

Техническое задание на разработку сайта

«Каталог игр»

1. Общие сведения

1.1. Наименование проекта

Веб-приложение «Каталог игр» (далее — Система).

1.2. Назначение проекта

Система предназначена для ведения каталога компьютерных игр с возможностью фильтрации, поиска и управления данными об играх и жанрах.

1.3. Технологический стек

Компонент	Технология
Backend	PHP 8.x, Laravel 10/11
Frontend	Blade, Bootstrap 5 / Tailwind CSS
База данных	MySQL / PostgreSQL
Аутентификация	Laravel Breeze / Laravel UI
Среда разработки	Laravel Sail / OpenServer / Docker

2. Функциональные требования

2.1. Модели данных и отношения

2.1.1. Общие требования к моделям

Модель «Жанр» (Genre)

Поле	Тип	Описание
id	integer (AI)	Первичный ключ
name	string(100), unique	Название жанра

Поле	Тип	Описание
description	text, nullable	Описание жанра
created_at	timestamp	Дата создания
updated_at	timestamp	Дата обновления

Модель «Игра» (Game)

Поле	Тип	Описание
id	integer (AI)	Первичный ключ
title	string(200)	Название игры
developer	string(150)	Разработчик
poster	string(255), nullable	Путь к постеру
release_date	date	Дата выхода
rating	decimal(3,1)	Рейтинг (0.0 – 10.0)
description	text, nullable	Краткое описание
created_at	timestamp	Дата создания
updated_at	timestamp	Дата обновления

2.1.2. Вариант 1 (на оценку 4) — связь «один ко многим»

text
Жанр (1) ----< (M) Игра

- Одна игра может принадлежать **только одному** жанру.
- Один жанр может содержать **множество** игр.

Дополнительное поле в таблице games:

```
sql
ALTER TABLE games ADD COLUMN genre_id INT;
ALTER TABLE games ADD FOREIGN KEY (genre_id) REFERENCES genres(id) ON DELETE SET NULL;
```

Отношения в коде:

php

```
// Модель Genre
public function games() {
    return $this->hasMany(Game::class);
}

// Модель Game
public function genre() {
    return $this->belongsTo(Genre::class);
}
```

2.1.3. Вариант 2 (на оценку 5) — связь «многие ко многим»

text

Жанр <----< (genre_game) >---- Игра

- Одна игра может принадлежать **нескольким** жанрам.
- Один жанр может содержать **множество** игр.

Дополнительная pivot-таблица:

```
sql
CREATE TABLE genre_game (
    id INT AUTO_INCREMENT PRIMARY KEY,
    genre_id INT NOT NULL,
    game_id INT NOT NULL,
    FOREIGN KEY (genre_id) REFERENCES genres(id) ON DELETE CASCADE,
    FOREIGN KEY (game_id) REFERENCES games(id) ON DELETE CASCADE
);
```

Отношения в коде:

```
php
// Модель Genre
public function games() {
    return $this->belongsToMany(Game::class);
}

// Модель Game
public function genres() {
    return $this->belongsToMany(Genre::class);
}
```

2.2. Требования к интерфейсу

2.2.1. Главная страница

Элементы главной страницы:

1. **Шапка сайта (navbar):**
 - Логотип / название сайта (ссылка на главную)
 - Ссылка «Добавить игру»

2. **Блок фильтрации** (располагается над списком игр):
 - Фильтр по жанру (выпадающий список из всех жанров)
 - Поиск по названию (поле ввода с текстовым поиском)
 - Фильтр по году выпуска (выпадающий список с годами)
 - Фильтр по рейтингу (выпадающий список: 0+, 5+, 7+, 8+, 9+)
3. **Блок «Список игр»** (сетка карточек):
 - Карточка игры содержит:
 - Постер (изображение)
 - Название игры
 - Рейтинг (звёздочки или число)
 - Карточка является **ссылкой** на страницу детального просмотра игры.
 - При отсутствии постера — отображается заглушка.
4. **Пагинация** (по 6–12 игр на странице).
5. **Сообщение** при отсутствии результатов: «Игры не найдены».

2.2.2. Страница детального просмотра игры

Содержит:

- Постер (крупное изображение)
- Название игры
- Разработчик
- Жанр(ы) (для варианта 1 — один жанр, для варианта 2 — список жанров)
- Дата выхода (год выпуска)
- Рейтинг (число и визуальное отображение)
- Краткое описание
- **Кнопки управления (только для авторизованных):**
 - «Редактировать» (ведёт на страницу редактирования)
 - «Удалить» (с подтверждением)

2.2.3. Страница добавления игры (только для авторизованных)

Форма добавления содержит поля:

Поле	Тип	Обязательно	Примечание
Название	text	✓	минимум 2 символа
Жанр(ы)	select / select multiple	✓	вариант 1 — 单选, вариант 2 — мультивыбор
Разработчик	text	✓	минимум 2 символа
Постер	file (jpg, png, webp)	✗	макс. 5 МБ
Дата выхода	date	✓	не может быть в будущем

Поле	Тип	Обязательно	Примечание
Рейтинг	number (step=0.1)	✓	от 0 до 10
Краткое описание	textarea	✗	макс. 1000 символов

Требования:

- Валидация всех полей на серверной стороне.
- После успешного добавления — редирект на страницу просмотра игры с сообщением об успехе.

2.2.4. Страница редактирования игры (только для авторизованных)

- Форма аналогична странице добавления, но с **предзаполненными** текущими значениями.
- После успешного обновления — редирект на страницу просмотра игры с сообщением об успехе.

2.2.5. Удаление игры

- Доступно со страницы просмотра игры.
- При нажатии на кнопку «Удалить» — запрос подтверждения (JavaScript confirm или модальное окно).
- После удаления — редирект на главную страницу с сообщением об успехе.

2.3. Требования к фильтрации

Фильтр	Тип	Примечание
По жанру	Выпадающий список	Значения из таблицы genres. Для М:М — поиск игр, содержащих выбранный жанр
Поиск по названию	Текстовое поле	Частичное совпадение (LIKE %...%), регистронезависимо
По году выпуска	Выпадающий список	Уникальные годы из существующих игр
По рейтингу	Выпадающий список	Варианты: «Любой», «≥ 5», «≥ 7», «≥ 8», «≥ 9»

Пример фильтра по рейтингу:

- «≥ 8» — показывать игры с рейтингом от 8.0 и выше

3. Требования к данным

3.1. Хранение постеров

- Изображения загружаются в директорию `storage/app/public/posters/`
- Для доступа через веб создаётся символическая ссылка (для загрузки файлов ввести в консоли: `php artisan storage:link`)

Подсказка как загрузить постер:

```
$file = $request->file('poster'); // запись объекта файла в переменную
$posterPath = $file->store('public');
$data['poster'] = basename($posterPath);
```

```
Game::create($data);
```

- Поддерживаемые форматы: JPG, JPEG, PNG, WEBP
- Максимальный размер файла: 5 МБ
- Постер по умолчанию: `/images/no-poster.jpg`

3.2. Заполнение базы данных (Seeders)

При развёртывании проекта должны быть сиды для:

Жанры (минимум 8):

- Action, RPG, Strategy, Adventure, Simulation, Sports, Puzzle, Horror

Игры (минимум 15):

- Разные годы выпуска (1990–2025)
- Разные рейтинги (2.0 – 9.8)
- Разные разработчики (Nintendo, Valve, CD Projekt, Ubisoft и др.)

4. Архитектурные требования

4.1. Маршруты (routes/web.php)

HTTP метод	URI	Назначение	Имя маршрута	Доступ
GET	/	Главная страница (список игр)	games.index	Все
GET	/games/{id}	Просмотр игры	games.show	Все
GET	/games/create	Форма добавления	games.create	Все
POST	/games	Сохранение игры	games.store	Все
GET	/games/{id}/edit	Форма редактирования	games.edit	Все
PUT/PATCH	/games/{id}	Обновление игры	games.update	Все
DELETE	/games/{id}	Удаление игры	games.destroy	Все

4.2. Модели и контроллеры

Модели:

- Genre (жанр)
- Game (игра)

Контроллеры:

- GameController (ресурсный контроллер со всеми методами CRUD)
- GenreController (опционально, для управления жанрами)

4.3. Валидация

Правила для GameRequest:

```
php
public function rules()
{
    return [
        'title' => 'required|min:2|max:200',
        'developer' => 'required|min:2|max:150',
        'release_date' => 'required|date|before_or_equal:today',
        'rating' => 'required|numeric|min:0|max:10',
        'description' => 'nullable|max:1000',
        'poster' => 'nullable|image|mimes:jpg,jpeg,png,webp|max:5120',
    ];
}
```

5. Критерии оценки

5.1. Базовый функционал (оценка 4)

№	Критерий
1	Реализованы модели «Жанр» и «Игра» со связью 1:M
2	Вывод всех игр на главной странице в виде карточек
3	Карточка является ссылкой на страницу игры
4	Страница детального просмотра игры со всей информацией
5	Фильтрация по жанру (1:M)
6	Поиск по названию
7	Фильтр по году выпуска
8	Фильтр по рейтингу
9	Добавление игры (форма + валидация + загрузка постера)
10	Редактирование игры (форма + предзаполнение)
11	Удаление игры с подтверждением
12	Аутентификация (регистрация/вход/выход)
13	Защита CRUD-операций middleware('auth')
14	Наличие seeders для демонстрации
15	Адаптивная вёрстка (Bootstrap/Tailwind)

Итого

5.2. Дополнительный функционал для оценки 5

№	Критерий
16	Реализована связь «многие ко многим» между играми и жанрами

№	Критерий
17	В форме добавления/редактирования — выбор нескольких жанров (select multiple / чекбоксы)
18	На странице игры отображаются все жанры
19	Фильтрация по жанру работает корректно для М:М
20	Дополнительный фильтр (например, по разработчику или алфавиту)
21	Пагинация с сохранением параметров фильтрации
Итого с бонусами	

5.3. Шкала перевода в оценку

Процент	Оценка
85–100%	5 (отлично)
70–84%	4 (хорошо)
50–69%	3 (удовлетворительно)
< 50%	2 (неудовлетворительно)

6. Дополнительные рекомендации

1. **Пагинация:** Использовать `->paginate(9)` на главной странице.
2. **Сохранение фильтров:** При пагинации параметры должны сохраняться в URL (`?genre=1&search=portal&year=2011&rating=8`).
3. **Сообщения:** Использовать Laravel Flash-сообщения (`session()->flash('success')`).
4. **Рейтинг:** Можно отображать звёздочками (CSS-звёзды или иконки).
5. **Валидация:** Вынести в `FormRequest` для чистоты контроллера.
6. **Постер:** Если не загружен — **ВЫВОДИТЬ** `/images/no-poster.jpg`.

7. Пример сценария использования

1. Пользователь открывает сайт → видит 9 карточек игр
2. Выбирает жанр «RPG» → отображаются только RPG-игры
3. Вводит в поиск «Witcher» → находит игры с этим словом в названии
4. Нажимает на карточку «The Witcher 3» → переходит на страницу игры
5. Видит полную информацию: описание, разработчик CD Projekt, рейтинг 9.5, жанры
6. Нажимает «Добавить игру» → заполняет форму → сохраняет
7. Новая игра появляется на главной странице
8. На странице игры нажимает «Редактировать» → меняет рейтинг
9. Нажимает «Удалить» → подтверждает → игра исчезает