Использование форм для передачи данных в программу РНР. Форма обратной связи.

Лабораторная работа № 4.

ЦЕЛЬ РАБОТЫ

Закрепление базовых знаний использования *РНР*; приобретение навыков работы с формами, как с основным инструментом получения входных данных для ВЕБ‐приложений. Обзор возможностей языка *РНР* для ввода/вывода данных.

ПРОДОЛЖИТЕЛЬНОСТЬ

2 академических часа (1 занятие)

РЕЗУЛЬТАТ РАБОТЫ

Размещенный на Веб‐сервере и доступный по протоколу *http* документ с формой обратной связи. При отправке формы осуществляется переход на страницу с ответом. Со страницы есть возможность заполнить форму еще раз.

ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ К РАБОТЕ

Результат лабораторной работы – одна *html*‐страница – header.html. Вывод формы, обработка данных выполняется в файле index.php. Обработка ответа и вывод реализованы в другом файле программы -home.php. При первой загрузке на странице сайта должна отображаться форма, содержащая следующие элементы.

* Однострочные поля для ввода текста:
  + ФИО;
  + Ваш е‐майл.
* Многострочные поля для ввода текста:
  + сообщение.
* Селектор со следующими опциями:
  + Предложение
  + Жалоба.
* Флажок:
  + Даю согласие на обработку данных.
* Две радио-кнопки:
  + Реклама из интернета
  + Рассказали друзья.
  + Кнопка «отправить».

Тема обращения определяется с помощью селектора. Флажок "Даю согласие на обработку данных" изначально не отмечен. В поле "Е‐мэйл" установлен placeholder.

Все элементы формы должны быть сгруппированы так, чтобы их было удобно использовать. Все они должны быть выровнены по левому краю, быть одинакового размера (кроме флажка и кнопки). Подписи к элементам должны располагаться слева от них.

При нажатии кнопки "Отправить" загружается страница с ответом на обращение home.php. На странице выводится ответ со следующими данными:

* ФИО
* Текст обращения
* Название прикрепленного файла, если он был прикреплен;
* Кнопка «Заполнить снова»

При переходе по кнопке должна вновь загрузиться форма обратной связи. Поля "*ФИО*" и "*Email*" должны быть заполнены ранее введенными значениями. Выбрана ранее отмеченная радио-кнопка.

РЕКОМЕНДАЦИИ К СТРУКТУРЕ ПРОГРАММЫ

Представленное в лабораторной работе задание является классическим обработчиком данных формы.

Так как мы используем две страницы в данной программе, то повторяющиеся элементы кода мы должны вынести в отдельный файл и с помощью метода include подключить этот файл в начале наших страниц. Таким элементом является объявление типа документа HTML страницы, блок head c meta-данными, структурный элемент страницы – header. Выносим этот код в файл header.html

Листинг 1

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Форма обратной связи</title>

    <link rel="stylesheet" href="styles/style.css">

</head>

<body>

    <header>

        <img src = "img\Mospolytech\_logo.jpg">

    </header>

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

Подключим файл в начале страниц нашей программы.

Листинг 2

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

<?php

include 'header.html';

?>

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

Явным и единственным признаком того, что пользователем была заполнена форма и ее данные переданы в скрипт для обработки – это наличие элементов с именами тегов формы в массиве

$\_POST или $\_GET (в зависимости от метода передачи данных). Для обработки данных по заданию лабораторной работы нужно указать в action формы относительный адрес нашего файла с обработкой данных home.php.

Для повторного заполнения формы обратной связи нам нужно передать несколько значений полей со страницы ответа. Поэтому используем массив $\_GET.

Кнопка «Заполнить снова» формируется в виде ссылки, оформление которой осуществляется в CSS‐файле. Т.к. после перехода по ссылке будет дана возможность оставить еще одно обращение, то, согласно условиям лабораторной работы, поля формы с ФИО, email, радио-кнопка уже должны быть заполнены: для этого в адрес ссылки передаются соответствующие GET‐параметры. При выводе самой формы довольно просто проверить существование указанных ключей в массиве $\_GET и, если они там есть, вывести элементы массива как значения полей.

Листинг 3

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

<?php

if(isset($\_GET['name'])){

$name=$\_GET['name'];

$email=$\_GET['email'];

$source=$\_GET['source'];

}

?>

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

А также в лабораторной из формы передается большой массив данных (включая многострочный текст), то разумно использовать *POST*‐форму. Следовательно, ее данные будут размещены в массиве $\_POST. Зная имена элементов формы, можно проверить наличие в массиве элементов с соответствующими ключами. Если они присутствуют – они были переданы из формы (им просто неоткуда взяться в массиве, кроме как быть переданными из формы), а значит пользователь заполнил ее и отправил для обработки, нажав кнопку "Отправить".

В данном случае достаточно проверить наличие в массиве только одного, т.к. страница обрабатывает только одну форму. Если в обработчик могут быть переданы данные из нескольких форм, необходимо так подбирать имена их элементов, чтобы можно было однозначно определить из какой формы были отправлены данные. Часто для этого используют невидимые текстовые поля в которых записывают имя формы или какой‐либо идентификатор.

Листинг 4

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

<input type="hidden" name="user\_id" value="6">

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

Первым шагом обработки данных, переданных пользователем с формы обратной связи является проверка выбранного типа обращения, который зависит от выбранной опции в соответствующем селекторе. Исходя из анализа переданного значения селектора с помощью ряда условных операторов, пользователю на страницу выводится соответствующий ответ.

Листинг 5

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

echo '<p> Здравствуйте,'.$\_POST['name'].'</p>'; //выводим ФИО

  if ($\_POST['category'] == 'propose'){ //проверяем тип обращения

         echo '<p>Спасибо за ваше предложение:</p>';

         echo '<textarea>'.$\_POST['message'].'</textarea>';//вывод текста сообщения

  }else{

         echo '<p>Мы рассмотрим Вашу жалобу:</p>';

         echo '<textarea>'.$\_POST['message'].'</textarea>';

  }

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

Второй этап обработки заключается в поиске прикрепленного файла. Если в массиве $\_POST найдено значение по ключу элемента вложения на форме, то мы выводим сообщение с названием приложенного пользователем файла. Иначе ничего не выводим в ответ.В примере письмо отправляется обычным текстом, а не *HTML*‐кодом, поэтому необходимо заменить все теги перевода строки <br> на соответствующие символы (символы "Возврат каретки" и "Перевод строки"). После отправки выводится требуемая надпись.

Листинг 6

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

if (isset($\_POST['attachment']) & $\_POST['attachment'] != '') echo 'Вы приложили следующий файл: '.$\_POST['attachment'];

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

Для завершения основной части РНР кода также необходимо сформировать *HTML*‐код кнопки, при нажатии которой будет повторно отображена форма обратной связи.

Листинг 7

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

echo '<a class="btn" href="index.php?N='.$\_POST['name'].'&E='.$\_POST['email'].'&S='.$\_POST['source'].'">Заполнить снова</a>';

– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –

СПРАВОЧНАЯ ИНФОРМАЦИЯ

|  |  |
| --- | --- |
| Получение данных о переданных в документ GET‐ и POST‐ параметрах. | $\_GET['el\_name'] – значение элемента *GET*‐формы с именем  "*el*\_*name*". Такие параметры также можно передавать и через *URL*.  $\_POST['el\_name'] – значение элемента *POST*‐формы с именем  "*el\_name*". |
| Что такое URL-адрес | <https://developer.mozilla.org/ru/docs/Learn/Common_questions/What_is_a_URL> |
| Обзор протокола HTTP | <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview> |
| Заголовки HTTP | <https://developer.mozilla.org/ru/docs/Web/HTTP/Headers> |
| Web-технологии. Протокол HTTP | <https://www.youtube.com/watch?v=HFt7Lm7hv1E> |
| Куки HTTP | <https://developer.mozilla.org/ru/docs/Web/HTTP/Cookies> |
| HTML5-формы | <https://html5book.ru/html5-forms/> |
| HTTP Pie | <https://httpie.io/> |
| httpbin.org | <https://httpbin.org/> |
| работа с протоколами HTTP через telnet | <https://lectureskpd.readthedocs.io/kpd/_checkpoint.html> |

КОНТРОЛЬНЫЕ ВОПРОСЫ К ЛАБОРАТОРНОЙ РАБОТЕ

Для успешной защиты работы помимо соответствующего требованиям результата необходимо уверенно отвечать на нижеперечисленные и другие вопросы, а также на контрольные вопросы всех предыдущих лабораторных работ.

1. Что такое форма?
2. Какие типы форм бывают?
3. В чем отличие POST‐ и GET‐форм?
4. Какие элементы формы Вы знаете?
5. Как передает свое значение каждый элемент формы в РНР‐программу?
6. Сколько символов может максимально содержать строка в которой буферизуется HTML‐код?

**Задание для самостоятельной работы.**

Форма аутентификации, реализованная по предыдущему заданию, должна отправляться методом **POST** . После отправки данных формы страница перезагружается и в случае совпадения введенных на форме значений с установленными в листинге страницы появляется строка с надписью: «Авторизация прошла успешно!»

**В качестве ответа укажите ссылку на git-репозиторий с кодом сайта, а также ссылку на сайт, размещённый на хостинге.**