

## Лабораторна робота №4-5

Тема: Створення MVC-додатку на PHP без використання фреймворків

Ознайомитись з теоретичними відомостями на сайті:

[https://www.w3schools.com/php/php\\_oop\\_what\\_is.asp](https://www.w3schools.com/php/php_oop_what_is.asp)

### Завдання на лабораторну роботу

Завдання 1. Створити структуру (скелет) веб-додатку з використанням моделі MVC.

В структурі має бути передбачено наявність таких сторінок: Головна (статична), форма реєстрації, Форма відправки запиту POST, Сторінка перегляду GET та POST параметрів. Передбачити єдину точку входу.

Примітка: тема сайту обирається здобувачем індивідуально і відповідає темі обраної курсової роботи.

Завдання 2. На основі створеної структури реалізувати окремі модулі:

1. Модуль обробки даних з форми – поля форми обираються самостійно, але не менше 7 полів 4 різних видів;
2. Модуль підтримки сеансу користувача зі збереженням даних у сеансі;
3. Модуль використання Cookie.

### Варіанти завдань для створення полів форми

| Варіант   | Поля форми  | Правила перевірки   |
|-----------|---|---|
| 1,7,13,19 | Прізвище – однорядкове поле<br>Ім'я – однорядкове поле<br>По-батькові – однорядкове поле<br>Вік – список зі значеннями: до 16, 16-21, 21-27, 27-35, 35-45, 45-55, більше 55<br>Про себе - багаторядкове поле введення | Перевірити чи введені П.І.Б. тільки англійськими символами, якщо ні то видати повідомлення із зазначенням, де саме використані не англійські символи.   |
| 2,8,14,20 | Логін – однорядкове поле введення<br>Пароль 1 – поле введення пароля<br>Пароль 2 – поле введення пароля<br>E-mail – однорядкове поле введення   | Перевірити чи збігаються паролі у двох полях «Пароль 1» і «Пароль 2», і якщо не збігаються, видати повідомлення про помилку.<br>Перевірити формат E-mail – має бути у форматі символи@хост.зона, де: Символи – будь-які символи, хост – набір символів (можливо з точкою), зона – набір символів без точки. |
| 3,9,15,21 | Ім'я – однорядкове поле<br>Підлога – радіо-кнопки ЧОЛ. та ЖІН. Дата народження – три однорядкові поля ДД, ММ, РРРР.   | Якщо обрано чоловічу стать та вік менше 21 року, то видати повідомлення 'Не можна зареєструватися'. Видати те саме повідомлення, якщо обрано жіночу стать і вік менше 18 років.   |

|            |  |   |
|------------|--|---|
| 4,10,16,22 | Прізвище – однорядкове поле<br>Ім'я – однорядкове поле<br>По-батькові – однорядкове поле<br>Уподобання – список з<br>можливістю вибору кількох<br>елементів одночасно<br>(значення: Навчання, Спорт,<br>Читання, Музика, Танці тощо) | Прізвище, ім'я та по батькові не повинні<br>бути порожніми, інакше вказати, які<br>поля потрібно заповнити. Якщо<br>вибрано у списку переваг більше трьох<br>елементів одночасно, то видати<br>повідомлення, що не можна вибрати<br>більше трьох.<br>По-батькові – однорядкове поле<br>Уподобання – список з можливістю<br>вибору кількох елементів одночасно<br>(значення: Навчання, Спорт,<br>Читання, Музика, Танці тощо)  |
| 5,11,17,23 | Число 1 – однорядкове поле<br>Число 2 – однорядкове поле<br>Функція – список зі значеннями<br>(sum, sub, div, mul, pow, sqrt).<br>Результат - однорядкове поле   | При натисканні на кнопку<br>перевіряється допустимість введених<br>значень для вибраної функції<br>1. sum - число 1 і число 2 будь-які<br>(додавання)<br>2. sub - віднімання - ті ж обмеження<br>3. mul – множення – ті ж<br>обмеження<br>4. pow - зведення в ступінь - ті ж<br>огр.<br>5. div - розподіл - число 2 не 0<br>6. sqrt – корінь квадратний, число 1<br>> 0, число 2 немає.<br>У разі помилки введення видати<br>повідомлення. Далі виконати операцію<br>та порівняти із введеним закінченням у<br>полі «Результат». Якщо не збіглося<br>видати повідомлення про помилку. |
| 6,12,18,24 | Логін – однорядкове поле введення<br>Пароль 1 – поле введення пароля<br>Пароль 2 – поле введення пароля<br>Про себе - багаторядкове поле<br>введення   | Якщо введене значення в полі «логін»<br>складається з кількох слів, видати<br>повідомлення про помилку. Перевірити<br>чи збігаються поля з паролями, якщо не<br>збігаються то видати помилку.<br>Перевірити що логін та пароль більше<br>4-х символів кожен. Якщо коротше 4<br>символів, то видати помилку.   |

### Методичні рекомендації до виконання роботи:

Для побудови скелету сайту з використанням моделі MVC див.відповідну лекцію.

На основі скелета сайту, необхідно реалізувати наступний набір модулів:

**Модуль обробки даних від форми** – повинна містити форму з полями для введення різнобарвної інформації, вказаної у вашому варіанті. Обробка полів має виконуватись відповідно до вимог, зазначених у варіанті завдання нижче. Форму реалізувати у відповідності до варіантів завдань.

**Модуль підтримки сеансу користувача із збереженням даних у сеансі** – на окремій сторінці користувач може встановити код кольору, яким за замовчуванням

залитиме фон сторінки. Після натискання на кнопку зберегти, сторінка перевантажується і далі цей колір завжди застосовується в рамках сеансу.

**Модуль використання Cookie** – необхідно на тій же сторінці, де і завдання кольору фону в сеансі, додати ще одну форму, яка дозволить користувачеві вказати своє ім'я та стать. Після збереження даних та перевантаження сторінки на всіх сторінках має виводитися ім'я користувача у фразі "Вітаємо Вас, пане ІМ'Я" або "Вітаємо Вас, пані Ім'я" залежно від зазначеної раніше статі.

## Паттерн MVC

Паттерн MVC включатиме такі компоненти (класи):

- **Application** - загальний клас застосунку, який матиме конструктор і метод Run() для запуску основного циклу обробки сторінки

- **Router** - клас для обробки запитуваної УРЛ адреси і параметрів. На основі цих даних роутер повинен створити екземпляр конкретного контролера, який повинен обробити запит і викликати відповідний метод дії (action)

- **Request** - клас хелпер для роботи з переданими параметрами в запиті. Фактично це обгортка для роботи з \$\_GET і \$\_POST масивами

- **Controller** - Базовий клас контролера від якого можуть бути породжені різні типи контролерів (наприклад, контролер для сторінок PageController, або контролер для Ajax обробки AjaxController, або ApiController)

- **PageController** - Конкретна реалізація базового класу контролера для сторінок сайту

- **View** - Загальний клас для View макетів

- **PageView** - Загальний клас для View макетів сторінок. Фактично має завантажувати загальний шаблон сторінки і в нього передавати змінювані блоки, інформацію для яких готує контролер. Як правило, не потрібно реалізовувати окремий клас view. Усі вищеописані класи є базовими класами створюваного фреймворку.

Так само на базі них реалізуються конкретні класи сторінок і View макетів:

- **IndexController** - Контролер головної сторінки

- **RegformController** - Контролер форми реєстрації

## Адресація сторінок (роутинг)

Як URL-адреси для створюваного сайту можна використовувати два підходи (або один на вибір)

- **Динамічні адреси** - коли явно урл адреси міститимуть єдину точку входу /index.php і їй як параметрам передають назви контролера і дії, плюс параметри. Наприклад, це може бути так:

- `index.php?route=regform/doreg` або так

- `index.php?ctrlname=regform&action=doreg`

- **Віртуальні адреси** - коли за допомогою .htaccess файлу і модуля mod\_rewrite для apache можна налаштувати правила підміни урл адрес. Тобто, наприклад, це може виглядати так:

- `/regform/doreg.html` а фактично буде викликатися `index.php?route=regform/doreg.html`

використовуючи

директиву RewriteRule в .htaccess можна перехопити всі віртуальні адреси

- RewriteRule (.\*)\$ index.php?route=\$1 [L]

### Принципи іменування класів/методів/файлів

Для прив'язки імен маршрутів до конкретних програмних елементів (класів і методів) рекомендується використовувати уніфіковану схему іменування, як зазвичай це робиться в сучасних фреймворках. Наприклад, для запиту контролерів можна застосувати таку схему іменування:

- Ім'я контролера із запиту перетворюється до слова з великої літери і додається слово Controller. Таким чином при запиті контролера regform отримуємо ім'я класу RegformController.

- Усі контролери складаються в папку controllers, а імена файлів формуються за принципом ІМ'ЯКОНТРОЛЛЕРА.php. Тобто для запитуваного regform контролера клас матиме ім'я RegformController, а файл буде такий controllers/RegformController.php

- Для всіх запитуваних дій у контролері мають бути реалізовані методи action\_ІМ'ЯДІЯПОДІЯ()

Організувати зберігання view макетів можна наступним чином:

Різні блоки для реалізації шаблону сторінки можна складати в папку views, де буде створено підкаталог під кожен сторінку (контролер) проєкту. Тобто якщо є контролер regform, то найімовірніше буде такий набір файлів:

- views/

- regform/

form.php - Це файл із кодом форми, який буде включено в загальний шаблон сторінки

done.php - Це файл із HTML кодом, який виведеться після відправлення запиту від форми в разі успішного заповнення форми

Для загального шаблону сторінки можна, наприклад, задіяти папку views/layouts, куди скласти необхідні файли.

Як сторінки сайту реалізувати той самий набір сторінок, що був реалізований у другій лабораторній роботі. Код реалізації, як правило, можна взяти з другої лабораторної роботи, привівши його до необхідного вигляду для MVC патерну.

Найімовірніше, на виході ви маєте отримати приблизно таку структуру папок, якщо дотримуватиметеся рекомендацій, описаних вище:

/ - корень вашого сайту classes/

Application.php

Controller.php

PageController.php

Request.php

Route.php

View.php PageView.php

config/

init.php controllers/

IndexController.php RegformController.php

*views/*  
*layout/*  
*header.php*  
*footer.php*  
*или*  
*main.tpl*  
*index/*  
*main.php*    *regform/*  
*form.php*    *done.php*  
*reqview/*  
*showrequest.php*  
*index.php*  
*.htaccess*

**Завдання 5. Закомітити весь сайт та звіт на [git.ztu.edu.ua](https://git.ztu.edu.ua) та надати доступ своїм викладачам.**