Министерство образования и молодежной политики

Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного

строительства»

Специальность 09.02.07

«Информационные системы и программирование»

Заведующий отделением Председатель цикловой комиссии

\_\_\_\_\_\_\_\_\_\_\_ Е.В.Дудель \_\_\_\_\_\_\_\_\_\_\_\_\_ Г.В.Мирошниченко

Разработка веб-сайта для ветеринарной клиники

**Пояснительная записка**

к дипломному проекту

ДП-ПР-42-06-2025-ПЗ

Разработал:

Студент гр. ПР-42 \_\_\_\_\_\_\_\_\_\_\_\_/ А.Ф. Пасечникова

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_/ Д.О. Гариев

Н. контроль: \_\_\_\_\_\_\_\_\_\_\_/ Г.В. Мирошниченко

Рецензент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_

2025Министерство образования и молодежной политики

Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного

строительства»

Специальность 09.02.07

«Информационные системы и программирование»

Разработка веб-сайта для ветеринарной клиники

**Пояснительная записка**

к дипломному проекту

ДП-ПР-42-06-2025-ПЗ

2025

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Специальность ***09.02.07 Информационные системы и программирование***

Рассмотрено: Утверждаю:

на заседании ЦМК зам. директора по

протокол № 8 УВР ГАПОУ СО «ЕКТС»

от «21» марта 2025г.

председатель ЦМК

\_\_\_\_\_\_\_\_\_\_\_\_/Г.В. Мирошниченко \_\_\_\_\_\_\_\_\_\_\_А.М. Шанин

ЗАДАНИЕ

НА ДИПЛОМНЫЙ ПРОЕКТ

Студентке : Пасечниковой Алине Фроловны

Группа ПР-42

Тема: Разработка веб-сайта для ветеринарной клиники

Руководитель: Гариев Денис Олегович

Срок дипломного проектирования:

С 18.05.2025г. по 28.06.2025г.

Сдача проекта на рецензию 13.06.2025г.

2025

**СОСТАВ ДИПЛОМНОГО ПРОЕКТА**

**Задание на дипломное проектирование**

**Отзыв руководителя**

**Рецензия**

**Пояснительная записка**

**Программный продукт (при отсутствии грифа «секретно»)**

**РЕКОМЕНДУЕМОЕ СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

* Титульный лист
* Задание дипломного проектирования
* Содержание
* Введение - о*ценка современного состояния проблемы, решаемой в проекте, обоснование необходимости выполнения проекта, новизна и актуальность темы*
* Основная часть;

1. Описание предметной области
2. Назначение и область применения программы- *описываются и анализируются объект и предмет исследования, обосновывается выбор применяемых средств, методов, технологий;*
3. Проектирование задачи- *предлагаются решения поставленных задач (проектирование и разработка базы данных, проектирование и разработка пользовательского интерфейса, разработка программ, разработка эксплуатационных документов – инструкций, руководств);*

3.1 Обоснование инструментов разработки

3.2 Описание алгоритма решения задачи

1. Программа решения задачи

4.1 Логическая структура

4.2 Физическая структура

1. Тестирование и отладка программы
2. Применение

6.1 Назначение программы

6.2 Требования к аппаратным ресурсам ПК

6.3 Руководство пользователя

1. Охрана труда и противопожарная безопасность

Заключение - *краткие выводы о результатах выполненной работы, оценка перспектив дальнейшего улучшения функциональности проекта, предложения по использованию результатов работы.*

* Список используемых источников;

1. Петрова, А. Н. Технологии WEB : учебное пособие / А. Н. Петрова. — Комсомольск-на-Амуре : КНАГУ, 2018. — 176 с.

2.Метанит – сайт для изучения языка c# [Электронный ресурс] – режим доступа : https://metanit.com/sharp/(дата обращения: 02.16.2024).

3.Илюшечкин В. М. Основы использования и проектирования баз данных: Учебное пособие. - М.: Юрайт, 2022 - 213 с.

4. Пармезан– интернет-магазин для доставки еды. [Электронный ресурс]. – Режим доступа: https://parmezan.bz/ (дата обращения: 02.16.2024).

* Приложения

Приложение А Схема алгоритма

Приложение Б Текст программы

Приложение В Структура данных

**График выполнения ДИПЛОМНОГО ПРОЕКТА**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование работ** | **Содержание** | **Выполнение работ в %** | **Список исполнения** |
|  | Выбор, выдача темы | Тема работы |  | 07.04.2025 |
|  | Внешнее проектирование (схемы проверки программы, тесты, контроль целостности данных) | Спецификация программы и данных | 20 | 19.05.2025 |
|  | Проектирование (укрупнённый проект)  а) Функциональная схема  б) Схемы данных  в) Структурная схема  г) Схема пользовательского интерфейса  д) Укрупнённый алгоритм программы  Детальное проектирование  Оформлены разделы в пояснительной записке (введение, Описание предметной области)  Разработка в среде программирования  а) Ядро программы  б) Функциональная часть  в) Пользовательский интерфейс | Готовая функциональная часть | 60 | 02.06.2025 |
|  | а) Защитная часть (авторизация, журнал пользователей, шифрование и т.д.)  б) Сервисная часть (настройки, справки и т.д.)  Оформлены разделы в пояснительной записке (Проектирование задачи, Программа решения задачи)  Программа написана в стадии отладки. Тестирование и анализ результатов  Оформлены разделы в пояснительной записке (Тестирование, Применение)Оформлена пояснительная записка полностью | Готовая пояснительная записка | 90 | 09.06.2025 |
|  | Оформление экспериментальной части, подготовка демонстрационных данных и проекта в целом  Подготовка к защите проекта (отзыв, рецензия, доклад) | Получены отзывы и рецензии | 100 | 13.06.2025 |
|  | Нормоконтроль |  |  | 13.06.2025 |
|  | Защита дипломных проектов |  |  | 21.06.2025  28.06.2025 |

Контроль проверки:

|  |  |
| --- | --- |
| 20% | 19.05.2025 |
| 60% | 02.06.2025 |
| 90% | 09.06.2025 |
| 100% | 13.06.2025 |

***ИСХОДНЫЕ ДАННЫЕ***

*В ходе выполнения дипломного проекта необходимо реализовать:*

1. *Спроектировать базу данных удовлетворяющая потребностям:*
2. *Клиенты (личные данные, история записей);*
3. *Врачи (специализация, график работы, отзывы);*
4. *Записи на приём (статус, дата, время);*
5. *Логи системы (действия пользователей).*
6. *Настройка индексов, связей и ограничений.*
7. *Разработка запросов для основных операций (выборка, фильтрация, обновление, удаление).*
8. *Веб-сайт, в котором будут следующие возможности для клиентов (зарегистрированных пользователей):*
9. *Авторизация и аутентификация (регистрация, вход, восстановление пароля).*
10. *Просмотр информации о ветеринарной клинике.*
11. *Онлайн-запись на приём.*
12. *Просмотр истории посещений.*
13. *Оставление отзывов и выставление рейтинга врачам.*
14. *Личный кабинет (редактирование).*
15. *Для администратора:*
16. *Управление записями клиентов.*
17. *Управление врачами (редактирование, удаление, добавление новых специалистов).*
18. *Просмотр и модерация отзывов.*
19. *Управление категориями услуг.*
20. *Просмотр логов системы.*
21. .*Для гостей (неавторизованных пользователей):*
22. *Просмотр информации о клинике.*
23. *Просмотр списка врачей и их специализации.*
24. *Возможность регистрации и входа в систему*

*Задание выдано* 07.04.2025 г.

Дипломник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Пасечникова А.Ф.

Руководитель Дипломного проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Гариев Д.О.

Данные руководителя (ВУЗ, год окончания ВУЗа, инженерный стаж после окончания ВУЗа)

Российский государственный профессионально-педагогический университет, 2019, 6 лет

Содержание

[Введение 9](#_Toc200115934)

[1. Описание предметной области 10](#_Toc200115935)

[2 Назначение и область применения программы 12](#_Toc200115936)

[3 Проектирование задачи 13](#_Toc200115937)

[3.1 Обоснование инструментов разработки 13](#_Toc200115938)

[Таблица 3.1 – Сравнение фреймворков. 13](#_Toc200115939)

[Таблица 3.2 – Сравнение систем управления базами данных 14](#_Toc200115940)

[3.2 Описание алгоритма решения задачи. 15](#_Toc200115941)

[4 Программа решения задачи 17](#_Toc200115942)

[4.1 Логическая структура 17](#_Toc200115943)

[Таблица 4.1 – Логическая структура таблицы Users 18](#_Toc200115944)

[Таблица 4.2 – Логическая структура таблицы Clients 18](#_Toc200115945)

[Таблица 4.3 – Логическая структура таблицы Doctors 19](#_Toc200115946)

[Таблица 4.4 – Логическая структура таблицы Services 19](#_Toc200115947)

[Таблица 4.5 – Логическая структура таблицы DoctorSchedules 20](#_Toc200115948)

[Таблица 4.6 – Логическая структура таблицы Appointments 21](#_Toc200115949)

[Таблица 4.7 – Логическая структура таблицы Reviews 21](#_Toc200115950)

[Таблица 4.8 – Логическая структура таблицы Logs 22](#_Toc200115951)

[5 Тестирование и отладка программы 23](#_Toc200115952)

[6 Применение 25](#_Toc200115953)

[6.1 Назначение программы 25](#_Toc200115954)

[6.2 Требования к аппаратным ресурсам ПК 25](#_Toc200115955)

# Введение

Современный этап развития цифровых технологий оказывает значительное влияние на все сферы деятельности, включая ветеринарные услуги. Внедрение информационных решений становится ключевым фактором повышения эффективности работы организаций, улучшения взаимодействия с клиентами и обеспечения конкурентных преимуществ. В связи с этим создание специализированного веб-сайта для ветеринарной клиники является актуальной задачей, позволяющей оптимизировать процессы записи на прием, информирования клиентов и управления услугами.

Несмотря на доступность цифровых инструментов, многие ветеринарные клиники до сих пор используют устаревшие методы коммуникации, такие как телефонные звонки или бумажные журналы записи. Это создает неудобства для владельцев животных, которые ожидают быстрого доступа к информации и возможности онлайн-взаимодействия. Кроме того, отсутствие удобных онлайн-платформ усложняет работу персонала, увеличивает нагрузку на администраторов и может приводить к ошибкам при записи.

Анализ существующих решений показывает, что большинство доступных веб-сайтов либо не учитывают специфику ветеринарных услуг, либо требуют значительных финансовых затрат на разработку и поддержку, что делает их недоступными для небольших клиник. Таким образом, возникает необходимость в создании адаптивного и функционального выгодного решения, которое сможет удовлетворить потребности как клиентов, так и сотрудников клиники.

Целью данной работы является проектирование и разработка веб-сайта для ветеринарной клиники. Реализация данного проекта позволит повысить качество обслуживания, оптимизировать рабочие процессы и усилить конкурентоспособность клиники на рынке ветеринарных услуг.

# 1. Описание предметной области

Ветеринарная клиника — это учреждение, предоставляющее медицинские услуги для домашних животных, включая диагностику, лечение, профилактику заболеваний и хирургические вмешательства. Основные функции ветеринарной клиники включают организацию приема животных, ведение медицинской документации, планирование и проведение профилактических осмотров, а также обеспечение высококачественного обслуживания клиентов и их питомцев. Участниками системы являются владельцы домашних животных и ветеринары.

Можно выделить несколько типов клиник, различающихся по специализации, уровню предоставляемых услуг и формату работы:

* общая ветеринарная клиника предоставляет широкий спектр услуг для домашних животных: диагностику, лечение заболеваний, вакцинацию, профилактические осмотры, несложные хирургические вмешательства. Обычно такие клиники обслуживают собак, кошек и других распространённых домашних питомцев;
* специализированная ветеринарная клиника ориентирована на определённые виды услуг или животных. Например, клиники, специализирующиеся на стоматологии, офтальмологии, кардиологии, дерматологии или на лечении экзотических животных (рептилий, птиц, грызунов);
* клиника скорой ветеринарной помощи работает круглосуточно и оказывает экстренную медицинскую помощь животным с острыми состояниями, травмами или критическими заболеваниями;
* госпиталь для животных — это крупное медицинское учреждение с развитой инфраструктурой, где доступны сложные диагностические процедуры, хирургия, стационарное лечение и реабилитация. Обычно такие госпитали имеют нескольких специалистов узкой направленности;
* мобильная ветеринарная клиника представляет собой выездную службу, которая оказывает ветеринарные услуги непосредственно на дому у клиента. Удобна для владельцев животных, которым сложно привезти питомца в стационар;
* ветеринарная лаборатория специализируется на проведении различных лабораторных исследований (анализ крови, мочи, биопсии и др.) для постановки точного диагноза;
* ветеринарный центр профилактики и вакцинации, основной акцент которой делается на профилактические меры: вакцинацию, дегельминтизацию, консультации по уходу и питанию животных.

Каждый тип клиники играет важную роль в комплексном обеспечении здоровья домашних животных. Выбор клиники зависит от характера проблемы, вида животного и необходимого уровня медицинской помощи.

# 2 Назначение и область применения программы

Разрабатываемый сайт для ветеринарной клиники предназначен для комплексной цифровизации процессов обслуживания домашних животных, обеспечения прозрачности взаимодействия между владельцами животных, ветеринарами и вспомогательным персоналом, а также повышения вовлеченности клиентов в уход за своими питомцами. Программа реализует следующие ключевые функции:

* регистрация пользователей с защитой данных;
* запись на прием к ветеринару с возможностью выбора времени и врача, что упрощает процесс организации визитов;
* отзывы, что помогает владельцам домашних животных составить мнение о клинике и оставить свое;
* предоставляет информацию об услугах врачей;
* просмотр истории посещений;
* доступ в личный кабинет;
* для администратора предоставляется возможность управлять записями клиентов и редактировать информацию о врачах;
* так же администратор может просматривать логи системы и модерировать отзывы.

Объект исследования: процессы обслуживания домашних животных в рамках ветеринарной клиники, включая документооборот, планирование медицинских услуг, коммуникацию с клиентами и нормативное регулирование.

Предмет исследования: разработка сайта, оптимизирующего указанные процессы за счет цифровизации и интеграции инструментов управления, что способствует повышению качества обслуживания и улучшению взаимодействия между клиентами и клиникой.

# 3 Проектирование задачи

## 3.1 Обоснование инструментов разработки

Для реализации системы управления ветеринарной клиникой выбран ASP.NET — это мощный фреймворк от Microsoft, предназначенный для разработки веб-приложений. Его ключевые преимущества включают:

Высокая производительность: Быстрая обработка запросов и эффективное управление ресурсами, что критично для работы с большим количеством пользователей.

Безопасность: Встроенные механизмы защиты от распространённых уязвимостей, а также поддержка аутентификации и авторизации.

Масштабируемость: легко адаптируется под растущие потребности клиники, включая увеличение числа пользователей и объёма данных.

Интеграция: Возможность интеграции с различными API и сторонними сервисами, что улучшает функциональность приложения.

Сравнение с другими фреймворками представлено в таблице 3.1.

### Таблица 3.1 – Сравнение фреймворков.

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | ASP.NET | Node.js | Django (Python) |
| Производительность | Высокая | Высокая | Средняя |
| Безопасность | Встроенные механизмы защиты | Требует дополнительных усилий | Хорошая |
| Масштабируемость | Легко адаптируется | Легко адаптируется | Сложнее |
| Интеграция | Простой доступ к API и сервисам | Легкая интеграция | Хорошая интеграция |
| Простота разработки | Высокая | Средняя | Высокая |

Microsoft SQL Server  
Microsoft SQL Server — это реляционная система управления базами данных, обеспечивающая надежное хранение и обработку данных. Основные преимущества включают:

Надёжность и производительность: Высокая скорость выполнения запросов и поддержка транзакций.

Сложные запросы: Возможность выполнения сложных SQL-запросов для анализа данных.

Безопасность данных: Многоуровневая защита информации, включая шифрование и управление правами доступа.

Резервное копирование: Надёжные механизмы резервного копирования и восстановления данных.

Сравнение с другими СУБД представлено в таблице 3.2.

### Таблица 3.2 – Сравнение систем управления базами данных

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Microsoft SQL Server | MySQL | PostgreSQL |
| **Надёжность** | Высокая | Средняя | Высокая |
| **Производительность** | Высокая | Высокая | Высокая |
| **Безопасность** | Многоуровневая защита | Хорошая | Хорошая |
| **Поддержка транзакций** | Да | Да | Да |
| **Сложные запросы** | Да | Ограниченная | Да |

Таким образом, выбор ASP.NET и Microsoft SQL Server обусловлен их высокой производительностью, безопасностью, удобством интеграции и возможностями масштабирования, что делает их лучшими инструментами для разработки системы управления ветеринарной клиникой.

## **3.2 Описание алгоритма решения задачи.**

Основная задача — автоматизация процессов записи на приём, ведения медицинской документации и управления взаимодействием между владельцами животных и ветеринарами.

Алгоритм решения включает следующие этапы:

1. Создание базы данных.

Создание таблиц и установление связей между ними.

1. Создание интерфейса сайта.

Интерфейс должен быть интуитивно понятен и прост.

1. Подключение базы данных к программе.
2. Регистрация и аутентификация пользователей.

Пользователи (владельцы животных и администратор) регистрируются в системе, создают учетные записи, проходят процедуру входа с проверкой прав доступа.

1. Обеспечение безопасности.

Реализуются меры по защите данных, включая шифрование паролей, разграничение доступа и защиту от атак.

1. Управление профилями.

Пользователи могут редактировать свои данные: владельцы — информацию о питомцах, контакты; администратор — специализацию, расписание врачей.

1. Просмотр и выбор услуг.

Владельцы животных просматривают список доступных ветеринаров, их специализации, услуги и расписание.

1. Запись на приём.

Пользователь выбирает ветеринара и удобные дату и время, система проверяет доступность и сохраняет запись.

1. Ведение медицинской документации.

Ветеринары фиксируют результаты осмотров, диагнозы, назначения и рецепты.

1. Управление отзывами и рейтингами.

Владельцы могут оставлять отзывы и оценки врачам, что помогает улучшать качество обслуживания. А администратор просматривать и редактировать их.

1. Логирование действий.

Все ключевые действия пользователей фиксируются в системе для обеспечения контроля и анализа. Администратору предоставляется возможность просматривать их.

1. Обработка исключений и ошибок.

Система должна корректно обрабатывать ошибки ввода, конфликтные ситуации (например, двойное бронирование) и обеспечивать информирование пользователей. Таким образом, алгоритм решения задачи обеспечивает комплексное управление процессами ветеринарной клиники, повышая эффективность работы и качество обслуживания клиентов.

# 4 Программа решения задачи

## 4.1 Логическая структура

Логическая модель данных — это усовершенствованная версия концептуальной модели. В ней представлены ограничение данных, имена сущностей, ключевые поля и связи.

На рисунке 4.1 представлена ER-диаграмма базы данных.

Между таблицами связь установлена с помощью ключевых значений.

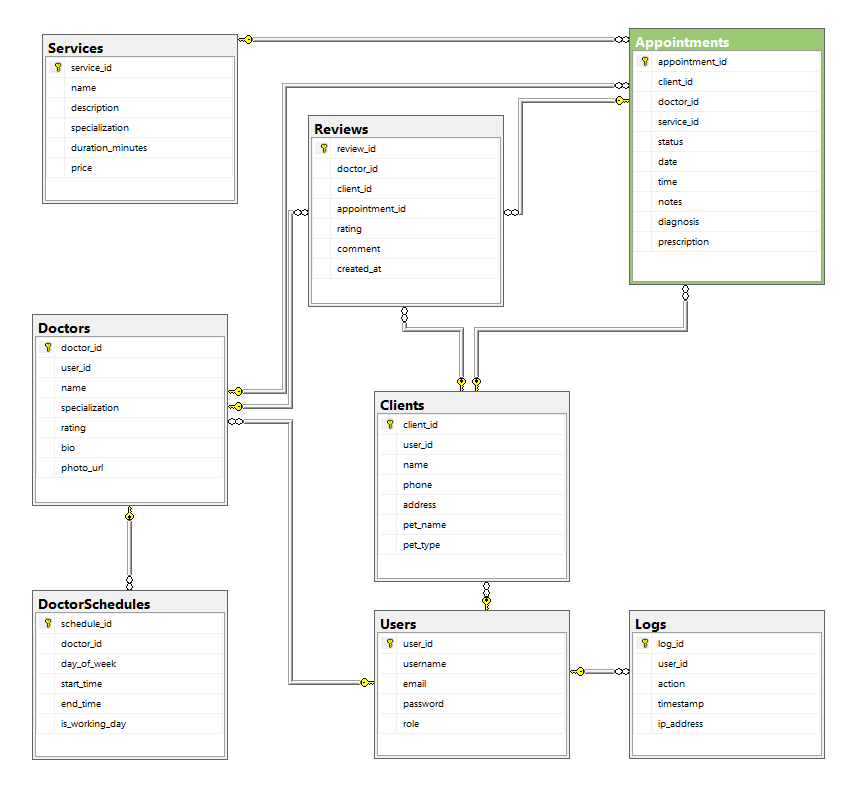


Рисунок 4.1 - ER-диаграмма базы данных

В таблице 4.1 построена логическая структура таблицы Users которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о пользователях, зарегистрированных в системе, такую как: индивидуальный код пользователя, имя пользователя, email и пароль.

### Таблица 4.1 – Логическая структура таблицы Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Users | Пользователи | user\_id | С | int |
| username | Т | nvarchar(50) |
| email | Т | nvarchar(50) |
| password | Т | nvarchar(50) |
| role | Т | nvarchar(10) |

В таблице 4.2 построена логическая структура таблицы Clients которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о животных и их владельцах, зарегистрированных в системе, такую как: индивидуальный код владельца питомца, индивидуальный код пользователя под которым он зарегистрирован, ФИО, телефон, адрес, имя и тип питомца.

### Таблица 4.2 – Логическая структура таблицы Clients

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Clients | Владельцы животных | client\_id | С | int |
| user\_id | С | int |
| name | Т | nvarchar(50) |
| phone | Т | nvarchar(15) |
| address | Т | nvarchar(100) |
| pet\_name | Т | nvarchar(50) |
| pet\_type | Т | nvarchar(50) |

В таблице 4.3 построена логическая структура таблицы Doctors которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о врачах, зарегистрированных в системе, такую как: индивидуальный код врача, индивидуальный код пользователя под которым зарегистрирован ветеринар, ФИО врача, специализация, рейтинг, биография и фотография ветеринара.

### Таблица 4.3 – Логическая структура таблицы Doctors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Doctors | Ветеринары | doctor\_id | С | int |
| user\_id | С | int |
| name | Т | nvarchar(50) |
| specialization | Т | nvarchar(50) |
| rating | Ч | int |
| bio | Т | text |
| photo\_url | Т | nvarchar(255) |

В таблице 4.4 построена логическая структура таблицы Services которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию об услугах, как: индивидуальный код услуги, название услуги, описание услуги, специализация врача, время выполнения процедуры и ее цена.

### Таблица 4.4 – Логическая структура таблицы Services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Services | Услуги | int | С | int |
| nvarchar(100) | Т | nvarchar(100) |
| text | Т | text |
| nvarchar(100) | Т | nvarchar(100) |

Продолжение таблицы 4.4 – Логическая структура таблицы Services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Services | Услуги | int | Ч | int |
| decimal(10, 2) | Д | decimal(10, 2) |

В таблице 4.5 построена логическая структура таблицы DoctorSchedules которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о расписании врача, такую как: индивидуальный код расписания, индивидуальный код врача, день начала работы, время начала работы, время окончания работы и рабочий ли это день на неделе.

### Таблица 4.5 – Логическая структура таблицы DoctorSchedules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| DoctorSchedules | Расписание врача | schedule\_id | С | int |
| doctor\_id | С | int |
| day\_of\_week | Ч | tinyint |
| start\_time | В | time(7) |
| end\_time | В | time(7) |
| is\_working\_day | Л | bit |

В таблице 4.6 построена логическая структура таблицы Appointments которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о записях на прием, такую как: индивидуальный код записи, индивидуальный код клиента, индивидуальный код врача, индивидуальный код услуг, статус записи, дата, время, примечание, диагноз и рецепт.

### Таблица 4.6 – Логическая структура таблицы Appointments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Appointments | Записи | appointment\_id | С | int |
| client\_id | С | int |
| doctor\_id | С | int |
| service\_id | С | int |
| status | Т | nvarchar(20) |
| date | ДТ | date |
| time | В | time(7) |
| notes | Т | text |
| diagnosis | Т | text |
| prescription | Т | text |

В таблице 4.7 построена логическая структура таблицы Reviews которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о записях на прием, такую как: индивидуальный код отзыва, индивидуальный код врача, индивидуальный код клиента, индивидуальный код записи, рейтинг, комментарий, время отправки и статус.

### Таблица 4.7 – Логическая структура таблицы Reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Reviews | Отзывы | review\_id | С | int |
| doctor\_id | С | int |
| client\_id | С | int |
| appointment\_id | С | int |
| rating | Ч | int |

Продолжение таблицы 4.7 – Логическая структура таблицы Reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Reviews | Отзывы | comment | Т | text |
| created\_at | ДТ | datetime |
| status | Т | nvarchar(20) |

В таблице 4.8 построена логическая структура таблицы Logs которая отображает названия сущностей, названия атрибутов, типы данных. Эта таблица содержит информацию о записях на прием, такую как: индивидуальный код отзыва, индивидуальный код врача, индивидуальный код клиента, индивидуальный код записи, рейтинг, комментарий, время отправки и статус.

### Таблица 4.8 – Логическая структура таблицы Logs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя сущности | Имя сущности в переводе | Имя атрибута | Тип | Формат |
| Logs | Логистика системы | log\_id | ДТ | int |
| user\_id | С | int |
| action | С | text |
| timestamp | Т | datetime |
| ip\_address | ДТ | nvarchar(50) |

# 5 Тестирование и отладка программы

Тестирование – это способ проверки разрабатываемой программы на соответствие ожидаемым результатам и поиск дефектов для их дальнейшего исправления. Этот этап помогает предотвратить будущие ошибки и сбои.

Тестирование предполагает анализ компонентов ПО с помощью ручных или автоматизированных тестов для оценки новой и текущей функциональности.

Так же существуют разные цели тестирования:

* функциональное тестирование – проверка, что программа работает в соответствии с требованиями.
* нефункциональное тестирование – проверка нефункциональных аспектов, таких как производительность, безопасность и удобство использования.

Для проверки программы будет использоваться ручное тестирование, с целью проверить функционал программы.

Преимущества данного способа:

* гибкость и адаптируемость. Ручное тестирование отличается высокой гибкостью и подходит для динамичных проектов, где требования могут меняться;
* экономическая эффективность. В небольших проектах может иметь преимущество;
* человеческое суждение и интуиция. Делает тестирование более направленным на пользователя;
* разнообразие тестовых примеров.

Но также у данного способа имеются и недостатки, такие как:

* субъективность. Результат зависит от того какие тесты будет проводить тестировщик.
* не подлежит повторному использованию. Тесты, написанные для более старых версий, станут не актуальны при будущих обновлениях. Из-за чего их придётся писать повторно

Так как проект является не большим и тестирование проводит один человек, недостатки не являются критичными, а преимущества берутся в больший расчет.

Для проверки работы гостевой части системы необходимо провести следующие тесты:

* проверка работоспособности навигационной панели и её свойств;
* проверка работоспособности кнопок и других интерактивных элементов;
* проверка работоспособности подключенных стилей;
* проверка работоспособности отображаемой информации пользователю;
* проверка работоспособности подгружаемых файлов
* проверка работоспособности страниц;
* проверка работоспособности стилей и страниц в различных браузерах (Microsoft Edge, Opera, Google Chrome, Яндекс Браузер);
* проверка адаптивности интерфейса к различным форматам;
* проверка доступности и лёгкости интерфейса в понимании.

Для проверки работы клиентской части системы необходимо провести следующие тесты:

* проверка отображения только нужных частей интерфейса;
* проверка на ввод корректных и некорректных данных.

Для проверки работы части системы администратора необходимо провести следующие тесты:

* проверка правильности отображения логистики системы.

# 6 Применение

## 6.1 Назначение программы

Веб-сайт для ветеринарной клиники предназначен для владельцев домашних животных и для администратора клиники. Данный сайт ускорит работу сотрудников клиники, за счет быстрого доступа к нужным данным. Владельцам станет проще записать своего питомца к ветеринару, узнать актуальные цены и расписания врачей, а так же мнение других пользователей о них.

## 6.2 Требования к аппаратным ресурсам ПК

Минимальные требования к аппаратным ресурсам ПК:

* процессор: 3 GHz Intel Pentium 4;
* оперативная память: 1 Гб ОЗУ;
* место на диске: 2 Гб;
* ОС: Windows 7/8/10/11;
* наличие манипулятора «мышь»;
* наличие средства вывода информации «монитор» с разрешением не менее 800\*600 точек;
* наличие средства ввода информации «клавиатура»;
* Для работы компьютеры должны быть оснащены сетевыми адаптерами или должны быть подключены проводным или иным соединением к сети Интернет.