Employee Payroll System

Code

```
using System;
using System.IO;
using System.Collections.Generic;

namespace EmployeePayrollSystem
{
    class BaseEmployee
    {
        //Properties
        public string Name { get; set; }
        public string Role { get; set; }
        public int ID { get; set; }
        public double Basic_Pay { get; set; }
        public double Allowances { get; set; }
        public double Deductions { get; set; }

        //Creating constructor
        public BaseEmployee(string name, string role, int id, double basic_pay, double
allowances, double deductions)
        {
            Name = name;
            Role = role;
            ID = id;
            Basic_Pay = basic_pay;
            Allowances = allowances;
            Deductions = deductions;
        }

        //Salary Calculation
        public virtual double SalaryCalculation()
        {
            return Basic_Pay + Allowances - Deductions;
        }

        //Employee Details
        public virtual void DisplayDetails()
        {

Console.WriteLine("-----------------------------------------------------------------------------
-----------------");
            Console.WriteLine($"ID: {ID}, Name: {Name}, Role: {Role}, Basic Pay:
{Basic_Pay}, Allowances: {Allowances}, Deductions: {Deductions}");

Console.WriteLine("-----------------------------------------------------------------------------
-----------------");

        }

    }

    // Developer class,inherits from BaseEmployee
    class Developer :BaseEmployee
    {
        public string Program_lang { get; set; }

        public Developer(string name,int id, double basicPay, double allowances, double
deductions, string program_lang)
            : base(name, "Developer",id, basicPay, allowances, deductions)
        {
            Program_lang = program_lang;
        }

        public override void DisplayDetails()
        {
            base.DisplayDetails();
            Console.WriteLine($"Programming Language: {Program_lang}");
        }
    }

    // Manager class inherits from BaseEmployee
    class Manager :BaseEmployee
    {
        public double Bonus { get; set; }

        public Manager(string name,int id,   double basic_pay, double allowances, double 
deductions, double bonus)
            : base(name,"Manager", id, basic_pay, allowances, deductions)
        {
            Bonus = bonus;
        }

        public override double SalaryCalculation()
        {
            return base.SalaryCalculation() + Bonus;
        }

        public override void DisplayDetails()
        {
            base.DisplayDetails();
            Console.WriteLine($"Bonus: {Bonus}");
        }
    }

    // Intern class inherits from BaseEmployee
    class Intern :BaseEmployee
    {
        public int Fees { get; set; }

        public Intern(string name, int id, double basicPay, double allowances, double 
deductions, int fees)
            : base(name,  "Intern",id, basicPay, allowances, deductions)
        {
            Fees = fees;
        }

        public override void DisplayDetails()
        {
            base.DisplayDetails();
            Console.WriteLine($"Internship Fees: {Fees}");
        }

    }
    class Program
    {
        static List<BaseEmployee> emp = new List<BaseEmployee>();

        static void Main(string[] args)
        {
            bool show_menu = true;
            while (show_menu)
            {
                Console.Clear();

Console.WriteLine("-------------------------------------------------------------------------
-----");
                Console.WriteLine("--------------------WELCOME TO EMPLOYEE PAYROLL 
SYSTEM----------------------");

Console.WriteLine("-------------------------------------------------------------------------
-----");
                Console.WriteLine("Select your option: ");
                Console.WriteLine("1. Add New Employee");
                Console.WriteLine("2. Display All Employees");
                Console.WriteLine("3. Calculate and Display Salary for an Employee");
                Console.WriteLine("4. Calculate Total Payroll");
                Console.WriteLine("5. Exit");
                string choice = Console.ReadLine();

                switch (choice)
                {
                    case "1":
                        AddNewEmployee();
                        break;
                    case "2":
                        DisplayAllEmployees();
                        break;
                    case "3":
                        CalculateAndDisplaySalary();
                        break;
                    case "4":
                        CalculateTotalPayroll();
                        break;
                    case "5":
                        show_menu = false;
                        break;
                    default:
                        Console.WriteLine("Invalid choice. Please try again...");
                        break;
                }
            }
        }

        // Add new employee
        static void AddNewEmployee()
        {
            Console.Clear();
            Console.WriteLine("Selected option: Add New Employee");

Console.WriteLine("---------------------------------------------------------------------------
-----");
            Console.WriteLine("------------------------------Add New
Employee-------------------------------");

Console.WriteLine("---------------------------------------------------------------------------
-----");
            Console.Write("Enter Name: ");
            string name = Console.ReadLine();
            Console.Write("Enter ID: ");
            int id = int.Parse(Console.ReadLine());
            Console.Write("Enter Basic Pay: ");
            double basicPay = double.Parse(Console.ReadLine());
            Console.Write("Enter Allowances: ");
            double allowances = double.Parse(Console.ReadLine());
            Console.Write("Enter Deductions: ");
            double deductions = double.Parse(Console.ReadLine());

            Console.WriteLine("Choose Role: ");
            Console.WriteLine("1. Manager");
            Console.WriteLine("2. Developer");
            Console.WriteLine("3. Intern");
            Console.Write("Select: ");
            string roleChoice = Console.ReadLine();

            BaseEmployee employee = null;

            switch (roleChoice)
            {
                case "1":
                    Console.Write("Enter Bonus for Manager: ");
                    double bonus = double.Parse(Console.ReadLine());
                    employee = new Manager(name, id, basicPay, allowances, deductions,
bonus);
                    break;
                case "2":
                    Console.Write("Enter Programming Language for Developer: ");
                    string program_lang = Console.ReadLine();
                    employee = new Developer(name, id, basicPay, allowances, deductions,
program_lang);
                    break;
                case "3":
                    Console.Write("Enter fees for Intern: ");
                    int fees = int.Parse(Console.ReadLine());
                    employee = new Intern(name, id, basicPay, allowances, deductions,
fees);
                    break;
                default:
                    Console.WriteLine("Invalid role.");
                    return;
            }

            emp.Add(employee);
            Console.WriteLine("New employee successfully added!!!");
            Console.WriteLine("Press enter key to go main menu...");
            Console.ReadKey();
        }

        // Display all employee details
        static void DisplayAllEmployees()
        {
            Console.Clear();
            Console.WriteLine("Selected option: Display All Employees");

Console.WriteLine("------------------------------------------------------------------
-----");
            Console.WriteLine("List of Employees:");

Console.WriteLine("------------------------------------------------------------------
-----");

            foreach (var employee in emp)
            {
                employee.DisplayDetails();
                Console.WriteLine($"Salary: {employee.SalaryCalculation()}\n");
            }
            Console.WriteLine("Press enter key to go main menu...");
            Console.ReadKey();
        }

        // Calculate and display individual salaries
        static void CalculateAndDisplaySalary()
        {
            Console.Clear();
            Console.WriteLine("Selected option: Calculate and Display Salary for an
Employee");
            Console.Write("Enter Employee ID: ");
            int id = int.Parse(Console.ReadLine());

            var employee = emp.Find(e => e.ID == id);

            if (employee != null)
            {
                Console.WriteLine($"Salary for {employee.Name}:
{employee.SalaryCalculation()}");
            }
            else
            {
                Console.WriteLine("Employee not found.");
            }
            Console.WriteLine("Press enter key to go main menu...");
            Console.ReadKey();
        }
        // Calculate total payroll for all employees
        static void CalculateTotalPayroll()
        {
            Console.Clear();
            Console.WriteLine("Selected option: Calculate Total Payroll");

            double totalPayroll = 0;

            foreach (var employee in emp)
            {
                totalPayroll += employee.SalaryCalculation();
            }

            Console.WriteLine($"Total Payroll: {totalPayroll}");
            Console.WriteLine("Press enter key to go main menu...");
            Console.ReadKey();
        }
    }
}
```