Министерство науки и высшего образования Российской Федерации **Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра информационных систем и программной инженерии

**КУРСОВОЙ ПРОЕКТ**

Приложение для информационной системы «Виртуальная арт-галерея».

Модуль «Эксплуатация»

Выполнил: студент гр. ПРИ-122

Шутова Т.Е.

Принял: доц. Вершинин В.В.

Владимир, 2024

**СОДЕРЖАНИЕ**

[ЭТАП 1 3](#_Toc181913530)

[1.1 Постановка задачи 3](#_Toc181913531)

[1.2 Описание предметной области 3](#_Toc181913532)

[1.3 Словарь предметной области 3](#_Toc181913533)

[1.4 Диаграмма прецедентов 4](#_Toc181913534)

[1.5 Диаграмма классов 7](#_Toc181913535)

[ЭТАП 2 9](#_Toc181913536)

[2.1 Диаграмма состояний 9](#_Toc181913537)

[2.2 Диаграмма последовательностей 10](#_Toc181913538)

[2.3 Диаграмма видов деятельности 12](#_Toc181913539)

ЭТАП 1

# 1.1 Постановка задачи

Цель работы: разработать программную информационную систему «Виртуальная арт-галерея» с функционалом, связанным преимущественно с областью эксплуатации.

Для достижения поставленной цели необходимо реализовать следующие функциональные требования:

* Авторизация и регистрация (администратор, художник, посетитель);
* Система поиска публикаций, выставок, художников (администратор, художник, посетитель);
* Прайс-лист в профиле художника (администратор, художник);
* Публикация работ (администратор, художник);
* Организация закрытых и открытых выставок (администратор, художник);
* Участие в открытых и закрытых выставках (администратор, художник);

# 1.2 Описание предметной области

Для художников необходимо иметь платформу, на которой можно размещать собственные работы с целью получения известности или поиска работодателя. Представителям компаний, в чьи ряды требуется графический художник той или иной специальности, важно иметь на руках портфолио кандидата, и именно платформы с графическими публикациями не только упрощают поиск, но и дают доступ к работам и ценам на услуги художника.

Виртуальная арт-галерея – это как раз платформа для художников и их нанимателей. Яркий пример – зарубежный сайт artstation.com. Российского аналога нет, следовательно, требуется заняться его разработкой.

# 1.3 Словарь предметной области

*Посетитель* – незарегистрированный пользователь. Имеет минимальные права, в основном может только просматривать работы, художников и выставки.

Атрибуты: ID.

*Художник* – зарегистрированный и прошедший авторизацию пользователь. Художнику не обязательно иметь в своём профиле работы, это только роль, присваиваемая при регистрации. Художник может публиковать работы, устраивать выставки, создавать прайс-листы.

Атрибуты: ID, никнейм, пароль, почта.

*Администратор* – зарегистрированный пользователь, имеющий доступ к редактированию содержимого чужих публикаций, чужих выставок и художников. Администратор следит за соблюдением правил платформы, удаляет нарушения, но также может позиционировать себя как художник.

Атрибуты: ID, никнейм, пароль, почта, статус.

*Публикация* – действие, после которого произведение становится доступным определенному кругу лиц, или обозначение доступной для публики работы художника.

Атрибуты: ID, название, описание, изображение, статус публикации, дата создания, категории.

*Прайс-лист* – общедоступное описание возможных услуг, которые художник готов предоставить, и примерная цена за них.

Атрибуты: ID, категории прайсов вместе с их названиями и ценой.

*Выставка* – собранная по определенному критерию коллекция работ художника или художников, доступная для редактирования заранее указанному кругу лиц.

Атрибуты: ID, название, описание, публичность, работы, категории.

*Категория* – характеристика того или иного объекта, по которой его можно сгруппировать с другими объектами, имеющими ту же характеристику.

Атрибуты: ID, название, описание.

# 1.4 Диаграмма прецедентов

Диаграмма прецедентов представлена на рисунке 1.



Рисунок 1. Диаграмма прецедентов

|  |
| --- |
| Расширенное описание прецедента «Публикация» |
| **Название:** «Публикация»  **Предусловие:** нет  **Действующее лицо:** художник, администратор  **Основной поток:** *Публикация*  Пользователь нажимает на кнопку «Публикация» и переходит в форму для публикации.  Обязательно нужно загрузить изображение. По желанию можно оформить описание и выбрать категории.  Работа публикуется в профиле художника и в соответствующих категориях.  **Альтернативный поток:** *нет*  **Постусловие:** пользователя автоматически перебрасывает на страницу с опубликованной работой. |

|  |
| --- |
| Расширенное описание прецедента «Поиск» |
| **Название:** Поиск  **Предусловие:** нет  **Действующее лицо:** посетитель, художник, администратор  **Основной поток:** *Поиск*  Пользователь вбивает в строку поиска то, что ему необходимо найти.  После его перебрасывает на страницу, где предлагается 3 варианта поиска: среди публикаций, среди художников и среди выставок. В каждой категории отображается несколько примеров, подходящих под условие поиска.  Выбрав конкретную категорию, пользователь может отфильтровать поиск под свои нужды.  **Альтернативный поток:** *Ничего не найдено*  В том случае, если в одной, двух или во всех категориях ничего подходящего под запрос не было найдено, на странице отображается текст в каждой категории: «Ничего не найдено».  **Постусловие:** после выбора необходимого варианта, пользователь переходит на нужную страницу, строка поиска автоматически очищается. |

|  |
| --- |
| Расширенное описание прецедента «Создание прайс-листа» |
| **Название:** Создание прайс-листа  **Предусловие:** нет  **Действующее лицо:** художник, администратор  **Основной поток:** *Создание прайс-листа*  Пользователь нажимает на кнопку «Создать прайс-лист», после чего его перебрасывает на форму заполнения.  После заполнения обязательных полей и необязательно по выбору, прайс-лист автоматически публикуется в профиле художника  **Альтернативный поток:** *нет*  **Постусловие:** после создания прайс-листа пользователя перебрасывает в свой профиль, где можно посмотреть прайс. |

|  |
| --- |
| Расширенное описание прецедента «Создание выставок» |
| **Название:** Создание выставок  **Предусловие:** нет  **Действующее лицо:** художник, администратор  **Основной поток:** *Создание выставок*  Пользователь нажимает на кнопку «Создать выставку», после чего его перебрасывает на страницу с заполняемой формой.  Выставке задаются категории, описание, открытость и др.  Пустая выставка автоматически добавляется на сайт и в профиль художника.  **Альтернативный поток:** *нет*  **Постусловие:** пользователя автоматически перебрасывает на страницу с созданной выставкой. |

# 1.5 Диаграмма классов

Диаграмма классов представлена на рисунке 2.

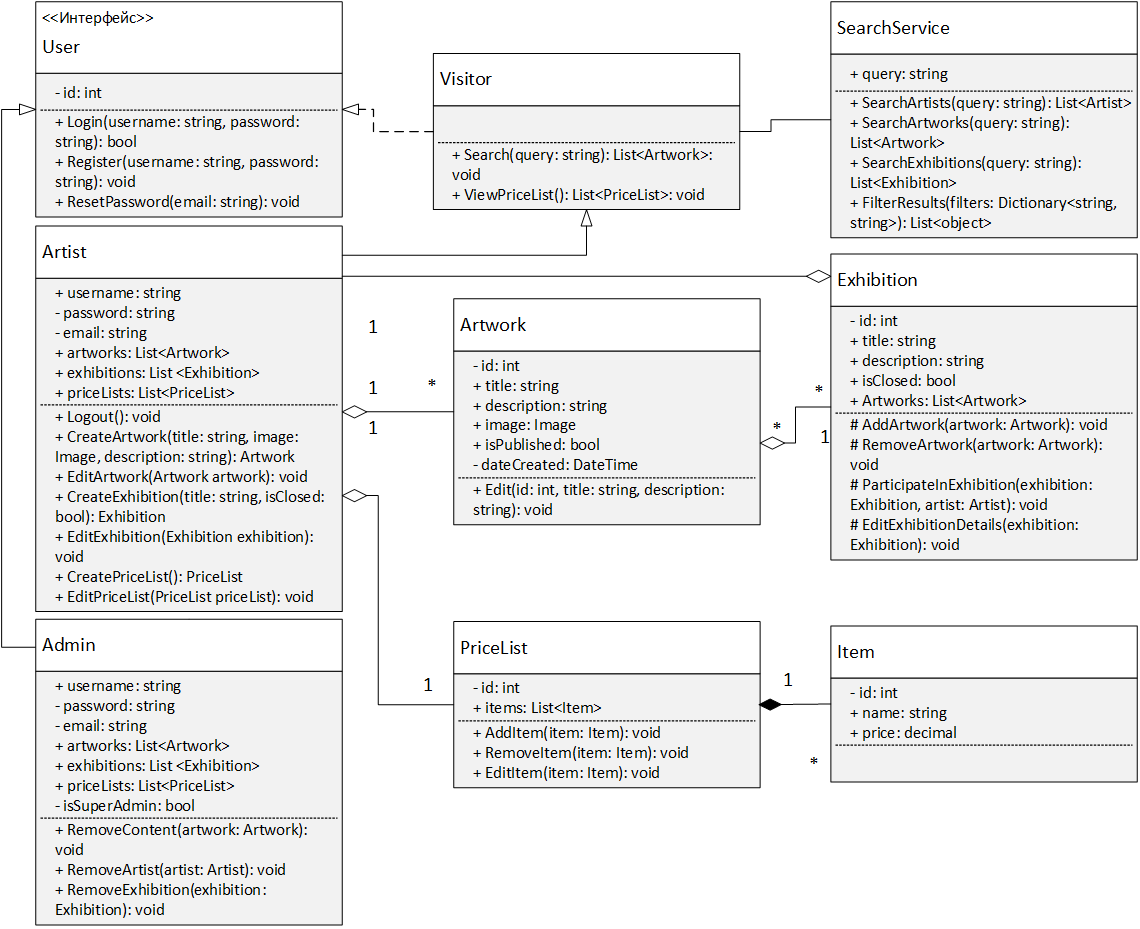


Рисунок 2. Диаграмма классов

ЭТАП 2

# 2.1 Диаграмма состояний

Диаграмма состояний для объекта «Выставка» представлена на рисунке 3.

**Описание диаграммы состояний для объекта «Выставка»**

**1) Форма заполнения выставки (черновик)**

Пользователь собирается создать выставку и нажимает на кнопку. Объект «Выставка» создаётся в системе, но он доступен только для создателя выставки. Создать выставку может только зарегистрированный пользователь.

**2) Ввод информации о выставке**

Ввод названия, описания, категории и прочих атрибутов выставки. К атрибутам объекта «Выставка» добавляются значения. Выставка публикуется.

**4) Добавление работ**

К объекту «Выставка» добавляются объекты «Публикация», которые отправляются на рассмотрение администратору.

**5) Предварительный просмотр**

Пользователь просматривает предварительный вариант выставки, находящейся в состоянии черновика.

**6) Работы отклонены (цикл)**

Администратор посчитал, что публикации не подходят правилам сайта, поэтому отклоняет их публикацию. Художник может попытаться снова до тех пор, пока работы не будут опубликованы.

**7) Выставка обновлена**

Если администратор одобрил работы, выставка обновляется. Выставку можно обновлять много раз, необязательно это должен делать только создатель выставки.

**8) Удаление выставки**

Администратор или создатель выставки может удалить выставку. Тогда данные о выставке не сохраняются.

**9) Закрытие выставки**

Создатель выставки закрывает доступ к выставке, после чего другие пользователи не могут с ней взаимодействовать. Фактически, выставка остаётся в архиве, её можно смотреть, но далее обновлений в ней не предвидится.

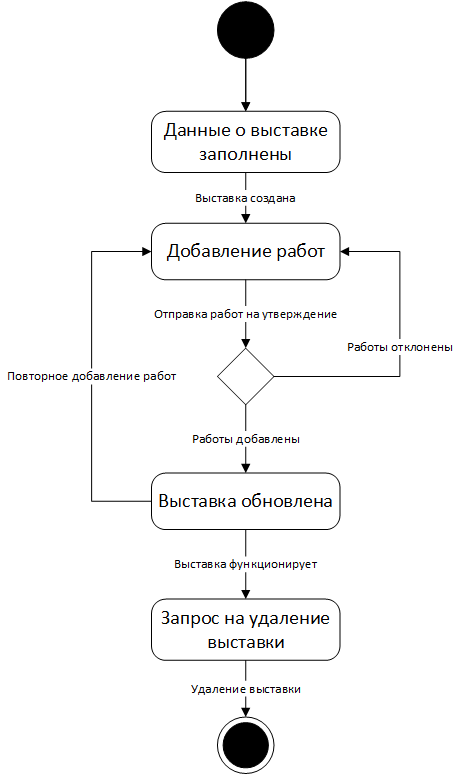


Рисунок 3. Диаграмма состояний

# 2.2 Диаграмма последовательностей

Диаграмма последовательностей для объекта «Выставка» представлена на рисунке 4.

**Описание диаграммы последовательностей объекта «Выставка».**

1. Художник делает запрос на создание выставки.

2. Выставка создаётся и автоматически публикуется.

3. Художник делает запрос на изменение выставки.

4. Если художнику нужно добавить публикацию, выставка обращается к созданию публикации.

5. Публикация отправляется на рассмотрение администратору.

6. Если администратор одобряет публикацию, она публикуется.

7. Публикация добавляется в выставку.

8. Выставка обновлена.

9. Те же операции по добавлению публикаций могут проделываться другие пользователи, если выставка открыта для редактирования.

10. Если художнику нужно удалить выставку, он делает запрос на удаление.

11. Выставка безопасно удаляет публикации, относящиеся к выбранной выставке.

12. Выставка удаляется.

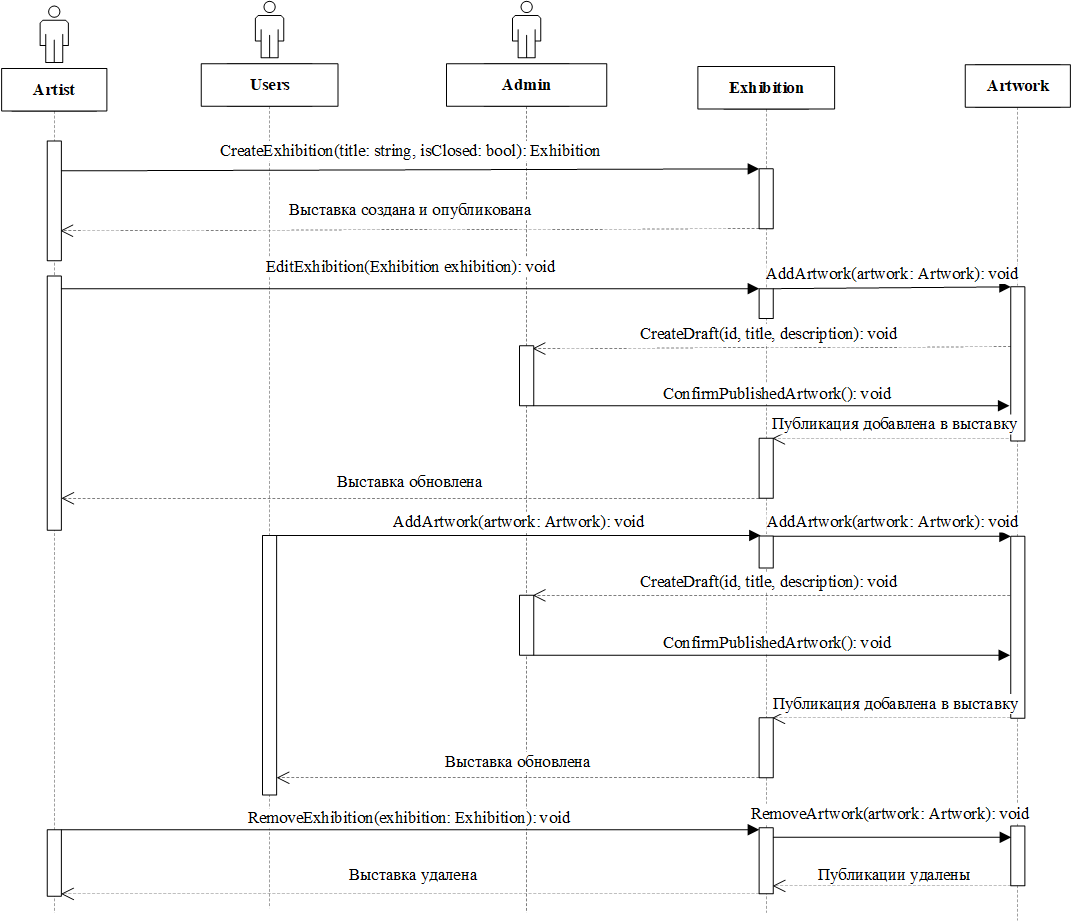


Рисунок 4. Диаграмма последовательностей

# 2.3 Диаграмма видов деятельности

Диаграмма видов деятельности для объекта «Выставка» представлена на рисунке 5.

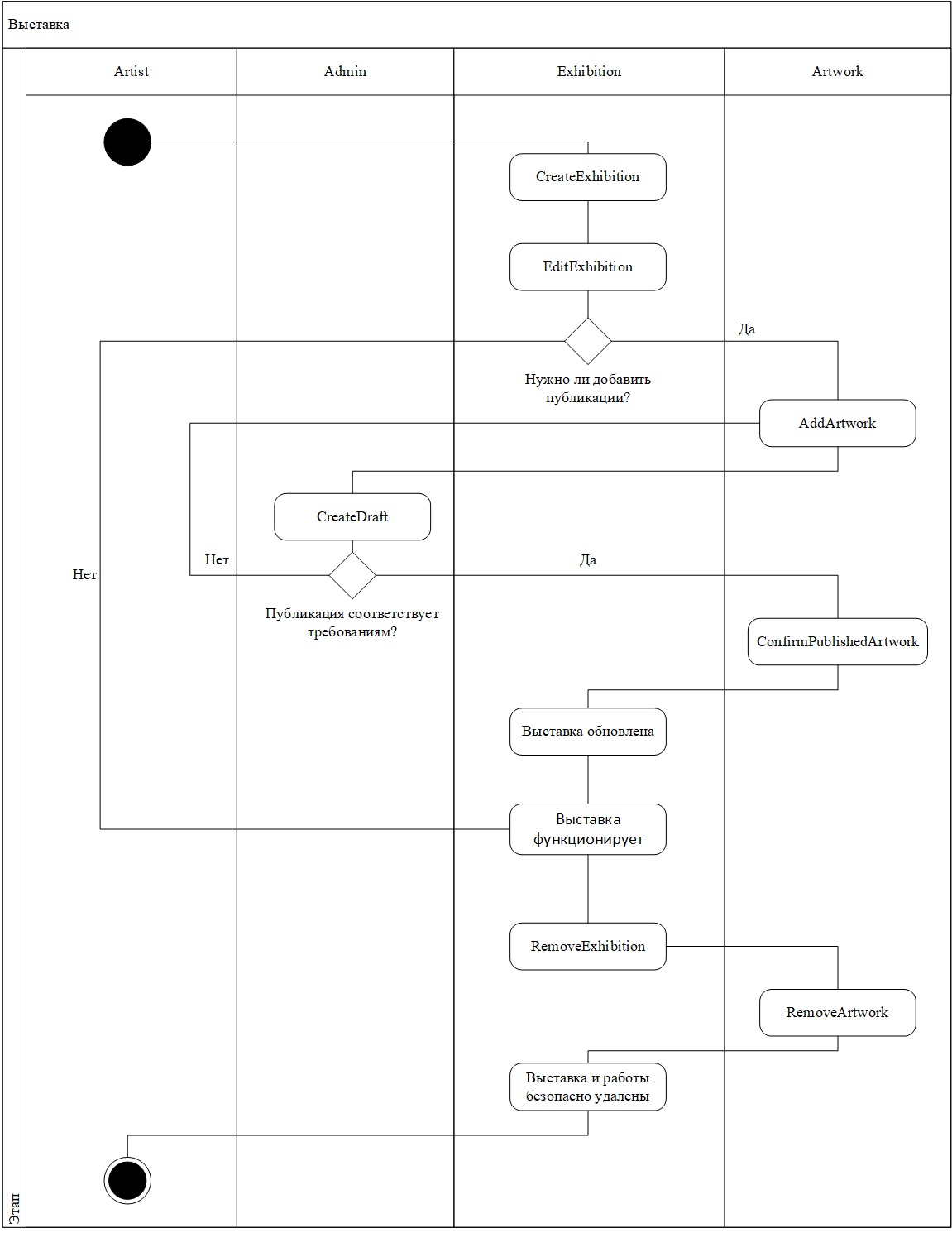


Рисунок 5. Диаграмма видов деятельности

ЭТАП 3

# Код ExhibitionModel:

using System.ComponentModel.DataAnnotations;

namespace Virtual\_Art\_Gallery.Models

{

public class ExhibitionModel

{

[Key]

public int Id { get; set; }

[Required]

public string Title { get; set; }

[Required]

public string Description { get; set; }

[Required]

public bool IsClosed { get; set; }

//public List<ArtworkModel> Artwork { get; set; }

}

}

# Код ExhibitionController:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using Virtual\_Art\_Gallery.Data;

using Virtual\_Art\_Gallery.Models;

namespace Virtual\_Art\_Gallery.Controllers

{

public class ExhibitionController : Controller

{

private readonly GalleryContext \_context;

public ExhibitionController(GalleryContext context)

{

\_context = context;

}

// GET: Exhibition

public async Task<IActionResult> Index()

{

return View(await \_context.Exhibitions.ToListAsync());

}

// GET: Exhibition/Details/5

public async Task<IActionResult> Details(int id)

{

if (id == null)

{

return NotFound();

}

var exhibitionModel = await \_context.Exhibitions

.FirstOrDefaultAsync(m => m.Id == id);

if (exhibitionModel == null)

{

return NotFound();

}

return View(exhibitionModel);

}

// GET: Exhibition/Create

public IActionResult Create()

{

return View();

}

// POST: Exhibition/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("Id,Title,Description,IsClosed")] ExhibitionModel exhibitionModel)

{

if (ModelState.IsValid)

{

\_context.Add(exhibitionModel);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(exhibitionModel);

}

// GET: Exhibition/Edit/5

public async Task<IActionResult> Edit(int id)

{

if (id == null)

{

return NotFound();

}

var exhibitionModel = await \_context.Exhibitions.FindAsync(id);

if (exhibitionModel == null)

{

return NotFound();

}

return View(exhibitionModel);

}

// POST: Exhibition/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, [Bind("Id,Title,Description,IsClosed")] ExhibitionModel exhibitionModel)

{

if (id != exhibitionModel.Id)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(exhibitionModel);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!ExhibitionModelExists(exhibitionModel.Id))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

return View(exhibitionModel);

}

// GET: Exhibition/Delete/5

public async Task<IActionResult> Delete(int id)

{

if (id == null)

{

return NotFound();

}

var exhibitionModel = await \_context.Exhibitions

.FirstOrDefaultAsync(m => m.Id == id);

if (exhibitionModel == null)

{

return NotFound();

}

return View(exhibitionModel);

}

// POST: Exhibition/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(int id)

{

var exhibitionModel = await \_context.Exhibitions.FindAsync(id);

if (exhibitionModel != null)

{

\_context.Exhibitions.Remove(exhibitionModel);

}

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

private bool ExhibitionModelExists(int id)

{

return \_context.Exhibitions.Any(e => e.Id == id);

}

}

}

# Код Exhibition/Index:

@model IEnumerable<Virtual\_Art\_Gallery.Models.ExhibitionModel>

@{

ViewData["Title"] = "Выставки";

}

<h1>Выставки</h1>

<p>

<a asp-action="Create">Создать выставку</a>

</p>

<table class="table">

<thead>

<tr>

<th>

Название

</th>

<th>

Описание

</th>

<th>

Открытость

</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Title)

</td>

<td>

@Html.DisplayFor(modelItem => item.Description)

</td>

<td>

@Html.DisplayFor(modelItem => item.IsClosed)

</td>

<td>

<a asp-action="Edit" asp-route-id="@item.Id">Редактировать</a> |

<a asp-action="Details" asp-route-id="@item.Id">Посмотреть</a> |

<a asp-action="Delete" asp-route-id="@item.Id">Удалить</a>

</td>

</tr>

}

</tbody>

</table>

# Код Exhibition/Create:

@model Virtual\_Art\_Gallery.Models.ExhibitionModel

@{

ViewData["Title"] = "Создание выставки";

}

<h1>Создать</h1>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="Create">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="form-group">

<**label** **asp-for**="Title" class="control-label">Название выставки</**label**>

<**input** **asp-for**="Title" class="form-control" />

<**span** **asp-validation-for**="Title" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Description" class="control-label">Описание выставки</**label**>

<**input** **asp-for**="Description" class="form-control" />

<**span** **asp-validation-for**="Description" class="text-danger"></**span**>

</div>

<div class="form-group form-check">

<label class="form-check-label">

<**input** class="form-check-input" **asp-for**="IsClosed" /> Разрешить другим пользователям добавлять работы в эту выставку

</label>

</div>

<div class="form-group">

<input type="submit" value="Создать выставку" class="btn btn-primary" />

</div>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Вернуться к выставкам</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

# Код Exhibition/Detail:

@model Virtual\_Art\_Gallery.Models.ExhibitionModel

@{

ViewData["Title"] = "О выставке";

}

<h1>@Html.DisplayFor(model => model.Title)</h1>

<div>

<hr />

<dl class="row">

<dt class = "col-sm-2">

Название

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Title)

</dd>

<dt class = "col-sm-2">

Описание

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.Description)

</dd>

<dt class = "col-sm-2">

Открытость

</dt>

<dd class = "col-sm-10">

@Html.DisplayFor(model => model.IsClosed)

</dd>

</dl>

</div>

<div>

<**a** **asp-action**="Edit" **asp-route-id**="@Model?.Id">Изменить</**a**> |

<**a** **asp-action**="Index">Вернуться к выставкам</**a**>

</div>

# Код Exhibition/Delete:

@model Virtual\_Art\_Gallery.Models.ExhibitionModel

@{

ViewData["Title"] = "Удаление выставки";

}

<h1>Удаление</h1>

<h3>Вы уверены, что хотите удалить выставку?</h3>

<div>

<hr />

<dl class="row">

<dt class="col-sm-2">

Название

</dt>

<dd class="col-sm-10">

@Html.DisplayFor(model => model.Title)

</dd>

<dt class="col-sm-2">

Описание

</dt>

<dd class="col-sm-10">

@Html.DisplayFor(model => model.Description)

</dd>

<dt class="col-sm-2">

Открытость

</dt>

<dd class="col-sm-10">

@Html.DisplayFor(model => model.IsClosed)

</dd>

</dl>

<**form** **asp-action**="Delete">

<**input** **type**="hidden" **asp-for**="Id" />

<input type="submit" value="Удалить" class="btn btn-danger" /> |

<**a** **asp-action**="Index">Отмена</**a**>

</**form**>

</div>

# Код Exhibition/Edit:

@model Virtual\_Art\_Gallery.Models.ExhibitionModel

@{

ViewData["Title"] = "Редактирование выставки";

}

<h1>Редактирование выставки</h1>

<hr />

<div class="row">

<div class="col-md-4">

<**form** **asp-action**="Edit">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<**input** **type**="hidden" **asp-for**="Id" />

<div class="form-group">

<**label** **asp-for**="Title" class="control-label">Название выставки</**label**>

<**input** **asp-for**="Title" class="form-control" />

<**span** **asp-validation-for**="Title" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="Description" class="control-label">Описание выставки</**label**>

<**input** **asp-for**="Description" class="form-control" />

<**span** **asp-validation-for**="Description" class="text-danger"></**span**>

</div>

<div class="form-group form-check">

<label class="form-check-label">

<**input** class="form-check-input" **asp-for**="IsClosed" /> Разрешить другим пользователям добавлять работы в эту выставку

</label>

</div>

<div class="form-group">

<input type="submit" value="Сохранить изменения" class="btn btn-primary" />

</div>

</**form**>

</div>

</div>

<div>

<**a** **asp-action**="Index">Вернуться к выставкам</**a**>

</div>

@section Scripts {

@{await Html.RenderPartialAsync("\_ValidationScriptsPartial");}

}

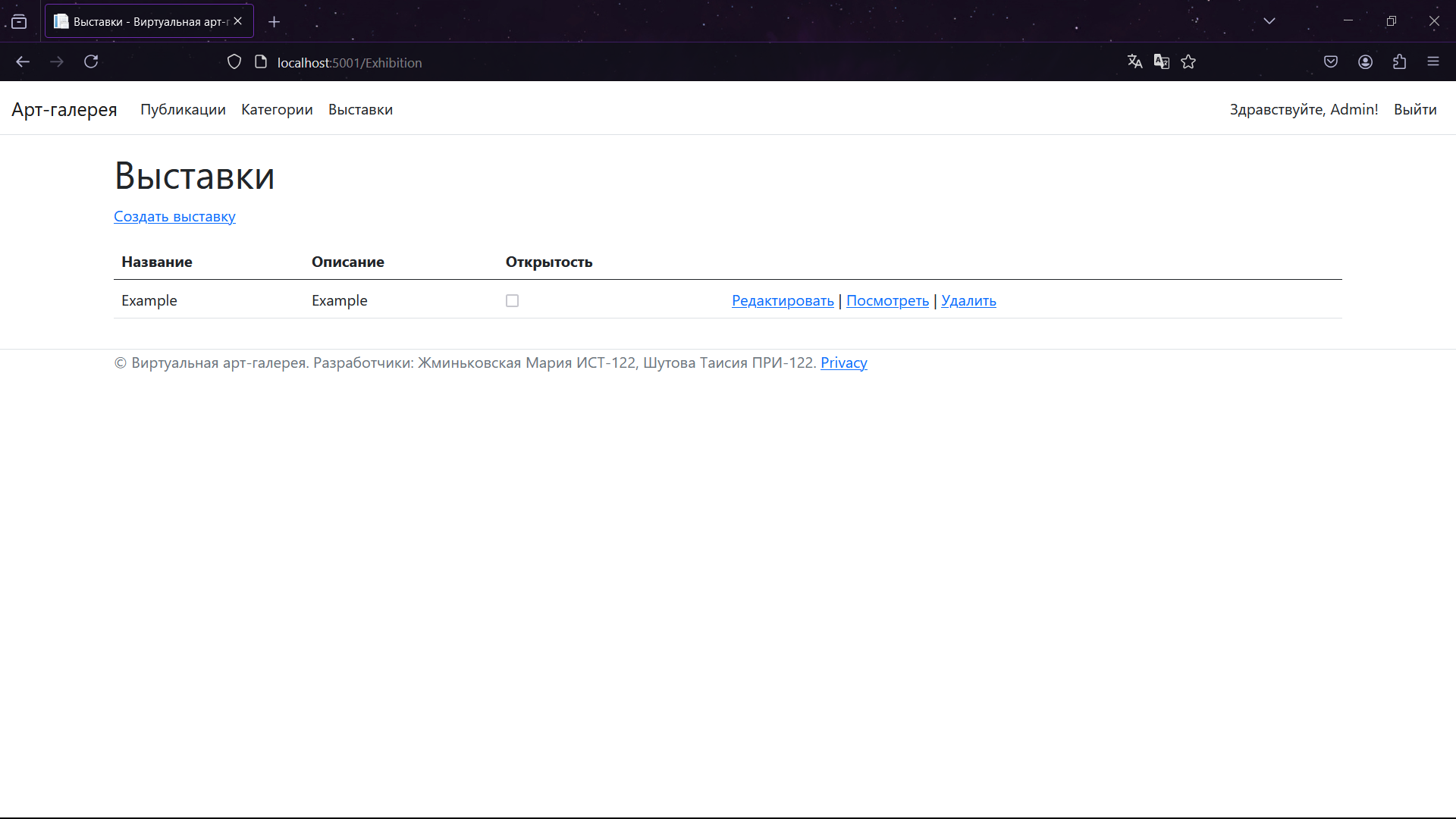


Рисунок 6. Exhibition/Index

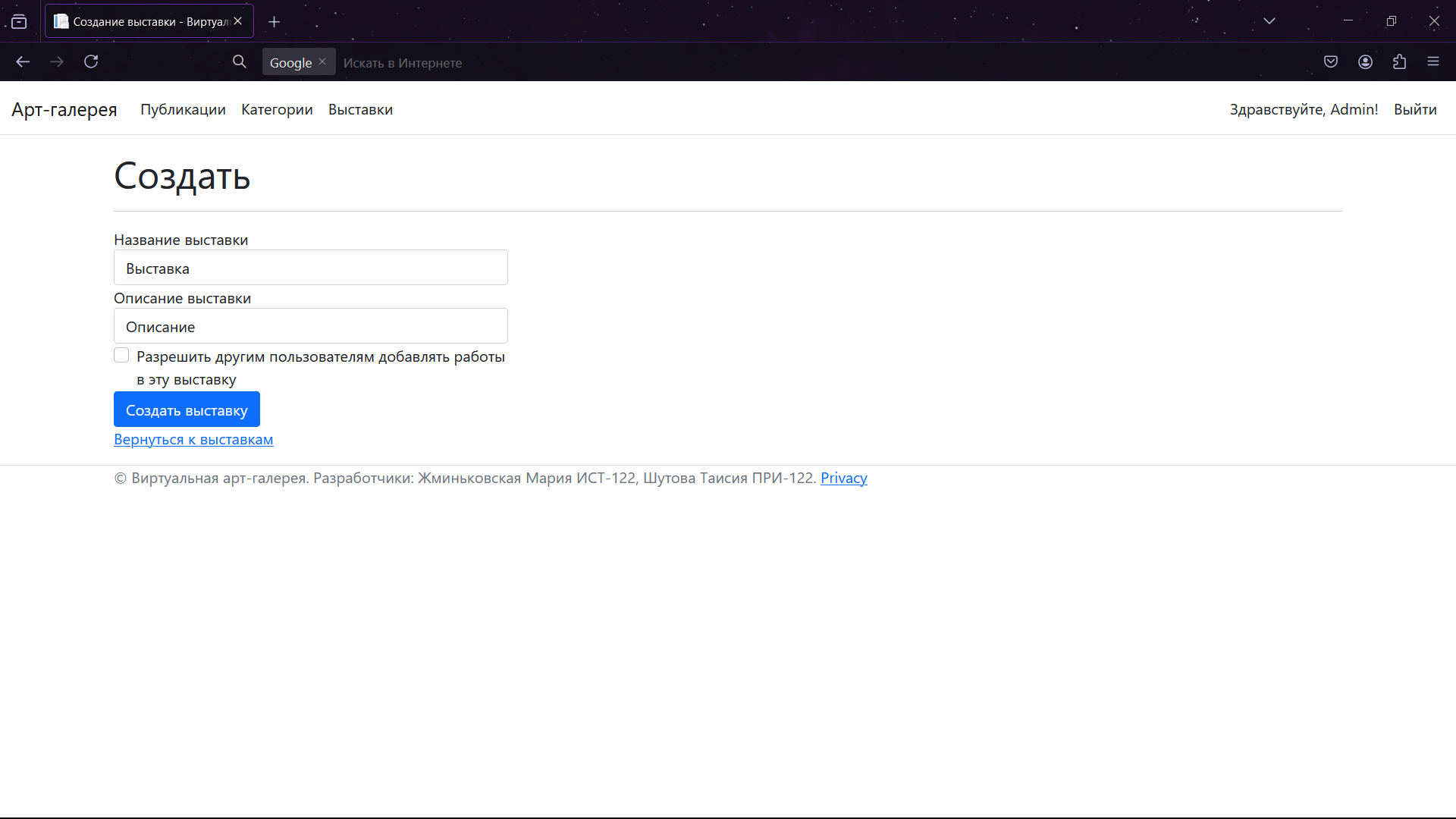


Рисунок 7. Exhibition/Create

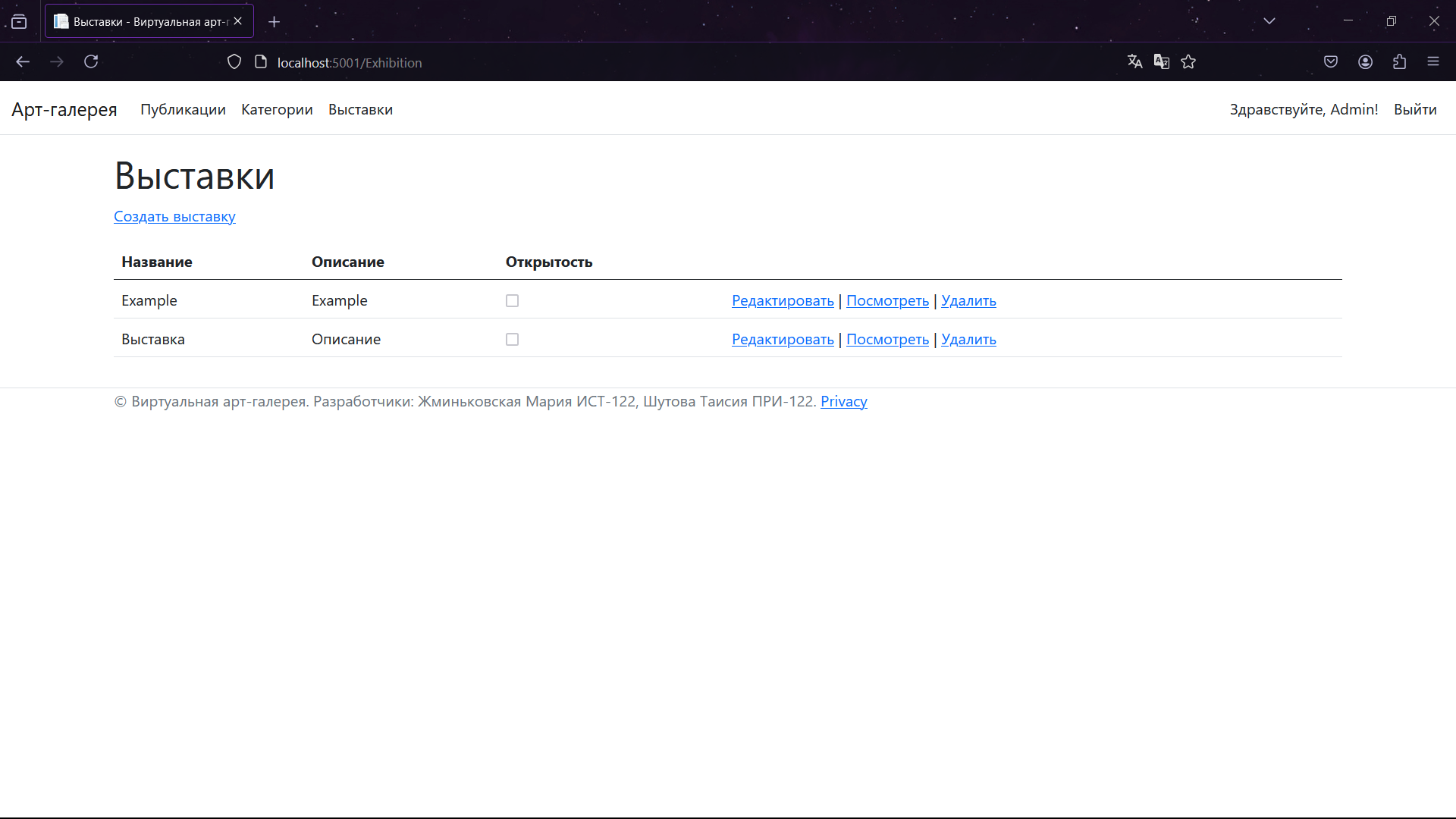


Рисунок 8. Exhibition/Index (обновлён)

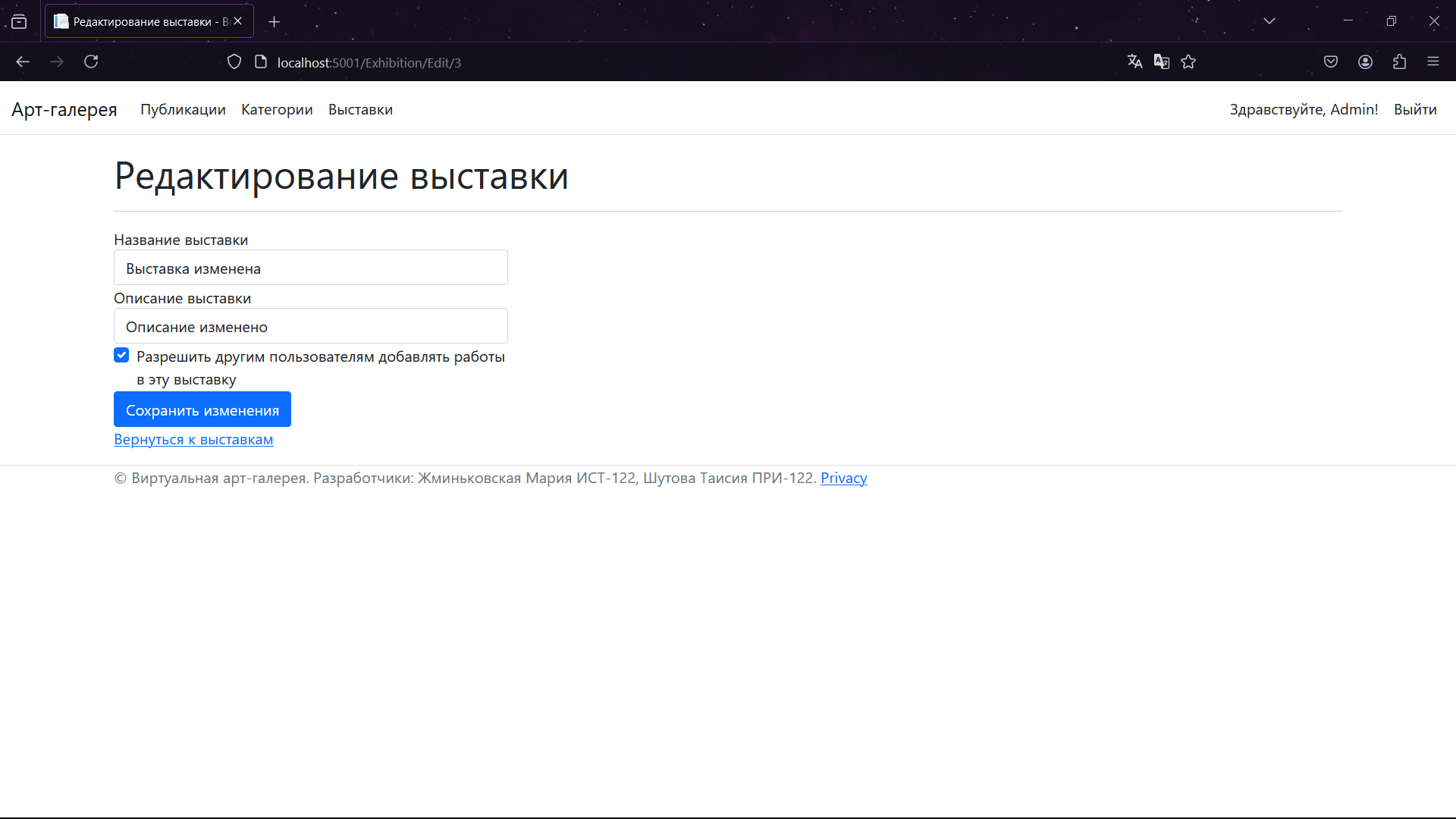


Рисунок 9. Exhibition/Edit

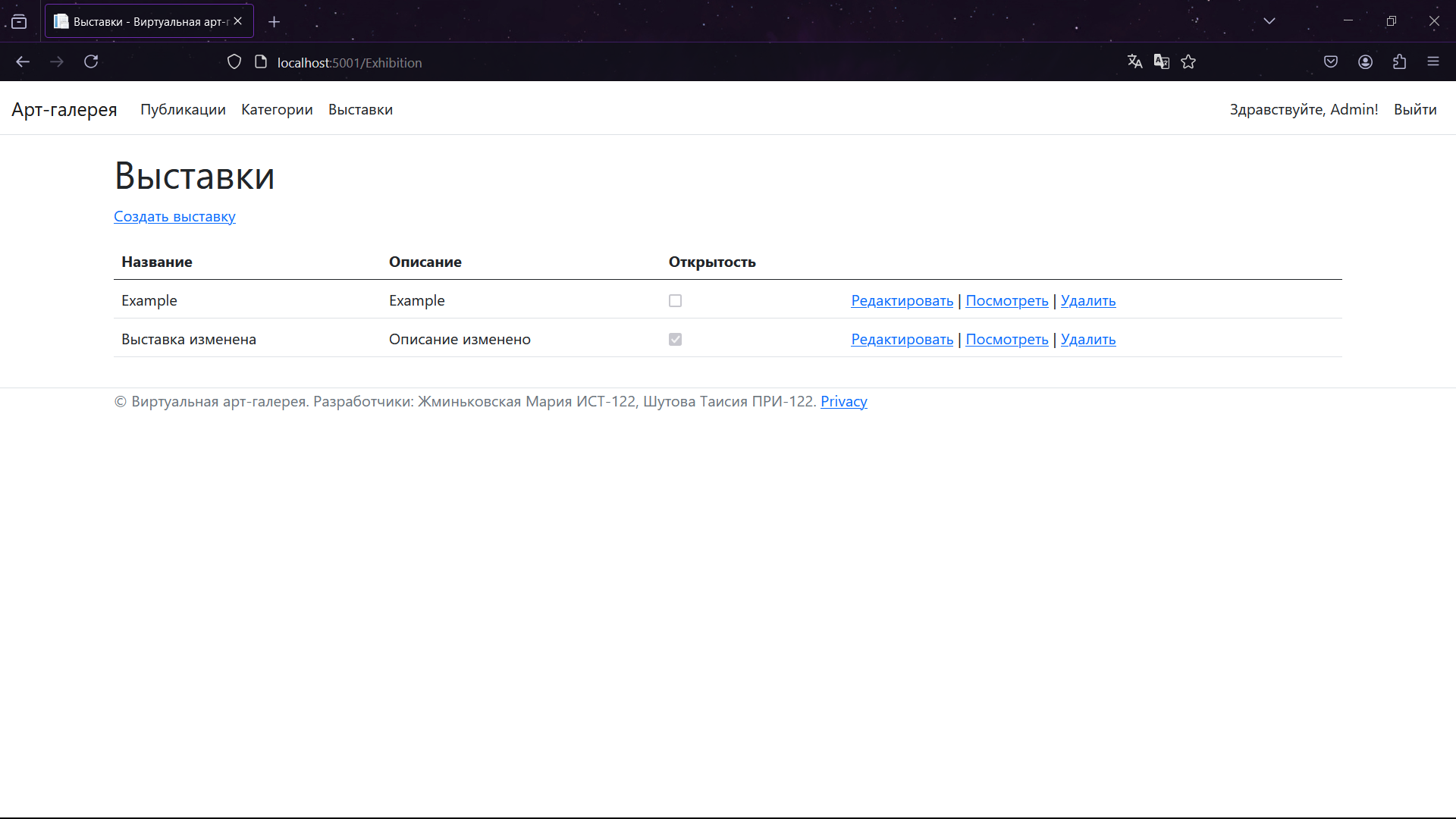


Рисунок 10. Exhibition/Index (обновлён)

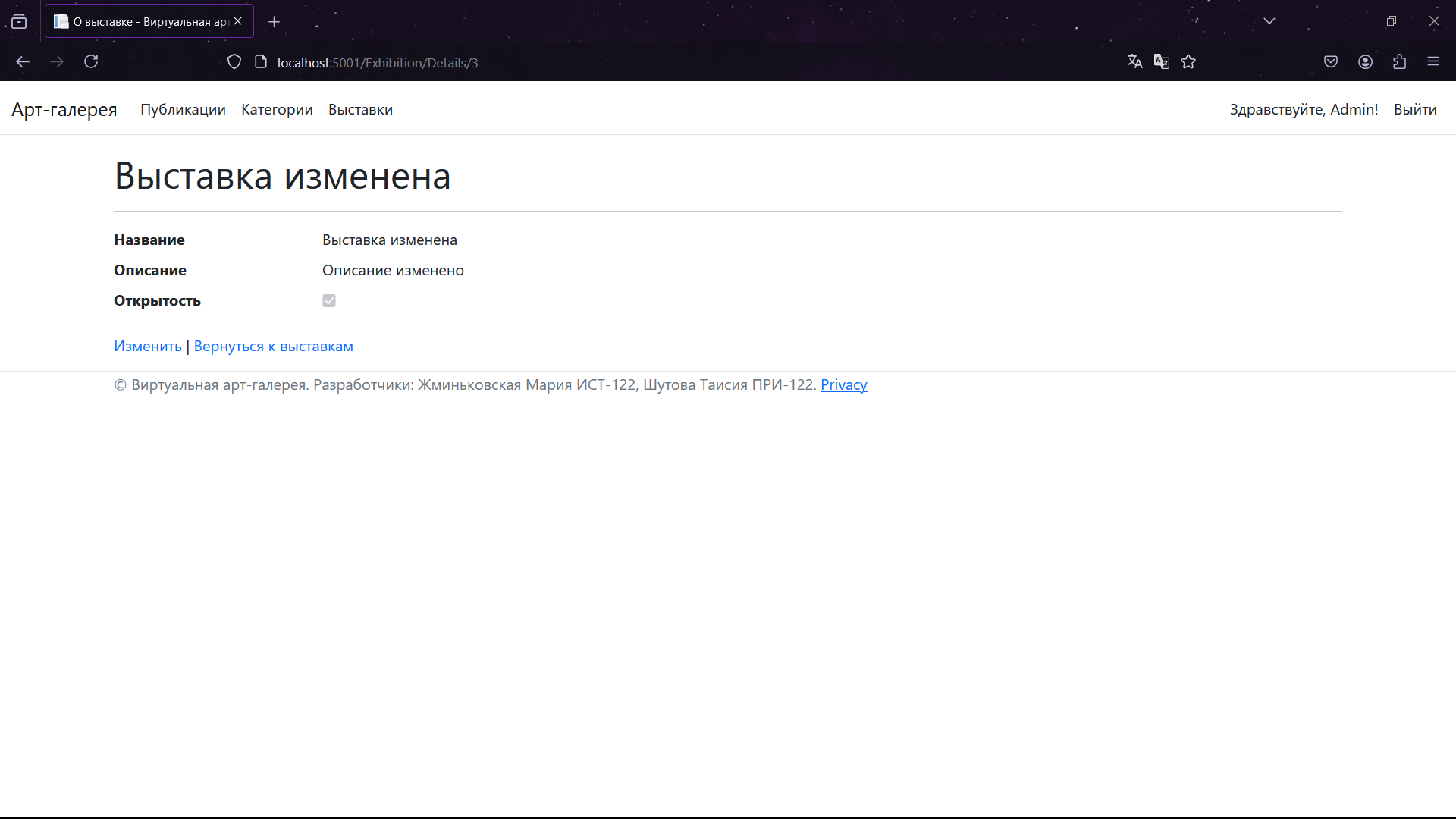


Рисунок 11. Exhibition/Edit

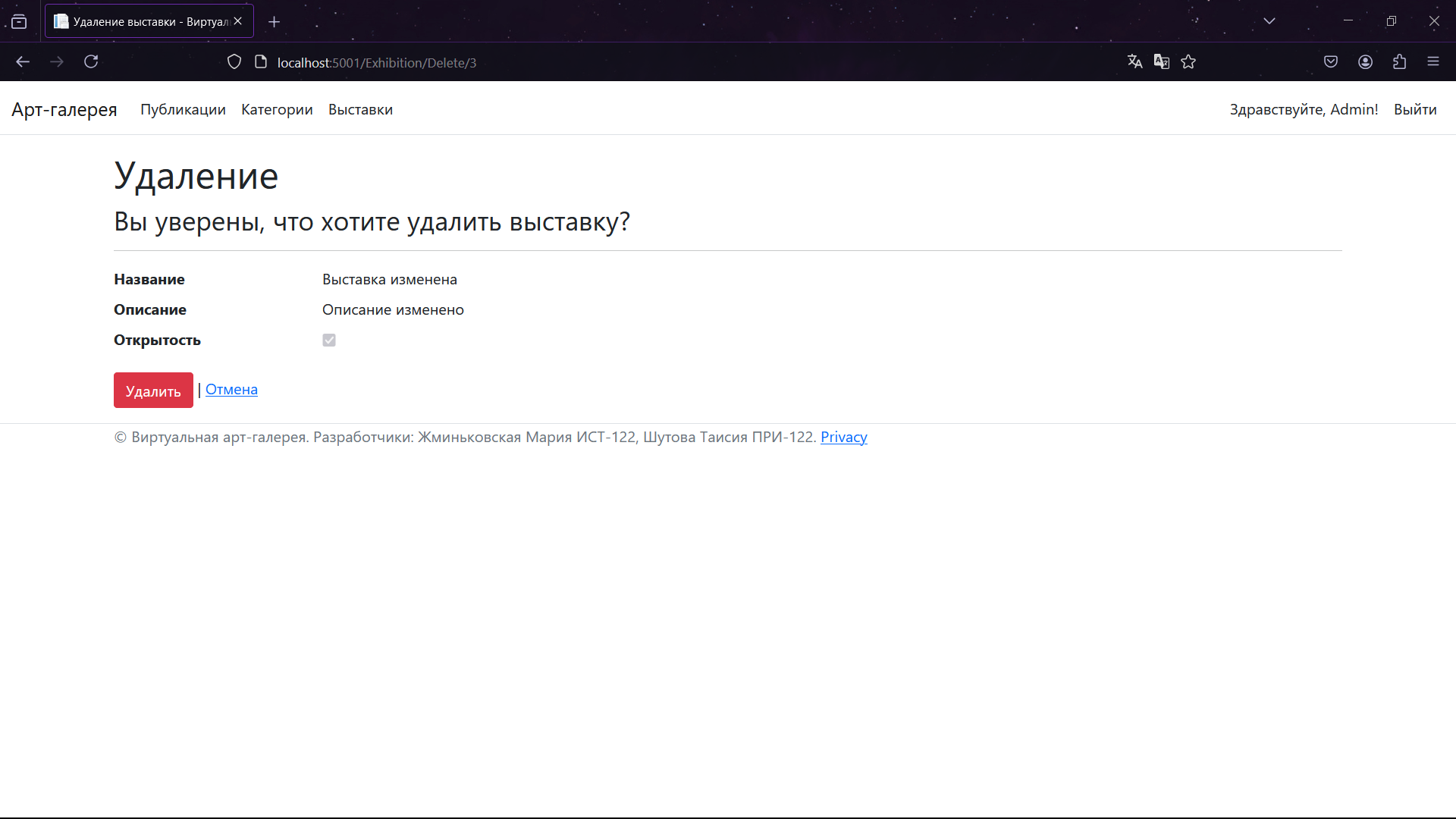


Рисунок 12. Exhibition/Delete

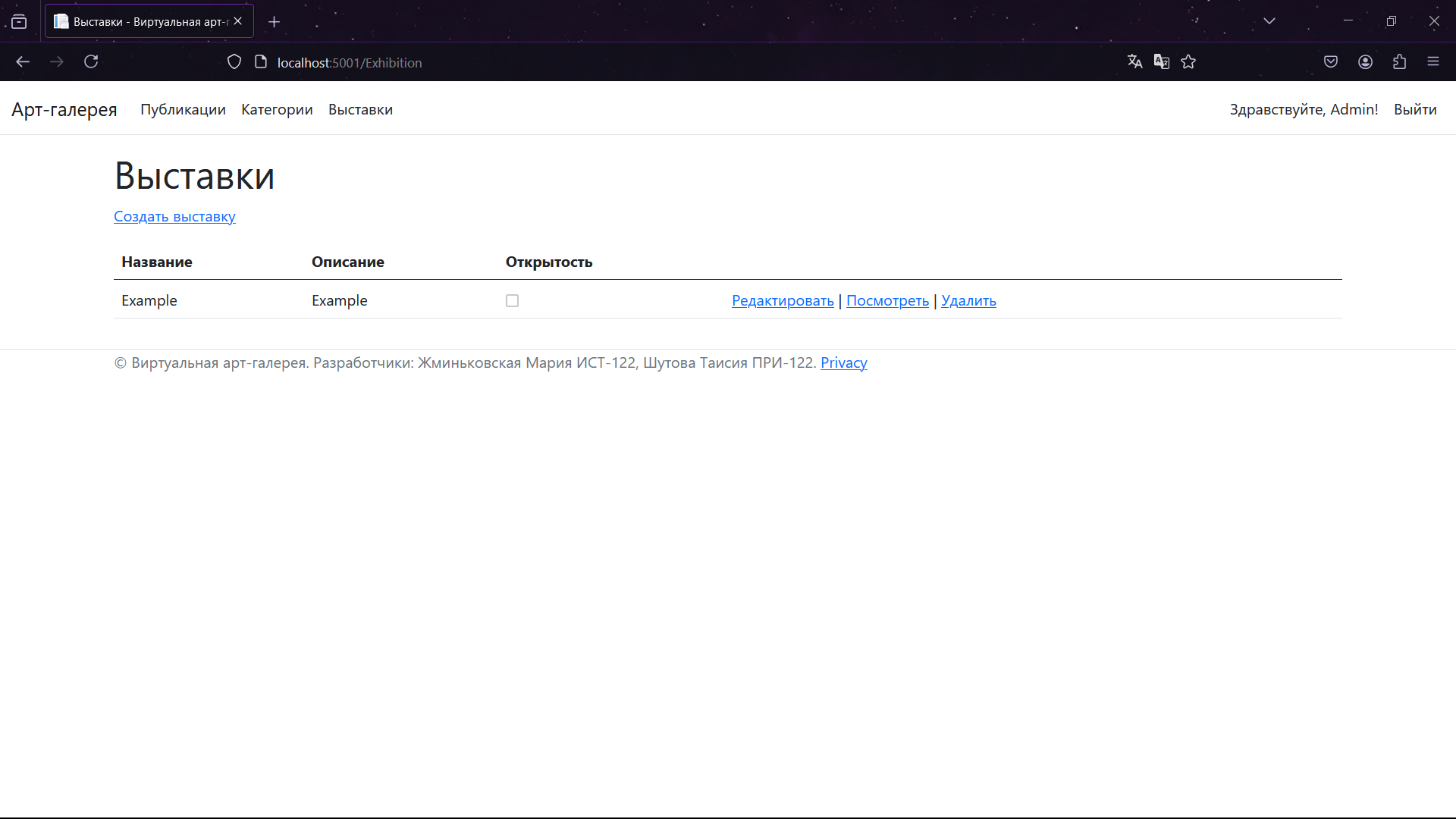


Рисунок 13. Exhibition/Index (обновлён)