Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет инженерно-экономический

Кафедра экономической информатики

Дисциплина «Распределённые системы обработки информации»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  ассистент кафедры ЭИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.П. Лыщик |
|  | \_\_\_.\_\_\_\_.2023 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

**«Разработка системы автоматизации работы санатория»**

|  |  |
| --- | --- |
|  | Выполнил студент группы 074002  Кленицкий Александр Анатольевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовой проект представлен на проверку \_\_\_.\_\_\_\_.2023  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2023

Содержание

[Введение 3](#_Toc484179613)

[1 Требования к Web-приложению 5](#_Toc484179614)

[1.1 Характер и анализ деятельности предприятия 5](#_Toc484179615)

[1.2 Общие требования к Web-приложению 6](#_Toc484179616)

[1.3 Требования к структуре Web-приложения 6](#_Toc484179617)

[1.4 Требования к дизайну Web-приложения 9](#_Toc484179618)

[1.5 Описание работы сервиса Web-приложения 13](#_Toc484179619)

[2 Обзор технологий, используемых для разработки 15](#_Toc484179620)

[и продвижения WEB-приложений 15](#_Toc484179621)

[2.1 Программное обеспечение для разработки дизайна 15](#_Toc484179624)

[Web-приложений 15](#_Toc484179625)

[2.2 Технологии разработки Web-приложений 20](#_Toc484179626)

[2.3 Технологии продвижения Web-приложений 24](#_Toc484179627)

[2.4 Выбор технологий для разработки Web-приложения 28](#_Toc484179628)

[3 Проектирование Web-приложения 30](#_Toc484179629)

[3.1 Разработка дизайна Web-приложения 30](#_Toc484179630)

[3.2 Разработка клиентской части Web-приложения 35](#_Toc484179631)

[3.3 Разработка серверной части Web-приложения 40](#_Toc484179632)

# Введение

С каждым днём число интернет-пользователей растёт, и всё больше людей получают необходимую информацию из сети интернет. Для каждого предприятия важно привлечь как можно больше потенциальных клиентов. Web-приложение – это отличный способ для решения этой задачи; к тому же, в Web-приложении можно разместить всю необходимую информацию о предприятии и его услугах, что поможет избежать большого числа звонков и направить освободившиеся время на развитие компании.

Web-приложение – это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – Web-сервер. Логика Web-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по сети.

Целью работы является разработка Web-приложения для стоматологической клиники.

Для реализации поставленной цели необходимо решить следующие задачи:

– дать характеристику деятельности предприятия и выявить, какие процессы деятельности предприятия нуждаются в автоматизации;

– определить требования к разрабатываемому продукту;

– изучить и выбрать технологии для разработки;

– описать процесс разработки;

– опубликовать и провести поисковую оптимизацию разработанного продукта;

– описать технико-экономическое обоснование процесса разработки.

Пояснительная записка выпускной квалификационной работы состоит из четырёх разделов:

1. требования к Web-приложению;
2. обзор технологий, используемых для разработки и продвижения Web-приложений;
3. проектирование Web-приложения;
4. внедрение Web-приложения.

В первом разделе дана характеристика и анализ деятельности предприятия, определены требования к структуре, дизайну, а также описана работа сервиса Web-приложения. Во втором разделе рассмотрены и выбраны технологии для разработки и продвижения приложения, а также рассмотрено программное обеспечение для разработки дизайна. В третьем разделе описан процесс разработки дизайна, клиентской и серверной части Web-приложения. В четвертом разделе описан процесс публикации, произведено продвижение, а также описано технико-экономическое обоснование разработки приложения.

# 1 Требования к Web-приложению

## **Характер и анализ деятельности предприятия**

Основной целью стоматологической клиники является полное и качественное удовлетворение потребностей пациентов в стоматологических услугах и получение прибыли в результате своей производственной деятельности.

Основными процессами деятельности стоматологической клиники являются:

– запись пациентов;

– приём и оказание платных стоматологических услуг (лечение, протезирование, хирургия и т.д.) пациентам;

– оформление медицинских карточек пациентов;

– повторная запись на приём.

Автоматизация данных процессов стоматологической клиники влияет на качество и скорость их выполнения, а, следовательно, на качество предоставляемых стоматологических услуг и успешность предприятия в целом.

Основными целями внедрения Web-приложения в стоматологическую клинику являются:

– повышение качества предоставления стоматологических услуг;

– планирование работы персонала, приём пациентов;

– унификация и систематизации данных о пациентах, персонале;

– анализ и мониторинг деятельности клиники её руководством для осуществления дальнейшего перспективного планирования и повышения эффективности работы.

## **Общие требования к Web-приложению**

Предметом разработки является Web-приложение стоматологической клиники.

Данное Web-приложение должно реализовывать следующие задачи:

– предоставление информации о стоматологической клинике;

– предоставление возможности записи на приём;

– предоставление пациентам просмотра своих записей на приём;

– предоставление персоналу возможности просмотра записи пациентов;

– предоставление персоналу медицинских карточек пациентов.

Для достижения поставленных задач необходимо реализовать следующие функции:

– добавление, удаление, редактирование записи пациентов;

– вход и регистрация персонала и пациентов в Web-приложение;

– формирование отчёта по медицинским карточкам пациентов.

Web-приложение должно быть ориентировано на совершеннолетних русскоговорящих пользователей.

Web-приложение должно хорошо отображаться на компьютерах, ноутбуках, планшетах и телефонах с различным разрешением экрана.

Данное Web-приложение рекомендовано просматривать в наиболее распространенных браузерах: Internet Explorer (версии 8.0 и выше), Opera (40-я версия и выше), Google Chrome (54-я версия и выше), Firefox (48-я версия и выше).

## **Требования к структуре Web-приложения**

Web-приложение должно состоять из следующих разделов:

– о клинике;

– врачи;

– услуги;

– прайс;

– записаться на приём;

– партнёры;

– контакты.

Web-приложение должно обеспечивать навигацию по всем доступным пользователю ресурсам и отображать соответствующую информацию. Для навигации должно использоваться меню, которое представляет собой текстовый блок (список гиперссылок) в верхней части страницы.

Для разделов, содержащих подразделы, должно быть предусмотрено выпадающее подменю. Взаимосвязь между разделами и подразделами представлена на рисунке 1.1.

Пользователей Web-приложения можно разделить на четыре группы в соответствии с правами доступа:

1. группа – посетители;
2. группа – зарегистрированный пользователь;
3. группа – сотрудники;
4. группа – администратор.

Для зарегистрированных пользователей раздел «Записаться на прием» должен быть изменен на «Запись на прием», в котором будут содержаться подразделы «Моя запись», «Записаться на прием». Для сотрудников клиники и администратора Web-приложение должно содержать разделы «Запись пациентов», «Медицинская карта». При этом раздел «Запись пациентов» для сотрудников клиники должен содержать подразделы «Запись на сегодня», «Запись на завтра», «Запись на неделю», «Записать на прием», а для администратора – «Запись на сегодня», «Запись на завтра», «Запись на неделю», «Запись на месяц», «Запись за все время», «Записать на прием». Кроме того, раздел «Врачи» для администратора должен содержать подразделы «Врачи», «Редактировать информацию о врачах», «Добавить врача».



Рисунок 1.1 – Карта Web-приложения

Таблица 1.1 – Группы пользователей

|  |  |
| --- | --- |
| Группа пользователей | Права доступа |
| Посетитель | – просмотр общедоступной части Web-приложения;  – записаться на приём |
| Зарегистрированный пользователь | – просмотр общедоступной части Web-приложения;  – записаться на приём;  – просмотр своих записей на приём |
| Сотрудники | – просмотр общедоступной части Web-приложения;  – просмотр, редактирование и удаление записи пациентов данного сотрудника;  – заполнение медицинских карточек пациентов и составление отчёта |
| Администратор | – просмотр общедоступной части Web-приложения;  – просмотр, редактирование и удаление записей пациентов;  – просмотр, редактирование и удаление информации о сотрудниках клиники;  – заполнение медицинских карточек пациентов и составление отчёта |

Доступ к личному кабинету авторизованных пользователей, сотрудников и администратора должен осуществляться с использованием уникального логина и пароля.

## **Описание работы сервиса Web-приложения**

Рассмотрим взаимодействие различных групп пользователей с Web-приложением.

В данном Web-приложении предусмотрены четыре группы пользователей – посетитель, зарегистрированный пользователь, сотрудники и администратор. Каждая группа пользователей сначала попадает на главную страницу, а затем переходит в различные разделы и подразделы меню.

Посетитель может посмотреть все разделы и подразделы меню, записаться на приём и зарегистрироваться в Web-приложении. Зарегистрированный пользователь может посмотреть все разделы и подразделы меню, авторизоваться, записаться на приём, а также посмотреть свои ближайшие записи на приём. Сотрудники клиники могут посмотреть все разделы и подразделы меню, авторизоваться, посмотреть, отредактировать и удалить ближайшие записи пациентов, а также заполнить медицинские карты пациентов и составить отчёт. Администратор может посмотреть все разделы и подразделы меню, авторизоваться, посмотреть, отредактировать и удалить запись пациентов любого сотрудника, заполнить медицинские карты пациентов и составить отчёт, а также добавить, отредактировать и удалить информацию о сотрудниках.

На рисунке 1.7 представлена диаграмма вариантов использования Web-приложения.



Рисунок 1.7 – Диаграмма вариантов использования

# Обзор технологий, используемых для разработки

# и продвижения WEB-приложений



## **Программное обеспечение для разработки дизайна**

## **Web-приложений**

Программное обеспечение для разработки дизайна Web-приложений делятся на две группы – программное обеспечение растровой графики и программное обеспечение векторной графики (таблица 2.1).

Таблица 2.1 – Программное обеспечение для разработки дизайна

|  |  |
| --- | --- |
| Вид программного обеспечения | Примеры |
| Программное обеспечение для растровой графики | Adobe Photoshop, GIMP |
| Программное обеспечение для векторной графики | CorelDRAW, Adobe Illustrator, Inkscape |

Adobe Photoshop – многофункциональный графический редактор для работы с растровыми изображениями. Сферами применения Adobe Photoshop являются:

– дизайн;

– обработка фото;

– полиграфия.

Основные возможности Adobe Photoshop:

– комплексная работа с изображениями, применение любых видов эффектов;

– быстрый доступ ко всем необходимым инструментам;

– поддержка многих графических форматов;

– работа с 3D;

– усовершенствованный интерфейс;

– возможность редактирования сразу нескольких слоев;

– специализированные фильтры для фотографов;

– усовершенствованная работа с текстом. Если в старых версиях Photoshop компания Adobe не придавала особого значения функции наложения текста, то теперь имеется возможность придать 3D-эффект обычному тексту, выбрать стиль абзацев и символов.

Преимущества и недостатки Adobe Photoshop представлены в таблице 2.2.

Таблица 2.2 – Преимущества и недостатки Adobe Photoshop

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Поддержка всех основных форматов изображений | Трудный для освоения начинающими |
| Огромный функционал как для редактирования изображений, так и для создания своих собственных | Высокая цена |
| Расширяемость при помощи плагинов |  |
| Бесплатная онлайн версия |  |

GIMP – программа для создания и обработки растровой графики и частичной поддержкой работы с векторной графикой. Типичные задачи, которые можно решать при помощи GIMP, включают в себя:

– создание графики и логотипов;

– масштабирование и кадрирование фотографий;

– раскраска, комбинирование изображений с использованием слоёв;

– ретуширование и преобразования изображений в различные форматы.

Особенности GIMP:

– понятный интерфейс;

– поддержка русского языка, благодаря чему легко разобраться во всех тонкостях программы;

– наличие всплывающих подсказок возле каждого инструмента, что также облегчает работу с программой;

– мощный функционал, с которым не сможет сравниться ни один из существующих бесплатных редакторов;

– поддержка разнообразных плагинов;

– наличие большого количества кистей;

– поддержка работы со слоями.

Преимущества и недостатки GIMP представлены в таблице 2.3.

Таблица 2.3 – Преимущества и недостатки GIMP

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Позволяет создавать оригинальные изображения | Нет полноценной поддержки цветовых моделей |
| Позволяет автоматизировать выполнение повторяющихся действий | Нет поддержки режима 16 и более разрядов на цветовой канал |
| Расширяемость при помощи плагинов | Нет процедурных (корректирующих) слоёв и эффектов (стилей) слоёв |
| Бесплатная лицензия |  |

CorelDRAW – это программа для создания и работы с графическими документами, выполненными в формате векторной графики. CorelDRAW в основном используется в оперативной полиграфии и в рекламных компаниях, использующих полотерную резку (т.е. создание логотипов, визиток, баннеров, и другой печатной продукции). Также этот продукт используется в Web-дизайне. Особенностями CorelDRAW являются:

– понятный интерфейс;

– все рабочие инструменты тесно связаны между собой, что позволяет эффективно оптимизировать рабочий процесс даже без тесной привязки к мышке;

– уникальные фильтры программы;

– в качестве линзы можно использовать любую векторную форму, в том числе и текст;

– допускается копирование только того участка, на который нацелена линза, с последующим его редактированием;

– огромная база шрифтов.

Преимущества и недостатки использования CorelDRAW [5] представлены в таблице 2.4.

Таблица 2.4 – Преимущества и недостатки CorelDRAW

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Увеличение масштаба без потери качества изображения | Сложность экспорта растрового формата в векторный, особенно если графика не черно-белая, а цветная |
| Небольшой размер векторного файла по сравнению с растровым | Интерфейс перегружен командными кнопками, что мешает отделить главное от второстепенного |
| Прекрасное качество полиграфической печати | Нет инструментов деловой графики, предоставляющих возможность быстро и просто создавать, например, графики и диаграммы |
| Возможность редактирования каждого элемента изображения в отдельности |  |
| Хорошая совместимость с другими программами |  |

Adobe Illustrator – векторный графический редактор, который будет полезен всем, чьей профессиональной деятельностью (или хобби) является рисование и оформление во всех их видах [25]. Это может быть разработка Web-страниц, оформление рекламы, полиграфия или просто рисование. К наиболее распространённым особенностям Adobe Illustrator относятся [29]:

– высокая точность работы с использованием пиксельной сетки;

– эффекты, четкие при любом разрешении (тени, текстуры, размытия и т.д.);

– русскоязычный интерфейс с возможностью настройки;

– создание 3D-эффектов;

– удобное построение красивых графиков;

– поддержка всех популярных графических форматов.

Преимущества и недостатки использования Adobe Illustrator [29] представлены в таблице 2.5.

Таблица 2.5 – Преимущества и недостатки Adobe Illustrator

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Интуитивный и понятный интерфейс | Сложность экспорта растрового формата в векторный |
| Возможность создания и совместной работы нескольких рабочих областей | Высокая цена |
| Обширный набор возможностей по работе с текстом | Нет огромного количества общедоступных наработок, разнообразных надстроек и шаблонов |
| Богатая библиотека инструментов для рисования |  |
| Возможность создания файлов видеоклипа flash, SVG, SVGZ форматов |  |

Inkscape – свободно распространяемый векторный графический редактор, удобен для создания как художественных, так и технических иллюстраций [22]. Сферами применения Inkscape являются:

– дизайн;

– создание технических иллюстраций;

– создание схем и планов.

Особенностями Inkscape являются [22]:

– поддержка большого количества форматов, как векторных, так и растровых;

– различные примитивы для создания изображений;

– векторизация растровых изображений;

– создание графиков и диаграмм;

– множество операций над контурами.

Преимущества и недостатки Inkscape [22] представлены в таблице 2.6.

Таблица 2.6 – Преимущества и недостатки Inkscape

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Поддержка большинства основных форматов изображений | Нестабильная работа некоторых инструментов и плагинов |
| Интуитивный и понятный интерфейс | Редактирование и сохранение только одной страницы PDF-документа |
| Расширяемость при помощи плагинов | Высокая нагрузка на ПК |
| Возможность векторизации растровых изображений |  |

Таким образом, рассмотрены основные назначения, преимущества и недостатки программного обеспечения для разработки дизайна Web-приложений.

## **Технологии разработки Web-приложений**

Технологии разработки Web-приложений делятся на две группы (таблица 2.7).

Таблица 2.7 – Технологии разработки Web-приложений

|  |  |
| --- | --- |
| Виды технологий | Примеры |
| Технологии, используемые на стороне клиента | HTML, CSS, JavaScript |
| Технологии, используемые на стороне сервера | Python, PHP, MySQL, PostgreSQL |

HTML – это язык разметки документа, описывающий форму отображения информации на экране компьютера. Просматривать HTML-документы можно с помощью специальных программ, которые называют Web-браузерами. Web-браузеры отображают документы с форматированием, выполненным на основе исходного кода, описывающего структуру документа.

Результат интерпретации HTML-документа, отображаемый в окне Web-браузера, называется Web-страницей. В отличие от HTML-документа, Web-страница может содержать не только текст, но и графику, видео, звуковое сопровождение, может реагировать на действия пользователя и т. д. Кроме того, Web-страница может быть результатом интерпретации сразу нескольких HTML-документов. Документы в формате HTML имеют расширение html или htm.

Весь текст HTML-документа расположен между тегами <html> и </html>. HTML-документ состоит из двух разделов – заголовка (между тегами <head> и </head>) и содержательной части (между тегами <body> и </body>).

XHTML – это семейство языков разметки Web-страниц на основе XML, повторяющих и расширяющих возможности HTML 5. Главное отличие XHTML от HTML заключается в обработке документа. Документы XHTML обрабатываются своим модулем (парсером) аналогично документам XML. В процессе этой обработки ошибки, допущенные разработчиками, не исправляются.

CSS – это каскадные таблицы стилей, которые позволяют существенно расширить возможности языка HTML за счёт более гибкого управления форматированием Web-страницы [26]. Применение стилей позволяет задавать точные характеристики практически всех элементов Web-страницы, а это значит, что можно точно контролировать внешний вид Web-страницы в окне Web-браузера.

Задать стиль можно тремя способами: встроить определение стиля в тег, встроить определения стилей в заголовок HTML-документа или вынести таблицу стилей в отдельный файл. Файл с таблицей стилей обычно имеет расширение css и может редактироваться любым текстовым редактором.

JavaScript – это язык программирования, позволяющий сделать Web-страницу интерактивной. Последовательность инструкций (называемая программой, скриптом или сценарием) выполняется интерпретатором, встроенным в обычный Web-браузер. Иными словами, код программы внедряется в HTML-документ и выполняется на стороне клиента. Для выполнения программы даже не нужно перезагружать Web-страницу. Все программы выполняются в результате возникновения какого-то события.

Основными задачами JavaScript являются:

* осуществление взаимодействий с пользователем в процессе его действий на страницах сайта;
* размещение математических вычислений на странице сайта;
* размещение элементов, которые двигаются, всплывают и проводят вычислительные операции.

PHP – язык программирования, выполняемый на стороне сервера. В отличие от языка JavaScript PHP не зависит от программного обеспечения клиента и поэтому будет выполнен в любом случае.

Последовательность инструкций (называемая программой или скриптом) выполняется интерпретатором языка PHP. Код программы может внедряться в HTML-код. Эта возможность отличает PHP от других языков, используемых в интернете, например, от языка Perl. Обработка PHP-кода производится на сервере до того, как страница будет передана Web-браузеру. В итоге Web-браузер получит обычный HTML-код или другой вывод.

Язык PHP используется для создания сайтов, GUI-приложений (графический интерфейс пользователя), хотя и не получил в данном направлении такого широкого распространения.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Язык программирования Python отличается сильной типизацией. Сильная типизация отличается тем, что язык не позволяет смешивать в выражениях различные типы и не выполняет автоматические неявные преобразования, например, нельзя вычесть из строки множество. В языке Python используется динамическая типизация – все типы выясняются уже во время выполнения программы [20].

Объявление переменных необязательно, названия восприимчивы к регистру (var и VAR – две разные переменные). Язык Python объектно-ориентирован, все в языке является объектом.

MySQL – одна из самых популярных и самых распространенных систем управления базами данных в интернете [13]. Она не предназначена для работы с большими объемами информации, но её применение идеально подходит как для небольших, так и для достаточно крупных интернет-сайтов.

MySQL отличатся хорошей скоростью работы, надежностью и гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP. Немаловажным фактором является ее бесплатный доступ.

PostgreSQL – это объектно-реляционная система управления базами данных, работающая как клиент-серверная система [6]. PostgreSQL ставит перед собой задачу полного соответствия как стандартам SQL, так и международным стандартам ANSI/ISO.

Главным отличием PostgreSQL от других систем управления базами данных является интеграция объектно-ориентированной и/или реляционной моделей. Несмотря на то, что PostgreSQL не настолько популярна, как MySQL, существует множество приложений сторонних производителей, значительно облегчающих работу с этой чрезвычайно мощной СУБД.

Таким образом, рассмотрены основные особенности технологий разработки Web-приложений.

## **Разработка серверной части Web-приложения**

Для разработки серверной части использованы технологии PHP, MySQL. Архитектура разрабатываемого Web-приложения является трехуровневой. Трёхуровневая архитектура предполагает наличие следующих компонентов приложения:

– клиентская часть;

– серверная часть (PHP);

– СУБД (MySQL).



Рисунок 3.7 – Общий принцип работы Web-приложения

Для регистрации пользователей, записи на приём, составления медицинских карточек и добавления врачей после загрузки формы пользователь заполняет все поля формы, затем происходит проверка введённых данных и после этого все данные представляются в виде параметров POST-запросов. POST-запрос посылается на сервер, который обрабатывает запрос, извлекает POST и передаёт управление и данные PHP-скрипту. Далее происходит извлечение данных из массива $\_POST и на основе этих данных составляется SQL-запрос к базе данных, который заносит данные в базу данных, а пользователь получает сообщение об успешном или неуспешном выполнении. В листинге 3.7 представлен SQL-запрос для внесения записи пациента в базу данных.

Листинг 3.7 – Внесение данных о пациенте в БД

if(isset($\_POST['surname']) && isset($\_POST['name']) && isset($\_POST['patronymic'])&&isset($\_POST['doctor'])&&isset($\_POST['date'])&&isset($\_POST['time'])&&isset($\_POST['tel']))

{

$surname = strip\_tags($\_POST["surname"]);

$name = strip\_tags($\_POST["name"]);

$patronymic=strip\_tags($\_POST["patronymic"]);

$doctor=strip\_tags($\_POST["doctor"]);

$date=strip\_tags($\_POST["date"]);

$time=strip\_tags($\_POST["time"]);

$tel=strip\_tags($\_POST["tel"]);

if(!$msql)

{

echo 'Ошибка в подключении к MySQL '. mysql\_error();

}

else

{

mysql\_select\_db("post", $msql);

if (!mysql\_query("INSERT INTO AH2.patients (Фамилия, Имя, Отчество, Врач, Дата, Время, Телефон) VALUES('$surname', '$name', '$patronymic', '$doctor', '$date', '$time','$tel')"))

}

После того, как сотрудники клиники или администратор заполнили медицинскую карточку и нажали на кнопку «Скачать» открывается окно с текстовым документом, который можно скачать. В листинге 3.8 представлена часть кода для создания текстового документа.

Листинг 3.8 – Создание текстового документа

for($code\_length = 3, $newcode = ''; strlen($newcode) < $code\_length; $newcode .= chr(!rand(0, 2) ? rand(48, 57) : (!rand(0, 1) ? rand(65, 90) : rand(97, 122))));

$today = date('d-M-Y');

$g="<br>";

$file=fopen('medical-card/'.$newcode.'\_'.'Medical Card'. '\_'.$today.'.doc','w+');

fputs($file,'Номер карточки: '.$newcode. "<br>" );

fputs($file,'Фамилия Имя Отчеств:' .$\_POST['surname']. " " .$\_POST['name']. " " .$\_POST['patronymic']."\n" );

fputs($file,'Пол:' .$\_POST['paul']."\n" );

fputs($file,'Дата рождения:' .$\_POST['date']."\n" );

fputs($file,'Адрес:' .$\_POST['address']."\n" );

fputs($file,'Место работы:' .$\_POST["work"]."\n" );

fputs($file,'Род занятий:' .$\_POST["profession"]."\n" );

fputs($file,'Результат лечения:'.$\_POST["therapy2"]. "\n" );

fputs($file,'К оплате в кассу:'.$\_POST["payment"]. "\n" );

fclose($file);

readfile('medical-card/'.$newcode.'\_'.'Medical Card'. '\_'.$today.'.doc','w+');

Для авторизации пользователей после загрузки формы пользователь вводит логин и пароль. Далее данные представляются в виде параметров POST-запросов, которые посылаются на сервер, занимающийся обработкой запроса, извлечением POST и передачей управления и данных PHP-скрипту. После чего происходит проверка введённого логина и пароля с теми данными, что есть в базе данных. Если в базе данных есть такой логин и пароль, то в качестве ответа пользователь переходит в личный кабинет, в противном случае выводится сообщение об ошибке. В листинге 3.9 представлена часть кода авторизации.

Листинг 3.9 – Авторизация пользователей

$login = $\_POST['login'];

$password = $\_POST['password'];

$result = mysql\_query("SELECT \* FROM AH2.registratsiya WHERE Логин = '$login' AND Пароль = '$password'") or die(mysql\_error());

$user1 = mysql\_fetch\_assoc($result);

mysql\_free\_result($result);

if ($user1)

{

$\_SESSION['login'] = stripslashes(htmlspecialchars($\_POST['login']));

$result1 = mysql\_query("SELECT \* FROM AH2.registratsiya WHERE Логин = '$login' AND Пароль = '$password' AND Пользователь= 'Доктор'");

$doctor1 = mysql\_fetch\_assoc($result1);

mysql\_free\_result($result1);

if ( $doctor1 )

{ ?>

<script>document.location.href='http://kur/the-entry-for-today.php'</script>;

<?php }

После того, как пользователь перешёл в личный кабинет создается сессия с его логином. По нажатию на кнопку «Выход» происходит удаление сессии (листинг 3.10).

Листинг 3.10 – Удаление сессии

session\_start();

if($\_POST['submit'])

{

$\_SESSION = array();

session\_destroy();

$\_SESSION = array();

header ('Location: index.php');

exit();

}

Представление данных в подразделах «Редактировать информацию о врачах», «Запись на прием», «Запись пациентов» осуществляется с помощью SQL-запроса. Данные SQL-запросы позволяют выводить информацию о записи на приём на сегодня, на завтра, на неделю, на месяц и за всё время. В SQL-запросе указаны поля, которые необходимо выводить при различных условиях. Пример SQL-запроса представлен в листинге 3.11.

Листинг 3.11 – SQL-запрос на вывод данных о записи пациентов на месяц

$query="select id, Фамилия, Имя, Отчество, Врач, Дата, Время, Телефон FROM patients where Дата BETWEEN current\_date and current\_date + interval 30 day order by Дата, Время";

Редактирование записей происходит согласно выбранной фамилии. Сотрудники клиники или администратор выбирает из списка фамилий необходимую фамилию, выбирает поле, которые хочет изменить и вводит новые данные в выбранное поле. По нажатию на кнопку «Изменить», запрос посылается на сервер. Где сервер обрабатывает полученные POST-запросы, извлекает POST и передает данные PHP-скрипту. После чего, выполняется запрос на изменение данных в базе данных и пользователь в качестве ответа получает новые данные из базы данных у себя на странице. Часть кода для редактирования данных представлена в листинге 3.12.

Листинг 3.12 – Редактирование данных

Изменить в <?php echo '<select name="where"><option > выберите фамилию';

$i=0;

while($data = mysql\_fetch\_array($query)){

$i++;

echo'<option value="'.$i.'">'.$data['Фамилия'];}

echo '</select>';

$query4 = mysql\_query('select Фамилия, Имя, Отчество, Врач, Дата, Время, Телефон from AH2.patients');

echo '<form method= "POST">'; ?>

поле

<?php echo '<select name="col1"><option>' ?> выберите поле для изменения

<?php for($j=0;$j<mysql\_num\_fields($query4);$j++){

echo'<option value="'.$j.'">'.mysql\_field\_name($query4, $j);}

echo '</select>'; ?>

на <input type= "text" name= "update" class="input" placeholder="введите текст изменения">

<input type= "submit" value= "Изменить" class="submit\_edit">

Удаление записей происходит согласно выбранным данным из таблицы. Сначала зарегистрированный пользователь, сотрудник или администратор отмечает необходимые данные для удаления и нажимает на кнопку «Удалить». После чего, запрос посылается на сервер. Сервер считывает POST-запросы согласно выбранным checkbox, далее происходит считывание данных из массива $\_POST и на основе этих данных составляется SQL-запрос к базе данных, который удаляет выбранные строки. В качестве ответа пользователь получает измененную базу данных у себя на странице. Часть кода для удаления данных из базы данных представлена в листинге 3.13.

Листинг 3.13 – Удаление данных

if($\_POST["dfile"])

{

$mass=$\_POST["dfile"];

$i=0;

while($mass[$i])

{

$link=mysql\_connect("localhost", "root","") or die("Could not connect");

mysql\_select\_db("AH2") or die("Could not select database");

$s1="delete from patients where id=$mass[$i]";

$result1=mysql\_query($s1) or die("Query failed");

$i++;

}

Header("Location: the-entry-for-today.php?i=2");

}

Таким образом, продемонстрированы примеры использования таких технологий для разработки серверной части Web-приложения как PHP и MySQL.