

Egy könyvtárkezelő alkalmazást fog megvalósítani, ahol a felhasználó könyveket adhat hozzá, listázhatja őket, vagy törölhet egy könyvet az adatbázisból.

---

### ### 1. Előkészületek

- **\*\*Adatbázis\*\***: Hozz létre egy MySQL adatbázist a következő szerkezettel:

```
```sql
```

```
CREATE DATABASE LibraryDB;
```

```
USE LibraryDB;
```

```
CREATE TABLE Books (
```

```
    Id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    Title VARCHAR(255) NOT NULL,
```

```
    Author VARCHAR(255) NOT NULL,
```

```
    YearPublished INT NOT NULL
```

```
);
```

```
```
```

- **\*\*C# projekt\*\***: Hozz létre egy új `Console App` projektet a Visual Studio-ban.

- **\*\*MySQL Connector\*\***: Telepítsd a MySQL.Data NuGet csomagot:

```
```bash
```

```
dotnet add package MySql.Data
```

```
```
```

---

### ### 2. Kód

Az alábbi kódot másold be a `Program.cs` fájlba:

```
``csharp
using System;
using MySql.Data.MySqlClient;

class Program
{
    private const string ConnectionString =
"Server=localhost;Database=LibraryDB;Uid=root;Pwd=yourpassword;";

    static void Main(string[] args)
    {
        while (true)
        {
            Console.WriteLine("\n--- Könyvtárkezelő ---");
            Console.WriteLine("1. Könyv hozzáadása");
            Console.WriteLine("2. Könyvek listázása");
            Console.WriteLine("3. Könyv törlése");
            Console.WriteLine("4. Kilépés");
            Console.Write("Válassz egy lehetőséget: ");
            string choice = Console.ReadLine();
```

```

switch (choice)
{
    case "1":
        AddBook();
        break;
    case "2":
        ListBooks();
        break;
    case "3":
        DeleteBook();
        break;
    case "4":
        return;
    default:
        Console.WriteLine("Érvénytelen választás. Próbáld újra.");
        break;
}
}
}

```

```

static void AddBook()
{
    Console.Write("Könyv címe: ");
    string title = Console.ReadLine();

    Console.Write("Szerző: ");
    string author = Console.ReadLine();
}

```

```

    Console.WriteLine("Kiadás éve: ");
    if (!int.TryParse(Console.ReadLine(), out int year))
    {
        Console.WriteLine("Érvénytelen év.");
        return;
    }

    using (var connection = new MySqlConnection(ConnectionString))
    {
        connection.Open();

        string query = "INSERT INTO Books (Title, Author, YearPublished)
VALUES (@Title, @Author, @YearPublished)";

        using (var command = new MySqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@Title", title);
            command.Parameters.AddWithValue("@Author", author);
            command.Parameters.AddWithValue("@YearPublished", year);
            command.ExecuteNonQuery();
        }
    }

    Console.WriteLine("Könyv sikeresen hozzáadva.");
}

static void ListBooks()
{
    using (var connection = new MySqlConnection(ConnectionString))

```

```

{
    connection.Open();
    string query = "SELECT * FROM Books";
    using (var command = new MySqlCommand(query, connection))
    using (var reader = command.ExecuteReader())
    {
        Console.WriteLine("\n--- Könyvek ---");
        while (reader.Read())
        {
            Console.WriteLine($"ID: {reader["Id"]}, Cím: {reader["Title"]},
Szerző: {reader["Author"]}, Év: {reader["YearPublished"]}");
        }
    }
}

```

```

static void DeleteBook()

```

```

{
    Console.Write("Törölni kívánt könyv ID-ja: ");
    if (int.TryParse(Console.ReadLine(), out int bookId))
    {
        Console.WriteLine("Érvénytelen ID.");
        return;
    }
}

```

```

using (var connection = new MySqlConnection(ConnectionString))
{

```



- Könyv törlése (`3`): Add meg a törlendő könyv ID-ját.
- Kilépés (`4`): Bezárja az alkalmazást.

---