

Лабораторна робота №1

Тема: Створення контролерів та представлень

Мета: Навчитись створювати контролери, та відображати дані за допомогою представлень.

Загальні теоретичні відомості

Контролери є частиною архітектури MVC. Це об'єкти класів, успадкованих від `yii\base\Controller` та відповідають за обробку запитів і генерування відповідей. Зокрема, після отримання контролю від додатків, контролери проаналізують вхідні дані, передадуть їх у моделі, додадуть результати моделі у представлення, і на сам кінець згенерують вихідні відповіді.

Контролери складаються з дій, які є основними блоками, до яких може звертатись кінцевий користувач і запитувати виконання того або іншого функціоналу. В контролері може бути одна або декілька дій.

Наступний приклад показує контролер `post` з двома діями: `view` та `create`:

```
namespace app\controllers;\n\nuse Yii;\nuse app\models\Post;\nuse yii\web\Controller;\nuse yii\web\\NotFoundException;\n\nclass PostController extends Controller\n{\n    public function actionView($id)\n    {\n        $model = Post::findOne($id);\n        if ($model === null) {\n            throw new NotFoundException;\n        }\n\n        return $this->render('view', [\n            'model' => $model,\n        ]);\n    }\n}
```

```
}

public function actionCreate()
{
    $model = new Post;

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}
}
```

У дії view (визначеній методом `actionView()`) код спочатку завантажує модель, відповідно до запитуваного ідентифікатора моделі; Якщо модель успішно завантажена, то код відобразить її за допомогою представлення, під назвою `view`. В іншому випадку - буде отримане виключення.

У дії `create` (визначеній методом `actionCreate()`) подібний код. Він спочатку намагається завантажити модель за допомогою даних із запиту і зберегти модель. Якщо все пройшло успішно, то код перенаправить браузер на дію `view` із ідентифікатором щойно створеної моделі. В іншому випадку - він відобразить представлення `create`, через яке користувач зможе вказати необхідні дані.

Представлення

Представлення є частиною архітектури MVC. Це код, який відповідає за відображення даних кінцевим користувачам. У веб-додатку представлення зазвичай створені у вигляді *шаблонів представлень*, які є файлами скриптів PHP, що містять переважно HTML-код та презентаційний PHP-код. Управління ними здійснюється за допомогою компонента додатку `view`, який надає часто використовувані методи для полегшення компонування та

формування представлення. Для простоти, часто шаблони представлень або файли шаблонів представлень називаються представленнями.

Створення представлень

Як зазначено вище, представлення є простим скриптом PHP, який містить HTML- та PHP-код. Далі наведено представлення, що представляє форму входу користувача. Як можна побачити, PHP-код використовується для генерування динамічного вмісту, такого як заголовок сторінки та форми, тоді як HTML-код організовує їх у презентабельну HTML-сторінку.

```
<?php
use yii\helpers\Html;
use yii\widgets\ActiveForm;

/* @var $this yii\web\View */
/* @var $form yii\widgets\ActiveForm */
/* @var $model app\models\LoginForm */

$this->title = 'Вхід';
?>
<h1><?= Html::encode($this->title) ?></h1>

<p>Будь ласка, заповніть наступні поля для входу:</p>

<?php $form = ActiveForm::begin(); ?>
    <?= $form->field($model, 'username') ?>
    <?= $form->field($model, 'password')->passwordInput() ?>
    <?= Html::submitButton('Увійти') ?>
<?php ActiveForm::end(); ?>
```

Всередині представлення ви можете використовувати `$this`, що відноситься до компоненту представлення, який забезпечує керування цим шаблоном та його формування.

Окрім `$this`, у представленні можуть бути інші попередньо визначені змінні, такі як `$model` у прикладі вище. Ці змінні представляють дані, які були *передані* у представлення через контролери або інші об'єкти, які спричинили формування представлення.

Завдання:

1. Встановити yiiframework за допомогою composer.
2. Створити єдиний базовий контролер.
3. Створити UserController та AdminController, що наслідують базовий контролер.
4. У кожному контролері зробити дії, що відповідають сторінкам Index та про користувача.
5. Реалізувати представлення, що будуть відображати сторінки.
6. Змінити шаблон Layout таким чином, щоб загальні посилання відповідали сторінкам представлень.
7. Реалізувати передачу даних від контролера до представлення(Користувач(юзер або адмін), масив(список дій і можливостей)).