

Income Tax Payment

IncomeTaxContent:

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IncomeTaxPaymentApp
{
    //Do NOT change the context name IncomeTaxContext or DbConnect name
    public class IncomeTaxContext:DbContext
    {
        public IncomeTaxContext():base("Name=DataConnection")
        {

        }
        public virtual DbSet<TaxPayerModel> TaxPayers{get;set;}
        public virtual DbSet<TaxPaymentModel> TaxPayments{get;set;}

        //Implement property for 'TaxPayers' with required 'DbSet' declaration
        //Implement property for 'TaxPayments' with required 'DbSet' declaration
        //use public access specifier and virtual keyword while declaring DbSet

    }
}
```

IncomeTaxRepository:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IncomeTaxPaymentApp
{
    public class IncomeTaxRepository
    {
        //Fill your code here
        public int AddTaxPayerDetails(TaxPayerModel taxPayer)
        {
            IncomeTaxContext context=new IncomeTaxContext();
            context.TaxPayers.Add(taxPayer);
            context.SaveChanges();
            return 1;
        }
        public int AddPaymentDetails(TaxPaymentModel payment)
        {
            IncomeTaxContext context=new IncomeTaxContext();
            context.TaxPayments.Add(payment);
            context.SaveChanges();
            return 1;
        }

    }
}
```

```
}
```

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IncomeTaxPaymentApp
{
    class Program
    {
        static void Main(string[] args)
        {
            TaxPayerModel taxPayerModel = new TaxPayerModel();
            TaxPaymentModel taxPaymentModel = new TaxPaymentModel();
            IncomeTaxRepository incomeTaxRepository = new IncomeTaxRepository();

            try
            {
                Console.WriteLine("Welcome to Tax Payment Application");
                Console.WriteLine("Enter Your Name:");
                taxPayerModel.TaxPayerName = Console.ReadLine();
                Console.WriteLine("Enter Your Email ID:");
                taxPayerModel.Email = Console.ReadLine();
                Console.WriteLine("Enter Your Pan Card Number:");
                taxPayerModel.PanNumber = Console.ReadLine();

                incomeTaxRepository.AddTaxPayerDetails(taxPayerModel);

                Console.WriteLine("Your Registration Process is completed");
            }
            catch
            {
                Console.WriteLine("Your Registration Process Failed.Check your entry");
                Console.WriteLine("Thank you.Have a nice day");
                Environment.Exit(0);
            }

            Console.WriteLine("\nWelcome to Payment Process");
            try
            {
                Console.WriteLine("Enter your Pan Number:");
                taxPaymentModel.PanNumber = Console.ReadLine();

                string panPart = taxPayerModel.PanNumber.Substring(0, 3);
                Random randomNumber = new Random();
                taxPaymentModel.PaymentReceiptNumber = panPart + randomNumber.Next(1001,
9999).ToString();

                Console.WriteLine("Enter Financial Year:");
                taxPaymentModel.FinancialYear = int.Parse(Console.ReadLine());
                Console.WriteLine("Enter Yearly Income:");
                taxPaymentModel.YearlyIncome = double.Parse(Console.ReadLine());
                Console.WriteLine("Enter House Rent (Enter 0 if you do not pay house
rent):");
                taxPaymentModel.HouseRent= double.Parse(Console.ReadLine());
            }
            catch
            {
                Console.WriteLine("Enter your Pan Number:");
                taxPaymentModel.PanNumber = Console.ReadLine();

                string panPart = taxPayerModel.PanNumber.Substring(0, 3);
                Random randomNumber = new Random();
                taxPaymentModel.PaymentReceiptNumber = panPart + randomNumber.Next(1001,
9999).ToString();

                Console.WriteLine("Enter Financial Year:");
                taxPaymentModel.FinancialYear = int.Parse(Console.ReadLine());
                Console.WriteLine("Enter Yearly Income:");
                taxPaymentModel.YearlyIncome = double.Parse(Console.ReadLine());
                Console.WriteLine("Enter House Rent (Enter 0 if you do not pay house
rent):");
                taxPaymentModel.HouseRent= double.Parse(Console.ReadLine());
            }
        }
    }
}
```

```

        Console.WriteLine("Enter LIC Premium Amount (Enter 0 if you do not have
LIC Policy):");
        taxPaymentModel.LICPremium = double.Parse(Console.ReadLine());
        Console.WriteLine("Enter Mediclaim Amount (Enter 0 if you do not have
Mediclaim):");
        taxPaymentModel.Mediclaim = double.Parse(Console.ReadLine());
        Console.WriteLine("Enter any other deduction amount like Loan/ELSS etc
"+ "(Enter 0 if you do not have any other deduction):");
        taxPaymentModel.OtherDeductions= double.Parse(Console.ReadLine());
        TaxCalculation taxCalculation = new TaxCalculation();
        taxPaymentModel.TaxPaid = taxCalculation.CalculateTax(taxPaymentModel);

        incomeTaxRepository.AddPaymentDetails(taxPaymentModel);
        Console.WriteLine("\nYour Tax Payment Completed");
        Console.WriteLine("Your Receipt Number is " +
taxPaymentModel.PaymentReceiptNumber);
        Console.WriteLine("Your Tax Payment in Financial Year " +
taxPaymentModel.FinancialYear + " is " + taxPaymentModel.TaxPaid.ToString("0.00"));
    }
    catch
    {
        Console.WriteLine("Tax Payment Failed.Check your entry");
    }

    Console.WriteLine("Thank you.Have a nice day");
}
}
}

```

TaxCalculation:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace IncomeTaxPaymentApp
{
    public class TaxCalculation
    {
        //Fill your code here
        public double CalculateTax(TaxPaymentModel taxPaymentModel)
        {
            double TaxableIncome=taxPaymentModel.YearlyIncome-
(taxPaymentModel.HouseRent+taxPaymentModel.LICPremium+taxPaymentModel.Mediclaim+taxPaym
entModel.OtherDeductions);
            if(TaxableIncome<=300000)
                return 0;
            else if(TaxableIncome>300000 && TaxableIncome<=1000000)
                return TaxableIncome/10;
            else
                return TaxableIncome*15/100;
        }
    }
}

```

TaxPayerModel:

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace IncomeTaxPaymentApp
{
    //Fill your code here
    [Table("tblTaxPayer")]
    public class TaxPayerModel
    {
        [Required] public string TaxPayerName{ get;set;}
        [Required] [RegularExpression("[a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+$")]
        public string Email {get; set;}

        [Required,Key] public string PanNumber {get;set;}

        public ICollection<TaxPaymentModel> TaxPaymentModel {get;set;}
    }
}

TaxPaymentModel:

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace IncomeTaxPaymentApp
{
    //Fill your code here
    [Table("tblTaxPayment")]
    public class TaxPaymentModel
    {
        [Key,Required] public string PaymentReceiptNumber { get;set;}
        [Required] public int FinancialYear {get;set;}
        [Required] public string PanNumber{get;set;} [ForeignKey("PanNumber")]
        public TaxPayerModel TaxPayerModel{get;set;}
        [Required] public double YearlyIncome {get;set;}
        public double HouseRent{get;set;}
        public double LICPremium {get;set;}
        public double Mediclaim{get;set;}
        public double OtherDeductions{get;set;}
        [Required] public double TaxPaid {get;set;}
    }
}

```