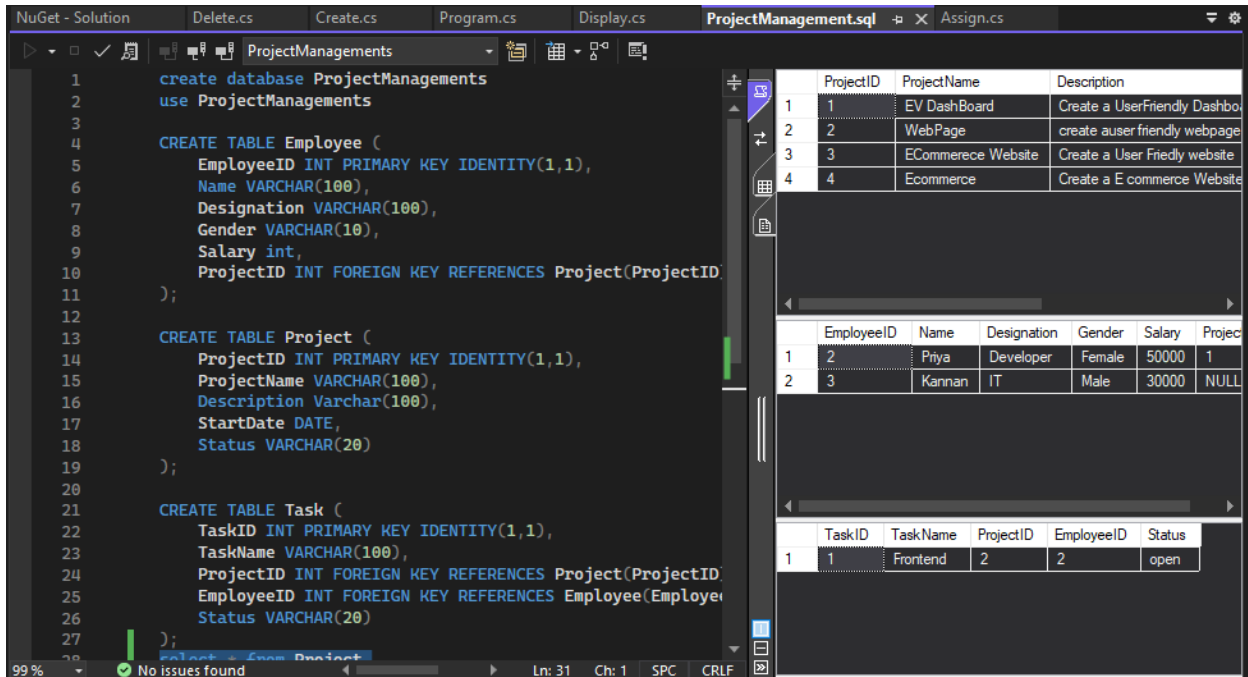


CASE STUDY PROJECT MANAGEMENT

CREATE DATABASE SCHEMA :



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'ProjectManagements' database schema. The right pane shows the 'ProjectManagement.sql' script, which contains the following SQL code:

```
1 create database ProjectManagements
2 use ProjectManagements
3
4 CREATE TABLE Employee (
5     EmployeeID INT PRIMARY KEY IDENTITY(1,1),
6     Name VARCHAR(100),
7     Designation VARCHAR(100),
8     Gender VARCHAR(10),
9     Salary int,
10    ProjectID INT FOREIGN KEY REFERENCES Project(ProjectID)
11 );
12
13 CREATE TABLE Project (
14     ProjectID INT PRIMARY KEY IDENTITY(1,1),
15     ProjectName VARCHAR(100),
16     Description Varchar(100),
17     StartDate DATE,
18     Status VARCHAR(20)
19 );
20
21 CREATE TABLE Task (
22     TaskID INT PRIMARY KEY IDENTITY(1,1),
23     TaskName VARCHAR(100),
24     ProjectID INT FOREIGN KEY REFERENCES Project(ProjectID)
25     EmployeeID INT FOREIGN KEY REFERENCES Employee(EmployeeID)
26     Status VARCHAR(20)
27 );
```

The right pane also displays three data tables:

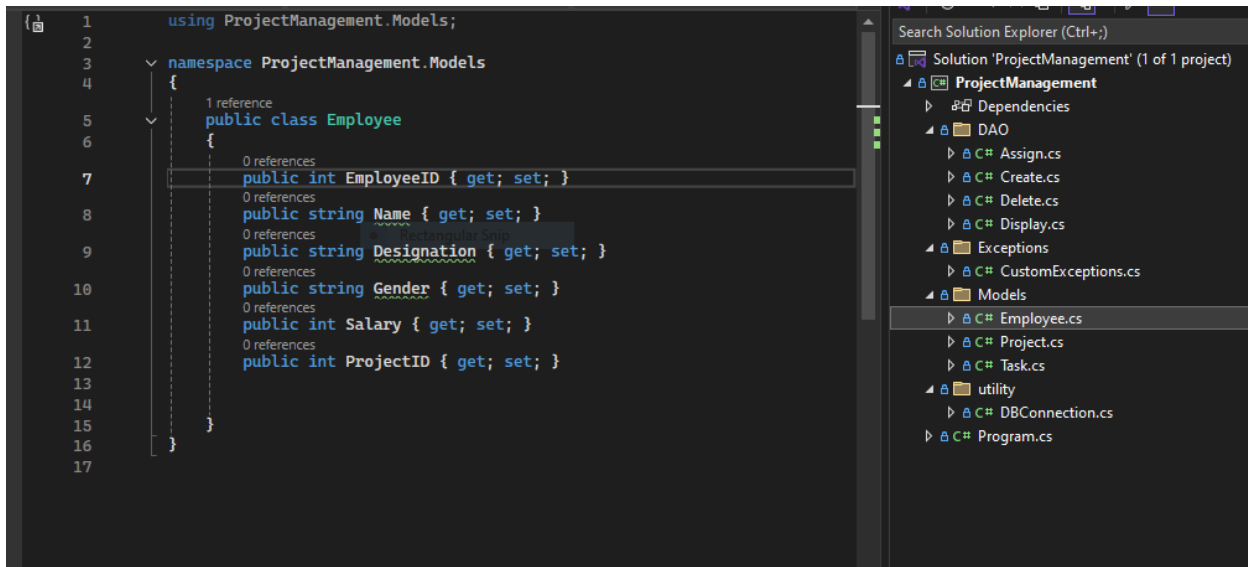
ProjectID	ProjectName	Description
1	EV DashBoard	Create a UserFriendly Dashbo
2	WebPage	create auser friendly webpage
3	ECommerce Website	Create a User Friedly website
4	Ecommerce	Create a E commerce Website

EmployeeID	Name	Designation	Gender	Salary	Project
1	Priya	Developer	Female	50000	1
2	Kannan	IT	Male	30000	NULL

TaskID	TaskName	ProjectID	EmployeeID	Status
1	Frontend	2	2	open

CREATE MODELS:

1. Employee

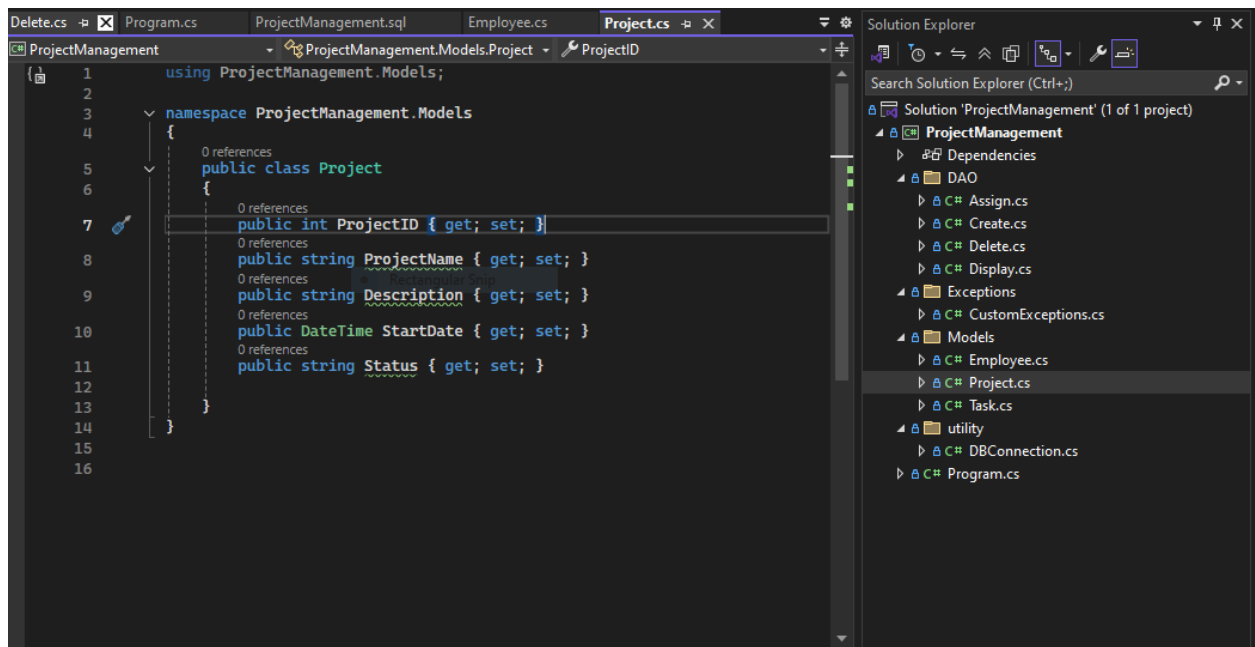


The screenshot shows the Visual Studio IDE. The left pane displays the C# code for the 'Employee' model class, which is part of the 'ProjectManagement.Models' namespace. The code is as follows:

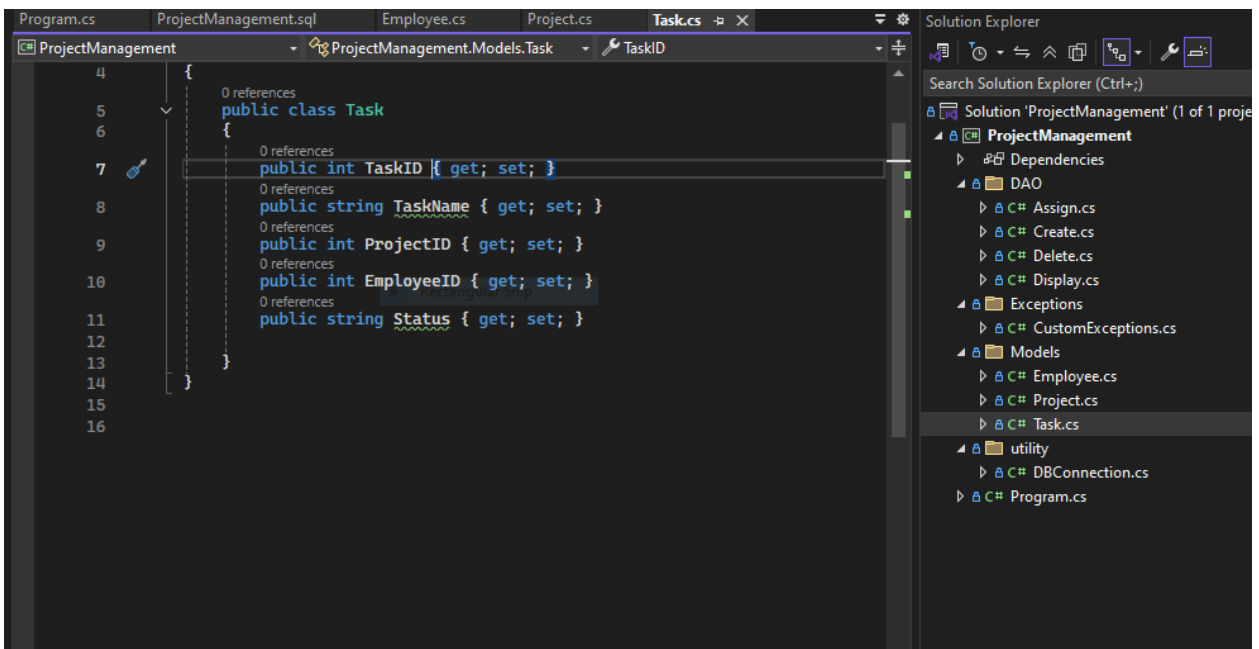
```
1 using ProjectManagement.Models;
2
3 namespace ProjectManagement.Models
4 {
5     1 reference
6     public class Employee
7     {
8         0 references
9         public int EmployeeID { get; set; }
10        0 references
11        public string Name { get; set; }
12        0 references
13        public string Designation { get; set; }
14        0 references
15        public string Gender { get; set; }
16        0 references
17        public int Salary { get; set; }
18        0 references
19        public int ProjectID { get; set; }
20    }
21 }
```

The right pane shows the 'Solution Explorer' with the 'ProjectManagement' solution. The 'Models' folder is expanded, showing the 'Employee.cs' file.

2. Project



3. Task



CREATE UTILITY

1. DBConnection

```
ProjectManagement
  ProjectManagement.Utility.DBConnection
    GetConnection()
1  using System.Data.SqlClient;
2
3  namespace ProjectManagement.Utility
4  {
5      11 references
6      public static class DBConnection
7      {
8          11 references
9          private static readonly string connectionString = "Server=(localdb)\\MSSQLLocalDB;Database=ProjectMan
10         public static SqlConnection GetConnection()
11         {
12             return new SqlConnection(connectionString);
13         }
14     }
```

CREATE DAO

1. Assign

```
using ProjectManagement.Exceptions;
using ProjectManagement.Models;
using ProjectManagement.Utility;
using System;
using System.Data.SqlClient;
using System.Threading.Tasks;
```

```
namespace ProjectManagement.DAO
```

```
{
    public class Assign
    {
        public static void AssignProject()
        {
```

```
            Console.WriteLine("Enter Employee ID to Assign Project: ");
            int employeeid = int.Parse(Console.ReadLine());
```

```
            Console.WriteLine("Enter ProjectID: ");
            int projectid = int.Parse(Console.ReadLine());
```

```
            using SqlConnection conn = DBConnection.GetConnection();
            string query = "UPDATE Employee SET ProjectID = @projectid WHERE EmployeeID = @employeeid";
```

```
            using SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@employeeid", employeeid);
            cmd.Parameters.AddWithValue("@projectid", projectid);
```

```
            conn.Open();
```

```
            int result = cmd.ExecuteNonQuery();
            Console.WriteLine($"{result} Project Assigned Successfully.");
```

```

    }
    public static void AssignTask()
    {
        Console.WriteLine("Enter Task ID to Assign Project and Employee: ");
        int taskid = int.Parse(Console.ReadLine());

        Console.WriteLine("Enter ProjectID: ");
        int projectid = int.Parse(Console.ReadLine());

        Console.WriteLine("Enter EmployeeID: ");
        int employeeid = int.Parse(Console.ReadLine());

        using SqlConnection conn = DBConnection.GetConnection();
        string query = "UPDATE Task SET ProjectID = @projectid, EmployeeID = @employeeid WHERE TaskID = @taskid";

        using SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@taskid", taskid);
        cmd.Parameters.AddWithValue("@employeeid", employeeid);
        cmd.Parameters.AddWithValue("@projectid", projectid);

        conn.Open();
        int result = cmd.ExecuteNonQuery();
        Console.WriteLine($"{result} Task Assigned Successfully.");
    }
}

```

2. Create

```

using ProjectManagement.Models;
using ProjectManagement.Utility;
using System;
using System.Data.SqlClient;

namespace ProjectManagement.DAO
{
    public class Create
    {
        public static void CreateEmployee()
        {
            Console.WriteLine("Enter Full Name: ");
            string name = Console.ReadLine();

            Console.WriteLine("Enter Designation: ");
            string designation = Console.ReadLine();

            Console.WriteLine("Enter Gender: ");
            string gender = Console.ReadLine();

            Console.WriteLine("Enter Salary: ");
            int salary = int.Parse(Console.ReadLine());

            int EmployeeID = 0;

```

```

using SqlConnection conn = DBConnection.GetConnection();

// Insert Employee
string studentQuery = "INSERT INTO Employee (Name,Designation,Gender,Salary) OUTPUT
INSERTED.EmployeeID VALUES (@name, @designation,@gender,@salary)";
using SqlCommand cmd = new SqlCommand(studentQuery, conn);
cmd.Parameters.AddWithValue("@name", name);
cmd.Parameters.AddWithValue("@designation", designation);
cmd.Parameters.AddWithValue("@gender", gender);
cmd.Parameters.AddWithValue("@salary", salary);

conn.Open();
EmployeeID = (int)cmd.ExecuteScalar();
conn.Close();
}
public static void CreateProject()
{
    Console.Write("Enter ProjectName: ");
    string projectname = Console.ReadLine();

    Console.Write("Enter Description: ");
    string description = Console.ReadLine();

    Console.Write("Enter Start Date (YYYY-MM-DD): ");
    string startdate = Console.ReadLine();

    Console.Write("Enter Status: ");
    string status = Console.ReadLine();

    int ProjectID = 0;

    using SqlConnection conn = DBConnection.GetConnection();

    // Insert Project
    string studentQuery = "INSERT INTO Project ( ProjectName,Description,StartDate,Status) OUTPUT
INSERTED.ProjectID VALUES (@projectname, @description,@startdate,@status)";
    using SqlCommand cmd = new SqlCommand(studentQuery, conn);
    cmd.Parameters.AddWithValue("@projectname", projectname);
    cmd.Parameters.AddWithValue("@description", description);
    cmd.Parameters.AddWithValue("@startdate", startdate);
    cmd.Parameters.AddWithValue("@status", status);

    conn.Open();
    ProjectID = (int)cmd.ExecuteScalar();
    conn.Close();
}
public static void CreateTask()
{
    Console.Write("Enter Task: ");
    string taskname = Console.ReadLine();

    //Console.Write("Enter ProjectID: ");
    //int projectid = int.Parse(Console.ReadLine());

    //Console.Write("Enter EmployeeID: ");
    //int employeeid = int.Parse(Console.ReadLine());

    Console.Write("Enter Status: ");

```

```

string status = Console.ReadLine();

int TaskID = 0;

using SqlConnection conn = DBConnection.GetConnection();

// Insert Task
string studentQuery = "INSERT INTO Task ( TaskName,Status) OUTPUT INSERTED.TaskID VALUES
(@taskname, @status)";

using SqlCommand cmd = new SqlCommand(studentQuery, conn);

cmd.Parameters.AddWithValue("@taskname", taskname);
//cmd.Parameters.AddWithValue("@projectid", projectid);
//cmd.Parameters.AddWithValue("@employeeid", employeeid);
cmd.Parameters.AddWithValue("@status", status);

conn.Open();
TaskID = (int)cmd.ExecuteScalar();
conn.Close();

}

internal bool CreateEmployee(Employee emp)
{
    throw new NotImplementedException();
}
}
}

```

3. Delete

```

using ProjectManagement.Models;
using ProjectManagement.Utility;
using System;
using System.Data.SqlClient;

namespace ProjectManagement.DAO
{
    public class Display
    {
        public static void DisplayEmployee()
        {
            using SqlConnection conn = DBConnection.GetConnection();
            string query = "SELECT * FROM Employee";

            using SqlCommand cmd = new SqlCommand(query, conn);
            conn.Open();
            using SqlDataReader reader = cmd.ExecuteReader();
            Console.WriteLine("\nEmployee Details:");
            while (reader.Read())
            {
                Console.WriteLine($"EmployeeID: {reader["EmployeeID"]}, Name: {reader["Name"]}, Designation:
{reader["Designation"]}, Gender: {reader["Gender"]},Salary: {reader["Salary"]},ProjectID: {reader["ProjectID"]}");
            }
        }
        public static void DisplayProject()
        {

```

```

using SqlConnection con = DBConnection.GetConnection();
string query = "SELECT * FROM Project";

using SqlCommand cmd = new SqlCommand(query, con);
con.Open();
using SqlDataReader reader = cmd.ExecuteReader();
Console.WriteLine("\nProject Details:");
while (reader.Read())
{
    Console.WriteLine($"ProjectID: {reader["ProjectID"]}, ProjectName: {reader["ProjectName"]}, Description: {reader["Description"]}, StartDate: {reader["StartDate"]}, Status: {reader["Status"]}");
}
}
public static void DisplayTask()
{
    using SqlConnection con = DBConnection.GetConnection();
    string query = "SELECT * FROM Task";

    using SqlCommand cmd = new SqlCommand(query, con);
    con.Open();
    using SqlDataReader reader = cmd.ExecuteReader();
    Console.WriteLine("\nTask Details:");
    while (reader.Read())
    {
        Console.WriteLine($"TaskID: {reader["TaskID"]}, TaskName: {reader["TaskName"]}, ProjectID: {reader["ProjectID"]}, EmployeeID: {reader["EmployeeID"]}, Status: {reader["Status"]}");
    }
}
public static void DisplayProjectAssigned()
{
    string query = @"
        SELECT p.ProjectId, p.ProjectName, e.Name, t.TaskName
        FROM Task t
        JOIN Employee e ON e.EmployeeID = t.EmployeeID
        JOIN Project p ON p.ProjectID = t.ProjectID";

    using SqlConnection conn = DBConnection.GetConnection();
    using SqlCommand cmd = new SqlCommand(query, conn);
    conn.Open();
    SqlDataReader reader = cmd.ExecuteReader();
    Console.WriteLine("\nProjects Assigned");
    while (reader.Read())
    {
        Console.WriteLine($"ProjectID: {reader["ProjectID"]}, ProjectName: {reader["ProjectName"]}, Name: {reader["Name"]}, TaskName: {reader["TaskName"]}");
    }
}
}
}

```

4. Display

```

using ProjectManagement.Models;
using ProjectManagement.Utility;
using System;
using System.Data.SqlClient;

```

```

namespace ProjectManagement.DAO
{
    public class Display
    {
        public static void DisplayEmployee()
        {
            using SqlConnection conn = DBConnection.GetConnection();
            string query = "SELECT * FROM Employee";

            using SqlCommand cmd = new SqlCommand(query, conn);
            conn.Open();
            using SqlDataReader reader = cmd.ExecuteReader();
            Console.WriteLine("\nEmployee Details:");
            while (reader.Read())
            {
                Console.WriteLine($"EmployeeID: {reader["EmployeeID"]}, Name: {reader["Name"]}, Designation: {reader["Designation"]}, Gender: {reader["Gender"]}, Salary: {reader["Salary"]}, ProjectID: {reader["ProjectID"]}");
            }
        }

        public static void DisplayProject()
        {
            using SqlConnection con = DBConnection.GetConnection();
            string query = "SELECT * FROM Project";

            using SqlCommand cmd = new SqlCommand(query, con);
            con.Open();
            using SqlDataReader reader = cmd.ExecuteReader();
            Console.WriteLine("\nProject Details:");
            while (reader.Read())
            {
                Console.WriteLine($"ProjectID: {reader["ProjectID"]}, ProjectName: {reader["ProjectName"]}, Description: {reader["Description"]}, StartDate: {reader["StartDate"]}, Status: {reader["Status"]}");
            }
        }

        public static void DisplayTask()
        {
            using SqlConnection con = DBConnection.GetConnection();
            string query = "SELECT * FROM Task";

            using SqlCommand cmd = new SqlCommand(query, con);
            con.Open();
            using SqlDataReader reader = cmd.ExecuteReader();
            Console.WriteLine("\nTask Details:");
            while (reader.Read())
            {
                Console.WriteLine($"TaskID: {reader["TaskID"]}, TaskName: {reader["TaskName"]}, ProjectID: {reader["ProjectID"]}, EmployeeID: {reader["EmployeeID"]}, Status: {reader["Status"]}");
            }
        }

        public static void DisplayProjectAssigned()
        {
            string query = @"
                SELECT p.ProjectId, p.ProjectName, e.Name, t.TaskName
                FROM Task t
                JOIN Employee e ON e.EmployeeID = t.EmployeeID
                JOIN Project p ON p.ProjectID = t.ProjectID";

            using SqlConnection conn = DBConnection.GetConnection();
            using SqlCommand cmd = new SqlCommand(query, conn);

```

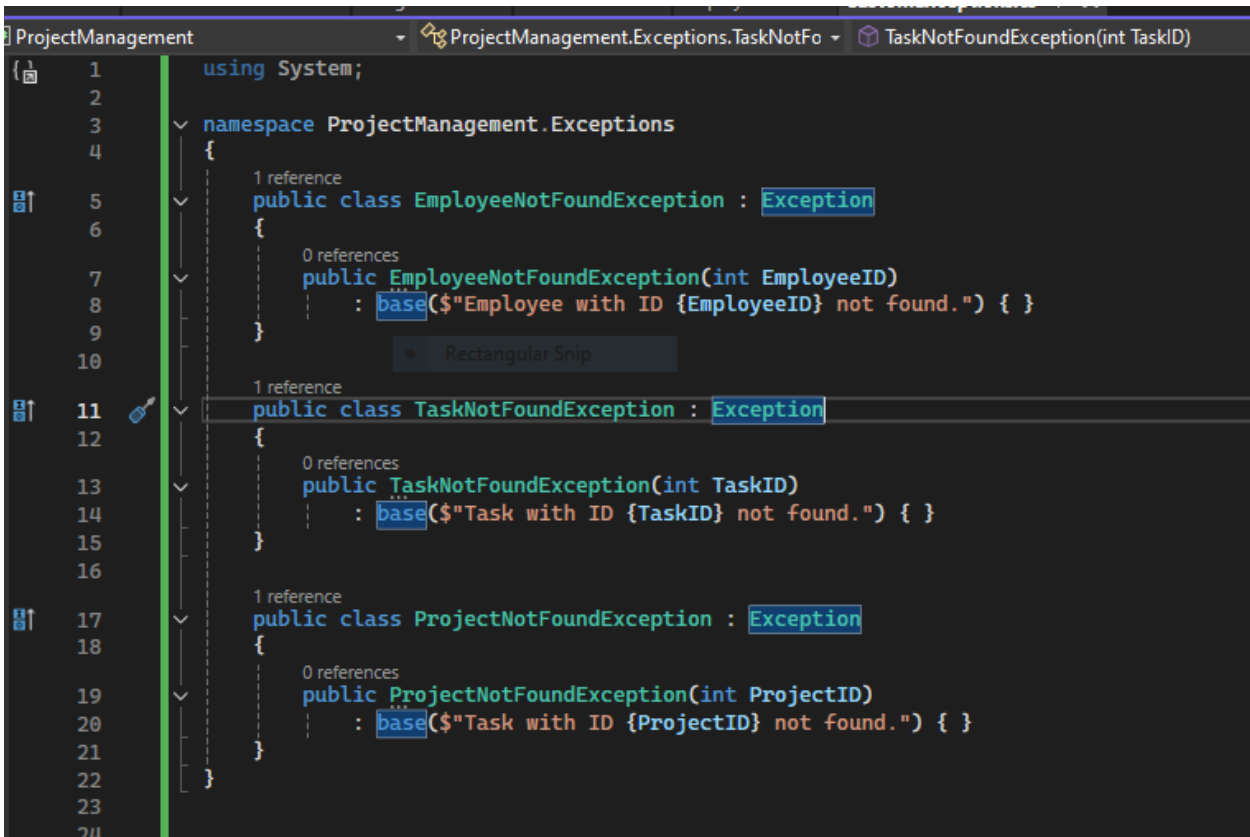


```

conn.Open();
SqlDataReader reader = cmd.ExecuteReader();
Console.WriteLine("\nProjects Assigned");
while (reader.Read())
{
    Console.WriteLine($"ProjectID: {reader["ProjectID"]}, ProjectName: {reader["ProjectName"]}, Name:
{reader["Name"]}, TaskName: {reader["TaskName"]}");
}
}
}
}

```

CREATE EXCEPTIONS



MAIN PROGRAM

```

using ProjectManagement.DAO;

namespace ProjectManagement
{
    public class MainProgram
    {
        static void Main()
        {
            bool exit = false;

            while (!exit)

```

```

{
    Console.WriteLine("\n----- MENU -----");
    Console.WriteLine("1. Add Project");
    Console.WriteLine("2. Add Employee");
    Console.WriteLine("3. Add Task");
    Console.WriteLine("4. Assign project to employee ");
    Console.WriteLine("5. Assign Task");
    Console.WriteLine("6. Display Employee Details");
    Console.WriteLine("7. Display Project Details");
    Console.WriteLine("8. Display Task Details");
    Console.WriteLine("9. Delete Employee");
    Console.WriteLine("10. Delete Task");
    Console.WriteLine("11. List all projects assigned with tasks to an employee ");

    Console.WriteLine("12. Exit");

    Console.Write("Enter your choice: ");
    string choice = Console.ReadLine();

    switch (choice)
    {
        case "1":
            Create.CreateProject();
            Console.WriteLine("Project Created SucessFully");
            break;
        case "2":
            Create.CreateEmployee();
            Console.WriteLine("Employee Data Created SucessFully");
            break;
        case "3":
            Create.CreateTask();
            Console.WriteLine("Task Created Sucessfully");
            break ;
        case "4":
            Assign.AssignProject();
            break;
        case "5":
            Assign.AssignTask();
            break;
        case "6":
            Display.DisplayEmployee();
            break;
        case "7":
            Display.DisplayProject();
            break;
        case "8":
            Display.DisplayTask();
            break;
        case "9":
            Delete.DeleteEmployee();
            break;
        case "10":
            Delete.DeleteTask();
            break;
        case "11":
            Display.DisplayProjectAssigned();
            break;
        case "12":
    
```

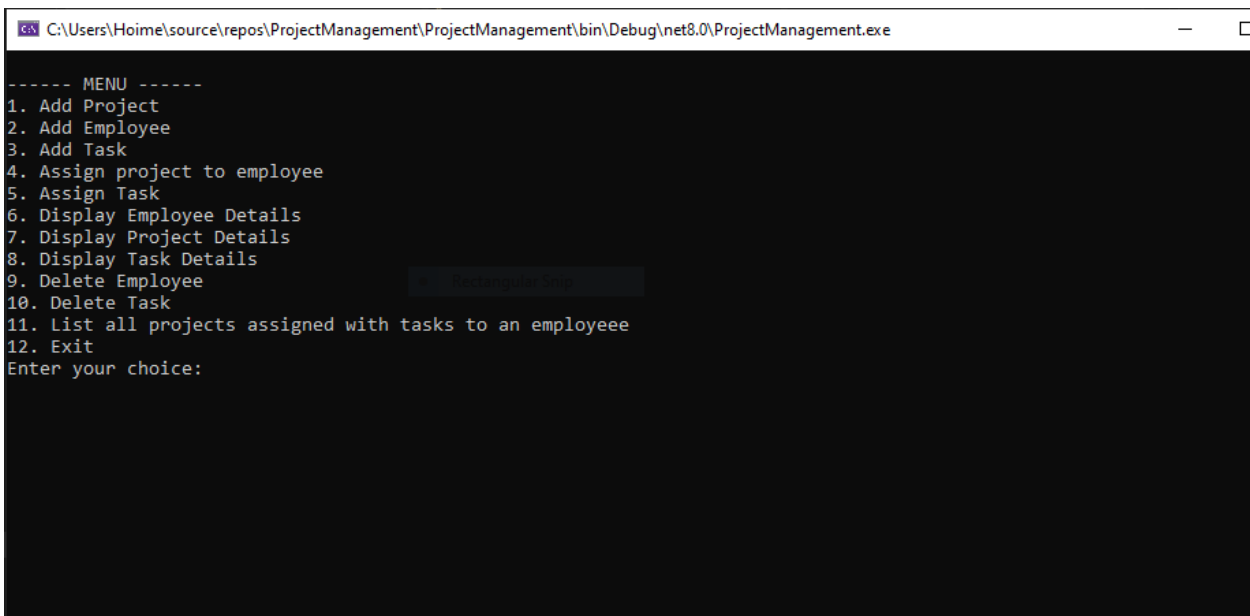
```

        exit = true;
        break;

default:
    Console.WriteLine("Invalid choice. Try again.");
    break;
    }
    }
    }
}
}
}

```

OUTPUT



```

C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe
----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice:

```

1. Add Project

C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectMar

```
----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 1
Enter ProjectName: DashBoard
Enter Description: Create a Vibrant Dash Board
Enter Start Date (YYYY-MM-DD): 2025.07.01
Enter Status: Started
Project Created SucessFully
```

2. Add Employee

C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectMar

```
Project Created SucessFully
----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 2
Enter Full Name: Kumar
Enter Designation: Developer
Enter Gender: Male
Enter Salary: 70000
Employee Data Created SucessFully
```

3. Add Task

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManag

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employeee
12. Exit
Enter your choice: 3
Enter Task: Frontend
Enter Status: started
Task Created Sucessfully
```

4. Assign Project to Employee

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employeee
12. Exit
Enter your choice: 4
Enter Employee ID to Assign Project: 5
Enter ProjectID: 5
1 Project Assigned Sucessfully.
```

5. Assign Task

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\Pro
----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 5
Enter Task ID to Assign Project and Employee: 5
Enter ProjectID: 5
Enter EmployeeID: 5
1 Task Assigned Sucessfully.
```

6. Display Employee Details

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe
1 Task Assigned Sucessfully.
----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 6
Employee Details:
EmployeeID: 2, Name: Priya, Designation: Developer, Gender: Female,Salary: 50000,ProjectID: 1
EmployeeID: 3, Name: Kannan, Designation: IT, Gender: Male,Salary: 30000,ProjectID:
EmployeeID: 5, Name: Kumar, Designation: Developer, Gender: Male,Salary: 70000,ProjectID: 5
```

7. Display Project Details

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 7

Project Details:
ProjectID: 1, ProjectName: EV DashBoard,Description: Create a UserFriendly Dashboard for E vehicles, StartDate: 5/2/2025
12:00:00 AM,Status: Assigned
ProjectID: 2, ProjectName: WebPage,Description: create auser friendly webpage, StartDate: 1/30/2025 12:00:00 AM,Status:
completed
ProjectID: 3, ProjectName: ECommerce Website,Description: Create a User Friedly website, StartDate: 6/30/2025 12:00:00
AM,Status: Open
ProjectID: 4, ProjectName: Ecommerce,Description: Create a E commerce Website, StartDate: 3/23/2025 12:00:00 AM,Status:
NotCompleted
ProjectID: 5, ProjectName: DashBoard,Description: Create a Vibrant Dash Board, StartDate: 7/1/2025 12:00:00 AM,Status: S
tarted

----- MENU -----
```

8. Display Task Details

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 8

Task Details:
TaskID: 1, TaskName: Frontend,ProjectID: 2, EmployeeID: 2,Status: open
TaskID: 5, TaskName: Frontend ,ProjectID: 5, EmployeeID: 5,Status: started

----- MENU -----
```

9. Delete Employee

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe
TaskID: 5, TaskName: Frontend ,ProjectID: 5, EmployeeID: 5,Status: started

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 9
Enter Employee ID to delete: 3
Employee deleted.
```

10. Delete Task

```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe
Employee deleted.

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employee
12. Exit
Enter your choice: 10
Enter Task ID to delete: 1
Task deleted.
```

11. List all projects assigned with tasks to an employee


```
C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe
Enter your choice: 10
Enter Task ID to delete: 1
Task deleted.

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employeee
12. Exit
Enter your choice: 11

Projects Assigned
ProjectID: 5, ProjectName: DashBoard, Name: Kumar, TaskName: Frontend

----- MENU -----
```

12. Exit

```
Microsoft Visual Studio Debug Console
Projects Assigned
ProjectID: 5, ProjectName: DashBoard, Name: Kumar, TaskName: Frontend

----- MENU -----
1. Add Project
2. Add Employee
3. Add Task
4. Assign project to employee
5. Assign Task
6. Display Employee Details
7. Display Project Details
8. Display Task Details
9. Delete Employee
10. Delete Task
11. List all projects assigned with tasks to an employeee
12. Exit
Enter your choice: 12

C:\Users\Hoime\source\repos\ProjectManagement\ProjectManagement\bin\Debug\net8.0\ProjectManagement.exe (process 8296) ex
ited with code 0 (0x0).
Press any key to close this window . . .
```