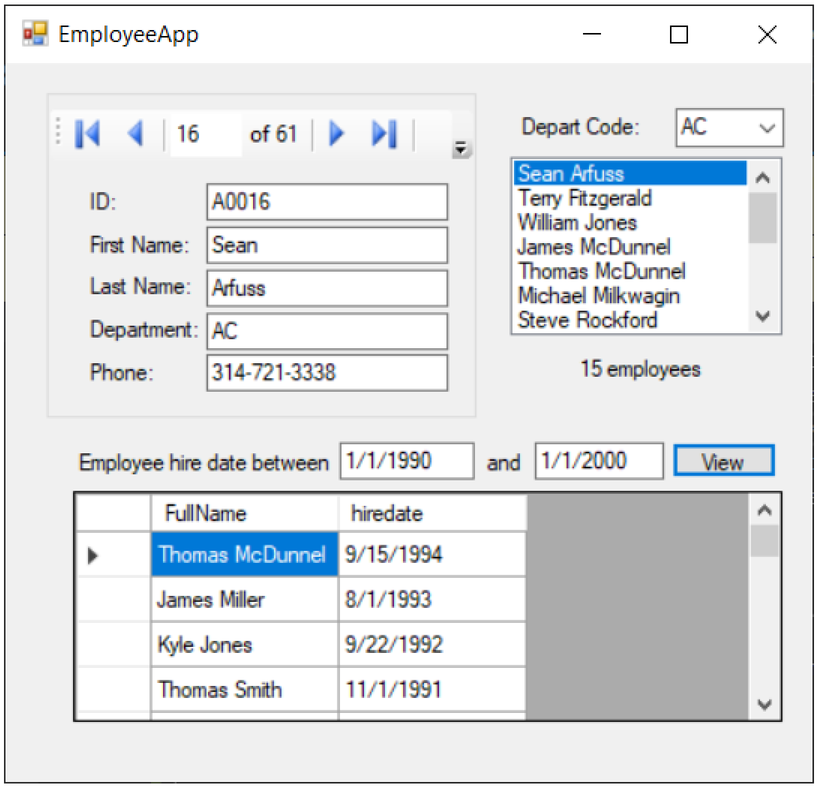
CIS 284 – EmployeeApp

In this enhanced version of the EmployeeApp, you will create a form that is bound to a database table containing employee data.



EmployeeApp

16 of 61

ID:	A0016
First Name:	Sean
Last Name:	Arfuss
Department:	AC
Phone:	314-721-3338

Depart Code: AC

Sean Arfuss
Terry Fitzgerald
William Jones
James McDunnel
Thomas McDunnel
Michael Milkwagin
Steve Rockford

15 employees

Employee hire date between 1/1/1990 and 1/1/2000 View

FullName	hiredate
Thomas McDunnel	9/15/1994
James Miller	8/1/1993
Kyle Jones	9/22/1992
Thomas Smith	11/1/1991

Create a data model that includes the Employee table in the *Practice.mdf* database provided. Bind a set of TextBox controls to the Employee table for viewing employee details. Include a BindingNavigator for navigating through the employee records. Save changes when leaving each record.

Add a *FullName* property to the Employee class that returns the concatenation of FirstName and LastName, as shown above.

Use LINQ method syntax to achieve the following.

- Provide a dropdown with a set of distinct department codes from the employee table.
- When a department code is selected from the list, display the names of all employees from the selected department in a ListBox in order by last name.
- Display the number of employees appearing in the list on the form.
- Change the current employee detail when a name is selected from the list.
- Use a TextBox, Button, and a DataGridView to list employees hired within the specified date range in descending order by hiredate. Note: One way to compare a DateTime property (x) to a literal date is to use the Parse method, as follows: x > DateTime.Parse("mm/dd/yyyy").