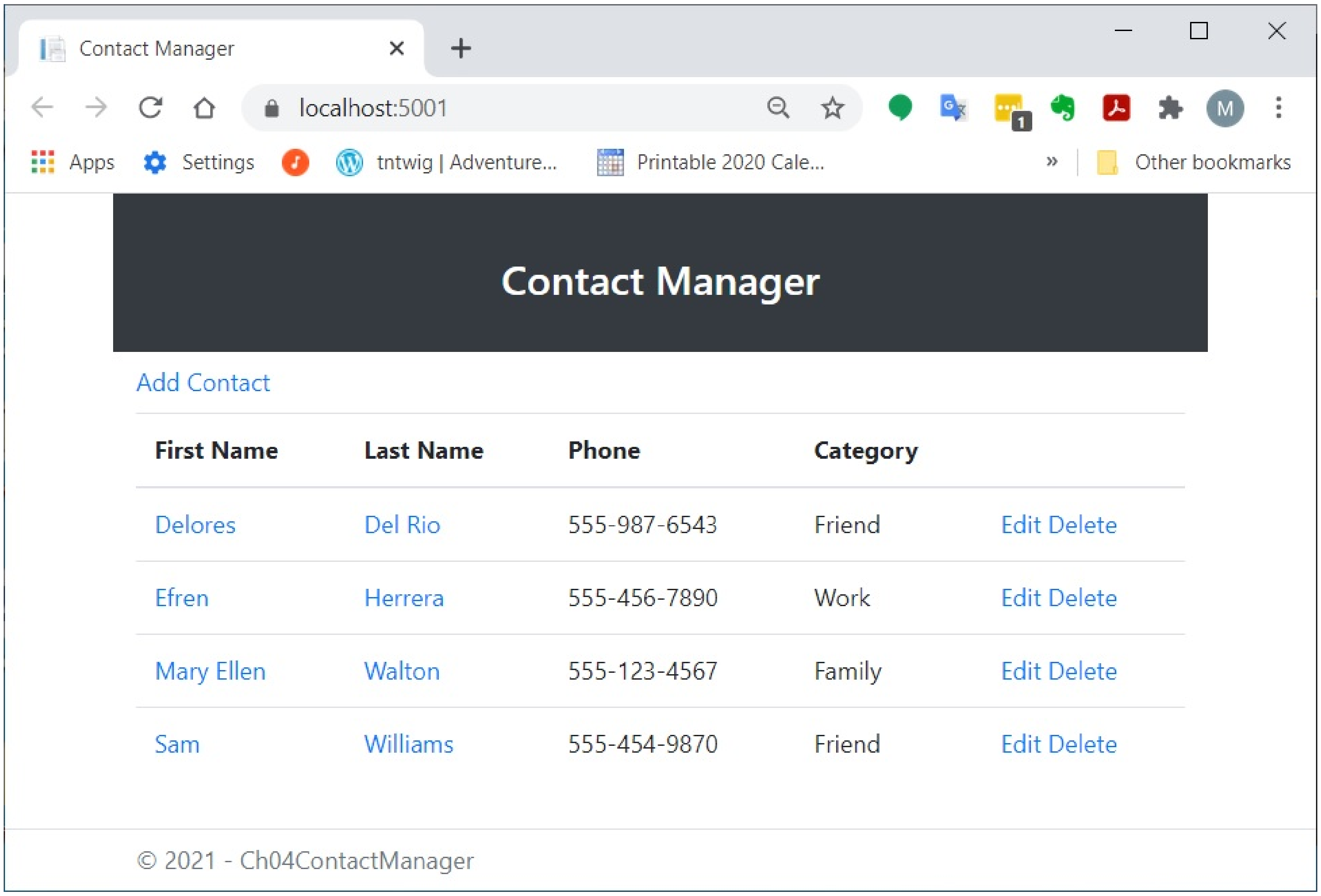
Homework 4-1a Build the Contact Manager app

In this assignment, you will build the first phase of a multi-page, data-driven Contact Manager app like the one that is shown below. An end-user will be able to see a list of their contacts and be able to add new contacts, edit contact information, and delete contacts.

Below are the screen prints for the entire application to give you an overview of what will be completed over the course of two homework assignments.

For this assignment, you will complete the Database, Home controller, and Home Page (Index view) for the Contact Manager app. The add, edit, and delete contacts will be developed in a future assignment.

The Home page



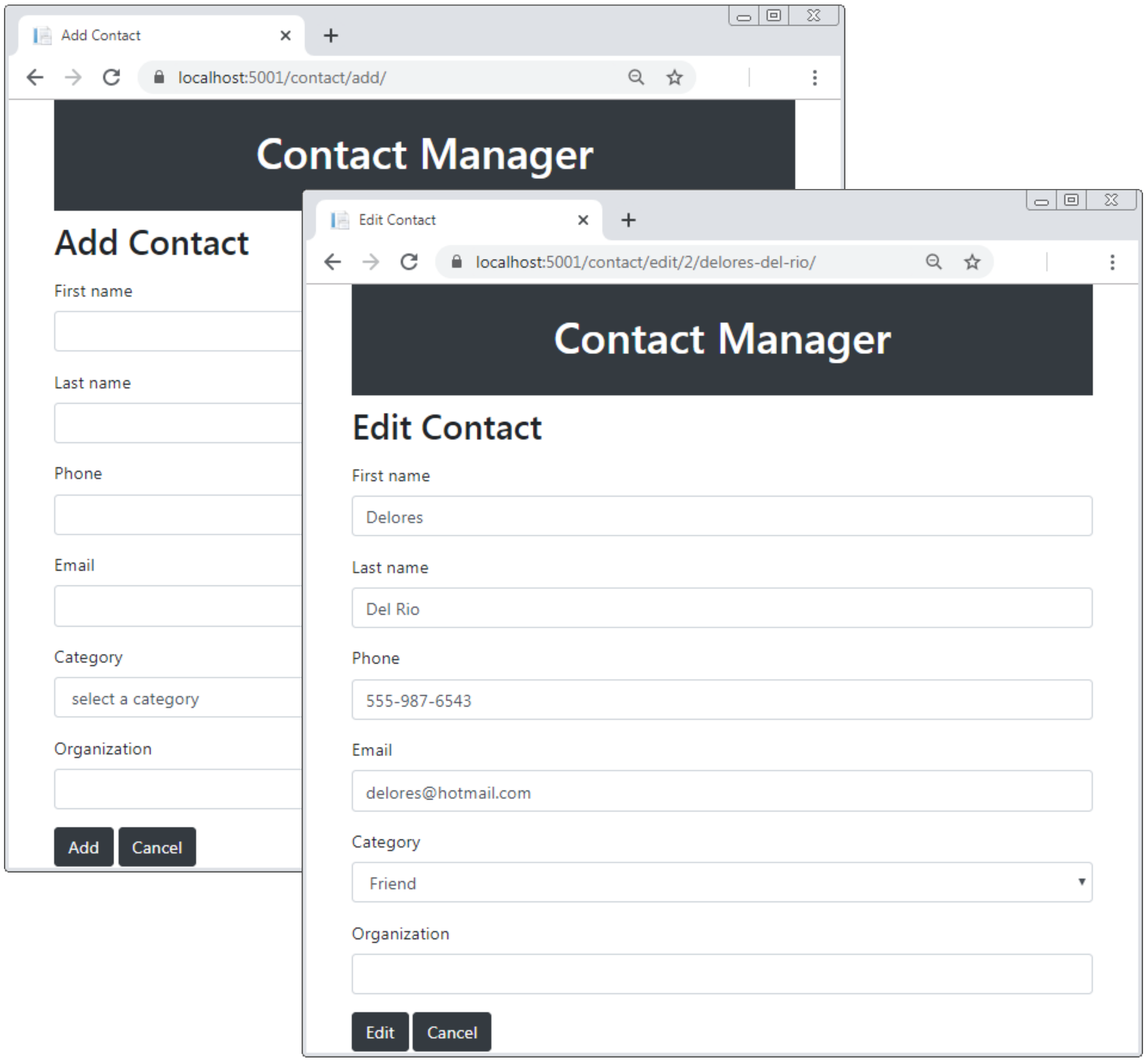
Contact Manager

Add Contact

First Name	Last Name	Phone	Category	
Delores	Del Rio	555-987-6543	Friend	Edit Delete
Efren	Herrera	555-456-7890	Work	Edit Delete
Mary Ellen	Walton	555-123-4567	Family	Edit Delete
Sam	Williams	555-454-9870	Friend	Edit Delete

© 2021 - Ch04ContactManager

The Add and Edit pages (both use the same view)


Add Contact
localhost:5001/contact/add/

Contact Manager

Add Contact

First name

Last name

Phone

Email

Category

select a category

Organization

Edit Contact
localhost:5001/contact/edit/2/delores-del-rio/

Contact Manager

Edit Contact

First name

Delores

Last name

Del Rio

Phone

555-987-6543

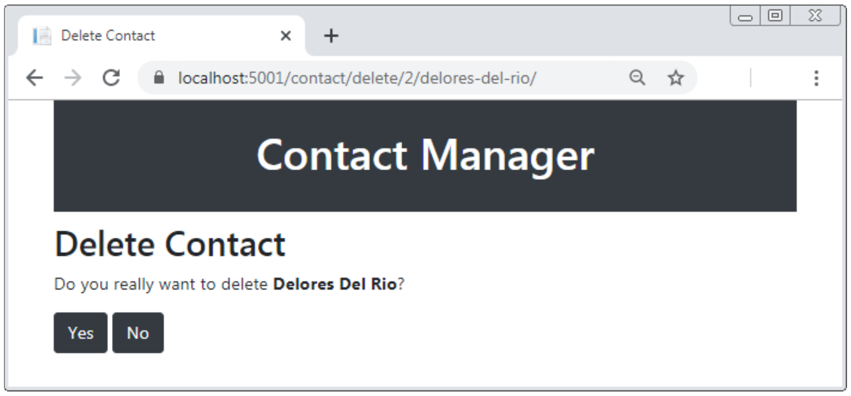
Email

delores@hotmail.com

Category

Friend

Organization



Contact Manager

Delete Contact

Do you really want to delete **Delores Del Rio**?

Specifications

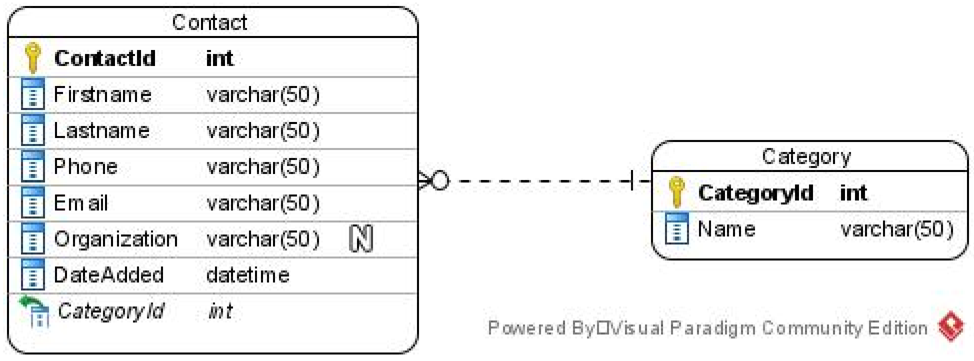
1. Create a new ASP.NET Core Web Application in your INFS 4950 folder with a project name of Ch04ContactManager and the solution name of Ch04ContactManager. Base this app on the Web Application (Model-View-Controller) template.

2. Remove all unnecessary files such Models/ErrorViewModel.cs, Views/Shared/Error.cshtml, Views/Home/Privacy.cshtml, etc.

3. Install the EF Core SqlServer and EF Core Tools NuGet packages.
4. You may accept the default installation of Bootstrap and client-side libraries, or you may instead choose to use a CDN. It is your choice.
5. Modify the _Layout.cshtml to remove the code within the <header>… </header> element and within the <header> </header> element include the following **<h1> Contact Manager </h1>**. The screenshot of the home page above shows the image of a header that includes a bootstrap container, dark background, centered, padding of 4 on all four sides, bottom margin of 2, and white text. You may format the header element of the _Layout to this same specification, or you may make any changes so that the _Layout looks the way you want it to appear to the end-user. (Note: You may also refer to the Movie List practice exercise as an example).
6. Remove all action methods from the HomeController except for the Index() method.

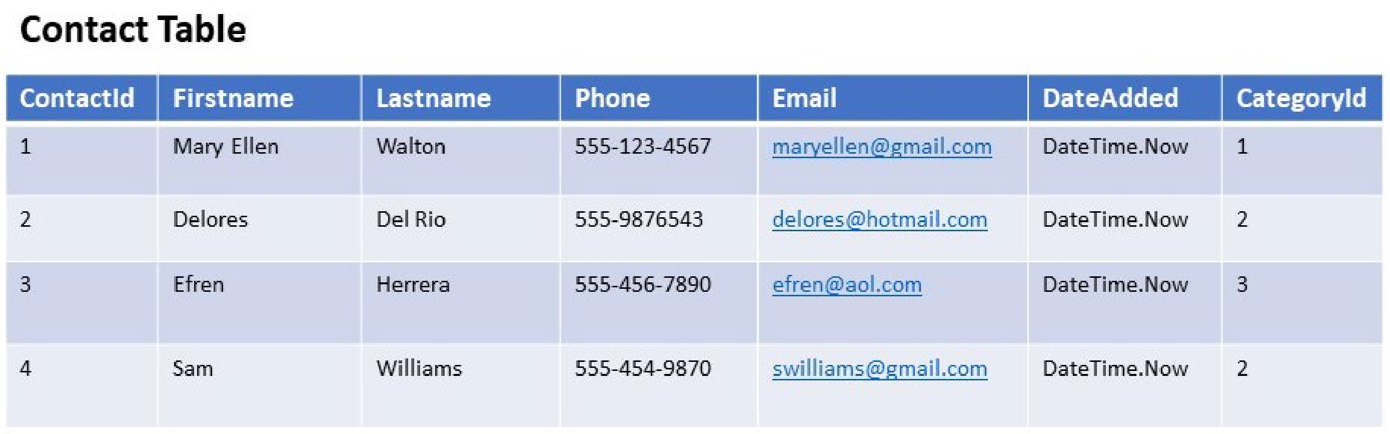
Code the Models and DB context

7. Code the Contact and Category classes for the data model and the DB context.

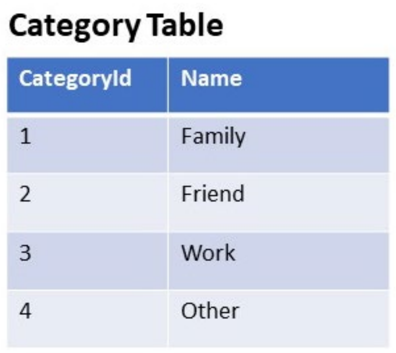


8. Seed initial data with the information provided below. Because Organization is optional, you do not need to include data for the Organization when you add the records below in the Contact table.
9. Populate the DateAdded column with the current date using DateTime.Now as follows when you are populating the seed data in the HasData() method:

    DateAdded = DateTime.Now


Contact Table

ContactId	Firstname	Lastname	Phone	Email	DateAdded	CategoryId
1	Mary Ellen	Walton	555-123-4567	maryellen@gmail.com	DateTime.Now	1
2	Delores	Del Rio	555-9876543	delores@hotmail.com	DateTime.Now	2
3	Efren	Herrera	555-456-7890	efren@aol.com	DateTime.Now	3
4	Sam	Williams	555-454-9870	swilliams@gmail.com	DateTime.Now	2


Category Table

CategoryId	Name
1	Family
2	Friend
3	Work
4	Other

Add the Connection String and Modify the Startup.cs file for Dependency Injection

10. Add a connection string to the appsettings.json file. Choose a descriptive name for your database such as Contacts, or ContactsExercise, etc.
11. Configure the Startup.cs file for Dependency Injection.

Create the Database

12. Use the Package Manager Console to add a new migration.
13. Update the database at the Package Manager Console. This should create the database.
14. View your dataset. To ensure the database was created correctly, display the SQL Server Object Explorer and expand the nodes until you can see the name of the database you just created.
15. View the seed database by expanding the Tables node. Then, right-click on each table (Category and Contact) and select View Data.

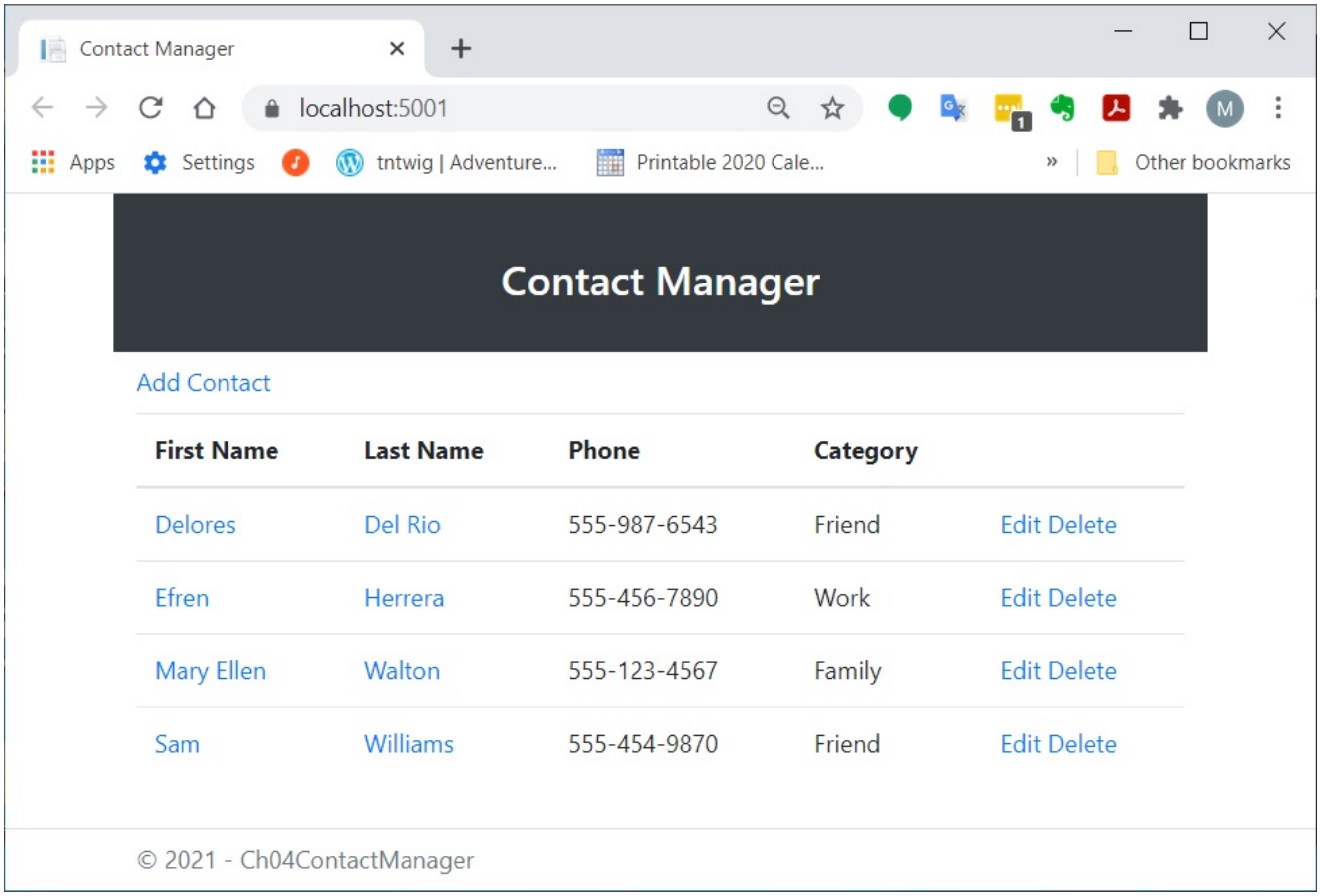
Modify the Home Controller

16. In the Home Controller include the context property and constructor so that other methods in this class can easily access the context object to work with the data in your database tables.
17. Code the Index() method to query a list of all the contacts. Include the Category information for each contact from the Category table. The Index() method should return a list of all the contacts.

Modify the Index View and Use Linq in the Home Controller to display a List of Friends

18. Modify the Home/Index view so that it looks like the web page shown below. We will create the code for Adding, Editing, and Deleting a contact in the next assignment. For this assignment, go ahead and include a link for Add Contact, Edit, and Delete. At this time, the links will not transfer to another page (use href="/"), they are placeholders for the functionality we will include in the next assignment (for example, the Add a New Movie link would be coded as: **<a href="/">Add a New Movie</a>**).

The Home page listing all contacts in the database



Contact Manager

Add Contact

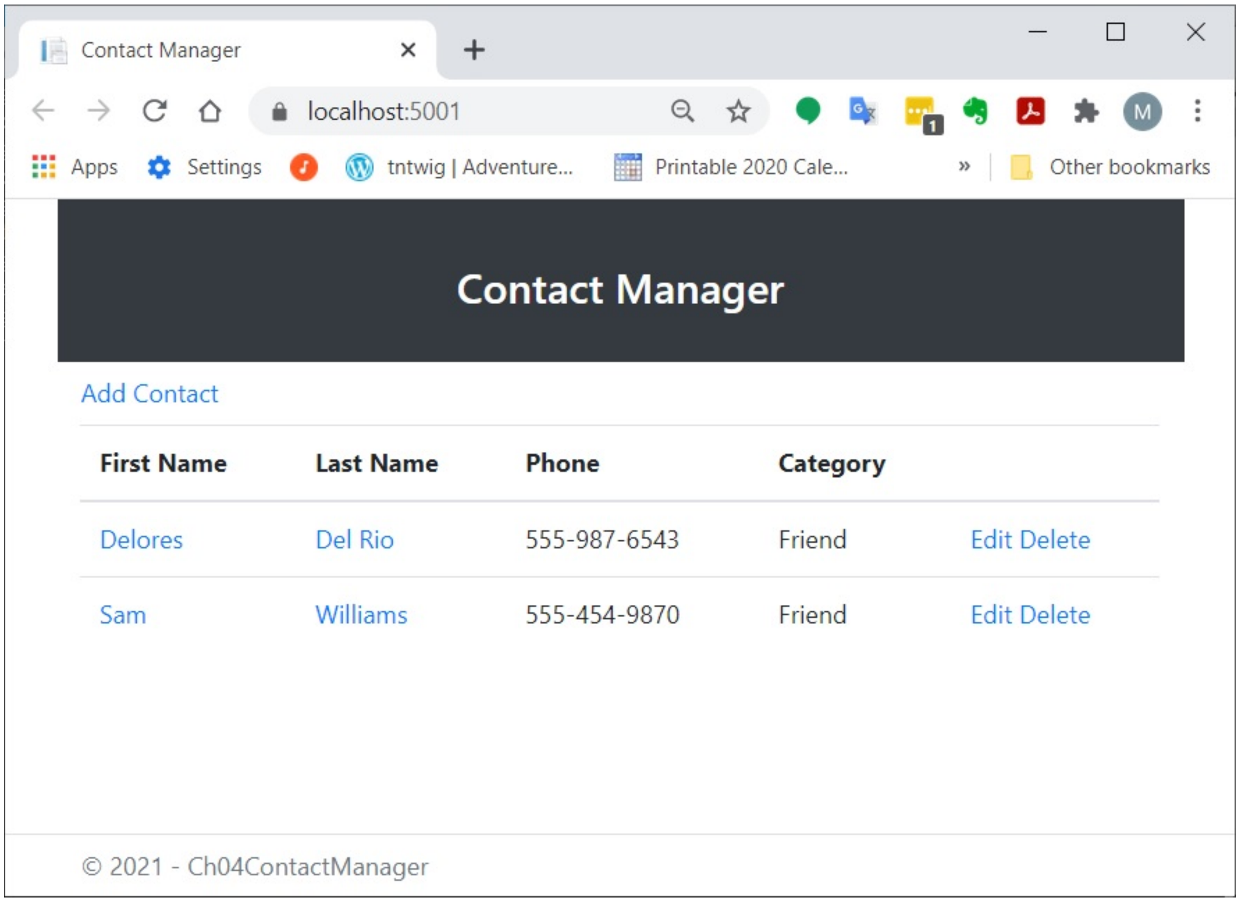
First Name	Last Name	Phone	Category	
Delores	Del Rio	555-987-6543	Friend	Edit Delete
Efren	Herrera	555-456-7890	Work	Edit Delete
Mary Ellen	Walton	555-123-4567	Family	Edit Delete
Sam	Williams	555-454-9870	Friend	Edit Delete

© 2021 - Ch04ContactManager

19. Test the application thoroughly. When the app starts, it should display a list of contacts, a link to add a contact, a link to Edit, and a link to Delete as shown in the above screen print.

20. Next, use Linq to write a query in the Index action of the Home controller that will list all your friends (note: refer to the Category table to find the CategoryId for the Friend category).

The Home page listing contacts whose category is Friend



Contact Manager

Add Contact

First Name	Last Name	Phone	Category	
Delores	Del Rio	555-987-6543	Friend	Edit Delete
Sam	Williams	555-454-9870	Friend	Edit Delete

© 2021 - Ch04ContactManager

21. Feel free to be creative and experiment with some of the bootstrap classes, CSS, and Font Awesome if you would like to make this home page look more aesthetic. When you are satisfied with your new look for the Contact Manager application and everything is working correctly, close the app and turn in your work.

Turn in the Contact Manager app

1. Once you are satisfied with your application, close Visual Studio, compress the Contact Manager app at the root level, and rename the compressed folder using the following naming convention:

   FirstInitial_LastName_Ch04ContactManager

   For example:

   M_Korzaan_Ch04ContactManager

2. Upload the renamed, compressed folder to the appropriate Dropbox in D2L.