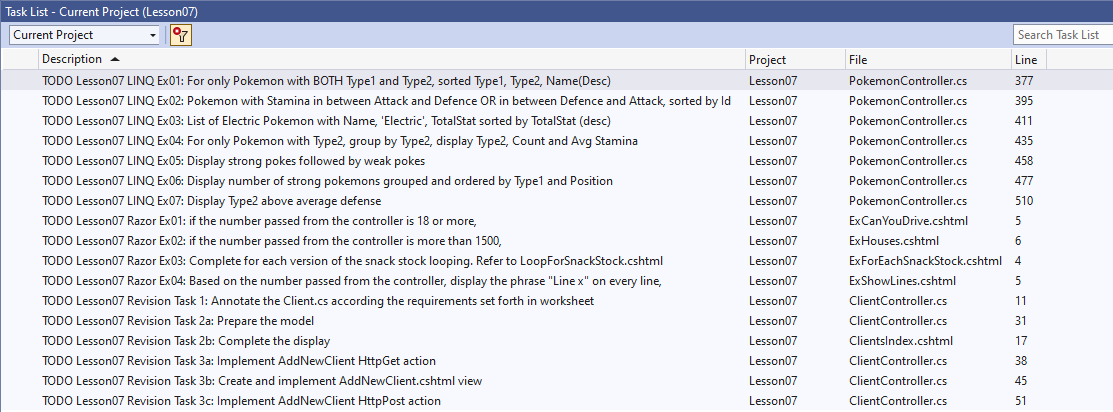
Lesson 07 – Putting It All Together

**Using Task List**

1. Open Task List window from the menu VIEW, Task List. Use the Filter function in the Project heading to display only the Task for Lesson07.



# Revision

Razor: Perform tasks Razor Ex01 to Ex04. Open ExCanYouDrive.cshtml, ExForEachSnackStock.cshtml, ExHouses.cshtml and ExShowLines.cshtml to view those tasks. Once you have completed these four exercises, screen shot the output and paste here.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. LINQ: Peform tasks LINQ Ex01 to Ex07.

# AttachyBridalHouse

1. The AttachyBridalHouse Webapp needs two database tables. Write down the names of the tables, the primary keys and foreign key (if any) here.

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Primary Key** | **Foreign Key** |
| Package | Id | - |
| Client | Id | PackageId |

1. What is the main controller for the AttachyBridalHouse Webapp. Complete the following table for the main use cases and the required files.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | **Controller/Action** | **Model** | **View** |
| List | ClientController/ ClientsIndex() | Client.cs | ClientsIndex.cshtml |
| Create | ClientsController/ AddNewClient() | Client.cs | AddNewClient.cshtml |
|  | ClientsController/ AddNewClient(HttpPost) | Client.cs | AddNewClient.cshtml |
| Update | ClientController/UpdateClient() | Client.cs | Update.cshtml |
|  | ClientController/UpdateClient(Httppost) | Client.cs | Update.cshtml |
| Delete | ClientController/DeleteClient() | Client.cs | ClientsIndex.html |
| List | ClientController/ ClientsIndex() | Client.cs | ClientsIndex.cshtml |

1. Data annotate the Client class based on the following validation requirements:

|  |  |  |
| --- | --- | --- |
| Property | Validation Requirement | Error Message |
| Name | Mandatory | Name cannot be empty. |
| PackageId | Mandatory | Please select a photo package. |

1. Confirmation dialog boxes should be used to allow user to cancel operations. Note that you might encounter the confirmation dialog box will appear even though user enters the invalid input. In this case, you can use **$(this).IsValid()** to check whether all information entered are valid.
2. For message to be passed to **Client Index** action:

|  |  |
| --- | --- |
| Scenario | Message |
| Insert/Update/Delete performed successfully | New Client added/Client updated/Client deleted |
| Insert/Update/Delete failed | Failed to add new Client./Failed to update client/Failed to delete client |
| Invalid input | Invalid information entered! |

1. Below are values for **PaymentMode**:

|  |  |
| --- | --- |
| Value | Text |
| C | Cash |
| I | Installment |
| CI | Cash and Installment |

1. If you have completed all the features in step 5 above, you can continue to attempt the **Advanced Task** in Appendix B of the Problem Statement.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Table

Description automatically generated

# Testing your App

Provide test cases here and screen shots to demonstrate that your web app is functioning well.

*— End of Worksheet —*