Аудит безопасности

1. Защита от XSS (Cross-Site Scripting)

Что нашли:

- В некоторых местах мы выводим данные от пользователей без защиты (например, в admin.php и form.php).
- Особенно это опасно, когда данные из базы данных показываются в таблицах и формах.

Что сделали для защиты:

- Используем функцию htmlspecialchars(), чтобы защитить все выводимые данные.
- Экранируем специальные символы перед тем, как показывать их в HTML.

Примеры исправлений:

```
<!-- Было -->
<td><?= $app['last_name'] ?></td>

<!-- Стало -->
<td><?= htmlspecialchars($app['last_name']) ?></td>
```

2. Защита от раскрытия информации

Что нашли:

- В сообщениях об ошибках подключения к базе данных показываются детали (логин, пароль).
- В admin.php есть жестко закодированные данные администратора.

Что сделали для защиты:

- Убрали чувствительную информацию из сообщений об ошибках.
- Перенесли данные администратора в отдельный конфигурационный файл.

Примеры исправлений:

```
// Было
} catch (PDOException $e) {
```

```
    die('Ошибка подключения: ' . $e->getMessage());
}

// Стало
} catch (PDOException $e) {
    die('Ошибка подключения к базе данных');
}
```

3. Защита от SQL Injection

Что нашли:

- В некоторых местах мы напрямую вставляем переменные в SQL-запросы.
- Не все параметры запросов правильно подготовлены.

Что сделали для защиты:

- Используем подготовленные выражения (prepared statements) везде.
- Применяем PDO с параметризованными запросами.

Примеры исправлений:

```
// Было
$stmt = $db->query("SELECT * FROM users WHERE login = '$login'");

// Стало
$stmt = $db->prepare("SELECT * FROM users WHERE login = ?");
$stmt->execute([$login]);
```

4. Защита от CSRF (Cross-Site Request Forgery)

Что нашли:

- В формах нет CSRF-токенов.
- Не проверяем, откуда приходят запросы.

Что сделали для защиты:

- Добавили CSRF-токены во все формы.
- Проверяем токены, когда обрабатываем POST-запросы.

Примеры исправлений:

```
// В форму добавлено:
<input type="hidden" name="csrf_token" value="<?= $_SESSION['csrf_token'] ?>">

// В обработчике:
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (!isset($_POST['csrf_token']) || $_POST['csrf_token'] !==
$_SESSION['csrf_token']) {
        die('Недействительный CSRF-токен');
    }
    // ...
}
```

5. Защита от уязвимостей с включением файлов и загрузкой

Что нашли:

- Есть риск включения произвольных файлов.
- Нет проверки загружаемых файлов (хотя функционал загрузки не реализован).

Что сделали для защиты:

- Четко указываем, какие файлы можно включать.
- Запрещаем включение файлов из переменных.

Примеры исправлений:

```
// Вместо динамического включения:
include($page . '.php');

// Используется прямое включение:
include('form.php');
```