A

PROJECT REPORT

ON

"Railway Reservation System"

[Submitted under partial fulfillment of Three years full time Bachelor of Computer Application (BCA)]

By-Ritik Raj

Roll No- 192750200053

Registration no.-16201804708

BCA Final Year(2018-2021)

(Maulana Mazharul Haque Arabic and Persian University, Patna)

DR. ZAKIR HUSAIN INSTITUTE, BHAGALPUR

CETIFICATE OF APPROVAL

This is to certify that this is a benefited record of the Project Report entitled" RailwayReservationSystem" IS done satisfactorily by **Ritik Raj ,Roll NO. 192750200053**

,BCA6th Semester in a partial fulfillment of BCA Bachelor degree Course for the year 2021.This report has not been submitted for any other examination and does not from part any other Course undergone by the Candidate.

Suman kr. singh

(CENTRE CO-ORDINATOR)

DECLARATION

The author of this Project, Ritik Raj , Roll No.----------,Registration No.----------BCA Part 3**,** declares that the Work reported in this **Project has been entirely done by under the Supervision of MR.GOPESH KUMAR .**

No part of this work has been Submitted in part or full in any other University.

Ritik Raj

Roll No.----------

Registration No.----------

BCA Part-3

CERTIFICATE OF ORIGINALITY

This is to Certify that the Project Report Entitled " Railway Reservation System "Project in to Submitted to Maulana Mazharul Haque Arabic & Persian University, patna is a partial fulfillment of the requirement for the award of the Degree of **BCA** is a Original Work carried out by **Ritik Raj, Roll No.----------**

---------- , BCA part-3, under the guidance of Gaurav KUMAR.

The matter embodied in this Project is a genuine work done by the student and has not been submitted whether to this University or Institute for the Fulfillment of any course of Study.

Signature of Student

ACKNOWLEDGEMENT

I express my sincere and profound gratitude to my Centre Head **Suman Kumar Singh** and teacher and guide**,** Gaurav KUMAR. He has been a constant source of inspiration throughout the trials and tribulations which went into the work up of this project.I will ever remain indebted to him. I find no words to him for this teachings.

With all sincerity,I Wish to put record my gratitude to my teachers, **Gaurav Sir, Gopesh Sir ,Rupak sir ,etc.** for providing me with guidance, full Co-operation, encouragement and suggestions during the course of study.

Ritik Raj

Roll No.----------

Registration No.----------

About Project:

This Train management system project is developed using VB.NET. Here at first, the user has to pass through login system (User and Admin) to get access then, from the user's login he/she can view train schedules and book tickets. Admin has all the control of the system, therefore from the admin login he/she can add, delete, view and update train information and view customer registrations. While adding train information, the user has to enter train id number, train name, source, destination, and seat availability. He/she can also register new users to handle the system. All the bookings, as well as train details, can be

managed through this system and it is not time-consuming. This project is easy to operate and understood by the users.

Development Tool Used in this Project:

1. PC configuration :

I. **Device name DESKTOP-VNH6BFU**

II. **Processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz**

III. **Installed RAM 8.00 GB (7.88 GB usable)**

IV. **Device ID 4ACAFDF1-EAA1-44C1-B994-CEA50D32799D**

V. **Product ID 00326-30000-00001-AA561**

VI. **System type 64-bit operating system, x64-based processor**

Pen and touch No pen or touch input is available for this display

2. Windows Specifiction:

- **Edition Windows 10 Home**
- **Version 21H1**
- **Installed on 07-10-2020**
- **OS build 19043.1165**

Experience Windows Feature Experience Pack 120.2212.3530.0

Microsoft Visual Studio:

- Microsoft Visual Studio Enterprise 2015
- Version 14.0.23107.0 D14REL
- Microsoft .NET Framework
- Version 4.8.04084

- Installed Version: Enterprise

- Architecture and Modeling Tools 00322-90150-00969-AA662
- Microsoft Architecture and Modeling Tools

- UML® and Unified Modeling Language™ are trademarks or registered trademarks of the Object Management Group, Inc. in the United States and other countries.

- Visual Basic 2015 00322-90150-00969-AA662
- Microsoft Visual Basic 2015

- Visual C# 2015 00322-90150-00969-AA662
- Microsoft Visual C# 2015

- Visual C++ 2015 00322-90150-00969-AA662
- Microsoft Visual C++ 2015

- Application Insights Tools for Visual Studio Package 1.0
- Application Insights Tools for Visual Studio

- ASP.NET and Web Tools 14.0.20626.0
- ASP.NET and Web Tools

- ASP.NET Web Frameworks and Tools 2013 5.2.30624.0
- For additional information, visit http://www.asp.net/

- Common Azure Tools 1.5
- Provides common services for use by Azure Mobile Services and Microsoft Azure Tools.
- GenerateUnitTest 1.0
- Generates unit test code for methods in classes under test.
- Microsoft Azure Mobile Services Tools 1.4
- Microsoft Azure Mobile Services Tools

- Microsoft Code Digger 0.9
- Microsoft Code Digger

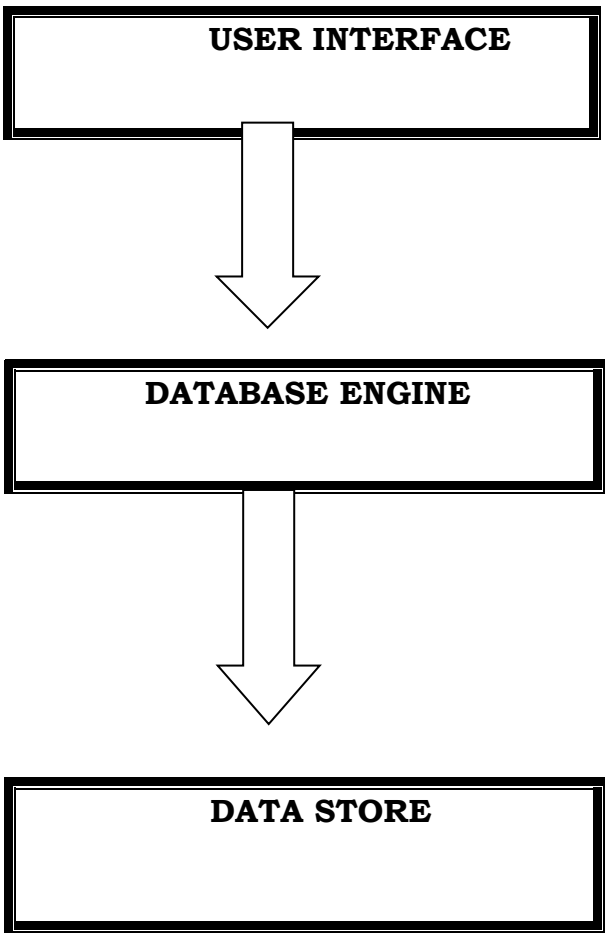
- Microsoft.Pex.VisualStudio 1.0
- Pex
- NuGet Package Manager 3.0.0
- NuGet Package Manager in Visual Studio. For more information about NuGet, visit http://docs.nuget.org/.
- PreEmptive Analytics Visualizer 1.2
- Microsoft Visual Studio extension to visualize aggregated summaries from the PreEmptive Analytics product.
- SQL Server Data Tools 14.0.50616.0
- Microsoft SQL Server Data Tools

TOOLS USED:-

- VB.NET for front-end.
- Microsoft jet OLEDB for connectivity.
- ACCESS for back-end.

VB.NET DATABASE ARCTITECTURE

A VB.NET database application consists of the following three components, as shown in the figure below:



USER INTERFACE: This is what the user interacts with. It contains the forms that display the data enables the user to view or update it. It also includes various data access techniques for database services like adding or deleting records and performing queries.

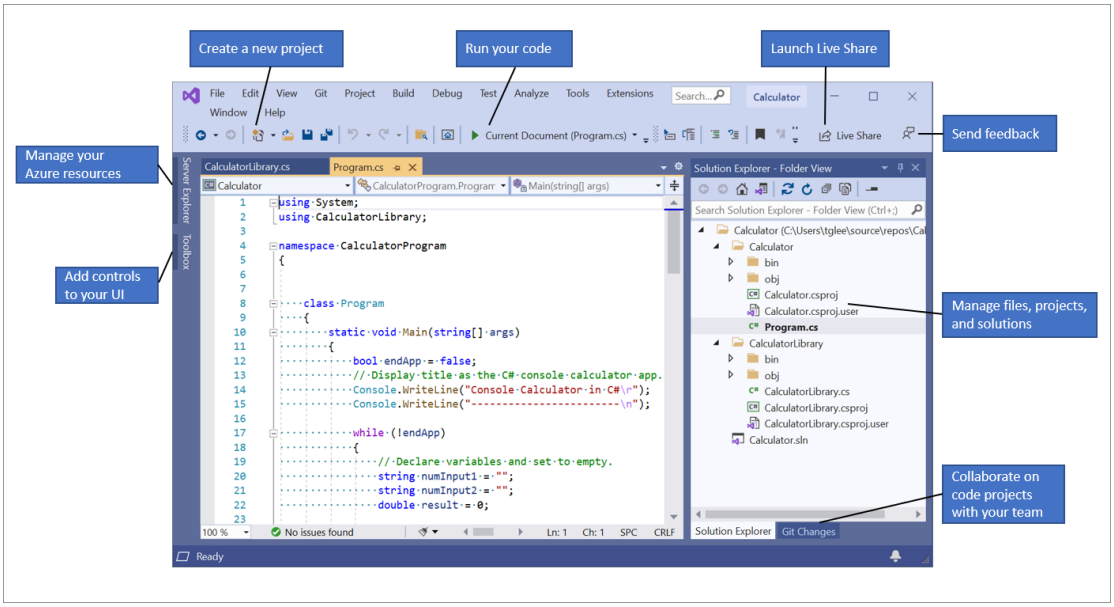
DATABASE ENGINE: It is contained in asset of Dynamic Link (DLL) files and is linked with the VB>NET program at run time. The engine is responsible for reading, writing and modifying the database. It also handles indexing, security and referential integrity issues in the database. It contains a query processor to handle SQL quires. THE database engine is logically placed between the program and the database files.

DATA STORES: It is the set of files containing the database tables. For example Microsoft Access has .mdb files containing several tables. A data is said to be passive because it does not world on the data on its own.

About Microsoft Visual Studio .NET: -

INTRODUCTION

The Visual Studio *integrated development environment* is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. An integrated development environment (IDE) is a feature-rich program that can be used for many aspects of software development. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process.



This image shows Visual Studio with an open project and several key tool windows you'll likely use:

- Solution Explorer (top right) lets you view, navigate, and manage your code files. **Solution Explorer** can help organize your code by grouping the files into solutions and projects.
- The editor window (center), where you'll likely spend a majority of your time, displays file contents. This is where you can edit code or design a user interface such as a window with buttons and text boxes.

- Git Changes (bottom right) lets you track work items and share code with others using version control technologies such as Git and GitHub.

Editions

Visual Studio is available for Windows and Mac. Visual Studio for Mac has many of the same features as Visual Studio 2019, and is optimized for developing cross-platform and mobile apps. This article focuses on the Windows version of Visual Studio 2019.

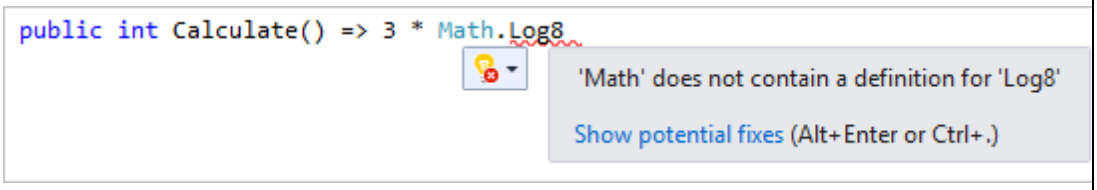
There are three editions of Visual Studio 2019: Community, Professional, and Enterprise. See Compare Visual Studio editions to learn about which features are supported in each edition.

Popular productivity features

Some of the popular features in Visual Studio that help you to be more productive as you develop software include:

- Squiggles and Quick Actions

  Squiggles are wavy underlines that alert you to errors or potential problems in your code as you type. These visual clues enable you to fix problems immediately without waiting for the error to be discovered during build or when you run the program. If you hover over a squiggle, you see additional information about the error. A light bulb may also appear in the left margin with actions, known as Quick Actions, to fix the error.

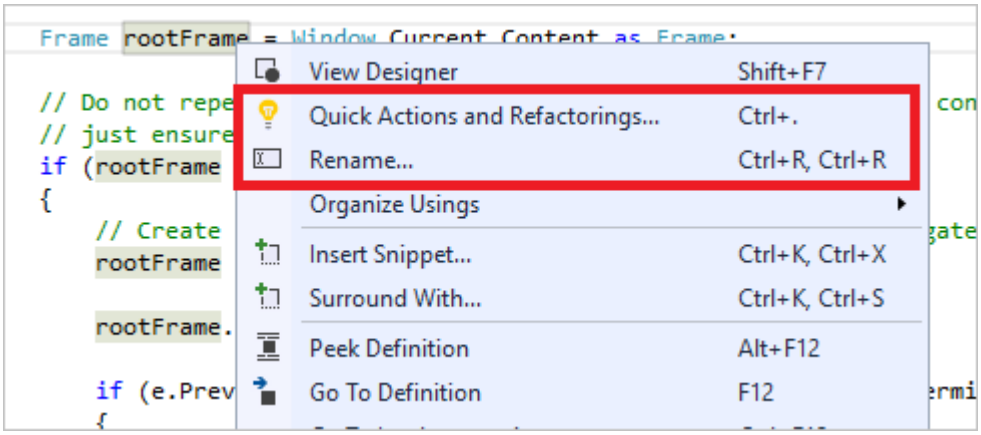


- Code Cleanup

  With the click of a button, format your code and apply any code fixes suggested by your code style settings, .editorconfig conventions, and Roslyn analyzers. **Code Cleanup** helps you resolve issues in your code before it goes to code review. (Currently available for C# code only.)

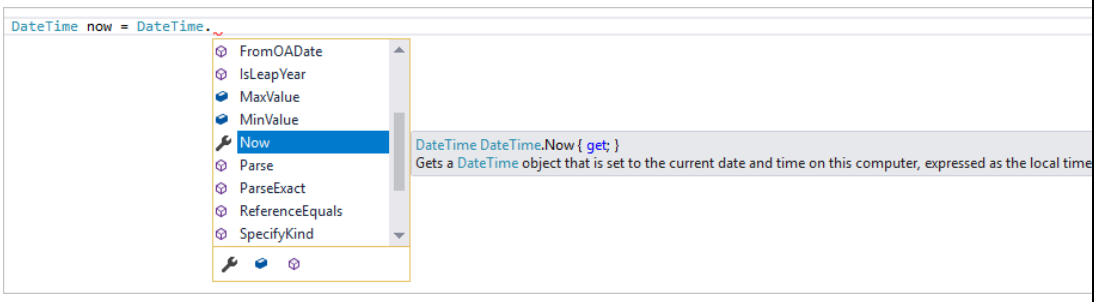
- Refactoring

  Refactoring includes operations such as intelligent renaming of variables, extracting one or more lines of code into a new method, changing the order of method parameters, and more.



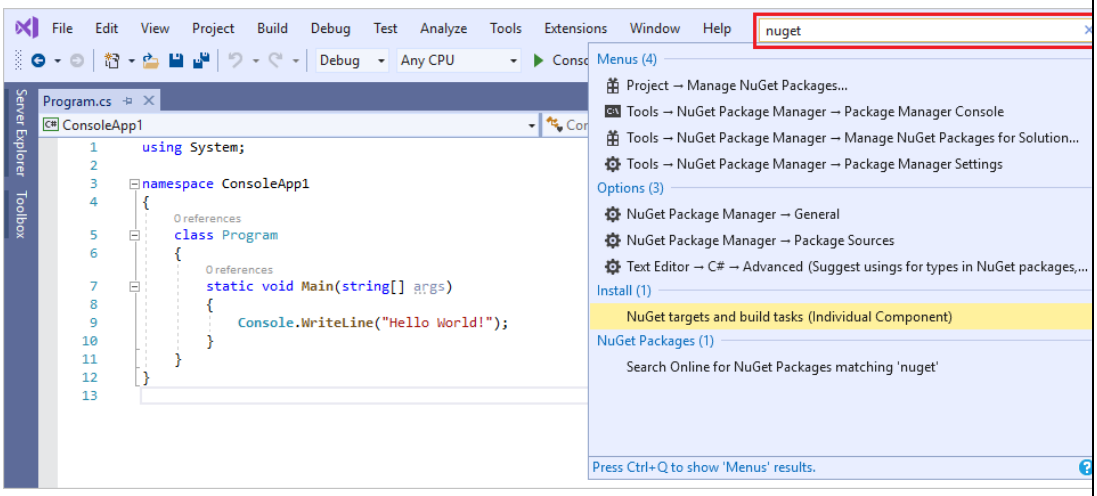
- IntelliSense

  IntelliSense is a term for a set of features that displays information about your code directly in the editor and, in some cases, write small bits of code for you. It's like having basic documentation inline in the editor, which saves you from having to look up type information elsewhere. IntelliSense features vary by language. For more information, see C# IntelliSense, Visual C++ IntelliSense, JavaScript IntelliSense, and Visual Basic IntelliSense. The following illustration shows how IntelliSense displays a member list for a type:



- Visual Studio search

  Visual Studio can seem overwhelming at times with so many menus, options, and properties. Visual Studio search (**Ctrl**+**Q**) is a great way to rapidly find IDE features and code in one place.



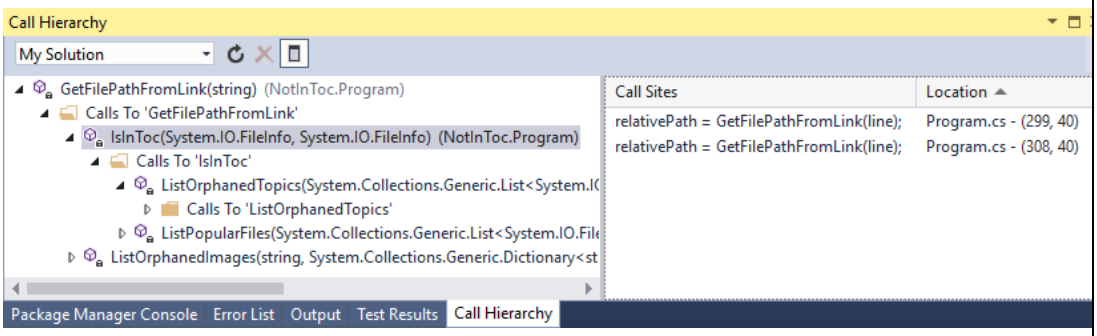
For information and productivity tips, see How to use Visual Studio search.

- Live Share

  Collaboratively edit and debug with others in real time, regardless of what your app type or programming language. You can instantly and securely share your project and, as needed, debugging sessions, terminal instances, localhost web apps, voice calls, and more.

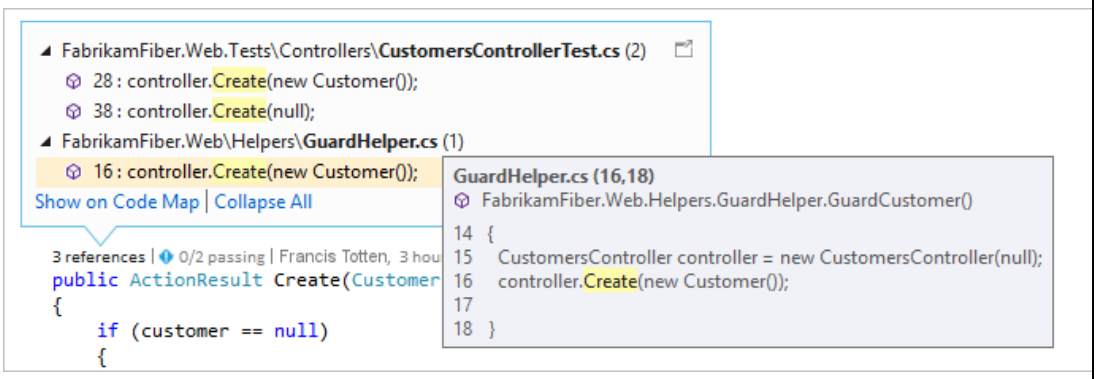
- Call Hierarchy

  The **Call Hierarchy** window shows the methods that call a selected method. This can be useful information when you're thinking about changing or removing the method, or when you're trying to track down a bug.



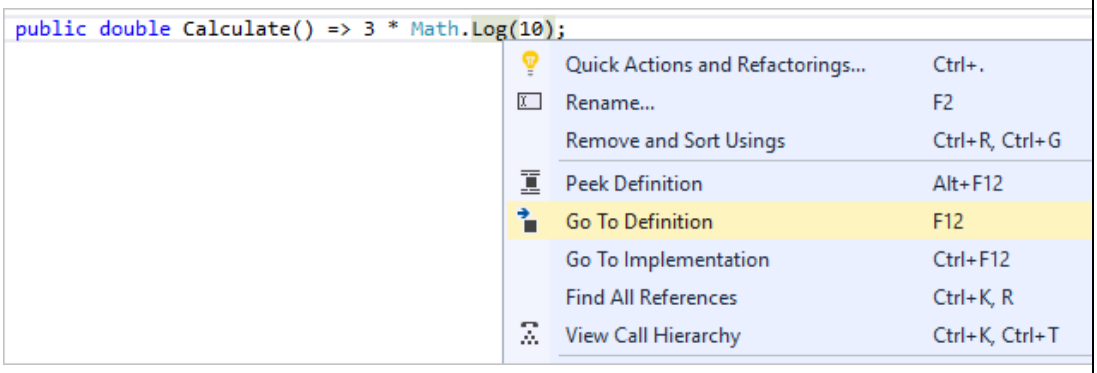
- CodeLens

  CodeLens helps you find references to your code, changes to your code, linked bugs, work items, code reviews, and unit tests, all without leaving the editor.



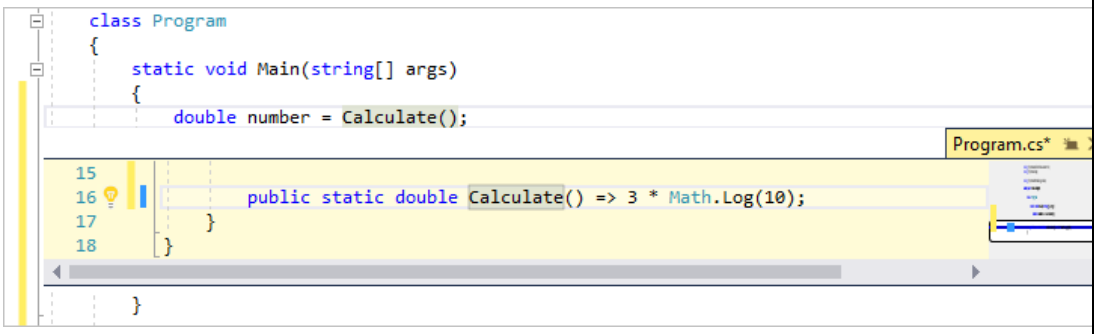
- Go To Definition

  The Go To Definition feature takes you directly to the location where a function or type is defined.



- Peek Definition

  The **Peek Definition** window shows the definition of a method or type without actually opening a separate file.

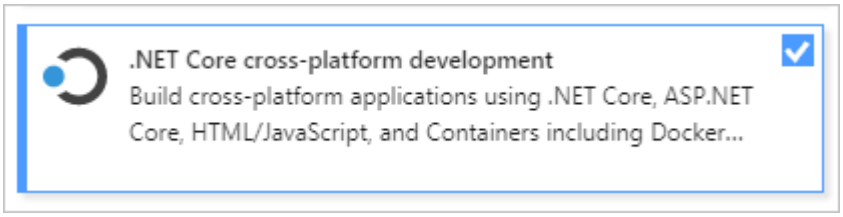


Install the Visual Studio IDE

In this section, you'll create a simple project to try out some of the things you can do with Visual Studio. You'll use IntelliSense as a coding aid, debug an app to see the value of a variable during the program's execution, and change the color theme.

To get started, download Visual Studio and install it on your system. The modular installer enables you to choose and install *workloads*, which are groups of features

needed for the programming language or platform you prefer. To follow the steps for creating a program, be sure to select the **.NET Core cross-platform development** workload during installation.



When you open Visual Studio for the first time, you can optionally sign in using your Microsoft account or your work or school account.

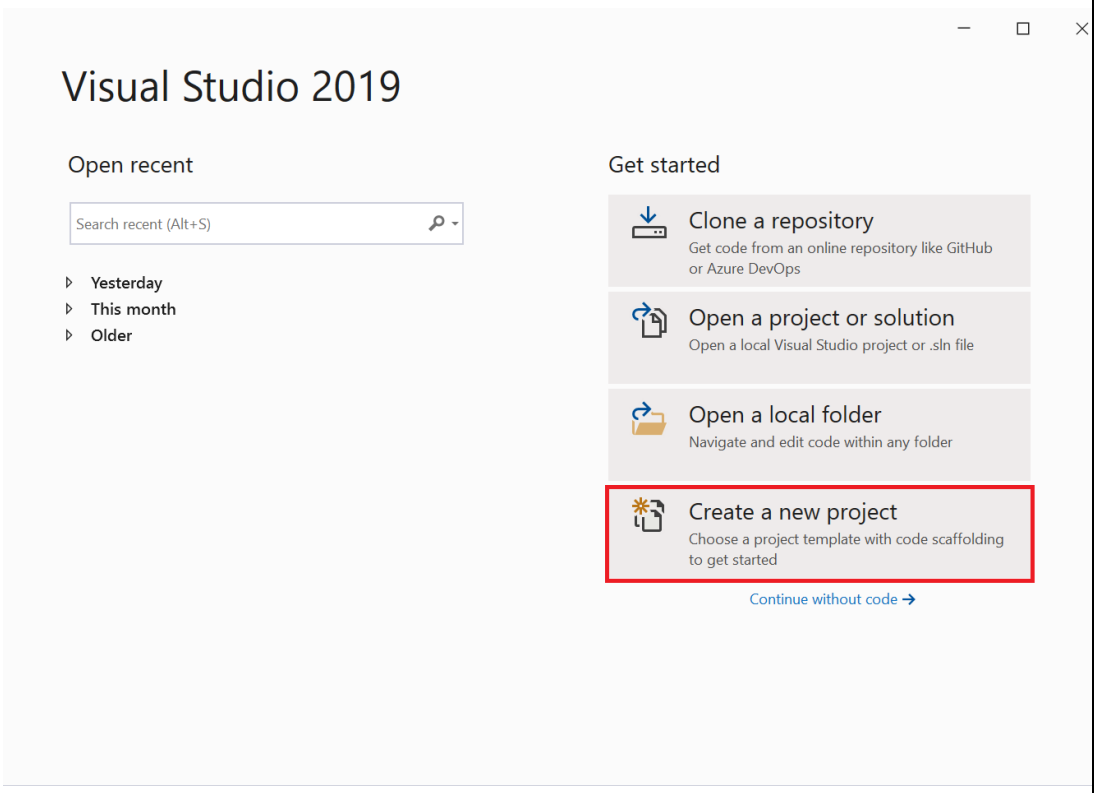
Create a program

Let's dive in and create a simple program.

1. Open Visual Studio.

   The start window appears with various options for cloning a repo, opening a recent project, or creating a brand new project.

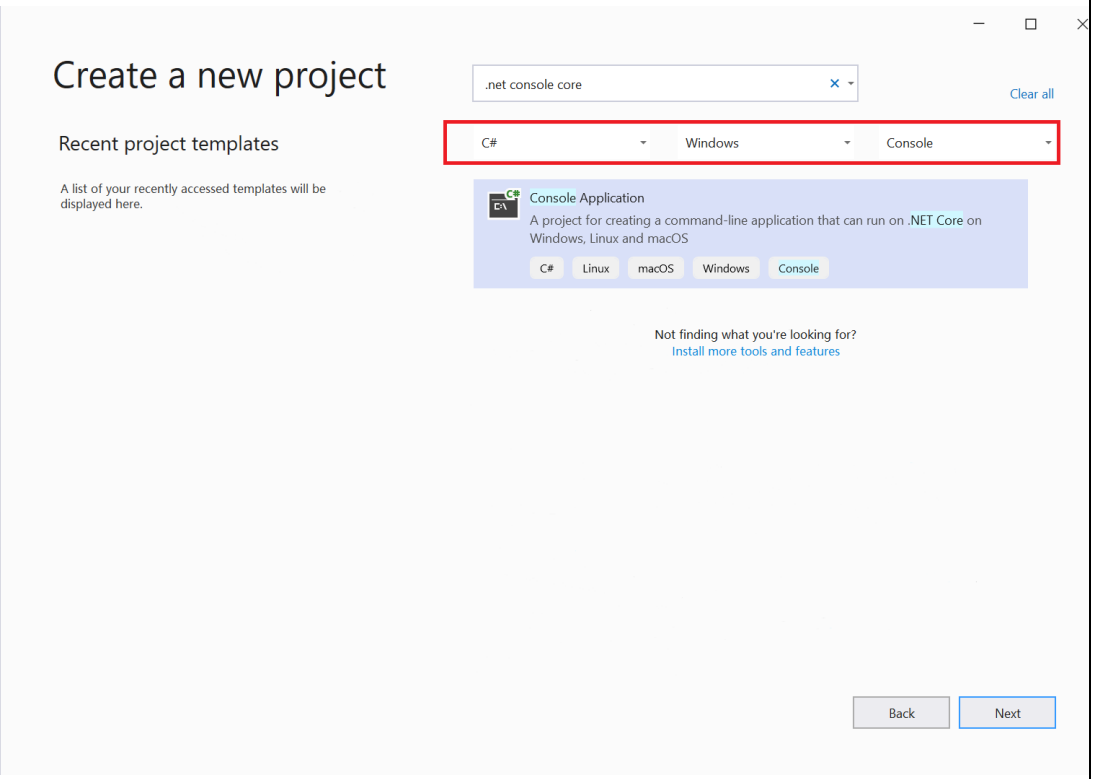
2. Choose **Create a new project**.



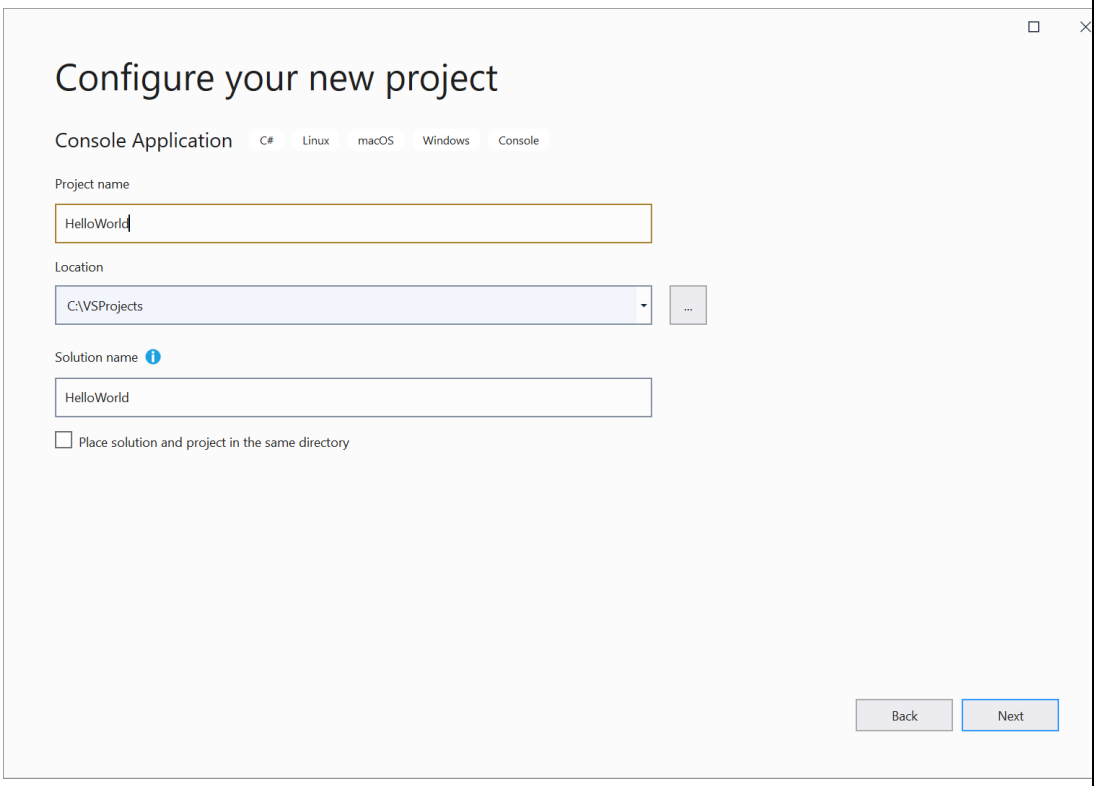
The **Create a new project** window opens and shows several project *templates*. A template contains the basic files and settings required for a given project type.

3. To find the template we want, type or enter **.net core console** in the search box. The list of available templates is automatically filtered based on the keywords you entered. You can further filter the template results by choosing **C#** from the **All language** drop-down list, **Windows** from the **All platforms** list, and **Console** from the **All project types** list .

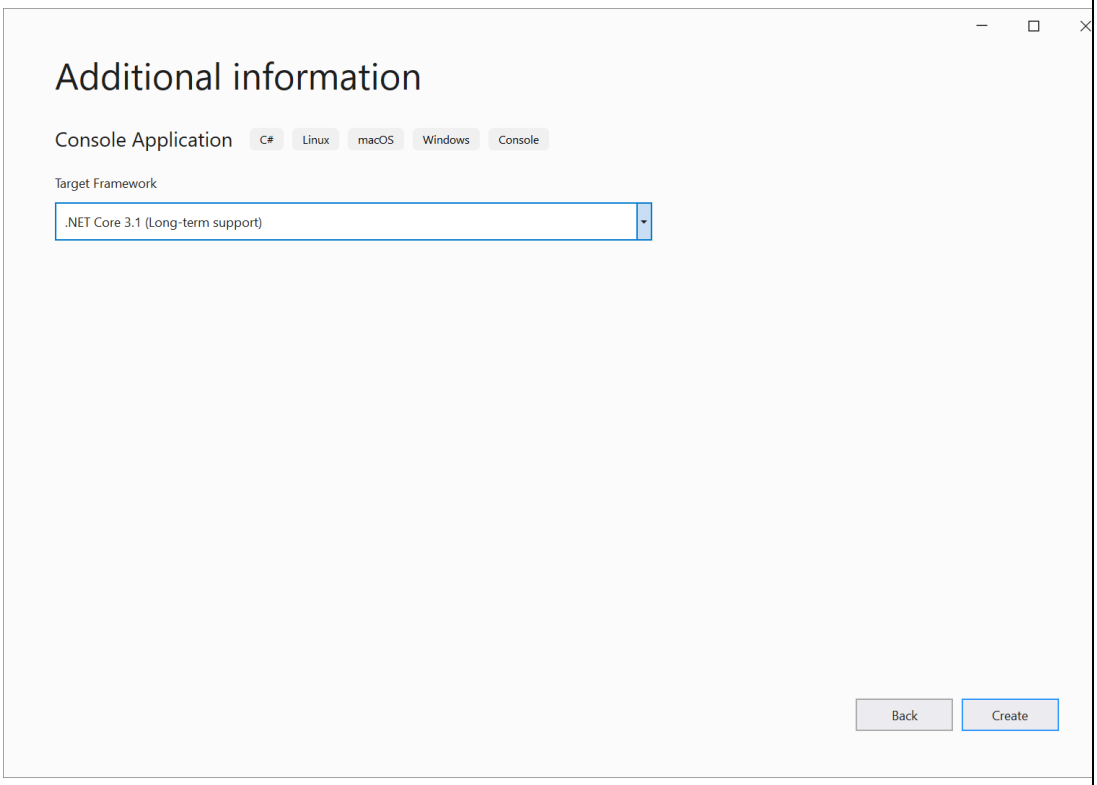
   Select the **Console Application** template, and then click **Next**.



4. In the **Configure your new project** window, enter **HelloWorld** in the **Project name** box, optionally change the directory location for your project files (the default locale is `C:\Users\<name>\source\repos`), and then click **Next**.

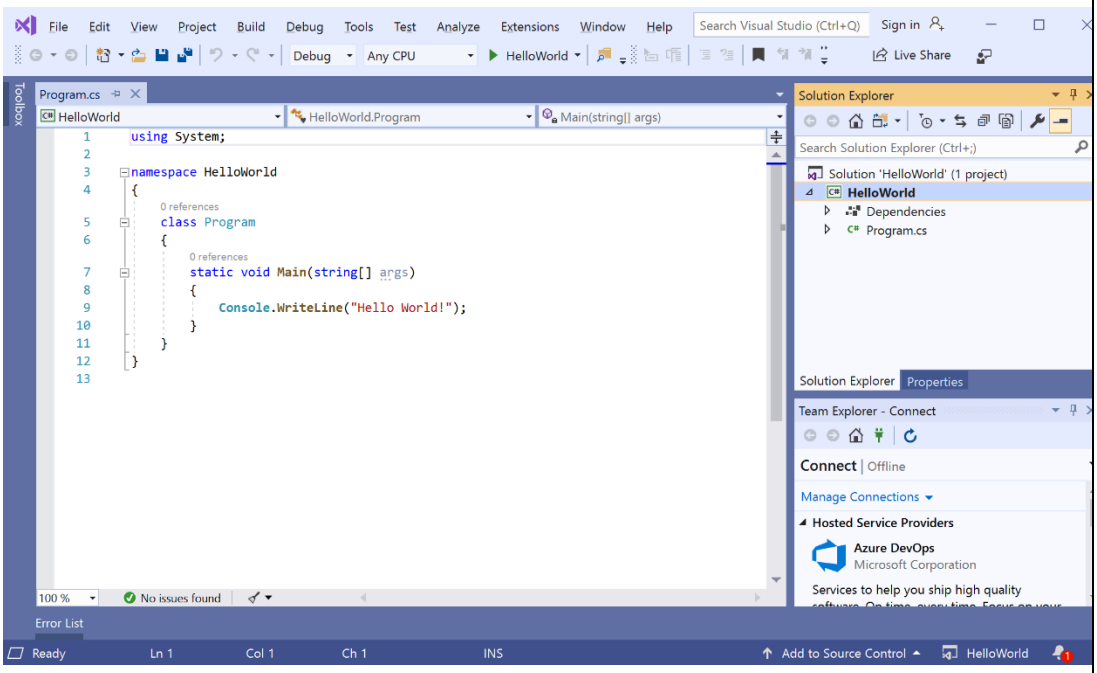


5. In the **Additional information** window, verify that **.NET Core 3.1** appears in the **Target Framework** drop-down menu, and then click **Create**.

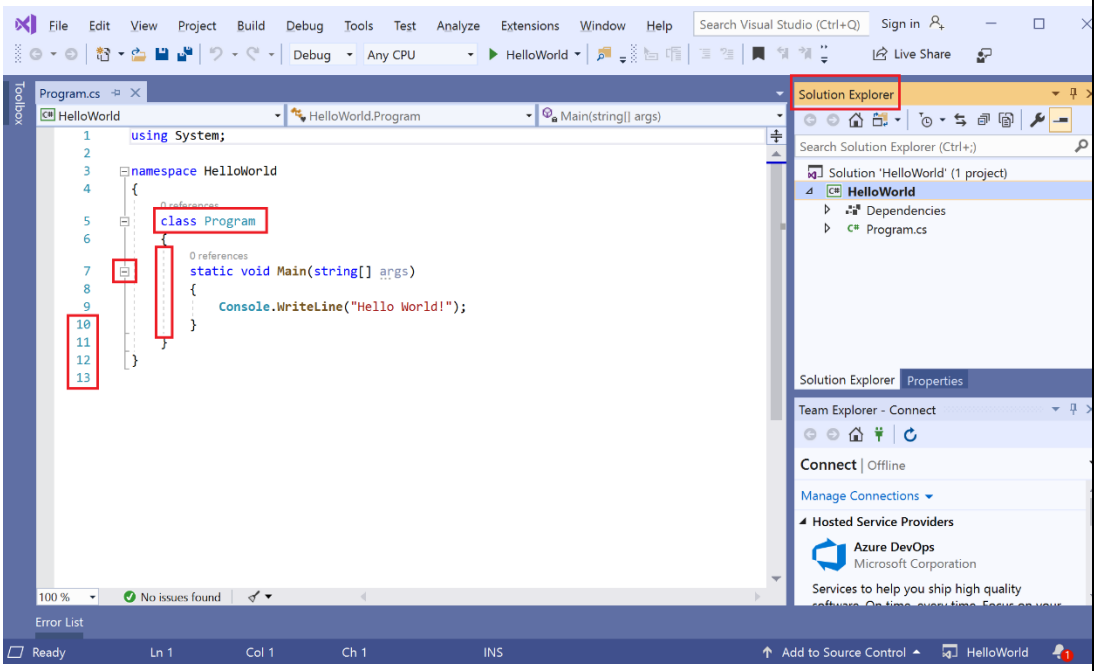


Visual Studio creates the project. It's a simple "Hello World" application that calls the Console.WriteLine() method to display the literal string "Hello World!" in the console (program output) window.

Shortly, you should see something like the following:

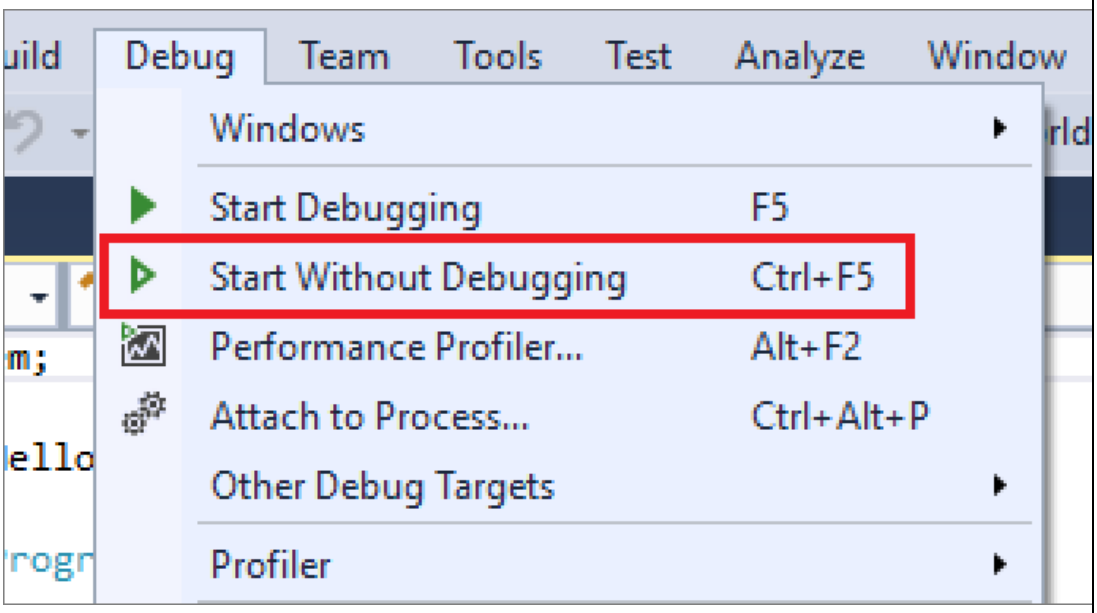


The C# code for your application shows in the editor window, which takes up most of the space. Notice that the text is automatically colorized to indicate different parts of the code, such as keywords and types. In addition, small, vertical dashed lines in the code indicate which braces match one another, and line numbers help you locate code later. You can choose the small, boxed minus signs to collapse or expand blocks of code. This code outlining feature lets you hide code you don't need, helping to minimize onscreen clutter. The project files are listed on the right side in a window called **Solution Explorer**.

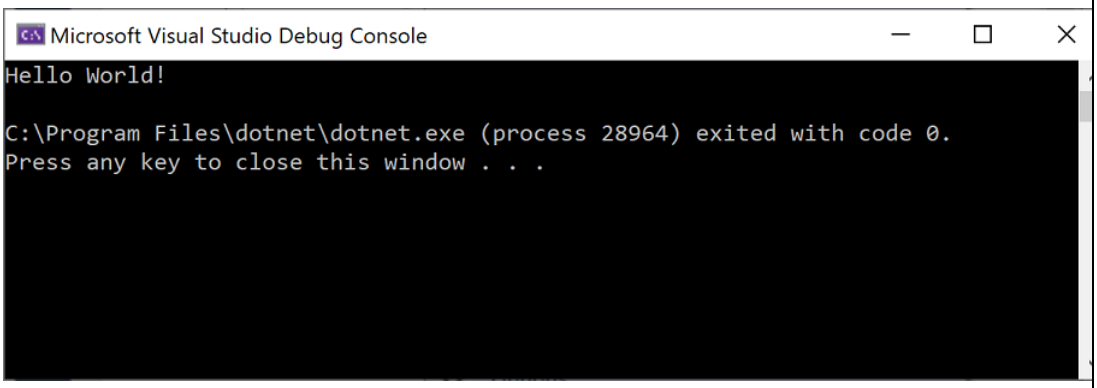


There are other menus and tool windows available, but let's move on for now.

6. Now, start the app. You can do this by choosing **Start Without Debugging** from the **Debug** menu on the menu bar. You can also press **Ctrl**+**F5**.



Visual Studio builds the app, and a console window opens with the message **Hello World!**. You now have a running app!



7. To close the console window, press any key on your keyboard.
8. Let's add some additional code to the app. Add the following C# code before the line that says `Console.WriteLine("Hello World!");`:

   C#Copy

   ```
   Console.WriteLine("\nWhat is your name?");
   var name = Console.ReadLine();
   ```

   This code displays **What is your name?** in the console window, and then waits until the user enters some text followed by the **Enter** key.

9. Change the line that says `Console.WriteLine("Hello World!");` to the following code:

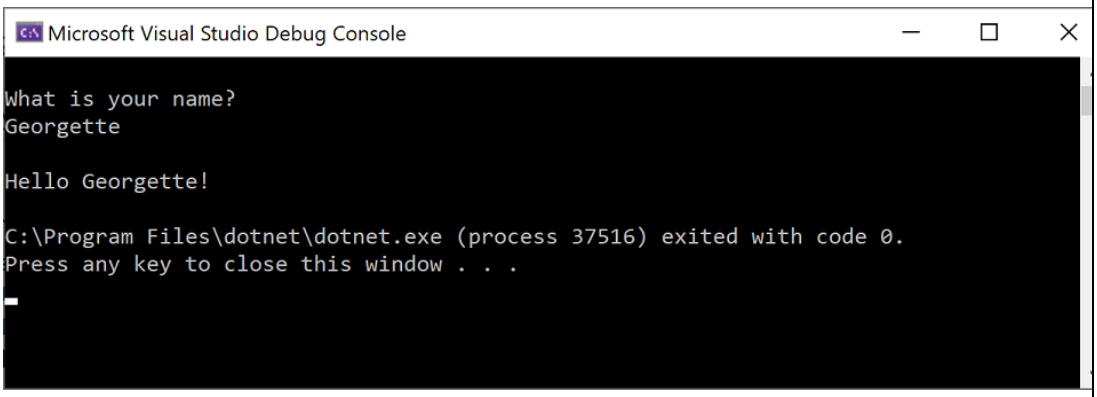
   C#Copy

   ```
   Console.WriteLine($"\nHello {name}!");
   ```

10. Run the app again by selecting **Debug** > **Start Without Debugging** or by pressing **Ctrl**+**F5**.

    Visual Studio rebuilds the app, and a console window opens and prompts you for your name.

11. Enter your name in the console window and press **Enter**.



12. Press any key to close the console window and stop the running program.

Use refactoring and IntelliSense

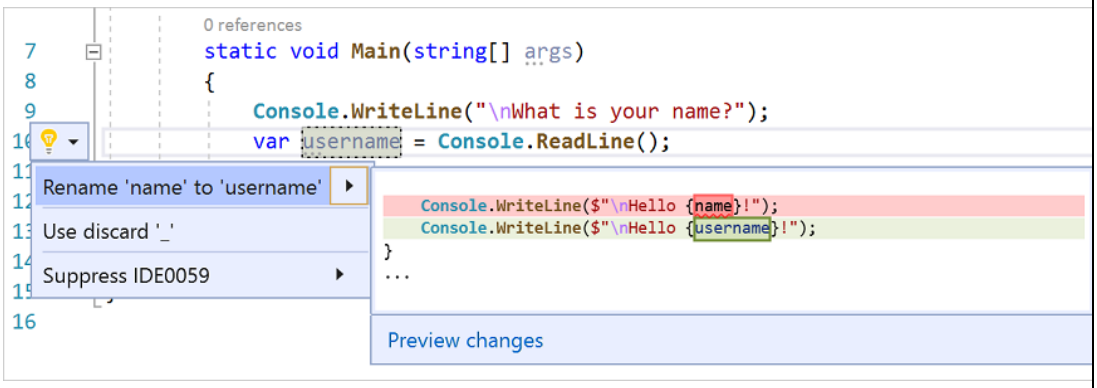
Let's look at a couple of the ways that refactoring and IntelliSense can help you code more efficiently.

First, let's rename the `name` variable:

1. Double-click the `name` variable to select it.
2. Type in the new name for the variable, **username**.

   Notice that a gray box appears around the variable, and a light bulb appears in the margin.

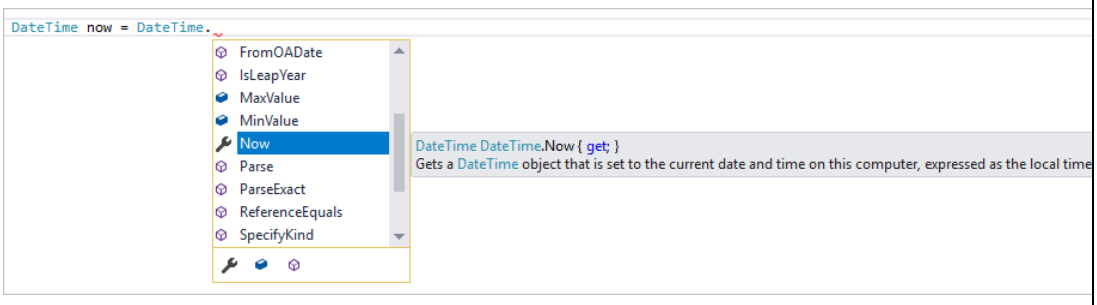
3. Select the light bulb icon to show the available Quick Actions.
   Select **Rename 'name' to 'username'**.



   The variable is renamed across the project, which in our case is only two places.

4. Now let's take a look at IntelliSense. Below the line that says `Console.WriteLine($"\nHello {username}!");`, type `DateTime now = DateTime.`.

   A box displays the members of the DateTime class. In addition, the description of the currently selected member displays in a separate box.



5. Select the member named **Now**, which is a property of the class, by double-clicking on it or pressing **Tab**. Complete the line of code by adding a semi-colon to the end.
6. Below that, type in or paste the following lines of code:

```
C#Copy

int dayOfYear = now.DayOfYear;

Console.Write("Day of year: ");
Console.WriteLine(dayOfYear);
```

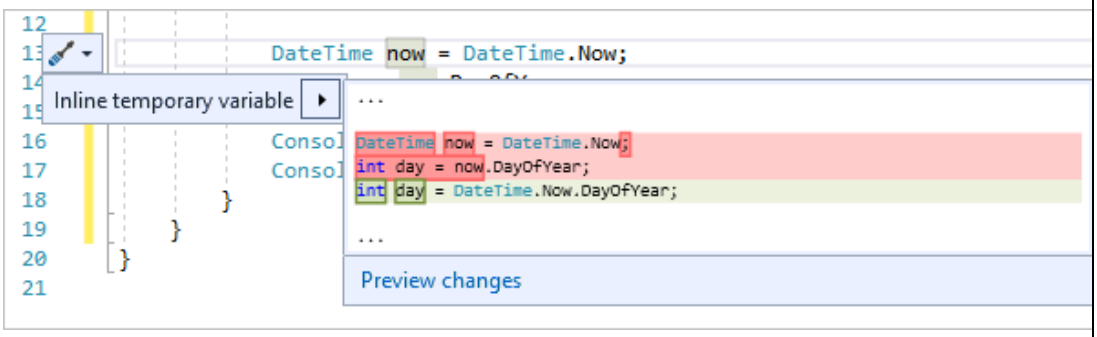
Tip

Console.Write is a little different to **Console.WriteLine** in that it doesn't add a line terminator after it prints. That means that the next piece of text that's sent to the output will print on the same line. You can hover over each of these methods in your code to see their description.

7. Next, we'll use refactoring again to make the code a little more concise. Click on the variable `now` in the line `DateTime now = DateTime.Now;`.

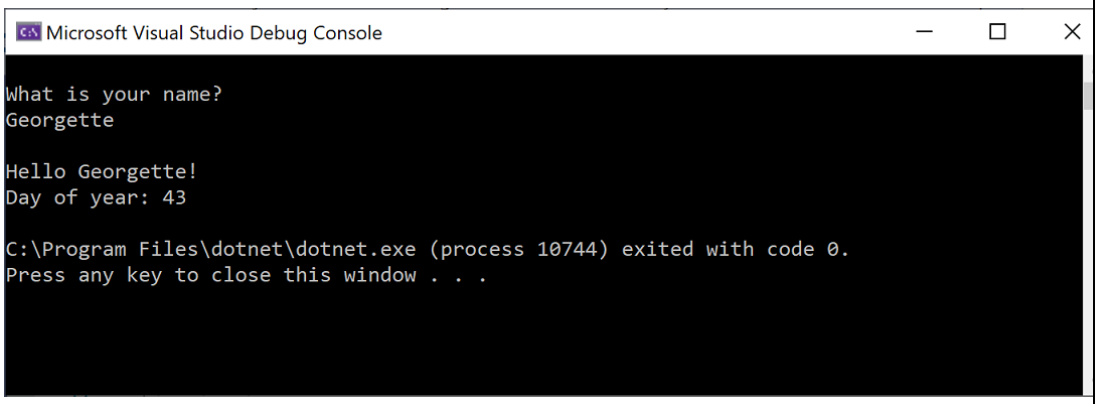
   Notice that a little screwdriver icon appears in the margin on that line.

8. Click the screwdriver icon to see what suggestions Visual Studio has available. In this case, it's showing the Inline temporary variable refactoring to remove a line of code without changing the overall behavior of the code:



9. Click **Inline temporary variable** to refactor the code.

10. Run the program again by pressing **Ctrl**+**F5**. The output looks something like this:



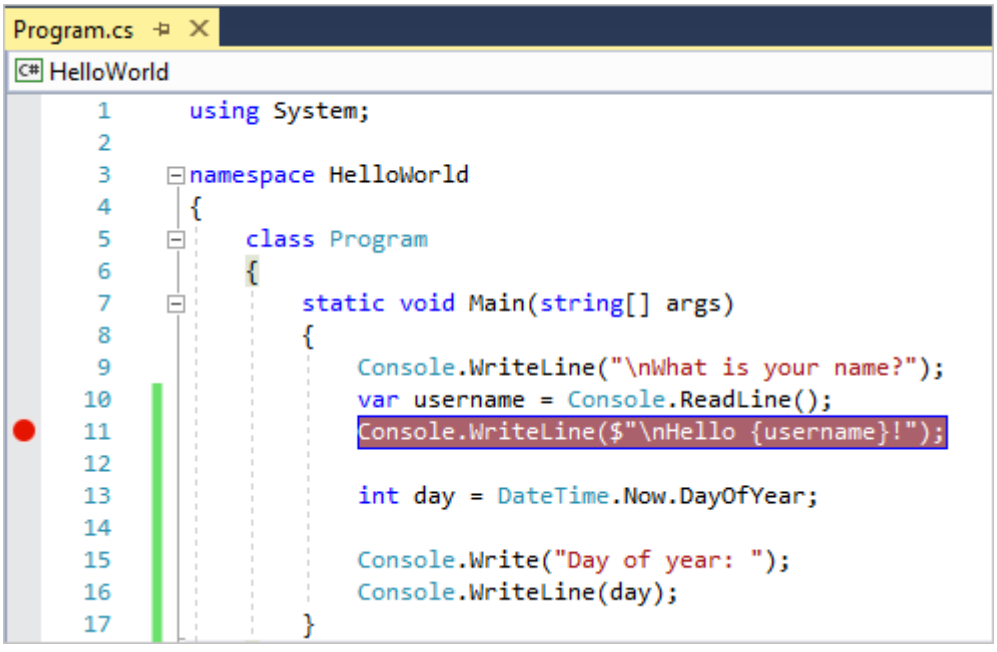
Debug code

When you write code, you need to run it and test it for bugs. Visual Studio's debugging system lets you step through code one statement at a time and inspect variables as you go. You can set *breakpoints* that stop execution of the code at a particular line. You can observe how the value of a variable changes as the code runs, and more.

Let's set a breakpoint to see the value of the `username` variable while the program is "in flight".

1. Find the line of code that says `Console.WriteLine($"\nHello {username}!");`. To set a breakpoint on this line of code, that is, to make the program pause execution at this line, click in the far left margin of the editor. You can also click anywhere on the line of code and then press **F9**.

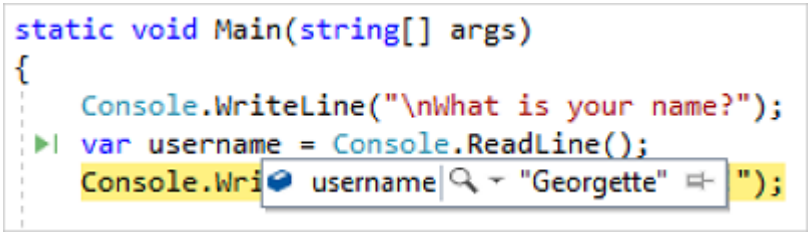
   A red circle appears in the far left margin, and the code is highlighted in red.



2. Start debugging by selecting **Debug** > **Start Debugging** or by pressing **F5**.
3. When the console window appears and asks for your name, type it in and press **Enter**.

   The focus returns to the Visual Studio code editor and the line of code with the breakpoint is highlighted in yellow. This signifies that it's the next line of code that the program will execute.

4. Hover your mouse over the `username` variable to see its value. Alternatively, you can right-click on `username` and select **Add Watch** to add the variable to the **Watch** window, where you can also see its value.



5. To let the program run to completion, press **F5** again.

To get more details about debugging in Visual Studio, see Debugger feature tour.

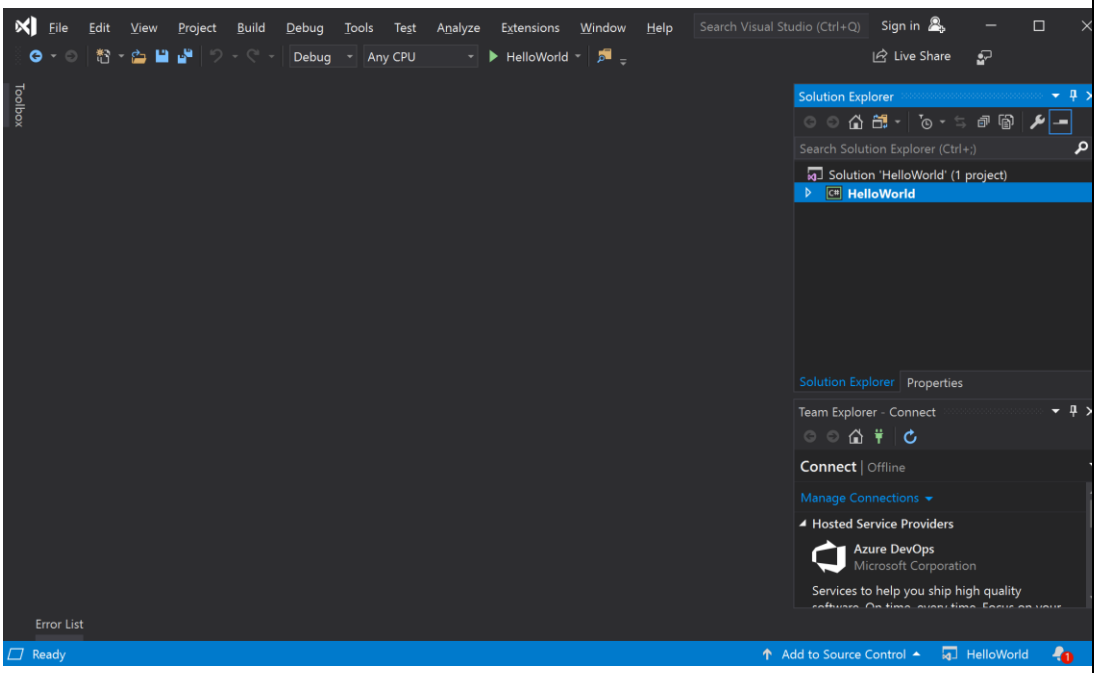
Customize Visual Studio

You can personalize the Visual Studio user interface, including change the default color theme. To change to the **Dark** theme:

1. On the menu bar, choose **Tools** > **Options** to open the **Options** dialog.

2. On the **Environment** > **General** options page, change the **Color theme** selection to **Dark**, and then choose **OK**.

   The color theme for the entire IDE changes to **Dark**.



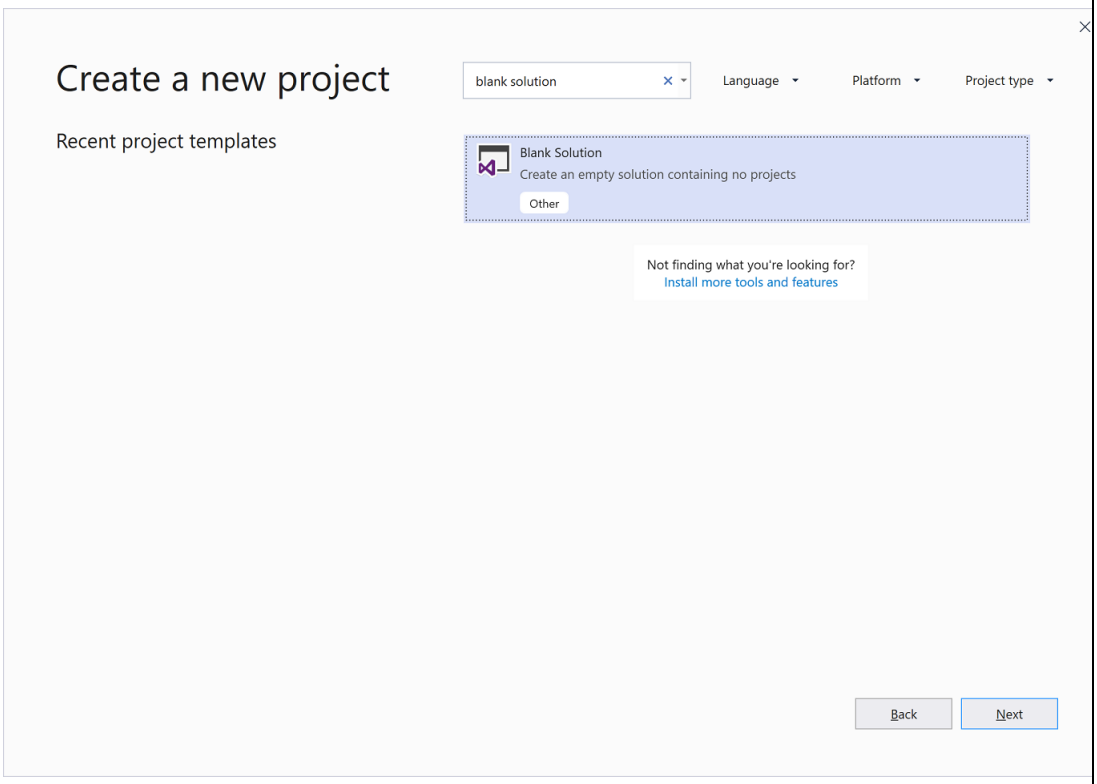
Solutions and projects

Despite its name, a solution is not an "answer". A solution is simply a container used by Visual Studio to organize one or more related projects. When you open a solution in Visual Studio, it automatically loads all the projects that the solution contains.

Create a solution

We'll start our exploration by creating an empty solution. After you get to know Visual Studio, you probably won't find yourself creating empty solutions very often. When you create a new project, Visual Studio automatically creates a solution to house the project if there's not a solution already open.

1. Open Visual Studio.
2. On the start window, select **Create a new project**.
3. On the **Create a new project** page, enter **blank solution** into the search box, select the **Blank Solution** template, and then select **Next**.



Tip

If you have several workloads installed, the **Blank Solution** template might not appear at the top of your list of search results. Try scrolling to the **Other results based on your search** section of the list. It should appear there.

4. Name the solution **QuickSolution**, and then select **Create**.

   A solution appears in **Solution Explorer** on the right-hand side of the Visual Studio window. You'll probably use **Solution Explorer** often, to browse the contents of your projects.

Add a project

Now let's add our first project to the solution. We'll start with an empty project and add the items we need to the project.

1. From the right-click or context menu of **Solution 'QuickSolution'** in **Solution Explorer**, select **Add** > **New Project**.

   A dialog box opens that says **Add a new project**.

2. Enter the text **empty** into the search box at the top, and then select **C#** under **Language**.

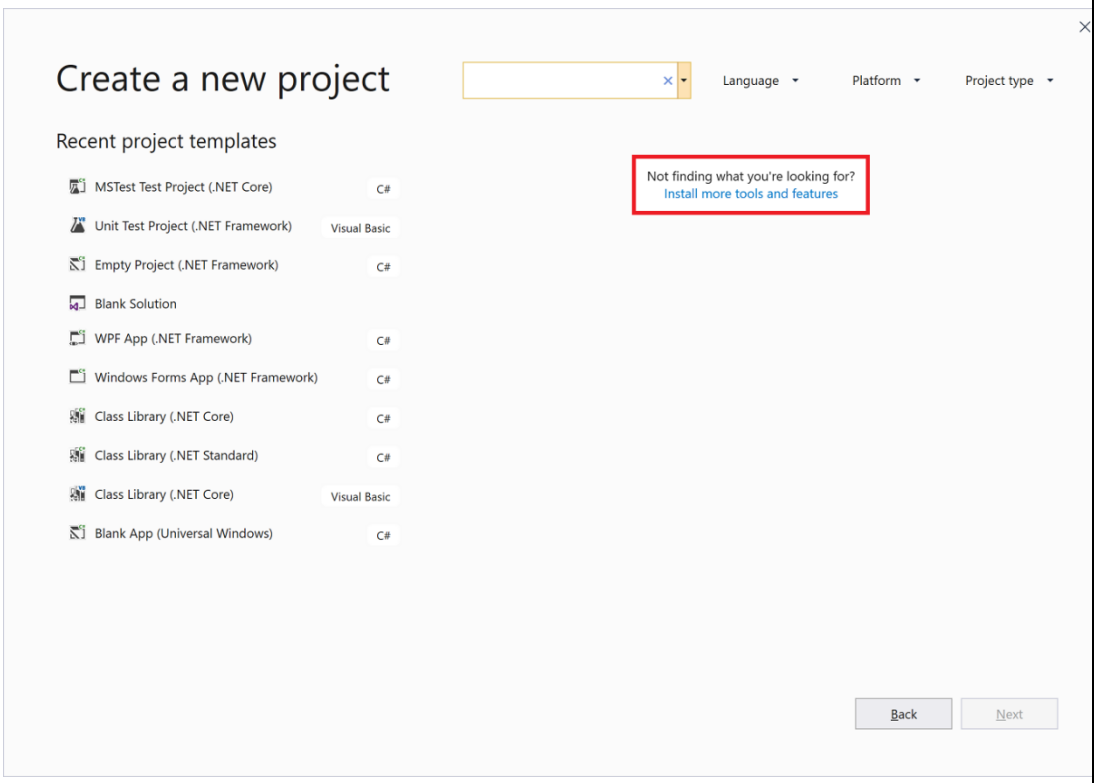
3. Select the **Empty Project (.NET Framework)** template, and then select **Next**.
4. Name the project **QuickDate**, then select **Create**.

   A project named QuickDate appears beneath the solution in **Solution Explorer**. Currently it contains a single file called *App.config*.

   Note

   If you don't see the **Empty Project (.NET Framework)** template, you must install the **.NET desktop development** Visual Studio workload. Visual Studio uses workload-based installation to install only the components you need for the type of development you do.

   An easy way to install a new workload when you're creating a new project is to select the **Install more tools and features** link under the text that says **Not finding what you're looking for?**. After Visual Studio Installer launches, select the **.NET desktop development** workload and then the **Modify** button.



Add an item to the project

We have an empty project. Let's add a code file.

1. From the right-click or context menu of the **QuickDate** project in **Solution Explorer**, select **Add** > **New Item**.

   The **Add New Item** dialog box opens.

2. Expand **Visual C# Items**, then select **Code**. In the middle pane, select the **Class** item template. Name the class **Calendar**, and then select the **Add** button.

   A file named *Calendar.cs* is added to the project. The *.cs* on the end is the file extension that is given to C# code files. The file appears in the visual project hierarchy in **Solution Explorer**, and its contents are opened in the editor.

3. Replace the contents of the *Calendar.cs* file with the following code:

   C#Copy

   ```
   using System;

   namespace QuickDate
   {
       internal class Calendar
       {
           static void Main(string[] args)
           {
               DateTime now = GetCurrentDate();
               Console.WriteLine($"Today's date is {now}");
               Console.ReadLine();
           }

           internal static DateTime GetCurrentDate()
           {
               return DateTime.Now.Date;
           }
       }
   }
   ```

   You don't need to understand what the code does, but if you want, you can run the program by pressing **Ctrl**+**F5** and see that it prints today's date to the console (or standard output) window.

Add a second project

It is common for solutions to contain more than one project, and often these projects reference each other. Some projects in a solution might be class libraries, some executable applications, and some might be unit test projects or websites.

Let's add a unit test project to our solution. This time we'll start from a project template so we don't have to add an additional code file to the project.

1. From the right-click or context menu of **Solution 'QuickSolution'** in **Solution Explorer**, select **Add** > **New Project**.

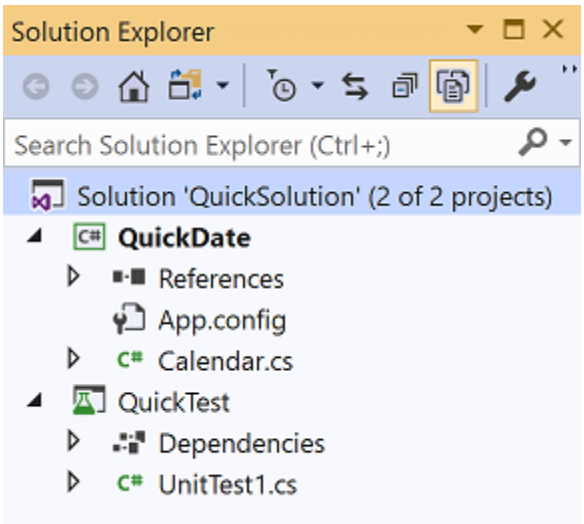
2. In the **Add a new project** dialog box, enter the text **unit test** into the search box at the top, and then select **C#** under **Language**.
3. Select the **Unit Test Project** project template for .NET Core, and then select **Next**.

   Note

   Starting in Visual Studio 2019 version 16.9, the MSTest project template name changed from **MSTest Unit Test Project (.NET Core)** to **Unit Test Project**. Several steps in the project creation changed in this update.

4. Name the project **QuickTest**, and then select **Next**.
5. Choose either the recommended target framework (.NET Core 3.1) or .NET 5, and then choose **Create**.

   A second project is added to **Solution Explorer**, and a file named *UnitTest1.cs* opens in the editor.



Add a project reference

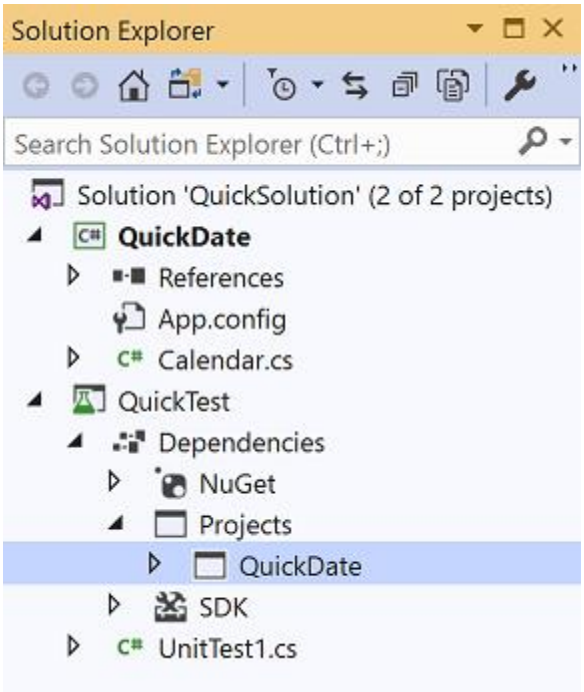
We're going to use the new unit test project to test our method in the **QuickDate** project, so we need to add a reference to that project. This creates a *build dependency* between the two projects, meaning that when you build the solution, **QuickDate** is built before **QuickTest**.

1. Select the **Dependencies** node in the **QuickTest** project, and from the right-click or context menu, select **Add Project Reference...**.

   The **Reference Manager** dialog box opens.

2. In the left pane, expand **Projects**, and then select **Solution**. In the middle pane, select the checkbox next to **QuickDate**, and then select **OK**.

   A reference to the **QuickDate** project is added.



Add test code

1. Now we'll add test code to the C# test code file. Replace the contents of *UnitTest1.cs* with the following code:

   C#Copy

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace QuickTest
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestGetCurrentDate()
        {
            Assert.AreEqual(DateTime.Now.Date,
QuickDate.Calendar.GetCurrentDate());
        }
    }
}
```

You'll see a red squiggle under some of the code. We'll fix this error by making the test project a friend assembly to the **QuickDate** project.

2. Back in the **QuickDate** project, open the *Calendar.cs* file if it's not already open. Add the following using statement and InternalsVisibleToAttribute attribute to the top of the file to resolve the error in the test project.

   C#Copy

   ```
   using System.Runtime.CompilerServices;

   [assembly: InternalsVisibleTo("QuickTest")]
   ```

   The code file should look like this:

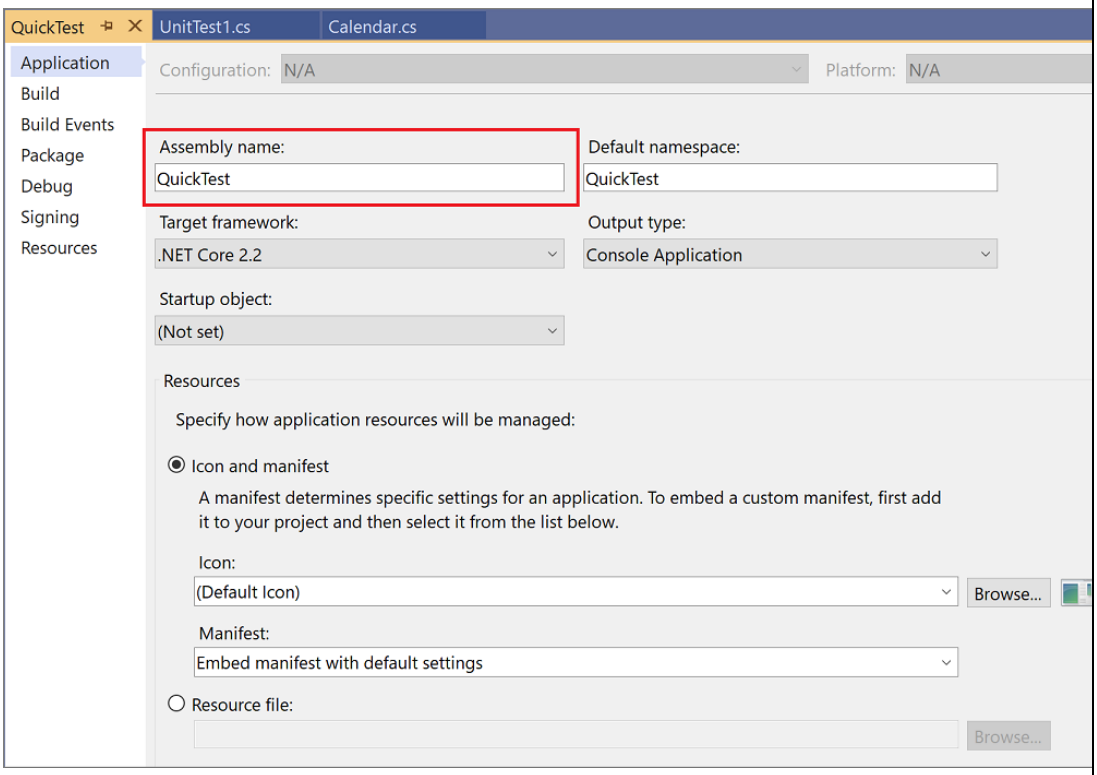
```
using System;
using System.Runtime.CompilerServices;

[assembly: InternalsVisibleTo("QuickTest")]

namespace QuickDate
{
    internal class Calendar
    {
        static void Main(string[] args)
        {
            DateTime now = GetCurrentDate();
            Console.WriteLine($"Today's date is {now}");
            Console.ReadLine();
        }

        internal static DateTime GetCurrentDate()
        {
            return DateTime.Now.Date;
        }
    }
}
```

Project properties

The line in the *Calendar.cs* file that contains the InternalsVisibleToAttribute attribute references the assembly name (file name) of the **QuickTest** project. The assembly name might not always be the same as the project name. To find the assembly name of a project, open the project properties.

1. In **Solution Explorer**, select the **QuickTest** project. From the right-click or context menu, select **Properties**, or just press **Alt**+**Enter**.

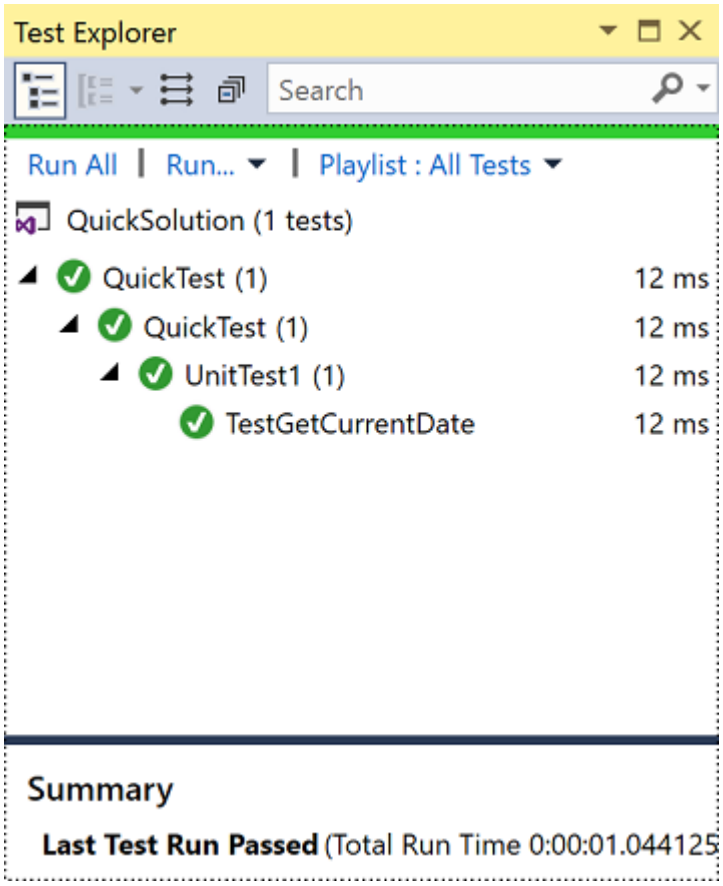
   The *property pages* for the project open on the **Application** tab. The property pages contain various settings for the project. Notice that the assembly name of the **QuickTest** project is indeed "QuickTest". If you wanted to change it, this is where you'd do that. Then, when you build the test project, the name of the resulting binary file would change from *QuickTest.dll* to whatever you chose.



2. Explore some of the other tabs of the project's property pages, such as **Build** and **Debug**. These tabs are different for different types of projects.

Next steps

If you want to check that your unit test is working, choose **Test** > **Run All Tests** from the menu bar. A window called **Test Explorer** opens, and you should see that the **TestGetCurrentDate** test passes.



OBJECTIVE

This software is developed to computerize the management system of **"RailwayReservationSystem"** Process data and

maintain all the records or database. It will provide the various advantages to the reservation, such as

- Increase train
- Decrease trains
- Ticket Reservation for candidate
- Cancel reserved ticket.
- Easy to generate reports.
- Easy to search data.
- Easy to maintain database.

Component we have used to create this project is –

- Microsoft Visual Studio interprise 2015
- OS Name Microsoft Windows 10 Home
- Version 10.0.19043 Build 19043
- Other OS Description Not Available
- OS Manufacturer Microsoft Corporation
- System Name DESKTOP-VNH6BFU

- System Manufacturer Acer
- System Model Aspire A515-51G
- System Typex64-based PC
- System SKU 0000000000000000
- Processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2712 Mhz, 2 Core(s), 4 Logical Processor(s)
- BIOS Version/Date Insyde Corp. V2.02, 03-01-2019
- SMBIOS Version 3.0
- Embedded Controller Version 1.17
- BIOS Mode UEFI
- BaseBoard Manufacturer KBL
- BaseBoard Product Charmander_KL
- BaseBoard Version V2.02
- Platform Role Mobile
- Secure Boot State On
- PCR7 Configuration Elevation Required to View
- Windows Directory C:\Windows
- System Directory C:\Windows\system32
- Boot Device \Device\HarddiskVolume1
- Locale United States

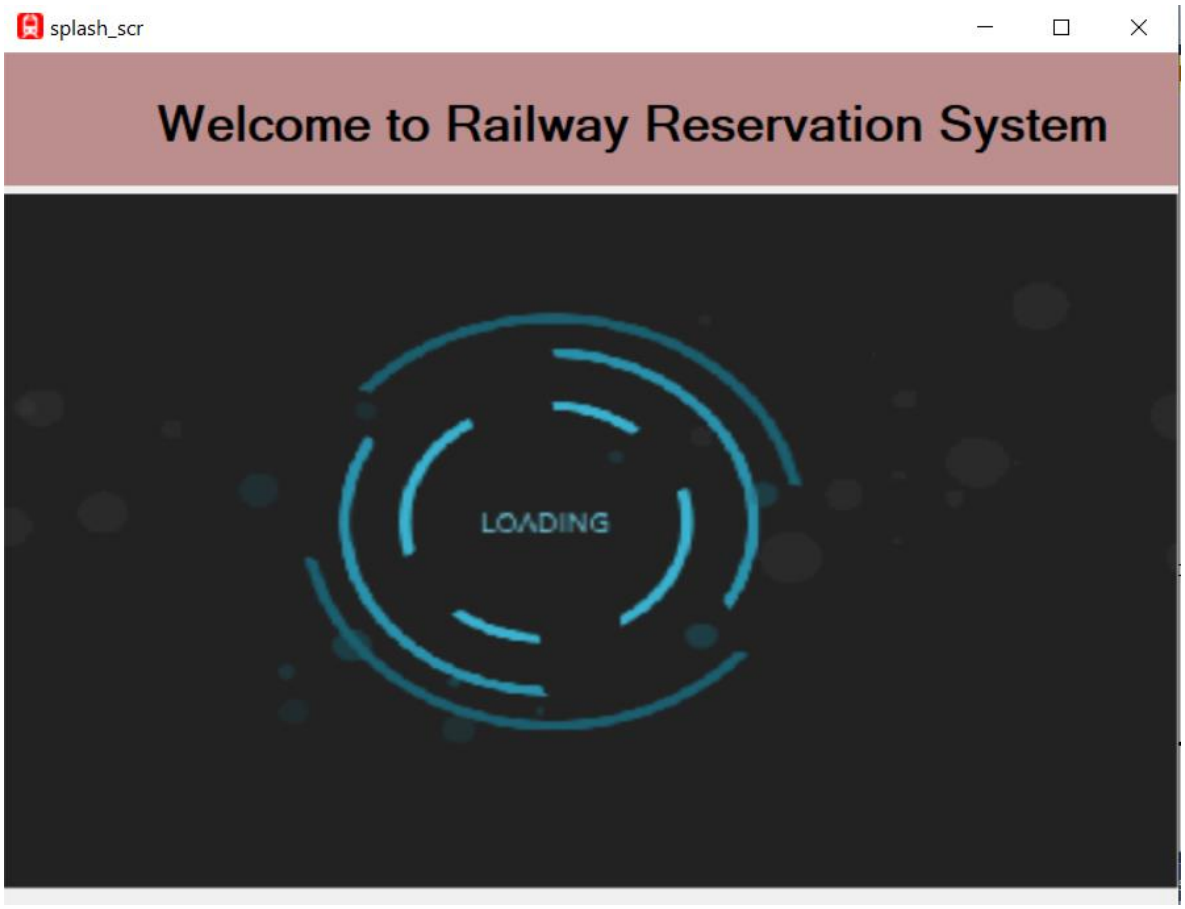
- Hardware Abstraction Layer Version = "10.0.19041.1151"
- User Name DESKTOP-VNH6BFU\aditya kumar
- Time Zone India Standard Time
- Installed Physical Memory (RAM) 8.00 GB
- Total Physical Memory 7.88 GB
- Available Physical Memory 3.75 GB
- Total Virtual Memory 9.13 GB
- Available Virtual Memory 3.81 GB
- Page File Space 1.25 GB
- Page File C:\pagefile.sys
- Kernel DMA Protection Not Available
- Virtualization-based security Not enabled
- Device Encryption Support Elevation Required to View
- Hyper-V - VM Monitor Mode Extensions Yes
- Hyper-V - Second Level Address Translation Extensions Yes
- Hyper-V - Virtualization Enabled in Firmware Yes
- Hyper-V - Data Execution Protection Yes

Splash_spr: it is a splash screen

```
Public Class splash_scr
```

```
    Private Sub splash_scr_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load

    End Sub

    Private Sub FrmLoading_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load
        Timer1.Start()
        ProgressBar1.Visible = False
    End Sub

    Private Sub Timer1_Tick(sender As
Object, e As EventArgs) Handles
Timer1.Tick
        ProgressBar1.Increment(2)
        If ProgressBar1.Value =
ProgressBar1.Maximum Then
            Me.Dispose()
            Form1.Show()
            Timer1.Stop()

        End If

    End Sub

    Private Sub Panel1_Paint(sender As
Object, e As PaintEventArgs) Handles
Panel1.Paint



    End Sub
```

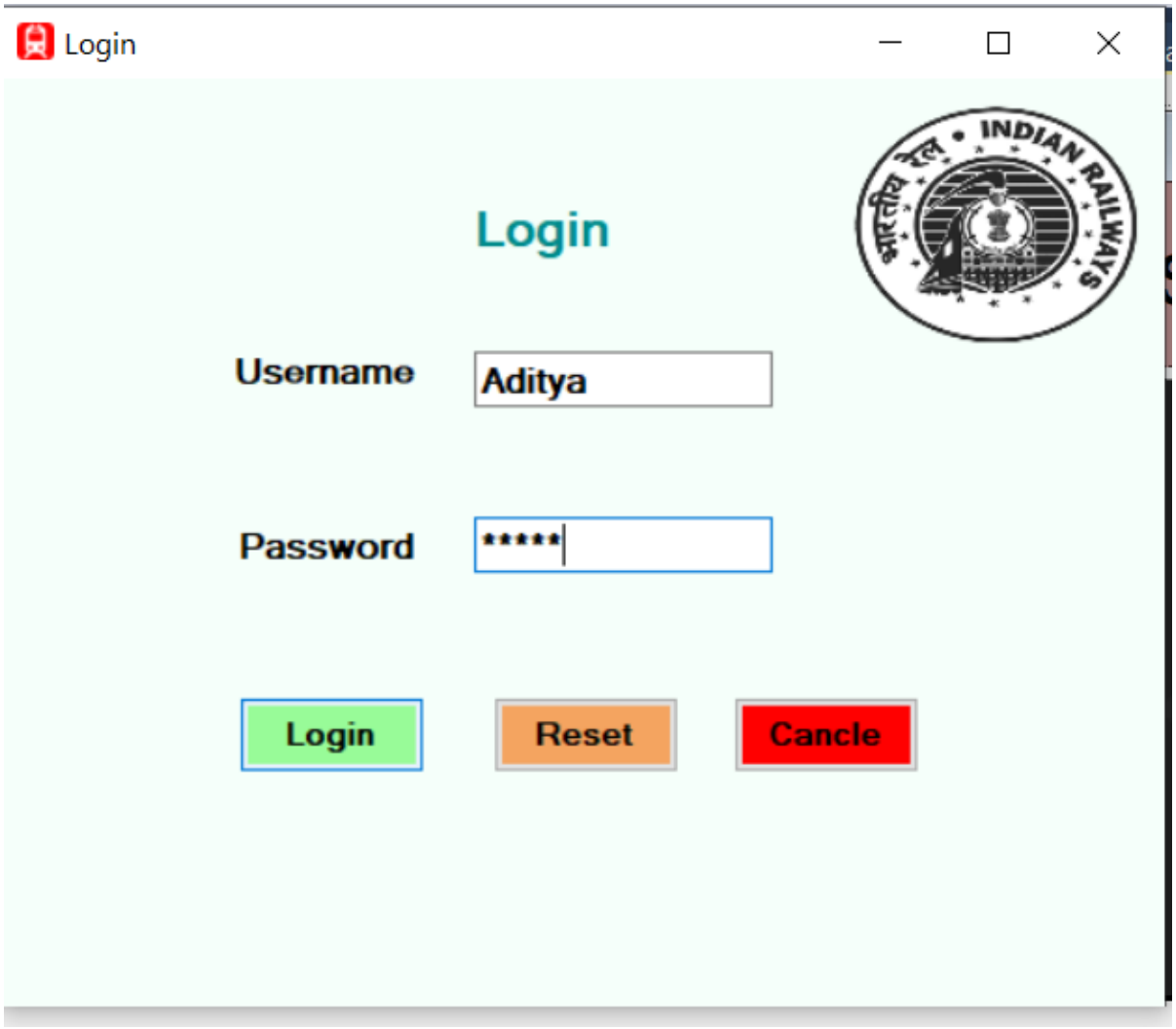
```
End Class
```

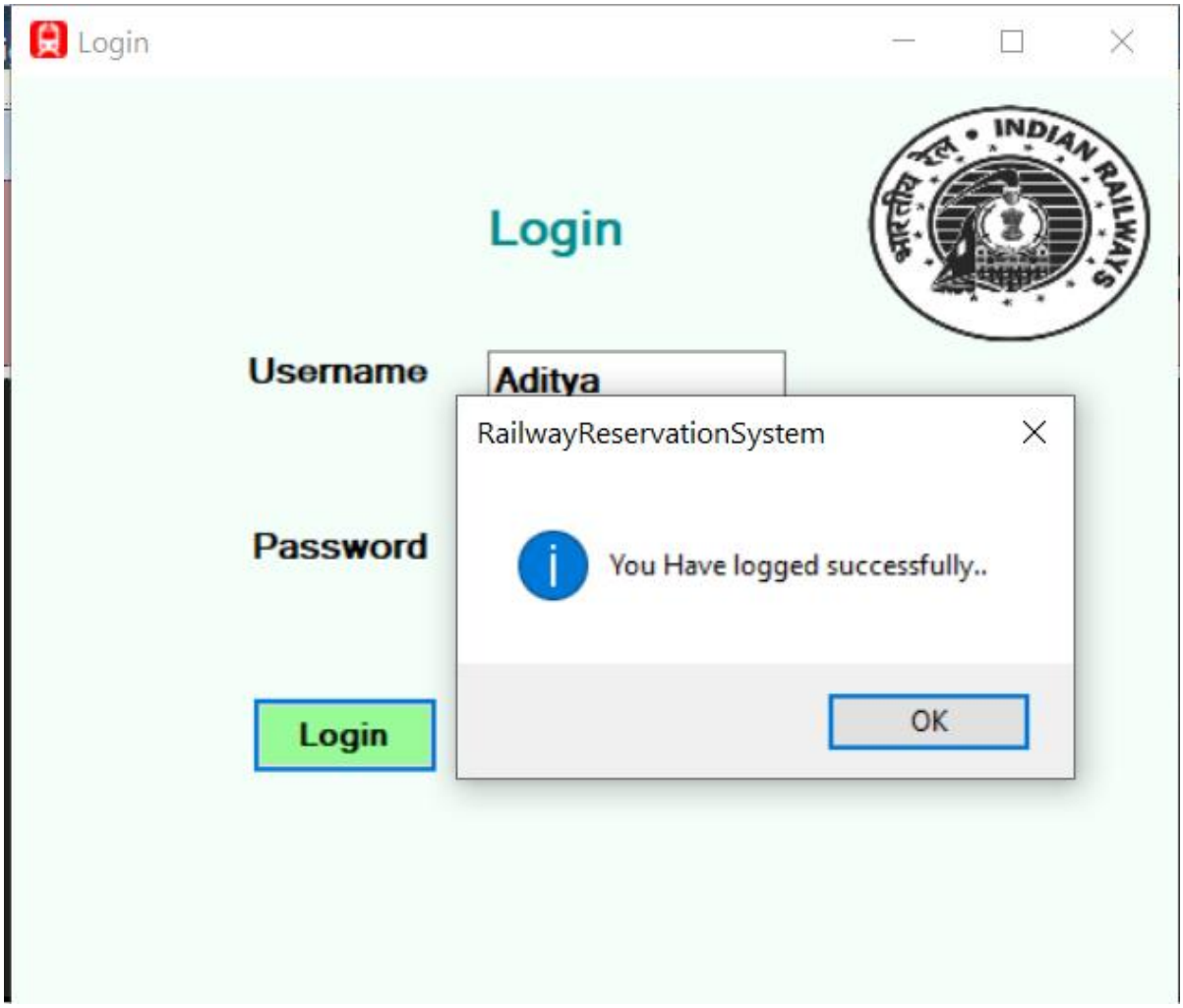


Form 1: this is a login form for officer userid= Aditya and password=kumar

```
Public Class Form1
    Private Sub Button1_Click(sender As
Object, e As EventArgs) Handles
Button1.Click
```

```
        If TextBox1.Text = "Aditya" And
TextBox2.Text = "kumar" Then
            MsgBox("You Have logged
successfully..", MsgBoxStyle.Information)
            Home.Show()
            Hide()
        Else
            MsgBox("login fail",
MsgBoxStyle.Critical +
MsgBoxStyle.OkCancel)
        End If
    End Sub

    Private Sub Button2_Click(sender As
Object, e As EventArgs) Handles
Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
    End Sub

    Private Sub Button3_Click(sender As
Object, e As EventArgs) Handles
Button3.Click
        Me.Dispose()
    End Sub
End Class
```


Login
Login
Username
Aditya
Password
*****
Login
Reset
Cancle

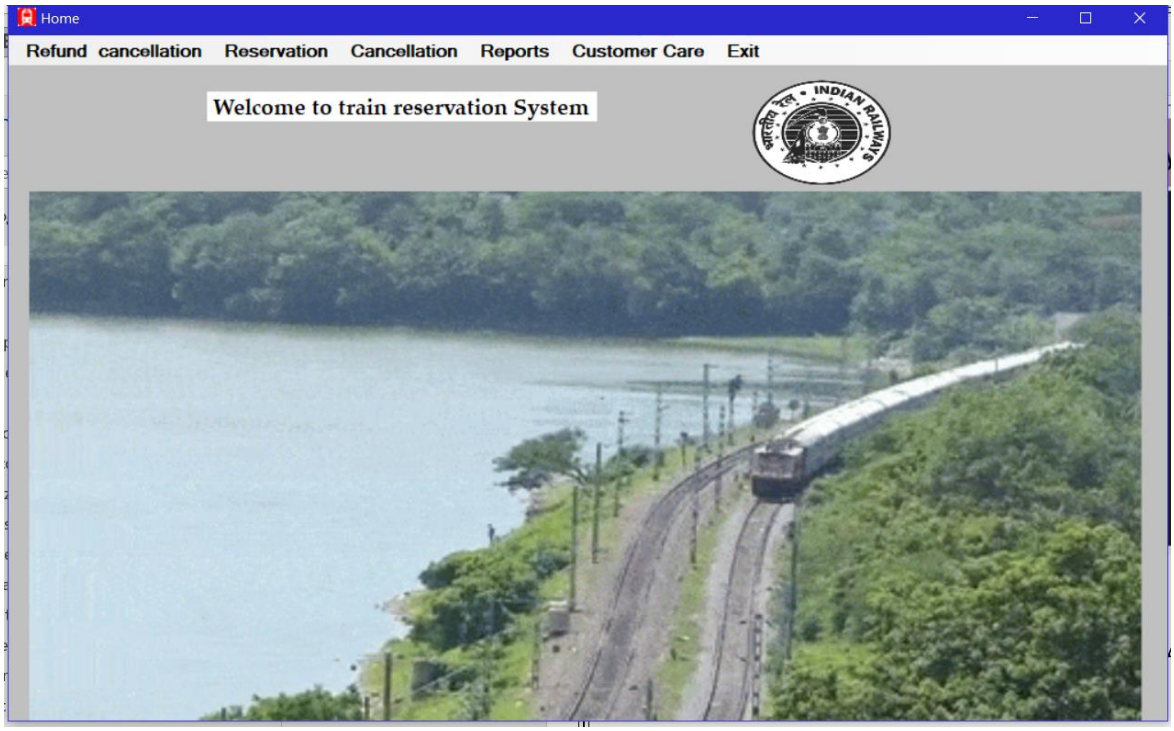


Home: this is main where you can perform multiple tasks.

```
Public Class Home
    Private Sub
TrainToolStripMenuItem_Click(sender
As Object, e As EventArgs) Handles
TrainToolStripMenuItem.Click
        NewTrain.ShowDialog()
    End Sub
```

```
    Private Sub
ReservationToolStripMenuItem_Click(se
nder As Object, e As EventArgs)
Handles
ReservationToolStripMenuItem.Click
        Reservation.ShowDialog()
    End Sub

    Private Sub
CancellationToolStripMenuItem_Click(s
ender As Object, e As EventArgs)
Handles
CancellationToolStripMenuItem.Click
        Cancellation.ShowDialog()
    End Sub

    Private Sub
ReportsToolStripMenuItem_Click(sender
As Object, e As EventArgs) Handles
ReportsToolStripMenuItem.Click
        Reports.ShowDialog()
    End Sub

    Private Sub
CustomerCareToolStripMenuItem1_Click(
sender As Object, e As EventArgs)
Handles
CustomerCareToolStripMenuItem1.Click
        CustomerCare.ShowDialog()
```

```
    End Sub

    Private Sub
ExitToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
        Me.Dispose()
    End Sub

    Private Sub Home_Load(sender As
Object, e As EventArgs) Handles
MyBase.Load

    End Sub
End Class
```



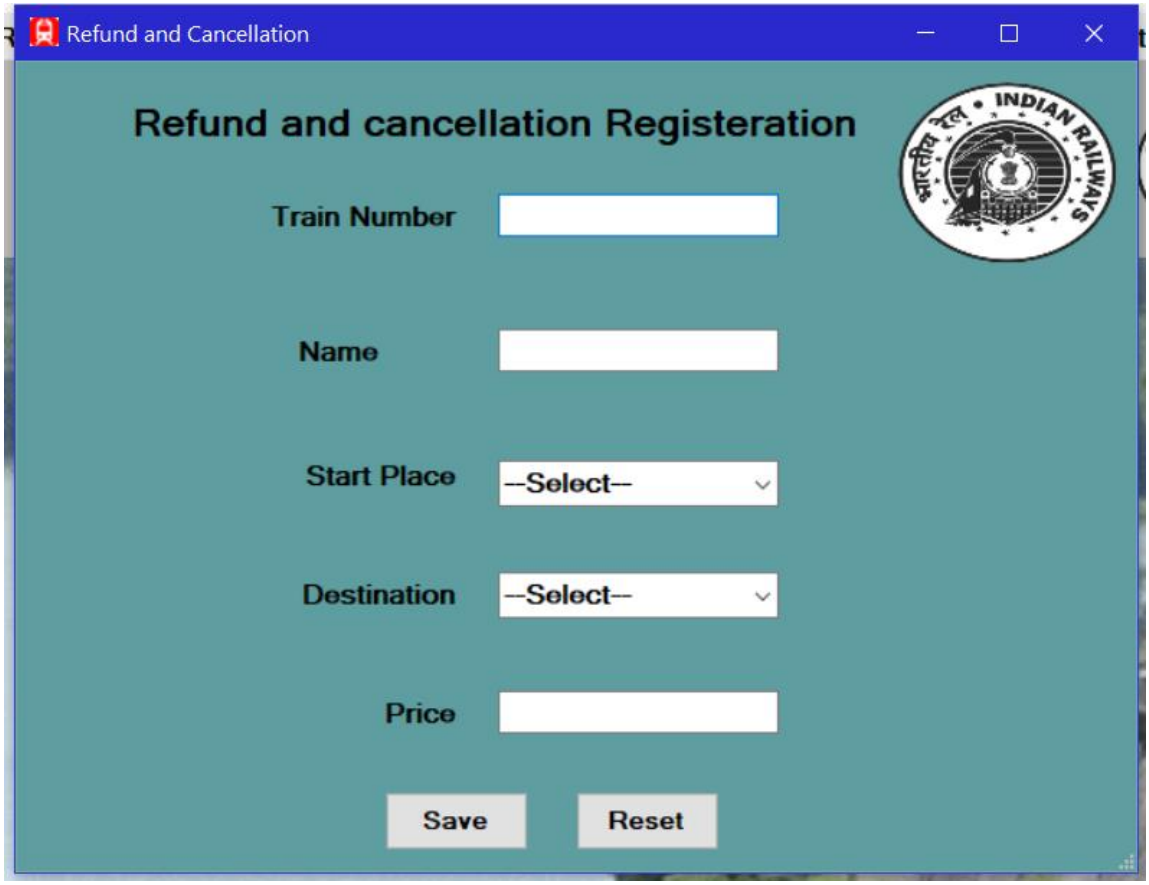
Refund & Cancellation Registration:

Here we can register for refund and cancellation.

```
Imports System.Data
Imports System.Data.SqlClient

Public Class NewTrain
    Dim con As New SqlConnection
    Dim da As New SqlDataAdapter
    Dim com As SqlCommand
    Dim ds As New DataSet
    Dim dr As SqlDataReader
    Private Sub Button1_Click(sender
As Object, e As EventArgs) Handles
Button1.Click
        con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
```

```
        com = New SqlCommand("insert
into
train(name,s_place,dest,price)values(
'" & TextBox2.Text & "','" &
ComboBox1.Text & "','" &
ComboBox2.Text & "','" &
TextBox3.Text & "')", con)
        con.Open()
        com.ExecuteNonQuery()
        MsgBox("New Train Information
Inserted Successfullyy..")
        Hide()
        con.Close()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        ComboBox1.Text = "--Select--"
        ComboBox2.Text = "--Select--"

    End Sub

    Private Sub Button2_Click(sender
As Object, e As EventArgs) Handles
Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        ComboBox1.Text = "--Select--"
```

```
        ComboBox2.Text = "--Select--"

    End Sub
End Class
```



Cancellation: here we can delete the cancel ticket from database. Enter id or pnr number to delete.

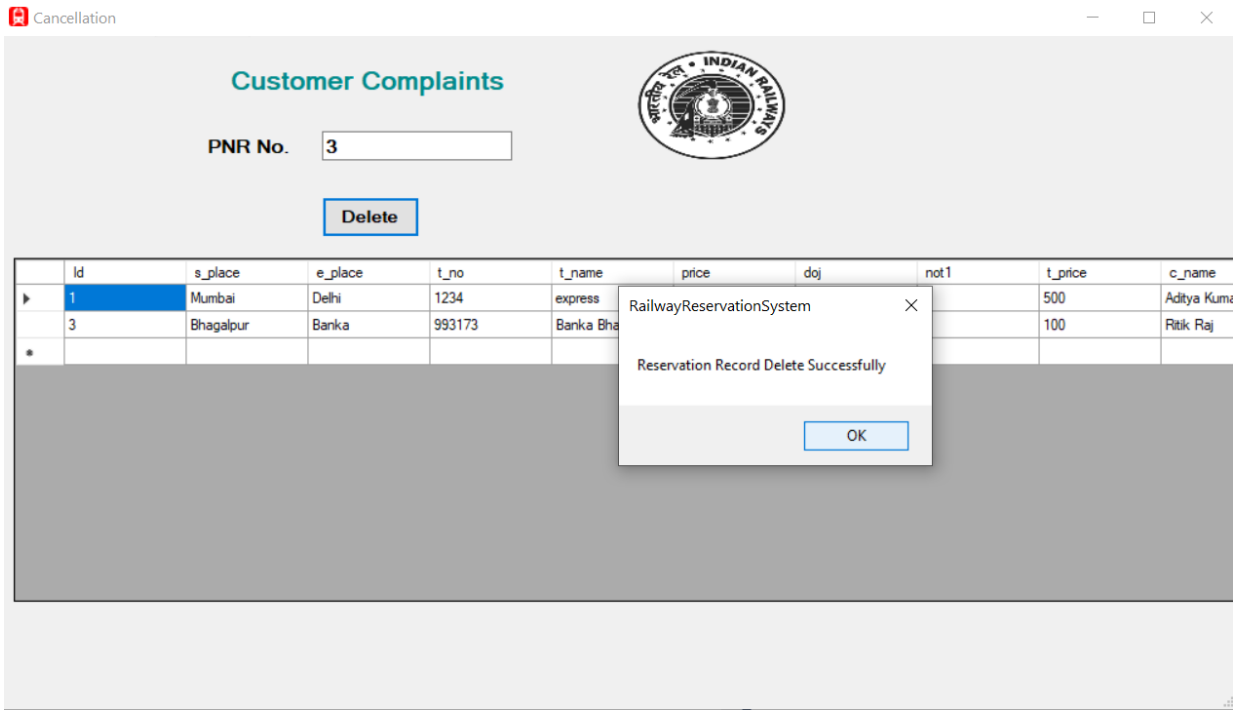
```
Imports System.Data
Imports System.Data.SqlClient
```

```
Public Class Cancellation
    Dim con As New SqlConnection
    Dim da As New SqlDataAdapter
    Dim com As SqlCommand
    Dim ds As New DataSet
    Dim dt As DataTable
    Dim dv As DataView
    Dim str As String
    Dim reser As Object


    Private Sub Button1_Click(sender
As Object, e As EventArgs) Handles
Button1.Click
        Try
            con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
            con.Open()
            str = "DELETE FROM
reservation WHERE id = '" &
TextBox1.Text & "'"
```

```
            com = New SqlCommand(Str,
con)
            com.ExecuteNonQuery()
            con.Close()
            MsgBox(" Reservation
Record Delete Successfully")
            Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                str = "SELECT * FROM
reservation"
                com = New
SqlCommand(str, con)
                da = New
SqlDataAdapter(com)
                dt = New DataTable()
                dv = New DataView()
                da.Fill(dt)


DataGridView1.DataSource = New
BindingSource(dt, reser)
                Hide()
                TextBox1.Text = ""
```

```
            End Using

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

    End Sub

    Private Sub
Cancellation_Load(sender As Object, e
As EventArgs) Handles MyBase.Load
        'TODO: This line of code
loads data into the
'RailwayDataSet.reservation' table.
You can move, or remove it, as
needed.

Me.ReservationTableAdapter.Fill(Me.Ra
ilwayDataSet.reservation)

    End Sub
End Class
```


Cancellation
Customer Complaints
PNR No.
3
Delete
Id
s_place
e_place
t_no
t_name
price
doj
not1
t_price
c_name
1
Mumbai
Delhi
1234
express
500
20/05/2021
1
500
Aditya Kuma
3
Bhagalpur
Banka
993173
Banka Bhagalpur...
100
16 September 20...
1
100
Ritik Raj

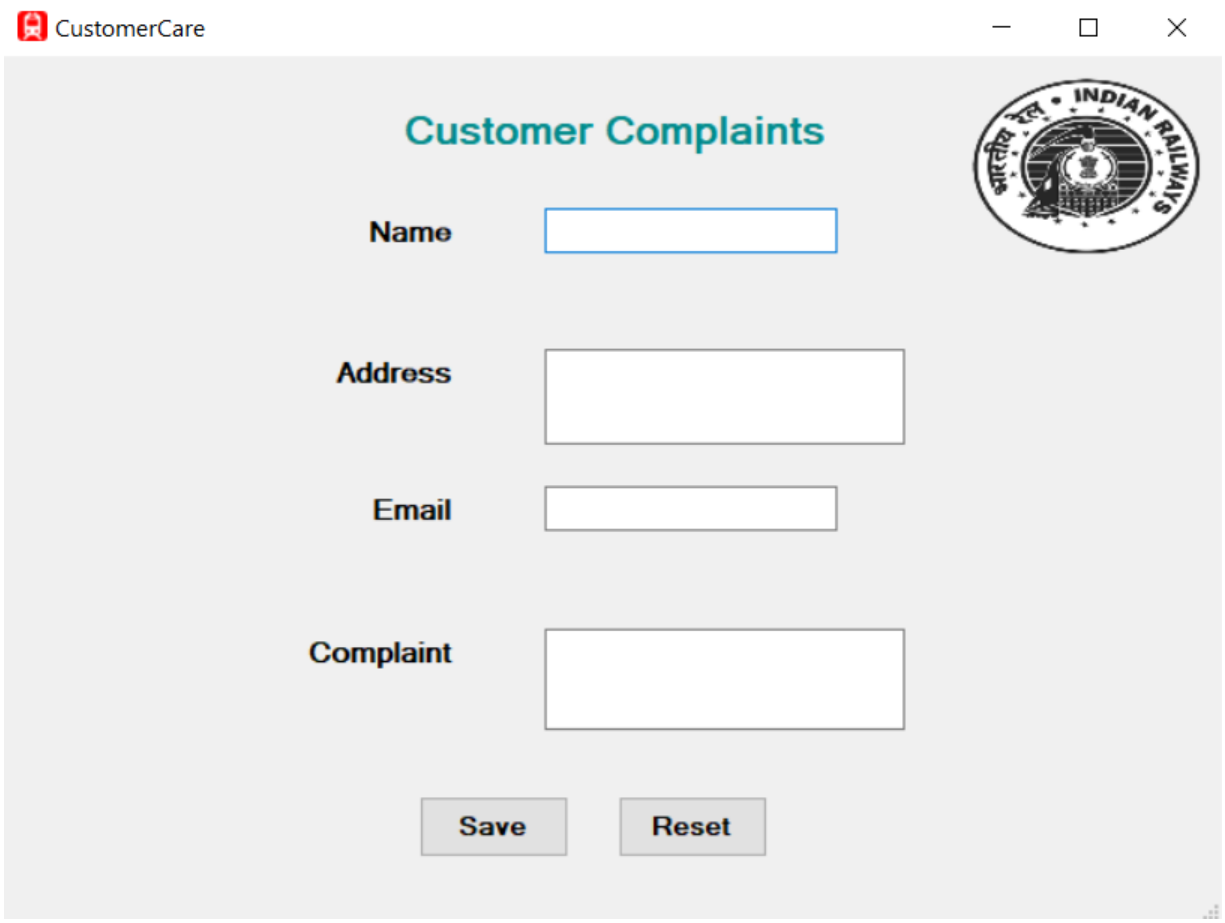


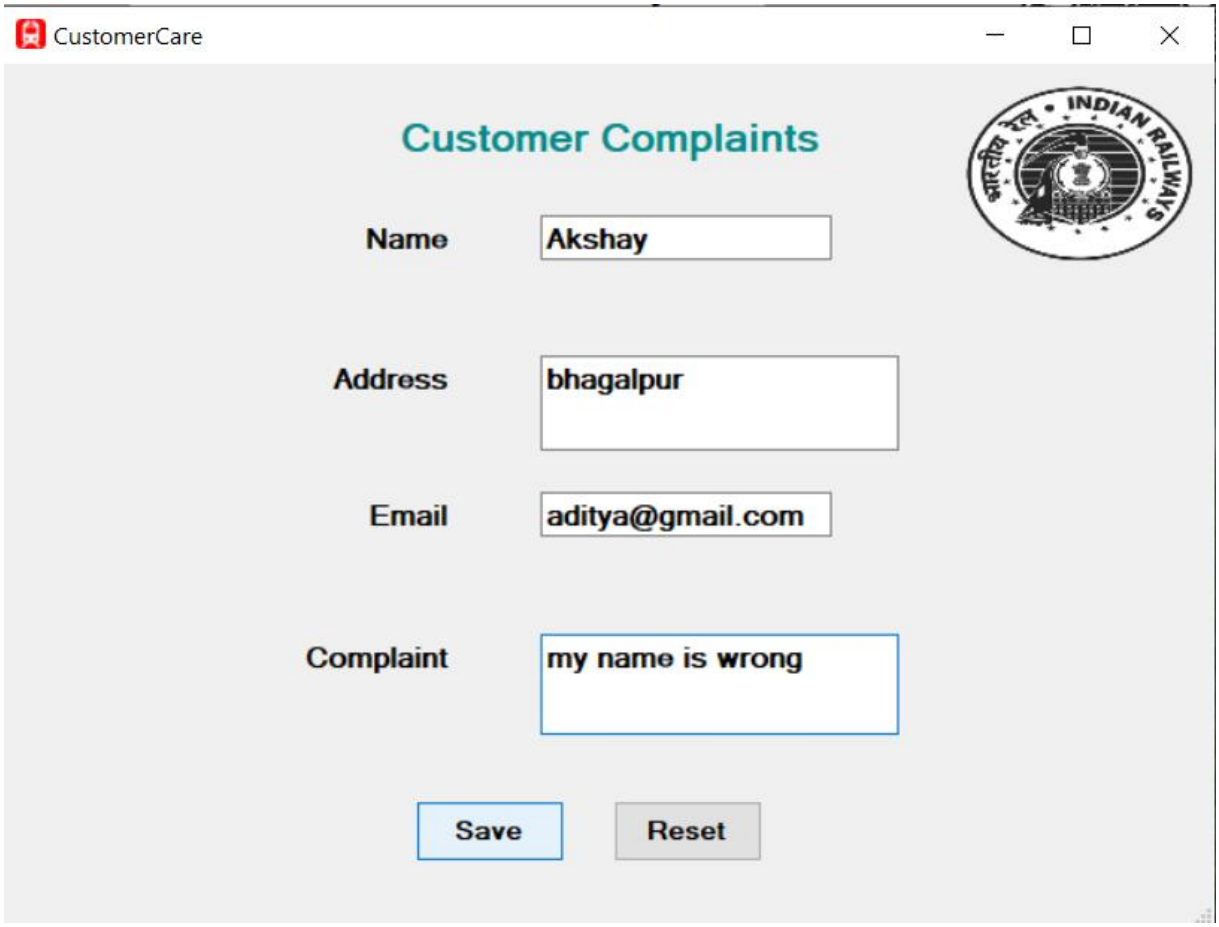
Customer Care: fill this for any complaints or feedback of customer

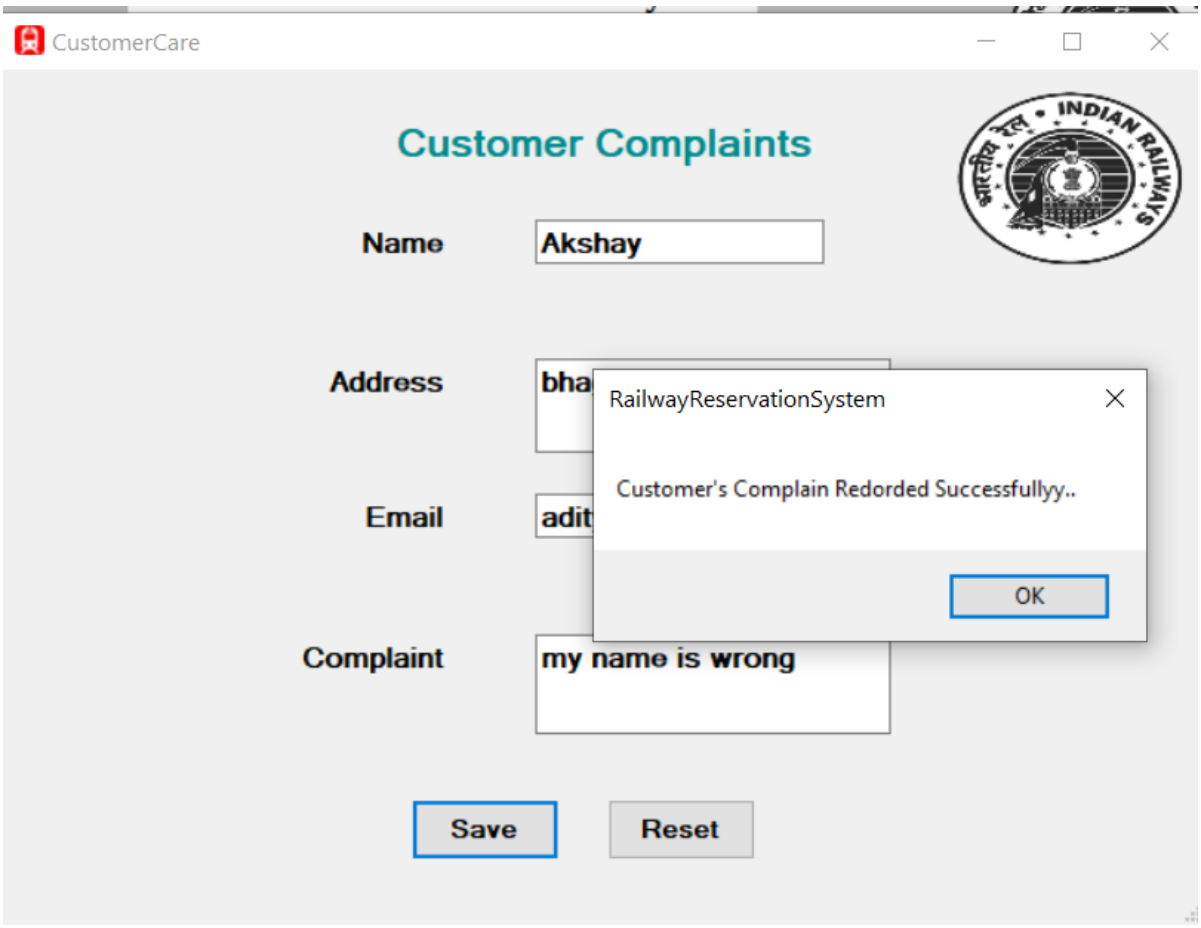
```
Imports System.Data
Imports System.Data.SqlClient

Public Class CustomerCare
    Dim con As New SqlConnection
    Dim da As New SqlDataAdapter
    Dim com As SqlCommand
    Dim ds As New DataSet
    Dim dr As SqlDataReader
```

```
    Private Sub Button1_Click(sender
As Object, e As EventArgs) Handles
Button1.Click
        con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")


        com = New SqlCommand("insert
into
complain(name,addr,email,complain)val
ues('" & TextBox1.Text & "','" &
TextBox2.Text & "','" & TextBox3.Text
& "','" & TextBox4.Text & "')", con)
        con.Open()
        com.ExecuteNonQuery()
        MsgBox("Customer's Complain
Redorded Successfullyy..")
        Hide()
        con.Close()
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
```

```
    End Sub

    Private Sub Button2_Click(sender
As Object, e As EventArgs) Handles
Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""

    End Sub
End Class
```


CustomerCare
Customer Complaints
INDIAN RAILWAYS
Name
Address
Email
Complaint
Save
Reset


CustomerCare
Customer Complaints
INDIAN RAILWAYS
Name
Akshay
Address
bhagalpur
Email
aditya@gmail.com
Complaint
my name is wrong
Save
Reset



Reports: here we can check history of booked tickets and cancel ticket and many more.

```
Imports System.Data
Imports System.Data.SqlClient

Public Class Reports
    Dim con As New SqlConnection
    Dim da As New SqlDataAdapter
```

```
    Dim com As SqlCommand
    Dim ds As New DataSet
    Dim dt As DataTable
    Dim dv As DataView
    Dim str As String
    Dim train As Object
    Dim reservation As Object
    Dim complain As Object
    Private Sub Button1_Click(sender
As Object, e As EventArgs) Handles
Button1.Click
        If ComboBox1.SelectedIndex =
"0" Then
            Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                str = "SELECT * FROM
train WHERE id = '" + TextBox3.Text +
"'"
                com = New
SqlCommand(str, con)
                da = New
SqlDataAdapter(com)
                dt = New DataTable()
```

```
                    dv = New DataView()
                    da.Fill(dt)


DataGridView1.DataSource = New
BindingSource(dt, train)
                End Using
            ElseIf
ComboBox1.SelectedIndex = "1" Then
                Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                    str = "SELECT * FROM
reservation WHERE id = '" +
TextBox3.Text + "'"
                    com = New
SqlCommand(str, con)
                    da = New
SqlDataAdapter(com)
                    dt = New DataTable()
                    dv = New DataView()
                    da.Fill(dt)
```

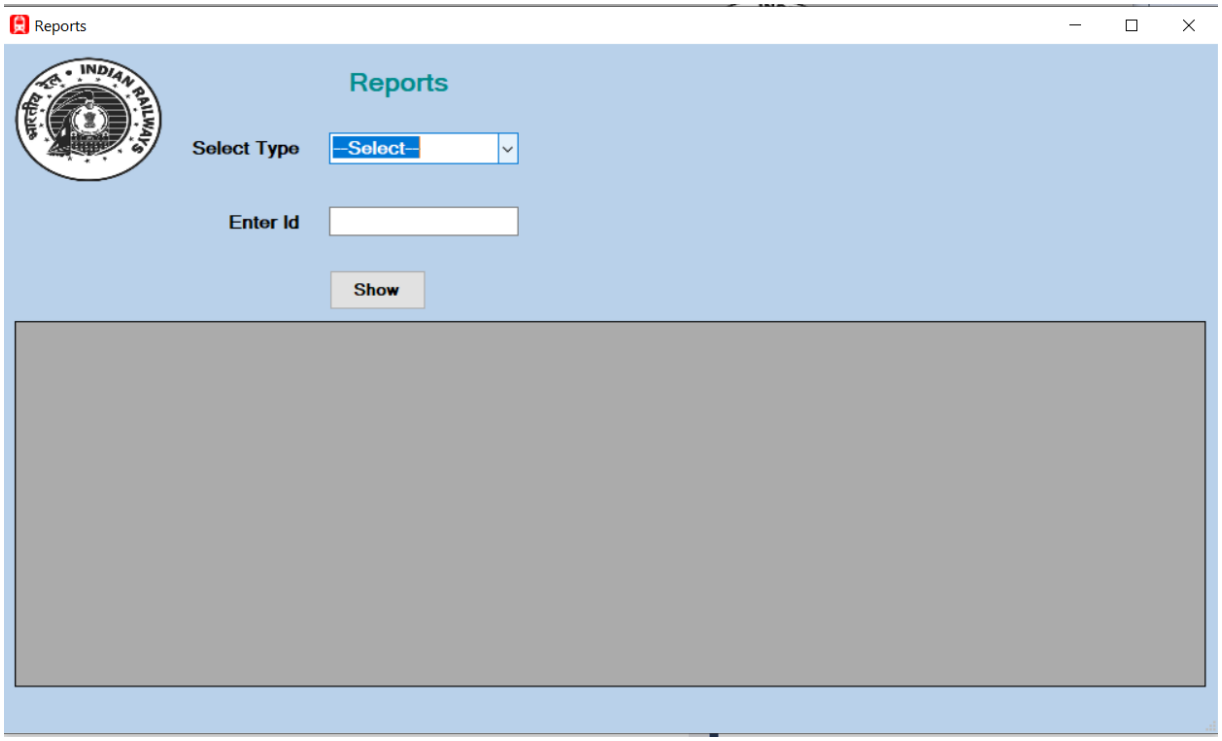
```
DataGridView1.DataSource = New
BindingSource(dt, reservation)
            End Using

        ElseIf
ComboBox1.SelectedIndex = "2" Then
            Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                str = "SELECT * FROM
complain WHERE id = '" +
TextBox3.Text + "'"
                com = New
SqlCommand(str, con)
                da = New
SqlDataAdapter(com)
                dt = New DataTable()
                dv = New DataView()
                da.Fill(dt)


DataGridView1.DataSource = New
BindingSource(dt, complain)
```

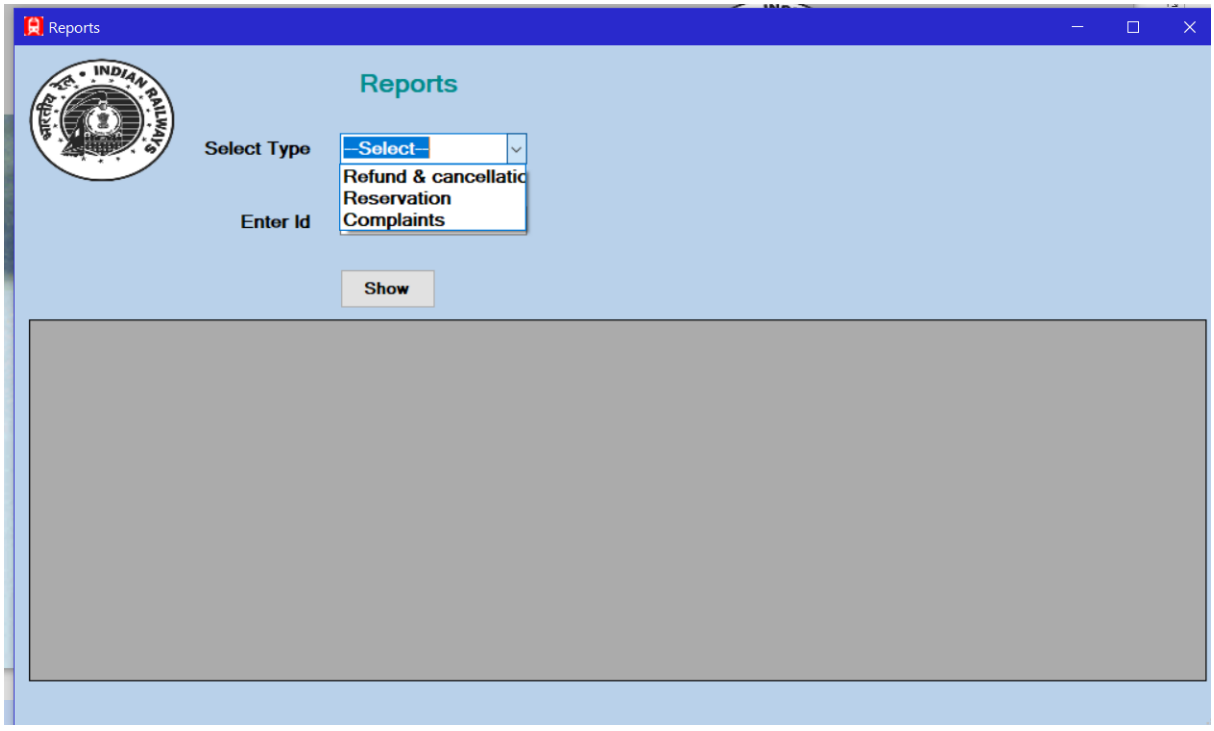
```
            End Using
            TextBox3.Text = ""
            ComboBox1.Text = "--
Select--"
        End If

    End Sub

    Private Sub
ComboBox1_SelectedIndexChanged(sender
As Object, e As EventArgs) Handles
ComboBox1.SelectedIndexChanged
        If ComboBox1.SelectedIndex =
"0" Then
            Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                str = "SELECT * FROM
train "
                com = New
SqlCommand(str, con)
                da = New
SqlDataAdapter(com)
                dt = New DataTable()
```

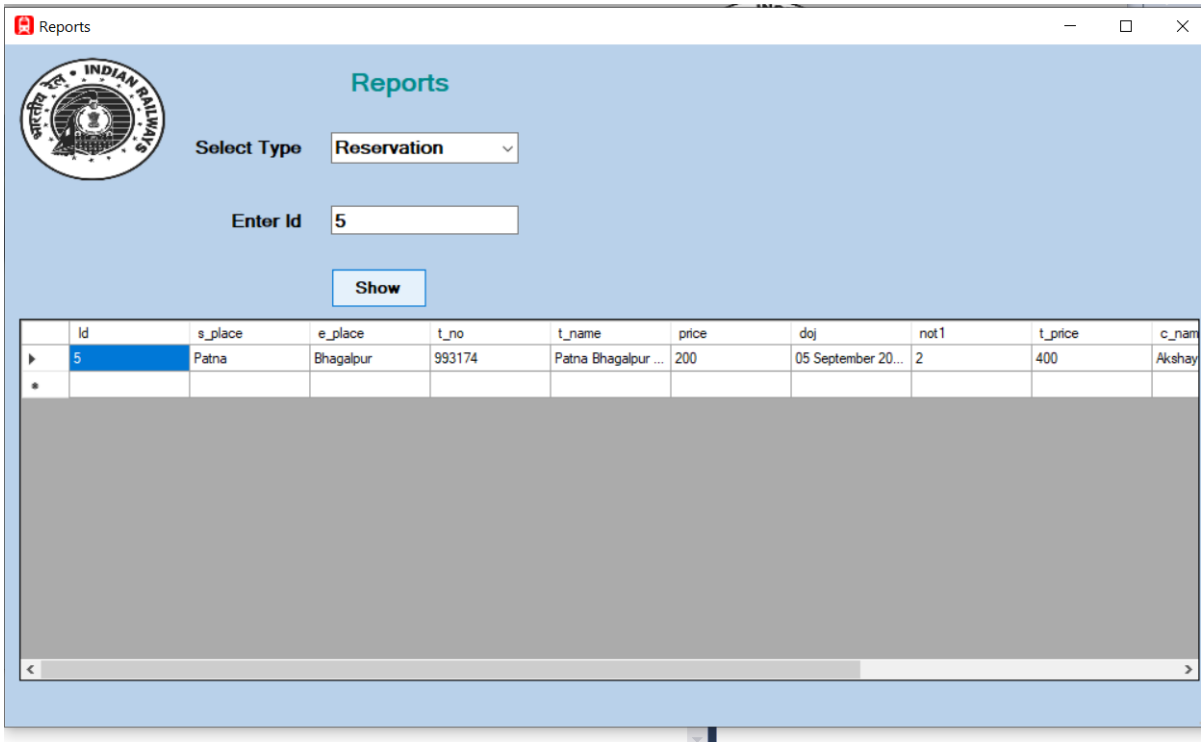
```
                    dv = New DataView()
                    da.Fill(dt)


DataGridView1.DataSource = New
BindingSource(dt, train)
                End Using
            ElseIf
ComboBox1.SelectedIndex = "1" Then
                Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                    str = "SELECT * FROM
reservation"
                    com = New
SqlCommand(str, con)
                    da = New
SqlDataAdapter(com)
                    dt = New DataTable()
                    dv = New DataView()
                    da.Fill(dt)
```

```
DataGridView1.DataSource = New
BindingSource(dt, reservation)
            End Using

        ElseIf
ComboBox1.SelectedIndex = "2" Then
            Using con = New
SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
                str = "SELECT * FROM
complain"
                com = New
SqlCommand(str, con)
                da = New
SqlDataAdapter(com)
                dt = New DataTable()
                dv = New DataView()
                da.Fill(dt)


DataGridView1.DataSource = New
BindingSource(dt, complain)
            End Using
```

```
            TextBox3.Text = ""
            ComboBox1.Text = "--
Select--"

        End If
    End Sub

    Private Sub Reports_Load(sender
As Object, e As EventArgs) Handles
MyBase.Load
        'TODO: This line of code
loads data into the
'RailwayDataSet.reservation' table.
You can move, or remove it, as
needed.

Me.ReservationTableAdapter.Fill(Me.Ra
ilwayDataSet.reservation)

    End Sub

    Private Sub
PictureBox1_Click(sender As Object, e
As EventArgs) Handles
PictureBox1.Click

    End Sub
End Class
```


Reports
Reports
Select Type
--Select--
Enter Id
Show


Reports
Reports
Select Type
--Select--
Refund & cancellatic
Reservation
Complaints
Enter Id
Show



Reservation: fill this form to book the tickets.

```
Imports System.Data
Imports System.Data.SqlClient

Public Class Reservation
    Dim con As New SqlConnection
    Dim da As New SqlDataAdapter
    Dim com As SqlCommand
    Dim ds As New DataSet
```

```
    Dim dr As SqlDataReader
    Dim getres As String

    Private Sub Button1_Click(sender
As Object, e As EventArgs) Handles
Button1.Click
        con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")


        com = New SqlCommand("insert
into
reservation(s_place,e_place,t_no,t_na
me,price,doj,not1,t_price,c_name,addr
,mob,email)values('" & ComboBox1.Text
& "','" & ComboBox2.Text & "','" &
TextBox2.Text & "','" & TextBox3.Text
& "','" & TextBox4.Text & "','" &
DateTimePicker1.Text & "','" &
TextBox6.Text & "','" & TextBox7.Text
& "','" & TextBox8.Text & "','" &
TextBox9.Text & "','" &
TextBox10.Text & "','" &
TextBox11.Text & "')", con)
```

```
            con.Open()
            com.ExecuteNonQuery()
            MsgBox("Reservation
Infromation Inserted
Successfullyy..")
            Hide()
            con.Close()
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            DateTimePicker1.Text = ""
            TextBox6.Text = ""
            TextBox7.Text = ""
            TextBox8.Text = ""
            TextBox9.Text = ""
            TextBox10.Text = ""
            TextBox11.Text = ""
            ComboBox1.Text = "--Select--"
            ComboBox2.Text = "--Select--"

        End Sub

        Private Sub
Reservation_Load(sender As Object, e
As EventArgs) Handles MyBase.Load

            con = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachD
bFilename=C:\Users\aditya
```

```
kumar\Desktop\Aditya_p\RailwayReserva
tionSystem\RailwayReservationSystem\r
ailway1.mdf;Integrated
Security=True")
        con.Open()



        Try
                getres = "SELECT
nextID=MAX(Id)+1 FROM reservation"
                com = New
SqlCommand(getres, con)
                dr =
com.ExecuteReader()
                If dr.Read() Then

                TextBox1.Text =
dr.GetValue(0).ToString()



            End If
            Catch ex As Exception
                MsgBox(ex.ToString)
        End Try
```

```
    End Sub


    Private Sub Button2_Click(sender
As Object, e As EventArgs) Handles
Button2.Click
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        DateTimePicker1.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        TextBox8.Text = ""
        TextBox9.Text = ""
        TextBox10.Text = ""
        TextBox11.Text = ""
        ComboBox1.Text = "--Select--"
        ComboBox2.Text = "--Select--"

    End Sub

    Private Sub Button3_Click(sender
As Object, e As EventArgs) Handles
Button3.Click
        If ComboBox1.Text =
"Bhagalpur" And ComboBox2.Text =
"Banka" Then
            TextBox4.Text = "100"
```

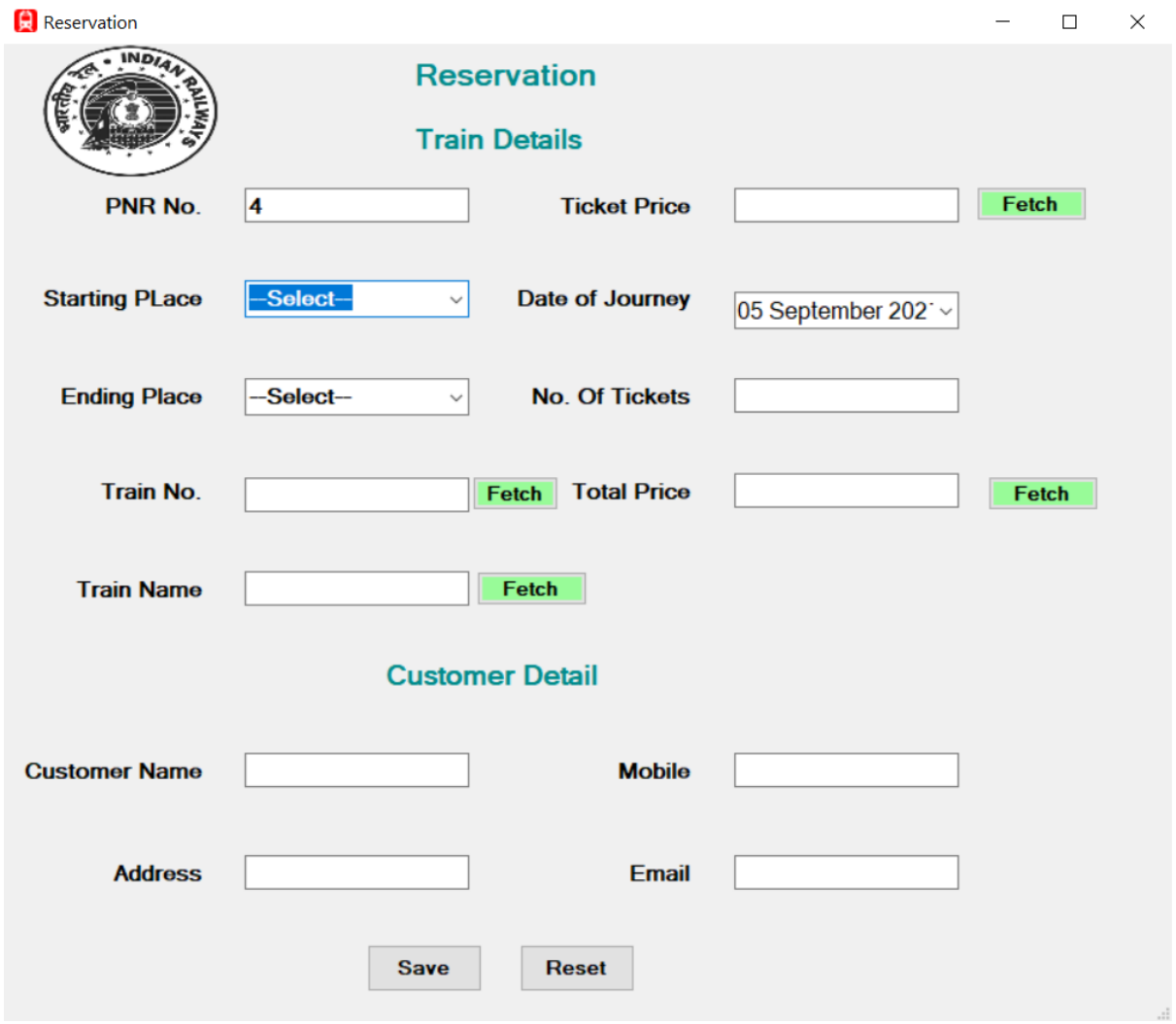
```
        ElseIf ComboBox1.Text =
"Patna" And ComboBox2.Text = "Banka"
Then
            TextBox4.Text = "200"


        ElseIf ComboBox1.Text =
"Banka" And ComboBox2.Text = "Patna"
Then
            TextBox4.Text = "300"




        ElseIf ComboBox1.Text =
"Banka" And ComboBox2.Text =
"Bhagalpur" Then
            TextBox4.Text = "100"

        ElseIf ComboBox1.Text =
"Patna" And ComboBox2.Text =
"Bhagalpur" Then
            TextBox4.Text = "200"


        ElseIf ComboBox1.Text =
"Bhagalpur" And ComboBox2.Text =
"Patna" Then
            TextBox4.Text = "300"
```

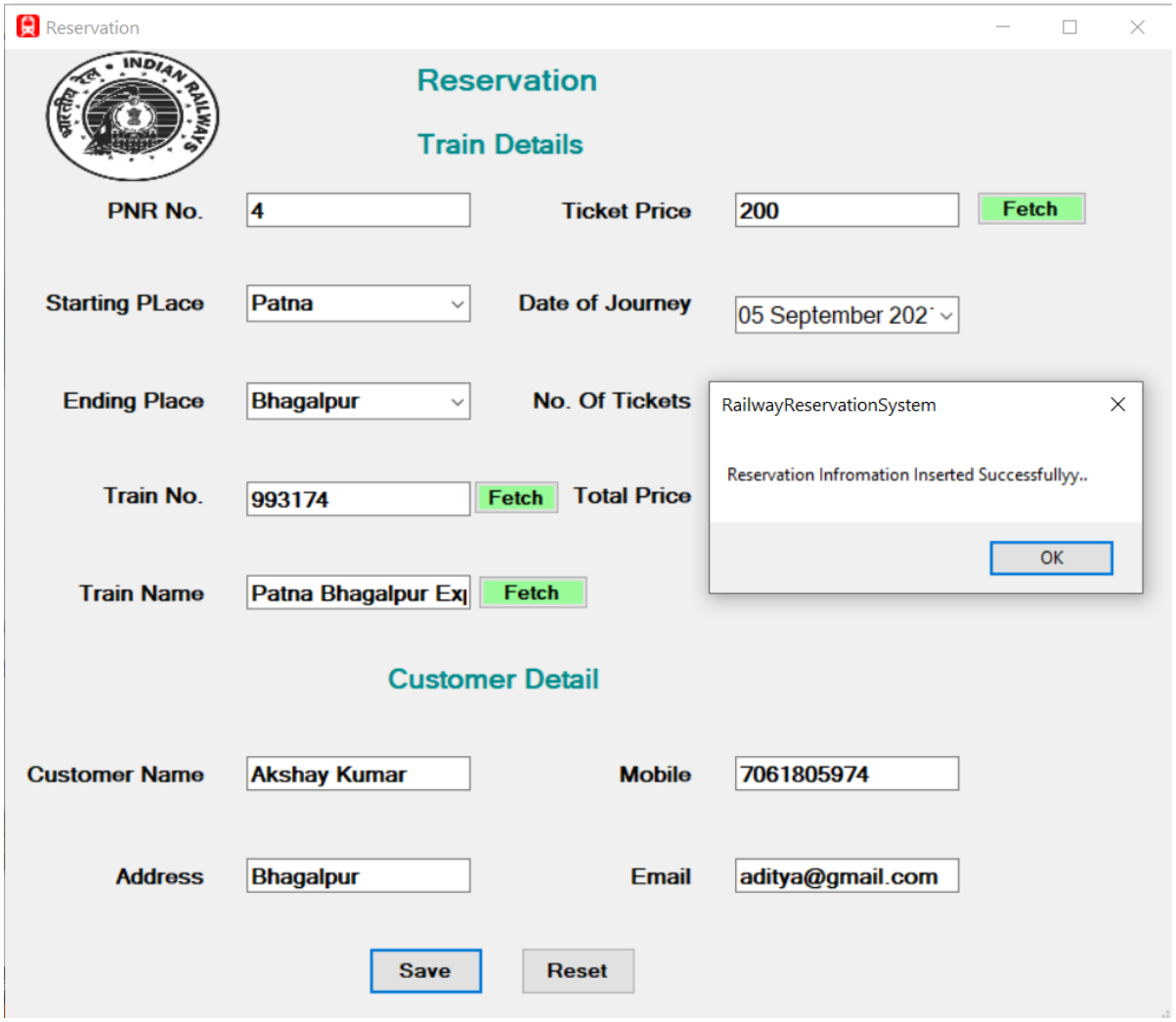
```
        Else
            TextBox4.Text = "0"
        End If
    End Sub

    Private Sub Button4_Click(sender
As Object, e As EventArgs) Handles
Button4.Click
        If TextBox6.Text = "enter" Or
TextBox6.Text = "enter" Then
            MsgBox("please enter
ticket no", MsgBoxStyle.Information +
vbOKOnly)
        Else
            TextBox7.Text =
TextBox4.Text * TextBox6.Text
        End If
    End Sub

    Private Sub Button5_Click(sender
As Object, e As EventArgs) Handles
Button5.Click
        If ComboBox1.Text =
"Bhagalpur" And ComboBox2.Text =
"Banka" Then
            TextBox2.Text = "993173"
            TextBox3.Text = "Banka
Bhagalpur Express"
```

```
        ElseIf ComboBox1.Text =
"Banka" And ComboBox2.Text =
"Bhagalpur" Then
            TextBox2.Text = "993173"
            TextBox3.Text = "Banka
Bhagalpur Express"
        ElseIf ComboBox1.Text =
"Patna" And ComboBox2.Text =
"Bhagalpur" Then
            TextBox2.Text = "993174"
            TextBox3.Text = "Patna
Bhagalpur Express"

        ElseIf ComboBox1.Text =
"Bhagalpur" And ComboBox2.Text =
"Patna" Then
            TextBox2.Text = "993174"
            TextBox3.Text = "Patna
Bhagalpur Express"

        ElseIf ComboBox1.Text =
"Patna" And ComboBox2.Text = "Banka"
Then
            TextBox2.Text = "993154"
            TextBox3.Text = "Patna
Banka Express"
```

```
        ElseIf ComboBox1.Text =
"Banka" And ComboBox2.Text = "Patna"
Then
            TextBox2.Text = "993154"
            TextBox3.Text = "Patna
Banka Express"




        Else
            TextBox2.Text = "00000"
And TextBox3.Text = "N/A"
        End If
    End Sub

    Private Sub TextBox4_Click(sender
As Object, e As EventArgs) Handles
TextBox4.Click
        TextBox4.Text = ""
    End Sub

    Private Sub TextBox6_Click(sender
As Object, e As EventArgs) Handles
TextBox6.Click
        TextBox6.Text = ""
    End Sub
```

```
    Private Sub
TextBox5_TextChanged(sender As
Object, e As EventArgs)

    End Sub
End Class
```




Reservation
Reservation
Train Details
PNR No.
4
Ticket Price
200
Fetch
Starting PLace
Patna
Date of Journey
05 September 202
Ending Place
Bhagalpur
No. Of Tickets
Train No.
993174
Fetch
Total Price
Train Name
Patna Bhagalpur Ex
Fetch
RailwayReservationSystem
Reservation Infromation Inserted Successfullyy..
OK
Customer Detail
Customer Name
Akshay Kumar
Mobile
7061805974
Address
Bhagalpur
Email
aditya@gmail.com
Save
Reset