

Министерство образования и науки
федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Отчет по дисциплине: «Web-программирование»

Практическая работа 3

Выполнила: Полтавец
Елена Андреевна

Группа: К3322

Проверила: Марченко
Елена Вадимовна

Санкт-Петербург

2024

Цель: отработать навыки разработки веб-страниц, применив знания PHP, MySQL и основ веб-программирования.

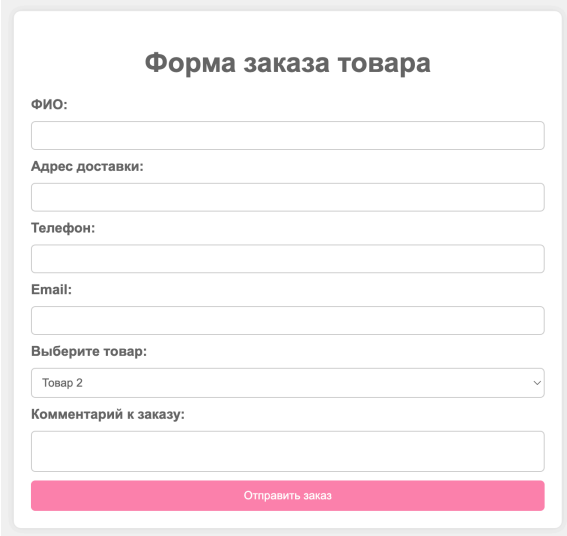
Задачи:

- Разработать веб-страницу с занесением данных в БД MySQL с помощью скрипта PHP;
- В движке WordPress модифицировать PHP скрипт согласно условиям задания;
- Написать веб-сервер с возможностью указания порта, на котором сервер будет работать.

Ход работы

Задание 1

В рамках задания было необходимо разработать веб-страницу (Рисунок 1), на которой пользователь может оставить данные о себе – фамилия, имя, отчество, адрес для физической доставки товара, телефон, адрес электронной почты. Товары покупатель выбирает из списка (реализовать меню), была предусмотрена отдельная позиция, в которой пользователь может оставить комментарий по заказу.



Форма заказа товара

ФИО:

Адрес доставки:

Телефон:

Email:

Выберите товар:

Комментарий к заказу:

Рисунок 1 – Страница с формой заказа товара

Был установлен MAMP (Macintosh, Apache, MySQL, PHP) – это программное обеспечение, которое создано для установки серверной среды на компьютерах Mac, позволяющее локально разрабатывать и тестировать веб-приложения, используя комбинацию Apache в качестве веб-сервера, MySQL в качестве системы управления базами данных и PHP в качестве языка программирования.

Данные, введенные в форму, обрабатываются PHP скриптом и заносятся в таблицу БД MySQL Server. Разработана и создана структура таблицы в PhpMyAdmin (Рисунок 2).

	#	Имя	Тип	Сравнение	Атрибуты
<input type="checkbox"/>	1	id	int		
<input type="checkbox"/>	2	fullname	varchar(100)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	3	address	varchar(200)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	4	phone	varchar(20)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	5	email	varchar(100)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	6	product	varchar(30)	utf8mb4_0900_ai_ci	
<input type="checkbox"/>	7	comments	text	utf8mb4_0900_ai_ci	

Рисунок 2 – Структура таблицы

В качестве обработчика POST-запроса при отправке формы используется PHP-скрипт, который добавляет данные из формы (Рисунок 3).

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $conn = new mysqli("localhost", "main_user", "1234", "lab_3_1");

    if($conn->connect_error){
        die("Ошибка: " . $conn->connect_error);
    }

    $fullname = $_POST["fullname"];
    $address = $_POST["address"];
    $phone = $_POST["phone"];
    $email = $_POST["email"];
    $product = $_POST["product"];
    $comments = $_POST["comments"];

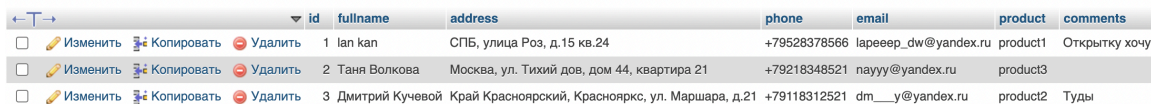
    $sql = "INSERT INTO persons (fullname, address, phone, email, product, comments) VALUES ('$fullname'";

    if($conn->query($sql)){
        header("Location: success.php");
    } else{
        header("Location: error.php");
    }

    $conn->close();
}
```

Рисунок 3 – PHP-скрипт

Если все прошло успешно, то пользователь видит сообщение на новой странице, а данные появляются в базе данных (Рисунок 4).



	id	fullname	address	phone	email	product	comments
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	1	Ian kan	СПб, улица Роз, д.15 кв.24	+79528378566	lapereep_dw@yandex.ru	product1	Открытку хочу
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	2	Таня Волкова	Москва, ул. Тихий дов, дом 44, квартира 21	+79218348521	nayyy@yandex.ru	product3	
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	3	Дмитрий Кучевой	Край Красноярский, Красноярск, ул. Маршара, д.21	+79118312521	dm___y@yandex.ru	product2	Туды

Рисунок 4 – База данных с данными из формы

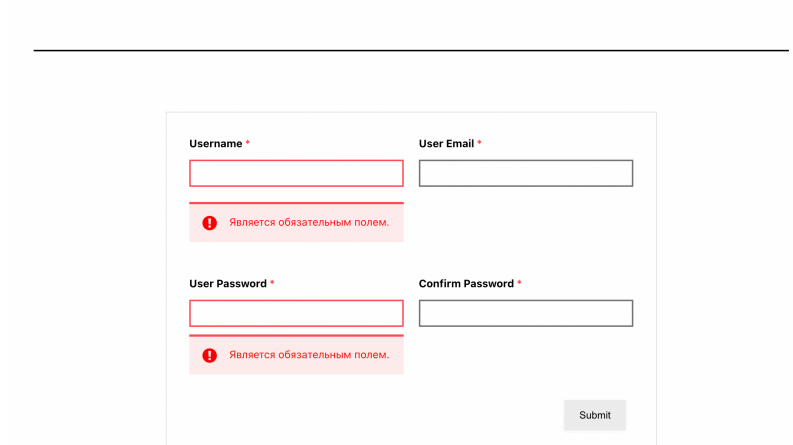
Задание 2

В WordPress используется форма для авторизации пользователей – пользователь вводит логин и пароль. Логин и хэш от пароля записываются в таблицу БД MySQL. Было необходимо модифицировать PHP скрипт, чтобы дополнительно логин и пароль записывались в отдельную новую таблицу в БД MySQL. Было необходимо предусмотреть два варианта записи:

- а) пароль записывается в исходном виде;
- б) инвертируются биты бат пароля (ноль меняется на единицу, единица меняется на ноль).

В начале была выбрана форма регистрации (Рисунок 5).

Страница регистрации



Username *

Является обязательным полем.

User Email *

Является обязательным полем.

User Password *

Является обязательным полем.

Confirm Password *

Является обязательным полем.

Submit

Рисунок 5 – Форма регистрации

Все данные о новых зарегистрированных пользователях по умолчанию сохраняются в таблицу wp_users в PhpMyAdmin. Для того, чтобы сохранить

модифицированные пароли была скопирована структура таблицы wp_users в таблицы wp_users_1 и wp_users_2 (Рисунок 6).

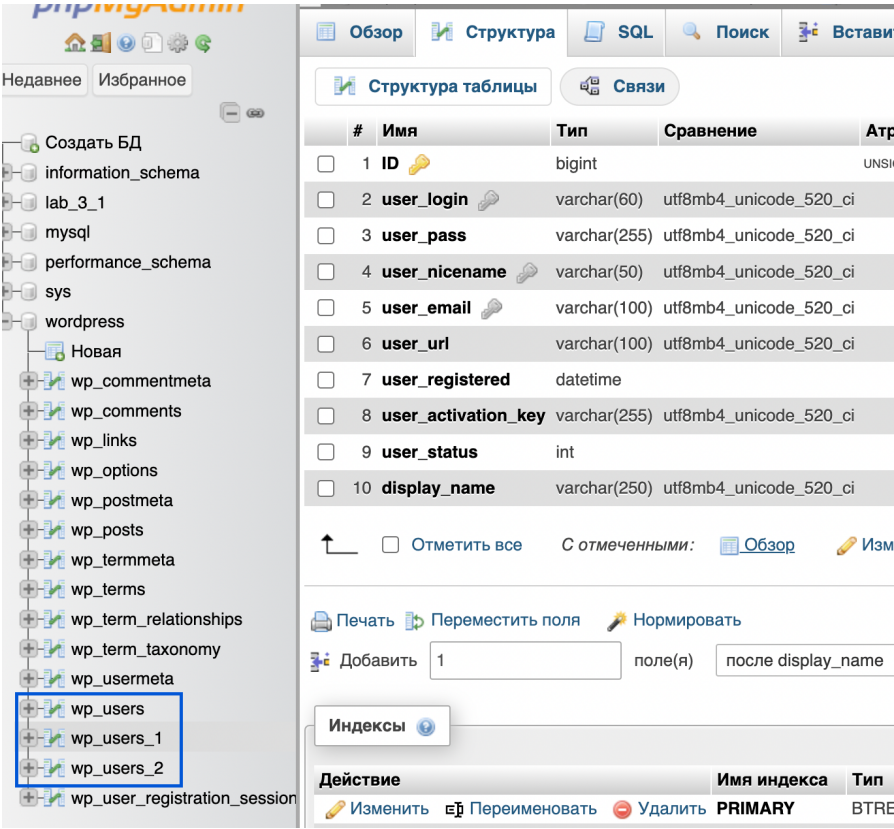


Рисунок 5 – Созданные таблицы в БД

WordPress использует процесс хэширования введенного пароля с помощью различных хэширующих алгоритмов. По умолчанию, WordPress использует MD5. Поэтому, чтобы сохранить пароль в исходном и инвертированном виде, необходимо убрать функцию хэширования.

По умолчанию в папке wp-includes содержится файл user.php, который содержит механизм отправки данных после регистрации пользователя. Измененный код представлен на рисунках 6–7.

```

$user_pass = $userdata['user_pass']; # пароль без хэширования
$compact1 = compact('user_pass', 'user_nicename', 'user_email', 'user_url', 'user_registered', 'user_activation_key', 'display_name' );
$user_pass = ~bindec(decbin(crc32($userdata['user_pass']))); # инвертированный пароль
$compact2 = compact('user_pass', 'user_nicename', 'user_email', 'user_url', 'user_registered', 'user_activation_key', 'display_name' );
$data = wp_unslash( $compact1 );
$data_1 = wp_unslash( $compact1 );
$data_2 = wp_unslash( $compact2 );

if ( ! $update ) {
    $data = $data + compact( 'user_login' );
    $data_1 = $data_1 + compact( 'user_login' );
    $data_2 = $data_2 + compact( 'user_login' );
}

if ( is_multisite() ) {
    $data = $data + compact( 'spam' );
    $data_1 = $data_1 + compact( 'spam' );
    $data_2 = $data_2 + compact( 'spam' );
}

$data = apply_filters( 'wp_pre_insert_user_data', $data, $update, ( $update ? $user_id : null ), $userdata );
$data_1 = apply_filters( 'wp_pre_insert_user_data', $data_1, $update, ( $update ? $user_id : null ), $userdata );
$data_2 = apply_filters( 'wp_pre_insert_user_data', $data_2, $update, ( $update ? $user_id : null ), $userdata );

if ( empty( $data ) || ! is_array( $data ) ) {
    return new WP_Error( 'empty_data', __( 'Not enough data to create this user.' ) );
}

```

Рисунок 6 – Работа с паролем

```

if ( empty( $data ) || ! is_array( $data ) ) {
    return new WP_Error( 'empty_data', __( 'Not enough data to create this user.' ) );
}

if ( $update ) {
    if ( $user_email !== $old_user_data->user_email || $user_pass !== $old_user_data->user_pass ) {
        $data['user_activation_key'] = '';
    }
    $wpdb->update( $wpdb->users, $data, array( 'ID' => $user_id ) );
} else {
    $wpdb->insert( $wpdb->users, $data );
    $user_id = (int) $wpdb->insert_id;
    $wpdb->insert( $wpdb->prefix.'users_1', $data_1 ); # дополнительная выгрузка
    $user_id = (int) $wpdb->insert_id;
    $wpdb->insert( $wpdb->prefix.'users_2', $data_2 ); # дополнительная выгрузка
    $user_id = (int) $wpdb->insert_id;

    $user = new WP_User( $user_id );

    /**
     * Filters a user's meta values and keys immediately after the user is created or updated
     * and before any user meta is inserted or updated.
     */
}

```

Рисунок 7 – Выгрузка данных в таблицы

Был протестирован функционал: зарегистрировано несколько пользователей, затем осуществлено сравнение занесенных в БД данных (Рисунки 8–10).

	ID	user_login	user_pass	user_nicename	user_email	us
<input type="checkbox"/> Изменить Копировать Удалить	1	admin	\$P\$BsbShBgrw0ltcBrw7QtL49FJnRMsB/	admin	Malenago.kay@yandex.ru	htt
<input type="checkbox"/> Изменить Копировать Удалить	2	name_1	\$P\$BvgM04tFog70hASALg8xerW8v7BT4.	name_1	name_1_pochta@yandex.ru	
<input type="checkbox"/> Изменить Копировать Удалить	3	name_2	\$P\$B5uhH9ztr8tEikvHB3IgPW8lW0u4Oj0	name_2	name_2_pochta@yandex.ru	
<input type="checkbox"/> Изменить Копировать Удалить	4	name_3	\$P\$BDQnvS7AJ\$6feg6wNOtuOvc9ux5sZ1	name_3	name_3_pochta@yandex.ru	

Рисунок 8 – Таблица wp_users с хэшированными паролями

			ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered
<input type="checkbox"/>	Изменить	Копировать	Удалить	1	name_1	1234	name_1	name_1_pochta@yandex.ru	2024-09-28 12:3
<input type="checkbox"/>	Изменить	Копировать	Удалить	2	name_2	2345qwerty	name_2	name_2_pochta@yandex.ru	2024-09-28 12:3
<input type="checkbox"/>	Изменить	Копировать	Удалить	3	name_3	sdf	name_3	name_3_pochta@yandex.ru	2024-09-28 12:3

Рисунок 9 – Таблица wp_users_1 с исходными паролями

				ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered
<input type="checkbox"/>	Изменить	Копировать	Удалить	1	name_1	-2615402660	name_1	name_1_pochta@yandex.ru		2024-09-28 12:3
<input type="checkbox"/>	Изменить	Копировать	Удалить	2	name_2	-3543157244	name_2	name_2_pochta@yandex.ru		2024-09-28 12:3
<input type="checkbox"/>	Изменить	Копировать	Удалить	3	name_3	-213280214	name_3	name_3_pochta@yandex.ru		2024-09-28 12:3

Рисунок 10 – Таблица wp_users_2 с инвертированным

Задание 3

В данном задании необходимо было написать web-сервер, предусмотреть возможность указать порт, на котором будет работать сервер.

В этом коде используются две библиотеки Python:

- *http.server* – это модуль Python, предоставляющий базовые классы для реализации HTTP серверов. В данном случае используется *SimpleHTTPRequestHandler*, который предоставляет простую реализацию HTTP обработчика запросов.
- *socketserver* – это также модуль Python, обеспечивающий базовые классы для создания сетевых серверов. В коде создается TCP сервер с помощью *TCPHandler*.

В данном коде (Рисунок 11) был создан простой HTTP сервер на порту 40. Класс *MyHandler* используется для обработки GET запросов. Если путь запроса равен '/', то он заменяется на 'index.html'. Затем запускается TCP сервер на указанном порту, и выводится сообщение о запуске сервера с указанием порта. Функция *serve_forever()* запускает сервер и обслуживает запросы, пока не будет принудительно остановлена.

```

1 import http.server
2 import socketserver
3
4 PORT = 40
5
6 class MyHandler(http.server.SimpleHTTPRequestHandler):
7     def do_GET(self):
8         if self.path == '/':
9             self.path = 'index.html'
10            return super().do_GET()
11
12 if __name__ == "__main__":
13     with socketserver.TCPServer(("", PORT), MyHandler) as httpd:
14         print(f"Сервер запущен на порту {PORT}")
15         httpd.serve_forever()

```

Рисунок 11 – Код для web-сервера

После запуска на верном порту появилась страница (Рисунок 12).

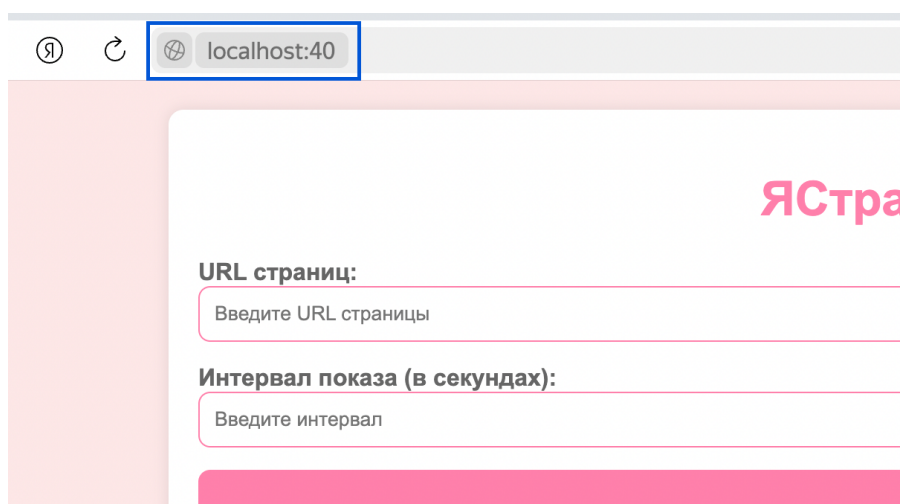


Рисунок 12 – Страница по заданному порту

Вывод: были отработаны навыки разработки веб-страниц, применены знания PHP, MySQL и основ веб-программирования на реальных задачах.