Models, Views, Controllers

Here inside the Visual Studio you'll have a window with the title Solution Explorer. It is the Solution Explorer window that allows you to get to all of your files, C sharp files, JavaScript files, CSS files, icons, images and everything that goes into your application. When we created this application using the internet project template, Visual Studio went ahead and populated our application with all of the folders and files that you see here. That's why we already have a running application with contact in about links working. Three of these folders have a special significance, the controllers, the models and the views folders. Let's talk about those names for a minute. The ASP.NET MVC framework derives its name from the model view controller design pattern. This design pattern has been present in software applications for several decades at this point and it's a design pattern to follow when you want to separate the responsibilities of the components in your user interface layer. The C and MVC is for controller. A controller is a software component that will be the target for some external stimulus. In the case of a web application that external stimulus is usually an incoming HTTP request. So, when someone launches a web browser and points it to the slash home slash about location of my application. That incoming request needs to go to a controller that is in my application. When the controller receives the request, it's responsible for building a model that M in MVC. It's the model that contains all the information that you need to present to the user to satisfy that incoming request. In the case, of slash home slash about, the model might just be some information about the website or about the company or the people behind the website