using System;

using System.Collections.Generic;

using System.Linq;

public enum LoanError

{

ApplicationRejected,

InvalidLoanAmount,

IneligibleForLoan,

InsufficientCreditScore,

Underage,

MissingEmail,

MissingLastName,

MissingFirstName

}

public interface ILoanApplicationResponse

{

string ToString();

}

public class LoanApplicationSuccessResponse : ILoanApplicationResponse

{

public int ApplicationId { get; }

public LoanApplicationSuccessResponse(int applicationId)

{

ApplicationId = applicationId;

}

public override string ToString() => $"Application Approved: ID {ApplicationId}";

}

public class LoanApplicationErrorResponse : ILoanApplicationResponse

{

public LoanError Error { get; }

public LoanApplicationErrorResponse(LoanError error)

{

Error = error;

}

public override string ToString() => $"Application Error: {Error}";

}

public interface ILoanRepository

{

int SaveApplication(Customer customer);

}

public class LoanRepository : ILoanRepository

{

public int SaveApplication(Customer customer)

{

Console.WriteLine("Saving application to the database...");

return new Random().Next(1000, 9999);

}

}

public class Customer

{

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public int Age { get; set; }

public int CreditScore { get; set; }

public bool HasCriminalRecord { get; set; }

public double AnnualIncome { get; set; }

public string EmploymentStatus { get; set; }

public List<string> PreviousLoans { get; set; }

public double LoanAmountRequested { get; set; }

public bool IsEligible()

{

if (Age < 18) return false;

if (CreditScore < 600) return false;

if (LoanAmountRequested <= 0 || LoanAmountRequested > 500000) return false;

return true;

}

}

public class LoanApplicationService

{

private readonly ILoanRepository \_repository;

public LoanApplicationService(ILoanRepository repository)

{

\_repository = repository;

}

public ILoanApplicationResponse ApplyForLoan(Customer customer)

{

if (string.IsNullOrWhiteSpace(customer.FirstName))

return new LoanApplicationErrorResponse(LoanError.MissingFirstName);

if (string.IsNullOrWhiteSpace(customer.LastName))

return new LoanApplicationErrorResponse(LoanError.MissingLastName);

if (string.IsNullOrWhiteSpace(customer.Email))

return new LoanApplicationErrorResponse(LoanError.MissingEmail);

if (!customer.IsEligible())

return new LoanApplicationErrorResponse(LoanError.IneligibleForLoan);

int applicationId = \_repository.SaveApplication(customer);

return new LoanApplicationSuccessResponse(applicationId);

}

}

class Program

{

static void Main(string[] args)

{

var customer = new Customer

{

FirstName = "John",

LastName = "Doe",

Email = "john@example.com",

Age = 25,

CreditScore = 700,

HasCriminalRecord = false,

AnnualIncome = 40000,

EmploymentStatus = "FullTime",

PreviousLoans = new List<string> { "SafeBankLoan" },

LoanAmountRequested = 200000

};

ILoanRepository repository = new LoanRepository();

var loanService = new LoanApplicationService(repository);

var response = loanService.ApplyForLoan(customer);

Console.WriteLine(response.ToString());

}

}