МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики и кибернетики

Кафедра программных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**

по дисциплине «Объектная распределенная обработка»

по теме «Игра пазл»

Обучающийся Д.О. Колбанов

Руководитель О.А. Гордеева

Самара 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева (Самарский университет)»

Институт информатики и кибернетики

Кафедра программных систем

ЗАДАНИЕ

на курсовую работу по дисциплине

«Технологии сети Интернет» обучающимся в группе № 6401-020302D Д.О. Колбанову

1. Тема проекта: «Веб-приложение каталог инди-игр»
2. Перечень вопросов, подлежащих разработке:
   1. Произвести анализ предметной области: изучить сайты–аналоги приложения, определить целевую аудиторию
   2. Выполнить обзор существующих систем-аналогов
   3. Подготовить информационное обеспечение сайта (текст, изображения)
   4. Разработать прототипы экранных форм
   5. Разработать и реализовать программное и информационное обеспечение, провести его тестирование и отладку
   6. Оформить документацию курсовой работы
3. Перечень графических разработок:
   1. Логическая схема (внутренняя и внешняя)
   2. Функциональная схема
4. Календарный план выполнения работ Задание принял

к исполнению Д.О. Колбанов

ПРИЛОЖЕНИЕ

к заданию на курсовую работу обучающимся в группе № 6401-020302D Д.О. Колбанову

Тема проекта: «Веб-приложение каталог инди-игр»

Исходные данные к проекту:

1. Характеристика объекта автоматизации:
2. объект автоматизации: веб-приложение игра «Пазл»;
3. виды автоматизируемой деятельности:
   * процесс авторизации/регистрации пользователей;
   * процесс просмотра игр, разделённых по категориям;
   * процесс игры и оценки игр;
   * процесс ведения базы данных.
4. количество ролей пользователей – 2;
5. минимальная длина пароля – 8 символов;
6. максимальная длина пароля – 30 символов;
7. минимальная длина логина – 1 символ;
8. максимальная длина логина – 30 символов;
9. минимальное количество категорий – 1;
10. минимальная длина названия игры – 1 символ;
11. максимальная длина названия игры – 50 символов;
12. минимальная длина описания игры – 5 символов;
13. максимальная длина описания игры – 500 символов;
14. минимальная длина текста имени автора – 1 символ;
15. максимальная длина текста имени автора – 50 символов.
16. Требования к информационному обеспечению:
17. структура базы данных (БД) разрабатывается на основании следующих сведений:
    * о песнях (название, текст, жанр, автор песни, автор записи на сайте, количество просмотров, дата добавления);
    * об игре (название, описание, категория, разработчик, картинка, ссылка на игру, проголосовавшие пользователи, количество лайков);
    * о пользователях (логин, почта, пароль);
    * о категориях (название).;
    * должна быть обеспечена целостность базы данных и защита от несанкционированного доступа.
18. Требования к техническому обеспечению:
    1. Требования к техническому обеспечению серверной части:
       1. тип ЭВМ – IBM PC совместимый;
       2. объем ОЗУ – не менее 2 Гб;
       3. объем свободного пространства на внешнем диске – не менее 50 Гб;
       4. наличие подключения к сети Интернет;
       5. манипулятор – мышь;
       6. технические характеристики определяются в процессе выполнения проекта.
    2. Требования к техническому обеспечению клиентской части:
       1. тип ЭВМ – IBM PC совместимый;
19. монитор с разрешающей способностью не ниже 800 х 600;
20. манипулятор – мышь, клавиатура;
21. технические характеристики определяются в процессе выполнения проекта.
22. Требования к программному обеспечению:
    1. Требования к программному обеспечению серверной части:
       1. тип операционной системы – Windows 7 и выше;
       2. СУБД – MongoDB;
    2. Требования к программному обеспечению клиентской части:
       1. тип операционной системы – Windows 7 и выше;
       2. браузер – Google Chrome 86.0.4240.183 (64-битный) и выше.
    3. Требования к программному обеспечению рабочего места разработчика:

* СУБД – MongoDB
* языки программирования –JavaScript (React, Express), SQL;
* среда программирования – Visual Studio Code;
* среда проектирования – StarUML.

1. Общие требования к проектируемой системе:
   1. Функции, реализуемые системой:
      1. функции системы:
         * аутентификация пользователя в системе, назначение ролей и настройка прав;
         * визуализация подборок игр;
         * обеспечение возможности работы с играми.
      2. функции пользователя:
         * регистрация пользователя в системе (ввод логина и пароля);
         * авторизация пользователя в системе (ввод логина и пароля);
         * просмотр игр;
         * голосование за понравившиеся игры.
      3. функции администратора:
         * добавление сущности в БД;
         * редактирование сущности в БД;
         * удаление сущности из БД.
   2. Технические требования к системе:
      1. режим работы – диалоговый;
      2. система должна удовлетворять санитарным правилам и нормам СанПин 2.2.2./2.4.2198-07;
      3. условия работы средств вычислительной техники (содержание вредных веществ, пыли и подвижность воздуха) должны соответствовать ГОСТ 12.1.005, 12.01.007;
      4. температура окружающего воздуха – 15-35°С;
      5. влажность воздуха – 45-75%.

Руководитель

проекта М.В. Додонов

Задание принял

к исполнению Д.О. Колбанов

СОДЕРЖАНИЕ [ВВЕДЕНИЕ 8](#_bookmark0)

1. [Описание и анализ предметной области 9](#_bookmark1)
   1. [Основные понятия и определения 9](#_bookmark2)
   2. [Описание систем-аналогов 11](#_bookmark3)
      1. [Яндекс игры 11](#_bookmark4)
      2. [Itch 12](#_bookmark5)
      3. [Сравнительный анализ систем-аналогов 13](#_bookmark6)
   3. [Постановка задачи 14](#_bookmark7)
      1. [Требования к информационному обеспечению 14](#_bookmark8)
      2. [Требования к техническому обеспечению 14](#_bookmark9)
      3. [Требования к программному обеспечению 15](#_bookmark10)
2. [Проектирование системы 15](#_bookmark11)
   1. [Функциональная модель 15](#_bookmark12)
      1. [Перечень исключительных ситуаций 19](#_bookmark13)
      2. [Структурная схема системы 19](#_bookmark14)
      3. [Диаграмма вариантов использования 21](#_bookmark15)
   2. [Логическая схема данных 24](#_bookmark16)
   3. [Разработка прототипа интерфейса пользователя системы 25](#_bookmark17)
3. [Реализация веб-приложения 30](#_bookmark18)
   1. [Разработка физической модели БД 30](#_bookmark19)
   2. [Разработка серверной части системы 30](#_bookmark20)
   3. [Разработка клиентской части 32](#_bookmark21)
   4. [Тестирование системы 33](#_bookmark22)
4. [Выводы 36](#_bookmark23)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 37](#_bookmark24)

[Приложение А 38](#_bookmark25)

ВВЕДЕНИЕ

В современном мире цифровых технологий компьютерные игры занимают важное место, предоставляя пользователям не только развлечение, но и возможность развивать логическое мышление, внимание и стратегические навыки. Одной из таких игр является "Пазл", которая сочетает в себе простоту правил и увлекательный игровой процесс. Эта игра не только помогает расслабиться, но и стимулирует когнитивные способности, такие как память, концентрация и пространственное мышление [1].

"Пазл" – это классическая логическая игра, в которой игроку необходимо открывать парные ячейки на игровом поле, пытаясь найти совпадения. Если открытые ячейки содержат одинаковые изображения или цвета, они остаются открытыми, в противном случае – закрываются обратно. Игра завершается, когда все ячейки на поле успешно открыты. Простота правил и возможность настройки сложности (за счет изменения размера поля) делают эту игру универсальной и интересной для широкой аудитории [1].

Разработка системы будет производиться по технологии быстрой разработки приложений Rapid Application Development (RAD)*,* которая поддерживается методологией структурного проектирования и включает элементы объектно-ориентированного проектирования и анализа предметной области [2].

При проектировании системы будут использоваться методология Object-Oriented Analysis/Design (ООАП), в основу которой положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов, и язык моделирования UML (Unified Modeling Language), который является стандартным инструментом для разработки «чертежей» программного обеспечения.

1. Описание и анализ предметной области

Под предметной областью (application domain) принято понимать ту часть реального мира, которая имеет существенное значение или непосредственное отношение к процессу функционирования программы. Другими словами, предметная область включает в себя только те объекты и взаимосвязи между ними, которые необходимы для описания требований и условий решения некоторой задачи [2].

* 1. Основные понятия и определения

Игра «Мемори» (Пазл) или другими словами «Найди пару». Мемори – популярная настольная игра для детей дошкольного возраста. Несмотря на простоту конструкции и правил, она несет в себе серьезный потенциал для развития психики ребенка. Это увлекательное и доступное любым возрастам развлечение, которое отлично развивает память и внимание. Игра «мемори» – это карточная настольная игра, состоящая из парных картинок, где основной целью игры является «открытие» как можно большего числа парных карточек, за что игру «мемори» еще называют игра «парочки». Пример игры «Пазл» приведен на рисунке 1.

  
Рисунок 1 – Пример игры «Пазл»

Правила игры:

1. игровое поле состоит из четного числа ячеек, расположенных в виде сетки (например, 4×4 или 6×6 и так далее);
2. каждая ячейка содержит уникальный элемент (изображение/цвет), имеющий пару на поле;
3. игрок попарно открывает ячейки:
   * если элементы совпадают, ячейки остаются открытыми;
   * если элементы разные, ячейки закрываются через короткий промежуток времени;
4. игра завершается, когда все пары найдены.
   1. Постановка задачи

В рамках выполнения курсового проекта требуется разработать распределённое клиент-серверное приложение «Пазл» на базе технологии Java Servlets (JSF), обеспечивающее:

* генерацию игрового поля с четным количеством ячеек;
* интерактивное взаимодействие пользователя с полем через веб-интерфейс;
* проверку совпадения элементов и управление состоянием ячеек;
* сохранение статистики игровых сессий.

В системе будет 1 роль игрок. Пользователь, который занимается поиском пар карточек.

Сервер будет заниматься генерацией игрового поля с уникальными парами элементов, проверять корректность ходов (открытие двух ячеек за ход). Фиксировать время игры и количество попыток. Вести статистику по рейтингу среди пользователей (имя пользователя, количество очков) и личную статистику по игре (время выполнения, количество ходов, начисленные очки, дата выполнения). Проводить аутентификацию и авторизацию пользователей.

На клиентской части для начала игрок должен будет авторизоваться, введя логин (минимум 4 символа, максимум 30 символов) и пароль (минимум 4 символа, максимум 30 символов), либо зарегистрироваться, если у пользователя еще нет аккаунта, введя логин, пароль, повтор пароля. После авторизации у пользователя будет возможность начать игру. Перед началом игры пользователь должен задать размер поля путем ввода количества пар (минимум 5, максимум 25). На странице игры будет происходить отображение игрового поля с закрытыми и открытыми ячейками, обработка кликов пользователя и отправка запросов на сервер, отображение прогресса (количество оставшихся пар, время).

Также в системе должна быть обеспечена возможность получения справочной информации как о самой системе, так и предоставляемых ею возможностях, а еще будет возможность просмотреть сведения о разработчике.

Функции системы:

* регистрация и авторизация пользователей;
* настройка игрового поля путем указания количества пар в игре;
* генерация случайных элементов (цвет/изображения) для каждой сессии;
* валидация ходов и обновление состояния поля;
* сохранение результатов игры в базе данных;
* формирование рейтинга игроков по количеству очков.

Функции пользователя:

* создание аккаунта, вход в систему;
* запуск новой игры с выбором количества пар для отгадывания;
* взаимодействие с ячейками через клики;
* просмотр истории игры и личной статистики, просмотр рейтинга среди игроков;
* просмотр справочной информации о системе и о разработчике.
* отвечают за поддержку многопользовательской игры и обмен данных между игроками. Они позволяют устанавливать соединение с серверами, передавать информацию о состоянии игры и обрабатывать события, связанные с сетевым взаимодействием.

Все эти компоненты игрового движка работают вместе, чтобы обеспечить правильное функционирование и взаимодействие элементов игры.

Инди-игра – компьютерная игра, разработанная небольшим коллективом без финансовой поддержки издателя. Как правило из-за этого они отличаются небольшим масштабом и простотой.

Веб-игры – игры, запускающиеся в браузере и не требующие установки. В разрабатываемом веб-приложении пользователь сможет запускать веб-игры, которые будут работать на удалённом сервере и отображаться в браузере у пользователя.

* 1. Описание систем-аналогов

Существует несколько веб-приложений, позволяющих загружать и запускать веб-игры, и все они обладают своими особенностями, достоинствами и недостатками. Далее рассмотрим несколько систем аналогов.

* + 1. Яндекс игры

Яндекс игры – платформа, которая позволяет разработчикам загружать свои веб-игры, а пользователям играть в них и выставлять им рейтинг. Главная страница приведена на рисунке 1.

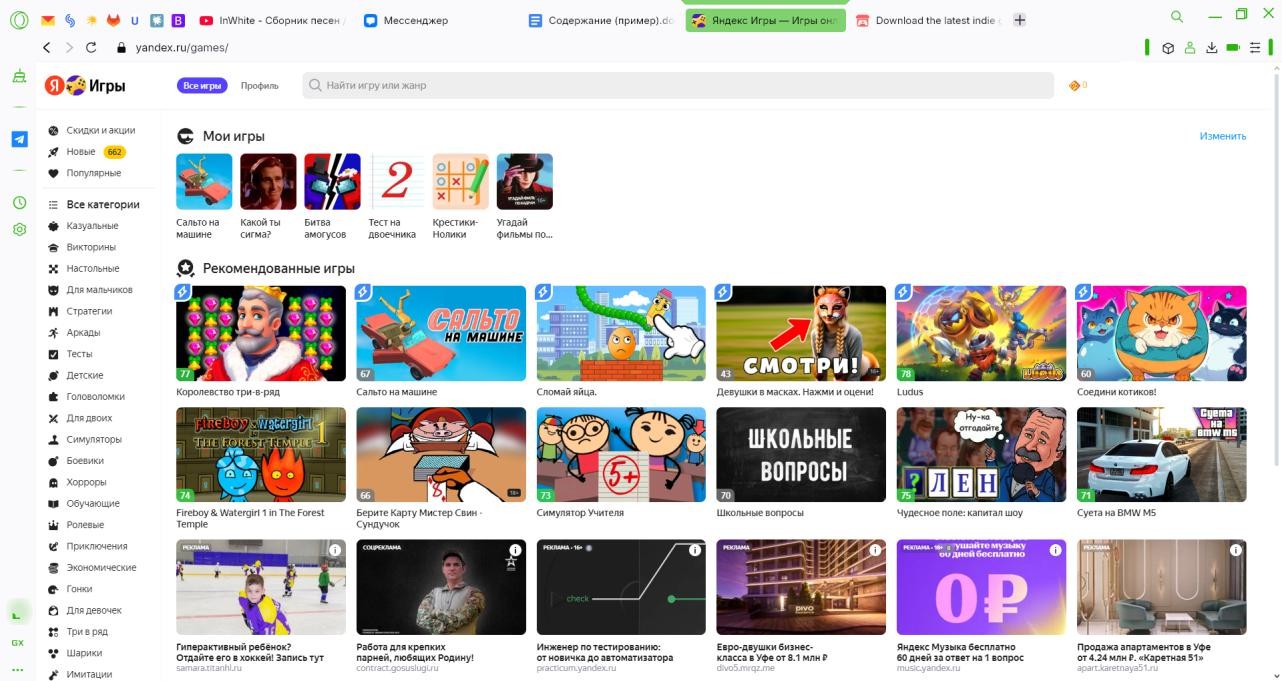


Рисунок 1 – Главная страница Яндекс игр

Как видно на рисунке, веб-приложение имеет большое количество жанров и категорий, оно простое в навигации и использовании. Также система запоминает игры, которые запускал пользователь и составляет его библиотеку. Однако из-за наличия рекламы, игровая область пользователя уменьшается, что является дискомфортным для игрока. Страница игры изображена на рисунке 2.

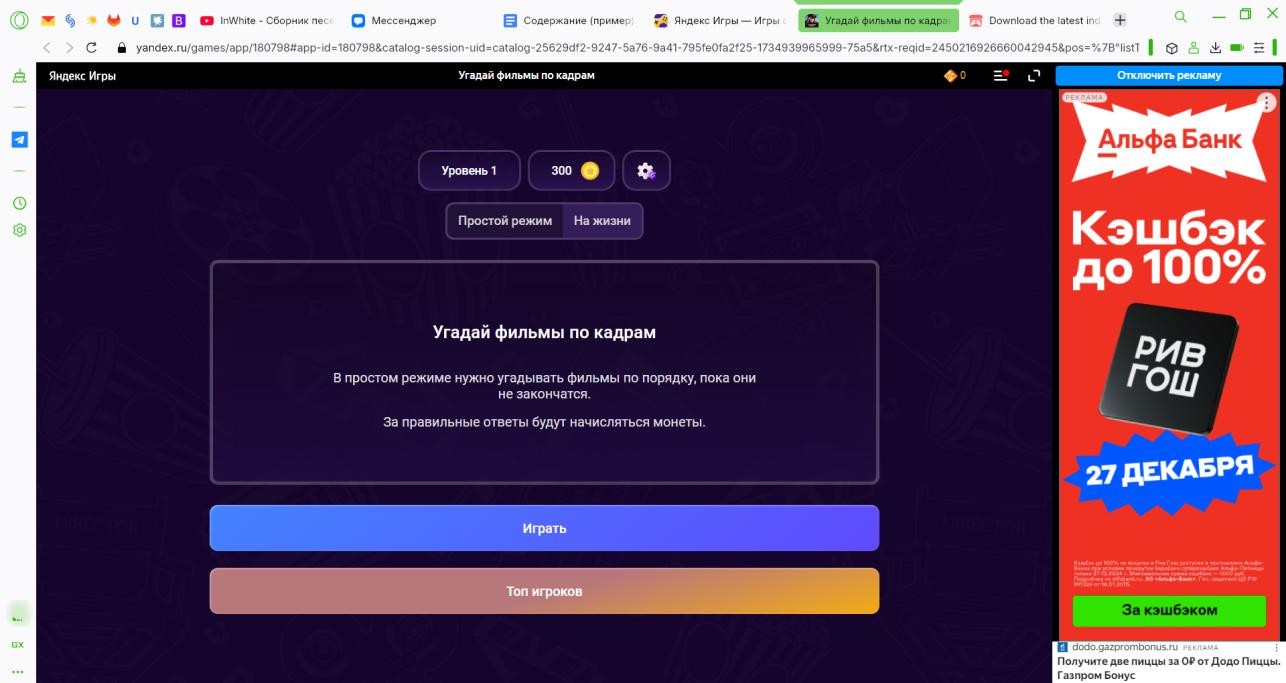


Рисунок 2 – Страница игры

Также к недостаткам системы можно отнести плохую модерацию, из-за которой система заполнена играми очень низкого качества, поэтому пользователю тяжело найти что-то качественное, исполненное на высоком уровне среди большого количества наспех собранных игр.

* + 1. Itch

Itch – это крупная зарубежная платформа, которая имеет большую функциональность и множество разделов. Главная страница изображена на рисунке 3. Как видно на рисунке, система хранит в себе не только игры, но и ассеты (ресурсы для разработки игр, например, звуки, картинки), книги. Itch также содержит раздел со статьями для общения пользователей, раздел с мероприятиями и турнирами для разработчиков. Также в веб-приложении размещаются не только веб-игры, но и десктопные игры с возможностью их покупки. Рассмотренная система является самой крупной системой- каталогом инди-игр, она обладает большим количеством преимуществ, но к недостаткам здесь можно отнести отсутствие русской локализации.

Страница игры изображена на рисунке 4.

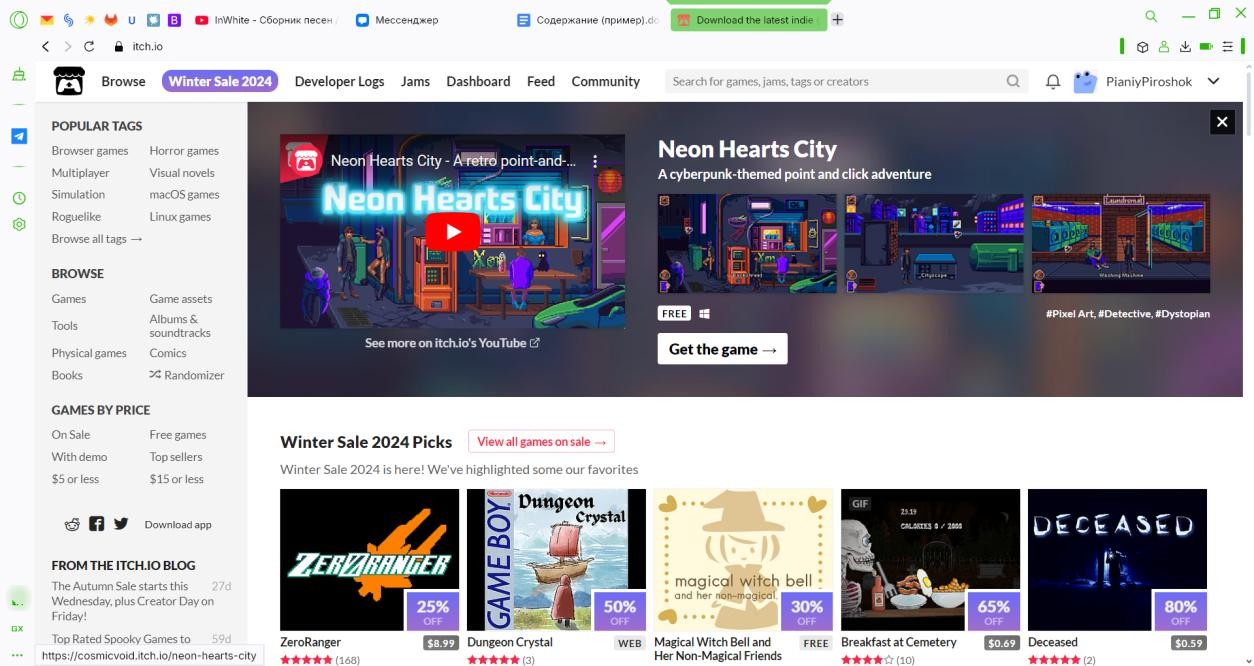


Рисунок 3 – Главная страница веб-приложения Itch

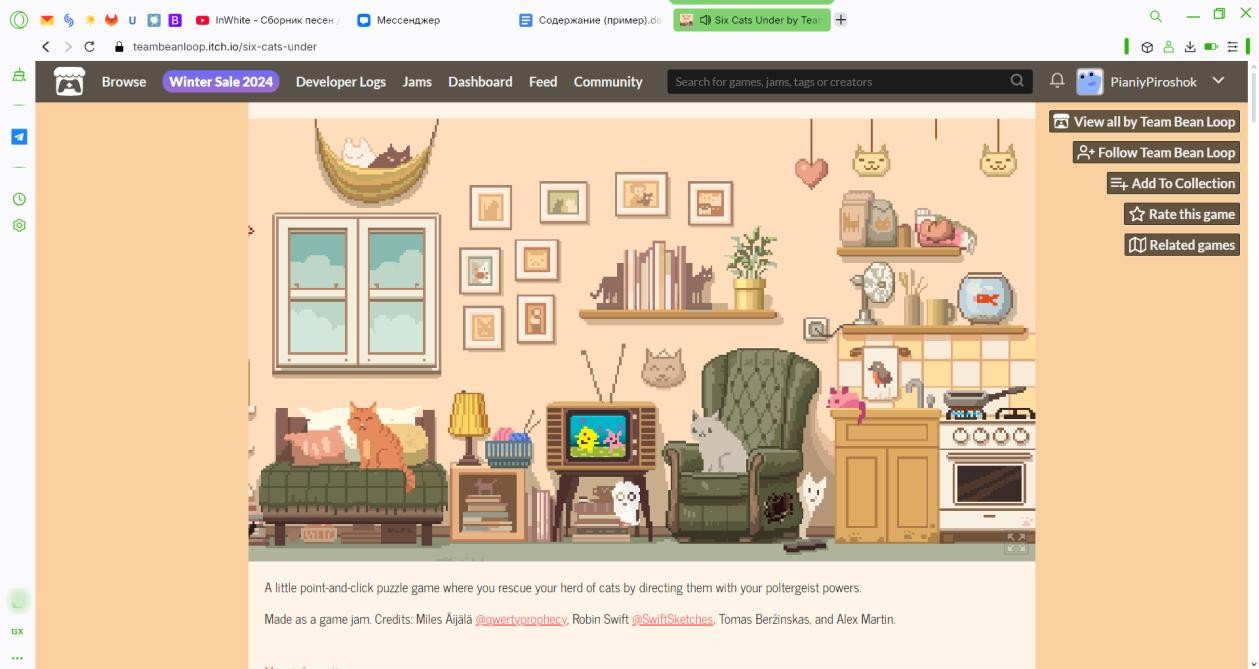


Рисунок 4 – Страница игры

* + 1. Сравнительный анализ систем-аналогов

В результате рассмотрения систем-аналогов были определены основные достоинства, которыми должна обладать разрабатываемая система. Так, она должна быть проста в навигации и удобна в использовании, а игры будет загружать администратор, что позволит строго контролировать качество игр, которые будут выкладываться в системе.

* 1. Постановка задачи
     1. Требования к информационному обеспечению

 структура базы данных разрабатывается на основе следующих сведений:

 об игре (название, описание, категория, разработчик, картинка, ссылка на игру, проголосовавшие пользователи, количество лайков);

 о пользователях (логин, почта, пароль);  о категориях (название).

 должна быть обеспечена целостность базы данных и защита от несанкционированного доступа.

* + 1. Требования к техническому обеспечению
       1. Требования к техническому обеспечению серверной части:  тип ЭВМ – IBM PC совместимый;

 объем ОЗУ – не менее 2 Гб;

 объем свободного пространства на внешнем диске – не менее 50 Гб;  наличие подключения к сети Интернет;

 манипулятор – мышь;

 технические характеристики определяются в процессе выполнения проекта.

* + - 1. Требования к техническому обеспечению клиентской части  тип ЭВМ – IBM PC совместимый;

 монитор с разрешающей способностью не ниже 800 х 600;  манипулятор – мышь;

технические характеристики определяются в процессе выполнения проекта.

* + 1. Требования к программному обеспечению
       1. Требования к программному обеспечению серверной части  тип операционной системы – Windows 7 и выше;

СУБД – MongoDB.

* + - 1. Требования к программному обеспечению клиентской части  тип операционной системы – Windows 7 и выше;

браузер – Google Chrome 86.0.4240.183 (64-битный) и выше.

* + - 1. Требования к программному обеспечению рабочего места разработчика

 тип операционной системы – Windows 7 и выше;

языки программирования – Html, CSS, JavaScript (React, Express), SQL;

среда программирования – Visual Studio Code; СУБД – MongoDB;

среда проектирования – StarUML 6.3.1.

1. Проектирование системы

Проектирование является процессом определения структуры, компонентов, интерфейсов и других характеристик системы или ее части. В первую очередь, необходимо определить архитектуру системы. Это включает в себя выбор архитектурного стиля, будь то монолит, микросервисы или клиент-серверная модель, и разработку общей структуры, которая будет служить основой для всех компонентов.

Разрабатываемая система будет веб-приложением, которое передает данные по протоколу HTTP, с двухзвенной клиент-серверной архитектурой и тонким типом клиента.

* 1. Функциональная модель

Функциональные требования задают «что» система должна делать; нефункциональные – с соблюдением «каких условий» (например, скорость

отклика при выполнении заданной операции). При разработке этих требований в первую очередь необходимо учитывать потребности пользователя (заказчика). Пользовательские требования (User Requirements) – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы.

Функциональная спецификация системы приведена в таблице 1.

Таблица 1 – Перечень функций, выполняемых системой

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название подсистемы | Название функции | Информационная среда | | | |
| Входные данные | | Выходные данные | |
| Назначение  (наименование) | Тип,  ограничения | Назначение  (наименование) | Тип,  ограничения |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Аутентификация | Аутентификация пользователя | Логин | Строка (string) | Токен с ID пользователя | Строка (string) |
| Пароль | Строка (string) |
| Работа с играми | Добавление игры | Название | Строка (string) | Игра | Объект «Игра» |
| Описание | Строка (string) |
| Категории | Массив строк |
| Разработчик | Строка (string) |
| Ссылка | Строка (string) |
| Картинка | Строка (string) |
| Оценившие  пользователи | Массив строк |

Продолжение таблицы 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Работа с играми | Редактирование игры | Название | Строка (string) | Изменённая игра | Объект «Игра» |
| Описание | Строка (string) |
| Категории | Строка (string) |
| Разработчик | Строка (string) |
| Ссылка | Строка (string) |
| Картинка | Строка (string) |
| Оценившие  пользователи | Строка (string) |
| Удаление игры | ID игры | Целое (int) | - | - |

* + 1. Перечень исключительных ситуаций

Исключительная ситуация – это ситуация, при которой система не может выполнить возложенных на нее функций или которая может привести к денормализации работы системы.

В таблице 2 приведен перечень исключительных ситуаций для разрабатываемой системы и описаны реакции системы на их возникновение.

Таблица 2 – Перечень исключительных ситуаций

|  |  |  |  |
| --- | --- | --- | --- |
| № | Подсистема | Исключительная  ситуация | Реакция системы |
| 1 | Взаимодействия с серверной частью | Невозможно  подключиться к серверу | Выдача сообщения  «Ошибка соединения» |
| 2 | Взаимодействия с БД | Внезапное отключение базы данных | Восстановление подключения или вывод сообщения «Разрыв  соединения» |
| 3 | Аутентификации | Пользователь передаёт  невалидные логин или пароль | Выдача сообщения  «Значения невалидны» |

* + 1. Структурная схема системы

Структурная схема – это схема, определяющая основные функциональные части изделия, их назначение и взаимосвязи.

Сущность структурного подхода к разработке системы заключается в её декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны [5].

На данном этапе разработки программной системы используется структурный подход к проектированию. Декомпозиция помогает сохранить целостность системы и облегчает процесс её разработки, тестирования и дальнейшего сопровождения.

Программная система для работы с песнями представляет собой клиент-серверное веб-приложение с двумя основными компонентами: клиентская часть и серверная часть.

На рисунке 5 приведена структурная схема разрабатываемой системы. В состав клиентской части входят следующие подсистемы:

* подсистема визуализации, которая отвечает за отображение пользовательского интерфейса (UI) на клиентской стороне;
* подсистема взаимодействия с серверной частью;
* подсистема «Пользователь» отвечает за взаимодействие с играми;
* подсистема «Администратор» управляет настройками системы, имеет возможность контролировать базу данных и управлять пользователями;
* подсистема регистрации и авторизации обеспечивает возможность пользователей, соответственно, зарегистрироваться и авторизоваться.

В состав серверной части входят следующие подсистемы:

* подсистема аутентификации обеспечивает процесс определения, что запрос отправлен именно от этого пользователя, проверяя подлинность отправленных данных;
* подсистема работы с играми отвечает за возможность добавлять, редактировать и удалять игры;
* подсистема взаимодействия с клиентской частью осуществляет обмен данными между клиентскими подсистемами и сервером, обрабатывая запросы пользователей;
* база данных хранит данные о пользователях, категориях и играх. Она обрабатывает запросы от клиентской и серверной частей, обеспечивая целостность и актуальность информации.
  + 1. Диаграмма вариантов использования

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. На ней изображаются отношения между актерами и вариантами использования.

Визуальное моделирование с использованием нотации UML можно представить как процесс поуровневого спуска от наиболее общей и абстрактной концептуальной модели исходной бизнес-системы к логической, а затем и к физической модели соответствующей ПС. Вначале строится модель в форме так называемой диаграммы вариантов использования (use case diagram), которая описывает функциональное назначение системы и является исходным концептуальным представлением в процессе ее проектирования и разработки.

На рисунке 6 приведена диаграмма вариантов использования пользователя. На диаграмме также представлен администратор, который тоже является ролью системы. Пользователь обобщает эту роль. Пользователь может войти в систему просматривать разделы и песни, играть и оценивать игры. Администратор осуществляет работу с сущностями в БД.

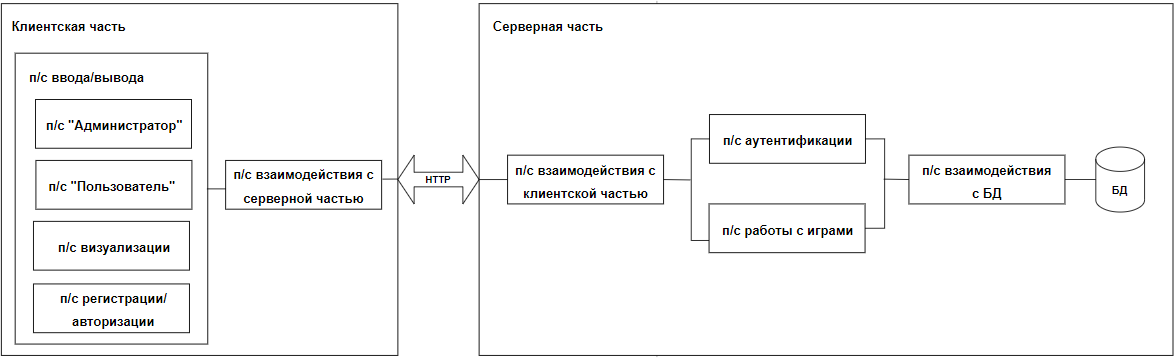


Рисунок 5 – Структурная схема системы

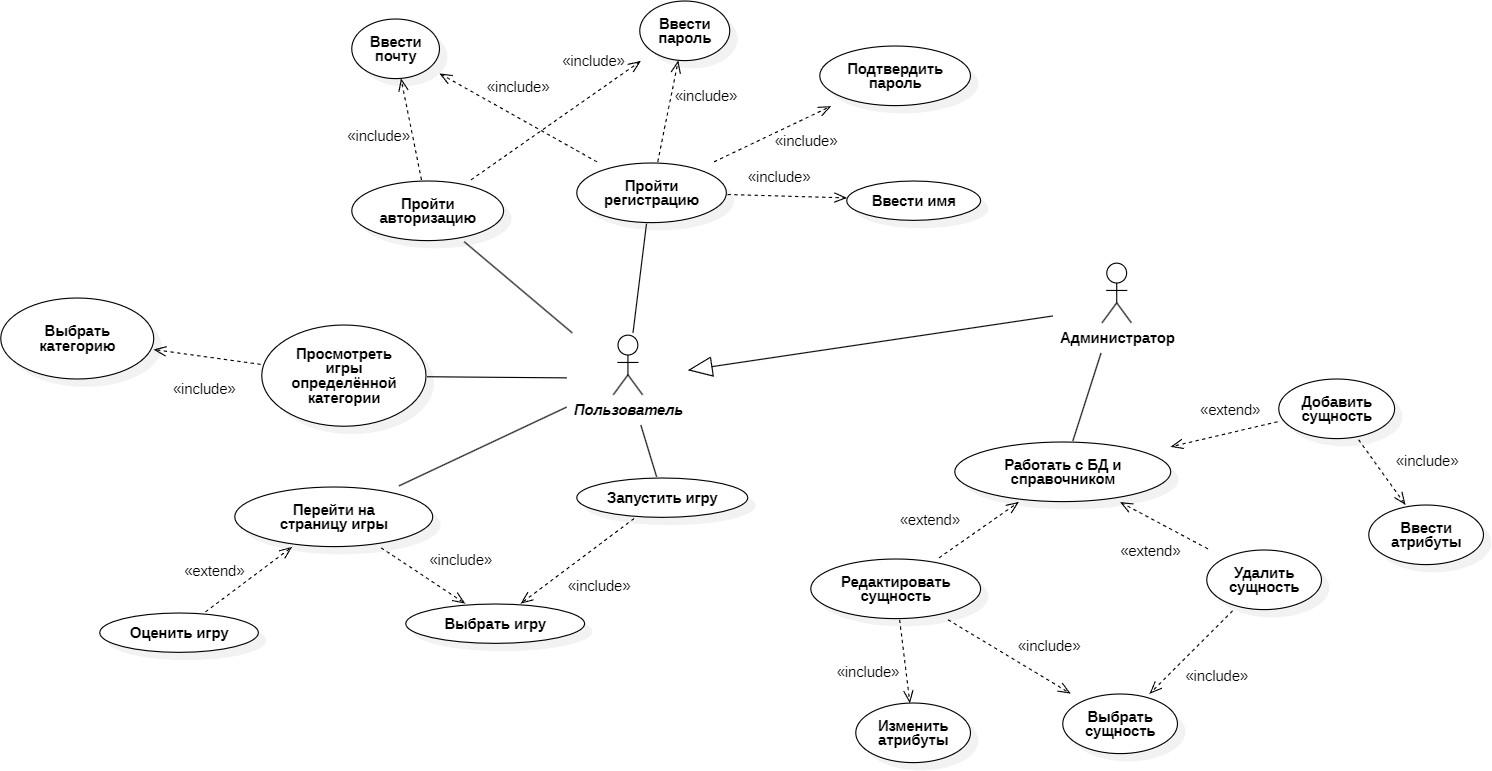


Рисунок 6 – Диаграмма вариантов использования

23

* 1. Логическая схема данных

На рисунке 7 приведена логическая схема БД.

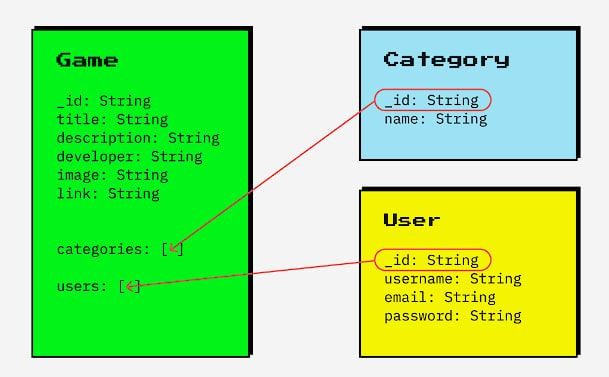


Рисунок 7 – ER-диаграмма базы данных веб-приложения

На диаграмме отображены сущности БД, которые необходимы для работы системы, и их атрибуты:

 Пользователь

 идентификатор пользователя, int (первичный ключ);  имя, string;

 пароль, string;  почта, string.

Категория

 идентификатор категории, int (первичный ключ);  название категории, string.

Игра

 идентификатор игры, int (первичный ключ);  название игры, string;

 описание игры, string;

 разработчик игры, string;  обложка игры, string;

 ссылка на игру, string. Это поле будет представлять собой

ссылку на html страницу с игрой, которая будет храниться в облаке Яндекса S3;

 категории игры, json набор, содержащий идентификаторы жанров игры;

 оценившие пользователи, json набор, содержащий идентификаторы оценивших игру пользователей.

* 1. Разработка прототипа интерфейса пользователя системы

Интерфейс пользователя является одним из важнейших элементов программы, это та часть программы, которая находится у всех на виду. Недочёты в пользовательском интерфейсе могут серьёзно испортить впечатление даже о самых многофункциональных программах. Именно поэтому разработке и проектированию пользовательского интерфейса нужно уделять особое внимание.

Прототипирование - это процесс создания первичной версии изделия, макета или модели с целью тестирования и оценки ее удобства использования, технических характеристик, эргономики и внешнего вида перед началом полной производственной реализации. Прототипирование является существенной частью процесса разработки продукта, который позволяет быстро и дешево проверить свои идеи, улучшить дизайн, избежать ошибок и снизить риски на стадии производства. Благодаря прототипированию можно убедиться в том, что продукт соответствует потребностям клиентов еще на этапе разработки и тестирования. Прототипирование также помогает сократить расходы на производство и улучшить его эффективность, поскольку мы можем избавиться от ненужных деталей, определить оптимальный способ производства и убедиться в том, что продукт или услуга соответствуют критериям качества.

В целом, прототипирование является неотъемлемой частью инновационного развития бизнеса, помогающей увеличить успешность

новых продуктов и услуг, а также снизить риски, связанные с инвестированием в новые проекты и продукты.

На рисунке 8 представлен прототип формы главной страницы. На ней пользователь сможет перейти в разделы «Новинки», «Популярное»,

«Шутеры», «Ранеры», «Пиксельные», «TDS». Также в правом верхнем углу будет кнопка для входа в учётную запись или выхода, если пользователь, соответственно, авторизован или нет. Помимо прочего на главной странице будут изображены игры, разбитые по категориям. При нажатии на карточку игры, соответственно, на её страницу.

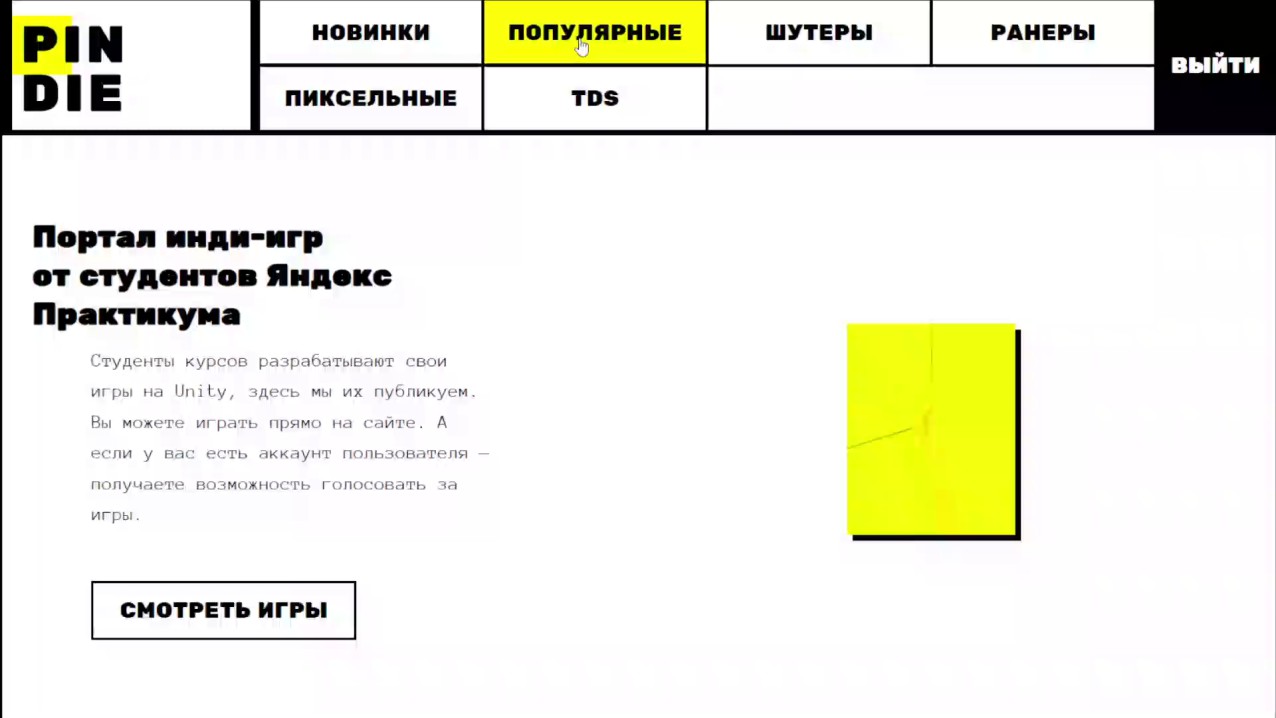


Рисунок 8 – Прототип формы главной страницы

На рисунке 9 изображён прототип страницы игры. На этой странице будет располагаться окно, в котором будет запускаться игра, а снизу этого окна будет располагаться название игры, её описание, имя автора игры и кнопка для того, чтобы можно было оценить игру с подписью о том, сколько человек уже оценило игру.



Рисунок 9 – Прототип страницы игры

На рисунке 10 приведён прототип раздела «Популярное». В этом разделе будут отображаться карточки игр с их обложками, названием, описанием, именем разработчика и сведениями о том, сколько человек оценило игру. При наведении на карточку она будет увеличиваться в размерах и подсвечиваться жёлтым.

Рисунок 10 – Прототип раздела «Популярное»

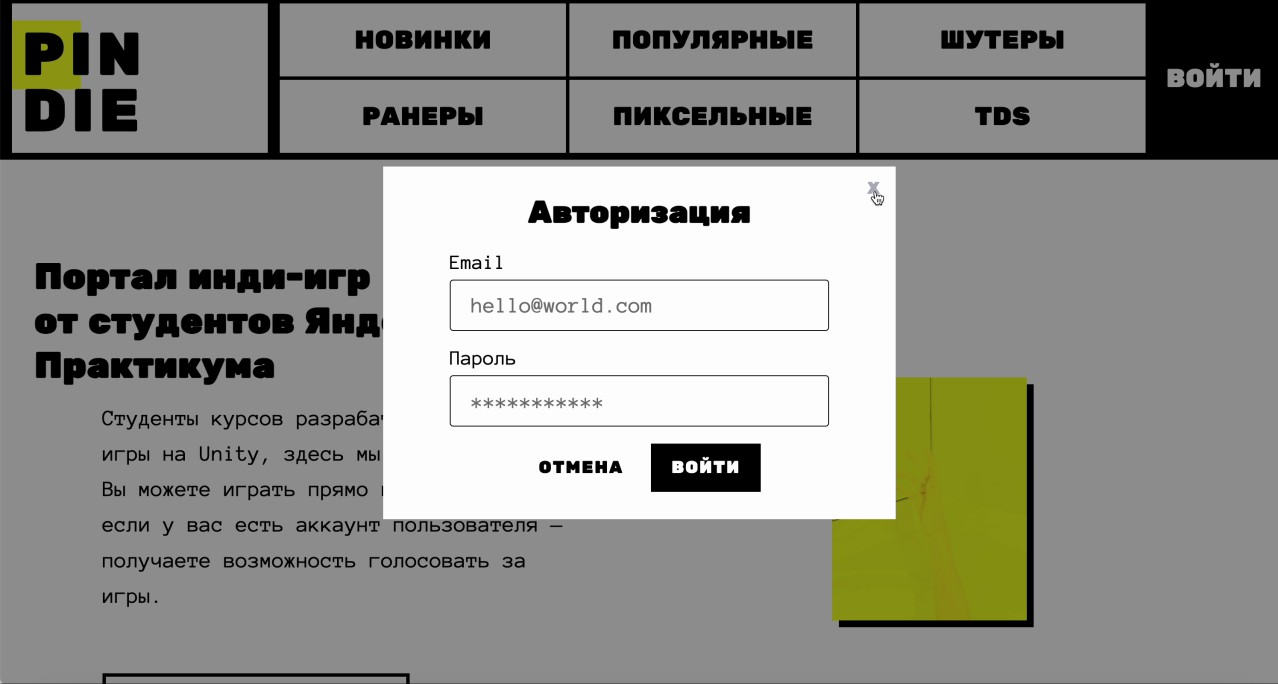
На рисунке 11 приведён прототип окна авторизации. В окне будут отображаться поля для ввода почты и пароля, кнопки для входа и закрытия окна.

Рисунок 11 – Прототип окна авторизации

На рисунках 12 и 13 приведены прототипы страницы администратора. На странице будет отображаться отображаться информация об играх, категориях и пользователях с возможностью редактирования информации.

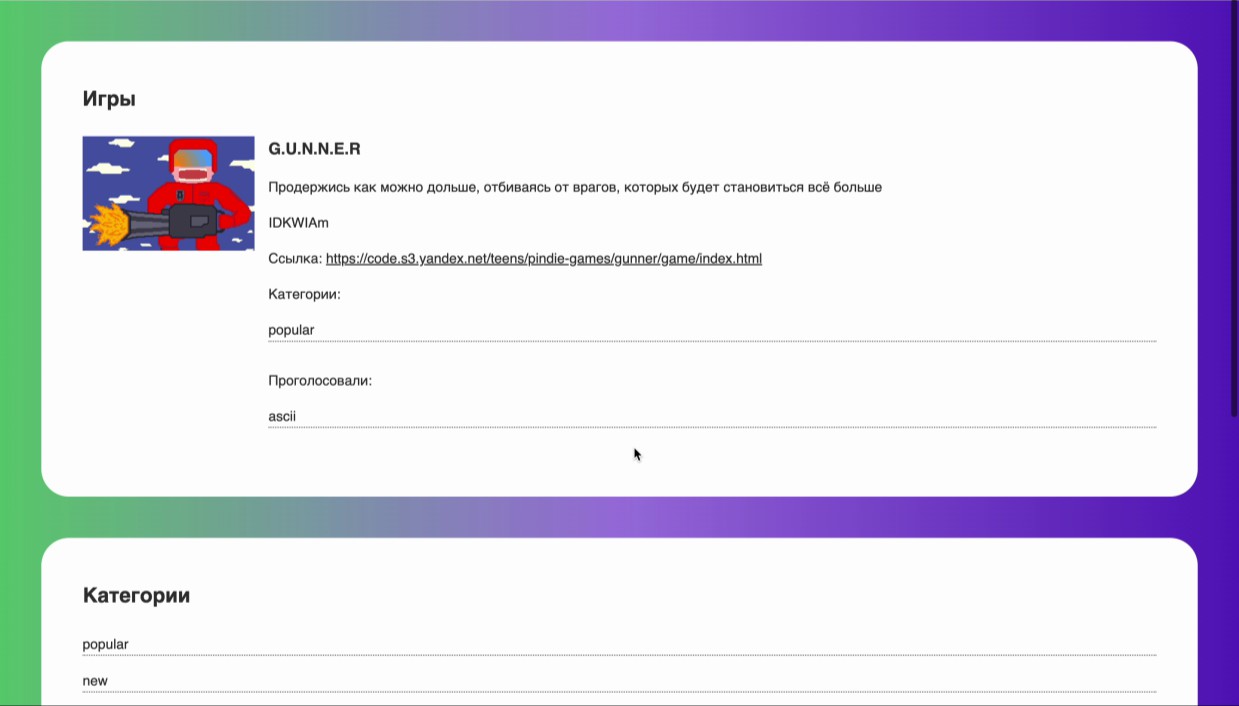


Рисунок 12 – Прототип страницы администратора часть 1

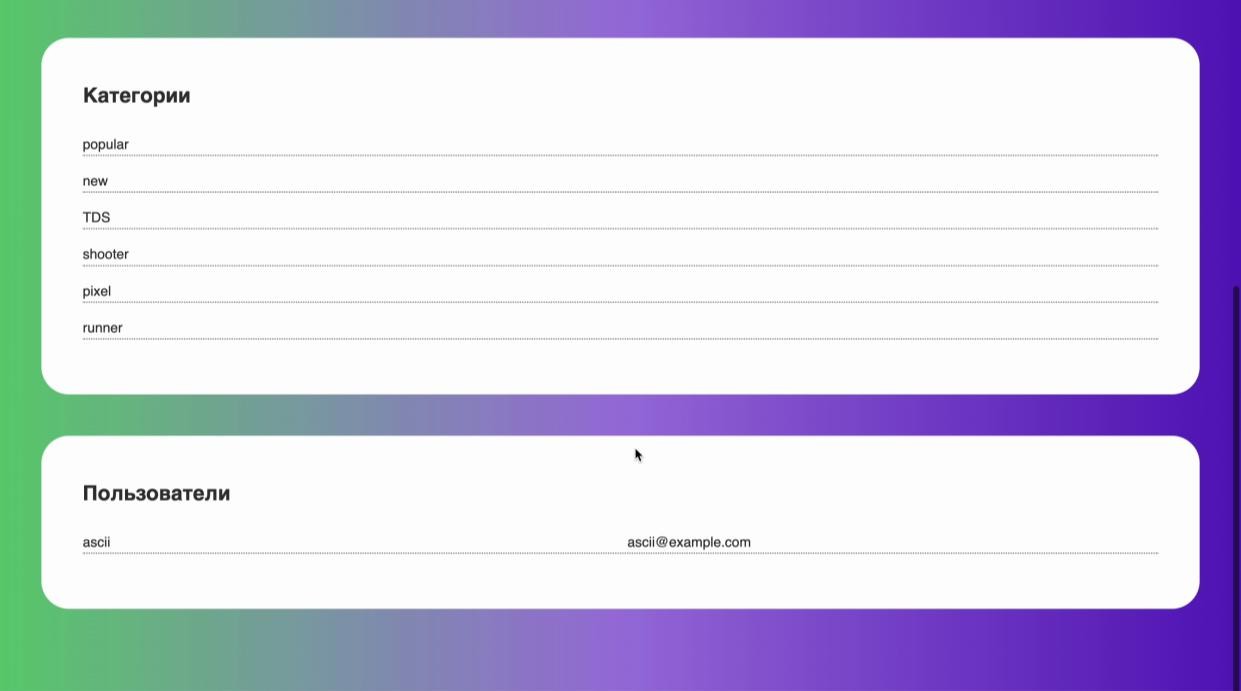


Рисунок 13 – Прототип страницы администратора часть 2

1. Реализация веб-приложения
   1. Разработка физической модели БД

Схема БД приведена на рисунке 7. Для реализации БД была выбрана СУБД MongoDB, так как она обладает рядом преимуществ:

 в MongoDB структуры данных могут быть более динамичными. Это означает, что вы можете легко добавлять новые поля в документы без необходимости изменять всю структуру базы данных, как это происходит в реляционных системах;

 MongoDB поддерживает вложенные структуры данных, такие как массивы и объекты, что позволяет лучше организовывать связанные данные и избегать необходимости связывать таблицы, как в реляционном мире;

 MongoDB хорошо подходит для хранения и обработки неструктурированных или полуструктурированных данных, таких как JSON- документы. Это делает его идеальным для приложений, которые требуют высокой гибкости в управлении данными.

* 1. Разработка серверной части системы

Разработка серверной части велась при помощи JavaScript библиотеки Express, и в результате работы было написано API.

 добавление игры – post-метод, добавляет новую игру в систему, входными данными здесь являются название игры, имя разработчика, описание игры, массив категорий игры, ссылка на игру, обложка и массив оценивших игру пользователей;

 получение всех игр – get-метод, возвращает список всех игр, выходные данные – массив объектов Игра;

 получить игру – get-метод, возвращает игру по её идентификатору, входные данные – идентификатор игру, выходные – объект Игра;

 обновление игры – put-метод, обновляет информацию об игре, входными данными здесь являются идентификатор игры, название игры, имя разработчика, описание игры, массив категорий игры, ссылка на игру, обложка и массив оценивших игру пользователей;

 удаление игры – delete-метод, удаляет игру по её идентификатору, входные данные – идентификатор игры;

 добавление пользователя – post-метод, добавляет пользователя в систему, входные данные – имя пользователя, почта пользователя, пароль пользователя;

 редактирование пользователя – pull-метод, обновляет информацию о пользователе, входные данные – идентификатор пользователя, имя пользователя, почта пользователя, пароль пользователя;

 удаление пользователя – delete-метод, удаляет пользователя по его идентификатору;

 добавление категории – post-метод, входные данные – название категории;

 редактирование категории – pull-метод, входные данные – название категории, идентификатор категории;

 удаление категории – delete-метод, входные данные – идентификатор категории.

Структура приведённого написанного API приведена на рисунке 14.

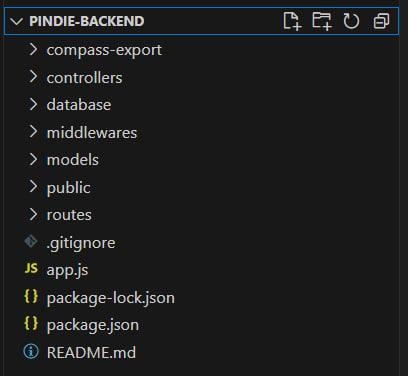


Рисунок 14 – структура серверной части

* 1. Разработка клиентской части

Клиентская часть была реализована согласно макетам, приведённым в разделе 2.3. Результат разработки одной из страниц изображён на рисунке 15.

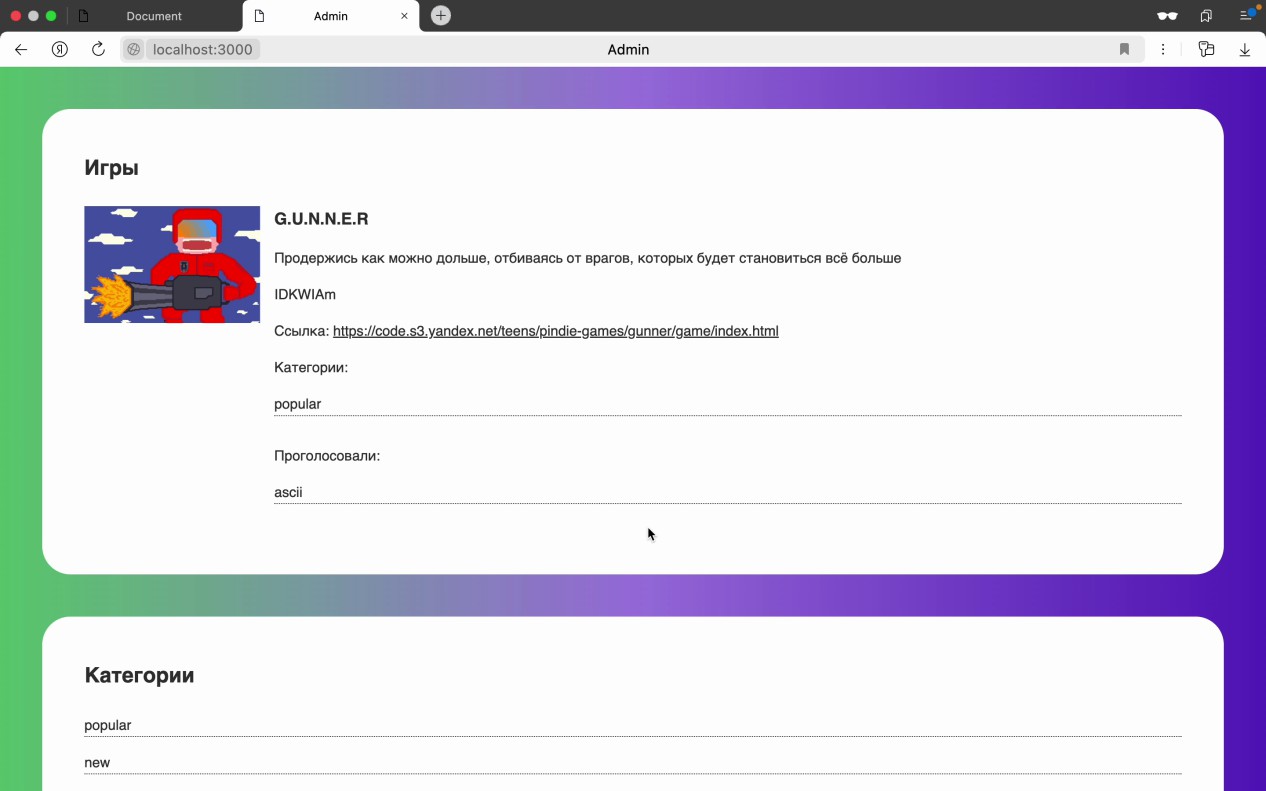


Рисунок 15 – разработанная страница администратора

Структура разработанной клиентской части приведена на рисунке 16.

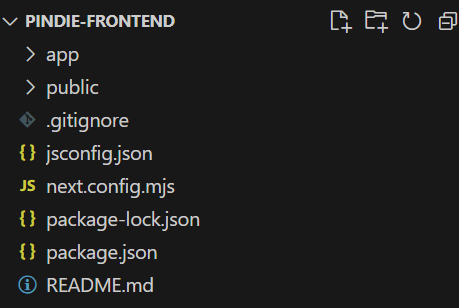


Рисунок 16 – структура клиентской части

* 1. Тестирование системы

При тестировании системы были проверены основные функции системы. Так, при выборе игры на главной странице успешно открывается её страница - рисунки 17 и 18.



Рисунок 17 – Разработанная главная страница

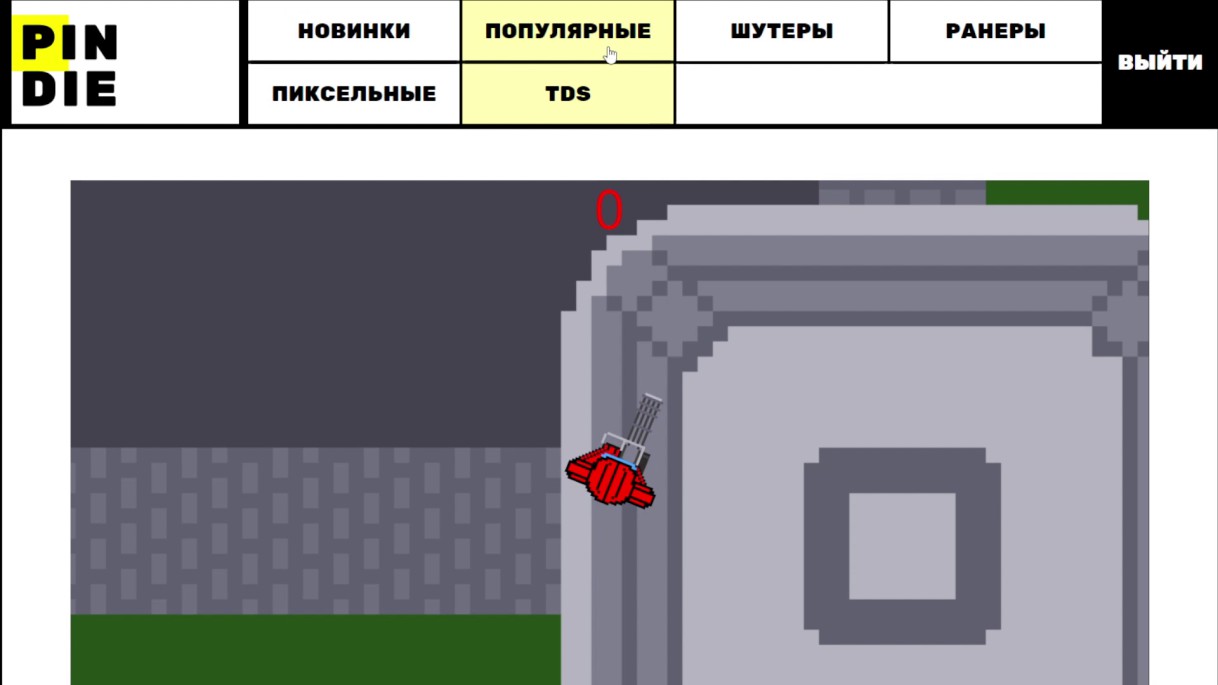


Рисунок 18 – Страница игры

Также успешна была проверена функция оценивания игры – рисунки 19 и 20.



Рисунок 19 – Страница игры до оценивания песни

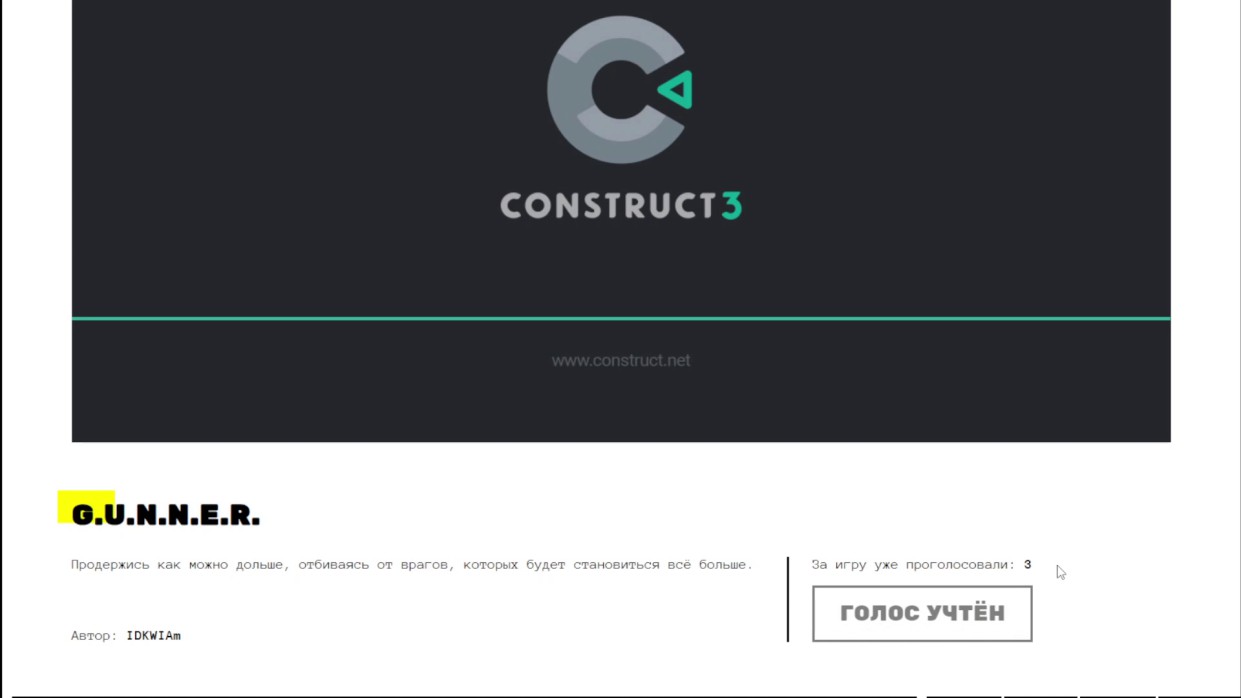


Рисунок 20 – Страница игры после оценивания песни

По результатам тестирования можно сказать, что функционал системы был реализован, и она работает корректно.

1. Выводы

В процессе выполнения курсового проекта было разработано веб- приложение «Каталог инди-игр». Этап разработки был поделён на разделы. Так, в первом разделе был произведён анализ предметной области, анализ систем аналогов, была определена актуальной разработанной системы. Во втором разделе было выполнено проектирование веб-приложение, в ходе которого была описана модель базы данных, были определены функции, выполняемые системой, возможные исключительные системы, была построена диаграмма вариантов использования и спроектированы макеты веб-приложения. А в третьем разделе были описаны результаты разработки веб-приложения, включая БД, фронтенд и бэкенд.

По итогам выполненной работы можно сказать, что все задачи курсовой работы были успешно выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Найди пару, запомни ячейку. [Электронный ресурс]. URL: https://www.vdm.ru/products/338/1261/ (дата обращения: 12.02.2025).
2. Зеленко Л.С. Технологии программирования и программная инженерия (1 часть) учебное пособие / Л.С. Зеленко. – Самара: изд-во СГАУ, 2006. – 96 с.: ил.
3. Компьютерная игра: основные понятия и термины [Электронный ресурс]. URL: https://[www.finam.ru/publications/item/kompyuternaya-igra-](http://www.finam.ru/publications/item/kompyuternaya-igra-) 20230629-0835/ (дата обращения: 20.12.2024).
4. Что такое движок в играх простыми словами [Электронный ресурс]. URL: https://uchet-jkh.ru/i/cto-takoe-dvizok-v-igrax-prostymi- slovami/#:~:text=Он%20включает%20в%20себя%20набор,Работа%20с%20гра фикой%20и%20анимацией. (дата обращения: 20.12.2024).
5. Что такое прототипирование [Электронный ресурс]. URL: https://[www.carrotquest.io/glossary/prototyping/#:~:text=Прототипирование%20](http://www.carrotquest.io/glossary/prototyping/#%3A~%3Atext%3DПрототипирование%20) позволяет%20быстро%20получить%20обратную,рынок%20новых%20продук тов%20и%20услуг (дата обращения: 21.12.2024).

ПРИЛОЖЕНИЕ А

Листинг модулей программы

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Chordy</title>

<link rel="shortcut icon" href="static/img/Icon.png" />

<link rel="stylesheet" href="static/styles/search.css">

<link rel="stylesheet" href="static/styles/navbar.css">

<link rel="stylesheet" href="static/styles/footer.css">

</head>

<body>

<div class="wrapper">

<div class="navbar">

<a href="index.html">Главная</a>

<a href="#">Исполнители</a>

<a href="#">Подборки</a>

<a href="#">Популярное</a>

<a href="addSong.html">Добавить подбор</a>

<div class="profile">

<a href="#"><img src="static/img/Профиль.png" alt="" class="profile-

img"></a>

</div>

<input type="text" placeholder="Поиск">

<div class="search">

<a href=""><img src="static/img/Search.png" alt="" class="search-img"></a>

</div>

</div>

<h2 id="message">Загрузка песен...</h2>

<ul class="song-list" id="songList"></ul>

<div class="footer">

<a href="">Обратная связь</a>

<a href="">Правила сайта</a>

</div>

</div>

<script>

async function fetchSongs() { try {

const response = await fetch('https://localhost:3000/api/songs'); const songs = await response.json();

const songsListDiv = document.getElementById('songsList'); songsListDiv.innerHTML = '';

songs.forEach(song => {

const songDiv = document.createElement('div'); songDiv.className = 'song';

songDiv.innerHTML=`<strong>Исполнитель:</strong>

${song.artist}<br><strong>Песня:</strong> ${song.title}<br><strong>Просмотры:</strong>

${song.views}`;

songsListDiv.appendChild(songDiv);

});

} catch (error) {

console.error('Ошибка при получении песен:', error);

}

}

// Вызов функции для загрузки песен fetchSongs();

</script>

</body>

</html>