

## Лабораторна робота №3

**Тема:** Створення блогу

**Мета:** навчитись створювати та підключатись до БД, опрацьовувати дані.

### Загальні теоретичні відомості

#### Налаштування підключення до БД

Перш ніж продовжити, переконайтеся, що у вас встановлено розширення PHP PDO та драйвер PDO для вашої БД (наприклад `pdo_mysql` для MySQL). Це є основною вимогою, якщо ваш додаток використовує реляційну базу даних.

Відкрийте файл `config/db.php` і змініть параметри для відповідності вашій БД. За замовчуванням, файл містить наступне:

```
<?php

return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=НАЗВА БД',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
];
```

#### Створення Active Record

Для репрезентації та отримання даних з таблиці `country` створіть похідний від Active Record клас з іменем `Country`, і збережіть його в файл `models/Country.php`.

```
<?php

namespace app\models;

use yii\db\ActiveRecord;

class Country extends ActiveRecord
```

```
{
}
```

Клас `Country` наслідує `yii\db\ActiveRecord`. Вам не потрібно писати ніякого коду всередині нього! Всього лише за допомогою описаного вище коду, Yii самостійно вгадає відповідне ім'я таблиці з імені класу.

Використовуючи клас `Country`, ви можете легко маніпулювати даними з таблиці `country`, як показано у цих фрагментах коду:

```
use app\models\Country;

// отримати всі рядки з таблиці country і відсортувати їх по "name"
$countries = Country::find()->orderBy('name')->all();

// отримати рядок, по основному ключу "US"
$country = Country::findOne('US');

// відобразити "United States"
echo $country->name;

// змінити назву країни на "U.S.A." і зберегти в БД
$country->name = 'U.S.A.';
$country->save();
```

## Створення дії

Щоб відобразити дані про країну кінцевим користувачам, необхідно створити нову дію. Замість розміщення нової дії у контролері `site`, який ви використовували в попередніх розділах, є сенс створити новий контролер спеціально для всіх дій, пов'язаних з даними країн. Створіть новий контролер з іменем `CountryController` і дію `index`, як показано нижче:

```
<?php

namespace app\controllers;

use yii\web\Controller;
use yii\data\Pagination;
use app\models\Country;

class CountryController extends Controller
{
```

```
public function actionIndex()
{
    $query = Country::find();

    $pagination = new Pagination([
        'defaultPageSize' => 5,
        'totalCount' => $query->count(),
    ]);

    $countries = $query->orderBy('name')
        ->offset($pagination->offset)
        ->limit($pagination->limit)
        ->all();

    return $this->render('index', [
        'countries' => $countries,
        'pagination' => $pagination,
    ]);
}
```

Збережіть цей код у файл `controllers/CountryController.php`.

Дія `index` викликає метод `Country::find()`. Цей метод Active Record будує запит до бази даних і отримує всі дані з таблиці `country`. Щоб обмежити кількість країн, які будуть отримуватись в кожному запиті, сам запит розділяється на сторінки, за допомогою об'єкта `yii\data\Pagination`. Об'єкт `Pagination` служить двом цілям:

- Встановлює директиви `offset` і `limit` для SQL-запиту так, щоб він повертав лише одну сторінку даних за один раз (не більше 5 рядків на сторінці).
- Використовується у представленнях для відображення пейджера, який складається із переліку кнопок переходу по сторінках, про що буде описано у наступному підрозділі.

Наприкінці, дія `index` формує представлення з іменем `index` та передає до нього дані по країнах із інформацією про посторінкове розділення.

### Створення представлення

---

В директорії `views` створіть спочатку під-директорію `country`. Цей каталог буде використовуватись для всіх представлень контролера `country`. В директорії `views/country`, створіть файл з іменем `index.php`, що містить наступне:

```
<?php
use yii\helpers\Html;
use yii\widgets\LinkPager;
?>
<h1>Країни</h1>
<ul>
<?php foreach ($countries as $country): ?>
    <li>
        <? = Html::encode("{ $country->name } ({ $country->code })") ?>:
        <? = $country->population ?>
    </li>
<?php endforeach; ?>
</ul>

<? = LinkPager::widget(['pagination' => $pagination]) ?>
```

Дане представлення містить дві секції для відображення даних про країни. У першій частині, відображаються передані дані про країни у вигляді неупорядкованого списку HTML. У другій частині, віджет `yii\widgets\LinkPager` з використанням інформації про нумерацію сторінок. Віджет `LinkPager` відображається у вигляді переліку кнопок. При натисканні на будь-якій з них будуть виводитись дані країн для відповідної сторінки.

### Завдання для виконання ЛР

- 1) Створити БД з таблицями `User`(Стандартний клас та реалізація інтерфейсу), `Post`(Заголовок, текстівка, категорія, хештеги), `Category`(Назва, напрям).
- 2) Реалізувати реєстрацію та авторизацію користувачів.
- 3) Організувати можливість перегляду постів, фільтрацію, пошук(за датою, за категорією).

- 4) Авторизований користувач може писати пости, обравши категорію, вказавши заголовок поста та написавши сам текст.
- 5) Додати до БД таблиці коментарів(коментар може залишати лише авторизований користувач). За допомогою пагінації відображати лише 2 останні коментаря, для решту розробити окрему сторінку.
- 6) Користувачу надати можливість перегляду всіх власних постів та вибірку усіх постів за автором. Редагування та видалення власних постів.
- 7) Реалізувати зручний графічний інтерфейс(jQuery, Bootstrap), змінити відображення адресного рядка(ЧПУ).
- 8) \* Розробити частину адміністратора. Адміністратор може редагувати пости та коментарі(як модератор) усіх користувачів. Змінювати дані користувача.
- 9) \* Реалізувати можливість оцінки посту(лайки). Передбачити, що один користувач може лише раз оцінити пост.

#### Контрольні запитання

- 1) Що таке Модель?
- 2) Яким чином виконується підключення до БД?
- 3) Як відобразити дані з БД у представлення
- 4) Які стандартні можливості Yii для роботи з користувачами?