МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Колледж Алтайского государственного университета

Отделение экономики и информационных технологий

ОТЧЕТ

Задание для должников

Конкурс Прожектор

Выполнил студент

4 курса К.205с9-4 группы

Васильев Денис Олегович

(подпись)

Преподаватель

Третьякова Ольга Николаевна

(подпись)

Работа защищена

«\_\_\_\_» 2025 г.

Оценка

(подпись)

Барнаул 2025

# Диаграммы

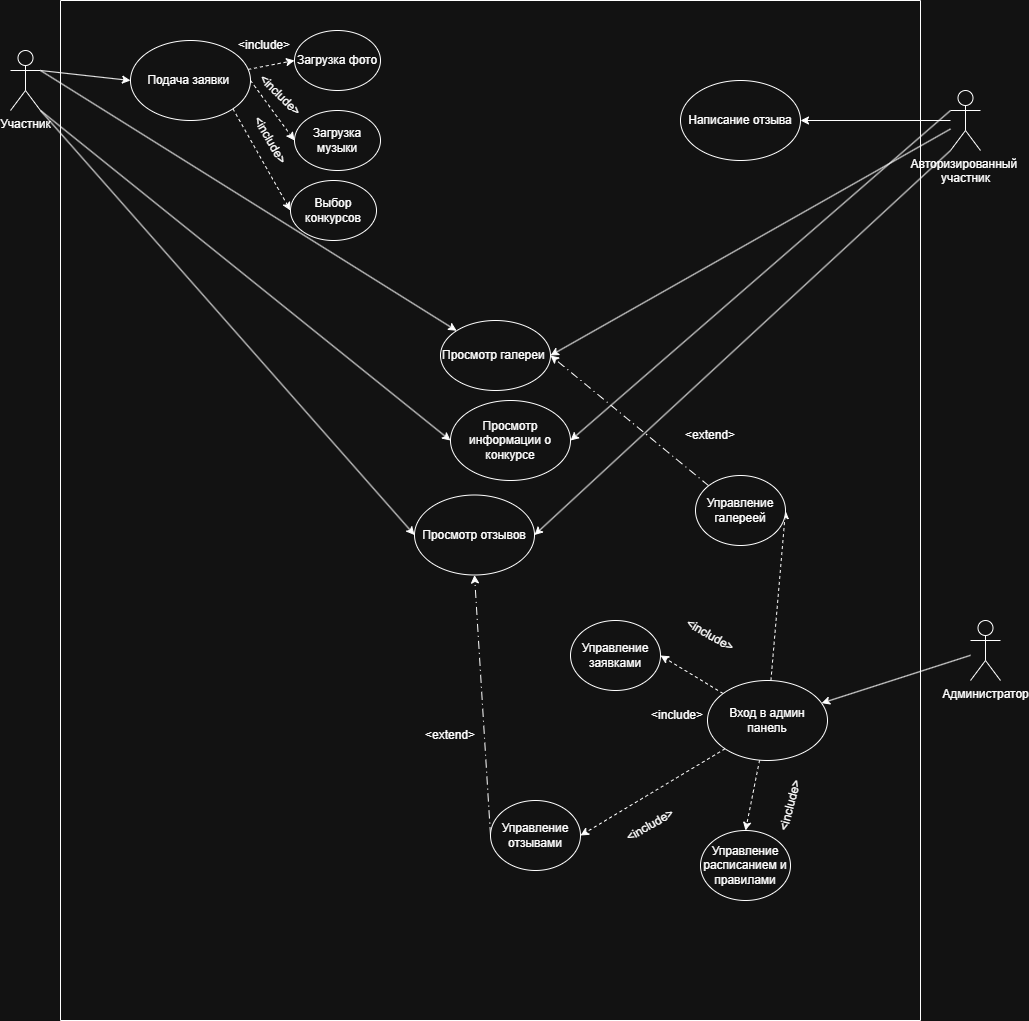


Рисунок 1 – Диаграмма вариантов использования

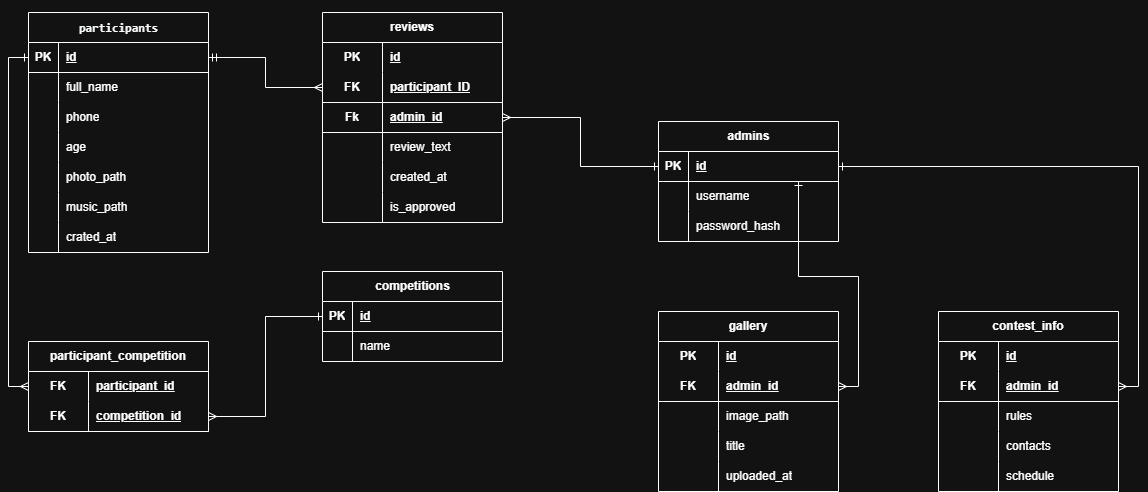


Рисунок 2 – ER диаграмма

# База данных

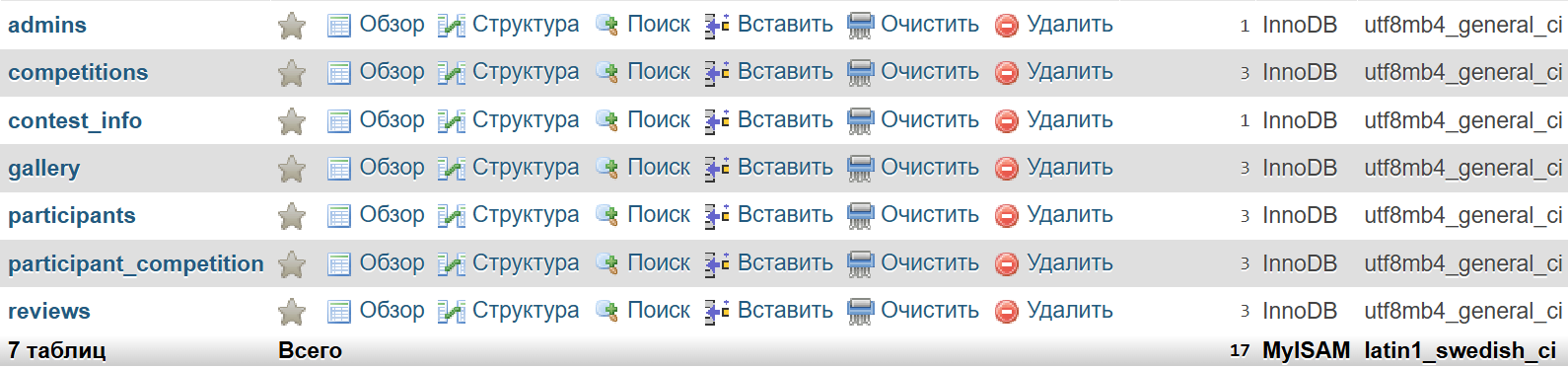


Рисунок 3 – База данных

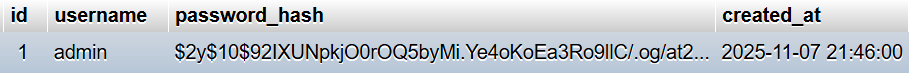


Рисунок 4 – Таблица admins

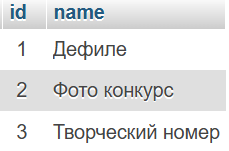


Рисунок 5 – Таблица competitions

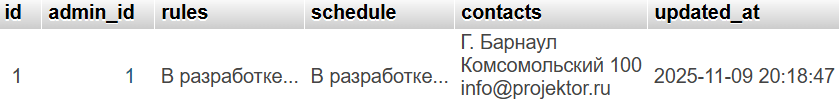


Рисунок 6 – Таблица contest\_info



Рисунок 7 – Таблица gallery

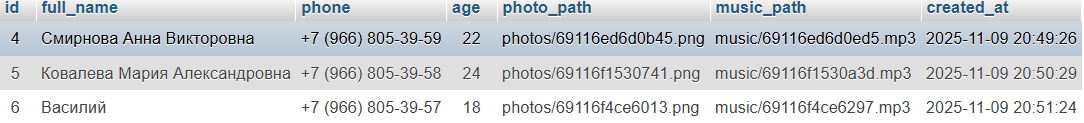


Рисунок 8 – Таблица participants

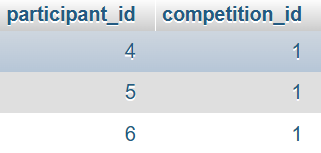


Рисунок 9 – Таблица participant­\_competition

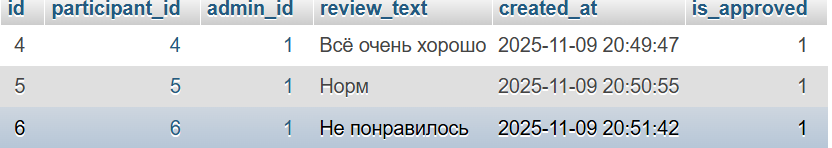


Рисунок 10 – Таблица reviews

# Код

Index.php

 // Загружаем конкурсы напрямую из БД

                                try {

                                    $stmt = $pdo->query("SELECT \* FROM competitions");

                                    $competitions = $stmt->fetchAll(PDO::FETCH\_ASSOC);

                                    if ($competitions) {

                                        foreach ($competitions as $competition) {

                                            echo '

                                            <div class="form-check mb-2">

                                                <input class="form-check-input" type="checkbox" name="competitions[]"

                                                       value="' . $competition['id'] . '" id="comp' . $competition['id'] . '">

                                                <label class="form-check-label" for="comp' . $competition['id'] . '">

                                                    ' . htmlspecialchars($competition['name']) . '

                                                </label>

                                            </div>';

                                        }

                                    } else {

                                        echo '<p class="text-muted">Конкурсы временно недоступны</p>';

                                    }

                                } catch (Exception $e) {

                                    echo '<p class="text-muted">Ошибка загрузки конкурсов</p>';

                                }

<?php

                // Загружаем галерею напрямую из БД

                try {

                    $stmt = $pdo->query("SELECT \* FROM gallery ORDER BY uploaded\_at DESC");

                    $gallery = $stmt->fetchAll(PDO::FETCH\_ASSOC);

                    if ($gallery) {

                        foreach ($gallery as $item) {

                            echo '

                            <div class="col-md-4 mb-4 fade-in-up">

                                <div class="card h-100">

                                    <img src="uploads/' . htmlspecialchars($item['image\_path']) . '" class="card-img-top"

                                         alt="' . htmlspecialchars($item['title']) . '" style="height: 250px; object-fit: cover;">

                                    <div class="card-body">

                                        <h6 class="card-title">' . htmlspecialchars($item['title']) . '</h6>

                                    </div>

                                </div>

                            </div>';

                        }

                    } else {

                        echo '<div class="col-12"><p class="text-muted">Галерея пуста</p></div>';

                    }

                } catch (Exception $e) {

                    echo '<div class="col-12"><p class="text-muted">Галерея временно недоступна</p></div>';

                }

                ?>

<?php

                // Загружаем отзывы напрямую из БД

                try {

                    $stmt = $pdo->query("

                        SELECT r.\*, p.full\_name

                        FROM reviews r

                        JOIN participants p ON r.participant\_id = p.id

                        WHERE r.is\_approved = 1

                        ORDER BY r.created\_at DESC

                    ");

                    $reviews = $stmt->fetchAll(PDO::FETCH\_ASSOC);

                    if ($reviews) {

                        foreach ($reviews as $review) {

                            // Правильно обрабатываем переносы строк в отзывах

                            $reviewText = nl2br(htmlspecialchars($review['review\_text']));

                            echo '

                            <div class="col-md-6 mb-4 fade-in-up">

                                <div class="card h-100">

                                    <div class="card-body">

                                        <h6 class="card-title">' . htmlspecialchars($review['full\_name']) . '</h6>

                                        <div class="card-text">' . $reviewText . '</div>

                                        <small class="text-muted">' . date('d.m.Y', strtotime($review['created\_at'])) . '</small>

                                    </div>

                                </div>

                            </div>';

                        }

                    } else {

                        echo '<div class="col-12"><p class="text-muted">Отзывов пока нет</p></div>';

                    }

                } catch (Exception $e) {

                    echo '<div class="col-12"><p class="text-muted">Отзывы временно недоступны</p></div>';

                }

                ?>

 <?php

                        // Загружаем контакты напрямую из БД

                        try {

                            $stmt = $pdo->query("SELECT contacts FROM contest\_info ORDER BY id DESC LIMIT 1");

                            $info = $stmt->fetch(PDO::FETCH\_ASSOC);

                            if ($info && !empty($info['contacts'])) {

                                // Правильно обрабатываем переносы строк в контактах

                                $contacts = nl2br(htmlspecialchars($info['contacts']));

                                echo '

                                <h4>Контактная информация</h4>

                                <div class="contact-text">' . $contacts . '</div>';

                            } else {

                                echo '

                                <h4>Контактная информация</h4>

                                <div class="contact-text">

                                    Телефон: +7 (999) 123-45-67<br>

                                    Email: info@projektor.ru<br>

                                    Адрес: г. Москва, ул. Примерная, д. 1

                                </div>';

                            }

                        } catch (Exception $e) {

                            echo '

                            <h4>Контактная информация</h4>

                            <div class="contact-text">

                                Телефон: +7 (999) 123-45-67<br>

                                Email: info@projektor.ru

                            </div>';

                        }

                        ?>

Actions.php

<?php

require\_once 'config.php';

header('Content-Type: application/json');

$action = $\_POST['action'] ?? '';

switch($action) {

    case 'register':

        registerParticipant();

        break;

    case 'submit\_review':

        submitReview();

        break;

    default:

        echo json\_encode(['success' => false, 'message' => 'Unknown action']);

}

function registerParticipant() {

    global $pdo;

    // Валидация

    if(empty($\_POST['full\_name']) || empty($\_POST['phone']) || empty($\_POST['age'])) {

        echo json\_encode(['success' => false, 'message' => 'Заполните все обязательные поля']);

        return;

    }

    // Загрузка фото

    $photoPath = uploadFile('photo', 'photos');

    if(!$photoPath) {

        echo json\_encode(['success' => false, 'message' => 'Ошибка загрузки фото']);

        return;

    }

    // Загрузка музыки (опционально)

    $musicPath = uploadFile('music', 'music');

    try {

        $pdo->beginTransaction();

        // Добавляем участника

        $stmt = $pdo->prepare("INSERT INTO participants (full\_name, phone, age, photo\_path, music\_path) VALUES (?, ?, ?, ?, ?)");

        $stmt->execute([$\_POST['full\_name'], $\_POST['phone'], $\_POST['age'], $photoPath, $musicPath]);

        $participantId = $pdo->lastInsertId();

        // Добавляем выбранные конкурсы

        if(isset($\_POST['competitions'])) {

            $stmt = $pdo->prepare("INSERT INTO participant\_competition (participant\_id, competition\_id) VALUES (?, ?)");

            foreach($\_POST['competitions'] as $competitionId) {

                $stmt->execute([$participantId, $competitionId]);

            }

        }

        $pdo->commit();

        echo json\_encode(['success' => true, 'message' => 'Регистрация успешна']);

    } catch(Exception $e) {

        $pdo->rollBack();

        echo json\_encode(['success' => false, 'message' => 'Ошибка базы данных: ' . $e->getMessage()]);

    }

}

function submitReview() {

    global $pdo;

    if(empty($\_POST['review\_name']) || empty($\_POST['review\_phone']) || empty($\_POST['review\_text'])) {

        echo json\_encode(['success' => false, 'message' => 'Заполните все поля']);

        return;

    }

    try {

        // Находим participant\_id по имени и телефону

        $stmt = $pdo->prepare("SELECT id FROM participants WHERE full\_name = ? AND phone = ? LIMIT 1");

        $stmt->execute([$\_POST['review\_name'], $\_POST['review\_phone']]);

        $participant = $stmt->fetch();

        if(!$participant) {

            echo json\_encode(['success' => false, 'message' => 'Участник не найден. Проверьте ФИО и телефон']);

            return;

        }

        // Добавляем отзыв

        $stmt = $pdo->prepare("INSERT INTO reviews (participant\_id, review\_text) VALUES (?, ?)");

        $stmt->execute([$participant['id'], $\_POST['review\_text']]);

        echo json\_encode(['success' => true, 'message' => 'Отзыв отправлен на модерацию']);

    } catch(Exception $e) {

        echo json\_encode(['success' => false, 'message' => 'Ошибка: ' . $e->getMessage()]);

    }

}

function uploadFile($fieldName, $folder) {

    if(!isset($\_FILES[$fieldName]) || $\_FILES[$fieldName]['error'] != UPLOAD\_ERR\_OK) {

        return null;

    }

    $uploadDir = "../uploads/$folder/";

    if(!is\_dir($uploadDir)) {

        mkdir($uploadDir, 0777, true);

    }

    $extension = pathinfo($\_FILES[$fieldName]['name'], PATHINFO\_EXTENSION);

    $fileName = uniqid() . '.' . $extension;

    $filePath = $uploadDir . $fileName;

    if(move\_uploaded\_file($\_FILES[$fieldName]['tmp\_name'], $filePath)) {

        return "$folder/$fileName";

    }

    return null;

}

?>

Admin.php

<?php

require\_once 'config.php';

session\_start();

// Устанавливаем заголовок ДО любого вывода

header('Content-Type: application/json; charset=utf-8');

header('Access-Control-Allow-Origin: \*');

// Включаем вывод ошибок, но перенаправляем в error\_log

error\_reporting(E\_ALL);

ini\_set('display\_errors', 0);

ini\_set('log\_errors', 1);

// Буферизация вывода чтобы перехватить любые случайные выводы

ob\_start();

try {

    // Определяем действие

    $action = $\_GET['action'] ?? $\_POST['action'] ?? '';

    error\_log("Admin action: " . $action);

    // Логин администратора

    if ($action === 'login') {

        $username = $\_POST['username'] ?? '';

        $password = $\_POST['password'] ?? '';

        $stmt = $pdo->prepare("SELECT \* FROM admins WHERE username = ?");

        $stmt->execute([$username]);

        $admin = $stmt->fetch();

        if ($admin && password\_verify($password, $admin['password\_hash'])) {

            $\_SESSION['admin\_id'] = $admin['id'];

            $\_SESSION['admin\_username'] = $admin['username'];

            $\_SESSION['admin\_logged\_in'] = true;

            echo json\_encode(['success' => true, 'message' => 'Login successful']);

        } else {

            echo json\_encode(['success' => false, 'message' => 'Неверный логин или пароль']);

        }

        exit;

    }

    // Проверка авторизации для всех остальных методов

    function checkAuth() {

        if (!isset($\_SESSION['admin\_logged\_in']) || $\_SESSION['admin\_logged\_in'] !== true) {

            http\_response\_code(401);

            echo json\_encode(['success' => false, 'message' => 'Not authorized']);

            exit;

        }

    }

    // Выход

    if ($action === 'logout') {

        session\_destroy();

        echo json\_encode(['success' => true]);

        exit;

    }

    // Все остальные методы требуют авторизации

    checkAuth();

    // Получение заявок

    if ($action === 'get\_applications') {

        $stmt = $pdo->query("

            SELECT p.\*, GROUP\_CONCAT(c.name SEPARATOR ', ') as competitions

            FROM participants p

            LEFT JOIN participant\_competition pc ON p.id = pc.participant\_id

            LEFT JOIN competitions c ON pc.competition\_id = c.id

            GROUP BY p.id

            ORDER BY p.created\_at DESC

        ");

        $applications = $stmt->fetchAll(PDO::FETCH\_ASSOC);

        echo json\_encode($applications);

        exit;

    }

    // Получение отзывов для модерации

    if ($action === 'get\_reviews\_moderation') {

        $stmt = $pdo->query("

            SELECT r.\*, p.full\_name, p.phone

            FROM reviews r

            JOIN participants p ON r.participant\_id = p.id

            ORDER BY r.is\_approved, r.created\_at DESC

        ");

        $reviews = $stmt->fetchAll(PDO::FETCH\_ASSOC);

        echo json\_encode($reviews);

        exit;

    }

    // Получение галереи

    if ($action === 'get\_gallery') {

        $stmt = $pdo->query("SELECT \* FROM gallery ORDER BY uploaded\_at DESC");

        $gallery = $stmt->fetchAll(PDO::FETCH\_ASSOC);

        echo json\_encode($gallery);

        exit;

    }

    // Получение информации о конкурсе

    if ($action === 'get\_contest\_info') {

        $stmt = $pdo->query("SELECT \* FROM contest\_info ORDER BY id DESC LIMIT 1");

        $result = $stmt->fetch(PDO::FETCH\_ASSOC);

        echo json\_encode($result ?: []);

        exit;

    }

    // Добавление в галерею

   if ($action === 'add\_to\_gallery') {

    error\_log("=== GALLERY UPLOAD DEBUG ===");

    error\_log("POST data: " . print\_r($\_POST, true));

    error\_log("FILES data: " . print\_r($\_FILES, true));

    error\_log("Content-Type: " . ($\_SERVER['CONTENT\_TYPE'] ?? 'not set'));

    // Детальная проверка файла

    if (!isset($\_FILES['image'])) {

        error\_log("❌ No 'image' key in FILES array");

        echo json\_encode(['success' => false, 'message' => 'Файл не найден в запросе']);

        exit;

    }

    $fileInfo = $\_FILES['image'];

    error\_log("File info - name: " . ($fileInfo['name'] ?? 'empty') .

              ", size: " . ($fileInfo['size'] ?? 'empty') .

              ", error: " . ($fileInfo['error'] ?? 'empty') .

              ", tmp\_name: " . ($fileInfo['tmp\_name'] ?? 'empty'));

    // Проверяем ошибки загрузки

    if ($fileInfo['error'] !== UPLOAD\_ERR\_OK) {

        $errorMessages = [

            UPLOAD\_ERR\_INI\_SIZE => 'Файл превышает максимальный размер',

            UPLOAD\_ERR\_FORM\_SIZE => 'Файл превышает размер указанный в форме',

            UPLOAD\_ERR\_PARTIAL => 'Файл загружен только частично',

            UPLOAD\_ERR\_NO\_FILE => 'Файл не был загружен',

            UPLOAD\_ERR\_NO\_TMP\_DIR => 'Отсутствует временная папка',

            UPLOAD\_ERR\_CANT\_WRITE => 'Не удалось записать файл на диск',

            UPLOAD\_ERR\_EXTENSION => 'Расширение PHP остановило загрузку'

        ];

        $errorCode = $fileInfo['error'];

        $errorMsg = $errorMessages[$errorCode] ?? "Неизвестная ошибка: $errorCode";

        error\_log("❌ File upload error: $errorMsg (code: $errorCode)");

        echo json\_encode(['success' => false, 'message' => $errorMsg]);

        exit;

    }

    // Проверяем что файл действительно загружен

    if (empty($fileInfo['tmp\_name']) || !file\_exists($fileInfo['tmp\_name'])) {

        error\_log("❌ Temporary file doesn't exist or is empty");

        echo json\_encode(['success' => false, 'message' => 'Временный файл не найден']);

        exit;

    }

    // Проверяем размер файла

    if ($fileInfo['size'] === 0) {

        error\_log("❌ File size is 0");

        echo json\_encode(['success' => false, 'message' => 'Файл пустой']);

        exit;

    }

    error\_log("✅ File validation passed");

    // Создаем папку если нужно

    $uploadDir = "../uploads/gallery/";

    if (!is\_dir($uploadDir)) {

        error\_log("Creating directory: $uploadDir");

        if (!mkdir($uploadDir, 0755, true)) {

            error\_log("❌ Failed to create directory");

            echo json\_encode(['success' => false, 'message' => 'Не удалось создать папку для загрузки']);

            exit;

        }

    }

    // Проверяем права доступа

    if (!is\_writable($uploadDir)) {

        error\_log("❌ Directory not writable: $uploadDir");

        echo json\_encode(['success' => false, 'message' => 'Папка не доступна для записи']);

        exit;

    }

    // Генерируем имя файла

    $extension = pathinfo($fileInfo['name'], PATHINFO\_EXTENSION);

    $fileName = uniqid() . '.' . $extension;

    $filePath = $uploadDir . $fileName;

    error\_log("Moving file from: " . $fileInfo['tmp\_name'] . " to: " . $filePath);

    // Перемещаем файл

    if (move\_uploaded\_file($fileInfo['tmp\_name'], $filePath)) {

        // Проверяем что файл действительно создан

        if (file\_exists($filePath)) {

            $fileSize = filesize($filePath);

            error\_log("✅ File successfully moved. Size: $fileSize bytes");

            // Сохраняем в БД

            $imagePath = "gallery/" . $fileName;

            $title = $\_POST['title'] ?? 'Без названия';

            $stmt = $pdo->prepare("INSERT INTO gallery (admin\_id, image\_path, title) VALUES (?, ?, ?)");

            $stmt->execute([$\_SESSION['admin\_id'], $imagePath, $title]);

            error\_log("✅ Database record created for: $imagePath");

            echo json\_encode(['success' => true, 'message' => 'Фото добавлено в галерею']);

        } else {

            error\_log("❌ File doesn't exist after move\_uploaded\_file");

            echo json\_encode(['success' => false, 'message' => 'Файл не был сохранен']);

        }

    } else {

        error\_log("❌ move\_uploaded\_file returned false");

        $lastError = error\_get\_last();

        error\_log("Last error: " . print\_r($lastError, true));

        echo json\_encode(['success' => false, 'message' => 'Ошибка при сохранении файла']);

    }

    exit;

}

    // Одобрение/отклонение отзыва

    if ($action === 'moderate\_review') {

        $reviewId = $\_POST['review\_id'];

        $approved = $\_POST['approved'] ? 1 : 0;

        $stmt = $pdo->prepare("UPDATE reviews SET is\_approved = ?, admin\_id = ? WHERE id = ?");

        $stmt->execute([$approved, $\_SESSION['admin\_id'], $reviewId]);

        echo json\_encode(['success' => true]);

        exit;

    }

    // Удаление отзыва

    if ($action === 'delete\_review') {

        $stmt = $pdo->prepare("DELETE FROM reviews WHERE id = ?");

        $stmt->execute([$\_POST['review\_id']]);

        echo json\_encode(['success' => true]);

        exit;

    }

    // Удаление из галереи

    if ($action === 'delete\_from\_gallery') {

        $stmt = $pdo->prepare("DELETE FROM gallery WHERE id = ?");

        $stmt->execute([$\_POST['image\_id']]);

        echo json\_encode(['success' => true]);

        exit;

    }

    // Обновление информации о конкурсе

    if ($action === 'update\_contest\_info') {

        $infoId = $\_POST['info\_id'] ?? 1;

        // Проверяем существует ли запись

        $stmt = $pdo->prepare("SELECT id FROM contest\_info WHERE id = ?");

        $stmt->execute([$infoId]);

        $exists = $stmt->fetch();

        if ($exists) {

            $stmt = $pdo->prepare("UPDATE contest\_info SET rules = ?, schedule = ?, contacts = ?, admin\_id = ? WHERE id = ?");

            $stmt->execute([$\_POST['rules'], $\_POST['schedule'], $\_POST['contacts'], $\_SESSION['admin\_id'], $infoId]);

        } else {

            $stmt = $pdo->prepare("INSERT INTO contest\_info (rules, schedule, contacts, admin\_id) VALUES (?, ?, ?, ?)");

            $stmt->execute([$\_POST['rules'], $\_POST['schedule'], $\_POST['contacts'], $\_SESSION['admin\_id']]);

        }

        echo json\_encode(['success' => true]);

        exit;

    }

    if ($action === 'delete\_application') {

    error\_log("=== DELETE APPLICATION DEBUG ===");

    error\_log("POST data: " . print\_r($\_POST, true));

    $appId = $\_POST['app\_id'] ?? 0;

    if (!$appId) {

        echo json\_encode(['success' => false, 'message' => 'ID заявки не указан']);

        exit;

    }

    try {

        $pdo->beginTransaction();

        // Сначала получаем пути к файлам для удаления

        $stmt = $pdo->prepare("SELECT photo\_path, music\_path FROM participants WHERE id = ?");

        $stmt->execute([$appId]);

        $participant = $stmt->fetch();

        if (!$participant) {

            echo json\_encode(['success' => false, 'message' => 'Заявка не найдена']);

            exit;

        }

        // Удаляем связанные записи из participant\_competition

        $stmt = $pdo->prepare("DELETE FROM participant\_competition WHERE participant\_id = ?");

        $stmt->execute([$appId]);

        // Удаляем связанные отзывы

        $stmt = $pdo->prepare("DELETE FROM reviews WHERE participant\_id = ?");

        $stmt->execute([$appId]);

        // Удаляем саму заявку

        $stmt = $pdo->prepare("DELETE FROM participants WHERE id = ?");

        $stmt->execute([$appId]);

        $pdo->commit();

        // Удаляем файлы с сервера

        if (!empty($participant['photo\_path'])) {

            $photoFile = "../uploads/" . $participant['photo\_path'];

            if (file\_exists($photoFile)) {

                unlink($photoFile);

            }

        }

        if (!empty($participant['music\_path'])) {

            $musicFile = "../uploads/" . $participant['music\_path'];

            if (file\_exists($musicFile)) {

                unlink($musicFile);

            }

        }

        error\_log("✅ Application #$appId successfully deleted");

        echo json\_encode(['success' => true, 'message' => 'Заявка успешно удалена']);

    } catch (Exception $e) {

        $pdo->rollBack();

        error\_log("❌ Error deleting application: " . $e->getMessage());

        echo json\_encode(['success' => false, 'message' => 'Ошибка при удалении заявки: ' . $e->getMessage()]);

    }

    exit;

}

    // Если действие не распознано

    echo json\_encode(['success' => false, 'message' => 'Unknown action: ' . $action]);

} catch (Exception $e) {

    // Перехватываем все исключения и возвращаем JSON ошибку

    error\_log("Admin PHP Error: " . $e->getMessage());

    echo json\_encode(['success' => false, 'message' => 'Server error: ' . $e->getMessage()]);

}

// Очищаем буфер на случай если что-то случайно вывелось

ob\_end\_clean();

?>

Config.php

<?php

$host = 'sql209.infinityfree.com';

$dbname = 'if0\_40349545\_projektor\_contest';

$username = 'if0\_40349545';

$password = 'MfmO1T5714nkH';

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8mb4", $username, $password);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->setAttribute(PDO::ATTR\_DEFAULT\_FETCH\_MODE, PDO::FETCH\_ASSOC);

} catch(PDOException $e) {

error\_log("Database connection failed: " . $e->getMessage());

die("Connection failed: " . $e->getMessage());

}

?>

Dashboard.php

<?php

session\_start();

// Отладка сессии

error\_log("Dashboard session: " . print\_r($\_SESSION, true));

if (!isset($\_SESSION['admin\_logged\_in']) || $\_SESSION['admin\_logged\_in'] !== true) {

    error\_log("Redirecting to login - not authorized");

    header('Location: login.php');

    exit;

}

?>

# Скриншоты

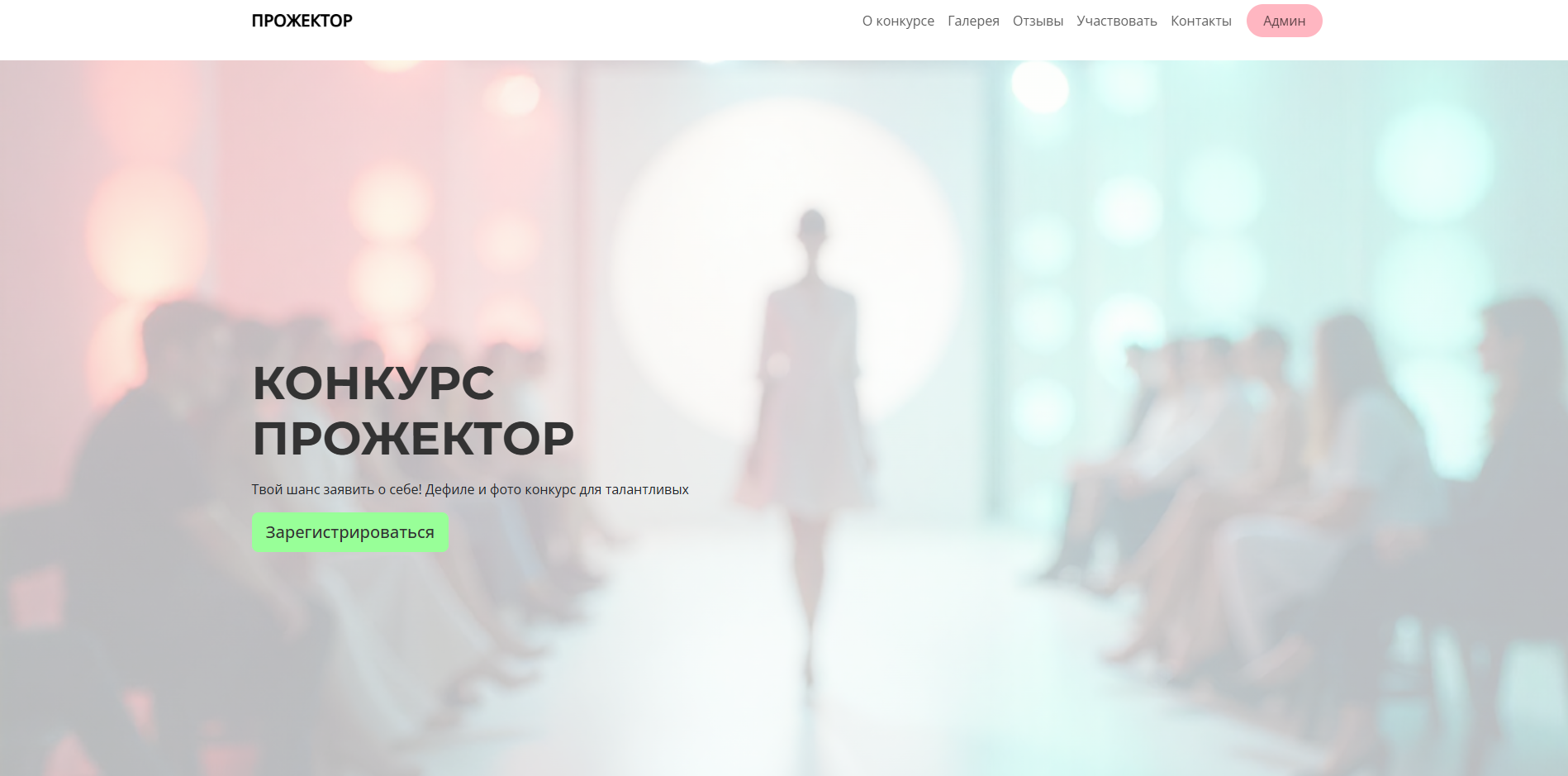


Рисунок 11 – Баннер

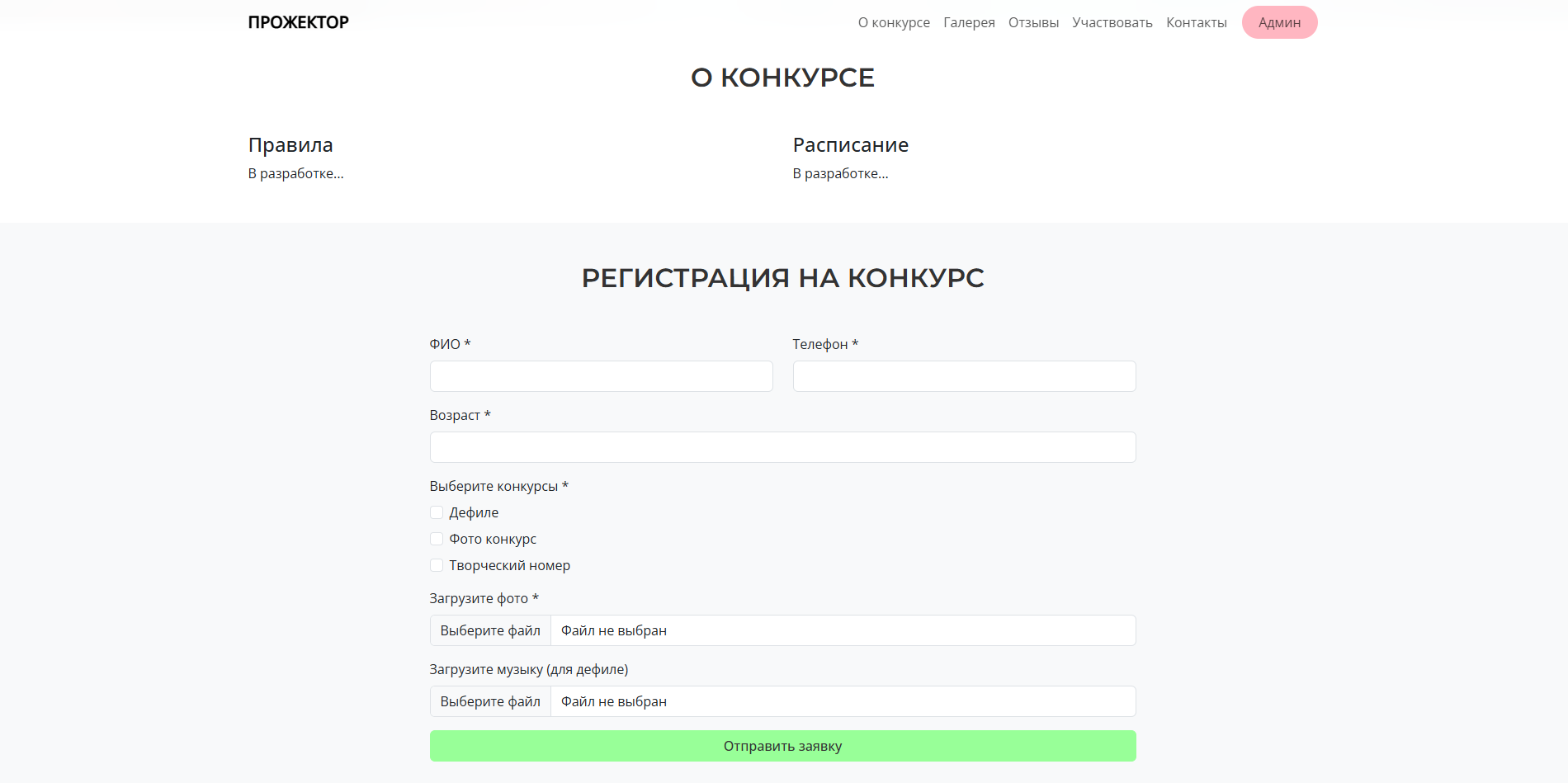


Рисунок 12 – Описание и регистрация

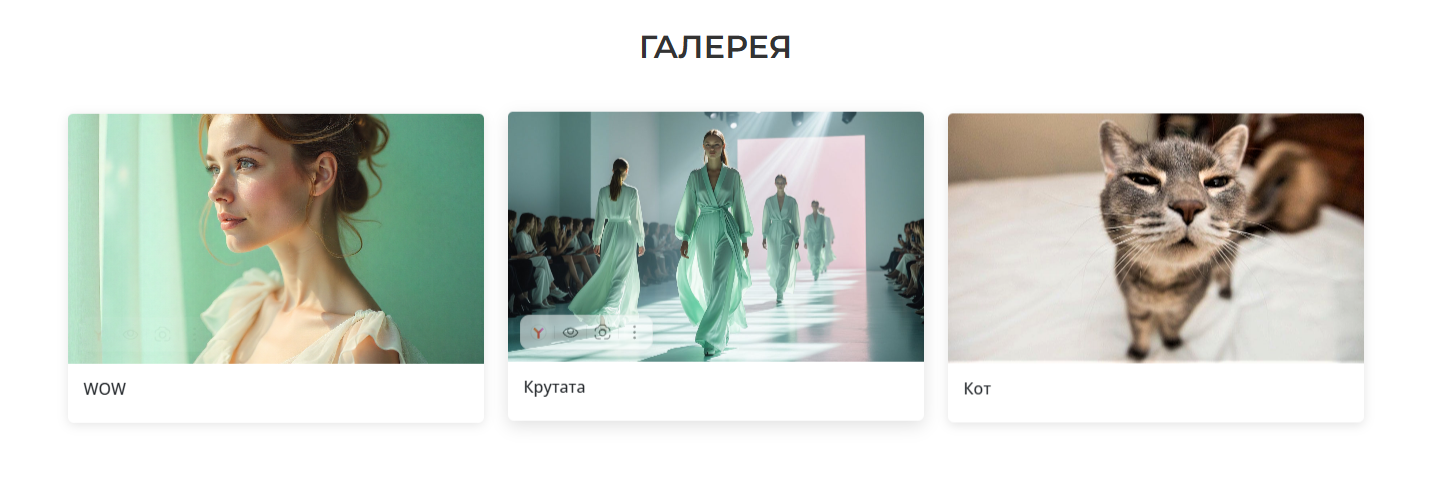


Рисунок 13 – Галерея

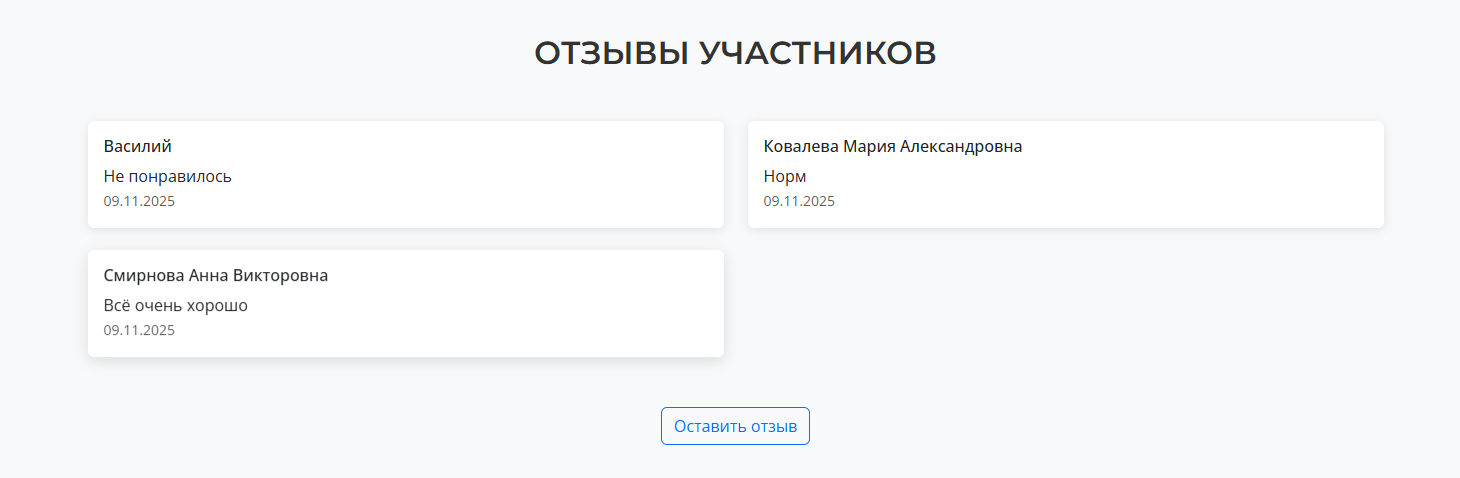


Рисунок 14 – Отзывы

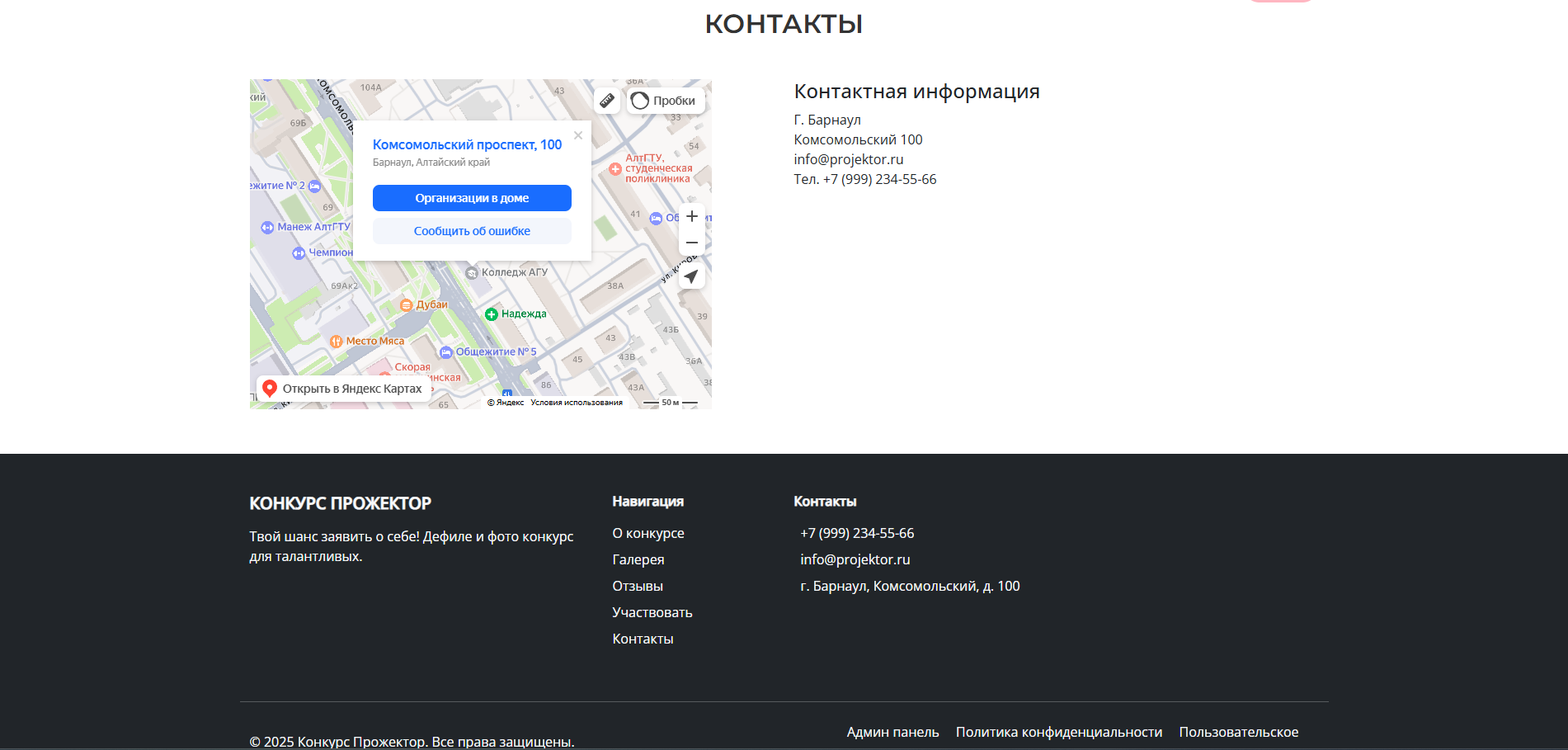


Рисунок 15 – Контакты и подвал

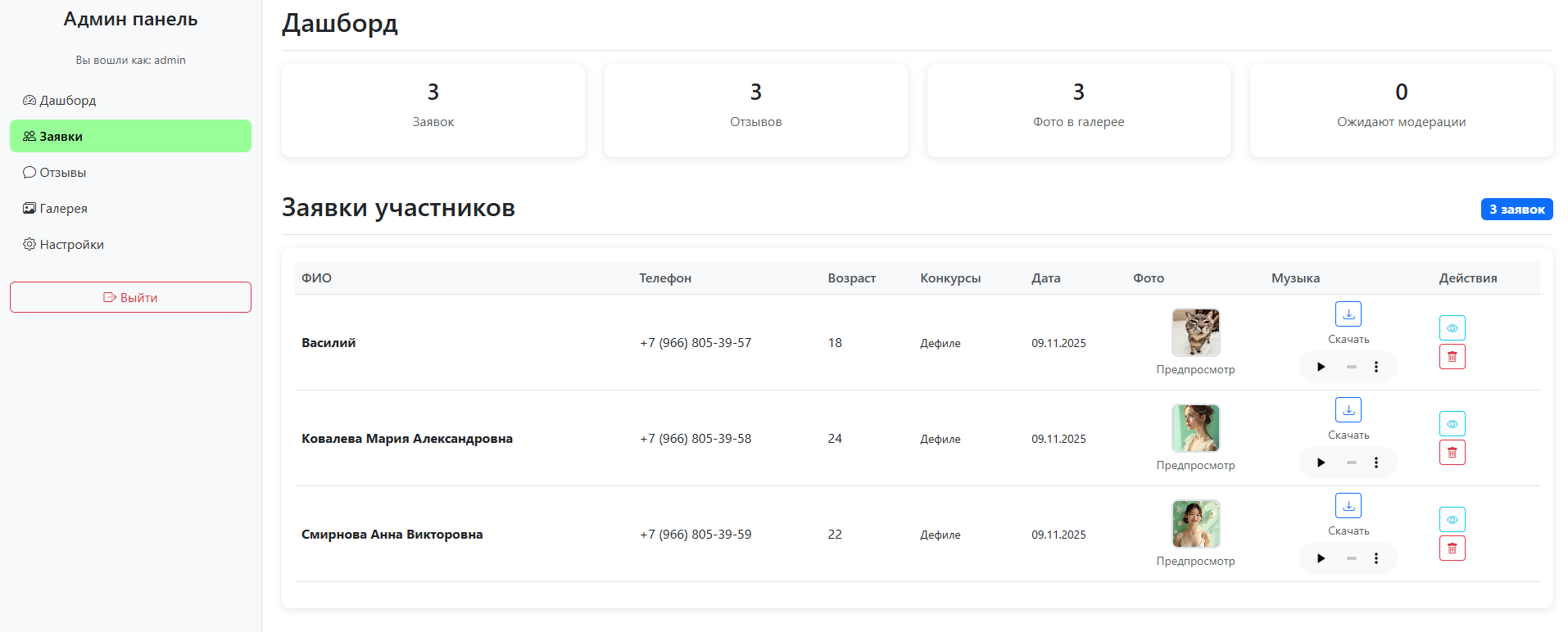


Рисунок 16 – Заявки

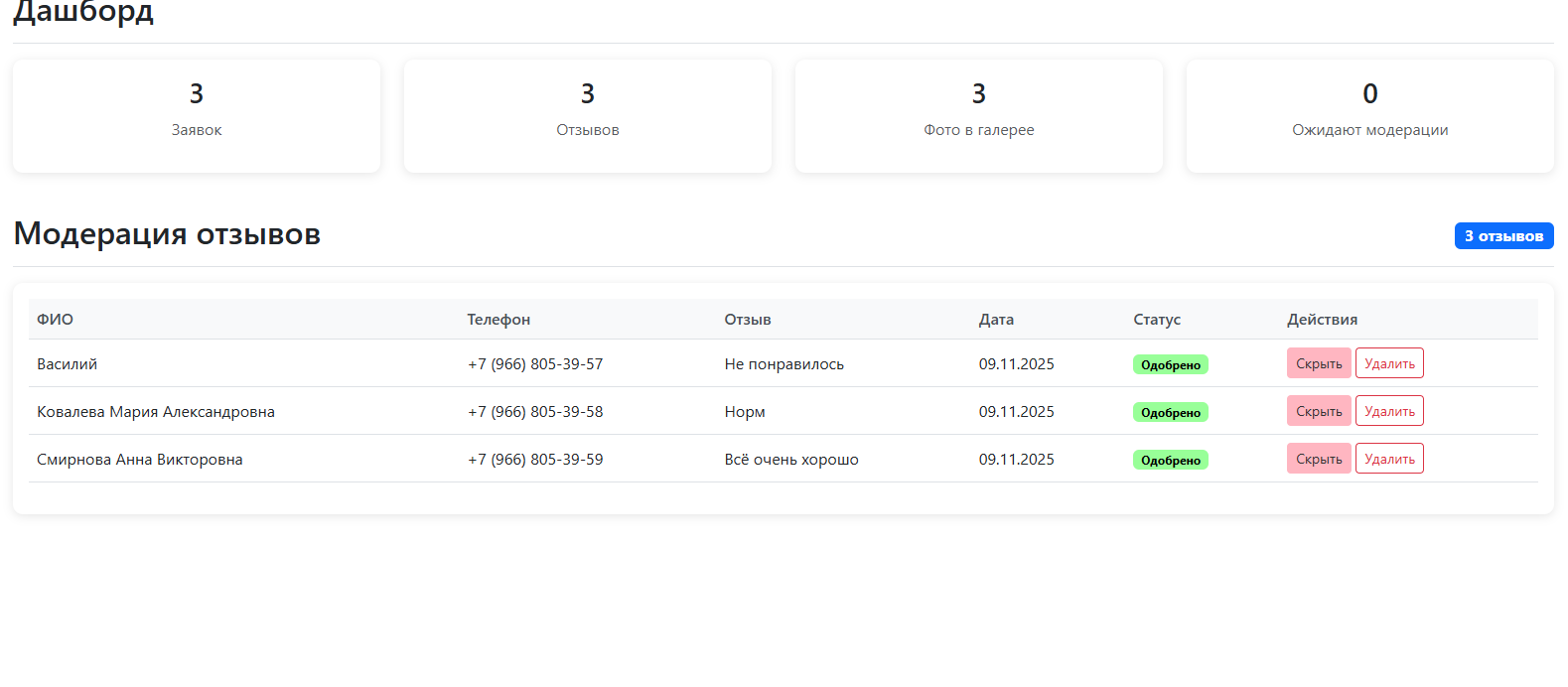


Рисунок 17 – Отзывы

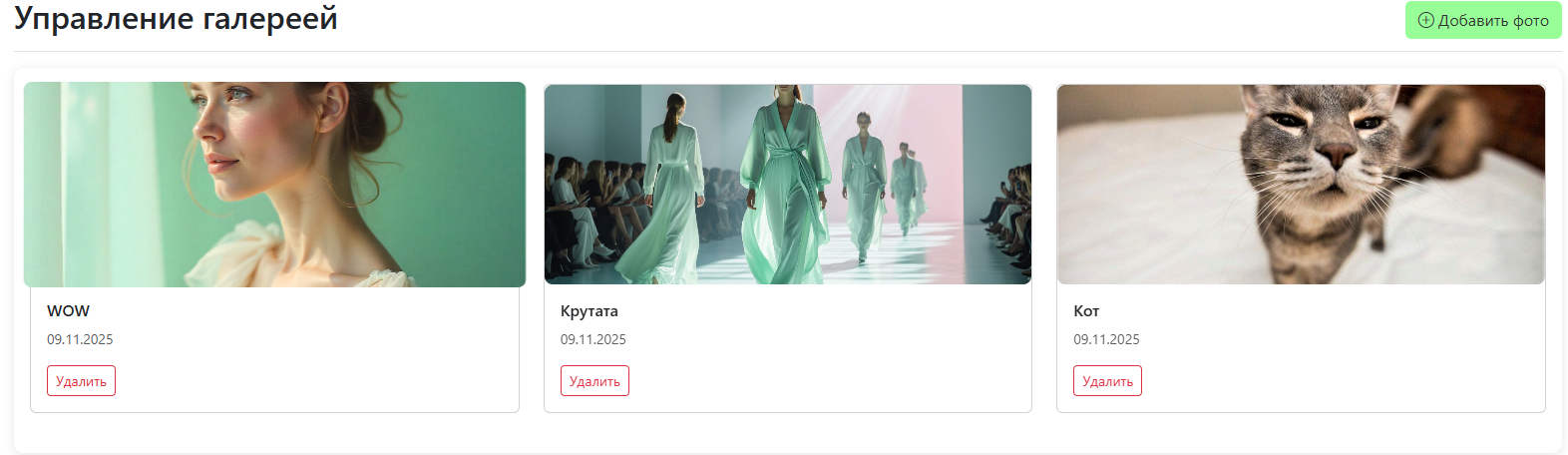


Рисунок 18 – Галерея

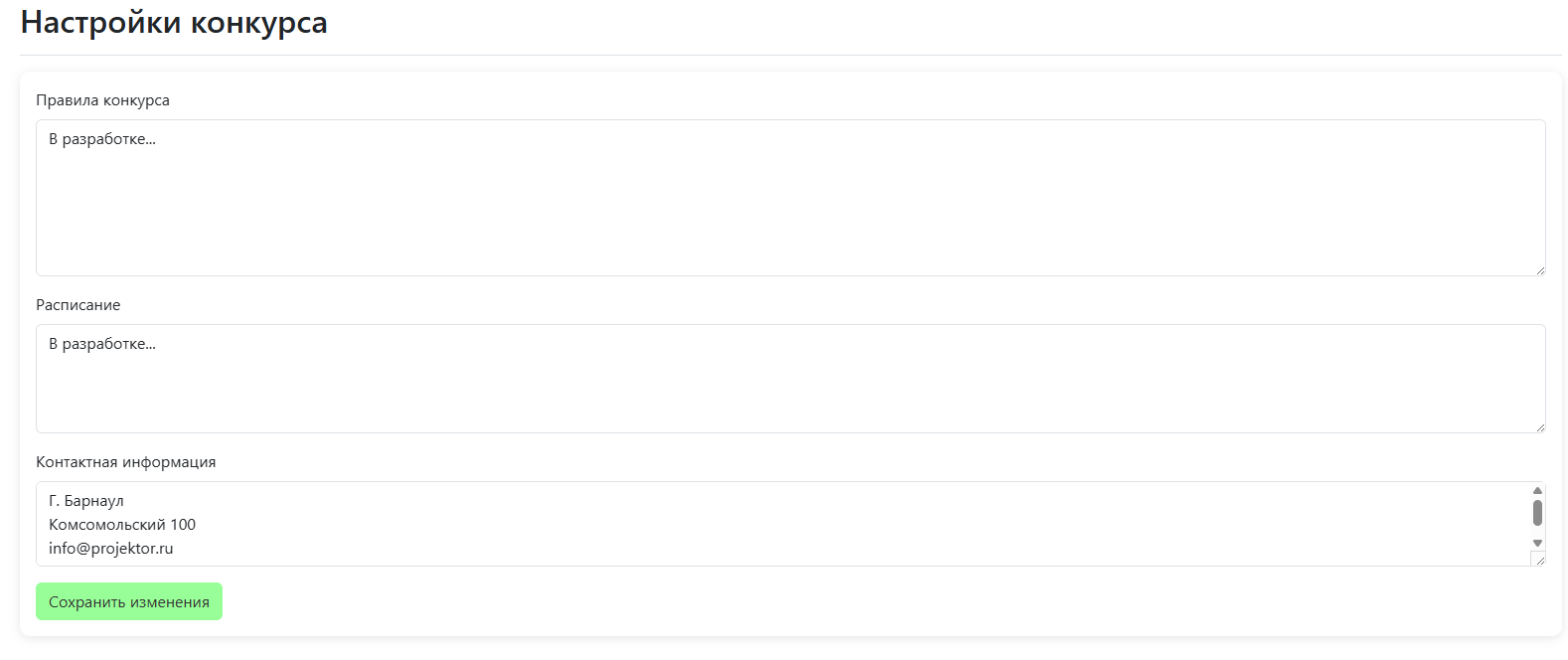


Рисунок 19 – Настройки

# Ссылка

https://vasil205.infinityfreeapp.com