

Министерство науки и высшего образования Российской Федерации



Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования

«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _ ИУК «Информатика и управление» _____

КАФЕДРА _ ИУК5 «Системы обработки информации» _____

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К курсовой работе на тему:

«Афиша мероприятий»

По дисциплине Модели данных

Студент гр. ИУК5-51Б

_____ (подпись)

(Жирнов С. А.)
(Ф.И.О.)

Руководитель

_____ (подпись)

(Кириллов В. Ю.)
(Ф.И.О.)

Оценка руководителя _____ баллов _____
30-50 (дата)

Оценка защиты _____ баллов _____
30-50 (дата)

Оценка практики _____ баллов _____
(оценка по пятибалльной шкале)

Комиссия: _____ (_____)
(подпись) (Ф.И.О.)

_____ (_____)
(подпись) (Ф.И.О.)

_____ (_____)
(подпись) (Ф.И.О.)

Калуга, 2024

Содержание

1. Техническое задание.....	3
1.1 Общие сведения.....	3
1.2 Цели и назначения создания системы.....	3
1.3 Характеристика объекта автоматизации.....	3
1.4 Требования к системе.....	4
1.5 Состав и содержание работ по созданию (развитию) системы.....	4
1.6 Порядок разработки информационной системы.....	5
1.7 Порядок контроля и приёмки системы.....	5
1.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие.....	5
1.9 Требования к документированию	6
1.10 Источники разработки.....	6
2. Научно-исследовательская часть	7
2.1 Введение	7
2.2 Задачи:.....	7
2.3 Описание предметной области	8
2.4 Анализ аналогов.....	9
2.5 требования к системе	17
2.6 Обоснование выбора инструментов и платформ для разработки.....	20
3. Проектно-конструкторская часть	25
3.1 Логическая схема базы данных:	25
3.2 Разработка интерфейса взаимодействия пользователя с системой. ...	25
3.3 Разработка архитектуры приложения	30
3.4 Реализация функционирующего приложения.....	31
4. Проектно-технологическая часть.....	32
4.1 Порядок развёртывания системы:.....	32
4.2 Разработка руководства пользователя и администратора	32
4.3 Тестирование программы реальными данными	35
Список использованных источников	38

1. Техническое задание

1.1 Общие сведения.

1.1.1 Полное наименование системы:

Афиша мероприятий

1.1.2 Наименование организации-заказчика, разработчика системы:

Заказчик: КФ МГТУ им. Н.Э. Баумана

Разработчик: Студент КФ МГТУ им. Н.Э. Баумана группы ИУК5-51Б Жирнов
Станислав Алексеевич

1.1.3 Плановые сроки начала и окончания работы по созданию системы:

- Плановый срок начала работ - 5 сентября 2024 года.
- Плановый срок окончания работ - 1 декабря 2024 года.

1.2 Цели и назначения создания системы.

1.2.1 Назначение системы:

Система предназначена для администрации города, чтобы управлять афишей мероприятий и обеспечивать доступ к информации о различных событиях (концерты, выставки, мастер-классы и т.д.) для населения.

1.2.2 Цели создания системы

Улучшение информированности населения о культурных, развлекательных и образовательных событиях в регионе и осуществление прямого контроля над этой информацией.

1.3 Характеристика объекта автоматизации.

1.3.1 Краткие сведения об объекте автоматизации

Выбор мероприятий для посещения осуществляется на основе информации собранной от различных листовок, наружной рекламы и общения с другими

людьми. На основе этого делается выбор, когда и куда пойти. В результате многие варианты остаются неизвестными, и посещаемость падает.

1.3.2 Сведения об условиях эксплуатации объекта автоматизации

Система используется администрацией города для управления афишей, а также пользователями для поиска информации о мероприятиях. Доступ к системе осуществляется через сайт.

1.4 Требования к системе.

1.4.1 Требование к системе в целом:

- Эргономичность: интерфейс должен быть интуитивно понятным и простым в использовании.
- Безопасность: защита данных пользователей и надежность системы.

1.4.2 Требования к функциям, выполняемым системой:

- Регистрация пользователей.
- Администрирование сайта (удаление нежелательной информации).
- Добавление, редактирование и удаление мероприятий администратором.
- Поиск и фильтрация мероприятий по дате, типу и цене.
- Возможность оставления отзывов и оценок мероприятий.
- Возможность пользователем добавить мероприятие в избранные.

1.4.3 Требования к входным и выходным данным

Входные данные:

- Данные пользователей (имя, электронная почта или телефон, пароль).
- Информация о мероприятиях (название, дата, время, место проведения, описание, цена).

Выходные данные:

- Список мероприятий с возможностью фильтрации.
- Вывод сообщения и покупки билетов.

1.5 Состав и содержание работ по созданию (развитию) системы.

1.5.1 Перечень стадий и этапов работ

Предпроектное проектирование: определение задач и целей системы.

Анализ предметной области: исследование требований и возможностей.

Разработка модели данных: создание структуры базы данных.

Проектирование интерфейса: разработка прототипа пользовательского интерфейса.

Реализация системы: программирование и тестирование системы.

Разработка документации: создание пользовательских и технических инструкций.

Внедрение: установка системы.

1.6 Порядок разработки информационной системы.

Формирование требований к системе

Разработка концепции системы

Разработка технического задания

Эскиз проекта

Технический проект

Документация

Введение системы

Сопровождение системы

1.7 Порядок контроля и приёмки системы.

Система разрабатывается с применением тестирования работоспособности функционала приложения. При добавлении новых функций предыдущие тесты должны сохранять работоспособность. Приёмка работы осуществляется комиссией КФ МГТУ им. Н. Э. Баумана кафедры ИУК5.

1.8 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие.

Подготовка объекта автоматизации к вводу системы в действие включает следующие работы:

Произвести ввод начальных данных в программу
Произвести выпуск программы в свободный доступ
Поддержка программы

1.9 Требования к документированию

Требуется предоставить:

А) Техническое задание в соответствии ГОСТ 34.602-2020

Б) Расчетно-пояснительную записку, включающую в исследовательскую часть, проектно-конструкторскую часть и проектно-технологическую часть, включающую в себя руководство пользователя и руководство программиста (администратора).

Расчётно-пояснительная записка выполняется с учётом требований, предусмотренных ГОСТ 7.32-2001 и 2.105-95.

1.10 Источники разработки

ГОСТ 34.602-2020. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы;

ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;

ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированной системы;

РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов;

ГОСТ 2.105-95. ЕСКД. Общие требования к текстовым документам;

ГОСТ Р. 50377-92 (МЭК 950-86) «Безопасность оборудования информационной технологии, включая электрическое контрольное оборудование».

2. Научно-исследовательская часть

2.1 Введение

Современные города предлагают множество культурных, развлекательных и спортивных мероприятий, однако жители и гости часто сталкиваются с проблемой недостаточной осведомленности о них. Рассеянность информации в различных источниках затрудняет поиск актуальных событий, что снижает интерес и посещаемость мероприятий.

Проект предполагает разработку функционала для отображения и управления событиями, а также администрирования контента, что поможет поддерживать актуальность данных и улучшать пользовательский опыт.

В условиях быстрого развития городской инфраструктуры и увеличения числа культурных, развлекательных и спортивных мероприятий, крайне важно создать удобный и эффективный инструмент для поиска и распространения информации о таких событиях. Существующие способы информирования населения, такие как рекламные листовки, наружная реклама и объявления в социальных сетях, часто оказываются недостаточно эффективными: важные события могут быть не замечены или трудно доступны для целевой аудитории.

Цель разработки: улучшение информированности населения о культурных, развлекательных и образовательных событиях в регионе и осуществление прямого контроля над этой информацией.

2.2 Задачи:

1. Анализ и проектирование базы данных. Определить структуру базы данных, которая будет хранить информацию о мероприятиях, пользователях, отзывах и площадках.
2. Разработка интерфейса пользователя. Создать понятный и удобный интерфейс для пользователей с различными уровнями технической подготовки.

3. Реализация поиска и фильтрации мероприятий. Разработать функционал, позволяющий пользователям легко находить мероприятия по ключевым критериям (тип мероприятия, дата).
4. Создание системы отзывов и оценок. Разработать модуль для добавления отзывов, оценок и формирования рейтинга мероприятий.
5. Реализация функционала избранных мероприятий. Создать возможность для пользователей сохранять интересующие их события в списке избранных и оценивать мероприятия.
6. Администрирование и управление контентом. Разработать панель администратора для управления мероприятиями, отзывами, площадками и комментариями пользователями.

Эта система позволит улучшить информированность населения о культурных, развлекательных и образовательных событиях в регионе и осуществление прямого контроля над этой информацией.

2.3 Описание предметной области

В выходные дни у людей часто возникает вопрос: «Чем заняться?». Некоторые отправляются гулять по паркам и случайно натыкаются на различные фестивали, другие остаются дома, а третьи — гости города, не знающие о том, куда можно сходить. В результате интересные и культурные мероприятия остаются незамеченными многими.

Данное приложение собирает информацию обо всех событиях города в одном месте, предоставляя важный источник информации для жителей и гостей, интересующихся культурной жизнью, развлечениями и спортивными мероприятиями. Пользователи могут просматривать актуальные мероприятия своего города без необходимости регистрации.

Регистрация в приложении открывает дополнительные возможности: только зарегистрированные пользователи могут оставлять отзывы о мероприятиях и добавлять их в список избранных. Для создания аккаунта пользователю требуется указать уникальный ник, пароль и номер телефона.

Пользователи могут просматривать различные мероприятия, организованные по типам (например, кино, спектакли, концерты). Каждое мероприятие включает информацию о продолжительности, дате проведения, стоимости, адресе, рейтинге, описании, изображении и ссылке на сайт организатора.

Рейтинг мероприятий формируется на основе отзывов пользователей, которые оставляют свои оценки и комментарии. Эти отзывы помогают другим пользователям получить представление о качестве мероприятий. Все мероприятия добавляются через администратора, который следит за актуальностью информации и при необходимости удаляет неподходящие отзывы или события.

Кроме того, пользователи создают список избранных мероприятий, что позволяет легко отслеживать интересующие их события. Каждое избранное мероприятие связано с конкретным мероприятием, и один пользователь может добавлять несколько событий в избранное.

Каждое мероприятие проходит на определенной площадке, которая также имеет свои атрибуты — название, адрес, телефон и email представителя. Одна площадка проводит множество мероприятий, но каждое из них происходит только на одной площадке.

2.4 Анализ аналогов

1 – Яндекс Афиша

Ссылка: <https://afisha.yandex.ru/kaluga?source=menu-city>

Присутствует полоска календаря, для выбора даты мероприятий. Есть строчка для поиска по названию, событию и артисту.

Все мероприятия представлены в списке. В описании каждого мероприятия присутствует адрес, дата, картинка, цена билета, с возможностью покупки и схемой зала (Рисунок – 1).

Мероприятия можно добавить в список желаемого.

Нет

оценки

и

отзывов.

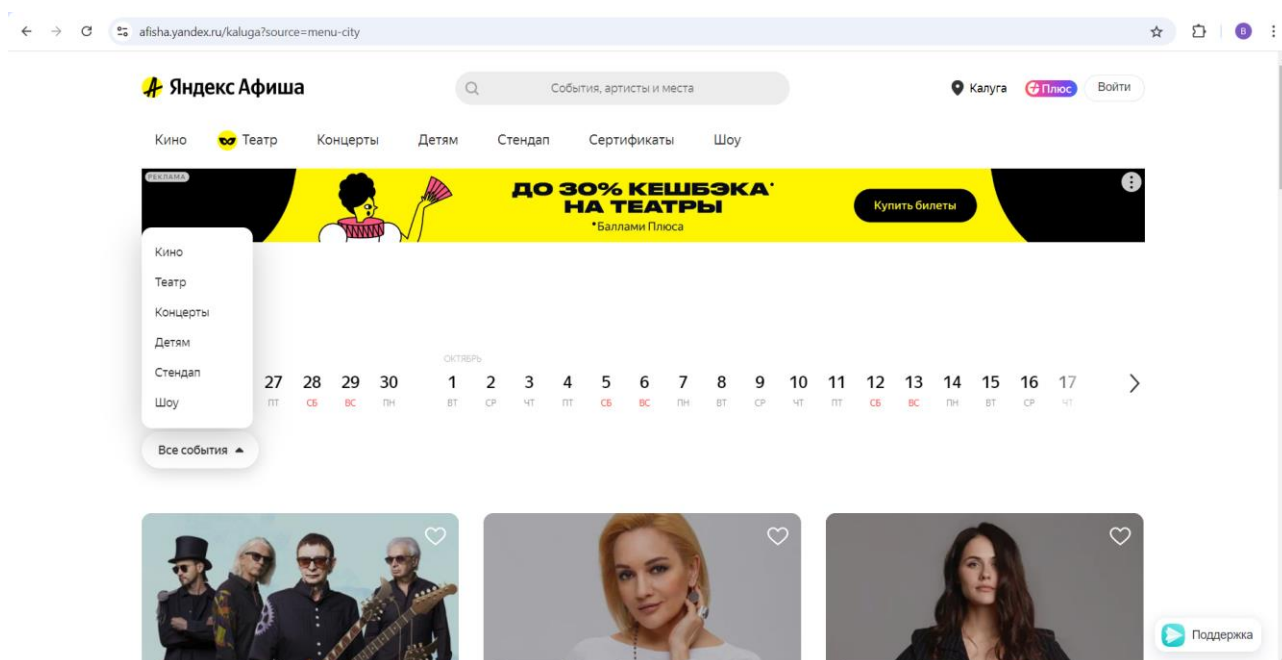


Рисунок 1 – сайт «Яндекс афиша»

2 – Топ Куда

Ссылка: <https://topkuda.ru/>

Очень по сути похоже на Яндекс Афишу.

Аналогичная полоска календаря с датами. Есть фильтр по возрасту, интересам, категориям и стоимости.

Мероприятия представлены списком и разбиты на секции. В описании мероприятий присутствует адрес, картинка, дата, цена билета, ссылка на источник, продолжительность, минимальный возраст и минимальное число участников (Рисунок – 2).

При нажатии на покупку билета переводит на официальный сайт организатора.

Нет оценки и отзывов.

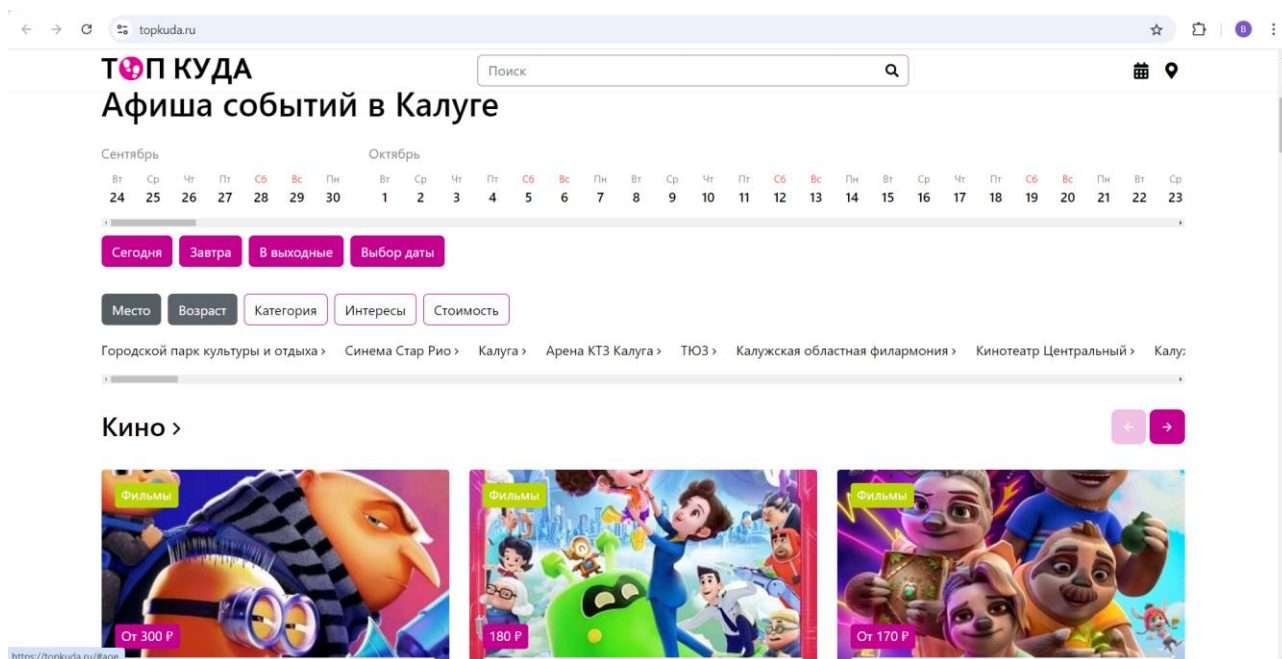


Рисунок 2 – сайт «Топ Куда»

3 – Афиша

Ссылка: <https://www.afisha.ru/kaluga/events/>

Присутствует полоска календаря, для выбора даты мероприятий. Есть сортировка по типам, датам, детским и по мероприятиям с акцией «Пушкинская карта».

Все мероприятия представлены в одном списке. В описании каждого мероприятия присутствует адрес, дата, картинка, цена билета, с возможностью покупки и схемой зала. На сайте можно посмотреть трейлер (Рисунок – 3). Есть оценка и отзывы пользователей.

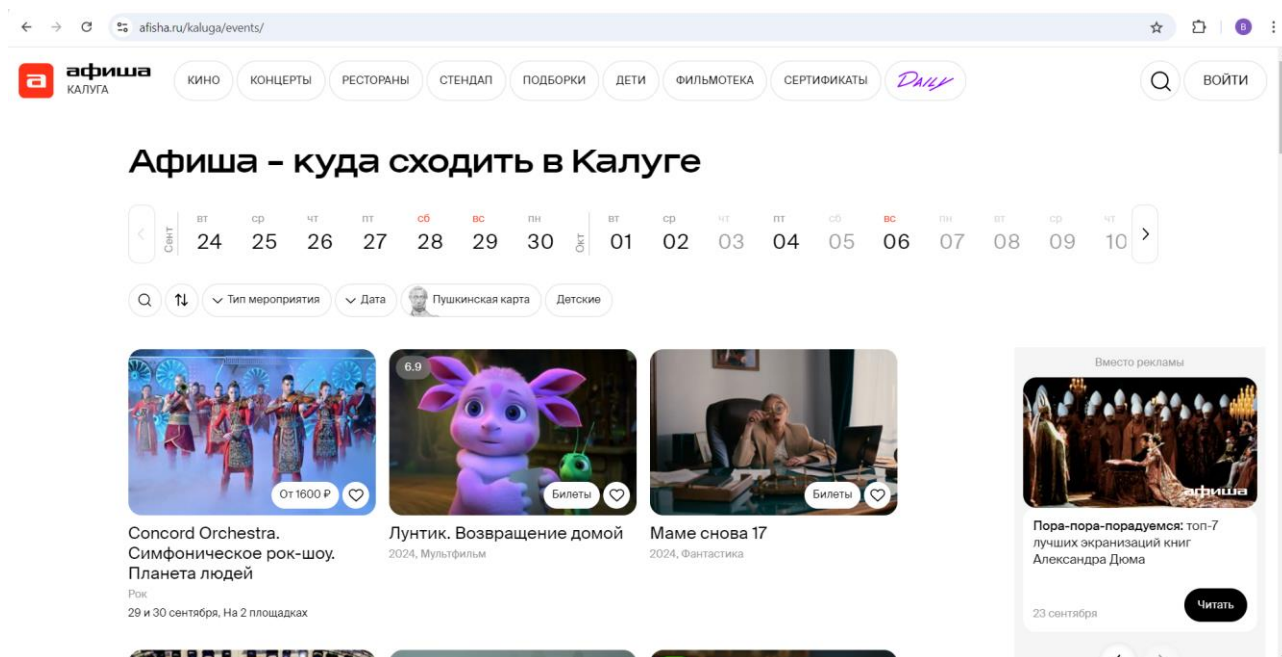


Рисунок 3 – сайт «Афиша»

4 – KUDAGO

Ссылка: <https://kudago.com/msk/#>

В верхней части находятся кнопки для сортировки мероприятий по типам. Есть сектор с новостями в городе.

Мероприятия находятся в одном сплошном списке. Разные мероприятия разного размера. В описании каждого мероприятия присутствует адрес, дата с таймером до самого мероприятия (дни, часы, минуты, секунды), картинка, возрастные ограничения и количество просмотров мероприятия (Рисунок – 4).

При нажатии на покупку билета переводит на официальный сайт организатора.

Есть оценка и отзывы пользователей.

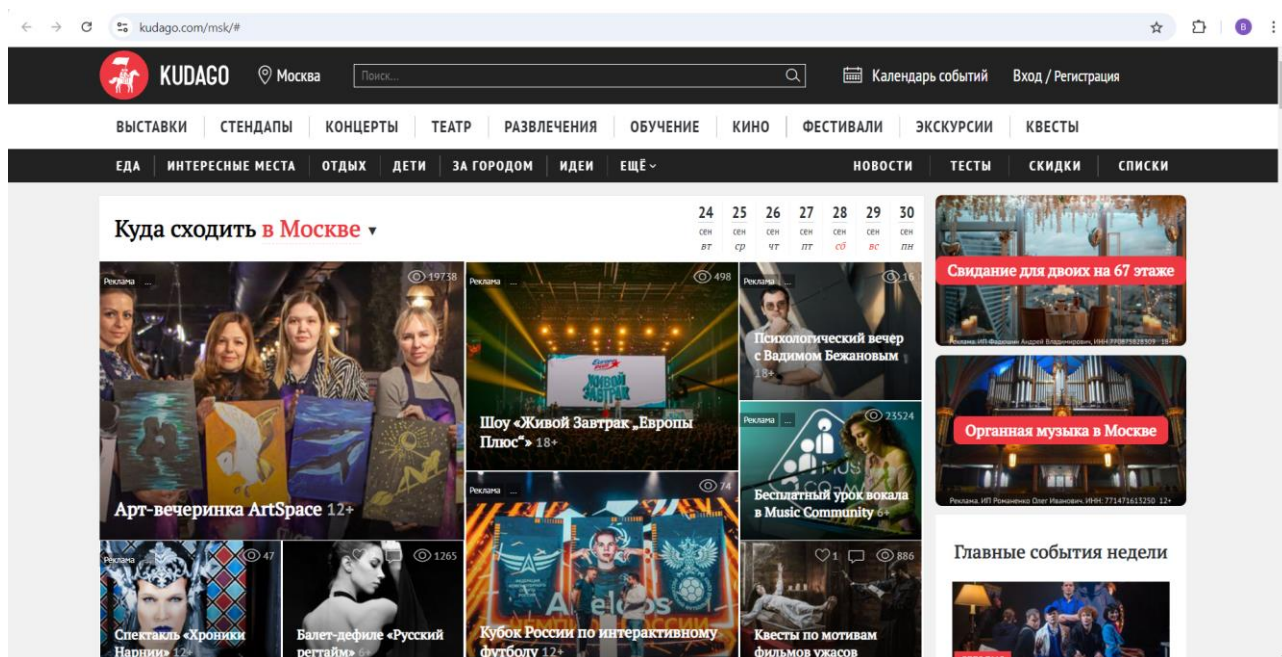


Рисунок 4 – сайт «Kudago»

5 – Kuda Moscow

Ссылка: <https://kudamoscow.ru/>

В верхней части можно выбрать дату.

Ниже список типов событий с картинками.

С боку есть табличка с мероприятиями, проходящими именно сегодня. Есть разделение по метро. Они представлены цифрой, при нажатии на которую включается соответственный фильтр (сегодняшняя дата).

Мероприятия находятся в одном сплошном списке. В описании каждого мероприятия присутствует адрес, дата, картинка, цена билета (Рисунок – 5).

При нажатии на покупку билета переводит на официальный сайт организатора.

Есть оценка и отзывы пользователей.

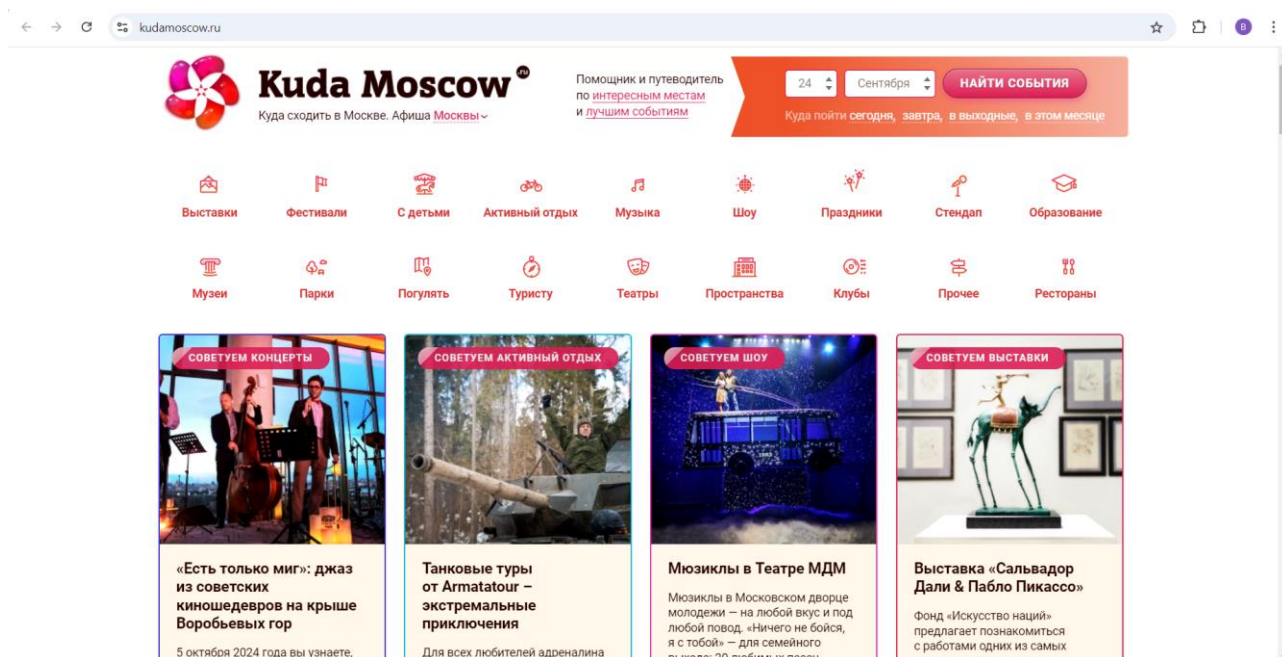


Рисунок 5 – сайт «Kuda Moscow»

6 – Мобильное приложение Яндекс Афиша

События можно отсортировать по дате и типу мероприятий. Все мероприятия представлены списком.

Все мероприятия представлены в списке. В описании каждого мероприятия присутствует адрес, дата, картинка, цена билета, с возможностью покупки и схемой зала.

В приложении есть поиск, избранное и можно посмотреть все купленные через это приложение билеты.

Есть оценки пользователей.

Можно пользоваться приложением не авторизовываясь.

Калуга ▾



Дата

Все события ▾



Пикник

Концерт 17 ноября, 18:00 Дворец культуры КТЗ
от 2 200 ₽



Татьяна Буланова

Концерт 21 ноября, 19:00 Филармония
от 2 000 ₽



Главная



Мои билеты



Избранное



Поиск

Рисунок 6 - Мобильное приложение Яндекс Афиша

7 – Мобильное приложение Kassir

Присутствует полоска календаря, для выбора даты мероприятий. Есть сортировка по типам, датам, детским и по мероприятиям с акцией «Пушкинская карта», а также можно выбрать фильтр «билеты по скидке».

Есть фильтры по стоимости и площадкам.

Есть строка для поиска по названию мероприятия.

Все мероприятия представлены в одном списке.

В описании каждого мероприятия присутствует адрес, дата, картинка, цена билета, с возможностью покупки и схемой зала.

В приложении есть поиск, избранное и можно посмотреть все купленные через это приложение билеты.

Нет отзывов.

Можно пользоваться приложением не авторизовываясь.

Можно подарить другому человеку сертификаты на различные номиналы.

При входе можно указать пожелания по типам мероприятий, по цене, и по интересам в каждом типе мероприятий. А также по городу и времени (Рисунок – 7).

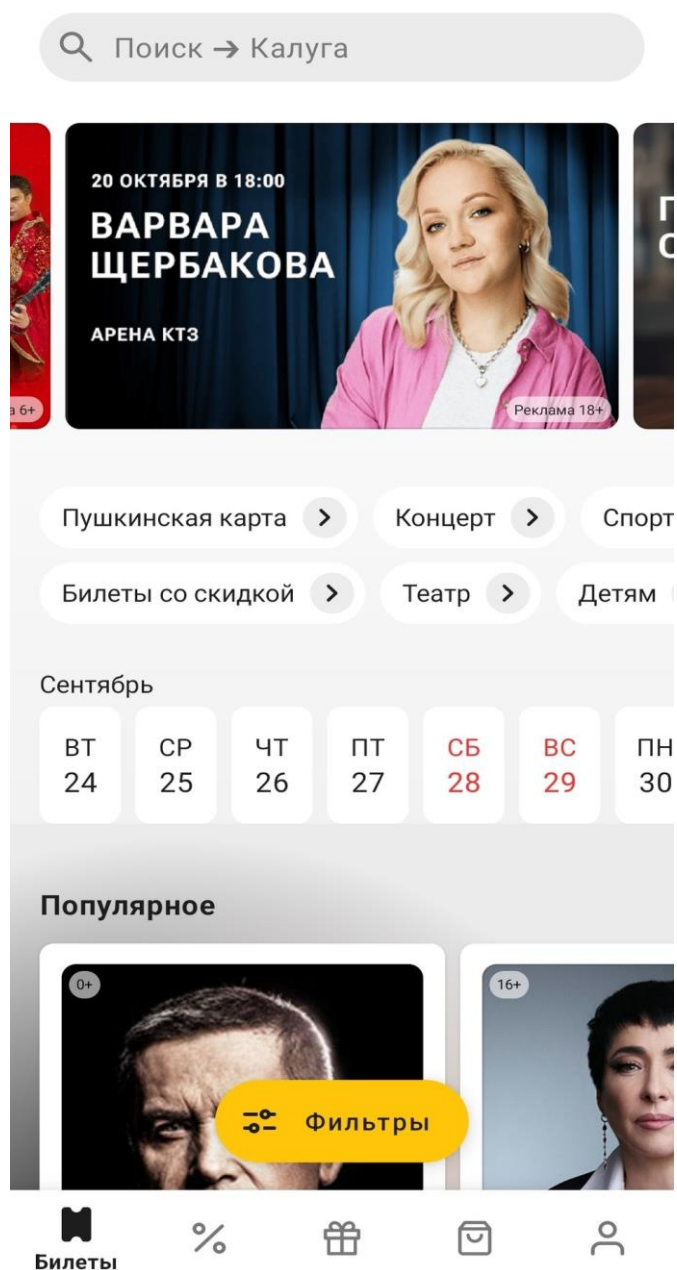


Рисунок 7 - Мобильное приложение Kassir

Изучая представленные аналоги, можно выделить несколько ключевых особенностей, которые позволяют им успешно выполнять свою задачу:

Большинство приложений имеют удобную навигацию с использованием полосы календаря для выбора дат. Это упрощает пользователям поиск интересных событий.

Существует возможность фильтрации мероприятий по типам, что делает поиск более целенаправленным.

Все приложения предоставляют основную информацию о мероприятиях: дату, время, адрес, стоимость билетов и изображения.

Некоторым приложениям удастся добавить дополнительные данные, такие как возрастные ограничения или продолжительность мероприятия.

Оценки и отзывы пользователей помогают сформировать общее представление о мероприятии. Однако не все аналоги предоставляют эту функцию.

Почти все приложения предлагают возможность покупки билетов, но обычно перенаправляют пользователей на сайты организаторов.

Мобильные приложения таких сервисов, как Яндекс Афиша и Kassir, предлагают функции офлайн-доступа, что является значительным плюсом для пользователей.

Эти функции кажутся мне наиболее удобными и практичными, позволяющие сервису выполнять свои основные функции.

2.5 требования к системе

1. Анализ и проектирование базы данных. Определить структуру базы данных, которая будет хранить информацию о мероприятиях, пользователях, отзывах и площадках.
2. Разработка интерфейса пользователя. Создать понятный и удобный интерфейс для пользователей с различными уровнями технической подготовки.

3. Реализация поиска и фильтрации мероприятий. Разработать функционал, позволяющий пользователям легко находить мероприятия по ключевым критериям (тип мероприятия, дата).
4. Создание системы отзывов и оценок. Разработать модуль для добавления отзывов, оценок и формирования рейтинга мероприятий.
5. Реализация функционала избранных мероприятий. Создать возможность для пользователей сохранять интересующие их события в списке избранных и оценивать мероприятия.
6. Администрирование и управление контентом. Разработать панель администратора для управления мероприятиями, отзывами, площадками и комментариями пользователями.

Данное приложение собирает информацию обо всех событиях города в одном месте, предоставляя важный источник информации для жителей и гостей, интересующихся культурной жизнью, развлечениями и спортивными мероприятиями. Пользователи могут просматривать актуальные мероприятия своего города без необходимости регистрации.

Регистрация в приложении открывает дополнительные возможности: только зарегистрированные пользователи могут оставлять отзывы о мероприятиях и добавлять их в список избранных. Для создания аккаунта пользователю требуется указать уникальный ник, пароль и номер телефона.

Пользователи могут просматривать различные мероприятия, организованные по типам (например, кино, спектакли, концерты). Каждое мероприятие включает информацию о продолжительности, дате проведения, стоимости, адресе, рейтинге, описании, изображении и ссылке на сайт организатора.

Рейтинг мероприятий формируется на основе отзывов пользователей, которые оставляют свои оценки и комментарии. Эти отзывы помогают другим пользователям получить представление о качестве мероприятий. Все мероприятия добавляются через администратора, который следит за

актуальностью информации и при необходимости удаляет неподходящие отзывы или события.

Кроме того, пользователи создают список избранных мероприятий, что позволяет легко отслеживать интересующие их события. Каждое избранное мероприятие связано с конкретным мероприятием, и один пользователь может добавлять несколько событий в избранное.

Каждое мероприятие проходит на определенной площадке, которая также имеет свои атрибуты — название, адрес, телефон и email представителя. Одна площадка проводит множество мероприятий, но каждое из них происходит только на одной площадке.

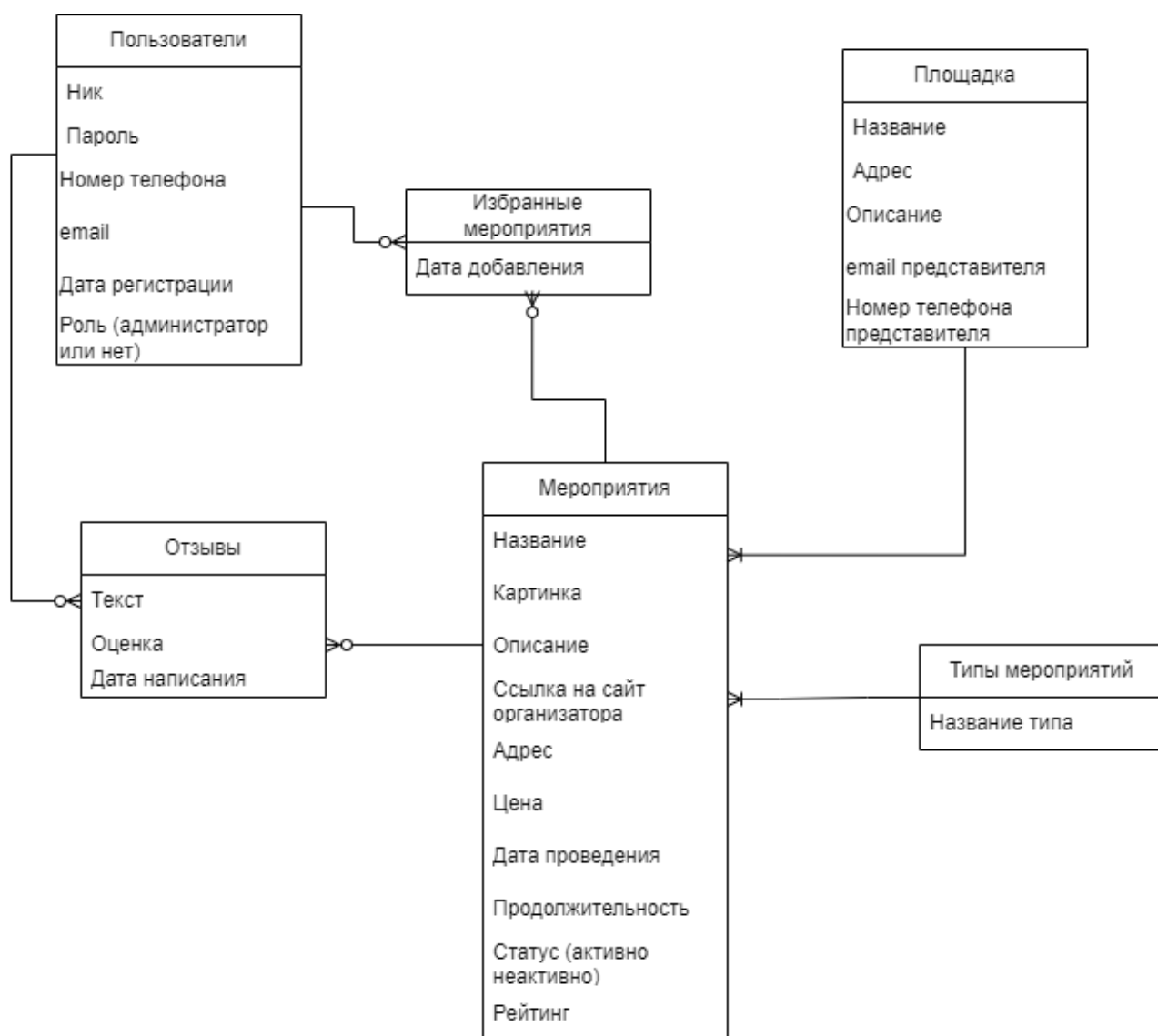


Рисунок 8 – концептуальная схема данных

2.6. Обоснование выбора инструментов и платформ для разработки

1. PostgreSQL

PostgreSQL — это оптимальный выбор для проекта профориентации по нескольким причинам:

Стандартный стек разработки исполнителя: PostgreSQL входит в стандартный стек разработки, что позволяет быстро и эффективно разрабатывать функционал, учитывая уже накопленный опыт работы с этой СУБД.

Поддержка сложных структур данных: база данных содержит множество таблиц, связанных друг с другом (например, вопросы, ответы, результаты тестов). PostgreSQL отлично справляется с реляционными структурами и позволяет эффективно работать с внешними ключами и сложными связями между таблицами. Это важно для поддержания связи между пользователями, тестами и результатами.

Расширенные возможности для работы с типами данных: в проекте используются различные типы данных, такие как текст, массивы. PostgreSQL предоставляет широкий выбор встроенных типов данных и поддерживает их эффективную обработку. Например, поле `suitable_professions` в таблице `test_results` — массив, и PostgreSQL поддерживает работу с массивами, что упрощает хранение и манипуляцию с данными.

Масштабируемость и производительность: PostgreSQL позволяет работать с большими объемами данных и при этом эффективно поддерживает производительность. В проекте профориентации, где хранятся результаты тестов и информация о пользователях, очень важно, чтобы система могла масштабироваться и обрабатывать множество запросов одновременно, особенно если количество пользователей вырастет.

Поддержка транзакций: В PostgreSQL реализована мощная система транзакций, что важно для обеспечения целостности данных. Например, когда пользователь проходит тест, важно, чтобы все данные — результаты, ответы и прогресс — были сохранены корректно и последовательно, без ошибок.

Открытость и активное сообщество: PostgreSQL — это бесплатно.

Таким образом, PostgreSQL предоставляет все необходимые возможности для успешной реализации системы профориентации, обеспечивает надежное хранение данных, высокую производительность и гибкость в работе с разными типами данных.

2. HTML:

HTML был выбран для создания структуры страниц и представления контента в системе профориентации по следующим причинам:

Стандарты веб-разработки: HTML является основой для всех веб-страниц, обеспечивая совместимость с любыми браузерами и устройствами. В контексте системы профориентации — это критично для доступности интерфейса, позволяя пользователям с различными устройствами (ПК, планшеты, мобильные телефоны) работать с системой.

Структура для пользовательских данных: HTML используется для создания форм регистрации и авторизации, а также для отображения личных кабинетов пользователей, где хранятся их данные, результаты тестов и рекомендации. Это позволяет организовать информацию о каждом пользователе (например, результаты тестов, рекомендации по профессиям) в структурированном виде.

Визуализация тестов и вопросов: Вопросы, ответы и изображения, которые сопровождают тесты, будут представлены в HTML-структурах, что упрощает их обработку и отображение. Это позволяет интегрировать текстовые и графические элементы, необходимые для отображения вопросов тестов и их вариантов ответов.

3. CSS:

CSS был выбран для стилизации интерфейса системы, чтобы обеспечить удобный и привлекательный пользовательский опыт:

Гибкость и адаптивность: с помощью CSS можно создать адаптивный дизайн для удобного отображения системы на любых устройствах, что особенно важно для пользователей с мобильных телефонов и планшетов. Это позволяет

пользователю продолжать проходить тесты на разных устройствах, не испытывая проблем с интерфейсом.

Интерактивные элементы: Для улучшения взаимодействия с пользователем CSS позволяет добавлять анимации и эффекты на кнопки, формы и другие элементы интерфейса. Например, кнопки для перехода к следующему вопросу или для сохранения прогресса тестирования могут быть стилизованы так, чтобы повышать удобство их использования.

Динамичность интерфейса: CSS помогает создавать понятную и привлекательную стилизацию для отображения результатов тестов, включая рекомендации по карьере и подходящие профессии, что важно для вовлеченности пользователя и визуального восприятия информации.

4. Node.js:

Node.js был выбран для серверной части системы, обеспечивая эффективную работу с данными и масштабируемость:

Обработка запросов в реальном времени: в системе профориентации важно поддерживать постоянное взаимодействие между клиентом и сервером. Например, пользователи могут завершать тесты в любое время, и их прогресс должен сохраняться мгновенно. Асинхронная модель Node.js позволяет обрабатывать множество запросов одновременно, что критично для масштабируемости системы, особенно если количество пользователей значительно вырастет.

Управление данными пользователей и тестами: Node.js эффективно взаимодействует с базой данных PostgreSQL, где хранятся все тесты, результаты, данные пользователей и их предпочтения. Это позволяет быстро и надежно обрабатывать запросы на сохранение результатов тестов, анализ данных и генерацию рекомендаций.

Модульность и масштабируемость: С использованием фреймворка Express.js на Node.js можно легко организовать маршрутизацию и управление данными, например, для регистрации, авторизации пользователей, создания и

редактирования тестов, управления учетными записями пользователей и хранения результатов тестирования.

Поддержка многозадачности: В проекте важно, чтобы система могла одновременно обрабатывать запросы от большого числа пользователей, проходящих тесты и получающих результаты. Node.js, благодаря своей неблокирующей модели ввода-вывода, идеально подходит для таких задач, обеспечивая высокую производительность даже при увеличении нагрузки.

Реализация алгоритмов анализа тестов: Сервер на Node.js будет управлять логикой тестирования, подсчета результатов на основе весов ответов и генерировать рекомендации по профессиям.

5. JavaScript

JavaScript — это один из наиболее популярных и широко используемых языков программирования, особенно в веб-разработке. Обоснование выбора JavaScript для разработки вашего проекта можно сформулировать на основе нескольких ключевых факторов:

1. Кроссплатформенность: JavaScript является универсальным языком, который работает на всех основных операционных системах и устройствах через браузер. Это позволяет создать веб-приложение, доступное на различных платформах без необходимости разработки отдельного кода для каждой из них.
2. Широкие возможности для фронтенд и бэкенд разработки: JavaScript используется как для разработки клиентской (фронтенд), так и серверной (бэкенд) части приложений. Node.js позволяет строить серверную логику, работая с асинхронными запросами и обеспечивая высокую производительность.
3. Поддержка асинхронности: JavaScript предоставляет мощные механизмы для работы с асинхронными задачами. Это идеально подходит для обработки запросов на сервер и взаимодействия с базами данных, что необходимо для обработки пользователей, событий, и их взаимодействий с системой в реальном времени.

4. Широкая экосистема и поддержка сообществом: JavaScript имеет огромное количество библиотек и фреймворков, что позволяет ускорить процесс разработки. Для создания REST API — Express.js, а для реализации аутентификации — Passport.js.
5. Популярность и доступность разработчиков: JavaScript является одним из самых популярных языков программирования, что означает, что существует большой выбор опытных разработчиков. Это облегчает найм, обучение и поддержку проекта.
6. Высокая производительность на серверной стороне: Node.js, основанный на JavaScript, позволяет обрабатывать тысячи параллельных соединений с минимальными затратами ресурсов, что делает его идеальным для масштабируемых приложений, таких как ваша система для поиска событий в реальном времени.

В результате, JavaScript предлагает гибкость, широкие возможности и поддержку для разработки как клиентской, так и серверной части приложения, что делает его отличным выбором для проекта.

3. Проектно-конструкторская часть

3.1 Логическая схема базы данных:

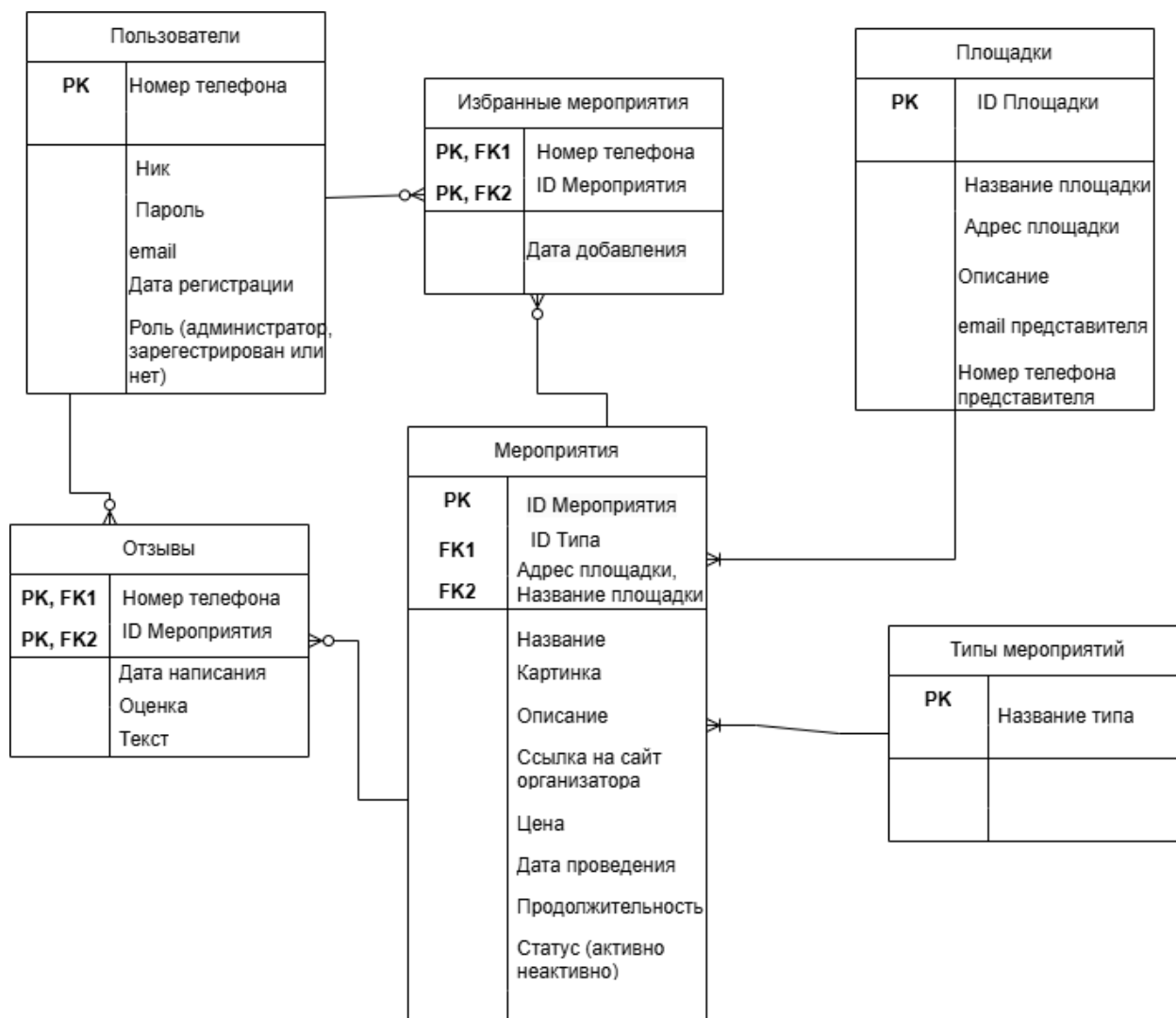


Рисунок 7 – Логическая схема базы данных

3.2 Разработка интерфейса взаимодействия пользователя с системой.

На главной странице (Рисунок – 8) нашего сайта пользователям предоставляются следующие возможности:

- **Регистрация:** Пользователи могут создать новый аккаунт, указав свой ник, номер телефона, адрес электронной почты и пароль. Этот процесс прост и интуитивно понятен, что позволяет легко присоединиться к нашему сообществу.

- Вход в систему: Зарегистрированные пользователи могут войти в свою учетную запись, используя номер телефона и пароль. Это обеспечивает безопасность и защиту личных данных.
- Просмотр без регистрации: У нас есть возможность для пользователей исследовать мероприятия без необходимости создания аккаунта, позволяя им ознакомиться с содержимым сайта.

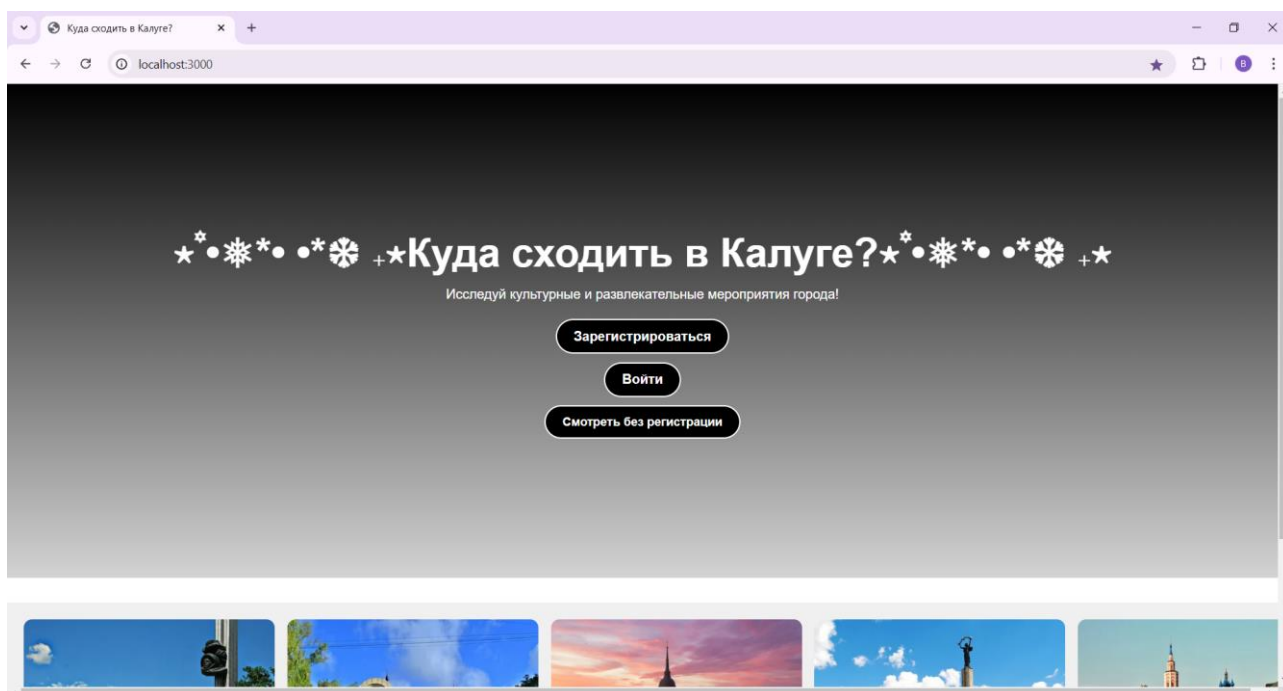


Рисунок 8 – главная страница

При регистрации (Рисунок – 9) на сайте пользователи должны указать:

- Ник:
- Номер телефона
- Email
- Пароль

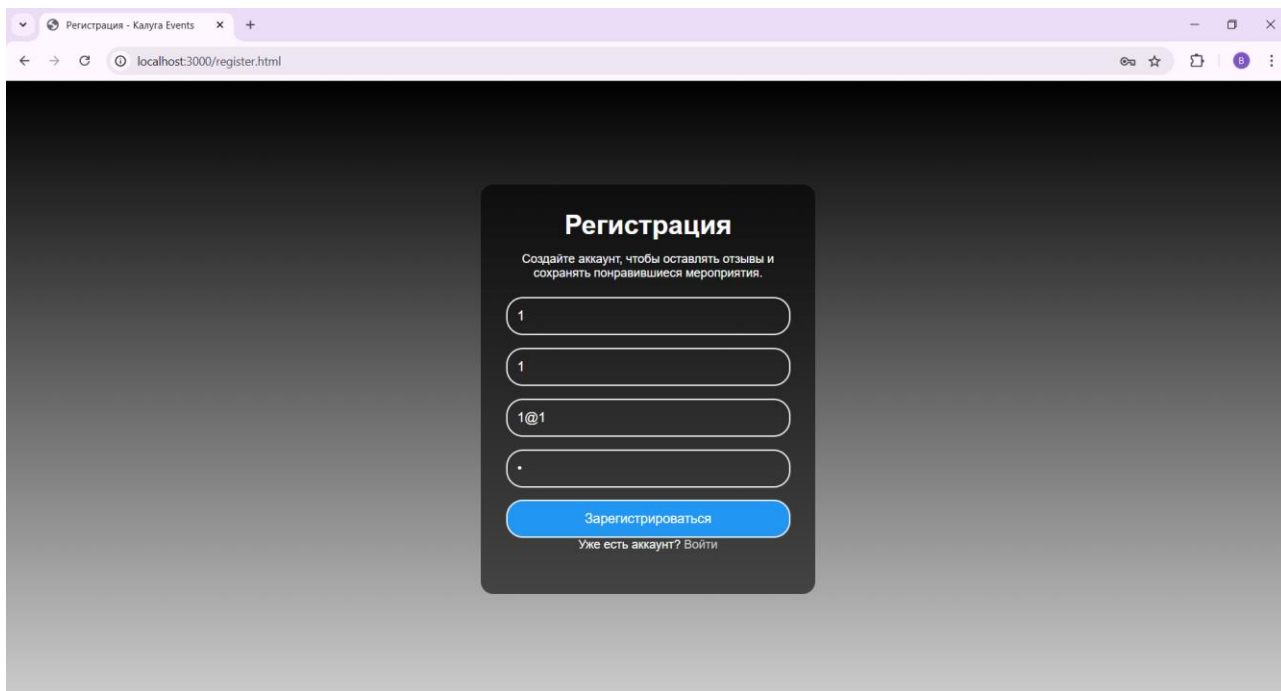


Рисунок 9 – Регистрация пользователя

Для входа (Рисунок – 10) в свою учетную запись пользователям необходимо указать номер телефона и пароль: Это шаг обеспечивает безопасность вашего профиля, позволяя только вам получать доступ к личной информации и предпочтениям.

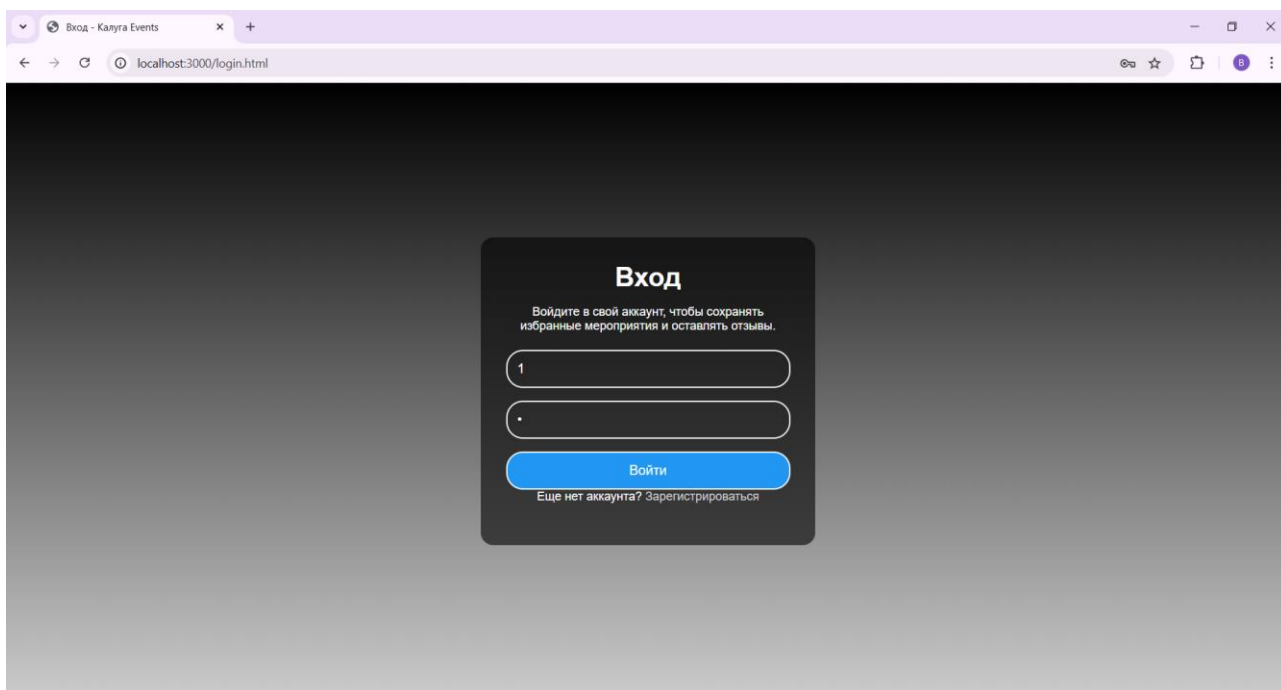


Рисунок 10 – страница входа

На странице мероприятий (Рисунок – 11) пользователи могут:

- Добавление в избранное: пользователь может легко добавлять интересные мероприятия в раздел "Избранные", чтобы к ним можно было вернуться позже.
- Оставление отзывов: пользователи могут делиться своими впечатлениями о мероприятиях, помогая другим пользователям сделать осознанный выбор и улучшая качество контента.

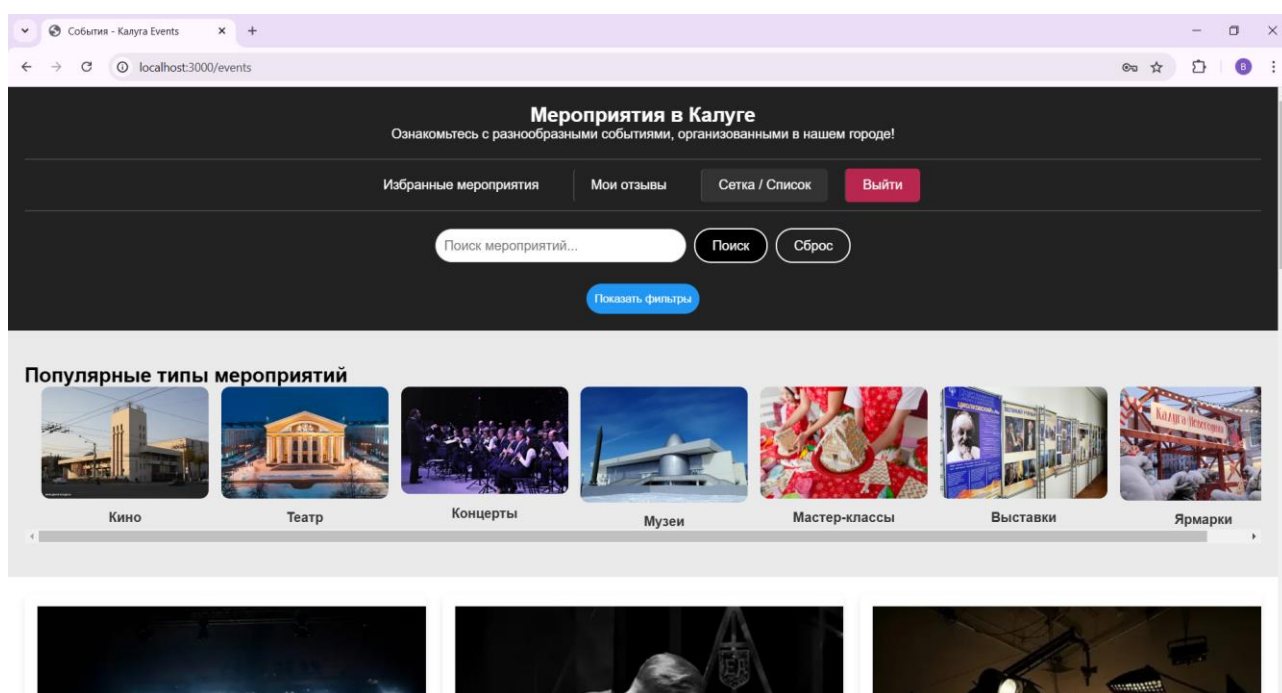


Рисунок 11 – список мероприятий

Просмотр избранных мероприятий: нажав на эту кнопку, пользователь может увидеть полный список всех мероприятий, которые были добавлены в избранные (Рисунок – 12).

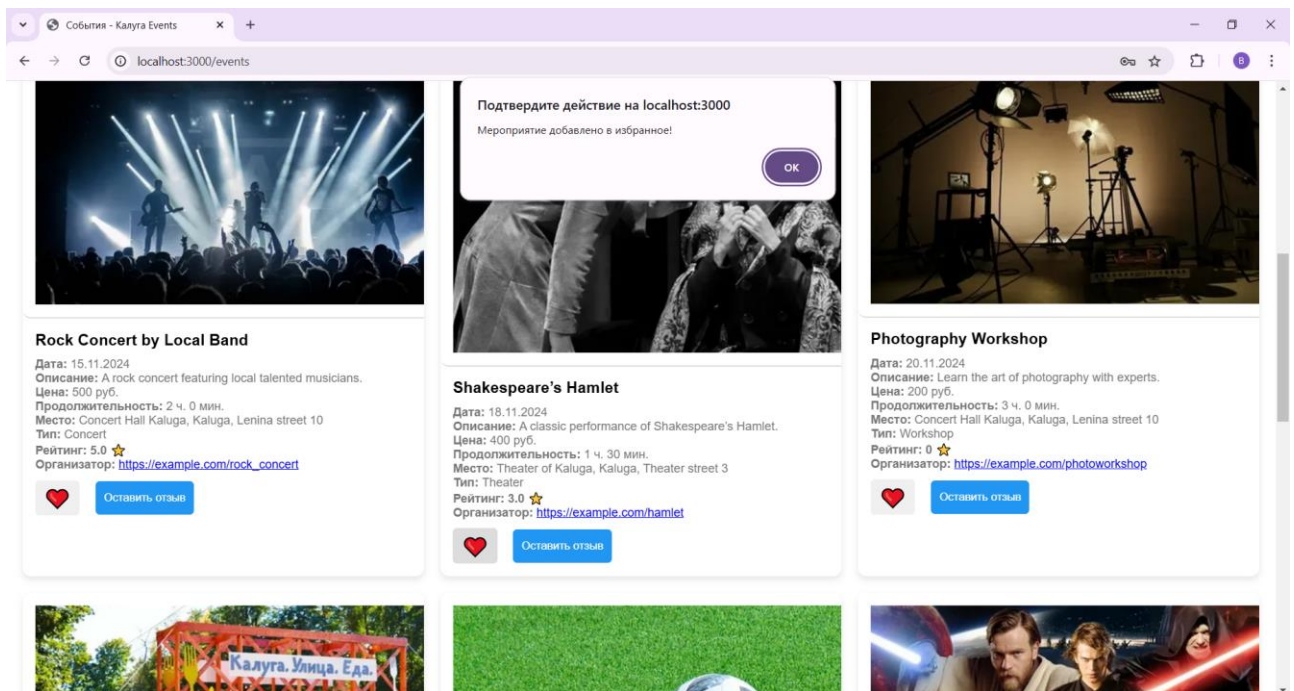


Рисунок 12 – добавление мероприятие в избранное

У каждого пользователя может добавлять мероприятия в свой список избранных мероприятий (Рисунок – 13).

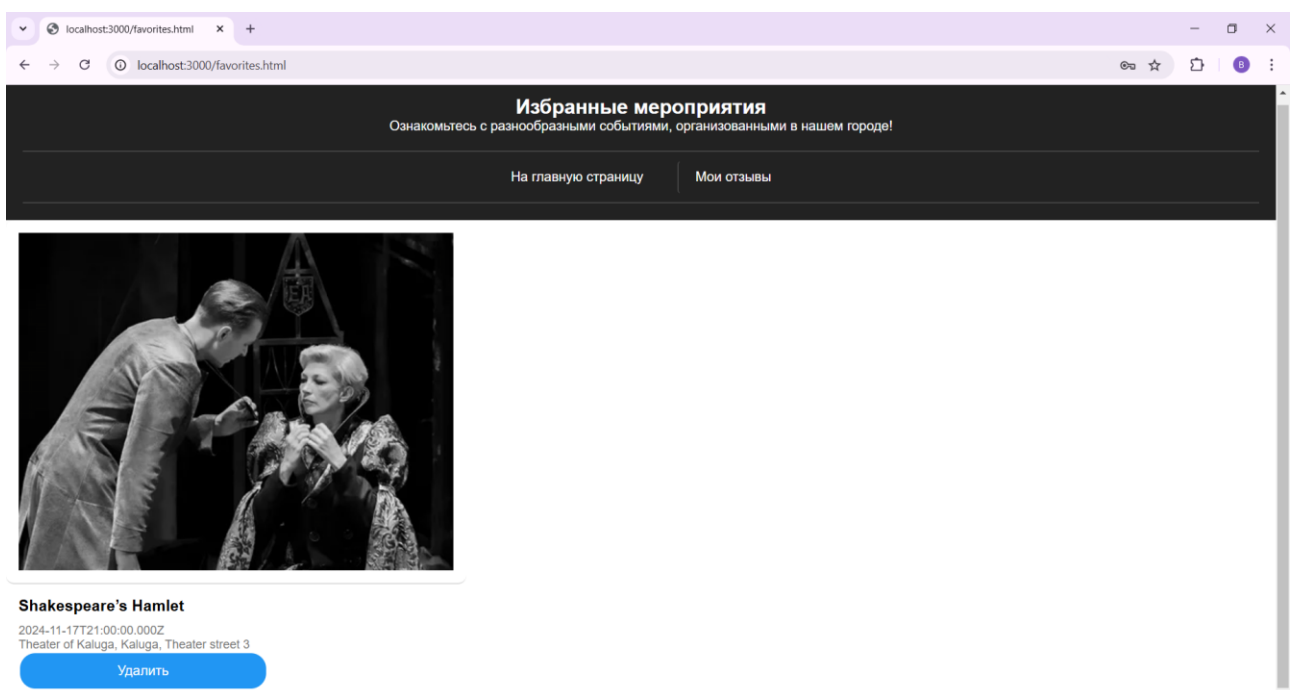


Рисунок 13 – список избранных мероприятий

Для удобства поиска пользователи могут воспользоваться функцией фильтрации:

Пользователи могут настроить параметры поиска по различным критериям, таким как цена, дата или тип мероприятия (Рисунок – 14).

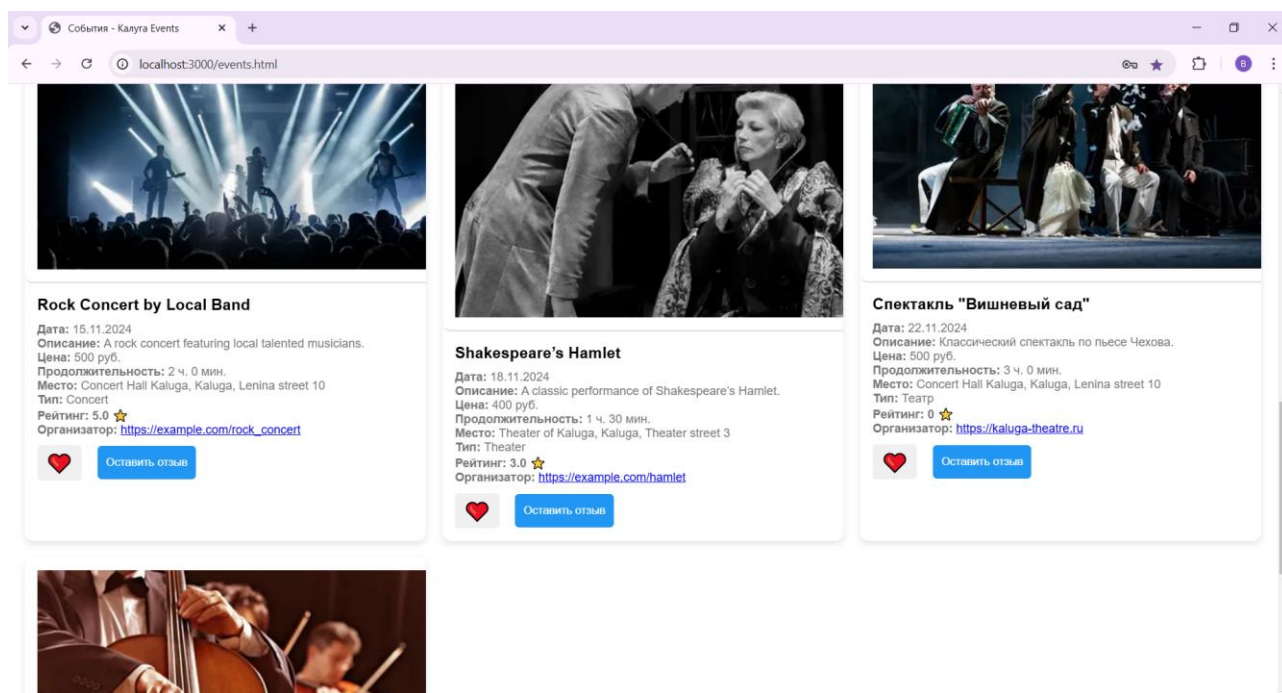


Рисунок 14 – вывод мероприятий, с ценой >400

3.3. Разработка архитектуры приложения

Оптимальным вариантом для реализации данного программного продукта является клиент-серверная архитектура. В этой модели серверная часть отвечает за выполнение бизнес-логики, хранение информации и взаимодействие с базой данных, тогда как клиентская часть (веб-интерфейс) устанавливает связь с конечным пользователем.

Клиентская часть

Это веб-приложение, разработанное с использованием HTML, CSS и JavaScript, а также фреймворков, таких как React или Vue.js. Пользователи взаимодействуют с системой через этот интерфейс, проходя тесты, получая рекомендации и просматривая результаты.

Серверная часть

Для обработки запросов и бизнес-логики тестирования используется Node.js вместе с Express.js, что обеспечивает плавное взаимодействие с базой данных.

База данных

В качестве базы данных выбрана PostgreSQL, которая отвечает за хранение информации о пользователях, тестах, результатах и других элементах системы.

Средства передачи данных

В системе реализован REST API, который обеспечивает взаимодействие между клиентской и серверной частями.

Методы и механизмы передачи информации:

Используется подход REST (Representational State Transfer) для обмена данными между фронтендом и бэкендом. Это позволяет веб-приложению и серверу общаться посредством HTTP-запросов, среди которых:

- GET — для извлечения данных, например, результатов тестирования.
- POST — для отправки данных на сервер, к примеру, для сохранения новых результатов тестов.
- PUT — для обновления информации, например, изменения сведений о пользователе.
- DELETE — для удаления данных, таких как удаление тестов.
- Аутентификация и авторизация

Для контроля доступа к данным и личным кабинетам применяются механизмы аутентификации, такие как JWT (JSON Web Tokens). Пользователям необходимо пройти авторизацию с помощью логина и пароля, после чего они получают токен доступа для дальнейших операций.

3.4 Реализация функционирующего приложения.

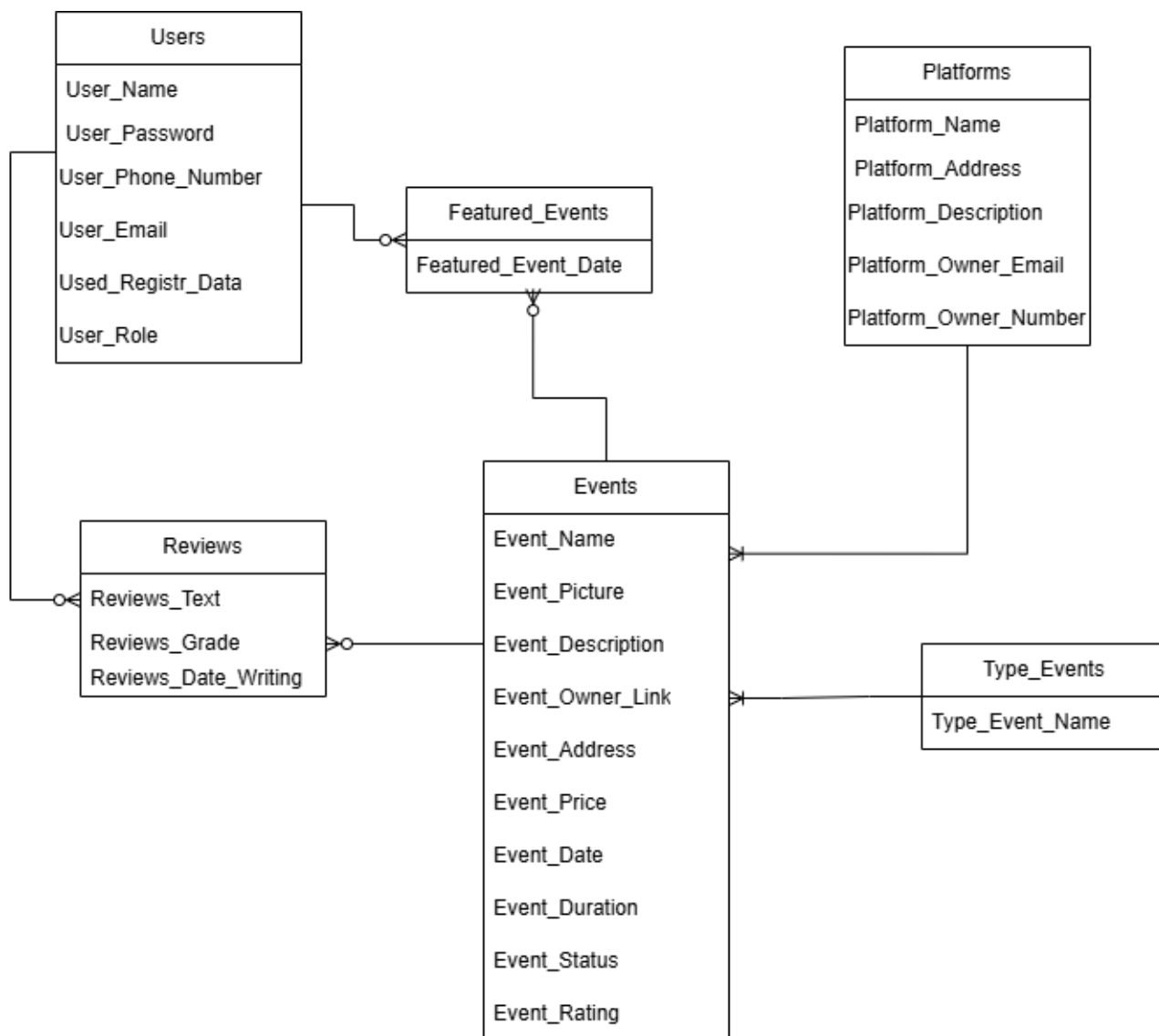


Рисунок 15 – Физическая схема

4. Проектно-технологическая часть

4.1 Порядок развёртывания системы:

Наполняем базу данных данными.

1. Запускаем консоль и переходим в папку с исполняемым файлом командой `cd C:\Пусть_к_исполняемому_файлу`
2. Запускаем серверную часть в консоли строчкой: `nodemon index.js`
3. Открываем поисковик и вводим <http://localhost:3000/>

4.2 Разработка руководства пользователя и администратора

Руководство пользователя представлено в пункте 3.2

Руководство администратора:

Вход и регистрация для администратора. При регистрации нового администратора необходимо указать в графе пароль «20040624» а в графе номер телефона «admin24» (Рисунок – 16). Ник и email не имеют значения и могут быть любыми. Далее необходимо нажать на кнопку «Войти» на странице входа. Требуется определённый пароль и ник.

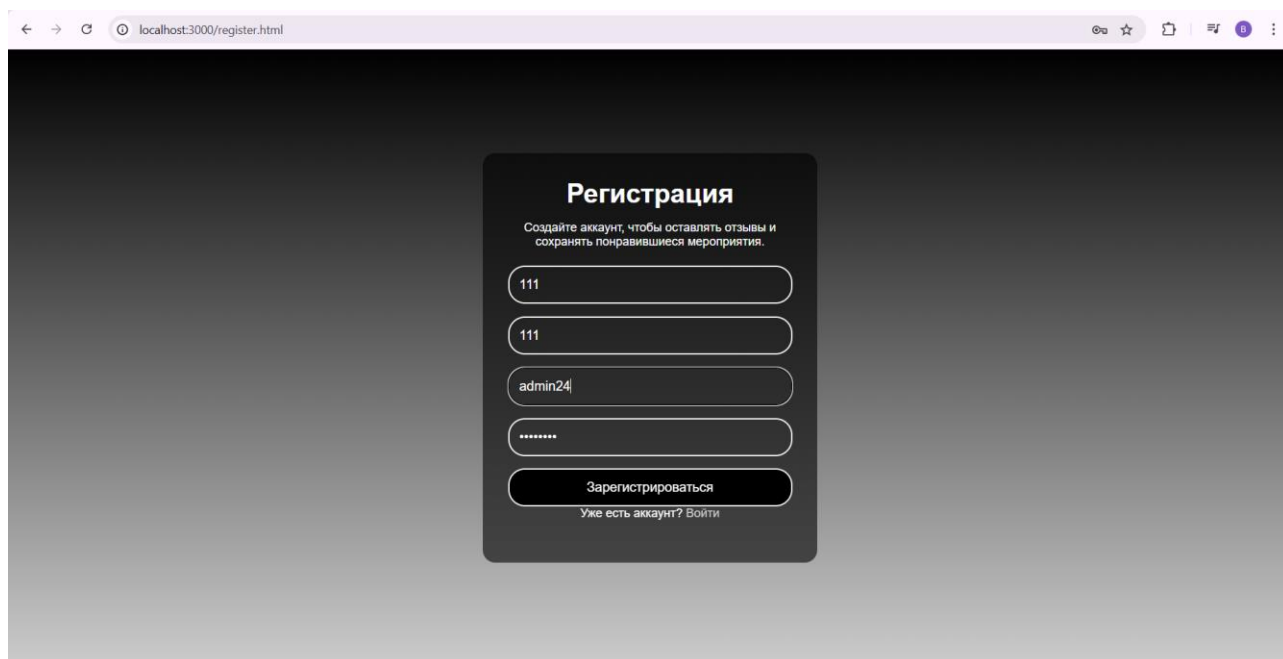


Рисунок 16 – Регистрация администратора

После входа администратор попадает на страницу со всеми мероприятиями. Тут он видит id всех мероприятий, и кнопку «Добавление мероприятия» (Рисунок – 17).

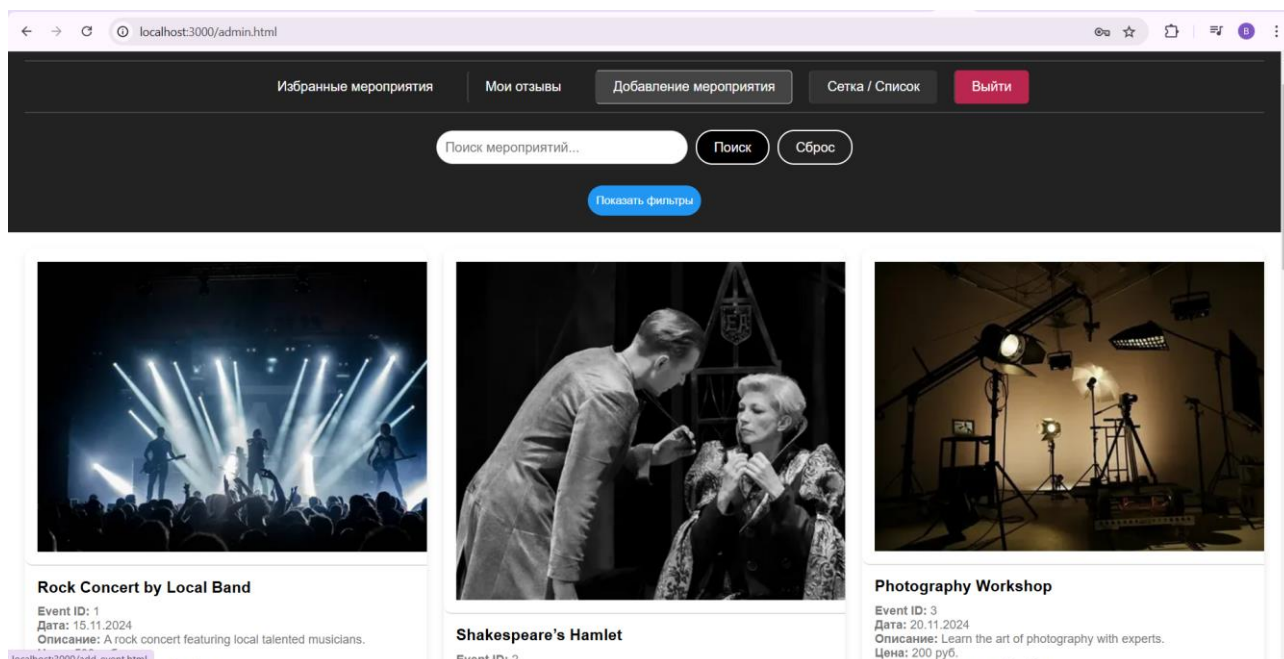


Рисунок 17 – страница мероприятий

Администратор может добавлять мероприятия. Для этого ему нужно нажать на кнопку «Добавление мероприятия» и вести данные в соответствующие поля (Рисунок – 18).

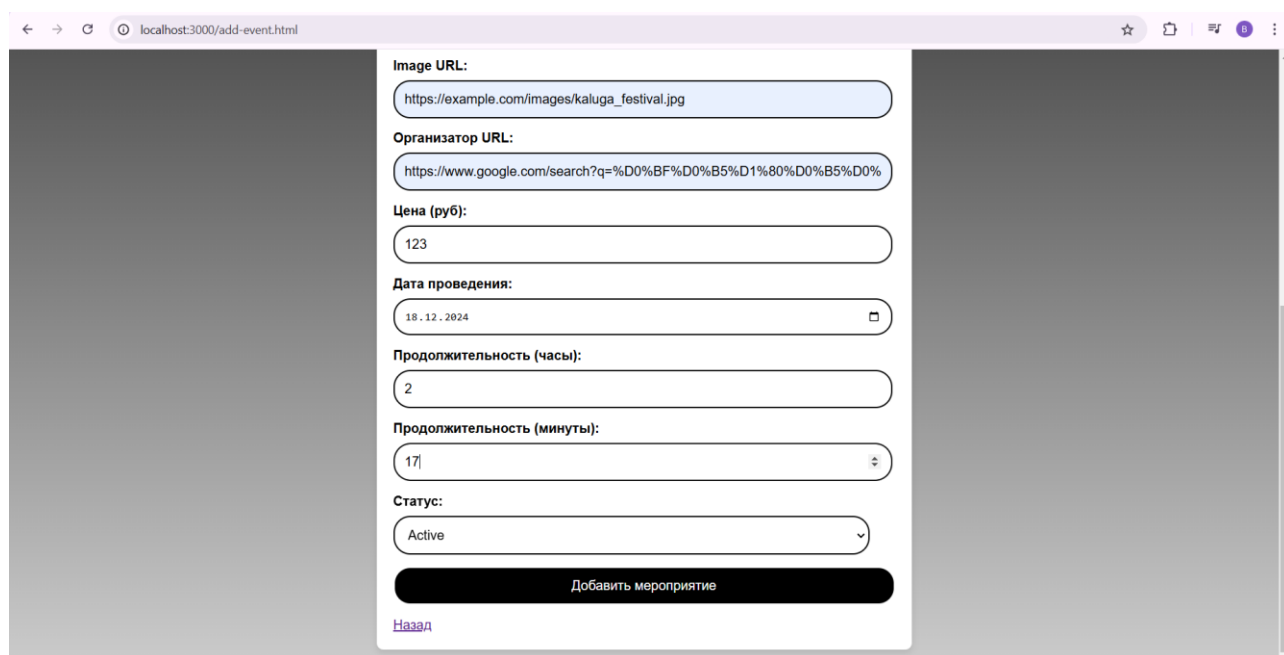
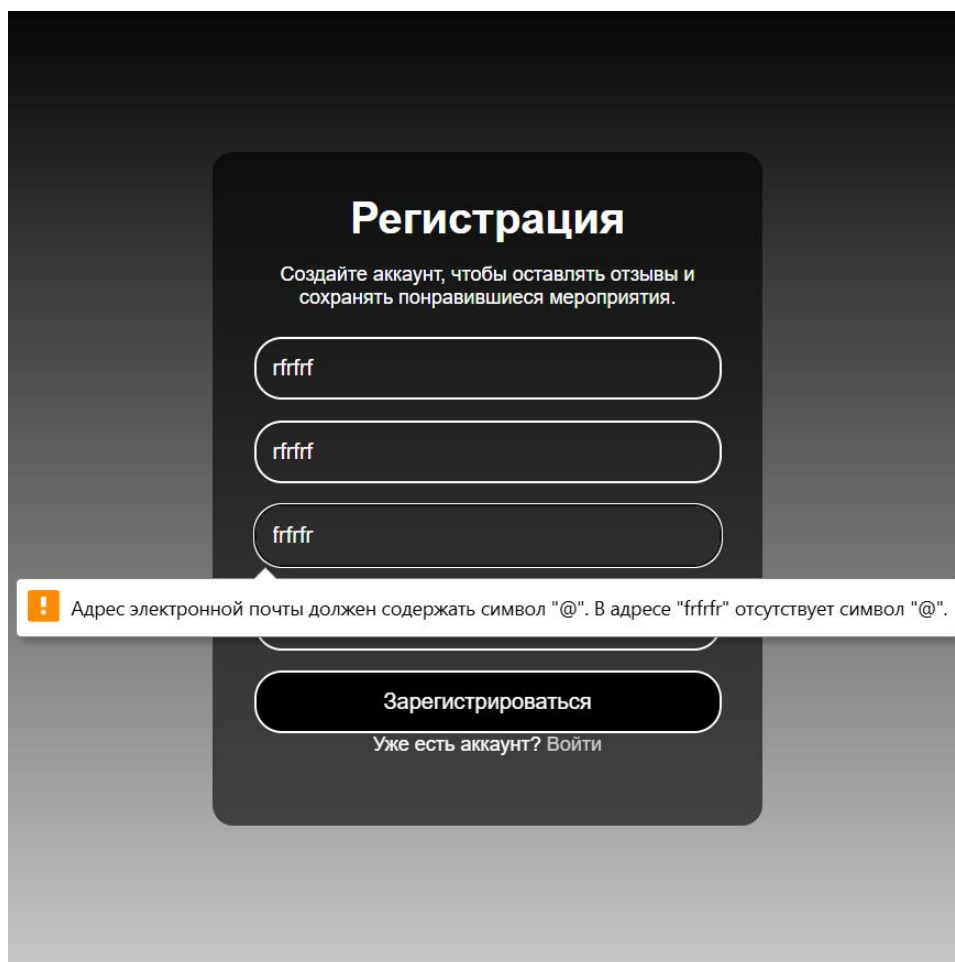


Рисунок 18 – добавление мероприятия

4.3 Тестирование программы реальными данными

Пример ввода некорректный данных

При регистрации, к примеру, укажем email адрес не верно. Тогда система выдаст ошибку и попросит ввести правильный email (Рисунок – 19):



The image shows a registration form titled "Регистрация" (Registration). Below the title is a subtitle: "Создайте аккаунт, чтобы оставлять отзывы и сохранять понравившиеся мероприятия." (Create an account to leave reviews and save liked events). There are three input fields for registration data. The first two fields contain the text "rfrfr". The third field contains "frfrfr". A red error message is displayed below the third field: "Адрес электронной почты должен содержать символ '@'. В адресе 'frfrfr' отсутствует символ '@'." (Email address must contain the symbol '@'. The address 'frfrfr' is missing the symbol '@'). Below the input fields is a button labeled "Зарегистрироваться" (Register). At the bottom of the form, there is a link: "Уже есть аккаунт? Войти" (Already have an account? Log in).

Рисунок 19 – некорректный ввод email при регистрации

Попробуем ни ввести какой-либо атрибут при регистрации и нажать кнопку «зарегистрироваться». Тогда система попросит ввести недостающие данные (Рисунок – 20):

Регистрация

Создайте аккаунт, чтобы оставлять отзывы и сохранять понравившиеся мероприятия.

Никнейм

Номер телефона ! Заполните это поле.

Email

Пароль

Зарегистрироваться

Уже есть аккаунт? [Войти](#)

Рисунок 20 – не введён обязательный атрибут при регистрации

Попробуем оставить отзыв на мероприятие, будучи при этом не зарегистрированными в системе. Оставлять отзывы могут только авторизованные пользователи, поэтому система выдаёт ошибку (Рисунок – 21):

Лекция по искусственному интеллекту

Дата: 25.11.2024

Описание: Лекция о будущем искусственного интеллекта и его влиянии на общество.

Цена: 100 руб.


Продолжительность: 1 ч. 45 мин.

Место: Concert Hall Kaluga, Kaluga, Lenina street 10

Тип: Лекции и семинары

Рейтинг: 0 ★

Организатор: https://example.com/ai_lecture

 **Оставить отзыв**

Оставлять отзывы могут только зарегистрированные пользователи.

Рисунок 21 – не авторизованный пользователь пытается оставить отзыв

Заключение

В ходе выполнения данного проекта была создана система, предназначенная для информирования жителей и гостей города о культурных, развлекательных и спортивных мероприятиях. Приложение автоматизирует процесс поиска информации о событиях, предлагая пользователям удобный доступ к актуальным мероприятиям без необходимости регистрации

Система включает в себя функционал для просмотра мероприятий, организованных по различным категориям, с полными сведениями о времени, месте, стоимости и других атрибутах каждого события. Заранее зарегистрированные пользователи получают дополнительные возможности: они могут оставлять отзывы о мероприятиях и добавлять их в свои избранные списки. Для создания аккаунта пользователю необходимо указать уникальный ник, пароль и номер телефона.

Все мероприятия и их рейтинги формируются на основе отзывов пользователей, что позволяет другим людям получать представление о качестве событий. Администратор системы отвечает за актуальность информации о мероприятиях.

Разработанная система использует современные веб-технологии, такие как HTML, CSS и Node.js, а база данных PostgreSQL обеспечивает надежное хранение информации о мероприятиях и пользователях. С применением клиент-серверной архитектуры приложение предоставляет удобный интерфейс для взаимодействия пользователей через веб-браузеры.

Таким образом, данное приложение представляет собой эффективный инструмент для автоматизации поиска и информирования о культурных событиях города, значительно упрощая процесс получения актуальной информации. В дальнейшем проект может быть расширен дополнительными функциями, такими как улучшение интерфейса и добавление новых типов мероприятий.

Список использованных источников

1. Сакулин С. А. Основы интернет-технологий: html, css, JavaScript, XML: учебное пособие. / С. А. Сакулин. – Москва: МГТУ им. Баумана, 2017. – 112 с. – ISBN 978-5-7038-4724-4. (дата обращения: 14.10.2024).
2. Ревунков Г. И. Проектирование баз данных. – М.: МГТУ им. Н. Э. Баумана, 2009. – 20 с. Режим доступа: <https://e.lanbook.com/book/52390> (дата обращения: 03.10.2024).
3. METANIT.COM. URL: <https://metanit.com/> (дата обращения: 26.09.2024).
4. Node.js – Установка Node.js и использование NVM URL: <https://nodejsdev.ru/guides/freecodecamp/#how-you-can-install-node-js-on-your-system-a-package-manager-the-official-website-installer-or-nvm> (дата обращения: 01.11.2024)