Сога Д.С.

студентка фізико-математичного факультету

Науковий керівник: А.Л. Федорчук

кандидат педагогічних наук,

доцент кафедри прикладної математики та інформатики

Житомирський державний університет імені Івана Франка

MVC PHP – ПОНЯТТЯ, ПЕРЕВАГИ, ПРИКЛАДИ

Близько десяти років тому стало популярним використовувати PHP фреймворки у розробці веб-додатків, для цього є ряд причин. Хороші фреймворки містять в собі бібліотеки, плагіни, модулі та розширення. Це дуже важливо для того, щоб реалізувати весь функціонал додатку,

вдосконалити та пришвидшити процес розробки. Фреймворки є базовою платформою для розробки та забезпечують основну структуру додатку. Тому основною метою статті є розглянути концепцію Модель-Вигляд-Контролер, її призначення та приклад використання з PHP.

Зараз майже всі PHP проекти побудовані на архітектурі Model View Controller(MVC). MVC – це архітектурний шаблон проектування, який використовується в більшості мов програмування і дозволяє виділити логіку проекту від користувацького інтерфейсу, а також виділити область логіки, що забезпечує обмін інформацією між базою даних і користувацьким інтерфейсом. Таким чином можна змінити логіку додатку, не торкаючись інтерфейсної частини, або навпаки, що дуже добре для дизайнерів та верстальників. Це дозволяє уникнути плутанини і спрощує весь процес розробки [1].

MVC (Model-view-controller) – це шаблон проектування додатків, при якому керуюча логіка поділена на три окремих компоненти таким чином, що модифікування одного з них дає мінімальний вплив на інші.

Шаблон MVC добре застосовувати при створенні складних проектів, де необхідно відокремити роботу php-програміста. MVC розділяє уявлення, дані, і обробку дій користувача на три окремих компоненти:

- MVC Модель (Model). Модель надає дані (зазвичай для View), а також реагує на запити (зазвичай від контролера), змінюючи свій стан.
- MVC Вигляд (View). Відповідає за відображення інформації (призначений для користувача інтерфейс).
- MVC Контролер (Controller). Інтерпретує дані, введені користувачем, і інформує модель і уявлення про необхідність відповідної реакції [2].

Такі компоненти як вигляд і контролер залежать від моделі, але ніяк не впливають на неї. Модель може мати декілька видів подання. Може

бути, концепція MVC складна для розуміння, але якщо її осмислити, вона стане незамінною при розробці додатків на PHP.

Особливістю при використанні MVC в PHP, є те, що існує одна точка входу в php-додаток, яка, наприклад, досягається наступним чином. Створюється index.php через який будуть оброблятися всі запити, для цього створюємо в папці з індексом файл .htaccess і поміщаємо в нього такий код:

```
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?route=$1 [L,QSA]
```

У наданому коді, першим рядком перевіряється існування запиту заданого файлу, і якщо його немає, то йде перенаправлення на index.php, інакше навіть запити малюнків сайту будуть перенаправлятися на index. Останній рядок коду перетворює запити виду index.php? Route = chat / index у вид index.php / chat / index. Якщо у вас немає можливості використовувати ModRewrite в своєму додатку, то вам доведеться робити переадресацію вручну.

PHP Модель. Дані про PHP моделі містяться в її атрибутах і можуть бути змінені тільки через спеціальні функції. Модель має декілька видів подання. Як правило, php модель це клас, що працює з БД, а саме виконує запис, читання, видалення. Читання інформації з БД може бути реалізовано кількома функціями. Як приклад модель статей на сайті: можна отримати конкретну статтю з БД, список останніх, популярних, якоїсь категорії – це все уявлення моделі. Для наочності нижче надано приклад php моделі [3].

```
<?php
function methodName()
{
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
if (!$link) {
```

```
die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($link);
$query_results= mysql_query('select * from searchNames order by firstname desc');

$data = array();
while ($row = mysql_fetch_objects($query_results)) {
$data[] = $row;
}
return $data;
}
?>
```

PHP контролер. PHP контролери отримують запити користувачів, які ми направляємо через index.php, і відповідно до них, коректують роботу моделі. Правильніше сказати контролюють роботу php додатку.

```
class Controller_Page extends Controller_Common {
    // Головна сторінка
   public function action_index()
   {
      $content = View::factory('/pages/show');
      $this->template->content = $content;
   }
```

PHP Вигляд. Вигляд відстежує зміну в моделі і створює або змінює інтерфейс php додатку.

```
<html>
<body>
<h1>List of Datas</h1>
```

<?php foreach ($data as $row) { ?>

<h2><?php echo $row->firstname ?></h2>

<h2><?php echo $row->lastname?></h2>

<?php } ?>

</body>

</html>

Як працює цей PHP MVC шаблон?

При зверненні користувачем за потрібною url вибирається відповідний контролер, який звертається до вигляду і моделі, і виводиться інформація. Іншими словами контролер в MVC є сполучною ланкою моделі та вигляду [4].

Рис.1. Концепція моделі MVC

Виникає питання, чи варто використовувати дану концепцію?

Відповідь на питання не однозначна. Вибір способу створення проекту на PHP залежить від об’ємності, призначення проекту, який буде реалізовуватися.

Розділена модель MVC забезпечує універсальність як частин окремо, так і системи в цілому. Ізоляція розмітки сторінки і ізоляція моделі даних від усього процесу забезпечує високу гнучкість продукту. Без MVC важко забезпечити такий рівень універсальності.

У той же час технологія MVC має свій недолік – складність розробки. На реалізацію такого проекту піде набагато більше часу, ніж на реалізацію проекту без MVC.

Технологію розділення моделі, контролера та вигляду слід застосовувати у великих проектах, а для реалізації сайтів візиток, тематичних сайтів краще відмовитися від MVC моделі, на користь менш універсального, але більш практичного програмного рішення.

Список використаних джерел:

1. MVC. [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/Model-View-Controller
2. MVC PHP. [Електронний ресурс] Режим доступу: http://sitear.ru/material/mvc-php
3. MVC для веб: проще некуда. [Електронний ресурс] Режим доступу: https://habrahabr.ru/post/181772/
4. Концепция MVC для чайников. [Електронний ресурс] Режим доступу: http://ruseller.com/lessons.php?id=666
5. Model View Controller (MVC) опыт использования, выводы. [Електронний ресурс] Режим доступу: https://habrahabr.ru/post/249263/