PRADITA University

IF120203-Web Programming

07:
Web Framework Laravel

PRU/SPMI/FR-BM-18/0222

Goals - Using Laravel

PRADITA University

- To learn about Laravel, a powerful and elegant PHP framework for web application development
- To understand the key concepts of Laravel, including routing, controllers, views, and models
- To explore Laravel's built-in features such as authentication, database migration, and templating
- To build dynamic and interactive web applications efficiently using Laravel's expressive syntax and developer-friendly tools
- To apply best practices in web development by utilizing Laravel's conventions and security mechanisms

Introduction to Laravel

PRADITA University

- Laravel is a popular open-source PHP framework known for its elegant syntax and robust features.
- It follows the Model-View-Controller (MVC) architectural pattern, separating application logic from presentation.
- Laravel provides a wide range of tools and libraries for tasks like routing, database interaction, and user authentication.
- Artisan, Laravel's command-line tool, simplifies common development tasks such as creating models, controllers, and migrations.
- Laravel promotes code reusability and maintainability through features like middleware and dependency injection.

Prerequisite

PRADITA University

- Good at OOP
- Know about MVC Architecture
- HTML, CSS, Javascript for Frontend
- PHP

Using MVC make Controller

PRADITA University

- Install Composer for using Laravel
- composer create-project laravel/laravel project-name
- php artisan make:controller ConverterController –resource
- –resource is to have a basic CRUD
- learn more https://laravel.com/docs/10.x

Handling Views in Laravel

PRADITA University

- All views are recommended to use the format `name`.blade.php.
- This convention is followed because Laravel uses the Blade template engine.
- Blade provides features like template inheritance, control structures, and more to simplify view rendering.
- With Blade, you can create dynamic and reusable components for your application's frontend.
- Blade templates are compiled into regular PHP code for better performance.
- Blade's syntax is intuitive and expressive, making it easy to work with complex views.

in index function

PRADITA University

```
public function index()
    {
        $data['title'] = 'Currency Converter';
        return view('converter', $data);
    }
```

JQuery Convert Button - Part 1

PRADITA University

```
$("#convert").click(function (e) {
    e.preventDefault(); // Prevent the form from submitting
    const fromSelect = $("#from");
    const toSelect = $("#to");
    const amount = $("#amount").val();
    const fromCurrency = fromSelect.val();
    const toCurrency = toSelect.val();
```

JQuery Convert Button - Part 2

PRADITA University

```
$.ajax({
        url: "converter.php",
        type: "POST",
        data: {
          amount: amount,
          from: fromCurrency,
          to: toCurrency,
        },
```

JQuery Convert Button - Part 3

PRADITA University

```
success: function (result) {
        const rate = result.rate;
        const convertedAmount = (rate * amount).
           toLocaleString(undefined, {
          minimumFractionDigits: 2,
          maximumFractionDigits: 2,
        });
```

JQuery Convert Button - Part 4

PRADITA University

```
let fromCurrencySymbol;
        if (fromCurrency === "IDR") {
          fromCurrencySymbol = "Rp. ";
        } else if (fromCurrency === "USD") {
          fromCurrencySymbol = "$ ";
        } else if (fromCurrency === "EUR") {
          fromCurrencySymbol = €" ";
        } else if (fromCurrency === "GBP") {
          fromCurrencySymbol = "£ ";
        }
```

JQuery Convert Button - Part 5

PRADITA University

```
let toCurrencySymbol;
        if (toCurrency === "IDR") {
          toCurrencySymbol = "Rp. ";
        } else if (toCurrency === "USD") {
          toCurrencySymbol = "$ ";
        } else if (toCurrency === "EUR") {
          toCurrencySymbol = €" ";
        } else if (toCurrency === "GBP") {
          toCurrencySymbol = "£ ";
        }
```

JQuery Convert Button - Part 6

PRADITA University

```
$("#from-result").text(fromCurrencySymbol + amount);
        $("#from-currency").text(fromCurrency);
        $("#to-result").text(toCurrencySymbol +
            convertedAmount);
        $("#to-currency").text(toCurrency);
        $("#exchange-rate").text('1 ${fromCurrency} = ${rate
            } ${toCurrency}');
    },
```

JQuery Convert Button - Part 7

PRADITA University

```
error: function (error) {
          console.error(error);
        },
      });
    });
});
```

Change HTML to PHP

PRADITA University

- change converter.html to converter.php
- adding PHP script on top of the file

Adding PHP Script on Top - Part 1

PRADITA University

```
<?php
function convertRate($fromCurrency, $toCurrency)
{
  $apiKey = "
     fca_live_iq1WvbaON67X9adHTaJiqEszDhP6jtDz7IouUbWv";
  $url = "https://api.freecurrencyapi.com/v1/latest?apikey=
     $apiKey&currencies=$toCurrency&base_currency=
     $fromCurrency";

  $ch = curl_init($url);
```

Adding PHP Script on Top - Part 2

PRADITA University

```
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    $data = json_decode($response, true);
    return $data['data'][$toCurrency];
}
```

Adding PHP Script on Top - Part 3

PRADITA University

```php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $amount = $_POST['amount'];
  $fromCurrency = $_POST['from'];
  $toCurrency = $_POST['to'];
  $rate = convertRate($fromCurrency, $toCurrency);
```

Adding PHP Script on Top - Part 4

PRADITA University

```
$response = array(
    'rate' => $rate
  );

  header('Content-Type: application/json');
  echo json_encode($response);
  exit;
}
?>
```

Thank You