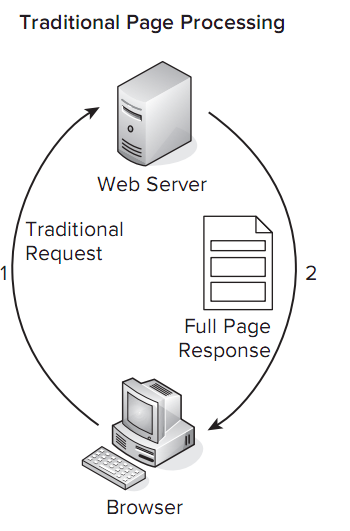
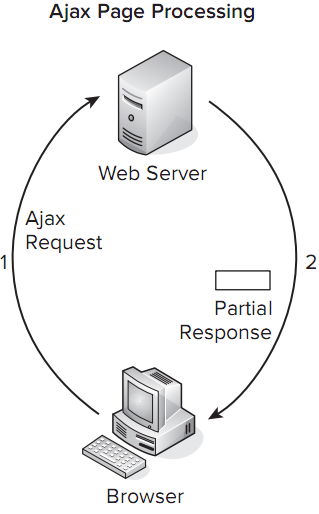
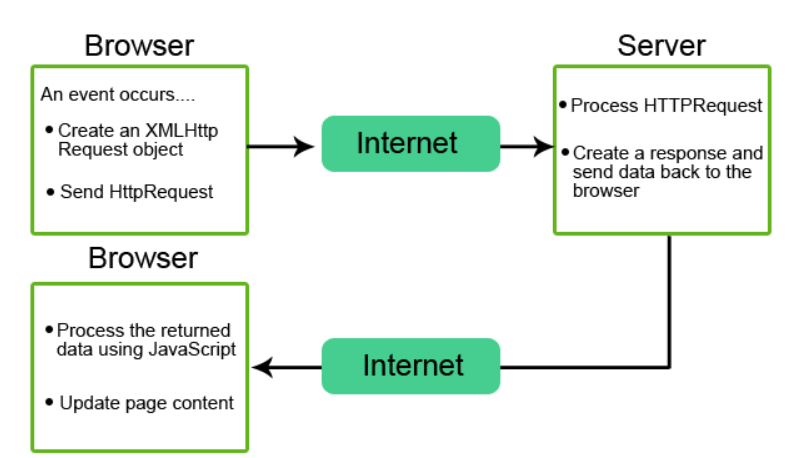
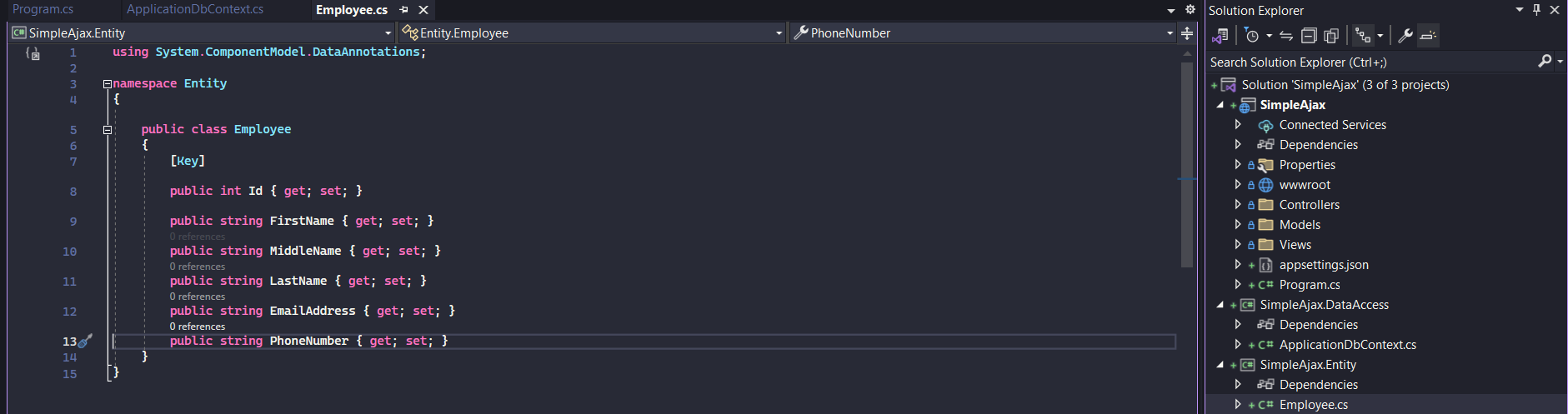
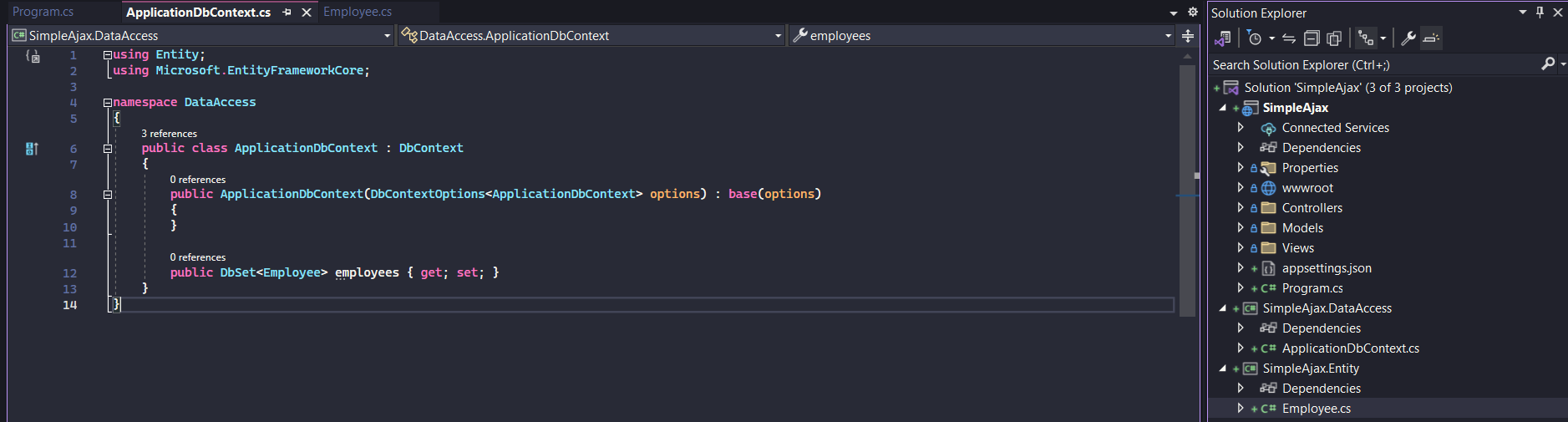
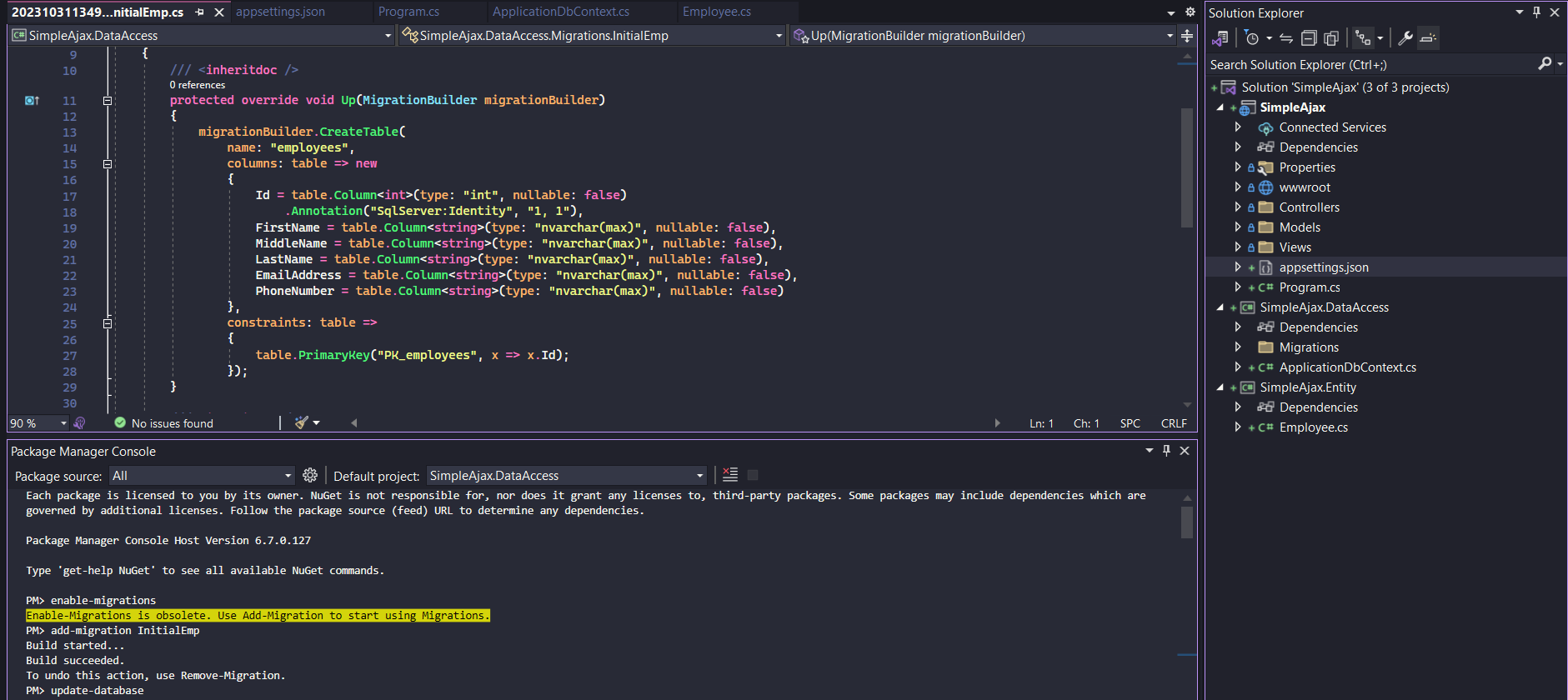
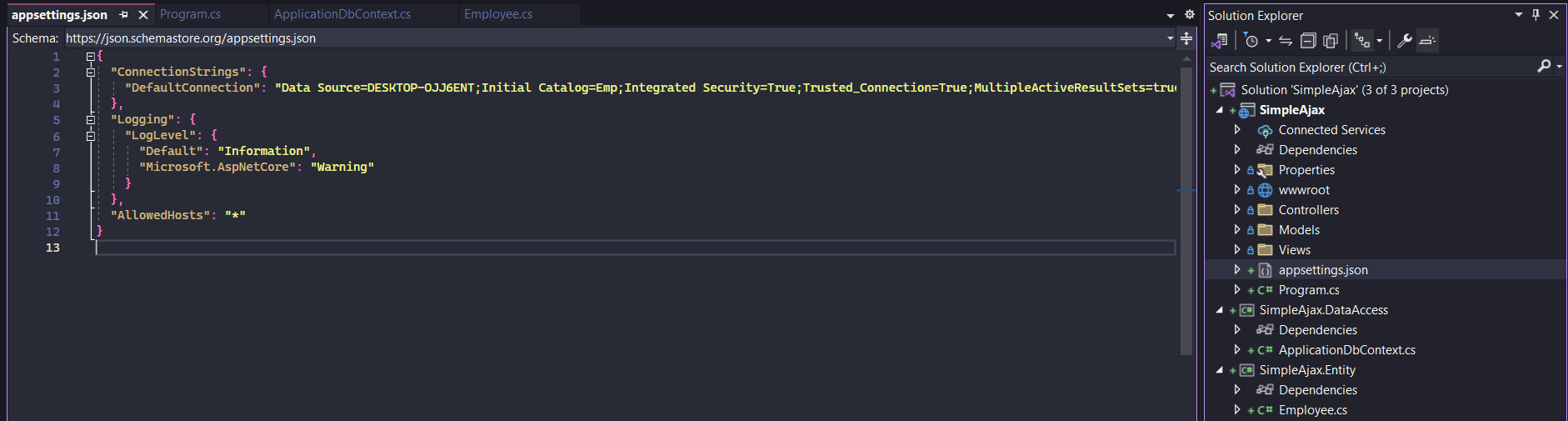
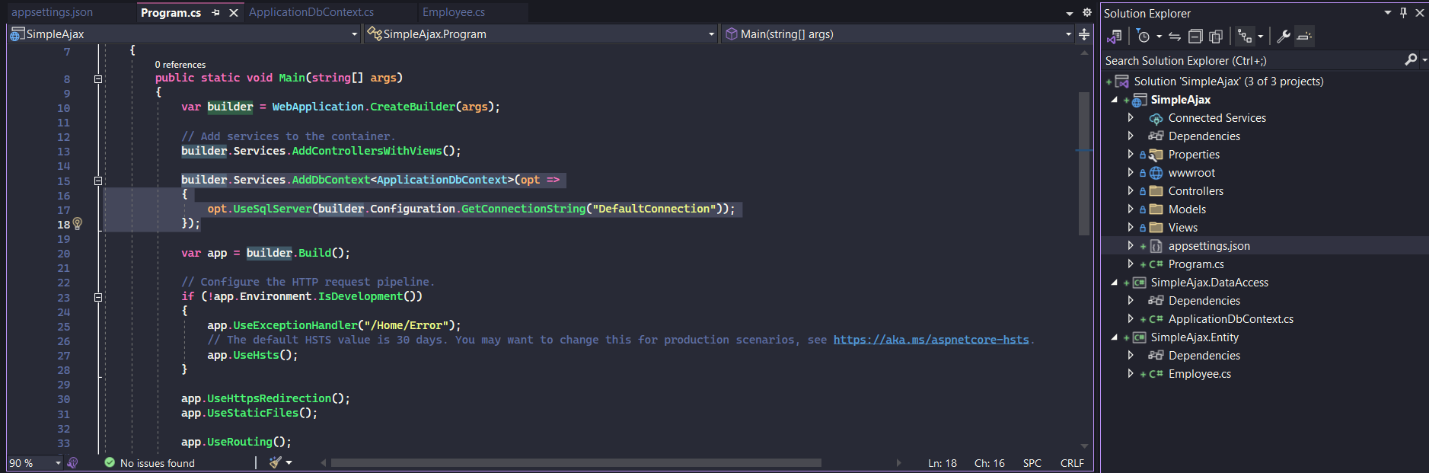
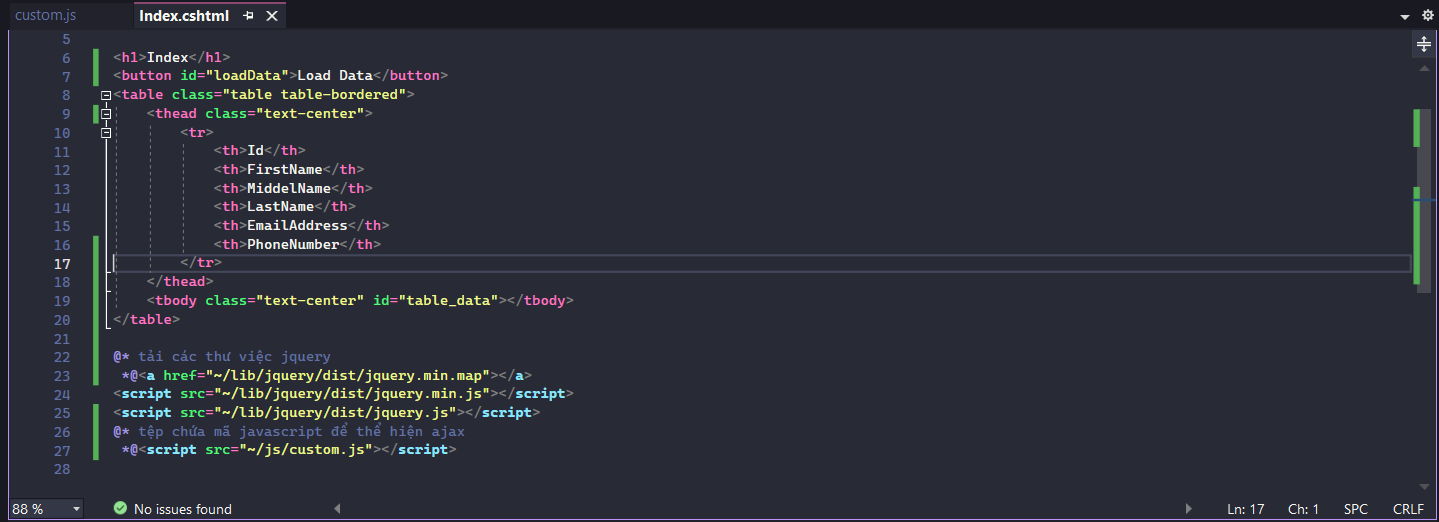
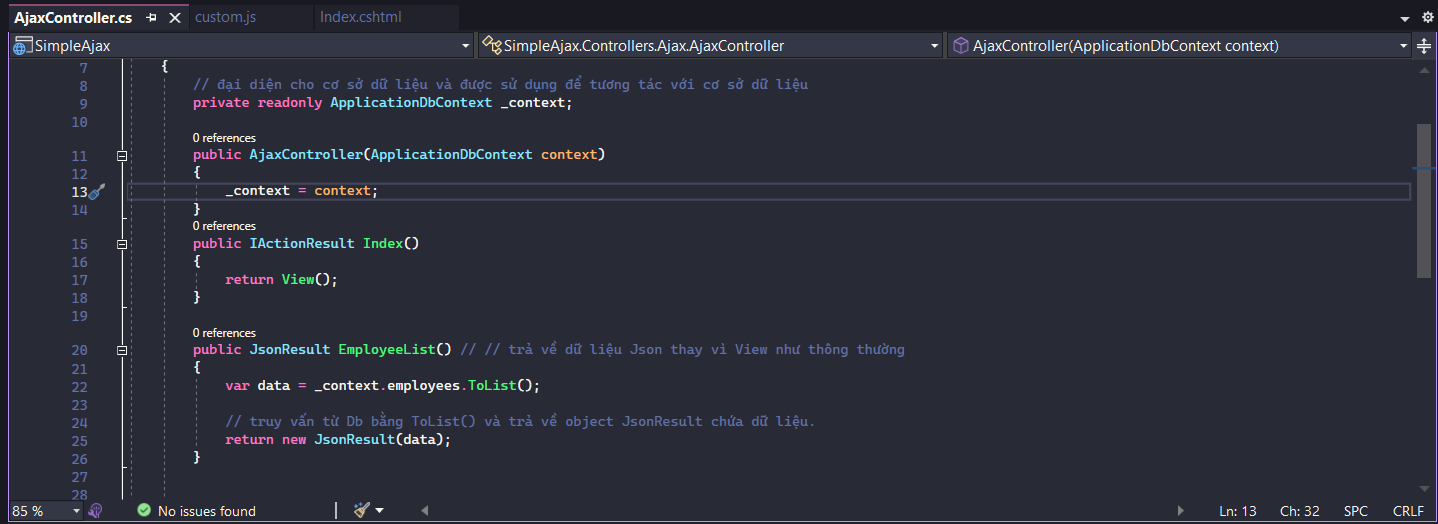
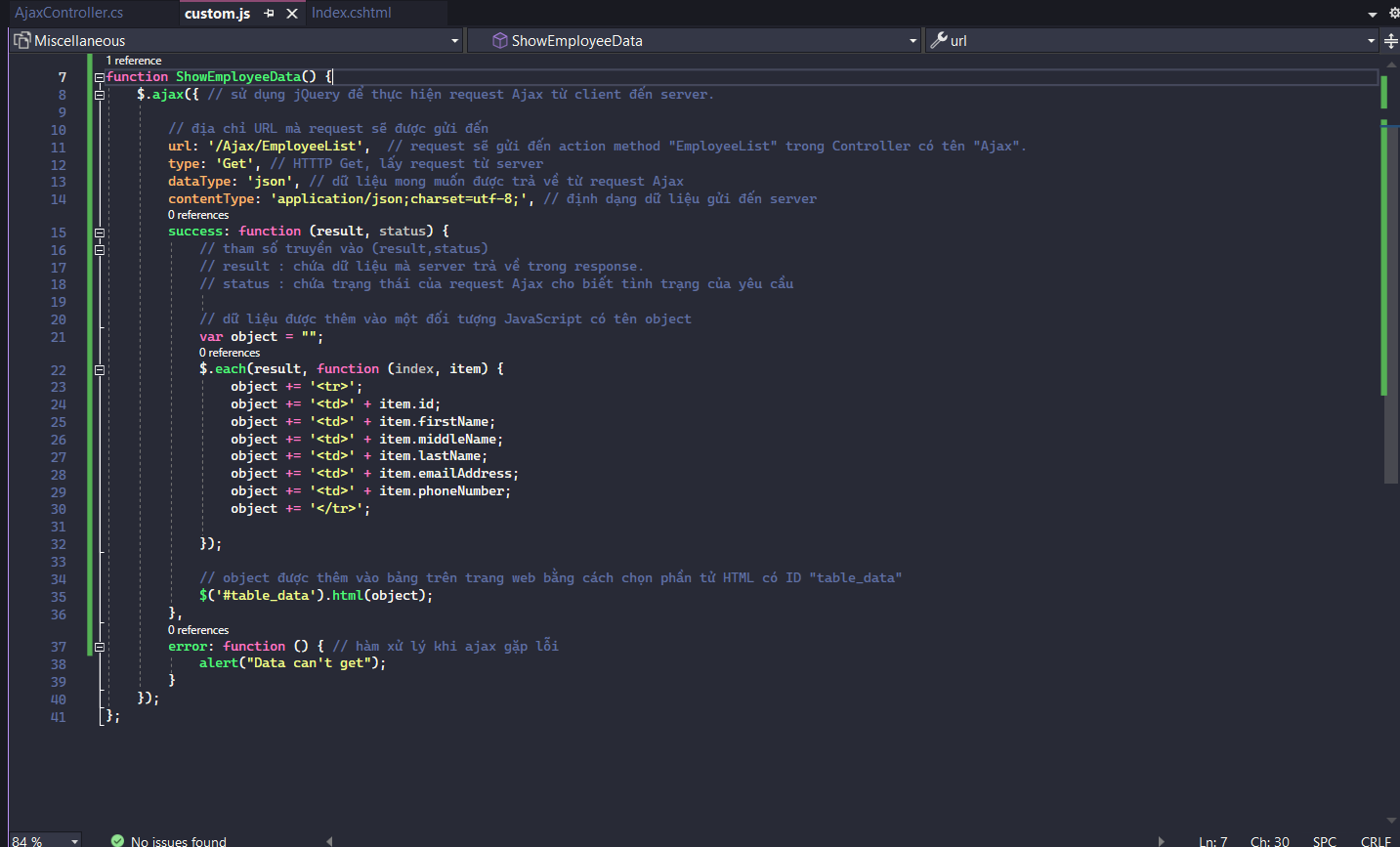
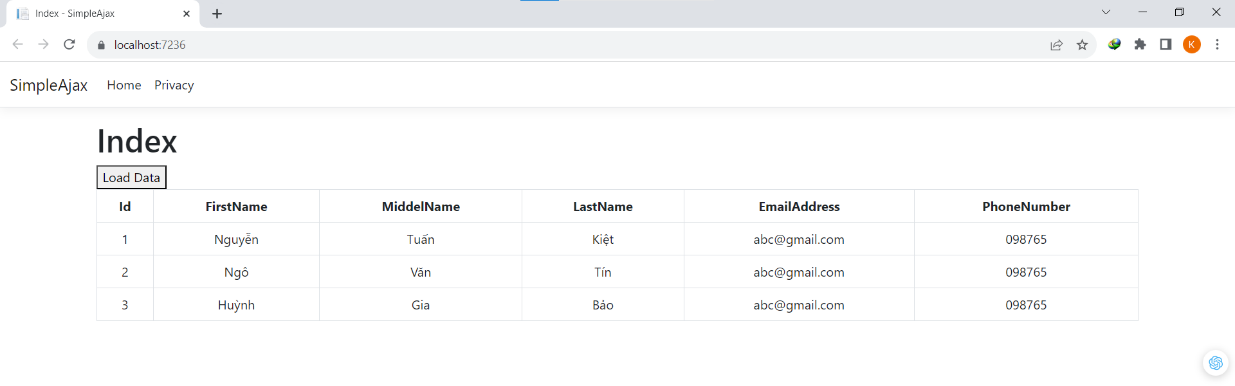
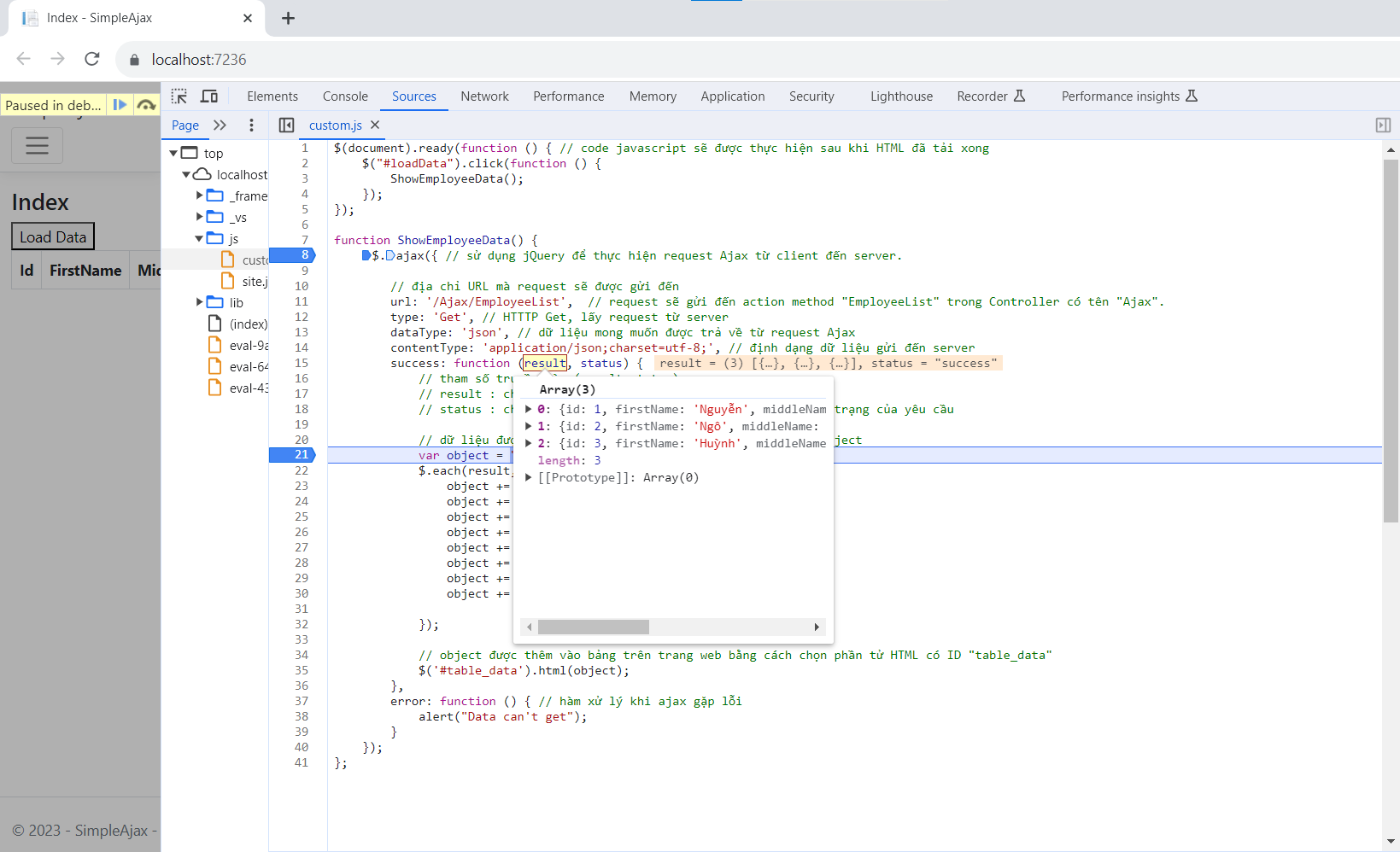
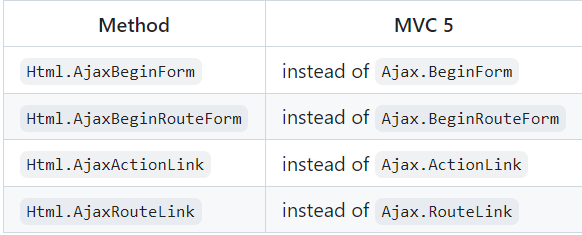
* Công nghệ AJAX ra đời từ năm 2005, viết tắt của Asynchronous JavaScript and XML, cho phép các trang web ở phía client trao đổi dữ liệu với server mà không làm gián đoạn giao diện người dùng.
* Các trình duyệt web tương tác với server bằng cách tạo ra các yêu cầu cho trang thông qua hai phương thức là **POST** và **GET**. Server xử lí yêu cầu và gửi lại kết quả (HTML, script, css) đến trình duyệt
* Nhìn vào mô hình trên chúng ta thấy có hai trở ngại chính là:
* Do toàn bộ trang được gửi lại từ server nên tạo ra một gánh nặng không cần thiết. Ví dụ trong Contact Form của chúng ta, mỗi lần người dùng nhập thông tin và nhấn Send thì toàn bộ trang sẽ được gửi đến browser, trong khi chỉ có một vài thông tin người dùng nhập là thay đổi còn phần lớn các thành phần như menu, banner, table,…không đổi.
* Cũng vì toàn bộ trang được gửi trở lại từ server nên những cái đã có trong quá trình tương tác trước đó sẽ được thay bằng những cái mới, quá trình thay thế này sẽ làm cho trang cảm giác như “nhấp nháy (flicker effect) ”, một hiệu ứng gây khó chịu cho người dùng.
* Microsoft đã tích hợp công nghệ **AJAX** vào nền tảng .**NET** (**ASP.NET AJAX**) để giải quyết những trở ngại trên.
* Một số lợi ích của **ASP.NET** **AJAX**:
* Chỉ một phần trang (được yêu cầu) được gửi trở lại từ server thay vì toàn bộ trang.
* Cung cấp phản hồi cho người dùng trong quá trình trang cập nhật.
* Cập nhật một phần trang và gọi các đoạn mã server đã được lập lịch bằng bộ định thời gian (timer).
* Với **AJAX**, mô hình trao đổi giữa browsers và servers sẽ như sau
* Một số nhược điểm của ASP.NET AJAX :
* Kích thước của yêu cầu dữ liệu tăng lên phần lớn vì tất cả các yêu cầu dữ liệu này đều được mã hóa URL.
* Phụ thuộc nhiều vào JavaScript vì nó được tích hợp sẵn JavaScript. Do đó, nếu người dùng tắt JavaScript trong trình duyệt thì AJAX sẽ ngừng hoạt động.
* Vì tất cả các tệp được tải xuống phía client – side trong ứng dụng AJAX nên tính bảo mật rất khá yếu trong các ứng dụng này.
* Trong AJAX, thông tin máy chủ hoàn toàn không thể truy cập được.
* Trong AJAX, thông thường nhiều công nghệ được sử dụng cùng nhau để tạo ra các trang web động bao gồm : XSLT – XHTML – CSS – Document Object Model – JavaScript – XMLHttpRequest Object
* Cách hoạt động :
* Phía máy khách (Client-side): Sử dụng JavaScript để gửi yêu cầu Ajax từ trình duyệt đến phương thức phía máy chủ đã tạo. Trong JavaScript, có thể sử dụng thư viện như jQuery hoặc XMLHttpRequest để thực hiện việc này. Yêu cầu này sẽ bao gồm thông tin như URL, dữ liệu cần gửi (nếu cần), kiểu yêu cầu, và cách xử lý kết quả.
* Xử lý kết quả: Khi máy chủ xử lý yêu cầu Ajax, nó sẽ trả về kết quả, thường dưới dạng dữ liệu JSON. JavaScript trên trình duyệt sẽ xử lý kết quả này và có thể cập nhật nội dung của trang web mà không làm trang web tải lại hoàn toàn.

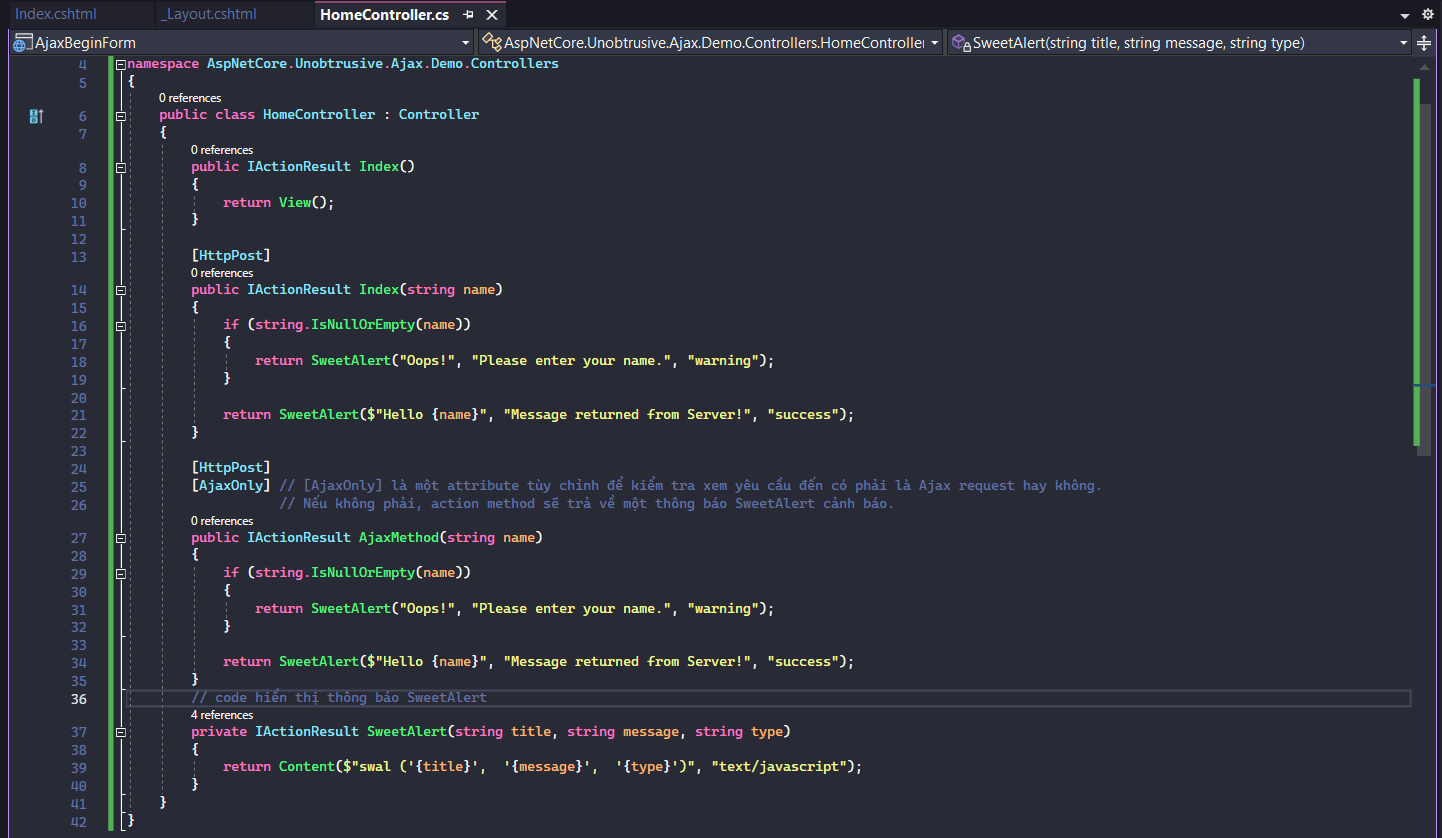
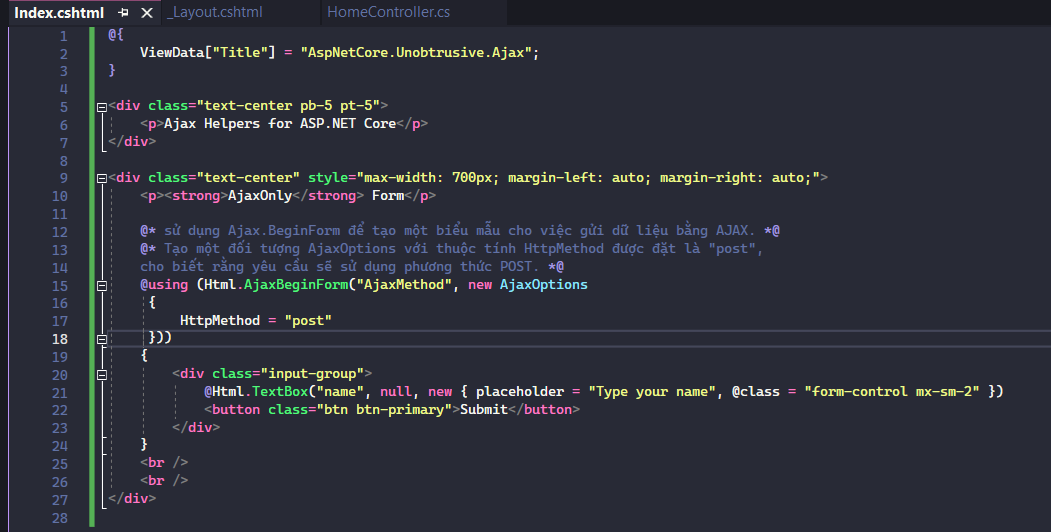
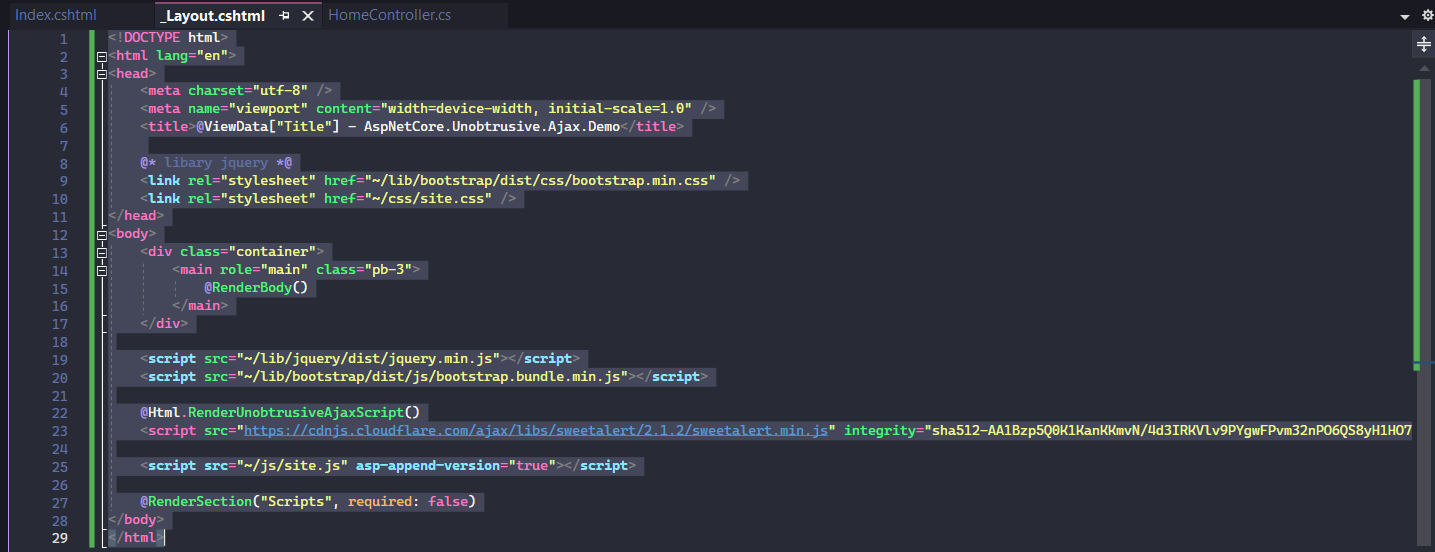
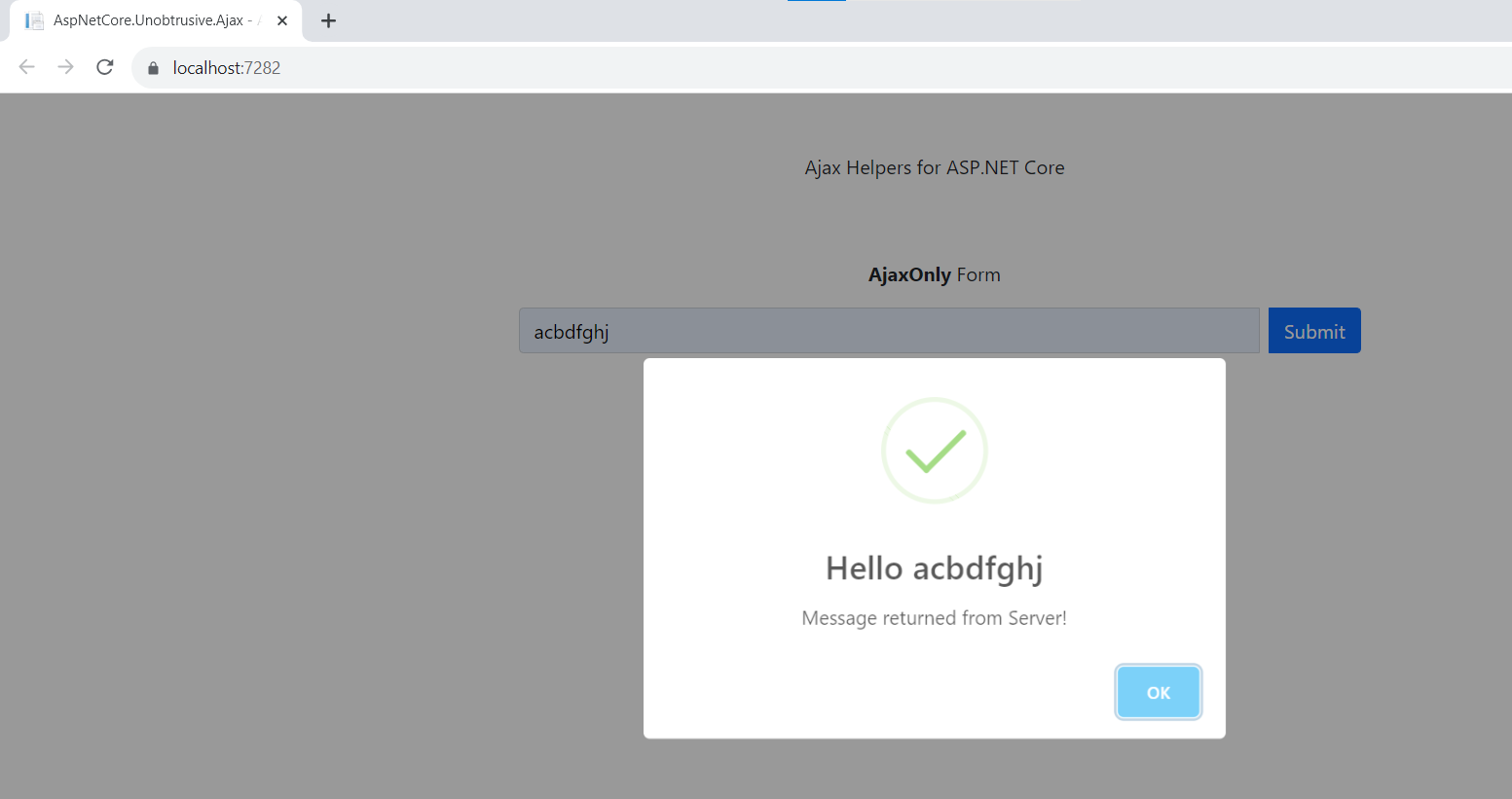
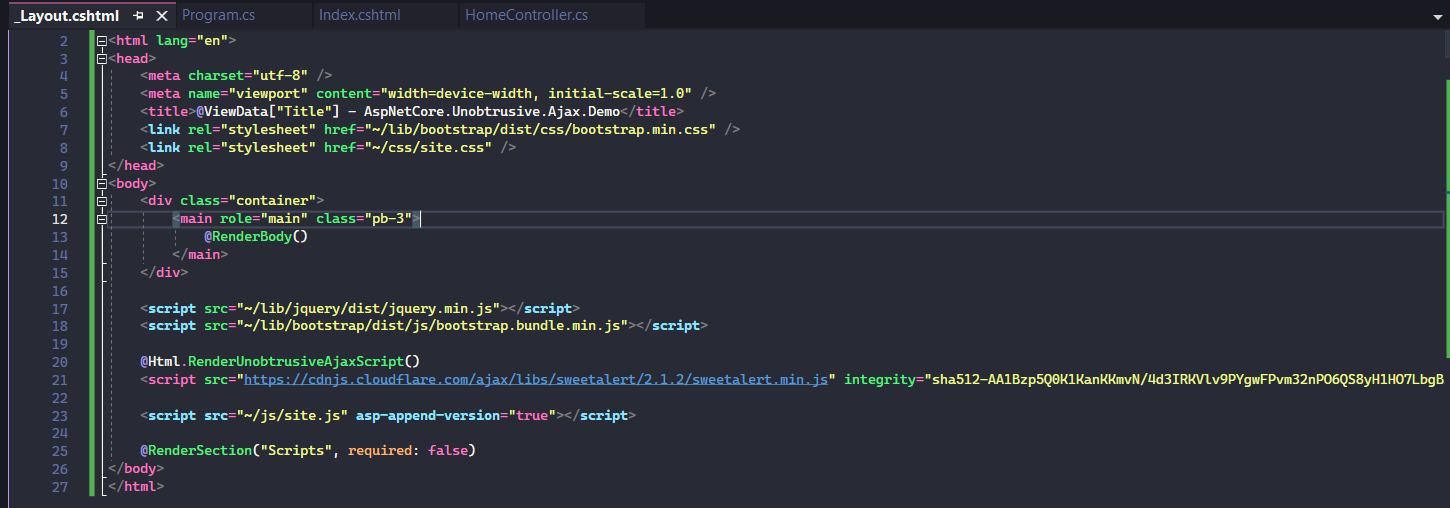
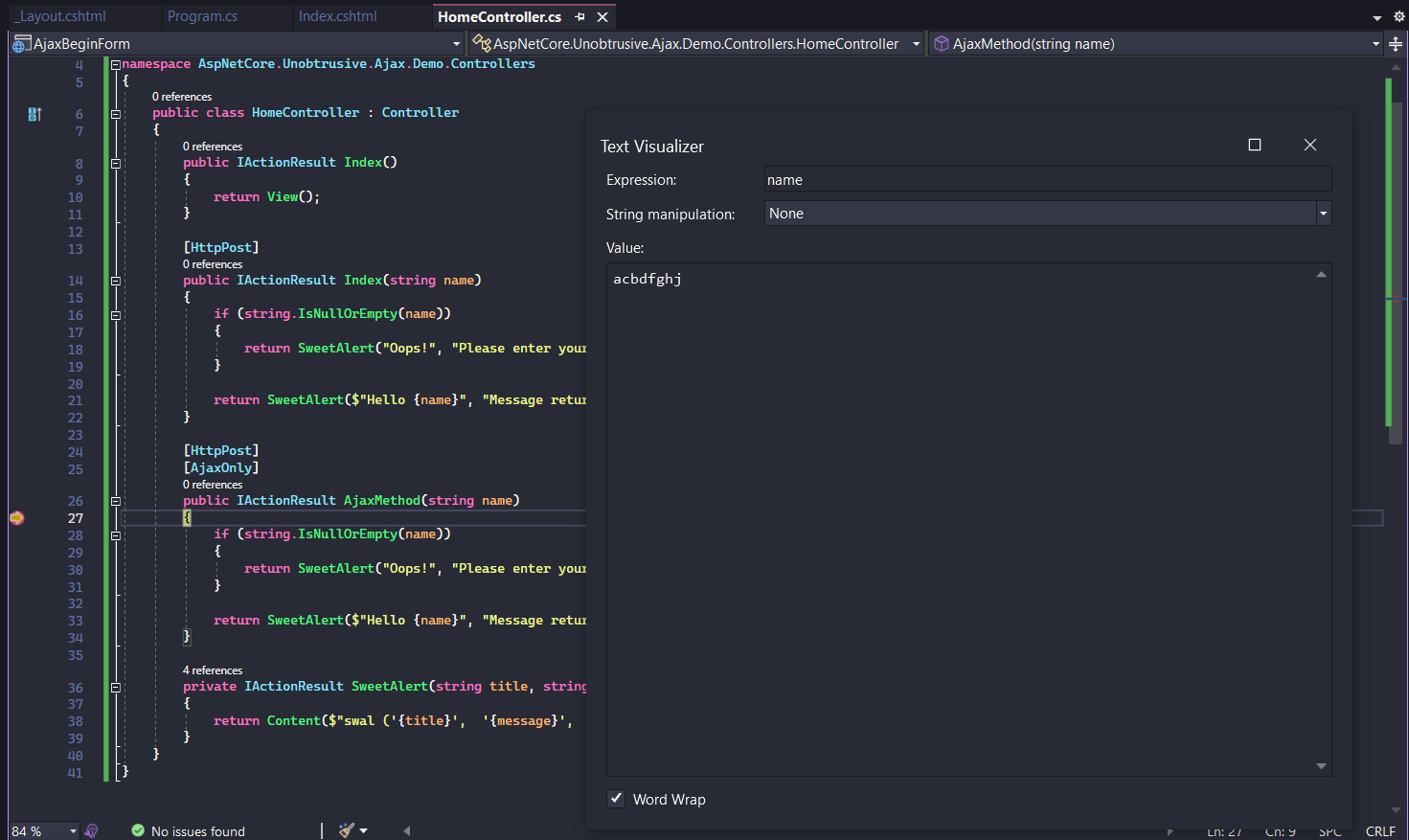


1. **Ajax implementation using**

* Triển khai Ajax trong ASP.NET bằng 2 cách phổ biến sau : UpdatePanel và sử dụng Jquery
* Sử dụng UpdatePanel :
* Cách hoạt động: UpdatePanel là một kiểu điều khiển (control) trong ASP.NET, được cung cấp bởi ASP.NET Ajax Extension. Có thể đặt các điều khiển muốn cập nhật bên trong UpdatePanel. Khi có một sự kiện tương tác xảy ra, chẳng hạn như sự kiện nhấp chuột, UpdatePanel sẽ thực hiện yêu cầu Ajax để cập nhật chỉ phần nội dung bên trong nó, mà không cần phải tải lại toàn bộ trang.
* Ưu điểm: Cách tiếp cận này đơn giản và không yêu cầu bạn viết JavaScript. Nó cung cấp một cách nhanh chóng triển khai Ajax cho các ứng dụng ASP.NET đã tồn tại.
* Hạn chế: UpdatePanel có thể gây thừa tải máy chủ và mạng, vì nó thực hiện một yêu cầu hoàn toàn đến máy chủ mỗi khi có sự kiện, thậm chí cho cả các tương tác nhỏ. Điều này có thể làm giảm hiệu suất ứng dụng khi có nhiều sự kiện Ajax trên trang.
* Sử dụng Jquery :
* Cách hoạt động: Sử dụng jQuery (hoặc các thư viện JavaScript khác), hoàn toàn kiểm soát việc gửi yêu cầu Ajax từ máy khách đến máy chủ và xử lý kết quả. Có thể xác định chính xác nội dung cần cập nhật, và có khả năng tối ưu hóa yêu cầu Ajax.
* Ưu điểm: Sử dụng jQuery cung cấp linh hoạt cao, giúp bạn tối ưu hóa trải nghiệm người dùng và hiệu suất ứng dụng. Bạn có thể điều khiển chính xác việc gửi và nhận dữ liệu Ajax và tùy chỉnh trang web một cách chi tiết.
* Hạn chế: Việc triển khai Ajax bằng cách sử dụng jQuery đòi hỏi kiến thức JavaScript và AJAX tốt hơn, và nó có thể đòi hỏi nhiều công sức hơn so với UpdatePanel.
* Demo Implementation of Ajax using Jquery : sử dụng Ajax để tải dữ liệu từ máy chủ và cập nhật giao diện người dùng trên client mà không cần phải tải lại toàn bộ trang

1. Tạo model tên “Employee” :
2. Cấu hình để kết nối đến csdl bằng entity framework
3. Tạo views để hiển thị
4.  Tạo controller “Ajax” để code chèn dữ liệu vào cơ sở dữ liệu, hiển thị dữ liệu vào views
5.  Thể hiện ajax
6. Kết quả
7. **Using Ajax.BeginForm Helper Method**

* Ajax.BeginForm là một phần mở rộng (extension method) trong ASP.NET MVC dùng để tạo biểu mẫu (form) và thực hiện gửi dữ liệu một cách bất đồng bộ bằng Ajax.
* Có 2 loại extension method BeginForm: “Html.BeginForm” và “Ajax.BeginForm”. “Html.BeginForm” tạo một biểu mẫu thông thường, trong khi “Ajax.BeginForm” tạo một biểu mẫu có khả năng gửi dữ liệu bằng Ajax.
* Ajax.BeginForm giúp tạo một biểu mẫu HTML giống như Html.BeginForm, nhưng nó được tối ưu hóa để thực hiện gửi dữ liệu bằng Ajax. Thay vì tải lại toàn bộ trang khi biểu mẫu được gửi, nó sử dụng Ajax để gửi yêu cầu và cập nhật nội dung trang mà không cần tải lại toàn bộ trang. Điều này giúp làm cho ứng dụng web trở nên nhanh hơn và mượt mà hơn.
* Ajax.BeginForm:
* Ưu điểm:
* Cung cấp trải nghiệm người dùng tốt hơn: Cho phép tải và cập nhật dữ liệu mà không cần tải lại toàn bộ trang, làm cho ứng dụng trở nên nhanh hơn và hiệu quả hơn.
* Nhược điểm:
* Đòi hỏi việc thêm thư viện Ajax và cài đặt nhiều tùy chọn cụ thể.
* Không phù hợp cho các tình huống mà bạn muốn tải lại toàn bộ trang hoặc thực hiện các thay đổi lớn trên trang.
* Html.BeginForm:
* Ưu điểm:
* Dễ dàng để tạo biểu mẫu HTML và thực hiện các thay đổi tức thì trên trang.
* Phù hợp cho các trường hợp mà bạn muốn tải lại toàn bộ trang hoặc không cần tích hợp Ajax.
* Nhược điểm:
* Không cung cấp trải nghiệm người dùng động và không giúp cải thiện hiệu suất tải trang.
* Phương thức Ajax.BeginForm không còn được hỗ trợ trong ASP.NET Core 6.0. Tuy nhiên, có nhiều cách khác để đạt được chức năng tương tự. Một cách là sử dụng gói jQuery Unobtrusive Ajax. Có thể cài đặt gói này bằng cách sử dụng PM> Install-Package AspNetCore.Unobtrusive.Ajax và thêm các tham chiếu đến nó trong views.
* Ví dụ : cách thức thay thế trên stackoverflow : <https://stackoverflow.com/questions/75549462/how-to-replace-ajax-beginform-helper-in-net-core-6-0-application-since-microsof>

1. Tạo controller
2.  Tạo views và \_Layout.cshtml (mẫu trang chung)
3. Kết quả :
4. **Jquery UI Controls**

* Jquery UI là viết tắt của Jquery GUI. Nó là một bộ sưu tập các hiệu ứng hoạt hình hình ảnh, vật dụng GUI, và chủ đề thực hiện với JQuery, CSS, HTML và JavaScript. Các plug-ins mới thêm rất nhiều chức năng mới trong các thư viện lõi jQuery.
* Vào thời điểm hiện tại, khi ASP.NET 7 (hoặc .NET 7) đã ra mắt, vẫn có thể sử dụng jQuery UI controls trong các ứng dụng ASP.NET, nhưng sự phụ thuộc vào jQuery UI đã giảm đáng kể và người ta đang dần chuyển sang các giải pháp và thư viện khác.
* Kích thước và hiệu suất: jQuery UI là một thư viện khá lớn, và việc sử dụng nó có thể làm tăng kích thước của trang web và làm giảm hiệu suất. Trong thời kỳ tập trung vào tối ưu hóa và hiệu suất trang web, nhiều người đã chuyển sang các giải pháp nhẹ hơn và tối ưu hóa hơn.
* Các thư viện JavaScript hiện đại: Có nhiều thư viện JavaScript hiện đại như React, Angular, Vue.js và một số khung làm việc khác đã trở nên phổ biến trong việc xây dựng giao diện người dùng và tạo các thành phần tương tác. Chúng cung cấp các giải pháp mạnh mẽ hơn và tương thích tốt hơn với các xu hướng hiện đại.
* Một số dự án lớn và frameworks đã dừng việc sử dụng jQuery và jQuery UI. Ví dụ, trong thế giới ASP.NET, .NET Core và .NET 5 trở đi đã tập trung vào Razor Pages và Blazor thay vì việc sử dụng jQuery.
* jQuery UI không còn được phát triển tích cực và không có nhiều bản cập nhật mới. Điều này có nghĩa rằng nó có thể không đáp ứng được các yêu cầu mới nhất về giao diện người dùng và an toàn.

1. **Jquery Templates**

* jQuery Templates là một tính năng của jQuery giúp tạo và quản lý mẫu HTML tương tác trong ứng dụng web.
* Trong ngữ cảnh của ASP.NET MVC, có thể sử dụng jQuery Templates để dễ dàng tạo và điều khiển nội dung HTML dựa trên dữ liệu từ máy chủ, thường thông qua AJAX và JSON.
* Cách thức hoạt động của jQuery Templates thường bao gồm định dạng mẫu HTML và sau đó sử dụng dữ liệu để lấp đầy mẫu này.
* Tuy nhiên, cần lưu ý rằng jQuery Templates không còn được phát triển và khuyến nghị sử dụng trong các ứng dụng web hiện đại. Dưới đây là một số lý do:
* Không còn được hỗ trợ: jQuery Templates không còn được duy trì và phát triển tích cực bởi cộng đồng jQuery hoặc bởi jQuery Core Team. Điều này có nghĩa rằng không có cập nhật mới và không có sự khắc phục lỗi hoặc cải tiến cho tính năng này.
* Tùy chọn thay thế: Có nhiều thư viện và cách tiếp cận hiện đại hơn để quản lý dữ liệu và tạo các mẫu HTML trong ứng dụng web. Ví dụ, có thể sử dụng các framework JavaScript như React, Angular, Vue.js, hoặc thậm chí là Razor Pages và Blazor trong ASP.NET Core để quản lý dữ liệu và tạo các mẫu dựa trên thành phần.
* Hiệu suất và hiệu quả: Các thư viện và framework hiện đại thường cung cấp hiệu suất và hiệu quả tốt hơn trong việc tải dữ liệu và tạo giao diện người dùng tương tác. Chúng cung cấp các tính năng như virtual DOM và các cách tối ưu hóa khác để làm cho ứng dụng web chạy nhanh và mượt mà.
* Phát triển dự án: Dự án ASP.NET Core và ASP.NET 5 trở đi đã tập trung vào việc sử dụng Razor Pages và Blazor để tạo ứng dụng web tương tác. Do đó, không còn có sự ưu tiên cho việc sử dụng jQuery Templates.