

MVCDatabase:

Tables:

First create one database and name it NerdDinner.

Now run following code to generate tables or create by your own if you know how to create and give relationship between two tables.

1) Dinners (First Table)

```
CREATE TABLE [dbo].[Dinners](
[DinnerID] [int] IDENTITY(1,1) NOT NULL,
[Title] [nvarchar](50) NOT NULL,
[EventDate] [datetime] NOT NULL,
[Description] [nvarchar](256) NOT NULL,
[HostedBy] [nvarchar](20) NOT NULL,
[ContactPhone] [nvarchar](20) NOT NULL,
[Address] [nvarchar](50) NOT NULL,
[Country] [nvarchar](30) NOT NULL,
[Latitude] [float] NOT NULL,
[Longitude] [float] NOT NULL,
CONSTRAINT [PK_Dinners] PRIMARY KEY CLUSTERED
(
    [DinnerID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Press F5 after paste above code in sql server for run code.

Create MVC Application and do CRUD: (CREATE, READ, UPDATE, DELETE)

Create New MVC Application:

File -> New Project

Expand Visual C# node -> select web node

Choose the ASP.NET MVC 3 Web Application

Now give name of the application.

Now choose internet application from three options.

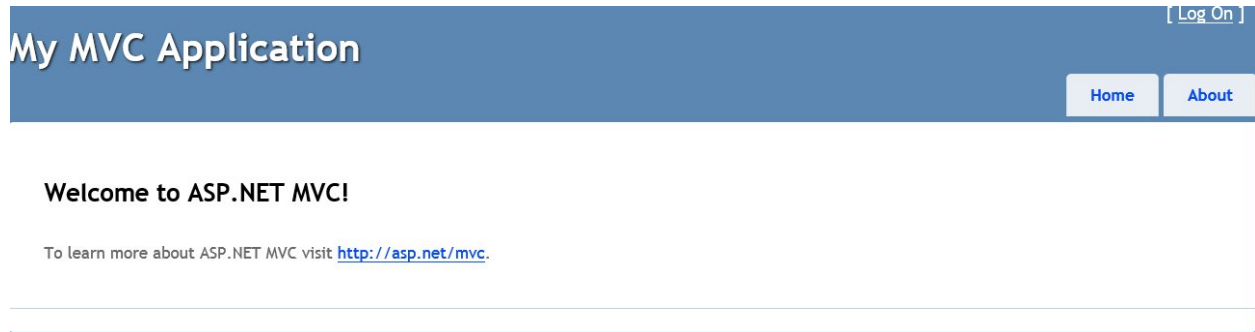
Click on unit test project check box if you want to test your application using unit test.

After clicking ok it will create your mvc application.

Just see in your solution explorer there are following basic files that has been automatically created:

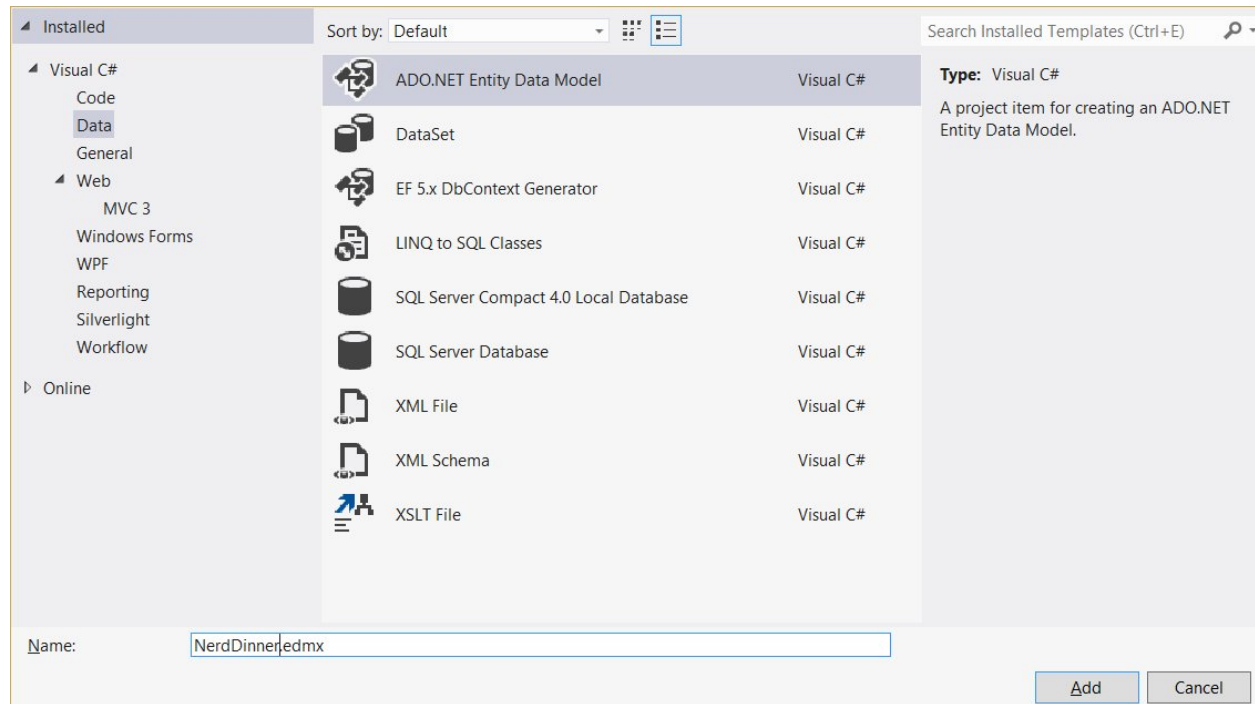
DIRECTORY	PURPOSE
/Controllers	Where you put Controller classes that handle URL requests
/Models	Where you put classes that represent and manipulate data and business objects
/Views	Where you put UI template files that are responsible for rendering output, such as HTML
/Scripts	Where you put JavaScript library files and scripts (.js)
/Content	Where you put CSS and image files, and other non-dynamic/non-JavaScript content
/App_Data	Where you store data files you want to read/write

Now you can run your application by F5 or Ctrl+F5 key and your basic auto configuration file set application will run.



Create Model classes for Database purpose:

Right Click on Models folder -> choose Add -> Select New Item just go to data category and Select the ADO.NET Entity Data Model.



Click generate from database because we have already created our database.

Next-> Click New Connection -> Enter your login Engine Name of your SQL Server into server name and choose your database from connect to database panel. Do test connection and OK.

Now you see in Entity Data Model Wizard above footer there is an entity name NerdDinnerEntities which we will use it to interact with the database in our code.

Next-> Choose you tables from the list and Finish.

Here is your tables showing in the NerdDinner.edmx save it and close it.

Now you see in your Models folder in solution explorer there is a NerdDinner.edmx in that there is a file called NerdDinner.tt in that there is a file Dinner.cs and RSVP.cs created automatically with the get set property of the field names of your tables.

Now we will create a new DinnerRepository class for the insert update delete and read queries to manage the code.

Right Click on Models folder -> choose Add -> Select New Item-> Select Code left side of your popup dialog box and select class and name it to DinnerRepository.

Write following code into DinnerRepository class file.

```
private NerdDinnerEntities entities = new NerdDinnerEntities();

public IQueryable<Dinner> FindAllDinners()
{
    return entities.Dinners;
}

public IQueryable<Dinner> FindUpcomingDinners()
{
    return from dinner in entities.Dinners
           where dinner.EventDate > DateTime.Now
           orderby dinner.EventDate
           select dinner;
}

public Dinner GetDinner(int id)
{
    return entities.Dinners.FirstOrDefault(d => d.DinnerID == id);
}

public void Add(Dinner dinner)
{
    entities.Dinners.Add(dinner);
}

public void Delete(Dinner dinner)
{
    entities.Dinners.Remove(dinner);
}

public void Save()
```

MVC CRUD

```
    entities.SaveChanges();  
}
```

Create Controller:

Now we create controller for handle URL request.

Right Click on Controller in Solution Explorer -> Add -> Controller

Name it as DinnersController and select template Empty Controller and Add.

Now we create action methods which is basically use for call the methods after controller call in URL.

We create two methods by which we can communicate with the database and get values as a list and as per the id vise.

Write following methods in DinnersController file:

```

DinnerRepository repository = new DinnerRepository();
public ActionResult Index()
{
    var dinners = repository.FindUpcomingDinners().ToList();
    return View("Index", dinners);
}

public ActionResult Details(int id)
{
    Dinner dinner = repository.GetDinner(id);

    if (dinner == null)
        return View("NotFound");
    else
        return View("Details", dinner);
}

```

Right click on NotFound and click on Add View it will create one view of NotFound and write following text in that between <h2></h2> tag.

Sorry - but the dinner you requested doesn't exist or was deleted.

Now we are going to display our details of Dinners table.

Right click on Details of View in Details Action Method in Dinners Controller and select add View.

Now->Click Create a strongly-typed view

Choose your model file from view data class dropdownlist.

Now choose view content as per requirement now we choose details for display our details as detail vise.

Click Add.

Run application write `http://localhost:portno/Dinners/Details/1` (1 or any id you have in your database Dinners table)

Create Index view template:

Create index now which is basically use for display whole table or all data. Right Click on Index which is in view function under Index ActionMethod. Click on Create a strongly typed View. Now choose your model file (for example: for us it is Dinner (NerdDinner.Models)) like wise.

Now click on Add button and system will automatically create the view having table and fill data which are in dinner table having name Index in Views/Dinners/Index.cshtml in solution explorer. Now Run your application after debug.

It will look like below picture:



NOTE: If your application not running properly ,you have no data in table or your date for event is not in a week go with the processes below and after create function add data and run this again.

Now you see in Index view in the last line:

```
@Html.ActionLink("Edit", "Edit", new { id=item.DinnerID })
@Html.ActionLink("Details", "Details", new { id=item.DinnerID }) |
@Html.ActionLink("Delete", "Delete", new { id=item.DinnerID })
```

is basically give us a link of particular data so we can get more detail by clicking on that data like in gridview. Detail or view or edit it or delete it.

Now We create ActionMethods and Views for CRUD:**Let's do Create Update Delete:****Create:**

Now we do Create action method for create for add new data in database.

Write following code of actionmethod in dinnerscontroller after Details actionmethod:

```
public ActionResult Create()
{
    Dinner dinner = new Dinner()
    {
        EventDate = DateTime.Now.AddDays(7)
    };
    return View(dinner);
}
```

Now we create view for Create ActionMethod:

So Right click on Create function method name and click on add view and click on create strongly type and select your dinner model file from model class and choose Create in scaffold template and click ok.

Now run your code in browser and write your url for example:

<http://localhost:49161/Dinners/Create>

And fill the form with new details and click on create button to save data but you are not able to save data in database now so for that you have to create same method for saving the data in database.

Now create one new method like below to take your all data from form and save into database.

```
[HttpPost]
public ActionResult Create(FormCollection formValues)
{
    Dinner dinner = new Dinner();
    if(TryUpdateModel(dinner))
    {
        repository.Add(dinner);
        repository.Save();
        return RedirectToAction("Details", new {id=dinner.DinnerID});
    }
    return View(dinner);
}
```


MVC CRUD

Now if you remember you have created repository file at starting of this tutorial that contains different different methods for add data, delete data and details etc etc...

So you are using now those methods in above actionmethod for CRUD in MVC. Like: (repository.Add(dinner))

There is one line where you see [HttpPost] which is basically use for sending the data as a post method for saving in database.

Now you run your application and just write /Dinners/Create after <http://localhost:49161> url and fill the form and click on Create button:

<http://localhost:49161/Dinners/Create>

You see your data filled successfully and your filled information display in a page with url Details.

The screenshot shows a web browser window with the title "My MVC Application". In the top right corner, there are two buttons: "Home" and "About". The main content area is titled "Create" and contains a form for creating a "Dinner" record. The form fields are as follows:

- Title:
- EventDate:
- Description:
- HostedBy:
- ContactPhone:
- Address:
- Country:
- Latitude:
- Longitude:

At the bottom left of the form is a "Create" button. In the bottom right corner of the page, there is a watermark that says "Activate Windows" and "Go to PC settings to activate Wind".

Update:

Now we create edit action method for get details from database and fill it in a textboxes so we can do update process.

Write following code of actionmethod in dinnerscontroller after create actionmethod which is [httppost]:

```
public ActionResult Edit(int id)
{
    Dinner dinner = repository.GetDinner(id);
    return View(dinner);
}
```

Now we create view for edit:

So Right click on Edit function method name and click on add view and click on create strongly type and select your dinner model file from model class and choose edit in scaffold template and click ok.

Now run your code in browser and write your url for example:

<http://localhost:49161/Dinners/Edit/1>

(1 is my database id you have to write your id which is available in your database table)

Still you just get the details as per your id passed but you cannot update so for update we create the same method for edit as we done in create process.

So write down following code in your application after Edit ActionMethod:

```
[HttpPost]
public ActionResult Edit(int id, FormCollection formValues)
{
    Dinner dinner = repository.GetDinner(id);
    UpdateModel(dinner);
    if (TryUpdateModel(dinner))
    {
        repository.Save();
        return RedirectToAction("Details", new { id = dinner.DinnerID });
    }
    return View(dinner);
}
```

Here you done the code for update information that you want to update.

MVC CRUD

There is a dinner object of a table and repository object by which you use it for use methods that you created in repository file.

repository. Save(); is used for saving details that you updated and after update done you have to display those data that you update so there is a RedirectToAction which is basically use like response.redirect it will display data that you update.

```
("Details", new { id = dinner.DinnerID });
```

that contains the view name and id that you want to display the detail.

Now run your application like below:

<http://localhost:49161/Dinners/Edit/1>

(1 is my database id you have to write your id which is available in your database)

and you see that your passed id's all information filled in textboxes so just change values those you want to change and click on Save button.

The screenshot shows a web application titled "My MVC Application" with a blue header. In the top right corner, there are two buttons: "Home" and "About". The main content area is titled "Edit" and contains a form for editing a "Dinner" record. The form fields are as follows:

Dinner	
Title	<input type="text" value=".Net Futures"/>
EventDate	<input type="text" value="07-Nov-14 9:52:47 PM"/>
Description	<input type="text" value="Lets talk about .Net"/>
HostedBy	<input type="text" value="Mahavirsinh"/>
ContactPhone	<input type="text" value="9879879879"/>
Address	<input type="text" value="Ahmedabad"/>
Country	<input type="text" value="India"/>
Latitude	<input type="text" value="1254.5454"/>
Longitude	<input type="text" value="-122.45"/>
<input type="button" value="Save"/>	

In the bottom right corner of the page, there is a watermark that says "Activate Windows Go to PC settings to activate Wind".

Delete:

For Delete you have to first get those data that you want to delete and than you can delete particular data:

So for that write following actionmethod first:

```
public ActionResult Delete(int id)
{
    Dinner dinner = repository.GetDinner(id);
    if (dinner == null)
        return View("NotFound");
    else
        return View(dinner);
}
```

Now we create view for Delete:

So Right click on Delete function method name and click on add view and click on create strongly type and select your dinner model file from model class and choose Delete in scaffold template and click ok.

Now run your code in browser and write your url for example:

<http://localhost:49161/Dinners/Delete/1>

Here you just displaying the data which user want to delete and ask that person that are you sure want to delete so after clicking delete here your data must be delete so for that we have to create Delete method again as a HttpPost.

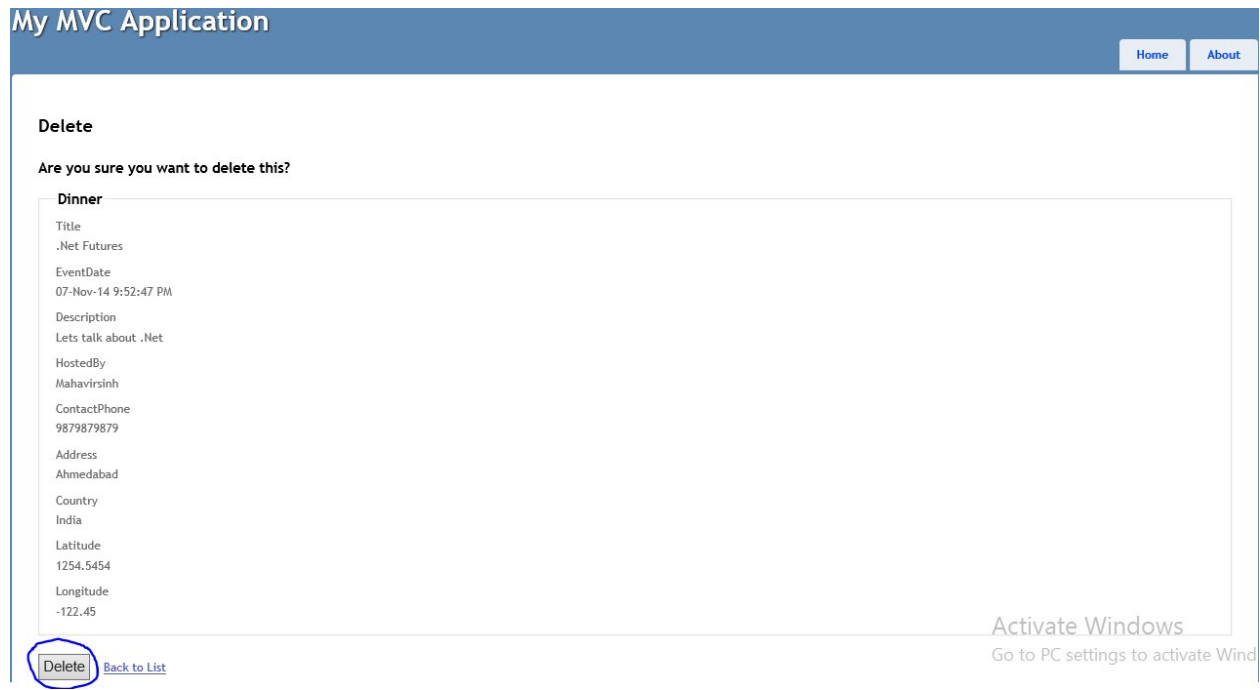
So write following code for delete after confirmation:

```
[HttpPost]
public ActionResult Delete(int id, string confirmButton)
{
    Dinner dinner = repository.GetDinner(id);
    if (dinner == null)
        return View("NotFound");
    repository.Delete(dinner);
    repository.Save();
    return View("Deleted");
}
```

So if you run your application like <http://localhost:49161/Dinners/Delete/1> and it will display data that you want to delete with confirmation that sure you want to delete? And now when you click on delete button it will delete your data from

MVC CRUD

database and if you choose Back to List than it will redirect you to the index page of your application.



So that's it for CRUD in MVC for more details send your feedback or query on:

dotnetexperteam@gmail.com

Or go on site for more study:

dotnetexperteam.wordpress.com