**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ**

**КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное**

**Образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

**Разрешаю**

**Допустить к защите**

**Зав. кафедрой ИСУиА**

**\_\_\_\_\_\_\_\_Воронова Л.И.**

**\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.**

**БАКАЛАВРСКАЯ РАБОТА**

**НА ТЕМУ**

**Разработка web-приложения для автоматизации**

**услуг предприятия культуры**

**Студент:**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Зверев Д.Н.**\_\_

**Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ларин А.И.** \_\_

**Москва 2023 гОбъемное задание**

**ОТЗЫВ  
на выпускную квалификационную работу**

студента 5 курса бакалавриата по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств», профиль «Промышленный интернет вещей и робототехника»,

ЦЗОПБ МТУСИ **Зверева Дмитрия Николаевича**

на тему: **«Разработка web-приложения для автоматизации услуг предприятия культуры»**

Выпускная квалификационная работа студента Зверева Д.Н. выполнена на актуальную тему — исследование в области повышения эффективности управления деятельностью предприятия в направлении автоматизации отдельных бизнес-процессов.

Целью ВКР является исследование и разработка web-приложения для автоматизации услуг предприятия культуры - музея.

В ходе выполнения ВКР автор, используя результаты проведенных исследований, определил требования к web-приложению предприятия культуры, выполнил проектирование базы данных как информационной основы и разработал web-приложение в виде структуры интернет-сайта и фрагментов его программной реализации. На основании анализа результатов проведенного автором тестирования обоснованы выводы о соответствии web-приложения предъявляемым требованиям, а также обосновано снижение стоимости внедрения по сравнению с известными аналогами.

С практической точки зрения, предложенное автором решение позволяет повысить эффективность работы музея с клиентами за счет автоматизации учета и планирования отдельных услуг.

По результатам обучения в бакалавриате компетенции у Зверева Д.Н. сформированы на достаточно высоком уровне. Материал работы изложен аккуратно, доступно, грамотно и логично.

В целом выпускная квалификационная работа соответствует всем требованиям, предъявляемым к ВКР. В качестве замечания отмечаю, что при тестировании предложенного решения автор в недостаточной степени использовал количественные методы оценки.

Считаю, что ВКР на тему «Разработка web-приложения для автоматизации услуг предприятия культуры» выполненная студентом 5 курса ЦЗОПБ МТУСИ по направлению подготовки 15.03.04 — «Автоматизация технологических процессов и производств» Зверевым Дмитрием Николаевичем может быть допущена к защите на ГЭК, а ее автор заслуживает по результатам защиты положительной оценки и присвоения квалификации (степени) «бакалавр».

Научный руководитель:

к.т.н., доцент кафедры ИСУиА А.И.Ларин

**Аннотация**

В выпускной квалификационной работе на актуальную тему «Разработка web-приложения для автоматизации услуг предприятия культуры» представлено исследование в предметной области улучшения качества обслуживания клиентов путем внедрения решений по автоматизации бизнес-процессов в учреждениях культуры. Используя результаты проведенных исследований, автор предложил собственный способ реализации web-сайта предприятия.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, приложения и списка использованных источников, содержащего 31 наименований.

В первой главе проведено исследование актуальности автоматизации отдельных аспектов деятельности предприятий культуры - музеев. На основе полученных результатов сформулированы требования к web-сайту музея.

Во второй главе выполнен сравнительный анализ систем, аналогичных разрабатываемой, помимо этого выбрана и обоснована модель базы данных на физическом и логическом уровнях.

В третьей главе представлены результаты разработки программной реализации web-сайта музея на основе предложенной модели базы данных. Представлены результаты тестирования, демонстрирующие возможности автоматизации отдельных аспектов работы предприятия культуры позволяющие оптимизировать обслуживание клиентов.

ВКР содержит 59 страниц, включая 38 рисунков, 8 таблиц, 7 листингов.

**Содержание**

[Введение 6](#_Toc129787373)

[Глава 1. Анализ предметной области, автоматизации услуг предприятия культуры 8](#_Toc129787374)

[1.1 Исследование проблематики управления клиентами учреждения культуры 8](#_Toc129787375)

[1.2 Обоснование актуальности эффективного web-приложения для управления клиентами предприятия культуры 11](#_Toc129787376)

[1.3. Формирование требований к web-приложению 13](#_Toc129787377)

[Выводы 15](#_Toc129787378)

[Глава 2. Исследование существующих программных решений для управления клиентами 16](#_Toc129787379)

[2.1 Сравнительный анализ существующих решений на рынке 16](#_Toc129787380)

[2.2 Обоснование выбора модели данных, разработка инфологической модели БД 18](#_Toc129787381)

[2.3 Логическое и физическое проектирование БД, нормализация, разработка схемы базы данных 23](#_Toc129787382)

[Выводы 30](#_Toc129787383)

[Глава 3. Разработка программного обеспечения для автоматизации услуг предприятия культуры 32](#_Toc129787384)

[3.1 Реализация БД для автоматизации услуг предприятия культуры 32](#_Toc129787385)

[3.2 Разработка web-приложения для автоматизации услуг предприятия культуры 36](#_Toc129787386)

[3.3 Тестирование web-приложения 40](#_Toc129787387)

[Выводы 41](#_Toc129787388)

[Заключение 43](#_Toc129787389)

[Список использованной литературы 44](#_Toc129787390)

[ПРИЛОЖЕНИЕ А 47](#_Toc129787391)

[ПРИЛОЖЕНИЕ Б 51](#_Toc129787392)

# Введение

На сегодняшнем уровне развития информационных технологий невозможно представить производство или работу какой-либо компании без использования web-сайтов и специального программного обеспечения, обеспечивающих деятельность практически всех сфер организации, начиная корпоративной почтой и заканчивая отделом бухгалтерии. Автоматизация отдельных бизнес-процессов создает условия повышения эффективности работы. Каждый рабочий день учреждения культуры пропускают через себя большой поток посетителей, что в текущих реалиях исчисляется сотнями или даже тысячами. Внедрение существующих или разработка собственных решений по автоматизации бизнес-процессов в учреждениях культуры является актуальной задачей.

Объектом исследования является деятельность предприятия культуры (музея).

Предметом исследования являются методы и способы автоматизации основной деятельности предприятия культуры.

Цель исследования – исследование и разработка web-приложения для автоматизации услуг предприятия культуры.

Для достижения цели работы сформулированы следующие задачи:

1) выполнить исследование в сфере управления предприятиями культуры и обосновать актуальность поставленной цели;

2) исследовать методы и способы внедрения автоматизированных систем на предприятиях культуры;

3)сформировать требования к разрабатываемому web-приложению;

4)выполнить сравнительный анализ существующих решений на рынке направленных на автоматизацию услуг предприятий культуры;

5)произвести проектирование требуемой базы данных на физическом и логическом уровнях;

6) разработать web-приложение на основе спроектированной базы данных;

7) произвести тестирование разработанного web-приложения.

Структура работы включает в себя введение, три главы, заключение и список использованной литературы в количестве 31 источников.

В первой главе проведено исследование актуальности автоматизации отдельных аспектов деятельности предприятий культуры - музеев. На основе полученных результатов сформулированы требования к web сайту музея.

Во второй главе выполнен сравнительный анализ систем, аналогичных разрабатываемой, помимо этого выбрана и обоснована модель базы данных на физическом и логическом уровнях.

В третьей главе представлены результаты разработки программной реализации web сайта музея на основе предложенной модели базы данных. Представлены результаты тестирования, демонстрирующие возможности автоматизации отдельных аспектов работы предприятия культуры позволяющие оптимизировать обслуживание клиентов.

Заключение содержит основные выводы и предложения.

ВКР содержит 59 страниц, включая 38 рисунков, 8 таблиц, 7 листингов.

# Глава 1. Анализ предметной области, автоматизации услуг предприятия культуры

## 1.1 Исследование проблематики управления клиентами учреждения культуры

Предприятие культуры это организация, основной задачей и деятельностью которой является предоставление разнообразных услуг социально-культурного, просветительского, оздоровительного, развлекательного характера, создание условий для развития любительского художественного творчества. Учреждение культуры создается в первую очередь в целях развития традиционного народного художественного творчества, социально-культурной активности, организации досуга и отдыха. Такие учреждения открыты для всех граждан, не зависимо от пола, возраста, национальности, образования, социального положения, политических убеждений, и религиозных верований [1]. К организациям культуры относятся:

1. Театры.
2. Кинотеатры.
3. Концертные организации и коллективы.
4. Цирки.
5. Библиотеки.
6. Музеи.
7. Клубные учреждения.
8. Парки культуры и отдыха.
9. Зоопарки.

В настоящем исследовании в качестве предприятия культуры будет пониматься Музей.

По статистике Минкульта в период с 2018 по 2022 год количество музеев увеличилось на 6%, что представлено на рисунке 1.1. При условии, что тенденция в последующие годы не будет нарушаться, то это число будет увеличиваться [2].

Рисунок 1.1 – Количество открытых музеев с 2018 по 2022 год

В России на 2022 год постоянно проживает 147,2 миллиона жителей, на посещение музеев приходится 95, 4 тысячи человека  за год, что представлено на рисунке 1.2. Получается в среднем в стране человек ходит в музей примерно раз в 16 месяцев. В статистике посещаемости музеев учитываются не только жители России, но и туристы.

Рисунок 1.2 – Количество посещений с 2018 по 2022 год

Анализ данных, представленных на рис. 2-1 показывает, что с 2019 по 2021 год посещаемость всех публичных мест уменьшилась из-за пандемии COVID-19. Посещаемость музеев упала на 45%. Это связано с тем, что публичные места, в том числе и музеи, закрывались на карантин. В 2021 году посещаемость выросла, но осталась ниже допандемийного.

Наиболее активный приток людей в Петербурге и Севастополе: 3,3 и 3 раза в год. Такая большая разница, обусловлена большим наплывом туристов в этих регионах. Выше средней отметки посещаемость на северо-западе европейской части. Информация была взята из Псковской, Ярославской, Калининградской областях. Меньше всего посещают в Карачаево-Черкесии, Дагестане, Тыве, Калмыкии и Еврейской автономной области: от 0,05 до 0,14 посещения в год. То есть среднестатистический житель этих регионов культурно просвещается примерно раз в семь лет [2].

По данным издания «The Art Newspaper» на год, представлен свой топ музеев по всему миру. В списке самых посещаемых культурных учреждений четыре российских музея: Третьяковская галерея (9-е место, 1,6 млн человек), Эрмитаж (6-е место, 1,6 млн человек), Мультимедиа Арт Музей (3-е место, 2,2 млн человек) и Русский музей (2-е место, 2,7 млн человек) [3].

Согласно финансовому отчету по музеям, за 2021 год, за все проведенные выставки и сторонние доходы они получили 17,2 миллиарда рублей. Пример нескольких таких организаций представлен в таблице 1.1 [4].

Таблица 1.1 – Выборка выручек нескольких организаций

|  |  |  |  |
| --- | --- | --- | --- |
| **Организация** | Выручка | активы | Регион |
| ООО Производственно-образовательный центр "ГРАНД МАКЕТ" | 302 млн. руб | 55,5 | Санкт-Петербург |
| АО "Эрарта" | 215 млн. руб | 109 | Санкт-Петербург |
| ООО "К-825" | 184 млн. руб | 390 | Москва |
| ООО "Историко-культурный комплекс "Вятское" имени Е.А. Анкудиновой" | 182 млн. руб | 45,4 | Ярославская область |
| ООО "Научный мир" | 142 млн. руб | 95,1 | Москва |

## 1.2 Обоснование актуальности эффективного web-приложения для управления клиентами предприятия культуры

CRM в переводе с английского языка означает «управление взаимоотношениями с клиентами». Это сервис, программное обеспечение или web-приложение, которое позволяет вести учет, обработку и работу с клиентами. Сегодня без CRM трудно представить успешный бизнес, который ведет свою деятельность в интернете. Крупные холдинги и частные специалисты, организации, связанные с торговлей или сферой услуг практически все имеют опыт взаимодействия с CRM-системами [5].

Наиболее важным показателем успешности любого предприятия связанного с обслуживанием являются посетители. В современных экономических и социальных условиях важно поддерживать уровень удовлетворенности своих потребителей на высоком уровне. Основополагающими факторами удовлетворенности посетителей являются эффективное управление и индивидуальный подход к каждому.

При наличии актуальной и достоверной информации о потребителях, предприятие может повысить свои успехи путем анализа этих потребителей. Такая информация позволяет изменить модель поведения, перевернув ее от неэффективной стратегии, более совершенной и продуктивной.

Анализируя все возможности CRM, можно обнаружить, что система имеет очень много достоинств. К основным преимуществам внедрения CRM-системы в музее можно отнести:

1) возможность своевременно анализировать, собирать, пересылать обрабатываемы данные. На выходе эти данные предоставляются пользователю в удобной и понятной для него форме;

2) возможность централизовать обрабатываемую информацию о процессах и клиентах. Метод позволяет более детально взаимодействовать с клиентами, что хорошо воздействует на предлагаемый компанией продукт/услугу;

3) возможность автоматизировать процессы, которые требуют постоянного повторения;

Внедрение CRM- системы на предприятии культуры создает условия для освобождения рабочего времени персонала, который можно будет переправить для решения других задач [6].

Одним из первых случаев внедрения системы CRM стал Музей современного искусства «MoMA». В 2011 году, в Нью-Йорке было внедрено облачное CRM решение для управления взаимоотношениями с клиентами Salesforce, чтобы помочь отслеживать большое количество данных о своих участниках. С момента его первоначального внедрения в платформу были интегрированы дополнительные данные из других музейных систем, такие как история покупок участника, онлайн-активность и история посещений, что позволяло музею иметь полное понимание о участниках своей программы. Эта интегрированная CRM-платформа позволила музею обеспечить более качественное и своевременное обслуживание клиентов, сохранить и увеличить количество участников и спонсоров, расширить взаимодействие со своими спонсорами и улучшить качество услуг за счет использования мобильных технологий.

Основная цель была достигнута за счет интеграции данных в платформу CRM из множества различных музейных каналов, таких как история посещений и активность в электронной коммерции. Это дало более целостное представление о деятельности участников программы, позволяя просматривать данные о человеке такие как:

1. Подробный просмотр истории пожертвований.
2. Эффективность маркетинговой кампании и отслеживание пожертвований по кампаниям.
3. Взгляд на отношения между участниками.
4. Покупка билетов на специальные мероприятия.
5. Участие в образовательных программах MoMA.
6. История посещений MoMA и MoMA PS1.
7. Розничные сделки.
8. Индивидуальные и совокупные личные предпочтения и интересы.
9. Вложения и подарки произведений искусства, сделанные музею.

Интегрированная платформа CRM также позволила повысить прозрачность внутренних отделов в отношении развития спонсоров или потенциальных клиентов. Стало возможным сотрудничество между отделами по маркетинговым кампаниям и отслеживание эффективности этих кампаний.

Несмотря на возникшие трудности, внедрение CRM-платформы оказалось успешным проектом, хотя и очень долгим. Это позволило музею лучше понять своих посетителей и, следовательно, предложить им лучший опыт и услуги [7].

## 1.3. Формирование требований к web-приложению

Так как музей может быть частным или государственным, бесплатным или платным, то к ним предоставляются разные требования по структуре разрабатываемого web-приложения. Разработать общее для всего сразу не имеет практического смысла. Универсальное решение либо будет слишком дорогим в разработке, либо иметь обобщенный функционал, что не очень скажется на продуктивности внутри предприятия. Для того чтобы web-приложение получилось более качественным нужно сделать его конкретно под свою область. Как основу для дальнейшей разработки и планированию системы был выбран бесплатный музей.

В штатном расписании любого музея имеется большое количество различных специалистов, начиная от уборщицы и заканчивая реставраторами и докторами наук [9]. Для работы непосредственно с посетителями требуются не все работники, а такие как:

Экскурсовод – специалист по проведению экскурсий. Специалист в данной области должен иметь хорошие знания по истории, культуре и экспонатам. Ему необходимо знать принципы организации и методики проведения экскурсий, особенности обустройства выставочных стендов.

Служба безопасности – подразделение,  задачей которого является обеспечение безопасности музейных экспозиций, фондохранилищ, служебных кабинетов сотрудников от пожаров, преступных посягательств, аварий, актов вандализма и стихийных бедствий. К основным задачам службы музейной безопасности относятся круглосуточная охрана Музея, обеспечение безопасности его посетителей, соблюдение строгого пропускного режима, противопожарная профилактика, контроль за правильностью эксплуатации систем жизнеобеспечения.

Администратор – сотрудник музея, осуществляющий надзор за музейной экспозицией. Основная деятельность музейного смотрителя связана с обеспечением порядка при проведении экскурсий, наблюдением за сохранностью экспонатов, консультированием посетителей по содержанию постоянной экспозиции, временных выставок, регистрацией посетителей на экскурсии.

Помимо сотрудников разработанной системой будут пользоваться как посетители, так и различные турагентства. Поэтому программное обеспечение должно выполнять такие задачи как:

Для посетителей:

1. запись на экскурсию;
2. обратная связь;

Для экскурсоводов:

1. просмотр расписания;

Для администратора:

1. просмотр, изменение и добавление мероприятий;
2. просмотр, изменение и добавление заказов;
3. просмотр, изменение и добавление пользователей;
4. просмотр, изменение и добавление мероприятий в расписание;
5. просмотр аналитики;

Для Турагенств:

1. возможность проектирования собственных мероприятий;
2. возможность добавления целых групп посетителей;

## Выводы

В результате проведенного исследования современного состояния обслуживания клиентов в учреждениях культуры установлено, что имеет место положительная ежегодная динамика роста количества людей, которые посещают музеи. Несмотря на резкий спад в 2019 году в связи невозможности посещать общественные места, тенденция начала нормализироваться к 2021 году и пошла в положительное русло и к 2022 году их общее кол-во в РФ составило 95 миллионов человек, что ненамного приблизило их к числам, которые были до пандемии.

Часть услуг предоставляемых музеями является платной. В связи с устойчивым ростом количества посетителей музеев растет и их доход. В связи с этим фиксируется увеличение объемов продаж услуг и составляет примерно 17,2  миллиарда рублей. Вместе с тем, выполненный анализ рынка услуг показывает, что имеются проблемы, основными из которых является клиентоориентированность учреждений культуры. В связи с большим наплывом посетителей учреждениям становится трудней обрабатывать их и при этом обеспечивать высокое качество обслуживания.

Основными методами и способами снижения негативного влияния на дальнейшее развитие рынка музейных услуг являются средства автоматизации и централизации предоставляемых услуг. В результате проведенных исследований установлено, что разработка способов упрощения и автоматизирования методов внутри организации является актуальной задачей.

# Глава 2. Исследование существующих программных решений для управления клиентами

## 2.1 Сравнительный анализ существующих решений на рынке

Для того чтобы сравнить и выяснить какие программные решения есть на рынке, нужно определить какими web-приложениями или ресурсами пользуются музеи. Для анализа был взят топ 10 популярных российских музеев за 2021 год, по мнению газеты The art newspaper Russia и представлен в таблице 2.1 [3].

Таблица 2.1 – Топ 10 популярных музеев в России.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название музея | Количество посещений | Город | Используемая CMS | Продажа билетов |
| Петергоф | 3300000 | Санк-Петербург | Самописный на фраемворке | Infotec |
| Казанский кремль | 3029068 | Казань | Wordpress | Infotec |
| Исаковский собор | 2581523 | Санк-Петербург | Самописный на фраемворке | Infotec |
| Русский музей | 2260231 | Санк-Петербург | bitrix | Infotec |
| Мультимедиа арт-музей | 2242405 | Москва | bitrix | Самописная |
| Сталинградская битва | 2188947 | Волгоград | bitrix | tn-cloud |
| Царское село | 1857585 | Санк-Петербург | Drupal | Infotec |
| Эрмитаж | 1649443 | Санк-Петербург | Wordpress | Infotec |
| Третьяковская галерея | 1580819 | Москва | bitrix | Infotec |
| Государственный Владимиро-Суздальский историко-архитектурный | 1162060 | Владимир | Bitrix | Infotec |

Из таблицы 2.1 видно, что большинство сайтов музеев использует за основу систему «bitrix», а для продажи билетов и взаимодействия с клиентами систему от «infotec».

Bitrix это программный комплекс, предназначенный для оптимизации ведения бизнеса, контролем производственных процессов. Основными его функциями являются:

1.Создание корпоративной сети компании.

2.Контроль работы организации.

3.Хранилище для файлов.

4.Конструктор для создания сайтов.

5.Приложение для осуществления звонков по IP-телефонии.

Система использует подписочную систему оплаты, где самый дешевый тариф предоставляется бесплатно, а самый дорогой стоит 11190 рублей в месяц [9].

Для продажи билетов музеи в основном используют решения от компании «infotec» под названием TicketNet.Музей**.** С**истема предназначена для автоматизации деятельности музея**, связанной с обслуживанием посетителей, включая:

1.Планирование.

2.Прием заказов, формирование расписаний, резервирование услуг.

3.Ведение договоров и баланса клиентов.

4.Продажу билетов в кассах.

5.Финансовый, бухгалтерский и организационный учет.

6.Мониторинг и статистический анализ.

TicketNet.Музей так же использует подписочную систему оплаты, где обеспечение как минимум для одного кассира обойдется компании в 9600р в месяц плюс 5% от продаж, что в год составляет 115 200р, не считая процент с дохода [10].

Основное достоинство систем «infotec» и TicketNet.Музей заключается в постоянном обновлении и поддержке от производителя, тем самым повышается работоспособность и устойчивость решения. Еще одним достоинством является возможность оплаты разными методами, что упрощает вывод денежных средств и ведения бухгалтерии. Поимо прочего предоставляет возможность интеграции с пушкинской картой. Эта возможность создает дополнительную привлекательность организации для студентов, тем самым привлекая новых посетителей.

Основным недостатком таких систем является отсутствие масштабируемости и сложности с настройкой под конкретные потребности предприятий. Изменить настройки или добавить функционал для каждого конкретного пользователя проблематично. Кроме того, у рассматриваемых систем существует платная подписка. Так же возможны проблемы, с внедрением системы на предприятии, так как основной функционал может не подходить под требования, которые могут быть специфичными для конкретного предприятия культуры.

Еще одним недостатком можно считать то, что потенциальным нарушителям информационной безопасности (хакерам) давно известны такие системы, их структура и уязвимости для сетевых атак, что создает значительные информационные риски, которые могут привести к раскрытию персональных данных посетителей.

## 2.2 Обоснование выбора модели данных, разработка инфологической модели БД

База данных это совокупность структурированных данных, относящихся к определенной области. Под предметной областью понимается определенная информация, которая представлена в базе данных. Система управления базами данных это комплекс программных средств, предназначенных для создания и сопровождения базы данных. Существует три основных типа моделей данных, таких как иерархическая, сетевая и реляционная [11].

Иерархическая модель данных представляет собой определенное количество данных, расположенных в порядке их подчинения. По своей сути образующих по структуре топологию типа «Дерево» , продемонстрированная на рисунке 2.1. К основным понятиям иерархической модели данных относятся уровень, узел и связь. Узел это совокупность атрибутов данных, описывающих информационный объект [12].

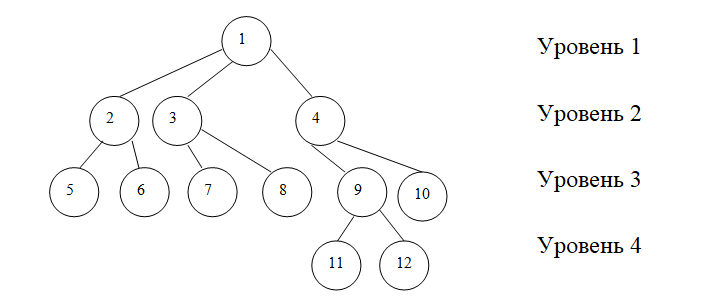


Рисунок 2.1 - Иерархическая структура данных

Иерархическая структура придерживается следующих принципов:

1) каждый узел на более низком уровне связан только с одним узлом,  
находящимся на более высоком уровне;

2) существует только один корневой узел на самом верхнем уровне, не  
подчиненный никакому другому узлу;

3) к каждому узлу существует ровно один путь от корневого узла;

Сетевая модель данных основана на похожих понятиях, что и иерархическая. Отличается она только тем, что каждый узел  
может быть связан с любым другим узлом. На рисунке 2.2 изображена сетевая структура организации данных [13].

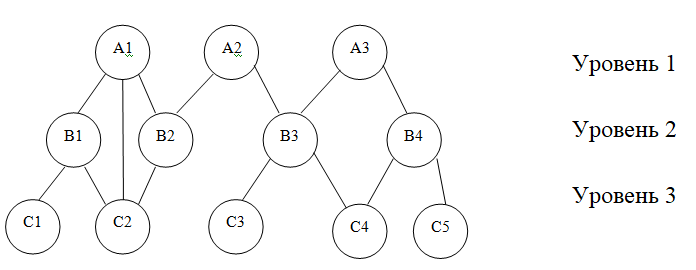


Рисунок 2.2 - Сетевая структура данных

Реляционная модель данных использует организацию данных в виде  
двумерных таблиц. Каждая такая таблица представляет собой двумерный массив и обладает следующими свойствами.

1) элементы в одном столбце имеют одинаковый тип и максимально допустимый размер;

2) каждый столбец имеет уникальное имя;

3) одинаковые строки в таблице отсутствуют;

4) порядок следования строк и столбцов в таблице не имеет значения;

Основными структурными элементами реляционной таблицы являются  
поле и запись, изображенные на рисунке 2.3. Поле (столбец реляционной таблицы) это элементарная единица логической организации данных, которая соответствует конкретному атрибуту информационного объекта. Запись (строка реляционной таблицы) это совокупность логически связанных полей, соответствующая конкретному экземпляру информационного объекта [14].

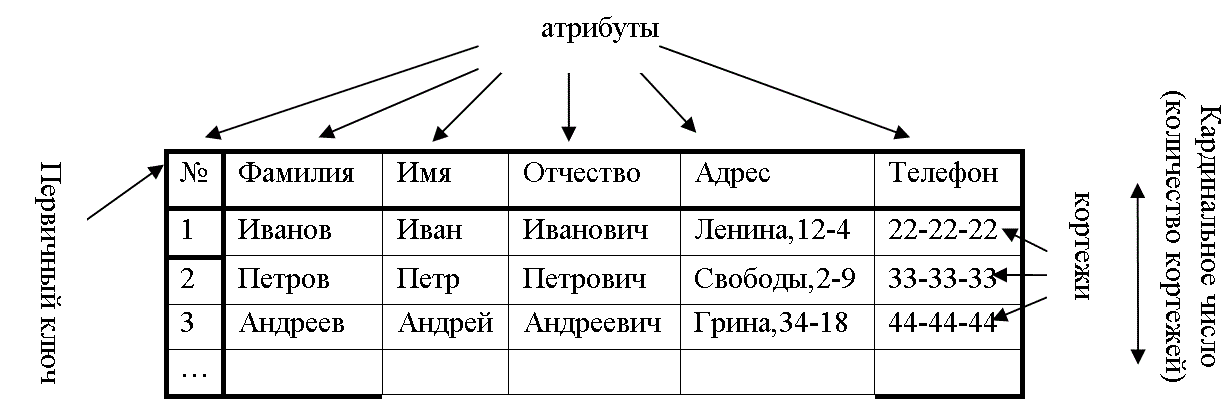


Рисунок 2.3 - Основные структурные элементы реляционной таблицы

Из всех вышеперечисленных моделей данных нам подходит реляционная модель, так как имеет много плюсов по сравнению с другими моделями.

1) Модель образует таблицу, которая наглядна и понятна обычному пользователю;

2) Модель позволяет корректно работать с типами данных, указывая жестко тип входящих и исходящих данных, а так же длину этих данных;

3) Модель позволяет избавиться от повторений;

4) При проектировании базы данных не нужно учитывать порядок следования строк и столбцов в таблице. К любой строке мы можем обратиться по ключу;

Целью инфологического проектирования является создание структурированной информационной модели предметной области, для которой будет разрабатываться база данных [15]. При проектировании на инфологическом уровне создается информационно-логическая модель, которая должна отвечать следующим требованиям:

1) обеспечение наиболее естественных для человека способов сбора и предоставления той информации, которую предполагается хранить в создаваемой базе данных;

2) корректность схемы БД (Адекватное отображение моделированной ПО);

3) простота и удобство использования на следующих этапах проектирования, то есть информационно-логическая модель может легко отображаться на модели базы данных, которые поддерживаются известным СУБД (Сетевые, иерархические, реляционные и др.);

4) информационно-логическая модель должна быть описана языком, понятным проектировщикам баз данных, программистам, администратору и будущим пользователям;

Суть инфологического моделирования состоит в выделении сущностей, которые, подлежат хранению в базе данных, а также в определении характеристик объектов и взаимосвязей между ними. Для информационной системы на основе проведенного системного анализа предметной области выделены следующие сущности [14]:

1. Пользователь, сущность содержит информацию о работниках, работающих в магазине.

2. Мероприятие, сущность содержит информацию о проводимых мероприятиях.

3. Заказ, сущность содержит информацию о записях на экскурсии.

4. Расписание, сущность содержит информацию о времени и дате проводимых мероприятий.

5. Турагентства, сущность содержит информацию о турагентстве которое, сотрудничает с организацией.

6. Экскурсоводы, сущность содержит информацию о ведущем мероприятие экскурсоводе;

Исходя из приведенных выше сущностей, построена инфологическая модель предметной области, которая представлена на рисунке 2.4.

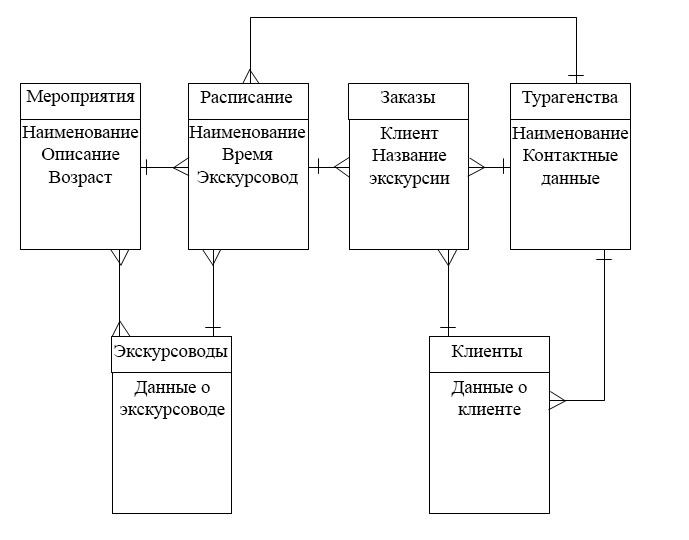


Рисунок 2.4 – Инфологическая модель предметной области

## 2.3 Логическое и физическое проектирование БД, нормализация, разработка схемы базы данных

Для того чтобы грамотно спроектировать структуру базы данных нужно провести нормализацию [16]. Нормализация это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости. На первоначальном этапе база данных будет иметь таблицы:

**Type of events** (типы мероприятий) - таблица содержит всю информацию о типах мероприятий и включает такие параметры как:

1. Aйди (id) – первичный ключ, уникальный адрес строки в базе. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

2. Тип (type) – столбик отвечает за хранение типа экскурсий. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

3. Активность (activity) – столбик отвечает за хранение значения о настройках отображения. Все значения в столбике будут типа integer с максимальной длинной в 1 символ.

Указанные выше параметры приведены в вид нормали 3НФ и отображены в таблице под номером 2.2.

Таблица 2.2 – Типы мероприятий с примером

|  |  |  |
| --- | --- | --- |
| Id | type | activity |
| 1 | Экскурсия | 1 |

**Event\_themes** (мероприятия) – таблица содержит всю информацию о проводимых мероприятиях и включает такие параметры как:

1. Айди (id) – первичный ключ, уникальный адрес строки в базе. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

2. Наименование (name) – столбик отвечает за хранение наименования экскурсий. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

3. Дней до скрытия (days to show) – столбик отвечает за хранение информации об автоматическом скрытие через определенное количество времени. Все значения в столбике будут типа integer с максимальной длинной в 2 символа.

4. Тип мероприятия (type id) – столбик отвечает за хранение информации о ссылке на таблицу «type of events» и содержит в себе id значения. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

5. Описание (description) – столбик отвечает за хранение объемного текста. Все значения в столбике будут типа text.

6. Возраст (age) – столбик хранит в себе минимальный возраст для прохода на экскурсию. Все значения в столбике будут типа integer с максимальной длинной в 2 символа.

7. Продолжительность (duration) – столбик хранит в себе время продолжительности экскурсии. Все значения в столбике будут типа varchar с максимальной длинной в 255 символа.

8. Количество мест (number of seats) – столбик хранит в себе информацию о количестве доступных мест. Все значения в столбике будут типа integer с максимальной длинной в 1 символ.

9. Изображение (picture) – столбик хранит в себе информацию о ссылке на изображение. Все значения в столбике будут типа varchar с максимальной длинной в 255 символа.

10. Активность (activity) – столбик отвечает за хранение значения о настройках отображения. Все значения в столбике будут типа integer с максимальной длинной в 1 символ.

11. Время активности (activity time) – столбик отвечает за хранение информации о длительности отображения. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

12. Время создания (event create time) – столбик отвечает за хранение информации о дате создания мероприятия. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

13. Айди пользователя (user id) – столбик отвечает за хранение информации о ссылке на таблицу «users» и содержит в себе id пользователя.

Указанные выше параметры приведены в вид нормали 3НФ и отображены в таблице под номером 2.3.

Таблица 2.3 – Таблица «Мероприятия» приведенная в вид нормали 3НФ.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Days  to  show | Type  id | Description | Age | Duration |
| 1 | ВИП | 7 | 4 | ВИП экскурсии | 7 | 60 |

Продолжение таблицы 2.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number  of  seats | Picture | Activity | Activity  time | Event  create  time | User  id |
| 50 | NULL | 1 | 1631544405 | 1631519831 | 1 |

**Event\_time** (расписание) – таблица содержит всю информацию о расписании и включает такие параметры как:

1. Айди (id) – первичный ключ, уникальный адрес строки в базе. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

2. Наименование (event themes id) – столбик отвечает за хранение информации о ссылке на таблицу «event themes» и содержит в себе id мероприятия.

3. Время (time) – столбик отвечает за хранение значения о дате и времени начала мероприятия. Все значения в столбике будут типа integer с максимальной длинной в 11 символ.

4. Количество заказов (orders count) – столбик отвечает за хранение значения о текущем количестве записей. Все значения в столбике будут типа integer с максимальной длинной в 11 символ.

5. Активность (activity) – столбик отвечает за хранение значения о настройках отображения. Все значения в столбике будут типа integer с максимальной длинной в 1 символ.

6. Время активности (activity time) – столбик отвечает за хранение информации о длительности отображения. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

7. Время создания (event create time) – столбик отвечает за хранение информации о дате создания мероприятия. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

8. Айди пользователя (user id) – столбик отвечает за хранение информации о ссылке на таблицу «users» и содержит в себе id пользователя.

Указанные выше параметры приведены в вид нормали 3НФ и отображены в таблице под номером 2.4.

Таблица 2.4 – Таблица «Расписание» приведенная в вид нормали 3НФ.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Event  themes  id | Time | Orders  count | Activity | Activity  Time | Create  time | User  id |
| 1 | 21 | 1632 | 5 | 1 | NULL | 1631266539 | 1 |

**Orders** (заказы) – таблица содержит всю информацию о текущих заказах с данными заказавших и включает такие параметры как:

1. Айди (id) – первичный ключ, уникальный адрес строки в базе. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

2. Наименование (event time id) – столбик отвечает за хранение наименования экскурсий. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

3. ФИО (full name) – столбик отвечает за хранение информации о имени покупателя. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

4. Дата рождения (date of birth) – столбик отвечает за хранение информации о имени покупателя. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

5. Номер телефона (telephone number) – столбик отвечает за хранение информации о телефоне покупателя. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

6. Почта (mail) – столбик отвечает за хранение информации о электронной почте покупателя. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

7. Паспорт (passport) – столбик отвечает за хранение информации о паспортных данных покупателя. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

8. Время заказа (order time) – столбик отвечает за хранение информации о времени когда был сделан заказ. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

9. Отметка о посещении (visit) – столбик отвечает за хранение информации о явки на мероприятие. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

10. Время визита (visit time) – столбик отвечает за хранение информации о времени прихода посетителя. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

11. Айди пользователя (user id) – столбик отвечает за хранение информации о ссылке на таблицу «users» и содержит в себе id пользователя.

12. Статус почты (mail status) – столбик отвечает за хранение информации о отправке письма приглашения пользователю. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

13. Активность (activity) – столбик отвечает за хранение значения о настройках отображения. Все значения в столбике будут типа integer с максимальной длинной в 1 символ.

14. Время активности (activity time) – столбик отвечает за хранение информации о длительности отображения. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

Указанные выше параметры приведены в вид нормали 3НФ и отображены в таблице под номером 2.5.

Таблица 2.5 – Таблица «Заказы» приведенная в вид нормали 3НФ.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Event  time  id | Full  name | Date  of  birth | Telephone  number | mail | passport |
| 5 | 9 | Науменко Екатерина Григорьевна | 484689600 | +7 (968) 975 0505 | [fawzi@mail.ru](mailto:fawzi@mail.ru) | NULL |

Продолжение таблицы 2.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Order  time | visit | Visit  time | User  id | Mail  status | Activity | Activity  time |
| 1631440409 | 0 | 1631520861 | 2 | 0 | 1 | 1631517674 |

**Type of users** (типы пользователей) – таблица содержит всю информацию о типах пользователей и включает такие параметры как:

1. Айди (id) – первичный ключ, уникальный адрес строки в базе. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

2. Имя (name) – столбик отвечает за хранение значения о названии предоставляемых прав. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

3. Права (rights) – столбик отвечает за хранение значения о самих правах доступа. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

Указанные выше параметры приведены в вид нормали 3НФ и отображены в таблице под номером 2.6.

Таблица 2.6 – Типы пользователей с примером

|  |  |  |
| --- | --- | --- |
| Id | name | rights |
| 1 | admin | full |

**Users** (пользователи) – таблица содержит всю информацию о типах пользователей и включает такие параметры как:

1. Айди (id) – первичный ключ, уникальный адрес строки в базе. Все значения в столбике будут типа integer с максимальной длинной в 11 символов.

2. Имя (login) – столбик отвечает за хранение значения о названии предоставляемых прав. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

3. Права (pass) – столбик отвечает за хранение значения о самих правах доступа. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

4. Права (username) – столбик отвечает за хранение значения о самих правах доступа. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

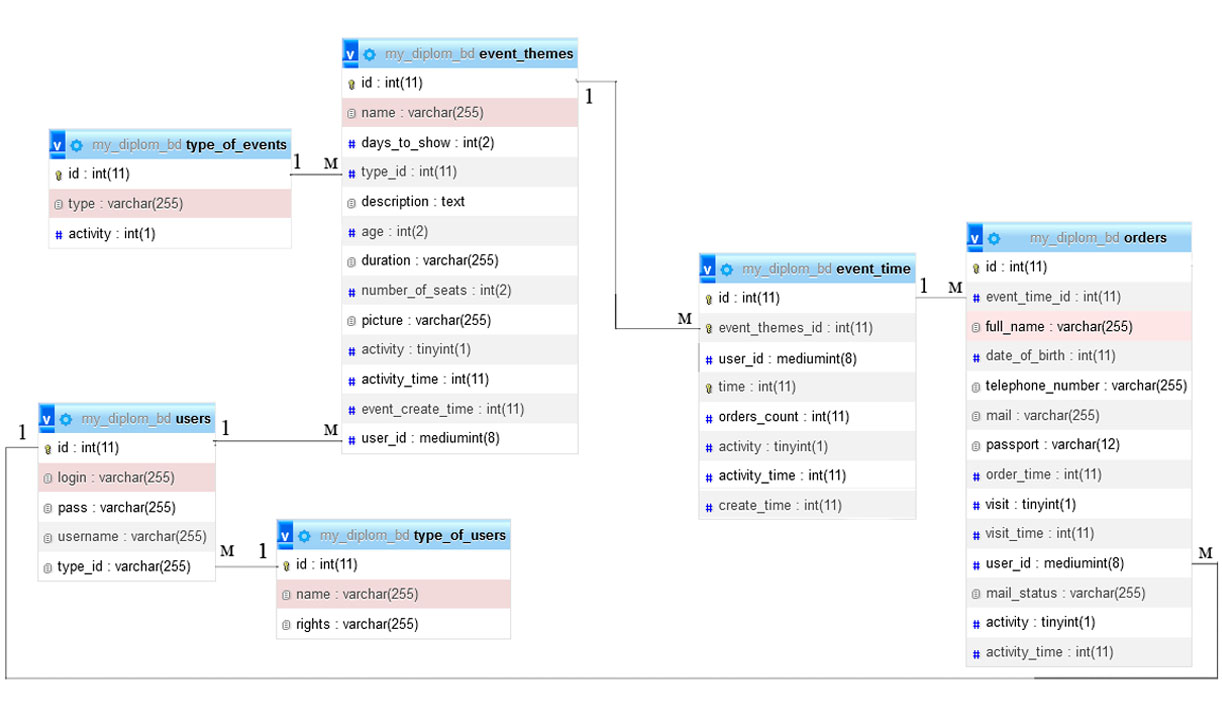
5. Права (type id) – столбик отвечает за хранение значения о самих правах доступа. Все значения в столбике будут типа varchar с максимальной длинной в 255 символов.

Указанные выше параметры приведены в вид нормали 3НФ и отображены в таблице под номером 2.7.

Таблица 2.7 – Заказы с примером

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | Login | pass | username | Type id |
| 1 | Admin | 1234 | admin | 1 |

Указанные выше таблицы организованны в даталогическую схему базы данных и отображены на рисунке 2.5 [17].

 Рисунок 2.5 – Даталогическая схема БД

## Выводы

В результате выполненного сравнительного анализа известных веб-решений в сфере оказания услуг клиентам музеев установлено, что имеющиеся решения в целом обеспечивают выполнение задач, но отмечаются существенные недостатки, главными из которых является дороговизна предлагаемых решений, отсутствие масштабируемости по функционалу, а также сложности с адаптацией под конкретные требования отдельного предприятия культуры.

В результате проведенных исследований установлено, что разработка собственного web-приложения является наиболее эффективным способом для повышения доходов и уменьшения расходов. Так подписка на модуль занимающийся продажами как минимум для 1 кассира обойдется компании в 9600р в месяц плюс 5% от продаж, что в год составляет 115 200р, не считая процент с дохода.

Помимо цены существенный недостаток заключается в том, что стороннее программное решение разрабатывалось усредненным, чтобы подходить под любую организационную систему. Подобный подход не всем подходит, так как требования конкретных предприятий культуры (музеев) достаточно сложно унифицировать и поэтому может возникнуть потребность в разработке дополнительных модулей.

# Глава 3. Разработка программного обеспечения для автоматизации услуг предприятия культуры

## 3.1 Реализация БД для автоматизации услуг предприятия культуры

Для реализации базы данных, в которой будет храниться вся информация, связанная с web-приложением стала СУБД «MariaDB» [18]. Она одна из самых популярных реляционных баз данных с открытым исходным кодом, создана первоначальными разработчиками MySQL и в дальнейшем останется с открытым исходным кодом. Она является частью большинства облачных предложений и используется по умолчанию в большинстве дистрибутивов Linux [19].

При работе с СУБД не стоит беспокоиться о том, что расширенное развертывание нарушит работу систем внутри компании. Дистрибутив имеет два варианта установки:

-в качестве автономного сервера;

-использование вместе с гибридной, аналитической обработкой;

Базовая структура обладает большой гибкостью. Имеется возможность для разворачивания в облаке или локально, так же имеется выбор запуска на сервере или компьютере.

Огромное количество данных, которые в процессе жизнедеятельности используют, собирают и анализируют компании, являются большими, поэтому важно иметь структуру базы данных, которая удобно предоставляет эту информацию. СУБД может обрабатывать несколько петабайт данных, что соответствует постоянно расширяющемуся списку источников данных. Использование хранилищем реляционной модели данных предоставляет аналитическим операциям множество преимуществ, таких как более высокая производительность при работе с огромными базами данных, поддерживаемыми платформой MariaDB, и 10-кратное сжатие. Дистрибутив может оптимизировать операции с базой данных, перенеся рабочие нагрузки на тот тип хранилища, который лучше всего подходит для определенных данных.

Языком обработки и записи данных служит язык SQL. Язык структурированных запросов SQL это стандартизированный язык программирования, который используется для управления реляционными базами данных и выполнения различных операций с данными в них. Первоначально созданный в 1970-х годах, SQL регулярно используется не только администраторами баз данных, но и разработчиками, пишущими сценарии интеграции данных, и аналитиками данных, которые хотят настроить и выполнить аналитические запросы. SQL используется для следующих целей:

-изменение структуры таблиц и индексов базы данных;

-добавление, обновление и удаление строк данных;

-извлечение подмножеств информации из систем управления реляционными базами данных;

SQL-запросы и другие операции принимают форму команд, написанных в виде инструкций, и объединяются в программы, которые позволяют пользователям добавлять, изменять или извлекать данные из таблиц базы данных [20].

Чтобы работать с базой данных ее нужно подключить на локальном сервере или хостинг сервере. После того как БД будет подключена и создана, есть два варианта наполнять ее данными:

-прописывать все в консоли вручную;

-создать скрипт;

Для работы с БД через консоль нужно к ней подключиться. Процесс подключения происходит при помощи команды “mysql –u root”, где:

1)mysql это подключение к БД

2) -u root это имя пользователя

Так как сервер локальный то пароль можно не указывать. Результат подключение к СУБД отображен на рисунке 3.1.

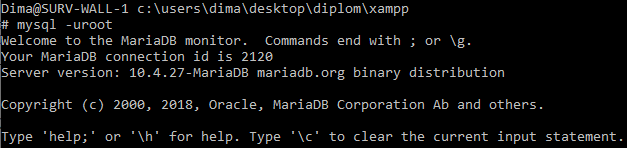


Рисунок 3.1 – Подключение к БД.

Создание БД производится с помощью специальной команды показанной на рисунке 3.2.



Рисунок 3.2 – Создание базы данных.

В состав команд входят такие операторы запроса как:

1)CREATE DATABASE diplom\_DB фрагмент запроса создает новую базу данных с именем “ diplom\_DB ”;

2)COLLATE стандартная кодировка на документе, которую ставят под кириллицу и устанавливают кодировку utf8\_general\_ci;

Для того чтобы создать таблицу в созданной базе данных нужно подключится к ней. Процесс подключение показан на рисунке 3.3.



Рисунок 3.3 – Подключение к созданной БД.

Как пример для описания процессов построения таблиц, была взята таблица «event\_themes». На рисунке 3.4 отображен процесс создания таблицы, все данные из которой взяты из параграфа 2.3.

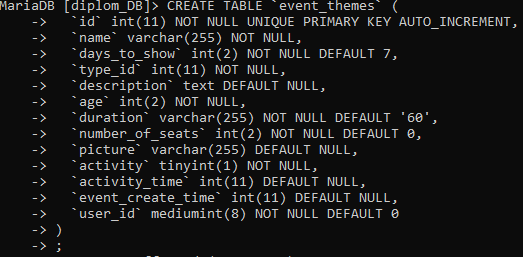


Рисунок 3.4 – Создание таблицы.

В состав команд запроса входят такие операторы как:

-CREATE TABLE event\_themes фрагмент, отвечающий за создание новой таблицы с именем “event\_themes”;

В круглых скобочках идут строчки, в которых указана информация о таблицах, которые чередуются через запятую. Имя каждого столбика указано в кавычках. После имени идут такие настройки как:

1)первым параметром после имени идет тип данных, в скобочках у которого стоит максимальное число знаков в строчке.

2)AUTO\_INCREMENT это параметр, который отвечает за автоматическое заполнение ячейки в порядке возрастания;

3)NOT NULL это параметр, который устанавливает значение на не нулевое;

4)UNIQUE это параметр, который устанавливает значение на уникальное;

5)PRIMARY KEY это параметр, который отвечает за установку первичного ключа;

После скобок идут параметры используемой подсистемы обработки и кодировки:

1)ENGINE это выбор используемой подсистемы;

2)DEFAULT CHARSET это кодировка, которая ставится по умолчанию;

3)COLLATE это параметры кодировки;

Чтобы отобразить в консоли, информацию о созданной выше таблице нужно подключиться заново, и использовать команду “DESCRIBE event\_themes”. Результаты команды отображены на рисунке 3.5.

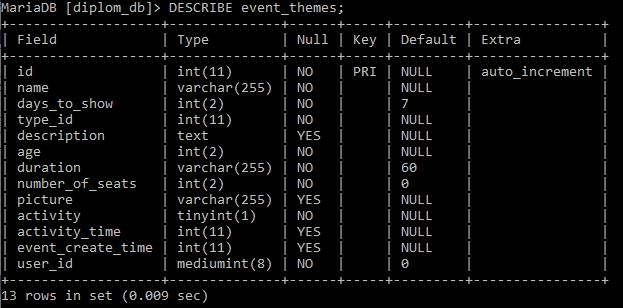


Рисунок 3.5 – Настройки таблицы event\_themes

## 3.2 Разработка web-приложения для автоматизации услуг предприятия культуры

Для реализации web-интерфейса был задействован язык разметки html и таблица стилей css. Обработкой серверных запросов занимался язык php, для работы с “DOM” объектами после прорисовки страницы jQuery.

Для разных групп пользователей существует разные уровни доступа.

Для группы “Администраторы” доступен весь перечень полномочий. По стандартным настройкам после авторизации они попадают на первую страницу в списке. Панель страниц отображена на рисунке 3.6.



Рисунок 3.6 – Панель доступных страниц для группы «Администраторы»

Пользователи, принадлежащие к этой группе, имеют доступ ко всем вкладкам и функциям web-приложения. Схематический перечень полномочий представлен на рисунке 3.7. Графическое отображение окон представлено в приложении А.1 по приложение А.12.

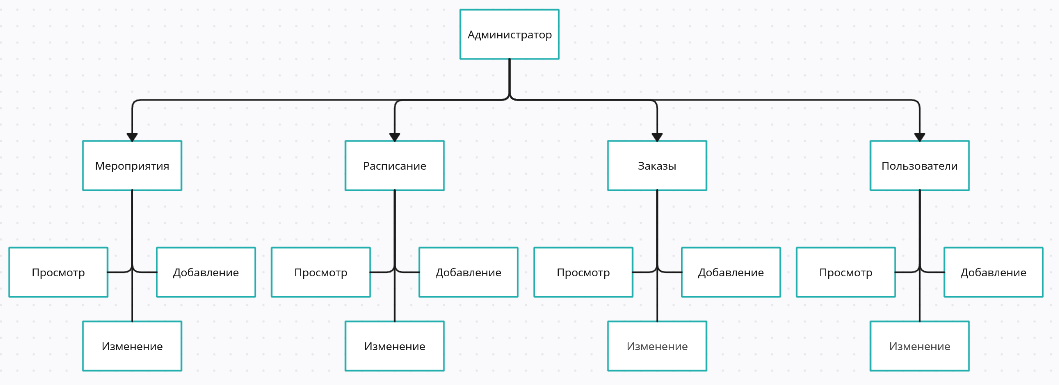


Рисунок 3.7 – Схема уровней доступа группы «Администраторы»

Группа «Управляющие» имеет схожие полномочия что и «Администраторы», отличаются они отсутствием доступа к странице пользователи. По стандартным настройкам после авторизации они попадают на первую страницу в списке. Панель страниц отображена на рисунке 3.8.



Рисунок 3.8 – Панель доступных страниц для группы «Управляющие»

Пользователи, принадлежащие к этой группе, имеют такой же доступ ко всем вкладкам и функциям web-приложения, кроме страницы с пользователями. Схематический перечень полномочий представлен на рисунке 3.9. Графическое отображение окон у этой группы такое же, как и у «Администраторов», отличие только заключается только в имени и должности отображаемой в верхнем правом углу. Отображение доступных окон представлено с приложения А.1 по приложение А.9.

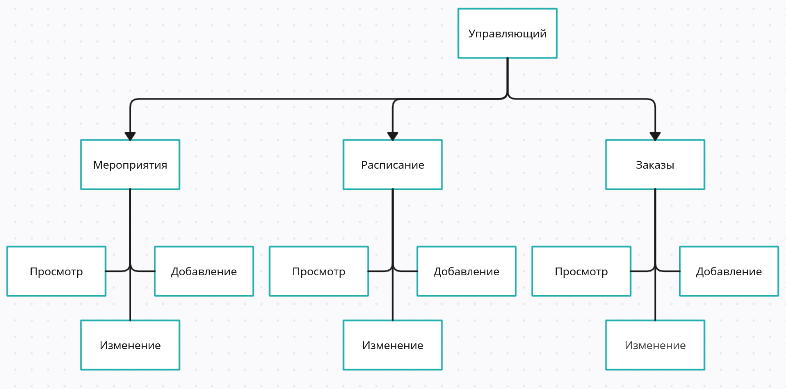


Рисунок 3.9 – Схема уровней доступа группы « Управляющие»

Группа «Экскурсоводы» имеет ограниченный доступ. Пользователи, относящиеся к данной группе, могут просматривать только расписание. В панели доступных страниц отображено только расписание и представлено на рисунке 3.10.



Рисунок 3. 10 – Панель доступных страниц для группы «Экскурсоводы»

Пользователи, принадлежащие к этой группе, имеют право только на просмотр расписания. Схематический перечень полномочий представлен на рисунке 3.11. Графическое отображение окон у этой группы отличается тем, что нет возможности для редактирования и добавление в расписание. Отображение основной доступной страницы представлено в приложении А.13.

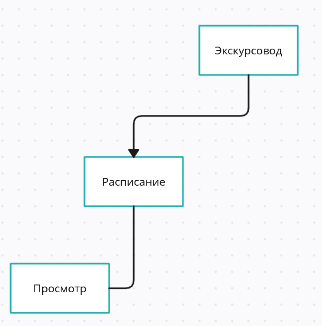


Рисунок 3.11 – Схема уровней доступа группы «Экскурсоводы»

Обычный посетитель имеет доступ только к публичной странице, где показано расписание экскурсий и возможность записаться. Окна доступные посетителю отображены в приложении А.14 и А.15. Схематический перечень полномочий представлен на рисунке 3.12.

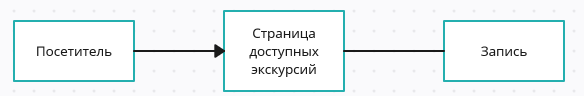


Рисунок 3.12 – Схема уровней доступа группы посетителя

Отображение страницы у пользователя, начинается с документа index.php. В этом файле прописаны все пути и ссылки на функции конструкторы. Чтобы загрузить и показать нужную страницу нужно подключиться к базе данных. Программный код отвечающий за это представлен в приложении Б.1.

После того как произошло подключение, нужно определить права пользователя. Это нужно для того чтобы пользователь мог использовать только те функции которые доступны его группе, программный код отвечающий за это представлен в приложении Б.2.

Так как сайт динамический то он конструируется с помощью функций конструкторов. Для того чтобы перейти на определенную страницу нужно получить определенный ключ из ссылки и ключ доступа из сессии, после чего путем оператора switch происходит переадресация на функцию конструктор которая начинает конструировать страницу пользователю, программный код отвечающий за это представлен в приложении Б.3.

После переадресации на страницу, идет процесс конструирования. Суть конструктора заключается в запросах в базу данных и упаковывания полученной информации в массив, программный код отвечающий за это представлен в приложении Б.4.

После помещения данных в массив идет конструирование самой страницы. Для этого собранные в массиве данные отсылаются на функцию шаблон, которая расставляет их в свои ячейки. Программный код отвечающий за это представлен в приложении Б.5.

После того как все данные полностью расфасованы они отсылаются на функцию вывода пользователю, программный код отвечающий за это представлен в приложении Б.6.

Основной элемент автоматизации состоит из части, которая записывает посетителей на экскурсии или другие мероприятия. Суть данной части кода заключается в обработке и создании записи в базе данных о брони человеком места. Данный метод полностью автоматизирует запись человека на экскурсию, что упрощает работу администратора и минимизирует затраты времени. Программный код отвечающий за это представлен в приложении Б.7.

По сравнению с имеющимися аналогами, разработанная система очень удобна и интуитивно понятна. Основными отличительными достоинствами являются:

1. Отсутствие ненужных фреймворков позволяет программному обеспечению быть независимым, что в свою очередь повышает производительность системы избавляя ее от ненужного софта.

2. Интуитивно понятный интерфейс, в подобных системах бывает очень запутанное меню с огромным функционалом. Разрабатываемая система лишена этого и сводит все к минимизированию и упрощению.

3. Полностью настраиваемая система, которая позволяет в полной мере раскрыть потенциал.

## 3.3 Тестирование web-приложения

Для тестирования разработанной системы направленной на автоматизацию нужно записаться на экскурсию. Самая первая экскурсия доступная это обзорная представленная на рисунке 3.11.

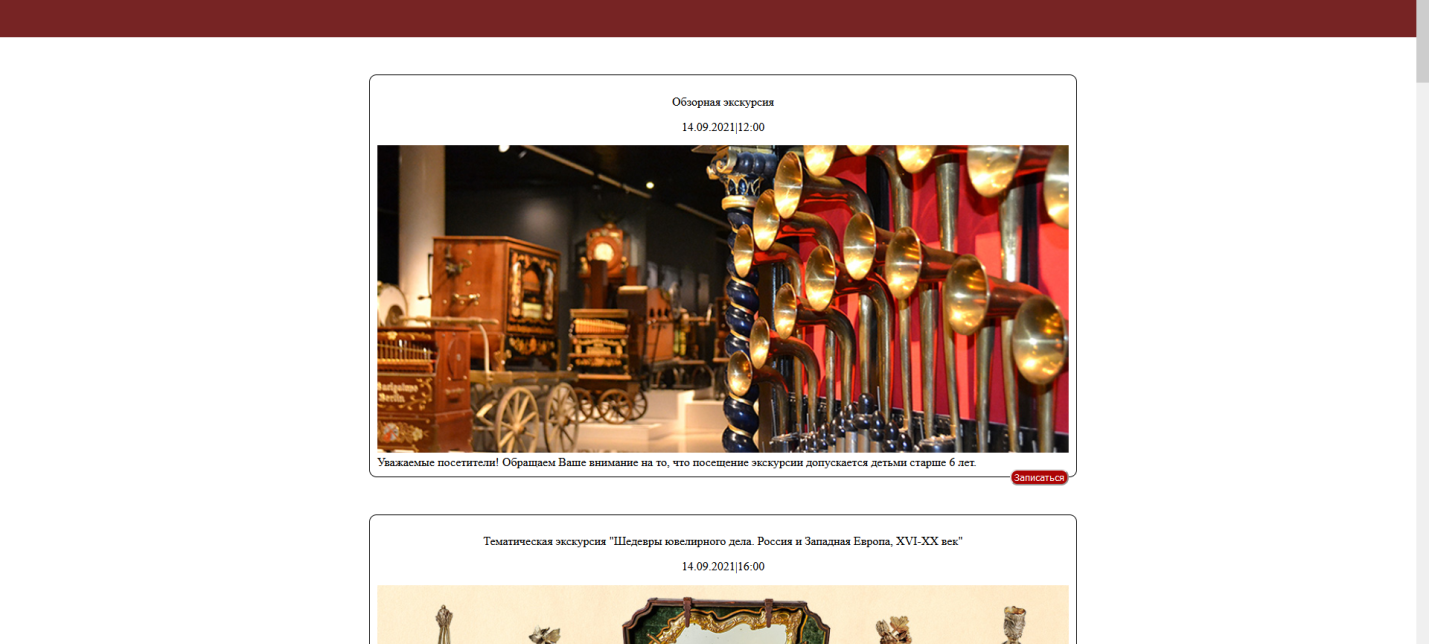


Рисунок 3.11 – Страница с экскурсиями

После нажатия на кнопку записаться появляется всплывающее окно, с контактными данными посетителя, которое представлено на рисунке 3.12.

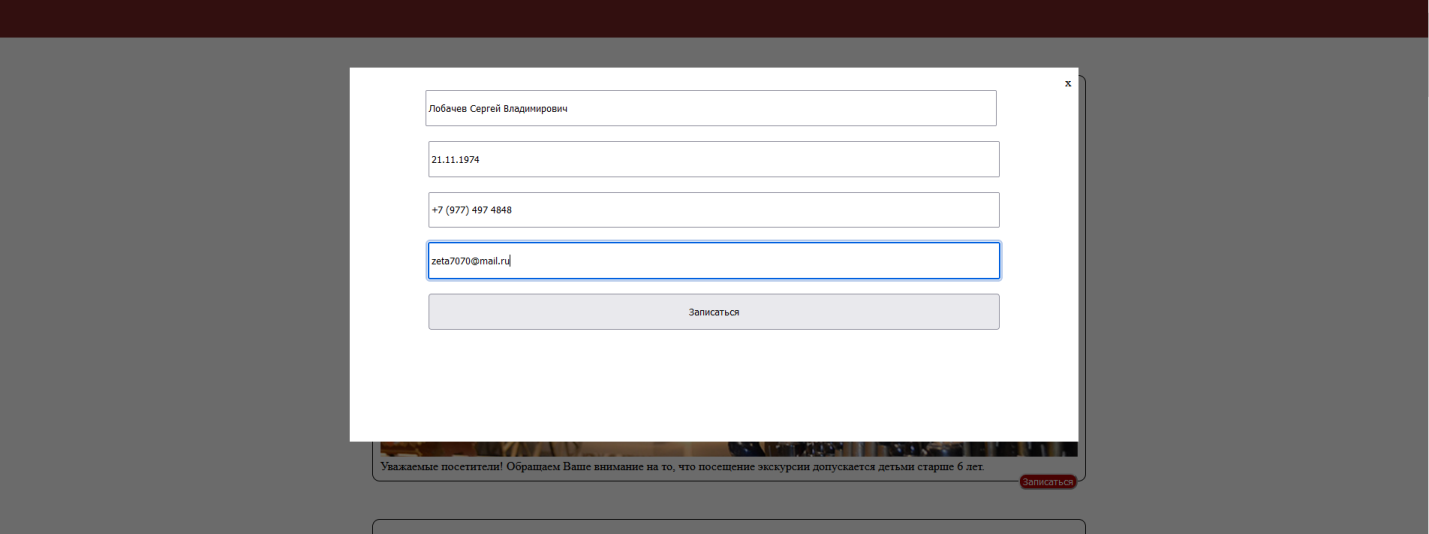


Рисунок 3.12 – Форма для записи

После того как посетитель записался можно посмотреть добавилась ли запись в БД. Для этого нужно войти в панель администратора. В процессе входа требуется ввести логин и пароль, форма для логина и пароля представлена на рисунке 3.13.

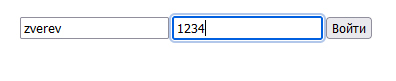


Рисунок 3.13 – Форма авторизации

После авторизации можно увидеть запись в заказах отображенной на рисунке 3.14.

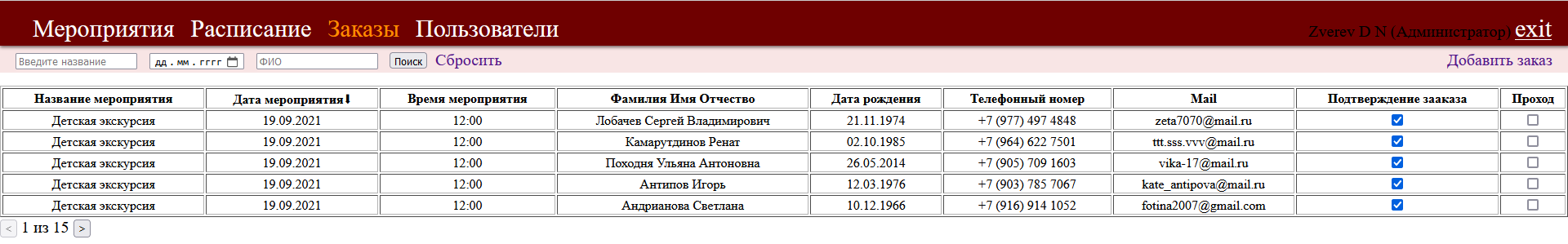


Рисунок 3.14 – Страница с заказами

Каждая экскурсия имеет определенное количество мест. После достижения данного количества экскурсия больше не выводится пользователям.

## Выводы

В результате анализа доступных готовых решений, направленных на автоматизацию, было спроектировано собственное web-решение. Результатом такой проектировки стало web-приложение, которое было разработано под нужды бесплатного музея, оно содержит только функции, которые нужны непосредственно в нутрии рабочих процессов осуществляемых в нем.

В результате проведенного тестирования сделан вывод, что разработанное web-приложение намного производительнее своих аналогов, так как оно не включает лишние блоки, которые могут замедлять работу системы. Интерфейс разработанной системы не нагружен большим скоплением функций, что упрощает понимание и работу в нем.

В связи с тем, разработанное решение намного быстрей справляется со своими задачами, сделан вывод, что оно намного эффективней и дешевле предлагаемых аналогов.

# Заключение

В ходе работы было проведено исследование в сфере управления предприятиями культуры. В результате проведенных исследований было обосновано, что повышение эффективности работы с клиентами за счет автоматизации отдельных бизнес-процессов является актуальной темой исследований.

В результате проведенного исследования методов и способов внедрения автоматизированных систем в работу музеев, было определено, что на сегодняшний день наиболее оптимальным является использование CRM-систем.

Используя полученные результаты сформированы требования к web-приложению, основным назначением которого является повышение эффективности работы музея с клиентами за счет автоматизации отдельных процессов.

В результате сравнительного анализа готовых решений – web-сайтов 10 популярных музеев была выбраны базовые системы для разработки собственного решения: битрикс, инфотех.

Разработана структура требуемой базы данных на физическом и логическом уровнях и программная реализация web-приложения на основах спроектированной базы данных и требований к системе.

Проведены тесты разработанной системы, результаты которых показали, что разработанное решение удобнее и быстрее в использовании, а также внедрение такого решения примерно в 2 раза дешевле предлагаемых аналогов.

Все поставленные задачи выполнены, цель работы достигнута.

# Список использованной литературы

1. Серяева М.Г "сфера культуры" Российской Федерации: понятие и источники финансирования 2019. URL: https://cyberleninka.ru/article/n/sfera-kultury-rossiyskoy-federatsii-ponyatie-i-istochniki-finansirovaniya (дата обращения: 28.12.2022).
2. Музеи. Сводные данные. Статистическая информация. [Электронный ресурс]. URL: https://opendata.mkrf.ru/opendata/7705851331-stat\_museum\_svod (дата обращения: 15.12.2022).
3. Рейтинг посещаемости российских музеев и художественных выставок за 2021 год [Электронный ресурс]. URL: https://www.theartnewspaper.ru/posts/20220603-rtsk/(дата обращения: 18.12.2022).
4. Рейтинг организаций по выручке [Электронный ресурс]. URL: https://www.testfirm.ru/rating/91\_02/(дата обращения: 19.12.2022).
5. CRM (мировой рынок) [Электронный ресурс]. URL: https://www.tadviser.ru/index.php/ Статья:CRM\_(мировой\_рынок) (дата обращения: 20.12.2022).
6. Преимущества и недостатки CRM-систем Электронный ресурс]. URL: https://www.bitrix24.ru/journal/preimushchestva-i-nedostatki-crm-sistem/(дата обращения: 20.12.2022).
7. The 360-degree view: Why an integrated CRM platform is important in growing a museum’s membership program | MWA2015: Museums and the Web Asia 2015 [Электронный ресурс]. URL: 33 https://mwa2015.museumsandtheweb.com/paper/the-360-degree-view-why-anintegrated-crm-platform-is-important-in-growing-a-museums-membershipprogram/ (дата обращения: 21.12.2022).
8. Кто работает в музее [Электронный ресурс]. URL: https://kedu.ru/press-center/articles/ekskursovody-smotriteli-nauchnye-sotrudniki-kto-rabotaet-v-muzee/(дата обращения: 28.12.2022).
9. Битрикс24 [Электронный ресурс]. URL: https://www.bitrix24.ru/(дата обращения: 28.12.2022).
10. Облачный сервис TicketNet.Cloud [Электронный ресурс]. URL: <http://museum-cloud.ru/>(дата обращения: 28.12.2022).
11. Что такое База Данных [Электронный ресурс]. URL: https://www.oracle.com/cis/database/what-is-database/(дата обращения: 30.12.2022).
12. Иерархическая база данных: что это такое, пример модели БД организации [Электронный ресурс]. URL: https://59i.ru/stati/tehnologii/ierarhicheskaja-baza-dannyh-chto-yeto-takoe-primer-modeli-bd-organizacii.html (дата обращения: 28.12.2022).
13. Сетевые базы данных. [Электронный ресурс]. URL: https://mega-obzor.ru/setevye-bazy-dannyx.html (дата обращения: 30.12.2022).
14. Что такое реляционная база данных [Электронный ресурс]. URL: https://www.oracle.com/cis/database/what-is-a-relational-database/(дата обращения: 30.12.2022).
15. Инфологическое проектирование баз данных. ER-диаграммы [Электронный ресурс]. URL: https://studme.org/93799/informatika/infologicheskoe\_proektirovanie\_dannyh\_er-diagrammy (дата обращения: 30.12.2022).
16. Как привести данные в форму: что такое нормализация и зачем она нужна [Электронный ресурс]. URL: https://practicum.yandex.ru/blog/chto-takoe-normalizaciya-dannyh/ (дата обращения: 30.12.2022).
17. Даталогическое проектирование базы данных. [Электронный ресурс]. URL: https://bstudy.net/732015/informatika/datalogicheskoe\_proektirovanie \_bazy\_dannyh(дата обращения: 5.01.2023).
18. Методическое пособие по проектированию БД [Электронный ресурс]. URL: http://rema44.ru/resurs/study/dbprj/dbprj.html(дата обращения: 5.01.2023).
19. MySQL vs MariaDB[Электронный ресурс]. URL: https://www.netangels.ru/support/overview/mysql-vs-mariadb/(дата обращения: 5.01.2023).
20. Методическое пособие по проектированию БД [Электронный ресурс]. URL: http://rema44.ru/resurs/study/dbprj/dbprj.html(дата обращения: 10.01.2023).
21. Язык SQL [Электронный ресурс]. URL: https://blog.skillfactory.ru/glossary/sql/(дата обращения: 15.12.2022).
22. Музей Петергоф [Электронный ресурс]. URL: https://peterhofmuseum.ru/(дата обращения: 15.01.2023).
23. Казанский кремль [Электронный ресурс]. URL: https://kazan-kremlin.ru/(дата обращения: 15. 01.2023).
24. Исаакиевский Собор [Электронный ресурс]. URL: https://www.cathedral.ru/ru(дата обращения: 15. 01.2023).
25. Русский музей [Электронный ресурс]. URL: http://www.rusmuseum.ru/(дата обращения: 15. 01.2023).
26. Мультимедиа Арт Музей [Электронный ресурс]. URL: https://mamm-mdf.ru/(дата обращения: 15. 01.2023).
27. Музей заповедник Сталинградская Битва [Электронный ресурс]. URL: /(дата обращения: 25.12.2022).
28. Государственный музей-заповедник «Царское Село» [Электронный ресурс]. URL: https://www.tzar.ru/(дата обращения: 25.12.2022).
29. Государственный Эрмитаж [Электронный ресурс]. URL: https://tickets.hermitagemuseum.org/ru/#id=1/(дата обращения: 25.12.2022).
30. Третьяковская галерея [Электронный ресурс]. URL: https://www.tretyakovgallery.ru/?lang=ru/(дата обращения: 25.12.2022).
31. Владимиро-суздальский музей заповедник [Электронный ресурс]. URL: https://vladmuseum.ru/ru/(дата обращения: 25.12.2022).

# ПРИЛОЖЕНИЕ А

****

Рисунок А.1 – Страница со списком «Мероприятий»

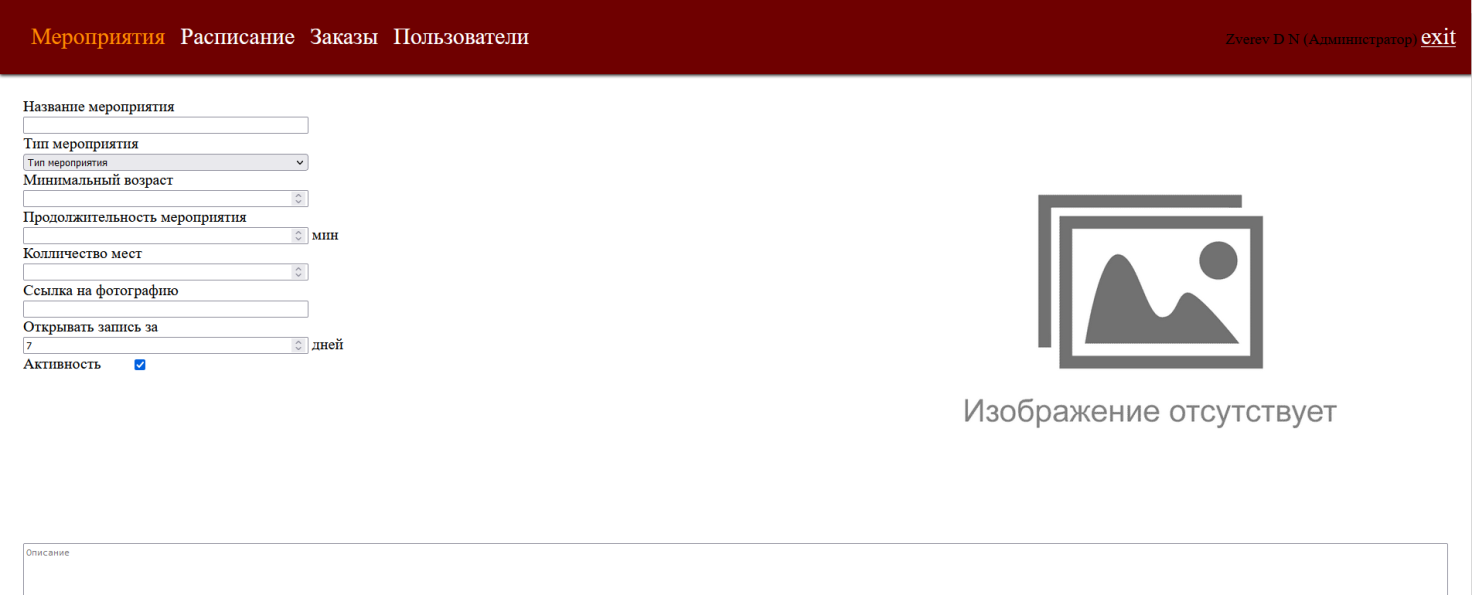


Рисунок А.2 – Страница с формой для добавления «Мероприятий»

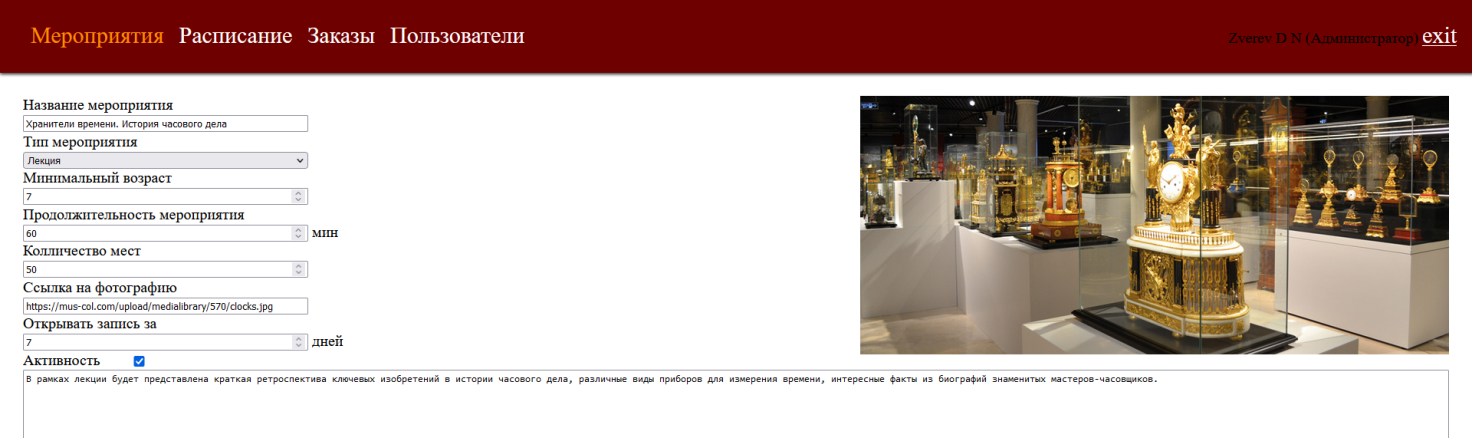


Рисунок А.3 – Страница с формой для изменения «Мероприятий»

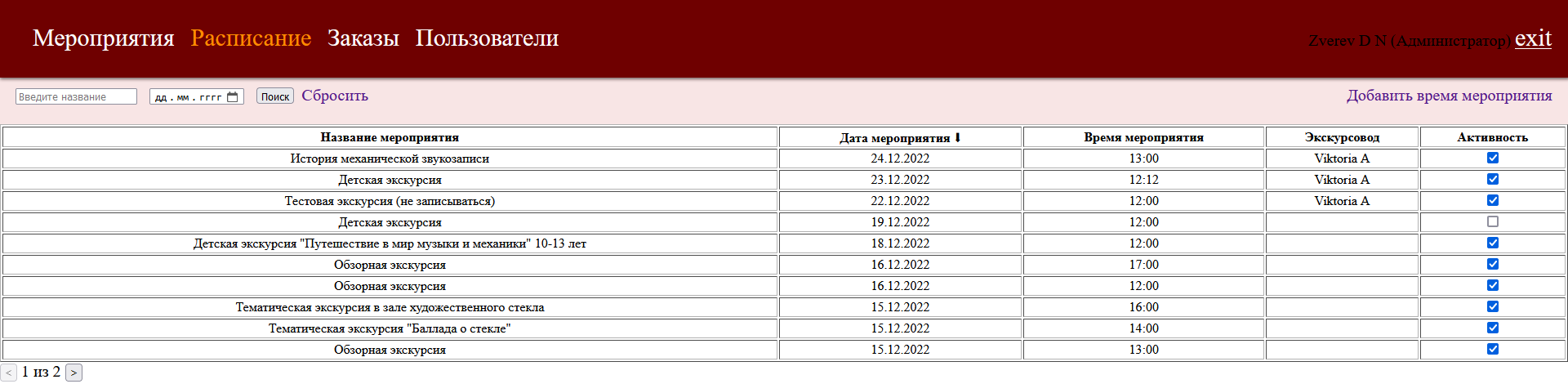


Рисунок А.4 – Страница с «Расписанием»

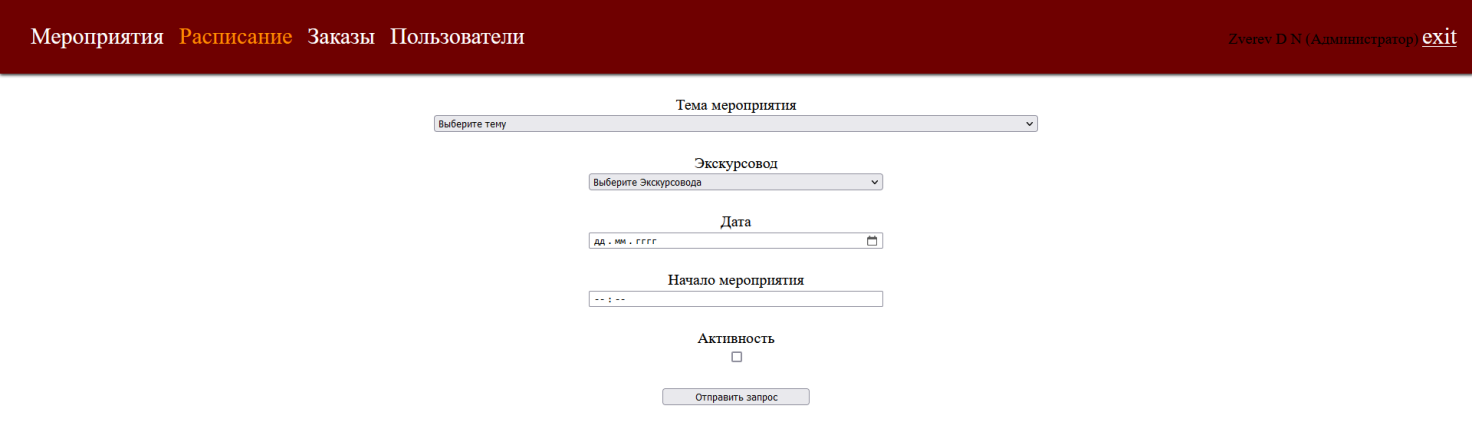


Рисунок А.5 – Страница с формой для добавления «Расписания»

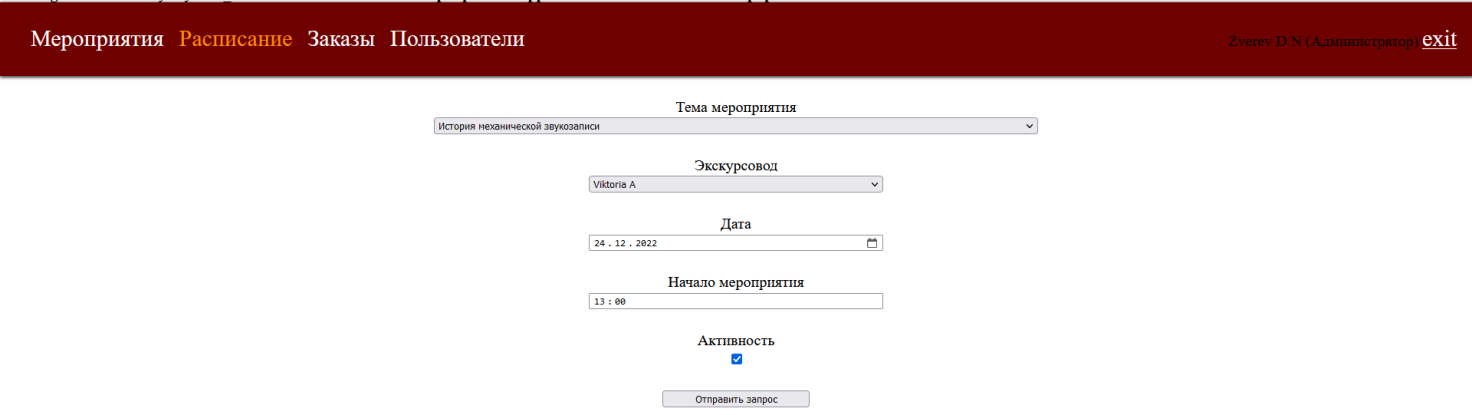


Рисунок А.6 – Страница с формой для изменения «Расписания» 

Рисунок А.7 – Страница с «Заказами»

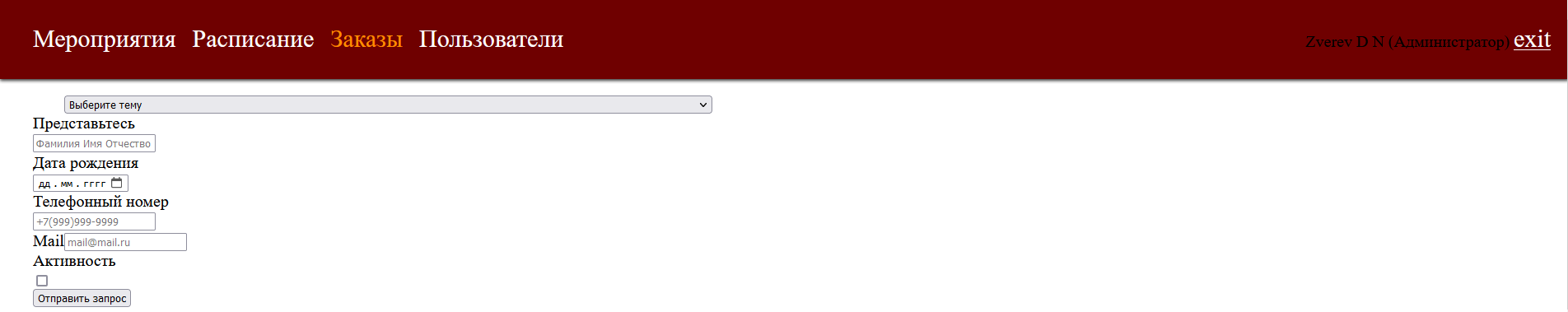


Рисунок А.8 – Страница с формой для добавления «Заказаов»

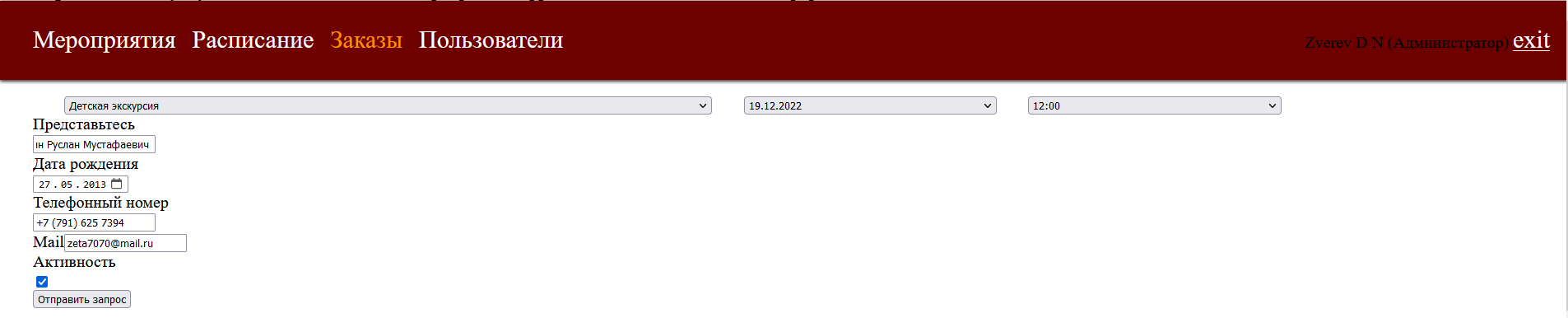


Рисунок А.9 – Страница с формой для изменения «Заказов»



Рисунок А.10 – Страница с «Пользователями»

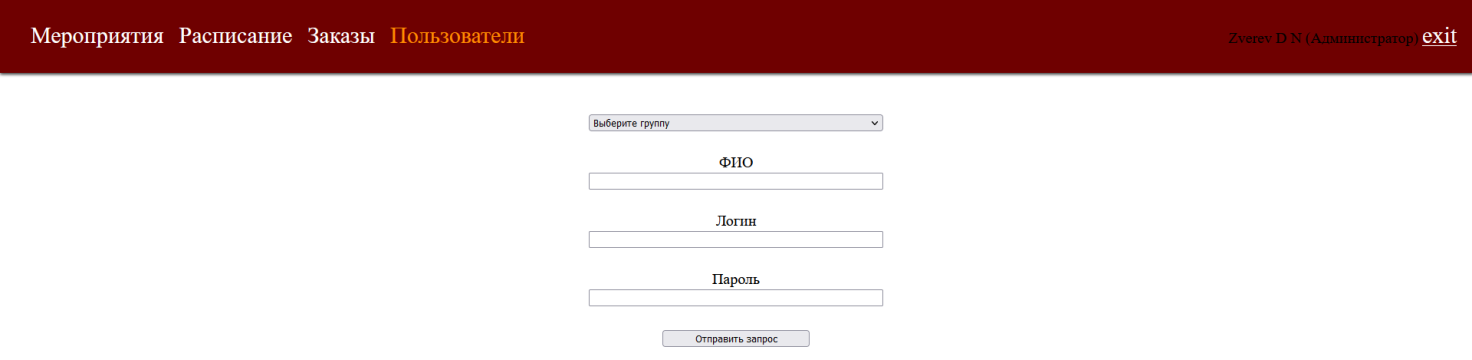


Рисунок А.11 – Страница с формой для добавления «Пользователей»



Рисунок А.12 – Страница с формой для изменения «Пользователей»

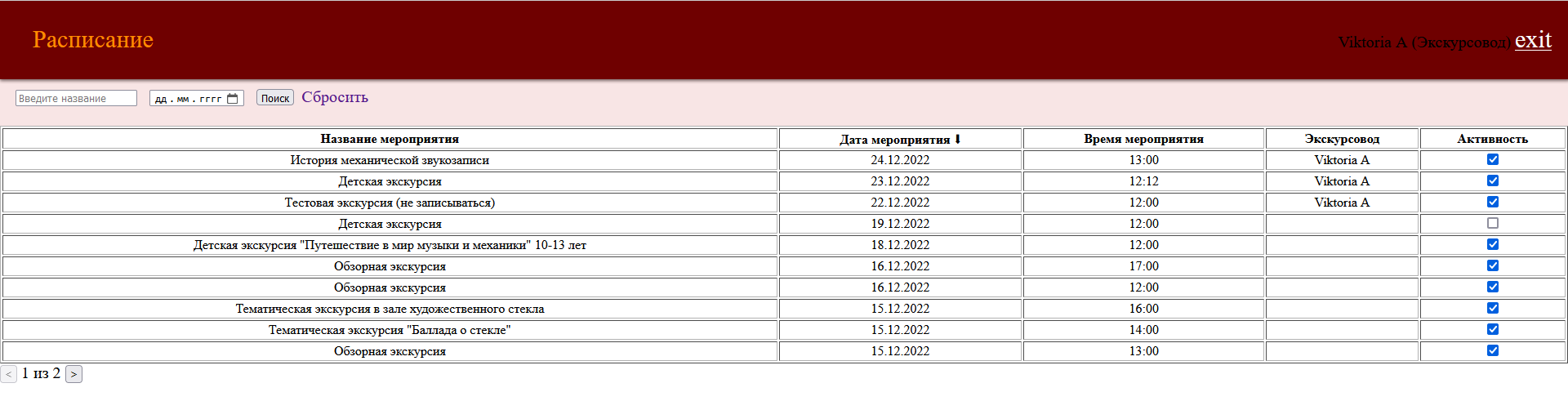


Рисунок А.13 – Страница с «Расписанием» для экскурсоводов

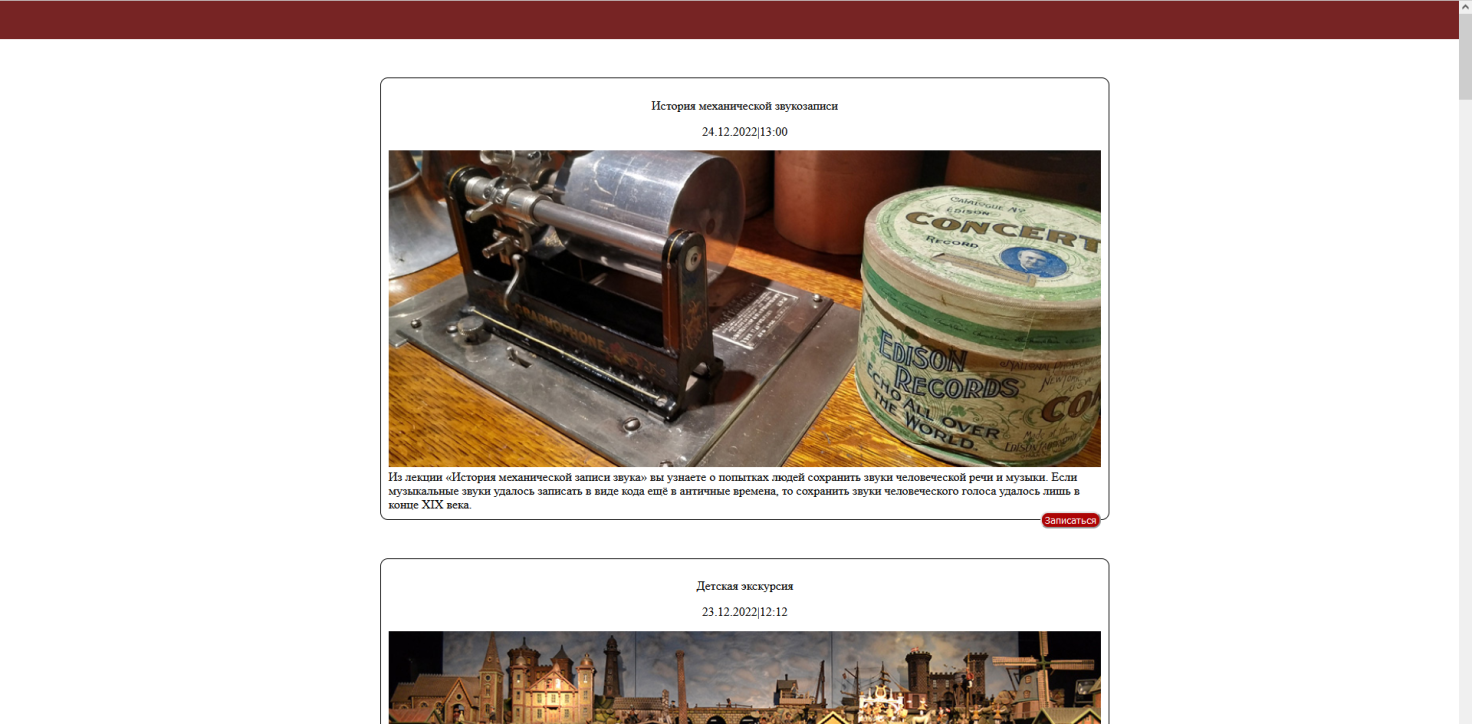
****

Рисунок А.14 – Страница с записью на экскурсию для «Посетителей»

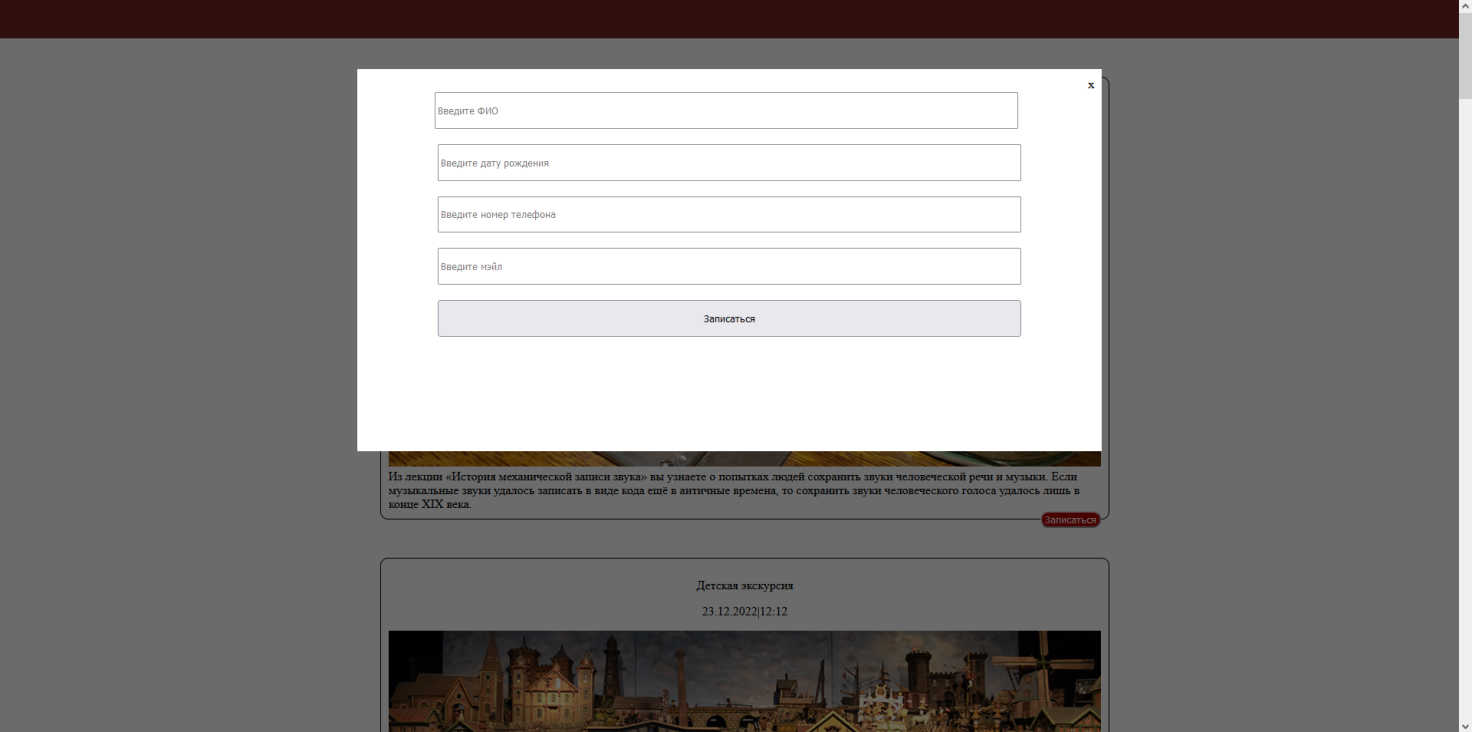
****

Рисунок А.15 – Страница с формой записи на экскурсию для «Посетителей»

# ПРИЛОЖЕНИЕ Б

Листинг 1

static public $obj = array ( "sql\_database" => "" ,

"sql\_user" => "" ,

"sql\_pass" => "" ,

"sql\_host" => "",

"sql\_port" => 3306 ,

"sql\_socket" => NULL ,

"persistent" => "0" ,

"sql\_tbl\_prefix" => "mkp\_" ,

"cached\_queries" => array(),

'debug' => 0,

);

static private $query\_id = "";

static private $connection\_id = "";

static private $query\_count = 0;

static private $record\_row = array();

static private $return\_die = 0;

static private $error = "";

static private $failed = 0;

static public function connect() {

define('SQL\_PREFIX', '');

self::$connection\_id = mysqli\_connect( self::$obj['sql\_host'],self::$obj['sql\_user'],self::$obj['sql\_pass'],NULL,self::$obj['sql\_port'],self::$obj['sql\_socket']);

if (!self::$connection\_id)

{

self::$fatal\_error('ERROR: Link-ID == false, connect failed');

return false;

}

if ( !mysqli\_select\_db(self::$connection\_id, self::$obj['sql\_database']) )

{

self::$fatal\_error("ERROR: Cannot find database ".self::$obj['sql\_database']);

return false;

}

}

Листинг 2

function \_\_construct(){

if(!isset($\_SESSION['rights'])){

$\_SESSION['rights'] = 0;

}

$login = (isset(GENERAL::$input['login'])!="")?GENERAL::$input['login']:"";

$pass = (isset(GENERAL::$input['pass'])!="")?GENERAL::$input['pass']:"";

if(!isset($\_SESSION['username'])!="" and !isset($\_SESSION['name'])!=""){

DB::query("SELECT\*FROM`users`LEFT JOIN type\_of\_users ON users.type\_id= type\_of\_users.id WHERE `login` LIKE '{$login}' AND `pass` LIKE '{$pass}'");

$row=DB::fetch\_row();

if($row!=""){

$\_SESSION['username']=$row["username"];

$\_SESSION['rights']=$row["rights"];

$\_SESSION['work\_group']=$row['name'];

$\_SESSION['data'] = time();

}

}else{if(intval(time() - $\_SESSION['data']) > 600){session\_destroy();}}}

Листинг 3

<?php

namespace FUNC;

error\_reporting(E\_ALL & ~E\_NOTICE);//скрывать notice

date\_default\_timezone\_set('Europe/Moscow');

session\_start();

require "autoload.php";

use FUNC\FUNC;

use FUNC\DB;

use FUNC\VERIFICATION;

use FUNC\AJAX;

use FUNC\AUTH;

class GENERAL{

public static $input = array();

public static $settings = array();

function \_\_construct(){

self::$input = VERIFICATION::parse\_incoming();

DB::$obj['sql\_database']="my\_diplom\_bd";

DB::$obj['sql\_user']="root";

DB::$obj['sql\_host']="localhost";

DB::connect();

DB::query("SELECT \* FROM `settings`");

while (($row=DB::fetch\_row())!=false) {

self::$settings[$row['name']]= $row['value'];

}

}

}

$start = new GENERAL();

$auth = new AUTH();

GENERAL::$input['act'] = (isset(GENERAL::$input['act']))? GENERAL::$input['act'] : "";

if(GENERAL::$input['act']=="exit"){

session\_destroy();

header("location:index.php");

exit;

}

if(GENERAL::$input['act']=="activ"){

AJAX::activ();

}

switch ($\_SESSION['rights']) {

case '1':

switch(GENERAL::$input['act']){

case 'event\_themes':

FUNC::event\_themes();

break;

case 'event\_time':

FUNC::event\_time();

break;

case 'orders':

FUNC::orders();

break;

case 'users':

FUNC::users();

break;

default:

FUNC::event\_themes();

break;

}

break;

case '2':

switch(GENERAL::$input['act']){

case 'event\_themes':

FUNC::event\_themes();

break;

case 'event\_time':

FUNC::event\_time();

break;

case 'orders':

FUNC::orders();

break;

default:

FUNC::event\_themes();

break;

}

break;

case '3':

switch(GENERAL::$input['act']){

case 'event\_time':

FUNC::event\_time();

break;

default:

FUNC::event\_time();

break;

}

break;

default:

FUNC::auth\_page();

break;

}?>

Листинг 4

public static function event\_themes(){

GENERAL::$input['do'] = (isset(GENERAL::$input['do']))? GENERAL::$input['do'] : "";

switch(GENERAL::$input['do']){

case'add':

self::event\_themes\_add\_page();

exit;

case'change':

self::event\_themes\_change\_page();

exit;

case'event\_themes\_insert':

self::event\_themes\_insert();

break;

case'event\_themes\_update':

self::event\_themes\_update();

break;

}

$select= array();

$return\_val=array("type\_even\_id"=>"","search\_name"=>"");

$data=array("search"=>"","content"=>"","type"=>"");

$page = array();

if(isset(GENERAL::$input['search\_name']) and GENERAL::$input['search\_name']!=""){

$return\_val['search\_name']=GENERAL::$input['search\_name'];

$select[]=" name LIKE '%".GENERAL::$input['search\_name']."%'";

}

if(isset(GENERAL::$input['type\_even\_id']) and GENERAL::$input['type\_even\_id']>0){

$return\_val['type\_even\_id']=GENERAL::$input['type\_even\_id'];

$select[]=" type\_id = ".GENERAL::$input['type\_even\_id'];

}

if(count($select)>0){

$data['search']="WHERE";

$data['search'].=implode(' AND' ,$select);

}

$page=self::pages("SELECT COUNT(id) FROM `event\_themes` {$data['search']}");

$sort=self::sorting(array('name','type','age','duration','number\_of\_seats'),'name');

DB::query("SELECT event\_themes.\*, type\_of\_events.type FROM event\_themes LEFT JOIN type\_of\_events ON type\_of\_events.id = event\_themes.type\_id {$data['search']} ORDER BY {$sort['orderby']} {$sort['ordertype']} {$page['limit']}");

while (($row=DB::fetch\_row())!=false) {

$data['content'].= TEMPL::event\_themes\_row($row);

}

DB::query("SELECT \* FROM type\_of\_events WHERE activity=1");

while (($row=DB::fetch\_row())!=false) {

$sel\_val=($row['id']==$return\_val['type\_even\_id'])?"selected":"";

$data['type'].=TEMPL::option($row['type'],$row['id'],$sel\_val);

}

self::print\_page(TEMPL::event\_themes($data,$return\_val,$page,$sort));

}

private static function event\_themes\_add\_page($post=array(),$error=''){

$content = '';

$sel\_val='';

$do='event\_themes\_insert';

DB::query("SELECT id,type FROM type\_of\_events WHERE activity=1");

while (($row=DB::fetch\_row())!=false) {

$content.=TEMPL::option($row['type'],$row['id'],$sel\_val);

}

if(isset(GENERAL::$input['do']) and GENERAL::$input['do']=="add"){

$row=array("activity"=>"false","id"=>"","picture"=>"","name"=>"","age"=>"","duration"=>"","number\_of\_seats"=>"","description"=>"");

}

self::print\_page(TEMPL::event\_themes\_add($content,$row,$do,$post,$error));

}

private static function event\_themes\_change\_page(){

$content = '';

$data=array();

$sel\_val='';

$id=(isset($\_GET['id']) and intval($\_GET['id']))?$\_GET['id']:"";

$do='event\_themes\_update';

DB::query("SELECT event\_themes.\*, type\_of\_events.type FROM event\_themes LEFT JOIN type\_of\_events ON type\_of\_events.id = event\_themes.type\_id WHERE event\_themes.id ='{$id}'");

$data=DB::fetch\_row();

DB::query("SELECT id,type FROM type\_of\_events");

while (($row=DB::fetch\_row())!=false) {

$sel\_val=($data['type\_id']==$row['id'])?"selected":"";

$content.=TEMPL::option($row['type'],$row['id'],$sel\_val);

}

self::print\_page(TEMPL::event\_themes\_add($content,$data,$do));

}

private static function event\_themes\_insert(){

$error='';

if(isset(GENERAL::$input['activity']) AND GENERAL::$input['activity']!=1){

GENERAL::$input['activity']=0;

}

if(intval(GENERAL::$input['type\_id'])!=0){

DB::query("SELECT \* FROM `type\_of\_events` WHERE `id` = ".GENERAL::$input['type\_id']." AND activity = 1");

if(DB::get\_num\_rows()==0){

$error.='Тема мероприятия</br>';

}

}

if(intval(GENERAL::$input['age'])<0 and intval(GENERAL::$input['age'])>100 ){

$error.='Возраст</br>';

}

if(intval(GENERAL::$input['duration'])<=0){

$error.='Продолжительность</br>';

}

if(intval(GENERAL::$input['number\_of\_seats'])<=0){

$error.='Колличество мест</br>';

}

if (!preg\_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-a-z0-9+&@#\/%=~\_|]/i",GENERAL::$input['picture'])) {

$error.= 'Ссылка на фото';

}

GENERAL::$input['name']=trim(GENERAL::$input['name']);

GENERAL::$input['name'] = str\_replace(" ", " ", GENERAL::$input['name']);

while( strpos(GENERAL::$input['name']," ")!==false){

GENERAL::$input['name'] = str\_replace(" ", " ", GENERAL::$input['name']);

}

GENERAL::$input['name']=htmlspecialchars (GENERAL::$input['name']);

GENERAL::$input['name']=addslashes(GENERAL::$input['name']);

GENERAL::$input['description']=trim(GENERAL::$input['description']);

GENERAL::$input['description'] = str\_replace(" ", " ", GENERAL::$input['description']);

while( strpos(GENERAL::$input['description']," ")!==false){

GENERAL::$input['description'] = str\_replace(" ", " ", GENERAL::$input['description']);

}

GENERAL::$input['description']=htmlspecialchars (GENERAL::$input['description']);

GENERAL::$input['description']=addslashes(GENERAL::$input['description']);

if($error!=''){

TEMPL::event\_themes\_add(GENERAL::$input,$error);

exit;

}

DB::do\_insert('event\_themes', array('name' => GENERAL::$input['name'],

'type\_id' => GENERAL::$input['type\_id'],

'age' => GENERAL::$input['age'],

'duration' => GENERAL::$input['duration'],

'description' => GENERAL::$input['description'],

'number\_of\_seats' => GENERAL::$input['number\_of\_seats'],

'picture' => GENERAL::$input['picture'],

'days\_to\_show' => GENERAL::$input['days\_to\_show'],

'activity' => GENERAL::$input['activity']));

header("location:index.php?act=event\_themes");

exit;

}

private static function event\_themes\_update(){

$id=(isset(GENERAL::$input['id']) and intval(GENERAL::$input['id']))?GENERAL::$input['id']:"";

DB::do\_update('event\_themes', array('name' => GENERAL::$input['name'],

'type\_id' => GENERAL::$input['type\_id'],

'age' => GENERAL::$input['age'],

'duration' => GENERAL::$input['duration'],

'number\_of\_seats' => GENERAL::$input['number\_of\_seats'],

'picture' => GENERAL::$input['picture'],

'days\_to\_show' => GENERAL::$input['days\_to\_show'],

'activity' => GENERAL::$input['activity']),"id={$id}");

header("location:index.php?act=event\_themes");

exit;

}

Листинг 5

public static function event\_themes\_row($row=array()){

$row['activity'] = ($row['activity']=='1')? "checked":"";

return <<<EOF

<tr><td><a href="index.php?act=event\_themes&do=change&id={$row['id']}">{$row['name']}</a></td><td>{$row['type']}</td><td>{$row['age']}</td><td>{$row['duration']} мин</td><td>{$row['number\_of\_seats']}</td><td><input class='activ' name = "event\_themes" type="checkbox" value='{$row['id']}' {$row['activity']}/></td><td><a href="index.php?act=event\_time&mName={$row['name']}">Посмотреть в расписании</a></td></tr>

EOF;

}

public static function event\_themes\_add($content='',$row=array(),$do='',$post=array(),$error=''){

$row['activity'] = ($row['activity'])? "checked":"";

$error=($error!='')?"<div class='error'>Проверьте правильность полей:</br>{$error}</div>":"";

$row['days\_to\_show']=(isset($row['days\_to\_show']))?$row['days\_to\_show']:"7";

return <<<EOF

<form action="index.php?act=event\_themes&do={$do}" method="post" enctype="multipart/form-data">

<ul class='event\_themes\_add'>

{$error}

<input name="id" value="{$row['id']}" type="hidden"/>

<li><img style='display:none;' src='{$row['picture']}' id='image'></li>

<li><img style='display:block;' src='not\_img.png' id='no\_image'></li>

<li>Название мероприятия </br><input required value="{$row['name']}" name="name"/></li>

<li>Тип мероприятия </br><select name="type\_id"><option value="0">Тип мероприятия</option>{$content}</select></li>

<li>Минимальный возраст</br><input required type="number" value="{$row['age']}" name="age"/></li>

<li>Продолжительность мероприятия</br><input required type="number" value="{$row['duration']}" name="duration"/> мин</li>

<li>Колличество мест</br><input required value="{$row['number\_of\_seats']}" type="number" name="number\_of\_seats"/></li>

<li>Ссылка на фотографию</br><input required id='picture' value="{$row['picture']}" type="url" name="picture"/></li>

<li>Открывать запись за</br><input required value="{$row['days\_to\_show']}" type="number" name="days\_to\_show"/> дней</li>

<li>Активность<input type="checkbox" value="1" name="activity" {$row['activity']}/></li>

<li><textarea required placeholder='Описание' name="description">{$row['description']}</textarea></li>

<input type="submit"/>

</ul>

</form>

EOF;

}

public static function event\_themes($data =array(),$return\_val=array(),$page=array(),$sort=array()){

return <<<EOF

<div class="opt\_line">

<form name="search" action="index.php?act=event\_themes" method="post" enctype="multipart/form-data">

<input autocomplete="off" name='search\_name' value='{$return\_val['search\_name']}' class='search' type='text' placeholder='Введите название' />

<select name='type\_even\_id' class='search'><option value='0'>Выберите тип</option>{$data['type']}</select>

<input type='hidden' name='orderby' value='{$sort['orderby']}'/>

<input type='hidden' name='page' value=''/>

<input type='hidden' name='ordertype' value='{$sort['ordertype']}'/>

<input class='search' value='Поиск' type="submit"/>

<a href='index.php?act=event\_themes'>Сбросить<a/>

</form>

<a href="index.php?act=event\_themes&do=add">Добавить мероприятие</a>

</div>

<div style="clear:both"></div>

<table border='1' class="output\_table">

<thead>

<tr>

<th class='sort' name='{$sort['name']}' id='name'>Название мероприятия {$sort['name\_arrow']}</th>

<th class='sort' name='{$sort['type']}'id='type'> Тип мероприятия {$sort['type\_arrow']}</th>

<th class='sort' name='{$sort['age']}'id='age'> Минимальный возраст {$sort['age\_arrow']}</th>

<th class='sort' name='{$sort['duration']}'id='duration'> Продолжительность {$sort['duration\_arrow']}</th>

<th class='sort' name='{$sort['number\_of\_seats']}'id='number\_of\_seats' > Колличество мест {$sort['number\_of\_seats\_arrow']}</th>

<th>Активность</th>

<th>Посмотреть в расписании</th>

</tr>

</thead>

<tbody>

{$data['content']}

</tbody>

</table>

<div> <button class = "page {$page['arrow\_left']}" value="{$page['previous\_page']}">&#60</button> {$page['this\_page']} из {$page['all\_pages']} <button class = "page {$page['arrow\_right']}" value="{$page['next\_page']}">&#62</button> </div>

EOF;

}

Листинг 6

private static function print\_page($content=""){

$body =TEMPL::head();

$body.=self::menu();

$body.= $content;

$body.= TEMPL::footer();

print $body;

}

private static function menu(){

$name\_color=(isset($\_GET['act']))?$\_GET['act']:"";

return TEMPL::menu($name\_color);

}

static public function head(){

return <<<EOF

<html>

<head>

<title>

</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<meta content="no-cache" http-equiv="Pragma"/>

<meta content="no-cache" http-equiv="no-cache"/>

<script src="https://code.jquery.com/jquery-3.1.1.js"></script>

<link rel="stylesheet" href="style/style.css">

</head>

<body>

EOF;

}

static public function footer(){

return <<<EOF

</div>

<div style="clear:both"></div>

<div class="site\_basement"></div>

<script src="script/script.js"></script>

</body>

</html>

EOF;

}

public static function homepage(){

return<<<EOF

<div>

<h1>This is homepage</h1>

</div>

EOF;

}

public static function menu($name\_color=''){

$color\_themes="style='";$color\_time="style='";$color\_orders="style='";$color\_users="style='";

switch($name\_color){

case'event\_themes':

$color\_themes.="color:#ff8d00;";

break;

case'event\_time':

$color\_time.="color:#ff8d00;";

break;

case'orders':

$color\_orders.="color:#ff8d00;";

break;

case'users':

$color\_users.="color:#ff8d00;";

break;

}

switch($\_SESSION['rights']){

case'1':

break;

case'2':

$color\_users.="display:none;";

break;

case'3':

$color\_orders.="display:none;";

$color\_themes.="display:none;";

$color\_users.="display:none;";

break;

}

$color\_themes.="'";$color\_time.="'";$color\_orders.="'";$color\_users.="'";

return <<<EOF

<div class="menu">

<ul>

<li><a {$color\_themes} href="index.php?act=event\_themes">Мероприятия</a></li>

<li><a {$color\_time} href="index.php?act=event\_time">Расписание</a></li>

<li><a {$color\_orders} href="index.php?act=orders">Заказы</a></li>

<li><a {$color\_users} href="index.php?act=users">Пользователи</a></li>

<li>{$\_SESSION['username']} ({$\_SESSION['work\_group']}) <a href="index.php?act=exit">exit</a></li>

</ul>

</div>

<div class="content">

EOF;}

Листинг 7

if(!DB::get\_num\_rows(DB::query("SELECT \* FROM `orders` WHERE `full\_name` = '".GENERAL::$input['name']."' AND event\_time\_id='{$id}'"))){

DB::do\_update('event\_time',array('orders\_count'=>$count++));

DB::do\_insert('orders',array('event\_themes\_id' =>$id, 'full\_name'=> GENERAL::$input['name'],

'date\_of\_birth'=> GENERAL::$input['date\_of\_birth'],

'telephone'=> GENERAL::$input['telephone'],

'mail'=> GENERAL::$input['mail']));

}