МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский ядерный университет «МИФИ»



Институт   
интеллектуальных кибернетических систем

Кафедра кибернетики (№ 22)

Направление подготовки 09.03.04 Программная инженерия

КУРСОВАЯ РАБОТА

«Разработка прототипа web-приложения»

по курсу

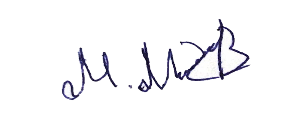
«Современные средства программирования»

Тема: «Билеты ж/д касс»

Преподаватель: Садчиков С.М

Студент: Махмудов М.А

Группа: Б18-504

****

Москва 2021

Содержание

[1. Описание предметной области 3](#_Toc61963863)

[2. Проектирование базы данных 4](#_Toc61963864)

[2.1 Пользователи системы 4](#_Toc61963865)

[2.2 События, изменяющие состояние базы данных 4](#_Toc61963866)

[2.3 Физическое проектирование 4](#_Toc61963867)

[2.4 Создание таблиц 5](#_Toc61963868)

[3. Работа с приложением 7](#_Toc61963869)

[3.1 Структура проекта 7](#_Toc61963870)

[3.2 Процесс работы 8](#_Toc61963871)

[3.3 Работа кассиров 13](#_Toc61963872)

# Описание предметной области

Разрабатываемая web-приложения предназначена для автоматизации работы касс железнодорожных вокзалов. И содержит общую информацию о направлениях поездов, станциях, поездах, вагонах и т.д. Его целью является продажа билетов, продажа билетов исходя из станции назначения. Таким образом приложения облегчает работу кассира, позволяя мгновенно продать билеты внеся данных о пассажире. Приложения предназначена для поездов, которые двигаются только между двумя станциями без остановок только одной кампании и только купейного типа. (также все поезда одинаковые и имеют по 4 вагона) Цены на билетах зависят от пунктов назначений. Приложением будут пользоваться кассиры, которые продают билеты пассажирам. Также пассажиры могут посмотреть расписание рейсов без регистрации на приложения. Пассажирам для покупки билета нужно:

1) обратится к кассиру железнодорожных вокзалов и сообщить: станцию отправления, станцию прибытия и дату отправления;

2)иметь при себе паспорт; (приложения предназначена только для лиц, у которых есть паспорт т.е. для детей билет не продается). Условия возврата билета или обмена также не предусмотрены.

Кассирам для получения доступа на приложения дается отдельные логины и пароли.

После того как пассажир сообщить кассиру станции и дату отправки, кассиру нужно выбрать из списков станцию отправления, станцию прибытие и дату отправки. После этого открывается окно где будут показаны все рейсы касающихся этих станций и дату (в нем будут показаны название станций, дата отправления, время отправления, дата прибытия, время прибытия и цена). После пассажиру необходима выбрать номер вагона. Дальше открывается другое окно, и кассир выбирает и бронирует место который выбрал пассажир. В процессе бронирования кассиру необходимо заполнить поля: ФИО пассажира, номер паспорта. Дальше открывается окно оплаты. (Если одновременно несколько кассиров смотрят один и тот же билет, то получит билет тот кассир, который продал первым). Оплатить покупку можно только наличными деньгами. После оплаты билет сохраняется на базе и на выходе пассажир получит мини выписку (в котором будут указаны данные пассажира, цена билета, станция отправления, станция прибытия, дата и время отправления, дата и время прибытия, номер вагона и место).

## 2. Проектирование базы данных

В базе данных хранятся информация о кассирах, о пассажирах, о поездах и о рейсах на которых продаются билеты.

## 2.1 Пользователи системы

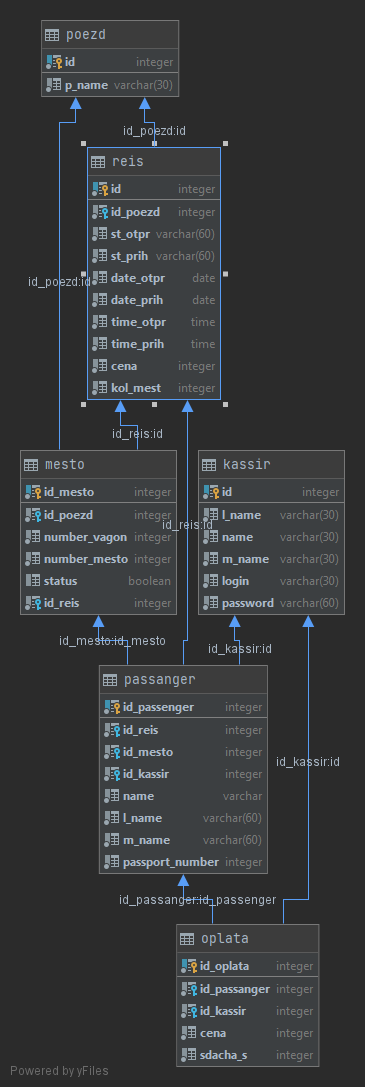
Данная система ориентирована на пользователей, которые продают билеты через веб приложения, то есть для кассиров.

## 2.2 События, изменяющие состояние базы данных

Состояние базы данных может меняться при регистрации новых пассажиров, при добавлении разработчиками новых кассиров, новых рейсов или новых поездов.

## 2.3 Физическое проектирование

Всвязи с проведённым анализом предметной области было проработано физическая схема базы данных:



## 2.4 Создание таблиц

**Таблица kassir**

CREATE TABLE kassir  
(  
 id SERIAL PRIMARY KEY,  
 l\_name varchar(30) NOT NULL,  
 name varchar(30) NOT NULL,  
 m\_name varchar(30) NOT NULL,  
 login varchar(30) NOT NULL,  
 password varchar(60) NOT NULL  
);

**Таблица passanger**

CREATE TABLE passanger  
(  
 id\_passenger SERIAL PRIMARY KEY,  
 id\_reis int NOT NULL,  
 id\_mesto int NOT NULL,  
 id\_kassir int NOT NULL,  
 name varchar NOT NULL,  
 l\_name varchar(60) NOT NULL,  
 m\_name varchar(60) NOT NULL,  
 passport\_number int NOT NULL,  
 FOREIGN KEY (id\_reis) references reis(id),  
 FOREIGN KEY (id\_mesto) references mesto(id\_mesto),  
 FOREIGN KEY (id\_kassir) references kassir(id)  
);

**Таблица poezd**

CREATE TABLE poezd  
(  
 id SERIAL PRIMARY KEY,  
 p\_name varchar(30) NOT NULL  
);

**Таблица oplata**

CREATE TABLE oplata  
(  
 id\_oplata SERIAL PRIMARY KEY,  
 id\_passanger int NOT NULL,  
 id\_kassir int NOT NULL,  
 cena int NOT NULL,  
 sdacha\_s int NOT NULL,  
 FOREIGN KEY (id\_passanger) references passanger(id\_passenger),  
 FOREIGN KEY (id\_kassir) references kassir(id)  
);

**Таблица reis**

CREATE TABLE reis  
(  
 id SERIAL PRIMARY KEY,  
 id\_poezd int NOT NULL,  
 st\_otpr varchar(60) NOT NULL,  
 st\_prih varchar(60) NOT NULL,  
 date\_otpr date NOT NULL,  
 date\_prih date NOT NULL,  
 time\_otpr time NOT NULL,  
 time\_prih time NOT NULL,  
 cena int NOT NULL,  
 kol\_mest int NOT NULL,  
 FOREIGN KEY (id\_poezd) references poezd(id)  
);

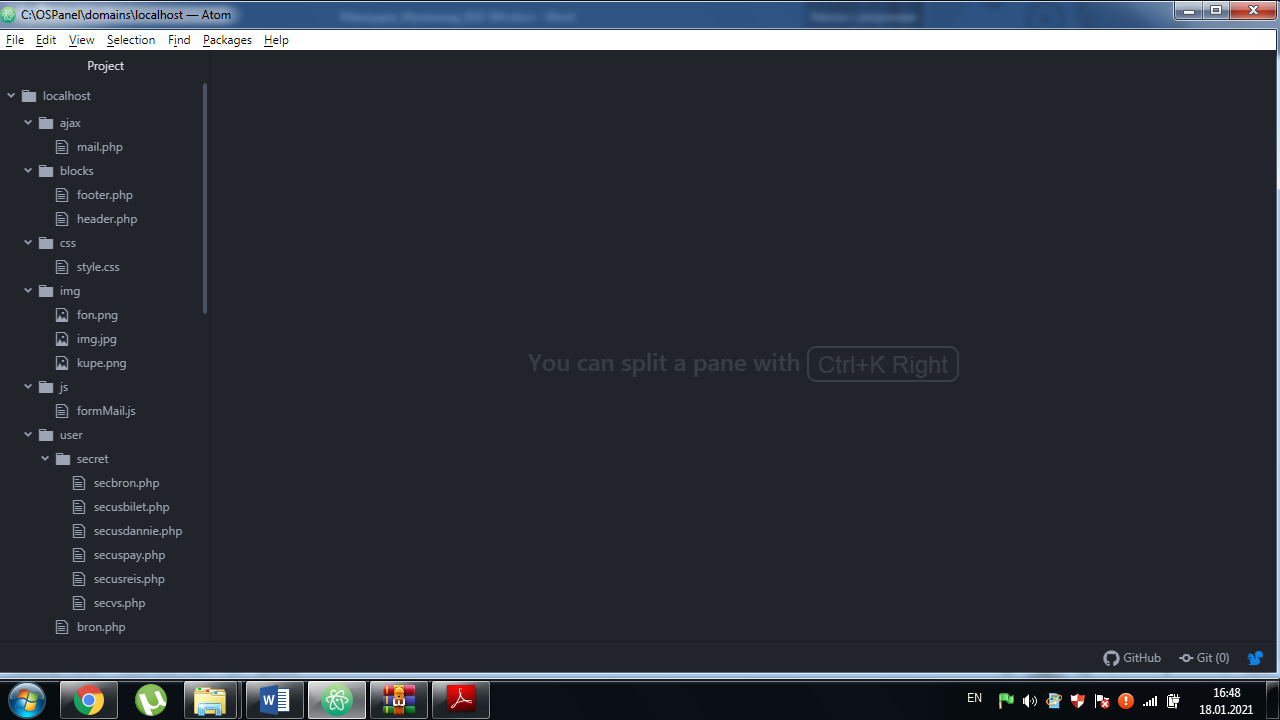
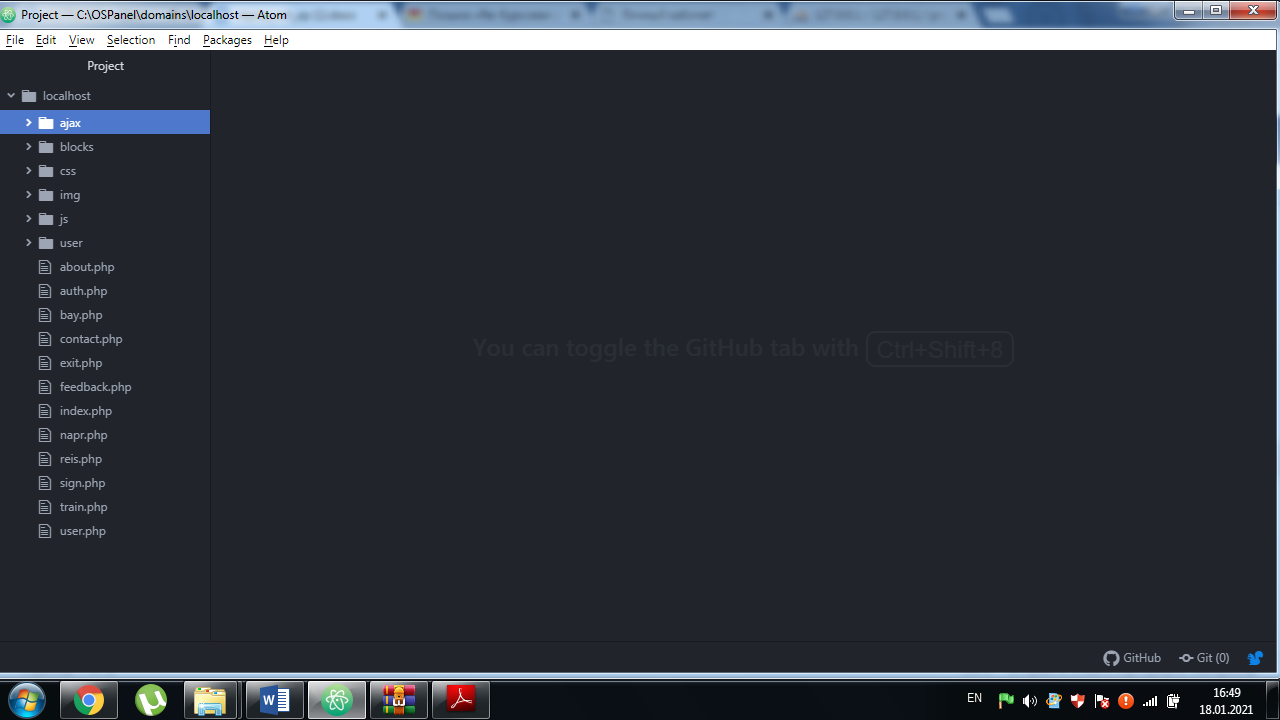
**Таблица mesto**

CREATE TABLE mesto  
(  
 id\_mesto SERIAL PRIMARY KEY,  
 id\_poezd int NOT NULL,  
 number\_vagon int NOT NULL,  
 number\_mesto int NOT NULL,  
 status boolean default true,  
 id\_reis int NOT NULL,  
 FOREIGN KEY (id\_poezd) references poezd(id),  
 FOREIGN KEY (id\_reis) references reis(id)  
);

В качестве данных было выбрано города Таджикистана (Худжанд, Душанбе, Куляб, Курган-Тюбе). А в качестве валюты было выбрано национальную таджикскую валюту-сомони.

## 3. Работа с приложением

## 3.1 Структура проекта

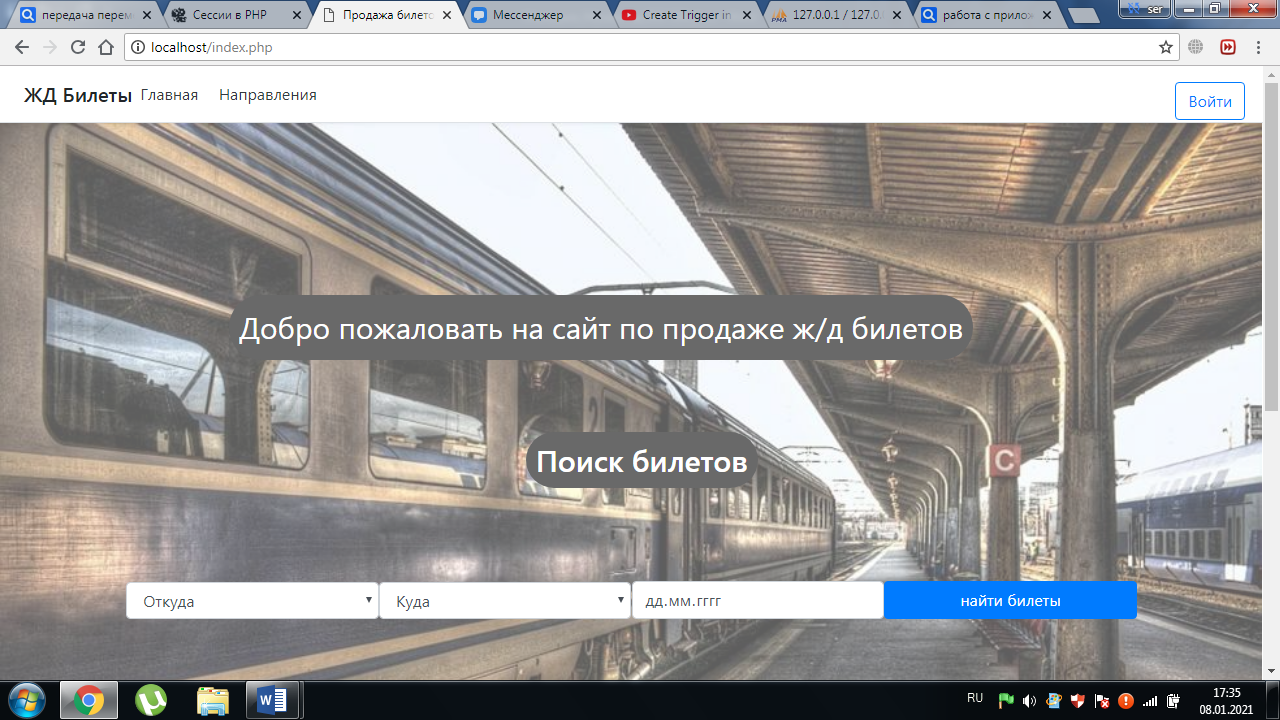
 

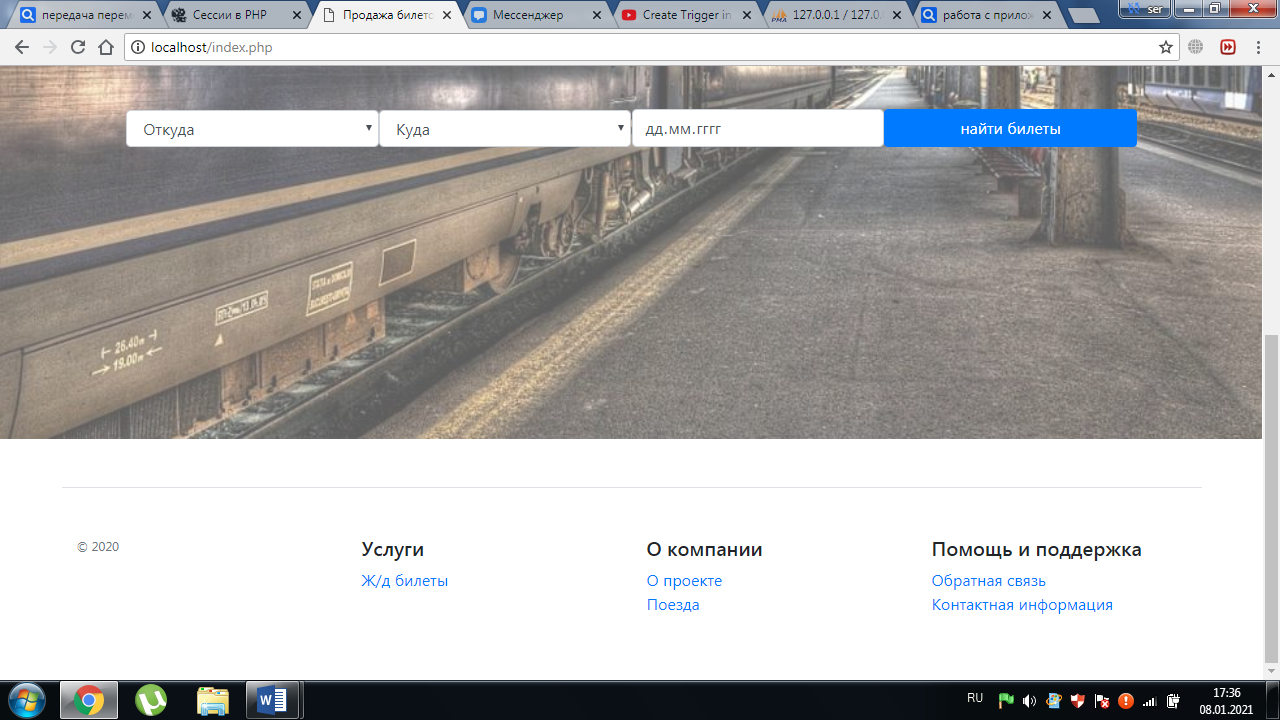
В директории /localhost/ находятся файлы с исходным кодом соответствующих страниц web-приложения

В папках ajax и js находятся файлы с исходным кодом, обеспечивающие работу AJAX-запроса в web-приложении. (Для обработки результатов, feedback-а)

## 3.2 Процесс работы

Приложением могут пользоваться два вида пользователей: кассиры (авторизованные пользователи) и обычные пользователи (неавторизованные пользователи). Только обычным пользователям не доступно покупка билетов, они могут только посмотреть на расписание рейсов. Также могут получать информацию о поездах компании и о контактах.

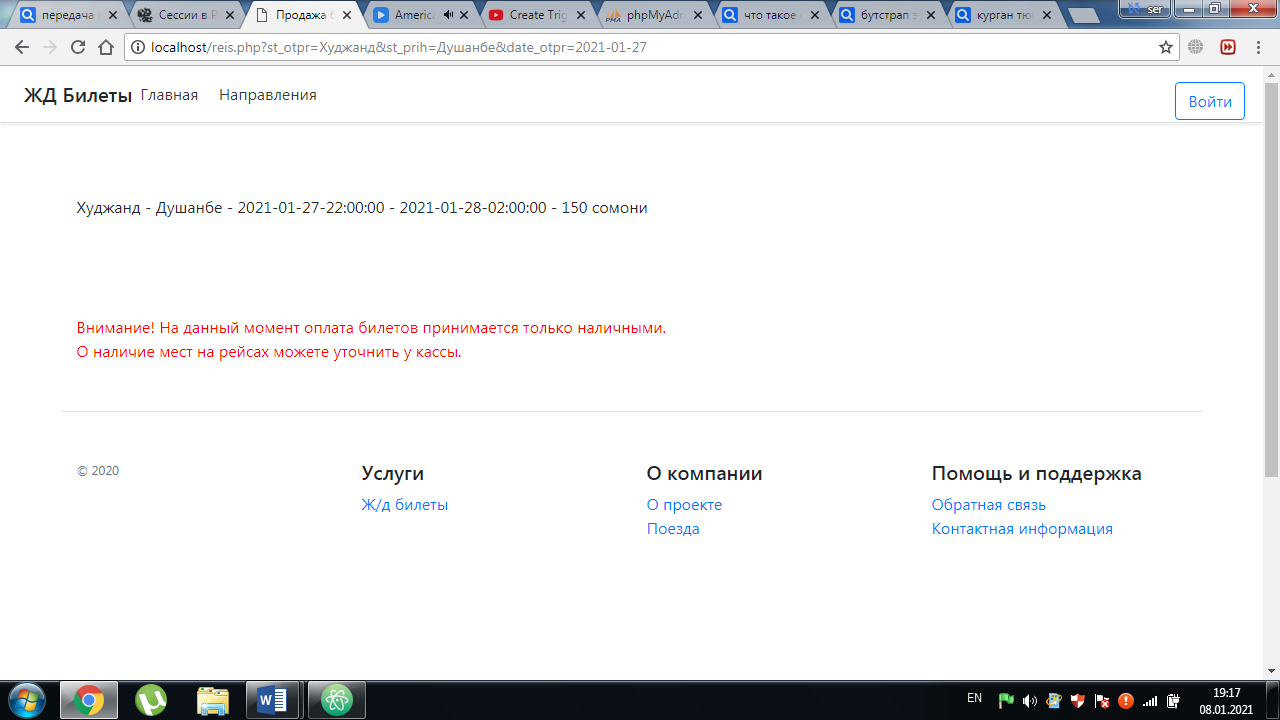




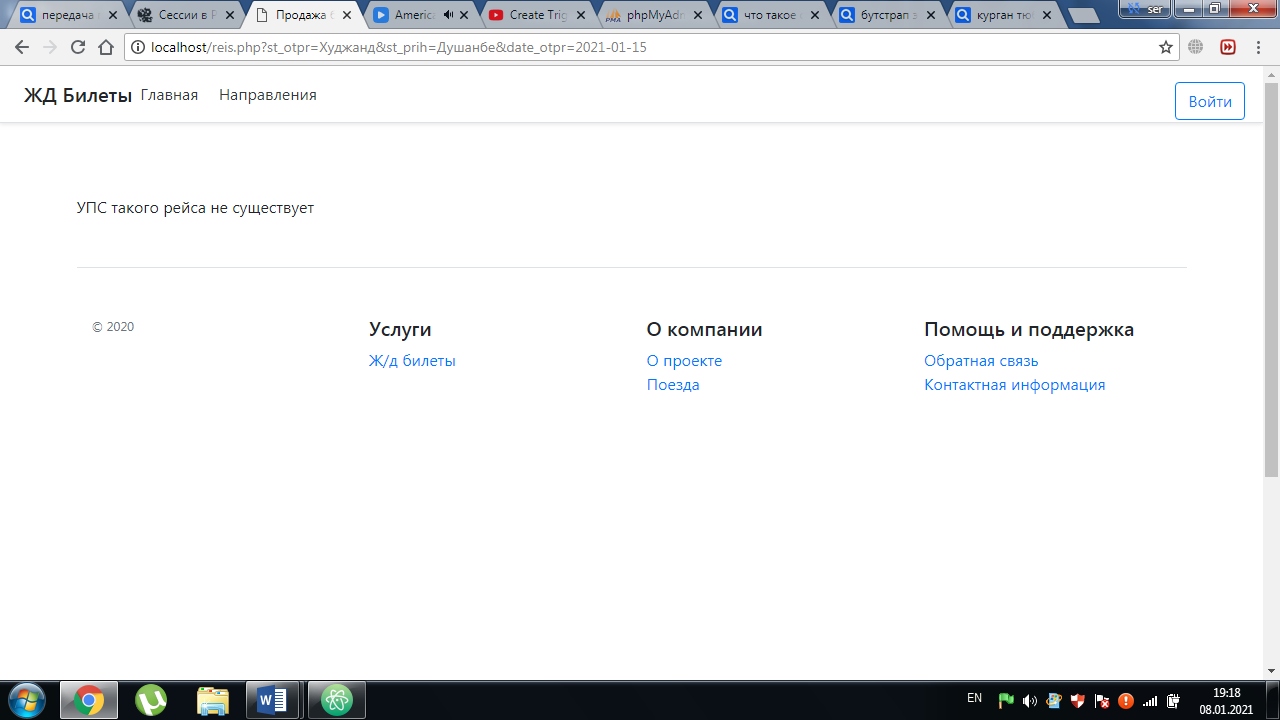
На этой картинке показано главная страница приложении.

Для добавления различных стилей в приложение было использовано набор инструментов Bootstrap.

Неавторизованные пользователи могут посмотреть список рейсов на определенную дату вводя в поля: откуда-станцию отправления, куда-станцию прибытия. Например посмотрим рейсы на 27.01.2021 из Худжанда в Душанбе:

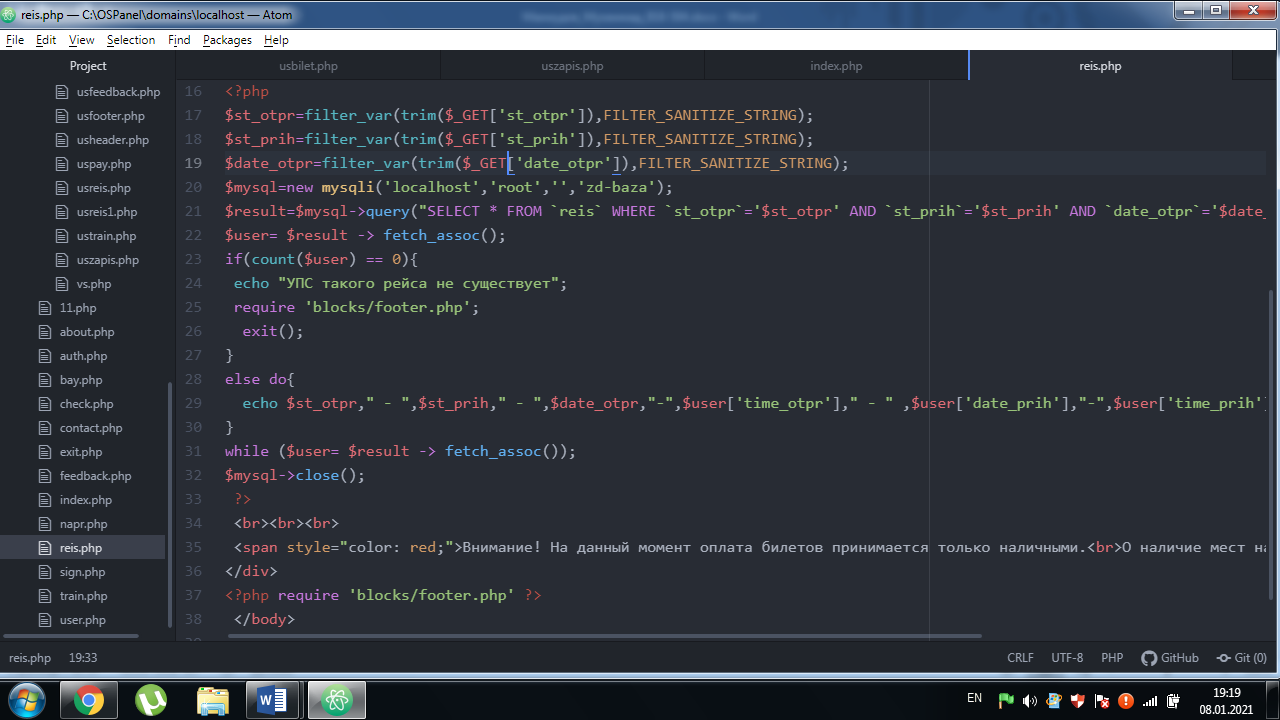


Как видно из картинки на эту дату планировано только один рейс и стоит оно 150 сомони. Давайте теперь посмотрим на другую дату, например, 15.01.2021:



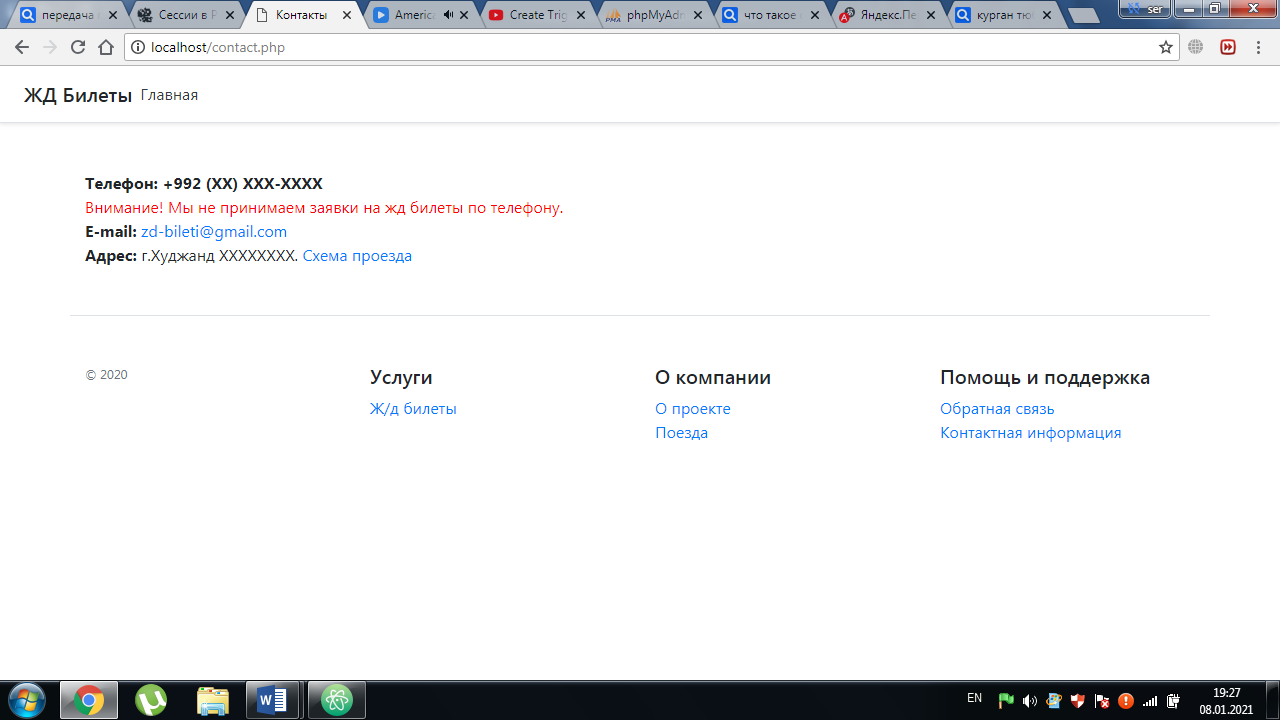
Если в базе данных нету такого рейса, то выводиться сообщения о том, что не существует такой рейс.

За проверку отвечает вот это часть кода:

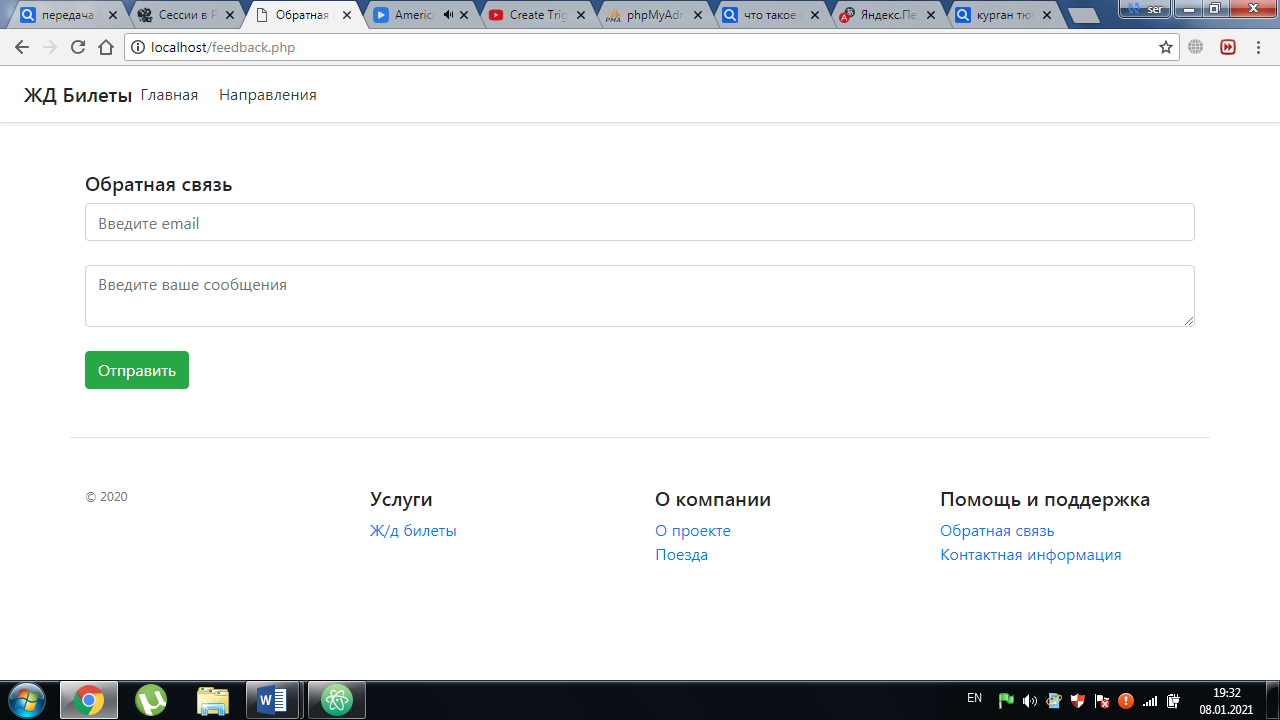


Из кода или из ссылки на страницу можно увидеть, что данные передаются между страницами методом GET так как это информация ни на что, не влияет и с точки зрения безопасности тоже никак не влияет на нашу систему (в остальных случаях используется метод POST).

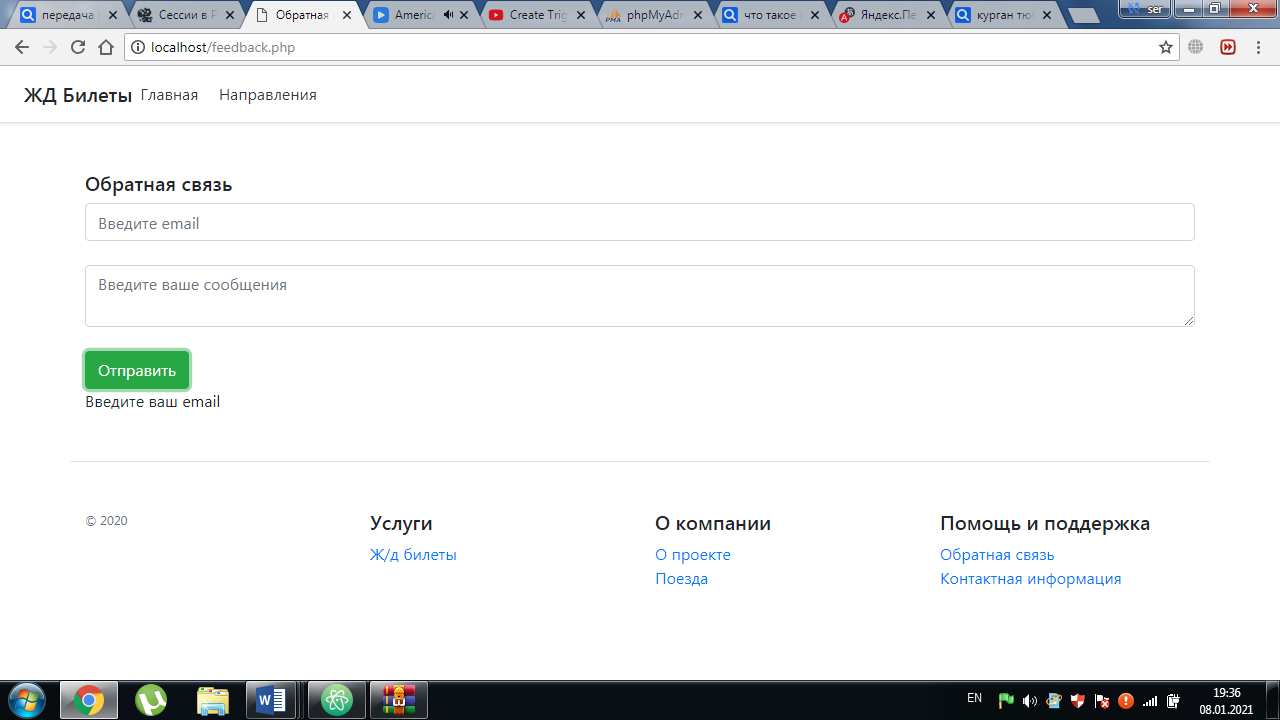
Также пользователи могут посмотреть на контакты и адресов касс нашей компании:

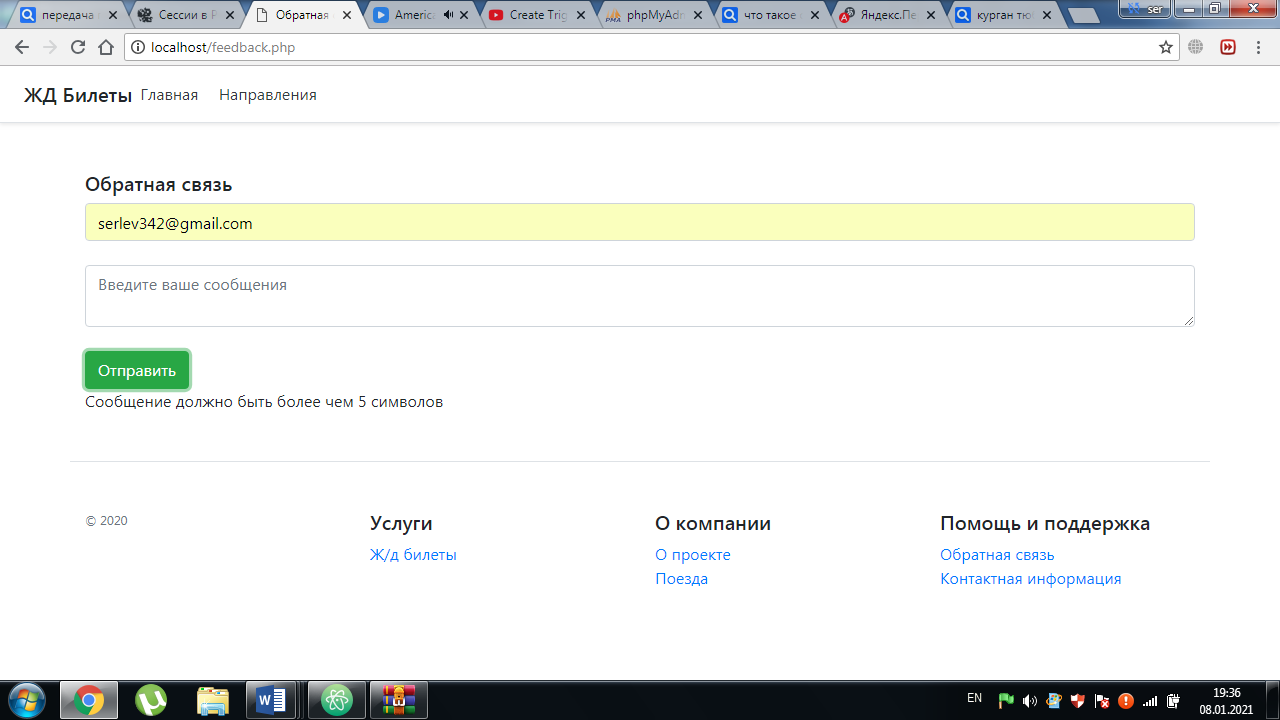


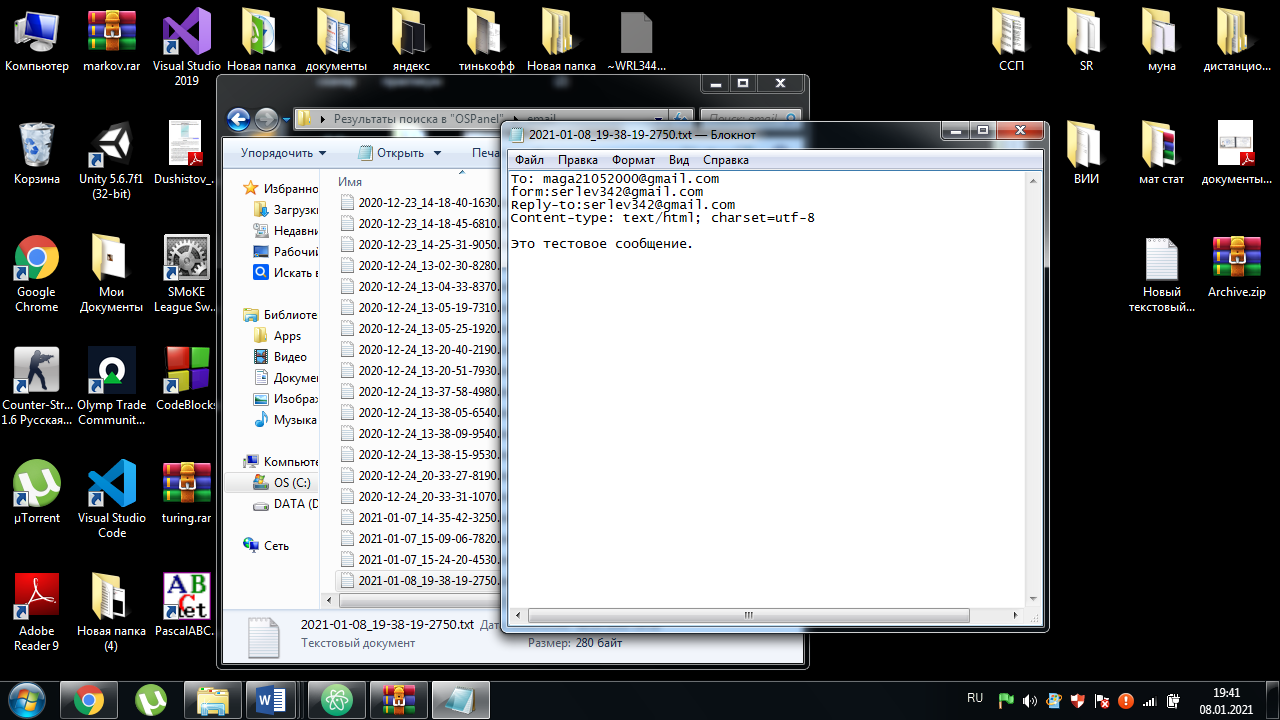
Если пользователям нужна помощь или у них есть какие-то идеи для улучшения сайта или же наоборот какие-то замечания, то они могут воспользоваться формой обратной связи:



В этой форме данные обрабатываются использую технологии Ajax:







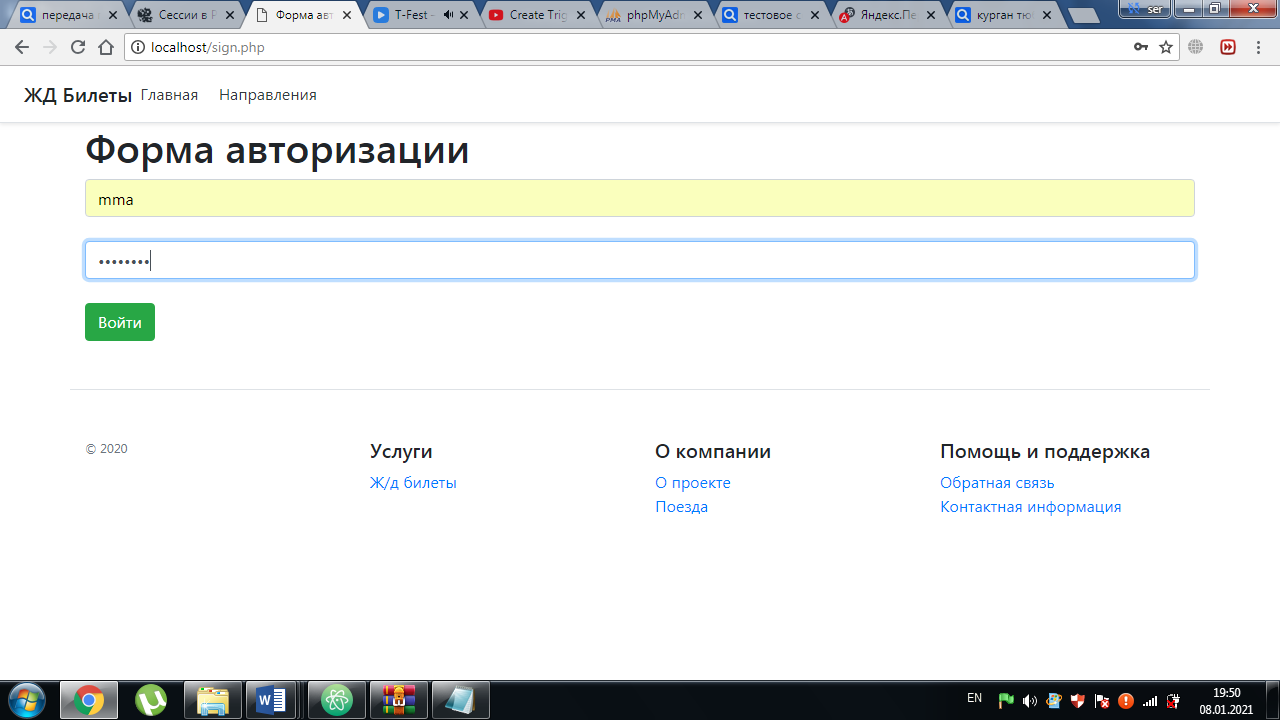
Также пользователи могут получить информацию о поездах и о компании нажимая соответствующие ссылки.

## 3.3 Работа кассиров

Кассирам после того как их примут на работу даются отдельные логины и пароли со стороны администрации.

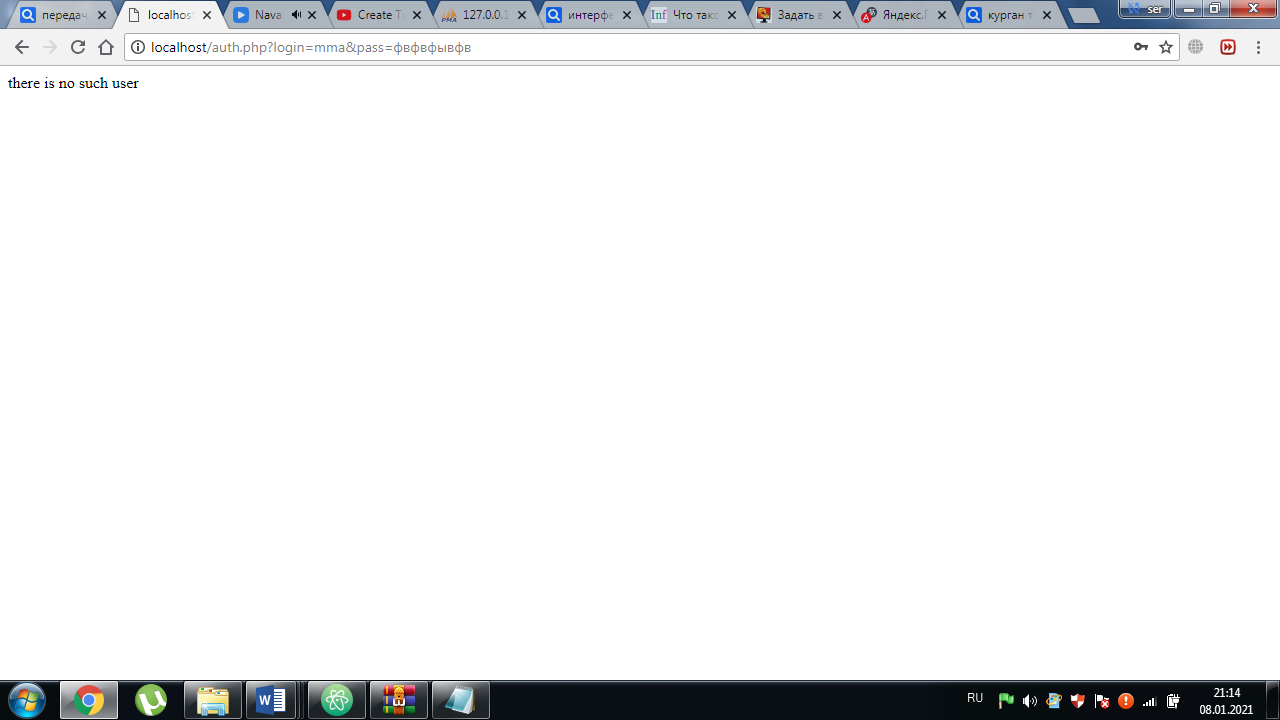


Для начала работы кассирам необходима войти в свои личные кабинеты вводя в поля логин и пароль:

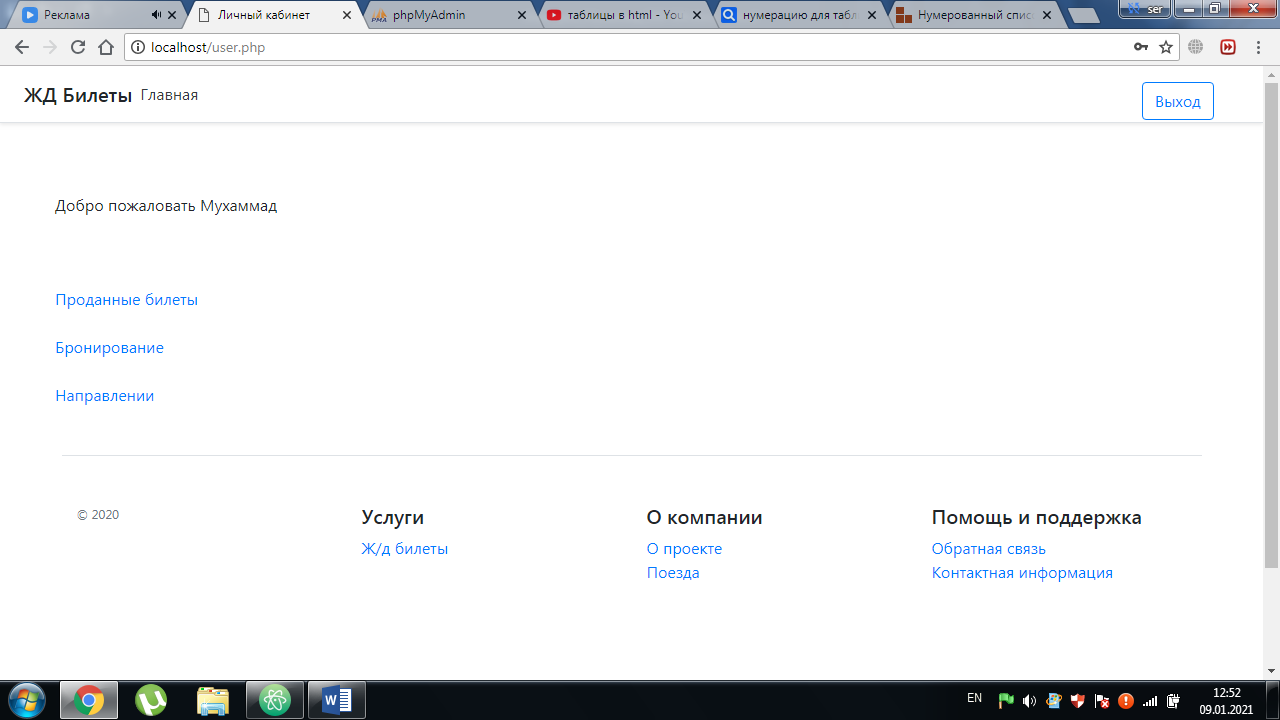


Тут, как и везде для интерфейса используются классы из bootstrap, например, для кнопки “Войти” использовался класс btn-success. В других местах решения по интерфейсу аналогичные.

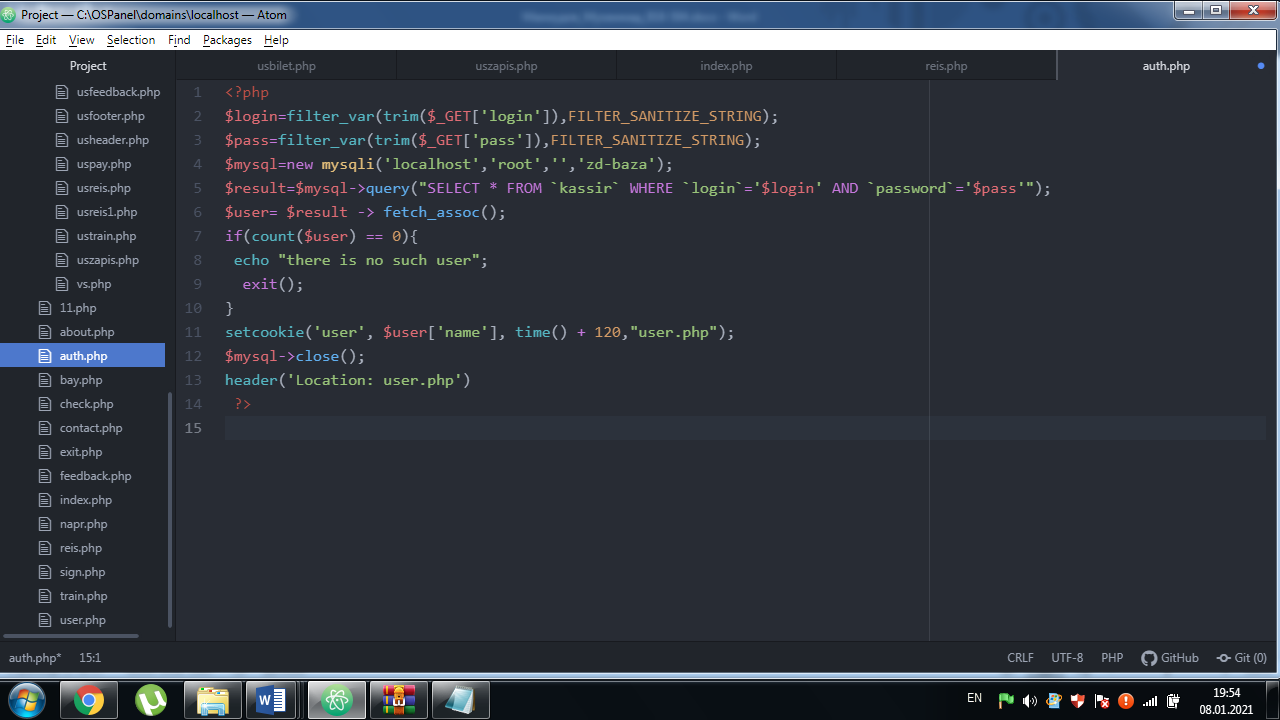
Когда вводимые данные неверные:



Так выглядит их рабочая страница:



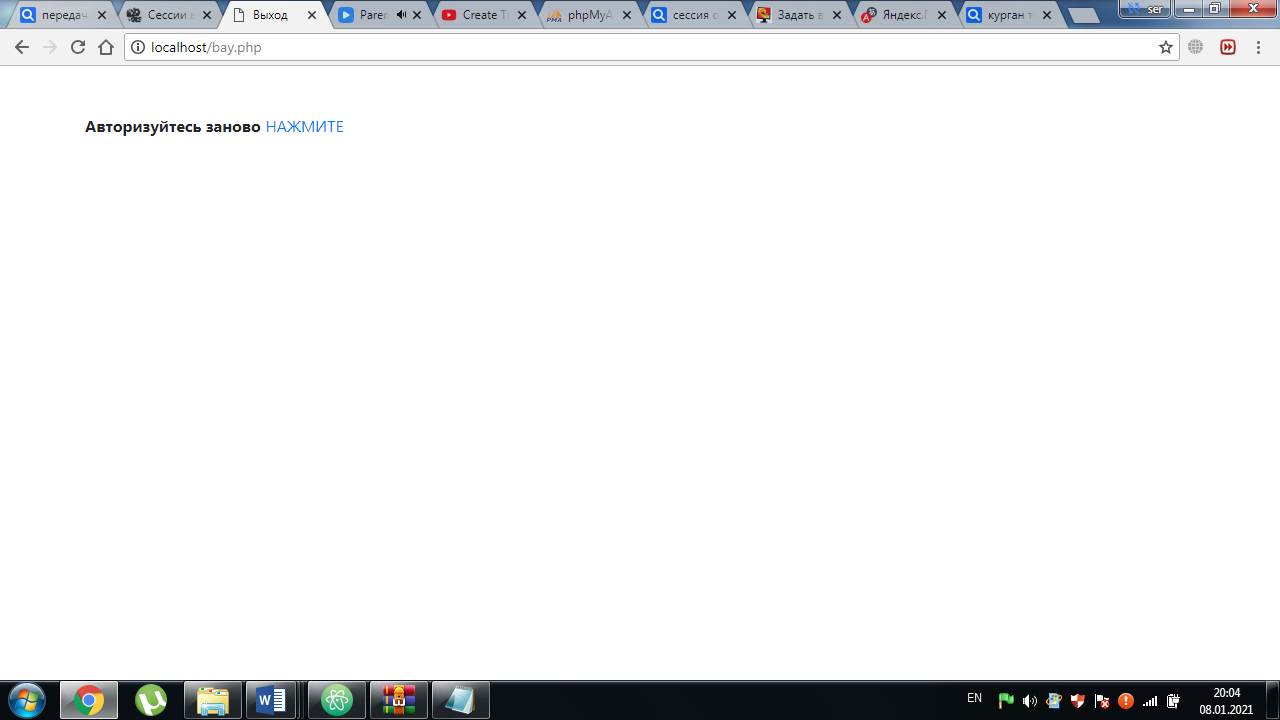
Согласна требованию нужно было организовать работу кассира 2 мин:



Это значение будет проверятся везде на страницах кассира:



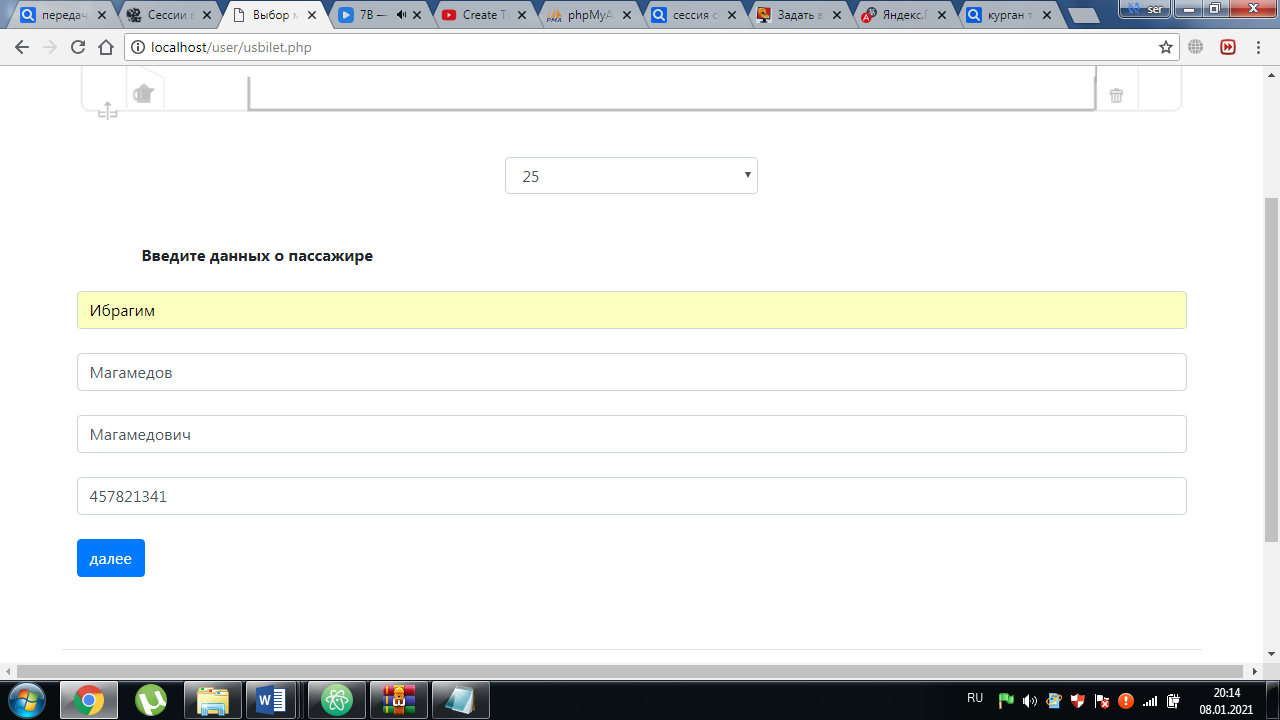
После истечения этого времени выводится вот такое сообщения:



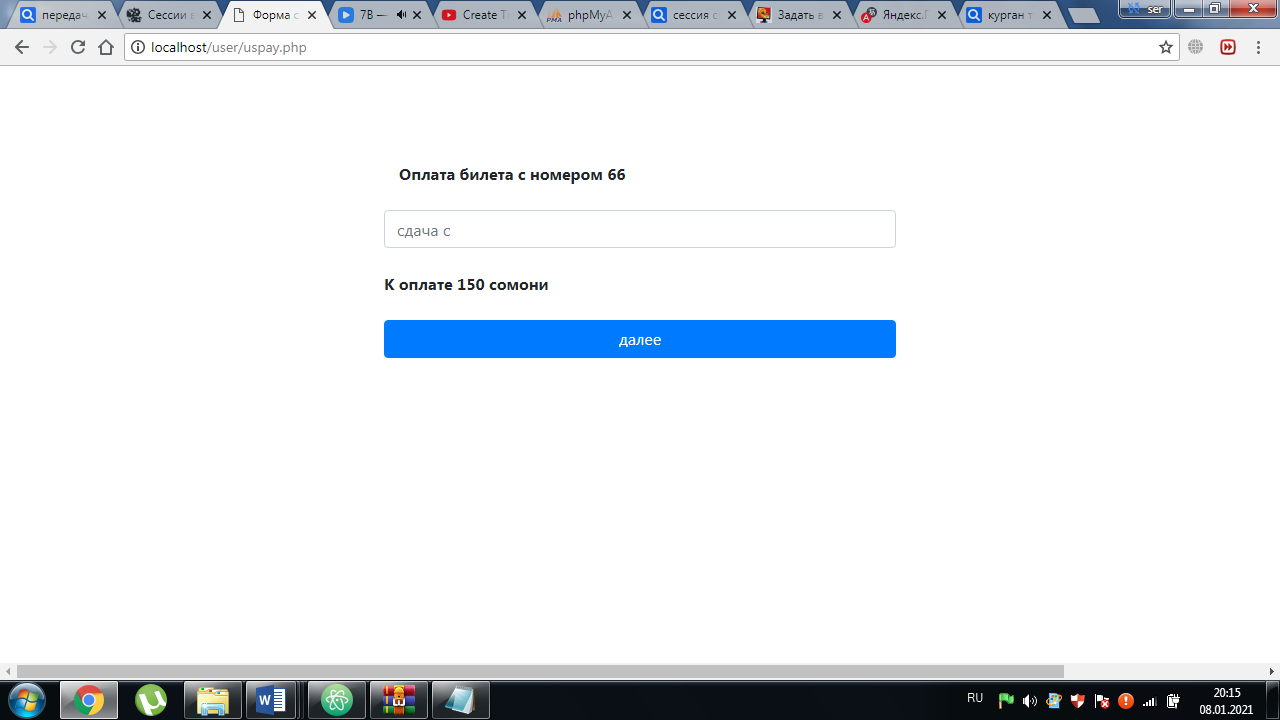
Далее кассир может приступит к работе и продавать билеты. Для этого он должен выбрать пункт бронирования из личного кабинета. И дальше продолжить, выбирая рейс, номер вагона. После открывается окно, в котором он должен выбрать свободное место и ввести данные пассажира (например, мы выбрали рейс худжад-душанбе на 27.01.21 22:00 и вагон номер 2):



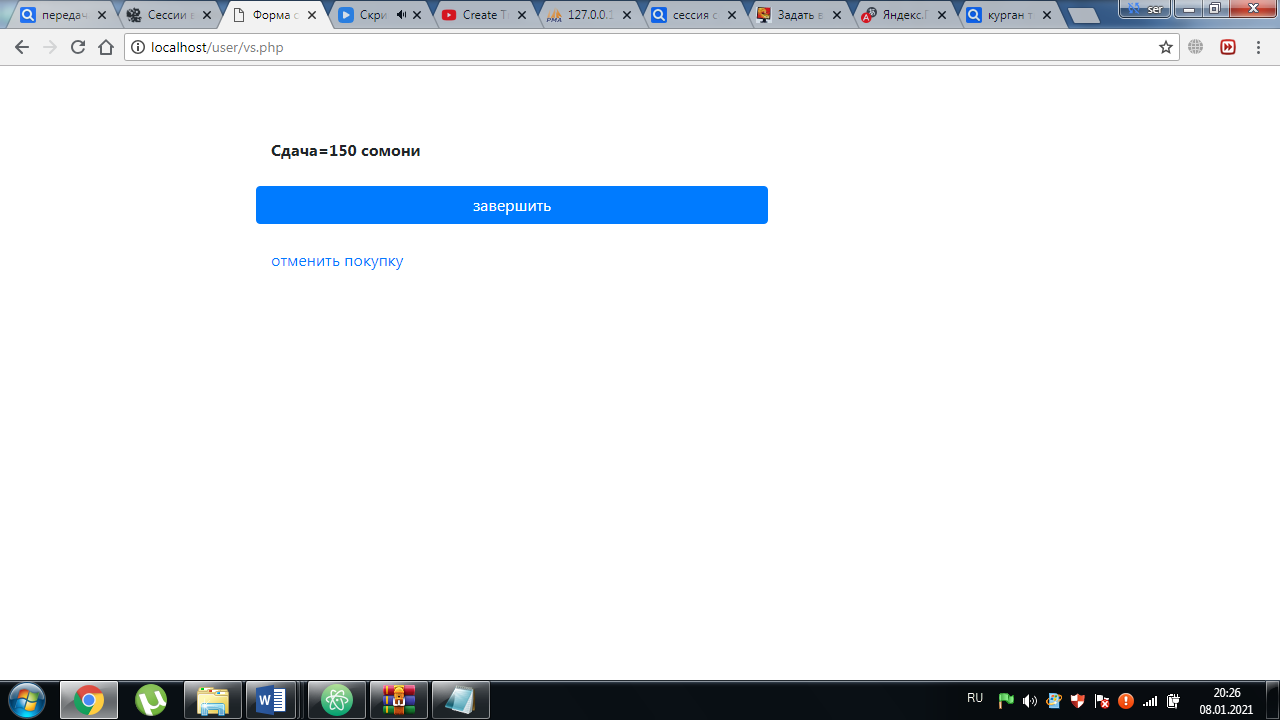
В появившейся окне кассир должен выбрать номер место (проданные места отображены серыми и их невозможно выбрать) и заполнит данные:



Дальше он должен указывать купюру с которой собирается дать сдачу пассажиру:

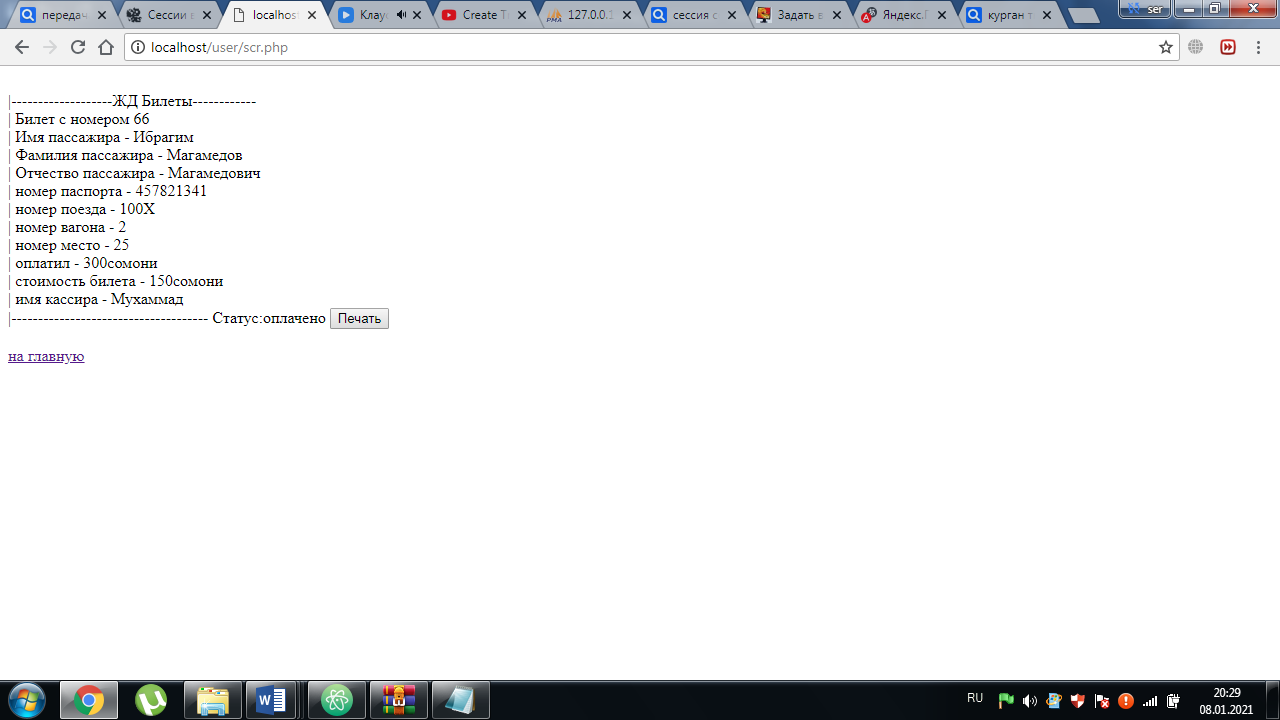


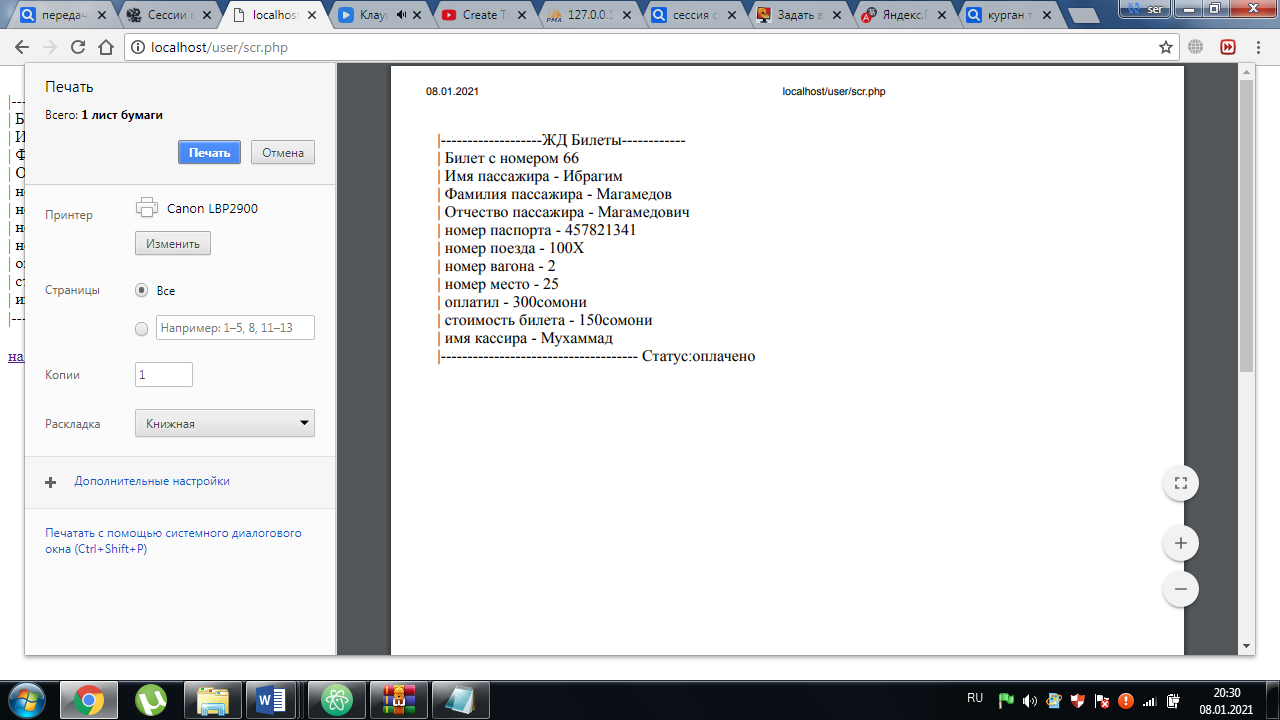
Например, 300.



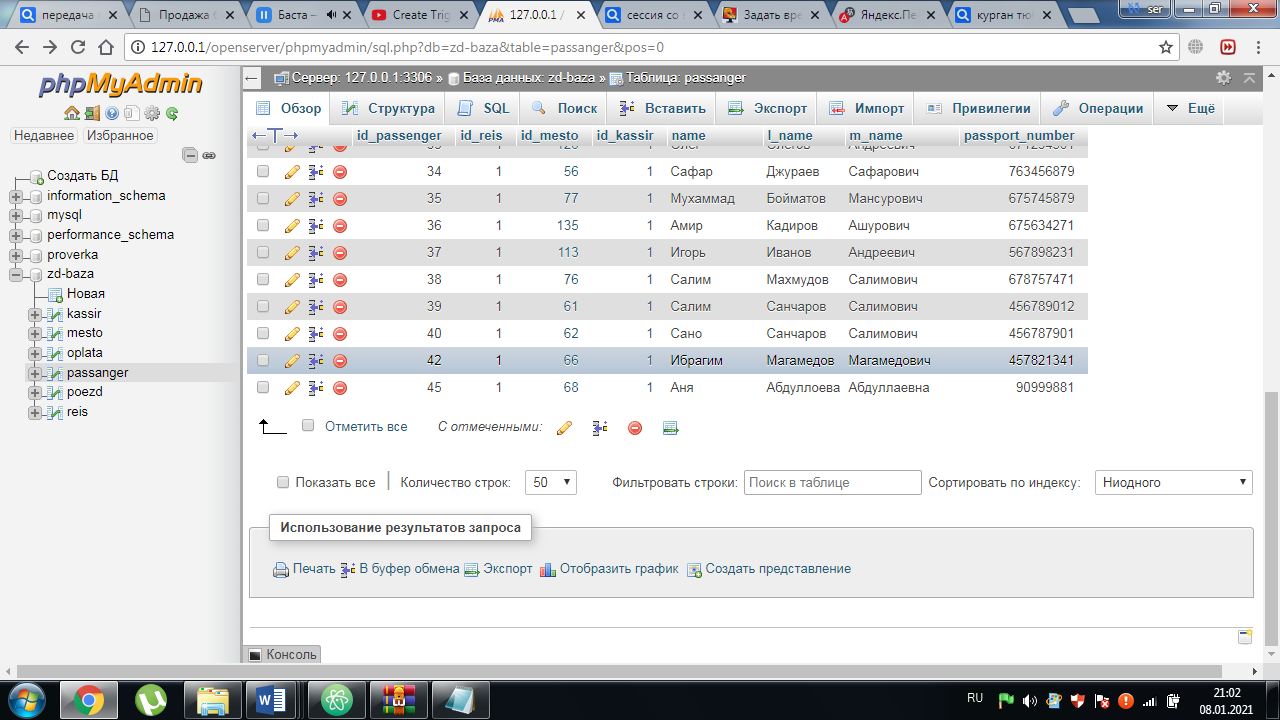
Пользователь также может отменить заказ.

После оплаты налычними кассиру кассир нажимает завершить и таким образом билет продается.После кассир напечатает мини выписку пассажиру нажимая на кнопку печать:





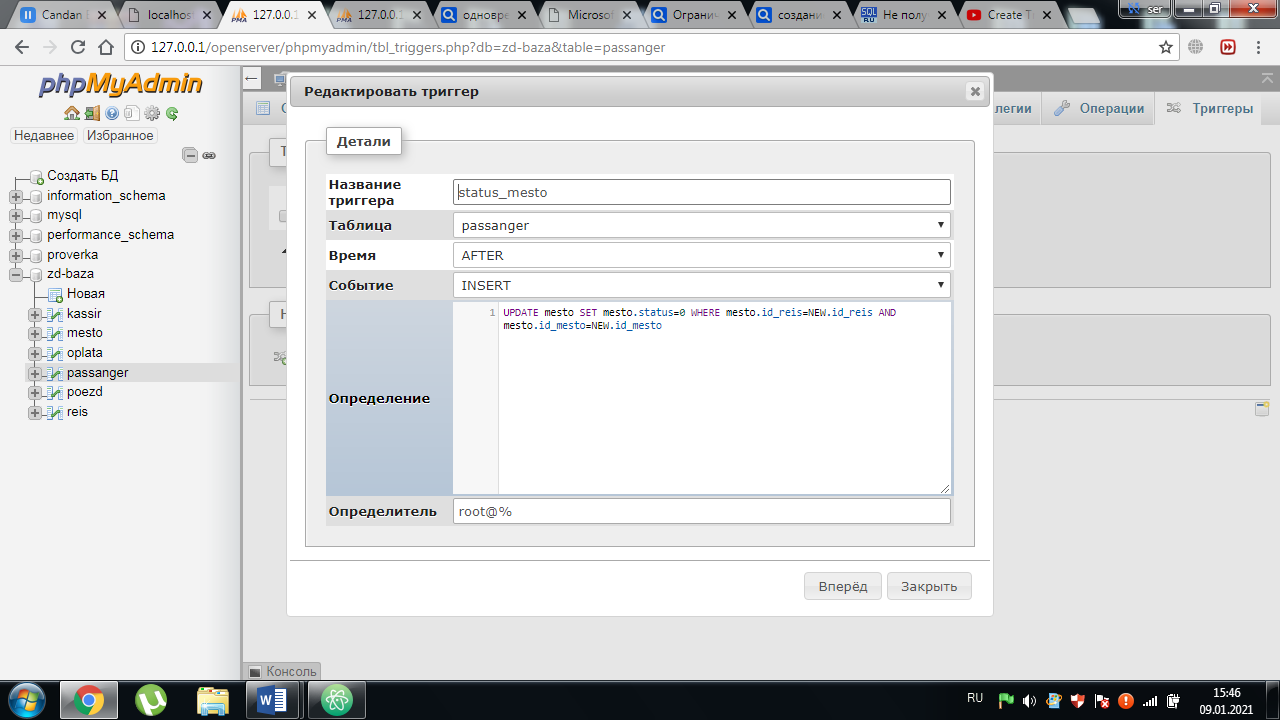
Таким образом завершается процесс продажи билета. Данные о пользователе и оплате вносятся на базу данных и автоматически статус выбранного место меняется на false.

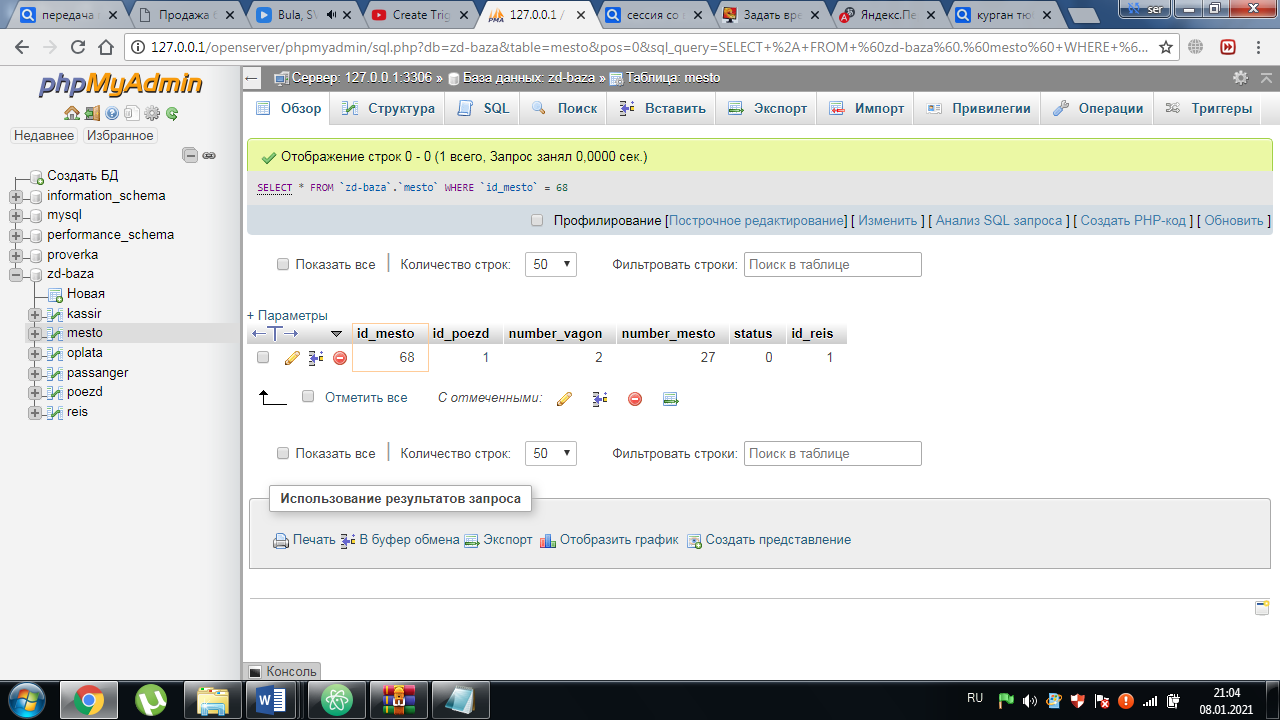


Данные об оплате:



Ибрагим добавился с номером id=42.Также место с id=68 изменил статус на занято для этого было написано триггер на phpmyadmin который отвечает за автосмену статуса:





Пользователь может посмотреть список своих проданных билетов:



Для этого было написано запрос, в котором данные берутся из несколько связанных таблиц:

SELECT passanger.name, passanger.l\_name, passanger.m\_name, poezd.p\_name, mesto.number\_mesto, mesto.number\_vagon, reis.date\_otpr, reis.date\_prih

FROM passanger

Join mesto on passanger.id\_mesto=mesto.id\_mesto

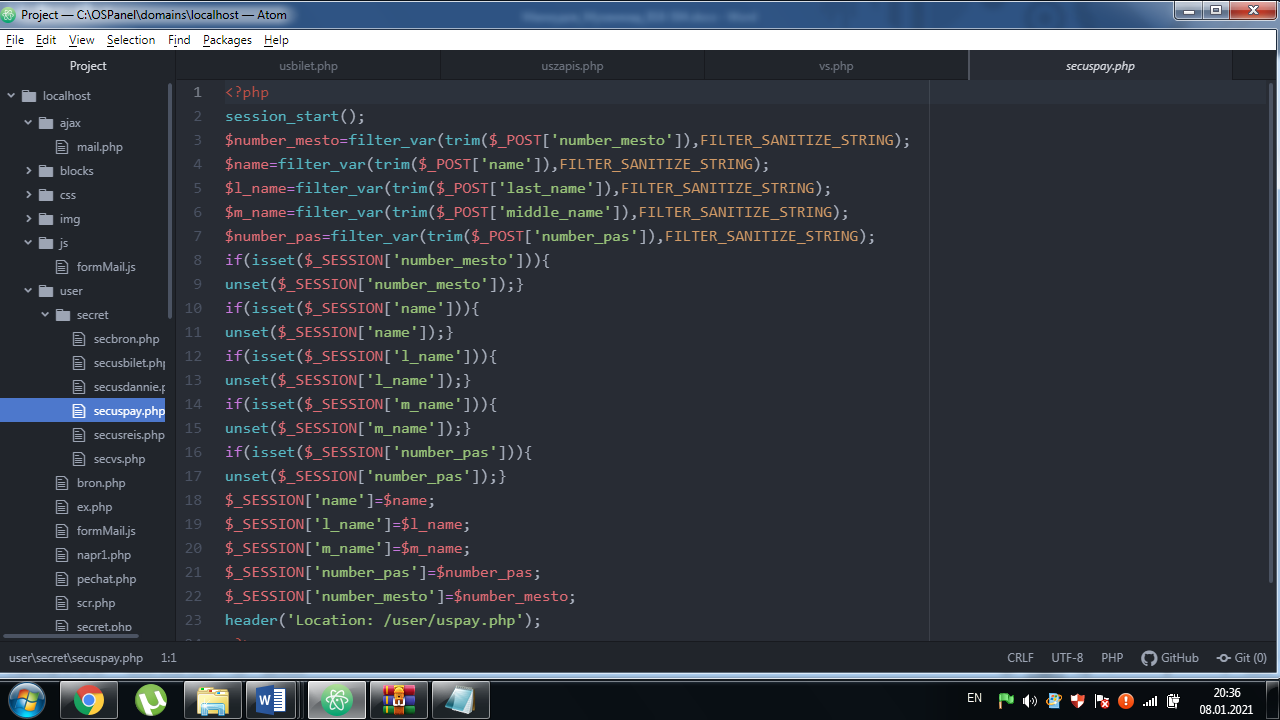
JOIN poezd on mesto.id\_poezd=poezd.id

JOIN reis on poezd.id=reis.id\_poezd

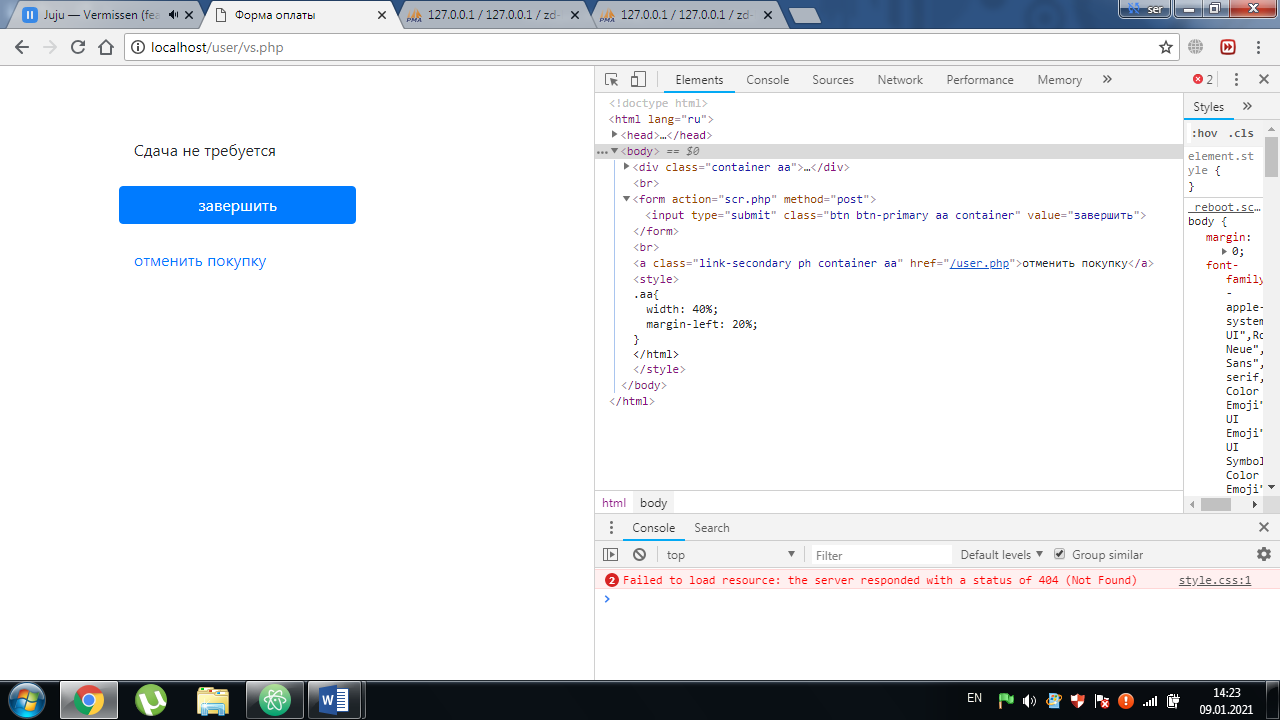
WHERE passanger.id\_kassir=$kassir

(тут $kassir это имя нашего кассира)

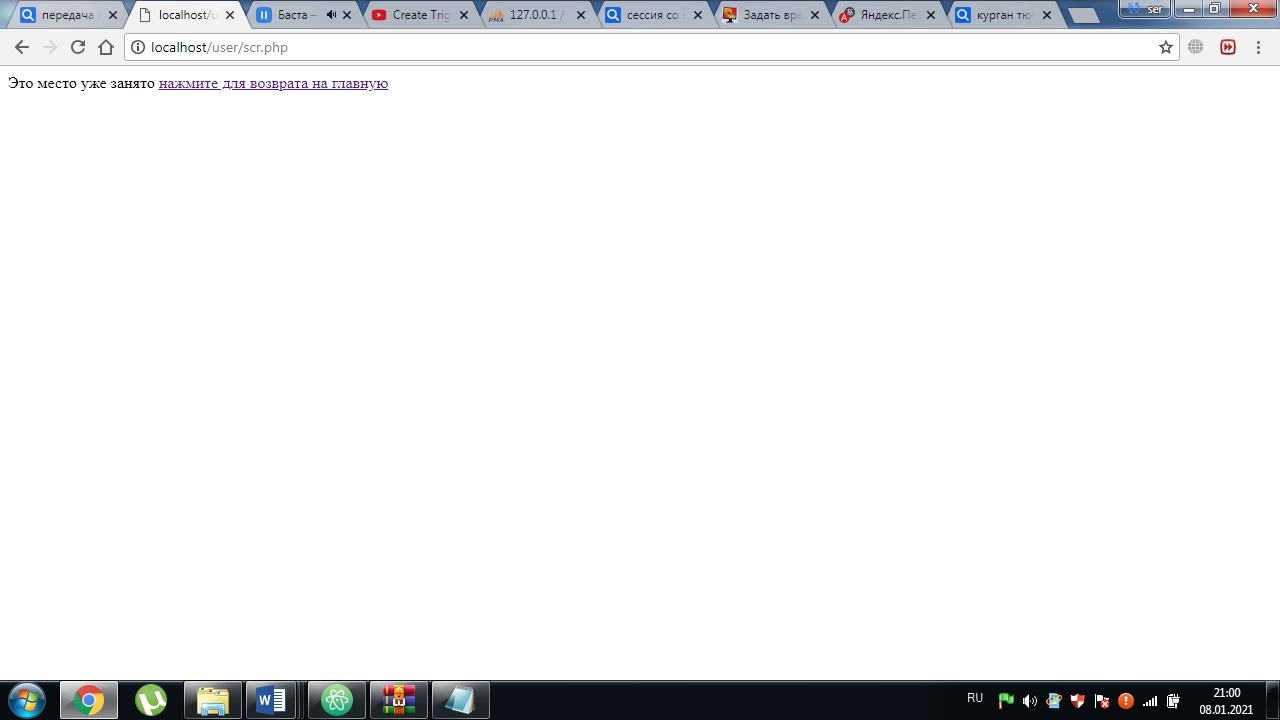
Для того чтобы защитить данные от пользователя(кассира) на изменение через кода страницы все данные принимаются на обработку в отдельных файлах где через скрипты на php записываются на глобальных переменных $\_SESSION на стороне сервера. Например:



Этих переменных невозможно будет увидеть через кода страницы. Например:

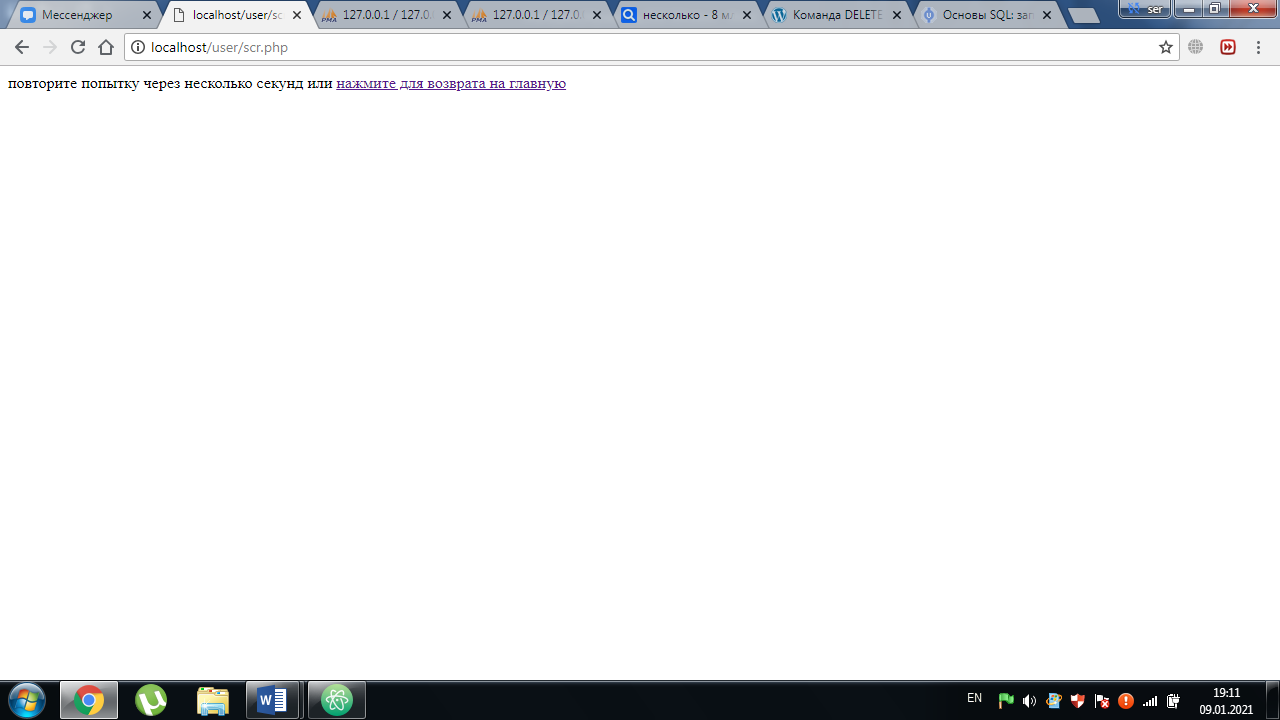


Все повторные отправления данных на базу данных также проверяются перед записью. Именно поэтому если два или несколько кассиров будут бронировать один и тот же билет то продает тот который бронирует первым. Например, повторное отправления:



Все ошибки пользователя обрабатываются прежде чем отправить вводимые или выбираемые данные и совершать какие-то операции с ними.

Также для защити данных данные записываются на базу в самом конце после оплаты. Если одновременно два или несколько кассиров собираются занести данные на базу данных, то для предотвращения дублирование записей было создана ещё одна база данных (можно было в существующей базе создать отдельную таблицу) в котором записывается, то место которое они хотят продать, и перед записью данных на основную базу сначала проверяется данные на дублирование:

Другой вариант предотвращении дублирования это через настройки mysql-разрешить внесение данных на базу только одному пользователю через max\_user\_connections.