
<body>
    <button onclick="alert('Hello!')">Click Me</button>
    <script>
        // Inline JavaScript
        document.getElementById('myButton').addEventListener('click',
function() {
            alert('Button clicked!');
        });
    </script>
</body>
```

4. Serving Static Assets:

- Place static assets such as CSS files, JavaScript files, images, etc., in the `public` or `assets` directory of your CodeIgniter project.
- Use CodeIgniter's `base_url()` and `site_url()` functions to generate the correct URLs for linking to static assets.

5. Using CodeIgniter Helpers and Libraries:

- CodeIgniter provides helpers and libraries to assist with common tasks such as form input generation, URL creation, and data formatting.
- You can use these helpers and libraries within your views by loading them in your controller and passing them to the view data.