**Лабораторная работа №5: Введение и правка данных с помощью формы**

В ходе выполнения данной лабораторной работы, мы рассмотрели принципы ввода информации и модификации контента сайта с помощью форм.

**Отправка почты**

1.С начала была создана страница **email.php.** Ей было присвоено название и пояснительный текст, и форма с двумя текстовым полями: **Тема сообщения** и **Текст сообщения**, **кнопка Отправить**, а также гиперссылка для возврата на главную страницу сайта (рис.1).

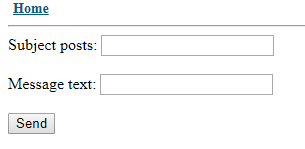


Рисунок 1 – страница отправки сообщения

2.Самостоятельно реализованная обработка данных формы с помощью функции **mail().**

3. Проверена корректность работы, создана гиперссылка с главной страницы на страницу **email.php** и с нее на страницу **blog.php.**

4.Реализована проверка заполнения всех полей формы для того, что бы исключить отправку “пустого” письма.

**Страница для добавления заметок**

1.В это упражнение демонстрируется создание страницы для добавления новых заметок – **newnote.php.**

2. Создана html-форма с именем **«newnote»**, методом обработки данных **POST.**

3. На форме мы разместили два поля:одно (типа text) для добавления заголовка заметки – с именем «**title**», другое (textarea) для добавления самой заметки – с именем «**article**» и добавлен параметр размера элементов формы.

4.На поле поместили кнопку отправки с именем «**submit**».

5. Добавили дату создания заметки. В таблице **notes,** заполняется через создаваемую нами форму. Разместили на форме после второго текстового поля скрытое поле с именем «**created**». Значение поля **created** будет получать через php-функцию **date().**

6. Обработана html-форма. Создан php-скрипт , который выполнит два шага: 1. Получает данные, введенные пользователем в поля созданной html- формы (т.е. новую заметку);

2.Передает эти данные в базу, где хранятся уже созданные ранее заметки.

Подключаем к серверу, выбираем базу, получаем данные из полей формы. Данные мы получаем из элементов формы используя названия (атрибут **name**) этих элементов. Данные формы помещаем в массив $\_POST, а затем присваиваем переменным php.

Передаем данные в базу по принципу: формирование SQL-запроса – реализация SQL-запроса. Формирование запроса: (в нем поле id получает свое значение автоматически)

7. Добавлены гиперссылки между страницами **blog.php** и **newnote.php.**



Рисунок 2 – страница добавления новых заметок

**Страница для редактирования заметок**

Создаем страницу **editnote.php**, добавляем название и пояснительный текст. Переход на эту страницу будет осуществляться со страницы **comments.php** (т.к. в начале этой страницы выводится текст комментируемой заметки).

1. Открываем страницу **comments.php**. Создаем между текстом комментируемой заметки и повторяющейся областью комментариев пустой абзац и введите текст «Изменить заметку». Делаем ее гиперссылкой для перехода на страницу **editnote.php.**

2. Гиперссылка на **editnote.php**. Для передачи информации на страницу **editnote.php** о том, какая именно заметка модифицируется (заметка с каким **id**), необходимо передать идентификатор заметки со страницы **comments.php** в строке URL-адреса через гиперссылку.

3. При его получении на странице **editnote.php** используем метод GET (принцип работы аналогичен тому, что был использован при передаче идентификатора заметки со страницы **blog.php** на страницу **comments.php**)

4. На странице **editnote.php** создаем html-форму с именем «**editnote**», метод обработки данных – POST. На форме размещаем два поля: одно (типа text) для изменения заголовка заметки – с именем «**title**», другое (textarea) для изменения самой заметки – с именем «**article**». Добавьте параметры размера элементов формы. Помещаем на поле кнопку отправки с именем «**submit**».

Создаем php-скрипт для обработки данных формы:

1. Отображаем редактируемую заметку в полях формы (т.е. помещать данные из базы в поля формы);

2. Получаем измененные данные из формы;

3. Передаем изменение данные в таблицу.

5. Заполняем форму. Вводим переменную **$note\_id**, которая получит в качестве значения идентификатор обрабатываемой заметки. Это значение она должна получить через массив $\_GET. Добавляем соединение с сервером. Выбираем базу данных. Далее формируем запрос на получение заметки с выбранным **id** из базы данных, для размещения ее в полях формы. Запрос реализуется с помощью оператора SELECT, условием запроса должно быть **id** выбранной заметки.

6.Отображаем заметки в соответствующих полях. Добавляем на html-форму скрытое поле с именем **note** (оно будет содержать **id** заметки).

7. Получаем данных из формы после изменения. Принцип реализации похож на добавление новой заметки. Получаем из формы измененные данные с помощью метода $\_POST; Передаем данные в таблицу с помощью SQL-запроса. Разница заключается только в SQL-запросе - при добавлении используем INSERT, а при обновлении UPDATE.

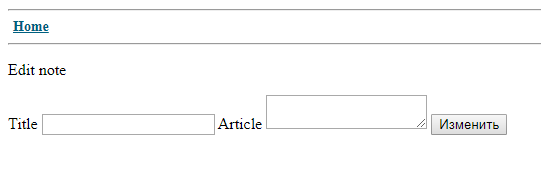


Рисунок 3 – Изменение заметки

**Создаем страницу удаления заметок**

Аналогично как с изменением заметки создаем новый файл. По выбранному id заметки удаляется заметка через sql-запрос.

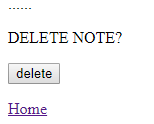


Рисунок 4 – страничка удаления заметки