«CRM-система для автоматизации детских центров Отмечалка» - это система автоматизации, предоставляющая полный набор инструментов, которые помогут раскрыть потенциал детского центра, повысить его прибыльность и конкурентоспособность.

Из плюсов это программы можно отметить то что в ней есть бесплатный 14 дневный период, постоянное обновление программы и высокая надёжность хранения данных.

В минусы этой программы также входят можно записать необходимость платить за её использование, а также необходимость подключения к сети интернет.

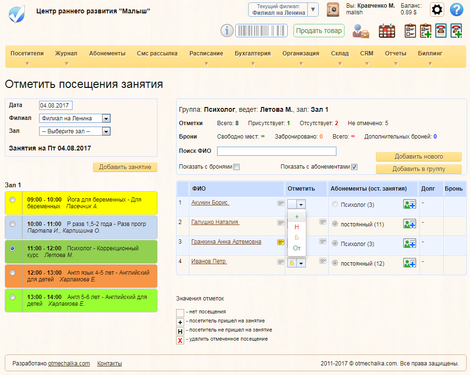


Рисунок 1.2 – «CRM-система Отмечалка»

Сравнительный анализ показал, что «Центр детского творчества», разрабатываемый в рамках курсового проектирования, по основным характеристикам не уступает существующим аналогам.

## 1.2 Цели и задачи проектирования

Целью данного курсового проекта является реализация программной системы «Цент детского творчества» для автоматизации деятельности и обеспечения хранения, накопления и предоставления информации.

Главная цель, которая стоит перед написанием курсового проекта – это приобретение практических навыков проектирования базы данных «Центр детского творчества» с использованием программ СУБД. В ходе выполнения данного курсового проекта необходимо будет разработать информационную базу данных и программное обеспечение для работы совместно с базой данных для центра детского творчества, которые помогут пользователю легко найти нужную информацию о сотруднике, детях, группах и событиях в любом момент времени

Предметом курсового проекта является проектирование информационной базы данных и создание программного обеспечения для работы совместно с базой данных для центра детского творчества.

Задачи проекта:

* Анализ предметной области;
* Построение инфологической (концептуальной) модели предметной области;
* Разработка логической структуры базы данных;
* Разработка физической структуры базы данных;
* Организация базы данных;
* Разграничение прав доступа.
* Разработка пользовательского интерфейса;
* Реализация проекта в конкретной среде программирования;
* Наполнение базы данных;

Входной информацией в создаваемой программе будут являться запросы пользователей.

Выходной информацией является подробная информация о детях, сотрудниках, группах и событиях представленная в виде таблиц.

Система будет много пользовательской.

Программное обеспечение системы не должно зависеть от аппаратных средств компьютера.

## ПРОЕКТНАЯ ЧАСТЬ

## Проектирование схемы данных

Проектирование базы данных осуществляется в три этапа: концептуальное (инфологическое) проектирование, логическое проектирование и физическое проектирование;

Цель инфологического этапа проектирования состоит в получении семантических (концептуальных) моделей, отражающих предметную область и информационные потребности пользователей. В качестве инструмента для построения семантических моделей данных на этапе инфологического проектирования является неформальная модель "Сущность-Связь" (ER-модель - Entity-Relationship). Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов.

Основными понятиями ER-модели являются сущность, связь и атрибут.

Сущность (объект) - это реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступна. Различают такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных предметов, событий, личностей, выступающих как единое целое. Экземпляр сущности относится к конкретной вещи в наборе. В диаграммах ER-модели сущность представляется в виде прямоугольника (в нотации Баркера), содержащего имя сущности.

Выделим базовые сущности данного курсового проекта:

* Дети. Атрибуты: информация о детях, родители.
* События. Атрибуты – описание, дата проведения, заметки, цена.
* Группы. Атрибуты – сотрудник, класс, расписание, размер группы, цена.
* Сотрудники. Атрибуты – ФИО, дата рождения, квалификация.

Родители и Информацию о детях будем рассматривать как связь с детьми. Атрибуты родителей: ФИО мамы, ФИО папы, дата рождения мамы, дата рождения папы. Атрибуты информации о детях: ФИО, дата рождения, адрес.

Класс будем рассматривать как связь с группой. Атрибуты класса – название, описание, заметки, категория. А Категории рассматриваем как связь с классом. Атрибуты категории – название, описание.

Лист в данной работе рассматривается как связь между такими сущностями как дети, события и группы.

Атрибут - поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества значений. Каждый атрибут обеспечивается именем, уникальным в пределах сущности.

Связь (Relationship) - это поименованная графически изображаемая ассоциация, устанавливаемая между сущностями и представляющая собой абстракцию набора отношений, которые систематически возникают между различными видами предметов в реальном мире. Большинство связей относятся к категории бинарных и имеют место между двумя сущностями.

Среди бинарных связей существуют три фундаментальных вида связи: один-к-одному (1:1), один-ко-многим (1:M), многие-ко-многим (M:M). Связь один-к-одному (1:1) существует, когда один экземпляр одной сущности связан с единственным экземпляром другой сущности. Связь один-ко-многим (1:M) имеет место, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан только с одним экземпляром первой сущности. Связь многие-ко-многим (М:N) существует, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан с одним или более экземпляром первой сущности.

В данной курсовой работе предоставлены все три вида связей. Инфологическая (концептуальная) модель предметной области представлена в Приложении В.

Функциональные возможности:

* ведение БД (запись, чтение, модификация, удаление);
* обеспечение логической непротиворечивости БД;
* реализация наиболее часто встречающихся запросов в готовом виде;
* предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

Запросы:

* получение списка всех групп, включая информацию о детях и классах;
* получение списка всех событий, включая информацию о детях;
* получение списка персонала;
* получение списка детей;
* получение списка полной информации о детях;
* получение списка родителей;
* получение списка классов;
* получение списка категорий;
* удаление данных из выше перечисленных списков;
* редактирование данных из выше перечисленных списков;
* добавление данных в выше перечисленные списки;

Логическое проектирование это проектированием логической структуры БД, что означает определение всех информационных единиц и связей между ними, задание их имен и типов, а также некоторых количественных характеристик (например, длины поля).

При проектировании логической структуры БД, осуществляется преобразование исходной инфологической модели в модель данных, поддерживаемую конкретной СУБД, и проверка адекватности, полученной логической модели отображаемой предметной области.

Схема базы «Системы управления «Центр детского творчества» представлена в приложении Б.

Описание и структура спроектированной базы данных:

Категории (id (счетчик (Длинное целое)), название (Текстовый), описание(Текстовый));

Дети (id (счетчик (Длинное целое)), id информации о ребёнке (счетчик (Длинное целое)), id родителей (счетчик (Длинное целое)));

Информация о детях (id (счетчик (Длинное целое)), ФИО (Текстовый), Дата рождения (Дата), адрес (Текстовый));

Класс (id (счетчик (Длинное целое)), название (Текстовый), описание (Текстовый), заметки (Текстовый), категория (счетчик (Длинное целое)));

События (id (счетчик (Длинное целое)), описание (Текстовый), дата проведения(Дата/Время), заметки (Текстовый), цена (Числовой (Длинное целое)));

Группы (id (счетчик (Длинное целое)), учитель (счетчик (Длинное целое)), тип группы (счетчик (Длинное целое)), расписание (Текстовый), размер группы (Числовой (Длинное целое)), цена (Числовой (Длинное целое)));

Список (id (счетчик (Длинное целое)), id ребёнка (счетчик (Длинное целое)), id группы (счетчик (Длинное целое)) id события (счетчик (Длинное целое)));

Родители (id (счетчик (Длинное целое)), ФИО матери (Текстовый), Дата рождения матери (Дата), ФИО отца (Текстовый), Дата рождения отца (Дата));

Персонал (id (счетчик (Длинное целое)), ФИО (Текстовый), Дата рождения (Дата), квалификация (Текстовый));

Удаленные события (id (счетчик (Длинное целое)), описание (Текстовый), дата проведения(Дата/Время), заметки (Текстовый), цена (Числовой (Длинное целое)));

Удаленные Группы (id (счетчик (Длинное целое)), учитель (счетчик (Длинное целое)), тип группы (счетчик (Длинное целое)), расписание (Текстовый), размер группы (Числовой (Длинное целое)), цена (Числовой (Длинное целое)));

Пользователь (id (счетчик (Длинное целое)), логин (Текстовый), пароль (Текстовый), если Админ (Числовое (Короткое целое)), последняя дата входа (Дата/Время), последняя дата выхода (Дата/Время));

## 2.2 Обоснование выбора технологий

C# — объектно-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Основные достоинства С#:

* C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности — как FCL, так и CLR;
* C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
* C# является мощным объектным языком с возможностями наследования и универсализации;
* C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от С++ к C#;
* сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
* благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;
* мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
* реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#

## Описание среды реализации

СУБД MySql — предоставляет мощные средства для доступа, настройки, администрирования, разработки всех компонентов базы данных и управления ими. MySql — это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySql — свободно распространяемая система. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД. Работать с MySql можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс для работы с этой СУБД — PhpMyAdmin. Этот интерфейс позволяет значительно упростить работу с базами данных в MySql.

PhpMyAdmin позволяет пользоваться всеми достоинствами браузера, включая прокрутку изображения, если оно не умещается на экран. Многие из базовых SQL-функций работы с данными в PhpMyAdmin сведены к интуитивно понятным интерфейсам и действиям, напоминающим переход по ссылкам в Internet.

PHPMyAdmin — это веб-приложение, которое распространяется с открытым кодом, написанное на языке web-программирования PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL (Что это такое?). PHPMyAdmin для работы с базой данных нужен браузер, который и будет передавать на сервер все команды. В качестве языка работы с БД используется широко известный SQL.

PHPMyAdmin широко распространён по всему миру и является одним из лидеров в работе СУБД. Интерфейс доступен более чем на 60 языках мира. Благодаря простому и понятному интерфейсу данным приложением могут пользоваться даже начинающие пользователи.

Возможности PHPMyAdmin

1. Создание, просмотр, редактирование и удаление: баз данных, пользователей, таблиц и записей в них.

2. Удобный и понятный импорт и экспорт БД, таблиц и записей.

3. Понятный поиск

4. Администрирование БД и пользователей

5. Подсказки при обработке SQL и подсветка синтаксиса

6. Сообщения о выполнении действия (в процессе, изменено и тп)

Microsoft Visual Studio — линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. VisualStudio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (например, Subversion и VisualSourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода). Главным преимуществом Visual Studio 2017 является производительность. Обеспечивает возможность создания разнообразных приложений на основе одного набора навыков.

Интегрированная среда разработки (IntegratedDevelopmentEnvironment - IDE) Visual Studio предлагает ряд высокоуровневых функциональных возможностей, которые выходят за рамки базового управления кодом.

Ниже перечислены основные преимущества IDE-среды Visual Studio.

Встроенный Web-сервер. Для обслуживания Web-приложения ASP.NET необходим Web-сервер, который будет ожидать Web-запросы и обрабатывать соответствующие страницы. Наличие в Visual Studio интегрированного Web-сервера позволяет запускать Web-сайт прямо из среды проектирования, а также повышает безопасность, исключая вероятность получения доступа к тестовомуWeb-сайту с какого-нибудь внешнего компьютера, поскольку тестовый сервер может принимать соединения только с локального компьютера.

Поддержка множества языков при разработке. Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках, используя все время один и тот же интерфейс (IDE). Более того, Visual Studio также еще позволяет создавать Web-страницы на разных языках, но помещать их все в одно и то же Web-приложение. Единственным ограничением является то, что в каждой Web-странице можно использовать только какой-то один язык (очевидно, что в противном случае проблем при компиляции было бы просто не избежать).

Меньше кода для написания. Для создания большинства приложений требуется приличное количество стандартного стереотипного кода, и Web-страницы ASP. NET тому не исключение. Например, добавление Web-элемента управления, присоединение обработчиков событий и корректировка форматирования требует установки в разметке страницы ряда деталей. В Visual Studio такие детали устанавливаются автоматически.

Интуитивный стиль кодирования. По умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам. Применяемые Visual Studio автоматически параметры форматирования можно даже настраивать, что очень удобно в случаях, когда разработчик предпочитает другой стиль размещения скобок (например, стиль K&R, при котором открывающая скобка размещается на той же строке, что и объявление, которому она предшествует).

## 4 ТЕСТИРОВАНИЕ

Тестирование (testing) программного обеспечения (ПО) - это процесс исследования ПО с целью выявления ошибок и определения соответствия между реальным и ожидаемым поведением ПО, осуществляемый на основе набора тестов, выбранных определённым образом. В более широком смысле, тестирование ПО - это техника контроля качества программного продукта, включающая в себя проектирование тестов, выполнение тестирования и анализ полученных результатов.

Очень часто современные программные продукты разрабатываются в сжатые сроки и при ограниченных бюджетах проектов. Программирование сегодня перешло из разряда искусства в разряд ремесел для многих миллионов специалистов. Но, к сожалению, в такой спешке разработчики зачастую игнорируют необходимость обеспечения защищённости своих продуктов, подвергая тем самым пользователей неоправданному риску. Контроль качества (тестирование) считается важным в процессе разработки ПО, потому что обеспечивает безопасность, надёжность, удобство создаваемого продукта. В настоящее время существует великое множество подходов и методик к решению задачи тестирования ПО, но эффективное тестирование сложных программных систем - процесс творческий, не сводящийся к следованию строгим и чётким правилам.

В рамках курсового проекта, программное средство будет протестировано одним из распространенных видов тестирования – функциональным тестированием.

Функциональное тестирование (functional testing) – вид тестирования, направленный на проверку корректности работы функциональности приложения (корректность реализации функциональных требований.

## Результаты тестирования программного продукта представлены в таблицах 4.1 – 4.3.

## Таблица 4.1 – Результаты тестирования формы пользователя

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Условия | Действие | Ожидаемый результат | Итог |
| 1 | 2 | 3 | 4 | 5 |
| 1 | При нажатии на вкладку | Нажатие на вкладку | Загрузка таблицы с данными | пройден |
| 2 | При поиске | Ввод данных в строку поиска и нажатие на кнопку «Search» | Обновление таблицы в соответствии с условиями поиска | пройден |

## Продолжение таблицы 4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | При сбросе данных поиска | Нажатие на кнопку «Reset» | Обновление таблицы | пройден |
| 4 | При сортировке данных | Нажатие на заголовки таблицы | Сортировка данных в таблице в соответствии с выбранными параметрами | пройден |

## Таблица 4.2 – Результаты тестирования формы авторизации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Условия | Действие | Ожидаемый результат | Итог |
| 1 | 2 | 3 | 4 | 5 |
| 1 | При верном вводе данных для авторизации администратора и нажатии на кнопку «Sign in» | Ввод данных для авторизации и нажатие на кнопку «Sign in» | Авторизация и открытие нового окна с формами для работы от лица администратора | пройден |
| 2 | При верном вводе данных для авторизации пользователя и нажатии на кнопку «Sign in» | Ввод данных для авторизации и нажатие на кнопку «Sign in» | Авторизация и открытие нового окна с формами для работы от лица пользователя | пройден |
| 3 | При неверном вводе данных для авторизации и нажатии на кнопку «Sign in» | Ввод данных для авторизации и нажатие на кнопку «Sign in» | Открытие нового окна с выводом ошибки | пройден |

Таблица 4.3 – Результаты тестирования формы администратора

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Условия | Действие | Ожидаемый результат | Итог |
| 1 | 2 | 3 | 4 | 5 |
| 1 | При нажатии на вкладку | Нажатие на вкладку | Загрузка таблицы с данными, отображение кнопки сохранения данных в таблицу | пройден |
| 2 | При добавлении данных | Ввод данных в таблицу и нажатие кнопки сохранить | Добавление данных в бд и обновление таблицы | пройден |
| 3 | При редактировании данных | Редактирование данных в таблице и нажатие кнопки сохранить | Обновление данных в бд и обновление таблицы | пройден |
| 4 | При удаление данных | Удаление данных в таблице | Удаление данных в бд и обновление таблицы | пройден |

Выше были приведены основные проверки форм программного средства. По итоговому результату можно сделать вывод о том, что программное средство выполняет все необходимые функции.

## ОПИСАНИЕ ПРИМЕНЕНИЯ

Программный продукт «Система управления «Центр детского творчества» поможет пользователю легко найти нужную информацию о сотруднике, детях, группах и событиях в любом момент времени, автоматизировать все рутинные процессы по учету и управлению данными.

Сфера применения программы «Система управления «Центр детского творчества» это администрирование детских развивающих центров, детских творческих клубов, центров раннего развития и предприятий подобной направленности.

С помощью программы «Центр детского творчества» можно будет выполнять следующие функции:

* Авторизация пользователей;
* Вести клиентскую базу детей с учетом их сопровождающих (родители, родственники);
* Вести клиентскую базу педагогов;
* Проводить мероприятия;
* В удобном виде регистрировать посещения детей на занятия;
* Планировать расписание занятий на любой период;
* Составлять график работы педагогов;
* И много другое…

Входной информацией в создаваемой программе будут являться запросы пользователей.

Выходной информацией является подробная информация о детях, сотрудниках, группах и событиях представленная в виде таблиц.

Система будет много пользовательской.

Программное обеспечение системы не должно зависеть от аппаратных средств компьютера.

## ЗАКЛЮЧЕНИЕ

Результатом выполнения курсовой работы стало разработанное приложение баз данных, позволяющее автоматизировать рутинные процессы по учету и управлению данными. в детских центрах. Разработанное приложение отвечает всем требованиям предметной области, таблицы созданной базы данных отвечают требованиям нормализации, что позволяет обеспечить целостность и непротиворечивость информации. В данной работе разработана база данных и приложение для работы совместно с базой данных.

В процессе выполнения курсовой работы были закреплены знания, полученные при изучении дисциплины. Были изучены такие пункты:

* анализ предметной области;
* построение концептуальной модели базы данных;
* организация базы данных;
* разработка прикладной программы;
* наполнение и сопровождение базы данных;

В процессе организации БД проведен до необходимого уровня абстракций анализ предметной области, построена реляционная модель БД, произведена нормализация реляционной БД. Реализация проекта была выполнена на современных программных платформах. В качестве технологии доступа к данным была использована объектно-реляционная модель, которая позволяет просто и лаконично осуществлять запросы к базе данных.

Даная курсовая работа была реализована на языке программирования C# с использованием среды разработки Microsoft Visual Studio.

Для проверки корректности работы программы были изучены методы тестирования и проведены тесты, по результатам которых были исправлены ошибки.

Цель и соответствующие задачи, поставленные перед выполнение курсового проекта выполнены в полном объёме.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] В.А.Гвоздева, И.Ю.Лаврентьева. Основы построения АИС, Москва, ИД «Форум» - ИНФРА-М, 2009.

[2] А.В.Рудаков, Технология разработки программных продуктов, Москва, Издательский центр «Академия», 2008

[3] Л.Г.Гагарина, Д.В.Киселев, Е.Л.Федотова, Разработка и эксплуатация АИС. Москва, ИД «Форум» - ИНФРА-М, 2009.

[4] Г.Ю.Максимович, А.Г.Романенко, О.Ф.Самойлюк. Информационные системы. Москва 2007, Федеральное агентство по образованию

[5] Диго С.М. Базы данных: проектирование и использование: Учебник. - М.: Финансы и статистика, 2005.

[6] Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. Учебник для вузов. - М.: Корона-принт, 2004.

[7] Кузин А.В., Левонисова С.В. Базы данных: Учебник. - М.: Academia, 2010.

[8] Джон Дей, Крейг Ван Слайк, Рэймонд Фрост Базы данных. Проектирование и разработка: Учебник. - М.: НТ Пресс, 2007.

[9] Домбровская Г.Р., Новиков Б.А. Настройка приложений баз данных: Учебник - М.: BHV, 2006.

[10] Туманов В.Е. Основы проектирования реляционных баз данных: Учебное пособие. - М.: Интернет-университет информационных технологий, 2010.

[11] Илюшечкин В. М. Основы использования и проектирования баз данных: Учебное пособие. - М.: Юрайт, 2010.

[12] Преснякова Г.В. Проектирование интегрированных реляционных баз данных: Учебник. - М.: КДУ, 2007.

[13] http://www.sql.ru/ - портал про язык SQL и клиент/серверные технологии.

[14] http://sql.itsoft.ru/ - интернет-справочник с примерами по языку SQL.

[15] http://www.cyberguru.ru/database/database-theory/ - статьи по теории баз данных.

## ПРИЛОЖЕНИЕ А

(ОБЯЗАТЕЛЬНОЕ)

ФРАГМЕНТ ИСХОДНОГО КОДА

using MaterialSkin;

using MaterialSkin.Controls;

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace ChildCentre

{

public partial class Main : MaterialForm

{

User user = new User();

public Main()

{

InitializeComponent();

var materialSkinManager = MaterialSkinManager.Instance;

materialSkinManager.AddFormToManage(this);

materialSkinManager.Theme = MaterialSkinManager.Themes.LIGHT;

materialSkinManager.ColorScheme = new ColorScheme(Primary.BlueGrey800, Primary.BlueGrey900, Primary.BlueGrey500, Accent.LightBlue200, TextShade.WHITE);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

Tables table = new Tables();

table.Show();

this.Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

Admin admin = new Admin();

admin.Show();

this.Hide();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{