

## BÀI THỰC HÀNH CÔNG NGHỆ .NET #4

(Thực hiện đồ án mẫu)

### Nội dung thực hiện đồ án:

- Xây dựng cơ sở dữ liệu (Bài thực hành #4)
- Thiết kế giao diện (Bài thực hành #5 và #6)
- Lập trình kết nối dữ liệu lên các giao diện đã thiết kế (Bài thực hành #7 đến #10)
- Tạo các biểu mẫu báo cáo (Bài thực hành #11)
- Đóng gói sản phẩm (Bài thực hành #12)

### 1. Tên đồ án: XÂY DỰNG PHẦN MỀM QUẢN LÝ CHO THUÊ SÁCH (BOOK RENTAL MANAGEMENT SOFTWARE)

### 2. Nội dung chuẩn bị:

- Thông tin về bài toán:** Công ty Beta-Books là công ty cho thuê sách. Hiện nay công ty đang quản lý công việc cho thuê sách dựa trên phần mềm Excel. Để nâng cao hiệu quả công việc quản lý cho thuê sách, công ty cần xây dựng một phần mềm quản công việc cho thuê sách. Hãy khảo sát nghiệp vụ bài toán và xây dựng một phần mềm đáp ứng các yêu cầu của công ty.
  - Tổ chức của công ty** (Đọc kỹ tài liệu đi kèm để hiểu bài toán, mục 1.2 trong Chương 1).
  - Tổng quan về các chức năng của hệ thống** (Đọc kỹ tài liệu đi kèm, mục 2.1 trong Chương 2).
  - Yêu cầu cụ thể của các chức năng** (Đọc kỹ tài liệu đi kèm, mục 2.2 trong Chương 2).
- Tạo CSDL có tên **BOOKRENTAL** trong Microsoft SQL Server.
  - Tạo các bảng sau trong CSDL trên (Chi tiết tại mục 2.3.2 trong Chương 2)
    - Bảng **Users** dùng để lưu danh sách nhân viên (người dùng).
    - Bảng **Categories** dùng để lưu trữ loại sách.
    - Bảng **Publishers** dùng để lưu trữ danh sách nhà xuất bản.
    - Bảng **Authors** lưu trữ danh sách tác giả.

- e. Bảng **BookAuthors** lưu trữ các tác giả của sách.
  - f. Bảng **Languages** lưu trữ ngôn ngữ của sách.
  - g. Bảng **Books** lưu chi tiết thông tin một quyển sách.
  - h. Bảng **RentalPrices** lưu bảng giá thuê sách.
  - i. Bảng **Customers** lưu danh sách khách hàng.
  - j. Bảng **Rentals** lưu trữ thông tin cho thuê sách.
  - k. Bảng **RentalDetails** lưu chi tiết sách cho thuê.
  - l. Bảng **Suppliers** lưu danh sách nhà cung cấp sách.
  - m. Bảng **Orders** lưu hóa đơn sách được nhập vào kho.
  - n. Bảng **OrderDetails** lưu chi tiết hóa đơn sách được nhập vào kho.
5. Thiết lập quan hệ giữa các bảng dữ liệu (Xem mục 2.3.3 trong Chương 3)

## BÀI THỰC HÀNH CÔNG NGHỆ .NET #5 VÀ #6

**Mục tiêu:** Sinh viên xây dựng được giao diện trên phần mềm Visual Studio 2019 dựa trên thông tin từ các bảng dữ liệu đã tạo ở bài thực hành số 4.

### Yêu cầu:

- Thiết kế giao diện Form cho toàn bộ các bảng dữ liệu;
- Thiết kế giao diện Form đăng nhập và giao diện chính để gọi các giao diện Form khác;
- Thời gian thực hiện 04 tiết, chia làm 02 buổi thực hành (mỗi buổi 08 Form)
- Đặt tên (thuộc tính Name) cho các điều khiển theo yêu cầu sau
  - o Điều khiển Button được đặt tên bắt đầu: **btn** (ví dụ: **btnSave**)
  - o Điều khiển TextBox được đặt tên bắt đầu: **txt** (ví dụ: **txtName**)
  - o Điều khiển ComboBox: **cb** (ví dụ: **cbFont**)
  - o Điều khiển DataGridView: **dgv** (ví dụ: **dgvCategory**)
  - o Điều khiển MenuStrip: **ms** (ví dụ: **msFile**)
  - o Điều khiển CheckBox: **chb**
  - o Điều khiển CheckedListBox: **clb**
  - o Điều khiển ListBox: **lb**
  - o ...
- Vị trí các điều khiển trong Form tùy ý.

### Ví dụ tạo các giao diện:

- Giao diện quản lý nhà cung cấp (Bảng **Suppliers**)

Nhập nhà cung cấp	
Tên nhà cung cấp (*):	<input type="text"/>
Họ tên người liên lạc:	<input type="text"/>
Địa chỉ công ty:	<input type="text"/>
Điện thoại (*):	<input type="text"/>
Email:	<input type="text"/>
Mô tả thêm:	<input type="text"/>
<div>Danh sách nhà cung cấp</div> <div></div>	
<div>Thêm mới</div> <div>Ghi</div> <div>Hủy bỏ</div> <div>Kết thúc</div>	

- Giao diện lưu chi tiết thông tin một quyển sách (Bảng **Books**) – Có 2 cách có thể trình bày như sau:

- Giao diện đăng nhập

Có thể thêm hình ảnh cho đẹp, ví dụ

- Tạo giao diện cho các bảng khác

## BÀI THỰC HÀNH CÔNG NGHỆ .NET #7 (Kết nối 1 bảng dữ liệu lên 1 Form)

### Mục đích:

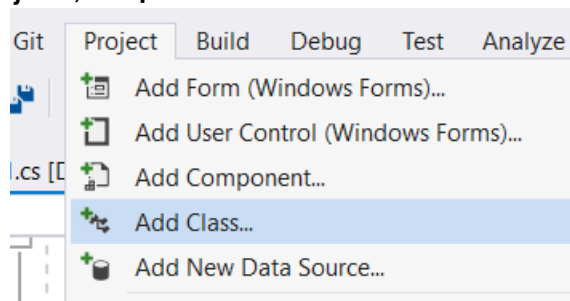
- Giải thích được các khái niệm trong ADO.NET
- Kết nối cơ sở dữ liệu SQL Server lên một giao diện bất kỳ

### Chuẩn bị:

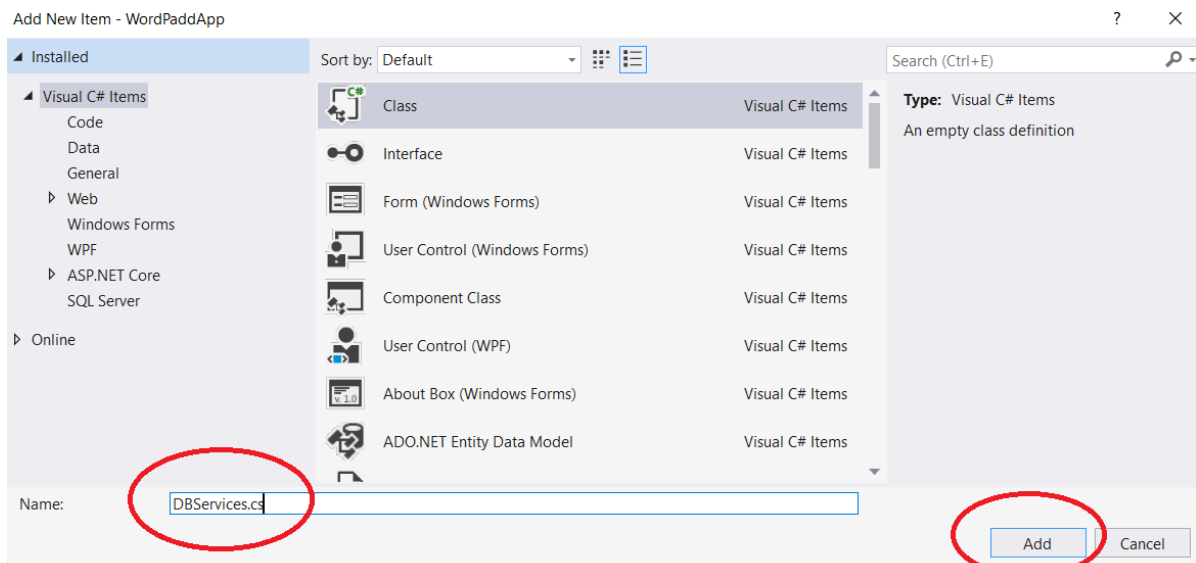
- Bảng **Users** được tạo trong CDSL **BOOKRENTAL** (bài thực hành trước)
- Form **frmUsers** được tạo ở bài thực hành #5 (VS 2019)
- Slide về lập trình ADO.NET

### Hướng dẫn:

1. Mở project đã tạo các form ở bài thực hành trước
2. Tạo một class **DBServices.cs** (chú ý đặt hoa thường, lớp này lưu trữ các phương thức để thao tác với dữ liệu qua câu lệnh SQL như SELECT, INSERT INTO, DELETE, UPDATE, CREATE...)
  - Trên menu Project, chọn Add Class



### - Gõ **DBServices.cs**, chọn Add



- Lớp DBServices.cs như sau

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace BookRentals
8  {
9      class DBServices
10     {
11     }
12 }
13
```

+ Dòng 1 đến 5 là khai báo các thư viện cần thiết (tương tự #include trong C và C++)

+ Dòng 7 là không gian tên, được đặt trùng với tên Project khi tạo mới (chú ý khi tạo project hay tạo các form không đặt dấu cách trống)

+ Dòng 9 là khai báo class DBServices

- Thêm các dòng lệnh sau để khai báo thư viện kết nối CSDL và Forms

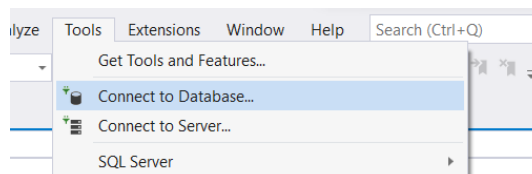
```
6  using System.Data;
7  using System.Data.SqlClient;
8  using System.Windows.Forms;
```

- Tạo chuỗi kết nối và một số đối tượng khác trong lớp DBServices (chú ý, các kiểu chuỗi kết nối có thể xem thêm ở slide bài giảng)

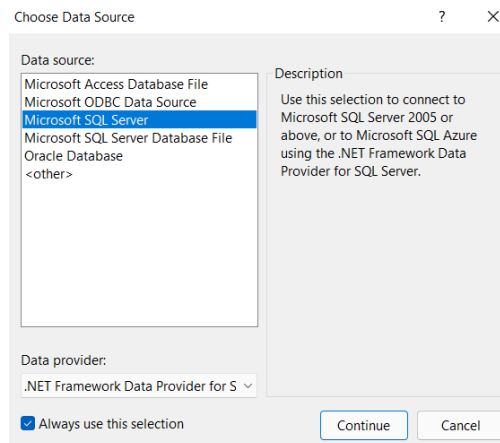
```
class DBServices
{
    private string conStr = @"Data Source=.;Initial Catalog=BOOKRENTAL;Integrated Security=True";
    private SqlConnection mySqlConnection;
}
```

**Chú ý:** Một cách khác để tạo conStr như sau:

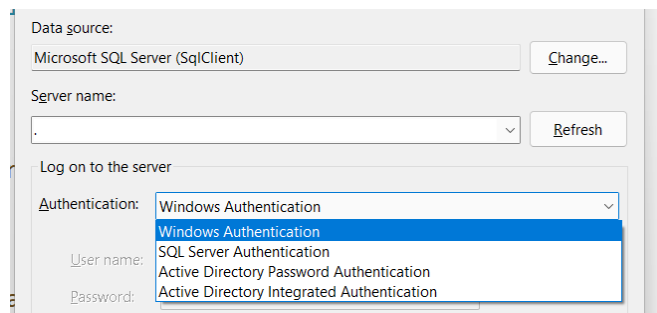
→ Vào Tools\Connect to Database



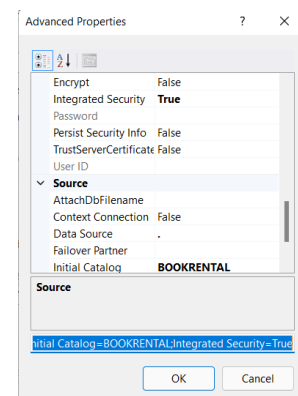
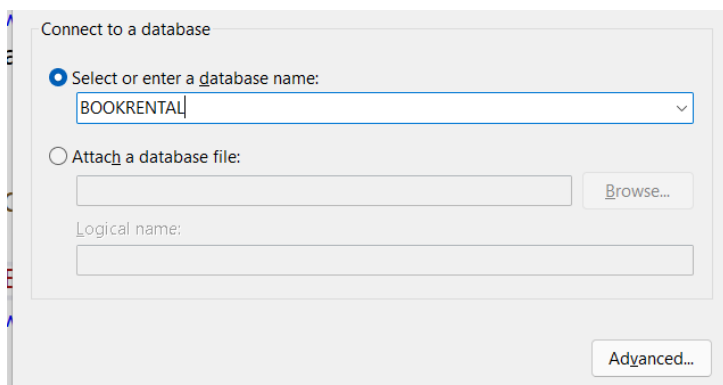
→ Chọn Microsoft SQL Server/ Continue



→ Gõ tên Server vào ô Server name (hoặc gõ dấu . nếu dữ liệu ngay trên máy đang làm), chọn xác thực bằng Windows (Windows Authentication) hoặc xác thực bằng SQL Server. Nếu xác thực bằng SQL Server thì phải điền User name và Password ở ô bên dưới.



→ Chọn tên CSDL (BOOKRENTAL) trong ô Select or enter database name



→ Chọn Advanced, copy dòng string connection phía dưới như hình trên, sau đó ấn mấy lần cancel để thoát khi đã copy xong kết nối.

→ Paste chuỗi kết nối trong cặp nháy kép là xong việc tạo chuỗi kết nối.  
private string conStr = @"<paste chuỗi kết nối vào đây>";

- Tạo các phương thức (hàm) sau trong lớp DBServices
- + Hàm thiết lập: public DBServices()
- + Lấy dữ liệu: public DataTable getData(string sSql), dùng cho câu lệnh SELECT
- + Thực hiện truy vấn: public void runQuery(string sSql), dùng cho các lệnh còn lại (INSERT INTO, DELETE, UPDATE, CREATE,...)
- + Cấu trúc các hàm như hình sau:

```
namespace BookRentals
{
    5 references
    class DBServices
    {
        private string conStr = @"Data Source=.;Initial Catalog=BOOKRENTAL;Integrated Security=True";
        private SqlConnection mySqlConnection;

        2 references
        public DBServices()...

        1 reference
        public DataTable getData(string sSql)...

        1 reference
        public void runQuery(string sSql)...
    }
}
```

- Nội dung cụ thể các hàm

```
public DBServices()
{
    mySqlConnection = new SqlConnection(conStr);
}

public DataTable getData(string sSql)
{
    try
    {
        SqlDataAdapter myDataAdapter = new SqlDataAdapter(sSql, mySqlConnection);
        DataTable myDataTable = new DataTable();
        myDataAdapter.Fill(myDataTable);
        return myDataTable;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        return null;
    }
}
```



```
public void runQuery(string sSql)
{
    try
    {
        mySqlConnection.Open();
        SqlCommand mySqlCommand = new SqlCommand(sSql, mySqlConnection);
        mySqlCommand.ExecuteNonQuery();
        mySqlConnection.Close();
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
        return;
    }
}
```

+ Ghi chú: SqlDataAdapter trong hàm getData có thể thay bằng SqlDataReader, đọc thêm thông tin trong slide.

→ Đến bước này là tạo xong class DBServices, lớp này sẽ được sử dụng cho toàn bộ các Forms sau này.

### 3. Giao diện User (đã tạo ở các buổi trước)

- Form được đặt tên

+ frmUsers

- Quy tắc đặt tên các TextBox theo thứ tự từ trên xuống:

- + txtUserID
- + txtFullName
- + txtPassword
- + txtPhone
- + txtEmail
- + txtDescription
- CheckBox được đặt tên
  - + chbStatus
- Các button được đặt tên
  - + btnAddNew
  - + btnSave
  - + btnCancel
  - + btnEdit
  - + btnDelete
  - + btnExit
- dataGridView được đặt tên: dgvUsers

Lưu ý: nếu các bạn đặt tên khác có thể đổi lại tên hoặc lập trình với tên tương ứng với những tên các bạn đã đặt từ giờ về sau.

#### 4. Lập trình đưa dữ liệu lên dgvUser

- Nhấp đúp chuột vào Form để tạo sự kiện Load của form

```
namespace BookRentals
{
    3 references
    public partial class frmUsers : Form
    {
        1 reference
        public frmUsers()
        {
            InitializeComponent();

            //Thêm các hàm ở đây

            1 reference
            private void frmUsers_Load(object sender, EventArgs e)
            {
            }
        }
    }
}
```

- Lập trình thêm các hàm sau (ở vị trí //Thêm các hàm ở đây)

```
private void setEnable(bool check)
{
    txtUserID.Enabled = false;
    txtFullName.Enabled = check;
    txtUserName.Enabled = check;
    txtPassword.Enabled = check;
    txtPhone.Enabled = check;
    txtEmail.Enabled = check;
    txtDescription.Enabled = check;
    chbStatus.Enabled = check;
    btnSave.Enabled = check;
    btnCancel.Enabled = check;
    btnAddNew.Enabled = !check;
    btnEdit.Enabled = !check;
    btnDelete.Enabled = !check;
    btnExit.Enabled = !check;
    dgvUsers.Enabled = !check;
}

private void LoadGridData()
{
    DBServices db = new DBServices();
    string sql = "SELECT * FROM Users WHERE Status = 1";
    dgvUsers.DataSource = db.getData(sql);
    setEnable(false);
}
```

- Trong sự kiện Load của form

```
private void frmUsers_Load(object sender, EventArgs e)
{
    LoadGridData();
}
```

- Chạy thử xem dữ liệu

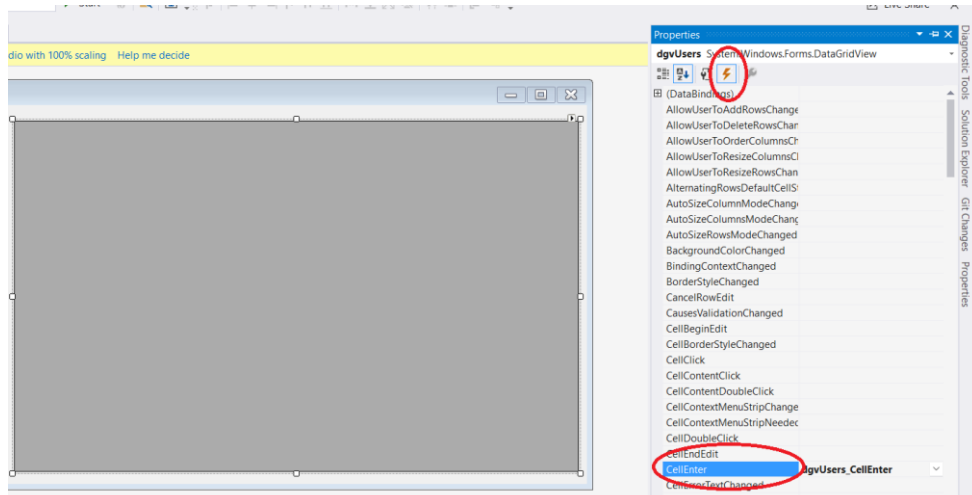
The screenshot shows a Windows application window titled 'Users'. On the left is a form with input fields for UserID (1), FullName (Tôn Ngộ Khổng), Username (khongn1), Password (masked), Phone (777888999), Email (khongn1@hoaquason.com), and a Description field. There are also checkboxes for 'Status' (checked) and 'Check for active account'. At the bottom of the form are buttons for 'Add New', 'Save', 'Cancel', 'Edit', 'Delete', and 'Exit'. On the right is a data grid with columns: UserID, FullName, Username, Password, Phone, and Email. The grid contains 11 rows of data. The first row is highlighted in blue.

UserID	FullName	Username	Password	Phone	Email
1	Tôn Ngộ Khổng	khongn1	123	777888999	khongn1@ho
2	Tru Bie Giê	gotb	123	666777666	gotb@tyedu
3	Tôn Ngộ Khổng 1	khongn1	1231	7778889991	khongn1@ho
6	2	2	2	2	2
10	6	6	6	6	6
11	8	6	8	8	8
*					

- Nếu chưa hiển thị dữ liệu xem lại từ các bước trước. Thông tin mật khẩu đang hiện, sẽ ẩn sau.
- Thêm lệnh cho nút thoát, nhấp đúp vào nút Exit tạo sự kiện click

```
private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}
```

- Tạo lệnh cho sự kiện CellEnter của dgvUsers
  - + Nhấp phải chuột vào dgvUsers → chọn properties → chọn Sự kiện → nhấp đúp vào CellEnter.



- Xuất hiện

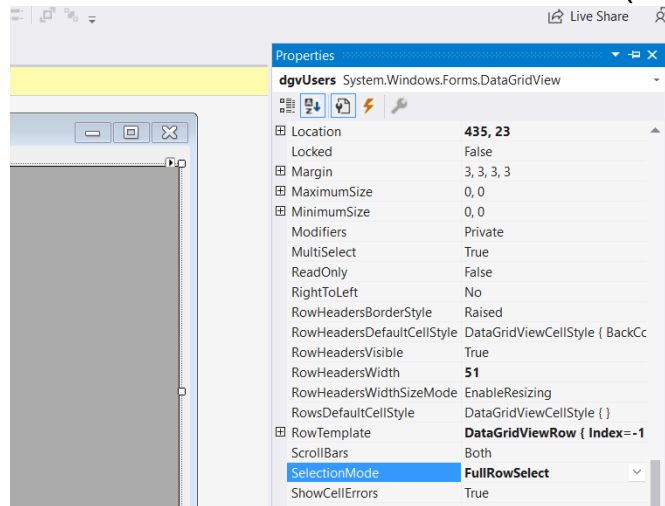
```
private void dgvUsers_CellEnter(object sender, DataGridViewCellEventArgs e)
{
}
```

- Lập trình nhấp vào từng ô trong dgvUsers thì xuất hiện thông tin lên các textBox tương ứng

```
private void dgvUsers_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    int i = e.RowIndex;
    if (i >= 0)
    {
        txtUserID.Text = dgvUsers.Rows[i].Cells["UserID"].Value.ToString();
        txtFullName.Text = dgvUsers.Rows[i].Cells["FullName"].Value.ToString();
        txtUserName.Text = dgvUsers.Rows[i].Cells["UserName"].Value.ToString();
        txtPassword.Text = dgvUsers.Rows[i].Cells["Password"].Value.ToString();
        txtPhone.Text = dgvUsers.Rows[i].Cells["Phone"].Value.ToString();
        txtEmail.Text = dgvUsers.Rows[i].Cells["Email"].Value.ToString();
        txtDescription.Text = dgvUsers.Rows[i].Cells["Description"].Value.ToString();
        if (dgvUsers.Rows[i].Cells["Status"].Value.ToString() == "1")
            chbStatus.Checked = true;
        else
            chbStatus.Checked = false;
    }
}
```

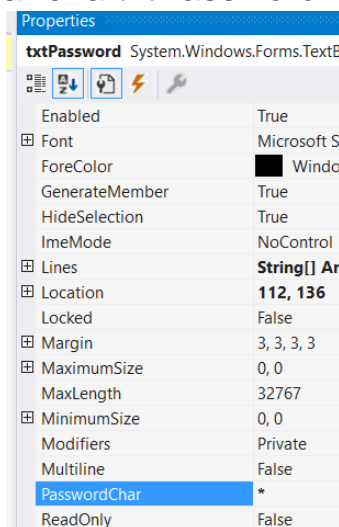
- Muốn nhấp chuột vào dgvUsers sẽ chọn cả hàng thực hiện các bước sau
  - + Nhấp chuột phải vào dgvUsers, chọn Properties

+ Chọn thuộc tính Selection Mode: FullRowSelect (như hình dưới)



+ Chạy và thử nghiệm

- Đặt thuộc tính PasswordChar của txtPassword là dấu \*



- Lập trình nút Add new: nhấp đúp vào Add new

+ Thêm biến AddNew vào vị trí như sau

```
public partial class frmUsers : Form
{
    bool AddNew = false;
    public frmUsers()
    {
        InitializeComponent();
    }
}
```

+ Trong sự kiện Click của nút Add New

```
private void btnAddNew_Click(object sender, EventArgs e)
{
    AddNew = true;
    setEnable(true);
    txtUserID.Clear();
    txtUserName.Clear();
    txtFullName.Clear();
    txtPassword.Clear();
    txtPhone.Clear();
    txtEmail.Clear();
    txtDescription.Clear();
    chbStatus.Checked = true;
    txtFullName.Focus();
}
```

- Lập trình cho nút Cancel: nhấp đúp vào nút Cancel

```
private void btnCancel_Click(object sender, EventArgs e)
{
    setEnable(false);
}
```

- Lập trình cho nút Save: nhấp đúp vào nút Save

+ Vì nút Save sẽ ghi thông tin cho cả khi thêm mới và khi sửa, nên ta thực hiện như sau:

```
private void btnSave_Click(object sender, EventArgs e)
{
    if (txtFullName.Text.Trim() == "")
    {
        MessageBox.Show("Thông tin FullName không được để trống!", "Thông báo");
        txtFullName.Focus();
        return;
    }
    //Tương tự kiểm tra các TextBox khác
    string fn = txtFullName.Text;
    string un = txtUserName.Text;
    string pw = txtPassword.Text;
    string ph = txtPhone.Text;
    string em = txtEmail.Text;
    string de = txtDescription.Text;
    int st = (chbStatus.Checked) ? 1 : 0;
    if (AddNew)
    {
        //Phần này ghi khi nhấp vào nút thêm mới
    }
    else
    {
        //Phần này ghi khi nhấp vào nút sửa
    }
}
```

+ Lập trình phần thêm mới

```
if (AddNew)
{
    //Phần này ghi khi nhấp vào nút thêm mới
    string sql = string.Format("INSERT INTO Users (FullName, UserName, Password, Phone, Email, Status, Description) VALUES " +
        "(N'{0}',N'{1}',N'{2}',N'{3}',N'{4}',{5},N'{6}')" , fn, un, pw, ph, em, st, de);
    //Chú thích:
    //N'{0}' kiểu xâu thì phải dùng N và dấu ' ở hai đầu, kiểu số không cần
    //Các số là thứ tự của các biến truyền vào ở sau
    //chẳng hạn fn truyền dữ liệu vào số 0, pw --> 1, ph --> 2,...
    DBServices db = new DBServices();
    db.runQuery(sql);
    LoadGridData();
}
```

#### - Lập trình phần sửa

```
else
{
    //Phần này ghi khi nhấp vào nút sửa
    string id = txtUserID.Text;
    string sql = string.Format("UPDATE Users SET " +
        "FullName=N'{0}'," +
        "UserName=N'{1}'," +
        "Password=N'{2}'," +
        "Phone=N'{3}'," +
        "Email=N'{4}'," +
        "Status={5}," +
        "Description=N'{6}' WHERE UserID={7}", fn, un, pw, ph, em, st, de, id);
    DBServices db = new DBServices(); //3 lệnh sau có thể đưa ra ngoài dùng chung cho cả AddNew
    db.runQuery(sql);
    LoadGridData();
}
```

#### - Lập trình nút Edit: nhấp đúp vào nút Edit

```
private void btnEdit_Click(object sender, EventArgs e)
{
    AddNew = false;
    setEnable(true);
}
```

#### - Lập trình nút Delete: nhấp đúp vào nút Delete

```
private void btnDelete_Click(object sender, EventArgs e)
{
    //Thêm câu lệnh hỏi trước khi xóa vào đây
    string id = txtUserID.Text;
    string sql = string.Format("DELETE FROM Users WHERE UserID={0}", id);
    DBServices db = new DBServices();
    db.runQuery(sql);
    LoadGridData();
}
```

**Còn nhiều cách làm khác để thực hiện các nhiệm vụ trên, các bạn có thể tham khảo trong Slide ADO.NET**

**BÀI THỰC HÀNH CÔNG NGHỆ .NET #8**  
(Kết nối 1 bảng dữ liệu lên 1 Form)

**- Xây dựng các form kết nối 1 bảng dữ liệu còn lại tương tự hướng dẫn ở trên**

**BÀI THỰC HÀNH CÔNG NGHỆ .NET #9**  
(Kết nối nhiều bảng dữ liệu lên 1 Form)

**1. Thông tin về bảng dữ liệu**

10. Bảng **Rentals** lưu trữ thông tin cho thuê sách. Bảng gồm các trường:

Stt	Tên trường	Kiểu dữ liệu	Khóa/Ràng buộc	Ý nghĩa
1	<b>RentalID</b>	int	PK	Mã đơn hàng tự tăng
2	UserID	int	FK, Not null	Mã nhân viên
2	CustomerID	int	FK, Not null	Mã khách hàng
3	RentalDate	nvarchar(10)	Not null	Ngày thuê sách
4	ReturnDate	nvarchar(10)	Not null	Ngày trả sách
5	Deposit	float	Not null	Tiền đặt cọc
6	Description	nvarchar(250)		Ghi chú thêm nếu có

- Bảng **Rentals** có 2 khoá ngoại (FK) là **UserID**, **CustomerID** được liên kết đến hai bảng **Users** và **Customers** tương ứng.
- Để nhập dữ liệu cho bảng **Rentals**, ta cần lưu vào cơ sở dữ liệu **Mã nhân viên** (UserID) và **Mã khách hàng** (CustomerID). Nhưng khi hiển thị trên form thì hiển thị Tên nhân viên thay vì Mã nhân viên, Tên khách hàng thay vì Mã khách hàng (vì thông thường không ai nhớ mã).



Hiển thị tên nhân viên

Hiển thị tên khách hàng

**Ghi chú: Khi ghi vào bảng Rentals thì ghi Mã nhân viên và Mã khách hàng tương ứng vì thiết kế lưu trữ trong CSDL là mã**

RentalID

User Name lbUser

Customer Name

RentalDate

ReturnDate

Deposit

Description

Add New Save Cancel Edit Delete Exit

- Đặt tên theo thứ tự từ trên xuống:

- + txtRentalID (TextBox)
- + lbUser (ListBox)
- + cbCustomer (ComboBox)
- + txtRentalDate (TextBox)
- + txtReturnDate (TextBox)
- + txtDeposit (TextBox)
- + txtDescription (TextBox)
- + dataGridView1 (dataGridView)
- + btnAddNew (Button)
- + btnSave (Button)
- + btnSave (Button)
- + btnCancel (Button)
- + btnEdit (Button)
- + btnDelete (Button)
- + btnExit (Button)

2. Lập trình đưa dữ liệu lên cho các điều khiển

- Nhấp phải chuột vào form chọn View Source
- Thêm dòng DBServices db = new DBServices();

```
public partial class frmRentals : Form
{
    DBServices db = new DBServices();
    1 reference
    public frmRentals()
    {
        InitializeComponent();
    }

    1 reference
    private void setEnable(bool check)
    {
    }
}
```

- Thêm hàm setEnable(bool check): làm tương tự như bài thực hành 7 để có thể tác động hoặc không được tác động đến control.
- Tạo hàm lấy dữ liệu lên dataGridView

```
private void LayDuLieuGridView()
{
    string sql = "SELECT * FROM Rentals";
    dataGridView1.DataSource = db.getData(sql);
}
```

- Tạo hàm đưa dữ liệu từ bảng Users lên listBox có tên lbUser

```
private void LayDuLieuUser()
{
    string sql = "SELECT * FROM Users WHERE Status=1";
    lbUser.DisplayMember = "FullName";
    lbUser.ValueMember = "UserID";
    lbUser.DataSource = db.getData(sql);
}
```

- + Thuộc tính DisplayMember được gán cho trường FullName trong bảng Users, là trường sẽ hiển thị lên listBox
- + Thuộc tính ValueMember được gán cho trường UserID trong bảng User, là trường sẽ lấy giá trị tương ứng khi chọn vào tên trên listBox, giá trị này sẽ được lưu vào trong bảng Rental tương ứng.

Ví dụ:

- Giả sử trong bảng Users có thông tin như sau

UserID	FullName	....
1	Nguyễn Văn Anh	
2	Trần Văn Trung	
3	Phan Văn Hải	

- Khi hiển thị lên listbox lbUser sẽ hiện

Nguyễn Văn Anh
Trần Văn Trung
Phan Văn Hải

- Khi nhấp vào Trần Văn Trung, giá trị lbUser.SelectedValue sẽ trả về giá trị 2 tương ứng,

- Tạo hàm đưa dữ liệu từ bảng Customers lên ComboBox: cbCustomer

```
private void LayDuLieuCustomer()
{
    string sql = "SELECT * FROM Customers";
    cbCustomer.DisplayMember = "FullName";
    cbCustomer.ValueMember = "CustomerID";
    cbCustomer.DataSource = db.getData(sql);
}
```

+ DisplayMember và ValueMember tương tự như trên

- Gọi các hàm trên trong sự kiện Form\_Load
  - + Chọn form giao diện, nhấp đúp chuột vào form
  - + Thêm đoạn chương trình sau trong sự kiện Form\_Load

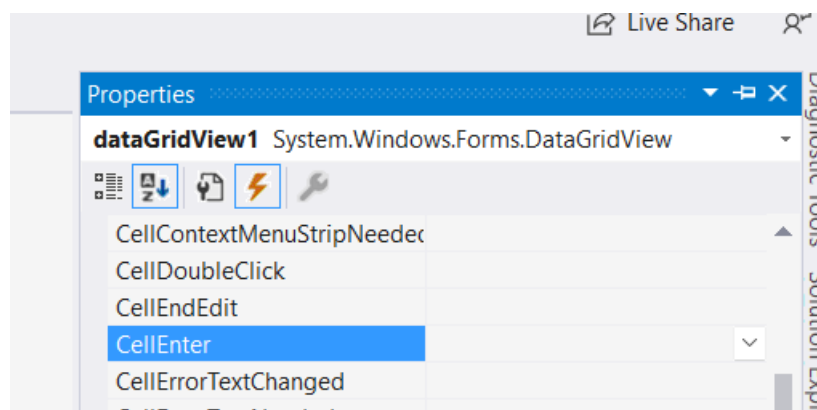
```
private void frmRentals_Load(object sender, EventArgs e)
{
    setEnable(false);

    LayDuLieuGridView();
    LayDuLieuUser();
    LayDuLieuCustomer();
}
```

- Nhập vài bản ghi trong các tệp **Users**, **Customers** và **Rentals** để kiểm thử.
- Chạy thử giao diện xem kết quả

RentalID	UserID	CustomerID	RentalDate	ReturnDate
1	1	2	10.10.2021	12.10.2021
2	12	1	05.03.2022	10.03.2022
3	3	2	10.04.2022	15.04.2022

- Dữ liệu đã được đưa lên các control.
- 3. Tạo sự kiện CellEnter cho dataGridView1, khi nhấp vào mỗi dòng, dữ liệu sẽ đưa lên các điều khiển bên tay trái.
- Chọn sự kiện CellEnter của dataGridView1, nhấp đúp chuột



- Thêm nội dung sau trong sự kiện CellEnter của dataGridView1

```
private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    int i = e.RowIndex;
    if (i >= 0)
    {
        txtRentalID.Text = dataGridView1.Rows[i].Cells["RentalID"].Value.ToString();
        lbUser.SelectedValue = int.Parse(dataGridView1.Rows[i].Cells["UserID"].Value.ToString());
        cbCustomer.SelectedValue = int.Parse(dataGridView1.Rows[i].Cells["CustomerID"].Value.ToString());
        txtRentalDate.Text = dataGridView1.Rows[i].Cells["RentalDate"].Value.ToString();
        //Nội dung khác làm tương tự ở đây
    }
}
```

#### 4. Thêm bản ghi mới

- Thêm biến addNew kiểu boolean vào đây

```
public partial class frmRentals : Form
{
    DBServices db = new DBServices();
    bool addNew = false;
    1 reference
    public frmRentals()
    {
        InitializeComponent();
    }
}
```

- Nếu biến này đặt bằng true ta xem đó là thao tác thêm mới, đặt bằng false ta xem đó là thao tác sửa.

- Nhấp đúp vào nút btnAddNew, thêm vào dòng lệnh

```
private void btnAddNew_Click(object sender, EventArgs e)
{
    txtRentalID.Clear();
    txtRentalDate.Clear();
    txtReturnDate.Clear();
    //Tương tự cho các txt khác
    addNew = true;
    setEnable(true);
}
```

- Nhấp đúp vào nút btnSave, thêm vào dòng lệnh

```
90 private void btnSave_Click(object sender, EventArgs e)
91 {
92     if (addNew)
93     {
94         string us = lbUser.SelectedValue.ToString(); //Để ý dòng này
95         string cn = cbCustomer.SelectedValue.ToString(); //và dòng này
96         string rd = txtRentalDate.Text;
97         string re = txtReturnDate.Text;
98         string dp = txtDeposit.Text;
99         string de = txtDescription.Text;
100        string sql = string.Format("INSERT INTO Rentals VALUES ({0},{1},N'{2}',N'{3}',{4},N'{5}']",
101            us, cn, rd, re, dp, de);
102        db.runQuery(sql);
103        LayDuLieuGridView();
104    }
105 }
```

- Dòng 94 và 95 lấy giá trị tương ứng khi chọn vào tên, giá trị tương ứng được lấy từ lệnh lbUser.ValueMember và cbCustomer.ValueMember đã được gán ở trên.

- Xoá 1 bản ghi, nhấp đúp vào nút btnDelete

```
private void btnDelete_Click(object sender, EventArgs e)
{
    string id = txtRentalID.Text;
    string sql = string.Format("DELETE FROM Rentals WHERE RentalID={0}", id);
    if (MessageBox.Show("Bạn muốn xoá bản ghi này không?", "Thông báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Error) == DialogResult.Yes)
    {
        db.runQuery(sql);
        LayDuLieuGridView();
    }
}
```

- Sửa thông tin

+ Nhấp đúp vào nút btnEdit

```
private void btnEdit_Click(object sender, EventArgs e)
{
    addNew = false;
    setEnable(true);
}
```

+ Trong nút btnSave

```
private void btnSave_Click(object sender, EventArgs e)
{
    if (addNew)
    {
        string us = lbUser.SelectedValue.ToString(); //Để ý dòng này
        string cn = cbCustomer.SelectedValue.ToString(); //và dòng này
        string rd = txtRentalDate.Text;
        string re = txtReturnDate.Text;
        string dp = txtDeposit.Text;
        string de = txtDescription.Text;
        string sql = string.Format("INSERT INTO Rentals VALUES ({0},{1},N'{2}',N'{3}',{4},N'{5}')",
            us, cn, rd, re, dp, de);
        db.runQuery(sql);
        LayDuLieuGridView();
    }
    else
    {
        //lệnh sửa làm ở đây, tương tự bài trước
    }
}
```

- Các nút khác làm tương tự bài thực hành trước

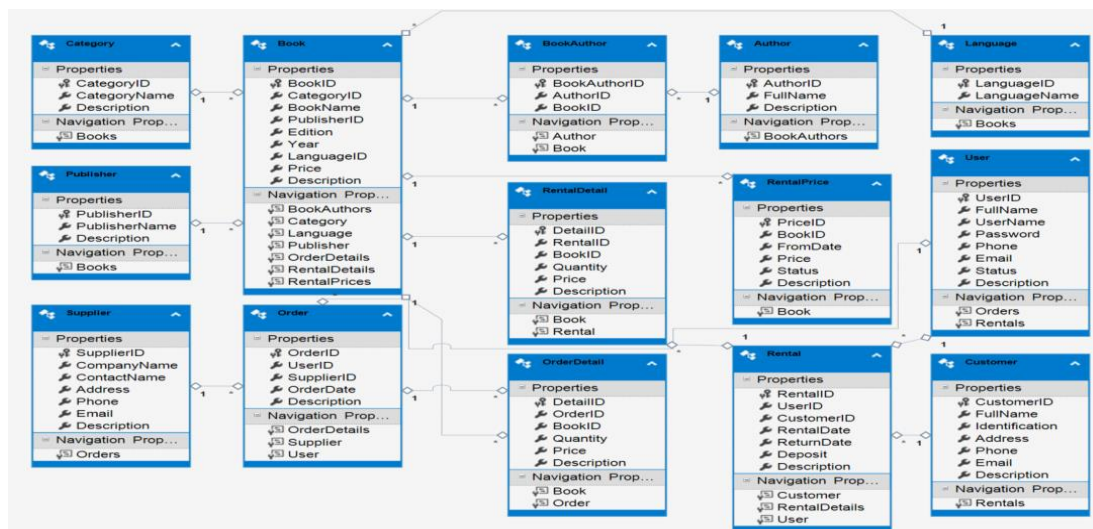
## BÀI THỰC HÀNH CÔNG NGHỆ .NET #10 (Kết nối cơ sở dữ liệu với LINQ) LINQ to Entities

### 1. Xây dựng mô hình thực thể

(Video tham khảo <https://www.youtube.com/watch?v=NopEZ7XI5JU>)

- Tạo tệp mô hình dữ liệu thực thể ADO.NET: Chọn menu Project, chọn Add > New Item.
- Chọn ADO.NET Entity Data Model, nhập tên tệp (ví dụ BooksEntities.edmx) và chọn Add.
- Chọn kiểu tạo mô hình dữ liệu thực thể: Trong cửa sổ Entity Data Model Wizard, chọn EF Designer from database và chọn Next.
- Thiết lập kết nối tới cơ sở dữ liệu: Chọn New Connection, xuất hiện cửa sổ Connection Properties.
- Chọn Server name > Authentication > Database name > chọn Ok.
- Chọn kiểu lưu trữ cấu trúc kết nối cơ sở dữ liệu: Chọn Yes, include the sensitive data và chọn Next.
- Nhập tên mô hình thực thể: Chọn Save entity connection settings, nhập tên mô hình thực thể (ví dụ BookEntities) và chọn Next.
- Chọn Entity Framework 6.x/ Entity Framework 5.x.
- Chọn các đối tượng của cơ sở dữ liệu: Chọn Tables, chọn các bảng dữ liệu, nhập không gian tên chứa mô hình thực thể (ví dụ BooksModel) và chọn Finish.
- Biên dịch mô hình: Chọn thực đơn Build > Build Solution.

Mô hình quan hệ giữa các lớp thực thể được tạo như sau



## 2. Tạo form như sau

The screenshot shows a Windows application window titled "frmUserLINQ". Inside the window, there is a form with the following elements:

- Seven text input fields stacked vertically on the left side, each with a label to its left: "UserID", "FullName", "User name", "Password", "Phone", "Email", and "Status".
- Below the input fields, there are six buttons arranged in three rows:
  - Row 1: "Add New" and "Delete"
  - Row 2: "Save" and "Cancel"
  - Row 3: "Edit" and "Exit"
- A large, empty gray rectangular area occupies the right half of the form.

+ txtUserID	TextBox
+ txtFullName	TextBox
+ txtUserName	TextBox
+ txtPassword	TextBox
+ txtPhone	TextBox
+ txtEmail	TextBox
+ txtStatus	TextBox
+ btnAddNew	Button
+ btnDelete	Button
+ btnSave	Button
+ btnCancel	Button
+ btnEdit	Button
+ btnExit	Button



+ dataGridView1      dataGridView

### 3. Khai báo thông tin

- Thêm vài đối tượng BookEntities trong lớp của form

```
BooksEntities myBook = new BooksEntities();  
private bool AddNew = false;
```

### 4. Lấy dữ liệu lên dataGridView

```
private void frmUserLINQ_Load(object sender, EventArgs e)  
{  
    LayDuLieuGridView();  
}
```

3 references

```
private void LayDuLieuGridView()  
{  
    var queryUser = from item in myBook.Users  
                     select item;  
    dataGridView1.DataSource = queryUser.ToList();  
}
```

### 4. Sự kiện cell\_enter của dataGridView

```
private void dataGridView1_CellEnter(object sender, DataGridViewCellEventArgs e)  
{  
    int i = e.RowIndex;  
    txtUserID.Text = dataGridView1.Rows[i].Cells["UserID"].Value.ToString();  
    //Tương tự trường còn lại  
}
```

### 5. Nhấn vào nút Add new

```
private void btnAddNew_Click(object sender, EventArgs e)  
{  
    AddNew = true;  
    setEnable(false);  
}  
  
1 reference  
private void setEnable(bool check)  
{  
    //Viết code tương tự các bài trước  
}
```

## 6. Nhấp vào nút Save

```
private void btnSave_Click(object sender, EventArgs e)
{
    //kiểm tra dữ liệu nhập ở đây
    if (AddNew)
    {
        int st = int.Parse(txtStatus.Text);
        User u1 = new User();
        u1.FullName = txtFullName.Text;
        u1.UserName = txtUserName.Text;
        u1.Password = txtPassword.Text;
        u1.Phone = txtPhone.Text;
        u1.Email = txtEmail.Text;
        u1.Status = st;
        myBook.Users.Add(u1);
        myBook.SaveChanges();
        LayDulieuGridView();
    }
    else
    {
    }
}
```

- Trong phần else của if AddNew

```
else
{
    int id = int.Parse(txtUserID.Text);
    var queryUsers = from item in myBook.Users
                      where (item.UserID == id)
                      select item;

    //Lấy đối tượng đầu tiên trong bảng dữ liệu truy vấn, ví dụ:
    User itemUse = queryUsers.First();
    //Gán giá trị mới cho đối tượng lớp thực thể, ví dụ:
    int st = int.Parse(txtStatus.Text);
    itemUse.FullName = txtFullName.Text;
    itemUse.UserName = txtUserName.Text;
    itemUse.Password = txtPassword.Text;
    itemUse.Phone = txtPhone.Text;
    itemUse.Email = txtEmail.Text;
    itemUse.Status = st;
    //Cập nhật lại bảng dữ liệu trong CSDL:
    myBook.SaveChanges();
}
```

## 7. Xoá dữ liệu có mã ID

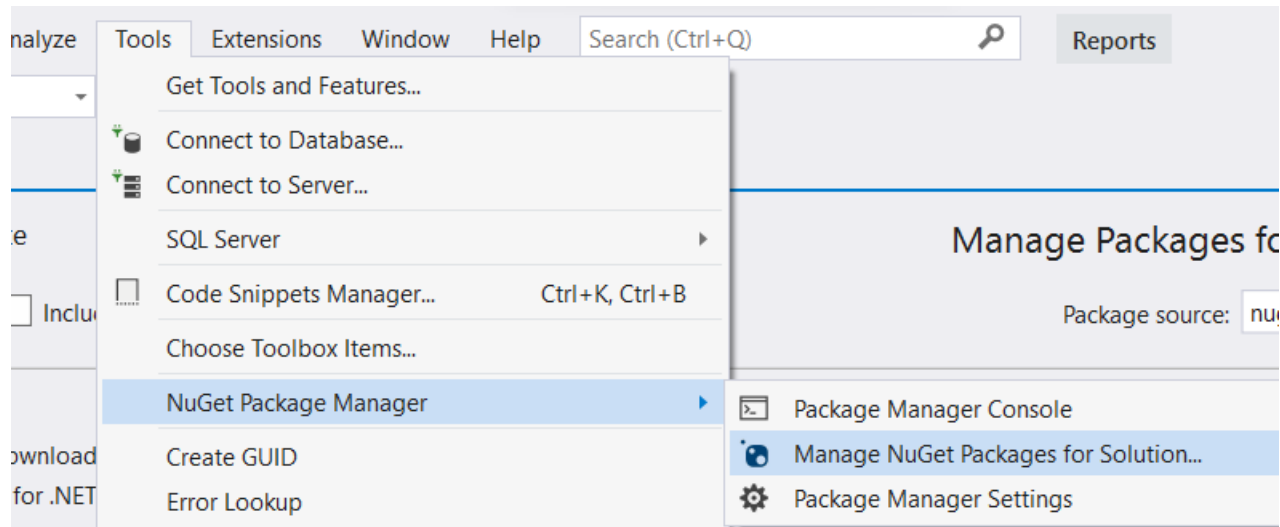
- Nhấp đúp vào nút xoá (btnDelete)

```
private void btnDelete_Click(object sender, EventArgs e)
{
    try
    {
        int id = int.Parse(txtUserID.Text);
        var queryDelete = from item in myBook.Users
                          where (item.UserID == id)
                          select item;
        myBook.Users.Remove(queryDelete.First());
        myBook.SaveChanges();
        LayDuLieuGridView(); // nạp lại dữ liệu sau khi xóa
    }
    catch (Exception myException)
    {
        MessageBox.Show(myException.Message);
    }
}
```

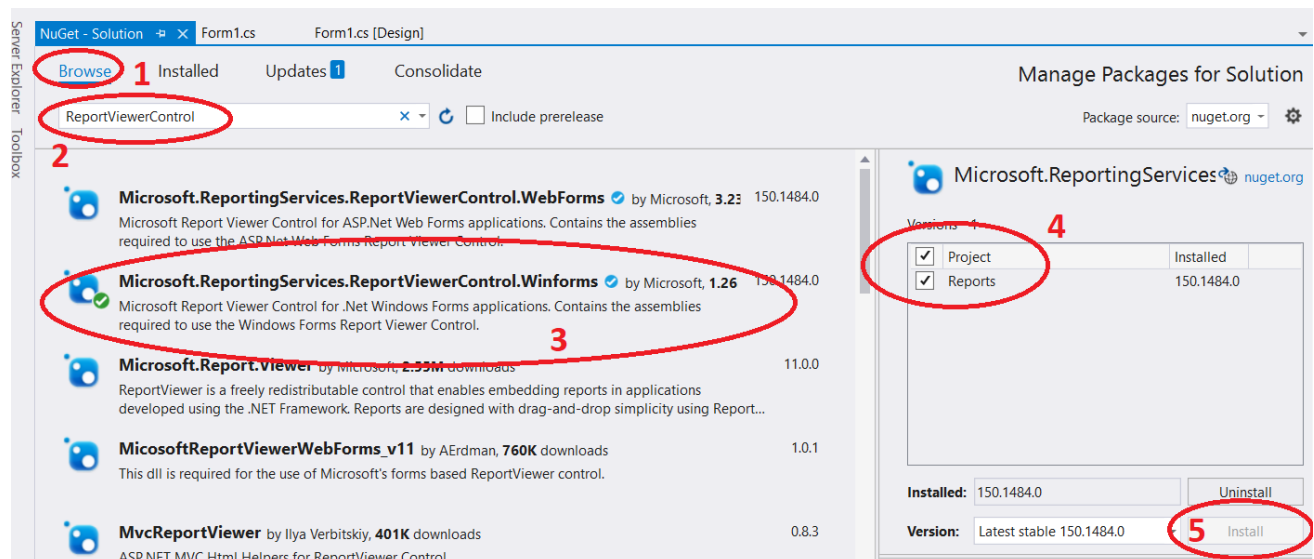
## BÀI THỰC HÀNH CÔNG NGHỆ .NET #11 (Tạo Report)

### 1. Cài đặt ReportViewer

Cách 1: Vào Manage NuGet theo hướng dẫn sau



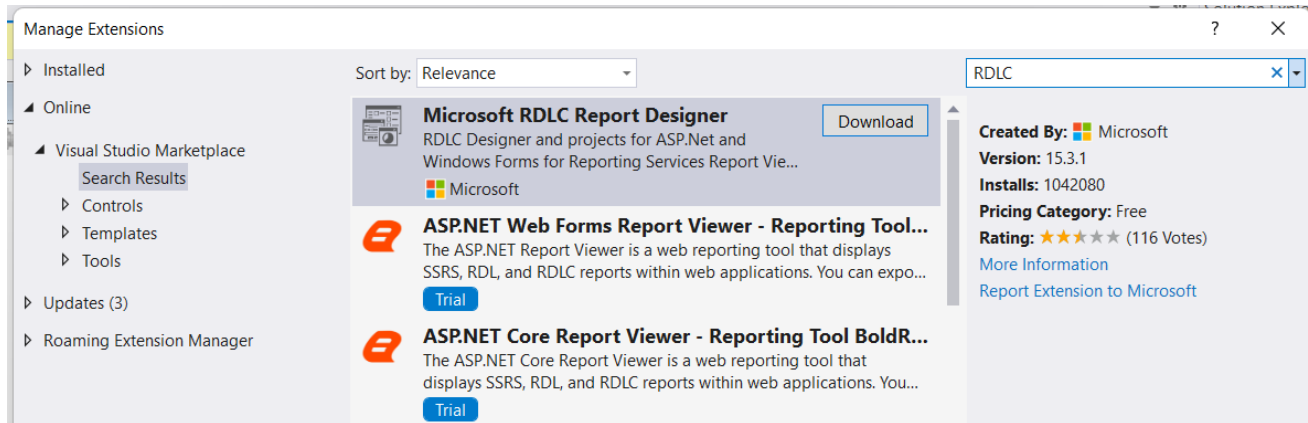
Thực hiện các bước như hình sau



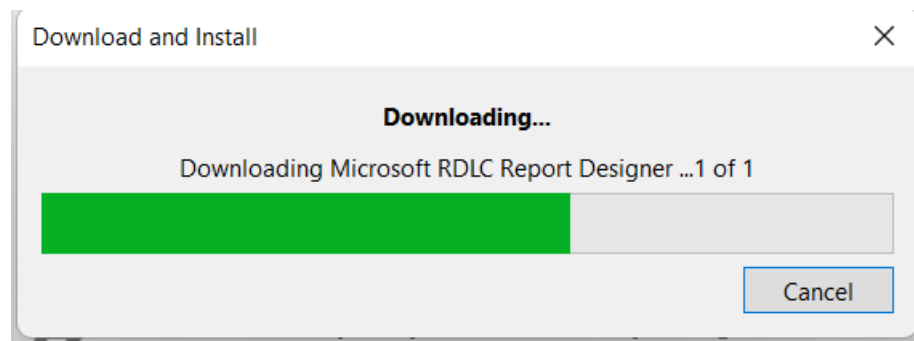
Sau khi cài đặt xong, khởi động lại Visual Studio để có ReportViewer

## 2. Cài đặt RDLC Report Designer

Vào menu Extensions/Manage Extensions, Gõ RDLC trong hộp search, chọn **Download** tại mục **Microsoft RDLC Report Designer**

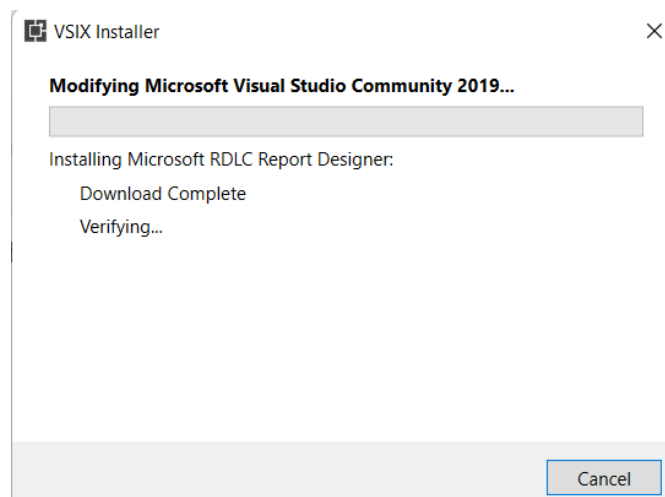


Quá trình thực hiện tải và cài đặt



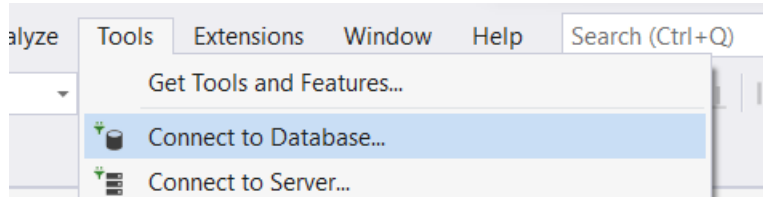
Đóng Visual Studio 2019 để tiếp tục quá trình cài đặt

Xuất hiện hộp thoại, chọn Modify, quá trình thực hiện xong, chọn Close.

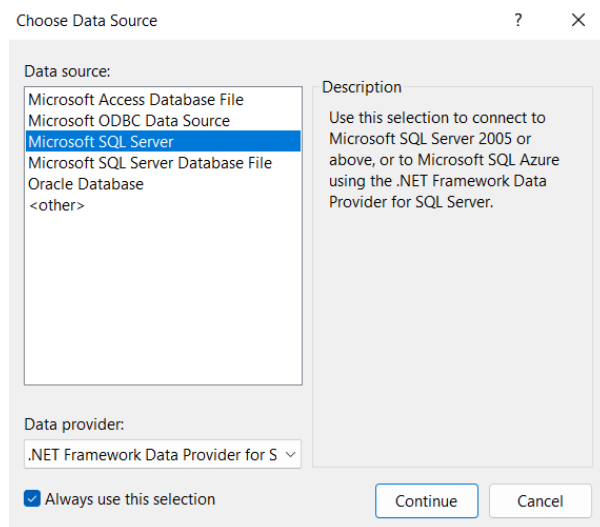


### 3. Kết nối cơ sở dữ liệu SQL Server

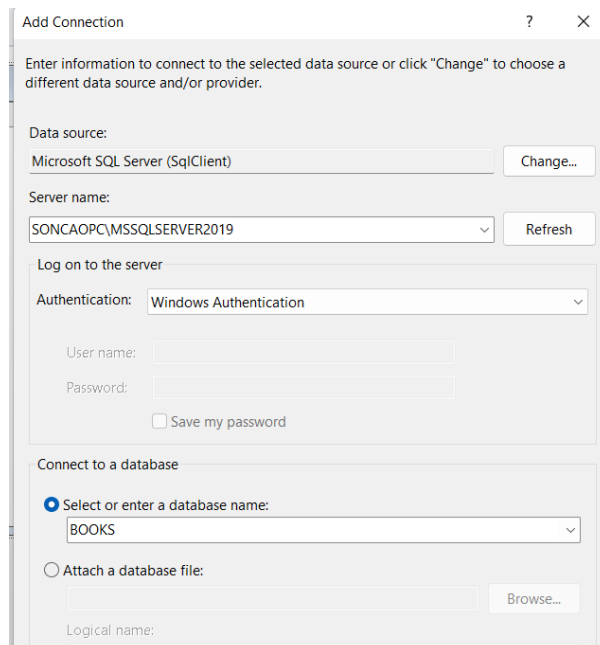
#### Tools/Connect to Database

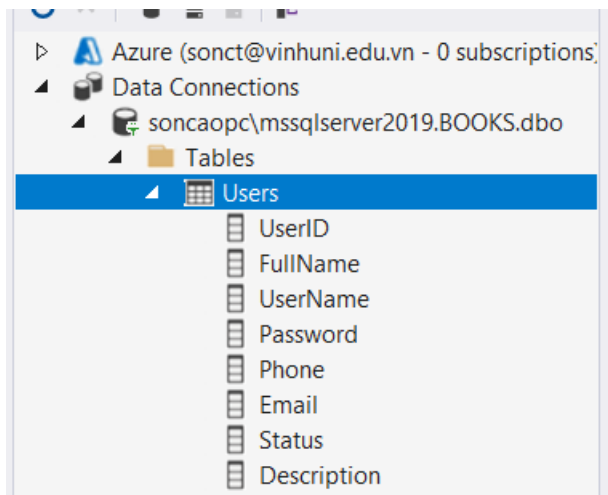


#### SQL Server > Continue



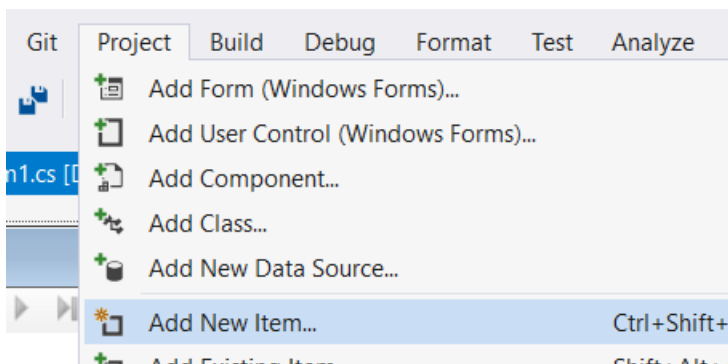
#### Gõ các tham số



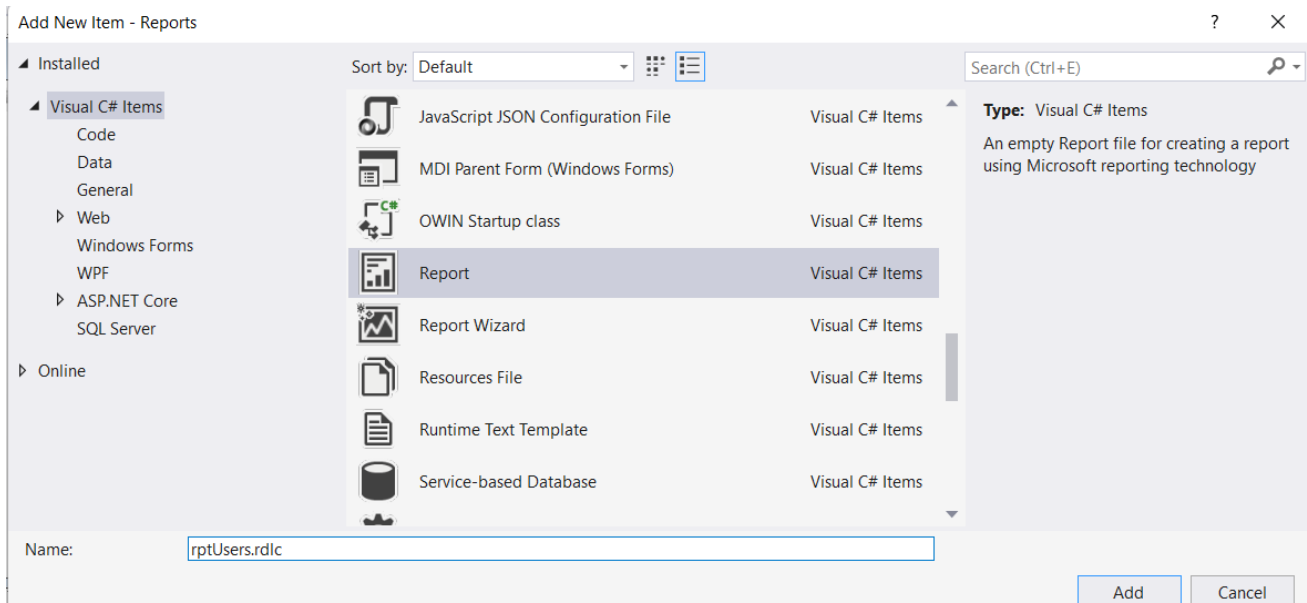


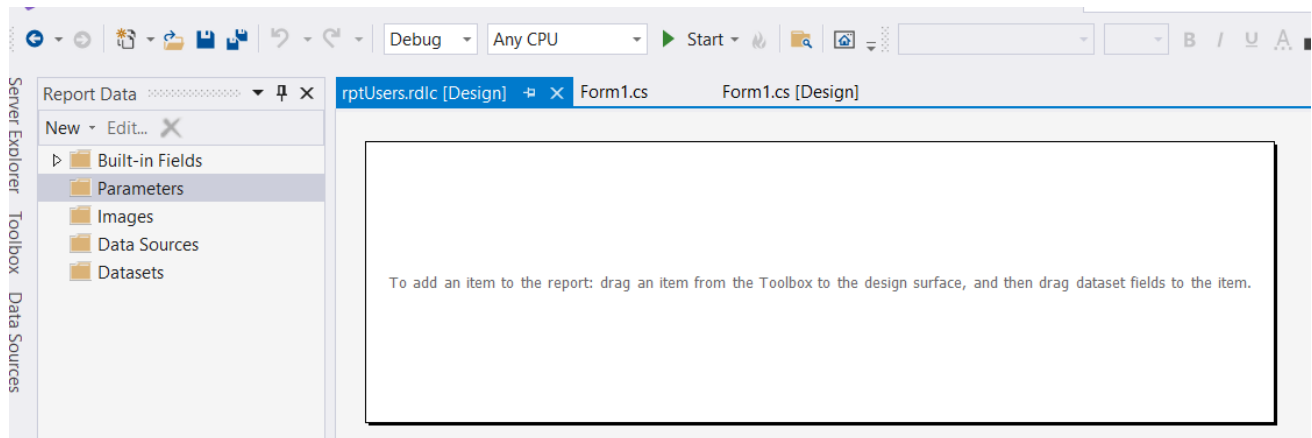
#### 4. Tạo report

Project/Add New Item...



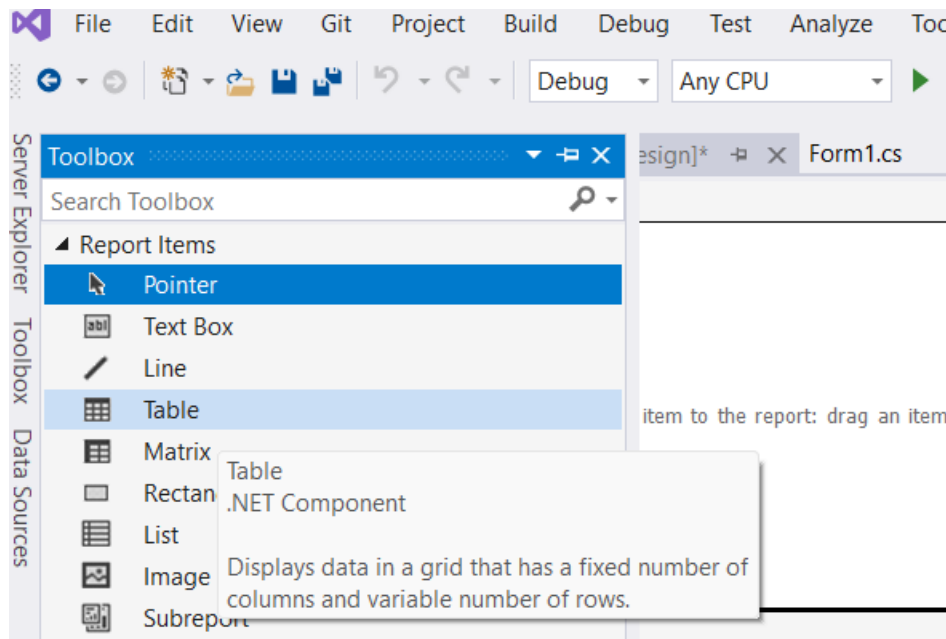
Chọn Report > Đặt tên (rptUsers.rdlc) > Add





Sau đây là một cách đưa dữ liệu lên report sử dụng bảng:

Kéo bảng ra report




Chọn dữ liệu cho bảng




?


×




**Choose a Data Source Type**

Where will the application get data from?

  
Database

  
Service

  
Object

Chọn Connection đã tạo ở bước trên



### Choose Your Data Connection

Which data connection should your application use to connect to the database?

soncaopc\mssqlserver2019.BOOKS.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password), which is required to connect to the database. However, storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

- ☐ No, exclude sensitive data from the connection string. I will set this information in my application code.
- ☐ Yes, include sensitive data in the connection string.

☒ Show the connection string that you will save in the application

Data Source=SONCAOPC\MSSQLSERVER2019;Initial Catalog=BOOKS;Integrated Security=True

< Previous

Next >

Finish

Cancel

### Chọn bảng cần đưa thông tin/Finish



### Choose Your Database Objects

Which database objects do you want in your dataset?

- ☒ Tables
- ☒ Users
- ☐ Views
- ☐ Stored Procedures
- ☐ Functions

DataSet name:

BOOKSDataset

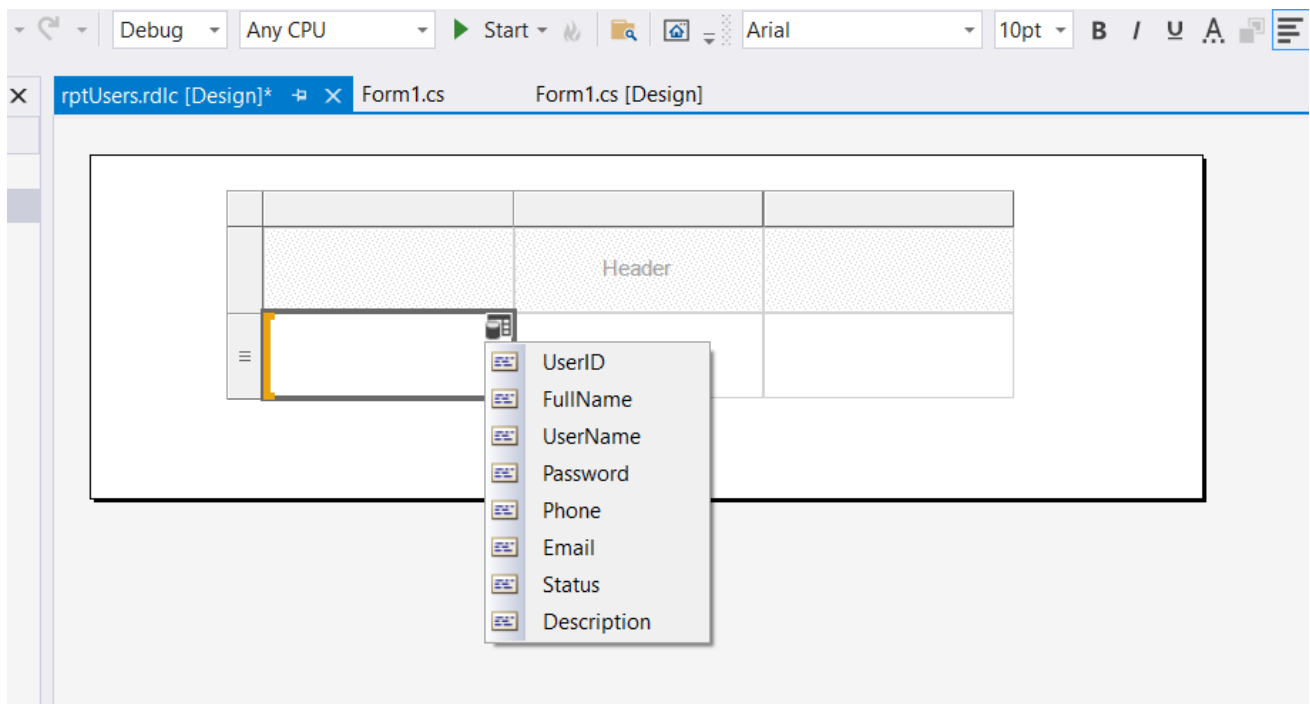
< Previous

Next >

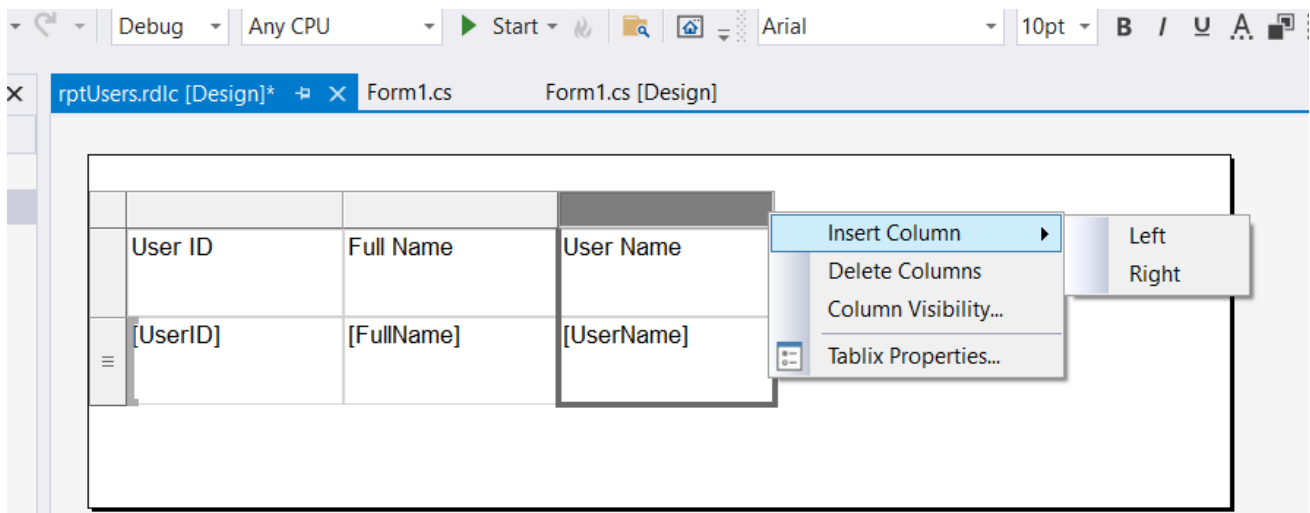
Finish

Cancel

35



Có thể thay đổi kích thước, thêm bớt cột tùy ý (nhấp chuột phải, chọn các chức năng)



Thêm các định dạng khác và trang trí

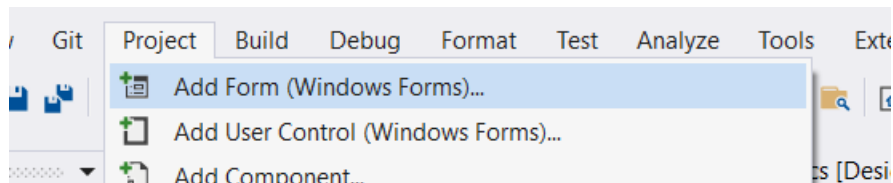
**DANH SÁCH NGƯỜI DÙNG**

User ID	Full Name	User Name	Phone
[UserID]	[FullName]	[UserName]	[Phone]

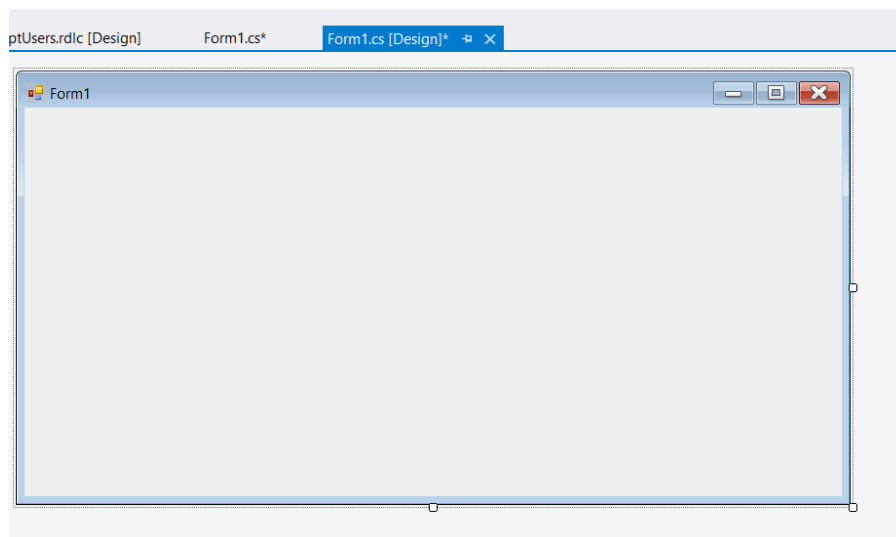
Ngày ..... tháng .... năm 2022

## 5. Tạo form mới để hiển thị report đã tạo

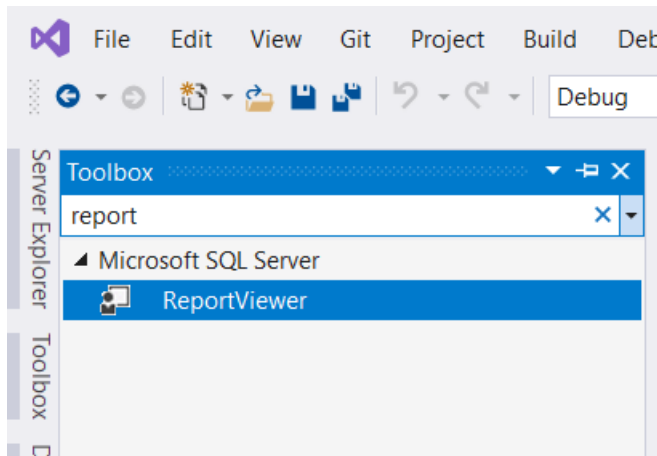
Project/Add Form



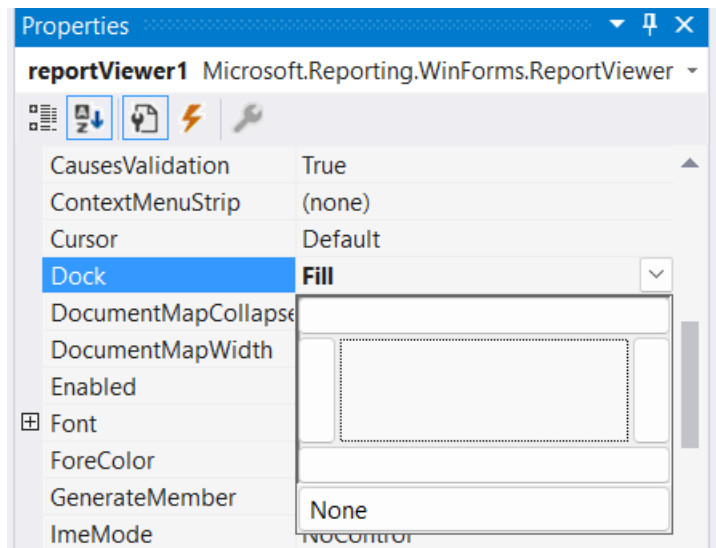
Ta có form



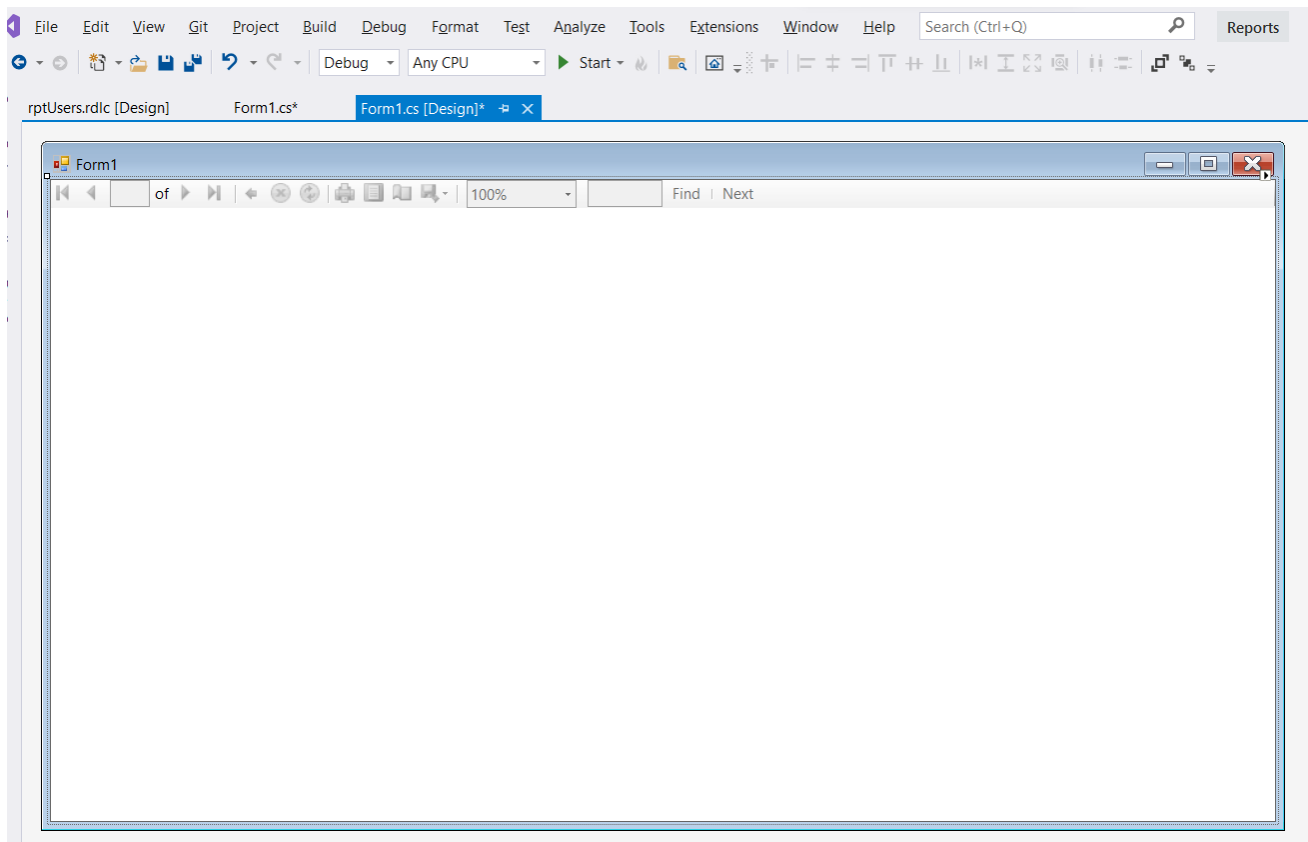
Vào Toolbox, chọn ReportViewer, kéo thả ra form



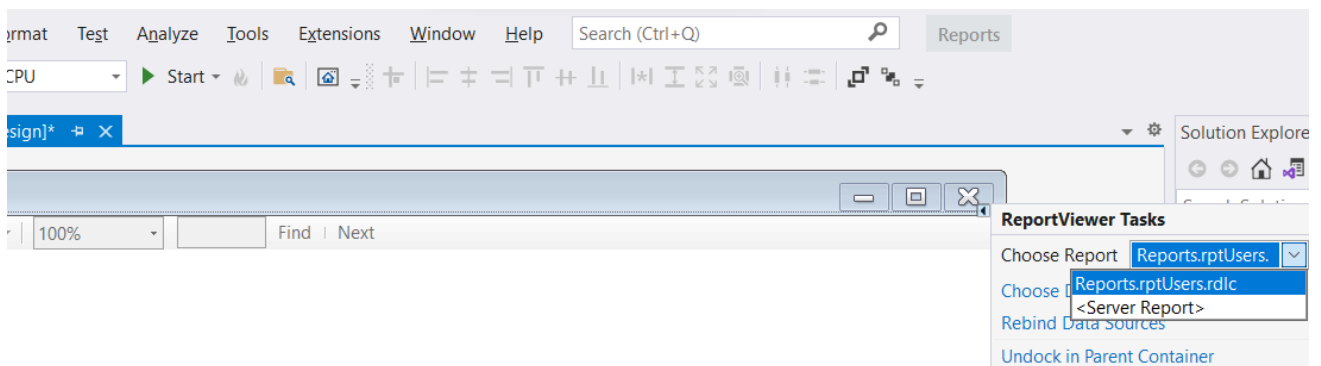
Sau khi có reportViewer, chọn thuộc tính Dock với giá trị Fill



Ta có kết quả



Nhấp vào mũi tên ở góc trên phải của ReportViewer, trong Choose Report chọn report đã tạo ở phần trước (Reports.rptUser.rdlc)



Sau khi xong, chạy thử form để xem kết quả.

Form1

1 of 1 100% Find | Next

### DANH SÁCH NGƯỜI DÙNG

User ID	Full Name	User Name	Phone
1	Nguyễn Thanh An	annt	0912345667
2	Lê Văn Dũng	dunglv	0913123321
3	Đinh Thị Quyên	quyendt	0914987678
4	Trần Nhâm Tiến	tientn	0912666554
5	Lê Quốc Vinh	vinhlq	0913876678

Ngày ..... tháng .... năm 2022

Trên report có thể in, phóng to thu nhỏ, tìm kiếm, xuất ra tệp excel, pdf,...

**6. Link tham khảo:** <https://www.youtube.com/watch?v=6GELtQpd7wc>



## BÀI THỰC HÀNH CÔNG NGHỆ .NET #12 (Một số nội dung mở rộng)

### 1 Thao tác dữ liệu với Excel

Link tham khảo

<https://howkteam.vn/course/radius-textbox-wpf-with-text-hint/thao-tac-voi-excel-trong-c-1575>

### 2. Xây dựng menu trong C# (Đã thực hành)

<https://plpsoft.vn/30241-Bai-tap-C-Bai-10-Cach-dung-MenuStrip-trong-C-winforms>

### 3. Xây dựng context menu trong C#

<https://freetuts.net/contextmenustrip-trong-c-sharp-3490.html>

### 4. Đóng gói sản phẩm

Link tham khảo: <https://www.youtube.com/watch?v=Ap2lgauS-ZE>