مستندات پروژه

موضوع پروژه : برنامه گل فروشی

نام استاد محترم:

تهیه کننده:



**فهرست مطالب . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . شماره صفحه**

چکیده 4

مقدمه 5

فصل اول: چشم انداز پروژه (Vision) 6

**1-1- Vision چیست** 7

**1-2- موضوع پروژه** 8

**1-3- اهداف پروژه** 8

**1-4- انجام پروژه چه سودی دارد** 8

**1-5- ذینفعان پروژه** 9

**1-6- سابقه انجام پروژه های مشابه** 9

**1-7- معایب پروژه های گذشته** 10

**1-8- زبان برنامه نویسی** 11

**1-9- پایگاه داده** 11

**1-10- سیستم عامل مورد نیاز برای پیاده سازی** 11

فصل دوم: طراحی پایگاه داده سیستم گل فروشی 12

**نمودار ER کلی پایگاه داده سیستم گل فروشی** 13

فصل سوم: بررسی Source Code C# 14

**در مورد زبان برنامه نویسی** 15

**مروري بر تاريخچه برنامه نويسي** 15

**برنامه نويسي ساخت يافته** 17

**تاریخچه سی شارپ** 19

**اهداف طراحی زبان** 20

**تاریخچه** 20

**ویژگی‌ها** 21

**سیستم یکپارچه شده** 22

**انواع داده** 22

Boxing و EnBoxing 23

**ویژگی‌های جدید در C# 2.0** 23

**کلاسهای partial** 23

**Generic‌ها** 24

**کلاس های static** 24

**یک شکل جدید از تکرار کننده با استفاده از سازنده توابع** 24

**Delegate‌های ناشناس** 25

**Delegate covariance and contravariance** 25

**نوع داده Nullable** 25

**کدهای مربوط به فرم ورود به سیستم** 27

**کدهای مربوط به فرم اصلی سیستم گل فروشی** 29

**کدهای مربوط به فرم ثبت دسته بندی گل** 33

**کدهای مربوط به فرم ثبت واحد شمارش گل** 37

**کدهای مربوط به فرم تعریف عنوان گل** 41

**کدهای مربوط به فرم ثبت موجودی گل** 46

**کدهای مربوط به فرم ثبت مشخصات مشتری** 51

**کدهای مربوط به فرم ثبت فاکتور فروش** 57

**کدهای مربوط به فرم ثبت کاربران سیستم** 66

چاپ فاکتور 73

فصل چهارم: جمع بندی ، نتیجه گیری و پیشنهادات 75

**جمع بندی** 76

**نتیجه گیری** 77

**پیشنهادات** 77

منابع 79

**کتابها** 79

**وب سایتها** 79

**چکیده**

در این نوشتار ، به بررسی و تحلیل سیستماتیک یکی از نرم افزارهای تحت ویندوز با عنوان گل فروشی پرداخته شده است.

هدف از تحلیل این سیستم نرم افزاری ، یافتن راه حلهای مناسب برای پیاده سازی می باشد. یافتن راه حل مناسب نیاز به بررسی دارد. برای بررسی بایستی از یکی از روشهای تحلیل مناسب استفاده نمود.. برای این منظور:

* در فصل اول Vision طرح مورد بررسی قرار گرفته است
* در فصل دوم پایگاه داده طراحی و پیاده سازی گردیده است
* در فصل سوم Source Code های برنامه به همراه واسط کاربر ارائه گردیده اند
* نهایتا در فصل چهارم به جمع بندی و نتیجه گیری پرداخته شده است.

**مقدمه**

نرم افزار مدیریت فروش گل فروشی ابزاری نیرومند است که به شایستگی قادر است پاسخ گوی تمام نیازهای مدیریتی باشد. در واقع این نرم افزار به همراه مشاوره و همکاری کارشناسان و متخصصان مجرب پیاده سازی سیستم گل فروشی ها تهیه گردیده که موجبات صرفه جویی در زمان و بر طرف نمودن کلیه مشکلات شیوه های دستی و قدیمی در گل فروشی ها را فراهم آورده است.

نرم افزار فوق قابل استفاده در کليه گل فروشی ها و کليه مراکز و محلهای مشابه می باشد. اين نرم افزار دارای برنامه حسابداری فاکتور می باشد اين امکان را به مديريت می دهد که با اطمينان کامل و بهتری بتواند مجموعه خود را کنترل و مديريت کند. در ادامه به بررسی ، تحلیل و پیاده سازی پروژه گل فروشی خواهیم پرداخت.

فصل اول: چشم انداز پروژه (**Vision**)

در این فصل ابتدا تعریفی از Vision ارائه می دهیم و سپس به بررسی مواردی می پردازیم که باید در چشم انداز پروژه یا همان Vision باید مورد بررسی قرار گیرند.

**1-1- Vision چیست**

Vision یا چشم انداز پروژه در حقیقت یک سری مستندات اولیه پروژه است که شامل بررسی موارد ذیل می باشد:

1. موضوع پروژه
2. اهداف پروژه
3. انجام پروژه چه سودی دارد
4. ویژگیهای پروژه
5. ذینفعان پروژه چه کسانی هستند
6. سابقه انجام پروژه های مشابه
7. معایب پروژه های گذشته
8. زبان برنامه نویسی
9. پایگاه داده
10. سیستم عامل مورد استفاده
11. روش تحلیل
12. نرم افزارهای مورد نیاز برای تحلیل و پیاده سازی

**1-2- موضوع پروژه**

موضوع پروژه جاری پیاده سازی یک نرم افزار با عنوان سیستم گل فروشی می باشد.

**1-3- اهداف پروژه**

* هدف از پیاده سازی ، سیستمی است که با آن بتوان عملیات فروش گل و ایجاد فاکتور مشتری را با سطوح دسترسی مختلف انجام داد.
* در پیاده سازی این پروژه اهداف تجاری مد نظر نمی باشد. بلکه فقط اهداف آموزشی زیر مد نظر می باشد: (البته ممکن است از این سیستم در مقاصد تجاری نیز استفاده شود)
  1. آشنایی با برنامه های تحت ویندوز
  2. آشنایی با تحلیل این گونه برنامه ها
  3. آشنایی با روش پیاده سازی این گونه نرم افزارها
  4. آموزش برنامه نویسی سیستمهای تجاری
  5. آموزش نحوه استفاده از ابزارهای برنامه نویسی برای پیاده سازی این گونه سیستمها
  6. آموزش نحوه ارتباط بین پایگاه داده و فرمها در این گونه سیستمها
  7. آشنایی با تکنولوژی برنامه نویسی LINQ

**1-4- انجام پروژه چه سودی دارد**

* هزینه دوره های آموزشی خارج از دانشگاه برای دانشجو حذف می شود
* انجام این پروژه از نظر کارآموزی و آموزش نحوه انجام یک پروژه تجاری برای کسانی که به نحوی با پروژه سررکار دارند بسیار مفید است
* Component هایی که در این پروژه تولید می شوند اکثرا در پروژه های آینده و یا پروژه های مشابه نیز قابل استفاده اند و این مساله خود باعث صرفه جویی در زمان اجرای پروژه های دیگر خواهد شد

**1-5- ذینفعان پروژه**

* **تحلیل گر**
  + کسی است که عملیات تحلیل سیستم را انجام می دهد و با سیستم به خوبی آشنایی دارد.
* **برنامه نویس**
  + کسی است که عملیات پیاده سازی و برنامه نویسی را با استفاده از مستندات تولید شده توسط تحلیلگر ، انجام می دهد.
* **افراد تحت تعلیم دوره**
  + افراد تحت تعلیم دوره معمولا دانشجویانی هستند که در این دوره آموزشی شرکت نموده اند.
* **مشتریان** 
  + مشتریان کسانی هستند که در آینده این نرم افزار را خریداری می نمایند و به نحوی از این برنامه استفاده می نمایند.

**1-6- سابقه انجام پروژه های مشابه**

نسخه های بسیار متوعی از برنامه گل فروشی در سرتاسر دنیا به زبانهای مختلف (چه از لحاظ برنامه نویسی و چه از لحاظ زبان ملی) پیاده سازی شده اند که زمینه فعالیت آنها به شرح ذیل نی باشد:

1. **فعالیتهای علمی**
   1. دانشگاهی
   2. پژوهشی
   3. آموزشی
2. **فعالیتهای تجاری**
   1. خرید
   2. فروش
   3. بازاریابی
3. **فعالیتهای عمومی**
   1. درج خلاصه فروش
   2. معرفی گل

**1-7- معایب پروژه های گذشته**

در بررسی به عمل آمده از ایرادهای زیر مشاهده شد که بیشتر این ایرادات در نرم افزارها معمولا مشترک می باشند:

* عدم وجود امکان جستجوی عادی
* وجود اشکال در جستجو با حروف فارسی در هنگام تایپ (مثلا با حرف ی و گ بیشتر مشکل دارند و نتایج مختلفی ارائه می شود و گاهی اوقات نتیجه مطلوبی از جستجو حاصل نمی شود)
* فرم ثبت نام مشتری بزرگ
* عدم وجود جستجوی پیشرفته برای اعضا ( در برخی موارد به یک جستجوی ساده اکتفا شده بود که آن هم نتایج مطلوبی ارائه نمی کرد)
* استفاده از کنترلهای غیر استاندارد
* گرافیک بسیار خسته گننده
* عدم پاسخدهی به موقع در بخش پشتیبانی
* عدم پشتیبانی از سیستم عاملهای مختلف مختلف
* عدم وجود دسته بندی گل
* . . . .

البته ایرادات فنی بیشماری در نرم افزارها موجود می باشد که به علت طولانی شدن بحث از ارائه آنها خودداری می شود.

**1-8- زبان برنامه نویسی**

زبان برنامه نویسی Base پروژه C# 2010 می باشد که C# در مجموعه Visual Studio 2010 موجود و قابل نصب می باشند. تکنولوژي مورد استفاده نیز LINQ می باشد.

**1-9- پایگاه داده**

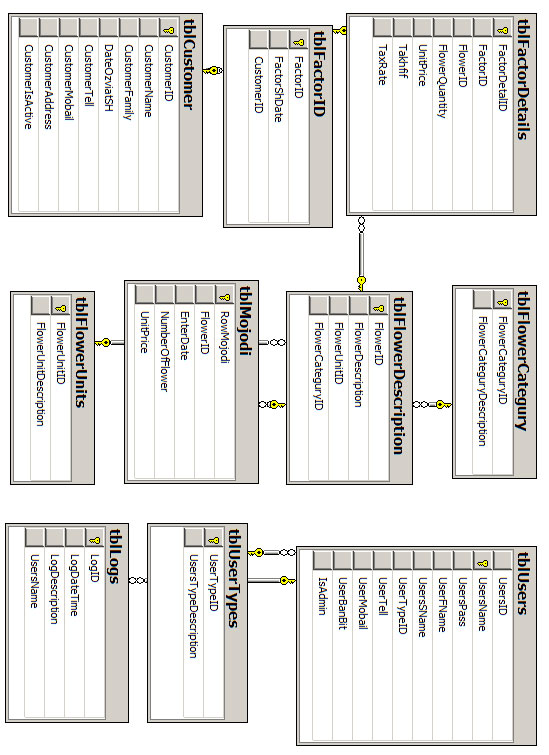
پایگاه داده مورد استفاده در سیستم SQL Server می باشد که مناسب برای نصب بر روی ویندوز است.

**1-10- سیستم عامل مورد نیاز برای پیاده سازی**

* سیتم عامل استاندارد برای کارهای حرفه ای Windows Server می باشد. ولی در این مورد ما از سیستم عامل ویندوز 10 استفاده خواهیم نمود.

فصل دوم: طراحی پایگاه داده سیستم گل فروشی

**نمودار ER کلی پایگاه داده سیستم گل فروشی**



نمایه - نمودارER

فصل سوم: بررسی Source Code C#

**در مورد زبان برنامه نویسی**

در این فصل ابتدا توضیحاتی در مورد زبان برنامه نویسی ارئه خواهد شد و سپس به بررسی کدهای نوشته شهد در کنار شکل فرمها پرداخته خواهد شد.

**مروري بر تاريخچه برنامه نويسي**

برنامه نويسي رايانه به گونه اي كه ما امروز مي شناسيم , در اواخر دهه ي 50 ميلادي شروع شد و در دهه ي 60 ميلادي بسرعت رشد نمود . در آن دوران برنامه نويسي يك كار تخصصي و مهندسي تلقي نميشد زيرا رايانه ها بسيار محدود بودند و كارهاي ساده اي انجام مي دادند . اغلب اينگونه بود كه افراد بسيار سريعتر از رايانه ها پاسخ محاسبات رياضي را مي يافتند . لذا برنامه نويسي براي چنين دستگاه هايي چندان ارزش و اهميت نداشت و اولين برنامه ها فقط چند دستور ساده ي متوالي بودند . پس از اجراي اين دستورها رايانه بايد از نو راه اندازي ميشد تا براي اجراي دوباره ي برنامه آماده شود .

كم كم سخت افزار متحول شد و رايانه ها قدرتمندتر و كارامدتر شدند اما به همان نسبت توليد برنامه براي آنها دشوار گشت طوري كه برنامه هاي ساده ي اوليه , مشكل جدي بر سر راه كاربري رايانه محسوب شدند . پس تلاش هايي صورت گرفت تا برنامه هاي كارامدتري نوشته شود . اما طبق يك قانون كلي , پيچيدگي نرم افزار بيشتر و بيشتر مي گردد تا جايي كه ديگر انسان نمي تواند آن را مديريت كند و از همان جا هرج و مرج شروع مي شود .

بزودي نهضت همگاني صورت گرفت تا راهكاري براي مديريت پيچيدگي و كارايي برنامه ها يافت شود . سرانجام در اواخر دهه ي 60 " برنامه نويسي ساخت يافته " ( Procedural programming ) بعنوان راه حل پذيرفته شد. برنامه نويسي ساخت يافته امكان ميداد كه برنامه به چند تابع تقسيم شود و چند نفر بطور همزمان كار توليد نرم افزار را پيش ببرند و همين امر سبب افزايش سرعت توليد نرم افزار و كاهش خطاهاي برنامه نويسي گرديد و برنامه هاي بزرگ و مفيد ساخته شدند . برنامه هايي كه در ابتدا با اين روش ايجاد شدند چيزي بين 10 تا 20 هزار خط كد داشتند , از ساختمان داده ي كوچكي بهره مي جستند و الگوريتم هاي ساده اي در آنها به كار مي رفت . با استفاده از برنامه نويسي ساخت يافته براحتي ميشد چنين برنامه اي را مديريت كرد .

برنامه ها آرام آرام به قصد بهبود كارايي , حجيم تر شدند . الگوريتم هاي پيشرفته اي بكار گرفته شد و ساختمان داده ي پيچيده تري در تركيب برنامه ها استفاده شد . برنامه نويسي ساخت يافته ديگر نمي توانست از پس كدهاي طولاني چنين برنامه هايي برآيد ( سيستم عامل ويندوز NT بيش از 10 ميليون خط كد دارد ) .

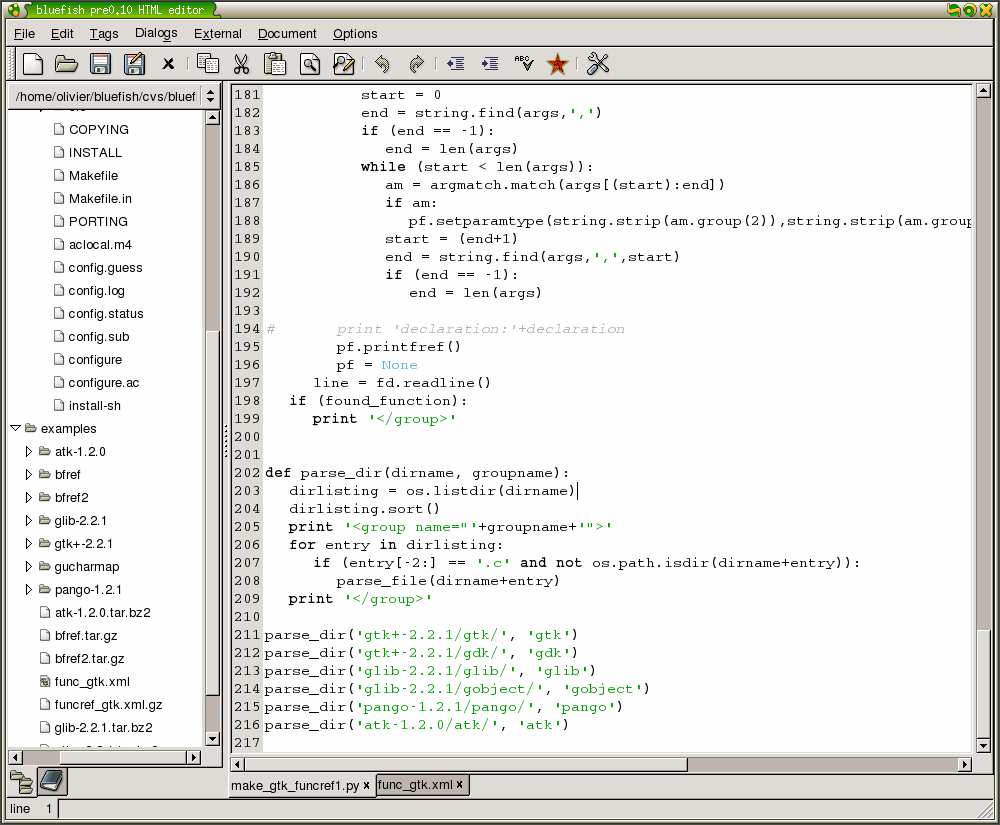


نمایه -اعداد باینری

بنظر ميرسيد كه روش ديگري لازم است تا توليد كنندگان نرم افزار را از دست متغيرهاي فراوان و توابع طولاني و سرگيجه آور خلاص كند . بر اساس اين تفكر بود كه برنامه نويسي شئ گرا ( Object oriented programming )پيش كشيده شد . تفكر شئ گرا تقريبا همزمان با تفكر ساخت يافته مطرح شد اما فقط براي طراحي برنامه استفاده ميشد . پس از عاجز ماندن برنامه نويسي ساخت يافته , شئ گرايي در برنامه نويسي هم استفاده شد . حالا برنامه نويسان مي توانند در برنامه به پانزده شيء فكر كنند كه اين اشياء با چهل پيام مشخص با هم ارتباط دارند , حال آنكه همين برنامه اگر به سبك ساخت يافته ايجاد شود بايد به 100 متغير , 20 ثابت , ششصد تابع و ... فكر كرد .

**برنامه نويسي ساخت يافته**

برنامه نويسي ساخت يافته روش منظمي براي توليد برنامه هاست . برنامه هايي كه به اين شيوه نوشته ميشوند خوانايي زيادي دارند ( يعني راحت درك ميشوند ) , آزمايش و اشكال زدايي و اصلاح آنها نيز آسان است . در اين روش برنامه بصورت مجموعه اي از فعاليت ها تصور ميشود كه بايد بر روي داده ها انجام شوند . هر كار پيچيده اي به چند كار كوچكتر تجزيه ميشود تا اين كارها براحتي قابل پياده سازي باشند . سپس براي انجام هريك از اين كارها , زير برنامه ي خاصي كه تابع ( Function ) نام گرفته نوشته ميشود . به همين دليل به اين شيوه " برنامه نويسي بالا به پايين " ( Up-Down programming ) نيز مي گويند . برنامه نويسي ساخت يافته روش موفقي در حل مسائل پيچيده است . توابع قابليت استفاده ي مجدد دارند . يعني ميتوان توابع يك برنامه را در برنامه هاي ديگر نيز بكار برد . به اين ترتيب كد نويسي كمتر ميشود و توليد نرم افزار با سرعت بيشتري صورت مي گيرد .



نمایه - محیط ویرایشگر کد

توابع امكان مي دهند تا توليد نرم افزار با استفاده از گروه ممكن شود . به اين شكل كه پس از مشخص شدن توابع مورد نياز , هريك از اعضاي گروه وظيفه ي نوشتن و آزمايش توابع مشخصي را برعهده مي گيرد و به اين شكل اعضاي گروه بصورت همزمان روي بخش هاي مختلف برنامه كار مي كنند بدون اينكه يكي منتظر ديگري بماند . در نهايت اين توابع به هم پيوند مي خورند تا جريان منطقي برنامه برقرار شود .

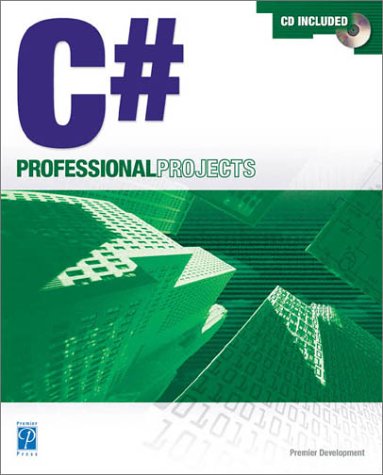
بديهي است كه عيب يابي و اصلاح برنامه نيز آسان ميشود , زيرا با برسي جريان برنامه , تابع معيوب بسرعت شناسايي و اصلاح ميشود بدون اينكه بقيه ي بخش هاي برنامه تغيير يابند . زبان هاي C و PASCAL دو نمونه از زبان هاي برنامه نويسي ساخت يافته هستند .

براي زبان C# همانند بيشتر زبانهاي برنامه‌سازي دو نوع كامپايلر وجود دارد. يك نوع كامپايلر كه به كامپايلر Command Line معروف است و نوع ديگر كامپايلر Visual است. كامپايلر‌هاي Command Line محيطي شبيه به محيط DOS دارند و با دادن يك سري دستورات به اجرا در مي‌آيند. كامپايلرهاي Visual محيطي همانند ويندوز دارند كه با دارا بودن محيط گرافيكي و ابزارهاي خاص، برنامه‌نويس را در امر برنامه‌سازي كمك مي‌كنند. از نمونه‌هاي هر يك از كامپايلرها، مي‌توان به Microsoft C# Command Line Compiler كه يك كامپايلر Command Line و Microsoft Visual C# كه يك كامپايلر Visual است، اشاره كرد. البته در حال حاضر بيشتر از كامپايلرهاي ويژوال استفاده مي‌شود.

اعلان Namespace به سيستم اعلان مي‌نمايد كه شما از توابع كتابخانه‌اي System جهت اجراي برنامه‌ها خود استفاده مي‌نماييد. دستوراتي مانند WriteLine و ReadLine جزو توابع كتابخانه‌اي System مي‌باشند. اغلب دستورات و توابع مهم و كليدي استفاده از كنسول ورودي/خروجي در اين كتابخانه مي‌باشد. در صورتيكه در ابتداي برنامه از using System استفاده نكنيد، يابد در ابتداي هر يك از دستورات برنامه كه مربوط اين كتابخانه است، از كلمة System استفاده نماييد.

**تاریخچه سی شارپ**

سی‌شارپ (C#) زبانی شیءگرا و سطح بالا (high level) از خانوادهٔ زبان‌های چارچوب دات‌نت شرکت مایکروسافت است. زبان #C، یک زبان برنامه نویسی چند الگویی است و منظم شده مدل های تابعی، امری، عمومی، شی گرا و جز گرا می باشد. این زبان توسط ماکروسافت و جزئی از دات نت به وجود آمد و بعدا استانداردهای ECMAو ISO را نیز در بر گرفت. #C یکی از 44 زبان برنامه نویسی ای است که توسط Common Language Runtime از .NET Framework پشتیبانی می شوند و در همه جا به وسیله Microsoft Visual Studio شناخته می شود.



نمایه - عکس جلد کتاب آموزش سی شارپ

این زبان برپایه سادگی، مدرن بودن، همه منظوره و شی گرا بودن ساخته شد.Anders Hejlsberg، طراح زبان برنامه نویسی دلفی، سرپرستی تیم طراحان زبان #C را بر عهده داشت.این زبان دارای دستوری شی گرا مشابه ++C است و به شدت از زبان های جاوا و دلفی تاثیر پذیرفته است.در ابتدا نام این زبان COOL بود که مخفف C like Object Oriented Language بود، هر چند در جولای 2000، زمانی که ماکروسافت پروژه را عمومی اعلام کرد، اسم آن به #C تغییر پیدا کرد.آخرین نسخه آن نسخه 3 است که از ویژوال استدیو 2008 استفاده می کند. نسخه بعدی 4 است که در دست ساخت است.

**اهداف طراحی زبان**

استاندارد ECMA این اهداف طراحی زبان را برای #C بر آورده می سازد:

سی شارپ یک زبان برنامه سازی ساده،مدرن،برای اهداف عمومی و شی گرا است.

به دلیل اهمیت داشتن موضوع نیرومندی و دوام و بهره وری برنامه نویس،زبان دارای چک کننده Strong Type ، چک کننده مرزهای آرایه، تشخیص حالتهایی که یک متغیر مقدار دهی اولیه نشده‌است،قابلیت انتقال کدها و Garbage Collection خودکار است.

این زبان برای استفاده در اجزای توسعه نرم افزار برای دستیابی به مزایای سیستم‌های توزیعی در نظر گرفته شده‌است.

قابلیت انتقال برنامه نویس بسیار مهم است،خصوصاً برای آن دسته از برنامه نویسانی که با زبان‌های C و C++ آشنا هستند.

پشتیبانی از این زبان برای بین المللی شدن بسیار مهم است.

زبان سی شارپ برای نوشتن برنامه‌ها برای سیستمهای تعبیه شده و میزبان در نظر گرفته شده‌است، از سیستم عامل های پیچیده بسیار بزرگ گرفه تا توابع اختصاصی بسیار کوچک.

هر چند برنامه های نوشته شده با #C طوری هستند که از لحاظ حافظه و پردازنده مورد نیاز مقرون به صرفه باشند، ولی خود زبان از لحاظ اندازه و کارایی به خوبی زبان های C و اسمبلی نیست.

**تاریخچه**

در سال 1999، شرکت سان اجازه استفاده از زبان برنامه نویسی جاوا را در اختیار ماکروسافت قرار داد تا در سیستم عامل خود از آن استفاده کند.جاوا در اصل به هیچ پلت فرم یا سیستم عاملی وابسته نبود، ولی ماکروسافت برخی از مفاد قرار داد را زیر پا گذاشت و قابلیت مستقل از سیستم عامل بودن جاوا را از آن برداشت.شرکت سان پرونده ای علیه ماکروسافت درست کرد و ماکروسافت مجبور شد تا زبان شی گرای جدیدی با کامپایل جدید که به ++C شبیه بود را درست کند. در طول ساخت دات نت، کلاس های کتابخانه ای با زبان و کامپایلر SMC نوشته شدند.در سال 1999 آندرس هلزبرگ گروهی را برای طراحی زبانی جدید تشکیل داد که در آن زمان نامش Cool بود و همانند C بود با خواص شی گرایی. ماکروسافت در نظر داشت اسم این زبان را تا آخر Cool قرار دهد، ولی به دلیل مناسب نبودن برای اهداف تجاری این کار را نکرد. در ارائه و معرفی رسمی .NET در PDC در سال 2000 این زبان به Cسی شارپ تغییر نام یافت و کتابخانه کلاس ها و runtime در ASP.NET به#C منتقل شدند. مدیر و سرپرست طراحان در ماکروسافت آندرس هلزبرگ بود که تجربه قبلی او در طراحی Framework و زبان‌های برنامه سازی++Borland , Delphi, Turbo Pascal, Visual C به آسانی در دستورالعمل‌های سی شارپ قابل رویت است و به همان خوبی در هسته CLR.

**ویژگی‌ها**

برخی از تفاوت‌های زبان سی شارپ با C و ++C عبارتند از:

هیچ تابع یا متغیر سراسری(Global) وجود ندارد، تمام متدها و اعضا بایستی در داخل کلاس‌ها تعریف شوند.این امر ممکن است، هر چند برای استفاده از متغیر ها و توابع عمومی باید از متدها و متغیرها در کلاس های عمومی استفاده کرد.

متغیر های عمومی، بر خلاف C و ++C ، نمی توانند بلاک های پیوستی را در بر بگیرند.

سی شارپ دارای یک نوع داده بولی است (bool).برخی از عبارت‌ها مانند while و دستور شرطی // if که شرطی هستند، نیازمند یک عبارت نوع بولی هستند.همان طور که ++C نیز دارای نوع داده بولی است،این نوع داده به راحتی میتواند به یا از Integerها تبدیل شود، و عبارتی مانند (دستور شرطی // if(a نیازمند این امر است که a از یک نوع قابل تبدیل به bool یا اشاره گر باشد.کامپایلر سی شارپ برنامه نویس را در این شرایط مجبور به استفاده از عباراتی می‌کند که به درستی یک مقدار bool را برمیگردانند. بنابراین دستوری مانند (دستور شرطی // if(a = b باعث بروز خطا می‌شوند.(به جای = بایستی از == استفاده شود)

در سی شارپ ، اشاره گرهای به حافظه بایستی فقط در داخل بلوکهای unsafe استفاده شوند و برنامه در این حالت برای اجرا نیاز به اجازه از کاربر دارد.بیشتر دسترسی شی از طریق شی امن است که یا همیشه در حال اشاره به شی صحیح موجود است یا یک مقدار Null دارد. اشاره گری به شی به درد نخور یا بلاک حافظه رندم غیر ممکن است.اشاره گر نا امن می تواند به نمونه ای از value-type ، آرایه، رشته یا بلاکی که حافظه به آن داده شده است اشاره نماید.کدی که به عنوان نا امن علامت نخورده باشد، هنوز می تواند اشاره گر ها را از سیستم بازیابی یا در آن ذخیره کند ولی نمی تواند مرجع جدیدی به آنها اختصاص دهد.

حافظه ساماندهی شده نمی تواند صریحا آزاد شود، ولی به طور خودکار به عنوان به درد نخور تلقی می شود.انتخاب آدرس های به درد نخور حافظه نفوذ ناپذیر است. هم چنین #C با استفاده از عبارات، پشتیبانی مستقیمی از پایان اجباری می کند(پشتیبانی از اصطلاح Resource Acquisition Is Initialization).

وراثت چندگانه از کلاس‌ها در این زبان پشتیبانی نمی‌شود.البته یک کلاس امکان ارث بری از تعداد نامحدود واسط‌ها را دارد.پشتیبانی نکردن از وراثت چندگانه به دلیل اهداف معماری این زبان در CLI و برای جلوگیری از پیچیدگی است.

سی شارپ بسیار typesafe تر از C++ است. تنها تبدیلات ضمنی مثل تبدیل نوع داده کوچکتر به بزرگتر یا تبدیل نوع مشتق شده به نوع پایه به طور پیش فرض و بدون خطا صورت می‌پذیرد.هیچ تبدیل ضمنی ای میانBooleanها و Integerها وجود ندارد و هر تبدیل user-defined بایستی به صراحت با یکی از کلمات explicit یا implicit نشانه گذاری شود. تبدیل b به a در حالتی که a یک Integer و b یک double باشد در زبان C++ مجاز است اما در سی شارپ به یک خطای زمان کامپایل منجر می‌شود(بایستی به صورت explicit تعریف شود) اعضای Enumeration در داخل محدوده شخصی خود قرار دارند. #C قابلیت syntactic sugar را برای توابع متداول، اکسسور ها و ماجول های کسول شده در یک کلاس به صورت ویژگی ها قرار داده است. اکسسورها که خاصیت نیز گفته می‌شوند در زبان سی شارپ قادر به کنترل دسترسی اعضا و معتبرسازی داده‌ها هستند. تمام انواع بازتابی(Reflection) و بازیابی(Recovery) قابل استفاده‌است. در حال حاظر (3 جوان 2008) دارای 77 کلمه رزرو شده است.

**سیستم یکپارچه شده**

سی شارپ دارای یک سیستم نوع یکپارچه‌است که به آن CTS می گویند.این بدان معناست که تمام انواع،شامل موارد اصلی مانند Integerها، مشتق شده از System.Object هستند.به عنوان مثال، هر نوع یک متد به نام ToString() را به ارث می‌برد.بخاطر کارائی،انواع اولیه (و انواع مقداری) به طور داخلی فضایی برای آنها بر روی پشته در نظر گرفته می‌شود.

**انواع داده**

CTS داده ها را به دو نوع تقسیم می کند:

* نوع مقداری
* نوع مرجعی

انواع داده ای توده ساده ای از داده می باشند.نمونه های انواع داده ای نه حویت مرجعی دارند و نه مفاهیم مقایسه مراجع را. برای مقایسه برابری یا عدم برابری انواع داده ای، خود مقدار داده ها را با یکدیگر مقایسه می کنیم مگر اینکه عملگر های مشابه دوباره تعریف شده باشند.مقادیر داده های مرجعی همیشه یک مقدار پیش فرض دارند و همیشه می توانند ایجاد و یا کپی شوند.یکی دیگر از محدودیت های انواع داده ای این ات که آنها نمی توانند از یکدیگر مشتق شوند(ولی میتوانند اشتراکاتی داشته باشند) و هم چنین نمی توانند در سازنده مقدار دهی اولیه شوند. مثالی از انواع داده ای، بعضی از انواع اولیه مانند int و float و char و System.DateTime می باشند. در مقابل، انواع مرجعی مفهوم تعریف مرجعی را دارند( که در آن هر نمونه از نوع مرجع، به طور ذاتی از دیگر نمونه ها جدا می شود، حتی اگر داده هر دو نمونه یکی باشد). این دقیقا نمونه مشابه مقایسه تساوی یا عدم تساوی داده های مرجعی است، که در آن آزمایش برای مرجع ها از داده ای ها سریع تر است. در کل نه همیشه امکان تعریف نمونه مرجعی وجود دارد و نه امکان کپی یا نمایش مقادیر مقایسه دو نمونه.ولی به هر حال انواع مرجعی خاص می توانند این اعمال را از طریق سازنده های عمومی یا اجرای واسط های مشابه(مثل ICloneable یا IComparable) انجام دهند.نمونه هایی از انواع مرجعی، اشیا، System.Stringو Sysmet.Array می باشند. هر دو نوع داده قابلیت انعطاف توسط تعریف به وسیله کاربر را دارند.

## Boxing و EnBoxing

Boxing عمل تبدیل مقدار نوع داده ای به نوع مرجع مشابه آن می باشد.

مثال:

int foo = 42; *// Value type...*

object bar = foo; *// foo is boxed to bar.*

UnBoxing عمل تبدیل نوع مرجع به نوع داده ای می باشد. مثال:

int foo = 42; *// Value type.*

object bar = foo; *// foo is boxed to bar.*

int foo2 = (int)bar; *// Unboxed back to value type.*

سی شارپ به [برنامه نویس](http://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87_%D9%86%D9%88%DB%8C%D8%B3) با استفاده از کلمه کلیدی Struct اجازه می‌دهد تا انواع مقداری User-defined را ایجاد کند. از دیدگاه برنامه نویسی، آنها کلاس‌های سبک وزن به نظر می‌رسند. برخلاف کلاس‌ها (که بر روی heap قرار میگیرند) و شبیه به انواع اولیه استاندارد مانند انواع مقداری Structها نیز بر روی پشته قرار می‌گیرند.آنها همچنین می‌توانند قسمتی از یک شئ باشند، یا در یک آرایه مرتب شوند، بدون حافظه غیر مستقیمی که به طور معمول برای انواع کلاس تخصیص می‌یابد.

**ویژگی‌های جدید در C# 2.0**

ویژگی‌های جدید در C# .NET SDK 2.0 (مطابق با سومین ویرایش استاندارد ECMA-334):

**کلاسهای partial**

کلاس‌های Partial اجازه اجرای کلاس‌ها از بیش از یک سورس فایل را می‌دهند.این امر اجازه می‌دهد تا کلاس‌های بسیار بزرگ را قطعه قطعه کنیم و همچنین برای زمانی که برخی قسمت‌های یک کلاس به طور خودکار تولید می‌شوند مفید است.

file1.cs:

public partial class MyClass

{

public MyClass()

{

*// implementation*

}

}

file2.cs:

public partial class MyClass

{

public void SomeMethod()

{

*// implementation*

}

}

**Generic‌ها**

genericها یا نوع های پارامتری شده یا چندریختی های پارامتری یک ویژگی جدید .NET 2.0 است که به وسیله سی شارپ پشتیبانی می‌شود.برخلاف Template‌های سی پلاس پلاس،در این انواع به جای اینکه نمونه سازی توسط کامپایلر انجام شود، در زمان اجرا صورت می‌گیرد، بنابراین می توانند چند زبلنه باشند در حالی که ++C نمی تواند.آنها دارای ویژگی‌هایی هستند که به طور مستقیم توسطTemplate‌های C++ پشتیبانی نمی‌شوند مانند نوع محدودیت‌ها در پارامترهای Generic با استفاده از رابط ها(Interface). سی شارپ از پارامترهای‌های Generic بدون نوع پشتیبانی نمی‌کند.بر خلاف genericهای جاوا،genericهای دات نت برای پارامتری کردن انواع داده ای در اشیا ماشین مجازی CLI، از مفاهیم شی گرایی استفاده می کنند که اجازه بهینه سازی و حفاظت انواع اطلاعات را می دهد.

**کلاس های static**

کلاس‌ها به صورت Static قابل تعریف نیستند مگر اینکه تمام اعضای آنها Static باشند.که این امر بسیار شبیه به مفهوم مدل در [زبانهای رویه‌ای](http://fa.wikipedia.org/w/index.php?title=%D8%B2%D8%A8%D8%A7%D9%86%D9%87%D8%A7%DB%8C_%D8%B1%D9%88%DB%8C%D9%87%E2%80%8C%D8%A7%DB%8C&action=edit&redlink=1) است.(زبان رویه‌ای : یک زبان برنامه نویسی که در آن عنصر اصلی برنامه نویسی یک زیربرنامه‌است.مانند زبانهای C، پاسکال و...)

**یک شکل جدید از تکرار کننده با استفاده از سازنده توابع**

یک شکل جدید از iterator(تکرار کننده) ، با استفاده از ساختار yield return بسیار شبیه به yield زبان Python.

*// Method that takes an iterable input (possibly an array) and returns all even numbers.*

public static IEnumerable<int> GetEven(IEnumerable<int> numbers)

{

foreach (int i in numbers)

{

دستور شرطی // if (i % 2 == 0) yield return i;

}

}

**Delegate‌های ناشناس**

Delegate‌های ناشناس که عملکردهای محدودی را در #C به وجود می‌آورند.کد کنار بدنه Deletage ناشناس، دسترسی کامل برای خواندن یا نوشتن در متغیرهای عمومی، پارامترهای توابع و اعضای کلاسهای دارای محدوده Deletage را دارد ولی پارامترهای out و ref را پشتیبانی نمی کند. برای مثال :

int SumOfArrayElements(int[] array)

{

int sum = 0;

Array.ForEach(

array,

delegate(int x)

{

sum += x;

}

);

return sum;

}

**Delegate covariance and contravariance**

تبدیل گروه های متد به نوع Deletage در برگشت دارای covariant و در انواع پارامترها دارای contravariant هستند.

اکسسورهای یک خاصیت(get و set) می‌توانند دارای سطح دسترسی متفاوتی باشند.

مثال :

string status = string.Empty; // حذف مقادیر متنی

public string Status

{

get { return status; } *// anyone can get value of this property,*

protected set { status = value; } *// but only derived classes can change it*

}

نکته مهم : سطح دسترسی خاصیت نمی‌تواند بالاتر از سطح دسترسی اکسسورها باشد.به عنوان مثال private بودن خاصیت و public بودن اکسسور باعث بروز خطا می‌شود.

**نوع داده Nullable**

نوع داده Nullable (که با یک علامت سوال قابل تشخیص است : int? i = null;)اجازه تخصیص مقدار null را برای انواع داده‌ای می‌دهد.این امر باعث بهبودی فعل و انفعال با پایگاه داده SQL می‌شود. در این حالت نوع ستونی INTEGER NULL در SQL به طور مستقیم به int? در سی شارپ تبدیل می‌شود.

داده های Nullable در آخرین لحظات آگوست 2005 اضافه شدند چند هفته متنده به اتمام کار اداری و برای بهبود زبان.متغیر Null در حقیقت خالی نیست، بلکه نمونه ای است از struct Nullable<T> با ویژگی HasValue مساوی false. وقتی در برنامه قرار می گیرد، خود به خود نمونه خالی در آن قرار می گیرد، نه مقدار خود آن، در نتیجه اشاره گر مقصد همیشه غیر Null می باشد، حتی برای مقادیر Null. کد زیر نضص بالا را مشخص می کند:

int? i = null;

object o = i;

دستور شرطی // if (o == null)

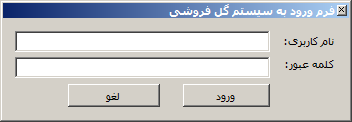
Console.WriteLine(«Correct behaviour - runtime version from September 2005 or later»);

else

Console.WriteLine(«Incorrect behaviour - pre-release runtime (from before September 2005)»);

وقتی درون شی ای کپی می شود، نمونه Nullable به صورت تشریفاتی در آن قرار می گیرد و در نتیجه مقادیر و منابع Null با هم برابر می شوند.در گذشته این خاصیت دارای مجادله بود تا زمانی که علاوه بر .NET2 ، به هسته CLR نیز مجهز شد و همه تکنولوژی ها نظیر #CوVBوSQL Server 2008و Visual Studio 2010 را شامل شد.

**کدهای مربوط به فرم ورود به سیستم**



نمایه - عکس فرم ورود به سیستم گل فروشی

namespace FlowerShop

{

public partial class frmLogin : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmLogin()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void frmLogin\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

}

private void frmLogin\_FormClosed(object sender, FormClosedEventArgs e) // این متد در زمان بسته شدن فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " بسته شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

Application.Exit(); // بسته شدن تمام برنامه

}

private void Btn\_Login\_Click(object sender, EventArgs e) // رخداد کلیک

{

Class\_tblUsers LogInUsr = new Class\_tblUsers(Txt\_UserName.Text.Trim(), Txt\_Password.Text.Trim());

if (LogInUsr.UDF\_LoginUser() == true) // بررسی اینکه آیا اطلاعات کاربری صحیح بوده یا خیر

{

Txt\_UserName.Text = String.Empty; // حذف مقادیر متنی

Txt\_Password.Text = String.Empty; // حذف مقادیر متنی

this.Hide();

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " ورود به سیستم انجام شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

frmMainMdi frm = new frmMainMdi();

frm.ShowDialog();

}

else

{

مایش پیغام // MessageBox.Show("نام کاربری و یا کلمه عبور صحیح نمی باشند.");

}

}

private void Btn\_Cancel\_Click(object sender, EventArgs e) // رخداد کلیک

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " درخواست ورود لغو شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

Application.Exit(); // بسته شدن تمام برنامه

}

private void Txt\_UserName\_KeyDown(object sender, KeyEventArgs e)

{

if (Txt\_UserName.Text != "" && Txt\_Password.Text != "")// بررسی اینکه فیلدهای نام کاربری و کلمه عبور خالی نباشند

{

if (int.Parse(e.KeyValue.ToString()) == 13) // بررسی فشرده شدن کلید اینتر

{

Btn\_Login.PerformClick();// اجرای متد لاگین

}

}

}

private void Txt\_Password\_KeyDown(object sender, KeyEventArgs e)

{

if (Txt\_UserName.Text != "" && Txt\_Password.Text != "")// بررسی اینکه فیلدهای نام کاربری و کلمه عبور خالی نباشند

{

if (int.Parse(e.KeyValue.ToString()) == 13)// بررسی فشرده شدن کلید اینتر

{

Btn\_Login.PerformClick();// اجرای متد لاگین

}

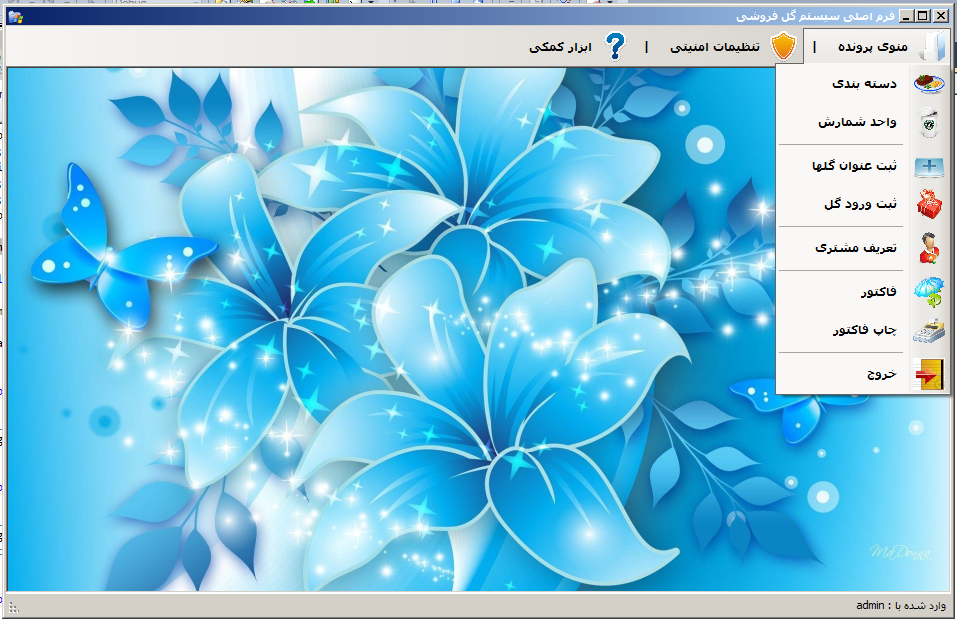
}

}

}

}

**کدهای مربوط به فرم اصلی سیستم گل فروشی**



نمایه - فرم اصلی سیستم گل فروشی

namespace FlowerShop

{

public partial class frmMainMdi : Form // علامت ارث بری از کلاس اصلی فرم

{

frmFoodCategury frm2; // ایجاد شئ از کلاس فرمهای برنامه

frmFoodUnits frm3; // ایجاد شئ از کلاس فرمهای برنامه

frmDefineFoods frm4; // ایجاد شئ از کلاس فرمهای برنامه

frmDefineCustomer frm5; // ایجاد شئ از کلاس فرمهای برنامه

frmEditUsers frm6; // ایجاد شئ از کلاس فرمهای برنامه

frmFactor frm7; // ایجاد شئ از کلاس فرمهای برنامه

frmMojodi frm8; // ایجاد شئ از کلاس فرمهای برنامه

frmCalculator frm9; // ایجاد شئ از کلاس فرمهای برنامه

public frmMainMdi()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void frmMainMdi\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

toolStripStatusLabel.Text = "وارد شده با : " + Class\_GlobalVar.Str\_LogedUserName.Trim();

}

private void frmMainMdi\_FormClosed(object sender, FormClosedEventArgs e) // این متد در زمان بسته شدن فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " بسته شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

Application.Exit(); // بسته شدن تمام برنامه

}

private void Mnu\_File\_Exit\_Click(object sender, EventArgs e) // رخداد کلیک

{

this.Close(); // بسته شدن شی

}

private void Mnu\_Reports\_Logs\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm1 == null || frm1.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm1 = new frmRptLogs();

frm1.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm1.Show(); // نمایش شئ

}

else

{

frm1.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_File\_FoodCategu\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm2 == null || frm2.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm2 = new frmFoodCategury();

frm2.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm2.Show(); // نمایش شئ

}

else

{

frm2.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_File\_FoodUnit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm3 == null || frm3.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm3 = new frmFoodUnits();

frm3.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm3.Show(); // نمایش شئ

}

else

{

frm3.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_File\_FoodDef\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm4 == null || frm4.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm4 = new frmDefineFoods();

frm4.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm4.Show(); // نمایش شئ

}

else

{

frm4.Activate(); // فعال سازی آبجکت

}

}

private void customerToolStripMenuItem\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm5 == null || frm5.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm5 = new frmDefineCustomer();

frm5.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm5.Show(); // نمایش شئ

}

else

{

frm5.Activate(); // فعال سازی آبجکت

}

}

private void mnuSecurity\_DefineUsers\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm6 == null || frm6.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm6 = new frmEditUsers();

frm6.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm6.Show(); // نمایش شئ

}

else

{

frm6.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_File\_Factor\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm7 == null || frm7.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm7 = new frmFactor();

frm7.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm7.Show(); // نمایش شئ

}

else

{

frm7.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_File\_InsertFood\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm8 == null || frm8.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm8 = new frmMojodi();

frm8.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm8.Show(); // نمایش شئ

}

else

{

frm8.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_Help\_About\_Click(object sender, EventArgs e) // رخداد کلیک

{

مایش پیغام // MessageBox.Show("برنامه گل فروشی - نسخه 1");

}

private void Mnu\_Help\_Calculator\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm9 == null || frm9.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm9 = new frmCalculator();

frm9.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm9.Show(); // نمایش شئ

}

else

{

frm9.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_Reports\_PrintFactor\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm10 == null || frm10.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm10 = new frmRptFactorPrint();

frm10.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm10.Show(); // نمایش شئ

}

else

{

frm10.Activate(); // فعال سازی آبجکت

}

}

private void Mnu\_Reports\_DailySale\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (frm11 == null || frm11.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm11 = new frmRptTotalSaleRate();

frm11.MdiParent = this; // تخصیص فرم والد به فرم فرزند

frm11.Show(); // نمایش شئ

}

else

{

frm11.Activate(); // فعال سازی آبجکت

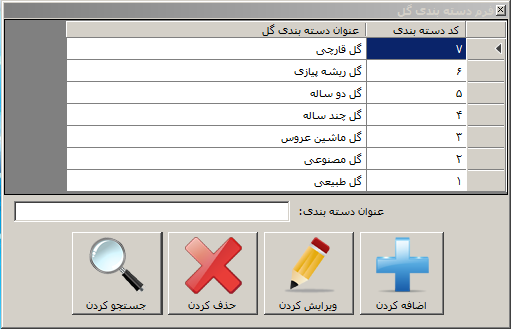
}

}

}

}

**کدهای مربوط به فرم ثبت دسته بندی گل**



نمایه - فرم دسته بندی گل

namespace FlowerShop

{

public partial class frmFoodCategury : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmFoodCategury()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void btnAdd\_Click(object sender, EventArgs e) // رخداد کلیک

{

Txt\_AssetCategury.Text = String.Empty; // حذف مقادیر متنی

btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

مایش پیغام // MessageBox.Show("لطفا اطلاعات را وارد نمایید");

}

private void frmFoodCategury\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

private void UDF\_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodCateguries

orderby Row.FoodCateguryID descending

select new

{

Row.FoodCateguryID,

Row.FoodCateguryDescription

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

private void frmFoodCategury\_FormClosed(object sender, FormClosedEventArgs e) // این متد در زمان بسته شدن فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " بسته شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount = dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (dataGridView1.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Txt\_CateguryID.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();

Txt\_AssetCategury.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

}

}

}

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_AssetCategury.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("ورود عنوان الزامی می باشد");

return; // خروج از متد

}

else

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodCategury FoodCtg = new tblFoodCategury();

FoodCtg.FoodCateguryDescription = Txt\_AssetCategury.Text.Trim();// دریافت مقدار از فیلد متنی

db.tblFoodCateguries.InsertOnSubmit(FoodCtg);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

btnAdd.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکورد جدید درج گردید.");

}

Txt\_AssetCategury.Text = "";

Txt\_CateguryID.Text = "";

}

private void Btn\_Edit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_CateguryID.Text == "" || Txt\_AssetCategury.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل ویرایش رکورد اطمینان دارید", "اخطار ویرایش", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodCategury FoodCtg = db.tblFoodCateguries.Single(n => n.FoodCateguryID == Decimal.Parse(Txt\_CateguryID.Text.Trim()));

FoodCtg.FoodCateguryDescription = Txt\_AssetCategury.Text.Trim();// دریافت مقدار از فیلد متنی

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_AssetCategury.Text.Trim() + " -ID= " + Txt\_CateguryID.Text.Trim() + "ویرایش");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

Txt\_AssetCategury.Text = "";

Txt\_CateguryID.Text = "";

}

private void Btn\_Delete\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_CateguryID.Text == "" || Txt\_AssetCategury.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل حذف رکورد اطمینان دارید", "اخطار حذف", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodCategury FoodCtg = db.tblFoodCateguries.Single(n => n.FoodCateguryID == Decimal.Parse(Txt\_CateguryID.Text.Trim()));

db.tblFoodCateguries.DeleteOnSubmit(FoodCtg);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_AssetCategury.Text.Trim() + " -ID= " + Txt\_CateguryID.Text.Trim() + "حذف شد");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

Txt\_AssetCategury.Text = "";

Txt\_CateguryID.Text = "";

}

private void Btn\_Search\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_AssetCategury.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان یا بخشی از عنوان را برای جستجو وارد نمائید.");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodCateguries

where SqlMethods.Like(Row.FoodCateguryDescription , "%" + Txt\_AssetCategury.Text.Trim() + "%")

select new

{

Row.FoodCateguryID,

Row.FoodCateguryDescription

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_AssetCategury.Text = "";

Btn\_Refresh.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

private void Btn\_Refresh\_Click(object sender, EventArgs e) // رخداد کلیک

{

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

Btn\_Refresh.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Search.BringToFront();// نمایش کلید در بالای سایر آبجکتها

}

}

}

**کدهای مربوط به فرم ثبت واحد شمارش گل**



نمایه - فرم ثبت واحد شمارش گل

namespace FlowerShop

{

public partial class frmFoodUnits : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmFoodUnits()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void frmFoodUnits\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

private void frmFoodUnits\_FormClosed(object sender, FormClosedEventArgs e) // این متد در زمان بسته شدن فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " بسته شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

}

private void btnAdd\_Click(object sender, EventArgs e) // رخداد کلیک

{

Txt\_AssetUnit.Text = String.Empty; // حذف مقادیر متنی

btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

مایش پیغام // MessageBox.Show("لطفا اطلاعات را وارد نمایید");

}

private void UDF\_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodUnits

orderby Row.FoodUnitID descending

select new

{

Row.FoodUnitID,

Row.FoodUnitDescription

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("ورود عنوان الزامی می باشد");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodUnit FoodUnt = new tblFoodUnit();

FoodUnt.FoodUnitDescription = Txt\_AssetUnit.Text.Trim();// دریافت مقدار از فیلد متنی

db.tblFoodUnits.InsertOnSubmit(FoodUnt);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

Txt\_UnitID.Text = "";

Txt\_AssetUnit.Text = "";

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount = dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (dataGridView1.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Txt\_UnitID.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();

Txt\_AssetUnit.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

}

}

}

}

private void Btn\_Edit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_UnitID.Text == "" || Txt\_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل ویرایش رکورد اطمینان دارید", "اخطار ویرایش", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodUnit FoodUnt = db.tblFoodUnits.Single(n => n.FoodUnitID == int.Parse(Txt\_UnitID.Text.Trim()));

FoodUnt.FoodUnitDescription = Txt\_AssetUnit.Text.Trim();// دریافت مقدار از فیلد متنی

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_AssetUnit.Text.Trim() + " -ID= " + Txt\_UnitID.Text.Trim() + "ویرایش");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

Txt\_UnitID.Text = "";

Txt\_AssetUnit.Text = "";

}

}

private void Btn\_Delete\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_UnitID.Text == "" || Txt\_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل حذف رکورد اطمینان دارید", "اخطار حذف", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_AssetUnit.Text.Trim() + " -ID= " + Txt\_UnitID.Text.Trim() + "حذف");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodUnit FoodUnt = db.tblFoodUnits.Single(n => n.FoodUnitID == int.Parse(Txt\_UnitID.Text.Trim()));

db.tblFoodUnits.DeleteOnSubmit(FoodUnt);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

Txt\_UnitID.Text = "";

Txt\_AssetUnit.Text = "";

}

private void Btn\_Search\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_AssetUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان یا بخشی از عنوان را برای جستجو وارد نمائید.");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodUnits

where SqlMethods.Like(Row.FoodUnitDescription, "%" + Txt\_AssetUnit.Text.Trim() + "%")

select new

{

Row.FoodUnitID,

Row.FoodUnitDescription

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_AssetUnit.Text = "";

Btn\_Refresh.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

private void Btn\_Refresh\_Click(object sender, EventArgs e) // رخداد کلیک

{

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

Btn\_Refresh.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Search.BringToFront();// نمایش کلید در بالای سایر آبجکتها

}

}

}

**کدهای مربوط به فرم تعریف عنوان گل**



نمایه - فرم ثبت عنوان گل

namespace FlowerShop

{

public partial class frmDefineFoods : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmDefineFoods()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void frmDefineFoods\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

private void frmDefineFoods\_FormClosed(object sender, FormClosedEventArgs e) // این متد در زمان بسته شدن فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " بسته شد.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

}

private void UDF\_CmbFill()

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query1 = from Row in db.tblFoodUnits

orderby Row.FoodUnitID descending

select new

{

Row.FoodUnitID,

Row.FoodUnitDescription

};

var Query2 = from Row in db.tblFoodCateguries

orderby Row.FoodCateguryID descending

select new

{

Row.FoodCateguryID,

Row.FoodCateguryDescription

};

Cmb\_FoodUnit.DisplayMember = "FoodUnitDescription";

Cmb\_FoodUnit.ValueMember = "FoodUnitID";

Cmb\_FoodCategury.DisplayMember = "FoodCateguryDescription";

Cmb\_FoodCategury.ValueMember = "FoodCateguryID";

Cmb\_FoodUnit.DataSource = Query1;

Cmb\_FoodCategury.DataSource = Query2;

}

private void UDF\_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodsDescriptions

orderby Row.FoodID descending

select new

{

Row.FoodID,

Row.FoodDescription,

Row.tblFoodCategury.FoodCateguryDescription,

Row.tblFoodUnit.FoodUnitDescription

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

private void btnAdd\_Click(object sender, EventArgs e) // رخداد کلیک

{

Txt\_FoodDescription.Text = String.Empty; // حذف مقادیر متنی

btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Txt\_FoodDescription.Clear();

مایش پیغام // MessageBox.Show("لطفا اطلاعات را وارد نمایید");

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodDescription.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("ورود عنوان الزامی می باشد");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodsDescription FoodDs = new tblFoodsDescription();

FoodDs.FoodDescription = Txt\_FoodDescription.Text.Trim();// دریافت مقدار از فیلد متنی

FoodDs.FoodCateguryID = Decimal.Parse(Cmb\_FoodCategury.SelectedValue.ToString());

FoodDs.FoodUnitID = int.Parse(Cmb\_FoodUnit.SelectedValue.ToString());

db.tblFoodsDescriptions.InsertOnSubmit(FoodDs);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_CmbFill();

btnAdd.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Txt\_FoodDescription.Clear();

مایش پیغام // MessageBox.Show("رکورد جدید درج گردید.");

}

private void Btn\_Edit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodDescription.Text == "" || Txt\_FoodID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب فرمائید.");

return; // خروج از متد

}

if (Cmb\_FoodUnit.Text == "" || Cmb\_FoodCategury.Text =="")

{

مایش پیغام // MessageBox.Show("انتخاب دسته بندی و واحد شمارش الزامی می باشد");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل ویرایش رکورد اطمینان دارید", "اخطار ویرایش", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n => n.FoodID == int.Parse(Txt\_FoodID.Text.Trim()));

FoodDs.FoodDescription = Txt\_FoodDescription.Text.Trim();// دریافت مقدار از فیلد متنی

FoodDs.FoodUnitID = int.Parse(Cmb\_FoodUnit.SelectedValue.ToString());

FoodDs.FoodCateguryID = Decimal.Parse(Cmb\_FoodCategury.SelectedValue.ToString());

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_FoodDescription.Text.Trim() + " -ID= " + Txt\_FoodID.Text.Trim() + "ویرایش");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

Txt\_FoodID.Clear();

Txt\_FoodDescription.Clear();

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount = dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (dataGridView1.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Txt\_FoodID.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();

Txt\_FoodDescription.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

}

UDF\_SelectCmbValue(int.Parse(Txt\_FoodID.Text.Trim()));

}

}

}

private void UDF\_SelectCmbValue(int int\_FoodUD)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n => n.FoodID == int\_FoodUD);

Cmb\_FoodUnit.SelectedValue = FoodDs.FoodUnitID;

Cmb\_FoodCategury.SelectedValue = FoodDs.FoodCateguryID;

}

private void Btn\_Delete\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodDescription.Text == "" || Txt\_FoodID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب فرمائید.");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل حذف رکورد اطمینان دارید", "اخطار حذف", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n => n.FoodID == int.Parse(Txt\_FoodID.Text.Trim()));

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_FoodDescription.Text.Trim() + " -ID= " + Txt\_FoodID.Text.Trim() + "حذف");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

db.tblFoodsDescriptions.DeleteOnSubmit(FoodDs);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

Txt\_FoodID.Clear();

Txt\_FoodDescription.Clear();

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void Btn\_Search\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodDescription.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان یا بخشی از عنوان را برای جستجو وارد نمائید.");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodsDescriptions

where SqlMethods.Like(Row.FoodDescription , "%" + Txt\_FoodDescription.Text.Trim() + "%")

select new

{

Row.FoodID,

Row.FoodDescription,

Row.tblFoodCategury.FoodCateguryDescription,

Row.tblFoodUnit.FoodUnitDescription

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_FoodDescription.Text = "";

Txt\_FoodID.Clear();

Btn\_Refresh.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

private void Btn\_Refresh\_Click(object sender, EventArgs e) // رخداد کلیک

{

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_CmbFill();

Btn\_Refresh.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Search.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Txt\_FoodDescription.Clear();

Txt\_FoodID.Clear();

}

}

}

**کدهای مربوط به فرم ثبت موجودی گل**



نمایه - ثبت موجودی گل

namespace FlowerShop

{

public partial class frmMojodi : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmMojodi()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void UDF\_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblMojodis

orderby Row.RowMojodi descending

select new

{

Row.RowMojodi,

Row.FoodID,

Row.tblFoodsDescription.FoodDescription,

Row.EnterDate,

Row.NumberOfFood

};

Grd\_Mojodi.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

private void UDF\_CmbFill()

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query1 = from Row in db.tblFoodsDescriptions

orderby Row.FoodID descending

select new

{

Row.FoodID,

Row.FoodDescription

};

Cmb\_FoodDes.DisplayMember = "FoodDescription";

Cmb\_FoodDes.ValueMember = "FoodID";

Cmb\_FoodDes.DataSource = Query1;

}

private void btnAdd\_Click(object sender, EventArgs e) // رخداد کلیک

{

Txt\_FoodUnit.Text = String.Empty; // حذف مقادیر متنی

btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Txt\_FoodUnit.Clear();

مایش پیغام // MessageBox.Show("لطفا موجودی گل را وارد نمایید");

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodUnit.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("ورود عنوان الزامی می باشد");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblMojodi mojodi = new tblMojodi();

Class\_ShamsiTools shamsi = new Class\_ShamsiTools();

shamsi.UDF\_GetCurrentShamsiDate();

mojodi.FoodID = int.Parse(Cmb\_FoodDes.SelectedValue.ToString());

mojodi.EnterDate = shamsi.CurrentShmasiDate;

mojodi.NumberOfFood =float.Parse( Txt\_FoodUnit.Text.Trim());

db.tblMojodis.InsertOnSubmit(mojodi);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_CmbFill();

btnAdd.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Txt\_FoodUnit.Clear();

مایش پیغام // MessageBox.Show("رکورد جدید درج گردید.");

}

private void frmMojodi\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

private void Grd\_Mojodi\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount = Grd\_Mojodi. GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (Grd\_Mojodi.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Txt\_FoodID.Text = Grd\_Mojodi.Rows[int.Parse(Grd\_Mojodi.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

Txt\_FoodUnit.Text = Grd\_Mojodi.Rows[int.Parse(Grd\_Mojodi.SelectedCells[i].RowIndex.ToString())].Cells[4].Value.ToString();

Cmb\_FoodDes.SelectedValue = int.Parse(Grd\_Mojodi.Rows[int.Parse(Grd\_Mojodi.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString());

}

}

}

}

private void Btn\_Edit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان یا بخشی از عنوان را انتخاب فرمائید.");

return; // خروج از متد

}

if (Cmb\_FoodDes.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان یا بخشی از عنوان را انتخاب فرمائید.");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل ویرایش رکورد اطمینان دارید", "اخطار ویرایش", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblMojodi mogodi = new tblMojodi();

Class\_ShamsiTools shamsi = new Class\_ShamsiTools();

shamsi.UDF\_GetCurrentShamsiDate();

mogodi.FoodID = int.Parse(Cmb\_FoodDes.SelectedValue.ToString());

mogodi.EnterDate = shamsi.CurrentShmasiDate;

mogodi.NumberOfFood = float.Parse(Txt\_FoodUnit.Text.Trim());

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_FoodUnit.Text.Trim() + " -ID= " + Txt\_FoodID.Text.Trim() + "ویرایش");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

Txt\_FoodID.Clear();

Txt\_FoodUnit.Clear();

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void Btn\_Delete\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان را انتخاب فرمائید.");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل حذف رکورد اطمینان دارید", "اخطار حذف", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFoodsDescription FoodDs = db.tblFoodsDescriptions.Single(n => n.FoodID == int.Parse(Txt\_FoodID.Text.Trim()));

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Cmb\_FoodDes.SelectedItem.ToString() + " -ID= " + Txt\_FoodID.Text.Trim() + "حذف");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

db.tblFoodsDescriptions.DeleteOnSubmit(FoodDs);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

Txt\_FoodID.Clear();

Txt\_FoodUnit.Clear();

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void Btn\_Search\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_Search.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا عنوان یا بخشی از عنوان را برای جستجو وارد نمائید.");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblMojodis

where SqlMethods.Like(Row.tblFoodsDescription.FoodDescription , "%" + Txt\_Search.Text.Trim() + "%")

select new

{

Row.RowMojodi,

Row.FoodID,

Row.tblFoodsDescription.FoodDescription,

Row.EnterDate,

Row.NumberOfFood

};

Grd\_Mojodi.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_FoodUnit.Text = "";

Txt\_FoodID.Clear();

Btn\_Refresh.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

private void Btn\_Refresh\_Click(object sender, EventArgs e) // رخداد کلیک

{

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_CmbFill();

Btn\_Refresh.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Search.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Txt\_FoodUnit.Clear();

Txt\_FoodID.Clear();

}

}

}

**کدهای مربوط به فرم ثبت مشخصات مشتری**



نمایه - فرم ثبت مشخصات مشتری

namespace FlowerShop

{

public partial class frmDefineCustomer : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmDefineCustomer()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void UDF\_TextBoxEmpty()

{

Txt\_CustomerID.Clear();

Txt\_Name.Clear();

Txt\_Family.Clear();

Txt\_Tell.Clear();

Txt\_Mobail.Clear();

Txt\_Address.Clear();

Chb\_IsActive.Checked = false;

}

private void UDF\_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblCustomers

orderby Row.CustomerID descending

select new

{

Row.CustomerID,

Row.CustomerName,

Row.CustomerFamily,

Row.DateOzviatSH,

Row.CustomerTell,

Row.CustomerMobail,

Row.CustomerAddress,

Row.CustomerIsActive

};

Grd\_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

private void Btn\_DeleteCustomer\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_CustomerID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا مشتری مورد نظر را انتخاب فرمائید.");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل حذف رکورد اطمینان دارید", "اخطار حذف", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblCustomer Customer = db.tblCustomers.Single(n => n.CustomerID == decimal.Parse(Txt\_CustomerID.Text.Trim()));

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_CustomerID.Text.Trim() + "-ID" + Txt\_CustomerID.Text.Trim() + "حذف");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

db.tblCustomers.DeleteOnSubmit(Customer);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_TextBoxEmpty();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void frmDefineCustomer\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

private void btnAdd\_Click(object sender, EventArgs e) // رخداد کلیک

{

Txt\_Name.Text = String.Empty; // حذف مقادیر متنی

Txt\_Family.Text = String.Empty; // حذف مقادیر متنی

Txt\_Tell.Text = String.Empty; // حذف مقادیر متنی

Txt\_Mobail.Text = String.Empty; // حذف مقادیر متنی

Txt\_Address.Text = String.Empty; // حذف مقادیر متنی

btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Txt\_Family.Clear();

مایش پیغام // MessageBox.Show("لطفا اطلاعات را وارد نمایید");

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_Name.Text == "" || Txt\_Family.Text == "" || Txt\_Tell.Text == "" || Txt\_Mobail.Text == "" || Txt\_Address.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("ورود موارد الزامی می باشد");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblCustomer Customer = new tblCustomer();

Class\_ShamsiTools shamsi = new Class\_ShamsiTools();

shamsi.UDF\_GetCurrentShamsiDate();

Customer.CustomerName = Txt\_Name.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerFamily = Txt\_Family.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.DateOzviatSH = shamsi.CurrentShmasiDate;

Customer.CustomerTell = Txt\_Tell.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerMobail = Txt\_Mobail.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerAddress = Txt\_Address.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerIsActive = Chb\_IsActive.Checked;

db.tblCustomers.InsertOnSubmit(Customer);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

btnAdd.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

UDF\_TextBoxEmpty();

مایش پیغام // MessageBox.Show("رکورد جدید درج گردید.");

}

private void Grd\_Customer\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount =Grd\_Customer. GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (Grd\_Customer.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Boolean Bln\_Check;

Txt\_CustomerID.Text = Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();

Txt\_Name.Text = Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

Txt\_Family.Text = Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[2].Value.ToString();

Txt\_Tell.Text = Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[4].Value.ToString();

Txt\_Mobail.Text = Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[5].Value.ToString();

Txt\_Address.Text = Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[6].Value.ToString();

Bln\_Check = Boolean.Parse(Grd\_Customer.Rows[int.Parse(Grd\_Customer.SelectedCells[i].RowIndex.ToString())].Cells[7].Value.ToString());

if (Bln\_Check == true)

{

Chb\_IsActive.Checked = true;

}

else if (Bln\_Check == false)

{

Chb\_IsActive.Checked = false;

}

}

}

}

}

private void Btn\_Edit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_CustomerID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا مشتری مورد نظر را انتخاب کنید");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل ویرایش رکورد اطمینان دارید", "اخطار ویرایش", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblCustomer Customer = db.tblCustomers.Single(n => n.CustomerID == Decimal.Parse(Txt\_CustomerID.Text.Trim()));

Customer.CustomerName = Txt\_Name.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerFamily = Txt\_Family.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerTell = Txt\_Tell.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerMobail = Txt\_Mobail.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerAddress = Txt\_Address.Text.Trim();// دریافت مقدار از فیلد متنی

Customer.CustomerIsActive = Chb\_IsActive.Checked;

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_Name.Text.Trim() + Txt\_Family.Text.Trim() + " -ID= " + Txt\_CustomerID.Text.Trim() + "ویرایش");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_TextBoxEmpty();

}

}

private void Btn\_SearchCustomer\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_Search.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا فیلد جستجو را وارد نمائید.");

return; // خروج از متد

}

if (Rdo\_CustomerID.Checked == true)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblCustomers

where Row.CustomerID == Decimal.Parse(Txt\_Search.Text.Trim())

select new

{

Row.CustomerID,

Row.CustomerName,

Row.CustomerFamily,

Row.DateOzviatSH,

Row.CustomerTell,

Row.CustomerMobail,

Row.CustomerAddress,

Row.CustomerIsActive

};

Grd\_Customer. DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_Search.Text = "";

button3.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_SearchCustomer.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

else if (Rdo\_CustomerName.Checked == true)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblCustomers

where SqlMethods.Like(Row.CustomerName , "%" +Txt\_Search.Text.Trim() + "%")

select new

{

Row.CustomerID,

Row.CustomerName,

Row.CustomerFamily,

Row.DateOzviatSH,

Row.CustomerTell,

Row.CustomerMobail,

Row.CustomerAddress,

Row.CustomerIsActive

};

Grd\_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_Search.Text = "";

button3.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_SearchCustomer.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

else if (Rdo\_CustomerFamily.Checked == true)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblCustomers

where SqlMethods.Like(Row.CustomerFamily , "%" + Txt\_Search.Text.Trim() + "%")

select new

{

Row.CustomerID,

Row.CustomerName,

Row.CustomerFamily,

Row.DateOzviatSH,

Row.CustomerTell,

Row.CustomerMobail,

Row.CustomerAddress,

Row.CustomerIsActive

};

Grd\_Customer.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_Search.Text = "";

button3.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_SearchCustomer.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

}

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

{

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

button3.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

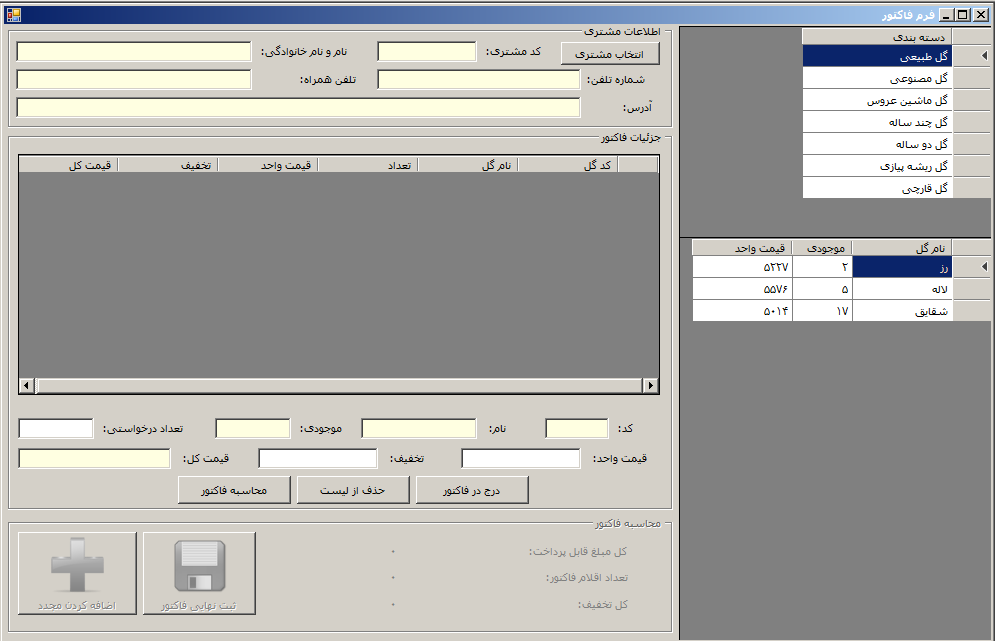
Btn\_SearchCustomer.BringToFront();// نمایش کلید در بالای سایر آبجکتها

}

}

}

**کدهای مربوط به فرم ثبت فاکتور فروش**



نمایه - فرم ایجاد فاکتور فروش

namespace FlowerShop

{

public partial class frmFactor : Form // علامت ارث بری از کلاس اصلی فرم

{

frmSearchCustomer frm001;

DataTable FactDt = new DataTable();

public frmFactor()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

//درج مقادیر گروه های گل در گریدویو اول

private void UDF\_GrdCategoryFill()

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodCateguries

orderby Row.FoodCateguryID ascending

select new

{

Row.FoodCateguryID,

Row.FoodCateguryDescription

};

Grd\_FoodCategory.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

// درج مقادیر گل در هر گروه با استفاده از گروه کالا در گرید ویوی دوم

private void UDF\_CatID(Decimal Dec\_CategoryID)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblFoodsDescriptions

where Row.FoodCateguryID == Dec\_CategoryID

select new

{

Row.FoodID,

Row.FoodDescription

};

DataTable dt = new DataTable();

dt.Columns.Add("FoodID", typeof(int));

dt.Columns.Add("FoodDescription", typeof(String));

dt.Columns.Add("FoodStock", typeof(float));

foreach (var Row in Query)

{

dt.Rows.Add(new Object[] { Row.FoodID, Row.FoodDescription, UDF\_GetStackRate(Row.FoodID) });

}

Grd\_FoodDescription.DataSource = dt;

}

/// <summary>

/// جهت دریافت موجودی گل

/// </summary>

/// <param name="int\_FoodID"></param>

/// <returns name="int\_Stock"></returns>

private float UDF\_GetStackRate(int int\_FoodID)

{

float flt\_Stock = 0;

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

try

{

xtblMandehGaza tblStock = db.xtblMandehGazas.Single(n => n.FoodsID == int\_FoodID);

flt\_Stock = float.Parse(tblStock.MandehGaza.ToString());

}

catch

{

flt\_Stock = 0;

}

return flt\_Stock;

}

private void Grd\_FoodCategory\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

UDF\_TXTFactorClear();

Int32 selectedCellCount = Grd\_FoodCategory.GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (Grd\_FoodCategory.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Decimal Dec\_CategoryID = Decimal.Parse(Grd\_FoodCategory.Rows[int.Parse(Grd\_FoodCategory.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString());

UDF\_CatID(Dec\_CategoryID);

}

}

}

}

private void frmFactor\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_GrdCategoryFill();

}

//private void PassData(object sender, int number)

//{

// //Sms\_Menu.Enabled = (Boolean)sender;

// مایش پیغام // MessageBox.Show(number.ToString());

//}

private void Grd\_FoodDescription\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount = Grd\_FoodDescription.GetCellCount(DataGridViewElementStates.Selected);

UDF\_TXTFactorClear();

if (selectedCellCount > 0)

{

if (Grd\_FoodDescription.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Txt\_FoodCode.Text = Grd\_FoodDescription.Rows[int.Parse(Grd\_FoodDescription.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();

Txt\_FoodName.Text = Grd\_FoodDescription.Rows[int.Parse(Grd\_FoodDescription.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

Txt\_Stock.Text = Grd\_FoodDescription.Rows[int.Parse(Grd\_FoodDescription.SelectedCells[i].RowIndex.ToString())].Cells[2].Value.ToString();

Txt\_TedadeDarkhast.Text = Txt\_FeeTotal.Text = Txt\_UnitPrice.Text = Txt\_Tajhfif.Text = "0";

}

}

}

}

private Decimal UDF\_GetFoodINFO(int int\_FoodID)

{

Decimal Dec\_FoodPRice = 0;

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

return Dec\_FoodPRice;

}

private void UDF\_TXTFactorClear()

{

Txt\_FoodCode.Text = "";

Txt\_FoodName.Text = "";

Txt\_Stock.Text = "";

Txt\_TedadeDarkhast.Text = "";

Txt\_UnitPrice.Text = "";

Txt\_Tajhfif.Text = "";

Txt\_FeeTotal.Text = "";

}

private void Txt\_TedadeDarkhast\_KeyPress(object sender, KeyPressEventArgs e)

{

int ascii = Convert.ToInt16(e.KeyChar);

if (ascii >= 48 && ascii <= 57)

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

private void Txt\_Tajhfif\_KeyPress(object sender, KeyPressEventArgs e)

{

int ascii = Convert.ToInt16(e.KeyChar);

if (ascii >= 48 && ascii <= 57)

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

private void Txt\_UnitPrice\_KeyPress(object sender, KeyPressEventArgs e)

{

int ascii = Convert.ToInt16(e.KeyChar);

if (ascii >= 48 && ascii <= 57)

{

e.Handled = false;

}

else

{

e.Handled = true;

}

}

private void Btn\_AddToList\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_FoodName.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("عنوان گل انتخاب نشده است");

return; // خروج از متد

}

if (Decimal.Parse(Txt\_UnitPrice.Text.Trim()) == 0 || Decimal.Parse(Txt\_TedadeDarkhast.Text.Trim()) == 0)

{

مایش پیغام // MessageBox.Show("ورود مقادیر تعداد , قیمت واحد الزامی می باشد");

return; // خروج از متد

}

if (Decimal.Parse(Txt\_TedadeDarkhast.Text.Trim()) > Decimal.Parse(Txt\_Stock.Text.Trim()))

{

مایش پیغام // MessageBox.Show("مقدار انتخاب شده بیش از مقدار موجودی می باشد");

return; // خروج از متد

}

if (FactDt.Rows.Count <= 0 && FactDt.Columns.Count <= 0)

{

FactDt.Columns.Add("FoodID", typeof(int));

FactDt.Columns.Add("FoodName", typeof(string));

FactDt.Columns.Add("FoodQuantity", typeof(float));

FactDt.Columns.Add("FoodUnitPrice", typeof(Decimal));

FactDt.Columns.Add("FoodDiscount", typeof(Decimal));

FactDt.Columns.Add("FoodTotalPrice", typeof(Decimal));

FactDt.Rows.Add(int.Parse(Txt\_FoodCode.Text.Trim()), Txt\_FoodName.Text.Trim(), float.Parse(Txt\_TedadeDarkhast.Text.Trim())

, Decimal.Parse(Txt\_UnitPrice.Text.Trim()) , Decimal.Parse(Txt\_Tajhfif.Text.Trim()), Decimal.Parse(Txt\_FeeTotal.Text.Trim()));

Grd\_Factor.DataSource = FactDt;

}

else

{

int int\_CntFound = 0;

float flt\_darkhast = 0;

foreach (DataRow Row in FactDt.Rows)

{

if (int.Parse(Row[0].ToString().Trim()) == int.Parse(Txt\_FoodCode.Text.Trim()))

{

int\_CntFound++;

flt\_darkhast = float.Parse(Row[2].ToString());

}

}

if (int\_CntFound == 0)// در صورتی که کلای تکراری نباشد در فاکتور درج می شود

{

FactDt.Rows.Add(int.Parse(Txt\_FoodCode.Text.Trim()), Txt\_FoodName.Text.Trim(), Decimal.Parse(Txt\_TedadeDarkhast.Text.Trim())

, Decimal.Parse(Txt\_UnitPrice.Text.Trim()) , Decimal.Parse(Txt\_Tajhfif.Text.Trim()), Decimal.Parse(Txt\_FeeTotal.Text.Trim()));

Grd\_Factor.DataSource = FactDt;

}

else

{//

float flt\_Mojoodi2 = UDF\_GetStackRate(int.Parse(Txt\_FoodCode.Text.Trim()));

//flt\_Mojoodi2 += flt\_darkhast;

if (flt\_Mojoodi2 < (flt\_darkhast + float.Parse(Txt\_TedadeDarkhast.Text.Trim())))

{

مایش پیغام // MessageBox.Show("مقدار موجودی کافی نمی باشد");

}

else

{

foreach (DataRow Row in FactDt.Rows)// درج اقلام تکراری به لیست به صورت مجموع

{

if (int.Parse(Row[0].ToString().Trim()) == int.Parse(Txt\_FoodCode.Text.Trim()))

{

int\_CntFound++;

float flt\_Quantity = float.Parse(Row[2].ToString());

flt\_Quantity += float.Parse(Txt\_TedadeDarkhast.Text.Trim());

Decimal Dec\_UnitPrice = Decimal.Parse(Row[3].ToString());

Decimal Dec\_Takhfif = (Decimal.Parse(Row[4].ToString()) + Decimal.Parse(Txt\_Tajhfif.Text.Trim()));

Decimal Dec\_TotalPrice = ((Dec\_UnitPrice \* Decimal.Parse(flt\_Quantity.ToString())) - Dec\_Takhfif);

Row[2] = flt\_Quantity.ToString();

Row[3] = Dec\_UnitPrice.ToString();

Row[4] = Dec\_Takhfif.ToString();

Row[5] = Dec\_TotalPrice.ToString();

}

}

}

}

}

}

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

{

if (frm001 == null || frm001.IsDisposed) // بررسی اینکه آیا فرم قبلا باز شده یا خیر

{

frm001 = new frmSearchCustomer();

frm001.passControl = new frmSearchCustomer.PassControl1(PassData);

frm001.ShowDialog();

}

else

{

frm001.Dispose();

frm001 = new frmSearchCustomer();

frm001.passControl = new frmSearchCustomer.PassControl1(PassData);

frm001.ShowDialog();

}

}

private void PassData(object sender, int number)

{

Boolean Bln\_Add\_Detail = (Boolean)sender;

int int\_Details\_Count = number;

if (Bln\_Add\_Detail == true)

{

Txt\_CustomerCode.Text = number.ToString();

}

else

{

مایش پیغام // MessageBox.Show("کد مشتری انتخاب نشده است");

}

}

private void Txt\_CustomerCode\_TextChanged(object sender, EventArgs e)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

try

{

tblCustomer tblCmr = db.tblCustomers.Single(n => n.CustomerID == Decimal.Parse(Txt\_CustomerCode.Text.Trim()));

Txt\_NameSname.Text = tblCmr.CustomerName + " " + tblCmr.CustomerFamily;

Txt\_Phone.Text = tblCmr.CustomerTell;

Txt\_Mobile.Text = tblCmr.CustomerMobail;

Txt\_Address.Text = tblCmr.CustomerAddress;

}

catch

{

}

}

private void Txt\_TedadeDarkhast\_TextChanged(object sender, EventArgs e)

{

try

{

Txt\_FeeTotal.Text = ((Decimal.Parse(Txt\_TedadeDarkhast.Text.Trim()) \* Decimal.Parse(Txt\_UnitPrice.Text.Trim())) - Decimal.Parse(Txt\_Tajhfif.Text.Trim())).ToString();

}

catch

{ }

}

private void Txt\_UnitPrice\_TextChanged(object sender, EventArgs e)

{

try

{

Txt\_FeeTotal.Text = ((Decimal.Parse(Txt\_TedadeDarkhast.Text.Trim()) \* Decimal.Parse(Txt\_UnitPrice.Text.Trim())) - Decimal.Parse(Txt\_Tajhfif.Text.Trim())).ToString();

}

catch

{ }

}

private void Txt\_Tajhfif\_TextChanged(object sender, EventArgs e)

{

try

{

Txt\_FeeTotal.Text = ((Decimal.Parse(Txt\_TedadeDarkhast.Text.Trim()) \* Decimal.Parse(Txt\_UnitPrice.Text.Trim())) - Decimal.Parse(Txt\_Tajhfif.Text.Trim())).ToString();

}

catch

{ }

}

private void Btn\_Delete\_Click(object sender, EventArgs e) // رخداد کلیک

{

Int32 selectedCellCount = Grd\_Factor.GetCellCount(DataGridViewElementStates.Selected);

int int\_FoodID = -5;

for (int i = 0; i < selectedCellCount; i++)

{

int\_FoodID = int.Parse(Grd\_Factor.Rows[int.Parse(Grd\_Factor.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString());

}

if (int\_FoodID == -5)

{

مایش پیغام // MessageBox.Show("آیتمی از فاکتور برای حذف انتخاب نشده است");

return; // خروج از متد

}

for (int i = FactDt.Rows.Count - 1; i >= 0; i--)

{

DataRow dr = FactDt.Rows[i];

if (int.Parse(dr["FoodID"].ToString()) == int\_FoodID)

{

dr.Delete();

}

}

Grd\_Factor.DataSource = FactDt;

}

private void Btn\_Close\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_CustomerCode.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا کد مشتری را انتخاب فرمائید");

return; // خروج از متد

}

GrPr\_Mohasebeh.Enabled = true;

Grpr\_Custumer.Enabled = false;

Grpr\_Factor.Enabled = false;

Decimal Dec\_TotalPrice = 0;

Decimal Dec\_TotalTakhfif = 0;

foreach (DataRow Row in FactDt.Rows)

{

Dec\_TotalPrice += Decimal.Parse(Row[5].ToString());

Dec\_TotalTakhfif += Decimal.Parse(Row[4].ToString());

}

Lbl\_Total\_Price.Text = Dec\_TotalPrice.ToString() + " ریال ";

Lbl\_Total\_Takhfif.Text = Dec\_TotalTakhfif.ToString() + " ریال ";

Lbl\_Total\_Aglam.Text = FactDt.Rows.Count.ToString() + " مورد ";

}

private void btnAdd\_Click(object sender, EventArgs e) // رخداد کلیک

{

GrPr\_Mohasebeh.Enabled = false;

Grpr\_Custumer.Enabled = true;

Grpr\_Factor.Enabled = true;

Lbl\_Total\_Aglam.Text = "0";

Lbl\_Total\_Price.Text = "0";

Lbl\_Total\_Takhfif.Text = "0";

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblFactorID tblFact = new tblFactorID();

Class\_ShamsiTools shDate = new Class\_ShamsiTools();

shDate.UDF\_GetCurrentShamsiDate();

tblFact.CustomerID = Decimal.Parse(Txt\_CustomerCode.Text.Trim());

tblFact.FactorShDate = shDate.CurrentShmasiDate;

db.tblFactorIDs.InsertOnSubmit(tblFact);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

tblFactorDetail tblFID;

foreach (DataRow Row in FactDt.Rows)

{

tblFID = new tblFactorDetail();

tblFID.FactorID = tblFact.FactorID;

tblFID.FoodID = int.Parse(Row[0].ToString());

tblFID.FoodQuantity = float.Parse(Row[2].ToString());

tblFID.UnitPrice = Decimal.Parse(Row[3].ToString());

tblFID.Takhfif = Decimal.Parse(Row[4].ToString());

tblFID.TaxRate = Decimal.Parse(Row[5].ToString());

db.tblFactorDetails.InsertOnSubmit(tblFID);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

}

Lbl\_Total\_Aglam.Text = "0";

Lbl\_Total\_Price.Text = "0";

Lbl\_Total\_Takhfif.Text = "0";

GrPr\_Mohasebeh.Enabled = false;

Grpr\_Custumer.Enabled = true;

Grpr\_Factor.Enabled = true;

FactDt.Rows.Clear();

for (int i = FactDt.Rows.Count - 1; i >= 0; i--)

{

DataRow dr = FactDt.Rows[i];

{

dr.Delete();

}

}

Grd\_Factor.DataSource = FactDt;

مایش پیغام // MessageBox.Show("کلیه موارد در پایگاه داده درج شدند");

UDF\_TXTFactorClear();

UDF\_TXTCustomerClear();

}

private void UDF\_TXTCustomerClear()

{

Txt\_CustomerCode.Text = "";

Txt\_NameSname.Text = "";

Txt\_Phone.Text = "";

Txt\_Mobile.Text = "";

Txt\_Address.Text = "";

}

}

}

**کدهای مربوط به فرم ثبت کاربران سیستم**



نمایه - فرم مربوط به کاربران سیستم

namespace FlowerShop

{

public partial class frmEditUsers : Form // علامت ارث بری از کلاس اصلی فرم

{

public frmEditUsers()

{

InitializeComponent(); // این متد در هنگام قبل از لود فرم فراخوانی شده و باعث می شود که اجزا و اشیاء استفاده شده بر روی فرم ظاهر و فعال شوند

}

private void UDF\_ClearTextBox()

{

Txt\_UseID.Clear();

Txt\_UserName.Clear();

Txt\_Password.Clear();

Txt\_Name.Clear();

Txt\_Family.Clear();

Txt\_Tell.Clear();

Txt\_Mobail.Clear();

Chb\_BanBit.Checked = false;

}

private void UDF\_CmbFill()

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query1 = from Row in db.tblUserTypes

orderby Row.UserTypeID descending

where Row.UserTypeID != 3

select new

{

Row.UserTypeID,

Row.UsersTypeDescription

};

Cmb\_UserType.DisplayMember = "UsersTypeDescription";

Cmb\_UserType.ValueMember = "UserTypeID";

Cmb\_UserType.DataSource = Query1;

}

private void UDF\_GrdFill() // توسط این متد اطلاعات فراخوانی شده در گرید ویو نمایش داده می شود

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblUsers

orderby Row.UsersID descending

where Row.UserTypeID != 3 select new

{

Row.UsersID,

Row.UsersName,

Row.UsersPass,

Row.UserFName,

Row.UsersSName,

Row.tblUserType.UsersTypeDescription,

Row.UserTell,

Row.UserMobail,

Row.UserBanBit

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

}

private void Btn\_Save\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_UserName.Text == "" || Txt\_Password.Text == "" || Txt\_Name.Text == "" || Txt\_Family.Text == "" || Txt\_Tell.Text == "" || Txt\_Mobail.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("ورود موارد خواسته شده الزامی می باشد");

return; // خروج از متد

}

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblUser user = new tblUser();

user.UsersName = Txt\_UserName.Text.Trim();// دریافت مقدار از فیلد متنی

user.UsersPass = Txt\_Password.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserFName = Txt\_Name.Text.Trim();// دریافت مقدار از فیلد متنی

user.UsersSName = Txt\_Family.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserTypeID = int.Parse(Cmb\_UserType.SelectedValue.ToString());

user.UserTell = Txt\_Tell.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserMobail = Txt\_Mobail.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserBanBit = Chb\_BanBit.Checked;

db.tblUsers.InsertOnSubmit(user);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_CmbFill();

btnAdd.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Save.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

UDF\_ClearTextBox();

مایش پیغام // MessageBox.Show("رکورد جدید درج گردید.");

}

private void btnAdd\_Click\_1(object sender, EventArgs e)

{

Txt\_UseID.Text = String.Empty; // حذف مقادیر متنی

Txt\_UserName.Text = String.Empty; // حذف مقادیر متنی

Txt\_Password.Text = String.Empty; // حذف مقادیر متنی

Txt\_Name.Text = String.Empty; // حذف مقادیر متنی

Txt\_Family.Text = String.Empty; // حذف مقادیر متنی

Txt\_Tell.Text = String.Empty; // حذف مقادیر متنی

Txt\_Mobail.Text = String.Empty; // حذف مقادیر متنی

Chb\_BanBit.Checked = false;

btnAdd.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

UDF\_ClearTextBox();

مایش پیغام // MessageBox.Show("لطفا اطلاعات را وارد نمایید");

}

private void Cmb\_UserType\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void frmEditUsers\_Load(object sender, EventArgs e) // این متد در هنگام لود فرم اجرا می شود

{

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " فعال گردید.");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

private void Btn\_Delete\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_UseID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا کاربر مورد نظر را انتخاب فرمائید.");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل حذف رکورد اطمینان دارید", "اخطار حذف", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblUser user = db.tblUsers.Single(n => n.UsersID == int.Parse(Txt\_UseID.Text.Trim()));

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_UseID.Text.Trim() + " -ID= " + Txt\_UseID.Text.Trim() + "حذف");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

db.tblUsers.DeleteOnSubmit(user);

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

UDF\_ClearTextBox();

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)//این متد در هنگام کلیک بر روی سطرهای گرید ویو انجام می شود و مقادیر مربوط به محتویات هر سلول را بر می گرداند

{

Int32 selectedCellCount = dataGridView1.GetCellCount(DataGridViewElementStates.Selected);

if (selectedCellCount > 0)

{

if (dataGridView1.AreAllCellsSelected(true))

{

مایش پیغام // MessageBox.Show("All cells are selected", "Selected Cells");

}

else

{

System.Text.StringBuilder sb = new System.Text.StringBuilder();

for (int i = 0; i < selectedCellCount; i++)

{

Boolean Bln\_Chk;

Txt\_UseID.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[0].Value.ToString();

Txt\_UserName.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[1].Value.ToString();

Txt\_Password.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[2].Value.ToString();

Txt\_Name.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[3].Value.ToString();

Txt\_Family.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[4].Value.ToString();

Txt\_Tell.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[6].Value.ToString();

Txt\_Mobail.Text = dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[7].Value.ToString();

Bln\_Chk = Boolean.Parse(dataGridView1.Rows[int.Parse(dataGridView1.SelectedCells[i].RowIndex.ToString())].Cells[8].Value.ToString());

if (Bln\_Chk == true)

{

Chb\_BanBit.Checked = true;

}

else if (Bln\_Chk == false)

{

Chb\_BanBit.Checked = false;

}

}

UDF\_SelectCmbValue(int.Parse(Txt\_UseID.Text.Trim()));

}

}

}

private void UDF\_SelectCmbValue(int int\_UserID)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblUser user = db.tblUsers.Single(n => n.UsersID == int\_UserID);

Cmb\_UserType.SelectedValue = user.UserTypeID;

}

private void Btn\_Edit\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_UseID.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا کاربر مورد نظر را انتخاب فرمائید.");

return; // خروج از متد

}

if (Cmb\_UserType.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("انتخاب نوع کاربر الزامی می باشد");

return; // خروج از متد

}

DialogResult dr = مایش پیغام // MessageBox.Show("آیا نسبت به عمل ویرایش رکورد اطمینان دارید", "اخطار ویرایش", مایش پیغام // MessageBoxButtons.OKCancel, مایش پیغام // MessageBoxIcon.Warning);

if (dr == DialogResult.OK)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

tblUser user = db.tblUsers.Single(n => n.UsersID == int.Parse(Txt\_UseID.Text.Trim()));

user.UsersName = Txt\_UserName.Text.Trim();// دریافت مقدار از فیلد متنی

user.UsersPass = Txt\_Password.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserFName = Txt\_Name.Text.Trim();// دریافت مقدار از فیلد متنی

user.UsersSName = Txt\_Family.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserTypeID = int.Parse(Cmb\_UserType.SelectedValue.ToString());

user.UserTell = Txt\_Tell.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserMobail = Txt\_Mobail.Text.Trim();// دریافت مقدار از فیلد متنی

user.UserBanBit = Chb\_BanBit.Checked;

db.SubmitChanges(); // ثبت تغییرات در پایگاه داده

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " رکورد " + Txt\_UserName.Text.Trim() + " -ID= " + Txt\_UseID.Text.Trim() + "ویرایش");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

UDF\_ClearTextBox();

UDF\_CmbFill();

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

}

}

private void Btn\_Search\_Click(object sender, EventArgs e) // رخداد کلیک

{

if (Txt\_Search.Text == "") // بررسی اینکه آیا فیلد متنی خالیست یا خیر

{

مایش پیغام // MessageBox.Show("لطفا فیلد جستجو را وارد نمائید.");

return; // خروج از متد

}

if (Rdo\_UserID.Checked == true)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblUsers

where Row.UsersID == int.Parse(Txt\_Search.Text.Trim())

select new

{

Row.UsersID,

Row.UsersName,

Row.UsersPass,

Row.UserFName,

Row.UsersSName,

Row.tblUserType.UsersTypeDescription,

Row.UserTell,

Row.UserMobail,

Row.UserBanBit

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_Search.Text = "";

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

else if (Rdo\_UserName.Checked == true)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblUsers

where SqlMethods.Like(Row.UsersName, "%" + Txt\_Search.Text.Trim() + "%")

select new

{

Row.UsersID,

Row.UsersName,

Row.UsersPass,

Row.UserFName,

Row.UsersSName,

Row.tblUserType.UsersTypeDescription,

Row.UserTell,

Row.UserMobail,

Row.UserBanBit

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_Search.Text = "";

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

else if (Rdo\_Family.Checked == true)

{

LSC\_FlowerShopDataContext db = new LSC\_FlowerShopDataContext(); // ایجاد شئ از کلاس لینک جهت اتصال به پایگاه داده

var Query = from Row in db.tblUsers

where SqlMethods.Like(Row.UsersSName, "%" + Txt\_Search.Text.Trim() + "%")

select new

{

Row.UsersID,

Row.UsersName,

Row.UsersPass,

Row.UserFName,

Row.UsersSName,

Row.tblUserType.UsersTypeDescription,

Row.UserTell,

Row.UserMobail,

Row.UserBanBit

};

dataGridView1.DataSource = Query; // تخصیص منبع داده جهت نمایش اطلاعات در گرید ویو

ثبت رخداد // Class\_tblLogs LogReg = new Class\_tblLogs(this.Text + " - عملیات جستجو");

LogReg.UDF\_SaveLogToDb(); // با استفاده از این متد رخداد ارسال شده به کلاس در پایگاه داده ثبت ممی شود

if (Query.Count() <= 0)

{

Txt\_Search.Text = "";

Btn\_Save.BringToFront();// نمایش کلید در بالای سایر آبجکتها

Btn\_Search.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

مایش پیغام // MessageBox.Show("رکوردی یافته نشد");

}

}

}

private void Btn\_Refresh\_Click(object sender, EventArgs e) // رخداد کلیک

{

UDF\_GrdFill(); // فراخوانی متد نمایش رکوردها در گرید ویو

UDF\_CmbFill();

Btn\_Refresh.SendToBack(); // نمایش کلید در پشت سایر آبجکتها

Btn\_Search.BringToFront();// نمایش کلید در بالای سایر آبجکتها

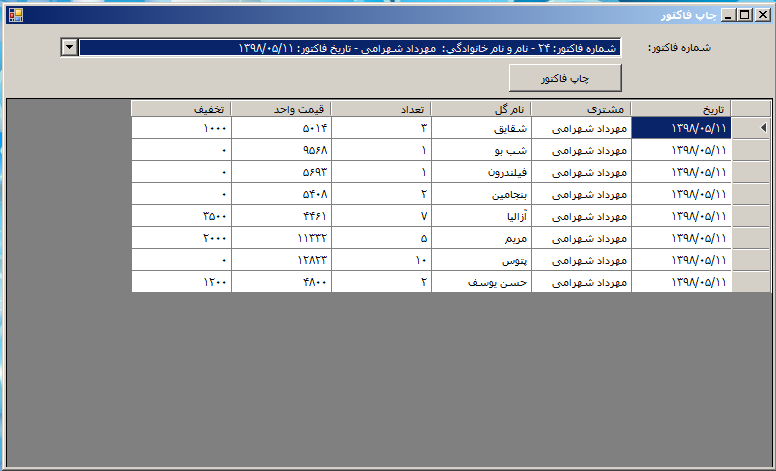
UDF\_ClearTextBox();

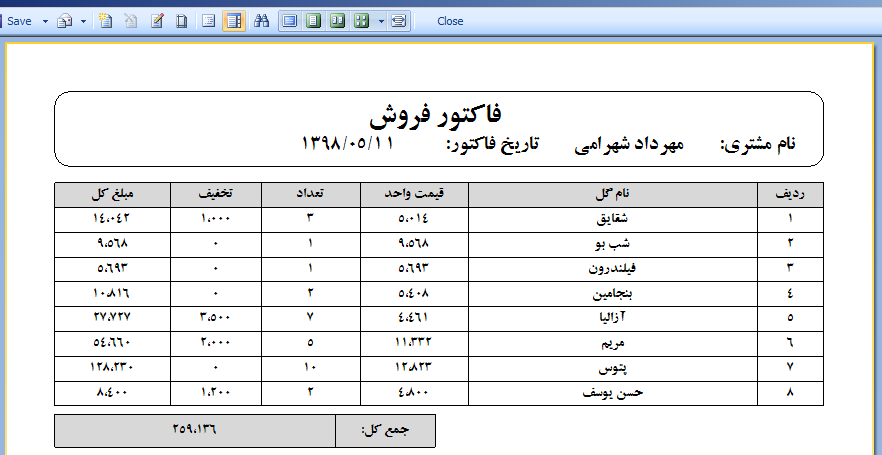
}

}

}

## چاپ فاکتور





namespace FlowerShop

{

public partial class frmFactorPrint : Form

{

public frmFactorPrint()

{

InitializeComponent();

}

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

{

UDF\_GetData();

}

private void UDF\_GetData()// واکشی دیتا

{

this.xtblDropFactorsTableAdapter.Fill(this.dataset.xtblDropFactors);

this.xtblFactorRptTableAdapter.FillByFactorID(this.dataset.xtblFactorRpt, int.Parse(comboBox1.SelectedValue.ToString()));

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

this.xtblFactorRptTableAdapter.FillByFactorID(this.dataset.xtblFactorRpt, int.Parse(comboBox1.SelectedValue.ToString()));

}

private void Btn\_AddToList\_Click(object sender, EventArgs e)//نمایش فاکتور

{

StiReport Report = new StiReport();

Report = (StiReport)stiReport1.Clone();

Report.RegData(this.dataset.xtblFactorRpt);

Report.Compile();

Report.Show();

}

}

}

فصل چهارم: جمع بندی ، نتیجه گیری و پیشنهادات

**جمع بندی**

سیستم ارائه شده نمونه ای از صدها مورد از سیستمهای نرم افزاری است که به صورت تحت ویندوز سازی شده است. زبان برنامه نویسی مورد استفاده نیز یک زبان برنامه نویسی شی گرا با عنوان C# می باشد که به علت وجود منابع آموزشی بسیار زیاد نسبت به سایر زبان های شئ گرا ، از این زبان برنامه نویسی به عنوان زبان برنامه سازی Base استفاده شده است.

C# یکی از آن زبان هایی است که بر خلاف C ، امکانات شی گرایی را در بطن وجودش پیاده سازی کرده است. برنامه نویسان میتوانند با استفاده از C# به صورت قدرتمندی اقدام به تولید برنامه های شی گرا نمایند. اما به خاطر داشته باشید وجود تفکر شی گرایی به هنگام برنامه نویسی مهم تر ار وجود امکانات شی گرایی در آن زبان خاص است.

حالا که با مفهوم شی گرایی در دنیای حقیقی آشنا شدید می توانید با دنبال کردن این سری پروژه ها ، تمام این مفاهیم را با استفاده از C# در دنیای برنامه نویسی هم پیاده سازی کنید. در این پروژه سعی شده است تا جدید ترین روش های شی گرایی در C# بررسی شود. به همین خاطر ما از کلاس های سبک جدید C# برای آموزش استفاده کرده ایم که به مراتب قابلیت های بالاتری نسبت به کلاس های کلاسیک #C دارند.

برای اتصال به پایگاه داده از تکنولوژی ADO.NET و LINQ استفاده خواهد شد. البته موتور پایگاه داده SQL Server 2005 می باشد که یک موتور بسیار قوی با مدیریت بسیار عالی برای دسترسی به داده است. این مورد تنها زمانی مشخص می گردد که داده ها در حجم بالا میان Server و Client در حال رد و بدل شدن هستند.

**نتیجه گیری**

پس از پیاده سازی این طرح نتایج حاصله از بررسی های به عمل آمده در مورد تکنولوژیهای برنامه نویسی این نتیجه حاصل شد که همیشه کوتاهترین راه بهترین راه برای رسیدن به هدف نیست. به عنوان مثال هنگام استفاده از ابزار آماده برای دسترسی به داده ها اگر چه عملیات برنامه نویسی کاهش یافت ، ولی مشکل اینجا بود که در هنگام کار با داده ها زمان واکشی و نمایش داده ها به میزان قابل توجهی افزایش یاقته بود و این مساله در طولانی مدت باعث برخی مشکلات خواهد شد.

**پیشنهادات**

متاسفانه در ایران به علت عدم وجود قانون کپی رایت ، نمی توان در این زمینه پیشنهادات چندانی را ارائه نمود. لیکن برای بهتر شدن کار ، برخی مسائل به شرح ذیل ارائه می گردد که در صورت ارائه باید قبلا زیرساختهای آن به نحو احسن پیاده سازی شده باشد:

1. برنامه برای برنامه نویس چه سودی می تواند داشته باشد
2. آیا پروژه می تواند دارای سرمایه گذار باشد یا نه
3. آیا با توجه نبود قانون برای جلوگیری از دزدی نرم افزار می تواند قفل مناسبی برای نرم افزار تهیه نمود
4. برای برنامه در ابتدا طرح و نقشه تهیه نمود (تجزیه و تحلیل اساسی)
5. در مشتریان ایجاد نیاز نمود
6. به مشتری مزایای این گونه طرح ها را گوشزد نمود
7. حدالامکان پروژه با تکنولوژیهای مختلف ساگاری داشته باشد
8. به گونه ای برای جلوگیری از دزدی نرم افزار هزینه نرم افزار در فروش به تعداد بالا گرفته شود و نه در فروش به صورت تکی و آن هم به یک مشتری خاص
9. افرادی که قرار است بر روی پروژه کار کنند قبلا مشخص شده باشند و از افراد قابل اطمینان باشند
10. برنامه نویسان در ابعادی کوچکتر از ذینفع باشند. چراکه این مساله خود موجب دلخوشی و افزایش بازدهی کارآنها خواهد شد.

**منابع**

**کتابها**

1. بولتن آموزشی برنامه تویسی نوشته آقای مهندس اسدی - انتشارات امیر کبیر - چاپ سوم - بهار 86
2. کتاب آموزش سی شارپ نوشته مهندس شایان- انتشارات جمیل - چاپ پنجم - تابستان 90
3. کتاب طراحی پایگاه داده نوشته مهندس رانکوهی - انتشارات جهاد دانشگاهی - چاپ چهارم - بهار 83
4. کتاب طراحی پایگاه داده ها نوشته مهندس ساها کیان - انتشارات جهاد دانشگاهی - چاپ سوم - سال 88
5. کتاب آموزش C# نوشته مهندس زاهد - انتشارات توحید - چاپ سوم - سال 88

**وب سایتها**

http://support.microsoft.com

http://www.codeproject.com

http://www.barnamenevis.org

http://www. www.w3schools.com

http://forum.asp.net

http://forum.persiantools.com

http://web.shaar.com

http://www.farsiebook.com

http://www.zytrax.com

http://www.regular-expressions.info

http://www.grymoire.com

http://www.developer.apple.com