validate_input.php

index.php

```php
<?php
// Функция полной очистки строк
function sanitize_and_validate($data, $max_length = 16) {
    // 1. Запрещаем пробелы, табы, переносы строк
    $data = preg_replace('/\s+/', '', $data);

    // 2. Обрезаем строго до 16 символов (п. 7)
    $data = mb_substr($data, 0, $max_length, 'UTF-8');

    // 3. Вырезаем любые HTML-теги, скрипты, конструкции типа <?php (п. 4, 5, 6, 8)
    $data = strip_tags($data);

    // 4. Оставляем ТОЛЬКО буквы (латиница), цифры и знаки подчеркивания/дефиса/точки
    // Это полностью уничтожает любые попытки SQL-инъекций (типа OR 1=1) и XSS-атак
    $data = preg_replace('/[^a-zA-Z0-9_\-\.]/', '', $data);

    return $data;
}

// Черные списки запрещенных слов (п. 4.1 и 4.2)
$forbidden_usernames = ['admin', 'administrator', 'test', 'root', 'user', 'guest'];
$forbidden_passwords = ['qwerty', 'abcdef', '123456', 'iamnumber1', 'password'];

// Пример проверки пришедших POST-данных (встраивается в api/start_install.php)
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $input_user = isset($_POST['username']) ? $_POST['username'] : '';
    $input_pass = isset($_POST['password']) ? $_POST['password'] : '';

    // Чистим данные
    $clean_user = sanitize_and_validate($input_user, 16);
    $clean_pass = sanitize_and_validate($input_pass, 16);

    // Проверяем по массивам запретов (вхождение подстроки)
    foreach ($forbidden_usernames as $bad_user) {
        if (strpos(strtolower($clean_user), $bad_user) !== false || empty($clean_user)) {
            echo json_encode(['error' => 'Недопустимое имя пользователя!']);
            exit;
        }
```

```
    }

    foreach ($forbidden_passwords as $bad_pass) {
        if (strpos(strtolower($clean_pass), $bad_pass) !== false ||
empty($clean_pass)) {
            echo json_encode(['error' => 'Слишком простой или запрещенный
пароль!']);
            exit;
        }
    }
}
```