

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЮЖНО-РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (НПИ)
имени М.И. ПЛАТОВА»

ФАКУЛЬТЕТ Информационных технологий и управления

КАФЕДРА Информационные и измерительные системы и технологии

НАПРАВЛЕНИЕ 09.03.03 – “Прикладная информатика”

КУРСОВАЯ РАБОТА

На тему: Web - сайт бронирования авиабилетов

По дисциплине: Разработка приложений для интернет

Выполнил студент 3 курса, группы 2б Решетилов Артем Николаевич

Фамилия, имя, отчество

Принял доцент, к.т.н Панфилов Александр Николаевич

Должность, звание

Фамилия, имя, отчество

« » Г.

Подпись

Новочеркасск 20 г.

Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего образования
«ЮЖНО-РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
(НОВОЧЕРКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ)» имени М.И.ПЛАТОВА

ФАКУЛЬТЕТ Информационных технологий и управления
КАФЕДРА Информационные и измерительные системы и технологии
НАПРАВЛЕНИЕ 09.03.03 - "Прикладная информатика"

УТВЕРЖДАЮ:
Заведующий кафедрой ИИСТ
_____ Горбатенко Н.И.
(подпись)
«__» _____ 2020 г.

ЗАДАНИЕ

на курсовую работу по дисциплине «РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ ИНТЕРНЕТ»

Студенту РЕШЕТИЛОВ АРТЕМ НИКОЛАЕВИЧ
(Фамилия, имя, отчество)

1. Тема курсовой работы: Web - сайт спортивной электронной газеты

Выполнение курсовой работы включает следующие этапы:

Этап первый. Проектирование сайта:

1. Сформулировать целевую аудиторию сайта, дать описание, определить ключевые слова тематического ядра. Оценить объем сайта (сколько страниц). Отразить в отчете.
2. Разработать структуру сайта. Отобразить в виде схемы в отчете.
3. Разработать систему связей между страницами сайта. Отобразить в виде схемы в отчете.
4. Разработать структуру страницы (отразить в отчете).
5. Определить используемую технологию (отразить в отчете).

Этап второй. Создание базовой структуры:

1. Создать структуру сайта в корневом каталоге.
2. Разработать и создать базовый шаблон страницы с учетом требований спецификации языка HTML.
3. Разработать систему навигации.
4. Доработать шаблон с учетом панелей навигации.
5. Создать индексные страницы сайта и разделов.
6. Создать дополнительные страницы.

Этап третий. Создание базового наполнения:

1. Создать базовое наполнение индексных страниц.
2. Создать базовое наполнение разделов.
3. Создать расширенное наполнение.

Этап четвертый. Подготовка документации:

1. Подготовить краткую информацию по сайту: содержание, ключевые слова, используемая технология и др.
2. Исчислить показатели эффективности сайта
3. Подготовить отчет по курсовой работе

Срок сдачи работы
Руководитель Панфилов А.Н.

« 25 » декабря 2020 г.

(Подпись)

Задание принял к исполнению
« 04 » сентября 2020 г.

(Подпись)

Дата выдачи задания

« 04 » сентября 2020

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ПОСТАНОВКА ЗАДАЧИ	4
1. Описание предметной области Интернет- приложения	5
1.1. Описание информационных и бизнес процессов в рамках предметной области Интернет- приложения.....	5
1.2 Построение модели Интернет- приложения	5
1.3 Техническое задание на разработку Web- сайта Интернет – проекта.....	6
2. Проектирование Web- сайта	7
2.1 Функциональная структура и модули сайта.....	7
2.2 Дизайн сайта	7
3. Информационное обеспечение сайта	10
3.1 Проектирование БД сайта	10
3.2. Описание пользовательских запросов	14
4. Программная реализация сайта	15
4.1 Описание программной реализации сайта	15
4.2 Описание пользовательского интерфейса	17
5. Оценка эффективности и тестирование сайта	25
ЗАКЛЮЧЕНИЕ	27
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	28
ПРИЛОЖЕНИЯ	29
Приложение А	29
Приложение В.....	32
Листинг css файла	32
Приложение С.....	37
Листинг php файлов	37

ВВЕДЕНИЕ

Интернет стал наиболее эффективным средством рекламы и продвижения и является одним из важных элементов современной цивилизации. Быстро растет количество изданий, посвященных Сети, что предвещает широкое ее распространение даже в далеких от техники областях. Интернет превращается из большой игрушки для интеллектуалов в полноценный источник разнообразной полезной информации для любой категории пользователей. Многие заходят в Интернет читать, развлекаться, общаться, узнавать что-то новое для себя. Поэтому создание сайтов стало таким популярным.

Сайты позволяют хранить, передавать, продавать различные типы информации, не отходя от экрана компьютера. А главное, сайты позволяют подать эту информацию в ярком, красочном виде, снабдить её графикой, звуком, анимацией, видеороликами, ссылками и при правильном спланировании работы, продемонстрировать эту информацию тысячам других пользователей интернета.

Для дальних путешествий идеально подходит авиатранспорт. Он имеет множество преимуществ: скорость, комфорт, возможность добраться в любые крупные города мира быстро и безопасно. Однако есть у него и существенный недостаток - это высокая стоимость авиабилетов. Те люди, кто часто пользуются самолётами, знают, что можно существенно сэкономить на передвижении.

Для этого используются специальные системы бронирования авиабилетов. Помимо возможности выбора направления, даты, стоимости маршрута, они также дают шанс поучаствовать в специальных акциях авиакомпаний, отдать приоритет самым дешёвым альтернативам. Именно с помощью интернета покупка недорогих билетов становится простой и быстрой.

ПОСТАНОВКА ЗАДАЧИ

Описание реализуемых функций:

1. Поиск доступных авиабилетов по заданным параметрам;
2. Возможность регистрации на сайте для бронирования авиабилетов;
3. Просмотр Ваших уже забронированных авиабилетов.

Требования к программной среде и средствам разработки:

- 1) формат представления Web-документов – HTML вер.4.0 и выше;
- 2) язык разработки серверных приложений – PHP вер. 5.0 и выше;
- 3) операционная система Windows 7 и выше;
- 4) Web-сервер – Open-Server;
- 5) Сервер БД –MySQL.

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ИНТЕРНЕТ-ПРИЛОЖЕНИЯ

1.1. ОПИСАНИЕ ИНФОРМАЦИОННЫХ И БИЗНЕС ПРОЦЕССОВ В РАМКАХ ПРЕДМЕТНОЙ ОБЛАСТИ ИНТЕРНЕТ- ПРИЛОЖЕНИЯ

Сайт представляет собой сервис по бронированию авиабилетов, с помощью которого пользователи получают доступ к информации о рейсах, о городах вылета, о городах прибытия, о цене авиабилетов, о количестве авиабилетов. Зарегистрированные пользователи получают доступ к поиску и бронированию этих авиабилетов, к просмотру уже забронированных авиабилетов. В связи с перечисленным выше можно выделить следующие бизнес-процессы:

- 1) Возможность регистрации на сайте
- 2) Поиск и бронирование доступных авиабилетов
- 3) Просмотр уже заказанных авиабилетов

Целевая аудитория сайта - лица от 18 лет и старше.

1.2 ПОСТРОЕНИЕ МОДЕЛИ ИНТЕРНЕТ- ПРИЛОЖЕНИЯ

Спроектируем структуру базы данных:

Название таблицы	Параметры таблицы
City	Id, Name
Users	Id, User_email, Password
Orders	Id, User_id, Ticket_id, Count
Tickets	Id, City_from_id, City_to_id, Date, Class_id, Price, Count
Class	Id, Name

Таблица 1.2.1 - Структура базы данных

1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ WEB- САЙТА ИНТЕРНЕТ – ПРОЕКТА.

Техническое задание представлено в приложении А.

2. ПРОЕКТИРОВАНИЕ WEB- САЙТА

В процессе проектирования Web-сайта, необходимо выполнить следующие задачи:

1. построить функциональную структуру;
2. описать модули Web-сайта;
3. описать дизайн Web-сайта.

2.1 ФУНКЦИОНАЛЬНАЯ СТРУКТУРА И МОДУЛИ САЙТА

К основным модулям системы относятся:

1. Поиск и бронирование авиабилетов
2. Просмотр уже забронированных авиабилетов
3. Вход/Регистрация

Функциональная модель сайта представлена на рисунке 2.1

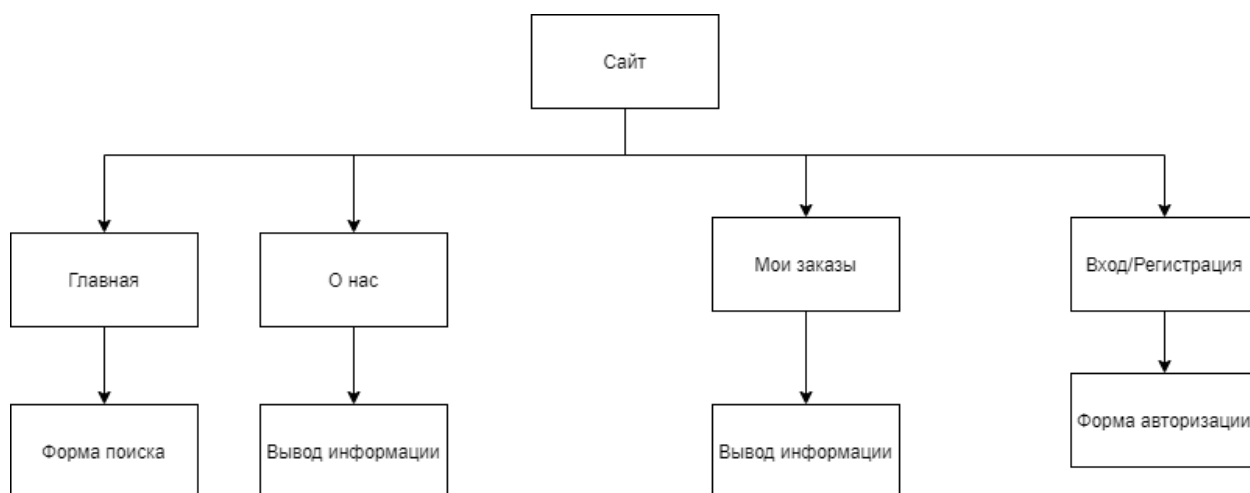


Рисунок 2.1 - Функциональная модель сайта

2.2 ДИЗАЙН САЙТА

Структура главной страницы Web-сайта представлена на рисунке 2.2. Web- сайт бронирования авиабилетов должен содержать в своей структуре размещения блоков следующие элементы:

- 1) блок шапки сайта, в котором имеется блок навигации, с помощью которого возможен переход на большую часть страниц.
- 2) блок содержания, в котором отображается блок с формой;
- 3) блок подвал, который отображает контактную информацию.

Пример главной страницы представлен на рисунке 2.3.

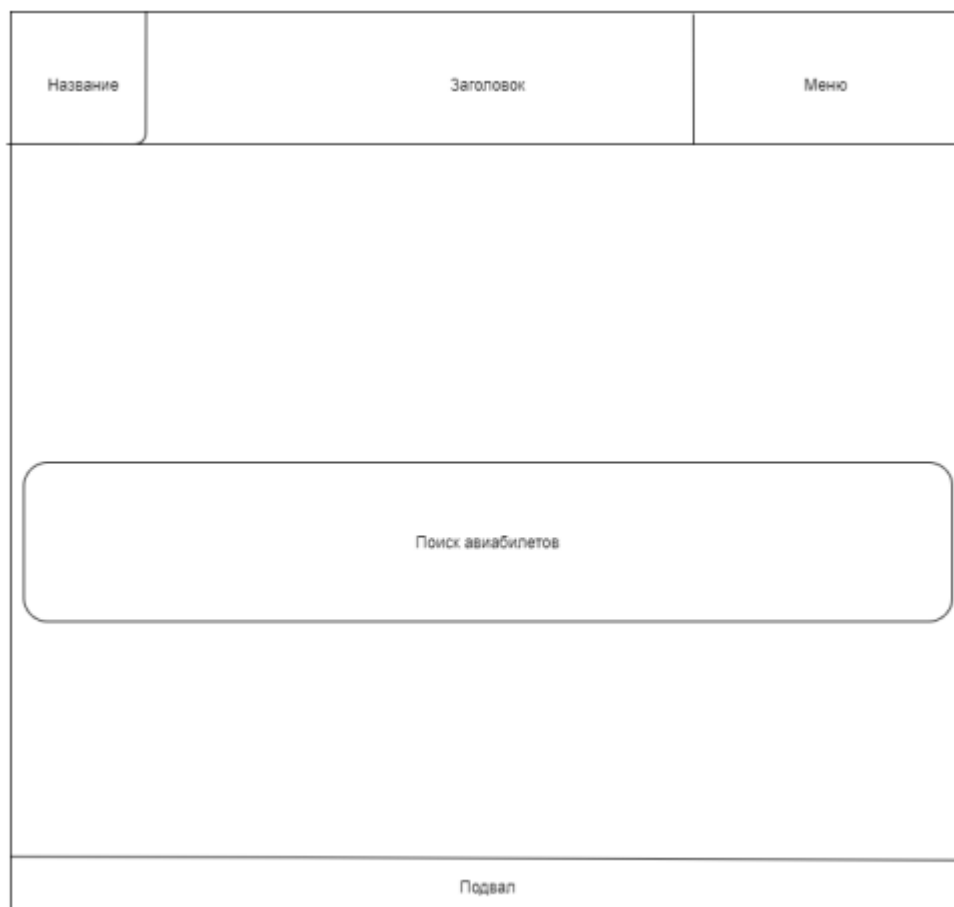


Рисунок 2.2 - Структура главной страницы

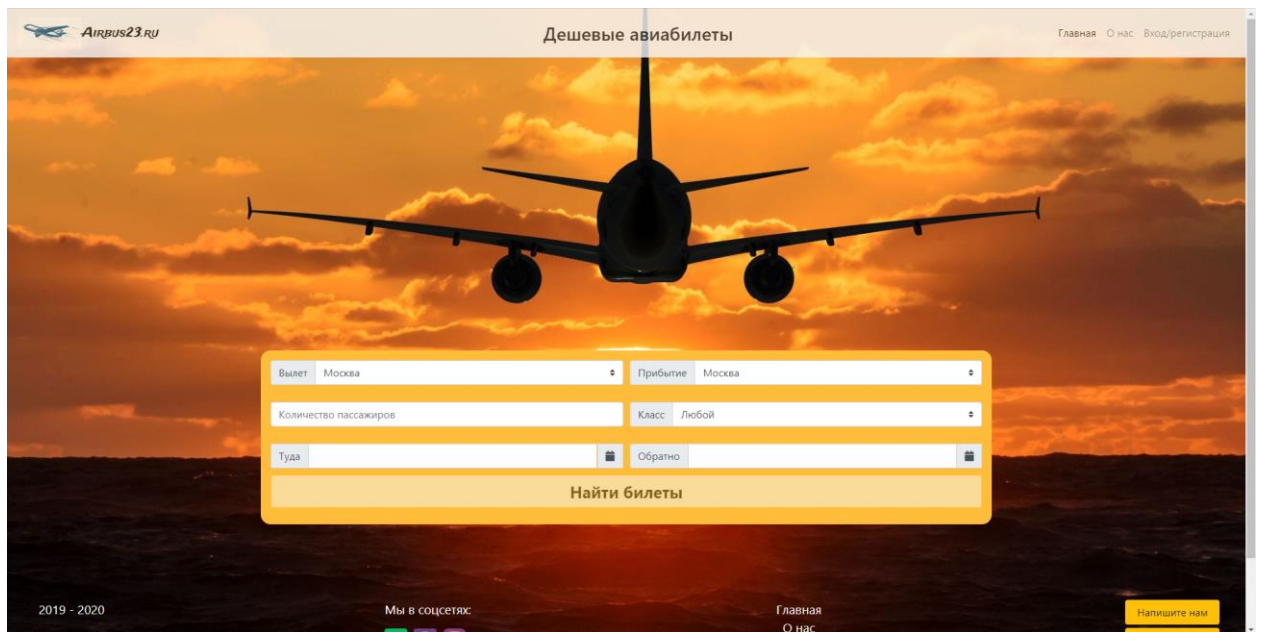


Рисунок 2.3 - Пример главной страницы

3. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ САЙТА

3.1 ПРОЕКТИРОВАНИЕ БД САЙТА

Схема базы данных была разработана в виде диаграммы сущность-связь в нотации Чена (Рисунок 3.1)

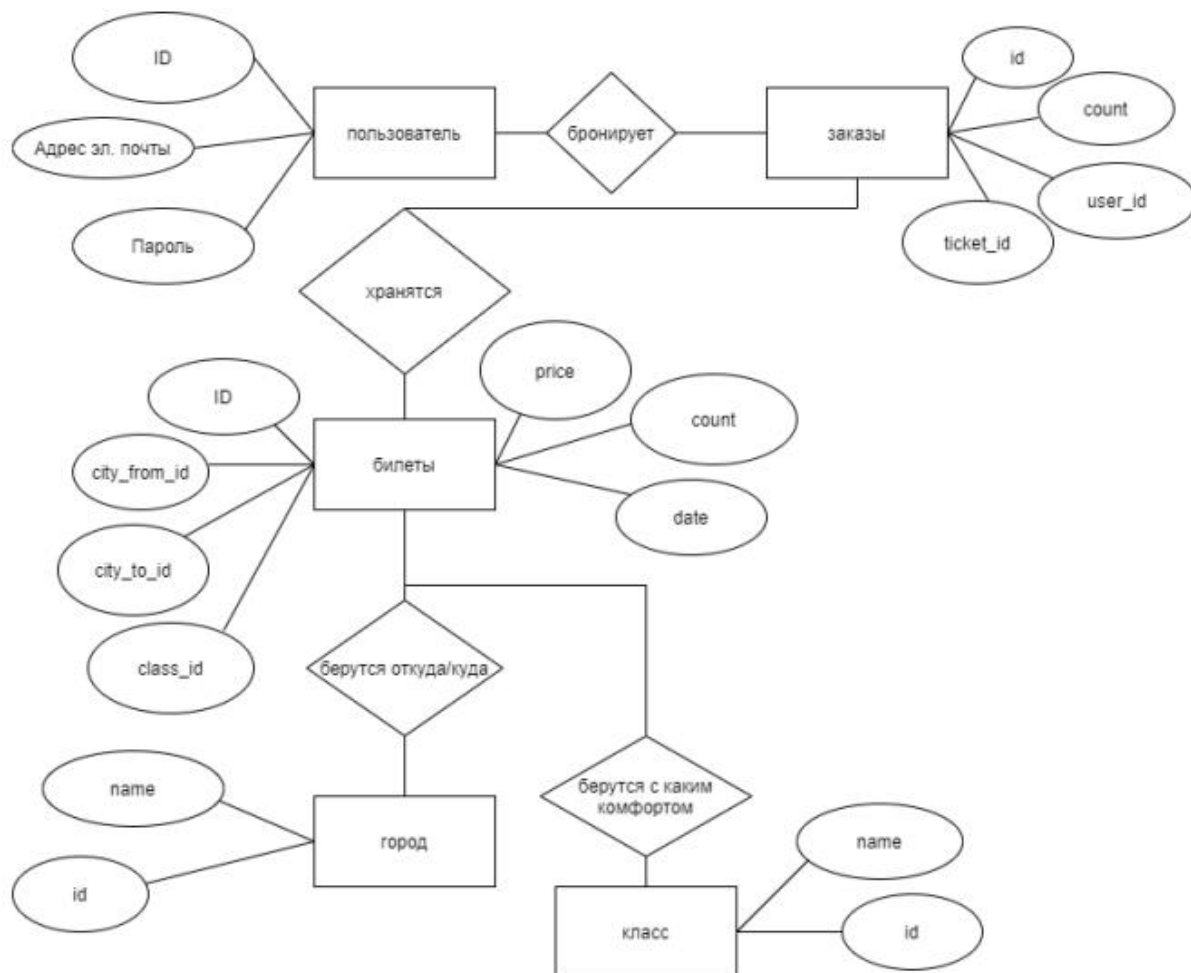


Рисунок 3.1 - Схема базы данных

В phpMyAdmin создадим базу данных для web-сайта, а в ней создадим пять таблиц – Users для пользователей, Tickets для авиабилетов, City для городов, Orders для заказов, Class для классов (Рисунок 3.1).

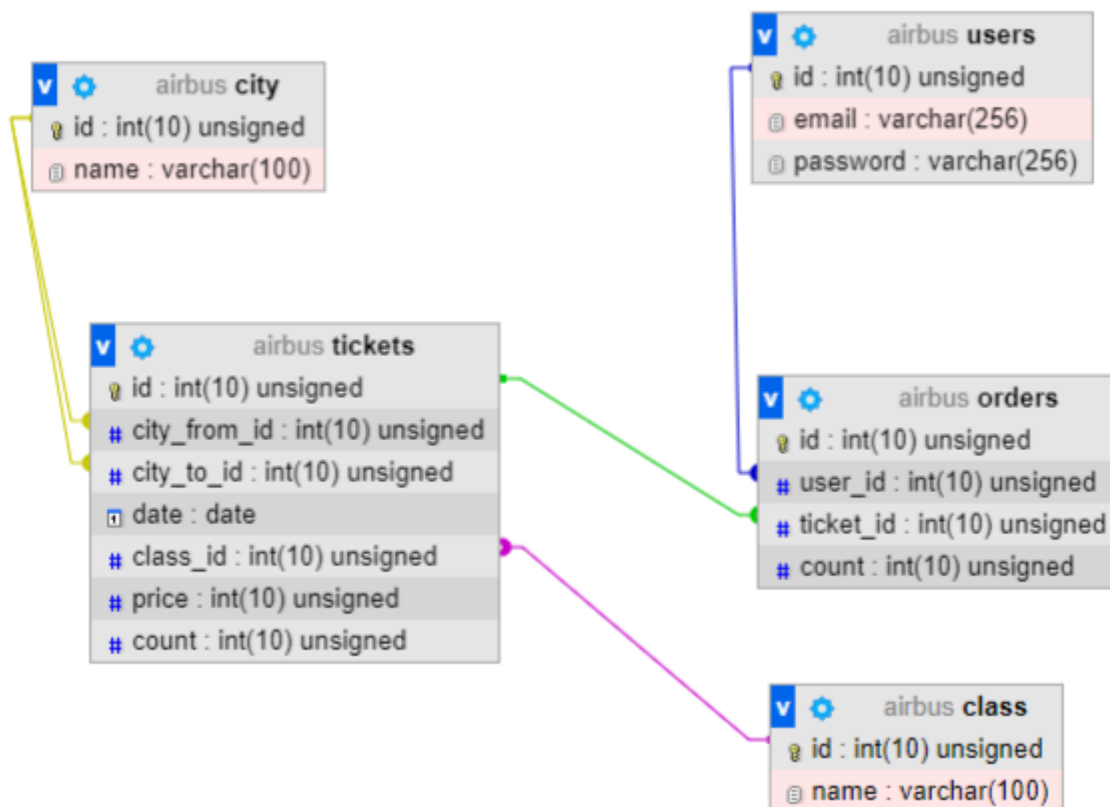


Рисунок 3.2 - Реализация связей

Содержание таблиц созданной базы представлено на рисунках 3.2-3.6:

В таблице City 2 поля: id и название города.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие		
1	id	int(10)		UNSIGNED	Нет	Нем		AUTO_INCREMENT			
2	name	varchar(100)	utf8_general_ci		Нет	Нем					

Рисунок 3.3 – Содержание таблицы City

В таблице Class 2 поля: id и название класса.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие		
1	id	int(10)		UNSIGNED	Нет	Нем		AUTO_INCREMENT			
2	name	varchar(100)	utf8_general_ci		Нет	Нем					

Рисунок 3.4 – Содержание таблицы Class

В таблице Orders 4 поля: id, id пользователя, id билета и их количество.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие		
1	id	int(10)		UNSIGNED	Нет	Нем		AUTO_INCREMENT			
2	user_id	int(10)		UNSIGNED	Нет	Нем					
3	ticket_id	int(10)		UNSIGNED	Нет	Нем					
4	count	int(10)		UNSIGNED	Нет	Нем					

Рисунок 3.5 – Содержание таблицы Orders

В таблице Tickets 7 полей: id, id города вылета, id города прибытия, дата вылета и прибытия, id класса, которым собирается полететь пользователь, цена за билет, количество билетов.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие		
1	id	int(10)		UNSIGNED	Нет	Нем		AUTO_INCREMENT			
2	city_from_id	int(10)		UNSIGNED	Нет	Нем					
3	city_to_id	int(10)		UNSIGNED	Нет	Нем					
4	date	date			Нет	Нем					
5	class_id	int(10)		UNSIGNED	Нет	Нем					
6	price	int(10)		UNSIGNED	Нет	Нем					
7	count	int(10)		UNSIGNED	Нет	Нем					

Рисунок 3.6 – Содержание таблицы Tickets

В таблице Users 3 поля: id, электронная почта пользователя, пароль пользователя к электронной почте.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие		
1	id	int(10)		UNSIGNED	Нет	Нем	User id	AUTO_INCREMENT			
2	email	varchar(256)	utf8_general_ci		Нет	Нем	User email				
3	password	varchar(256)	utf8_general_ci		Нет	Нем					

Рисунок 3.7 – Содержание таблицы Users

Далее заполним каждую таблицу базы данных макетными данными, используя phpMyAdmin (Рисунок 3.8-3.12)

←T→				id	name
<input type="checkbox"/>				1	Москва
<input type="checkbox"/>				4	Новороссийск
<input type="checkbox"/>				6	Владивосток
<input type="checkbox"/>				7	Ростов-на-Дону
<input type="checkbox"/>				9	Санкт-Петербург
<input type="checkbox"/>				12	Новосибирск
<input type="checkbox"/>				14	Казань
<input type="checkbox"/>				15	Екатеринбург
<input type="checkbox"/>				17	Нижний Новгород
<input type="checkbox"/>				18	Челябинск
<input type="checkbox"/>				21	Самара
<input type="checkbox"/>				22	Омск

Рисунок 3.8 - Заполненная таблица City

←T→				id	name
<input type="checkbox"/>				1	Бюджет
<input type="checkbox"/>				2	Комфорт
<input type="checkbox"/>				3	Бизнес

Рисунок 3.9 - Заполненная таблица Class

←T→				id	user_id	ticket_id	count
<input type="checkbox"/>				7	13	1	2

Рисунок 3.10 - Заполненная таблица Orders (Заполняется пользователем, когда он заказывает авиабилеты)

←T→				id	city_from_id	city_to_id	date	class_id	price	count
<input type="checkbox"/>				1	1	7	2020-12-30	2	9000	28
<input type="checkbox"/>				2	1	7	2020-12-30	1	6000	17
<input type="checkbox"/>				3	21	12	2020-12-30	3	7000	50
<input type="checkbox"/>				4	1	7	2020-12-30	3	12000	4
<input type="checkbox"/>				5	1	7	2020-12-30	1	5000	15
<input type="checkbox"/>				6	1	7	2020-12-30	2	6500	17

Рисунок 3.11 - Заполненная таблица Tickets

←T→				id	email	password
				User id	User email	
<input type="checkbox"/>				13	TEST@mail.ru	c16ea6ac5479fb64005672a005a5a7c8

Рисунок 3.12 - Заполненная таблица Users (С помощью формы регистрации)

3.2. ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ЗАПРОСОВ

Запросы к базе данных:

Функциональные пользовательские запросы:

1. Вывести список городов, в которых есть аэропорты

```
SELECT DISTINCT `name` FROM `city`
```

2. Вывести количество доступных билетов без одинакового количества.

```
SELECT DISTINCT `count` FROM `tickets`
```

Функциональные запросы для организации:

1. Добавить новый город.

```
INSERT INTO `city`(`id`, `name`) VALUES (23, "Архангельск")
```

2. Удалить новый город.

```
DELETE FROM `city` WHERE `city`.`id` = 23
```

3. Изменение цены билета.

```
UPDATE `tickets` SET `price` = '9500' WHERE `tickets`.`id` = 1
```

4. Изменение количества билетов.

```
UPDATE `tickets` SET `count` = '30' WHERE `tickets`.`id` = 1
```

5. Изменение даты вылета.

```
UPDATE `tickets` SET `date` = '2021-01-20' WHERE `tickets`.`id` = 1
```

6. Изменение названия города.

```
UPDATE `city` SET `name` = 'Новосибирск' WHERE `city`.`id` = 1
```

7. Изменение количества забронированных билетов пользователем.

```
UPDATE `orders` SET `count` = '3' WHERE `orders`.`id` = 7
```

4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ САЙТА

4.1 ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ САЙТА

Web-сайт представляет собой набор связанных между собой страниц. Структурная схема проекта представлена на рисунке 4.1.

Имя	Дата изменения	Тип	Размер
.idea	20.12.2020 11:16	Папка с файлами	
app	20.12.2020 11:16	Папка с файлами	
images	20.12.2020 11:16	Папка с файлами	
pages	20.12.2020 11:16	Папка с файлами	
templates	31.12.2020 20:46	Папка с файлами	
airbus	20.12.2020 11:17	Файл "SQL"	6 КБ
index.php	20.12.2020 11:14	Файл "PHP"	6 КБ
script	20.12.2020 1:14	файл JavaScript	4 КБ
style	20.12.2020 1:26	CSS-документ	4 КБ

Рисунок 4.1 - Структурная схема проекта

Страницы представляют собой следующие php-файлы:

1. Orders.php – файл для вывода информации о заказах
2. AboutUs.php – файл для вывода дополнительной информации.
3. footer.php - файл, содержащий “подвал” сайта, он подключается после основного блока с помощью функции require.
4. header.php – файл, содержащий “шапку” сайта, он подключается перед основным блоком с помощью функции require.
5. index.php – основной файл страницы сайта, задающий её структуру, на нем имеется возможность использования поисковой строки.
6. Auth.php – файл авторизации
7. Book.php – файл для заказа авиабилетов
8. Cities.php - файл для добавления городов в базу данных
9. Classes.php - файл для добавления классов в базу данных

10. Logout.php - файл для выхода из личного кабинета
11. Register.php - файл для регистрации пользователей
12. Search.php - файл для поиска авиабилетов
13. Database.php – файл базы данных
14. Admin.php – файл Админ панели
15. Create.php – создание новых билетов
16. Remove.php – удаление билетов

При написании программного кода, который содержится в выше перечисленных файлах использовались следующие функции:

1. `<?php ?>` - объявляет участок кода php
2. `include имя_файла` - функция для подключения дополнительных файлов
3. `mysqli_connect()` - открывает новое соединение с сервером MySQL
4. `mysqli_query ()` - выполняет запрос к базе данных
5. `mysqli_fetch_row ()` - получение строки результирующей таблицы в виде массива
6. `mysqli_num_rows ()` - получает число рядов в результирующей выборке
7. `mysqli_fetch_assoc ()` - извлекает результирующий ряд в виде ассоциативного массива
8. `mysqli_close ()` - закрывает ранее открытое соединение с базой данных
9. `trim ()` - удаляет пробелы из начала и из конца строки.
10. `setcookie()` - устанавливает COOKIE
11. `empty ()` - проверяет, пуста ли переменная
12. `exit` - заканчивает выполнение скрипта
13. `isset ()` - определяет, была ли установлена переменная значением, отличным от null

4.2 ОПИСАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

На главной странице находится поиск авиабилетов по параметрам (Рисунок 4.2).

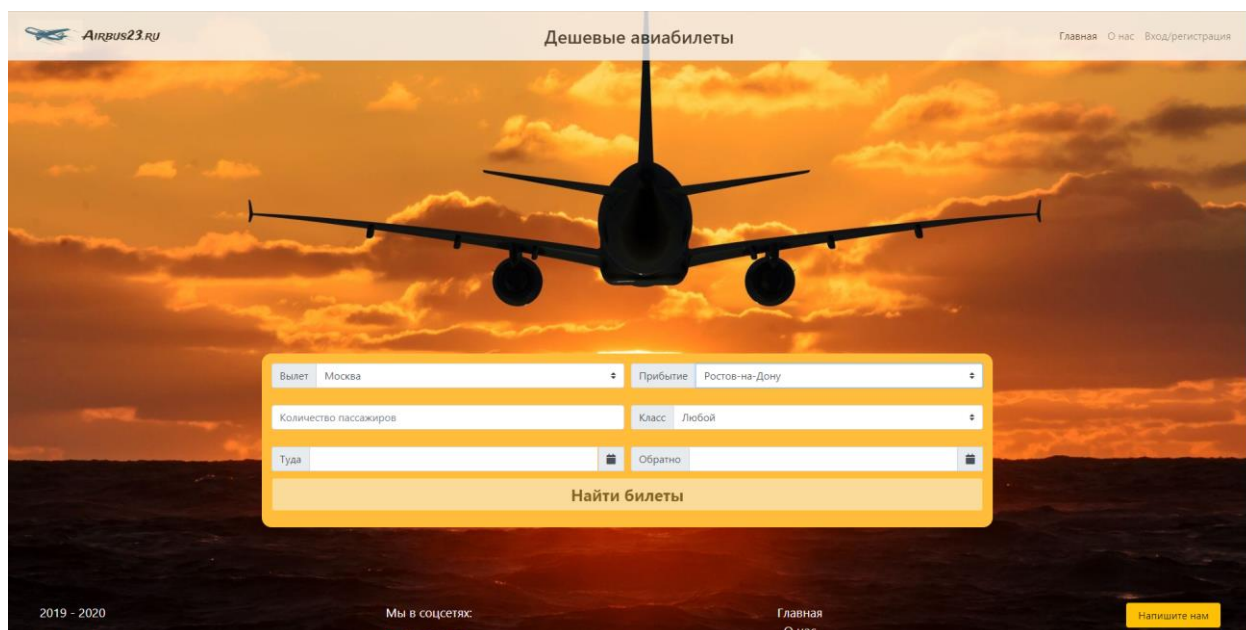


Рисунок 4.2 - Главная страница

Найдём билет (рисунок 4.3) по нужным параметрам, результат представлен на рисунке 4.4.

This image is a close-up of the search form from the previous screenshot. The 'Вылет' field is set to 'Москва' and the 'Прибытие' field is set to 'Ростов-на-Дону'. The 'Количество пассажиров' field contains the number '1'. The 'Класс' field is set to 'Любой'. The 'Туда' date field shows '30.12.2020' and the 'Обратно' date field shows '07.01.2021'. Both date fields include a calendar icon. The 'Найти билеты' button is prominent at the bottom.

Рисунок 4.3 – Форма поиска

ВылетМосква

ПрибытиеРостов-на-Дону

1

КлассЛюбой

Туда30.12.2020

Обратно07.01.2021

Найти билеты

Город отправки: Москва

Город прибытия: Ростов-на-Дону

Класс: Комфорт

Цена: 9000 Р

Билетов в наличии: 28

Забронировать

Город отправки: Москва

Город прибытия: Ростов-на-Дону

Класс: Бюджет

Цена: 6000 Р

Билетов в наличии: 17

Забронировать

Город отправки: Москва

Город прибытия: Ростов-на-Дону

Класс: Бизнес

Цена: 12000 Р

Билетов в наличии: 4

Забронировать

Город отправки: Москва

Город прибытия: Ростов-на-Дону

Класс: Бюджет

Цена: 5000 Р

Билетов в наличии: 15

Забронировать

Город отправки: Москва

Город прибытия: Ростов-на-Дону

Класс: Комфорт

Цена: 6500 Р

Билетов в наличии: 17

Забронировать

Рисунок 4.4 – Результат поиска

Переходим на вкладку “О нас” (рисунок 4.9). Здесь представлена дополнительная информация

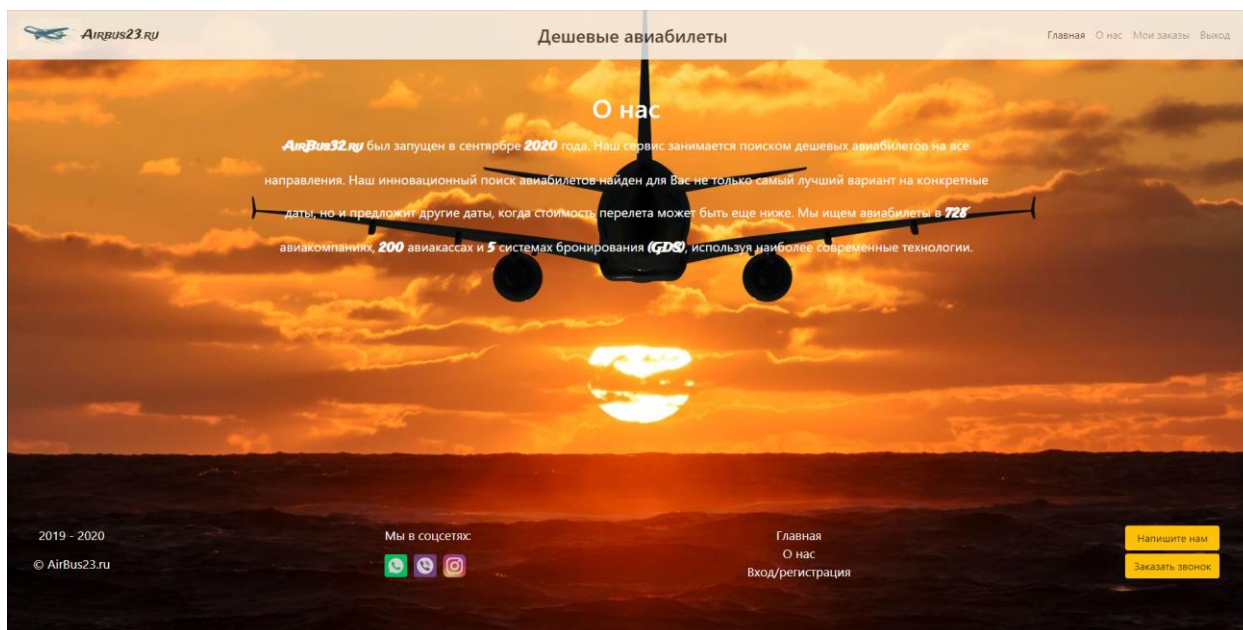


Рисунок 4.5 – Вкладка “О нас”

Переходим на вкладку “Регистрация” (рисунок 4.6). Здесь представлена форма регистрации, необходимо ввести следующие данные: e-mail, пароль. Если не заполнить хотя бы одно поле или ввести недопустимые символы, то регистрация не произойдет, произойдет вывод ошибки (рисунок 4.7).

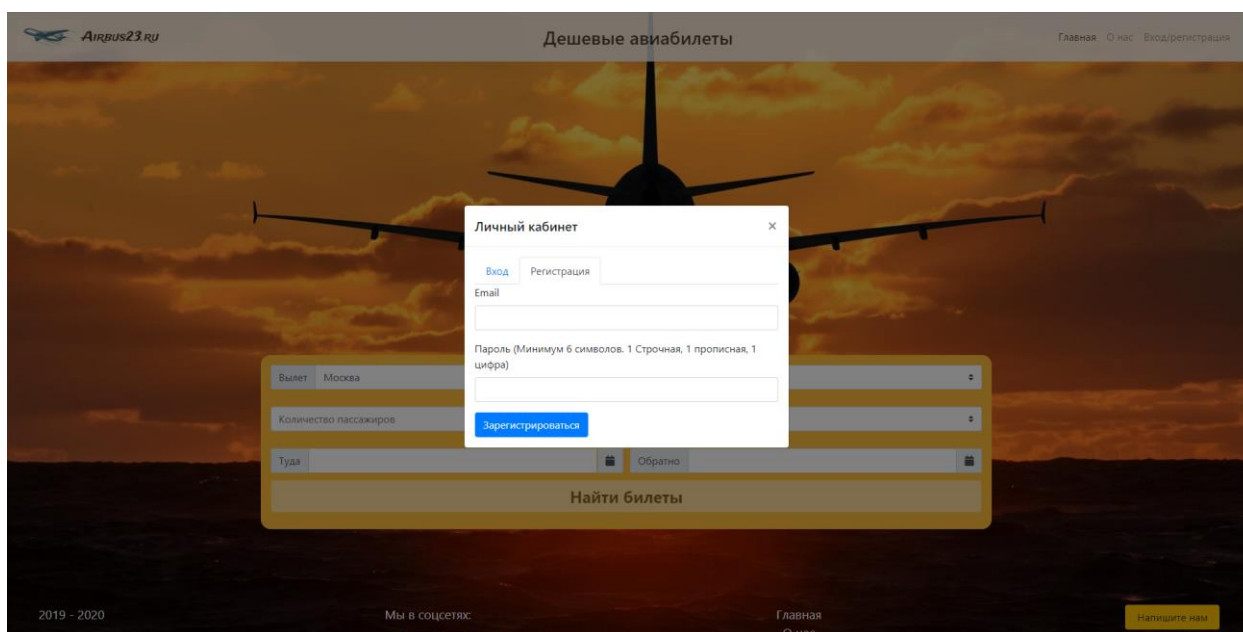


Рисунок 4.6 - Вкладка “Регистрация”

Рисунок 4.7 – Вывод сообщения об ошибках

Рисунок 4.8 – Вывод сообщения об ошибках

Заполним форму данными, которые показаны на рисунке 4.9, и нажмем кнопку отправить (рисунок 4.10). Данные с формы были записаны в таблицу users (рисунок 4.11)

Личный кабинет ×

[Вход](#) [Регистрация](#)

Email

TEST23@mail.ru 🔑

Пароль (Минимум 6 символов. 1 Строчная, 1 прописная, 1 цифра)

.....

[Зарегистрироваться](#)

Рисунок 4.9 – Заполнение формы

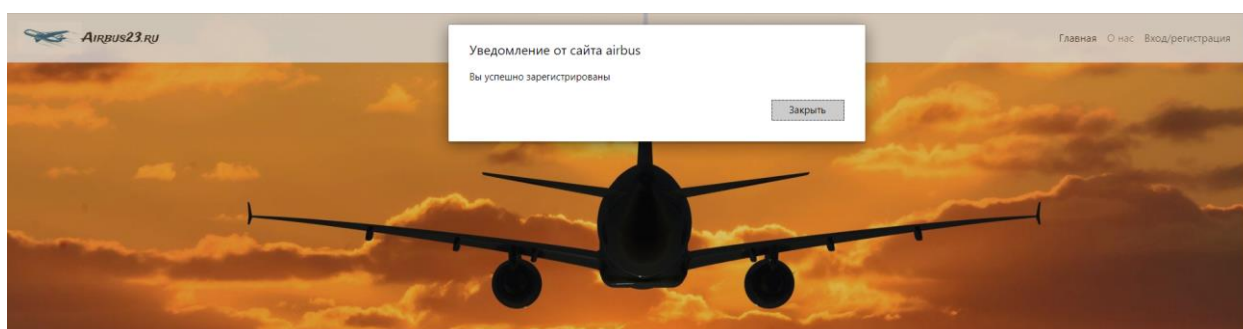


Рисунок 4.10 – Вывод сообщения об успехе










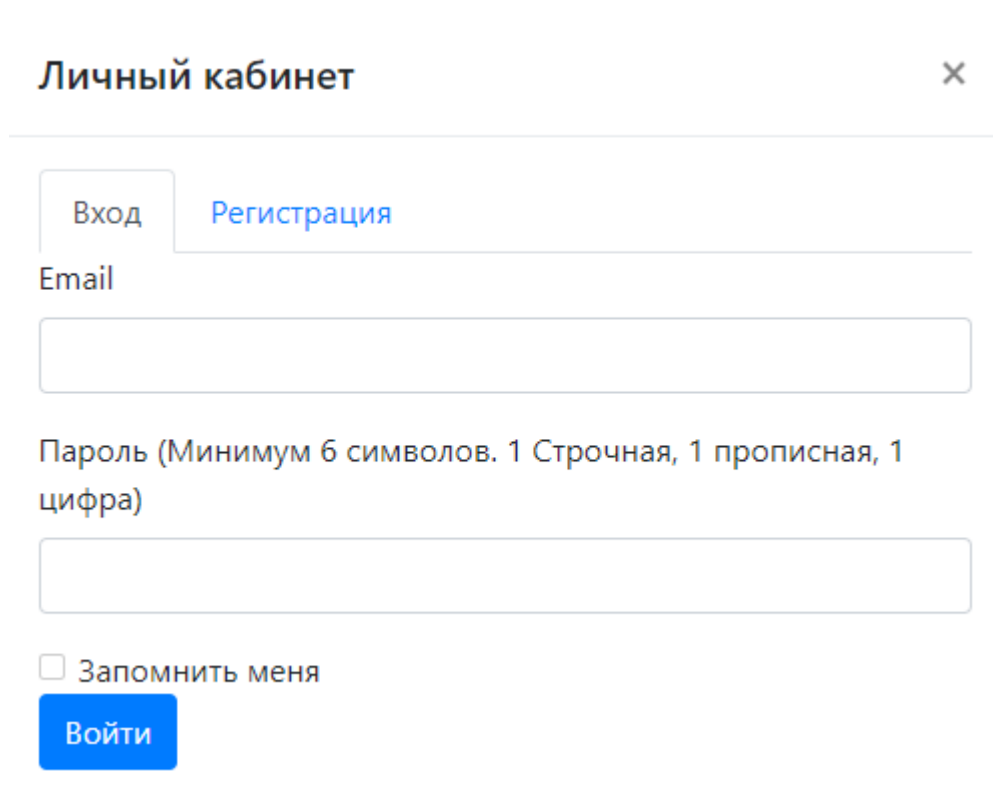
<div>← T →</div>					id	email	password
					User id	User email	
<input type="checkbox"/>					13	TEST@mail.ru	c16ea6ac5479fb64005672a005a5a7c8
<input type="checkbox"/>					15	TEST@mail.ru	bd9883ea6062ac01c74846eb45752f32
<input type="checkbox"/>					16	TEST23@mail.ru	f630fc575a411d9c3fdcac352f92e0e7

Рисунок 4.11 – Результат внесения

Переходим на вкладку “Вход” (рисунок 4.12). Здесь представлена форма входа, необходимо ввести данные регистрации, а именно: email и пароль. Если не заполнить хотя бы одно поле, то произойдет вывод ошибки (рисунок 4.13).

Заполним форму данными (рисунок 4.14) и нажмем кнопку отправить (рисунок 4.15).



The screenshot shows a web form titled "Личный кабинет" (Personal Account) with a close button (X) in the top right corner. Below the title, there are two tabs: "Вход" (Login) and "Регистрация" (Registration). The "Регистрация" tab is currently selected. The form contains the following elements:

- An "Email" label followed by a text input field.
- A "Пароль (Минимум 6 символов. 1 Строчная, 1 прописная, 1 цифра)" (Password (Minimum 6 symbols. 1 Lowercase, 1 Uppercase, 1 digit)) label followed by a password input field.
- A checkbox labeled "Запомнить меня" (Remember me).
- A blue button labeled "Войти" (Login).

Рисунок 4.12 - Вкладка "Вход"

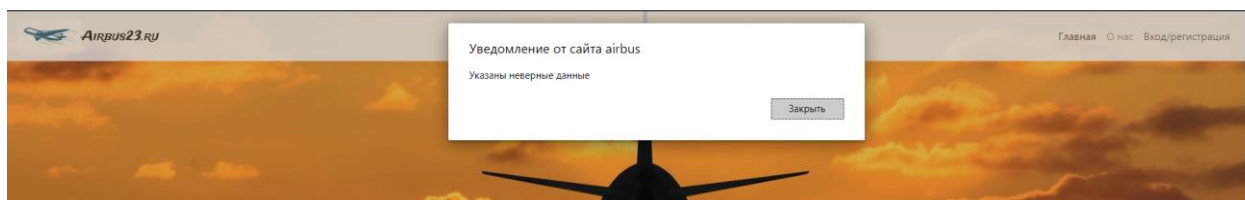


Рисунок 4.13 - Вывод сообщения об ошибках

Рисунок 4.14 – Заполнение формы

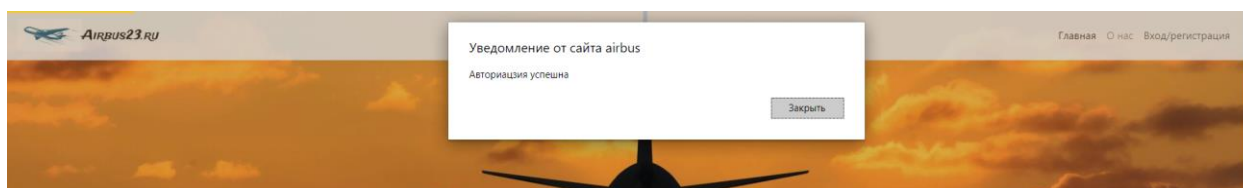


Рисунок 4.15 - Вывод сообщения об успехе

Переходим на вкладку “Мои заказы”, предварительно заказав авиабилеты (рисунок 4.16). Здесь представлена информация об уже забронированных авиабилетах. Эта вкладка доступна только для авторизованных пользователей.

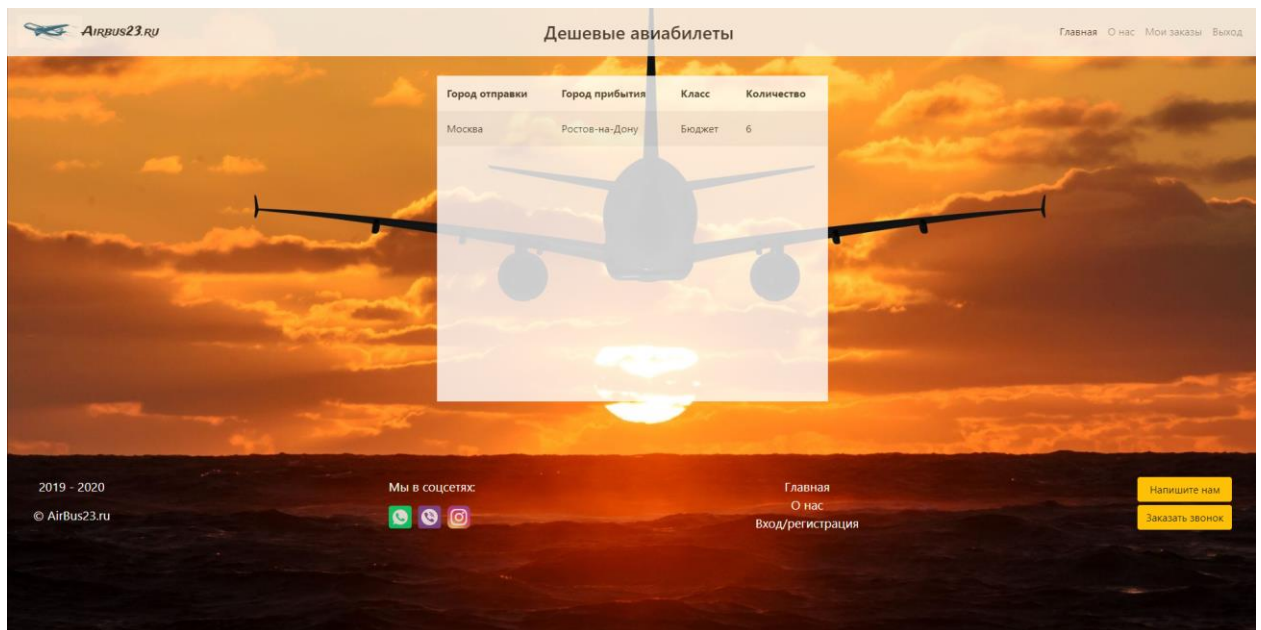


Рисунок 4.16 - Вкладка “Мои заказы”

Также, администратор сайта может менять данные не только в phpMyAdmin, но и на сайте.

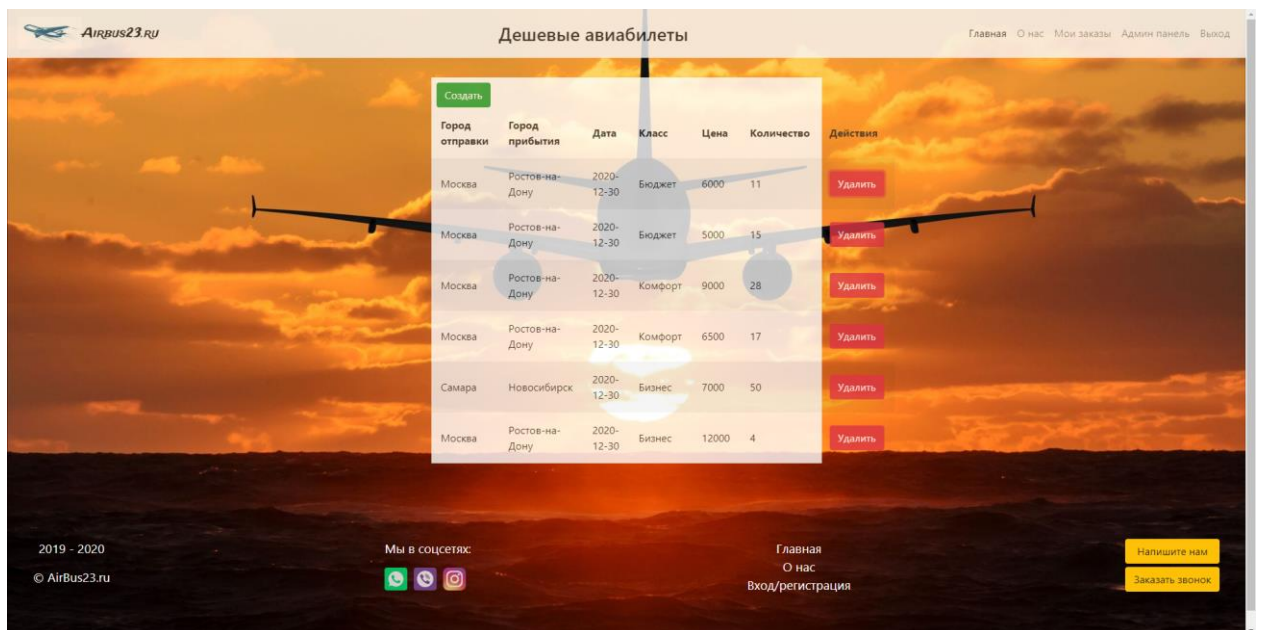


Рисунок 4.17 – Вкладка Админ панель.

5. ОЦЕНКА ЭФФЕКТИВНОСТИ И ТЕСТИРОВАНИЕ САЙТА

Тестирование сайта является одним из важных этапов разработки. Тестирование и отлов ошибок предшествуют запуску проекта и выполняются после всех остальных этапов, ведь досконально проверять имеет смысл только уже готовый продукт.

Тестирование сайта производилось с помощью встроенного в браузер решения Lighthouse, который проверяет сайт по следующим параметрам:

1. Performance - время до взаимодействия, время ожидания, индекс скорости, оптимизация ресурсов, TTFB, доставка ресурсов, время выполнения сценариев и тд.
2. Accessibility - Проверяет наличие распространенных проблем, которые могут помешать пользователям получить доступ к вашему контенту.
3. Best Practices - Ищет все, от использования HTTPS до исправления пропорций изображения.

4. SEO - Проверяет наличие базовых рекомендаций, таких как canonical, title, различные мета-теги.
5. Progressive Web App - Оценивает страницу по контрольному чек-листу для прогрессивного веб-приложения.

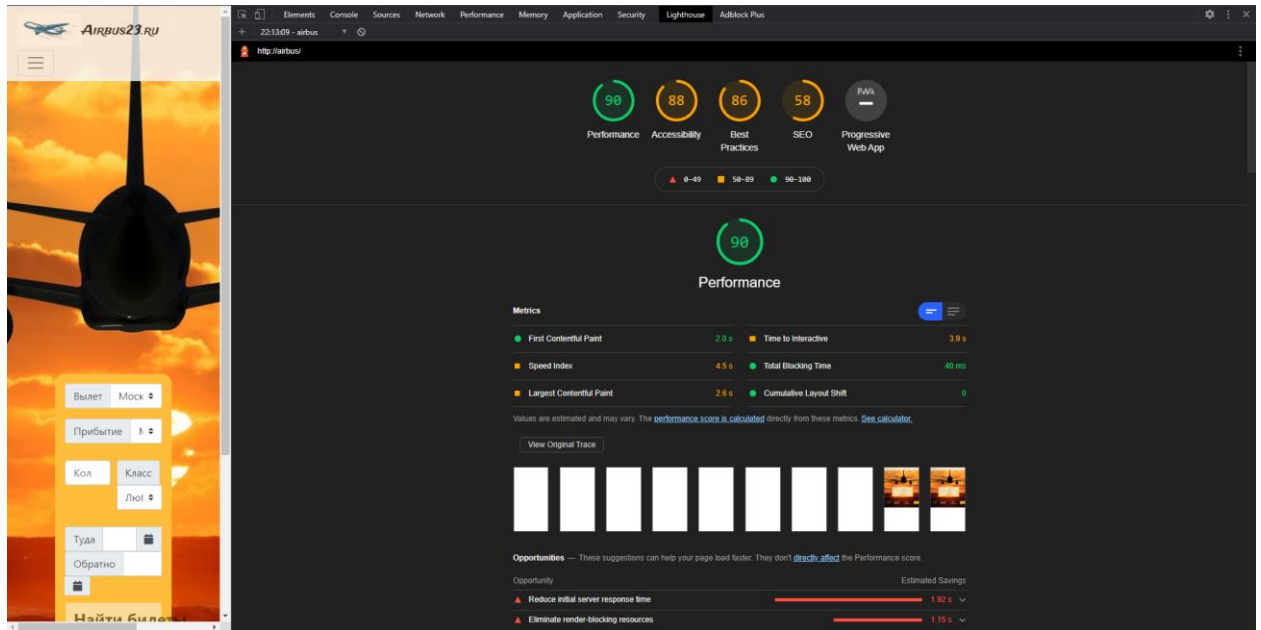


Рисунок 5.1 – Оценка сайта

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы были изучены принципы разработки web-сайта, основы работы с СУБД MySQL, язык программирования PHP, HTML. На основе полученных знаний был спроектирован и разработан собственный web-сайт

Также были решены следующие задачи:

1. Описать предметную область Интернет- приложения
2. Спроектировать Web- сайт
3. Описать информационное обеспечение сайта
4. Выполнить программную реализацию сайта
5. Оценить эффективность и протестировать сайт

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Разработка приложений для Интернет: Методические указания по выполнению курсовой работы/Панфилов Юж.-Рос. гос. политехн. ун-т (НПИ) имени М.И. Платова.- Новочеркасск: ЮРГПУ (НПИ) имени М.И. Платова, 2016.- 28с.
2. <https://www.php.net/manual/ru/ref.mysql.php> - интернет-ресурс содержащий справочные материалы для работы с СУБД MySQL.
3. <http://htmlbook.ru/html> - интернет-ресурс содержащий справочные материалы по языку гипертекстовой разметки HTML5.
4. <https://www.php.net> - интернет-ресурс содержащий справочные материалы по языку программирования PHP.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ WEB САЙТА

1 ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Сайт по бронированию авиабилетов. Имя сайта (название домена):

www.Airbus23.ru

1.3 Наименования организации-заказчика и организаций участников работ

Заказчиком системы является Панфилов Александр Николаевич.

1.4 Перечень документов, на основании которых создается система

Основанием для разработки является типовый договор на разработки Web-проекта.

1.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ по созданию – 4 сентября 2020 года.

Плановый срок окончания работ - 25 декабря 2020 года.

1.6 Порядок оформления и предъявления заказчику результатов работ по

созданию системы Система передается в виде функционирующего

комплекса на базе средств вычислительной техники Заказчика и

Исполнителя в сроки, установленные Договором. Приемка системы

осуществляется комиссией в составе уполномоченных представителей

Заказчика и Исполнителя. 25

1.7 Перечень нормативно-технических документов, методических

материалов, использованных при разработке ТЗ При разработке

автоматизированной системы и создании проектно-эксплуатационной

документации Исполнитель должен руководствоваться требованиями

следующих нормативных документов:

–ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы.

Автоматизированные системы. Стадии создания;

–ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на

автоматизированные системы. Виды, комплексность и обозначение

документов при создании автоматизированных систем; –РД 50-34.698-90.

Методические указания. Информационная технология. Комплекс

стандартов на автоматизированные системы. Автоматизированные

системы. Требования к содержанию документов.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Представление в Интернете: сервиса по бронированию авиабилетов,

информации о сервисе бронирования. Доступ к контактной информации,

справочному материалу, покупке авиабилетов, ценам на оказываемые

услуги, истории сервиса.

2.2 Цели создания системы Сайт должен способствовать привлечению

клиентов.

3 ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ На web-Сайте

доступно бронирование онлайн, просмотр специальных предложений и

популярных направлений. Информация для устройства на работу

4 ТРЕБОВАНИЯ К СИСТЕМЕ

4.1 Язык сайта. Русский.

4.2 Объем и состав текстовой информации. Согласно Приложению 1.

4.3 Основные ключевые слова, по которым сайт должны находить по

запросам в поисковых системах и Интернет – каталогах: билет, авиабилет,

самолёт, путешествие, сервис, дешевые авиабилеты, бронирование, поиск

билетов, полёт, небо, бизнес-класс, эконо-класс, отдых, перелёт,

авиакомпания, онлайн, пилот, стюардесса, взлёт, посадка, места, аэропорт,

город, маршрут, страны.

- 4.4 Билет, авиабилет, самолёт, путешествие, сервис, дешёвые авиабилеты, бронирование, поиск билетов, полёт, небо, бизнес-класс, эконом-класс, отдых, перелёт, авиакомпания, онлайн, пилот, стюардесса, взлёт, посадка, места, аэропорт, город, маршрут, страны.
- 4.5 Объём и состав графической информации не оговорён.
- 4.6 Общий объём и состав текстовой и графической информации, переданный Заказчиком в электронном виде.
- 4.7 Предполагаемая возрастная аудитория сайта. От 18 лет и старше.
- 4.8 Количество страниц сайта. Сайт должен содержать следующие отдельные html страницы: 1 – Главная (домашняя) страница; 2 – Поиск авиабилетов; 3 – О нас; 4 – Регистрация/Вход;
- 4.9 Кнопки управления (навигация сайта). С каждой страницы сайта должен быть обеспечен переход (установлена гиперссылка) на главную (начальную) страницу сайта.
- 4.10 Схема сайта. Главная (начальная) страница сайта должна содержать гиперссылки, обеспечивающие переход с нее на не менее чем 95% страниц сайта, но не более чем 50 гиперссылок. Раздел дополняется графической схемой навигации.
- 4.11 Общий объём сайта, Мб. Не оговаривается.
- 4.12 Оформление рисунков. Все рисунки объемом более 1 Мб должны быть выполнены с замещающим текстом. Рисунки размером более 5 Мб должны быть выполнены с предпросмотром (предпросмотр – уменьшенное изображение рисунка, при нажатии мышкой на которое открывается полное или более крупное изображение того же рисунка). Формат всех рисунков gif или jpg (jpeg), а также png.
- 4.13 Пропускная способность линии связи. Среднее время загрузки страниц не должно превышать 28 секунд при скорости соединения 28.8 Кбит/сек. Допускается увеличение времени загрузки отдельных страниц до 36 секунд, но не более чем на 30% общего числа страниц сайта. Главная (начальная) страница должна иметь время загрузки не более 40 секунд.
- 4.14 Основной диапазон разрешения мониторов, на которых будет просматриваться сайт: 1920x1080.
- 4.15 Основной браузер, которым будет просматриваться сайт, и его минимальная версия: Yandex Browser 20.12.1.
- 4.16 Цветовая палитра. Основной режим мониторов, на которых будет просматриваться сайт: 15 разрядов цветов и выше (число цветов 65536 и выше). При разработке сайта должен быть обеспечена возможность его просмотра при использовании безопасной цветовой палитры (разрядность цветов 8). Изменения оттенков цветов, при просмотре сайта с использованием безопасной цветовой палитры, не оговариваются.
- 4.17 Общий фон сайта. Общий фон сайта светлый. Допускается использование фоновой рисунка, а также использование эффекта градиент от темного цвета до светлого и обратно.
- 4.18 Размер и вид шрифта сайта. Размер шрифта сайта должен быть в пределах 10-16 для оформления текста. Размер шрифта для оформления заголовков, названия страниц и т.д. не оговаривается. Вид (название) шрифта не оговаривается.
- 4.19 Срок разработки сайта. Три недели со дня зачисления 100 % предоплаты на расчётный счёт Исполнителя.
- 4.20 Порядок передачи сайта. Исполнитель выгружает сайт на хостинг,

оформленный на Заказчика, а также передаёт логин, пароль и название (код передачи данных) по протоколу ftp. Заказчик обязан проверить наличие грамматических и орфографических ошибок на сайте в течение трех рабочих дней. Обнаруженные ошибки Исполнитель обязан устранить в течение трех рабочих дней.

4.21 Дополнительные условия. 28 Каждая страница сайта должна содержать логотип и название Фирмы. Внизу на каждой странице сайта должна быть указана контактная информация.

Приложение к ТЗ

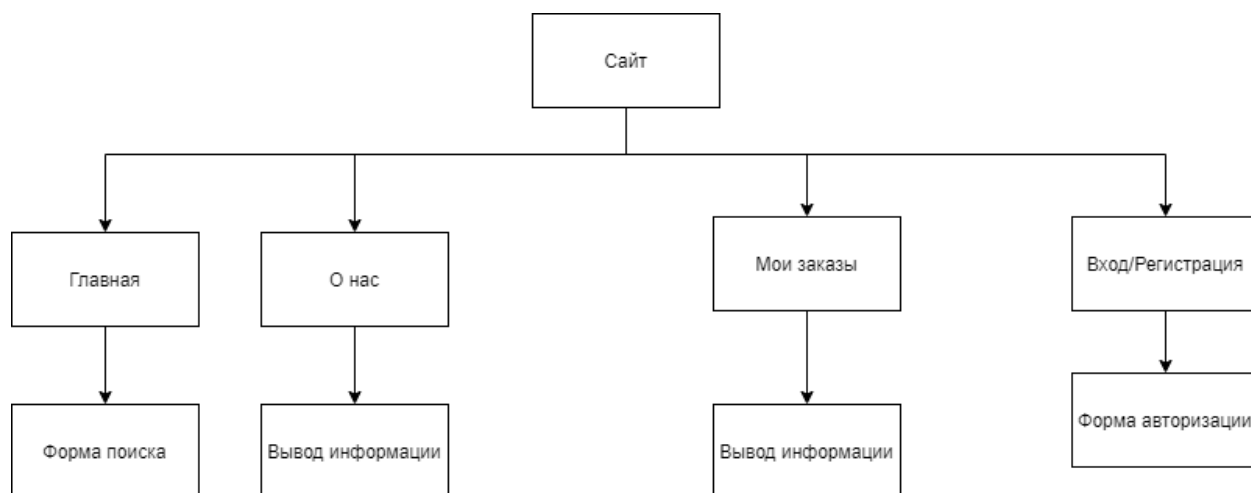


Рисунок – Схема сайта

ПРИЛОЖЕНИЕ В
ЛИСТИНГ CSS ФАЙЛА

Style.css

```
/* header */
#navbarNavAltMarkup {
    justify-content: flex-end;
}

.bg-light {
    opacity: 0.8;
}

a.navbar-brand {
    width: 300px;
    font-family: 'Trade Winds', cursive;
}

.main-text {
    display: block;
    margin: 0 auto;
}

.logo {
    height: 50px;
}

body {
    background: url("images/main.jpg") no-repeat top;
    background-size: cover;
}

.content {
    min-height: 250px;
}

.aboutUs {
    line-height: 50px;
    width: 60%;
    margin: 50px auto 0;
    height: 550px;
    color: white;
    text-align: center;
    font-size: 20px;
}

.marked {
    font-family: 'Trade Winds', cursive;
    font-weight: bold;
}

.search-form-block {
    width: 60%;
```

```

        margin: 450px auto 0;
        background-color: #FFBD3B;
        padding: 15px;
        border-radius: 15px;
    }

    .form-row {
        margin-bottom: 10px;
    }

    #searchButton {
        background-color: #f8f9fa;
        opacity: 0.5;
        color: black;
        width: 100%;
        font-size: 25px;
        font-weight: bold;
    }

    /* footer */
    .footer-container {
        color: white;
        font-size: 1.2rem;
        height: 260px;
        padding: 30px 40px;
        display: flex;
        align-items: center;
        margin-top: 30px;
    }

    .footer-row {
        display: flex;
        flex-direction: row;
        justify-content: space-between;
        width: 100%;
    }

    .social-networks-column {;
        display: flex;
        flex-direction: column;;
    }

    .social-networks {
        display: flex;
        flex-direction: row;
        justify-content: center;
    }

    .social-networks div {
        width: 35px;
        height: 35px;
        margin-right: 10px;
    }

```

```

}

.social-networks div img {
    width: 35px;
    opacity: 0.8;
}

.social-networks div img:hover {
    opacity: 1;
}

.menu-footer {
    display: flex;
    flex-direction: column;
    align-items: center;
}

.menu-footer a {
    color: white !important;
    text-decoration: none;
}

.menu-footer a:hover {
    text-decoration: underline;
}

.footer-modals {
    display: flex;
    flex-direction: column;
}

.footer-modals button {
    margin-bottom: 5px;
}

.footer-modals a {
    text-decoration: none;
    color: white;
}

.footer-modals a:hover {
    text-decoration: underline;
}

#toTop {
    width: 50px;
    background-color: #424242;
    height: 50px;
    text-align: center;
    border-radius: 4px;
    margin: 30px;
    position: fixed;
}

```

```

        bottom: 20px;
        right: 0;
        z-index: 1000;
        opacity: 1;
        display: none;
        font-size: 26px;
    }

    #toTop:hover {
        cursor: pointer;
        opacity: .8;
    }

    #toTop:active {
        background-color: #555
    }

    #toTop i:before {
        display: block;
        position: relative;
        top: 7px;
        color: white;
    }

    .noFoundTickets {
        color: white;
        text-align: center;
    }

    .search-result {
        display: flex;
        flex-flow: row wrap;
        justify-content: start;
    }

    .ticketInfo {
        background-color: #f8f9fa;
        border-radius: 8px;
        width: 250px;
        margin-right: 20px;
        padding: 15px;
        margin-top: 15px;
    }

    .orders {
        min-height: 500px;
        width: 600px;
        background-color: white;
        margin: 30px auto 0;
        opacity: 0.8;
    }

```

```
.orders table {  
    width: 100%;  
}  
table th, td{  
    padding: 15px;  
}
```

ПРИЛОЖЕНИЕ С
ЛИСТИНГ PHP ФАЙЛОВ

1) Index.php

```
<?php
include_once 'templates/header.php';

$cities = include_once './app/cities.php';
?>
<div class="container-fluid content">
  <div class="search-form-block">
    <form action="" novalidate class="search-form">
      <div class="form-row">
        <div class="col">
          <div class="input-group mb-3">
            <div class="input-group-prepend">
              <label class="input-group-text" for="from">Вылет</label>
            </div>
            <select class="custom-select" id="from" required name="from">
              <? foreach ($cities as $city): ?>
                <option value="<?= $city['id'] ?>"><?= $city['name'] ?></option>
              <? endforeach; ?>
            </select>
          </div>
        </div>
        <div class="col">
          <div class="input-group mb-3">
            <div class="input-group-prepend">
              <label class="input-group-text" for="to">Прибытие</label>
            </div>
            <select class="custom-select" id="to" required name="to">
              <? foreach ($cities as $city): ?>
                <option value="<?= $city['id'] ?>"><?= $city['name'] ?></option>
              <? endforeach; ?>
            </select>
          </div>
        </div>
      </div>
      <div class="form-row">
        <div class="col">
          <input type="number" class="form-control" placeholder="Количество
пассажиров" required min="1"
          name="count">
          <div class="invalid-feedback">
            Пожалуйста, укажите количество пассажиров
          </div>
        </div>
        <div class="col">
          <div class="input-group mb-3">
            <div class="input-group-prepend">
              <label class="input-group-text" for="inputGroupSelect01">Класс</label>
            </div>
            <select class="custom-select" id="inputGroupSelect01" required name="class">
```

```

        <option value="0">Любой</option>
        <option value="1">Бюджет</option>
        <option value="2">Комфорт</option>
        <option value="3">Бизнес</option>
    </select>
</div>
</div>
</div>
<div class="form-row">
    <div class="col">
        <div class="input-group date" id="datetimepicker1" data-target-input="nearest">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon1">Туда</span>
            </div>
            <input type="text" class="form-control datetimepicker-input" data-
target="#datetimepicker1"
                aria-describedby="basic-addon1" required name="dateTo">
            <div class="input-group-append" data-target="#datetimepicker1" data-
toggle="datetimepicker">
                <div class="input-group-text"><i class="fa fa-calendar"></i></div>
            </div>
            <div class="invalid-feedback">
                Пожалуйста, укажите дату вылета
            </div>
        </div>
    </div>
    <div class="col">
        <div class="input-group date" id="datetimepicker2" data-target-input="nearest">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon2">Обратно</span>
            </div>
            <input type="text" class="form-control datetimepicker-input" data-
target="#datetimepicker2"
                required name="dateBack">
            <div class="input-group-append" data-target="#datetimepicker2" data-
toggle="datetimepicker">
                <div class="input-group-text"><i class="fa fa-calendar"></i></div>
            </div>
            <div class="invalid-feedback">
                Пожалуйста, укажите дату прибытия
            </div>
        </div>
    </div>
</div>
<div class="form-row">
    <div class="col">
        <button id="searchButton" class="btn btn-warning" type="submit">Найти
билеты</button>
    </div>
</div>
<div class="search-result"></div>
</form>

```

```
</div>
```

```
</div>
```

```
<?php
```

```
include_once 'templates/footer.php'
```

```
?>
```

2) Footer.php

```
<footer>
```

```
<div class="container-fluid footer-container">
```

```
<div class="row-width footer-row">
```

```
<div class="text-center">
```

```
<p>2019 - 2020</p>
```

```
<p>© AirBus23.ru</p>
```

```
</div>
```

```
<div class="social-networks-column">
```

```
<p>Мы в соцсетях:</p>
```

```
<div class="social-networks">
```

```
<div class="social">
```

```
<a href="#">
```

```

```

```
</a>
```

```
</div>
```

```
<div class="social">
```

```
<a href="#">
```

```

```

```
</a>
```

```
</div>
```

```
<div class="social">
```

```
<a href="#">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="menu-footer">
```

```
<a href="../index.php">Главная</a>
```

```
<a href="../pages/aboutUs.php">О нас</a>
```

```
<a href="#">Вход/регистрация</a>
```

```
</div>
```

```
<div class="footer-modals">
```

```
<button class="btn btn-warning footer-button" data-toggle="modal" data-  
target="#contactUsModal">Напишите нам</button>
```

```
<button class="btn btn-warning footer-button" data-toggle="modal" data-  
target="#callbackModal">Заказать звонок</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<a id="toTop" >
```

```
<i class="fa fa-chevron-up"></i>
```

```
</a>
```

```
</footer>
```



```

<script src="https://code.jquery.com/jquery-3.5.1.js" integrity="sha256-
QWo7LDvxbWT2tbbQ97B53yJnYU3WhH/C8ycbRAkjPDc="
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.1/moment.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/tempusdominus-bootstrap-
4/5.1.2/js/tempusdominus-bootstrap-4.min.js" integrity="sha512-
2JBCbWoMJP+Uj7Wq5OLub8E5edWHITM4ar/YJkZh3plwB2INhhOC3eDoqHm1Za/ZOSks
rLIURLoyXVdfQXqwg==" crossorigin="anonymous"></script>
<script src="./script.js"></script>
</body>
</html>

```

3) Header.php

```

<?php session_start() ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Airbus23.ru</title>
    <link
href="https://use.fontawesome.com/releases/v5.0.13/css/all.css"
rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css2?family=Trade+Winds&displ
ay=swap" rel="stylesheet">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootst
rap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAis6JXm
" crossorigin="anonymous">
    <link rel="stylesheet" href="../style.css">
</head>
<body>
<header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Airbus23.ru</a>
        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse"
id="navbarNavAltMarkup">
            <h3 class="main-text">Дешевые авиабилеты</h3>
            <div class="navbar-nav">

```

```

        <a class="nav-item nav-link active"
href="../index.php">Главная<span
class="sr-
only">(current)</span></a>
        <a class="nav-item nav-link"
href="../pages/aboutUs.php">О нас</a>
        <?php if ($_SESSION['name']): ?>
        <a class="nav-item nav-link"
href="../pages/orders.php">Мои заказы</a>
        <?php endif; ?>
        <?php if ($_SESSION['name']): ?>
        <a class="nav-item nav-link"
href="../app/logout.php">Выход</a>
        <?php else: ?>
        <a class="nav-item nav-link" href="#" data-
toggle="modal" data-target="#loginModal">Вход/регистрация</a>
        <?php endif; ?>
    </div>
</div>
</nav>
<!-- Modal -->
<div class="modal fade" id="loginModal" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-
hidden="true">
    <div class="modal-dialog modal-dialog-centered"
role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title"
id="exampleModalLongTitle">Личный кабинет</h5>
                <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <ul class="nav nav-tabs">
                    <li class="nav-item">
                        <a class="nav-link active" data-
toggle="tab" href="#login">Вход</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" data-toggle="tab"
href="#registration">Регистрация</a>
                    </li>
                </ul>
                <div class="tab-content">
                    <div class="tab-pane fade show active"
id="login">
                        <form id="loginForm">
                            <div class="form-group">
                                <label
for="loginEmail">Email</label>

```

```
<input type="text"
class="form-control" name="loginEmail" id="loginEmail" aria-
describedby="emailHelp" required>
</div>
<div class="form-group">
<label
for="loginPassword">Пароль (Минимум 6 символов. 1 Строчная, 1
прописная, 1 цифра)</label>
<input name="loginPassword"
type="password" class="form-control" id="loginPassword" required
minlength="6" pattern="(?!.*\d) (?!.*[a-z]) (?!.*[A-Z]) (?!.*\s).*$">
</div>
<div class="form-check">
<input type="checkbox"
class="form-check-input" id="rememberMe">
<label class="form-check-
label" for="rememberMe">Запомнить меня</label>
</div>
<button type="submit" class="btn
btn-primary">Войти</button>
</form>
</div>
<div class="tab-pane fade"
id="registration">
<form id="registrationForm">
<div class="form-group">
<label
for="registrationEmail">Email</label>
<input type="email"
class="form-control" id="registrationEmail" aria-
describedby="emailHelp" required name="email">
</div>
<div class="form-group">
<label
for="registrationPassword">Пароль (Минимум 6 символов. 1 Строчная,
1 прописная, 1 цифра)</label>
<input type="password"
class="form-control" id="registrationPassword" required
minlength="6" pattern="(?!.*\d) (?!.*[a-z]) (?!.*[A-Z]) (?!.*\s).*$" name="password">
</div>
<button type="submit" class="btn
btn-primary">Зарегистрироваться</button>
</form>
</div>
</div>
</div>
</div>
</div>
</header>
```

4) AboutUs.php

```
<?php
include_once '../templates/header.php';
?>

<div class="container-fluid">
    <div class="aboutUs">
        <h1>О нас</h1>
        <p><span class="marked">AirBus32.ru</span> был
запущен в сентябре <span class="marked">2020</span> года.
        Наш сервис занимается поиском дешевых авиабилетов
на все направления.
        Наш инновационный поиск авиабилетов найден для Вас
не только самый лучший вариант на конкретные даты,
        но и предложит другие даты, когда стоимость
перелета может быть еще ниже. Мы ищем авиабилеты в <span
        class="marked">728</span> авиакомпаниях,
        <span class="marked">200</span> авиакассах и
<span class="marked">5</span> системах бронирования <span
        class="marked">(GDS)</span>, используя
наиболее современные технологии.</p>
    </div>
</div>
```

```
<?php
include_once '../templates/footer.php';
5) championship.php
```

```
<?php
include_once '../templates/header.php';
include_once '../app/Database.php';
$db = new Database();
$orders = $db->getUserOrders();
?>

<div class="container-fluid">
    <div class="row">
        <div class="col">
            <div class="orders">
                <?php if ($_SESSION['name']): ?>
                    <table class="table-striped">
                        <thead>
                            <tr>
                                <th>Город отправки</th>
                                <th>Город прибытия</th>
                                <th>Класс</th>
                                <th>Количество</th>
                            </tr>
                        </thead>
                        <tbody>
                            <?php foreach ($orders as $order): ?>
                                <tr>
```

```
 <?= $order['fromCity'] ?></td>   <?= $order['toCity'] ?></td>  <?= $order['class'] ?></td>  <?= $order['count'] ?></td> </tr> <?php endforeach; ?> </tbody> </table> <? else: ?> <p style="text-align: center">Для просмотра заказов необходимо авторизоваться</p> <? endif; ?> </div> </div> </div> </div>  <?php include_once '../templates/footer.php'; ?> | | | |
```

6) auth.php

```

<?php
session_start();

include_once 'Database.php';

$data = $_POST;

$email = $data['email'];
$password = $data['password'];
$result = [];
header('Content-Type: application/json');

if (!$email || !$password) {
    $result['success'] = false;
    echo json_encode([
        'success' => false,
        'message' => 'Заполнены не все обязательные поля',
    ]);
    return;
}

$db = new Database();
if (!$userData = $db->findForAuth($email, md5($password))) {
    echo json_encode([
        'success' => false,
        'message' => 'Указаны неверные данные',
    ]);
}

```

```

        return;
    }
    list($_SESSION['id'], $_SESSION['name']) = $userData;
    echo json_encode([
        'success' => true,
        'message' => 'Авторизация успешна',
    ]);
    return;
}
7) book.php
<?php
include_once 'Database.php';
session_start();
header('Content-Type: application/json');

$data = $_POST;

if (!$count = $data['count']) {
    $count = 1;
}

if (!$SESSION['name']) {
    echo json_encode([
        'success' => false,
        'message' => 'Для бронирования билетов необходимо авторизоваться',
    ]);
    return;
}

if (!$id = $data['id']) {
    echo json_encode([
        'success' => false,
        'message' => 'Отсутствует идентификатор билета'
    ]);
    return;
}

$db = new Database();
if (!$db->bookTickets($id, $count)) {
    echo json_encode([
        'success' => false,
        'message' => 'Не удалось забронировать'
    ]);
    return;
}

echo json_encode([
    'success' => true,
    'message' => 'Бронирование успешно'
]);
return;
}
8) cities.php

```

```

<?php

require_once 'Database.php';

$db = new Database();
return $db->getCities();
    9) Classes.php
<?php

require_once 'Database.php';

$db = new Database();
return $db->getClasses();
    10) reg.php
<?php
session_start();
unset($_SESSION['name']);

header('Location: http://airbus/');
die();
    11) register.php
<?php

include_once 'Database.php';

$data = $_POST;

$email = $data['email'];
$password = $data['password'];

$result = [
    'success' => true,
    'message' => 'Вы успешно зарегистрированы'
];

if (!$email || !$password) {
    $result['success'] = false;
    $result['message'] = 'Заполнены не все обязательные поля';
    echo json_encode($result);
    return;
}

$db = new Database();
try {
    $db->register($email, $password);
} catch (Exception $e) {
    $result['success'] = false;
    $result['message'] = $e->getMessage();
}
echo json_encode($result);
    12) search.php
<?php

```

```

include_once 'Database.php';
header('Content-Type: application/json');
$data = $_POST;

$db = new Database();
$result = $db->findTickets($data);
echo json_encode($result);
13) Database.php
<?php

class Database {

    private $connection;

    public function __construct() {
        if (!$this->connection = mysqli_connect(
            'localhost',
            'mysql',
            'mysql'
        )) {
            die('Отсутствует подключение к базе данных');
        }
        if (!$this->connection->select_db('airbus')) {
            die('Отсутствует база данных airbus');
        }
    }

    public function register(string $email, string $password):
void {
        $password = md5($password);
        $query = sprintf("INSERT INTO users (email, password)
VALUES ('%s', '%s')", $email, $password);
        if (!$result = $this->connection->query($query)) {
            throw new Exception('Не удалось зарегистрировать
пользователя');
        }
        return;
    }

    public function findForAuth(string $email, string $password):
?array {
        $result = $this->connection->query(sprintf(
            "SELECT id, email, password FROM users WHERE email =
'%s' AND password = '%s'",
            $email,
            $password
        ));
        return mysqli_fetch_row($result);
    }

    public function findTickets(array $data): array {

```



```

        $where = [];
        $dateFrom = (new DateTime($data['dateFrom']))->format('Y-
m-d');
        $dateTo = (new DateTime($data['dateTo']))->format('Y-m-
d');
        $where[] = sprintf("(t.date = '%s' OR t.date = '%s')",
        $dateFrom, $dateTo);
        $where[] = sprintf('t.count >= %s', (int)$data['count']);
        $where[] = sprintf('t.city_from_id = %s', $data['from']);
        $where[] = sprintf('t.city_to_id = %s', $data['to']);
        if ($class = $data['class']) {
            $where[] = sprintf('t.class_id = %s', (int)$class);
        }

        $query = sprintf(
            "SELECT t.id, t.date, fc.name as fromCity, tc.name as
toCity, t.price, t.count, c.name as class
            FROM tickets t
            INNER JOIN city fc on fc.id = t.city_from_id
            inner join city tc on tc.id = t.city_to_id
            INNER JOIN class c on c.id = t.class_id
            WHERE %s",
            implode(' AND ', $where)
        );
        $result = $this->connection->query($query);
        return (array)mysqli_fetch_all($result, MYSQLI_ASSOC);
    }

    public function bookTickets(int $ticketId, int $count): bool
    {
        $result = mysqli_fetch_row($this->connection-
>query(sprintf(
            "SELECT COUNT(*) FROM orders WHERE user_id = %s AND
ticket_id = %s",
            $_SESSION['id'],
            $ticketId
        )));
        try {
            $transactionStarted = $this->connection-
>query("BEGIN");
            if ($result[0]) {
                $transactionStarted && $this-
>updateTicketCount($ticketId, $count);
            } else {
                $transactionStarted && $this-
>insertOrderRow($ticketId, $count);
            }
            $this->decreaseTicketsCount($ticketId, $count);
            $this->connection->query("COMMIT");
            return true;
        } catch (Exception $e) {
            $this->connection->query("ROLLBACK");
        }
    }

```

```

        return false;
    }
}

public function getClasses(): array {
    $result = $this->connection->query("SELECT * FROM class");
    return (array)mysqli_fetch_all($result, MYSQLI_ASSOC);
}

public function getCities(): array {
    $result = $this->connection->query("SELECT * FROM city");
    return (array)mysqli_fetch_all($result, MYSQLI_ASSOC);
}

public function getUserOrders(): array {
    $result = $this->connection->query(sprintf(
        "SELECT fc.name as fromCity, tc.name as toCity, c.name
as class, o.count
        FROM orders o
        INNER JOIN tickets t on t.id = o.ticket_id
        INNER JOIN city fc on fc.id = t.city_from_id
        INNER JOIN city tc on tc.id = t.city_to_id
        INNER JOIN class c on c.id = t.class_id
        WHERE user_id = %s",
        $_SESSION['id']
    ));
    return (array)mysqli_fetch_all($result, MYSQLI_ASSOC);
}

private function updateTicketCount(int $ticketId, int $count)
{
    if (!$this->connection->query(sprintf(
        "UPDATE orders SET `count` = `count` + %s WHERE user_id
= %s AND ticket_id = %s",
        $count,
        $_SESSION['id'],
        $ticketId
    ))) {
        throw new Exception('Невозможно увеличить количество
билетов');
    }
}

private function insertOrderRow(int $ticketId, int $count):
void {
    if (!$this->connection->query(sprintf(
        "INSERT INTO orders (user_id, ticket_id, `count`)
VALUES (%s, %s, %s)",
        $_SESSION['id'],
        (int)$ticketId,
        (int)$count
    ))) {

```

```

        throw new Exception('Не удалось забронировать
билеты');
    }
}

private function decreaseTicketsCount(int $ticketId, int
$count): void {
    if (!$this->connection->query(sprintf(
        "UPDATE tickets SET `count` = `count` - %s WHERE id =
%s",
        $count,
        $ticketId
    ))) {
        throw new Exception('Невозможно уменьшить количество
билетов');
    }
}
}

```

14. Admin.php

```

<?php
include_once './templates/header.php';
include_once './app/Database.php';
$db = new Database();
$tickets = $db->getTickets();
$cities = $db->getCities();
$classes = $db->getClasses();
?>

<div class="container-fluid">
    <div class="row">
        <div class="orders">
            <?php if ($_SESSION['name'] ==
'admin@mail.ru'): ?>
                <div class="row">
                    <div class="col ml-2 mt-2">
                        <button class="btn btn-success"
data-toggle="modal" data-target="#createModal">Создать
                        </button>
                    </div>
                </div>
            </div>
            <table class="table-striped">
                <thead>
                    <tr>
                        <th>Город отправки</th>
                        <th>Город прибытия</th>
                        <th>Дата</th>
                        <th>Класс</th>
                        <th>Цена</th>
                        <th>Количество</th>
                        <th>Действия</th>
                    </tr>
                </thead>
            </table>
        </div>
    </div>
</div>

```

```

        <tbody>
        <?php foreach ($tickets as $ticket):
?>
            <tr data-id="<?= $ticket['id']
?>">
                <td> <?= $ticket['fromCity']
?></td>
                <td> <?= $ticket['toCity']
?></td>
                <td> <?= $ticket['date']
?></td>
                <td> <?= $ticket['class']
?></td>
                <td> <?= $ticket['price']
?></td>
                <td> <?= $ticket['count']
?></td>
                <td>
                    <button class="btn btn-
danger"
onclick="removeTicket(<?= $ticket['id'] ?>)">
                        Удалить
                    </button>
                </td>
            </tr>
        <? endforeach; ?>
        </tbody>
    </table>
    <? else: ?>
        <p style="text-align: center">Для
редактирования билетов необходимо быть администратором</p>
    <? endif; ?>
</div>
</div>
</div>

    <div class="modal fade" id="createModal" tabindex="-1"
role="dialog" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <form action="" novalidate
id="ticketCreationForm" class="ticket-creation-form">
                    <div class="modal-header">
                        <h5 class="modal-title"
id="staticBackdropLabel">Создание билета</h5>
                        <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
                            <span
hidden="true">&times;</span>
                        </button>
                    </div>

```

```

        <div class="modal-body">
            <div class="form-row">
                <div class="col">
                    <div class="input-group mb-
3">
                        <div class="input-group-
prepend">
                            <label class="input-
group-text" for="from">Вылет</label>
                        </div>
                        <select class="custom-
select" id="from" required name="from">
                            <? foreach ($cities
as $city): ?>
                                <option
value="<?= $city['id'] ?>"><?= $city['name'] ?></option>
                                <? endforeach; ?>
                            </select>
                        </div>
                    </div>
                    <div class="col">
                        <div class="input-group mb-
3">
                            <div class="input-group-
prepend">
                                <label class="input-
group-text" for="to">Прибытие</label>
                            </div>
                            <select class="custom-
select" id="to" required name="to">
                                <? foreach ($cities
as $city): ?>
                                    <option
value="<?= $city['id'] ?>"><?= $city['name'] ?></option>
                                    <? endforeach; ?>
                                </select>
                            </div>
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="col">
                            <input
class="form-control" type="number"
placeholder="Количество пассажиров"
required
min="1"
name="count">
                            <div
class="invalid-
feedback">
                                Пожалуйста, укажите
количество пассажиров
                            </div>
                        </div>
                    </div>

```

```

3">
    <div class="col">
        <div class="input-group mb-
            <div class="input-group-
prepend">
                <label class="input-
group-text" for="inputGroupSelect01">Класс</label>
            </div>
                <select class="custom-
select" id="inputGroupSelect01" required name="class">
                    <option value="1"
selected>Бюджет</option>
                    <option
value="2">Комфорт</option>
                    <option
value="3">Бизнес</option>
                </select>
            </div>
        </div>
    </div>
    <div class="form-row">
        <div class="col">
            <div class="input-group
date" id="datetimepicker1" data-target-input="nearest">
                <div class="input-group-
prepend">
                    <span class="input-
group-text" id="basic-addon1">Туда</span>
                </div>
                <input type="text"
class="form-control datetimepicker-input"
data-
target="#datetimepicker1"
aria-
describedby="basic-addon1" required name="dateTo">
                    <div class="input-group-
append" data-target="#datetimepicker1"
data-
toggle="datetimepicker">
                        <div class="input-
group-text"><i class="fa fa-calendar"></i></div>
                    </div>
                    <div class="invalid-
feedback">
                        Пожалуйста, укажите
дату вылета
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col">
        <input class="form-control"
placeholder="Цена" required

```

```

min="1"
name="price">


54


```

```

header('Content-Type: application/json');

$result = [
    'success' => true,
    'message' => 'Билет успешно удален',
];

if ($_SESSION['name'] !== 'admin@mail.ru') {
    echo json_encode([
        'success' => false,
        'message' => 'Удаление доступно только
администратору',
    ]);
    return;
}

if (!$id = $_POST['id']) {
    echo json_encode([
        'success' => false,
        'message' => 'Укажите ID билета'
    ]);
    return;
}

$db = new Database();
try {
    $db->removeTicket($id);
} catch (Exception $e) {
    $result['success'] = false;
    $result['message'] = $e->getMessage();
}
echo json_encode($result);

```