Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

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

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

(подпись и ФИО руководителя)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**КУРСОВАЯ РАБОТА**

|  |  |
| --- | --- |
| Тема | Учет медицинских карт |
| По МДК 09.01 | Проектирование и разработка веб-приложений |
| Выполнил | Корнев В.В. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Научный руководитель | Мартыненко В.А. |

г. Белореченск

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ 4

1.1 Анализ предметной области 4

1.2 Цели и задачи выполняемой работы 4

1.3 Метод реализации процесса проектирования 5

1.4 Обоснование проектных решений 5

2 ПРОЕКТНАЯ ЧАСТЬ 7

2.1 Техническое задание 7

2.2 Установка и настройка среды разработки 12

2.3 Руководство пользователя 12

2.4 Руководство администратора 13

ЗАКЛЮЧЕНИЕ 18

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 19

ПРИЛОЖЕНИЕ

ВВЕДЕНИЕ

Актуальность исследования – успешная работа компании напрямую связанна с деятельностью логистической системой. В зависимости от того, как реализована работа, деятельности основного предприятия, но и от точности работы медицинских карт. Оценка работы. Грамотно структурированная работа – залог успешного развития компании. Актуальность темы заключается в том, что ведение мед учета позволяет сделать работу «прозрачной» и минимизировать временные затраты на проведение различных складских операций.

Цель исследование – рассмотреть и установить повышение эффективности процессов, добавления, редактирования, удаления, мед карт на на сайте за счет разработки и внедрения информационной учета медицинских карт, с использованием адресного хранения мед карт.

Задачи исследования:

– изучить предметную область;

– рассмотреть ряд существующих разработок для решения поставленной цели;

– выбрать метод реализации работы учета медицинских карт;

– спроектировать работу разрабатываемой системы медицинских карт;

– разработать информационную систему;

– протестировать разработанную информационную систему медицинских карт.

Методы исследования – адресное хранение это способ размещения мед карт в профиле, при котором каждому месту хранения присваивается индивидуальный, id, или уникальный идентификатор.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ предметной области

Разрабатываемая система предназначена для более удобной и лёгкой работы с мед картами.

Система будет очень удобной, лёгкой в своём использовании. Для пациентов, администрации будет всё понятно и доступно.

Администратор сайта сможет очень легко и быстро вносить данные о пациенте и наблюдать сколько их. Тем самым будет очень легко и просто следить за пациентами. Учет медицинских карт будет не заторможенным, соответственно будет хорошая работа и пациенты скорее всего больше не вернуться.

1.2 Цели и задачи выполняемой системы

Основными целями создания мед карт на сайте:

– удобное добавление мед карт;

– следить сколько осталось товаров на складе;  
– изменение их параметров;

– удаления мед карт;

Задачами являются:

– Достичь поставленных целей;

– Разработать приложение учет медицинских карт;

– Автоматическое обновление данных;

– Оформить отчёт;

– Освоить навыки полученные на учении;

– Связать программу с БД;

– Сделать структурированную программу.

1.3 Метод реализации процесса проектирования

В ходе проведения работ по разработке учета медицинских карт автоматизируются процессы и проверки. Либо просто добавить ее к пациенту.

При внедрении системы будет достигнуто:

– добавление мед карты;

– редактирование мед карты;

– удаление данных.

– правильная и удобная работа учета медицинских карт.

Объекты автоматизации характеризуются:

– обеспечивать максимальную работоспособность;

– быть надежной и технологичной;

– обслуживание должно проводиться минимальным количеством персонала;

1.4 Обоснования проектного решения

В целом, выбор Bootstrap, HTML, PHP, JavaScript и API для разработки электронного расписания занятий обеспечивает гибкость, надежность и адаптивность. Использование этих технологий позволит нам создать функциональное и интуитивно понятное приложение, которое будет удовлетворять потребностям пользователей и быть легко масштабируемым.

Bootstrap - это популярная библиотека CSS и JavaScript, которая позволяет создавать адаптивную и мобильно-оптимизированную верстку. Мы выбрали Bootstrap для разработки электронного расписания занятий, потому что он обеспечивает быструю и удобную разработку, а также имеет большой набор готовых компонентов и стилей, которые можно использовать для создания пользовательского интерфейса.

HTML - это язык разметки, который используется для создания структуры веб-страницы. Мы выбрали HTML для разработки электронного расписания занятий, потому что он является основным языком для создания веб-страниц и обеспечивает гибкость и адаптивность.

PHP - это язык программирования, который используется для создания динамических веб-страниц. Мы выбрали PHP для разработки электронного расписания занятий, потому что он является одним из самых популярных языков для разработки веб-приложений и обеспечивает гибкость и надежность.

JavaScript - это язык программирования, который используется для создания интерактивных веб-страниц. Мы выбрали JavaScript для разработки электронного расписания занятий, потому что он позволяет создавать динамические и интерактивные элементы, такие как слайдеры, валидацию форм и анимации.

API (Application Programming Interface) - это набор инструментов и протоколов, которые позволяют разным приложениям обмениваться данными и взаимодействовать друг с другом. Мы выбрали API для разработки электронного расписания занятий, потому что он позволяет получать данные о занятиях, преподавателях и студентах из внешних источников, таких как учебные планы и расписания.

2. ПРОЕКТНАЯ ЧАСТЬ

2.1 Техническое задание

Общие сведение организации:

Полное наименование системы и ее условное обозначение

Полное наименование системы: ЧУПОО Техникум Бизнес и Право.

Краткое наименование системы: «ЧУПОО БИП».

Шифр темы или шифр ИЭ9965-20-49142272 договора

Номер договора: №ИЭ9965-20-49142272 от 15.11.2023

Наименование предприятия разработчика и заказчика системы, их реквизиты

Полное наименование заказчика: Скрыльников Дмитрий Константинович

Полное наименование разработчика: Корнев Владимир Владимирович

Частный почтовый адрес 352680, Краснодарский край, Усть-Лабиснкий район, станица Ладожская, улица длинная 9.

Контактный телефон: 8-967-663-30-96.

Расчётный счёт: 4070265380042032001.

Порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств.

Система представляется в виде полностью функционирующей программы и исполнителя в сроки, установленным договором. Приемка системы осуществляется комиссией в составе уполномоченных представителей заказчика и исполнителя.

Весь комплект документации на «ЧУПОО БИП», разработанный исполнителем передается заказчику на бумажных( в двух экземплярах), один экземпляр передается после подписания исполнителю.

Назначение и цели создания (развитие системы)

Назначение системы

Система требовалась для вывода медицинских карт пациентов на сайте

Требования к функциям (ВИ):

Имя: Получение медицинских карт

Описание: Пациент зашел на сайт для просмотра своей медицинской карты

Действующий субъект: Пациент

Предусловие: Пользователь зашел на сайт

Сценарий:

1. Главная страница сайта
2. Пользователь нажимает на вкладку “Профиль”
3. Открывается страница
4. Пользователь нажимает на просмотр своих мед карт
5. Вывелась медицинская карта где есть информация о нем и можно узнать больше если нажать на кнопку развернуть.

Для функционирования понадобится ПК, на котором будет установлена операционная система Windows 7 или выше:

Провести интернет-соединение

Обеспечить 1 работника ПК

Установить на компьютер сервер

Изучить персоналом «Руководство пользователя»

Эксплуатация системы

При анализе предметной области разрабатываемой системы были выделены следующие сущности, на основе которых будет разрабатываться сайт:

В коде мы разбиваем, приводим к читаемому виду и выводим результат в виде медицинских карт заполненными администраторами

Требования к эргономике и технической эстетике.

Требования к размещению элементов управления на экранных формах

Общие требования:

- Фоновый цвет всех страниц – темно-синий-градиент, футер и навбар имеют тот же градиент.

- Кастомный шрифт от Boostrap.

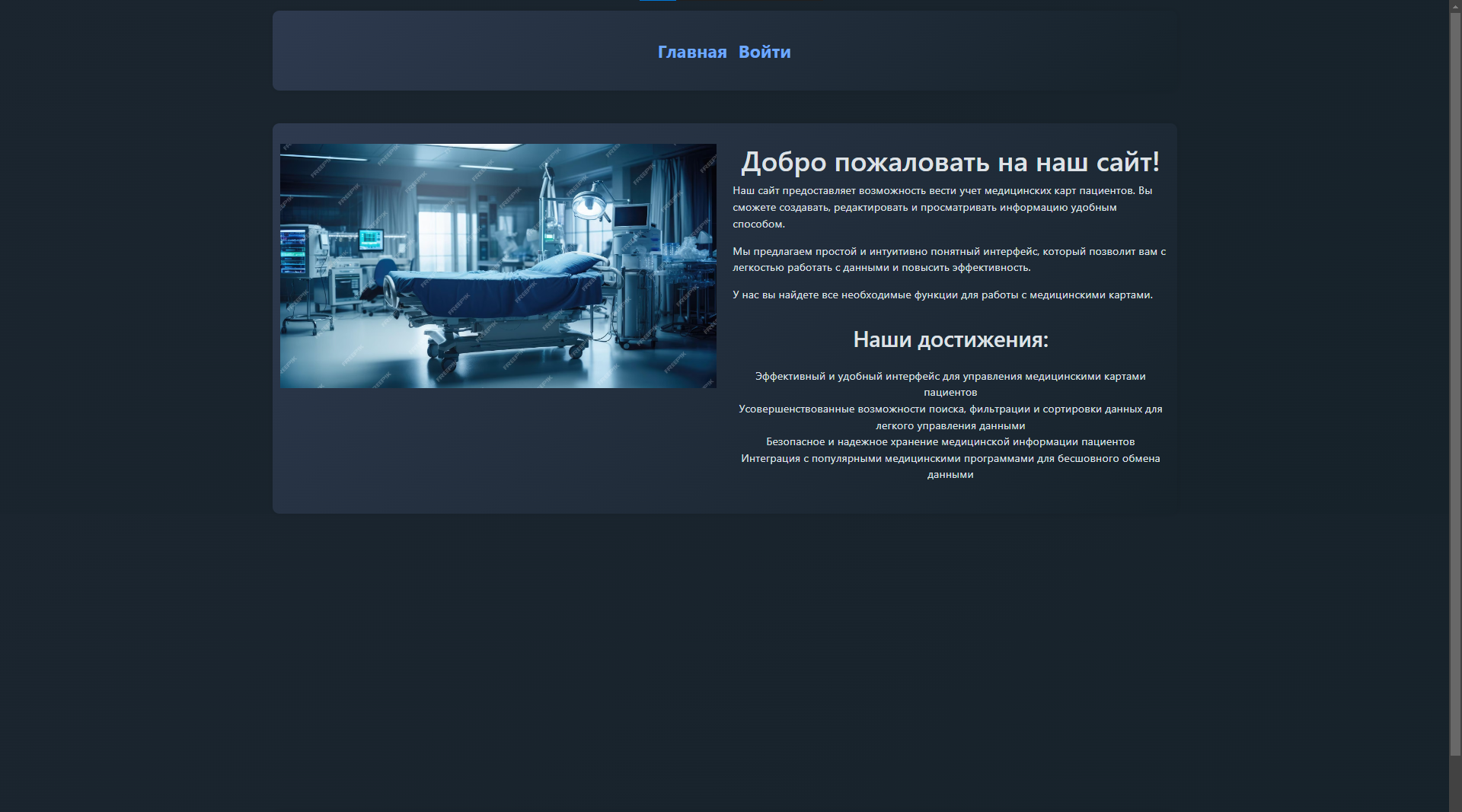
- Страница адаптивная.

- При запуске страницы будут видны 3 вкладки для выбора, в зависимости от выбора будет результат.

Требования к содержанию и оформлению вводимых данных

Требования к видам обеспечения

Для функционирования системы требуется сервер, приложение и компьютер, на которых будет производиться запуск системы.



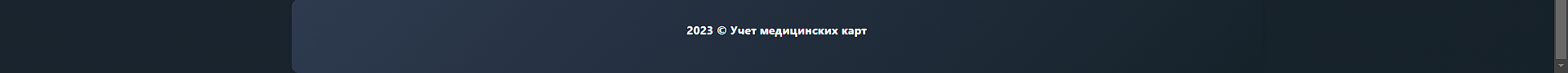


Рисунок 1 – Окно главной страницы (отдалено на 25% для лучшего вида всего оформления)

Требования для сервера БД:

Процессор 2 ядра(4 логических потока) частота - 2 – 2.53

Оперативная память не ниже 4Гб ddr3

Свободного места на диске не менее 1Гб

HDD (жесткий диск), тип – SATA2, скоростью вращения 7200 лб/мин

Требования к клиентским устройствам:

Доступ к интернету со скоростью не менее 15 Мбит/с

В качестве ОС клиентских устройств может использоваться Microsoft Windows 7 и выше.

Состав и содержание работ по созданию системы

Комплекс работ по созданию «ЧУПОО БИП» должен осуществляться в несколько этапов. Этапы приведены в таблице 2.3.:

Таблица 2.3. – Этапы создание «УМД»

|  |  |  |  |
| --- | --- | --- | --- |
| Стадии работ | Выполняемые работы | Сроки | Итог выполнения работ |
| 1 | 2 | 3 | 4 |
| Формирование требований | Обследование объектов | Выполнено | Отчёт о результатах; |
| Разработка технического задания на создание системы | Утверждение ТЗ на создание системы; |
| Проектирование | Разработка технического проекта на систему  Разработка прототипа системы | 15.11.2023  -  16.11.2023 | Технический проект на систему;  Спецификация программно-аппаратных средств системы; |
| Разработка программных средств | Разработка откладки и тестирование программных средств Подсистемы | 04.05.2023  -  07.05.2023 | Комплект  проектов  организационно-распорядительной |

Продолжение таблицы 2.4. – Этапы создание «УМД»

|  |  |  |  |
| --- | --- | --- | --- |
| Приемка работ | Проведение предварительных испытаний на стенде исполнителя | 08.05.2023  -  12.05.2023 | Акт готовности подсистемы к развертыванию в опытной зоне. |

Порядок контроля и приемки системы

Система передается в виде полностью функционирующего комплекса на базе средств вычислительной техники заказчика и исполнителя в сроки, установленные договором. Приемка системы осуществляется комиссией в составе уполномоченных представителей заказчика и исполнителя.

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Заказчик обязан предоставить сервер и персональные компьютеры, требования к которым указаны в разделе 2.4.3 «Требования к видам обеспечения».

Необходимое программное обеспечение для сервера:

- Операционная система Windows 10

Требования к документированию

Перечень подлежащих разработке комплектов и видов документов:

Техническое задание

Эскизный проект

Программа и методика испытаний

Источники разработки

Источниками разработки настоящего Технического Задания являются:

перечень документации, представленный в разделе 2.1.2 «Основания для проведения работ»;

ГОСТ 24.601 Исследование и обоснование создания АС.

ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы;

ГОСТ 24.601-86 Автоматизированные системы. Стадии создания;

2.2 Установка и настройка среды разработки

Скачайте или скопируйте проект на ваш компьютер.

Откройте терминал и перейдите в директорию проекта.

В терминале выполните команду "npm install", чтобы установить все необходимые зависимости проекта.

После установки зависимостей, выполните команду "npm start", чтобы запустить проект на локальном сервере.

Откройте браузер и перейдите на адрес <http://localhost:3000> или подругизите ссылку с названием проекта на основной сайт, чтобы увидеть работу странички.

После проверки работы странички, вы можете загрузить проект на сервер, используя FTP или другой метод.

На сервере выполните команду "npm install", чтобы установить все необходимые зависимости проекта.

После установки зависимостей, выполните команду "npm start", чтобы запустить проект на сервере.

Теперь вы можете открыть браузер и перейти на адрес вашего сайта, чтобы увидеть работу странички.

2.3 Руководство пользователя

Добро пожаловать в руководство пользователя по учете медицинских карт! Это руководство предназначено для пациентов, как просматривать свою медицинскую карту и тд.

Учет медицинских карт - это удобный инструмент, который позволяет пациентам и Администраторам просматривать медицинские карты. В этом руководстве мы расскажем вам, как просматривать профиль и медицинские карты.

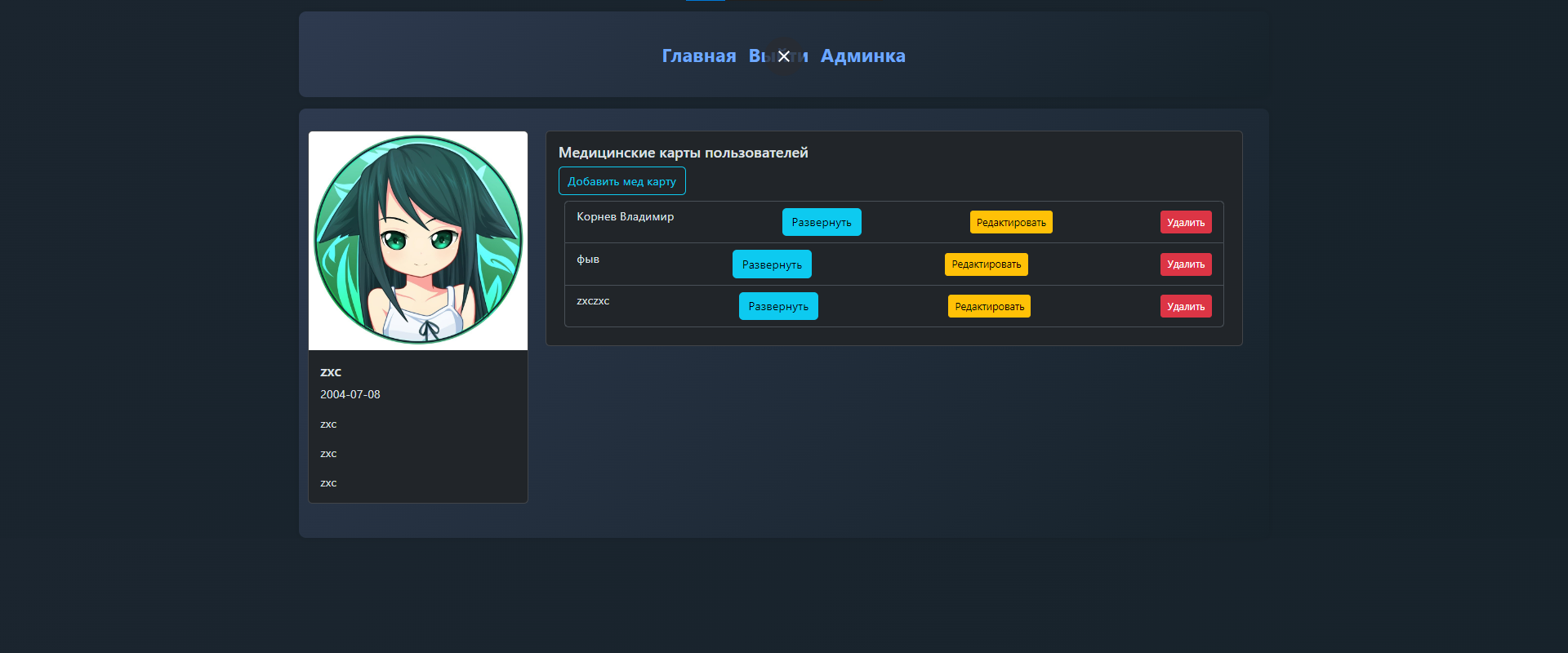


Рисунок 2 Руководство пользователя

Как просматривать Медицинские карты пациента?

Выбираем в шапке главной страницы сайта “Вход” затем регистрируемся и заполняем данные которые необходимы для регистрации.

После регистрации мы попадаем на главную страницу и у нас появляется в шапке сайта текст с Профилем нажимаем, попадаем на страницу профиля где мы уже можем просмотреть и развернуть более подробную информацию по медицинской карте, правда у вас не будет медицинской карты пока Администратор не заполнит ее.

Данная html страница выполнена в темно-синем градиенте, футер и навбар подтягивает стили с основной страницы, общая информация разбита на блоки, интерфейс понятен

2.4 Руководство администратора

Добро пожаловать в руководство администратора по Учету медицинских карт! Это руководство предназначено для тех, кто хочет узнать, как работать с медицинскими картами в качестве администратора.

Учет медицинских карт - это удобный инструмент, который позволяет пациентам и Администраторам просматривать медицинские карты. В этом руководстве мы расскажем вам, как просматривать профиль и медицинские карты.

Как работать с Учетом медицинских карт администратором?

Вход в систему:

Чтобы начать работу с Учетом медицинских карт администратором в качестве администратора, вам нужно войти в систему. Для этого нажмите на соответствующую ссылку на сайте и введите свои учетные данные.

Работа с API:

В качестве администратора вы можете работать с API, чтобы вносить изменения в медицинских картах. Для этого вам нужно использовать специальные инструменты, которые позволяют взаимодействовать с API.

Общие положения

Полное наименование системы и ее условное обозначение

Полное наименование системы:

Краткое наименование системы: «УМД».

Назначение системы

«УМД» предназначена для упрощения и более качественной работы:

Просмотра расписания и заполнения медицинских карт

Цели создания системы

Разрабатываемая система предназначена для просмотра учета медицинских карт, в случае не работы приложения.

Реализация вышеизложенных пунктов преследуют ключевую цель – повышение работоспособности предприятия.

Описание процесса деятельности

Результатом анализа предметной области стало выявление информации о том, что для максимально продуктивной работы компании, нужно чтоб, система должна была выполнять запросы на api, чтоб вся информация о расписании выводилась в таблицы.

Решения по режимам функционирования, диагностированию работы системы

Система должна поддерживать работу в двух режимах:

- основной режим, в котором все подсистемы выполняют свои основные функции;

В основном режиме функционирования «УМД» должна обеспечивать:

- выполнение функций – просмотр информации.

Решения по численности, квалификации и функциям персонала АС, режимам его работы, порядку взаимодействия

В состав персонала, необходимого для обеспечения эксплуатации «БИП» в рамках соответствующих подразделений Заказчика, необходимо

выделение следующих ответственных лиц, там описано кто требуется для обслуживания системы, так же указано с чем он работает, для работы не нужно представленных в таблице 3.1:

Таблица 3.1. – Состав персонала, необходимого для обеспечения эксплуатации «УМД»

|  |  |  |
| --- | --- | --- |
| Должность | Количество | Подсистема |
| Администратор | 1 | Администрирование системы и api |

Администратор 1 Администрирование системы и БД

Для функционирования системы требуется наличие сервера и клиентских компьютеров, на которых будет производится работа с системой. Перед вводом системы в действие необходимо:

- провести в компанию интернет-соединение.

- обеспечить каждого специалиста персональным компьютером.

- установить на один ПК данный клиент.

Решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам

Для функционирования системы требуется сервер и клиентских компьютеры, на которых будет производится запуск системы.

Требования для сервера БД:

- Процессор 2 ядра (8 логических потоков), частота – 2—2,53

- Оперативная память 8 Гб и выше

- Свободное дисковое пространство не менее 3 Гб

- HDD (жесткий диск), тип - SATA2, скорость вращения 7200 об/мин

Решения по составу программных средств, языкам деятельности, алгоритмам процедур и операций и методам их реализации

При разработке ИС использованы следующие программные компоненты:

Средство разработки PhP Shtorm

Облачный сервер для проверки запросов на api

В качестве ОС клиентских устройств может использоваться:

- Microsoft Windows 7 и выше

Для реализации клиентской части могут использоваться:

PhP

Css

JavaScript

Bootsrap

Html

Эскиз пользовательского интерфейса «ЧУПОО БИП»

Эскизы пользовательского интерфейса:

Данная страница будет дополнять сайт УМД, она будет встраиваться в уже существующий проект, так будет выглядить создаваемая страница:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Автоматически созданное описание

Рисунок 3 Так выглядит создаваемая страница

ЗАКЛЮЧЕНИЕ

При разработке приложения было сделано простота использования и удобство интерфейса, а также защищенность от постороннего вмешательства.

Также были достигнуты цели курсовой работы. Удалось создать трёхуровневую архитектуру приложения. Цель работы достигнута, а именно создано приложение для работы с базой данных.

Куда проще вносить данные о товаре и хранить их там.

Задачи решены, а именно создать трёхуровневую архитектуру приложения, быстрый доступ к базе данных, добавлению товара, просмотра, редактирования и удаления любого товара.

В результате выполнения курсовой, был создан динамический сайт для внесения и хранения там данных о товарах. Для его разработки применялось актуальное программное обеспечение, без помощи которого не было бы возможности добиться поставленной цели. Это программное обеспечение включает в себя:

– язык программирования PHP;

– фреймворк bootstrap;

– веб-сервер XAMPP;

– реляционная система управления БД MySQL;

– приложение PHPMyAdmin.

С помощью этих инструментов значительно сокращена ручная работа и повторяющийся код. Чтобы добиться ЧПУ без применения bootstrap и дополнительных библиотек, необходимо вносить много изменений в файлы.

Таким образом, затрачено меньше времени на разработку сайта. Всё информационное наполнение сайта находится в БД.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вандюк, Джон К. CMS Drupal. Руководство по разработке системы управления сайтом / Вандюк, Джон К., Мэтт Вестгейт,. - М.: Вильямс, 2019. - 400 c.
2. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2020. - 464 c.
3. Дакетт, Джон HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - Москва: Наука, 2019. - 480 c.
4. Дронов, В. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 c.
5. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 c.
6. Дронов, В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / В.А. Дронов. - М.: БХВ-Петербург, 2019. - 820 c.
7. Дронов, Владимир JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 c.
8. Дронов, Владимир PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: "БХВ-Петербург", 2019. - 448 c.
9. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 c.
10. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2021. - 364 c.
11. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2020. - 895 c.
12. Кузнецов, М. PHP 5. Практика создания Web-сайтов / М. Кузнецов, И. Симдянов, С. Голышев. - М.: БХВ-Петербург, 2020. - 960 c.
13. Леонтьев, Борис PHP 5.0 для начинающих, или как создать динамический WEB-сайт / Борис Леонтьев. - М.: Новый издательский дом, 2019. - 176 c.
14. Петюшкин, Алексей HTML в Web-дизайне / Алексей Петюшкин. - Москва: Машиностроение, 20. - 400 c.
15. Шкрыль, А. PHP - это просто. Программируем для Web-сайта / А. Шкрыль. - М.: БХВ-Петербург, 2021. - 368 c

ПРИЛОЖЕНИЕ

Style.css - Прописи для стилей

.btn-primary{

}

body {

background: linear-gradient(135deg, #1b252f, #16222A); /\* Темный градиент \*/

}

body {

font-family: 'Roboto', sans-serif;

}

pumbaforms{

background: linear-gradient(135deg, #2E3A4F, #16222A); /\* Темный градиент \*/

}

.container {

background: linear-gradient(135deg, #2E3A4F, #16222A); /\* Темный градиент \*/

border-radius: 10px; /\* Закругление углов контейнера \*/

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Тень контейнера \*/

padding: 30px; /\* Внутренний отступ контейнера \*/

margin-top: 50px; /\* Внешний отступ сверху \*/

}

.tur

{

width: 25%;

}

.log{

width: 25%;

}

.h1, h1 {

text-align: center;

}

.footer {

position: absolute;

left: 0;

bottom: 0;

width: 100%;

height: 80px;

}

p {

margin-top: 0;

margin-bottom: 0rem;

}

.list-group{

padding-top: 2%;

}

Javascript.js

Функционал кнопок и тд

function submitEditForm() {

var medcartId = $("#medcart").val();

var newDoctor = $("#editDoctor").val();

fetch("EditMedcarts.php", {

method: "POST",

headers: {

"Content-Type": "application/x-www-form-urlencoded"

},

body: "id=" + medcartId + "&doctor=" + newDoctor

})

.then(response => response.text())

.then(data => {

console.log(data);

// Закрываем модальное окно после успешного редактирования

document.querySelector("#editModal").classList.remove("show");

document.querySelector("#editModal").style.display = "none";

document.querySelector(".modal-backdrop").remove();

// Можете добавить здесь обновление данных на странице, если необходимо

location.reload(); // Обновление страницы

})

.catch(error => {

console.error("Error during fetch request:", error);

// Обработка ошибок при отправке данных на сервер

});

}

document.addEventListener('click', function(event) {

if (event.target.classList.contains('delete-btn')) {

// Получаем идентификатор медкарты из атрибута data

var medcartId = event.target.dataset.medcartId;

// Вызываем fetch для удаления

fetch("DeleteMedcart.php", {

method: "POST",

headers: {

"Content-Type": "application/x-www-form-urlencoded"

},

body: "id=" + medcartId

})

.then(response => response.text())

.then(data => {

console.log(data);

// По желанию, можно перезагрузить страницу или обновить интерфейс здесь

location.reload(); // Обновление страницы

})

.catch(error => {

console.error("Error during fetch request:", error);

// Обработка ошибок, если это необходимо

});

}

});

function toggleButtonText(button) {

// Получаем текущий текст кнопки

var buttonText = button.innerText;

// Меняем текст в зависимости от текущего состояния

if (buttonText.trim() === 'Развернуть') {

button.innerText = 'Свернуть';

} else {

button.innerText = 'Развернуть';

}

}

AddMedcarts Добавление мед карт

<?php

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Проверка, была ли отправлена форма

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Получение данных из формы

$datepost = $\_POST["datepost"];

$diagnoz = $\_POST["diagnoz"];

$heal = $\_POST["heal"];

$doctor = $\_POST["doctor"];

$user\_id = $\_POST["user\_id"];

$sql = "INSERT INTO medcarts (datepost, diagnoz, heal, doctor, user\_id) VALUES ('$datepost', '$diagnoz', '$heal', '$doctor', '$user\_id')";

if(mysqli\_query($conn, $sql)){

header("location:strprofile.php");

} else{

echo "Ошибка: " . mysqli\_error($conn);

}

$sql = "UPDATE medcarts SET doctor='$doctor' WHERE user\_id='$user\_id'";

}

mysqli\_close($conn);

?>

AddPacients

<?php

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

if (!$conn) {

die("Ошибка: " . mysqli\_connect\_error());

}

$name=$\_POST["name"];

$lastname=$\_POST["lastname"];

$surname=$\_POST["surname"];

$date=$\_POST["date"];

$karts=$\_POST["karts"];

$oms=$\_POST["oms"];

$adres=$\_POST["adres"];

$sql = "INSERT INTO pacients (name, lastname,surname, date, karts, oms, adres) VALUES ('$name', '$lastname', '$surname', '$date', '$karts', '$oms', '$adres')";

if(mysqli\_query($conn, $sql)){

echo "Данные успешно добавлены";

} else{

echo "Ошибка: " . mysqli\_error($conn);

}

mysqli\_close($conn);

?>

Config Подключение mysql

<?php

return [

'db\_name' => 'php-auth-demo',

'db\_host' => '127.0.0.1',

'db\_user' => 'mysql',

'db\_pass' => 'mysql',

];

DeleteMedcart Удаленние мед карт

<?php

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

// Get the medcart id from the POST data

$medcartId = $\_POST['id'];

// Delete the medcart from the database

$conn->query("DELETE FROM medcarts WHERE id='$medcartId'");

// Return a success message

echo "Success";

} else {

// Return an error if the request method is not POST

http\_response\_code(400);

echo "Bad Request";

}

?>

EditMedcarts Редактирование мед карт

<?php

var\_dump($\_POST);

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

if ($conn->connect\_error) {

die("Ошибка подключения: " . $conn->connect\_error);

}

// Проверка метода запроса

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

// Проверка существования и непустоты переданных данных

if (isset($\_POST['id']) && isset($\_POST['doctor'])) {

// Получение данных из запроса

$medcartId = $conn->real\_escape\_string($\_POST['id']);

$newDoctor = $conn->real\_escape\_string($\_POST['doctor']);

// Подготовленный запрос для обновления данных в базе данных

$stmt = $conn->prepare("UPDATE medcarts SET doctor=? WHERE id=?");

$stmt->bind\_param("ss", $newDoctor, $medcartId);

// Выполнение запроса

if ($stmt->execute()) {

// Возвращение успешного ответа

echo "Success";

} else {

// Возвращение ошибки в случае неудачи

http\_response\_code(500);

echo "Ошибка при выполнении запроса: " . $stmt->error;

}

// Закрытие подготовленного запроса

$stmt->close();

} else {

// Возвращение ошибки, если не переданы необходимые данные

http\_response\_code(400);

echo "Bad Request: Не переданы необходимые данные";

}

} else {

// Возвращение ошибки, если запрос не является POST

http\_response\_code(400);

echo "Bad Request: Запрос не является POST";

}

// Закрытие подключения к базе данных

$conn->close();

?>

Glavnaya Главная страница сайта

<?php include "headet.php";

?>

<body>

<div class="container mt-5">

<div class="row">

<!-- Левая сторона с изображением -->

<div class="col-md-6">

<img src="https://img.freepik.com/premium-photo/a-photo-of-hospital\_931878-23795.jpg?w=2000" class="img-fluid" alt="Placeholder Image">

</div>

<!-- Правая сторона с текстовым контентом -->

<div class="col-md-6">

<h1>Добро пожаловать на наш сайт!</h1>

<p>Наш сайт предоставляет возможность вести учет медицинских карт пациентов. Вы сможете создавать, редактировать и просматривать информацию удобным способом.</p>

<p>Мы предлагаем простой и интуитивно понятный интерфейс, который позволит вам с легкостью работать с данными и повысить эффективность.</p>

<p>У нас вы найдете все необходимые функции для работы с медицинскими картами.</p>

<h2 class="text-center p-3">Наши достижения:</h2>

<ul class="list-unstyled text-center">

<li>Эффективный и удобный интерфейс для управления медицинскими картами пациентов</li>

<li>Усовершенствованные возможности поиска, фильтрации и сортировки данных для легкого управления данными</li>

<li>Безопасное и надежное хранение медицинской информации пациентов</li>

<li>Интеграция с популярными медицинскими программами для бесшовного обмена данными</li>

</ul>

</div>

</div>

</div>

</body>

<footer class="footer text-white p-3">

<div class="container text-center">

<p class="text-white fw-bold">2023 © Учет медицинских карт</p>

</div>

</footer>

</html>

Headet Шапка всего сайта

<!DOCTYPE html>

<?php

session\_start();

// Проверка, вошел ли пользователь

if (!isset($\_SESSION['auth'])){

$\_SESSION['auth']=false;

}

?>

<!DOCTYPE html>

<html lang="en" data-bs-theme="dark">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<script src="js/ja.js"></script>

<link rel="stylesheet" href="style/style.css">

<link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">

</head>

<body>

<div class="container text-center mt-3">

<ul class="nav justify-content-center nav-underline " style="font-size: 26px;font-weight: bold;">

<li class="nav-item">

<a class="nav-link" href="glavnaya.php">Главная</a>

</li>

<?php if ($\_SESSION["auth"] === true) { ?>

<li class="nav-item">

<a class="nav-link" href="logout.php">Выйти</a>

</li>

<?php } else { ?>

<li class="nav-item">

<a class="nav-link" href="user.php">Войти</a>

</li>

<?php } ?>

<?php if ($\_SESSION["idrolle"] === 1) { ?>

<li class="nav-item">

<a class="nav-link" href="UsersProfile.php">Профиль пациента</a>

</li>

<?php } ?>

<?php if ($\_SESSION["idrolle"] === 2) { ?>

<li class="nav-item">

<a class="nav-link" href="strprofile.php">Админка</a>

</li>

<?php } ?>

</ul>

</div>

</body>

</html>

Index Форма регистрации

<?php include "headet.php";

?>

<div class="container d-flex align-items-center min-vh-100">

<form class="row g-3" method="post" action="reg.php" enctype="multipart/form-data" style="margin: 5%;" onsubmit="return validateForm()">

<div class="col-md-12 text-center">

<h1 class="display-4">Регистрация</h1>

</div>

<div class="col-md-6">

<label for="fio" class="form-label">Фамилия Имя Отчество</label>

<input type="text" class="form-control" id="fio" name="fio" required>

</div>

<div class="col-md-6">

<label for="date" class="form-label">Дата рождения</label>

<input type="date" class="form-control" id="date" name="date" required>

</div>

<div class="col-md-6">

<label for="sex" class="form-label">Пол</label>

<input type="text" class="form-control" id="sex" name="sex" required>

</div>

<div class="col-md-6">

<label for="adres" class="form-label">Адрес проживания</label>

<input type="text" class="form-control" id="adres" name="adres" required>

</div>

<div class="col-md-6">

<label for="telephone" class="form-label">Номер телефона</label>

<input type="text" class="form-control" id="telephone" name="telephone" pattern="[0-9]{10}" title="Введите 10 цифр" required>

</div>

<div class="col-md-12">

<label for="username" class="form-label">Логин</label>

<input type="text" class="form-control" id="username" name="username" required style="

width: 50%;

">

</div>

<div class="col-md-12">

<label for="password" class="form-label">Пароль</label>

<input type="password" class="form-control" id="password" name="password" required style="

width: 50%;

">

</div>

<div class="col-md-12">

<label for="avatars" class="form-label">Изображение</label>

<input class="form-control" type="file" id="avatars" name="avatars" required>

</div>

<div class="col-md-12 mt-3">

<button type="submit" class="btn btn-primary btn-lg">Зарегистрироваться</button>

</div>

</form>

</div>

<script>

function validateForm() {

var inputs = document.querySelectorAll('input[required]');

for (var i = 0; i < inputs.length; i++) {

if (!inputs[i].value) {

alert("Пожалуйста, заполните все обязательные поля.");

return false;

}

}

return true;

}

</script>

Kursov Добавление пациента

<!DOCTYPE html>

<?php include "headet.php";?>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">

</head>

<body>

<form action="addPacientKarts.php" method="post" class="row g-3">

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Фамилия</label>

<input type="text" class="form-control" name="surname">

</div>

<div class="col-md-6">

<label for="inputPassword4" class="form-label">Имя</label>

<input type="text" class="form-control" name="name">

</div>

<form class="row g-3">

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Отчество</label>

<input type="text" class="form-control" name="lastname">

</div>

<div class="col-md-6">

<label for="inputPassword4" class="form-label">Дата рождения</label>

<input type="text" class="form-control" name="date">

</div>

<form class="row g-3">

<div class="col-md-6">

<label for="inputEmail4" class="form-label">№ амб.карты</label>

<input type="text" class="form-control" name="karts">

</div>

<div class="col-md-6">

<label for="inputPassword4" class="form-label">№ полоса ОМС</label>

<input type="text" class="form-control" name="oms">

</div>

<form class="row g-3">

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Адрес проживания</label>

<input type="text" class="form-control" name="adres">

</div>

<div class="col-12">

<button onclick="da()" type="submit" class="btn btn-primary">Добавить пациента</button>

</div>

</form>

<script>

</script>

</body>

</html>

Login Функционал авторизации

<?php

session\_start(); // Начинаем сессию

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

// Подключение к базе данных

$link = new PDO('mysql:host=localhost;dbname=zoxan', 'root', '');

// Получаем данные из формы

$username = $\_POST["username"];

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

// Поиск пользователя в базе данных по имени пользователя

$q = $link->prepare("SELECT \* FROM users WHERE username = :username");

$q->execute(["username" => $username]);

$user = $q->fetch(PDO::FETCH\_ASSOC);

// Проверка пароля

if ($user && password\_verify($password, $user['password'])) {

// Пароль верен, устанавливаем сессию

$\_SESSION['auth'] = true;

$\_SESSION['user\_id'] = $user['id'];

$\_SESSION['username'] = $user['username'];

$\_SESSION['idrolle'] = $user['idrolle'];

$\_SESSION['avatars'] = $user['avatars'];

// Перенаправляем пользователя на защищенную страницу

header('Location: glavnaya.php');

exit;

} else {

echo 'Неверные имя пользователя или пароль.';

}

}

?>

Logout Выход со страницы

<?php

session\_start();

// Уничтожение сессии и перенаправление на главную страницу

$\_SESSION['auth'] = false;

$\_SESSION['idrolle']= 0;

header("Location: glavnaya.php");

exit();

?>

MedCart Добавление мед карты

<!doctype html>

<?php include "headet.php";?>

<?php

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

$query = $conn->query("SELECT \* FROM users");

while($user = mysqli\_fetch\_assoc($query))

{

$users[] = $user;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport"

content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

</head>

<body>

<form action="AddMedcarts.php" method="post" class="pumbaforms row g-3 align-items-center flex-column mx-auto text-center" onsubmit="return validateForm()" style="border-radius: 10px; /\* Закругление углов контейнера \*/box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /\* Тень контейнера \*/background: linear-gradient(135deg, #2c3742, #16222A); /\* Темный градиент \*/width: 50%;margin-top: 5%;">

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Дата поступления</label>

<input type="date" class="form-control" name="datepost" required>

</div>

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Диагноз</label>

<input type="text" class="form-control" name="diagnoz" required>

</div>

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Лечение</label>

<input type="text" class="form-control" name="heal" required>

</div>

<div class="col-md-6">

<label for="inputEmail4" class="form-label">Доктор</label>

<input type="text" class="form-control" name="doctor" required>

</div>

<div class="col-md-6">

<label for="cars" class="form-label">Выберите пользователя</label>

<select class="form-select" name="user\_id" id="cars" required>

<?php foreach ($users as $user): ?>

<option value="<?=$user["id"]?>"><?= $user['username'] ?></option>

<?php endforeach; ?>

</select>

</div>

<div class="col-12 mt-3">

<button type="submit" class="btn btn-primary" style="

margin-bottom: 20px;

">Добавить мед карту</button>

</div>

</form>

<script>

function validateForm() {

// Простая проверка на заполненность полей

var inputs = document.querySelectorAll('input[required], select[required]');

for (var i = 0; i < inputs.length; i++) {

if (!inputs[i].value) {

alert("Пожалуйста, заполните все обязательные поля.");

return false;

}

}

return true;

}

</script>

</body>

</html>

Profile Профиль пацинета

<?php

session\_start();

$id=$\_SESSION['user\_id'];

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

$query = $conn->query("SELECT \* FROM users WHERE id='$id'");

while($user = mysqli\_fetch\_assoc($query))

{

$users[] = $user;

}

if ($\_SESSION['idrolle'] != 2){

header("location:reg.php");

}

?>

<!DOCTYPE html>

<?php include "headet.php"; ?>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9" crossorigin="anonymous">

</head>

<body>

<img src= class="img-thumbnail" alt="...">

</body>

</html>

Reg Форма регистрации

<?php

session\_start();

// Создаем соединение

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Для начала нужно войти в аккаунт";

// Обработка данных из формы

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$fio = $\_POST["fio"];

$date = $\_POST["date"];

$sex = $\_POST["sex"];

$adres = $\_POST["adres"];

$telephone = $\_POST["telephone"];

$username = $\_POST["username"];

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

// Хэширование пароля (рекомендуется для безопасности)

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

// Вставка данных в таблицу users

$sql = "INSERT INTO users (fio, date, sex, adres, telephone, idrolle, username, password) VALUES ('$fio', '$date', '$sex', '$adres', '$telephone', 1, '$username', '$hashed\_password')";

if ($conn->query($sql) === TRUE) {

$last\_id = $conn->insert\_id;

// Обработка файла аватара

$avatar\_dir = "avatars/"; // Укажите путь к папке для сохранения аватаров

$avatar\_path = $avatar\_dir . $last\_id . "\_" . basename($\_FILES["avatars"]["name"]);

move\_uploaded\_file($\_FILES["avatars"]["tmp\_name"], $avatar\_path);

// Обновление записи в базе данных с путем к файлу аватара

$sql\_update\_avatar = "UPDATE users SET avatars='$avatar\_path' WHERE id=$last\_id";

$conn->query($sql\_update\_avatar);

echo "Регистрация успешна";

header('Location: glavnaya.php');

} else {

echo "Ошибка при регистрации: " . $conn->error;

}

}

// Закрытие соединения с базой данных

$conn->close();

?>

Strprofile

<?php include "headet.php"; ?>

<?php

$id=$\_SESSION['user\_id'];

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

$query = $conn->query("SELECT \* FROM users WHERE id='$id'");

$users = [];

while($user = mysqli\_fetch\_assoc($query))

{

$users[] = $user;

}

if ($\_SESSION['idrolle'] != 2){

header("location:reg.php");

}

$query = $conn->query("SELECT \* FROM medcarts");

$medcarts = [];

while ($medcart = mysqli\_fetch\_assoc($query)) {

$medcarts[] = $medcart;

}

?>

<div class="container mt-3">

<div class="row">

<div class="col-md-3">

<!-- ... Код для информации о пациенте ... -->

</div>

<div class="row">

<div class="col-md-3">

<div class="card mb-3">

<img src="<?=$users[0]['avatars']?>" class="card-img-top" alt="Avatar">

<div class="card-body">

<h5 class="card-title"><?=$users[0]['fio']?></h5>

<p class="card-text"><?=$users[0]['date']?></p>

<p class="card-text"><?=$users[0]['sex']?></p>

<p class="card-text"><?=$users[0]['adres']?></p>

<p class="card-text"><?=$users[0]['telephone']?></p>

</div>

</div>

</div>

<div class="col-md-9">

<div class="card mb-4">

<div class="card-body">

<h5 class="card-title">Медицинские карты пользователей</h5>

<div>

<button type="button" class="btn btn-outline-info" onclick="redirectToMedCarts()">Добавить мед карту</button>

<script>

function redirectToMedCarts() {

// Выполните переход на страницу MedCarts.php

window.location.href = 'MedCarts.php';

}

</script>

</div>

<ul class="list-group p-2">

<?php foreach ($medcarts as $medcart): ?>

<li class="list-group-item d-flex justify-content-between align-items-center">

<p class="card-text"><?php

$user\_id=$medcart['user\_id'];

$query = $conn->query("SELECT \* FROM users WHERE id='$user\_id'");

$us = mysqli\_fetch\_assoc($query);

echo $us['fio'];

?></p>

<button type="button" class="btn btn-info" data-toggle="collapse" data-target="#medcart<?=$medcart['id']?>" aria-expanded="false" aria-controls="medcart<?=$medcart['id']?>" id="toggleBtn<?=$medcart['id']?>" onclick="toggleButtonText(this)">

Развернуть

</button>

<button type="button" class="btn btn-warning btn-sm edit-btn" data-toggle="modal" data-target="#editModal" data-medcart-id="<?=$medcart['id']?>">

Редактировать

</button>

<button type="button" class="btn btn-danger btn-sm delete-btn" data-target="#deleteModal" data-medcart-id="<?=$medcart['id']?>">

Удалить

<!-- Развернутая информация -->

<div class="collapse" id="medcart<?=$medcart['id']?>">

<div class="card card-body mt-2">

<input type="hidden" id="medcart" value="<?=$medcart['id']?>">

<p><strong>Дата поступления:</strong> <?=$medcart['datepost']?></p>

<p><strong>Диагноз:</strong> <?=$medcart['diagnoz']?></p>

<p><strong>Лечение:</strong> <?=$medcart['heal']?></p>

<p><strong>Доктор:</strong> <?=$medcart['doctor']?></p>

</div>

</div>

</li>

<?php endforeach; ?>

</ul>

</div>

</div>

</div>

</div>

</div>

<!-- Модальное окно для редактирования -->

<div class="modal fade" id="editModal" tabindex="-1" role="dialog" aria-labelledby="editModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="editModalLabel">Редактировать карту</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<!-- Форма редактирования -->

<form id="editForm" method="post" action="EditMedcarts.php">

<div class="form-group">

<label for="editDoctor">Имя нового доктора</label>

<input type="text" class="form-control" id="editDoctor" name="doctor">

</div>

<button type="button" class="btn btn-primary" onclick="submitEditForm()">Сохранить изменения</button>

</form>

</div>

</div>

</div>

</div>

<!-- Модальное окно для удаления -->

<div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="deleteModalLabel">Удалить карту?</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<p>Вы уверены, что хотите удалить эту карту?</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Отмена</button>

<button type="button" class="btn btn-danger" onclick="deleteMedcart()">Удалить</button>

</div>

</div>

</div>

</div>

<!-- Подключение необходимых скриптов Bootstrap и jQuery -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

User Роли

<?php

// Генерация рандомных чисел для капчи, только если их еще нет в сессии

if (!isset($\_SESSION['num1']) || !isset($\_SESSION['num2']) || !isset($\_SESSION['captchaResult'])) {

$\_SESSION['num1'] = rand(1, 10);

$\_SESSION['num2'] = rand(1, 10);

$\_SESSION['captchaResult'] = $\_SESSION['num1'] + $\_SESSION['num2'];

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Проверка капчи

if (!empty($\_POST['captcha']) && intval($\_POST['captcha']) === $\_SESSION['captchaResult']) {

// Правильный ответ на капчу

// Ваша логика для успешной обработки формы

echo "Форма успешно отправлена!";

} else {

// Неправильный ответ на капчу

echo "Пожалуйста, введите правильный ответ на капчу.";

}

// Генерация новых рандомных чисел для капчи после отправки формы

$\_SESSION['num1'] = rand(1, 10);

$\_SESSION['num2'] = rand(1, 10);

$\_SESSION['captchaResult'] = $\_SESSION['num1'] + $\_SESSION['num2'];

}

?>

<body>

<?php include "headet.php";

?>

<!-- Форма с капчей -->

<div class="container form-container d-flex align-items-center justify-content-center">

<form class="form" method="post" action="login.php" onsubmit="return validateForm() "style="

width: inherit;

width: 50%;

">

<h1 class="text-center mb-6">Авторизация <span class="badge bg-secondary"></span></h1>

<label for="username" class="form-label">Username</label>

<input type="text" class="form-control" id="username" name="username" required>

<label for="password" class="form-label mt-3">Password</label>

<input type="password" class="form-control" id="password" name="password" required>

<!-- Простое математическое уравнение в качестве капчи -->

<label for="captcha" class="form-label mt-3">Сколько будет <?= $\_SESSION['num1'] ?> + <?= $\_SESSION['num2'] ?>?</label>

<input type="text" class="form-control" id="captcha" name="captcha" required>

<button type="submit" class="btn btn-primary d-grid gap-2 col-3 mx-auto mt-3">Авторизация</button>

<p class="text-center mt-3">

<a class="icon-link-hover" href="index.php" style="color: #fff;">Зарегистрироваться</a>

</p>

</form>

</div>

<script>

function validateForm() {

// Простая проверка на заполненность полей

var inputs = document.querySelectorAll('input[required]');

for (var i = 0; i < inputs.length; i++) {

if (!inputs[i].value) {

alert("Пожалуйста, заполните все обязательные поля.");

return false;

}

}

// Дополнительная проверка на правильный ответ на капчу

var captchaInput = document.getElementById('captcha');

if (isNaN(captchaInput.value) || parseInt(captchaInput.value) !== <?= $\_SESSION['captchaResult'] ?>) {

alert("Пожалуйста, введите правильный ответ на капчу.");

return false;

}

return true;

}

</script>

</body>

UsersProfile Панель Администратора

<?php include "headet.php"; ?>

<?php

$id=$\_SESSION['user\_id'];

$conn = mysqli\_connect("localhost", "root", "", "zoxan");

$query = $conn->query("SELECT \* FROM users WHERE id='$id'");

$users= [];

while($user = mysqli\_fetch\_assoc($query))

{

$users[] = $user;

}

if ($\_SESSION['idrolle'] != 1){

}

$query = $conn->query("SELECT \* FROM medcarts WHERE user\_id='$id'");

$medcart = mysqli\_fetch\_assoc($query)

?>

<div class="container mt-3">

<div class="row">

<div class="col-md-3">

<!-- ... Код для информации о пациенте ... -->

</div>

<div class="row">

<div class="col-md-3">

<div class="card mb-3">

<img src="<?=$users[0]['avatars']?>" class="card-img-top" alt="Avatar">

<div class="card-body">

<h5 class="card-title"><?=$users[0]['fio']?></h5>

<p class="card-text"><?=$users[0]['date']?></p>

<p class="card-text"><?=$users[0]['sex']?></p>

<p class="card-text"><?=$users[0]['adres']?></p>

<p class="card-text"><?=$users[0]['telephone']?></p>

</div>

</div>

</div>

<div class="col-md-9">

<div class="card mb-4">

<div class="card-body">

<h5 class="card-title">Медицинские карты</h5>

<ul class="list-group">

<?php foreach ($users as $user): ?>

<li class="list-group-item d-flex justify-content-between align-items-center">

<p class="card-text"><?=$user['fio']?></p>

<button type="button" class="btn btn-info" data-toggle="collapse" data-target="#medcart<?=$medcart['id']?>" aria-expanded="false" aria-controls="medcart<?=$medcart['id']?>" id="toggleBtn<?=$medcart['id']?>" onclick="toggleButtonText(this)">

Развернуть

</button>

<!-- Развернутая информация -->

<div class="collapse" id="medcart<?=$medcart['id']?>">

<div class="card card-body mt-2">

<p><strong>Дата поступления:</strong> <?=$medcart['datepost']?></p>

<p><strong>Диагноз:</strong> <?=$medcart['diagnoz']?></p>

<p><strong>Лечение:</strong> <?=$medcart['heal']?></p>

<p><strong>Доктор:</strong> <?=$medcart['doctor']?></p>

</div>

</div>

</li>

<?php endforeach; ?>

</ul>

</div>

</div>

</div>

</div>

</div>

<!-- Подключение необходимых скриптов Bootstrap и jQuery -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>