

MVC PROGRAMMING

COUNTINUOUS ASSESSEMENT - 1

Name : Sushmita Jaiswal

Roll No: 17

Registration : 12222771

Section : K22GW

Date : 07/04/25

Que No. 17: Passing Data to Views Scenario:

You are building a blog application and need to pass data to views. You need to pass a list of blog posts to the view, along with additional details like the author's name. Instructions: 1. Create a route that will pass an array of blog posts to a view. 2. Create a method to define an array of blog posts and pass this data to the view using the with() method or compact() function. 3. Create a Blade view (resources/views/posts/index.blade.php) to display the list of posts along with the author's name.

Expected Output: • The page should display a list of blog posts with the following information:

- o Title: Post Title
- o Author: Author Name
- o Content: Short description of the post

Solution:

Step 1: Create a Route in web.php

File: routes/web.php

We define a new route /posts and link it to a controller method.

//CODE

```
<?php
```

```
use App\Http\Controllers\PostController;
```

```
Route::get('/posts', [PostController::class, 'index']);
```

Step 2: Create a Controller

File: app/Http/Controllers/PostController.php

//CODE

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class PostController extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        $posts = [
```

```
            [
```

```

        'title' => 'Understanding Laravel Blade',
        'author' => 'Sushmita Jaiswal',
        'content' => 'Blade is a simple and powerful templating engine
provided with Laravel.'
    ],
    [
        'title' => 'Getting Started with PHP',
        'author' => 'John Doe',
        'content' => 'This post covers the basics of PHP for web
development.'
    ],
    [
        'title' => 'Deploying Laravel Applications',
        'author' => 'Jane Smith',
        'content' => 'Learn how to deploy your Laravel app with best
practices.'
    ]
];

return view('posts.index', compact('posts'));
}
}

```

Step 2: Create a Blade View

Create the file at: resources/views/posts/index.blade.php

//CODE

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Creative Blog Posts</title>

  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;7
00&display=swap" rel="stylesheet">

  <style>

    body {

      font-family: 'Poppins', sans-serif;

      background: #f7f9fc;

      padding: 30px;

      color: #333;

    }


    h1 {

      text-align: center;

      font-size: 36px;

      color: #1d3557;

      margin-bottom: 40px;

    }


    .container {

      display: grid;

      grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
```

```
gap: 30px;
max-width: 1200px;
margin: 0 auto;
}
```

```
.post-card {
  background: #ffffff;
  padding: 25px;
  border-radius: 15px;
  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}
```

```
.post-card:hover {
  transform: translateY(-8px);
  box-shadow: 0 15px 25px rgba(0, 0, 0, 0.15);
}
```

```
.title {
  font-size: 22px;
  color: #e63946;
  font-weight: 700;
  margin-bottom: 10px;
}
```

```
.author {
```

```
font-size: 14px;
color: #6c757d;
margin-bottom: 15px;
}
```

```
.content {
font-size: 16px;
color: #343a40;
}
```

```
.tag {
display: inline-block;
background: #a8dadc;
color: #1d3557;
font-size: 12px;
padding: 5px 10px;
border-radius: 5px;
margin-top: 15px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>  Creative Blog Posts</h1>
```

```
<div class="container">
```

```
  @foreach ($posts as $post)
```

```
<div class="post-card">
    <div class="title">Title: {{ $post['title'] }}</div>
    <div class="author">Author: {{ $post['author'] }}</div>
    <div class="content">Content: {{ $post['content']
}}</div>

    <div class="tag">#Laravel</div>
</div>

@endforeach
</div>
</body>
</html>
```

Commands:

Now at last in the terminal run the command :

1. cd proj1
2. php artisan serve

Now command will get executed and it will provide the link: <http://127.0.0.1:8000/posts>

Simply you have to click on the link and then you will be able to see the output of the executed code.

Output:



