1. **Общие сведения о программе**

Программа предназначена для автоматизации процесса управления заявками в системе обслуживания клиентов. Она позволяет пользователям создавать, редактировать, просматривать и удалять заявки, а также отслеживать их статус и взаимодействие с клиентами и исполнителями. Система используется для повышения эффективности работы с заявками, улучшения организации работы сотрудников и сокращения времени обработки заявок.

Основные функции:

Создание заявки — Пользователи могут создавать заявки, указав информацию о клиенте, исполнителе, статусе и другие данные.

Редактирование заявки — Пользователи могут изменять данные заявок, такие как статус, исполнитель, описание и другие параметры.

Просмотр заявок — Возможность просмотра детальной информации по каждой заявке, включая информацию о клиенте, исполнителе, статусе и времени выполнения.

Удаление заявки — Пользователи могут удалять заявки из системы.

Отслеживание статусов — Заявки можно отслеживать по текущему статусу, что позволяет быстро получить информацию о ходе выполнения заявки.

Управление клиентами и исполнителями — Система предоставляет возможность добавлять, редактировать и удалять клиентов и исполнителей, а также привязывать их к заявкам.

1. **Структура программы**

Составные части программы:

Программа состоит из нескольких основных компонентов:

* Контекст базы данных (AppDbContext): Отвечает за взаимодействие с базой данных, включая таблицы заявок, клиентов, исполнителей и статусов.
* Модели данных: Каждая сущность (заявка, клиент, исполнитель, статус) представлена отдельной моделью, которая содержит соответствующие свойства.
* Страницы Razor: Представляют собой интерфейс для взаимодействия с пользователем (например, создание, редактирование, просмотр заявок).
* Логика обработки данных: Включает методы для работы с заявками, такими как создание, обновление, удаление и поиск.

Связь и структура компонентов:

* Модели данных (Client, Request, Executor, Status) связаны с базой данных через AppDbContext.
* Страницы Razor взаимодействуют с данными через контроллеры, которые используют AppDbContext для выполнения операций с базой данных.
* Данные заявок, клиентов, исполнителей и статусов передаются через контроллеры на страницы, где они отображаются пользователю.

1. **Настройка программы**

Требования к программному обеспечению

СУБД: Microsoft SQL Server или PostgreSQL для хранения данных.

Среда разработки: Microsoft Visual Studio или Visual Studio Code.

Фреймворк: .NET Core для работы с серверной логикой.

Настройка базы данных

Установите и настройте Microsoft SQL Server или PostgreSQL.

Создайте базу данных с названием, например, TaskRequestDb.

Проверьте успешность подключения к базе данных через AppDbContext.

1. **Проверка программы**

Проверка установки программы

* После установки и настройки базы данных, выполните запуск программы.
* Проверьте, что приложение подключается к базе данных без ошибок.
* Введите данные в форму создания заявки и сохраните их. Убедитесь, что заявка сохраняется в базе данных.

Тестирование функциональности

* Создание заявки: Протестируйте процесс создания заявки. Проверьте, что все введенные данные корректно сохраняются.
* Редактирование заявки: Измените статус заявки и убедитесь, что изменения применяются.
* Удаление заявки: Убедитесь, что заявка удаляется из системы и больше не отображается в списке.

1. **Дополнительные возможности**

Сортировка списка заявок:

Возможность сортировки списка заявок по различным критериям, таким как дата добавления, номер заявки, статус или исполнитель. Пользователь может выбирать, по какому полю сортировать, и в каком порядке (по возрастанию или убыванию).

1. **Сообщения системному программисту**

В процессе работы программы могут быть выведены следующие сообщения, которые помогут в диагностике ошибок:

* Ошибка подключения к базе данных:
* «Не удается подключиться к базе данных. Проверьте строку подключения в конфигурации.»
* Ошибка валидации данных:
* «Ошибка: обязательные поля не могут быть пустыми.»
* Ошибка обновления данных:
* «Ошибка при обновлении заявки. Повторите попытку позже.»
* Ошибка удаления заявки:
* «Заявка с указанным ID не существует в базе данных.»