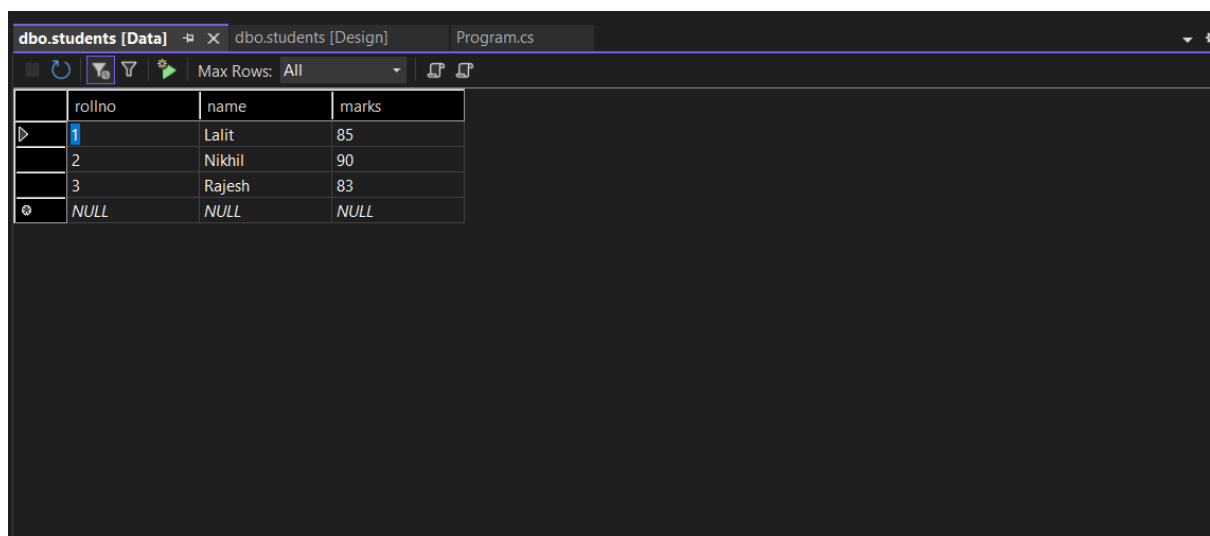
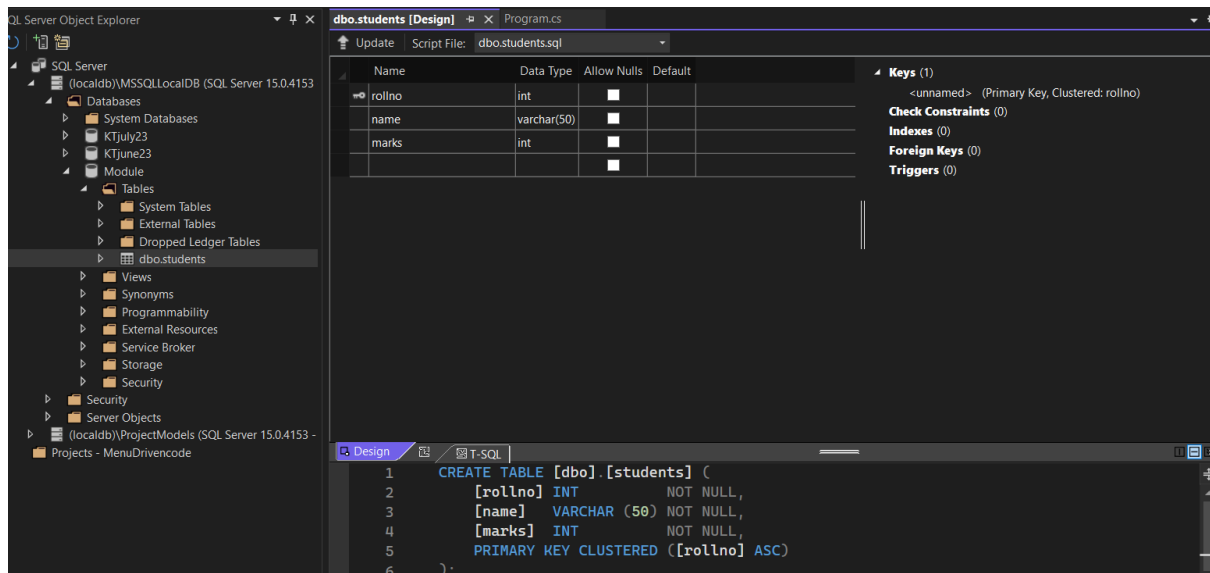


Name:Lalit Chaudhari

LAB EXAM

MS.Net Technologies

1. Create a menu driven C# application to interact with the database. You have to insert data into "Students" table and retrieve a list of all the students from the Students table and display their names, roll numbers, and their marks. In the above question, perform update and delete operations as well.



Name:Lalit Chaudhari

```
Program.cs
MenuDrivenCode
MenuDrivenDatabaseProgram.Program
Update()

1 using Microsoft.Data.SqlClient;
2 using System.Data;
3
4 namespace MenuDrivenDatabaseProgram
5 {
6     0 references
7     internal class Program
8     {
9         0 references
10        static void Main()
11        {
12            while (true)
13            {
14                Console.WriteLine("Menu:");
15                Console.WriteLine("1. Update");
16                Console.WriteLine("2. Delete");
17                Console.WriteLine("3. View Data");
18                Console.WriteLine("5. Exit");
19                Console.Write("Enter your choice: ");
20
21                string choice = Console.ReadLine();
22
23                switch (choice)
24                {
25                    case "1":
26                        Update();
27                        break;
28                    case "2":
29                        Delete();
30                        break;
```

```
31                    break;
32                    case "4":
33                        Console.WriteLine("Exiting...");
34                        return;
35                default:
36                    Console.WriteLine("Invalid choice. Please try again.");
37                    break;
38            }
39
40            Console.WriteLine();
41        }
42    }
43 }
```

Name:Lalit Chaudhari

```
static void Update()
{
    SqlConnection cn = new SqlConnection();
    cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Module;
Integrated Security=True";
    cn.Open();

    try
    {
        Console.WriteLine("Enter the Student rollno ID to update: ");
        int Rollno = int.Parse(Console.ReadLine());

        Console.WriteLine("Enter the new Name: ");
        string newName = Console.ReadLine();

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "UPDATE students SET name = @NewName WHERE rollno = @Rollno";
        cmd.Parameters.AddWithValue("@NewName", newName);
        cmd.Parameters.AddWithValue("@Rollno", Rollno);

        Console.WriteLine(cmd.ExecuteNonQuery());
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        cn.Close();
    }
}
```

Name:Lalit Chaudhari

```
static void Delete()
{
    SqlConnection cn = new SqlConnection();
    cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Module;
    Integrated Security=True";
    cn.Open();

    try
    {
        Console.WriteLine("Enter the student rollno to delete: ");
        int Rollno = int.Parse(Console.ReadLine());

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "DELETE FROM students WHERE rollno = @Rollno";
        cmd.Parameters.AddWithValue("@Rollno", Rollno);

        Console.WriteLine(cmd.ExecuteNonQuery());
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        cn.Close();
    }
}
```

```
11 static void DataReader()
12 {
13     SqlConnection cn = new SqlConnection();
14     cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Module;
15     Integrated Security=True";
16     cn.Open();
17     try
18     {
19         SqlCommand cmd = new SqlCommand();
20         cmd.Connection = cn;
21         cmd.CommandType = CommandType.Text;
22         cmd.CommandText = "select * from students";
23
24         SqlDataReader dr = cmd.ExecuteReader();
25
26         while (dr.Read())
27         {
28             Console.WriteLine(dr["rollno"] + " ");
29             Console.WriteLine(dr["name"] + " ");
30             Console.WriteLine(dr["marks"] + " ");
31
32             Console.WriteLine();
33         }
34         dr.Close();
35     }
36     catch (Exception ex)
37     {
38     }
39 }
40
```

Name:Lalit Chaudhari

```
138     }
139     catch (Exception ex)
140     {
141     }
142     Console.WriteLine(ex.Message);
143 }
144 finally
145 {
146     cn.Close();
147 }
148 }
149 }
150
151 0 references
152 class Students
153 {
154     0 references
155     public string name { get; set; }
156     0 references
157     public int rollno { get; set; }
158     0 references
159     public int marks { get; set; }
160 }
```

Update and read

```
Menu:
1. Update
2. Delete
3. View Data
5. Exit
Enter your choice: 1
Enter the Student rollno ID to update: 3
Enter the new Name: Rajesh
1

Menu:
1. Update
2. Delete
3. View Data
5. Exit
Enter your choice: 3
1 Lalit 85
2 Nikhil 90
3 Rajesh 83

Menu:
1. Update
2. Delete
3. View Data
5. Exit
Enter your choice:
E:\CDACTVM\NET\Lab\Moduleend\Moduleend\MenuDrivencode\bin\Debug\net7.0\MenuDrivencode.exe (process 21364) exited with code -1.
```

Name:Lalit Chaudhari

Delete and read

```
Menu:
1. Update
2. Delete
3. View Data
5. Exit
Enter your choice: 2
Enter the student rollno to delete: 3
1

Menu:
1. Update
2. Delete
3. View Data
5. Exit
Enter your choice: 3
1 Lalit 85
2 Nikhil 90

Menu:
1. Update
2. Delete
3. View Data
5. Exit
Enter your choice:
```

```
using Microsoft.Data.SqlClient;
using System.Data;

namespace MenuDrivenDatabaseProgram
{
    internal class Program
    {
        static void Main()
        {
            while (true)
            {
                Console.WriteLine("Menu:");
                Console.WriteLine("1. Update");
                Console.WriteLine("2. Delete");
                Console.WriteLine("3. View Data");
                Console.WriteLine("5. Exit");
                Console.Write("Enter your choice: ");

                string choice = Console.ReadLine();

                switch (choice)
                {
                    case "1":
                        Update();
                        break;
                    case "2":
                        Delete();
                        break;
                    case "3":
                        DataReader();
                        break;
                    case "4":
                        Console.WriteLine("Exiting...");
                        return;
                    default:
                        Console.WriteLine("Invalid choice. Please try again.");
                        break;
                }

                Console.WriteLine();
            }
        }

        static void Update()
        {
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Module;
```

Name:Lalit Chaudhari

```
Integrated Security=True";
cn.Open();

try
{
    Console.Write("Enter the Student rollno ID to update: ");
    int Rollno = int.Parse(Console.ReadLine());

    Console.Write("Enter the new Name: ");
    string newName = Console.ReadLine();

    SqlCommand cmd = new SqlCommand();
    cmd.Connection = cn;
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "UPDATE students SET name = @NewName WHERE rollno = @Rollno";
    cmd.Parameters.AddWithValue("@NewName", newName);
    cmd.Parameters.AddWithValue("@Rollno", Rollno);

    Console.WriteLine(cmd.ExecuteNonQuery());

}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    cn.Close();
}
}

static void Delete()
{
    SqlConnection cn = new SqlConnection();
    cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Module;
Integrated Security=True";
    cn.Open();

    try
    {
        Console.Write("Enter the student rollno to delete: ");
        int Rollno = int.Parse(Console.ReadLine());

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "DELETE FROM students WHERE rollno = @Rollno";
        cmd.Parameters.AddWithValue("@Rollno", Rollno);

        Console.WriteLine(cmd.ExecuteNonQuery());

    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        cn.Close();
    }
}

static void DataReader()
{
    SqlConnection cn = new SqlConnection();
    cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Module;
Integrated Security=True";
    cn.Open();
    try
    {
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = cn;
        cmd.CommandType = CommandType.Text;
```

Name:Lalit Chaudhari

```
cmd.CommandText = "select * from students";

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())
{
    Console.Write(dr["rollno"] + " ");
    Console.Write(dr["name"] + " ");
    Console.Write(dr["marks"] + " ");

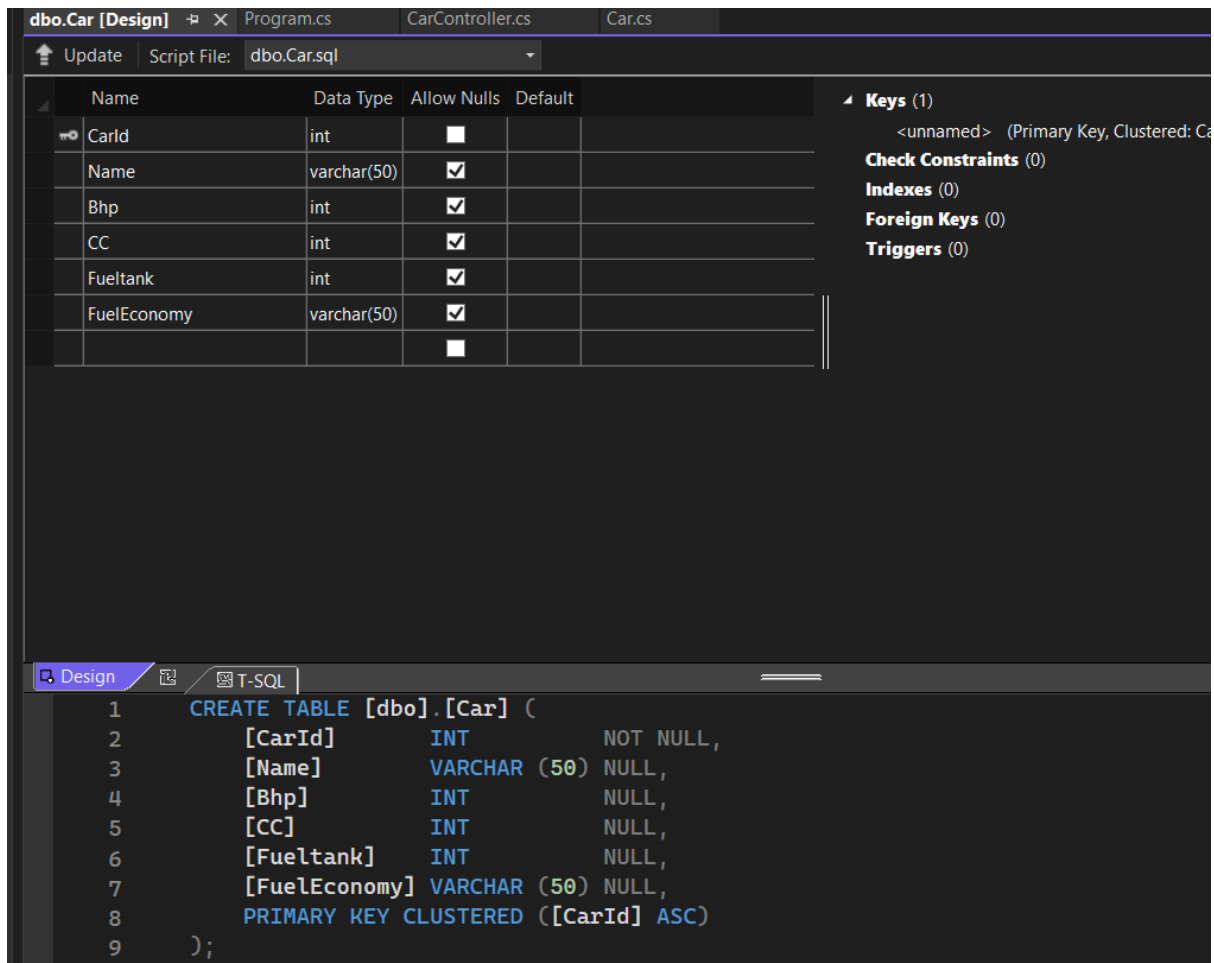
    Console.WriteLine();
}
dr.Close();

}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    cn.Close();
}
}

class Students
{
    public string name { get; set; }
    public int rollno { get; set; }
    public int marks { get; set; }
}
}
```


Name:Lalit Chaudhari

2. Create a model class named Car. Add an Index by using model binding, perform create, read update and Delete operations for details such as CarId, Name, bhp, cc, fuel tank and fuelEconomy.



dbo.Car [Design] | Program.cs | CarController.cs | Car.cs

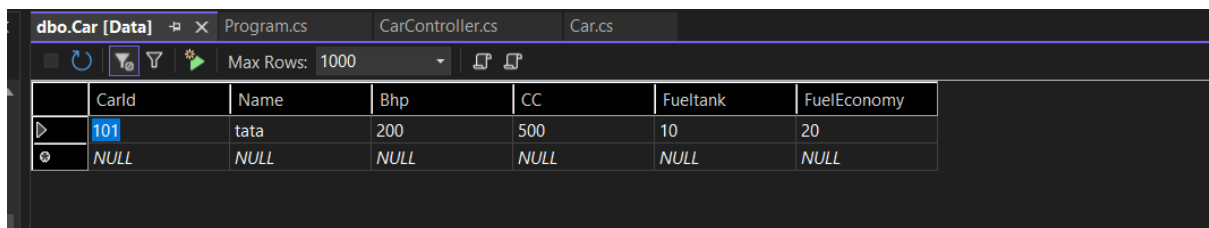
Update | Script File: dbo.Car.sql

Name	Data Type	Allow Nulls	Default
CarId	int	<input type="checkbox"/>	
Name	varchar(50)	<input checked="" type="checkbox"/>	
Bhp	int	<input checked="" type="checkbox"/>	
CC	int	<input checked="" type="checkbox"/>	
Fueltank	int	<input checked="" type="checkbox"/>	
FuelEconomy	varchar(50)	<input checked="" type="checkbox"/>	

Keys (1)
<unnamed> (Primary Key, Clustered: Ca)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Design | T-SQL

```
1 CREATE TABLE [dbo].[Car] (  
2     [CarId] INT NOT NULL,  
3     [Name] VARCHAR (50) NULL,  
4     [Bhp] INT NULL,  
5     [CC] INT NULL,  
6     [Fueltank] INT NULL,  
7     [FuelEconomy] VARCHAR (50) NULL,  
8     PRIMARY KEY CLUSTERED ([CarId] ASC)  
9 );
```

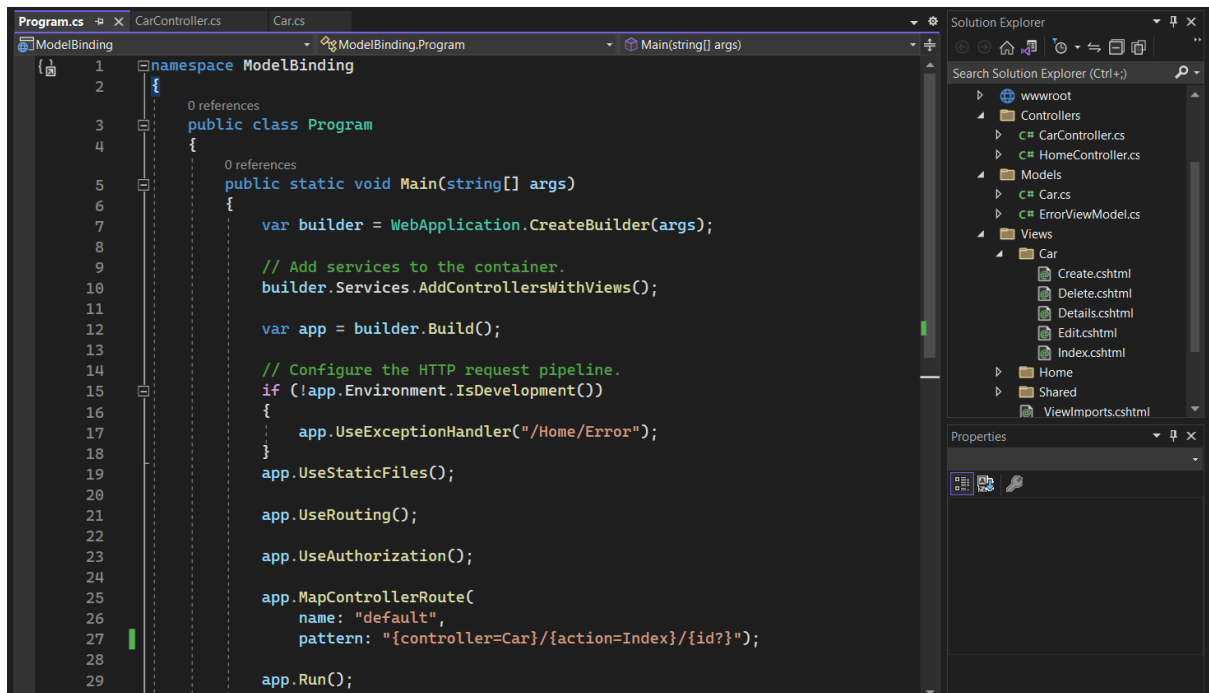


dbo.Car [Data] | Program.cs | CarController.cs | Car.cs

Max Rows: 1000

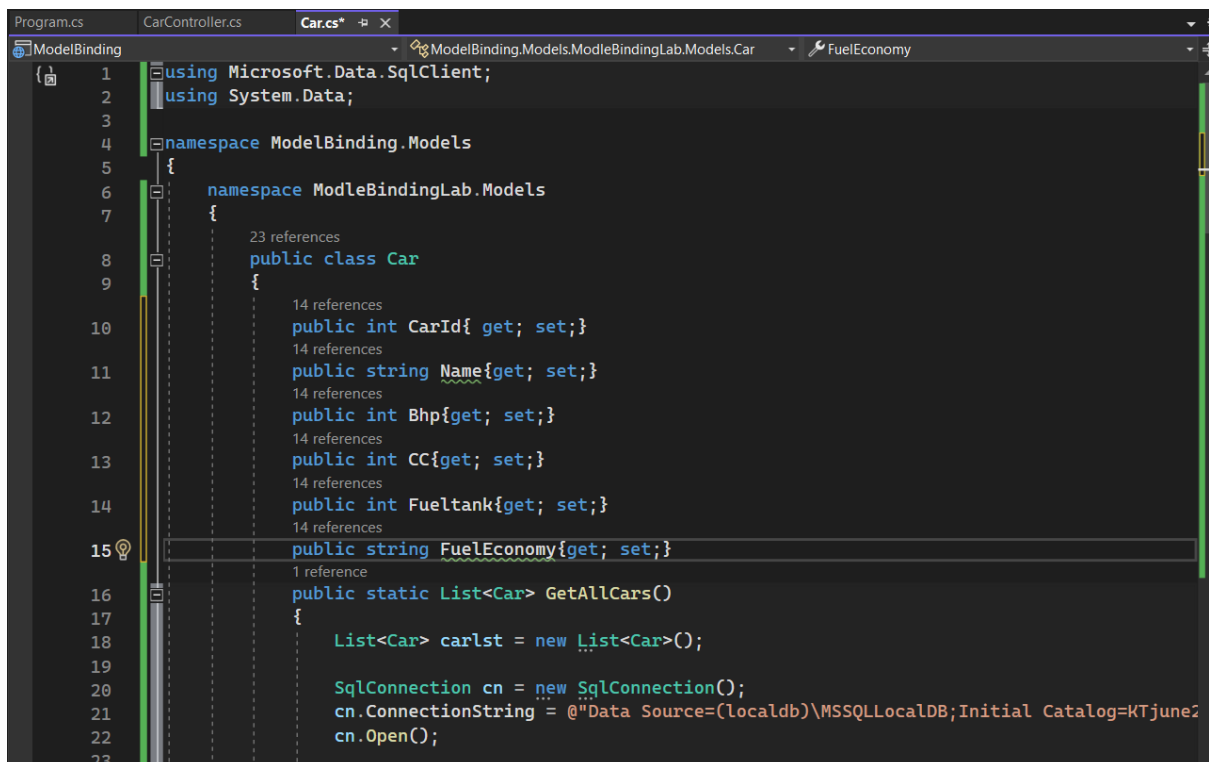
	CarId	Name	Bhp	CC	Fueltank	FuelEconomy
▶	101	tata	200	500	10	20
⊕	NULL	NULL	NULL	NULL	NULL	NULL

Name:Lalit Chaudhari



This screenshot shows the Visual Studio IDE with the `Program.cs` file open. The code defines a `Program` class with a `Main` method that sets up a web application. The `ModelBinding` namespace is shown in the Solution Explorer on the right.

```
1 namespace ModelBinding
2 {
3     public class Program
4     {
5         public static void Main(string[] args)
6         {
7             var builder = WebApplication.CreateBuilder(args);
8
9             // Add services to the container.
10            builder.Services.AddControllersWithViews();
11
12            var app = builder.Build();
13
14            // Configure the HTTP request pipeline.
15            if (!app.Environment.IsDevelopment())
16            {
17                app.UseExceptionHandler("/Home/Error");
18            }
19            app.UseStaticFiles();
20
21            app.UseRouting();
22
23            app.UseAuthorization();
24
25            app.MapControllerRoute(
26                name: "default",
27                pattern: "{controller=Car}/{action=Index}/{id?}");
28
29            app.Run();
30        }
31    }
32 }
```



This screenshot shows the Visual Studio IDE with the `Car.cs` file open. The code defines a `Car` class with properties for `CarId`, `Name`, `Bhp`, `CC`, `FuelTank`, and `FuelEconomy`. It also includes a `GetAllCars` method that connects to a database. The `ModelBinding.Models` namespace is shown in the Solution Explorer on the right.

```
1 using Microsoft.Data.SqlClient;
2 using System.Data;
3
4 namespace ModelBinding.Models
5 {
6     namespace ModelBinding.Models
7     {
8         public class Car
9         {
10             public int CarId { get; set; }
11             public string Name { get; set; }
12             public int Bhp { get; set; }
13             public int CC { get; set; }
14             public int FuelTank { get; set; }
15             public string FuelEconomy { get; set; }
16
17             public static List<Car> GetAllCars()
18             {
19                 List<Car> carList = new List<Car>();
20
21                 SqlConnection cn = new SqlConnection();
22                 cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=KTjune2";
23                 cn.Open();
24             }
25         }
26     }
27 }
```

Name:Lalit Chaudhari

```
Program.cs  CarController.cs  Car.cs*  X
ModelBinding  ModelBinding.Models.ModleBindingLab.Models.Car  GetAllCars()

24      try
25      {
26          SqlCommand cmd = new SqlCommand();
27          cmd.Connection = cn;
28          cmd.CommandType = CommandType.Text;
29          cmd.CommandText = "SELECT * FROM Car";
30
31          SqlDataReader dr = cmd.ExecuteReader();
32
33          while (dr.Read())
34          {
35              Car cr = new Car();
36              {
37                  cr.CarId = Convert.ToInt32(dr["CarId"]);
38                  cr.Name = dr["Name"].ToString();
39                  cr.Bhp = Convert.ToInt32(dr["Bhp"]);
40                  cr.CC = Convert.ToInt32(dr["CC"]);
41                  cr.Fueltank = Convert.ToInt32(dr["Bhp"]);
42                  cr.FuelEconomy = dr["Name"].ToString();
43              };
44
45              carlst.Add(cr);
46          }
47          foreach (Car c in carlst)
48          {
49              Console.WriteLine($"CarId: {c.CarId}");
50              Console.WriteLine($"Name: {c.Name}");
51              Console.WriteLine($"Bhp: {c.Bhp}");
52              Console.WriteLine($"CC: {c.CC}");
53              Console.WriteLine($"Fueltank: {c.Fueltank}");
54              Console.WriteLine($"FuelEconomy: {c.FuelEconomy}");
```

```
Program.cs  CarController.cs  Car.cs*  X
ModelBinding  ModelBinding.Models.ModleBindingLab.Models.Car  GetAllCars()

48      {
49          Console.WriteLine($"CarId: {c.CarId}");
50          Console.WriteLine($"Name: {c.Name}");
51          Console.WriteLine($"Bhp: {c.Bhp}");
52          Console.WriteLine($"CC: {c.CC}");
53          Console.WriteLine($"Fueltank: {c.Fueltank}");
54          Console.WriteLine($"FuelEconomy: {c.FuelEconomy}");
55
56          Console.WriteLine();
57      }
58      dr.Close();
59  }
60  catch (Exception ex)
61  {
62      Console.WriteLine(ex.Message);
63  }
64  finally
65  {
66      cn.Close();
67  }
68
69  return carlst;
70  }
71  }
72  }
73  }
74  }
```

Name:Lalit Chaudhari

```
1 using Microsoft.AspNetCore.Http;
2 using Microsoft.AspNetCore.Mvc;
3 using ModelBinding.Models.ModelBindingLab.Models;
4
5 namespace ModelBinding.Controllers
6 {
7     0 references
8     public class CarController : Controller
9     {
10         // GET: CarController
11         3 references
12         public ActionResult Index()
13         {
14             List<Car> lstcar = Car.GetAllCars();
15             return View(lstcar);
16         }
17
18         // GET: CarController/Details/5
19         0 references
20         public ActionResult Details(int id)
21         {
22             return View();
23         }
24
25         // GET: CarController/Create
26         0 references
27         public ActionResult Create()
28         {
29             return View();
30         }
31     }
32 }
```

```
22
23 // GET: CarController/Create
24 0 references
25 public ActionResult Create()
26 {
27     return View();
28 }
29
30 // POST: CarController/Create
31 [HttpPost]
32 [ValidateAntiForgeryToken]
33 0 references
34 public ActionResult Create(IFormCollection collection)
35 {
36     try
37     {
38         return RedirectToAction(nameof(Index));
39     }
40     catch
41     {
42         return View();
43     }
44 }
45
46 // GET: CarController/Edit/5
47 0 references
48 public ActionResult Edit(int id)
49 {
50     Car obj = Car.GetSingleCar(id);
51     return View(obj);
52 }
```

Name:Lalit Chaudhari

```
ModelBinding
Program.cs
CarController.cs
Car.cs
ModelBinding.Controllers.CarController
Delete(int id, Car obj)

50
51 // POST: CarController/Edit/5
52 [HttpPost]
53 [ValidateAntiForgeryToken]
54 0 references
55 public ActionResult Edit(Car cr, IFormCollection collection)
56 {
57     try
58     {
59         Car.Update(cr);
60         return RedirectToAction(nameof(Index));
61     }
62     catch
63     {
64         return View();
65     }
66 }
67
68 // GET: CarController/Delete/5
69 0 references
70 public ActionResult Delete(int id)
71 {
72     Car obj = Car.GetSingleCar(id);
73     return View(obj);
74 }
75
76 // POST: CarController/Delete/5
77 [HttpPost]
78 [ValidateAntiForgeryToken]
79 0 references
```

```
80
81 // GET: CarController/Delete/5
82 0 references
83 public ActionResult Delete(int id)
84 {
85     Car obj = Car.GetSingleCar(id);
86     return View(obj);
87 }
88
89 // POST: CarController/Delete/5
90 [HttpPost]
91 [ValidateAntiForgeryToken]
92 0 references
93 public ActionResult Delete(int id, Car obj)
94 {
95     try
96     {
97         Car.DeleteCar(id);
98         return RedirectToAction(nameof(Index));
99     }
100    catch
101    {
102        return View();
103    }
104 }
105 }
```

Name:Lalit Chaudhari

Index - ModelBinding

localhost:5182

ModelBinding Home Privacy

Index

[Create New](#)

CarId	Name	Bhp	CC	FuelTank	FuelEconomy	
101	tata	200	500	200	tata	Edit Details Delete

Edit

Car

CarId

101

Name

tata

Bhp

200

CC

500

FuelTank

10

FuelEconomy

20

Save

[Back to List](#)

ModelBinding Home Privacy

Delete

Are you sure you want to delete this?

Car

CarId	101
Name	tata
Bhp	200
CC	500
FuelTank	10
FuelEconomy	20

Delete

[Back to List](#)