|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка программных приложений»

**Тема курсовой работы** АРМ Служба безопасности банка

**Студент группы** ИКБО-11-18 Кубышкина Полина Владимировна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель курсовой работы** доцент Синицын И.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена к защите «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Допущен к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Москва 2020

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Утверждаю

Заведующий кафедрой МОСИТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Головин С.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

«Разработка программных приложений»

Студент Кубышкина Полина Владимировна Группа ИКБО-11-18

**Тема работы:** АРМ Служба безопасности банка

**Исходные данные:**

**З**адание на курсовую работу -АРМ Служба безопасности банка. Разработать систему тестирования знаний по теме дисциплины.

**Функционал разрабатываемой системы должен** предоставлять возможности полностью интерактивной системы с понятным дружественным графическим интерфейсом и обеспечивать необходимую функциональность, которая формируется в зависмости от заданной темы и предметной области изучаемых вопросов.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

Создание дистрибутива общего программного обеспечения (Реализация мультизагрузочного мобильного носителя с дистрибутивами операционных систем MS Windows 10 x64,Linux Mint 19 Mate x64, portable версий данных операционных систем, программных комплеков восстановления и сохранения, программных комплексов диагностики и системного сопровождения).

Освоение применения мультизагрузчиков Grub и Grub-2, освоение языков программирования для формирования специальных скриптов для управления Grub и Grub-2.

Установка и настройка MS Windows и Linux. В том числе настройка интерфейса с применением 3D столов типа Куб и доккетной линейки быстрого запуска приложений, практическое освоение командных языков платформы MS Windows и Linux(конфигурационное программирование).

Установка и настройка Web сервера Apache 2.x, FileZilla FTP Server 0.9.x, почтового сервера Mercury Mail Transport System 4.x, СУБД MySQL 5.x, интерпретаторы языков WEB программирования PHP 5.x и Perl 7.x, утилита обеспечения графического интерфейса при работе с СУБД MySQL.

Изучение жизненного цикла программ и создание программного комплекса с применением языков программирования PHP, PERL,JAVA и т.п., согласно темы курсового проектирования. Обеспечение безопасности информации при работе планового программного комплекса(обеспечение работы ролевой модели безопасности).

Тестирование и диагностика созданного программного продукта.

Портирование написанной программной системы на внешних хостах в сети Интернет.

Отчет по курсовой работе в виде расчетно-пояснительной записки .

**Срок представления к защите курсовой работы:** до «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Задание на курсовую работу выдал** \_\_\_\_\_\_\_\_\_\_\_\_ (Синицын И.В

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Задание на курсовую работу получил**  \_\_\_\_\_\_\_\_\_\_\_\_ (Кубышкина П.В.)

# **СОДЕРЖАНИЕ**

[1 ВВЕДЕНИЕ 5](#_Toc42146010)

[2 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc42146011)

[3 РЕАЛИЗАЦИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 7](#_Toc42146012)

[3.1 Установка необходимого программного обеспечения 7](#_Toc42146013)

[3.1.1 Создание загрузочной мультизагрузочного мобильного носителя с дистрибутивами ОС MS Windows 10 x64,Linux Mint 19 Mate x64 7](#_Toc42146014)

[3.1.2 Установка ОС Windows 9](#_Toc42146015)

[3.1.3 Установка ОС Linux 10](#_Toc42146016)

[3.1.4 Установка программного комплекса XAMPP 11](#_Toc42146017)

[3.2 Создание модели будущей системы 13](#_Toc42146018)

[3.3 Проектирование модели базы данных 14](#_Toc42146019)

[3.4 Создание и разметка страниц 16](#_Toc42146020)

[3.4.1 Начальная страница index.php 16](#_Toc42146021)

[3.4.2 Страница авторизации для администратора и сотрудника 16](#_Toc42146022)

[3.4.3 Добавление данных о сотруднике 18](#_Toc42146023)

[3.4.4 Страница удаления данных о сотруднике 21](#_Toc42146024)

[3.4.5 Страница вывода данных по посетителям и сотрудникам 22](#_Toc42146025)

[3.4.6 Просмотр данных о последней инкассации 24](#_Toc42146026)

[3.5 Добавление сайта на хостинг 25](#_Toc42146027)

[4 ЗАКЛЮЧЕНИЕ 26](#_Toc42146028)

[5 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc42146029)

# **ВВЕДЕНИЕ**

В современном мире, где подавляющее большинство информации хранится на цифровых носителях, особое внимание уделяется ограничению доступа к информации и защите данных. Особенно это важно для крупных коммерческих банковских компаний, которые сохраняют большое количество данных своих клиентов. Утечка таких данных может крайне негативно отразиться как на имидже компании, так и привести к падению ее акций и потере прибыли.

Банки тратят большие суммы на обеспечение безопасности. Работа службы безопасности затрагивает все подразделения банка. Объектами безопасности являются персонал, финансовые средства, материальные ценности, банковские технологии и информационные ресурсы. Тем не менее, учитывая довольно широкую спецификацию данного подразделения, не всегда можно быстро и оперативно решать поставленные задачи. Именно поэтому были созданы автоматизированные рабочие места, направленные на автоматическое решение задач и упрощение работы.

Локально служба безопасности также отвечает за отчетность посещений и доступа на объект, то есть в конкретное здание банка.

# **ПОСТАНОВКА ЗАДАЧИ**

В данной курсовой работе стоит задача разработать программный комплекс службы безопасности банка. Для того, чтобы определить необходимый функционал системы, необходимо обратиться к сторонним источникам.

Первое, что необходимо определить, это задачи, выполняемые сотрудником службы безопасности. Служба безопасности сама по себе является комплексом со сложной вложенной структурой, поэтому необходимо определить, на каком уровне будет находиться сотрудник службы, использующий данную систему. Так, как в рамках курсовой работы нет возможности обратиться к сторонним системам, которые позволили бы на более высоком уровне создать функционал, работа службы безопасности в данной курсовой работе будет сводиться к следующим функциям:

* Добавление сотрудников в базу;
* Удаление сотрудников из базы;
* Добавление данных о посещениях сотрудников;
* Учет данных по инкассации;
* Просмотр данных о посетителях;

Сам по себе процесс выполнения курсовой работы будет сводиться к следующим этапам:

1. Установка необходимого программного обеспечения для выполнения поставленной задачи;
2. Создание модели будущей системы;
3. Проектирование модели базы данных;
4. Создание и разметка страниц в соответствии с построенной моделью;
5. Написание программного кода для автоматизации добавления, удаления и просмотра данных о сотрудниках и инкассации;
6. Регистрация хостинга;
7. Добавление работы на хостинг;
8. Проверка работоспособности системы.

# **РЕАЛИЗАЦИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ**

## **Установка необходимого программного обеспечения**

### **Создание загрузочной мультизагрузочного мобильного носителя с дистрибутивами ОС MS Windows 10 x64,Linux Mint 19 Mate x64**

Для создания мультизагрузочного носителя нам потребуется программа Bootice. Для начала работы с программой Bootice необходимо установить соответствующий софт. После установки запускаем программу и выбираем диск назначения, в нашем случае флешку с именем «General USB Flash Disk».

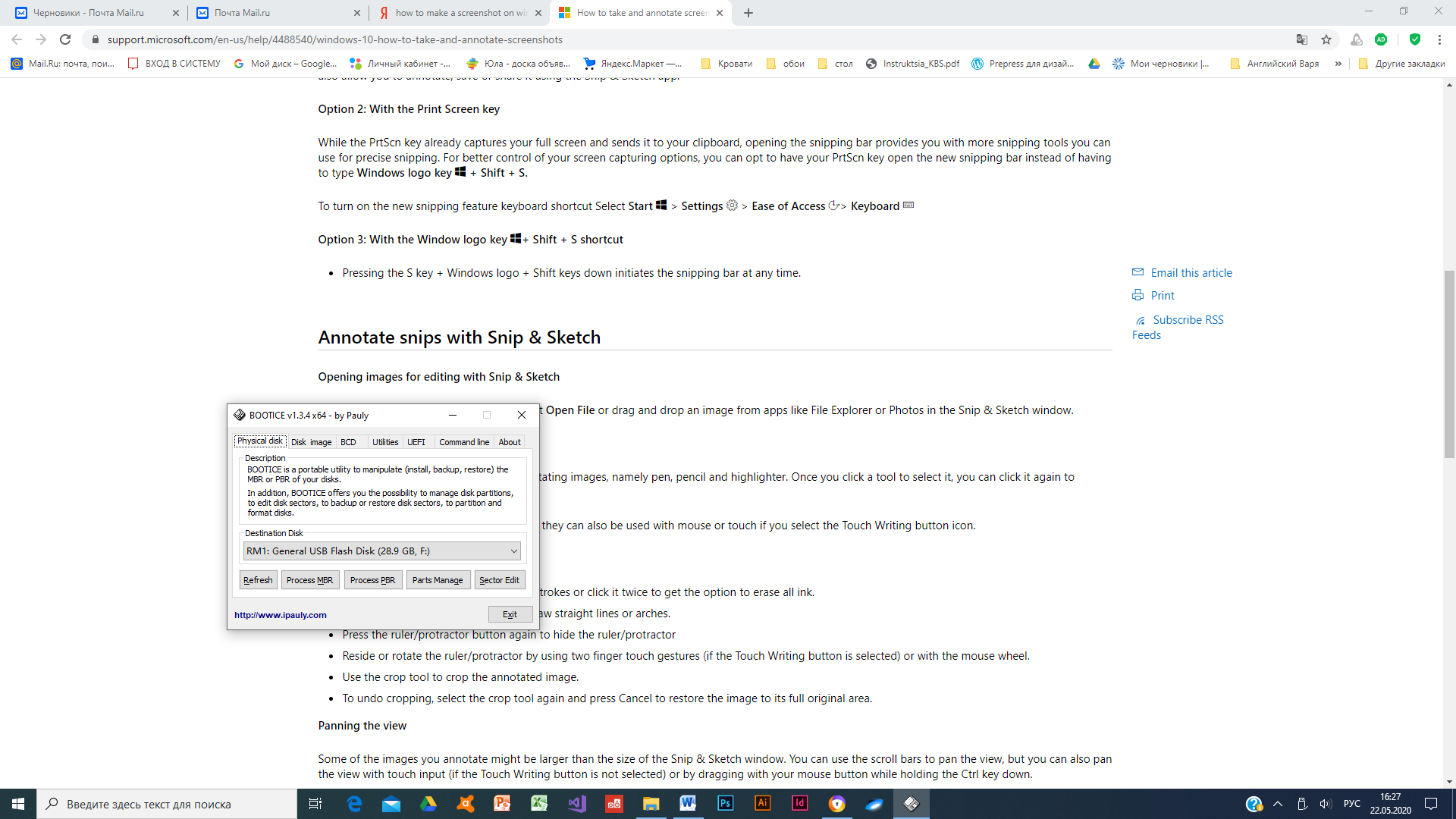


Рисунок 3.1.1.1 – Запуск программы Bootice

Затем выбираем «Process MBR» для настройки основной загрузочной записи «Master Boot Record». Выбираем «GRUB4DOS 0.4.5c/0.4.6a». Данный тип MBR является универсальным и потому выбран для системы по умолчанию.

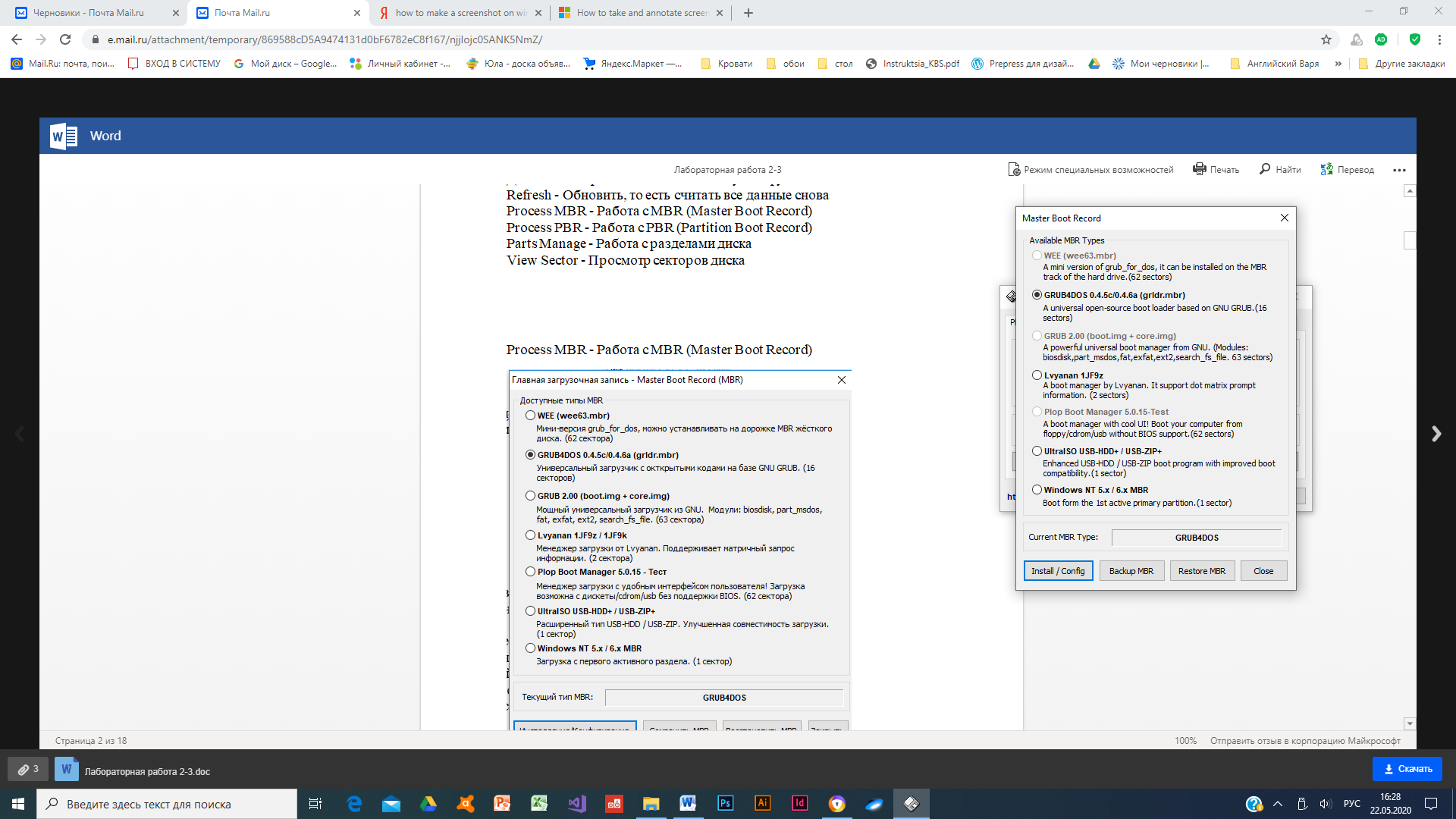


Рисунок 3.1.1.2 – Окно «Process MBR»

После выбора необходимой конфигурации и исталляции, если она необходима, переходим к настройке PBR. Выбираем также «GRUB4DOS», так как он подходит к нашей системе и нажимаем «Install/Config», чтобы применить изменения и изменить тип на нужный нам.

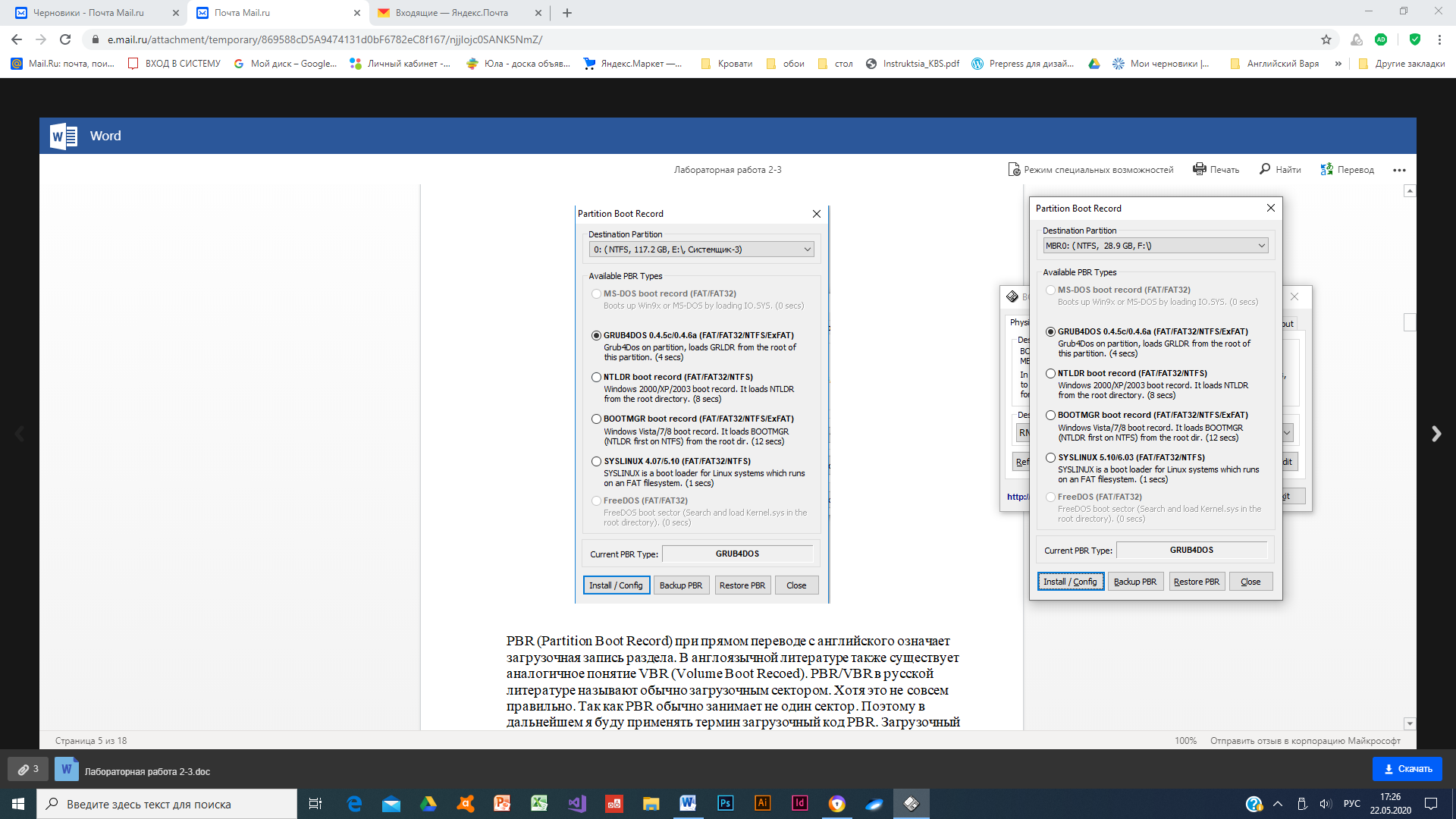


Рисунок 3.1.1.3 – Настройка PBR

При нажатии на кнопку «Parts Manage» открывается окно управления разделами диска, которое позволяет осуществить некоторый набор операций, в т.ч. изменить идентификатор типа раздела, отформатировать раздел или переформатировать USB диск. В нашем случае ничего делать не требуется.

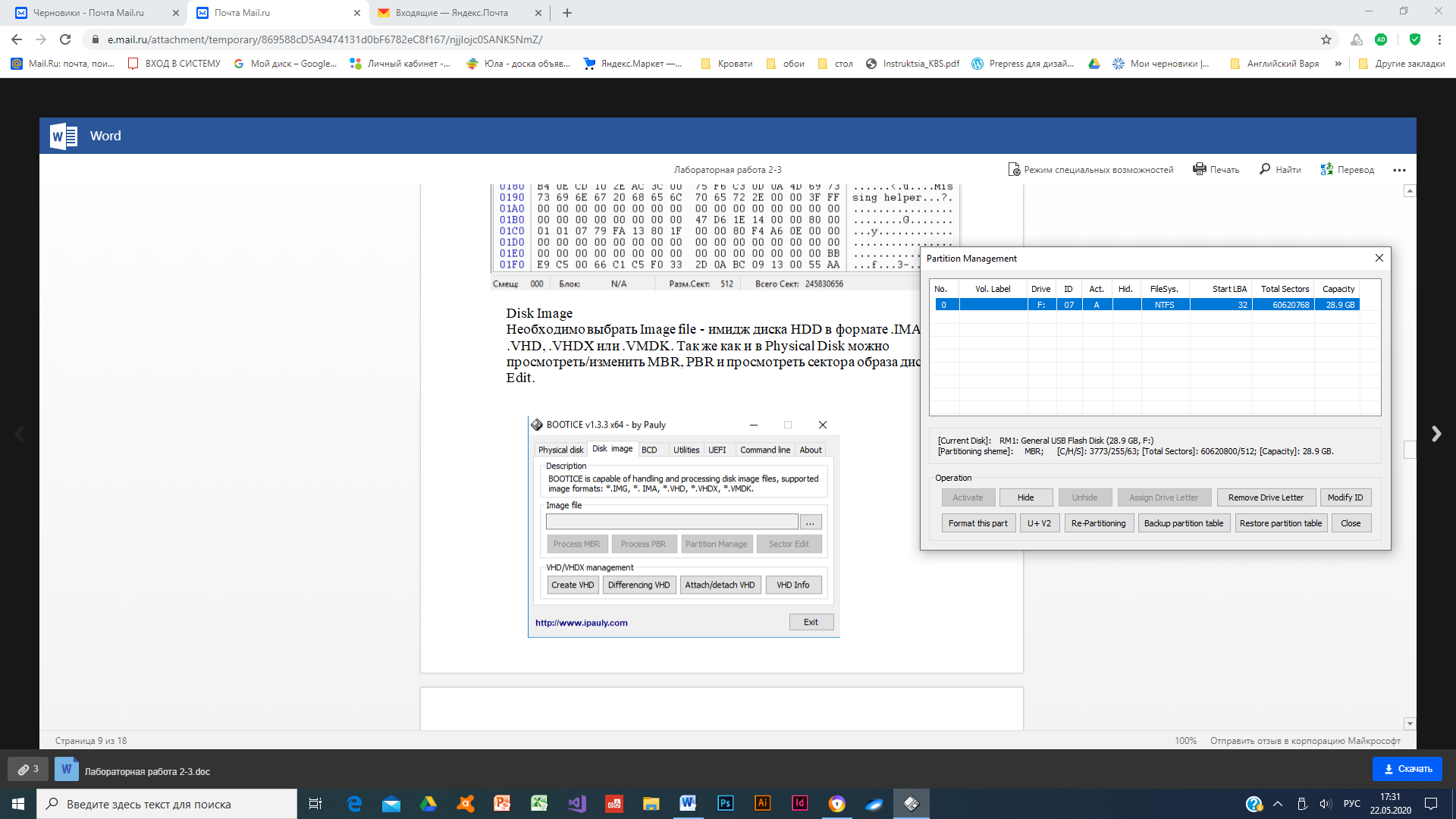


Рисунок 3.1.1.4 – Окно управления разделами диска

Окно «Disk Image» позволяет выбрать для диска image-файл одного из доступных форматов для монтирования образов оптических и жестких дисков.

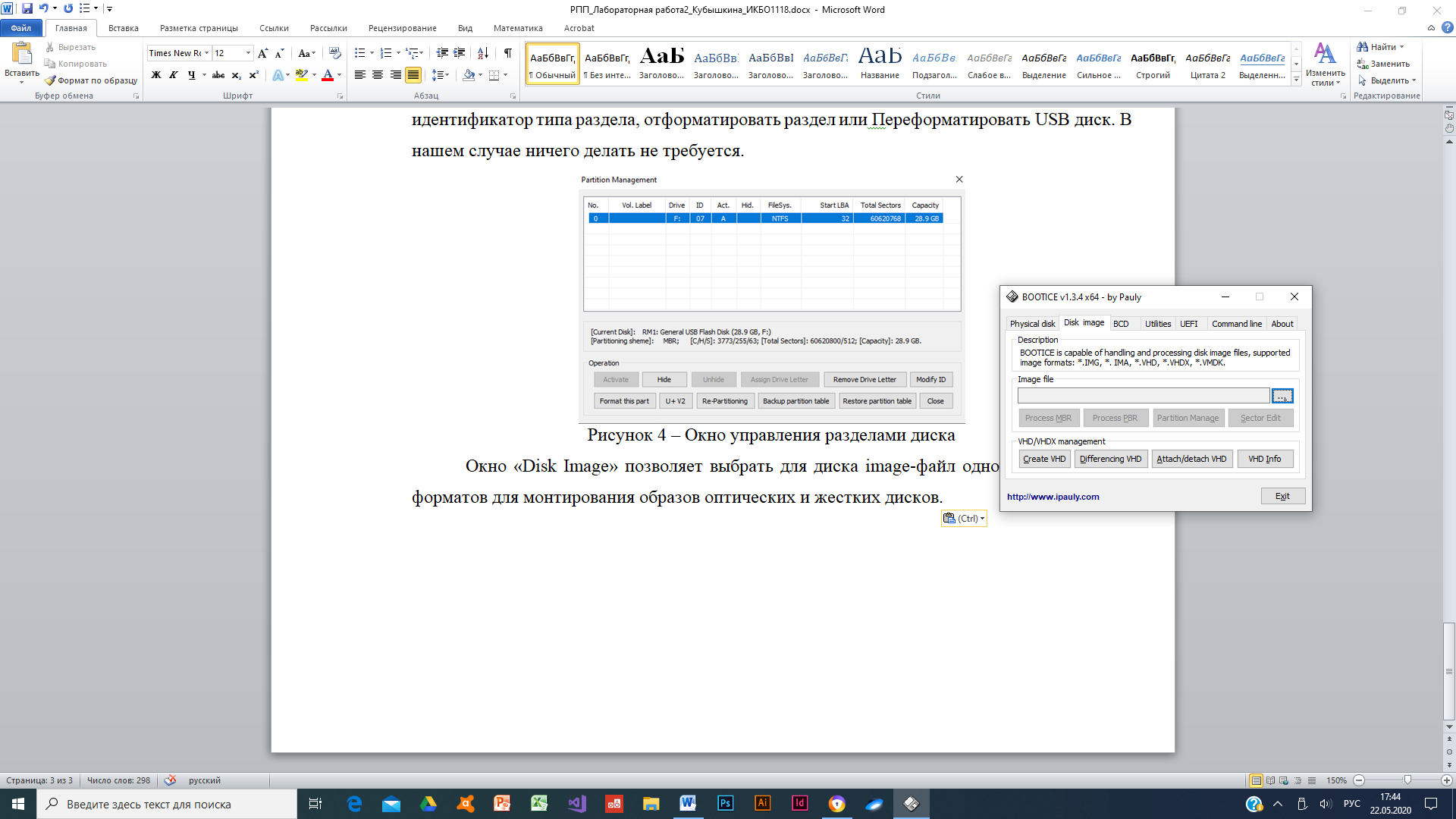


Рисунок 3.1.1.5 – Окно «Disk Image»

Для создания загрузочной флешки фоспользуемся программой «WinSetupFromUSB». После запуска программы в стартовом окне программа автоматически укажет наименование флеш носителя, который будет загрузочным. Необходимо выбрать необходимую ОС и указать путь до образа данной ОС. В данном случае нам необходимо только перенести предоставленные преподавателем образы на флешку.

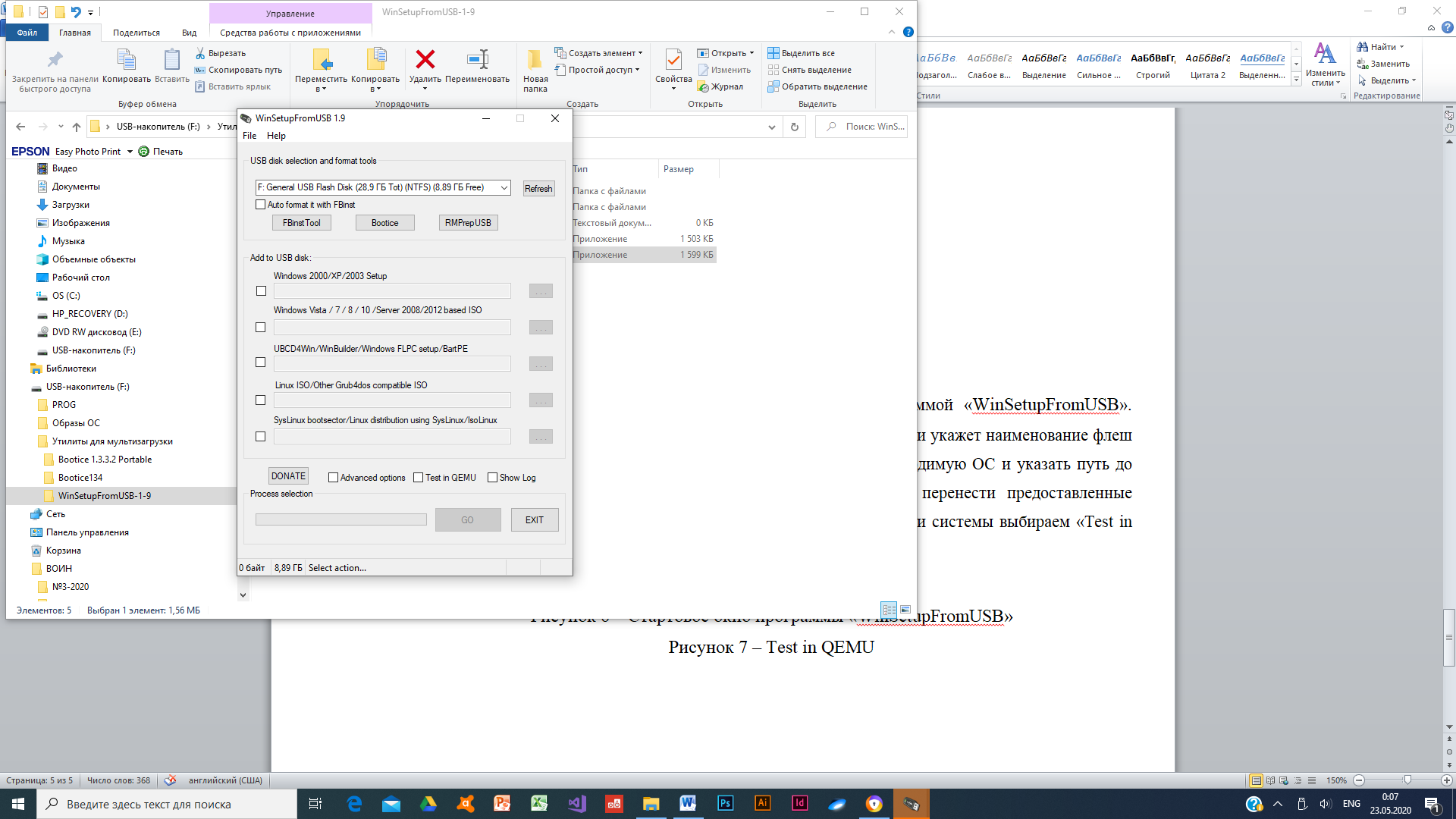


Рисунок 3.1.1.6 – Стартовое окно программы «WinSetupFromUSB»

### **Установка ОС Windows**

Для установки ОС Windows 10. Для этого сначала необходимо зайти в BIOS и указать очередность запуска, чтобы флеш-накопитель, с которого будет вестись установка был на первом месте. Затем, компьютер нужно перезагрузить.

При перезагрузке утилита установщика запустится автоматически. Дальше настройка будет заключаться в выборе параметров системы. Сначала необходимо выбрать формат времени и валюты, и часовой пояс, после чего указать раскладку клавиатуры. Для удобства выберем несколько раскладок – русскую и английскую. Дальше Windows запросит ключ для активации операционной системы, здесь необходимо выбрать «У меня нет ключа продукта». Система запросит подтверждение условий лицензионного соглашения. После выставления галочки нужно выбрать тип «Полная установка системы». Система предложит произвести настройку разделов жесткого диска. Оставляем параметры по умолчанию.

После осуществления основных настроек система предлагает создать пользователя или войти в уже существующую учетную запись. Так, как учетной записи нет, ее нужно создать. После чего можно считать, что установка завершена.

Для корректной работы нужно также настроить сеть. Основная настройка сети происходит в центре управления сетями и общим доступом. Здесь можно поменять сетевое расположение, если во время установки был выбран неправильный вариант, или сеть была изменена. Сетевое расположение компьютера важно, так как от его настройки зависит настройка строгости брандмауэра.

### **Установка ОС Linux**

При запуске компьютера через флеш накопитель Linux Mint запускает живую сессию. Он автоматически открывает рабочий стол с установщиком на нём. Чтобы начать установку необходимо кликнуть на иконку «Установить Linux Mint». После чего запустится установщик.

Сначала система предложит выбрать язык. После чего нужно выбрать параметры подключения к интернету.

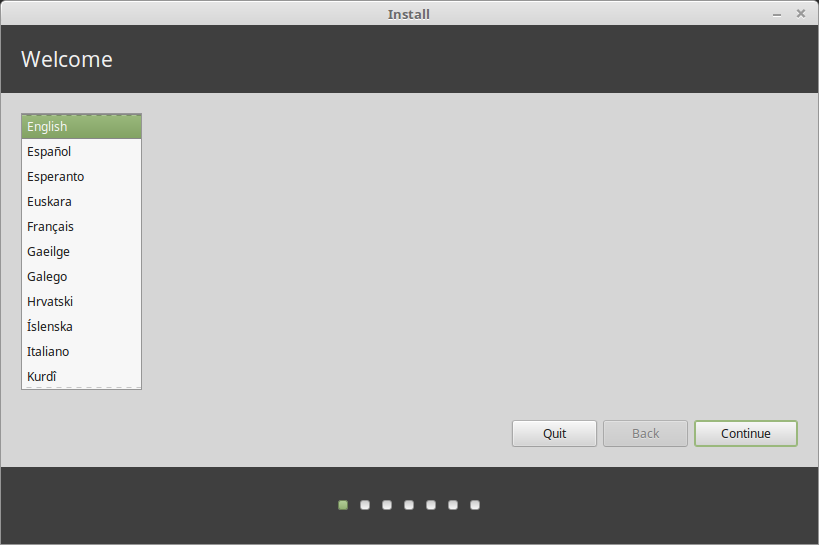


Рисунок 3.1.3.1 – Выбор языка

Затем система предложит выбрать, нужно ли устанавливать мультимедийные кодеки. В рамках курсовой работы, это не нужно.

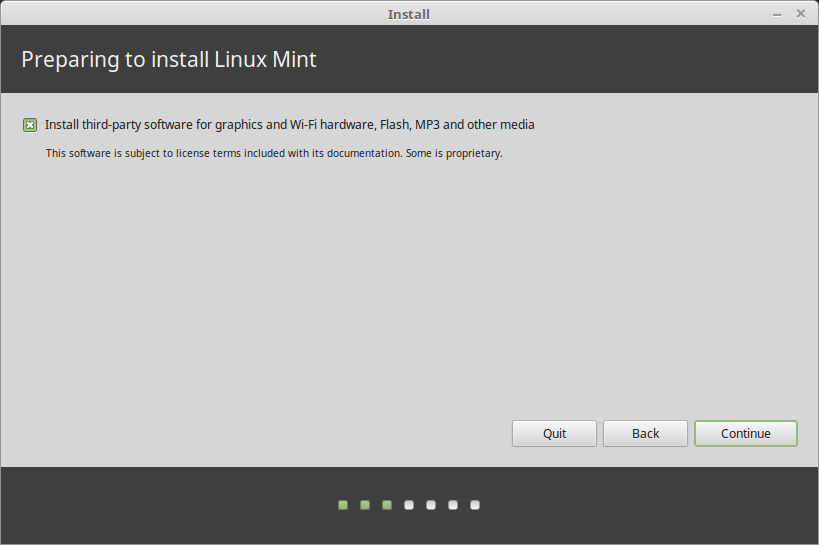


Рисунок 3.1.3.2 – Установка мультимедийных кодеков

На следующем шаге система предложит выбрать тип установки. Так, как установка ведется на виртуальную машину, можно нажать «Стереть все и установить Linux Mint». Дальше нужно указать часовой пояс и выбрать раскладку клавиатуры. Последнее, что попросит система это создать учетную запись. После добавления данных о имени и пароле, Linux начнет полноценную установку, после чего система попросить перезагрузить компьютер и будет готова к использованию.

### **Установка программного комплекса XAMPP**

Компоненты XAMPP:

* X – обозначает, что сборка будет работать на всех операционных системах;
* Apache – веб-сервер, поддерживающий основные версии ОС, гибко

настраиваемый и надёжный;

* MySQL;
* PHP;
* Perl;

Основные этапы настройки:

1.Смена корневой директории

По умолчанию директория для localhost будет "C:\Server\web\", именно в ней находятся файлы, которые отображают в браузере стартовую страницу XAMPP. При необходимости ее можно изменить в настройках сервера Apache, но это делать не обязательно. Для изменения директории остановите сервер Apache, в файле «C:\Server\web\apache\conf\httpd.conf» (этот файл отвечает за конфигурацию сервера) укажите путь к желаемой директории, где будет находиться корень сервера: DocumentRoot "C:/Server/data/htdocs" и <Directory " C:/Server/data/htdocs">. Запустите сервер снова. Для проверки работы можно в этой директории создать какой-нибудь index.html или index.php файл и набрать в браузере адрес - http://localhost/, должно загрузиться содержимое этого файла. Если вы изменили директорию, доступ к phpMyAdmin останется тот же: http://localhost/phpmyadmin/

2.Установка уровня отображения ошибок для PHP

Изначально XAMPP сконфигурирован для разработчика, т.е. в браузере будут отображаться не только критические ошибки PHP, но и различного рода предупреждения, например, "Strict Standards: ...", "Notice: ..." и другие. Для изменения уровня отображения ошибок PHP нужно открыть файл "C:\Server\web\php\php.ini" в любом редакторе и в нем поставить значение: error\_reporting = E\_ALL. При этом значении будут отображаться только критические ошибки PHP.

После внесения изменений в файл конфигурации PHP (php.ini) так же нужно перезагружать сервер Apache. В контрольной панели XAMPP напротив "Apache" нажать кнопку "Stop", если сервис уже запущен, дальше после остановки нажать кнопку "Start".

## **Создание модели будущей системы**

Для создания сайта в соответствии с заданием необходимо первоначально разработать структуру входящих в него элементов. Это значит упростит процесс создания системы и настройки связей внутри.

Так, как тема курсовой работы имеет отношение к службе безопасности делать панель регистрации пользователя нет необходимости. Осуществлять регистрацию пользователей, как и их удаление будет «Администратор» системы. Поэтому, всю структуру условно можно разделить на два блока: блок сотрудника и блок администратора. «Сотрудником» будет являться сотрудник службы безопасности, в обязанности которого входит учет посетителей на объекте и данных по проведенным инкассациям. На рисунке 3.2.1 представлена условная структура разрабатываемой системы.



Рисунок 3.2.1 – Модель портала

## **Проектирование модели базы данных**

В рамках данной курсовой работы в PHPMyAdmin была создана одна база данных, содержащая 5 таблиц. Наличие лишь одной базы данных обусловлено отсутствием необходимости в создании большего количества, а также ограничениями по количеству допустимых в бесплатной версии баз данных у хостинга, на который будет выкладываться система.

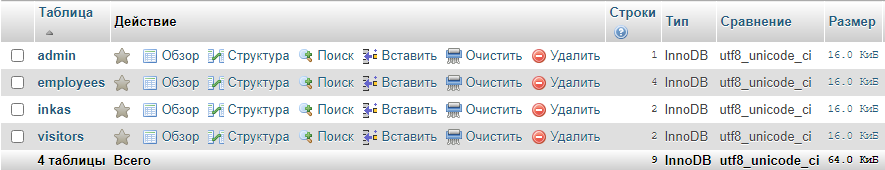


Рисунок 3.3.1 – Структура базы данных

База данных имеет название «security» и содержит таблицы с названиями «employees», «inksas», «admin» и «visitors».



Рисунок 3.3.2 – Структура таблицы employee

Таблица «employees» содержит данные о сотрудниках компании и имеет следующие поля:

* id

Поле содержит информацию об айди сотрудника, и является полем с типом «int»;

* initials

Данное поле содержит данные о ФИО сотрудника. Тип данных поля – text.

* rank

Поле содержит данные о должности сотрудника и также является переменной типа «text».

* department\_id

Поле содержит данные о номере отдела, в котором работает сотрудник. Сотрудники определенных отделов не имеют доступа к системе;

* login

Логин пользователя создается для сотрудника администратором. Содержит параметр UNIQUE, чтобы в системе не было двух одинаковых пользователей. Тип данных – varchar(20);

* password

Пароль пользователю также присваивается администратором. Пароль является набором цифр. Тип данных – varchar(80), что необходимо для хэширования пароля;

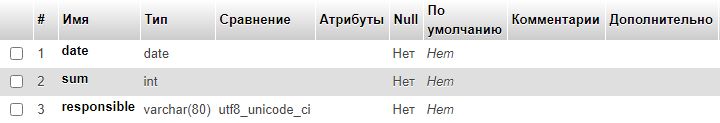


Рисунок 3.3.3 – Структура таблицы inkas

Таблица «inkas» предназначена для хранения информации о проведенных инкассациях. Поля таблицы:

* date

Поле содержит информацию о дате инкассации и является полем типа «date»;

* sum

Сумма инкассации, тип данных – «int»;

* responsible

Данное поле содержит информацию о лице, ответственном за проведение инкассации. Тип данных – «varchar(80)».

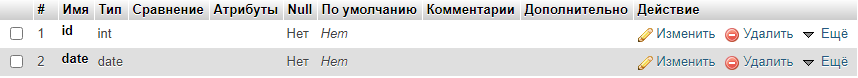


Рисунок 3.3.4 – Структура таблицы visitors

Таблица «visitors» используется для хранения данных о посетителях, т.е. сотрудниках банка. Содержит 2 поля.

* date

Поле типа date, содержит информацию о дате посещения;

* id

Поле для ввода данных о id сотрудника, тип данных – «int».

Таблица «admin» - таблица, содержащая всего одну запись – логин и пароль для администратора, так как в системе он всего один.

* login (тип varchar(30))
* password (тип int)

## **Создание и разметка страниц**

### **Начальная страница index.php**

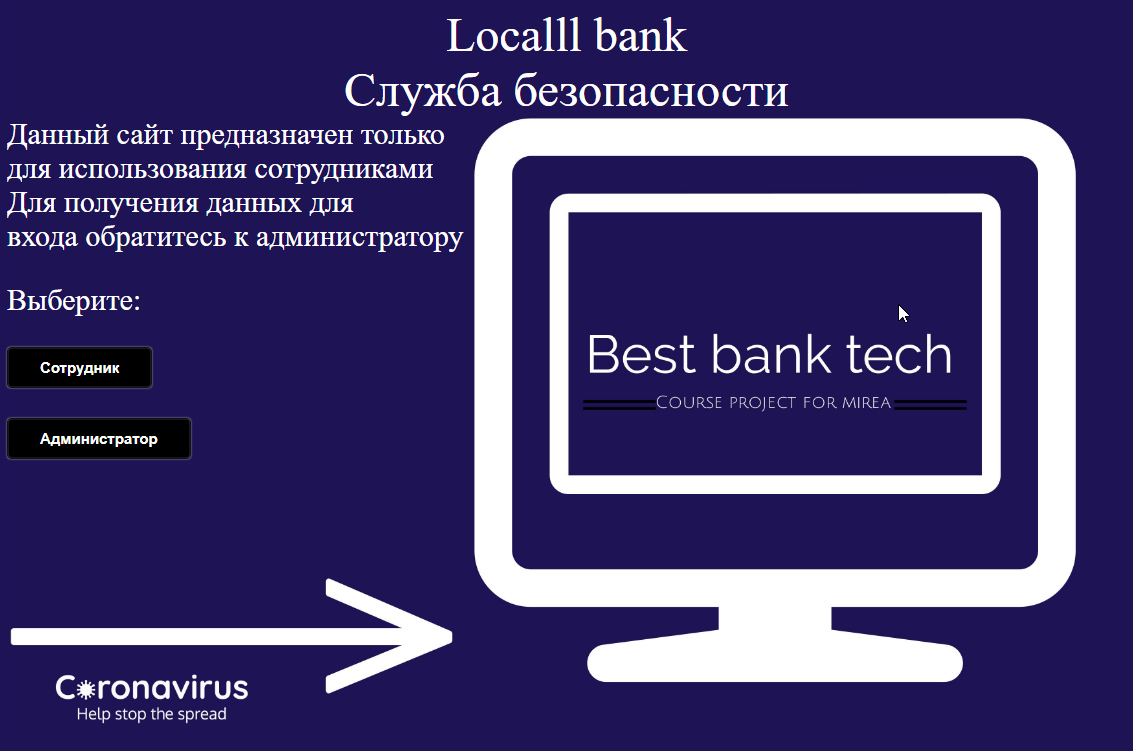


Рисунок 3.4.1.1 – Стартовая страница

На стартовой странице пользователю предлагается выбрать, в качестве кого он хочет войти: администратора или сотрудника.

### **Страница авторизации для администратора и сотрудника**

Структура страниц авторизации для администратора и сотрудника совпадает. Предлагается ввести логин и пароль.

$db = mysqli\_connect("localhost","root","") or die ("Проверьте подключение к сети");

mysqli\_select\_db($db, "security") or die ("Подключение не установлено");

В первой строке производится подключение к базе данных. Если не удается установить подключение, выдается сообщение об ошибке. Вторая строка отвечает за поиск в системе базы данных с названием «security».

$result=mysqli\_query($db, "SELECT login, password, department\_id FROM employees") or die (mysqli\_error());

$array = mysqli\_fetch\_array($result);

Данный фрагмент отвечает за получение из базы данных необходимых строк, а именно логина и пароля, чтобы произвести сравнение введенных данных с имеющимися.

do{

$x = True;

IF ($\_POST['loginn'] == $array['login'] && $\_POST['passw'] == $array['password'])

{

IF ($array['department\_id'] == 1)

{

header("Location: empmenu.php");

$x=False;

}

}

}

while($array=mysqli\_fetch\_array($result));

If ($x = True)

{

echo "Логин или пароль введены неверно. Введите данные повторно или обратитесь к администратору";

}

Производится сравнение введенных логина и пароля с теми, что есть в системе, а также проверка id отдела. Отделом безопасности является только отдел с id равным 1.



Рисунок 3.4.2.1 – Данные для входа администратора

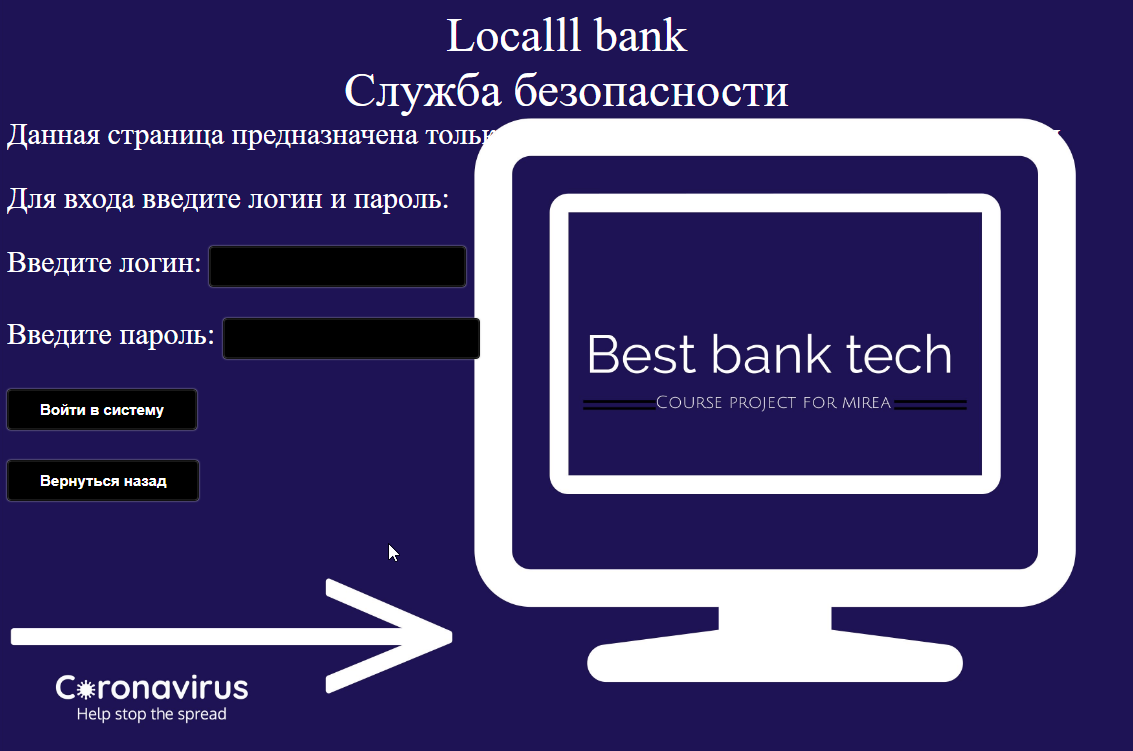


Рисунок 3.4.2.2 – Страница авторизации(admin.php)

На странице авторизации сотрудника интерфейс выглядит аналогичным образом. При вводе данные сверяются с данными в таблице employee. При проверке пароля используется функция password\_verify. Эта функция отвечает за проверку того, соответствует ли пароль хэшу.

$verifypasswd = password\_verify($\_POST['passw'],$array['password']); #Проверка пароля при помощи функции password\_verify   
IF ($\_POST['loginn'] == $array['login'] && ($verifypasswd))   
{   
IF ($array['department\_id'] == 1)   
{   
header("Location: empmenu.php");   
$x=False;   
}   
}

Также, так как в банковской системе существует ограничение прав доступа для сотрудников различных отделов, происходит проверка совпадения айди отдела с единицей – это отдел Безопасности.

### **Добавление данных о сотруднике**

Администратор может добавить данные о новом сотруднике. Для этого необходимо выбрать соответствующую кнопку в основном меню.

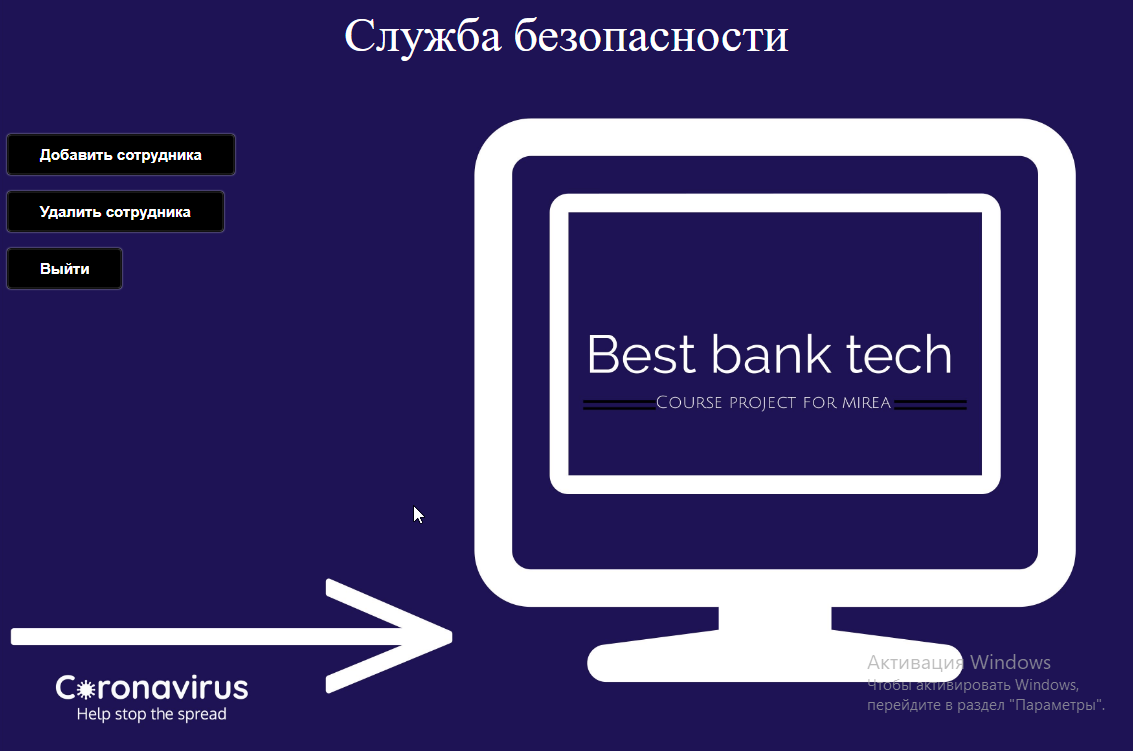


Рисунок 3.4.3.2 – Основное меню для администратора(adminmenu.php)

После перехода на соответствующую страницу (admadd.php) администратору нужно заполнить соответствующие поля.

if ((IS\_NUMERIC($cl\_password)) && (IS\_NUMERIC($cl\_id)))   
{

$hash=password\_hash($cl\_password, PASSWORD\_DEFAULT); #Хэширование паролей

Переменная hash отвечает за шифрование пароля. При помощи функции «password\_hash», на вход которой подается переменная введенного пароля и алгоритм, при помощи которого будет производиться хэширование. «PASSWORD\_DEFAULT» означает, что будет использоваться алгоритм bcrypt.

$t=mysqli\_connect("localhost","root","") OR DIE("Ошибка подключения");    
  
mysqli\_select\_db($t,$dbName) or die(mysqli\_error());   
mysqli\_set\_charset( $t,'utf8' );   
  
$query = "INSERT INTO $itemstable VALUES ('$cl\_initials','$cl\_rank','$cl\_id','$cl\_department\_id','$cl\_login', '$cl\_password')";   
mysqli\_query($t,$query) or die(mysqli\_error($t));   
$query = "SELECT id FROM $itemstable WHERE $initials='$cl\_initials'";   
$result = mysqli\_query($t,$query) or die(mysqli\_error());   
while ($row = mysqli\_fetch\_object($result)) {   
echo "<i>Сотрудник <b>$cl\_initials</b> был добавлен в БД под номером <b>$cl\_id</b>.</i>";   
}   
  
mysqli\_close($t);   
}   
else   
echo 'Проверьте правильность введенных данных';   
}

Система подключается к базе данных, как и в случае авторизации, и аналогичным образом выдает ошибку, если сделать это не получается. Дальше идет проверка на соблюдение правил заполнения, и, если все верно, администратору выводится сообщение о том, что пользователь добавлен в систему.

Результат хэширования пароля в базе данных будет выглядеть следующим образом:

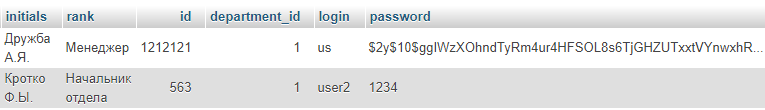


Рисунок 4.3.4.3 – Пример хэширования пароля в таблице employees

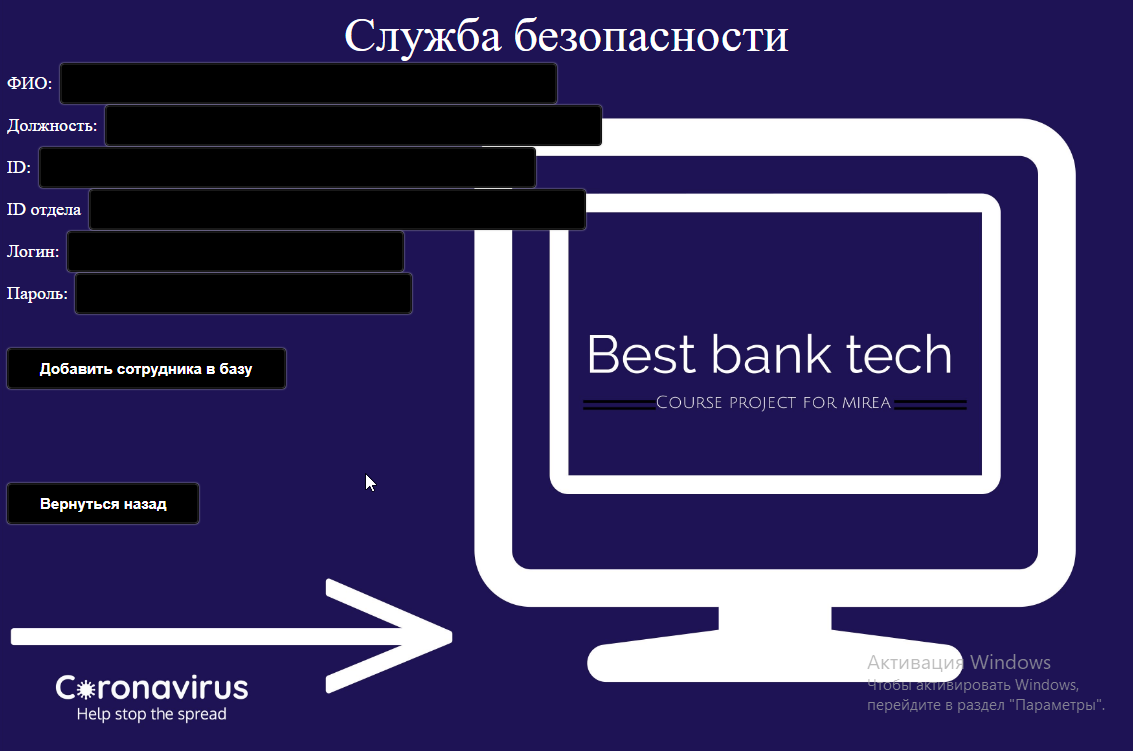


Рисунок 4.3.4.4 – Страница добавления пользователя(admadd.php)

Страницы добавления данных о посетителях и инкассации сделаны аналогичным образом.

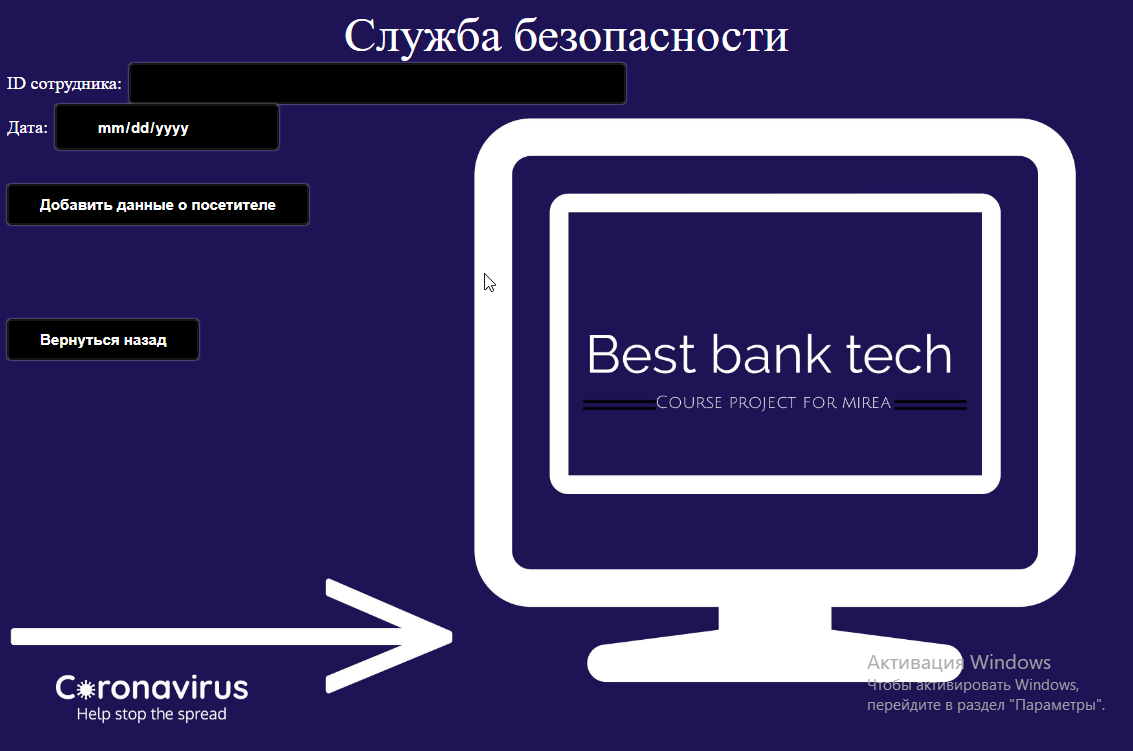


Рисунок 4.3.4.5 – Страница добавления данных о посетителях(empadd.php)

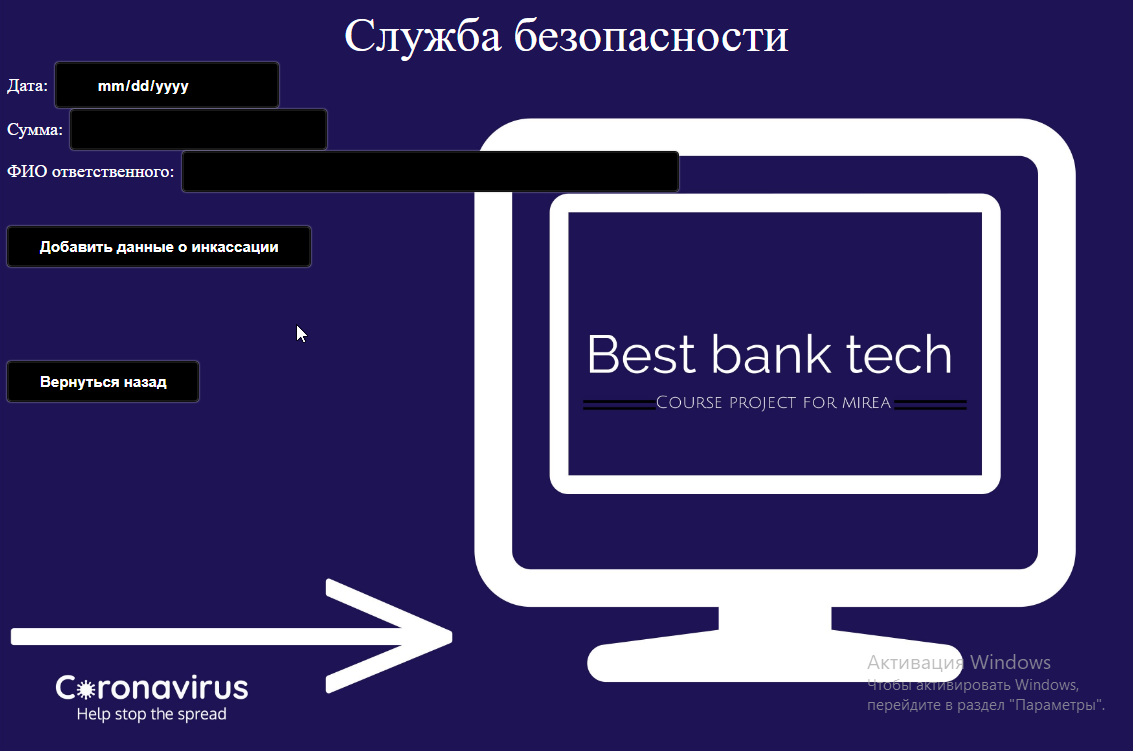


Рисунок 4.3.4.6 – Страница добавления данных о инкассации(empink.php)

### **Страница удаления данных о сотруднике**

Вторая функция администратора – удаление данных о сотрудниках, которые больше не работают в компании. Для выбора сотрудника необходимо в выпадающем меню выбрать соответствующее поле.

$t=mysqli\_connect("localhost","root","") OR DIE("Ошибка подключения");   
mysqli\_select\_db($t,$dbName) or die(mysqli\_error());   
mysqli\_set\_charset( $t,'utf8' );

Система проверяет подключение к базе данных. Если подключение успешно, то выводится таблица с данными фамилии и id сотрудника из таблицы employees.

$query = "SELECT $id, $initials FROM $itemstable";   
$result = mysqli\_query($t,$query) or die(mysqli\_error());   
while ($row = mysqli\_fetch\_object($result)) {   
echo '<option value="'. $row->id . '">'. $row->initials . '</option>';   
}

После того, как администратор нажимает на кнопку «Удалить», система производит проверку совпадения значений и удаляет из таблицы совпавший элемент.

echo '</select><br><input class="second" type="submit" value="Удалить" /><br>';   
  
if (isset($\_POST['user']))   
{   
$user = $\_POST['user'];   
  
$query ="DELETE FROM $itemstable WHERE $id=$user";   
mysqli\_query($t,$query) or die(mysqli\_error($t));   
echo '<i> Сотрудник <b>',$user,'</b> удален.</i> <br />';    
mysqli\_close($t);   
}    
?>

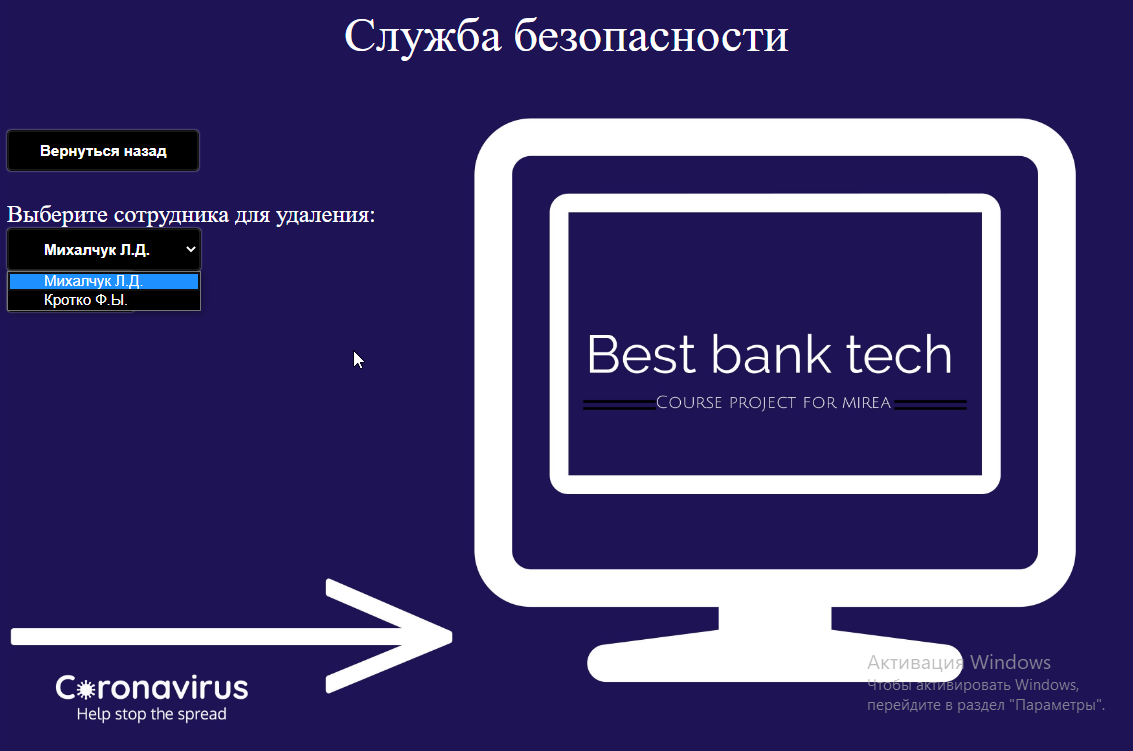


Рисунок 3.4.4.1 – Страница удаления пользователя(admdel.php)

### **Страница вывода данных по посетителям и сотрудникам**

Сотрудник службы безопасности может посмотреть данные по посетителям. Для этого ему необходимо нажать на соответствующую кнопку в основном меню.

Сначала система также производит проверку подключения.

$t=mysqli\_connect("localhost", "root", "") OR DIE("Ошибка соединения");   
mysqli\_select\_db($t,$dbName) or die(mysqli\_error());   
mysqli\_set\_charset($t,'utf8'); 

После этого из таблицы «visitors» выбираются все значения и записываются в «query».

$query = "SELECT \* FROM $itemstable";   
$result = mysqli\_query($t,$query) or die(mysqli\_error());   
  
 Создается таблица, куда построчно выводятся значения.

echo '<table class="zebra"><caption></caption><tr><th><b>ID</b></th><th><b>Дата</b></th></tr></tr>';   
  
while ($row = mysqli\_fetch\_array($result))   
{   
echo '<tr align="center">   
<td><b>'.$row[0].'</b></td>   
<td>'.$row[1].'</td>';   
}

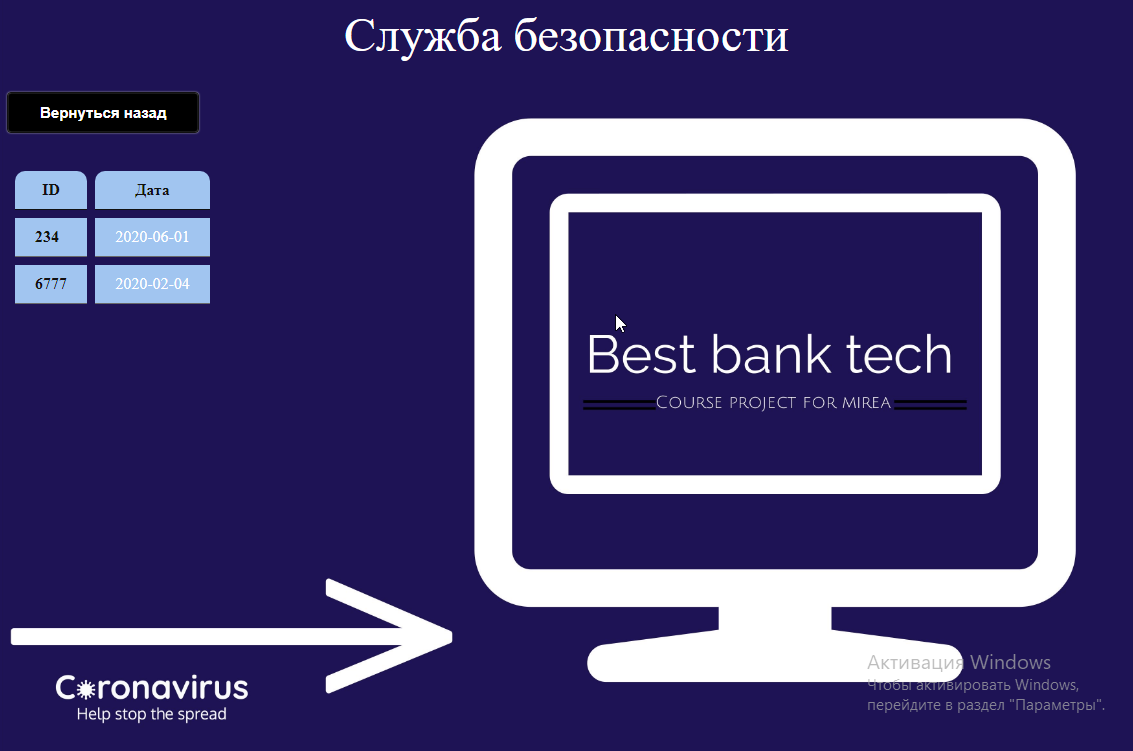


Рисунок 3.4.5.1 – Страница вывода данных по посетителям(emplook.php)

Страница вывода данных по сотрудникам имеет схожий синтаксис, но на вывод в таблицу подаются только три поля: ФИО, данные о должности и айди отдела. Это сделано из соображений безопасности, чтобы пароли и логины не были видны даже администратору.

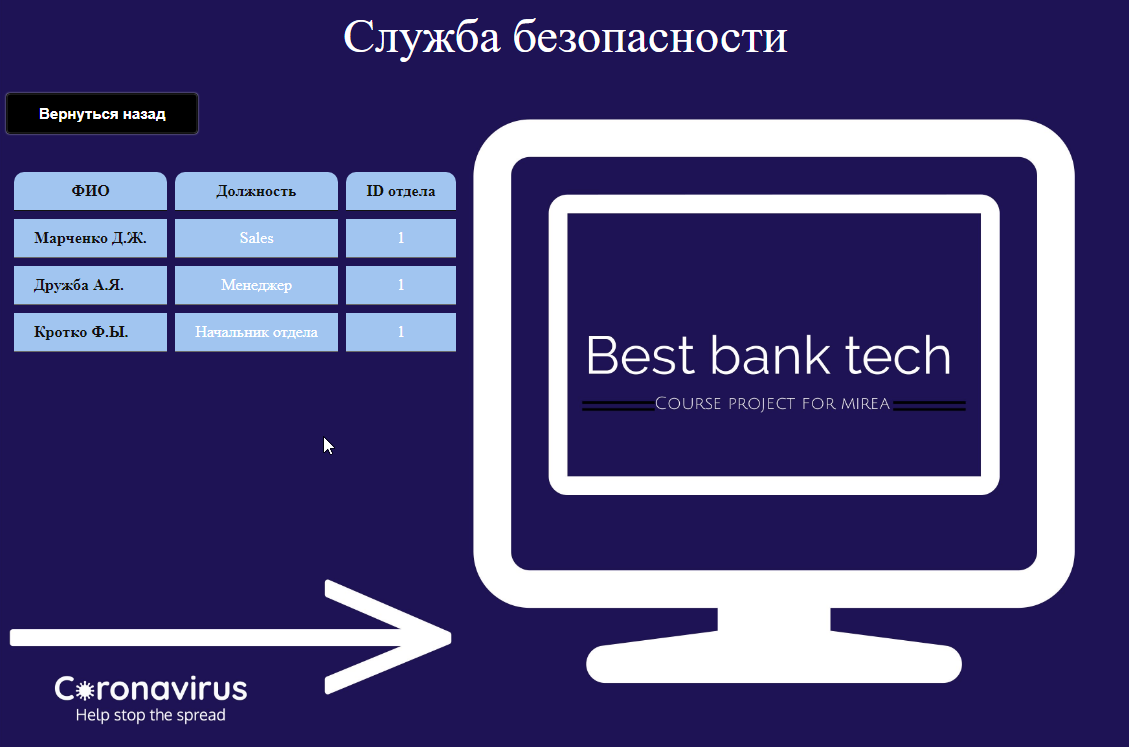


Рисунок 3.4.5.2 – Вывод данных по сотрудникам(admlook.php)

### **Просмотр данных о последней инкассации**

Последняя из функций данного сайта – просмотр данных о последней инкассации. Чтобы осуществить просмотр, нужно выбрать соответствующую вкладку в основном меню. После чего загрузится страница с данными последней инкассации. При этом, данные будут выводиться не о инкассации, данные о которой были добавлены в последнюю очередь, а о инкассации, дата которой ближе всего к сегодняшней.

$t=mysqli\_connect("localhost", "root", "") OR DIE("Ошибка подключения");   
mysqli\_select\_db($t,$dbName) or die(mysqli\_error());   
mysqli\_set\_charset($t,'utf8'); 

Из отсортированной по дате таблицы выбирается одно значение и результат выводится в строку.

$query = "SELECT \* FROM $itemstable ORDER BY date DESC LIMIT 1";   
$result = mysqli\_query($t,$query) or die(mysqli\_error($t));   
while ($row = mysqli\_fetch\_array($result))   
{   
echo '<tr align="center">   
<td><b>'.$row[0].'</b></td>   
<td>'.$row[1].'</td>   
<td>'.$row[2].'</td>';   
}

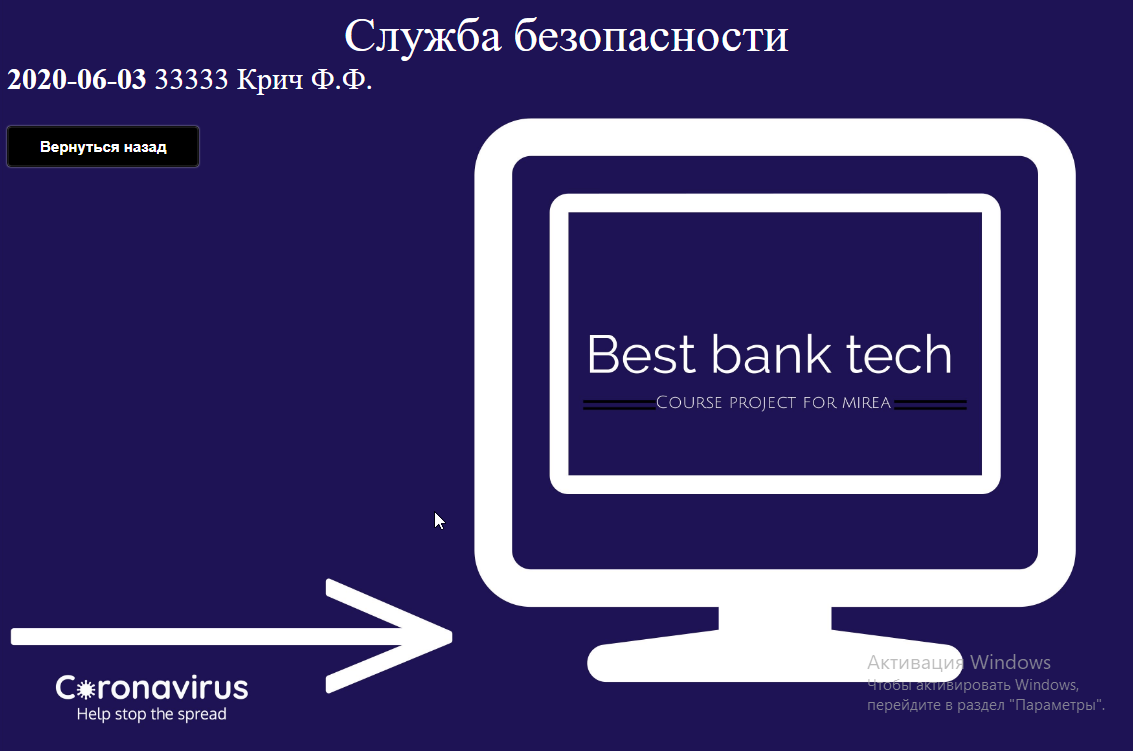


Рисунок 3.4.6.1 – Вывод данных о последней инкассации(lastink.php)

## **Добавление сайта на хостинг**

Для того, чтобы веб-сайт попал в общий доступ, его необходимо загрузить на хостинг. На рисунке ниже приведен скриншот интерфейса хостинга, выбранного для данной курсовой работы.

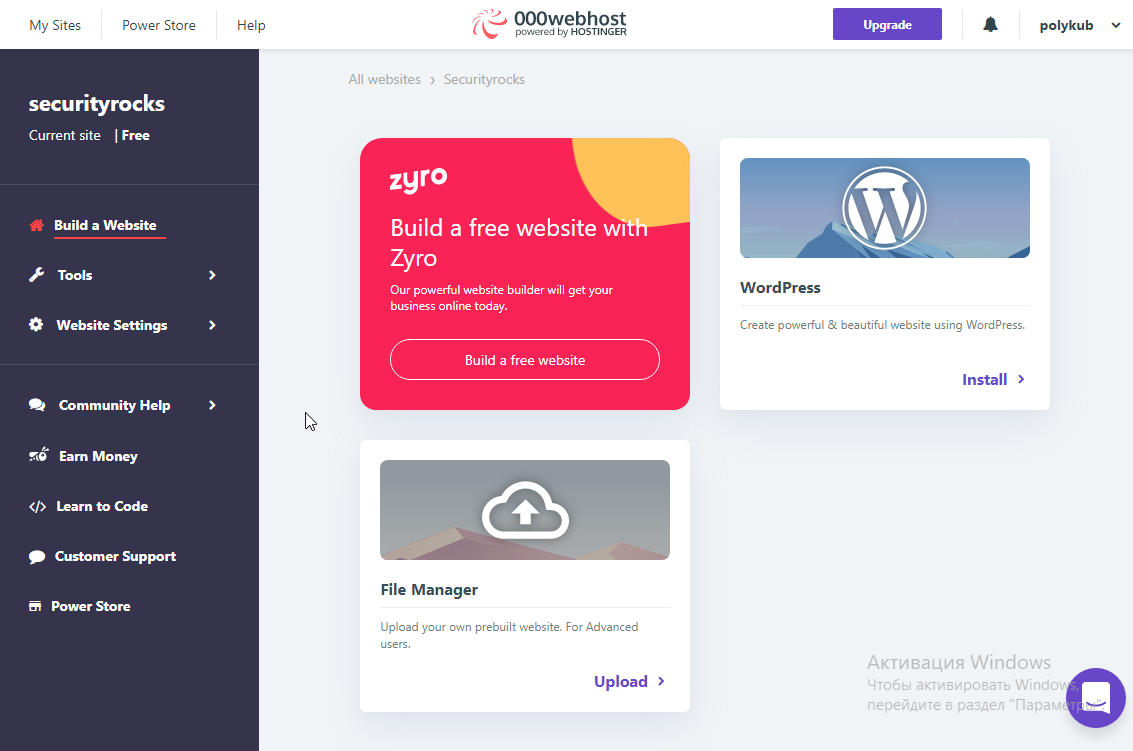


Рисунок 3.5.1 – Интерфейс хостинга 000webhost.com

В файл-менеджере хостинга необходимо импортировать соответствующие файлы.

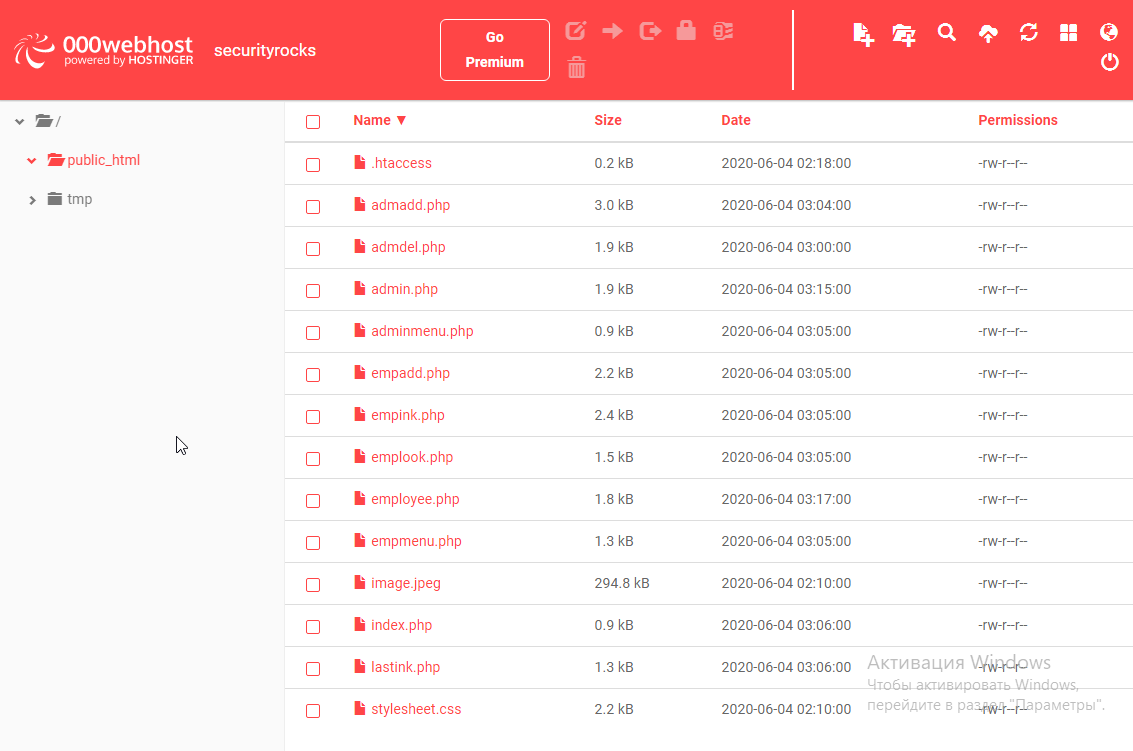


Рисунок 3.5.2 – Файл-менеджер хостинга

Последнее, что необходимо сделать, чтобы система была работоспособной – это добавить созданные базы данных. Для этого нужно выполнить экспорт .sql файлов, в данной курсовой он один, и выгрузить их через специальную вкладку в тулбоксе хостинга. После этого, необходимо изменить данные о названии хоста, базы данных и пароле в файлах, загруженных ранее, и проверить работоспособность системы.

Ссылка на созданный сайт: [https://securityrocks.000webhostapp.com](https://vk.com/away.php?to=https%3A%2F%2Fsecurityrocks.000webhostapp.com%2Findex.php&cc_key=)

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы были получены навыки работы с рядом программных средств, изучены основные команды использования языков PHP, HTML, CSS и MySQL.

Мною были проанализированы материалы по структуре работы, и был разработан и спроектирован проект, реализующий основные задачи, касаемые работы сотрудника службы безопасности банка.

В дальнейшем данный проект можно доработать и усовершенствовать, сделав его полноценным порталом с возможностями сохранения пользовательских данных и реального взаимодействия с компаниями.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Как работают службы безопасности банков [Электронный ресурс] URL: <https://www.investmentrussia.ru/sluzhba-bezopasnosti-banka> (Дата обращения: 02.05.2020)
2. Documentation [Электронный ресурс] URL: <https://www.php.net/docs.php> (Дата обращения: 08.05.2020)
3. Колисниченко Д.Н. PHP и MySQL. Разработка Web-приложений. – 4 изд., перераб. И доп./Д.Н. Колисниченко – СПб.: БХВ-Петербург, 2013 – 560с.: ил.