**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка веб-приложения для бронирования билетов в театр | | | | | |
|  |  | | | | | |
|  | |  | БР.БИСТ.02.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | В.П. Безухов |  |  |  | канд. техн. наук, доц.  В. Г. Кобылянский |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**С И Б И Р С К И Й Г О С У Д А Р С Т В Е Н Н Ы Й У Н И В Е Р С И Т Е Т П У Т Е Й С О О Б Щ Е Н И Я ( С Г У П С )**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационные системы и технологии» Профиль: Интеллектуальные транспортные системы

|  |
| --- |
| ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные*  *технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
| *« » 2022 г.* |

## З А Д А Н И Е

## на выполнение выпускной квалификационной работы

студенту Безухову Владиславу Павловичу

1. Тема «Разработка веб-приложения для бронирования билетов в театр» утверждена приказом № 203/с от «30» мая 2022 г.
2. Задание выдано «12» мая 2022 г.
3. Срок сдачи законченной работы на кафедру «17» июня 2022 г.
4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики
5. Содержание расчетно-пояснительной записки

|  |  |  |
| --- | --- | --- |
| Наименование разделов и вопросов | Примерное  количество страниц | График (сроки)  выполнения |
| Введение | 4 | 02.05.2022 |
| Анализ предметной области | 12 | 05.05.2022 |
| Описание бизнес-процессов | 10 | 09.05.2022 |
| Проектирование информационной системы | 12 | 11.05.2022 |
| Руководство пользователя | 10 | 16.05.2022 |
| Заключение | 1 | 20.05.2022 |

1. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количеств о  слайдов | График (сроки) выполнения |
| Презентация PowerPoint | 12 | 29.05.2022 |

1. Консультанты по разделам

|  |  |  |
| --- | --- | --- |
| Наименование раздела | Фамилия, И. О. консультанта | Подпись консультанта, дата выдачи задания |
|  |  |  |

|  |  |
| --- | --- |
| Руководитель | В. Г. Кобылянский |
| *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял | В. П. Безухов |
| *(подпись студента)* |  |

УДК 004.41

# АННОТАЦИЯ

В работе 56 страниц, на которых размещены 38 рисунков, 2 таблицы. При написании использовалось 14 источников.

Ключевые слова: *информационная система для театра, автоматизация, веб-сайт, php, база данных.*

Предметная область – веб-приложение для театра.

Объект исследования – информационная система бронирования и продажи билетов в театре.

В работе проведено создание веб-приложения, которое автоматизирует процесс бронирования билетов. Приложение работает с сервером, откуда берет данные о свободном времени для записи.

## ABSTRACT

In the work 56 pages, which contain 38 drawings and 2 tables. When writing, 14 sources were used.

Keywords: *theater information system, automation, website, php, database*.

The subject area is a web application for the theater.

The object of research is an information system for booking and selling tickets in the theater.

The work carried out the creation of a web application that automates the process of booking tickets. The application works with the server, from where it takes data on free time for recording.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

PHP – это язык сценариев общего назначения, ориентированный на веб-разработку.

XAMPP – кроссплатформенная сборка локального веб-сервера.

ИС – информационная система.

АИС – автоматизированная операционная система.

СУБД – комплекс программ, позволяющих создать базу данных.

CSS – мобильная операционная система компании Google.

VISUAL STUDIO CODE – редактор кода для разработки PHP.

SQL (Structured Query Language) – декларативный язык программирования, применяемый для управления данными в реляционной базе данных.

URL (Uniform Resource Locator) – стандарт записи ссылок на объекты в Интернете.

MySQL – система управления реляционными базами данных компании Microsoft.

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc106379462)

[1 Анализ предметной области 10](#_Toc106379463)

[1.1 Общее положение 10](#_Toc106379464)

[1.2 Описание проблемы и постановка задачи 12](#_Toc106379465)

[1.3 Обзор существующих систем автоматизации деятельности 14](#_Toc106379466)

[1.4 Выводы к разделу 22](#_Toc106379467)

[2 Описание бизнес-процессов бронирования билетов в театре 23](#_Toc106379468)

[2.1 Основные понятия 23](#_Toc106379469)

[2.2 Диаграмма деятельности 26](#_Toc106379470)

[2.3 Диаграмма вариантов использования 28](#_Toc106379471)

[3 Проектирование информационной системы 30](#_Toc106379472)

[3.1 Описание архитектуры общей системы 30](#_Toc106379473)

[3.2 Обоснование выбора базы данных 34](#_Toc106379474)

[3.3 Выбор инструментов и технологий для решения задачи 37](#_Toc106379475)

[3.4 Разработка запросов и внешнего вида страниц 38](#_Toc106379476)

[4 Руководство пользователя 43](#_Toc106379477)

[4.1 Руководство для системного администратора по установке программы 43](#_Toc106379478)

[4.2 Руководство для пользователя администратор 47](#_Toc106379479)

[4.3 Руководство для пользователя кассир 49](#_Toc106379480)

[4.4 Руководство для пользователя клиент 51](#_Toc106379481)

[Заключение 54](#_Toc106379482)

[Список использованных источников 55](#_Toc106379483)

[Приложение А исходный код Login 57](#_Toc106379484)

[Приложение Б исходный код Reg 60](#_Toc106379485)

# 

# ВВЕДЕНИЕ

На данный момент любая отрасль человеческой деятельности так или иначе связана с применением электронно-вычислительных устройств, персональных компьютеров, мобильных устройств. Компьютер стал использоваться во многих отраслях – например, в медицине, в бухгалтерии, в прогнозировании погоды, в составлении статистики, в сфере развлечений и услуг, для обучения. Преимущество использования компьютера для обработки информации заключается в его безупречном выполнении инструкций, то есть программного кода. Вследствие чего компьютер не ошибается (при условии, что в него введена правильная программа). Кроме того, компьютерная обработка экономит массу времени для человека, который может потратить данное время на отдых или на улучшение условий своего труда (например, заняться самообразованием или продолжить оптимизировать свой труд путем еще большей автоматизации).

В мире существует множество разнообразных аппаратных и программных платформ. Программное обеспечение сильно привязано к операционным системам (ОС). Это приводит к их несовместимости. Например, программа, разработанная под Windows, не запускается под ОС Linux или под ОС IOS.

Программное обеспечение, как правило, разрабатывается под определенную операционную систему. И это негативно сказывается на совместимости программ, так как приложение, разработанное, например, под Windows отказывается запускаться под операционной системой IOS без дополнительных манипуляций (установка эмулятора Windows). Это вынуждает создавать мультиплатформенные приложения с использованием платформ типа .NET или Java.

Вторым вариантом создавать многоплатформенные приложения – использовать то, что уже изобретено – универсальность веб-браузеров. Веб-браузеры являются обязательным компонентом всех современных операционных систем. А веб-приложение запускается в любом веб-браузере. Следовательно, веб-приложение можно считать мультиплатформенным. Предлагается разработать именно веб-приложение в данном проекте.

В работе будет рассмотрено внедрение веб-приложения.

Театральная сфера актуальна несмотря на то, что на данный момент готовится проект указа президента «Об утверждении Основ государственной политики по сохранению и укреплению традиционных российских духовно-нравственных ценностей», влияющий на театр и кино и о котором некоторые известные актеры отзываются строго негативно, так как он фактически вводит цензуру. Больше всего театральных деятелей в законопроекте возмутило, что, согласно нему, культурно-образовательную сферу возьмут под дополнительный контроль, а также создадут соответствующий координационный орган.

«Государственная поддержка мероприятий в культурно-образовательной сфере нуждается в дополнительном контроле на предмет максимального соответствия целям государственной политики в сфере традиционных ценностей», – говорится в тексте документа. Кроме того, «условием успешной реализации государственной политики в сфере традиционных ценностей» авторы законопроекта видят «создание механизма межведомственной координации», предполагающий создание «органа межведомственной координации» [1].

Бронирование билетов – процесс, требующий внимательности и серьезного подхода. До появления персональных компьютеров бронирование билетов осуществлялось с помощью записи на бумажных носителях. В связи с этим присутствие человеческого фактора очень часто приводило к коллизиям, когда два или несколько человека бронировали на один и тот же спектакль одно и то же место. Приходилось предлагать этим людям другие места, либо договариваться каким-то другим образом, чтобы уладить ситуацию. С появлением персональных компьютеров стало возможным вести учет не на бумаге, а в электронном виде, с помощью системы управления базами данных. В этом случае база данных отвечала за нахождение свободных мест, а также выдавала бы сообщение, если произошла попытка бронировать места, которые уже оказались забронированы на запрашиваемый спектакль. Применение компьютерных технологий поможет снизить влияние человеческого фактора на процесс проверки брони на место и ускорить процесс бронирования в целом [2].

В данной работе необходимо разработать предложения по автоматизации для продажи билетов в театре.

Театр – коммерческое предприятие со зрительным залом, оборудованным для показа спектаклей. В зале располагается сцена и зрительные места.

С точки зрения функционирования или структуры театра, можно сказать, что он располагает зрительными местами с разным уровнем сервиса, комфортности и, соответственно, оплаты. Места могут быть разных типов [3]:

* (VIP) – самые дорогие места с максимально комфортными для просмотра условиями;
* (Comfort) – место меньшей, чем VIP, стоимости и комфортности, находящиеся в зоне наилучшего обзора, более удобные и соответственно дорогие чем C;
* (Normal) – наиболее экономные места, без каких-либо выраженных преимуществ. В театре ведется учет состояния зрительных мест.

Все клиенты, желающие приобрести билет, должны указать, на какое представление они хотят его приобрести и класс зрительного места, оплатить стоимость билета.

Любое место зрительного зала имеет номер, по которому ведется учет занято оно или свободно для продажи.

Также театр предоставляет возможность бронирования билетов.

Таким образом, в функционирование театра входит:

* продажа билетов;
* контроль наполняемости зала;
* предоставление информации о репертуаре театра;
* услуги бронирования билетов и снятия брони;
* возврат билетов.

Объектом исследования является молодежный театр «Драма».

Предмет – автоматизированная информационная система для театра.

В данном сервисе можно будет вести учетов представлений и клиентов, бронировать билеты на представления.

Цель работы – оптимизация процессов управления работой театра за счет разработки автоматизированной информационной системы.

Для достижений указанной цели придется решить следующие задачи:

* проанализировать имеющиеся аналоги программного обеспечения и сформировать требования к разрабатываемому приложению;
* смоделировать бизнес-процессы;
* выбрать инструменты реализации проекта;
* реализовать автоматизированную информационную систему.

Данная работа актуальна, так как предполагает создание нового программного обеспечения с необходимым для решения задачи функционалом.

# 1 Анализ предметной области

## 1.1 Общее положение

Организация представляет собой театр, который организует представления для детей и взрослых, реализует билеты на них. Театр организует показы представлений для различных слоев населения. Реализацией билетов занимается касса. Сейчас продажа билетов осуществляется через телефон, а после идет запись в систему, что неудобно, так как приходится вручную считать итоговые суммы продаж. Хотелось бы реализовать возможность продажи или хотя бы бронирования билетов на представления через интернет. Необходимо внедрить личный кабинет пользователей, чтобы пользователи могли зарегистрироваться в программе и иметь личный кабинет с корзиной или подобным функционалом. Кроме того, необходимо сделать систему индивидуальных скидок для пользователей.

Под театром понимается организация, основной деятельностью которой является подготовка и показ спектаклей, других публичных представлений и предоставление сопутствующих этому услуг в целях формирования и удовлетворения потребностей населения в сценическом искусстве.

Театр является некоммерческой организацией, то есть прибыль не является целью его деятельности. Согласно положению правительства о принципах финансирования государственных и муниципальных театров в РФ финансирование должно обеспечить условия для предотвращения коммерциализации театра. Вместе с тем театр имеет право самостоятельно формировать свою экономическую программу, определять порядок реализации билетов, услуг и продукции, устанавливать на них цены, заниматься предпринимательской деятельностью в ограниченных рамках, вести внешнеэкономическую деятельность, иметь валютные счета в банках. Театр самостоятельно распоряжается продуктами своего интеллектуального труда.

Учредителями театра могут быть федеральные органы исполнительной власти, органы исполнительной власти субъектов РФ, органы местного самоуправления, гражданские и юридические лица, в том числе иностранные, и лица без гражданства. Учредитель утверждает устав, назначает на должность и освобождает от должности руководителей театра, осуществляет контроль. Руководство деятельностью государственного или муниципального театра осуществляется по выбору учредителя. Устанавливается либо единоначалие (в лице директора или художественного руководителя), либо разделение сфер влияния между директором и художественным руководителем.

К основным видам деятельности театра относятся:

* создание и показ спектаклей, организация гастролей, концертов, проведение творческих вечеров, фестивалей и конкурсов;
* подготовка спектаклей, концертов, представлений по договорам с другими юридическими и физическими лицами для показа на их собственных площадках, по телевидению и так далее;
* организация других мероприятий художественно-творческого характера;
* проведение стажировок ведущими мастерами и деятелями театра;
* предоставление другим организациям по договорам постановочных услуг;
* изготовление по заказам предметов художественного оформления спектаклей, концертов, представлений;
* подготовка, тиражирование и реализация информационно-справочных изданий, копий видеоматериалов и фонограмм;
* прокат и реализация костюмов, обуви, оборудования, реквизита, бутафории, гримерных и так далее.

Театр вправе осуществлять предпринимательскую деятельность лишь постольку, поскольку это служит целям, для которых он создан. К предпринимательской деятельности театра относятся:

* реализация и сдача в аренду основных фондов и имущества театра для целей, не связанных с культурной деятельностью;
* оказание посреднических услуг;
* долевое участие в деятельности коммерческих организаций;
* приобретение акций, облигаций и иных ценных бумаг;
* деятельность, приносящая доход и не предусмотренная уставом, соответствующая целям создания театра.

## 1.2 Описание проблемы и постановка задачи

Данная выпускная квалификационная работа направлена на создание прототипа информационной системы для сотрудников театра, для того чтобы автоматизировать процесс бронирования билетов. На данный момент бронирование билетов в Новосибирском молодёжном театре драмы осуществляется с помощью ведения учета брони на бумажных носителях. Из-за этого присутствует человеческий фактор, что часто приводит к неприятным коллизиям: два или более клиентов бронируют билеты на одно и то же место внутри одного спектакля.

На рисунке 1.1 представлена модель «AS IS» (с англ. «как есть») в виде контекстной диаграммы.

Основными действующими лицами в процессе бронирования являются клиент и кассир. Клиент никак не может повлиять на процесс бронирования, кроме как сообщить свою контактную информацию. Основную ответственность за правильность бронирования несет кассир. Если он ошибся, это будет исключительно его вина.

Если автоматизировать процесс бронирования таким образом, чтобы клиент мог сам инициировать процесс бронирования, в том числе без участия кассира, это приведет к разгрузке кассира и создаст максимальное удобство для пользователя (так как он сможет в любое время самостоятельно зайти в приложение и выбрать представление и место в соответствии со своими личными предпочтениями). Особенно это важно для клиентов-интровертов, которые испытывают дискомфорт при личном общении или даже при общении с людьми по телефону.

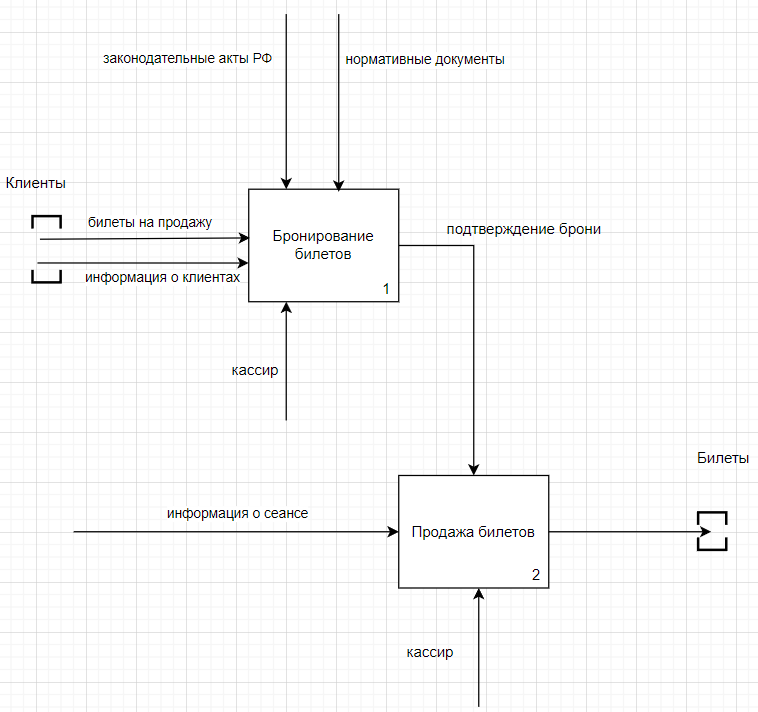


Рисунок 1.1 – Контекстная диаграмма IDEF0

Процесс продажи билетов всегда состоит из двух этапов: бронирование и продажа. Поэтому при продаже билета, бронирование происходит фактически в момент выбора и печати билета.

Декомпозиция бронирования и продажи билетов выполнена в виде контекстной диаграммы и отображает более детально происходящие процессы в кассе театра. Данная диаграмма представлена на рисунке 1.2.

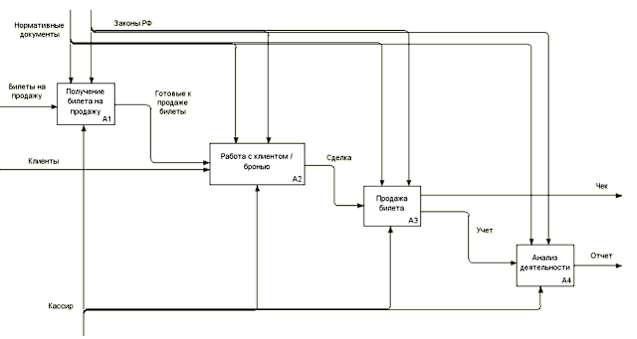


Рисунок 1.2 – Контекстная диаграмма декомпозиции бронирования и продажи билетов

Необходимо разработать автоматизированную информационную систему, которая автоматизирует:

* отображение актуальных представлений с описанием, датой события, ценой на билеты;
* регистрацию и учет пользователей;
* бронирование и/или продажу билетов;
* управление акциями и скидками для отдельных клиентов.

## 1.3 Обзор существующих систем автоматизации деятельности

Для того, чтобы определить каким критериях должно соответствовать веб-приложение, были проанализированы 10 различных сервисов предоставляющие услуги по бронированию, а также содержащая информацию о предоставляемой услуге.

На рисунке 1.3 приведено изображение сайта культурного агентства «Энергия».

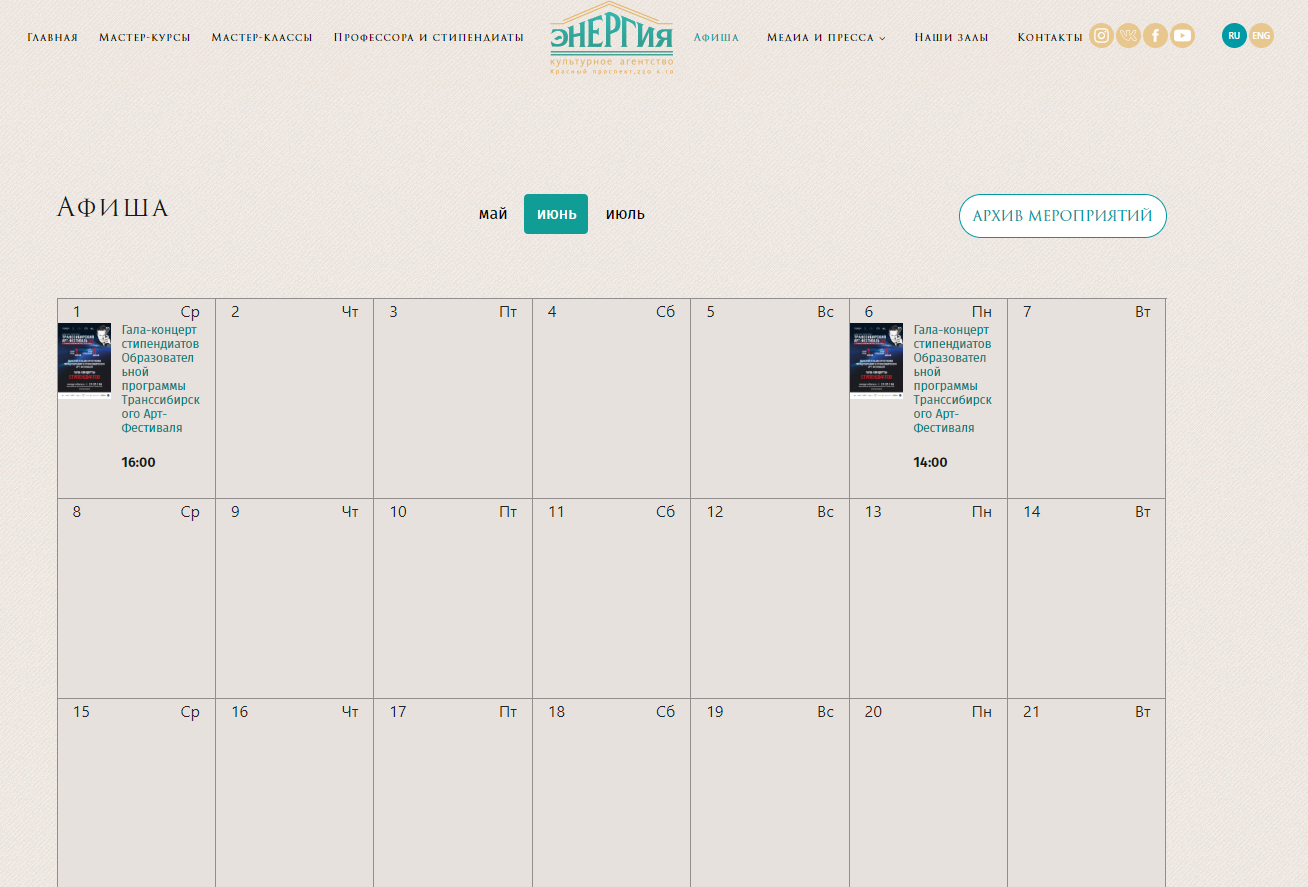


Рисунок 1.3 – Энергия

Сервис для организации мероприятий в области искусства. Среда «Энергия» служит для организации театральных, музыкальных, хореографических и образовательных кинопроектов, презентаций и корпоративных мероприятий. Бронь осуществляется по телефону.

На рисунке 1.4 приведено изображение главного меню театра «Среда».



Рисунок 1.4 – Среда

Данный Театр находится в городе Новосибирск. Сайт позволяет клиентам выбрать нужный им спектакль и осуществить бронирование билета, а также ознакомиться со всей необходимой информацией по телефону.

На рисунке 1.5 приведено изображение окна сервиса Единое поле.

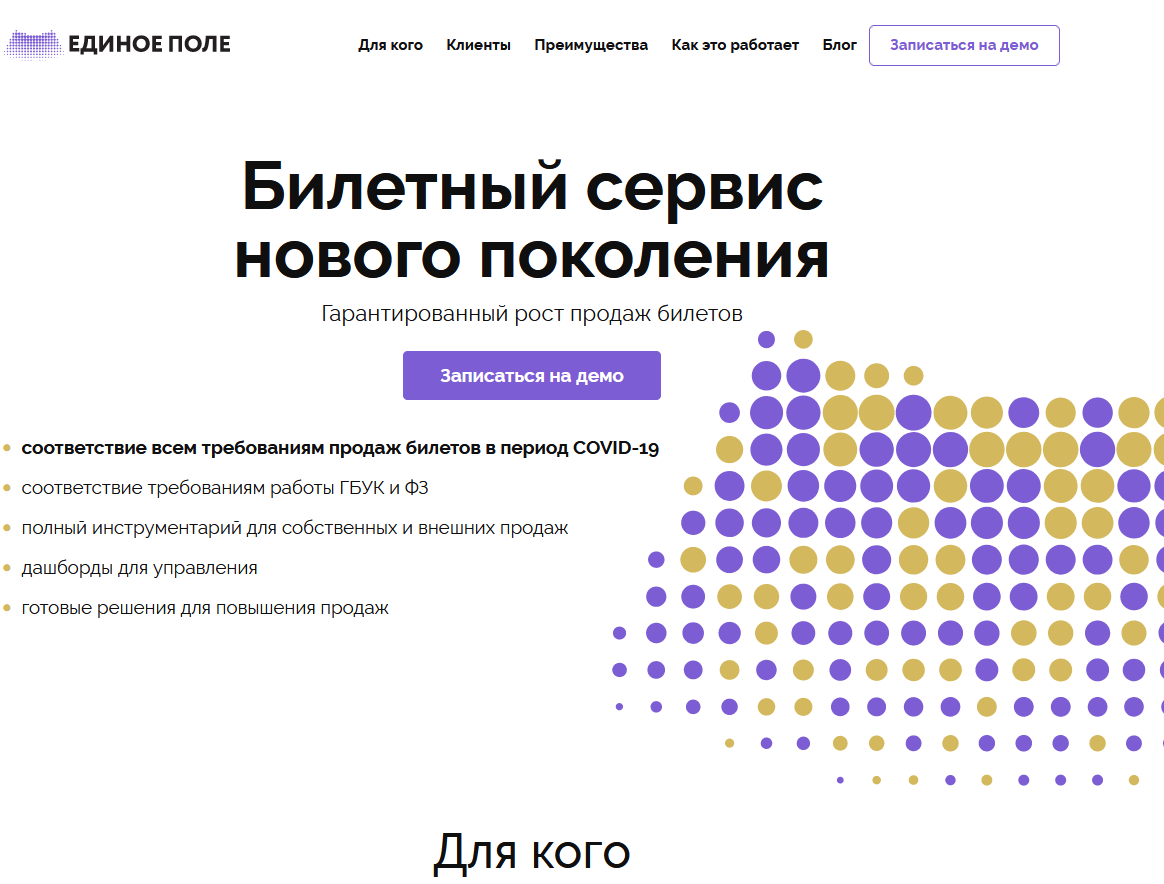


Рисунок 1.5 – Единое поле

«Единое поле» является билетным системой предоставляющий свои услуги, объединив в себе 3 направления: билетную систему, внешние продажи и собственные продажи. Если у клиентов есть база зрителей, но они с ней не работают. Клиенты хотели бы запустить рекламу в интернете, но у них нет специалистов. Есть намерение сделать сайт лучше, собирать и работать с офлайн-аудиторией – нет ресурсов, тогда данных сервис может с этим помочь. Присутствует простая и бесплатная интеграция.

На рисунке 1.6 приведено изображение окна сервиса TicketsCloud.

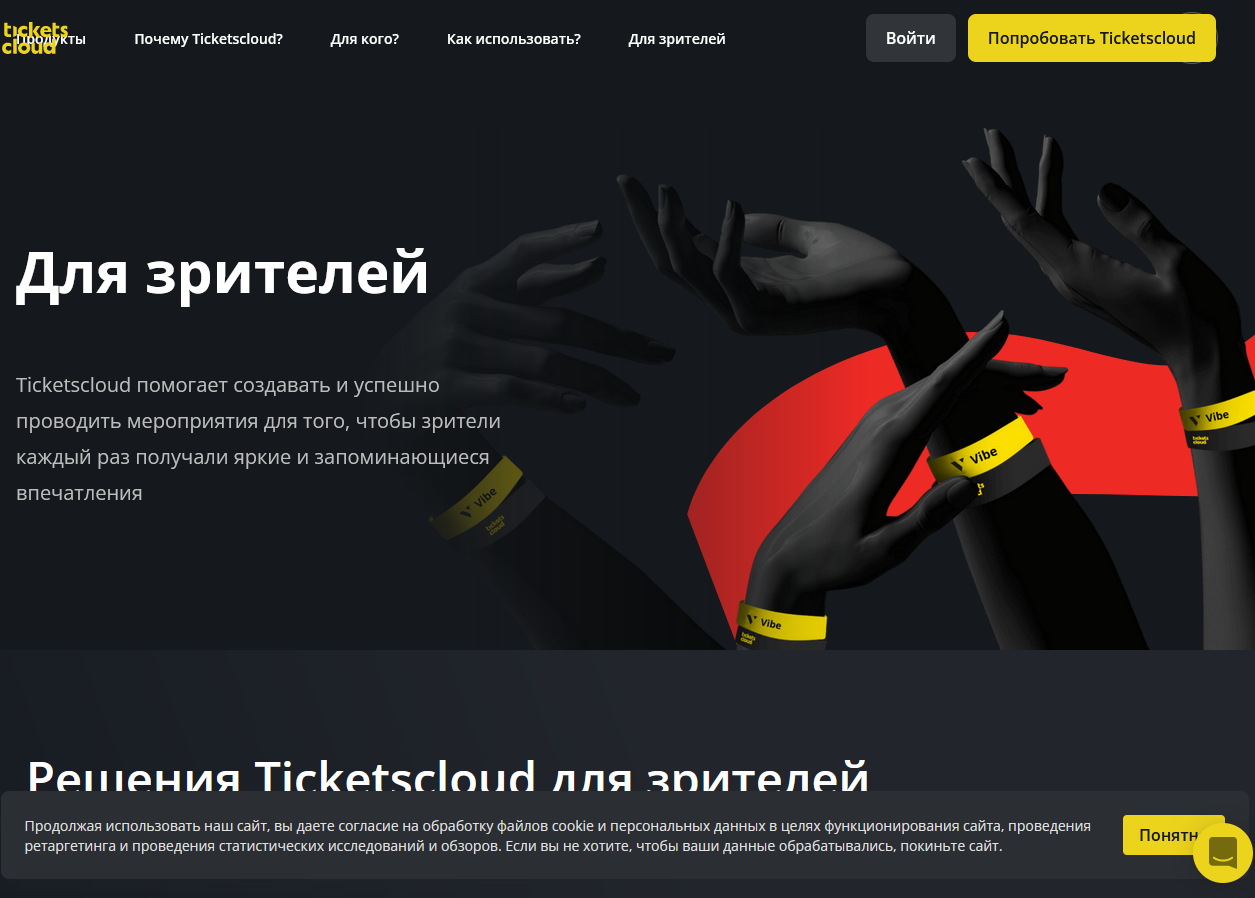


Рисунок 1.6 – TicketsCloud

Облачная система продажи билетов. Можно продавать билеты на любых сайтах или в социальных сетях. Один договор с платформой для работы со всеми партнерами. Онлайн доступ к денежным средствам с возможностью моментального вывода.

На рисунке 1.7 приведено изображение окна сайта театра кукол «Пилигримы».

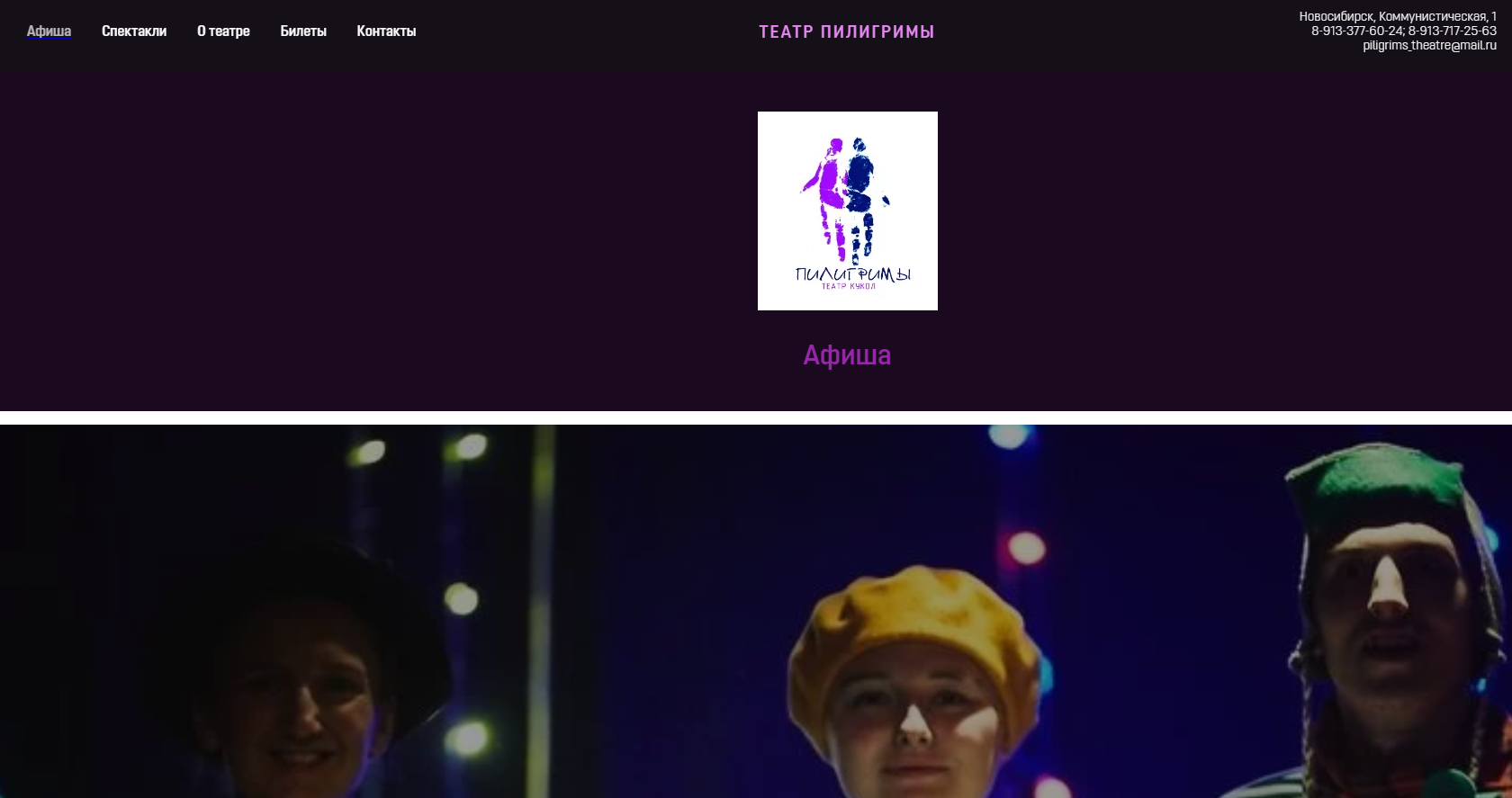


Рисунок 1.7 – InTickets

Сайт для учета и брони билетов. Позволяет посмотреть нужную информацию по предстоящему представлению. Оплата происходит через отдельные сервис TimePad или на странице в VK.

На рисунке 1.8 приведено изображение окна сервиса TicketSteam.

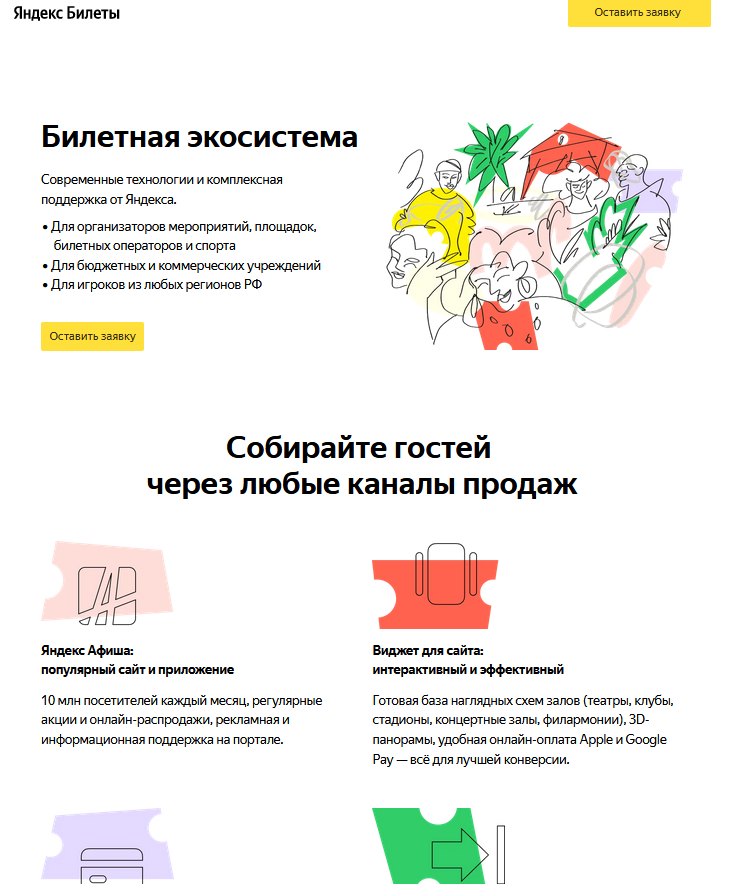


Рисунок 1.8 – TicketSteam

Профессиональная билетная система. Управление кассами. Учет продаж партнеров. Виджет для сайта. Конструктор сайта. Мобильное приложение.

На рисунке 1.9 приведено изображение окна сервиса Qtickets.

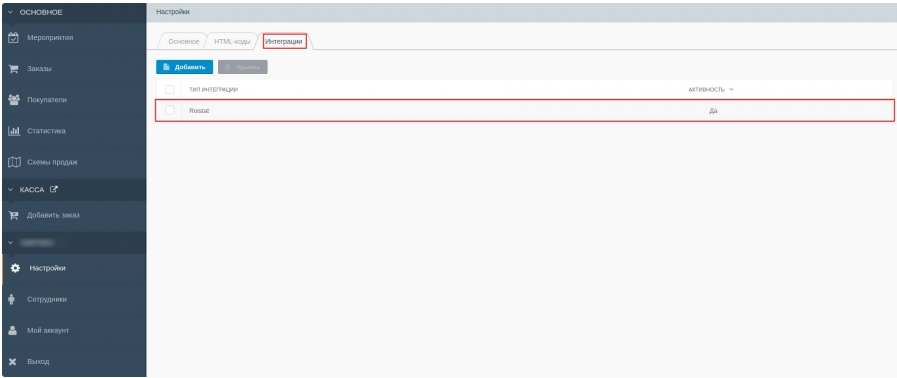


Рисунок 1.9 – Qtickets

Сервис для организаторов, площадок, концертных и рекламных агентств. Позволяет распространять, продавать и контролировать все билеты, автоматизировать рутину, снижать налоговую нагрузку, собирать информацию о покупателях, моментально получать деньги и отчётность в режиме реального времени. Покупатели могут оплатить билеты с помощью Apple Pay, Google Pay, банковской карты, наличными через терминалы, салоны связи или по счёту. Для клиентов разработаны различные тарифы, с которыми можно ознакомиться на сайте.

На рисунке 1.10 приведено изображение окна сайта Новат.

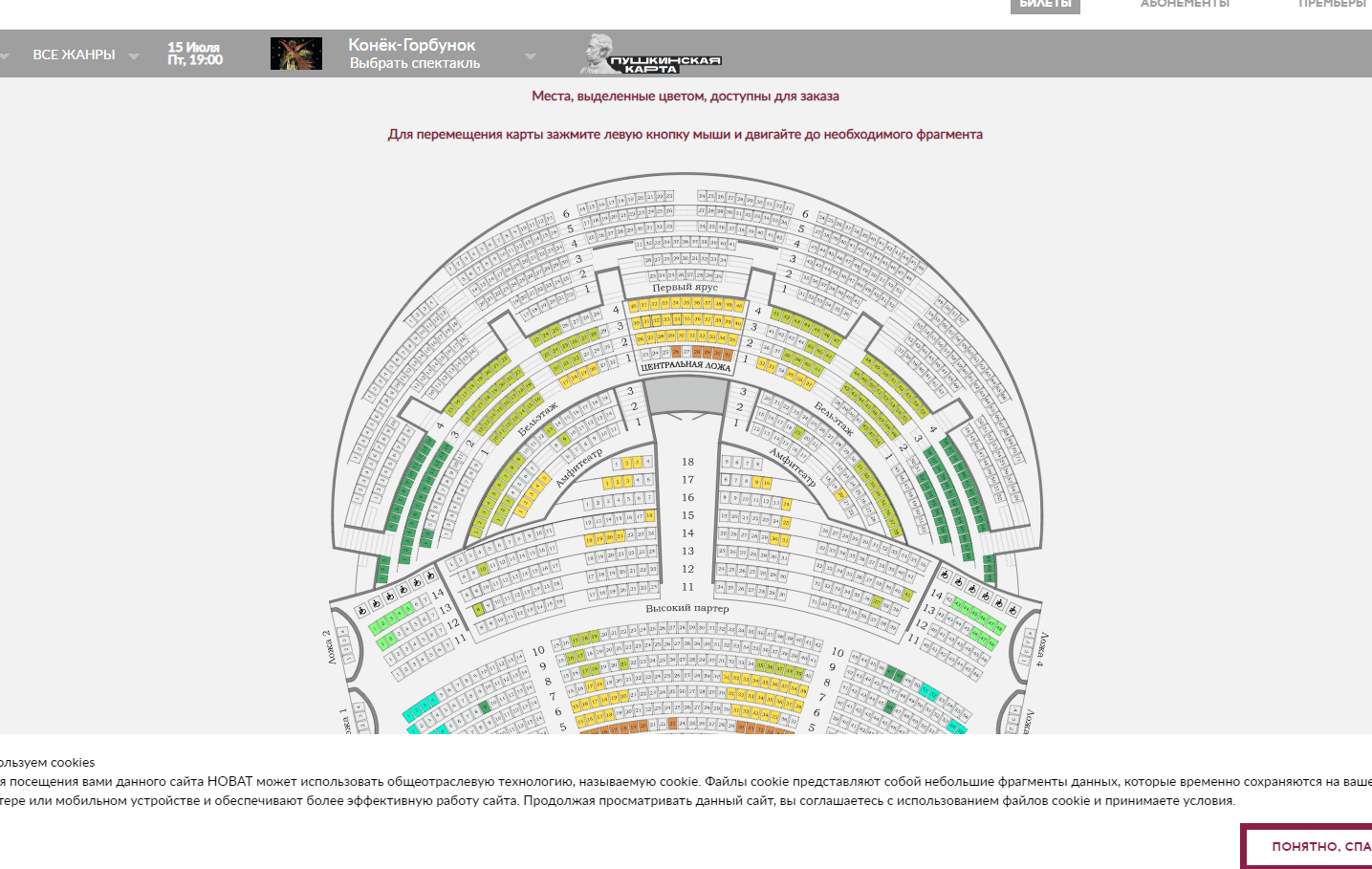


Рисунок 1.10 – Online tickets

Система, позволяющая автоматизировать процесс бронирования и покупки билетов в театр. Можно посмотреть подробное описание любого спектакля, а также информацию о театре. Сайт позволяет покупать билеты по «Пушкинской карте», что приведет к увеличению посещаемости. На сайте присутствует система регистрации и авторизации пользователей, что позволит пользователям в любое удобное время авторизироваться на сайте и просмотреть дату и время купленных ими билетов.

На рисунке 1.11 приведено изображение окна сервиса Чудобилет.



Рисунок 1.11 – Чудобилет

Система продажи билетов, позволяющая на базе единой платформы автоматизировать кассы предприятий культурно-массовой и спортивно-зрелищной сферы: кинотеатров, театров, филармоний, музеев, клубов, стадионов и многих других.

На рисунке 1.12 приведено изображение окна сервиса TicketTool.net.

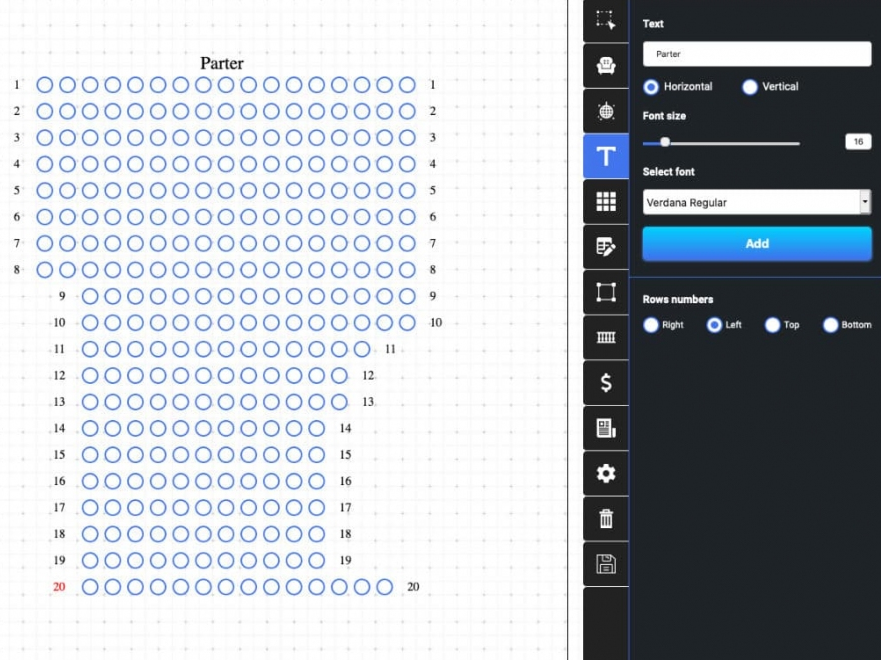


Рисунок 1.12 – TicketTool.net

Система онлайн бронирования и продажи билетов. Электронные билеты. Система онлайн оплаты с низкой комиссией.

Сравнительный анализ приведен в таблице 1.1.

Таблица 1.1 – Сравнительный анализ

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Характеристика | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Многоязычность | - | - | - | - | - | - | - | - | - | - |
| Личный кабинет и регистрация | - | - | + | + | + | + | + | + | + | + |
| Возможность выбора места в зале | + | - | + | + | + | - | + | + | + | + |
| Возможность онлайн-оплаты | - | - | + | + | + | + | + | + | + | + |
| Простота интерфейса | + | + | + | + | - | + | + | + | + | + |

## 1.4 Выводы к разделу

Разработанная АИС для молодежного театра драмы должна быть выполнена на одном языке (русском), иметь возможность регистрации и авторизации с переходом в личный кабинет. Программное обеспечение должно позволять бронировать билеты и обладать простым интерфейсом. Подключение онлайн-оплаты реализовать в учебном проекте невозможно, так как для подключения платежных систем необходимо иметь зарегистрированное юридическое лицо.

# 2 Описание бизнес-процессов бронирования билетов в театре

## 2.1 Основные понятия

В таблице 2.1 приведены проблемы, с которыми сталкивается театр. Данные проблемы будут решены с помощью АИС.

Таблица 2. 1 – Проблемы

|  |  |
| --- | --- |
| Проблема | В театре отсутствует возможность вести автоматизированный учет проданных билетов |
| Воздействует на | Кассиры, посетители театра, бухгалтеры |
| Результатом чего является | Приходится вести учет проданных билетов вручную |
| Сейчас решается | Работы ведутся вручную с использованием офисных приложений типа Microsoft Excel |
| Имеющиеся решения | Большое количество ошибок за счет человеческого фактора |

Будет разработана автоматизированная информационная система, которая автоматизирует:

* отображение актуальных представлений с описанием, датой события, ценой на билеты;
* регистрацию и учет пользователей;
* бронирование и/или продажу билетов.

На рисунке 2.1 приведена карта взаимодействий.

Назначение – автоматизация продажи билетов в театре.

Обзор проекта и предпосылки. Разрабатываемое приложение должно:

* поддерживать регистрацию и авторизацию пользователей (администратора, кассира, посетителей) с разными правами доступа;
* иметь разный интерфейс в зависимости от прав доступа;
* быть выполнено в минималистическом стиле, с неброским дизайном.

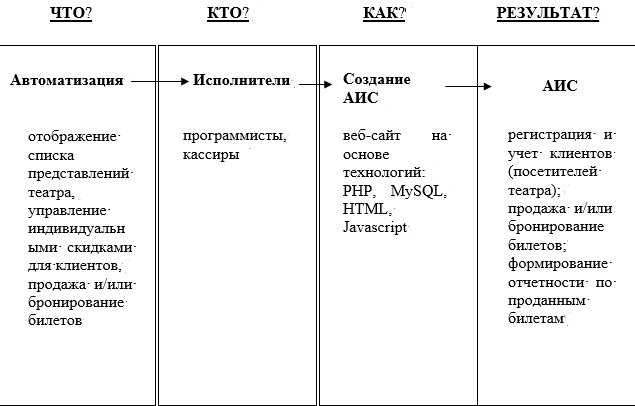


Рисунок 2.1 – Карта взаимодействий

Зависимости проекта. Зависимостей от другого программного обеспечения не предусматривается.

Заинтересованные стороны:

* посетители;
* кассиры.

Основные предположения и допущения:

* АИС будут пользоваться русскоговорящие пользователи;
* АИС будут пользоваться люди без специальных знаний компьютерной грамотности.

Ограничения.

Веб-сервер будет работать на PHP7 с библиотекой MySQLi, поэтому обязательно необходима поддержка в коде данного сочетания языка программирования и используемых функций библиотеки MySQLi [4].

Не использовать CMS.

Не использовать платные библиотеки и прочие модули.

Отсутствует официальная регистрация данной организации в виде юридического лица на данном этапе, поэтому подключение платежных систем реализовывать не нужно.

На рисунке 2.2 приведена контекстная диаграмма IDEF0 «To-be» («Как будет»).

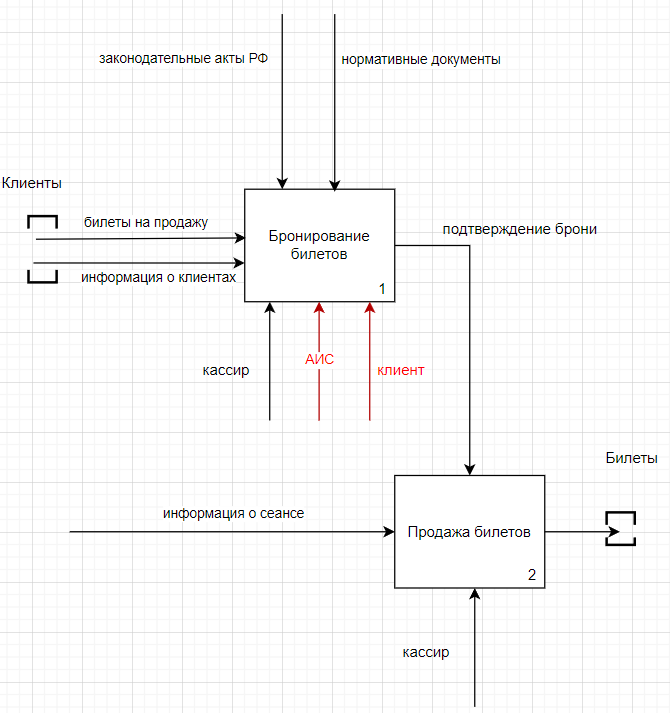


Рисунок 2.2 – Контекстная диаграмма IDEF0

На данной диаграмме красным цветом выделены те части, которые претерпели изменения по сравнению с существующей моделью «AS IS».

Клиент в данном случае самостоятельно регистрируется на сайте, входит и бронирует себе необходимые места с помощью автоматизированной информационной системы. Участие кассира в этом процессе не предусматривается, так как процесс полностью автоматизирован и реализован с помощью веб-приложения, доступного через сеть интернет. Однако кассир имеет возможность отменять бронь, либо же бронировать места за пользователей, которые занимали места по телефону.

## 2.2 Диаграмма деятельности

На рисунке 2.3 приведена диаграмма последовательности для процесса взаимодействия пользователя с веб-сайтом, ввод запроса и получения ответа.

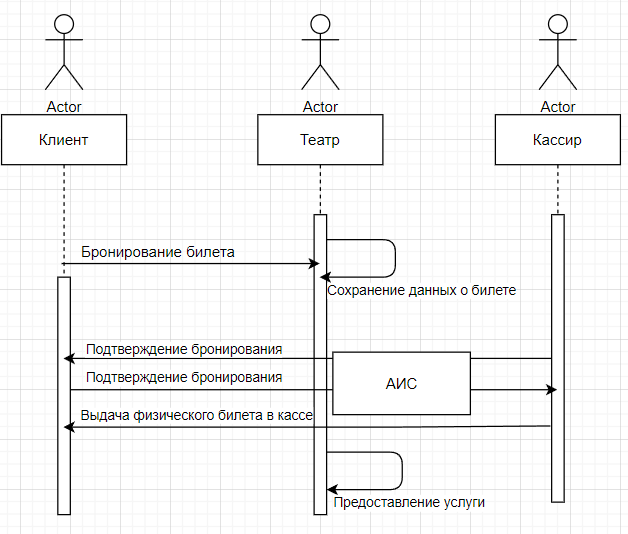


Рисунок 2.3 – Диаграмма последовательности для процесса взаимодействия пользователя с веб-сайтом

На ней видны процессы покупки билета, движения информации от одного действующего лица к другому.

На рисунке 2.4 приведена диаграмма деятельности с этапами бронирования и продажи билета.

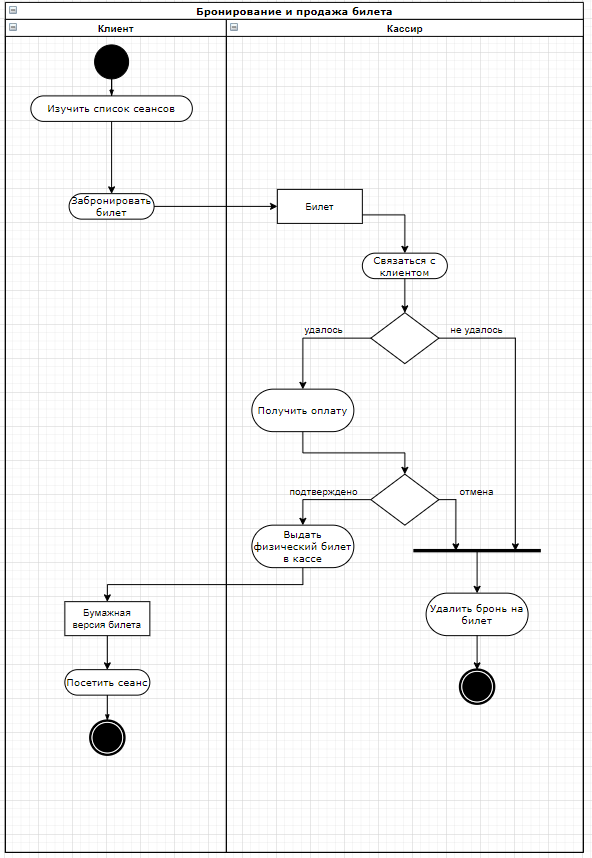


Рисунок 2.4 – Диаграмма деятельности с этапами бронирования и продажи билета

В ней описан принцип работы театра с точки зрения двух ролей – покупателя и кассира – как двух участников бизнес-процессов рассматриваемого предприятия. Процесс продажи фактически состоит из трех этапов:

1. Этап бронирования билета – данный этап доступен клиентам (посетителям театра) с помощью веб-приложения в онлайн-режиме (пользователь может забронировать билет, находясь дома или в дороге по пути в театр).
2. Этап оплаты забронированного билета.
3. В случае отказа от оплаты забронированного билета, кассир или администратор АИС может удалить бронь, тем самым забронированное место становится доступным для бронирования другим пользователем.

## 2.3 Диаграмма вариантов использования

На рисунке 2.5 приведена диаграмма вариантов использования (другое название – «диаграмма прецедентов»). На ней хорошо видны перечисленные группы пользователей.

Администратор может регистрировать в ручном режиме (в том числе и создавать) новых пользователей с любым уровнем доступа.

Кассир имеет весь остальной функционал администратора кроме доступа к созданию новых пользователей.

Пользователь наиболее ограниченный по функционалу класс пользователей программы.

После регистрации пользователь получает возможность авторизоваться на сайте. Он может переходить по приложению, просматривать список представлений, выбирать места в зале на определенный выбранный спектакль и осуществлять покупку или бронирование билета.

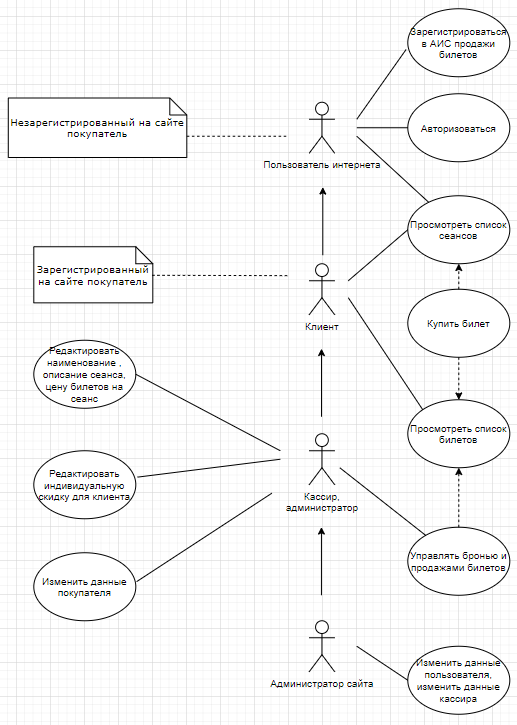


Рисунок 2.5 – Диаграмма прецедентов

Таким образом, можно сделать вывод, что необходимо разработать приложение для продажи билетов, которое будет выполнено (иметь интерфейс) на одном языке, иметь личный кабинет и регистрацию, иметь возможность выбора места в зале, иметь возможность онлайн-оплаты, простоту интерфейса.

# 3 Проектирование информационной системы

## 3.1 Описание архитектуры общей системы

АИС представляет собой веб-приложение, которое должно выполняться на веб-сервере в сети интернет. При отсутствии подключения к интернету, пользователи приложением воспользоваться не смогут.

Разрабатывается АИС в виде веб-приложения. Веб-приложение будет работать на сервере. Подключение к серверу осуществляется через интернет. Приложение будет открываться в браузере как обычный веб-сайт по определенному веб-адресу [5].

Описание возможных технических решений

В качестве инструментов будут использоваться технологии и языки: HTML 5.0, Javascript, CSS, PHP 7. В качестве веб-сервера будет Apache.

Для разработки на операционной системе Microsoft Windows 10, в которой велась работа, очень удобно использовать готовую сборку XAMPP 3.2.2. В состав этой сборки входит настроенный веб-сервер Apache с языком PHP и базой данных MySQL. Необходимо лишь установить данную программу.

Проектирование базы данных.

Проектирование базы данных удобно производить с помощью специализированного программного обеспечения. Одним из представителей этого класса программ является MySQLWorkBench. С помощью данного приложения можно с легкостью создать архитектуру базы данных, в том числе, рисунок 3.1 [6]:

* сформировать структуру таблиц;
* указать наименования и типы данных в таблицах;
* нарисовать схему БД;
* нанести связи между таблицами.

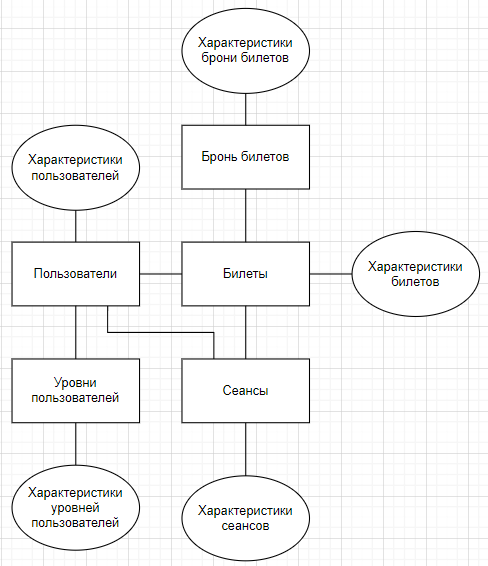


Рисунок 3.1 – Инфологическая модель

Как видно из данного рисунка, основными сущностями являются:

* пользователи;
* представления;
* билеты;
* продажи билетов;
* уровни пользователей.

Пользователи бывают нескольких типов в зависимости от необходимого уровня доступа. Далее эти уровни доступа будут влиять на возможности работы с функционалом сайта (у администратора – максимальный уровень доступа, у рядового пользователя – минимальный уровень доступа). В АИС имеются следующие уровни доступа:

* администратор;
* кассир;
* пользователь.

Администратор может регистрировать в ручном режиме (в том числе и создавать) новых пользователей с любым уровнем доступа.

Кассир имеет весь остальной функционал администратора кроме доступа к созданию новых пользователей.

Пользователь наиболее ограниченный по функционалу класс пользователей программы.

Состав веб-приложения приведен на рисунке 3.2.

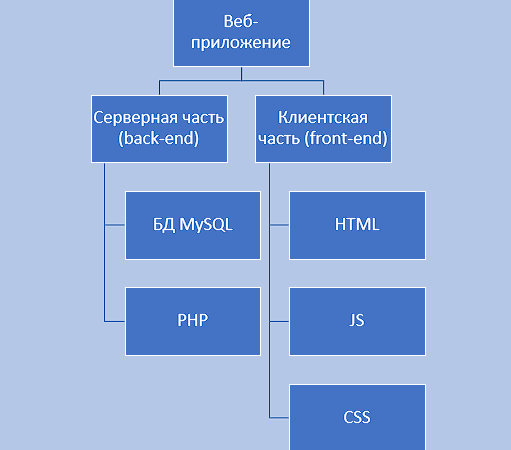


Рисунок 3.2 – Взаимодействие компонентов системы

В создании АИС не используются специальные модели, такие как MVC (Model-view-controller), так как содержимое страниц несложное и строится по одному шаблону. Каждая страница состоит из четырех основных частей [7]:

* заголовочная часть (файл top.php);
* меню (menu.php);
* основная часть (контент) (файл – в зависимости от страницы имеет соответствующее название);
* нижняя часть («подвал») (footer.php).

Так как каждая часть находится в отдельном файле (top.php, menu.php, footer.php), можно быстро изменить интерфейс приложения (стили оформления содержатся в файле style.css).

При разработке АИС необходимо:

* придерживаться единообразия интерфейса;
* придерживаться единообразия обработки исключительных ситуаций (обработка ошибок);
* унифицировать работу со справочниками;
* разграничить права доступа разных групп пользователей.

Из диаграммы развертывания, рисунок 3.3, видно, что АИС состоит из трех компонентов:

* клиентская часть;
* серверная часть;
* БД.

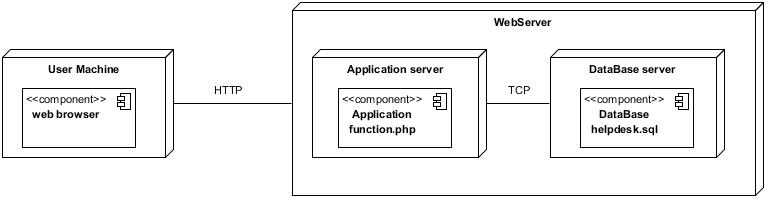


Рисунок 3.3 – Диаграмма развертывания

## 3.2 Обоснование выбора базы данных

Для работы с большими объемами информации пользуются понятиями:

* СУБД – система управления базой данных;
* SQL ­– язык написания запросов к базе данных.

СУБД бывают разные. Например, в зависимости от типа расположения:

* серверные;
* локальные.

Часто используются следующие СУБД:

* Microsoft Access – приложение входит в состав Microsoft Office – набор офисных программ. Это простое в освоении приложение, которое позволяет создавать базы данных разной сложности, однако не имеет инструментов для одновременного (многопользовательского доступа) [8];
* Oracle ­­– известная БД одноименной корпорации. Характеризуется высокой сложностью, специалисты по данной СУБД редки и их услуги очень дороги. Oracle обладает высокой надежностью и производительностью, является одним из лидеров в области СУБД;
* SQLite – свободно распространяемая СУБД. Она характеризуется простотой в эксплуатации, высокой производительностью. Используется, например, в браузере Mozilla Firefox для хранения всех данных [9];
* серверные базы данных MySQL, PostgreSQL – часто используемые в веб-приложениях СУБД. MySQL наиболее распространена на веб-хостингах в интернете и является стандартом де-факто в веб-разработках различного уровня сложности. По MySQL имеется множество литературы, она обладает неплохой производительностью, кроме того, она бесплатная. MySQL поддерживается на уровне системных библиотек в языке программирования PHP [10].

В данной работе выбрана MySQL как бесплатная, наиболее простая и распространенная среди СУБД для веб-разработки.

Разработанная схема база данных приведена на рисунке 3.4.



Рисунок 3.4 – Схема базы данных

На схеме БД видны следующие сущности:

* events – справочник представлений;
* levels – справочник уровней доступа;
* tickets – билеты;
* users – пользователи;
* sales – продажи.

В каждой сущности имеется её структура (сущности), которые позволяют хранить всю информацию в таблицах.

Структуры таблиц приведены на рисунках 3.5-3.9.



Рисунок 3.5 – Структура таблицы events

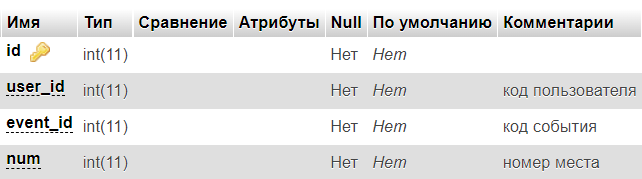


Рисунок 3.6 – Структура таблицы tickets

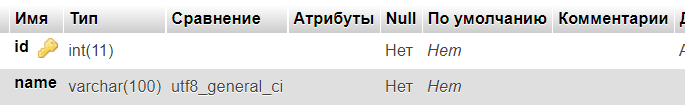


Рисунок 3.7 – Структура таблицы levels



Рисунок 3.8 – Структура таблицы sales



Рисунок 3.9 – Структура таблицы users

В данных таблицах подробно видны атрибуты, и их типы данных, которые содержатся в сущностях. Это позволит хранить всю необходимую информацию в базе данных.

## 3.3 Выбор инструментов и технологий для решения задачи

PHP – один из серверных языков программирования. Язык работает по принципу интерпретатора, а не компилятора, как работают многие другие языки программирования: C, С++, Delphi, Python и другие. Принцип его работы, следующий: интерпретатор получает текстовый файл (текст программы), затем обрабатывает и выдает результат в виде веб-страницы в HTML-формате. Данный язык один из самых широко используемых в мире веб-программирования, благодаря чему он присутствует почти на всех веб-хостингах в сети интернет [11].

Итак, в качестве инструментов будут использоваться технологии и языки: HTML, Javascript, CSS, PHP. В качестве веб-сервера будет Apache.

Настройка веб-сервера – сложный процесс, требующий высокой квалификации. Поэтому часто используют готовые образы или сборки операционной системы с уже установленным и правильно настроенным веб-сервером, и базой данных. Для ОС Microsoft Windows 10, существует приложение под названием XAMPP. В нем уже все настроено для совместной работы всех модулей с корректными настройками. Рекомендуется для начинающих программистов, так как не требует высокой квалификации для установки и настройки веб-сервера и сопутствующих веб-приложений [12].

## 3.4 Разработка запросов и внешнего вида страниц

Код, который отвечает за бронирование билета в файле confirm\_pay.php:

$event\_id=abs(intval($\_POST['hidden\_event\_id']));

$query="

INSERT INTO sales

SET

user\_id=$\_SESSION[id],

event\_id=$event\_id,

sum=$cnt

\*(SELECT price FROM events WHERE id=$event\_id LIMIT 1) # цена за 1 билет

\*ROUND(1-

(SELECT discount\_value FROM users WHERE id=$\_SESSION[id] LIMIT 1) # величина индивидуальной скидки

/100,2)

";

$res=mysqli\_query($con, $query) or die(mysqli\_error($con));

if ($res) {

echo "<p>Успешно забронировано билетов: $cnt. Ваши билеты можно просмотреть <a href='tickets.php'>здесь</a><br>

Суммы покупок <a href='sales.php'>здесь</a></p>";

};

};

В переменной $query запрос в БД, который осуществляет запись в таблицу sales. Данная таблица отвечает за хранение информации о ценах забронированных билетов. Её атрибутами являются пользователи, выбранные спектакли и сумма продажи. Таблица с этими с данными приведена на рисунке 3.10.



Рисунок 3.10 – Таблица sales

Запросы в остальных скриптах зачастую формируются «налету», внутри PHP-скриптов, поэтому в данном разделе не приводятся.

Разработка внешнего вида страниц.

Каждая страница строится по шаблону, который состоит из заголовка, меню, общей информации и подвала веб-приложения. Данный внешний вид веб-приложения является удобным для использования пользователем, поскольку имеет вид большинства веб-сайтов.

Схематически внешний вид каждой страницы построен по шаблону как на рисунке 3.11.

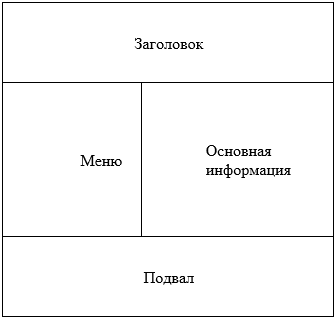


Рисунок 3.11 – Схема шаблона страницы

Заголовок содержит баннер со ссылкой на главную страницу.

Подвал сайта имеет в себе реквизиты создателей сайта.

Меню для каждого уровня пользователей свое: для кассиров и администраторов оно кардинальным образом отличается от меню пользователей-клиентов.

После авторизации с правами администратора системы отображается страница как на рисунке 3.12. Стоит отметить, что для защиты аккаунтов пользователей, необходимо использовать алгоритм хеширования. Таким образом пароли пользователей будут зашифрованы в битовую строку определенной блин. Был выбран алгоритм MD5 [13]. Код представлен на рисунке 3.13.

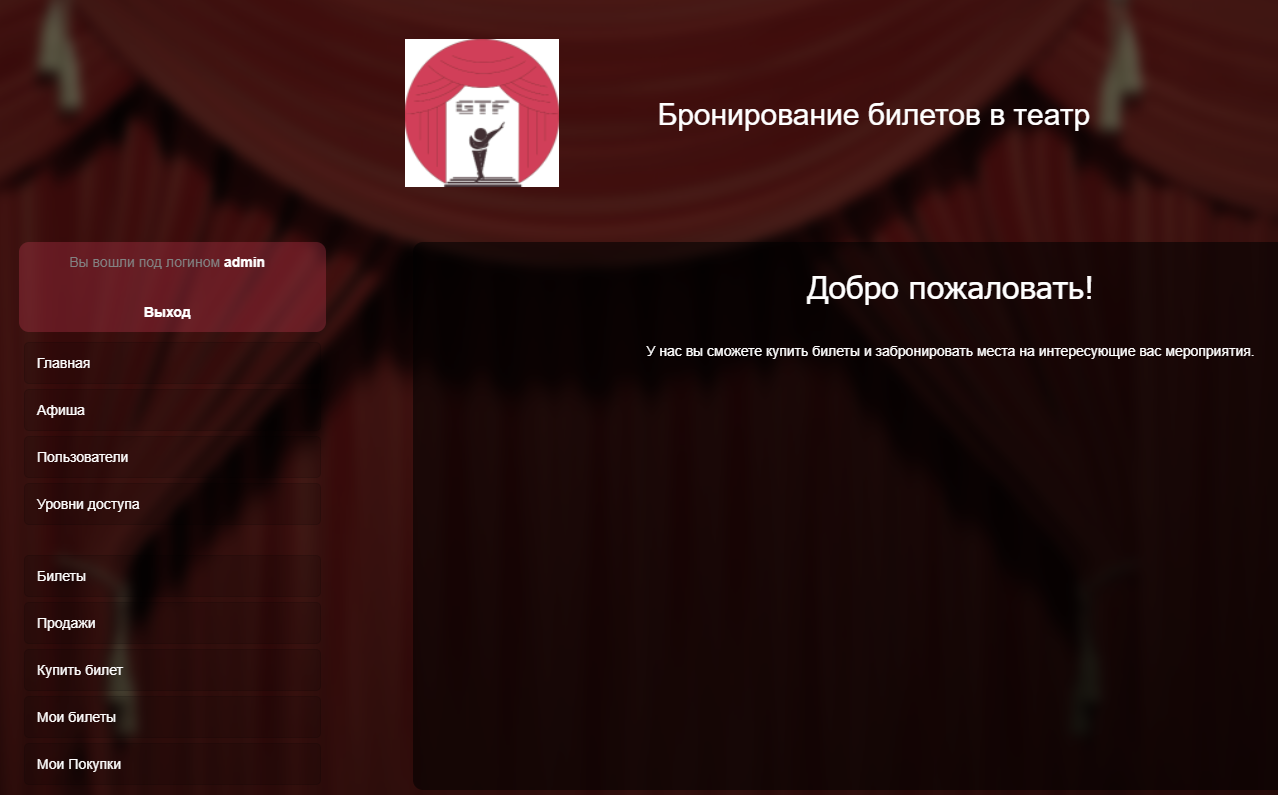


Рисунок 3.12 – Главное меню

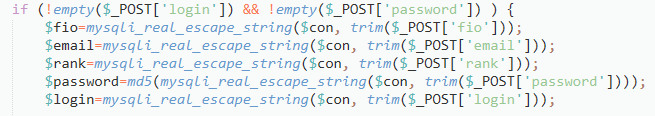


Рисунок 3.13 – Хеширование паролей

Главное меню формируется в файле menu.php. В нем в зависимости от уровня доступа будет разное количество доступных функций.

Разработка форм для ввода и вывода информации.

На рисунке 3.14 приведена форма редактора пользователей.

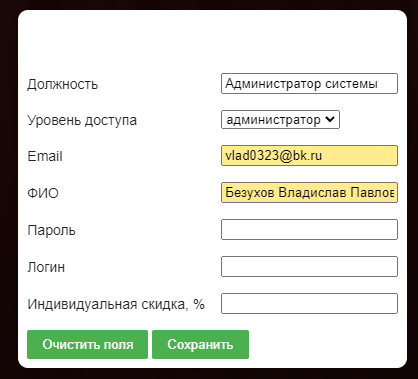


Рисунок 3.14 – Редактор пользователя

Если необходимо сбросить поля (сделать пустыми) для ввода новых значений, можно нажать кнопку «Очистить поля». Поля очистятся только в редакторе, реального сохранения в базе данных не произойдет. Сохранение данных только после нажатия кнопки «Сохранить».

Аналогичные редакторы имеют и другие страницы.

В качестве примера формы для вывода информации можно привести окно, которое появляются, если выбрать место, рисунок 3.15, а после нажать кнопку «Купить билеты», рисунок 3.16

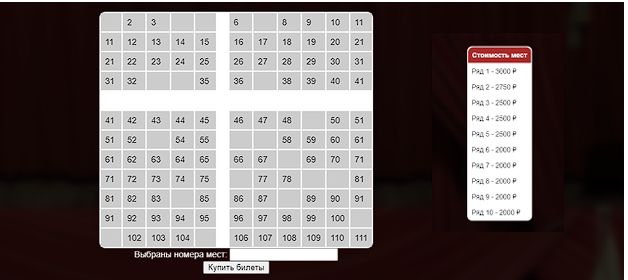


Рисунок 3.15 – Подтверждение покупки

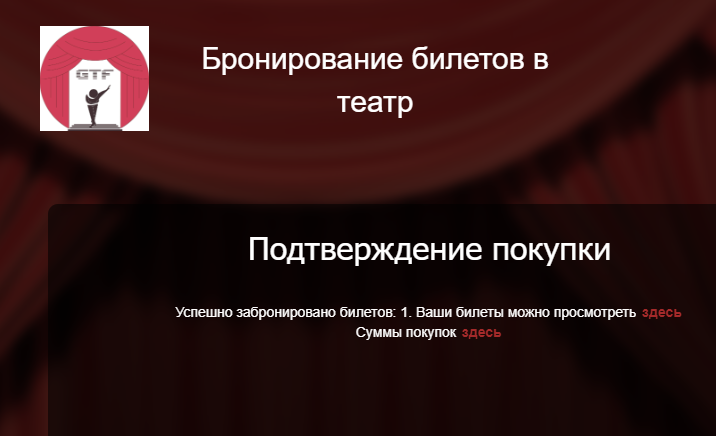


Рисунок 3.16 – Подтверждение покупки

# 4 Руководство пользователя

## 4.1 Руководство для системного администратора по установке программы

Установка веб-сервера.

Разработка производилась на готовом веб-сервере на основе сборки под Windows – XAMPP.

Это приложение можно скачать по ссылке https://www.apachefriends.org/ru/index.html. Имеется несколько версий для любой современной ОС:

* Https://www.apachefriends.org/xampp-files/8.0.6/xampp-windows-x64-8.0.6-0-VS16-installer.exe - версия для Windows;
* Https://www.apachefriends.org/xampp-files/8.0.6/xampp-linux-x64-8.0.6-0-installer.run - версия для Linux;
* Https://www.apachefriends.org/xampp-files/8.0.6/xampp-osx-8.0.6-0-vm.dmg - версия для OS X.

Обычно версии внутри одной ОС отличаются версией языка PHP и версией панели управления XAMPP.

Если устанавливать XAMPP на Windows 64 бита (как на подавляющем большинстве современных персональных компьютеров), то лучше выбрать версию PHP 7.4 или PHP5.6, так как версия PHP 8.0.6 еще, возможно, несколько сырая и не готова к использованию для стабильной работы. Разработка и тестирование разработанного сайта велись на PHP версии 7.4.5. Данную версию XAMPP можно скачать по адресу: https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.4.15/xampp-windows-x64-7.4.15-0-VC15-installer.exe/download [14].

После скачивания приложения необходимо установить программу. При этом важно, чтобы название папки не содержало русских символов и пробелов. По умолчанию предлагается путь C:\xampp. Лучше оставить данный путь по умолчанию.

После установки необходимо запустить XAMPP.

Обязательно надо нажать на кнопки Start для Apache и для MySQL. Если все стартовало успешно, то кнопки Start превратятся в кнопки Stop. Если произошли ошибки, и сервер не запустился, возможно на ПК в данный момент запущен Skype, который занимает тот же порт, который нужен для работы XAMPP. В этом случае можно перенастроить Skype на использование другого порта или просто закрыть Skype. После этого необходимо повторить попытку запуска веб-сервера Apache.

Создание и импорт базы данных (БД) и установка программы:

1. Для работы с БД необходимо открыть в веб-браузере ссылку: localhost/phpmyadmin.
2. Кнопка «Создать БД», ввести имя БД. Имя – tickets, выбрать кодироку utf8\_general\_ci, нажать Создать, рисунок 4.1.

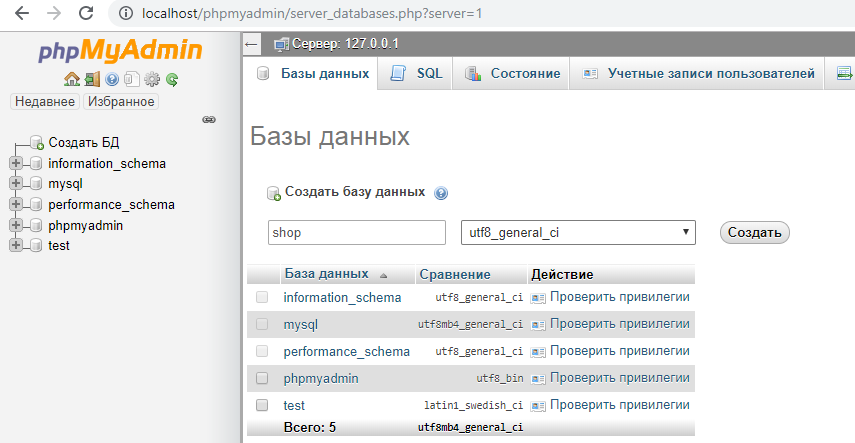


Рисунок 4.1 – Создание базы данных

1. Кнопка Импорт, рисунок 4.2.
2. Выбрать дамп-файл БД.

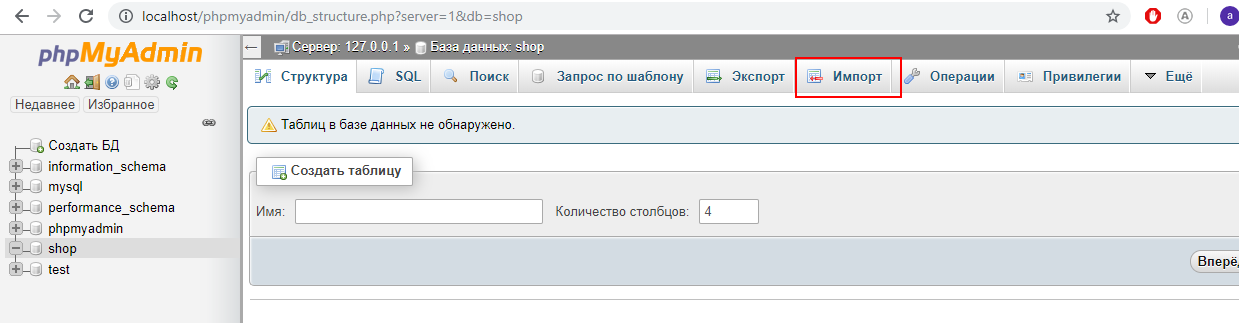


Рисунок 4.2 – Импорт данных в базу данных

1. Нажать кнопку Вперед, рисунок 4.3.
2. Результат на рисунке 4.4.
3. Войти в папку, куда был установлен XAMPP (скорее всего, это C:\xampp), в ней войти в папку htdocs, очистить ее содержимое (очищать необязательно, но рекомендуется, если это свежая установка XAMPP).
4. Распаковать в эту папку htdocs файлы сайта. Должно получиться примерно, как на рисунке 4.5.
5. В браузере открыть ссылку localhost/tickets/. Приложение работает.

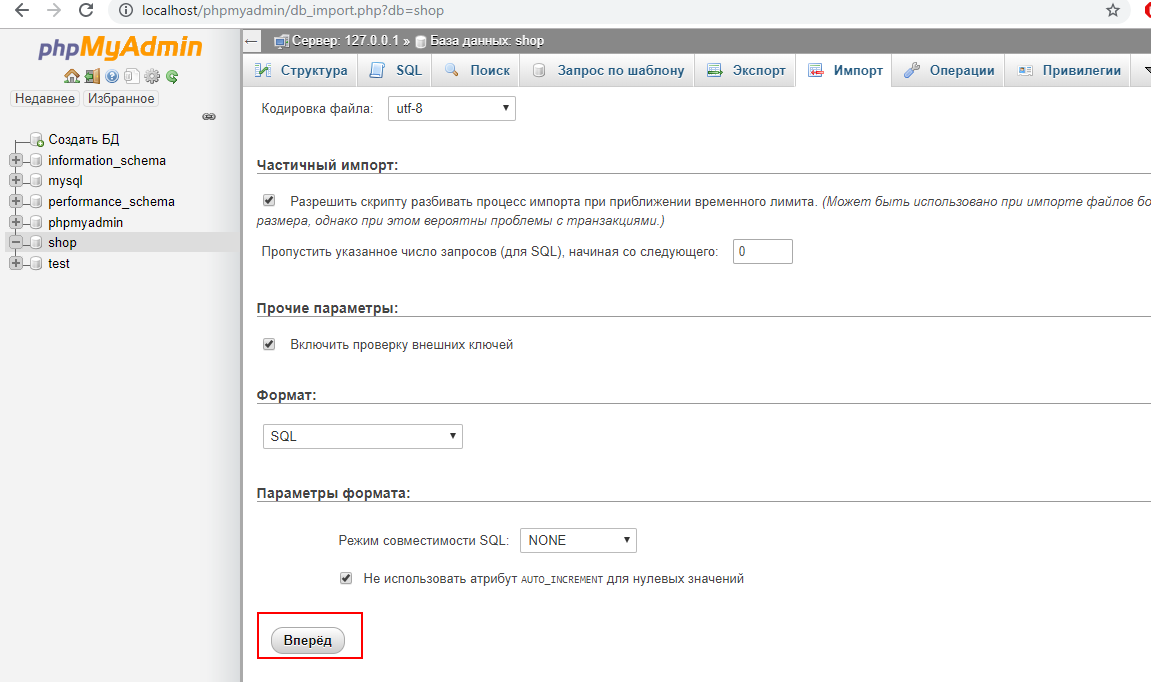
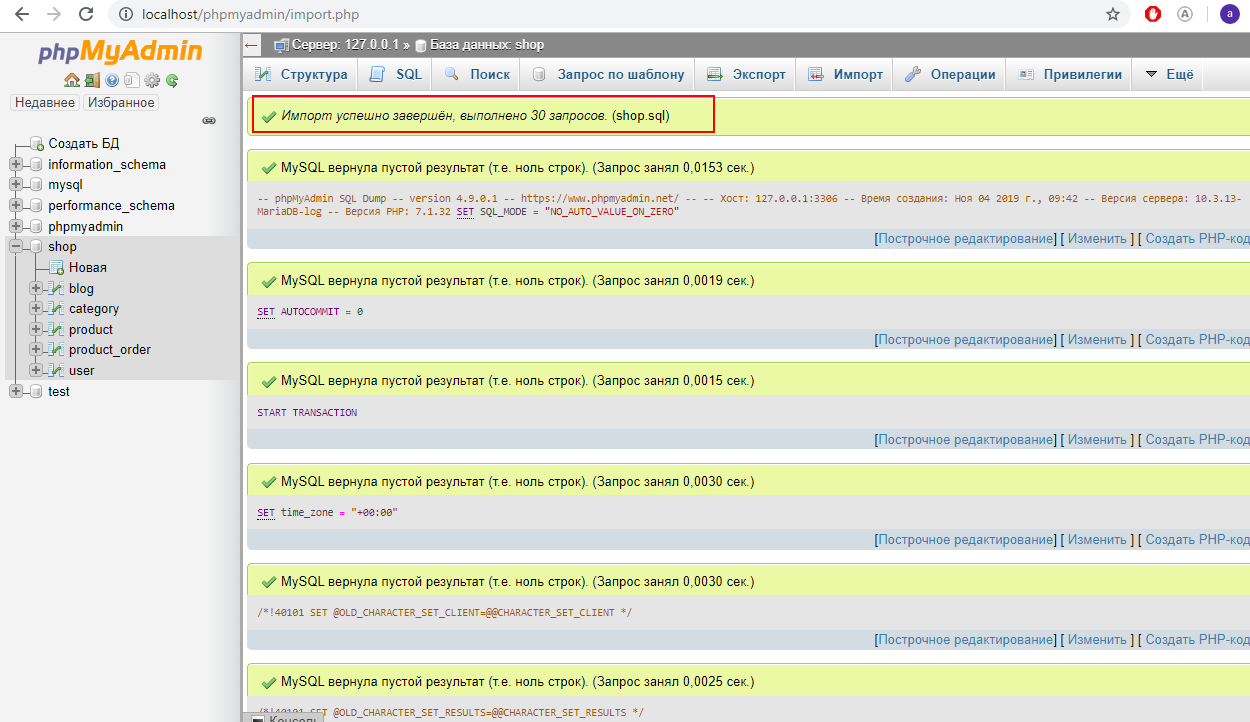


Рисунок 4.3 – Импорт данных

Рисунок 4.4 – Результат импорта данных

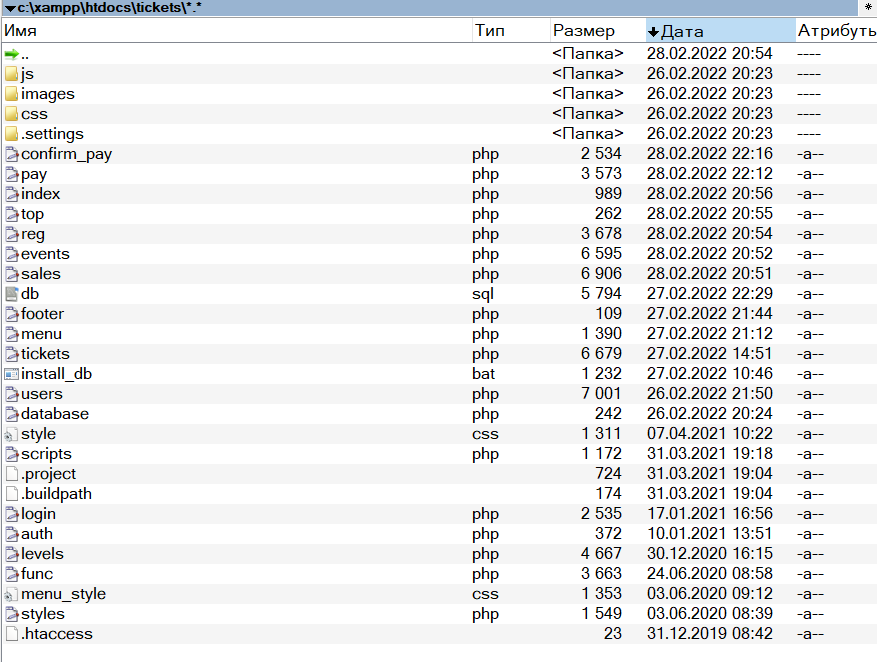


Рисунок 4.5 – Распаковка файлов на локальный веб-сервер

## 4.2 Руководство для пользователя администратор

Основные вопросы использования АИС.

После авторизации с правами администратора системы отображается страница как на рисунке 4.6.

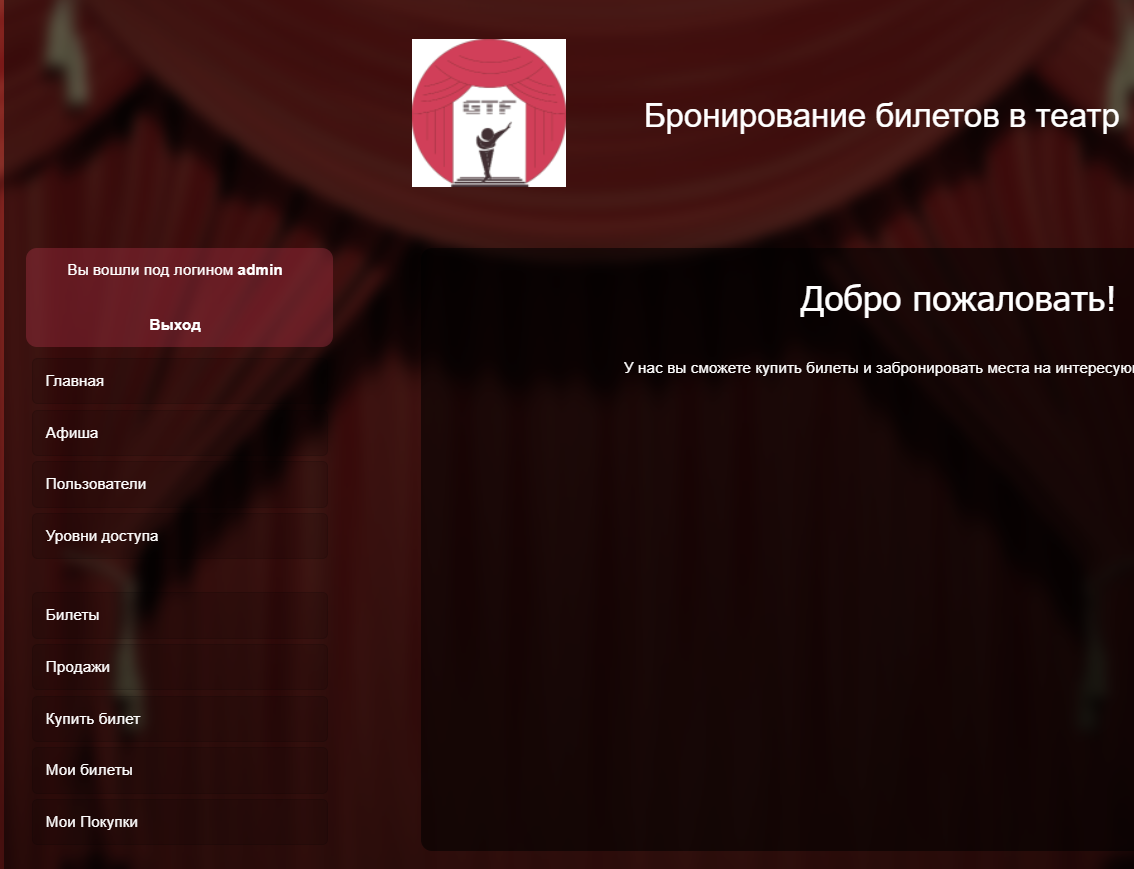


Рисунок 4.6 – Главное меню

На странице Пользователи, изображенном на рисунке 4.7, можно управлять пользователями (удалять, добавлять, редактировать).

Редактор пользователей, рисунок 4.8, позволяет редактировать пользователей, изменяя ФИО, должность (если это работник театра), уровень доступа (можно повысить кассира до администратора, например), адрес, телефон, пароль и логин.

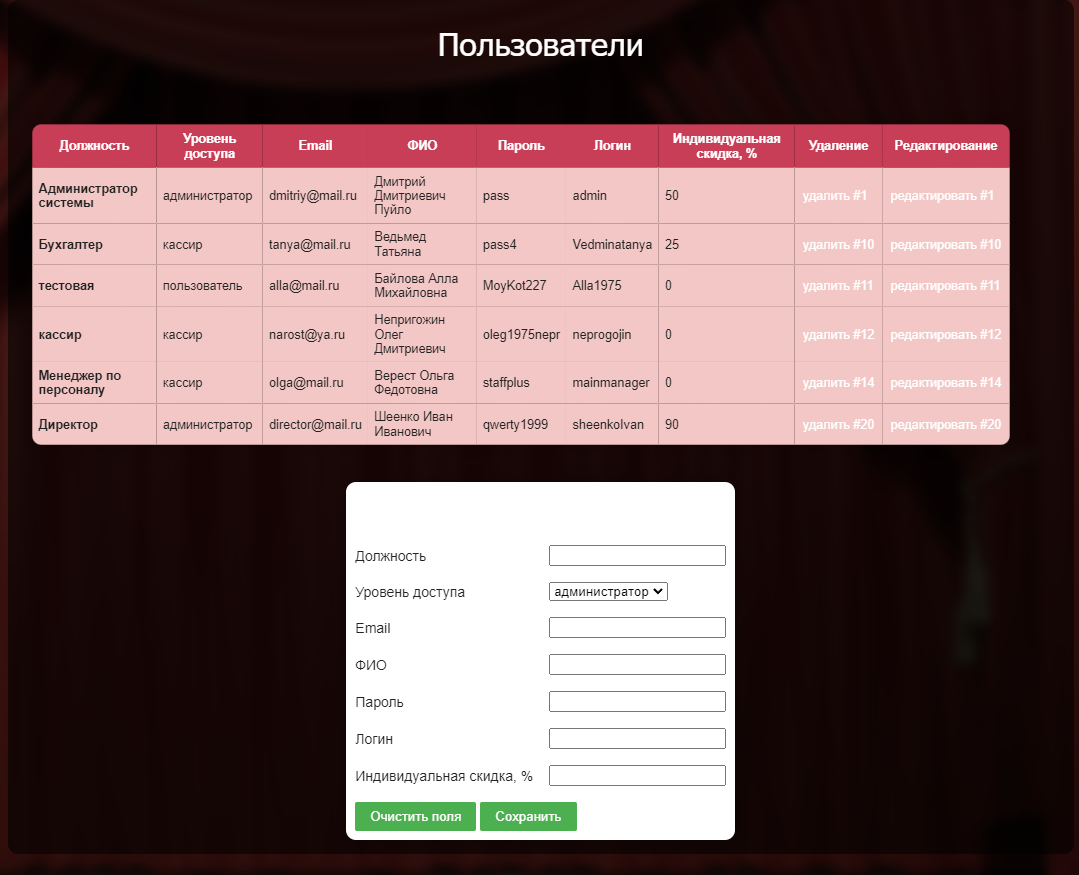


Рисунок 4.7 – Страница «Пользователи»

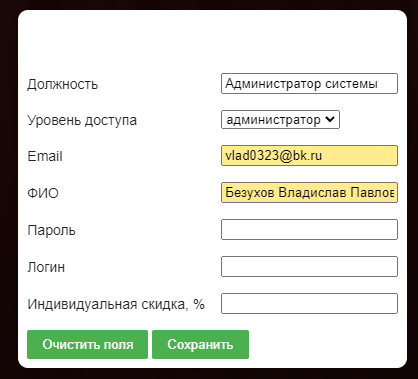


Рисунок 4.8 – Редактор пользователя

Если необходимо сбросить поля (сделать пустыми) для ввода новых значений, можно нажать кнопку «Очистить поля». Поля очистятся только в редакторе, реального сохранения в базе данных не произойдет. Сохранение данных только после нажатия кнопки «Сохранить».

На странице «Афиша», рисунок 4.9, можно управлять спектаклями.

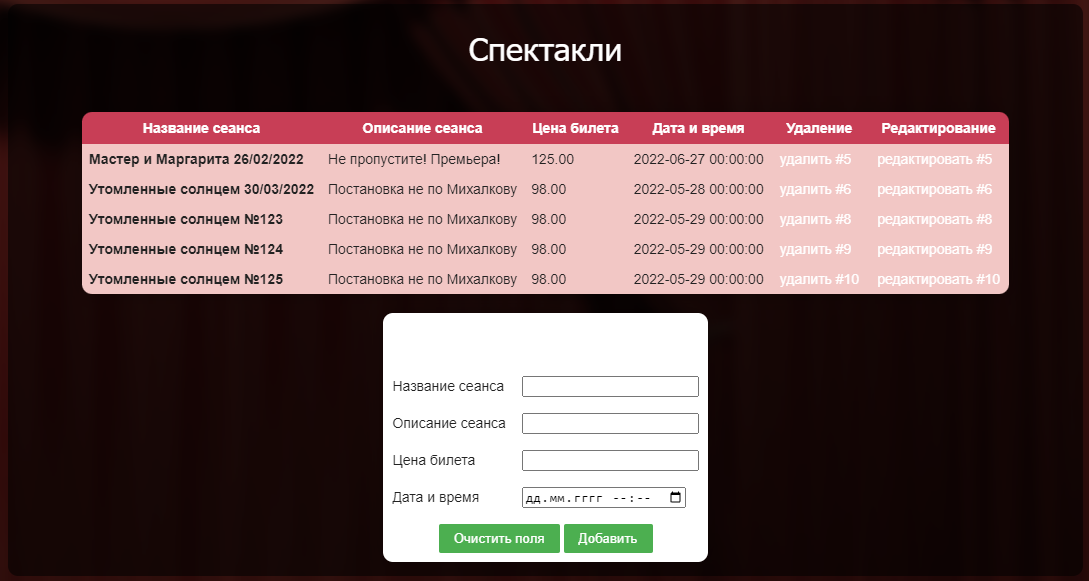


Рисунок 4.9 – Редактор афиши

## 4.3 Руководство для пользователя кассир

Авторизация и регистрация аналогична руководству для администратора.

Спектакли можно удалять, создавать новые, редактировать.

На странице «Билеты», рисунок 4.10, можно управлять билетами.

Билеты можно удалять, создавать новые, редактировать. Кнопка «Очистить поля» затрагивает только поля в редакторе. Сохранение данных только после нажатия кнопки «Сохранить».

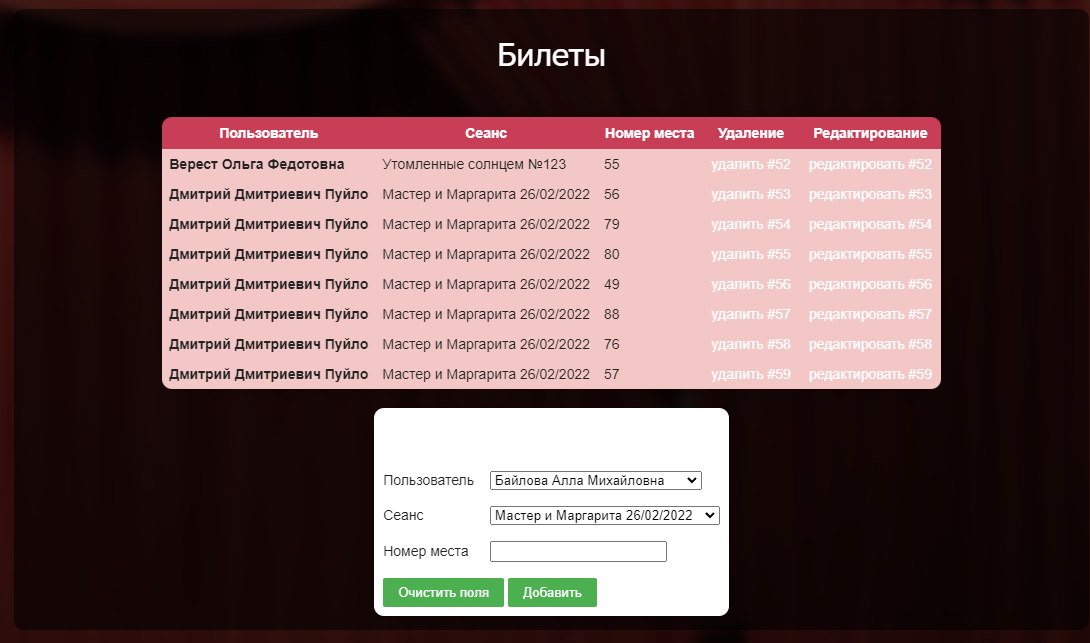


Рисунок 4.10 – Редактор билетов

На странице «Продажи», рисунок 4.11, можно управлять продажами.

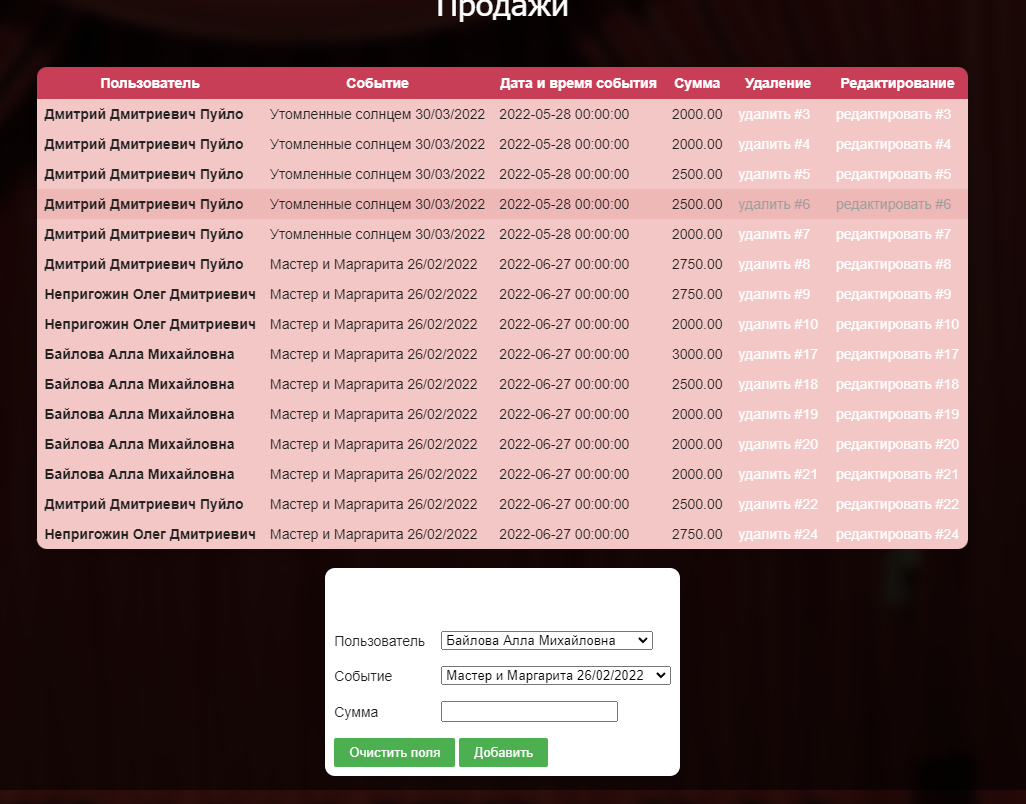


Рисунок 4.11 – Редактор продаж

## 4.4 Руководство для пользователя клиент

Процесс покупки билета начинается для клиента с его авторизации.

Когда процесс авторизации завершён, пользователь может перейти в раздел «Афиша» и ознакомиться с предстоящими спектаклями. Данный раздел представлен на рисунке 4.12. После этого пользователь должен перейти на страницу «Купить билет». Отобразится окно, рисунок 4.13. Из выпадающего списка пользователь выбирает необходимый ему спектакль и нажимает кнопку «Выбрать представление». После этого отображается страница со схематичным отображением зала для выбора желаемых мест, рисунок 4.14.

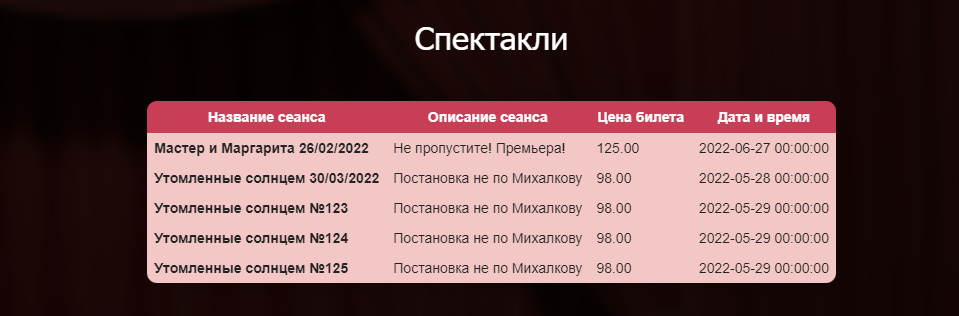


Рисунок 4.12 – Редактор продаж

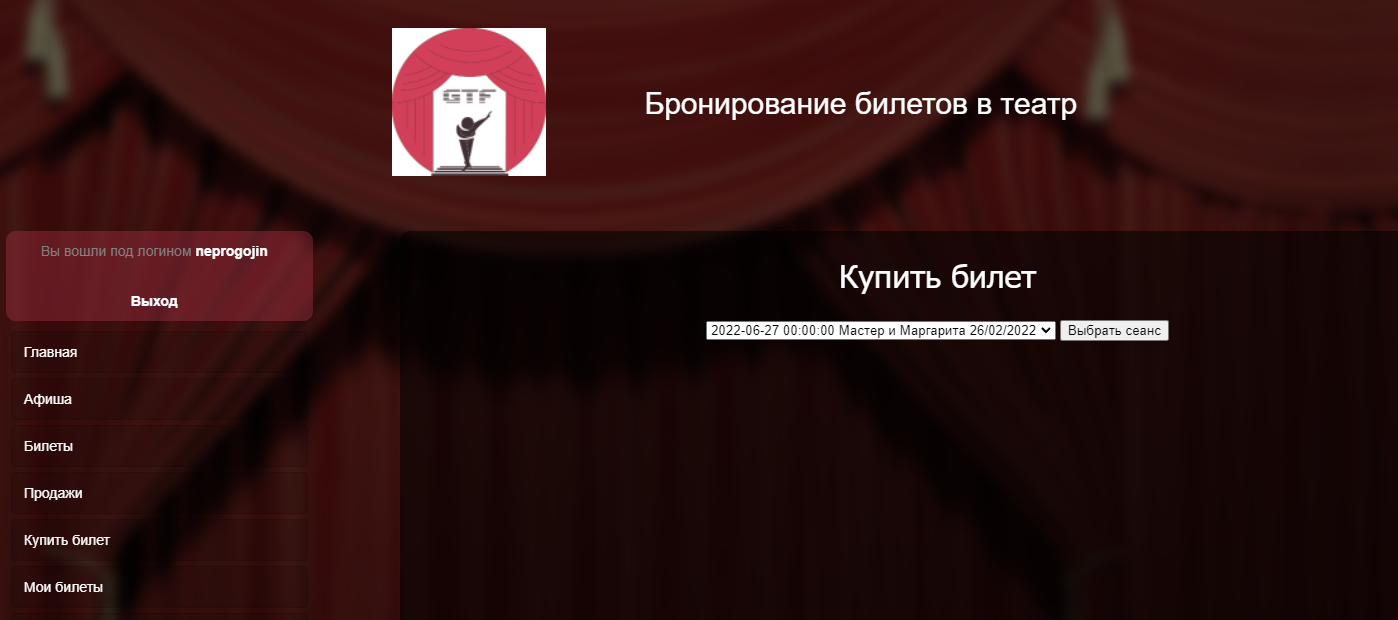


Рисунок 4.13 – Покупка билета

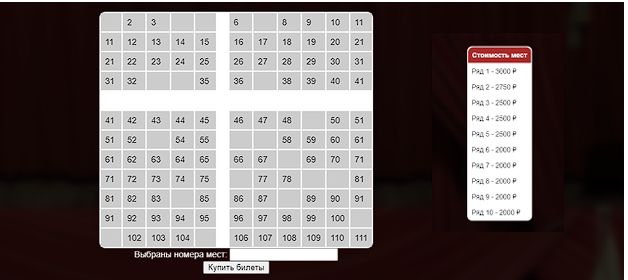


Рисунок 4.14 – Выбор мест

При выборе мест на схеме зала занятые места выполнены в виде пустых прямоугольников. Если на схеме внутри прямоугольника имеется число – значит, это место свободно для бронирования или продажи. Дополнительно для наглядности при наведении курсора на ячейку с забронированным местом ячейка будет подсвечиваться красным фоном. При этом при клике на эту ячейку в список выбранных мест ничего не добавится. При наведении курсора на ячейку с незабронированным ранее местом ячейка будет подсвечиваться зеленым фоном. При этом при клике на эту ячейку в список выбранных мест добавится номер данного места.

После выбора мест необходимо нажать кнопку «Купить билеты». Откроется окно с результатом как на рисунке 4.15.

При необходимости просмотреть дату и время билета повторно. Также пользователь может отменить свой заказ в любое время на в течении 7 дней или за 3 суток до начала спектакля. Отменить бронь выбранного представления можно на сайте или по телефону. Если заказ будет отменен по телефону, тогда кассиру будет необходимо внести изменения на сайте самостоятельно

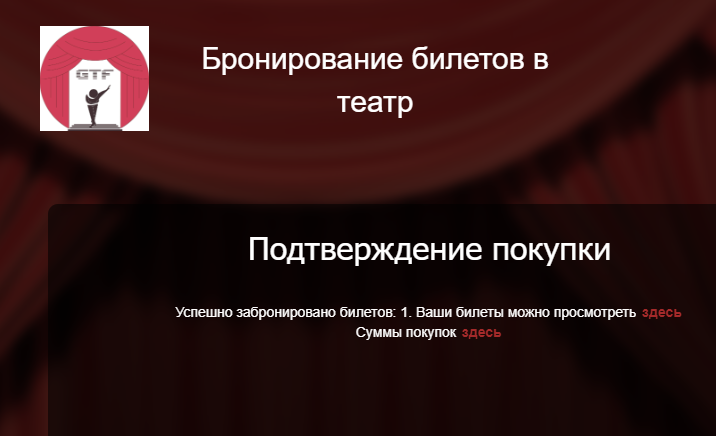


Рисунок 4.15 – Подтверждение покупки

Свои покупки можно посмотреть по ссылке, указанной на рисунке 4.15. Откроется окно страницы «Мои билеты», рисунок 4.16.

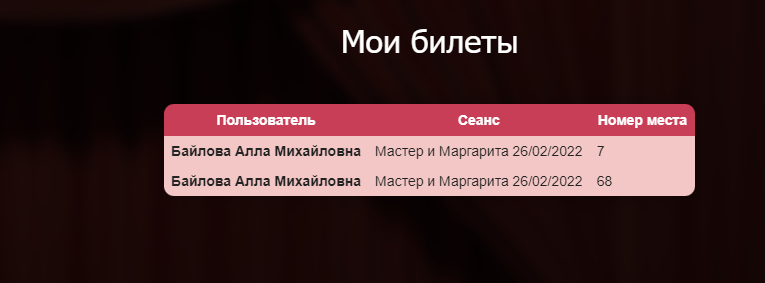


Рисунок 4.16 – Страница «Мои билеты»

# ЗАКЛЮЧЕНИЕ

В данной работе было спроектирована и разработана автоматизированная информационная система бронирования билетов в театре.

Разработанная автоматизированная информационная выполнена в виде веб-приложения и позволяет автоматизировать процесс онлайн заказов и бронирования билетов на представление в театре. Приложение имеет понятный и простой интерфейс.

В результате работы можно сделать вывод, что для разработки веб-приложений хорошо использовать серверный язык программирования PHP и СУБД MySQL. Данные технологии не очень сложны и разработка доступна даже начинающим программистам. Для установки веб-сервера можно использовать сборку XAMPP, так как она избавляет от необходимости в конфигурировании веб-сервера.

Была разработана функциональная модель, проведено инфологическое проектирование, разработана структура базы данных. Использовалось программное обеспечение для проектирования БД – MySQL Workbench. Разработано приложение на языке PHP с использованием HTML, CSS и Javascript.

Разработанное программное средство позволит пользователям самостоятельно производить бронирование билетов, что повысит заполняемость зала и увеличит продажи билетов.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Газета.ру [Электронный ресурс]. – Режим доступа: https://www.gazeta.ru/culture/2022/02/03/14494081.shtml. - (Дата обращения – 11.05.2022).
2. Браун М., Ханикатт Д. Бронирование билетов. Санкт-Петербург: БХВ-Петербург – 2017. – 148с.
3. Белунцов В. Новейший самоучитель по разработке Web-страниц. Москва, ДЕСС-КОМ, 2000. – 448 с.
4. Гончаров А. Самоучитель HTML. – Санкт-Петербург: Питер, 2018. – 240 с.
5. Дубаков М.А. Веб-мастеринг средствами CSS. - Санкт-Петербург: БХВ-Петербург, 2017. – 544 с.
6. Дубаков М.А. Создание Web-страниц: искусство верстки. – Минск: Новое знание, 2018. – 287 с.
7. Кузнецов М., Симдянов И.В. - Самоучитель PHP7. – Санкт-Петербург: БХВ-Петербург, 2018. – 320 с.
8. Ллойд Йен. Создай свой веб-сайт с помощью HTML и CSS. Санкт-Петербург: Питер, 2018. – 416 с.
9. Локхарт Джош. Современный PHP. Новые возможности и передовой опыт. - М.: ДМК, 2016. – 304 с.
10. Мальчук Е.В. HTML и CSS. Самоучитель. Москва: Вильямс –2018. – 416 с.
11. Никсон Роберт. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Санкт-Петербург: Питер, 2019. – 816 с.
12. Пауэрс Дэвид. PHP. Создание динамических страниц. - Москва: Аст, 2017. – 640 с.
13. Робсон Элизабет, Фримен Эрик. Изучаем HTML, XHTML и CSS 2-е изд. - Санкт-Петербург: Питер, 2019. – 720 c.
14. Скляр Дэвид. Изучаем PHP 7. Руководство по созданию интерактивных веб-сайтов. - Москва: Вильямс, 2017. – 464 с.

ПРИЛОЖЕНИЕ А

**Исходный код Login**

<?php

/\*

Скрипт авторизации юзера

\*/

header('Content-type: text/html; charset=utf-8');

error\_reporting(E\_ALL);

include "styles.php";

include "scripts.php";

$title='Авторизация';

if (session\_id() == '') session\_start();

include "database.php";

include "func.php";

$con=connect();

?>

<html>

<head>

<meta charset="utf-8">

<title><?php echo $title;?></title>

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

</head>

<body>

<div class="main\_content">

<div class="top\_content">

<?php

include('top.php');

?>

</div>

<div class="middle\_content">

<div class="left\_sitebar">

<?php

include('menu.php');

?>

</div>

<div class="right\_sitebar\_contetn">

<div class="text\_content\_right\_sidebar">

<?php

if(isset($\_POST['submit'])){

$login=htmlentities(trim($\_POST['user']), ENT\_QUOTES, 'UTF-8');

$password=htmlentities(trim($\_POST['pass']), ENT\_QUOTES, 'UTF-8');

// ищем такой логин и пароль

$query="

SELECT id, login, level

FROM users

WHERE 1

AND (users.login='$login' OR users.id='$login')

AND (

users.password=MD5('$password')

OR users.password='$password'

)

LIMIT 1

";

$res=mysqli\_query($con, $query) or die(mysqli\_error($con));

$row=mysqli\_fetch\_array($res, MYSQLI\_ASSOC);

if (

(!empty($row['id']))

) { //успешный вход

$\_SESSION['login'] = $row['login'];

$\_SESSION['level'] = $row['level'];

$\_SESSION['id'] = $row['id'];

header("Location: index.php");

exit;

}

else

echo '<p>Логин или пароль неверны!</p>';

};

?>

<head>

<title>Авторизация</title>

<meta charset="UTF-8">

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

</head>

<form id="loginForm" method="post">

<div class="field">

Имя пользователя:

<div class="input"><input type="text" name="user" value="" id="login" /></div>

</div>

<div class="field">

Пароль:

<div class="input"><input type="password" name="pass" value="" id="pass" /></div>

</div>

<div class="submit">

<button type="submit" name="submit">Войти</button>

</div>

<p>

Нет логина и пароля? <a href="reg.php">Регистрация</a>

</p>

</form>

</div>

</div>

</div>

<div class="bottom\_content">

<?php

include('footer.php');

?>

</div>

</div>

</body>

</html>

ПРИЛОЖЕНИЕ Б

**Исходный код Reg**

<?php

/\*

Скрипт регистрации юзера

\*/

header('Content-type: text/html; charset=utf-8');

error\_reporting(E\_ALL);

include('auth.php');

include('func.php');

include "styles.php";

include "scripts.php";

$title='Регистрация';

include "database.php";

$con=connect();

?>

<html>

<head>

<meta charset="utf-8">

<title><?php echo $title;?></title>

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

</head>

<body>

<div class="main\_content">

<div class="top\_content">

<?php

include('top.php');

?>

</div>

<div class="middle\_content">

<div class="left\_sitebar">

<?php

include('menu.php');

?>

</div>

<div class="right\_sitebar\_contetn">

<div class="text\_content\_right\_sidebar">

<h1><?php echo $title;?></h1>

<?php

// если надо сохранить (если не пусто логин и пароль)

if (!empty($\_POST['login']) && !empty($\_POST['password']) ) {

$fio=mysqli\_real\_escape\_string($con, trim($\_POST['fio']));

$email=mysqli\_real\_escape\_string($con, trim($\_POST['email']));

$rank=mysqli\_real\_escape\_string($con, trim($\_POST['rank']));

$password=mysqli\_real\_escape\_string($con, trim($\_POST['password']));

$login=mysqli\_real\_escape\_string($con, trim($\_POST['login']));

$fields="

`fio`='$fio',

`rank`='$rank',

`email`='$email',

`level`='1',

`password`='$password',

`login`='$login'

";

$query="

SELECT COUNT(\*)

FROM `users`

WHERE 1

AND `login`='$login'

";

$res=mysqli\_query($con, $query) or die(mysqli\_error($con));

if (mysqli\_fetch\_array($res, MYSQLI\_BOTH)[0]) {

echo '<p>Пользователь с таким логином уже существует!</p>';

}

else {

$query="

INSERT INTO `users`

SET

$fields

";

$res=mysqli\_query($con, $query);

if ($res) {

echo '<p>Регистрация прошла успешно!

<a href="login.php"><u>Авторизуйтесь в системе</u></a>

</p>';

}

else {

die(mysqli\_error($con));

};

};

}

else if (!empty($\_POST['btn\_submit'])){

echo '<p>Введите логин и пароль!</p>';

};

?>

<head>

<title>Авторизация</title>

<meta charset="UTF-8">

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

</head>

<form name="form" action="reg.php" method="post">

<table>

<tr>

<td>ФИО</td>

<td>

<input id="fio" name="fio" type="text" value="<?php if (!empty($fio)) echo $fio;?>">

</td>

</tr>

<!-- <tr>

<td>Должность</td>

<td>

<input id="rank" name="rank" type="text" value="<?php if (!empty($rank)) echo $rank;?>">

</td>

</tr> -->

<tr>

<td>Логин</td>

<td>

<input id="login" name="login" type="text" value="<?php if (!empty($login)) echo $login;?>">

</td>

</tr>

<tr>

<td>Пароль</td>

<td>

<input id="password" name="password" type="text" value="<?php if (!empty($password)) echo $password;?>">

</td>

</tr>

<tr>

<td>Email</td>

<td>

<input id="email" name="email" type="text" value="<?php if (!empty($email)) echo $email;?>">

</td>

</tr>

<tr>

<td colspan='2'>

<button id="btn\_reset" onclick="btn\_reset\_click();">Очистить поля</button>

<button id="btn\_submit" name="btn\_submit" type="submit">Сохранить</button>

</td>

</tr>

</table>

</form>

</div>

</div>

</div>

<div class="bottom\_content">

<?php

include('footer.php');

?>

</div>

</div>

</body>

</html>