

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт вычислительной математики и информационных технологий
Кафедра системного анализа и информационных технологий

Направление подготовки: 10.03.01 – Информационная безопасность
Профиль: Безопасность компьютерных систем

КУРСОВАЯ РАБОТА

Электронная медицинская карта

Студент 3 курса
группы 09-641

«__» _____ 2019 г. _____ Гарифуллина В.Р.

Научный руководитель
к.н., доцент

«__» _____ 2019 г. _____ Шаймухаметов Р.Р.

Казань-2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
2 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ СИСТЕМЫ	8
3 РЕАЛИЗАЦИЯ	10
3.1 Авторизация.....	10
3.2 Поиск пациентов	12
3.3 Вывод данных пациента/пользователя в таблицу	12
3.4 Добавление новой записи.....	14
3.5 Загрузка файла.....	16
3.6 Завершение сеанса	17
4 БЕЗОПАСНОСТЬ СИСТЕМЫ	18
4.1 Хэширование паролей	18
4.2 Защита от SQL-инъекций.....	18
5 ОПИСАНИЕ РАБОТЫ СИСТЕМЫ.....	23
5.1 Личный кабинет медицинского работника.	23
5.2 Личный кабинет пользователя.....	27
ЗАКЛЮЧЕНИЕ	29
ПРИЛОЖЕНИЕ	31

ВВЕДЕНИЕ

Современный мир с каждым днем становится все мобильнее, что приводит к созданию таких интерфейсов и систем, которые минимизируют количество времени, затраченное человеком на получение той или иной услуги. В наши дни время является одним из основных богатств, поэтому вопрос о создании удобной и быстрой системы, которая позволит пользователю дистанционно без личного присутствия и огромных временных затрат получить какую-либо услугу, достаточно актуален. Наряду с потребностью создания интерактивных услуг возникает вопрос хранения информации. Бумажные носители все чаще заменяются электронными, а хранение большого количества информации в бумажном виде становится все более трудоемким и неудобным процессом. Современные технологии в достаточной мере позволяют перенести данный процесс в электронный вид, что помогает повысить безопасность хранения данных, предотвратить потерю информации, а также получить доступ к информации, находясь в любой точке земного шара.

К сожалению, многие услуги, в частности государственные, могут быть получены, только при личном посещении, что делает этот процесс достаточно неудобным для клиента. Медицинская сфера не является исключением, несмотря на то, что здравоохранение – одна из ведущих отраслей сферы производства услуг.

Появляется необходимость в создании для медицинских работников и клиентов такого сервиса, который сможет облегчить и автоматизировать хранение данных, сократить время работы с пациентом, и в свою очередь помочь пациентам избавиться от ненужной процедуры «стояния» в очередях, для получения результатов медицинского обследования или данных о состоянии своего здоровья.

В рамках данной проблемы был разработан сервис «Электронная медицинская карта». Данный проект направлен на автоматизацию работы медицинских работников, перевода амбулаторных медицинских карт в

электронный вид, а также добавление новых записей в историю болезней пациента, данных о результатах медицинского обследования и рекомендаций к лечению. Пользователям в свою очередь предоставляется возможность удаленно просматривать историю своих болезней, рекомендации к лечению, справки и результаты анализов.

Используя электронную медицинскую карту, врач сможет максимально быстро получить доступ к информации пациента, необходимой для принятия решения о диагнозе и лечении, уменьшится трата времени на заполнение амбулаторных карт и историй болезни, будет уделяться больше внимания непосредственно работе с пациентом. Вследствие чего, ожидается повышение качества оказания медицинской помощи населению.

1 ИДЕЯ ПРОЕКТА

Возникает необходимость в создании такого сервиса, который будет удобен в использовании, как для обычных пользователей, так и медицинских работников, не имеющих особых знаний в сфере IT-технологий. Сервис должен содержать минимум лишней информации, а также иметь максимально понятный функционал, так как предполагается, что пользоваться данным сервисом будут люди разных возрастов.

Предполагается создание личного кабинета для всех медицинских работников, прошедших процедуру повышения квалификации и допущенных для работы с пациентами. Личный кабинет должен содержать информацию о личности медицинского специалиста, а также о его стаже и специализации. Одной из возможностей медперсонала предполагается поиск пациента в базе данных по уникальному номеру его страхового медицинского полиса.

После успешной аутентификации, врачу должна быть предоставлена личная информация пациента: ФИО, дата рождения, адрес, а также, в удобном табличном виде, выдана информация о его предыдущих посещениях медицинских учреждений. Данная информация должна содержать данные о дате приема, жалобах, диагнозе, медицинском специалисте, к которому обращался клиент, а также данные о прошедших обследованиях пациента и справках, выданных ему по какой-либо причине.

Также в базе данных пациента предполагается хранение документов или результатов медицинских анализов, которые пациент проходил за время лечения. Доступ к данным файлам должен быть максимально быстрым и удобным. Кроме того, вся дополнительная информация о лечении должна быть закреплена за определенной записью в журнале. Дополнительные документы при их наличии должны быть загружены самостоятельно медицинским работником.

Все врачи, имеющие доступ к базе данных пациентов, должны иметь возможность добавлять новые записи, для сохранения информации о новом посещении пациентом медицинского учреждения.

Для сокращения времени работы с пациентом поля, информация для которых может быть получена без помощи медицинского работника, такие как ФИО пациента, дата посещения, ФИО врача, должны заполняться автоматически.

Чтобы исключить потерю/порчу документа (больничный лист, освобождение от учебы и т.д.), освобождающего пациента от какой-либо деятельности, необходимо создание электронного документа, который будет генерироваться, исходя из рекомендаций специалиста. Вся подробная информация должна быть введена врачом при необходимости (кроме тех данных, которые также могут быть получены исходя из личных данных специалиста и пациента). Если такой необходимости нет, поля ввода справки следует оставлять незаполненными.

Для сохранения введенных данных на сервер, данная форма должна быть оснащена кнопкой «Сохранить», при нажатии на которую, вся информация будет записана в базу данных и выведена на экран в таблице истории болезней пациента.

Во избежание случайного или намеренного удаления/изменения данных пациента, возможность редактирования записей у медицинских специалистов должна отсутствовать.

Функционал обычных пользователей должен предусматривать лишь возможность просмотра собственной истории болезней, а также информацию о диагнозе, дате посещения, ФИО специалиста и рекомендациях к лечению. Вся информация должна представляться в табличном виде и быть отсортирована по дате.

Все дополнительные файлы, содержащие данные об анализах и обследованиях пациента, а также полученные им справки от специалиста, при необходимости могут быть просмотрены пользователем и в случае необходимости загружены на устройство.

Возможность редактирования записей, а также добавления новых у обычного пользователя также должна отсутствовать.

Таким образом, пользование данным сервисом будет максимально удобным всем пользователям, независимо от уровня владения персональным компьютером и знаний в области IT-технологий.

2 ПРОЕКТИРОВАНИЕ СТРУКТУРЫ СИСТЕМЫ

Сервис «Электронная медицинская карта» разработан с помощью свободной реляционной системы управления базами данных MySQL.

Для хранения всей необходимой информации в базе данных «medcard» было создано 4 основных таблицы: user, doctor, journal, spravka. Система связей приведена на рис.1.

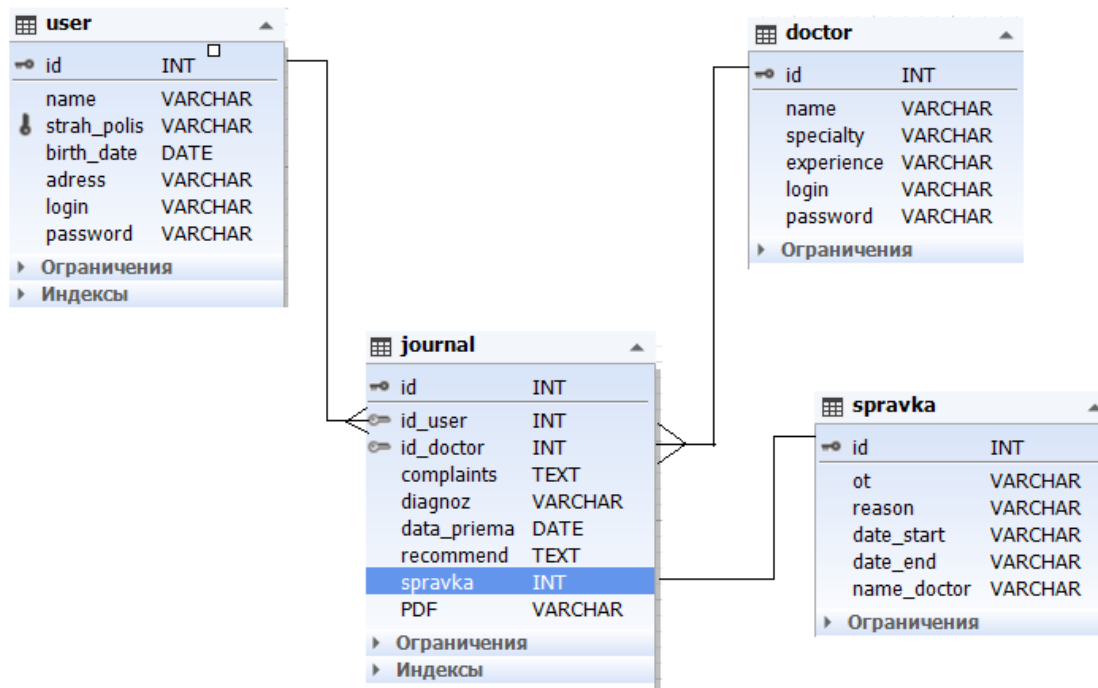


Рисунок 1 Схема базы данных

Таблица «user» предусматривает хранение всех необходимых данных о пользователе: ФИО (name), номер страхового медицинского полиса (strah_polis), дату рождения (birth_date), адрес проживания (address), логин (login), пароль (password).

Таблица «doctor» аналогично таблице «user» хранит информацию обо всех сотрудниках медицинских учреждений, имеющих право работать с пациентами: ФИО врача (name), специализация (specialty), стаж (experience), логин (login), пароль (password).

Таблица «journal» содержит всю информацию о посещениях гражданами медицинских учреждений. Данные в эту таблицу могут вноситься исключительно медицинскими сотрудниками без возможности их последующего изменения. Поле id_user необходимо для поиска всех записей

в журнале, относящихся к конкретному пользователю/пациенту. Поле `id_doctor` хранит информацию о враче, проводившем консультацию. Врач также обязан заполнить поля жалобы (`complaints`), диагноз (`diagnoz`) и рекомендации (`recommend`) к лечению для фиксирования сведений о состоянии здоровья пациента. Поле `PDF` хранит путь к файлу на сервере, загрузку которого производит сам специалист. Это может быть справка, результат медицинского обследования, анализы. Тип загружаемого файла может быть любым. Поле `справка (spravka)` хранит идентификатор записи данного документа в таблицы «`spravka`»

Таблица «`spravka`» необходима для хранения информации о причине освобождения пациента (`reason`) от какой-либо деятельности (`ot`) и сроках действия данной справки (`date_start`, `date_end`). Справка генерируется автоматически и не хранится на сервере в виде изображения.

3 РЕАЛИЗАЦИЯ

Данная система реализована с помощью языка разметки HTML с использованием каскадных таблиц стилей CSS [8]. Связь с локальным сервером устанавливалась с помощью специального языка PHP, предназначенного для работы в среде web-серверов. Для придания интерактивности некоторым элементам web-страниц, использовался язык программирования JavaScript.

3.1 Авторизация

Окно авторизации (рис. 2) представляет собой набор обязательных к заполнению полей, необходимых для прохождения процедуры аутентификации. После введения всех необходимых данных и установление флага Checkbox (при необходимости), активация кнопки «Вход» обрабатывает специальный PHP-запрос на проверку корректности введенных данных. Данный запрос обрабатывается на специальной странице «login.php».

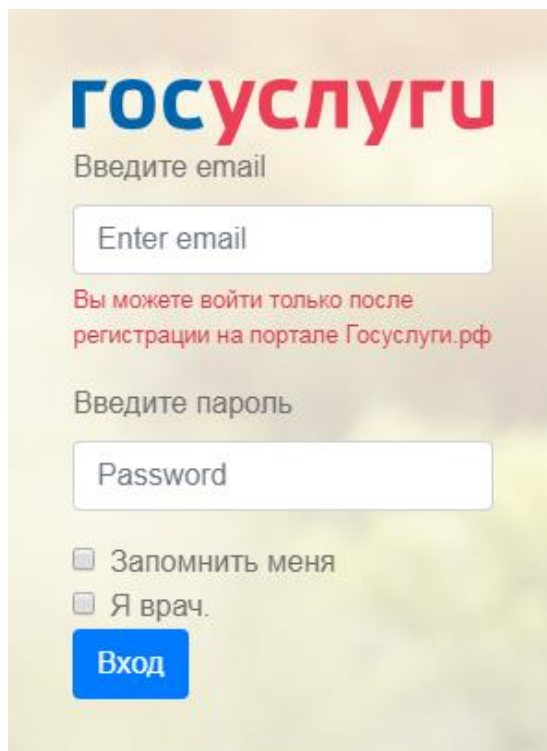
The image shows a login form for the 'Госуслуги' (Gosuslugi) portal. At the top, the word 'ГОСУСЛУГИ' is displayed in blue and red. Below it, the text 'Введите email' (Enter email) is followed by a text input field containing the placeholder 'Enter email'. A red message states: 'Вы можете войти только после регистрации на портале Госуслуги.рф' (You can only log in after registration on the Gosuslugi.ru portal). Below this, the text 'Введите пароль' (Enter password) is followed by a password input field with the placeholder 'Password'. There are two checkboxes: 'Запомнить меня' (Remember me) and 'Я врач.' (I am a doctor). At the bottom is a blue button labeled 'Вход' (Login).

Рисунок 2 Форма авторизации

В базе данных из таблицы «user» с помощью специального SQL-запроса извлекается необходимая информация о пользователе: уникальный идентификатор пользователя, его имя и пароль.

Поиск нужного пользователя происходит исходя из информации поля «email».

SQL - запрос для получения данных пользователя:

```
SELECT id, name, password FROM user WHERE login = "login"
```

Если пользователь с таким логином отсутствует, система оповестит его специальным сообщением и предоставит попытку повторного ввода данных. В этом случае пользователь будет перенаправлен на главную страницу системы (index.php)

Проверка корректности введенных данных:

```
$sql->execute([':login' => $login]);  
if ($sql->rowCount() == 0){  
    echo "не сущ";  
    header("location: index.php");}
```

При успешной аутентификации пользователю открывается сессия (SESSION). Сессия является простым способом хранения нужной информации об отдельных пользователях с уникальным идентификатором сессии. Она необходима для сохранения состояния между запросами страниц. Как правило, идентификатор сессии отправляется браузеру через специальный сессионный cookie и используется для получения имеющихся данных сессии. Отсутствие данного идентификатора сообщает PHP о необходимости создания новой сессии и генерирования нового идентификатора [1].

Если же пароль, введенный пользователем, и пароль, хранящийся в базе данных, не совпали, пользователь получает сообщение «Неверный пароль» и возможность повторной авторизации.

Создание сессии при успешной авторизации пользователя:

```
if (md5($password) == $result["password"]){  
    $_SESSION['USER_ID']=true;  
    $_SESSION['USER_ID']=$result["id"];  
    $_SESSION['USER_NAME']=$result["name"];
```

```
header("location:profil.php");
```

3.2 Поиск пациентов

Поиск пациентов осуществляется только медицинскими работниками, поэтому данный функционал не доступен обычным пользователям. Для поиска пациента врачу необходимо знать номер его страхового медицинского полиса.

После ввода данных и активации кнопки «Поиск» специальный PHP-запрос обрабатывает введенные данные, производя поиск пациента из таблицы user, по значению поля strah_polis:

```
$strah = $_GET["strah"];  
SELECT * FROM user WHERE strah_polis = :strah;
```

Медицинский сотрудник перенаправляется на личную страницу пациента doctor.php.

3.3 Вывод данных пациента/пользователя в таблицу

При переходе медицинским сотрудником на страницу пациента, а также при авторизации пользователя в данной системе, в специальной таблице отображается история болезни пациента (рис. 3). Поиск данных для данной таблицы происходит поэтапно, так как необходимая для вывода информация хранится во всех таблицах, содержащихся в базе данных.

Если авторизованный пользователь не является медицинским специалистом, ему открывается специальная сессия, в которой хранится идентификатор данного пользователя:

```
$id_user = $_SESSION['USER_ID'];
```

По значению идентификатора с помощью SQL-запроса, извлекается вся необходимая информация о данном пользователе из таблицы user:

```
SELECT * FROM user WHERE id = :id_user;
```

А также все данные о посещениях пациентом медицинского учреждения. Для поиска записей используется SQL-запрос с использованием оператора JOIN, позволяющего получить необходимые данные путем объединения нескольких таблиц:

```
SELECT journal.*,doctor.name FROM journal JOIN doctor ON
journal.id_doctor = doctor.id WHERE journal.id_user =
:id_user ORDER by journal.data_priema DESC
```

Результатом данного запроса является массив, содержащий все необходимые для вывода в таблицу данные: дата приема, жалобы, заболевание, рекомендации к лечению, врач, справка, приложения.

Полученные данные в цикле выводятся в таблицу при помощи PHP-кода:

```
<? $i=1;
while ($row = $table->fetch(PDO::FETCH_ASSOC)) {?>
    <tr>
        <th scope="row"><?=$i?></th>
        <td><?=$row["data_priema"]?></td>
        <td><?=$row["complaints"]?></td>
        <td><?=$row["diagnoz"]?></td>
        <td><?=$row["recommend"]?></td>
        <td><?=$row["name"]?></td>
    /*...*/
    <? $i = $i+1;}
```

Поля «Справка» и «Приложения» заполняются только при их наличии. На месте записи в таблице, предназначенной для хранения информации о дополнительных документах и приложениях при наличии записи в таблице journal, устанавливается специальная иконка, являющаяся ссылкой на конкретный документ. Если сведения о справке/приложении в базе данных отсутствуют, данное поле в таблице посещения пользователя остается пустым.

Вывод иконки справки в таблицу:

```
<? if (($row["spravka"]) != 0) {?>
<a href="spravka_gen.php?id=<?=$row["id"]?>">

</a>
<?}??>
```

Вывод иконки приложения происходит аналогичным образом:

```
<? if (($row["PDF"]) != NULL) {?>
<a href="http://medcard.ru/<?=$row["PDF"]?>">

</a>
<?}??>
```

Полученная ссылка содержит путь к файлу на сервере.

#	Дата приема	Жалобы	Заболевание	Рекомендации к лечению	Врач	Справка	Приложения
1	2019-04-14	Тошнота, рвота, диарея, головокружение	Пищевое отравление	Не есть молочное, жареное, фрукты/овощи, пить Смекту 4 раза в день по 1 пакету	Коровин Леонид Викторович		
2	2019-04-14	Резкое ухудшение зрения, частые головокружения, плохой сон	Нарушение зрения	Капли Глазалан в теч. недели 2 раза в день, утром и вечером, ограничить телевизор/компьютер	Каримова Лилия Эдуардовна		
3	2019-03-18	боль в горле, насморк, слезотечение и резь в глазах (в начале болезни), головная боль, кашель, вялость.	Грипп	Лежать, спать, есть и отдыхать, горячий чай с лимоном и мед	Коровин Леонид Викторович		
4	2019-03-15	Упал на лестнице 2.04.19. Боль в области поясницы, отдышка, нервный тик	Ушиб поясницы	1-5 день мазь Антиболь на место ушиба, 10 дней Мелоксикам 3 таблетки в день, покой, ограничение фмз. нагрузки	Валеев Андрей Арсланович		
5	2019-03-14	Температура 38, головная боль, озноб	ОРВИ	Пить больше воды, постельный режим, Антигриппин 2 таблетки 1 раз в день во время еды.	Коровин Леонид Викторович		

Рисунок 3 Вывод данных в таблицу

3.4 Добавление новой записи

Для контроля состояния здоровья пациента, а также для создания структурированного информативного блока записей о каждом посещении пациентом медицинского учреждения, необходимо ведение учета всех посещений гражданина. Главной функцией медицинского работника в данном сервисе является добавление новых записей в историю болезней пациента.

Для получения данной функции на странице пациента создана специальная желтая кнопка «Новая запись» (рис. 4):



ФИО: Иванов Иван Иванович
Дата рождения: 1974-06-19

После заполнения данных и нажатия кнопки "Сохранить", введенные данные изменить будет нельзя. Внимательно проверяйте данные перед отправкой!

Новая запись





#	Дата приема	Жалобы	Заболевание	Рекомендации к лечению	Врач	Справка	Приложения
1	2019-04-18	23456y	2345ty	werty	Коровин Леонид Викторович		
2	2019-04-14	Тошнота, рвота, диарея, головокружение	Пищевое отравление	Не есть молочное, жареное, фрукты/овощи, пить Смекту 4 раза в день по 1 пакету	Коровин Леонид Викторович		

Рисунок 4 Добавление новой записи

При нажатии, появляется модальное окно (рис. 5), содержащее специальную форму для добавления новых данных.

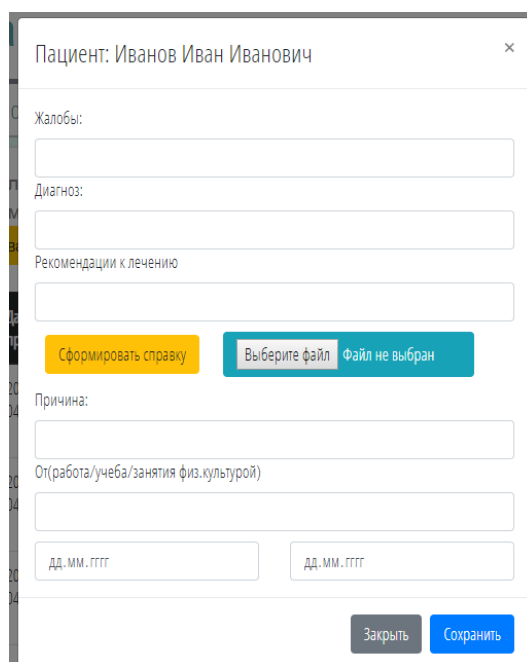


Рисунок 5 Форма создания новой записи

После ввода всех необходимых данных и загрузки файла (при наличии), врач активирует кнопку «Сохранить». Все данные обрабатываются на странице send.php.

Так как данные справки, а также добавление приложений является необязательным действием, перед добавлением новой информации в базу данных происходит проверка на пустоту поля формы «Причина», а также проверка на наличие загруженного файла, во избежание добавления в базу данных пустого поля.

Если поле «Причина» в форме ввода не является пустым, то данные о посещении заносятся в базу данных journal, а поле spravka принимает значения идентификатора справки в таблице spravka, в которой хранится вся необходимая информация для ее генерации:

```
if (!empty($reason)){  
    $sql = $mysqli->prepare("INSERT INTO journal (id_user,  
    id_doctor, complaints, diagnoz,data_priema, recommend, PDF,  
    spravka) VALUES (:user_id, :doctor_id, :jaloba, :diagnoz,  
    CURRENT_DATE(), :rec, :uploadfile, :last_id)");  
}
```

Иначе, поле справки в journal примет значение NULL.

3.5 Загрузка файла

При нажатии кнопки «Выберите файл», пользователю открывается диалоговое окно загрузки файла с компьютера пользователя. Данное поле ввода содержит тип «file». После выбора нужного файла, заполнения обязательных полей и активации кнопки «Сохранить» в специальную переменную `$_FILES['file']` записываются данные о загруженном файле: название, размер, расширение.

Обработка поступившего массива с данными файла для его последующего использования:

```
if ($_FILES['file']['size']!=0){
$upload_dir = 'files/vlojeniya/';
$type_file = pathinfo($_FILES['file']['name'],
PATHINFO_EXTENSION);
$current_date = date("Y-m-d_H-i-s");
$new_name=$user_id."_".$current_date."_".$type_file;
$uploadfile = $upload_dir . $new_name;
if(!move_uploaded_file($_FILES['file']['tmp_name'],$uploadfile)){
echo "Файл не загружен, повторите попытку!\n";
exit();}
}
else{
$uploadfile=NULL;
```

Переменная `$uploadfile` содержит начальный путь файла на сервере. После успешной загрузки файла из переменной `$_FILES['file']` извлекается имя и расширение файла для его последующей корректировки. Новое имя составляется при помощи объединения идентификатора пациента, к которому принадлежит данное приложение, даты посещения пользователем медицинского учреждения, а также типа загружаемого файла. Так как данные заполняются непосредственно во время консультации пациента, дата посещения принимает значение текущей даты: `$current_date = date("Y-m-d_H-i-s")`. После проведения всех необходимых процедур переменная `$uploadfile` будет содержать полноценный путь к файлу на сервере с удобным для классификации названием. Данная переменная заносится в базу данных в поле PDF.

Если произошел сбой загрузки файла, система уведомит пользователя об ошибке и предоставит возможность повторной загрузки.

3.6 Завершение сеанса

Для намеренного завершения сеанса в данном сервисе в шапке профиля располагается специальная кнопка «Выход»

При нажатии на данную кнопку происходит закрытие всех имеющихся сессий и пользователь перенаправляется на главную страницу `index.php`:

```
unset($_SESSION['DOCTOR_ID']);  
unset($_SESSION['DOCTOR_NAME']);  
unset($_SESSION['USER_ID']);  
unset($_SESSION['USER_NAME']);  
header("location: index.php");
```

Для повторного входа в систему, пользователю будет необходимо заново ввести все необходимые для авторизации данные: логин и пароль.

4 БЕЗОПАСНОСТЬ СИСТЕМЫ

В связи с тем, что данные о состоянии здоровья пациентов строго конфиденциальны, вопрос о безопасности данной системы и сохранении целостности информации является очень важным. Для безопасности системы «Электронная медицинская карта» был произведен ряд мер, защищающих данные пользователей от несанкционированного доступа.

4.1 Хэширование паролей

Все пароли в таблице user и doctor хранятся в закодированном виде (рис.6). Кодирование производится с помощью 128-битного алгоритма хэширования MD5. Если злоумышленник получит доступ к базе данных пользователей, то он сможет получить только хэш пароля, а не его настоящее значение. Данная технология хранения не является абсолютно неуязвимой и в настоящее время уже существуют методы взлома MD5-хэша. Поэтому данный метод безопасности является лишь дополнением ко всему комплексу защиты, реализованному на данном сервисе.

id	name	strah_polis	birth_date	adress	login	password
1	Иванов Иван Иванович	1000	1974-06-19	ул. Пушкина 32-279	111	698d51a19d8a121ce581499d7b701668
2	Семенова Юлия Викторовна	1100000000000000	1956-03-04	Дзержинского 54-34	222	bcbe3365e6ac95ea2c0343a2395834dd
4	Киркоров Филипп Васильевич	1234000000000000	1993-07-15	Горького 14-45	123	202cb962ac59075b964b07152d234b70
5	Шариков Дмитрий Андреевич	5890025694390001	2019-01-17	ул. Баумана, д. 14 кв.23	дима	1a814dab1591d4714a0d6db73b9245c2
6	Крутой Игорь Витальевич	4509006999983233	1998-11-30	ул. братьев Касимовых д.3, кв.117	игорь	4177c578ae19eda1861730dabea151bc

Рисунок 6 Хранение паролей в базе данных

4.2 Защита от SQL-инъекций

SQL-инъекции - это один из самых доступных способов взлома сайта, основанный на работе с базами данных. Смысл данных инъекций состоит во внедрение произвольного SQL-кода в запрос. Данный способ атаки можно произвести в любом месте сайта, в котором происходит обмен данными с сервером: поля ввода-вывода, строковые параметры. Такое внедрение позволит злоумышленнику беспрепятственно выполнить произвольный запрос к базе данных и в свою очередь даст возможность изменения/чтения/удаления файлов и выполнения произвольных команд на атакуемом сервере[2-3].

Ввиду того, что на сервисе «Электронная медицинская карта» происходит постоянный обмен информацией с базой данных, защита от SQL-инъекций является важным этапом в организации безопасности всех пользовательских данных.

Чтобы злоумышленник не имел возможность получить доступ к базе данных путем внедрения произвольного запроса, важно, чтобы данные могли попадать в запрос не напрямую, а через некоторое постановочное выражение.

Данный способ работы с базой данных предоставляет специальный драйвер PDO_MYSQL, который обеспечивает безопасный доступ к БД MySQL из PHP [4]. Добавление новых данных или обновление старых являются одними из самых распространенных операций с базами данных. С использованием PDO данный процесс происходит в два этапа: `prepare` и `execute`.

`PDO::prepare` — производит подготовку запроса к выполнению и возвращает объект, связанный с этим запросом [5].

`PDOStatement::execute` — запускает подготовленный запрос на выполнение [6].

`Prepare statement` – это заранее подготовленное корректное SQL-выражение, которое может быть выполнено неоднократно путем отправки серверу лишь различных наборов данных. Параметры подготовленного выражения не нуждаются в экранировании кавычками, так как драйвер делает это автоматически. Использование на сервисе лишь подготовленных выражений дает гарантию защиты от атаки путем внедрения SQL-инъекций.

Подключение PDO к базе данных «medcard»:

```
$mysqli = new PDO('mysql:host=localhost;dbname=medcard',  
$user,$password);
```

Пример использования PDO в системе «Электронная медицинская карта»:

```
$sql = $mysqli->prepare("SELECT * FROM doctor WHERE id  
=:id_doctor");  
$sql->execute([':id_doctor' => $id_doctor]);  
$result = $sql->fetch(PDO::FETCH_ASSOC);
```

Работа с данными на портале «Электронная медицинская карта» производится исключительно путем использования подготовленных выражений, что обеспечивает гарантированную защиту от sql-инъекций, а также конфиденциальность и целостность всех пользовательских данных.

4.2 Шифрование данных с помощью протокола SSL

Данный сервис работает с конфиденциальными данными пользователей, что предполагает постоянный обмен информацией между клиентом и сервером. Такая информация может представлять большой интерес злоумышленникам, поэтому использование протокола передачи данных гипертекстовых документов HTTP является небезопасным, так как данные могут быть легко перехвачены и использованы в корыстных целях.

Для решения данной проблемы был создан криптографический протокол SSL (Secure Sockets Layer), который обеспечивает безопасную передачу информации в сети [7]. Данный протокол позволяет передавать зашифрованные данные по незасекреченным каналам, что обеспечивает надежный обмен информацией между двумя удаленными устройствами. Протокол SSL состоит из нескольких слоев. Первый слой является протоколом транспортного уровня TCP, на котором происходит формирование пакета и непосредственная передача данных по сети. Вторым слоем – защитный SSL Record Protocol. Для обеспечения защищенной передачи данных представленные два слоя являются обязательными, так как совместно они образуют некое ядро SSL.

При шифровании данных используются специальные криптографические ключи различной степени сложности. Сервис «Электронная медицинская карта» использует 2048-битный RSA ключ, являющийся одним из самых надежных для шифрования.

Чтобы иметь возможность передачи данных с помощью SSL протокола, необходимо наличие на сервере специального SSL-сертификата, который содержит некоторые сведения: владелец ключа, центр сертификации, данные об открытом ключе. Данный сертификат может быть получен в центре

сертификации или же выписан пользователем самостоятельно. В ходе создания защищенного соединения для данного сервиса был использован второй вариант.

Существует два типа ключа – public (открытый) и private(закрытый). С помощью открытого ключа можно зашифровать все имеющиеся данные, но расшифровать их можно только используя закрытый ключ. Если у пользователя отсутствует закрытый ключ, расшифровка сообщения становится невозможной. Открытый и закрытый ключ используется исключительно в паре.

В первую очередь возникает необходимость в создании закрытого ключа для использования самоподписанного сертификата. Для его формирования использовалась криптографическая библиотека OpenSSL.

Генерация приватного ключа и сертификата для сервера:

```
openssl req -newkey rsa:2048 -nodes -keyout server.key -x509  
-days 365 -out server.crt
```

Создаются два файла server.crt и server.key, в которых хранятся данные о владельце сертификата (открытый ключ «.crt») и закрытый ключ (.key). После указания пути на сервере к данным файлам, активируется защищенное соединение по протоколу HTTPS (рис. 7).



Рисунок 7 Активация протокола HTTPS

Для проверки работы обновленного протокола используется специальная программа-анализатор трафика компьютерных сетей Wireshark.

Перед началом передачи данных сервер и клиент обмениваются приветственными сообщениями, а также информацией о данных открытых ключей (рис. 8):

24 0.116494	192.168.43.1	192.168.43.60	TLSv1.2	583 Client Hello
25 0.127368	192.168.43.60	192.168.43.1	TLSv1.2	1187 Server Hello, Certificate, Server Key Exchange, Server Hello Done
27 0.141152	192.168.43.1	192.168.43.60	TLSv1.2	192 Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
28 0.141153	192.168.43.1	192.168.43.60	TLSv1.2	795 Application Data
30 0.142296	192.168.43.60	192.168.43.1	TLSv1.2	324 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
32 2.145758	192.168.43.60	192.168.43.1	TLSv1.2	522 Application Data
34 12.646481	192.168.43.60	192.168.43.1	TLSv1.2	97 Encrypted Alert

Рисунок 8 Настройка связи клиент-сервер

Передача данных происходит в зашифрованном виде (рис. 9):

```

> Frame 57: 795 bytes on wire (6360 bits), 795 bytes captured (6360 bits) on interface 0
> Ethernet II, Src: SamsungE_f6:4e:5d (ec:10:7b:f6:4e:5d), Dst: HonHaiPr_28:db:dd (44:1c:a8:28:db:dd)
> Internet Protocol Version 4, Src: 192.168.43.1, Dst: 192.168.43.60
> Transmission Control Protocol, Src Port: 39866, Dst Port: 443, Seq: 644, Ack: 1122, Len: 729
  Secure Sockets Layer
    TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 724
      Encrypted Application Data: 00000000000000014df7bf0bbf5ab09532e3296d4bd8a887...

```

Рисунок 9 Пример передачи зашифрованных данных

Шифрование данных производится с помощью алгоритма RSA и SHA512 (рис. 10).

```

  TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 333
  Handshake Protocol: Server Key Exchange
    Handshake Type: Server Key Exchange (12)
    Length: 329
  EC Diffie-Hellman Server Params
    Curve Type: named_curve (0x03)
    Named Curve: secp256r1 (0x0017)
    Pubkey Length: 65
    Pubkey: 0460763ebd5af72a486cd85e60f87f18b9c3dcdad5846611...
  Signature Hash Algorithm: 0x0601
    Signature Hash Algorithm Hash: SHA512 (6)
    Signature Hash Algorithm Signature: RSA (1)
    Signature Length: 256
    Signature: 078bf543cd861f2315ff76ee2b86faa05fd783390e496faf...

```

Рисунок 10 Шифрование

Весь представленный комплекс защиты данного сервиса обеспечивает гарантированную надежность хранения и передачи данных.

5 ОПИСАНИЕ РАБОТЫ СИСТЕМЫ

Система «Электронная медицинская карта» разработана для удобной организации работы, а также для быстрого доступа ко всем интересующим данным, как сотрудников здравоохранения, так и обычных пользователей.

Управление системой осуществляется с помощью функционала «Личный кабинет». Регистрация в данной системе отсутствует, так как предполагается, что доступ к личному кабинету в этом сервисе могут иметь только лица, зарегистрировавшиеся на официальном государственном портале и подтвердившие свою личность.

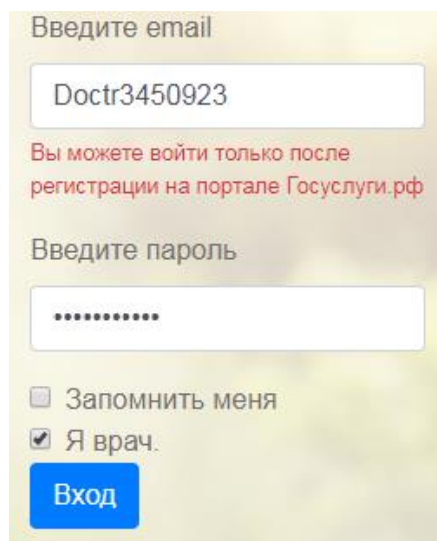
Пользоваться системой «Электронная медицинская карта» могут не только сотрудники медицинских учреждений, но также и обычные граждане, которые в любое время смогут получить информацию о своем здоровье, а также иметь быстрый доступ к справкам и результатам анализов. В связи с этим функция «Вход» в систему разделяется на категории: «Врач» и обычный пользователь, что подразумевает различие в функционале и разрешенных действиях для данных категорий пользователей.

5.1 Личный кабинет медицинского работника.

Так как в данной системе полностью отсутствует регистрация, медицинский персонал может осуществить вход с помощью учетной записи, содержащей уникальный идентификатор пользователя: логин и пароль. Данные учетной записи выдаются руководству медицинского учреждения только для врачей, подтвердивших свою квалификацию и имеющих все полномочия для работы с пациентами. После успешного прохождения всех необходимых процедур, сотрудник получает индивидуальные данные учетной записи.

Для входа в личный кабинет медицинский сотрудник в разделе «Вход» должен установить специальный флажок в разделе «Я врач» (рис. 11). Данное действие является необходимым, ввиду того, что даже при введении верных данных учетной записи, попытка авторизации будет неудачной, так как поиск сотрудника будет производиться в базе данных, предназначенной

для хранения данные только гражданских лиц. Идентификационные данные сотрудников располагаются в специально созданной для этого таблице базы данных, не связанной с базой данных обычных пользователей.



Введите email

Doctr3450923

Вы можете войти только после регистрации на портале Госуслуги.рф

Введите пароль

.....

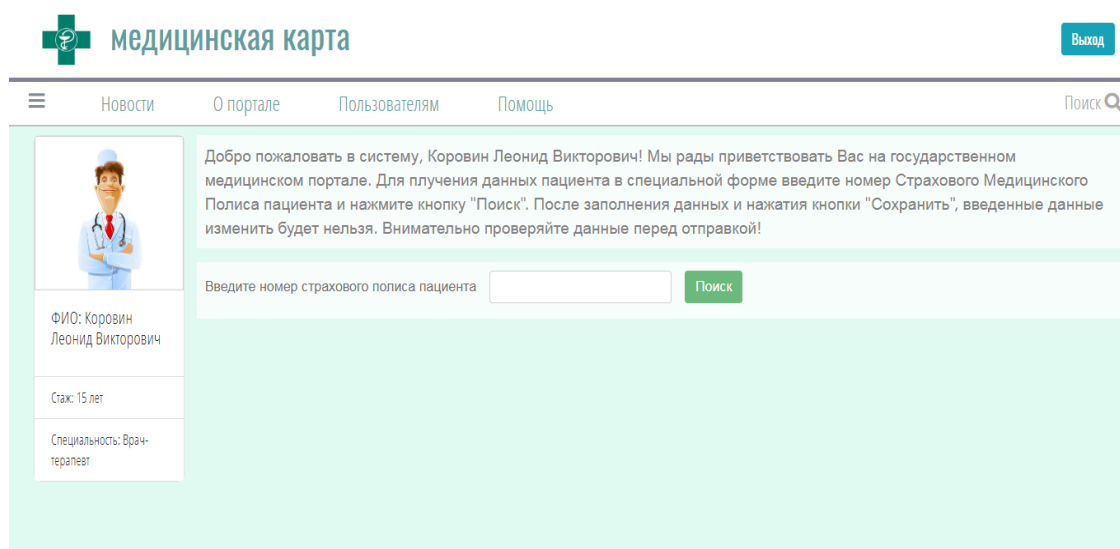
☐ Запомнить меня

☒ Я врач.

Вход

Рисунок 11 Авторизация медицинского специалиста

После успешной аутентификации сотрудник попадает в «Личный кабинет» (рис. 12), содержащий данные о его личности, специальности и опыте работы, где также имеется специальное поле ввода, с помощью которого производится поиск пациентов по номеру страхового полиса.



медицинская карта

Выход

Новости О портале Пользователям Помощь Поиск

Добро пожаловать в систему, Коровин Леонид Викторович! Мы рады приветствовать Вас на государственном медицинском портале. Для получения данных пациента в специальной форме введите номер Страхового Медицинского Полиса пациента и нажмите кнопку "Поиск". После заполнения данных и нажатия кнопки "Сохранить", введенные данные изменить будет нельзя. Внимательно проверяйте данные перед отправкой!

Введите номер страхового полиса пациента

Поиск


ФИО: Коровин Леонид Викторович

Стаж: 15 лет

Специальность: Врач-терапевт

Рисунок 12 Личный кабинет специалиста

Кнопка «Поиск» осуществляет поиск пациента с данным номером страхового медицинского полиса и перенаправляет медработника на страницу пользователя с его личными данными (рис. 13).



ФИО: Иванов Иван Иванович
Дата рождения: 1974-06-19

Страховой полис: 1000

Адрес: ул. Пушкина 32-279

После заполнения данных и нажатия кнопки "Сохранить", введенные данные изменить будет нельзя. Внимательно проверяйте данные перед отправкой!

Новая запись

#	Дата приема	Жалобы	Заболевание	Рекомендации к лечению	Врач	Справка	Приложения
1	2019-04-14	Тошнота, рвота, диарея, головокружение	Пищевое отравление	Не есть молочное, жареное, фрукты/овощи, пить Смекту 4 раза в день по 1 пакету	Коровин Леонид Викторович		
2	2019-04-14	Резкое ухудшение зрения, частые головокружения, плохой сон	Нарушение зрения	Капли Глазалан в теч. недели 2 раза в день, утром и вечером, ограничить телевизор/компьютер	Каримова Лилия Эдуардовна		
3	2019-03-18	боль в горле, насморк, слезотечение и резь в глазах (в начале болезни), головная боль, кашель, вялость.	Грипп	Лежать, спать, есть и отдыхать, горячий чай с лимоном и мед	Коровин Леонид Викторович		

Рисунок 13 Страница пациента

На странице пациента содержится вся необходимая информация о нем: ФИО, дата рождения, номер страхового медицинского полиса и адрес. Основным объектом данной страницы является таблица посещений пользователем врачей, в которой хранятся записи, содержащие информацию о дате посещения пациентом медицинского учреждения, его жалобах, диагнозах, врачах, проводивших консультацию, а также необходимые рекомендации для лечения. Врач может посмотреть данные об анализах пациента, а также справки, выданные пациенту другими специалистами.

Личный кабинет врачей оборудован кнопкой «Новая запись», при нажатии на которую открывается специальная форма для создания новой записи посещения пациента (рис. 14). Данная форма содержит поля «Жалобы», «Диагноз», «Рекомендация к лечению». Данные о личности пациента, дате посещения и враче, производившего консультацию, заполняются автоматически.

Пациент: Иванов Иван Иванович

Жалобы:

Диагноз:

Рекомендации к лечению

Сформировать справку

Выберите файл

Файл не выбран

Причина:

От(работа/учеба/занятия физ.культурой)

Дата начала

Дата окончания

Заккрыть

Сохранить

Рисунок 14 Форма добавления новой записи

Медицинские сотрудники также имеют возможность загрузить результаты анализов или данные об обследованиях пациента с помощью кнопки «Выбрать файл». Необходимые файлы могут быть загружены как с компьютера врача, так и с внешних носителей.

Если пациенту по каким-либо обстоятельствам необходимо получение справки, освобождающей его от работы/ учебы / занятий физической культуры, врач может заполнить данные, необходимые для создания и формирования справки: причина освобождения, вид деятельности, от которого освобождается пациент, а также сроки действия данной справки. Данные о пациенте, враче, дате получения справки заполняются автоматически. Если необходимость в получении справки отсутствует, медицинский работник может оставить данные поля ввода незаполненными, что никак не повлияет на корректность добавления новой записи пациента.

Кнопка «Сохранить» записывает введенную информацию в таблицу базы данных, предназначенной для хранения записей посещений всех пациентов. После нажатия данной кнопки, введенные данные не будут подлежать изменению.

Информация о предыдущих посещениях пациента не может быть изменена или удалена никем из медицинских сотрудников.


Для завершения работы в системе имеется кнопка завершения сеанса «Выход», располагающаяся в правом верхнем углу шапки.

5.2 Личный кабинет пользователя.

Аутентификация пользователя осуществляется по данным учетной записи, которая используется пользователем для доступа к личному кабинету на официальном государственном портале. Вход в «Личный кабинет» производится аналогично сотрудникам медицинского учреждения. Флажок «Я врач» должен быть отключен, иначе пользователю не удастся получить доступ к сервису. При вводе неверной информации пользователь получает сообщение об ошибке и при необходимости может повторить попытку авторизации.

После успешной авторизации пользователь попадает в свой личный кабинет (рис. 15), содержащий его личные данные, а также данные о посещениях медицинских учреждений. Интерфейс пользователя идентичен интерфейсу страницы пациентов у медицинских сотрудников. Пользователю предоставляется возможность просматривать историю своих посещений, содержащую жалобы, диагноз, дату посещения, ФИО лечащего врача, а также все необходимые рекомендации. Также пользователю доступны к просмотру данные о результатах медицинского обследования и справки, полученные им ранее (рис. 16).

Пользователь не может изменять, добавлять или удалять данные.



ФИО: Иванов Иван Иванович

Дата рождения: 1974-06-19

Страховой полис: 1000

Адрес: ул. Пушкина 32-279

В таблице показаны ваши обращения к врачу

Добро пожаловать, Иванов Иван Иванович

#	Дата приема	Жалобы	Заболевание	Рекомендации к лечению	Врач	Справка	Приложения
1	2019-04-14	Тошнота, рвота, диарея, головокружение	Пищевое отравление	Не есть молочное, жареное, фрукты/овощи, пить Смекту 4 раза в день по 1 пакету	Коровин Леонид Викторович		
2	2019-04-14	Резкое ухудшение зрения, частые головокружения, плохой сон	Нарушение зрения	Капли Глазалан в теч. недели 2 раза в день, утром и вечером, ограничить телевизор/компьютер	Каримова Лилия Эдуардовна		
3	2019-03-18	боль в горле, насморк, слезотечение и резь в глазах (в начале болезни), головная боль, кашель, вялость.	Грипп	Лежать, спать, есть и отдыхать, горячий чай с лимоном и мед	Коровин Леонид Викторович		
4	2019-03-15	Упал на лестнице 2.04.19. Боль в области поясницы, отдышка, нервный тик	Ушиб поясницы	1-5 день мазь Антиболь на место ушиба, 10 дней Мелоксикам 3 таблетки в день, покой,	Валеев Андрей		

Рисунок 15 Личный кабинет пользователя

ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ УПРАВЛЕНИЯ
ЗДРАВООХРАНЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОРОДА МОСКВЫ

"ГОРОДСКАЯ ПОЛИКЛИНИКА №00"

ИНДЕКС, Г. ГОРОД УЛ. УЛИЦА, Д. 1, СТ. 1.
ТЕЛ.: 8-012-345-67-89; 8-012-345-67-89

СПРАВКА № 31

Выдана: Иванов Иван Иванович


Дата рождения: 1974-06-19

О том, что он(она) освобождается от
От работы на ООО "АВТОВАЗ"

по причине:
Пищевое отравление

с 14.04.19 по 27.04.19

Врач: Коровин Леонид Викторович



Дата выдачи: 2019-04-14

Рисунок 16 Пример сгенерированной справки

ЗАКЛЮЧЕНИЕ

В ходе данной работы был разработан медицинский сервис «Электронная медицинская карта», в котором реализована возможность хранения данных о посещениях гражданами медицинских учреждений, хранение информации о личности граждан, состоянии их здоровья, а также сведений о результатах различных медицинских обследований, просмотр истории болезней, рекомендаций к лечению, справок и результатов анализов. Сервис оснащен удобным функционалом для комфортной работы пользователей всех возрастов, а также понятной системой авторизации. Приведено полное описание возможностей данной системы.

На сервисе «Электронная медицинская карта» реализована система защиты пользовательских данных от различных видов атак, таких как получение данных паролей пользователей, SQL-инъекции, просмотр трафика передачи данных. Был произведен ряд мер для создания надежной системы защиты, а также произведена проверка на уязвимость данной системы с помощью специальных программ.

Результаты данной работы могут помочь структурировать и упростить работу медицинских учреждений, а также организовать удобный доступ гражданам к информации о состоянии своего здоровья. Внедрение данной системы в работу медицинских учреждений предполагает перенос данных с ненадежных бумажных носителей в удобный для поиска и хранения, устойчивый к потерям и повреждениям электронный вид. Пользователи смогут дистанционно просматривать рекомендации к лечению, диагнозы и результаты анализов, не тратя без необходимости время на посещение медицинских учреждений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Основы использования сессий. URL:
<https://www.php.net/manual/ru/session.examples.basic.php>
2. Статья SQL инъекции. Проверка, взлом, защита. URL:
<https://habr.com/ru/post/130826/>.
3. Статья Внедрение SQL-кода. Википедия. URL:
https://ru.wikipedia.org/wiki/Внедрение_SQL-кода
4. Статья: Подключения и управление подключениями PDO. URL:
<https://www.php.net/manual/ru/pdo.connections.php>
5. Статья: PDO::prepare. URL: <https://www.php.net/manual/ru/pdo.prepare.php>
6. Статья: PDOStatement::execute. URL:
<https://www.php.net/manual/ru/pdostatement.execute.php>
7. Статья: Что такое SSL и зачем он нужен. URL:
<https://www.colocat.ru/texts/ssl.html>
8. Хрусталеv А.А, Кириченко А.В, HTML5 + CSS3. Основы современного WEB-дизайна, 2018. – 352 с.

ПРИЛОЖЕНИЕ

«index.php»

```
<?session_start();?>
<!DOCTYPE html>
<html>
<head>
<meta charset="urf-8">
<title>Медицинская карта</title>
<link rel="stylesheet" href="css/bootstrap/bootstrap.min.css">
<script src="js/bootstrap/bootstrap.min.js"></script>
<link rel="stylesheet" href="css/main.css" type="text/css">
<link rel="stylesheet" href="css/main_page.css" type="text/css">
<meta name="description" content="Сайт про котиков">
<meta name="keywords" content="котики, собачки">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1" />
<link href="img/2157b91a9977b19.ico" rel="shortcut icon" type="image/x-icon">
<!-- <link rel="stylesheet" href="css/fontawesome.min.css">-->
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-
fnnOCqbTlWIlj8LyTjo7m0UStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
<!-- Bootstrap CSS -->
<script src="http://travistidwell.com/jsencrypt/bin/jsencrypt.js"></script>
</head>
<body>
<div id="wrapper">
<div id="content">
<header>
<div id="logo">
<a href="http://medcard.ru/" title="На главную">

<h1 style="display: inline-block; margin-left: 10px; position: relative; top: -
5px;">медицинская карта</h1>
</a>
</div>
<div id="registr" >
<a href="https://www.gosuslugi.ru/" title="Войти">
<div class="btn btn-info">
ГОСУСЛУГИ.РФ
</div>
</a>
</div>
</header>
<nav>
<div id="menuShow"><i class="fas fa-bars"></i></div>
<div id="hideMenu">
<a href="">Новости</a>
<a href="">О портале</a>
<a href="">Пользователям</a>
<a href="">Помощь</a>
</div>
<div id="search">
<span>Поиск</span>
<i class="fas fa-search"></i>
</div>
<div id="mobilMenu">
<a href="">Новости</a>
```

```

<a href="">О портале</a>
<a href="">Пользователям</a>
<a href="">Помощь</a>
<hr>
<a href="">Регистрация</a>
<a href="">Войти</a>
</div>
</nav>
<div style="clear:both;"></div>
<div class="row row-margin-0" id="main_content_bg">
<div class="col-12 col-md-6 col-xl-8">
</div>
<div class="col-12 col-md-5 col-xl-3">
<div class="reg-in">

<form id="login_form" onsubmit="showMessage()" action="login.php"
method="post">
<div class="form-group">
<label for="exampleInputEmail1">Введите email</label>
<input name="login" type="text" class="form-control" id="email" aria-
describedby="emailHelp" placeholder="Enter email">
<small style="color:#dc3545;" id="emailHelp" class="form-text">Вы можете
войти только после регистрации на портале Госуслуги.рф</small>
</div>
<div class="form-group">
<label for="exampleInputPassword1">Введите пароль</label>
<input name="password" type="password" class="form-control" id="password"
placeholder="Password">
</div>
<div class="form-check">
<input type="checkbox" class="form-check-input" id="exampleCheck1">
<label class="form-check-label" for="exampleCheck1">Запомнить меня</label>
</div>
<div class="form-check">
<input class="form-check-input" type="checkbox" name="doctor" value="yes">
<label class="form-check-label" for="defaultCheck1">
Я врач.
</label>
</div>
<button type="submit" id="send_btn" class="btn btn-primary">Вход
</button>
</form>
</div>
</div>
</div>
</div>
<div id="site_name">
<span>Государственный медицинский портал</span>
</div>
<div id="clear"></div>
<div id="footer_menu">
<a href="" title="Узнать про рекламу">Реклама</a>
<a href="" title="Поддержать проект">Поддержать проект</a>
<a href="" title="Написать письмо">Обратная связь</a>
</div>
<div id="rights">
<a href="">Все права защищены &copy; <?=date('Y')?></a>
</div>
<div id="social">
<a href="" title="Группа Вк"><i class="fab fa-vk"></i></a>
<a href="" title="Facebook"><i class="fab fa-facebook-square"></i></a>
<a href="" title="Twitter"><i class="fab fa-twitter"></i></a>
</div>

```



```

</footer>
</div>
<!--jQuery-->
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></scri
pt>
<script>
$('#menuShow').click(function () {
if ($('#mobilMenu').is(':visible'))
$('#mobilMenu').hide();
else
$('#mobilMenu').show();
});
$(document).scroll (function () {
if ($(document).width() > 785){
if ($(document).scrollTop() > $('#header').height() + 10)
$('#nav').addClass('fixed');
else
$('#nav').removeClass('fixed');
}
});

window.onresize = function(event) {
$('#mobilMenu').hide();
};
//шифрование
function showMessage() {
var pub =
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDiXxKjoXywT8cOsXsAY8Qy99TvznFxxvQEf2Xrg
ddTBmFKBoilEio4CQF4VNNTqEF/HWvCcOhCKXNvko/uM0YrhxTQIGlUIxrlzJxTiznzhY3SZg6sDy
bykBMHU8n55PPwKskd6v34QvsuP8Lx1kOpvQtdpZT7AXNa1L1XYlmmFTwIDAQAB";
var crypt = new JSEncrypt();
crypt.setPublicKey(pub);
var data = $('#email').val();
$('#email').val(crypt.encrypt(data));
var data2 = $('#password').val();
$('#password').val(crypt.encrypt(data2));
}
</script>
</body>
</html>

```

«bd.php»

```

<?php
function print_res ($result){
    while ($row = $result->fetch_assoc()) {
        //print_r ($row);
        echo $row["name"]." ".$row["adress"];
        echo "<br/>";
    }
    echo "Кол-во записей = ".$result->num_rows."<br/>-----";
}
$host = 'localhost'; // адрес сервера
$database = 'medcard'; // имя базы данных
$user = 'root'; // имя пользователя
$password = ''; // пароль
// подключаемся к серверу
// $mysqli = new mysqli ($host, $user, $password, $database);
$mysqli = new PDO('mysql:host=localhost;dbname=medcard', $user, $password);
?>

```

«doctor.php»

```
<?
session_start();
if (!isset($_SESSION['DOCTOR_ID'])) {
header("location: index.php");
}
require "bd.php";
require "templates/head.php";
$strah = $_GET["strah"];
$user = $mysqli->prepare("SELECT * FROM user WHERE strah_polis = :strah");
$user->execute([':strah' => $strah]);
$result = $user->fetch(PDO::FETCH_ASSOC);
$table = $mysqli->prepare("SELECT journal.*,doctor.name FROM journal JOIN
doctor ON journal.id_doctor = doctor.id WHERE journal.id_user = :id_user
ORDER by journal.data_priema DESC");
$table->execute([':id_user' => $result['id']]);
$filee = $mysqli->prepare("SELECT PDF FROM journal WHERE id_user =
:id_user");
$filee->execute([':id_user' => $result['id']]);
?>
<body>
<div id="wrapper">
<div id="content">
<?require "templates/header.php"?>
<div class="row row-margin-0" id="profil_img">
<div class="col-3">
<div class="profile_block_left">
<div class="card" >

<div class="card-body">
<h5 class="card-title">ФИО: <?=$result["name"]?></h5>
<p class="card-text">Дата рождения: <?=$result["birth_date"]?></p>
</div>
<ul class="list-group list-group-flush">
<li class="list-group-item">Страховой полис: <?=$result["strah_polis"]?></li>
<li class="list-group-item">Адрес: <?=$result["adress"]?></li>
</ul>
</div>
</div>
</div>
<div class="col-9">
<div class="table_block">
<blockquote class="blockquote">
<p class="mb-0">После заполнения данных и нажатия кнопки "Сохранить",
введенные данные изменить будет нельзя. Внимательно проверяйте данные перед
отправкой!
</p>
<button type="button" class="btn btn-warning" data-toggle="modal" data-
target="#myModal">Новая запись</button>
</blockquote>
<table class="table table-hover">
<thead class="thead-dark">
<tr>
<th scope="col">#</th>
<th scope="col">Дата приема</th>
<th scope="col">Жалобы</th>
<th scope="col">Заболевание</th>
<th scope="col">Рекомендации к лечению</th>
<th scope="col">Врач</th>
<th scope="col">Справка</th>
<th scope="col">Приложения</th>
</tr>
</thead>
```

```

<tbody >
<? $i=1;
while ($row = $table->fetch(PDO::FETCH_ASSOC)) {>
<tr>
<th scope="row"><?=$i?></th>
<td><?=$row["data_priema"]?></td>
<td><?=$row["complaints"]?></td>
<td><?=$row["diagnoz"]?></td>
<td><?=$row["recommend"]?></td>
<td><?=$row["name"]?></td>
<td>
<? if(($row["spravka"])!=0){>
<a href="spravka_gen.php?id=<?=$row["id"]?>">

</a>
<?}>
</td>
<td>
<? if(($row["PDF"])!=NULL){>
<a href="http://medcard.ru/<?=$row["PDF"]?>">

</a>
<?}>
</td>
</tr>
<? $i = $i+1;;
?>
</tbody>
</table>
</div>
</div>
</div>
<div class="row row-margin-0" id="profil_img">
<div class="col-3">
</div>
<div class="col-9">
</div>
</div>
</div>
</div>
<!-- Модальное окно -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h4 class="modal-title">Пациент: <?=$result["name"]?></h4>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<form action = "send.php?act=addZapis" enctype="multipart/form-data"
method="post">
<div class="modal-body">
<label>Жалобы:</label>
<input type="hidden" name="id_user" value="<?=$result["id"]?>">
<input type="hidden" name="id_doctor" value="<?=$_SESSION['DOCTOR_ID']?>">
<input type="text" class="form-control" name="jalobi">
<label>Диагноз:</label>
<input type="text" class="form-control" name="diagnoz">
<label>Рекомендации к лечению:</label>
<input type="text" class="form-control" name="recommend">
<button type="button" class="btn btn-warning"
style="margin:10px;">Сформировать справку</button>
<input type="hidden" name="MAX_FILE_SIZE" value="30000000" />

```

```

<input type="file" class="btn btn-info" name="file" id="file"
title="Результаты анализов" style="margin:10px;">
<label>Причина:</label>
<input type="text" class="form-control" name="reason">
<label>От (работа/учеба/занятия физ.культурой)</label>
<input type="text" class="form-control" name="ot">
<div class="row" style="margin-top: 10px">
<div class="col">
<input type="date" class="form-control" placeholder="Дата начала" name =
"date_start">
</div>
<div class="col">
<input type="date" class="form-control" placeholder="Дата окончания" name =
"date_end">
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal"
>Закрыть</button>
<button type="submit" class="btn btn-primary" >Сохранить</button>
</div>
</form>
</div><!-- /.модальное окно-Содержание -->
</div><!-- /.модальное окно-диалог -->
</div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.0.0/jquery.min.js"
></script>
<script src="js/bootstrap/bootstrap.min.js"></script>
</body>

```

«login.php»

```

<?
session_start();
include "bd.php";
$key =<<<SOMEDATA777
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDiXxKjoXyWT8cOsXsAY8Qy99TvznFxxvQEf2XrgddTBmFKBOile
io4CQF4VNNtqEF/HWvCcOhCKXNvko/uM0YrhxTQIGlUIxrlzJxTiznzhY3SZg6sD
ybykBMHU8n55PPwKskd6v34QvsuP8LxlKOpvQtdpZT7AXNa1L1XYlmmFTwIDAQAB
AoGAQXnWX1X7RtQMc4eKWFaDDWr5wFMqJQfSJ3A0RnBolaCFXLOB9D1PTf9oNyIM
45bQ3UzEg6uX1S1+vOdhfx2s2Ylee3UwMqSUUpWDvCVyBBDrfiH+OMIqhvA/QZqsc
Eyw9kx0YeJmmSrK1bbiFkE8khzCZaKjGuqARIFI8jD27/yECQQD6ZAeEVH41nPKH
+Bo8TUSawcPmOQMdz6ZAzDPlbK1T8M4NU7RQPvFCVgEpPXX1Ny6mMwRcy+z2Qhp+
vRjSxsx9AkEA53FLWpW/EEycNE9C0vpI8lnfTABKIKHCYHLeGZ6AxFq1c14eBq97
5NL18OJ7yeLP9x4Hw+fXQJufIMi9P5unewJBAJTAVE7j009yfAsW888bQESIFG9z
M0zEjcolBieoFDhP+LmmHpsEsn+sRGhRoPaZf9cwu8i9RXt07hqZEGQ3vykCQAWo
TVE/i9YYKVFwBqolmBbkf4LaFvXJPnkhFbDGoRsrpHfXeyBqtbbqYNSq4PpJmyvKd
d08gobBgnXktrwKZCXUCQQDDPbrAS06e4keB+Csn7rUjg9GDzgFm+f8XcTWPufFD
/WhBE0A/4Fbb9itPzb2vgFLHs5KVXPY+gmv753MMqNRJ
-----END RSA PRIVATE KEY-----
SOMEDATA777;
$login = $_POST["login"];
$password = $_POST["password"];
$pk = openssl_get_privatekey($key);
openssl_private_decrypt(base64_decode($login), $login, $pk);
openssl_private_decrypt(base64_decode($password), $password, $pk);

if(isset($_POST['doctor']) &&
$_POST['doctor'] == 'yes')
{

```

```

$sql = $mysqli->prepare("SELECT id,name,password FROM doctor WHERE login =
:login ");
$sql->execute([':login' => $login]);
if ($sql->rowCount() == 0){
echo "не сущ";
header("location: index.php");
}
else{
$result = $sql->fetch(PDO::FETCH_ASSOC);
if (md5($password) == $result["password"]){
$_SESSION['DOCTOR_ID']=true;
$_SESSION['DOCTOR_ID']=$result["id"];
$_SESSION['DOCTOR_NAME']=$result["name"];
header("location: profil_doc.php");
}else{
echo "Пароль неверный ыыы.";
}
}
}
else
{
$sql = $mysqli->prepare("SELECT id,name,password FROM user WHERE login =
:login");
$sql->execute([':login' => $login]);
if ($sql->rowCount() == 0){
echo "не сущ";
header("location: index.php");
}
else{
$result = $sql->fetch(PDO::FETCH_ASSOC);
if (md5($password) == $result["password"]){
$_SESSION['USER_ID']=true;
$_SESSION['USER_ID']=$result["id"];
$_SESSION['USER_NAME']=$result["name"];
header("location: profil.php");
}else{
echo "Пароль неверный.";
}
}
}
?>

```

«profil.php»

```

<?
session_start();
if (!isset($_SESSION['USER_ID'])){
header("location: index.php");
}
require "bd.php";
require "templates/head.php";
$id_user = $_SESSION['USER_ID'];
$sql = $mysqli->prepare("SELECT * FROM user WHERE id = :id_user ");
$sql->execute([':id_user' => $id_user]);
$result = $sql->fetch(PDO::FETCH_ASSOC);
$table= $mysqli->prepare("SELECT journal.*,doctor.name FROM journal JOIN
doctor ON journal.id_doctor = doctor.id WHERE journal.id_user = :id_user
ORDER by journal.data_priema DESC");
$table->execute([':id_user' => $id_user]);
?>
<body>
<div id="wrapper">
<div id="content">

```

```

<?require "templates/header.php"?>
<div class="row row-margin-0" id="profil_img">
<div class="col-3">
<div class="profile_block_left">
<div class="card" >

<div class="card-body">
<h5 class="card-title">ФИО: <?=$_SESSION['USER_NAME'];?></h5>
<p class="card-text">Дата рождения: <?=$result["birth_date"]?></p>
</div>
<ul class="list-group list-group-flush">
<li class="list-group-item">Страховой полис: <?=$result["strah_polis"]?></li>
<li class="list-group-item">Адрес: <?=$result["adress"]?></li>
</ul>
</div>
</div>
</div>
<div class="col-9">
<div class="table_block ">
В таблице показаны ваши обращения к врачу
<div class="alert alert-primary" role="alert">
Добро пожаловать, <?=$_SESSION['USER_NAME']?>
</div>
<table class="table table-hover">
<thead class="thead-dark">
<tr>
<th scope="col">#</th>
<th scope="col">Дата приема</th>
<th scope="col">Жалобы</th>
<th scope="col">Заболевание</th>
<th scope="col">Рекомендации к лечению</th>
<th scope="col">Врач</th>
<th scope="col">Справка</th>
<th scope="col">Приложения</th>
</tr>
</thead>
<tbody >
<? $i=1;
while ($row = $table->fetch(PDO::FETCH_ASSOC)) {?>
<tr>
<th scope="row"><?=$i?></th>
<td><?=$row["data_priema"]?></td>
<td><?=$row["complaints"]?></td>
<td><?=$row["diagnoz"]?></td>
<td><?=$row["recommend"]?></td>
<td><?=$row["name"]?></td>
<td><? if (($row["spravka"]) !=0) {?>
<a href="spravka_gen.php?id=<?=$row["id"]?>">

</a>
<?}?>
</td>
<td>
<? if (($row["PDF"]) !=NULL) {?>
<a href="http://medcard.ru/<?=$row["PDF"]?>">

</a>
<?}?>
</td>
</tr>
<? $i = $i+1;};
?>
</tbody>
</table>
</div>

```



```

<button style="background-color:#6cb97d; border-color:#6cb97d;" type="submit"
class="btn btn-primary mb-2">Поиск</button>
</form>
</div>
</div>
<div class="row row-margin-0" id="profil_img">
<div class="col-3">
</div>
<div class="col-9">
</div>
</div>
</div>
</div>
</div>
</body>

```

«send.php»

```

<?
session_start();
include "bd.php";
$act = $_GET['act'];
switch ($act){
case 'addZapis':
$jaloba = $_POST['jalobi'];
$user_id = $_POST['id_user'];
$doctor_id = $_POST['id_doctor'];
$diagnoz = $_POST['diagnoz'];
$rec = $_POST['recommend'];
$reason = $_POST['reason'];
$ot = $_POST['ot'];
$date_start = $_POST['date_start'];
$date_end = $_POST['date_end'];
/*Загрузка файла*/
if ($_FILES['file']['size']!=0){
$uploadaddir = 'files/vlojeniya/';
$type_file = pathinfo($_FILES['file']['name'], PATHINFO_EXTENSION);
$current_date = date("Y-m-d_H-i-s");
$new_name=$user_id."_".$current_date."_".$type_file;
$uploadfile = $uploadaddir . $new_name;
if (!move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile)) {
echo "Файл не загружен, повторите попытку!\n";
exit();
}
}else{
$uploadfile=NULL;
}
/*загрузка данных справки*/
$name_doctor = $mysqli->prepare("SELECT name FROM doctor WHERE id =
:doctor_id");
$name_doctor->execute([':doctor_id' => $doctor_id]);
$name_doctor = $name_doctor->fetch(PDO::FETCH_ASSOC) ["name"];
if (!empty($reason)){
$sql2 = $mysqli->prepare("INSERT INTO spravka (ot, reason, date_start,
date_end, name_doctor) VALUES (:ot, :reason,:date_start, :date_end,
:name_doctor)");
$sql2->execute([':ot' => $ot, ':reason' => $reason, ':date_start' =>
$date_start, ':date_end' => $date_end, ':name_doctor' => $name_doctor ]);
}
/*добавление записи в журнал*/
if (!empty($reason)){
$sql = $mysqli->prepare("INSERT INTO journal (id_user, id_doctor, complaints,
diagnoz,data_priema, recommend, PDF, spravka) VALUES (:user_id, :doctor_id,
:jaloba, :diagnoz, CURRENT_DATE(), :rec, :uploadfile, :last_id)");

```



```

$sql->execute([':user_id' => $user_id, ':doctor_id' => $doctor_id, ':jaloba'
=> $jaloba, ':diagnoz' => $diagnoz, ':rec' => $rec, ':uploadfile' =>
$uploadfile, ':last_id' => $mysqli->lastInsertId() ]]);
}
else {
$sql = $mysqli->prepare("INSERT INTO journal (id_user, id_doctor, complaints,
diagnoz,data_priema, recommend,PDF) VALUES (:user_id, :doctor_id, :jaloba,
:diagnoz, CURRENT_DATE(),:rec, :uploadfile)");
$pdo = $sql->execute([':user_id' => $user_id, ':doctor_id' => $doctor_id,
':jaloba' => $jaloba, ':diagnoz' => $diagnoz, ':rec' => $rec, ':uploadfile'
=> $uploadfile ]]);

}
header("location: {$_SERVER['HTTP_REFERER']}");
break;
case 'logout':
unset($_SESSION['DOCTOR_ID']);
unset($_SESSION['DOCTOR_NAME']);
unset($_SESSION['USER_ID']);
unset($_SESSION['USER_NAME']);
header("location: index.php");
break;
}

```

«Spravka_gen.php»

```

<?
session_start();
require "bd.php";
header("Content-type: image/png");
$id_zapisi = $_GET['id'];
$sql = $mysqli->prepare("SELECT * FROM journal WHERE id = :id_zapisi ");
$sql->execute([':id_zapisi' => $id_zapisi]);
$result = $sql->fetch(PDO::FETCH_ASSOC);
$sql2 = $mysqli->prepare("SELECT * FROM spravka WHERE id = :id ");
$sql2->execute([':id' => $result['spravka']]);
$result2 = $sql2->fetch(PDO::FETCH_ASSOC);
$user = $mysqli->prepare("SELECT * FROM user WHERE id = :id_us");
$user->execute([':id_us' => $result['id_user']]);
$result3 = $user->fetch(PDO::FETCH_ASSOC);
if (!isset($_SESSION['DOCTOR_ID'])) {
if($_SESSION['USER_ID']!= $result['id_user'])
header("location: index.php");
}
$fio= $result3['name'];
$date=$result3['birth_date'];
$ot = $result2['ot'];
$reason = $result2['reason'];
$c = $result2['date_start'];
$po = $result2['date_end'];
$doctor = $result2['name_doctor'];
$date_take = $result['data_priema'];
$im = imagecreatefrompng("img/spravka2.png");
$black = imagecolorallocate($im, 20, 20, 20);
$text_size=18;
$font = 'arial.ttf';
imagefttext($im, $text_size, 0, 648, 70, $black, $font, $id_zapisi);
imagefttext($im, $text_size, 0, 210, 185, $black, $font, $fio);
imagefttext($im, $text_size, 0, 306, 235, $black, $font, $date);
imagefttext($im, $text_size, 0, 108, 336, $black, $font, $ot);
imagefttext($im, $text_size, 0, 108, 428, $black, $font, $reason);
imagefttext($im, $text_size, 0, 125, 490, $black, $font, $c);
imagefttext($im, $text_size, 0, 410, 490, $black, $font, $po);
imagefttext($im, $text_size, 0, 192, 576, $black, $font, $doctor);

```

```
imagetfttext($im, $text_size, 0, 778, 672, $black, $font, $date_take);  
imagepng($im);  
imagedestroy($im);
```

```
?>
```