1.ADO.NET\_WorkshopEnrollmentApp\_HandsOn1

**Hands On: Data Insert Operation with ADO.NET**

**Scenario:**

 Your organization has decided to enroll trainees for a workshop on ***ART of LIVING***. Help admin by creating an application to register trainees for the workshop.

**Hands- On Description:**

Here you will practice how to store data into Sql Server database using **ADO.NET** and **C#** by implementing 3-layer application architecture.

**Data Design:**

|  |  |  |
| --- | --- | --- |
| **Table name: tblTrainee** | |  |
| **Column Name** | **Data type** | **Constraints** |
| Trainee\_Id | bigint | Primary Key, Not Null |
| Trainee\_Name | varchar(100) | Not Null |
| Batch\_Code | varchar(100) | Not Null |

The database connection information is stored in the “App.config” file, which is also provided as part of code skeleton/template.

**Component Specification:**

Create a class called **TraineeBO** (business object) with the below public auto-implemented properties:

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **DataType** | **Properties** |
| TraineeBO | long | TraineeId |
| string | TraineeName |
| string | BatchCode |

Also add a default constructor and parameterized constructor in this class.

Create a class called **TraineeDA** with the below method: (This class contains data operations logic such as insert, update, delete or retrieve records to/from database)

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **Method** | **Responsibility** |
| TraineeDA | public bool AddTraineeDetails(TraineeBO objBO) | This method should accept a TraineeBO object and execute a sql query to insert a trainee record into the database. It returns *true* if the insertion is successful and *false* if it is failed to insert. |

Use (as it is) the below code snippet in order to read the connection string values from the App.config file.

public string ConnectionString

{

  get

  {

       return ConfigurationManager.ConnectionStrings["SqlCon"].ConnectionString;

  }

}

Create a class called **TraineeBL** with the below method: (This class contains business logic, validations and calculations related to the data. The Presentation Layer will communicate to Data Layer via this Business Layer)

|  |  |  |
| --- | --- | --- |
| **Type (Class)** | **Method** | **Responsibility** |
| TraineeBL | public bool SaveTraineeDetails(TraineeBO objBO) | This method should accept a TraineeBO object and call the TraineeDA’s AddTraineeDetails method that returns true if it successfully inserts data to the database; else returns false. |

From the **Program** class’s Main method, call the TraineeBL’s method and test your application.

**Business Rule:**

Use **try… catch** block while working with TraineeDA and TraineeBLclass methods to catch exception.

**Sample Input / Output1:**

2.Seminar Ticket Booking

**SEMINAR TICKET BOOKING USING ADO DOT NET**

**Scenario:**

Seminar ticket booking capture the name and seat number.  We need to implement New booking, View all the booking and update booking using ado .net with SQLSERVER database.

**Table Details:**

**Table Name  : Booking**

|  |  |
| --- | --- |
| **Field Name** | **Datatype** |
| Id – Primary key – Auto Increment | int |
| Name | varchar(50) |
| Seatno | varchar(20) |

**Implement following steps:**

1)      Use below methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Name** | **Argument** | **Return Type** | **Access Specifier** | **Responsibilities** |
| NewBooking | string name, string seatno | void | public | New booking Insert a record into table. ‘Id’ not need to pass, is auto increment by one. |
| GetAllBooking |  | void | public | Retrieve all the booking from the booking table.  Note : Use query to retrieve only name and seatno column from the table.  Display ONLY these 2 column from all the booking details. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UpdateBooking | string name, string seatno, | void | public | Here we having update option for only the name.  Update the record name against the seat number |

2)      For the main logic,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Name** | **Argument** | **Return type** | **Access Specifier** | **Responsibilities** |
| MainMenu | N/A | bool | private static | Here we are,  Call the ''GetAllBooking" method to display the data from database.  Then we have 3 menu options,  Seminar Ticket Booking  1. New Booking  2. View All Booking  3. Update Booking  4. Exit  If '1' then, need to be capture name and seat number insert to table.  If '2' then, retrieve all the data from the table and display the name and seat number  If '3' then, capture both name and seat number, name only wee can update here against the seat number. |

 3) Use '**ConnectionString**' as the connection string name to connect to the database. This is given in App.config.

*YOU NEED NOT MAKE ANY CHANGES TO****App.config***

**Sample Input/Output:**

|  |  |
| --- | --- |
| **S.No** | **Sample Input & Output** |
| **1** | **Seminar Ticket Booking**  **1. New Booking**  **2. View All Booking**  **3. Update Booking**  **4. Exit**  **1**  **Booking Name :**  **Shyam**  **Seat No. :**  **A20**  **---New Booking Saved---**  **1   Shyam   A20** |
| **2** | **Seminar Ticket Booking**  **1. New Booking**  **2. View All Booking**  **3. Update Booking**  **4. Exit**  **2**  **Shyam  A20** |

|  |  |
| --- | --- |
| **3** | **Seminar Ticket Booking**  **1. New Booking**  **2. View All Booking**  **3. Update Booking**  **4. Exit**  **3**  **Booking Name :**  **Shyam Saravan**  **Seat No. :**  **A20**  **---Booking Updated---**  **1   Shyam Saravan    A20** |