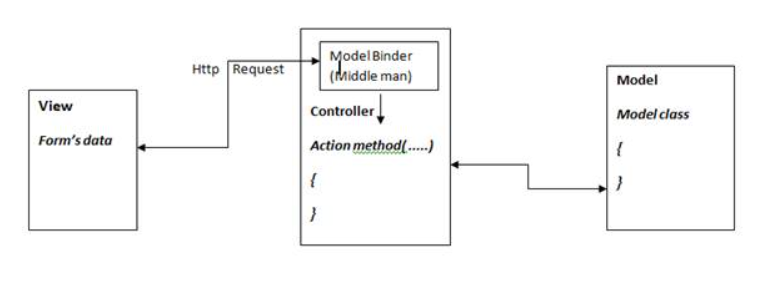
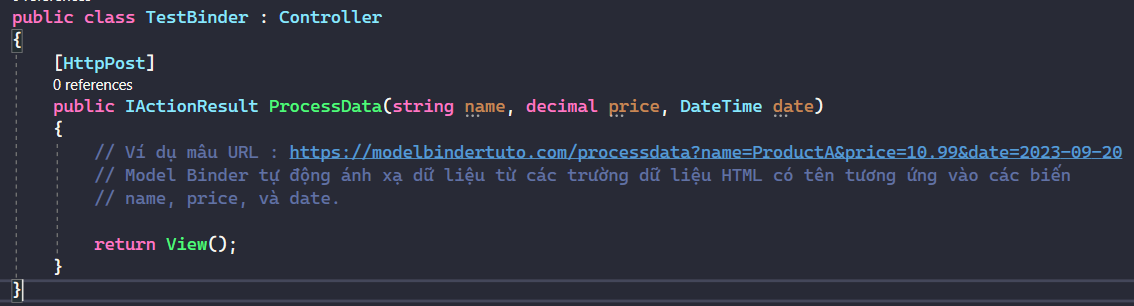
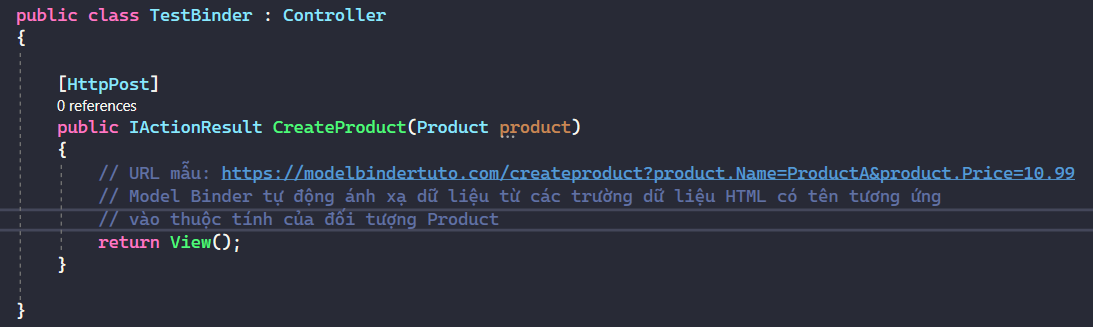
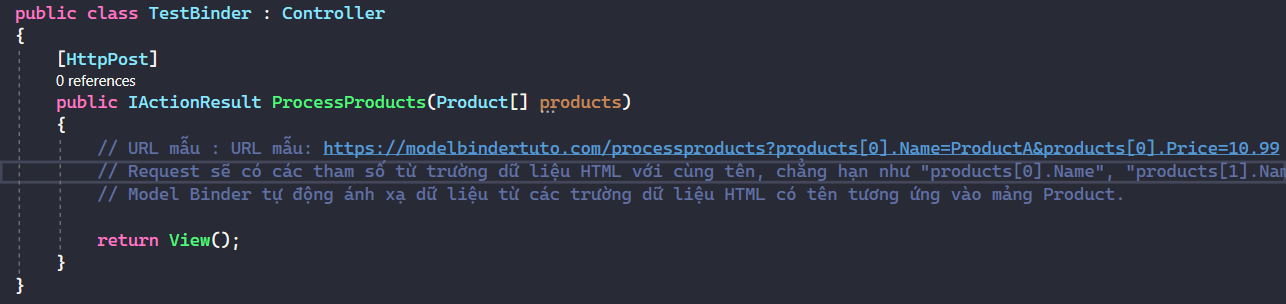
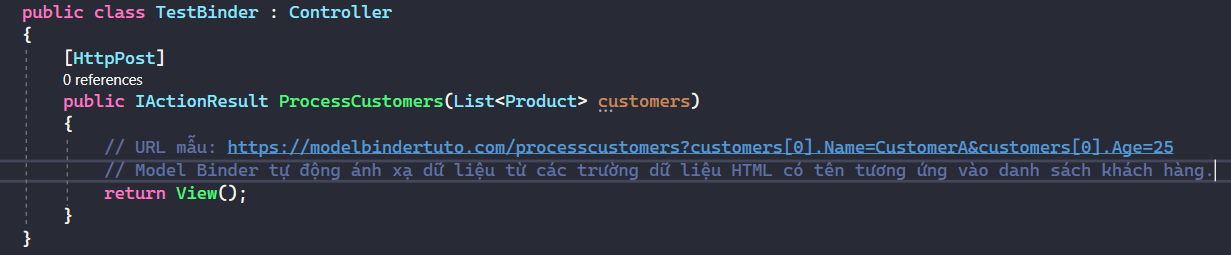
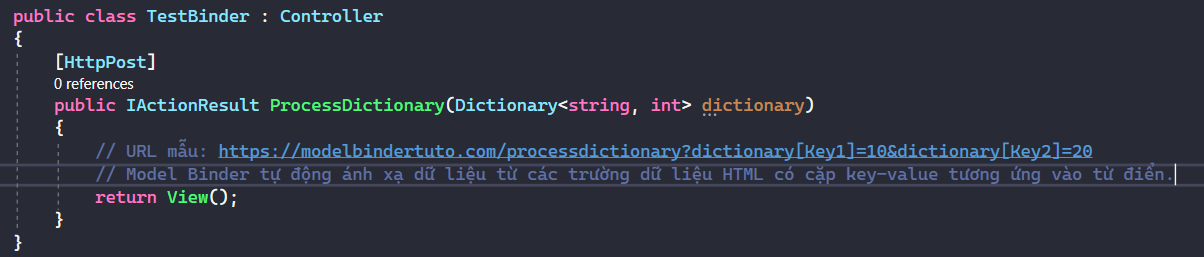
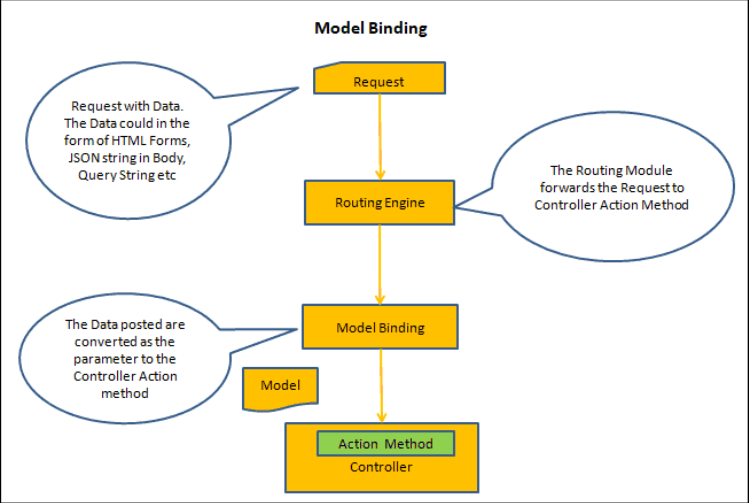
* Model Binder có tác dụng ánh xạ một request từ browser vào trong một đối tượng
* Các Model Binder giống như công cụ chuyển đổi kiểu dữ liệu bởi vì chúng có thể chuyển đổi các HTTP Request thành các đối tượng được truyền vào phương thức action.

1. **Default Model Binder.**

* Default Model Binder trong ASP.NET rất linh hoạt và đủ tốt để ánh xạ dữ liệu từ request của trình duyệt vào nhiều loại đối tượng, bao gồm :
* Các kiểu thông dụng: Model Binder có thể ánh xạ các kiểu thông dụng như string, decimal, và DateTime v.v... Điều này cho phép lấy dữ liệu từ các trường dữ liệu HTML và biến đổi nó thành các kiểu dữ liệu .NET phổ biến.
* Các lớp tự xây dựng (custom classes): Model Binder cũng có khả năng ánh xạ dữ liệu vào các lớp tự xây dựng như Product hoặc Customer v.v . Có thể tạo các đối tượng của các lớp này và sử dụng Model Binder để điền dữ liệu vào các thuộc tính của chúng.

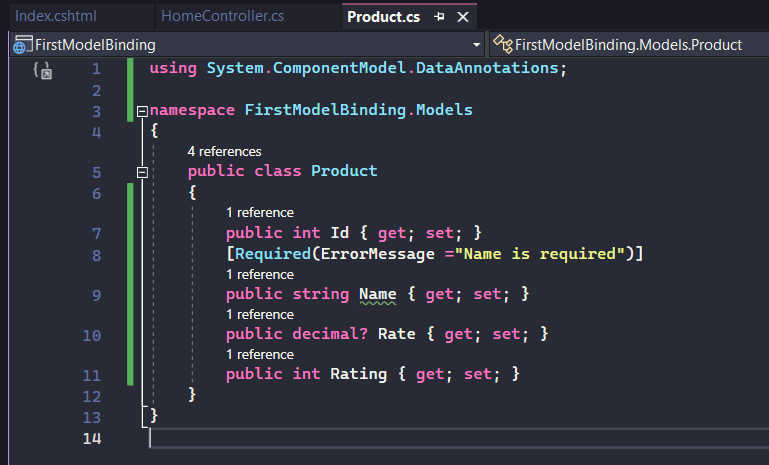
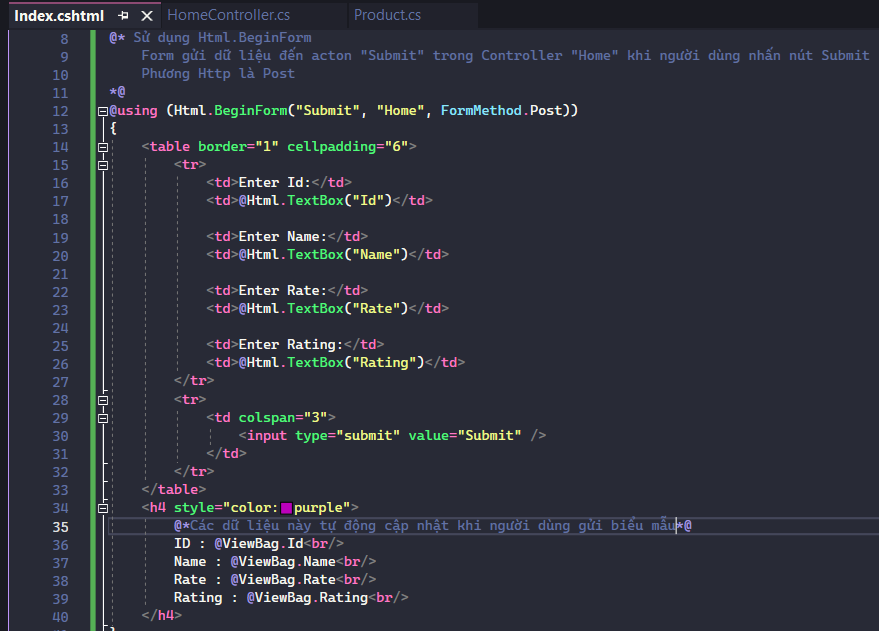


* Mảng (arrays): Model Binder hỗ trợ ánh xạ dữ liệu vào các mảng .NET, chẳng hạn như Product[] hoặc string[]. Điều này giúp thu thập dữ liệu từ nhiều trường dữ liệu có cùng tên trong request.
* Tập hợp (collections): Model Binder hỗ trợ ánh xạ dữ liệu vào các tập hợp .NET như IEnumerable<T>, ICollection<T>, IList<T>, T[], Collection<T>, và List<T>. Có thể dễ dàng thu thập dữ liệu từ các trường dữ liệu HTML và biến đổi chúng thành các tập hợp.
* Từ điển (dictionaries): Model Binder cũng hỗ trợ ánh xạ dữ liệu vào các kiểu từ điển .NET như IDictionary<TKey, TValue> và Dictionary<TKey, TValue>. Điều này cho phép tổ chức dữ liệu theo cặp key – value.
* Model binding là cơ chế tự động map giá trị từ truy vấn HTTP và ánh xạ chúng vào tham số của action method trong Controller hoặc vào thuộc tính của model class tương ứng.
* Cơ chế Model Binding giải phóng lập trình viên khỏi việc trích xuất dữ liệu thủ công, khỏi truy vấn rồi tự gán cho từng biến cục bộ. Cho phép dễ dàng Bind các giá trị này vào các tham số của action method. Các tham số này có thể là kiểu nguyên thủy hoặc kiểu đối tượng phức tạp.
* Cơ chế hoạt động của Model Binding :

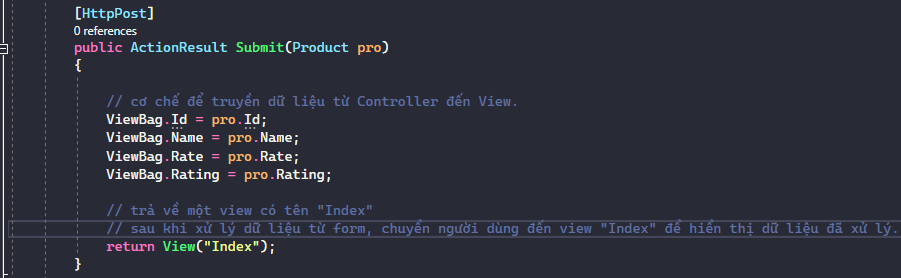
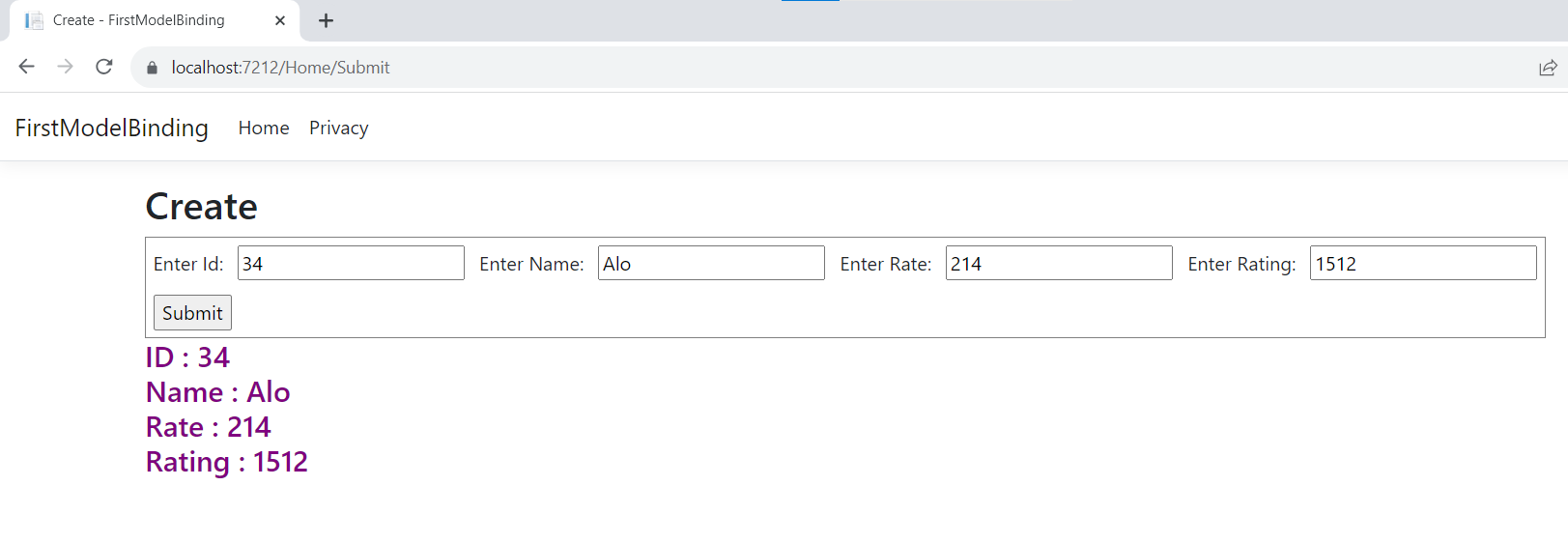
Khi người dùng click vào nút Submit thì một request Post được gửi lên server với dữ liệu có thể là : Form Data, QueryString, Route Parameter...

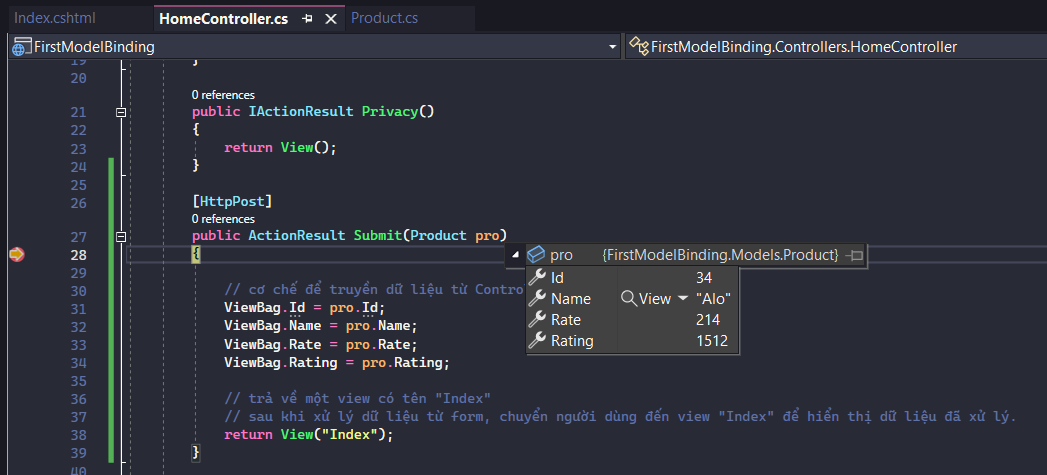
Model Binder sẽ được kích hoạt trước khi action method trong Controller được gọi. Nó tìm dữ liệu thỏa mãn trong form data, query string và request parameter trong HTTP Request.

Sau đó nó sẽ Binding các giá trị vào tham số của action method qua tên.

1. Tạo model Product
2. Tạo và hiển thị form

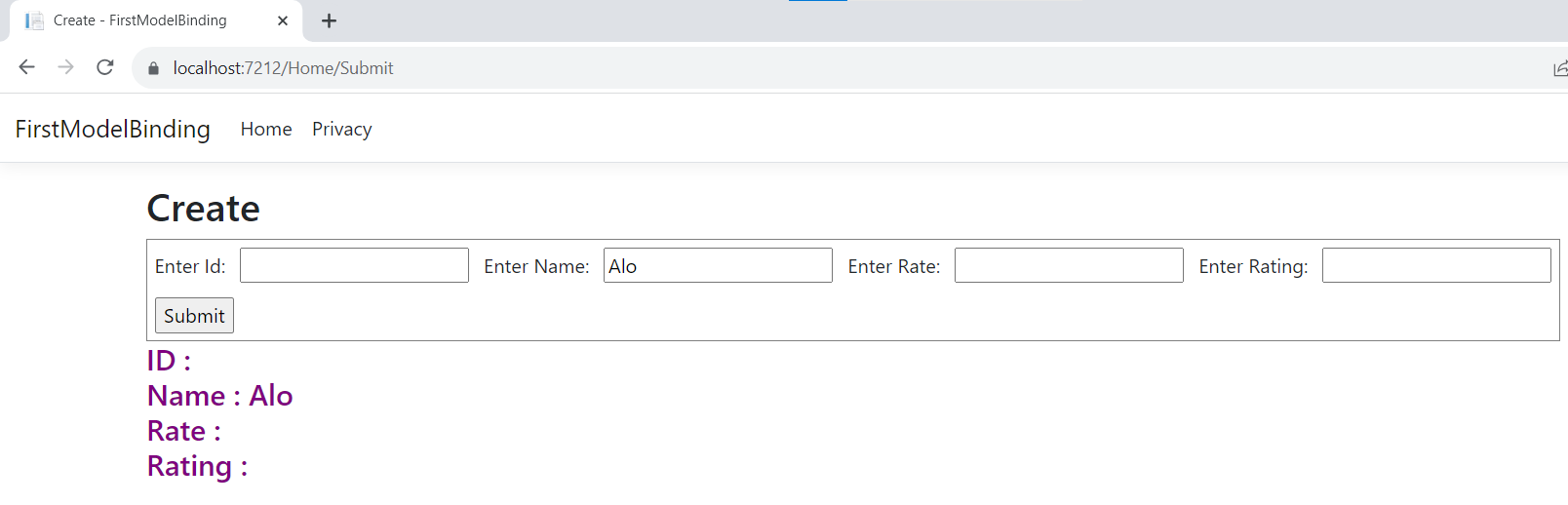
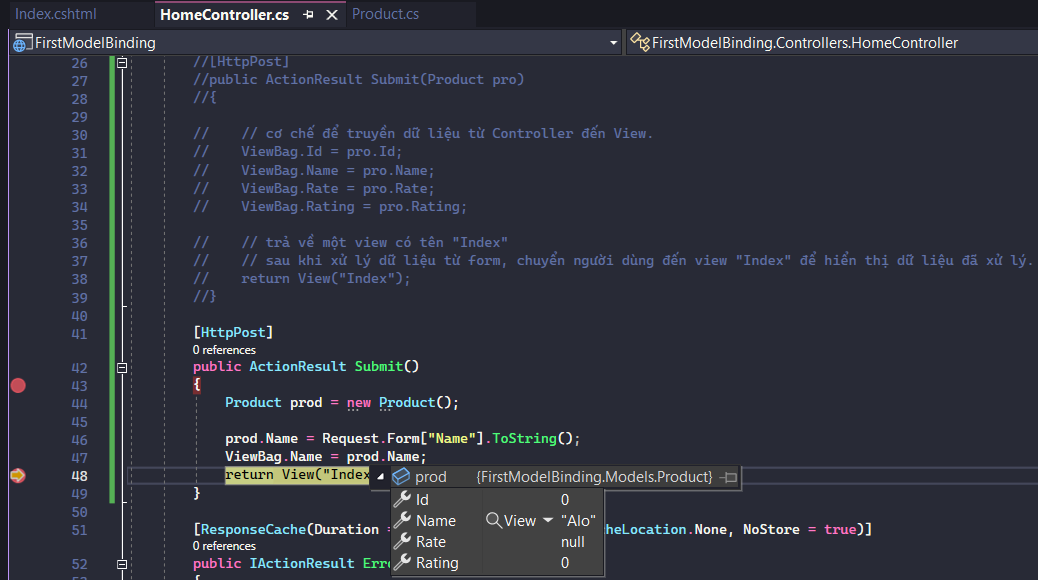
* Khi người dùng gửi biểu mẫu, dữ liệu từ các trường ID, Name, Rate, Rating sẽ tự động ánh xạ vào “ViewBag.ID”, “ViewBag.Name”,…vào action “Submit” trong Controller.

1. Tạo Controller với action method để xử lý Submit method
2. Kết quả



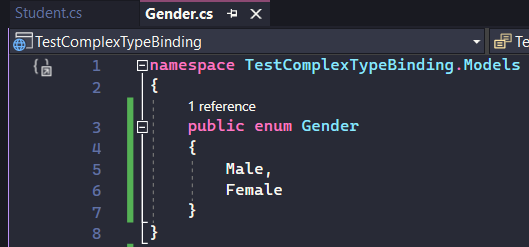
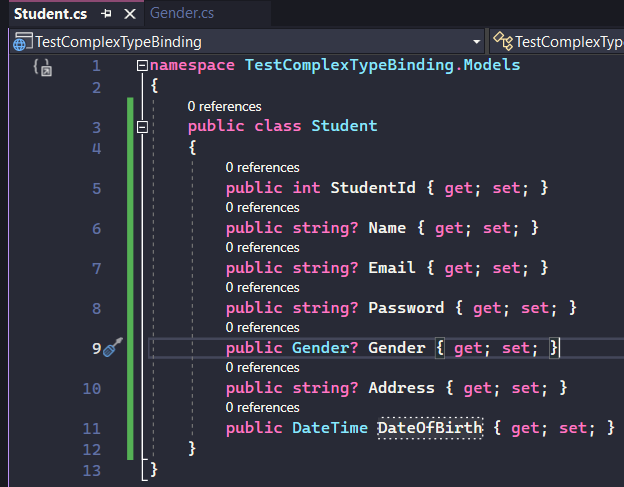
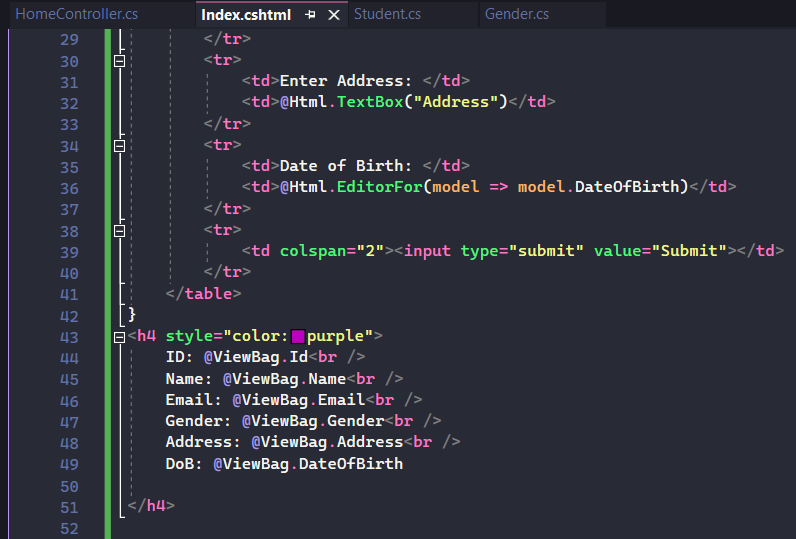
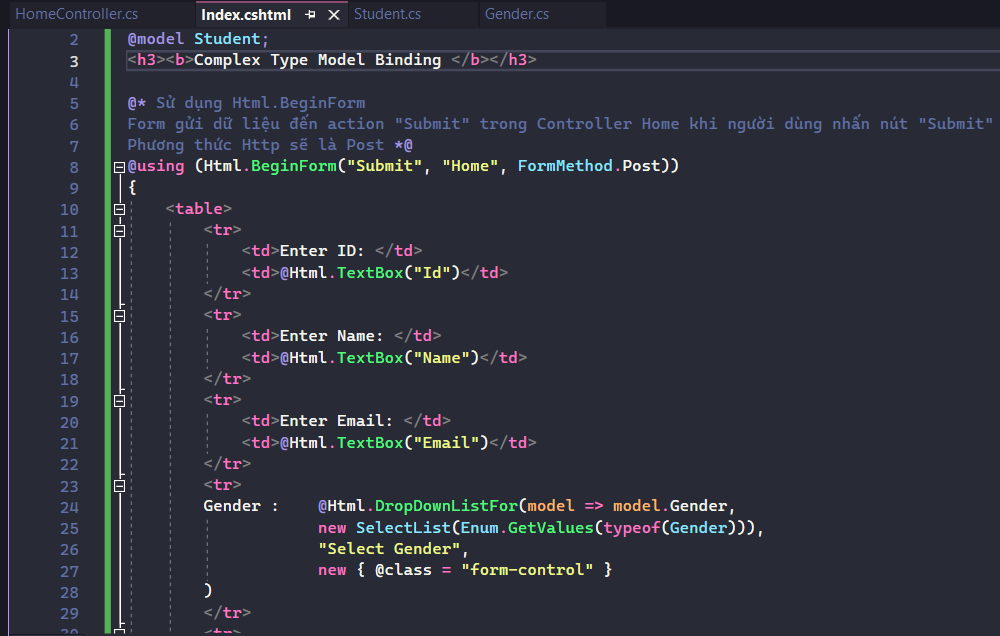
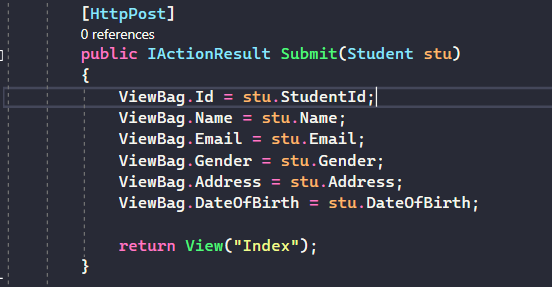
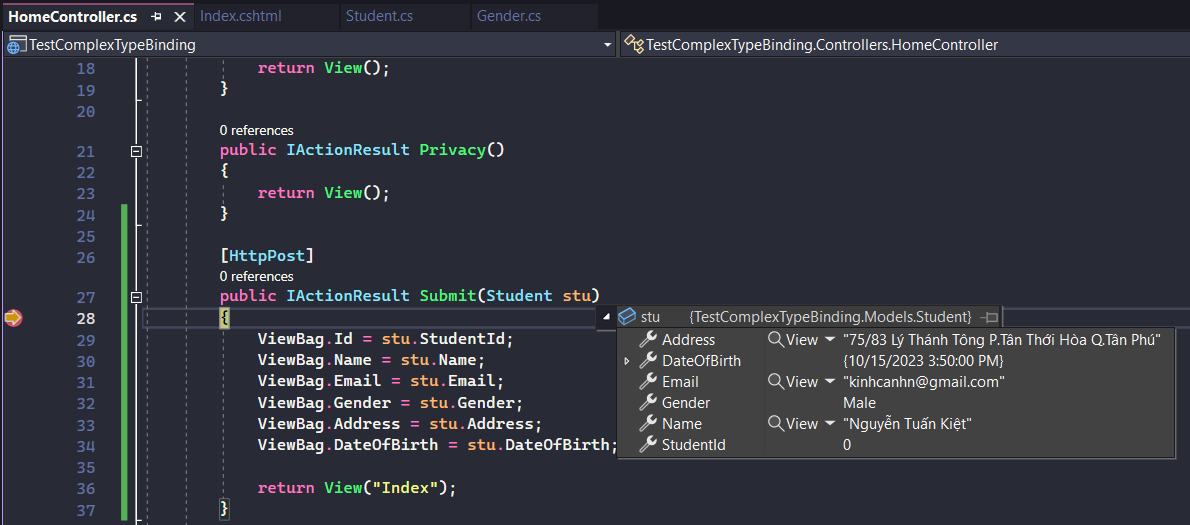
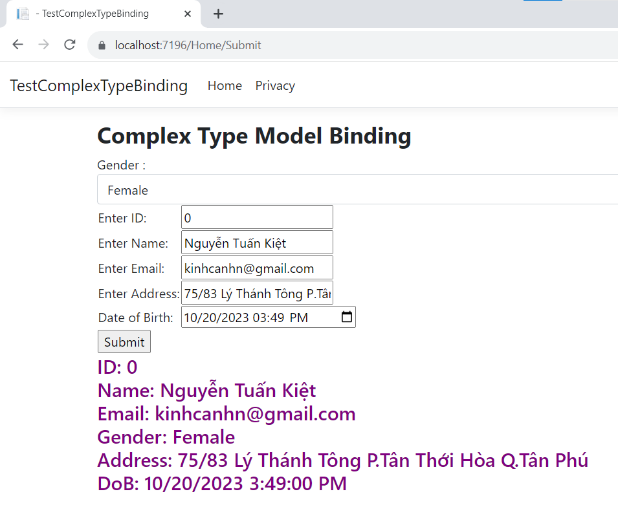
**Lưu ý : Để Model Binding làm việc đúng**

* Thuộc tính phải match với request data
* Các thuộc tính phải đặt public set

1. So với việc sử dụng trích xuất dữ liệu thông thường và không sử dụng Binding

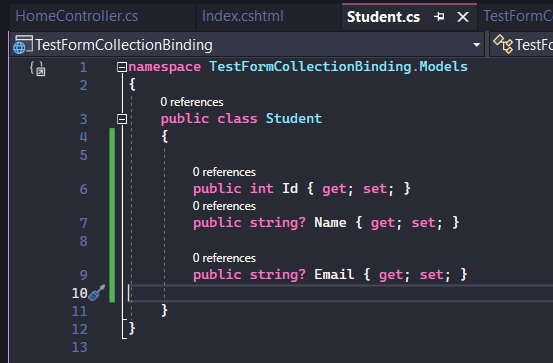
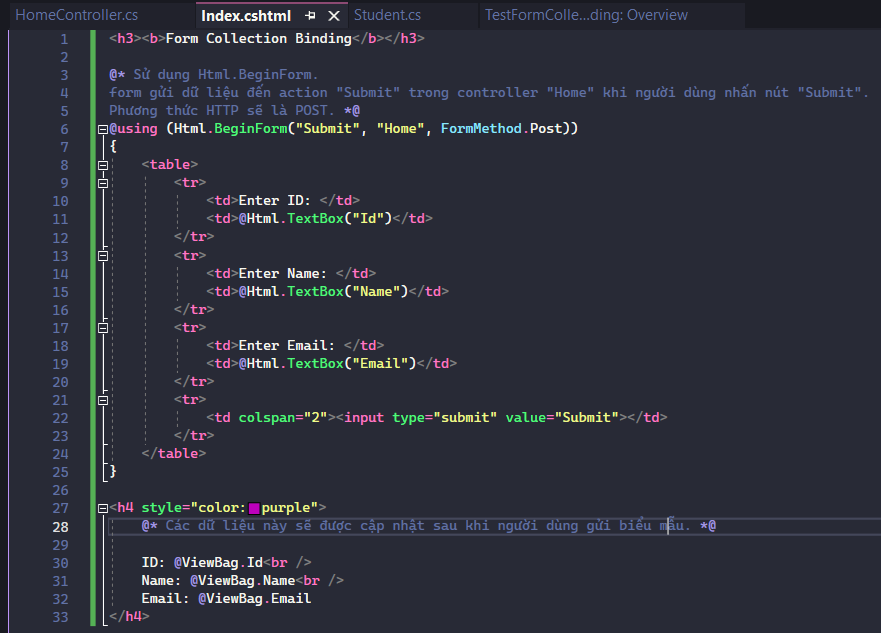
* **Cơ chế Model Binding giải phóng lập trình viên khỏi việc trích xuất dữ liệu thủ công, khỏi truy vấn rồi tự gán cho từng biến cục bộ. Cho phép dễ dàng Bind các giá trị này vào các tham số của action method.**

1. **Complex Type Model Binding**

* Complex Type Model Binding là khả năng ánh xạ dữ liệu từ truy vấn HTTP vào các kiểu dữ liệu phức tạp hoặc đối tượng tùy chỉnh (custom objects)
* Điều này cho phép tự định nghĩa cấu trúc dữ liệu phức tạp dựa trên nhu cầu của ứng dụng và sau đó sử dụng Model Binding để tự động ánh xạ dữ liệu từ các trường dữ liệu HTML hoặc truy vấn URL vào các đối tượng phức tạp này.
* Ví dụ với custom objects:
* Tạo ra đối tượng enum Gender đại diện cho thuộc tính giới tính đối tượng Student chứa dữ liệu học sinh
* Tạo và hiển thị form cho việc submit form thông tin học sinh (Student)
* Tạo Controller với các action method để xử lý Submit method
* ****Kết quả
* ASP.NET Core’s Model Binding systemmạnh mẽ và linh hoạt, có khả năng xử lý nhiều tình huống khác nhau từ simple types đến complex nested objects và các collections phức tạp.

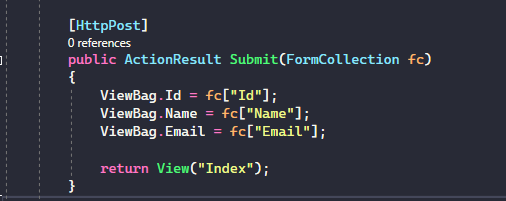
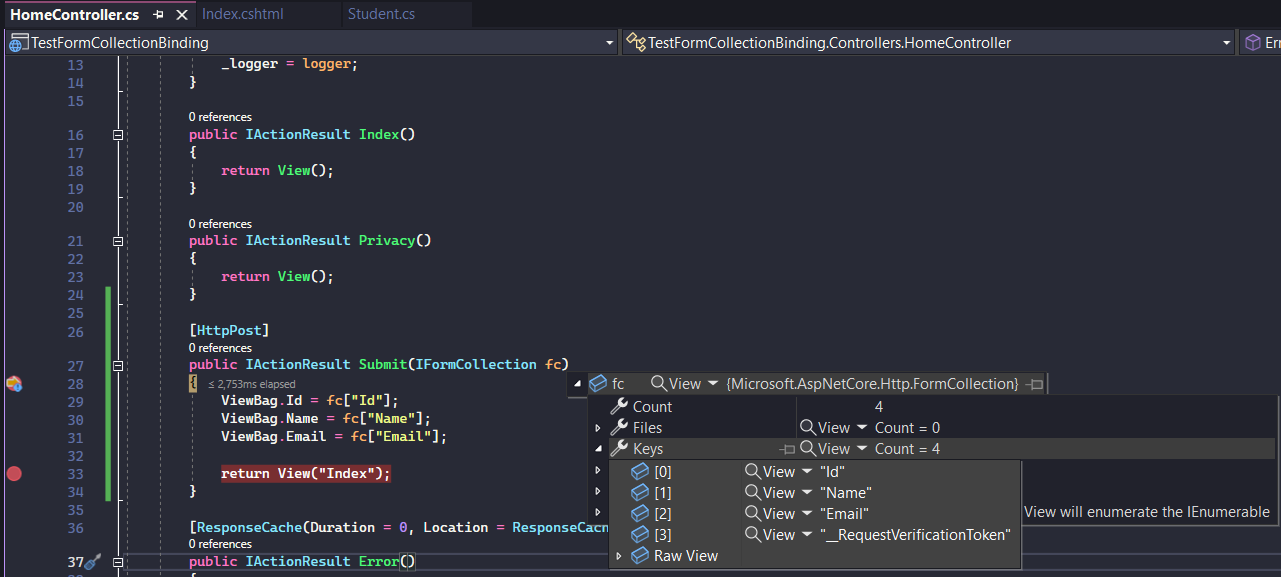
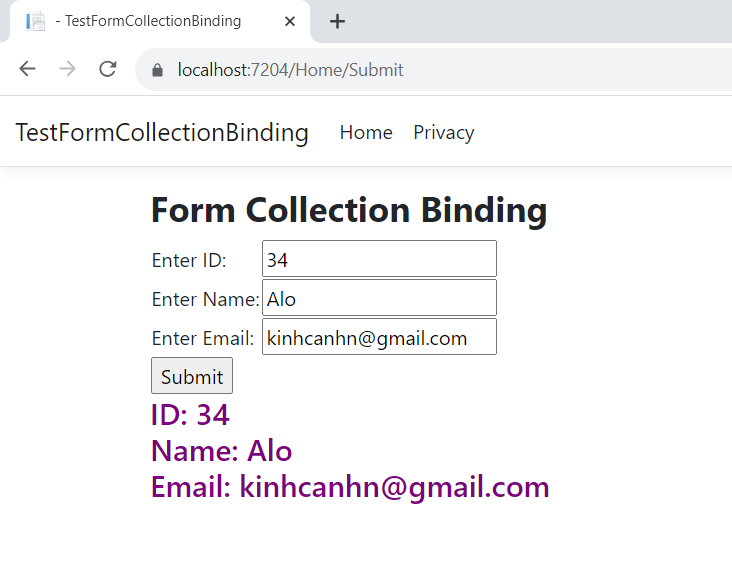
1. **Form Collection Binding**

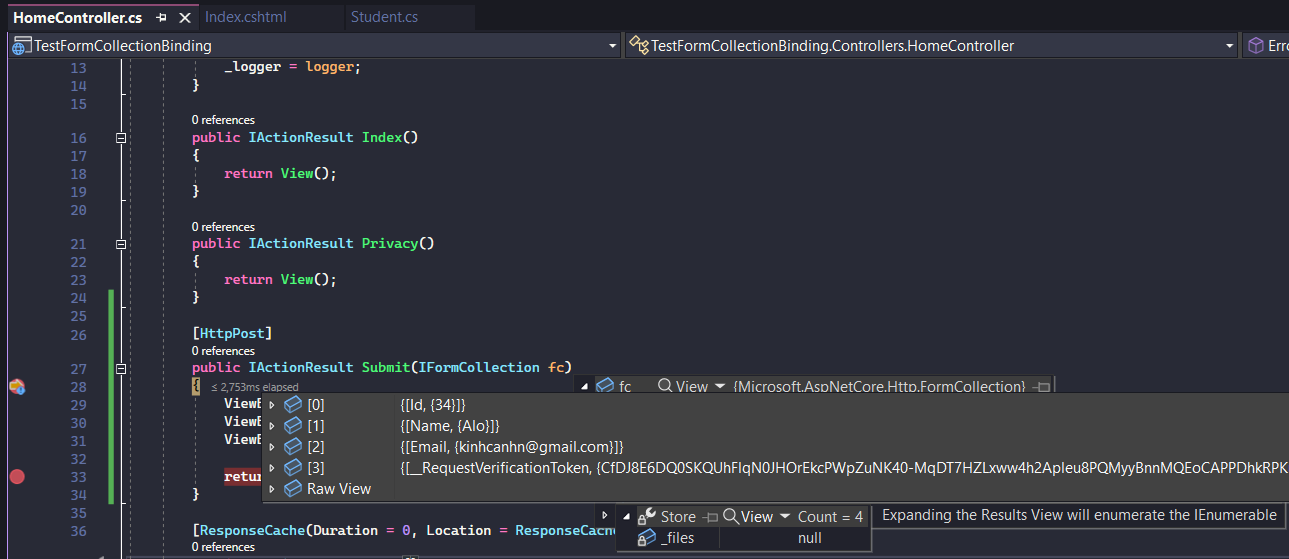
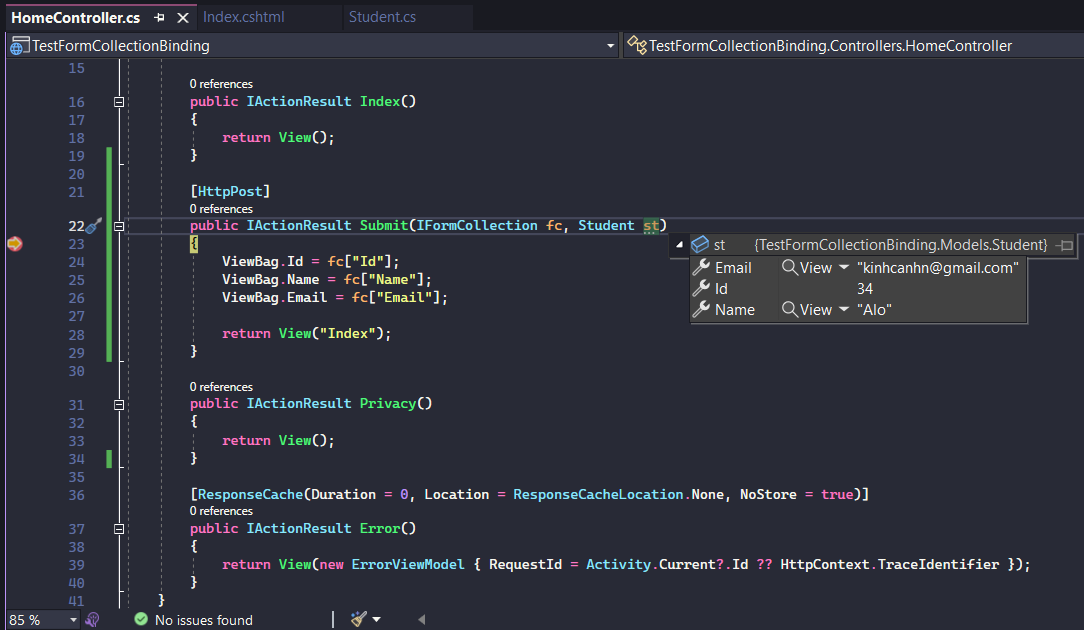
* Form Collection Model Binding là một tính năng cho phép tự động ánh xạ dữ liệu từ biểu mẫu HTML (form data) vào một đối tượng FormCollection.
* Điều này hữu ích khi cần xử lý dữ liệu từ một biểu mẫu mà không biết trước cấu trúc của nó hoặc khi cần xử lý dữ liệu từ nhiều trường dữ liệu với các tên khác nhau.
* Ví dụ về Form Collection Model Binding:

1. Tạo model Student
2. Tạo và hiển thị form cho việc submit form thông tin học sinh (Student)

Khi người dùng gửi biểu mẫu, dữ liệu từ các trường ID, Name, Email sẽ tự động ánh xạ vào “ViewBag.ID”, “ViewBag.Name”, “ViewBag.Email” trong action “Submit” trong Controller.

Form Collection Binding cho phép bạn thu thập dữ liệu từ form HTML mà không cần tạo một đối tượng model cụ thể, và dữ liệu này có thể truyền vào controller dễ dàng.

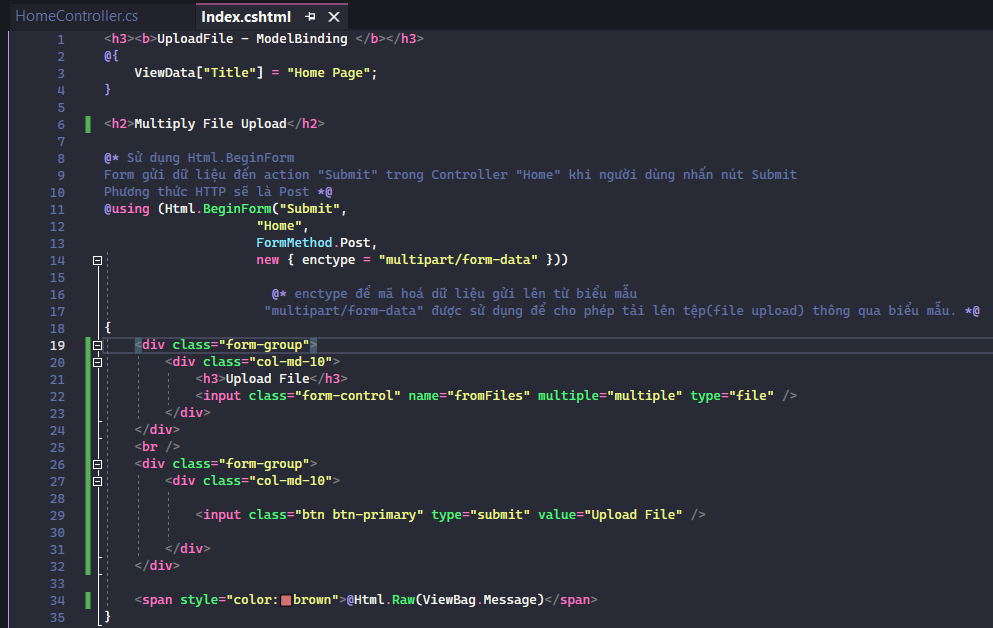
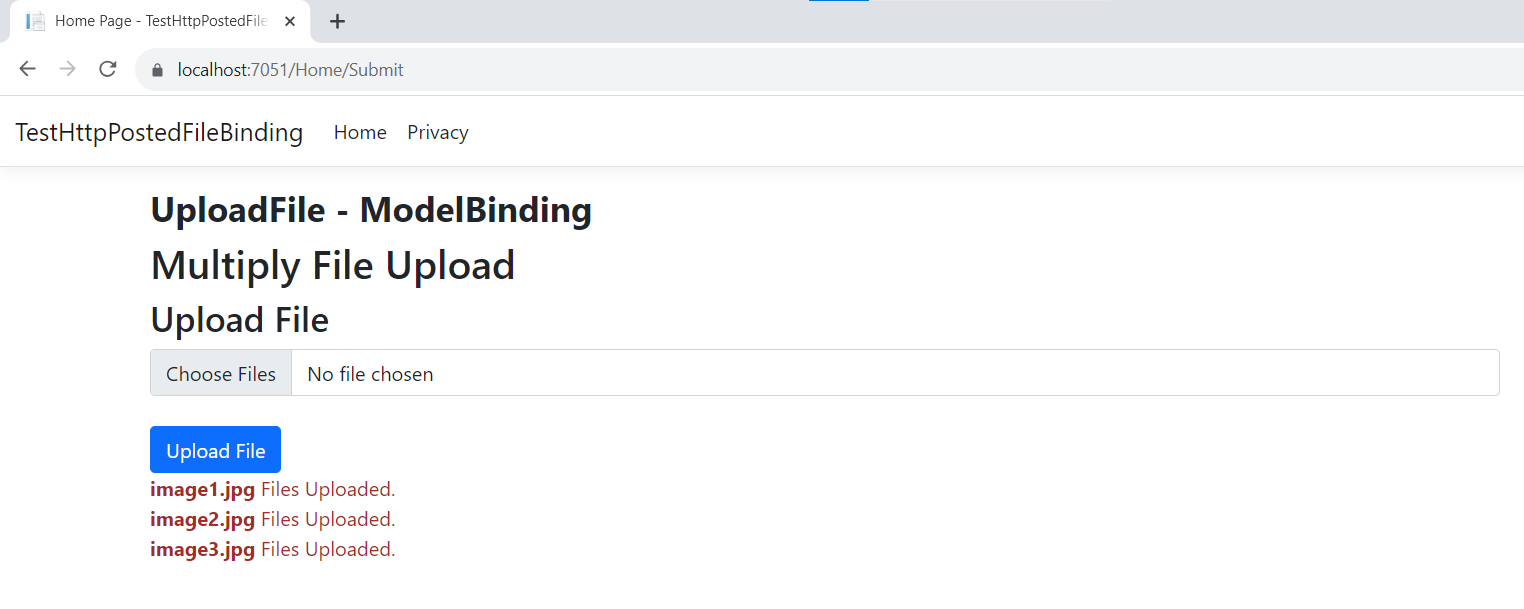
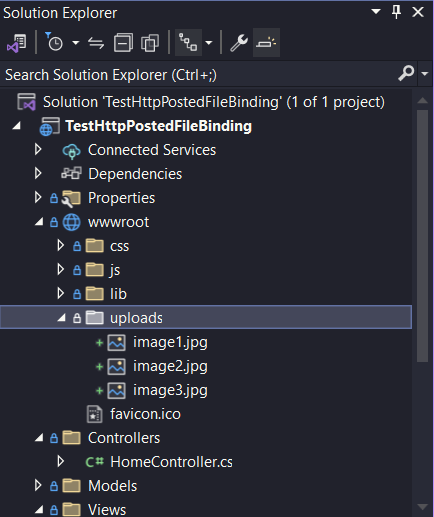
1. Tạo Controller với các action method để xử lý Submit method
2. ****Kết quả

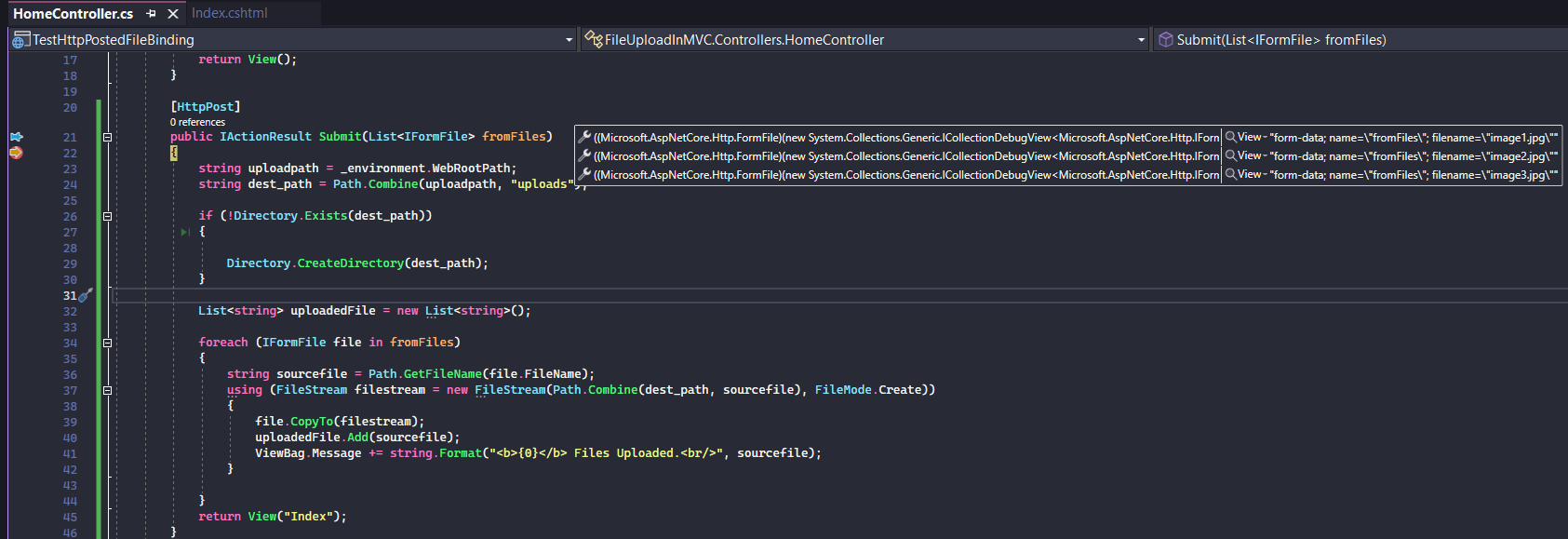


* Tạo một đối tượng mô hình (model object) như Student trong các ứng dụng ASP.NET giúp quản lý dữ liệu từ Form Collection một cách hiệu quả hơn và giúp mã nguồn trở nên dễ đọc và dễ bảo trì hơn.
* Đối tượng mô hình giúp định nghĩa cấu trúc dữ liệu cho dữ liệu được nhập từ Form Collection. Trong trường hợp Student, định nghĩa các thuộc tính như Id, Name, và Email để biểu thị dữ liệu từ form. Điều này tạo ra một cấu trúc dữ liệu rõ ràng và dễ quản lý.
* Đối tượng Student giúp quản lý dữ liệu đầu vào từ Form Collection. Khi nhận được dữ liệu từ Form Collection, có thể ánh xạ nó vào Student và sử dụng thuộc tính của đối tượng này một cách dễ dàng trong action method, đồng thời giảm bớt sự phức tạp trong việc xử lý và quản lý dữ liệu.

1. **HTTP Posted File Model Binding**

* HTTP Posted File Model Binding là một tính năng cho phép tự động ánh xạ các tệp (files) gửi lên từ trình duyệt của người dùng vào đối tượng IFormFile.
* Điều này rất hữu ích khi cần cho phép người dùng tải lên các tệp, chẳng hạn như hình ảnh, tài liệu, hoặc bất kỳ loại tệp nào khác.
* Microsoft đã vĩnh viễn loại bỏ lớp HttpPostedFileBase khỏi ASP.NET Core và giới thiệu một giao diện mới là IFormFile để tải lên các tệp
* Ví dụ về HTTP Posted File Model Binding upload multiply file:

1. Tạo và hiển thị cho việc upload file
2. Tạo Controller với các action method để xử lý Submit method
3.  Kết quả



1. **Bind Attribute**

* Bind Attribute trong ASP.NET Core MVC là một cách để kiểm soát việc ánh xạ dữ liệu từ truy vấn HTTP hoặc form HTML vào thuộc tính của Model hoặc tham số của method xử lý trong Controller. Giúp bảo vệ ứng dụng khỏi các cuộc tấn công gửi dữ liệu độc hại (phân tích dữ liệu) từ client.
* Bind Attribute cung cấp một số thuộc tính để điều chỉnh cách ánh xạ dữ liệu hoạt động. Dưới đây là một số thuộc tính chính của Bind Attribute:
* Include : Thuộc tính Include cho phép bạn chỉ định danh sách các tên thuộc tính được ánh xạ. Chỉ các thuộc tính có tên nằm trong danh sách Include mới được ánh xạ, và các thuộc tính không nằm trong danh sách sẽ bị loại bỏ.
* Exclude : Thuộc tính Exclude cho phép bạn chỉ định danh sách các tên thuộc tính không được ánh xạ. Nếu một thuộc tính có tên trùng khớp với danh sách Exclude, thì nó sẽ bị loại trừ và không được ánh xạ.
* Prefix : Thuộc tính Prefix cho phép bạn thiết lập một tiền tố được sử dụng khi tạo các trường HTML cho việc ánh xạ. Điều này thường được sử dụng để tạo ra các tên trường dữ liệu HTML với tiền tố cụ thể cho việc ánh xạ dữ liệu.
* IsPropertyAllowed : Phương thức IsPropertyAllowed cho phép bạn kiểm tra xem một thuộc tính cụ thể có được ánh xạ hay không. Nó trả về true nếu thuộc tính được phép ánh xạ, và false nếu không được phép.
* **Kể từ phiên bản ASP.NET Core, đặc biệt là từ ASP.NET Core 2.0 trở đi, BindAttribute không còn được hỗ trợ nữa. ASP.NET Core không còn sử dụng BindAttribute để kiểm soát việc ánh xạ dữ liệu từ truy vấn HTTP hoặc biểu mẫu HTML vào model**