**Steps for Creating Online Shopping StoreWithLinq Using MVC:**

1. Manage Project
   1. Create BO's Folder .
   2. Util Folder

->Helper Class.

* 1. Create Area For Admin.
  2. Create ShoppingImages Folder.

1. Manage UserProfile
   1. Create
   2. Edit
   3. Delete
   4. Details
2. Manage Category
   1. Create
   2. Edit
   3. Delete
   4. Details
3. Manage SubCategory
   1. Create
   2. Edit
   3. Delete
   4. Details
4. Manage Product
   1. Create
   2. Edit
   3. Delete
   4. Details
5. Manage Order
   1. Edit
   2. Delete
   3. Details

**Setting up the Project**

Tables

**Create a UserProfile Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKUserId | Int | PK, Not Null |
| FKRoleId | Int | FK, Not Null |
| UserName | varcahar | Not null,Unique |
| FirstName | Varchar(50) | Not null |
| LastName | Varchar(50) | Allow nulls |
| EmailId | Varchar(50) | Not null |
| password | Varchar(50) | Not Null |
| PhoneNo | Varchar(50) | Not null |
| AddressLine1 | Varchar(50) | Not null |
| AddressLine2 | Varchar(50) | Allow nulls |
| Street | Varchar(50) | Not null |
| City | Varchar(50) | Not null |
| Country | Varchar(20) | Not null |
| ZipCode | BigInt | Not Null |
| CreatedDate | DateTime | Not Null |
| UpdatedDate | DateTime | Not Null |
| CreatedByUserId | Int | FK, Allow Nulls |
| UpdatedByUserId | Int | FK,Allow Nulls |
| IsActive | Bit | Not Null |

**Create a Category Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKCategoryId | Int | PK, Not Null |
| CategoryName | varchar(50) | Not null,Unique |
| CreatedDate | DateTime | Not null |
| UpdatedDate | DateTime | Allow null |
| FKCreatedByUserId | Int | FK, Not Null |
| FKUpdatedByUserId | Int | FK, Allow Null |
| IsActive | bit | Not null |

**Create a SubCategory Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKSubCategoryId | Int | PK, Not Null |
| FKCategoryId | Int | FK,Not null |
| SubCategoryName | VarChar(50) | Not Null,Unique |
| ImageName | VarChar(50) | Not Null |
| ImagePath | VarChar(Max) | Not Null |
| CreatedDate | DateTime | Not null |
| UpdatedDate | DateTime | Allow null |
| FKCreatedByUserId | Int | FK, Not Null |
| FKUpdatedByUserId | Int | FK, Allow Null |
| IsActive | bit | Not null |

**Create a Product Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKProductId | Int | PK, Not Null |
| FKCategoryId | Int | FK,Not null |
| FKSubCategoryId | Int | FK,Not null |
| ProductName | VarChar(50) | Not Null,Unique |
| Description | VarChar(500) | Not Null |
| Quantity | Int | Not Null |
| Price | Money | Not Null |
| CreatedDate | DateTime | Not null |
| UpdatedDate | DateTime | Allow null |
| FKCreatedByUserId | Int | FK, Not Null |
| FKUpdatedByUserId | Int | FK, Allow Null |
| IsActive | bit | Not null |

**Create a ProductImage Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKImageId | Int | PK, Not Null |
| FKProductId | Int | FK,Not Null |
| ImageName | VarChar(50) | Not Null |
| ImagePath | VarChar(Max) | Not Null |

**Create a Order Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKOrderId | Int | PK, Not Null |
| FKCustomerUserId | Int | FK,Not Null |
| FKBillingAddressId | Int | FK,Not Null |
| FKProcessByUserId | Int | FK,Allow null |
| OrderDate | DateTime | Not Null |
| DeliveryDate | DateTime | Allow Null |
| CustomerComment | VarChar(Max) | Allow Null |
| AdminComment | VarChar(Max) | Allow Null |
| Status | VarChar(50) | Not Null |

**Create a OrderDetail Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKOrderDetailId | Int | PK, Not Null |
| FKProductId | Int | FK,Not Null |
| FKOrderId | Int | FK,Not Null |
| Quantity | Int | Not Null |
| Cost | Money | Not Null |

**Create a OrderHistory Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKOrderHistoryId | Int | PK, Not Null |
| FKOrderId | Int | FK,Not Null |
| Description | VarChar(max) | Allow Null |
| Status | VarChar(20) | Allow Null |
| DateChanged | DateTime | Allow Null |

**Create a Role Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKRoleId | Int | PK, Not Null |
| RoleName | VarChar(50) | Not Null |
| IsActive | Bit | Not Null |

**Create a UserLoginHistory Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKUserLoginHistoryId | Int | PK, Not Null |
| FKUserId | Int | Not Null |
| LoginDateTime | DateTime | Not Null |
| IPAddress | VarChar(1000) | Not Null |

**Create a BillingAddress Table**

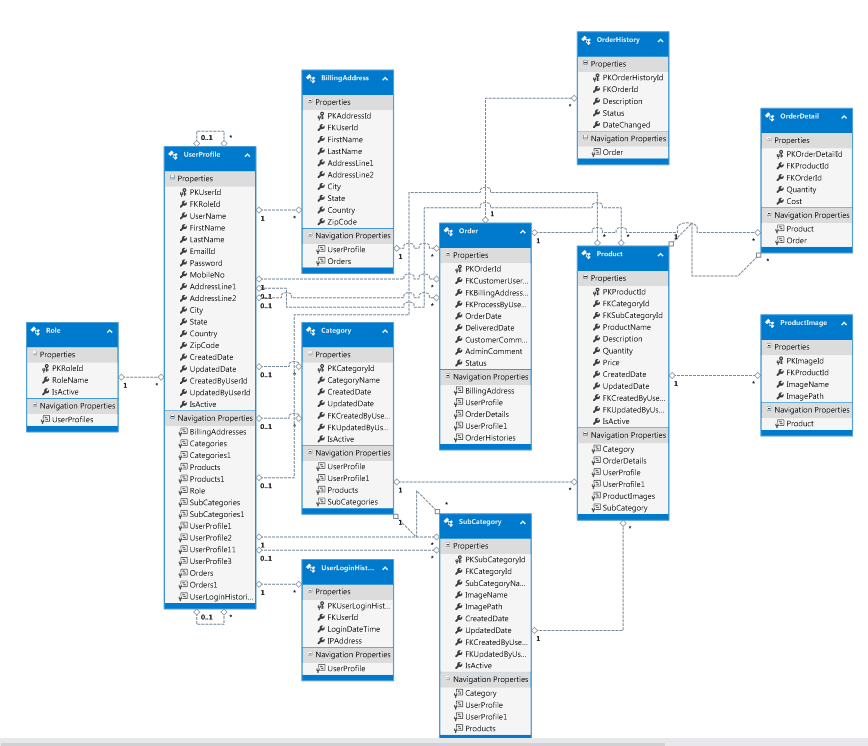
|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKAddressId | Int | PK, Not Null |
| FKUserId | Int | FK, Not Null |
| FirstName | Varchar(50) | Not null |
| LastName | Varchar(50) | Allow nulls |
| AddressLine1 | Varchar(50) | Not null |
| AddressLine2 | Varchar(50) | Allow nulls |
| City | Varchar(50) | Not null |
| Country | Varchar(20) | Not null |
| ZipCode | BigInt | Not null |
|  |  |  |

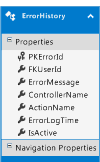
**Create a ErrorHistory Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data type** | **Constraints** |
| PKErrorHistoryId | Int | PK, Not Null |
| FKUserId | Int | Null |
| ErrorLogTime | DateTime | Not Null |
| ErrorMessage | VarChar(max) | Not Null |
| ActionName | VarChar(50) |  |

Goto Solution- Add new Item---Data templete--> EntityFrameWork---->Add name=OnlineShoppingStore-->generate from database  Next Click  New Connection  Provide Server Name  Select or Enter Database name Ok  Next  Checked Tables  Finish

1. Add edmx file using EntityFramework





1. Go to Solution Explore->Project-> Add the following to Web.Config, under <configuration>

|  |
| --- |
| <connectionStrings>  <add name="OnlineShoppingStoreDBEntities" connectionString="metadata=res://\*/OnlineShoppingStoreDB.csdl|res://\*/OnlineShoppingStoreDB.ssdl|res://\*/OnlineShoppingStoreDB.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=SERVER\SERVERDB;initial catalog=OnlineShoppingStoreDB;user id=sa;password=Pentag0n;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />  </connectionStrings> |

1. Go to Solution Explore->Project->Right click on project->Add->Class->Class name=Helper.cs

|  |
| --- |
| public class Helper  {  public static string \_UserData;  public static string UserData  {  get  {  return HttpContext.Current.User.Identity.Name;  }  }  public static int RoleId  {  get  {  return Convert.ToInt32(UserData.Split('^')[1]);  }  }  public static string RoleName  {  get  {  return UserData.Split('^')[2];  }  }  public static int UserId  {  get  {  return Convert.ToInt32(UserData.Split('^')[0]);  }  }  public static string UserName  {  get  {  return UserData.Split('^')[3];  }  }  public static string Firstname  {  get  {  return UserData.Split('^')[4];  }  }  public static string LastName  {  get  {  return UserData.Split('^')[5];  }  }  public static string EmailId  {  get  {  return UserData.Split('^')[6];  }  }  public static string Password  {  get  {  return UserData.Split('^')[7];  }  }  public static string UserAddress  {  get  {  return UserData.Split('^')[8];  }  }  public static DateTime CreatedDate  {  get  {  return Convert.ToDateTime(UserData.Split('^')[9]);  }  }  public static DateTime UpdatedDate  {  get  {  return Convert.ToDateTime(UserData.Split('^')[10]);  }  }  }  public enum Status  {  Processing = 1,  Pending,  Delivered,  Cancelled  } |

1. Go to Solution Explore->Project->Right Click on project->Add->Add a New Folder ->Folder Name=BO->Under the BO Folder-> Right click on BO Folder ->Add-> Class->class name= UserProfileBO-> Click OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class UserProfileBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public UserProfile GetUser(int userId)  {  return context.UserProfiles.Where(u => u.PKUserId == userId).SingleOrDefault();  }  public UserProfile AuthenticateUser(string userName)  {  try  {  UserProfile objUserProfile = context.UserProfiles.Where(u => u.UserName == userName).SingleOrDefault();  return objUserProfile;  }  catch (Exception ex)  {  throw ex;  }  }  public List<UserProfile> GetUsers(string emailId = "", string mobileNo = "", DateTime? fromDate = null, DateTime? toDate = null, bool? isActive = null)  {  IQueryable<UserProfile> qry = context.UserProfiles;  if (emailId != "")  qry = qry.Where(u => u.EmailId.Contains(emailId));  if (mobileNo != "")  qry = qry.Where(u => u.MobileNo.Contains(mobileNo));  if (fromDate.HasValue && toDate.HasValue)  qry = qry.Where(u => u.CreatedDate >= fromDate.Value && u.CreatedDate <= toDate.Value);  if (isActive != null)  qry = qry.Where(u => u.IsActive == isActive);  return qry.ToList();  }  public void InsertUser(UserProfile objUser)  {  try  {  objUser.CreatedDate = DateTime.Now;  if (!string.IsNullOrEmpty(Helper.UserData))  objUser.CreatedByUserId = Helper.UserId;  context.UserProfiles.Add(objUser);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateUser(UserProfile objUser)  {  try  {  objUser.UpdatedDate = DateTime.Now;  objUser.UpdatedByUserId = Helper.UserId;  context.Entry(objUser).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteUser(int userId)  {  try  {  UserProfile objUser = context.UserProfiles.Find(userId);  context.UserProfiles.Remove(objUser);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to solution ->Project->Right Click on Project->Add ->New Folder->Folder name = Data->Under the Data Folder->Right Click on Folder->Add->Class->Class name=CategoryData->Click OK and Change the namespace of class OnlineShoppingStoreMVCUsingLinq.Data To OnlineShoppingStoreMVCUsingLinq (Remove .Data),Make the class as Partial & Change the code as below

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public partial class Category  {  public string CreatedByUser  {  get;  set;  }  public string UpdatedByUser  {  get;  set;  }  public string ImageName  {  get;  set;  }  public string ImagePath  {  get;  set;  }  public int ImageId  {  get;  set;  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name= CategoryBO->Click OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class CategoryBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public Category GetCategory(int categoryId)  {  return context.Categories.Where(c => c.PKCategoryId == categoryId).SingleOrDefault();  }  public List<Category> GetCategories(bool? isActive = null, string categoryName = null)  {  try  {  IQueryable<Category> qry = context.Categories;  if (isActive != null)  qry = qry.Where(c => c.IsActive == isActive);  var q = (from c in qry  join u in context.UserProfiles on c.FKCreatedByUserId equals u.PKUserId  select new  {  PKCategoryId = c.PKCategoryId,  CategoryName = c.CategoryName,  CreatedDate = c.CreatedDate,  UpdatedDate = c.UpdatedDate,  FKCreatedByUserId = c.FKCreatedByUserId,  FKUpdatedByUserId = c.FKUpdatedByUserId,  CreatedByUser = u.UserName,  UpdatedByUser = u.UserName,  IsActive = c.IsActive  }).AsEnumerable().Select(x => new Category  {  PKCategoryId = x.PKCategoryId,  CategoryName = x.CategoryName,  CreatedDate = x.CreatedDate,  UpdatedDate = x.UpdatedDate,  FKCreatedByUserId = x.FKCreatedByUserId,  FKUpdatedByUserId = x.FKUpdatedByUserId,  CreatedByUser = x.CreatedByUser,  UpdatedByUser = x.UpdatedByUser,  IsActive = x.IsActive  });  if (!string.IsNullOrEmpty(categoryName))  qry = qry.Where(cat => cat.CategoryName == categoryName);  return q.ToList();  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertCategory(Category objCategory)  {  try  {  objCategory.CreatedDate = DateTime.Now;  objCategory.FKCreatedByUserId = Helper.UserId;  context.Categories.Add(objCategory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateCategory(Category objCategory)  {  try  {  objCategory.UpdatedDate = DateTime.Now;  objCategory.FKUpdatedByUserId = Helper.UserId;  context.Entry(objCategory).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteCategory(int categoryId)  {  try  {  Category objCategory = context.Categories.Find(categoryId);  context.Categories.Remove(objCategory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select Data Folder->Under the Data Folder->Right Click on Folder->Add

->Class->Class name=SubCategoryData ->Click OK and Change the namespace of class OnlineShoppingStoreMVCUsingLinq.Data To OnlineShoppingStoreMVCUsingLinq (Remove .Data)& Make the class as Partial & Change the code as below

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public partial class SubCategory  {  public string CreatedByUser  {  get;  set;  }  public string UpdatedByUser  {  get;  set;  }  public string CategoryName { get; set; }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name=SubCategoryBO->Click OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class SubCategoryBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public SubCategory GetsubCategory(int subCategoryId)  {  return context.SubCategories.Where(s => s.PKSubCategoryId == subCategoryId).SingleOrDefault();  }  public List<SubCategory> GetSubCategories(int categoryId = 0, bool? isActive = null)  {  try  {  context.Configuration.LazyLoadingEnabled = false;  IQueryable<SubCategory> qry = context.SubCategories;  if (categoryId != 0)  qry = qry.Where(s => s.FKCategoryId == categoryId);  if (isActive != null)  qry = qry.Where(s => s.IsActive == isActive);  var q = (from s in qry  join u in context.UserProfiles on s.FKCreatedByUserId equals u.PKUserId  join c in context.Categories on s.FKCategoryId equals c.PKCategoryId  select new  {  PKSubCategoryId = s.PKSubCategoryId,  FKCategoryId = s.FKCategoryId,  CategoryName = c.CategoryName,  SubCategoryName = s.SubCategoryName,  CreatedDate = s.CreatedDate,  UpdatedDate = s.UpdatedDate,  FKCreatedByUserId = s.FKCreatedByUserId,  FKUpdatedByUserId = s.FKUpdatedByUserId,  CreatedByUser = u.UserName,  UpdatedByUser = u.UserName,  ImageName = s.ImageName,  ImagePath = s.ImagePath,  IsActive = s.IsActive  }).AsEnumerable().Select(x => new SubCategory  {  PKSubCategoryId = x.PKSubCategoryId,  CategoryName = x.CategoryName,  FKCategoryId = x.FKCategoryId,  SubCategoryName = x.SubCategoryName,  CreatedDate = x.CreatedDate,  UpdatedDate = x.UpdatedDate,  FKCreatedByUserId = x.FKCreatedByUserId,  FKUpdatedByUserId = x.FKUpdatedByUserId,  CreatedByUser = x.CreatedByUser,  UpdatedByUser = x.UpdatedByUser,  ImageName = x.ImageName,  ImagePath = x.ImagePath,  IsActive = x.IsActive  });  return q.ToList();  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertSubCategory(SubCategory objSubCategory)  {  try  {  objSubCategory.CreatedDate = DateTime.Now;  objSubCategory.FKCreatedByUserId = Helper.UserId;  context.SubCategories.Add(objSubCategory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateSubCategory(SubCategory objSubCategory)  {  try  {  objSubCategory.UpdatedDate = DateTime.Now;  objSubCategory.FKUpdatedByUserId = Helper.UserId;  context.Entry(objSubCategory).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteSubCategory(int subCategoryId)  {  try  {  SubCategory objSubCategory = context.SubCategories.Find(subCategoryId);  context.SubCategories.Remove(objSubCategory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select Data Folder->Under the Data Folder->Right Click on Folder->Add

->Class->Class name=Product Data->Click OK and Change the namespace of class OnlineShoppingStoreMVCUsingLinq.Data To OnlineShoppingStoreMVCUsingLinq (Remove .Data)& Make the class as Partial & Change the code as below

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public partial class Product  {  public string CreatedByUser  {  get;  set;  }  public string UpdatedByUser  {  get;  set;  }  public string CategoryName  {  get;  set;  }  public string SubCategoryName  {  get;  set;  }  public string ImageName  {  get;  set;  }  public string ImagePath  {  get;  set;  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name=ProductBO-.Click ->OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class ProductBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public Product GetProduct(int productId)  {  return context.Products.Where(p => p.PKProductId == productId).SingleOrDefault();  }  public List<Product> GetProducts(int categoryId = 0, int subCategoryId = 0, bool? isActive = null)  {  try  {  IQueryable<Product> qry = context.Products;  if (categoryId != 0)  {  qry = context.Products.Where(p => p.FKCategoryId == categoryId);  }  if (subCategoryId != 0)  {  qry = context.Products.Where(p => p.FKSubCategoryId == subCategoryId);  }  if (isActive != null)  {  qry = context.Products.Where(p => p.IsActive == isActive);  }  var q = (from p in qry  join u in context.UserProfiles on p.FKCreatedByUserId equals u.PKUserId  join c in context.Categories on p.FKCategoryId equals c.PKCategoryId  join s in context.SubCategories on p.FKSubCategoryId equals s.PKSubCategoryId  join pi in context.ProductImages on p.PKProductId equals pi.FKProductId  select new  {  PKProductId = p.PKProductId,  FKCategoryId = p.FKCategoryId,  FKSubCategoryId = p.FKSubCategoryId,  CategoryName = c.CategoryName,  SubCategoryName = s.SubCategoryName,  ProductName = p.ProductName,  ImageName = pi.ImageName,  ImagePath = pi.ImagePath,  Description = p.Description,  Quantity = p.Quantity,  Price = p.Price,  CreatedDate = p.CreatedDate,  UpdatedDate = p.UpdatedDate,  CreatedByUser = u.UserName,  UpdatedByUser = u.UserName,  IsActive = p.IsActive  }).AsEnumerable().Select(x => new Product  {  PKProductId = x.PKProductId,  FKCategoryId = x.FKCategoryId,  FKSubCategoryId = x.FKSubCategoryId,  CategoryName = x.CategoryName,  SubCategoryName = x.SubCategoryName,  ProductName = x.ProductName,  ImageName = x.ImageName,  ImagePath = x.ImagePath,  Description = x.Description,  Quantity = x.Quantity,  Price = x.Price,  CreatedDate = x.CreatedDate,  UpdatedDate = x.UpdatedDate,  CreatedByUser = x.CreatedByUser,  UpdatedByUser = x.UpdatedByUser,  IsActive = x.IsActive  });  return q.ToList();  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertProduct(Product objProduct)  {  try  {  objProduct.CreatedDate = DateTime.Now;  objProduct.FKCreatedByUserId = Helper.UserId;  context.Products.Add(objProduct);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateProduct(Product objProduct)  {  try  {  objProduct.UpdatedDate = DateTime.Now;  objProduct.FKUpdatedByUserId = Helper.UserId;  context.Entry(objProduct).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteProduct(int productId)  {  try  {  Product objProduct = context.Products.Find(productId);  context.Products.Remove(objProduct);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name=ProductImageBO-> Click ->OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class ProductImageBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public ProductImage GetImage(int imageId)  {  return context.ProductImages.Where(i => i.PKImageId == imageId).SingleOrDefault();  }  public List<ProductImage> GetImages(int productId = 0)  {  try  {  IQueryable<ProductImage> qry = context.ProductImages;  if (productId != 0)  qry = qry.Where(x => x.FKProductId == productId);  return qry.ToList();  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertImage(ProductImage objImage)  {  try  {  context.ProductImages.Add(objImage);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateImage(ProductImage objImage)  {  try  {  context.Entry(objImage).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteImage(int imageId)  {  try  {  ProductImage objImage = context.ProductImages.Find(imageId);  context.ProductImages.Remove(objImage);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select Data Folder->Under the Data Folder->Right Click on Folder->Add

->Class->Class name=OrderData ->Click OK and Change the namespace of class OnlineShoppingStoreMVCUsingLinq.Data To OnlineShoppingStoreMVCUsingLinq (Remove .Data)& Make the class as Partial & Change the code as below

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public partial class Order  {  public string UserName  {  get;  set;  }  public int FKProductId  {  get;  set;  }  public int Quantity  {  get;  set;  }  public decimal Cost  {  get;  set;  }  public int OrderId  {  get;  set;  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name=OrderBO->Click OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class OrderBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public Order GetOrder(int orderId)  {  return context.Orders.Where(o => o.PKOrderId == orderId).SingleOrDefault();  }  public List<Order> GetOrders(DateTime? fromDate = null, DateTime? toDate = null, string userName = "", string status = "")  try  {  {  IQueryable<Order> qry = context.Orders;  if (status != "" && status != "All")  qry = qry.Where(order => order.Status == status);  if (fromDate.HasValue && toDate.HasValue)  qry = qry.Where(order => order.OrderDate > fromDate.Value && order.OrderDate < toDate.Value);  var q = (from o in qry  join u in context.UserProfiles on o.FKCustomerUserId equals u.PKUserId  join od in context.OrderDetails on o.PKOrderId equals od.FKOrderId  where u.UserName.Contains(userName == "" ? u.UserName : userName)  select new  {  PKOrderId = o.PKOrderId,  FKCustomerUserId = o.FKCustomerUserId,  FKBillingAddressId = o.FKBillingAddressId,  OrderDate = o.OrderDate,  DeliveredDate = o.DeliveredDate,  FKProcessByUserId = o.FKProcessByUserId,  CustomerComment = o.CustomerComment,  AdminComment = o.AdminComment,  Status = o.Status,  UserName = u.UserName,  Cost = od.Cost,  Quantity = od.Quantity,  FKProductId = od.FKProductId  }).AsEnumerable().Select(x => new Order  {  PKOrderId = x.PKOrderId,  FKCustomerUserId = x.FKCustomerUserId,  FKBillingAddressId = x.FKBillingAddressId,  OrderDate = x.OrderDate,  DeliveredDate = x.DeliveredDate,  FKProcessByUserId = x.FKProcessByUserId,  CustomerComment = x.CustomerComment,  AdminComment = x.AdminComment,  Status = x.Status,  UserName = x.UserName,  Cost = x.Cost,  Quantity = x.Quantity,  FKProductId = x.FKProductId  });  return q.ToList();  catch (Exception ex)  {  throw ex;  }  }  public void InsertOrder(Order objOrder)  {  try  {  objOrder.FKCustomerUserId = Helper.UserId;  objOrder.FKProcessByUserId = Helper.UserId;  objOrder.OrderDate = DateTime.Now;  context.Orders.Add(objOrder);  context.SaveChanges();  catch (Exception ex)  {  throw ex;  }  }  public void UpdateOrder(Order objOrder)  {  try  {  context.Entry(objOrder).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteOrder(int orderId)  {  try  {  Order objOrder = context.Orders.Find(orderId);  context.Orders.Remove(objOrder);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name=OrderDetailsBO->Click OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class OrderDetailBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public OrderDetail GetOrderDetail(int orderDetailId)  {  return context.OrderDetails.Where(o => o.PKOrderDetailId == orderDetailId).SingleOrDefault();  }  public List<OrderDetail> GetOrderDetails(int orderId = 0)  {  try  {  context.Configuration.LazyLoadingEnabled = false;  IQueryable<OrderDetail> qry = context.OrderDetails;  if (orderId > 0)  {  qry = qry.Where(ordDetails => ordDetails.FKOrderId == orderId);  }  return qry.ToList();  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertOrderDetail(OrderDetail objOrderDetail)  {  try  {  context.OrderDetails.Add(objOrderDetail);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateOrderDetail(OrderDetail objOrder)  {  try  {  context.Entry(objOrder).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteOrderDetail(int orderId)  {  try  {  OrderDetail objOrder = context.OrderDetails.Find(orderId);  context.OrderDetails.Remove(objOrder);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select Data Folder->Under the Data Folder->Right Click on Folder->Add

->Class->Class name=OrderHistoryData-> Click OK and Change the namespace of class OnlineShoppingStoreMVCUsingLinq.Data To OnlineShoppingStoreMVCUsingLinq (Remove .Data), Make the class as Partial & Change the code as below

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public partial class OrderHistory  {  public int CustomerUserId  {  get;  set;  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name=OrderHistoryBO->Click OK

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class OrderHistoryBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public OrderHistory GetOrderHistory(int orderId)  {  return context.OrderHistories.Where(o => o.FKOrderId == orderId).SingleOrDefault();  }  public List<OrderHistory> GetOrderHistories(int userId = 0)  {  IQueryable<OrderHistory> qry = context.OrderHistories;  var q = (from oh in qry  join o in context.Orders on oh.FKOrderId equals o.PKOrderId  where o.FKCustomerUserId == (userId == 0 ? o.FKCustomerUserId : userId)  select new  {  FKOrderId = oh.FKOrderId,  Description = oh.Description,  Status = oh.Status,  DateChanged = oh.DateChanged,  CustomerUserId = o.FKCustomerUserId  }).AsEnumerable().Select(x => new OrderHistory  {  FKOrderId = x.FKOrderId,  Description = x.Description,  Status = x.Status,  DateChanged = x.DateChanged,  CustomerUserId = x.CustomerUserId  });  return q.ToList();  }  public void InsertOrderHistory(OrderHistory objOrderHistory)  {  objOrderHistory.DateChanged = DateTime.Now;  context.OrderHistories.Add(objOrderHistory);  context.SaveChanges();  }  public void UpdateOrderHistory(OrderHistory objOrderHistory)  {  context.Entry(objOrderHistory).State = EntityState.Modified;  context.SaveChanges();  }  public void DeleteOrderHistory(int orderHistoryId)  {  OrderHistory objOrderHistory = context.OrderHistories.Find(orderHistoryId);  context.OrderHistories.Remove(objOrderHistory);  context.SaveChanges();  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name =BillingAddressBO

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class BillingAddressBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public BillingAddress GetBillingAddress(int addressId)  {  return context.BillingAddresses.Where(b => b.PKAddressId == addressId).SingleOrDefault();  }  public List<BillingAddress> GetBillingAddresses()  {  try  {  return context.BillingAddresses.ToList();  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertBillingAddress(BillingAddress objBillngAddress)  {  try  {  objBillngAddress.FKUserId = Helper.UserId;  context.BillingAddresses.Add(objBillngAddress);  context.SaveChanges();  }  catch(Exception ex)  {  throw ex;  }  }  public void UpdateBillingAddress(BillingAddress objBillngAddress)  {  try  {  context.Entry(objBillngAddress).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteBillingAddress(int addressId)  {  try  {  BillingAddress objBillngAddress = context.BillingAddresses.Find(addressId);  context.BillingAddresses.Remove(objBillngAddress);  context.SaveChanges();  }  catch(Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

->Class->Class name =UserLoginHistoryBO

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class UserLoginHistoryBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public UserLoginHistory GetUserLoginHistory(int userId)  {  return context.UserLoginHistories.Where(u => u.FKUserId == userId).SingleOrDefault();  }  public List<UserLoginHistory> GetUserLoginHistories(int userId = 0)  {  try  {  IQueryable<UserLoginHistory> qry = context.UserLoginHistories;  if (userId != 0)  {  return context.UserLoginHistories.Where(u => u.FKUserId == userId).ToList();  }  else  {  return context.UserLoginHistories.ToList();  }  }  catch (Exception ex)  {  throw ex;  }  }  public void InsertUserLoginHistory(UserLoginHistory objUserLoginHistory)  {  try  {  objUserLoginHistory.LoginDateTime = DateTime.Now;  context.UserLoginHistories.Add(objUserLoginHistory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateUserLoginHistory(UserLoginHistory objUserLoginHistory)  {  try  {  context.Entry(objUserLoginHistory).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteUserLoginHistory(int userLoginHistoryId)  {  try  {  UserLoginHistory objUserLoginHistory = context.UserLoginHistories.Find(userLoginHistoryId);  context.UserLoginHistories.Remove(objUserLoginHistory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. Go to Solution Explore->Project->Select BO Folder->Under the BO Folder->Right Click on Folder->Add

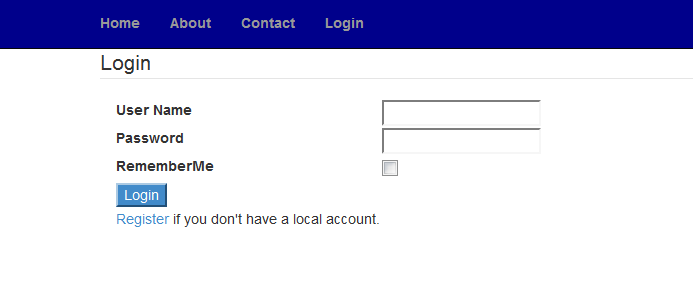
->Class->Class name =ErrorHistoryBO

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.Entity;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.BO  {  public class ErrorHistoryBO  {  OnlineShoppingStoreEntities context = new OnlineShoppingStoreEntities();  public ErrorHistory GetErrorHistory(int errorHistoryId)  {  return context.ErrorHistories.Where(e => e.PKErrorId == errorHistoryId).SingleOrDefault();  }  public List<ErrorHistory> GetErrorHistories()  {  try  {  return context.ErrorHistories.ToList();  }  catch(Exception ex)  {  throw ex;  }  }  public void InsertErrorHistory(ErrorHistory objErrorHistory)  {  try  {  objErrorHistory.ErrorLogTime = DateTime.Now;  if (!string.IsNullOrEmpty(Helper.UserData))  objErrorHistory.FKUserId = Helper.UserId;  context.ErrorHistories.Add(objErrorHistory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void UpdateErrorHistory(ErrorHistory objErrorHistory)  {  try  {  context.Entry(objErrorHistory).State = EntityState.Modified;  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  public void DeleteErrorHistory(int errorHistoryId)  {  try  {  ErrorHistory objErrorHistory = context.ErrorHistories.Find(errorHistoryId);  context.ErrorHistories.Remove(objErrorHistory);  context.SaveChanges();  }  catch (Exception ex)  {  throw ex;  }  }  }  } |

1. GoTo Solution Explorer->Project->Views->Shared Folder -> Under Shared folder ->Check the \_Layout.cshtml and Change the code as below or if \_layout.cshtml not available-> Under Shared folder ->Add-> MVC5 LayoutPage->name=\_Layout.cshtml->Click Ok -> Change the code as below

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>@ViewBag.Title - My ASP.NET Application</title>  @Styles.Render("~/Content/css")  @Scripts.Render("~/bundles/modernizr")  </head>  <body>  <div class="navbar navbar-inverse navbar-fixed-top" style="background-color:darkblue">  <div class="container">  <div class="navbar-collapse collapse" style="font-weight:bolder">  <ul class=" nav navbar-nav">  <li>@Html.ActionLink("Home", "Index", "Home")</li>  <li>@Html.ActionLink("About", "About", "Home")</li>  <li>@Html.ActionLink("Contact Us", "Contact", "Home")</li>  <li>@Html.ActionLink("Login", "Login", "Account")</li>  </ul>  </div>  </div>  </div>  <div class="container body-content">  <div class="col-sm-10">  @RenderBody()  </div>  <hr />  </div>  @Scripts.Render("~/bundles/jquery")  @Scripts.Render("~/bundles/bootstrap")  @RenderSection("scripts", required: false)  </body>  </html> |

1. Login:-



1. Create Account Controller:

GoTo- Solution Explorer->Project->Select Controller Folder->Right Click on Controllers->Add-> Controller->Select Template As EmptyMVC Controller ->Click Add-> Controller name= AccountController->Click Add

=>Login ,Register,IsUserNameValid and SignOut Methods In This Controller.

|  |
| --- |
| using Microsoft.Ajax.Utilities;  using OnlineShoppingStoreMVCUsingLinq.BO;  using OnlineShoppingStoreMVCUsingLinq.Models;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using System.Web.Security;  namespace OnlineShoppingStoreMVCUsingLinq.Controllers  {  public class AccountController : Controller  {  UserProfileBO objUserProfileBO = new UserProfileBO();  UserLoginHistory objUserLoginHistory = new UserLoginHistory();  UserLoginHistoryBO objUserLoginHistoryBO = new UserLoginHistoryBO();  //  // GET: /Account/  public ActionResult Index()  {  return View();  }  public ActionResult Login(string returnurl)  {  ViewBag.ReturnUrl = returnurl;  return View();  }  [HttpPost]  public ActionResult Login(LoginViewModel model,string returnurl)  {  if(ModelState.IsValid)  {  UserProfile objUserProfile = objUserProfileBO.AuthenticateUser(model.UserName);  if(objUserProfile!=null)  {  if(model.Password==objUserProfile.Password)  {  string userData = objUserProfile.PKUserId + "^" + objUserProfile.FKRoleId + "^" + objUserProfile.Role.RoleName + "^" + objUserProfile.UserName + "^" + objUserProfile.FirstName + "^" + objUserProfile.LastName + "^" + objUserProfile.EmailId + "^" + objUserProfile.Password + "^" + objUserProfile.AddressLine1 + "^" + objUserProfile.AddressLine2 + "^" + objUserProfile.CreatedDate + "^" + objUserProfile.UpdatedDate;  System.Web.HttpContext.Current.User = new System.Security.Principal.GenericPrincipal(new System.Security.Principal.GenericIdentity(userData), new string[] { userData });  FormsAuthentication.RedirectFromLoginPage(Helper.UserData, model.RememberMe);  objUserLoginHistory.LoginDateTime = DateTime.Now;  objUserLoginHistory.FKUserId = Helper.UserId;  string hostName = Dns.GetHostName();  string clientIpAddress = Dns.GetHostAddresses(hostName).GetValue(1).ToString();  objUserLoginHistory.IPAddress = clientIpAddress;  objUserLoginHistoryBO.InsertUserLoginHistory(objUserLoginHistory);  if(objUserProfile.Role.RoleName=="Admin")  {  return RedirectToAction("Index", "Category", new { area = "Admin" });  }  else  {  if(returnurl!=null)  {  return Redirect(returnurl);  }  else  {  return RedirectToAction("Index", "Home");  }  }  }  else  {  ModelState.AddModelError("", "Invalid Password");  return View();  }  }  else  {  ModelState.AddModelError("", "User Doesnot Exists");  return View();  }  }  return View();  }  public ActionResult Register()  {  return View();  }  [HttpPost]  public ActionResult Register(UserProfile objUser)  {  if(ModelState.IsValid)  {  objUser.FKRoleId = 2;  objUserProfileBO.InsertUser(objUser);  return RedirectToAction("Login");  }  return View(User);  }  public JsonResult IsUserNameValid(string userName)  {  if (objUserProfileBO.AuthenticateUser(userName) != null)  {  return Json("Sorry! UserName Already Exist", JsonRequestBehavior.AllowGet);  }  return Json(true, JsonRequestBehavior.AllowGet);  }  public ActionResult Signout()  {  FormsAuthentication.SignOut();  return RedirectToAction("Login", "Account");  }  }  } |

1. Create Models For Login And Register Methods:

GoTo Solution Explorer->Project->Select Models Folder->Under Models Folder->Right Click On Model->Add->Class->Class name= LoginViewModel->Click Ok& Type Below Code.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel.DataAnnotations;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq.Models  {  public class LoginViewModel  {  [Required(ErrorMessage = "Please Enter User Name")]  [Display(Name="User Name")]  public string UserName  {  get;  set;  }  [Required(ErrorMessage = "Please Enter Password")]  [DataType(DataType.Password)]  public string Password  {  get;  set;  }  public bool RememberMe  {  get;  set;  }  }  } |

1. Create RegisterModel Follow Same as above steps

GoTo Solution Explorer->Project->Select Models Folder->Under Models Folder->Right Click On Model->Add->Class->Class name=RegisterModel and the change the name space of the class OnlineShoppingStoreMVCUsingLinq.Models To OnlineShoppingStoreMVCUsingLinq

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel.DataAnnotations;  using System.Linq;  using System.Web;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVCUsingLinq  {  public class RegisterModel  {  [Required]  [Remote("IsUserNameValid","Account")]  [Display(Name="User Name")]  public string UserName { get; set; }  [Required(ErrorMessage="Please enter first name")]  [Display(Name = "First Name")]  public string FirstName { get; set; }  [Display(Name = "Last Name")]  public string LastName { get; set; }  [Required(ErrorMessage = "Please enter emailId")]  [DataType(DataType.EmailAddress)]  public string EmailId { get; set; }  [Required(ErrorMessage="Please enter password")]  [DataType(DataType.Password)]  public string Password { get; set; }  [Required(ErrorMessage="Please enter mobileNo")]  [DataType(DataType.PhoneNumber)]  public string MobileNo { get; set; }  [Required(ErrorMessage = "Please enter addressLine1")]  [Display(Name = "Address Line1")]  public string AddressLine1 { get; set; }  [Display(Name = "Address Line2")]  public string AddressLine2 { get; set; }  [Required]  public string City { get; set; }  [Required]  public string State { get; set; }  [Required]  public string Country { get; set; }  [DataType(DataType.DateTime)]  public System.DateTime CreatedDate { get; set; }  [DataType(DataType.DateTime)]  public Nullable<System.DateTime> UpdatedDate { get; set; }  }  [MetadataType(typeof(RegisterModel))]  public partial class UserProfile  { }  } |

1. Create Views For AccountController Methods

GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Account Controller->Double click on it->Login Method In Account Controller->Right Click On Login Method->Add View->Check the checkbox of create a strongly typed view ->Select Model Class As LoginViewModel-> Select Scaffold template as Create->Click Add.

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Models.LoginViewModel  @{  ViewBag.Title = "Login";  }  @using (Html.BeginForm("Login", "Account", new { ReturnUrl = ViewBag.ReturnUrl }, FormMethod.Post))  {<div class="container">  <fieldset>  <legend>Login</legend>  @Html.AntiForgeryToken()  @Html.ValidationSummary(true)  <div class="col-sm-10">  <table>  <tr>  <td>  @Html.LabelFor(model => model.UserName)  </td>  <td>@Html.EditorFor(model => model.UserName)</td>  </tr>  <tr>  <td>  @Html.LabelFor(model => model.Password)  </td>  <td>@Html.EditorFor(model => model.Password)</td>  </tr>  <tr>  <td>  @Html.LabelFor(model => model.RememberMe)  </td>  <td>  @Html.EditorFor(model => model.RememberMe)  </td>  </tr>  <tr>  <td>  <input type="submit" id="btnLogin" value="Login" class="btn-primary" />  </td>  </tr>  <tr>  <td>  <p>@Html.ActionLink("Register", "Register", "Account") if you don't have a local account.</p>  </td>  </tr>  </table>  </div>  </fieldset>  </div>  } |

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Account Controller->Double click on it->Register Method In Account Controller->Right Click On Register Method->Add View->Check the checkbox of create a strongly typed view ->Select Model Class As UserProfileModel-> Select Scaffold template as Create->Click Add.

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.UserProfile  @{  ViewBag.Title = "Register";  }  <h2>Register</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>UserProfile</h4>  <hr />  @Html.ValidationSummary(true)  <div class="form-group">  @Html.LabelFor(model => model.UserName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.UserName)  @Html.ValidationMessageFor(model => model.UserName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.FirstName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.FirstName)  @Html.ValidationMessageFor(model => model.FirstName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.LastName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.LastName)  @Html.ValidationMessageFor(model => model.LastName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.EmailId, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.EmailId)  @Html.ValidationMessageFor(model => model.EmailId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Password, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Password)  @Html.ValidationMessageFor(model => model.Password)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.MobileNo, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.MobileNo)  @Html.ValidationMessageFor(model => model.MobileNo)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AddressLine1, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AddressLine1)  @Html.ValidationMessageFor(model => model.AddressLine1)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AddressLine2, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AddressLine2)  @Html.ValidationMessageFor(model => model.AddressLine2)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.City)  @Html.ValidationMessageFor(model => model.City)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.State, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.State)  @Html.ValidationMessageFor(model => model.State)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Country, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Country)  @Html.ValidationMessageFor(model => model.Country)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ZipCode, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.ZipCode)  @Html.ValidationMessageFor(model => model.ZipCode)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Create" class="btn btn-primary" />  </div>  </div>  </div>  } |

1. GoTo Solution Explorer->Project->Select Web.config ->Add the following code in Web.Config under System.web:

|  |
| --- |
| <authentication mode="Forms"> |
| <forms name="\_\_UserLoginCookie" loginUrl="/Account/Login" defaultUrl="/Account/Login"></forms> |
| </authentication> |

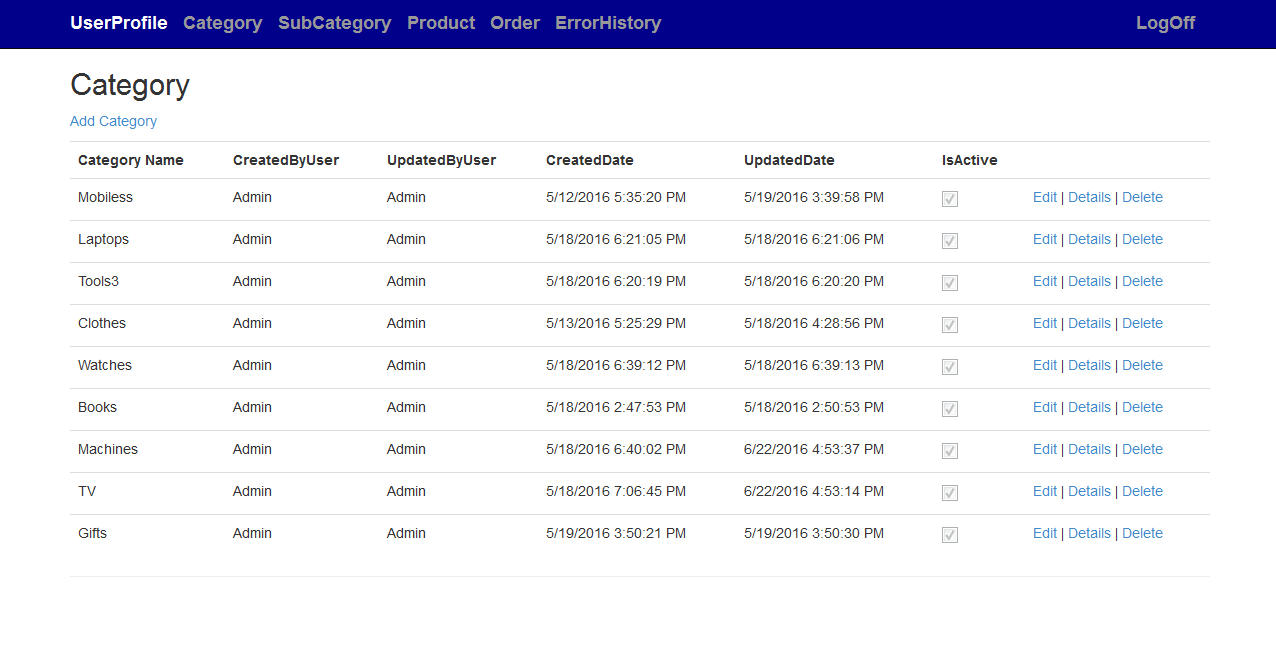
Admin Area:

14. Goto Solution Explorer->Project->Right click on project → Add → Areas→ Name=Admin→ OK

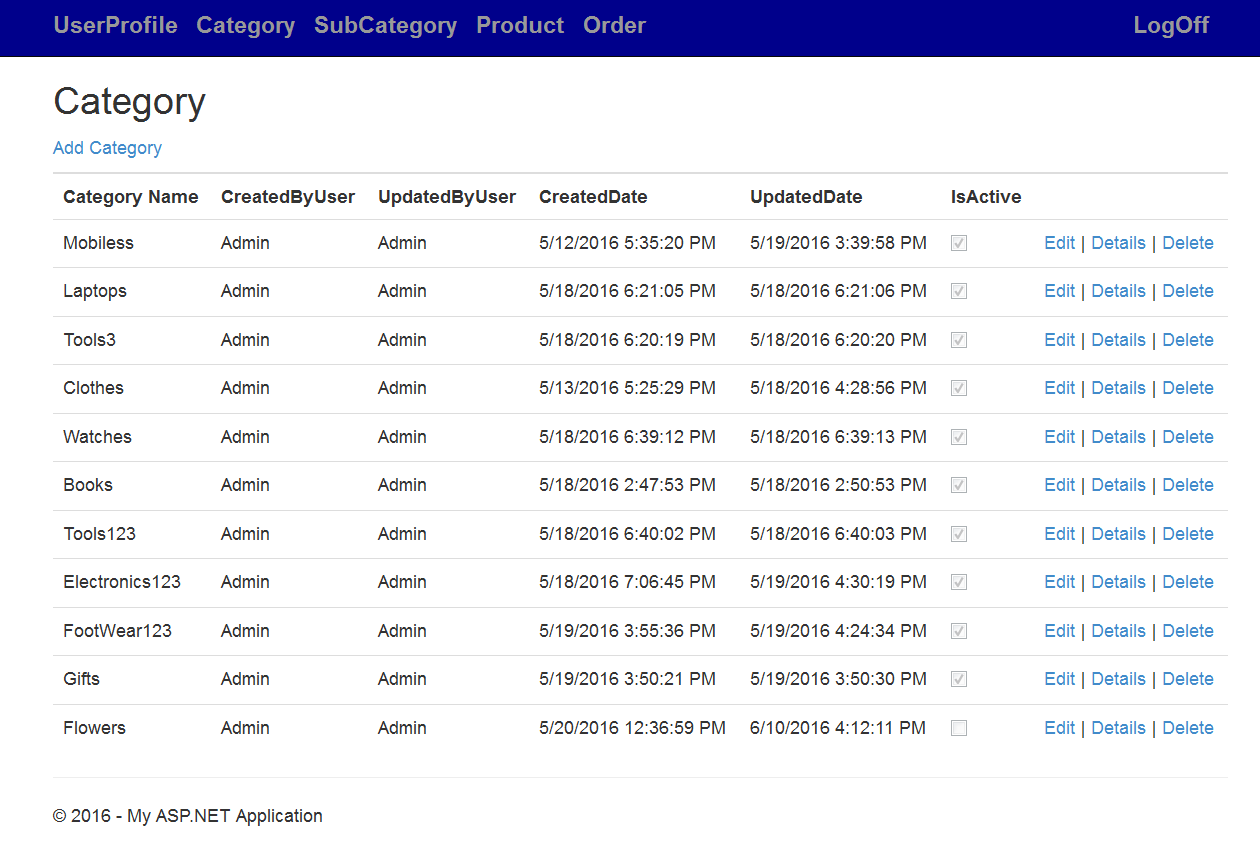
Under the Areas->Admin->Views->Shared->Right Click on Folder->Add ->MVC5 Layout->nma=\_Layout.cshtml-> Click Ok and write the code as below.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>@ViewBag.Title - OnlineShoppingStore</title>  @Styles.Render("~/Content/css")  @Scripts.Render("~/bundles/modernizr")  </head>  <body style="font-family:sans-serif">  <div class="navbar navbar-inverse navbar-fixed-top" style="background-color:darkblue">  <div class="container">  <div class="navbar-header" style="font-weight:bolder;">  @Html.ActionLink("UserProfile", "Index", "UserProfile", null, new { @class = "navbar-brand" })  @Html.ActionLink("Category", "Index", "Category", null, new { @class = "navbar-brand" })  @Html.ActionLink("SubCategory", "Index", "SubCategory", null, new { @class = "navbar-brand" })  @Html.ActionLink("Product", "Index", "Product", null, new { @class = "navbar-brand" })  @Html.ActionLink("Order", "Index", "Order", null, new { @class = "navbar-brand" })  @Html.ActionLink("ErrorHistory", "Index","ErrorHistory", null, new { @class = "navbar-brand" })  </div>  <div style="float:right; font-weight:900">  @Html.ActionLink("LogOff", "Signout", "Account", new { area = "" }, new { @class = "navbar-brand" })  </div>  </div>  </div>  <div class="container body-content" >  @RenderBody()  <hr />  </div>  @Scripts.Render("~/bundles/jquery")  @Scripts.Render("~/bundles/bootstrap")  @RenderSection("scripts", required: false)  </body>  </html> |

1. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=CategoryController -> Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=Category (OnlineShoppingStoreMVCUsingLinq) → DataContextClass=OnlineShoppingStoreEntities (OnlineShoppingStoreMVCUsingLinq) ->Click Add->→ Click Add.



|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  [Authorize]  public class CategoryController : Controller  {  CategoryBO objCategoryBO = new CategoryBO();  // GET: /Admin/Category/  public ActionResult Index()  {  return View(objCategoryBO.GetCategories());  }  // GET: /Admin/Category/Details/5  public ActionResult Details(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Category category = objCategoryBO.GetCategory(id.Value);  if (category == null)  {  return HttpNotFound();  }  return View(category);  }  // GET: /Admin/Category/Create  public ActionResult Create()  {  return View();  }  // POST: /Admin/Category/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(Category category)  {  if (ModelState.IsValid)  {  objCategoryBO.InsertCategory(category);  return RedirectToAction("Index");  }  return View(category);  }  // GET: /Admin/Category/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Category category = objCategoryBO.GetCategory(id.Value);  if (category == null)  {  return HttpNotFound();  }  return View(category);  }  // POST: /Admin/Category/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Edit(Category category)  {  if (ModelState.IsValid)  {  objCategoryBO.UpdateCategory(category);  return RedirectToAction("Index");  }  return View(category);  }  // GET: /Admin/Category/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Category category = objCategoryBO.GetCategory(id.Value);  if (category == null)  {  return HttpNotFound();  }  return View(category);  }  // POST: /Admin/Category/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  objCategoryBO.DeleteCategory(id);  return RedirectToAction("Index");  }  }  } |

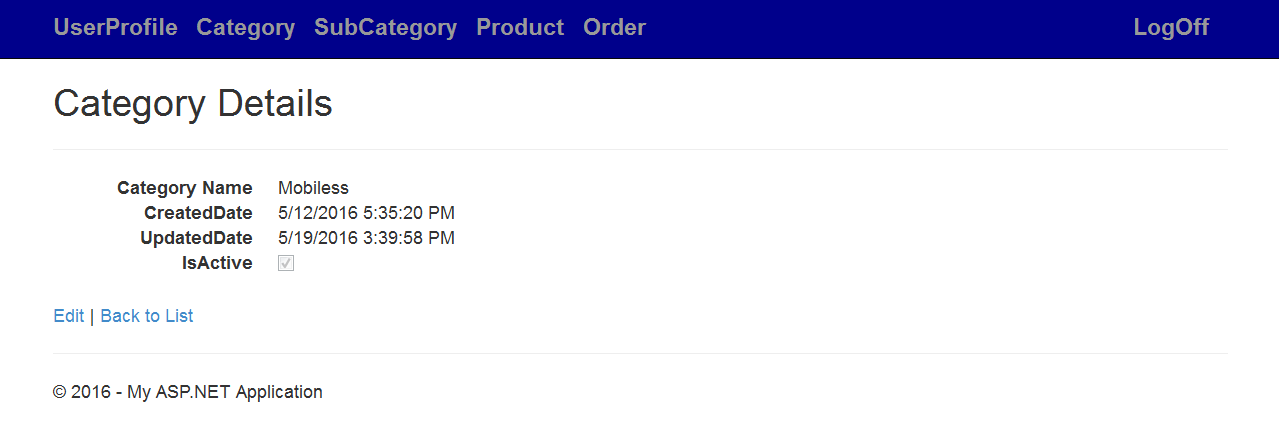


1. Go to Solution Explorer->Project->Areas->Models->Under Models->Right Click on Folder->Add->Class->Class name=CategoryVal->click OK and Change the name space of class OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Models TO OnlineShoppingStoreMVCUsingLinq (remove .Areas.Admin.Models)

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel.DataAnnotations;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public class CategoryVal  {  [Required(ErrorMessage="Please enter category name")]  [Display(Name="Category Name")]  public string CategoryName { get; set; }  [Display(Name="Active")]  public bool IsActive { get; set; }  }  [MetadataType(typeof(CategoryVal))]  public partial class Category  { }  } |

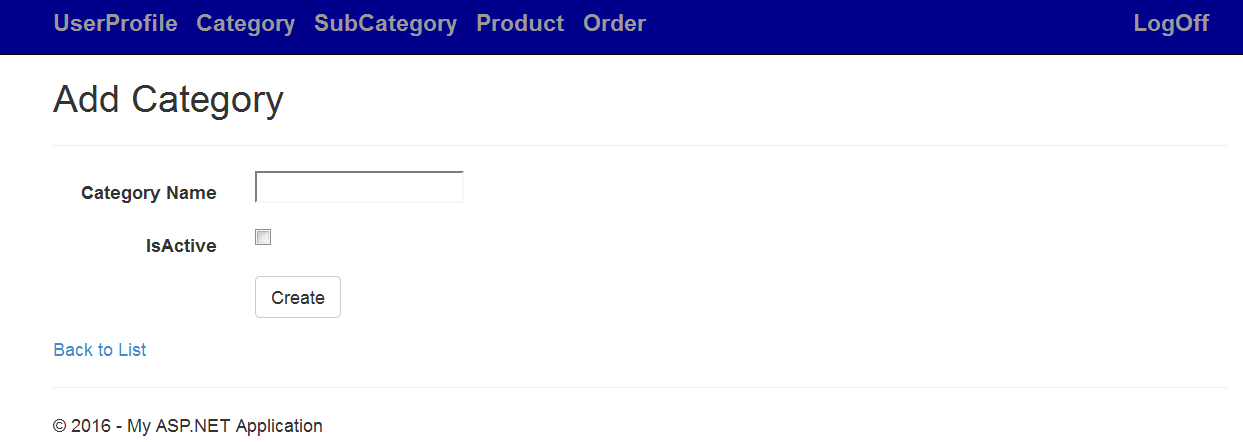
1. Go to Solution Explorer->Project->Areas->Admin->Views->Category->In Category,Select Index.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.Category>  @{  ViewBag.Title = "Index";  }  <h2>Category</h2>  <p>  @Html.ActionLink("Add Category", "Create")  </p>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.CategoryName)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedByUser)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedByUser)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.IsActive)  </th>    <th></th>  </tr>  @foreach (var item in Model) {  <tr>  <td>  @Html.DisplayFor(modelItem => item.CategoryName)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedByUser)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedByUser)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.IsActive)  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id=item.PKCategoryId }) |  @Html.ActionLink("Details", "Details", new { id=item.PKCategoryId }) |  @Html.ActionLink("Delete", "Delete", new { id=item.PKCategoryId })  </td>  </tr>  }  </table> |



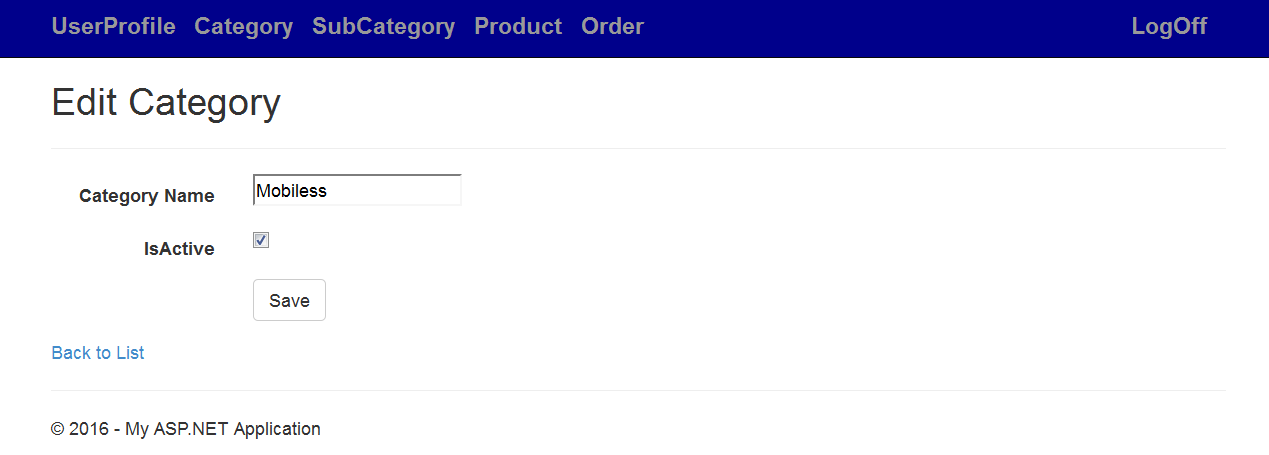
1. Go to Solution Explorer->Project->Areas->Admin->Views->Category->In Category,Select Details.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Category  @{  ViewBag.Title = "Details";  }  <h2> Category Details</h2>  <div>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.CategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.CategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  </dl>  </div>  <p>  @Html.ActionLink("Edit", "Edit", new { id = Model.PKCategoryId }) |  @Html.ActionLink("Back to List", "Index")  </p> |



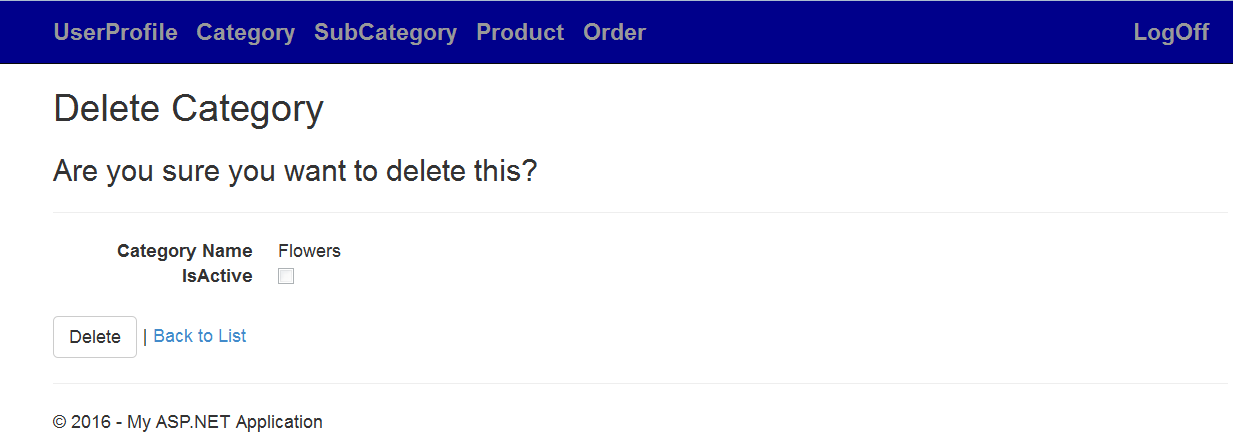
1. Go to Solution Explorer->Project->Areas->Admin->Views->Category->In Category,Select Create.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Category  @{  ViewBag.Title = "Create";  }  <h2>Add Category</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <hr />  @Html.ValidationSummary(true)  <div class="form-group">  @Html.LabelFor(model => model.CategoryName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.CategoryName)  @Html.ValidationMessageFor(model => model.CategoryName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Create" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |



1. Go to Solution Explorer->Project->Areas->Admin->Views->Category->In Category,Select Edit.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Category  @{  ViewBag.Title = "Edit";  }  <h2>Edit Category</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKCategoryId)  @Html.HiddenFor(model=>model.CreatedDate)  @Html.HiddenFor(model=>model.FKCreatedByUserId)  <div class="form-group">  @Html.LabelFor(model => model.CategoryName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.CategoryName)  @Html.ValidationMessageFor(model => model.CategoryName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |



1. Go to Solution Explorer->Project->Areas->Admin->Views->Category->In Category,Select Delete.cshtml->Change the code as below

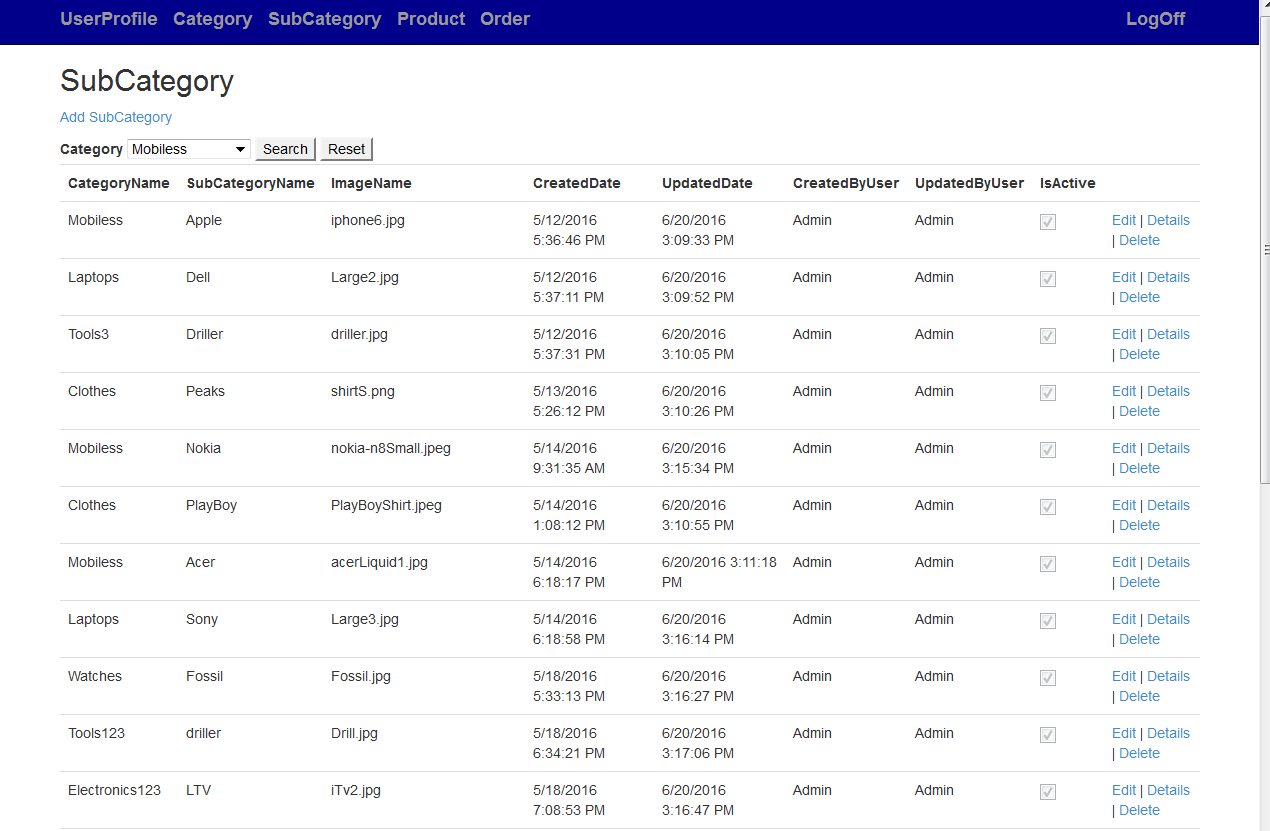
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Category  @{  ViewBag.Title = "Delete";  }  <h2>Delete Category</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.CategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.CategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index")  </div>  }  </div> |

1. Go to Solution Explorer->Project->RightClick on Project -> Add ->New Folder->Folder Names=Images->Click OK(To store Subcategory,Product Images)
2. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=SubCategoryController→ Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=SubCategory(OnlineShoppingStoreMVCUsingLinq)→ DataContextClass=OnlineShoppingStoreEntities (OnlineShoppingStoreMVCUsingLinq) ->Click Add.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  using System.Web.UI.WebControls;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  [Authorize]  public class SubCategoryController : Controller  {  SubCategoryBO objSubCategoryBO = new SubCategoryBO();  CategoryBO objCartegoryBO = new CategoryBO();  // GET: /Admin/SubCategory/  public ActionResult Index()  {  ViewBag.CategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  return View(objSubCategoryBO.GetSubCategories());  }  [HttpPost]  public ActionResult Index(FormCollection fc)  {  if (fc["btnSearch"] == "Search"&&fc["Category"]!="")  {  int categoryId = Convert.ToInt32(fc["Category"]);  ViewBag.CategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName", categoryId);  return View(objSubCategoryBO.GetSubCategories(categoryId, true));  }  else  {  ViewBag.CategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  return View(objSubCategoryBO.GetSubCategories());  }  }  // GET: /Admin/SubCategory/Details/5  public ActionResult Details(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  SubCategory subcategory = objSubCategoryBO.GetsubCategory(id.Value);  if (subcategory == null)  {  return HttpNotFound();  }  return View(subcategory);  }  // GET: /Admin/SubCategory/Create  public ActionResult Create()  {  ViewBag.FKCategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  return View();  }  // POST: /Admin/SubCategory/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(SubCategory subcategory, HttpPostedFileBase Images, FormCollection fc)  {  if (ModelState.IsValid)  {  subcategory.FKCategoryId = Convert.ToInt32(fc["CategoryName"]);  string imageName = System.IO.Path.GetFileName(Images.FileName);  Images.SaveAs(Server.MapPath("~/Images/" + imageName));  subcategory.ImageName = imageName;  subcategory.ImagePath = "/Images/" + imageName;  objSubCategoryBO.InsertSubCategory(subcategory);  return RedirectToAction("Index");  }  ViewBag.FKCategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName", subcategory.FKCategoryId);  return View(subcategory);  }  // GET: /Admin/SubCategory/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  SubCategory subcategory = objSubCategoryBO.GetsubCategory(id.Value);  if (subcategory == null)  {  return HttpNotFound();  }  ViewBag.FKCategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName", subcategory.FKCategoryId);  return View(subcategory);  }  // POST: /Admin/SubCategory/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Edit(SubCategory subcategory, HttpPostedFileBase txtFile)  {  if (ModelState.IsValid)  {  if (txtFile != null)  {  string imageName = System.IO.Path.GetFileName(txtFile.FileName);  txtFile.SaveAs(Server.MapPath("~/Images/" + imageName));  subcategory.ImageName = imageName;  subcategory.ImagePath = "/Images/" + imageName;  }  objSubCategoryBO.UpdateSubCategory(subcategory);  return RedirectToAction("Index");  }  ViewBag.FKCategoryId = new SelectList(objCartegoryBO.GetCategories(true), "PKCategoryId", "CategoryName", subcategory.FKCategoryId);  return View(subcategory);  }  // GET: /Admin/SubCategory/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  SubCategory subcategory = objSubCategoryBO.GetsubCategory(id.Value);  if (subcategory == null)  {  return HttpNotFound();  }  return View(subcategory);  }  // POST: /Admin/SubCategory/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  objSubCategoryBO.DeleteSubCategory(id);  return RedirectToAction("Index");  }  }  } |

1. Goto Solution Explorer->Project->Areas->Admin->Models->Under Models->Right Click on Models-.Add->Class->Class name=SubCategoryVal->Click OK and Change the Namespace of class OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Models To OnlineShoppingStoreMVCUsingLinq (remove .Areas.Admin.Models)

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel.DataAnnotations;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public class SubCategoryVal  {  [Required(ErrorMessage="Please enter a subcategory")]  [Display(Name = "SubCategory Name")]  public string SubCategoryName { get; set; }  [Display(Name="Category Name")]  public string CategoryName { get; set; }  [Display(Name="Active")]  public bool IsActive { get; set; }  }  [MetadataType(typeof(SubCategoryVal))]  public partial class SubCategory  { }  } |

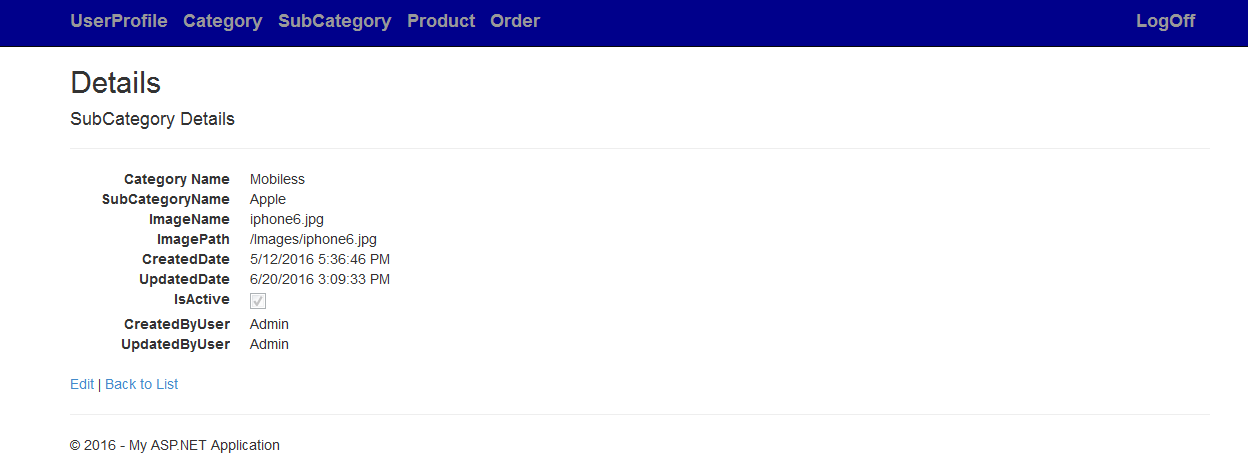


Views To The All Methods Of SubCategories Controller.

1. Go to Solution Explorer->Project->Areas->Admin->Views->Subcategory->Select Index.cshtml->Change the code as below

Index.cshtml

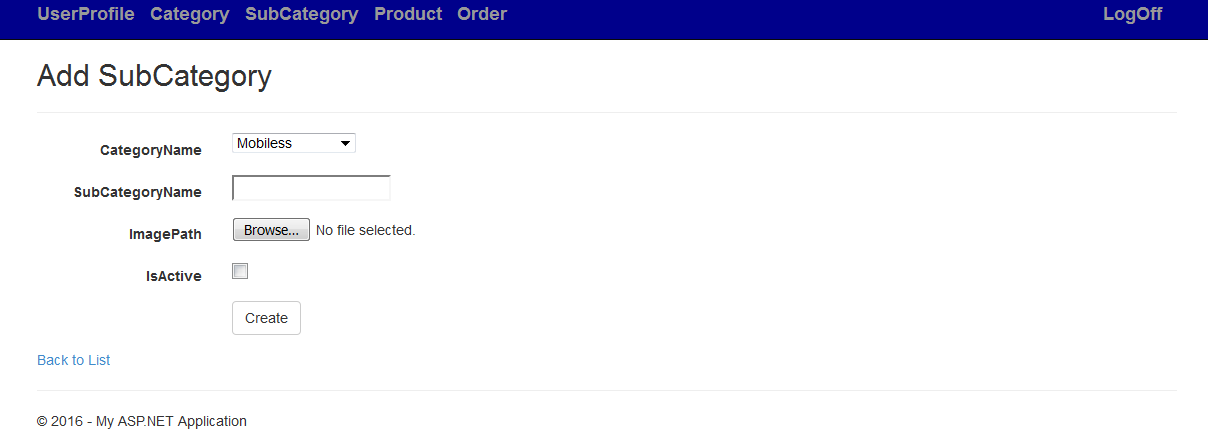
|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.SubCategory>  @{  ViewBag.Title = "Index";  }  <h2>SubCategory</h2>  <p>  @Html.ActionLink("Add SubCategory", "Create")  </p>  @using (Html.BeginForm())  {  <div>  @Html.Label("Category")  @Html.DropDownList("Category", ViewBag.CategoryId as SelectList," Select Category")  <input type="submit" name="btnSearch" value="Search" />  <input type="submit" name="btnReset" value="Reset" />  </div>  }  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.CategoryName)  </th>  <th>  @Html.DisplayNameFor(model => model.SubCategoryName)  </th>  <th>  @Html.DisplayNameFor(model => model.ImageName)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedByUser)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedByUser)  </th>  <th>  @Html.DisplayNameFor(model => model.IsActive)  </th>  <th></th>  </tr>  @foreach (var item in Model)  {  <tr>  <td>  @Html.DisplayFor(modelItem => item.CategoryName)  </td>  <td>  @Html.DisplayFor(modelItem => item.SubCategoryName)  </td>  <td>  @Html.DisplayFor(modelItem => item.ImageName)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedByUser)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedByUser)  </td>  <td>  @Html.DisplayFor(modelItem => item.IsActive)  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id = item.PKSubCategoryId }) |  @Html.ActionLink("Details", "Details", new { id = item.PKSubCategoryId }) |  @Html.ActionLink("Delete", "Delete", new { id = item.PKSubCategoryId })  </td>  </tr>  }  </table> |



1. Go to Solution Explorer->Project->Areas->Admin->Views-> SubCategory->Select Details.cshtml and change the code as below

Details.cshtml

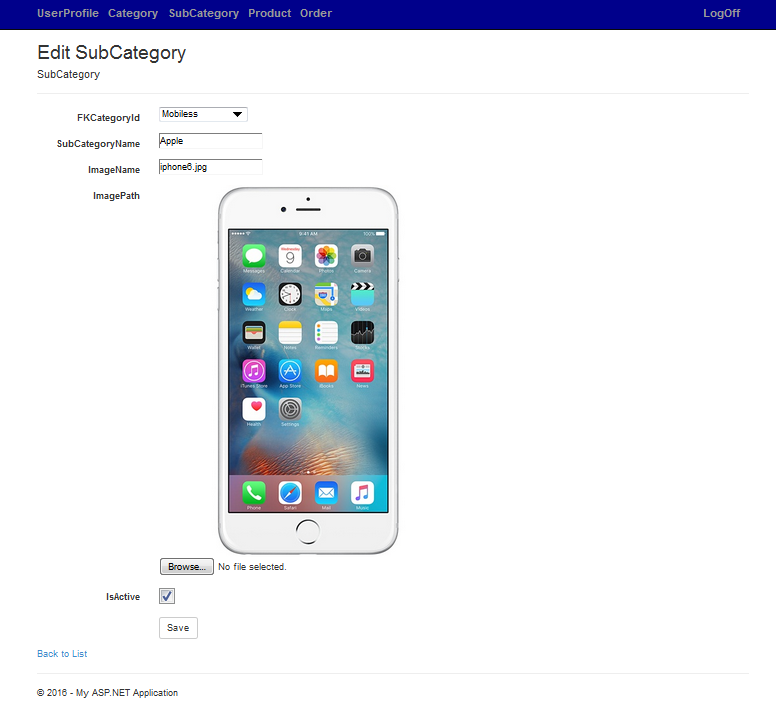
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.SubCategory  @{  ViewBag.Title = "Details";  }  <h2>Details</h2>  <div>  <h4>SubCategory Details</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.Category.CategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.Category.CategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.SubCategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.SubCategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ImageName)  </dt>  <dd>  @Html.DisplayFor(model => model.ImageName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ImagePath)  </dt>  <dd>  @Html.DisplayFor(model => model.ImagePath)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedByUser)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedByUser)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile1.UserName)  </dd>  </dl>  </div>  <p>  @Html.ActionLink("Edit", "Edit", new { id = Model.PKSubCategoryId }) |  @Html.ActionLink("Back to List", "Index")  </p> |



1. Go to Solution Explorer->Project->Areas->Admin->Views-> SubCategory->Select Create.cshtml and change the code as below

Create.cshtml

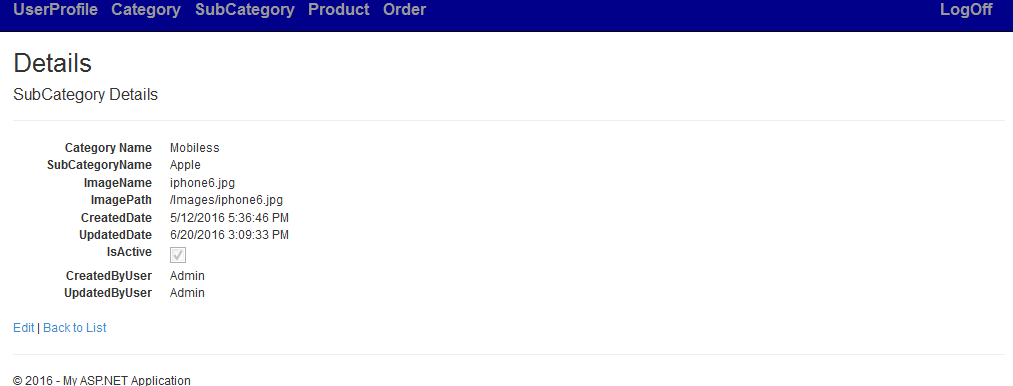
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.SubCategory  @{  ViewBag.Title = "Create";  }  <h2>Add SubCategory</h2>  @using (Html.BeginForm("Create", "SubCategory", FormMethod.Post, new { enctype = "multipart/form-data" }))  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <hr />  @Html.ValidationSummary(true)  <div class="form-group">  @Html.LabelFor(model => model.Category, "CategoryName", new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownList("CategoryName",ViewBag.FKCategoryId as SelectList)  @Html.ValidationMessageFor(model => model.CategoryName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.SubCategoryName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.SubCategoryName)  @Html.ValidationMessageFor(model => model.SubCategoryName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ImagePath, new { @class = "control-label col-md-2" })  <div class="col-md-10">  <input type="file" name="Images" id="Images" />  @Html.ValidationMessageFor(model => model.ImagePath)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Create" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div>  @section Scripts {  @Scripts.Render("~/bundles/jqueryval")  } |



1. Go to Solution Explorer->Project->Areas->Admin->Views-> SubCategory->Select Edit.cshtml and change the code as below

Edit.cshtml

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.SubCategory  @{  ViewBag.Title = "Edit";  }  <h2>Edit SubCategory</h2>  @using (Html.BeginForm("Edit", "SubCategory", FormMethod.Post, new { enctype = "multipart/form-data" }))  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>SubCategory</h4>  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKSubCategoryId)  @Html.HiddenFor(model => model.FKCreatedByUserId)  @Html.HiddenFor(model => model.FKUpdatedByUserId)  @Html.HiddenFor(model => model.CreatedDate)  @Html.HiddenFor(model => model.ImageName)  @Html.HiddenFor(model => model.ImagePath)  <div class="form-group">  @Html.LabelFor(model => model.FKCategoryId, "FKCategoryId", new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownList("FKCategoryId", String.Empty)  @Html.ValidationMessageFor(model => model.FKCategoryId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.SubCategoryName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.SubCategoryName)  @Html.ValidationMessageFor(model => model.SubCategoryName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ImageName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.ImageName)  @Html.ValidationMessageFor(model => model.ImageName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ImagePath, new { @class = "control-label col-md-2" })  <div class="col-md-10">  <img src="@Model.ImagePath" alt="loading...."/>  <input type="file" name="txtFile" value="" />  @Html.ValidationMessageFor(model => model.ImagePath)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |



Delete.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views-> SubCategory->Select Delete.cshtml and change the code as below

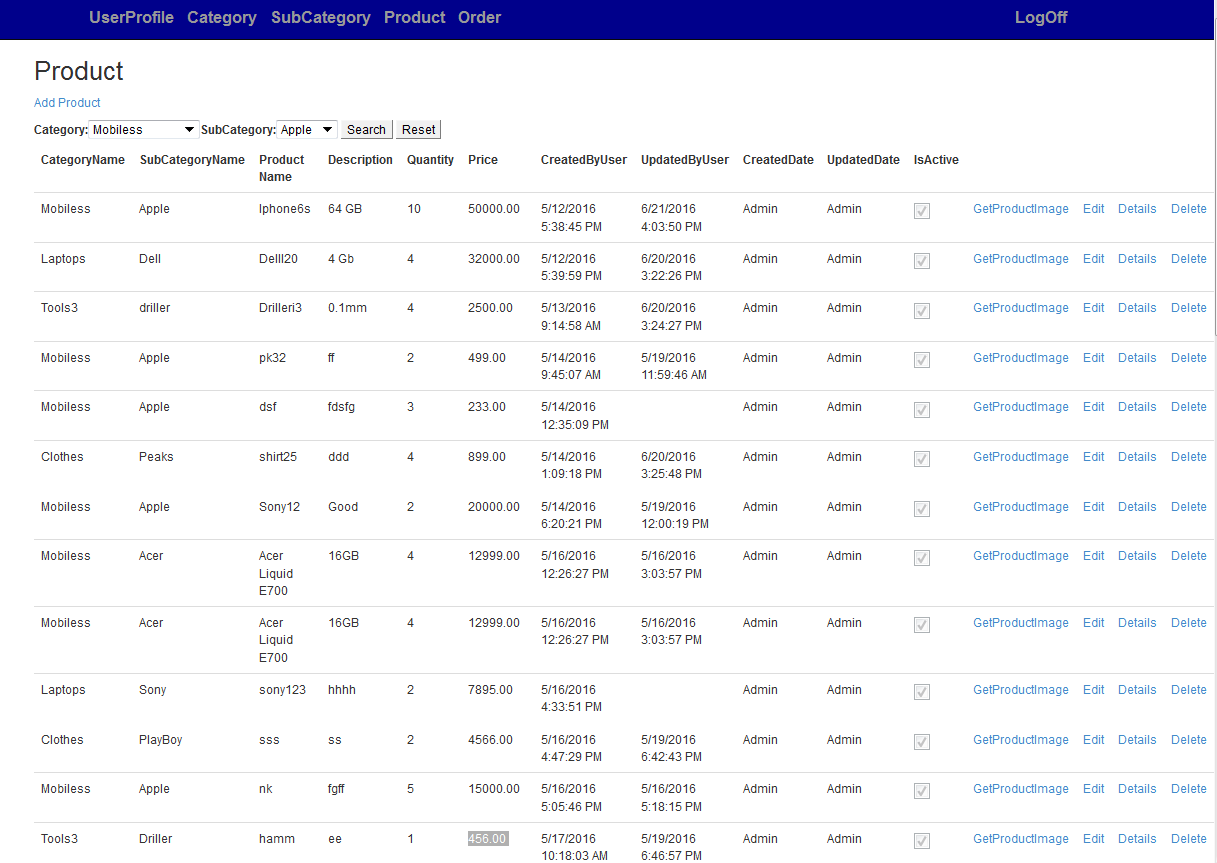
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.SubCategory  @{  ViewBag.Title = "Delete";  }  <h2>Delete Subcategory</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <h4>SubCategory</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.Category.CategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.Category.CategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.SubCategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.SubCategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ImageName)  </dt>  <dd>  @Html.DisplayFor(model => model.ImageName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ImagePath)  </dt>  <dd>  @Html.DisplayFor(model => model.ImagePath)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>    <dt>  @Html.DisplayNameFor(model => model.CreatedByUser)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedByUser)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile1.UserName)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index")  </div>  }  </div> |

1. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=ProductController → Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=Product(OnlineShoppingStoreMVCUsingLinq)→ DataContextClass=OnlineShoppingStoreEntities (OnlineShoppingStoreMVCUsingLinq) ->Click Add.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  [Authorize]  public class ProductController : Controller  {  ProductBO objProductBO = new ProductBO();  CategoryBO objCategoryBO = new CategoryBO();  SubCategoryBO objSubCategoryBO = new SubCategoryBO();  ProductImageBO objProductImageBO = new ProductImageBO();  // GET: /Admin/Product/  public ActionResult Index()  {  ViewBag.Category = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  ViewBag.SubCategory = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName");  return View(objProductBO.GetProducts());  }  [HttpPost]  public ActionResult Index(FormCollection fc)  {  if (fc["btnSearch"] == "Search")  {  int category = 0;  int subCategory = 0;  if (fc["Category"] != "")  category = Convert.ToInt32(fc["Category"]);  if (fc["SubCategory"] != "")  subCategory = Convert.ToInt32(fc["SubCategory"]);  ViewBag.Category = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName",category);  ViewBag.SubCategory = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName",subCategory);  return View(objProductBO.GetProducts(category, subCategory));  }  else  {  ViewBag.Category = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  ViewBag.SubCategory = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName");  return View(objProductBO.GetProducts());  }  }  // GET: /Admin/Product/Details/5  public ActionResult Details(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Product product = objProductBO.GetProduct(id.Value);  if (product == null)  {  return HttpNotFound();  }  return View(product);  }  public ActionResult GetSubCategories(int id=0)  {  var lstSubCategories = objSubCategoryBO.GetSubCategories(id);  return Json(lstSubCategories, JsonRequestBehavior.AllowGet);  }  // GET: /Admin/Product/Create  public ActionResult Create()  {  ViewBag.FKCategoryId = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  ViewBag.FKSubCategoryId = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName");  return View();  }  // POST: /Admin/Product/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(Product product, IEnumerable<HttpPostedFileBase> images)  {  ProductImage objProductImage = new ProductImage();  if (ModelState.IsValid)  {  objProductBO.InsertProduct(product);  foreach (HttpPostedFileBase img in images)  {  string imageName = System.IO.Path.GetFileName(img.FileName);  img.SaveAs(Server.MapPath("~/Images/" + imageName));  objProductImage.FKProductId = product.PKProductId;  objProductImage.ImageName = imageName;  objProductImage.ImagePath = "/Images/" + imageName;  objProductImageBO.InsertImage(objProductImage);  }  return RedirectToAction("Index");  }  ViewBag.FKCategoryId = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  ViewBag.FKSubCategoryId = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName");  return View(product);  }  // GET: /Admin/Product/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Product product = objProductBO.GetProduct(id.Value);  if (product == null)  {  return HttpNotFound();  }  ViewBag.FKCategoryId = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName",product.FKCategoryId);  ViewBag.FKSubCategoryId = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName",product.FKSubCategoryId);  return View(product);  }  // POST: /Admin/Product/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Edit(Product product, FormCollection fc)  {  if (ModelState.IsValid)  {  objProductBO.UpdateProduct(product);  if (fc["btnEdit"] == "Edit product images")  {  return RedirectToAction("Index", "Productimage", new { id=product.PKProductId });  }  return RedirectToAction("Index");  }  ViewBag.FKCategoryId = new SelectList(objCategoryBO.GetCategories(true), "PKCategoryId", "CategoryName");  ViewBag.FKSubCategoryId = new SelectList(objSubCategoryBO.GetSubCategories(0, true), "PKSubCategoryId", "SubCategoryName");  return View(product);  }  // GET: /Admin/Product/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Product product = objProductBO.GetProduct(id.Value);  if (product == null)  {  return HttpNotFound();  }  return View(product);  }  // POST: /Admin/Product/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  List<ProductImage> lstProductImage = objProductImageBO.GetImages(id);  foreach (ProductImage p in lstProductImage)  {  objProductImageBO.DeleteImage(p.PKImageId);  }  objProductBO.DeleteProduct(id);  return RedirectToAction("Index");  }  }  } |

1. Goto Solution Explorer->Project->Areas->Admin->Models->Under Models->Right Click on Models-.Add->Class->Class name=ProductVal->Click OK and Change the Namespace of class OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Models To OnlineShoppingStoreMVCUsingLinq (remove .Areas.Admin.Models)

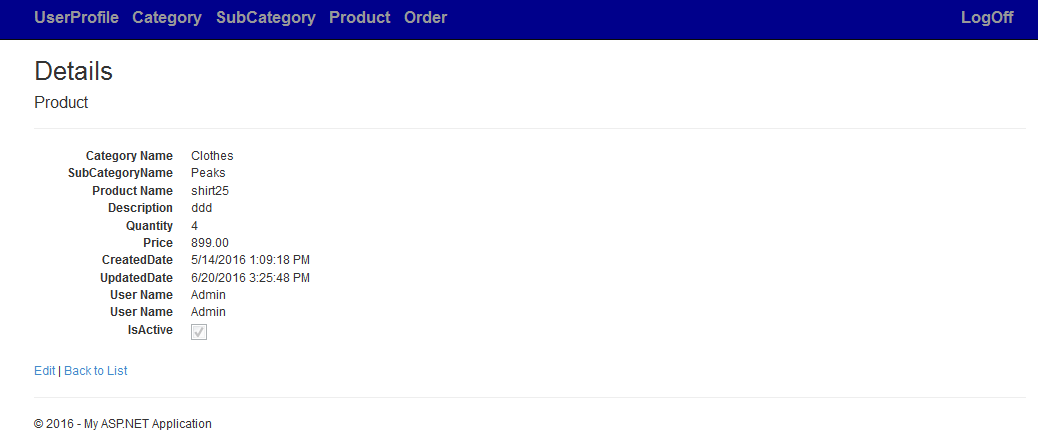
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel.DataAnnotations;  using System.Linq;  using System.Web;  namespace OnlineShoppingStoreMVCUsingLinq  {  public class ProductVal  {  [Required(ErrorMessage = "Please enter product name")]  [Display(Name = "Product Name")]  public string ProductName { get; set; }  [Required(ErrorMessage = "Please enter Descrption")]  public string Description { get; set; }  [Required]  public int Quantity { get; set; }  [Display(Name = "Active")]  public bool IsActive { get; set; }  }  [MetadataType(typeof(ProductVal))]  public partial class Product  {  }  } |



Index.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->Product->In Product,Select Index.cshtml->Change the code as below

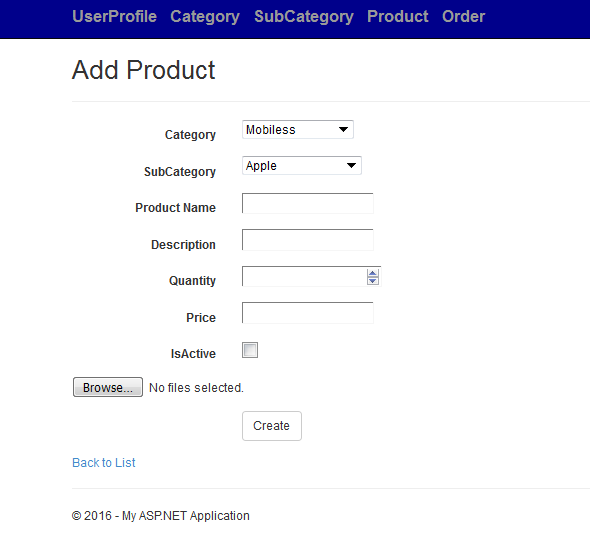
|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.Product>  @{  ViewBag.Title = "Index";  }  <h2>Product</h2>  <p>  @Html.ActionLink("Add Product", "Create")  </p>  <div>  @using (Html.BeginForm())  {  @Html.Label("Category:")  @Html.DropDownList("Category", ViewBag.Category as SelectList)  @Html.Label("SubCategory:")  @Html.DropDownList("SubCategory", ViewBag.SubCategory as SelectList)  <input type="submit" name="btnSearch" value="Search" />  <input type="submit" name="btnReset" value="Reset" />  }  </div>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.CategoryName)  </th>  <th>  @Html.DisplayNameFor(model => model.SubCategoryName)  </th>  <th>  @Html.DisplayNameFor(model => model.ProductName)  </th>  <th>  @Html.DisplayNameFor(model => model.Description)  </th>  <th>  @Html.DisplayNameFor(model => model.Quantity)  </th>  <th>  @Html.DisplayNameFor(model => model.Price)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedByUser)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedByUser)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.IsActive)  </th>  <th></th>  </tr>  @foreach (var item in Model)  {  <tr>  <td>  @Html.DisplayFor(modelItem => item.CategoryName)  </td>  <td>  @Html.DisplayFor(modelItem => item.SubCategoryName)  </td>  <td>  @Html.DisplayFor(modelItem => item.ProductName)  </td>  <td>  @Html.DisplayFor(modelItem => item.Description)  </td>  <td>  @Html.DisplayFor(modelItem => item.Quantity)  </td>  <td>  @Html.DisplayFor(modelItem => item.Price)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedByUser)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedByUser)  </td>  <td>  @Html.DisplayFor(modelItem => item.IsActive)  </td>  <td>  <a href="@Url.Action("Index", "ProductImage", new {id=item.PKProductId })">GetProductImage</a>  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id = item.PKProductId })  </td>  <td>  @Html.ActionLink("Details", "Details", new { id = item.PKProductId })  </td>  <td>  @Html.ActionLink("Delete", "Delete", new { id = item.PKProductId })  </td>  </tr>  }  </table>  @section scripts  {  <script>  $(document).ready(function () {  $('#Category').change(function (e) {  var id = $(this).val();  if (id != "") {  $.ajax({  url: "Product/GetSubCategories",  data: { id: id },  type: "GET"  }).success(function (data) {  $("#SubCategory").empty();  $.each(data, function (key, value) {  $("#SubCategory").append($("<option></option>").val(value.PKSubCategoryId).html(value.SubCategoryName));  });  })  }  })  $('#Category').trigger('change');  })  </script>  } |



Details.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->Product->In Product,Select Details.cshtml->Change the code as below

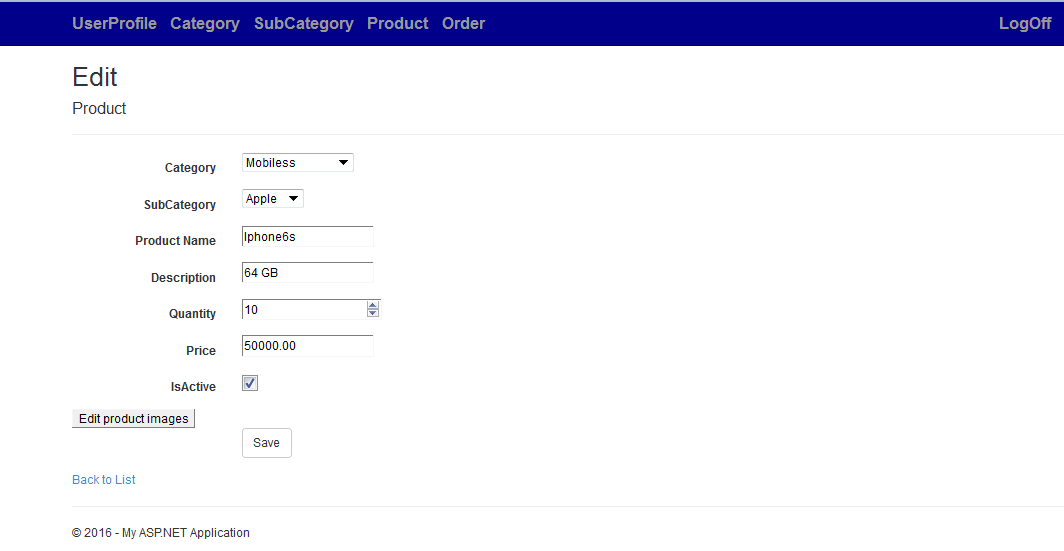
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Product  @{  ViewBag.Title = "Details";  }  <h2> ProductDetails</h2>  <div>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.Category.CategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.Category.CategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.SubCategory.SubCategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.SubCategory.SubCategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ProductName)  </dt>  <dd>  @Html.DisplayFor(model => model.ProductName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Description)  </dt>  <dd>  @Html.DisplayFor(model => model.Description)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Quantity)  </dt>  <dd>  @Html.DisplayFor(model => model.Quantity)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Price)  </dt>  <dd>  @Html.DisplayFor(model => model.Price)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile1.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile1.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  </dl>  </div>  <p>  @Html.ActionLink("Edit", "Edit", new { id = Model.PKProductId }) |  @Html.ActionLink("Back to List", "Index")  </p> |



Create.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->Product->In Product,Select Create.cshtml->Change the code as below

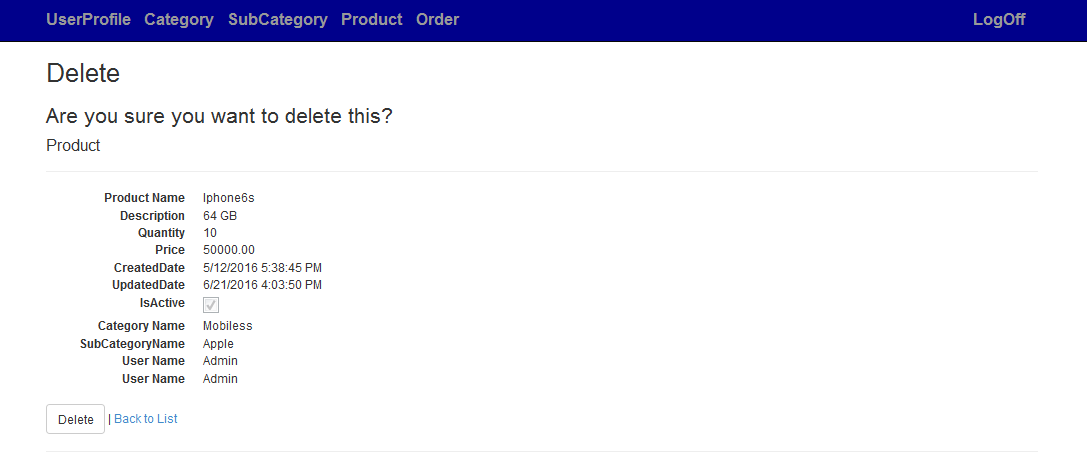
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Product  @{  ViewBag.Title = "Create";  }  <h2>Add Product</h2>  @using (Html.BeginForm("Create", "Product", FormMethod.Post, new { enctype = "multipart/form-data" }))  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>Product</h4>  <hr />  @Html.ValidationSummary(true)  <div class="form-group">  @Html.LabelFor(model => model.FKCategoryId, "Category", new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownList("FKCategoryId")  @Html.ValidationMessageFor(model => model.FKCategoryId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.FKSubCategoryId, "SubCategory", new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownList("FKSubCategoryId")  @Html.ValidationMessageFor(model => model.FKSubCategoryId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ProductName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.ProductName)  @Html.ValidationMessageFor(model => model.ProductName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Description, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Description)  @Html.ValidationMessageFor(model => model.Description)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Quantity, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Quantity)  @Html.ValidationMessageFor(model => model.Quantity)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Price, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Price)  @Html.ValidationMessageFor(model => model.Price)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  Image  <div class="col-md-10">  <input type="file" name="images" value="" multiple="multiple" />  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Create" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div>  @section scripts  {  <script>  $(document).ready(function()  {  $('#FKCategoryId').change(function (e)  {  var id = $(this).val();  $.ajax({  url: "GetSubCategories",  data: { id: id},  type: "GET"  }).success(function (data) {    $("#FKSubCategoryId").empty();  $.each(data, function (key, value) {  $("#FKSubCategoryId").append($("<option></option>").val(value.PKSubCategoryId).html(value.SubCategoryName));  });  })  })  $('#FKCategoryId').trigger('change');  })  </script>  } |



Edit.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->Product->In Product,Select Edit.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Product  @{  ViewBag.Title = "Edit";  }  <h2>Edit</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()  <div class="form-horizontal">  <h4>Product</h4>  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKProductId)  @Html.HiddenFor(model => model.CreatedDate)  @Html.HiddenFor(model => model.FKCreatedByUserId)  <div class="form-group">  @Html.LabelFor(model => model.FKCategoryId, "Category", new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownList("FKCategoryId")  @Html.ValidationMessageFor(model => model.FKCategoryId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.FKSubCategoryId, "SubCategory", new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownList("FKSubCategoryId")  @Html.ValidationMessageFor(model => model.FKSubCategoryId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ProductName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.ProductName)  @Html.ValidationMessageFor(model => model.ProductName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Description, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Description)  @Html.ValidationMessageFor(model => model.Description)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Quantity, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Quantity)  @Html.ValidationMessageFor(model => model.Quantity)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Price, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Price)  @Html.ValidationMessageFor(model => model.Price)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div>  <input type="submit" value="Edit product images" name="btnEdit" />  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div>  @section scripts  {  <script>  $(document).ready(function () {  $('#FKCategoryId').change(function (e) {  var id = $(this).val();  $.ajax({  url: "../GetSubCategories",  data: { id: id },  type: "GET"  }).success(function (data) {  $("#FKSubCategoryId").empty();  $.each(data, function (key, value) {  $("#FKSubCategoryId").append($("<option></option>").val(value.PKSubCategoryId).html(value.SubCategoryName));  });  })  })  $('#FKCategoryId').trigger('change');  })  </script>  } |



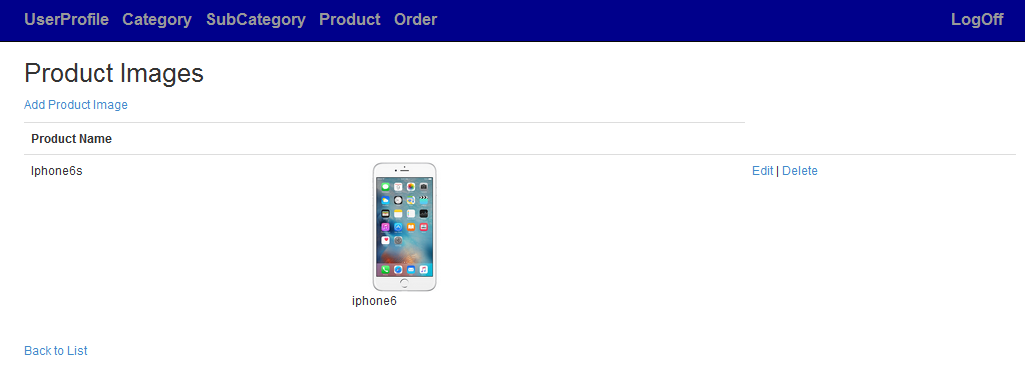
Delete.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->Product->In product,Select Delete.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Product  @{  ViewBag.Title = "Delete";  }  <h2>Delete</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <h4>Product</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.ProductName)  </dt>  <dd>  @Html.DisplayFor(model => model.ProductName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Description)  </dt>  <dd>  @Html.DisplayFor(model => model.Description)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Quantity)  </dt>  <dd>  @Html.DisplayFor(model => model.Quantity)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Price)  </dt>  <dd>  @Html.DisplayFor(model => model.Price)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Category.CategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.Category.CategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.SubCategory.SubCategoryName)  </dt>  <dd>  @Html.DisplayFor(model => model.SubCategory.SubCategoryName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile1.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile1.UserName)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index")  </div>  }  </div> |

1. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=ProductImageController → Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=ProductImage(OnlineShoppingStoreMVCUsingLinq)→ DataContextClass=OnlineShoppingStoreEntities (OnlineShoppingStoreMVCUsingLinq) ->Click Add.

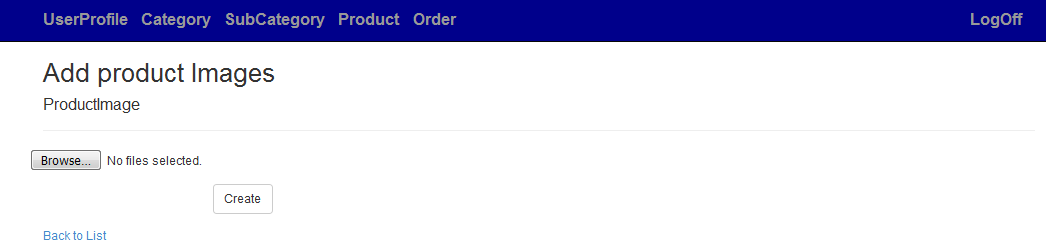
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  [Authorize]  public class ProductImageController : Controller  {  ProductImageBO objProductImageBO = new ProductImageBO();  // GET: /Admin/ProductImage/  public ActionResult Index(int? id)  {  TempData["productId"] = id.Value;  TempData.Keep();  return View(objProductImageBO.GetImages(id.Value));  }  // GET: /Admin/ProductImage/Create  public ActionResult Create()  {  return View();  }  // POST: /Admin/ProductImage/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(ProductImage productimage, IEnumerable<HttpPostedFileBase> images)  {  if (ModelState.IsValid)  {  foreach (HttpPostedFileBase img in images)  {  string imageName = System.IO.Path.GetFileName(img.FileName);  img.SaveAs(Server.MapPath("~/Images/" + imageName));  productimage.FKProductId = (int)TempData["productId"];  productimage.ImageName = imageName;  productimage.ImagePath = "/Images/" + imageName;  objProductImageBO.InsertImage(productimage);  }  return RedirectToAction("Index", new {id=productimage.FKProductId });  }  return View(productimage);  }  // GET: /Admin/ProductImage/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  ProductImage productimage = objProductImageBO.GetImage(id.Value);  if (productimage == null)  {  return HttpNotFound();  }  return View(productimage);  }  // POST: /Admin/ProductImage/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Edit(ProductImage productimage,HttpPostedFileBase image)  {  if (ModelState.IsValid)  {  string imageName = System.IO.Path.GetFileName(image.FileName);  image.SaveAs(Server.MapPath("~/Images/" + imageName));  productimage.ImageName = imageName;  productimage.ImagePath = "/Images/" + imageName;  objProductImageBO.UpdateImage(productimage);  }  return RedirectToAction("Index", new { id = productimage.FKProductId });  }  // GET: /Admin/ProductImage/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  ProductImage productimage = objProductImageBO.GetImage(id.Value);  if (productimage == null)  {  return HttpNotFound();  }  return View(productimage);  }  // POST: /Admin/ProductImage/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  objProductImageBO.DeleteImage(id);  int productId = Convert.ToInt32(TempData["productId"]);  return RedirectToAction("Index", new { id = productId });  }  }  } |



Index.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->ProductImage->In ProductImage,Select Index.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.ProductImage>  @{  ViewBag.Title = "Index";  }  <h2>Product Images</h2>  <p>  @Html.ActionLink("Add Product Image", "Create")  </p>  <table class="table">  <tr>    <th>  @Html.DisplayNameFor(model => model.Product.ProductName)  </th>  <th></th>  </tr>  @foreach (var item in Model) {  <tr>    <td>  @Html.DisplayFor(modelItem => item.Product.ProductName)  </td>  <td>  <img src="@item.ImagePath" alt="loading...." style="max-height:150px;max-width:120px"/>  <br/>  <p>@item.ImageName.Split('.')[0].ToString()</p>  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id=item.PKImageId }) |  @Html.ActionLink("Delete", "Delete", new { id=item.PKImageId })  </td>  </tr>  }  </table>  <p>  @Html.ActionLink("Back to List", "Index", "Product")  </p> |



Create.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->ProductImage->In ProductImage,Select Create.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.ProductImage  @{  ViewBag.Title = "Create";  }  <h2>Add product Images</h2>  @using (Html.BeginForm("Create", "ProductImage", FormMethod.Post, new { enctype = "multipart/form-data" }))  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>ProductImage</h4>  <hr />  @Html.ValidationSummary(true)  <div class="form-group">  <input type="file" name="images" value="" multiple="multiple" />  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Create" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |

Edit.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->ProductImage->In ProductImage,Select Edit.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.ProductImage  @{  ViewBag.Title = "Edit";  }  <h2>Edit</h2>  @using (Html.BeginForm("Edit", "ProductImage", FormMethod.Post, new { enctype = "multipart/form-data" }))  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>ProductImage</h4>  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKImageId)  @Html.HiddenFor(model=>model.FKProductId)  <img src="@Model.ImagePath" alt="loading...." style="max-height:150px;max-width:120px" />  <input type="file" name="image" value="" />  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index", new { id=Model.FKProductId})  </div> |

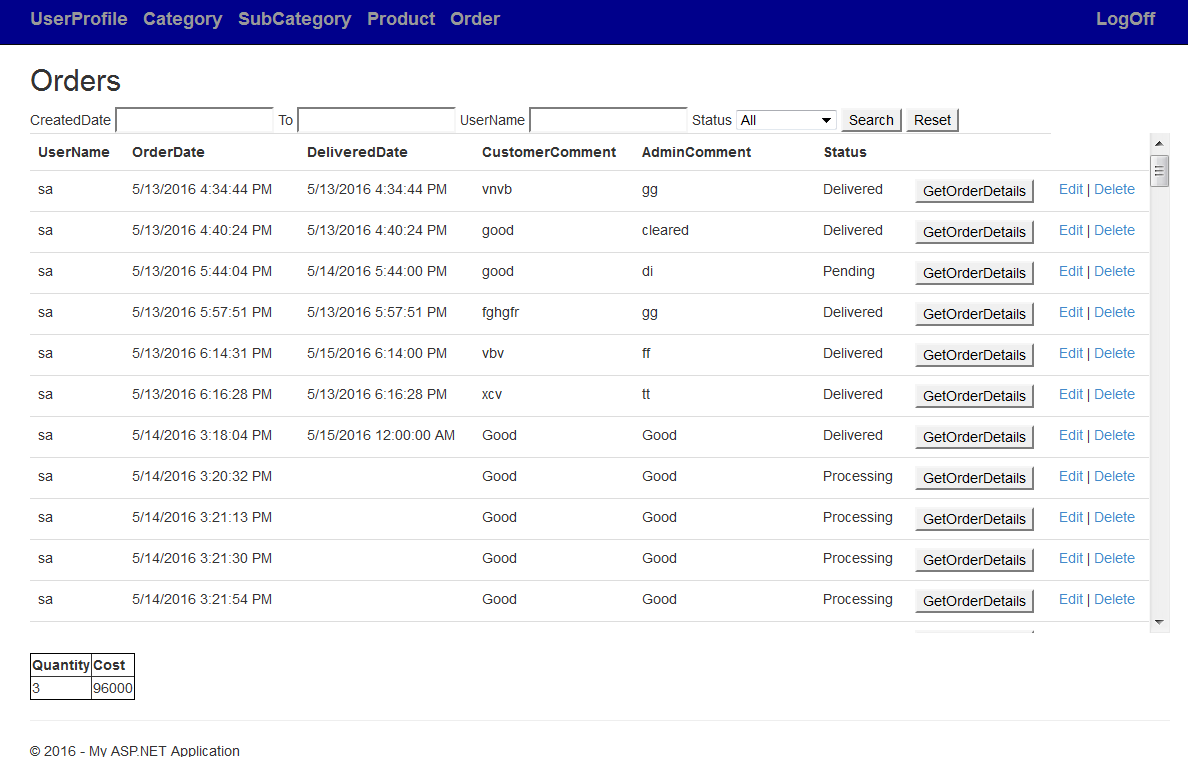
Delete.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->ProductImage->In ProductImage,Select Delete.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.ProductImage  @{  ViewBag.Title = "Delete";  }  <h2>Delete</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <h4>ProductImage</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.ImageName)  </dt>  <dd>  @Html.DisplayFor(model => model.ImageName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ImagePath)  </dt>  <dd>  @Html.DisplayFor(model => model.ImagePath)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Product.ProductName)  </dt>  <dd>  @Html.DisplayFor(model => model.Product.ProductName)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index", new { id=Model.FKProductId})  </div>  }  </div> |

1. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=Order Controller→ Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=Order(OnlineShoppingStoreMVCUsingLinq) → DataContextClass=OnlineShoppingStoreEntities (OnlineShoppingStoreMVCUsingLinq) ->Click Add.

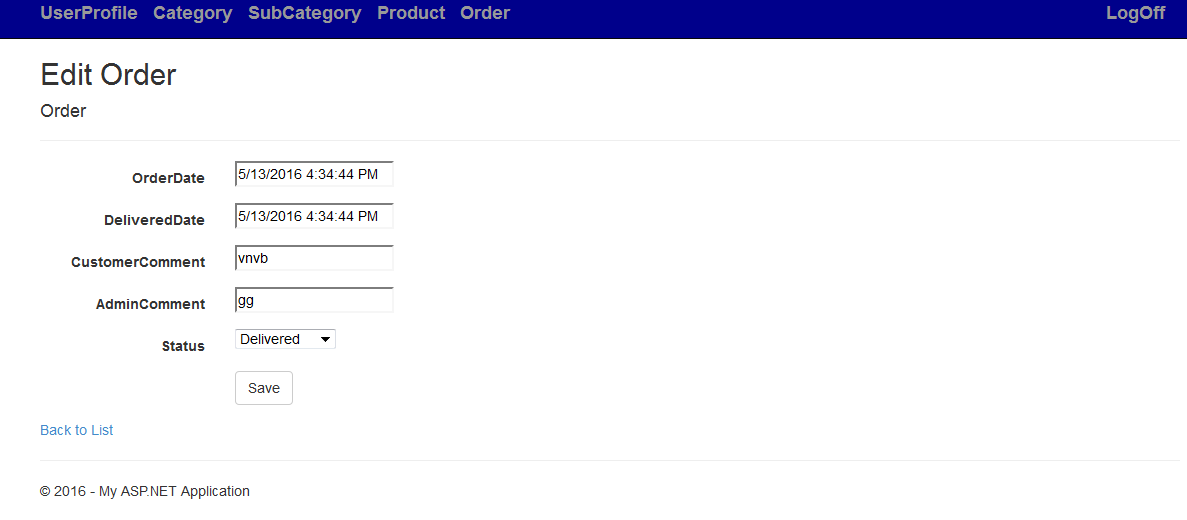
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  [Authorize]  public class OrderController : Controller  {  OrderBO objOrderBO = new OrderBO();  OrderHistory objOrderHistory = new OrderHistory();  OrderHistoryBO objOrderHistoryBO = new OrderHistoryBO();  // GET: /Admin/Order/  public ActionResult Index()  {  return View(objOrderBO.GetOrders());  }  [HttpPost]  public ActionResult Index(FormCollection fc)  {  DateTime fromDate;  DateTime toDate;  string userName;  string status;  if (fc["txtFromDate"] == "")  {  userName = fc["txtUserName"];  status = fc["Status"];  return View(objOrderBO.GetOrders(null, null, userName, status));  }  else if (fc["btnSearch"] == "Search")  {  fromDate = DateTime.Parse(fc["txtFromDate"]);  toDate = DateTime.Parse(fc["txtToDate"]);  userName = fc["txtUserName"];  status = fc["Status"];  return View(objOrderBO.GetOrders(fromDate, toDate, userName, status));  }  else  {  return View(objOrderBO.GetOrders());  }  }  // GET: /Admin/Order/Details/5  public ActionResult Details(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Order order = objOrderBO.GetOrder(id.Value);  if (order == null)  {  return HttpNotFound();  }  return View(order);  }  // GET: /Admin/Order/Create  public ActionResult Create()  {  return View();  }  // POST: /Admin/Order/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(Order order)  {  if (ModelState.IsValid)  {  objOrderBO.InsertOrder(order);  return RedirectToAction("Index");  }  return View(order);  }  // GET: /Admin/Order/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Order order = objOrderBO.GetOrder(id.Value);  if (order == null)  {  return HttpNotFound();  }  else  {  IEnumerable<Status> values = Enum.GetValues(typeof(Status)).Cast<Status>();  IEnumerable<SelectListItem> items = from value in values  select new SelectListItem  {  Text = value.ToString(),  Value = value.ToString(),  };  ViewBag.StatusData = items;  }  return View(order);  }  // POST: /Admin/Order/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Edit(Order order)  {  if (ModelState.IsValid)  {  objOrderBO.UpdateOrder(order);  objOrderHistory = objOrderHistoryBO.GetOrderHistory(order.PKOrderId);  if (objOrderHistory != null)  {  objOrderHistory.Status = order.Status;  objOrderHistory.Description = order.AdminComment;  objOrderHistoryBO.UpdateOrderHistory(objOrderHistory);  }  return RedirectToAction("Index");  }  return View(order);  }  // GET: /Admin/Order/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  Order order = objOrderBO.GetOrder(id.Value);  if (order == null)  {  return HttpNotFound();  }  return View(order);  }  // POST: /Admin/Order/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  objOrderBO.DeleteOrder(id);  return RedirectToAction("Index");  }  public JsonResult GetOrderDetails(int id)  {  OrderDetailBO objOrderDetailBO = new OrderDetailBO();  List<OrderDetail> lstOrderdetails = objOrderDetailBO.GetOrderDetails(id);  return Json(lstOrderdetails, JsonRequestBehavior.AllowGet);  }  }  } |



Index.xshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->Order->In Order,Select index.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.Order>  @{  ViewBag.Title = "Index";  }  <h2>Orders</h2>  @using (Html.BeginForm())  {  @Html.DisplayName("CreatedDate")  <input type="text" name="txtFromDate" class="Date"/>  @Html.DisplayName("To")  <input type="text" name="txtToDate" class="Date"/>  @Html.DisplayName("UserName")  <input type="text" name="txtUserName" />  @Html.DisplayName("Status")  <select name="Status">  <option value="" selected>All</option>  <option value="Pending">Pending</option>  <option value="Processing">Processing</option>  <option value="Delivered">Delivered</option>  <option value="Cancelled">Cancelled</option>  </select>  <input type="submit" name="btnSearch" value="Search" />  <input type="submit" name="btnReset" value="Reset" />  }  <div style="max-height:500px;overflow:auto;">  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.UserName)  </th>  <th>  @Html.DisplayNameFor(model => model.OrderDate)  </th>  <th>  @Html.DisplayNameFor(model => model.DeliveredDate)  </th>  <th>  @Html.DisplayNameFor(model => model.CustomerComment)  </th>  <th>  @Html.DisplayNameFor(model => model.AdminComment)  </th>  <th>  @Html.DisplayNameFor(model => model.Status)  </th>  <th></th>  </tr>  @foreach (var item in Model)  {  <tr>  <td>  @Html.DisplayFor(modelItem => item.UserName)  </td>  <td>  @Html.DisplayFor(modelItem => item.OrderDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.DeliveredDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.CustomerComment)  </td>  <td>  @Html.DisplayFor(modelItem => item.AdminComment)  </td>  <td>  @Html.DisplayFor(modelItem => item.Status)  </td>  <td>  <input type="button" id="@item.PKOrderId" value="GetOrderDetails" class="anchor" />  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id = item.PKOrderId }) |  @Html.ActionLink("Delete", "Delete", new { id = item.PKOrderId })  </td>  </tr>  }  </table>  </div>  <br />  <table border="1" id="tblOrderDetails" style="column-width: auto;border:1px solid black;">  <thead>  <tr style="padding-left:2%">  <th>Quantity</th>  <th>Cost</th>  </tr>  </thead>  <tbody></tbody>  </table>  @section scripts  {  <link rel="stylesheet" href="//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css">  <script src="//code.jquery.com/jquery-1.10.2.js"></script>  <script src="//code.jquery.com/ui/1.11.4/jquery-ui.js"></script>  <script type="text/javascript">  $(document).ready(function () {  $('.Date').datepicker();  });  <script type="text/javascript">  $(document).ready(function () {  $('.anchor').click(function () {  var id = $(this).attr("id");  $.ajax({  url: "Order/GetOrderDetails/" + id,  type: 'Get',  }).success(function (data) {  $("#tblOrderDetails tbody").empty();  for (var i = 0; i < data.length; i++) {  var rows = "<tr><td>" + data[i].Quantity + "</td><td>" + data[i].Cost + "</td></tr>";  $('#tblOrderDetails tbody').append(rows);  }  });  });  });  </script>  } |



Edit.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->Order->In Order,Select Edit.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Order  @{  ViewBag.Title = "Edit";  }  <h2>Edit Order</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKOrderId)  @Html.HiddenFor(model=>model.FKBillingAddressId)  @Html.HiddenFor(model=>model.FKCustomerUserId)  @Html.HiddenFor(model=>model.FKProcessByUserId)  @Html.HiddenFor(model=>model.FKProductId)  <div class="form-group">  @Html.LabelFor(model => model.OrderDate, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.OrderDate)  @Html.ValidationMessageFor(model => model.OrderDate)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.DeliveredDate, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.DeliveredDate)  @Html.ValidationMessageFor(model => model.DeliveredDate)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.CustomerComment, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.CustomerComment)  @Html.ValidationMessageFor(model => model.CustomerComment)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AdminComment, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AdminComment)  @Html.ValidationMessageFor(model => model.AdminComment)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Status, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.DropDownListFor(model => model.Status, ViewBag.StatusData as IEnumerable<SelectListItem>)  @Html.ValidationMessageFor(model => model.Status)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |

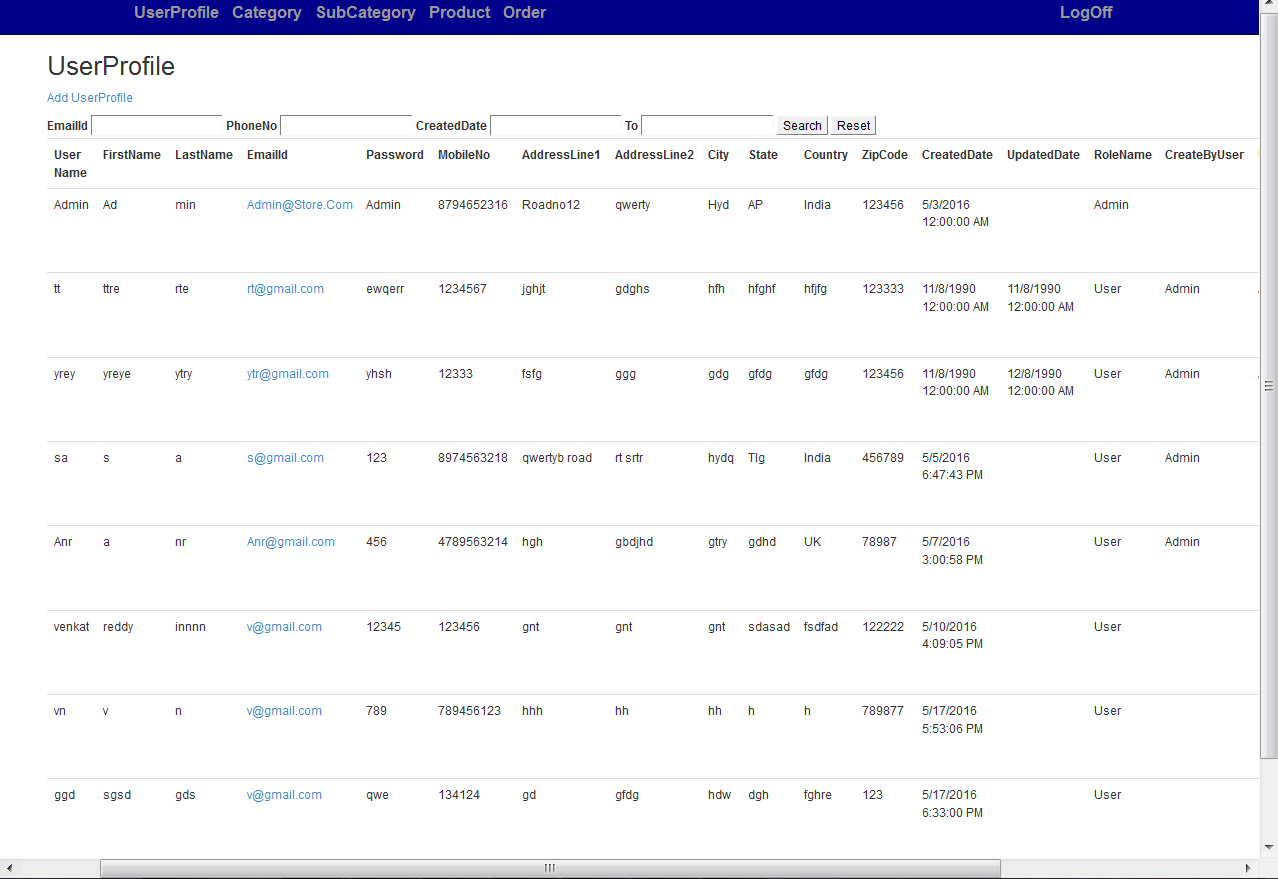
Delete.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->Order->In Order,Select Delete.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Order  @{  ViewBag.Title = "Delete";  }  <h2>Delete Order</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <h4>Order</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.OrderDate)  </dt>  <dd>  @Html.DisplayFor(model => model.OrderDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.DeliveredDate)  </dt>  <dd>  @Html.DisplayFor(model => model.DeliveredDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CustomerComment)  </dt>  <dd>  @Html.DisplayFor(model => model.CustomerComment)  </dd>  <dt>  @Html.DisplayNameFor(model => model.AdminComment)  </dt>  <dd>  @Html.DisplayFor(model => model.AdminComment)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Status)  </dt>  <dd>  @Html.DisplayFor(model => model.Status)  </dd>  <dt>  @Html.DisplayNameFor(model => model.BillingAddress.FirstName)  </dt>  <dd>  @Html.DisplayFor(model => model.BillingAddress.FirstName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile1.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile1.UserName)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index")  </div>  }  </div> |

1. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=UserProfileController → Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=UserProfile(OnlineShoppingStoreMVCUsingLinq) → DataContextClass=(OnlineShoppingStoreMVCUsingLinq) ->Click Add.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  [Authorize]  public class UserProfileController : Controller  {  UserProfileBO objUserProfileBO = new UserProfileBO();  // GET: /Admin/UserProfile/  public ActionResult Index()  {  return View(objUserProfileBO.GetUsers());  }  [HttpPost]  public ActionResult Index(FormCollection fc)  {  string emailId = fc["txtEmailId"];  string phoneNo = fc["txtPhoneNo"];  DateTime fromDate = DateTime.Parse(fc["txtFromDate"]);  DateTime toDate =DateTime.Parse(fc["txtToDate"]);  if (fc["btnSearch"] == "Search")  {  return View(objUserProfileBO.GetUsers(emailId, phoneNo, fromDate, toDate));  }  else  {  return View();  }  }  // GET: /Admin/UserProfile/Details/5  public ActionResult Details(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  UserProfile userprofile = objUserProfileBO.GetUser(id.Value);  if (userprofile == null)  {  return HttpNotFound();  }  return View(userprofile);  }  // GET: /Admin/UserProfile/Create  public ActionResult Create()  {  //ViewBag.FKRoleId = new SelectList(db.Roles, "PKRoleId", "RoleName");  //ViewBag.CreatedByUserId = new SelectList(db.UserProfiles, "PKUserId", "UserName");  //ViewBag.UpdatedByUserId = new SelectList(db.UserProfiles, "PKUserId", "UserName");  return View();  }  // POST: /Admin/UserProfile/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(UserProfile userprofile)  {  if (ModelState.IsValid)  {  objUserProfileBO.InsertUser(userprofile);  return RedirectToAction("Index");  }  return View(userprofile);  }  // GET: /Admin/UserProfile/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  UserProfile userprofile = objUserProfileBO.GetUser(id.Value);  if (userprofile == null)  {  return HttpNotFound();  }  return View(userprofile);  }  // POST: /Admin/UserProfile/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Edit(UserProfile userprofile)  {  if (ModelState.IsValid)  {  objUserProfileBO.UpdateUser(userprofile);  return RedirectToAction("Index");  }    return View(userprofile);  }  // GET: /Admin/UserProfile/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  UserProfile userprofile = objUserProfileBO.GetUser(id.Value);  if (userprofile == null)  {  return HttpNotFound();  }  return View(userprofile);  }  // POST: /Admin/UserProfile/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  objUserProfileBO.DeleteUser(id);  return RedirectToAction("Index");  }  public ActionResult UserLoginHistory(int id)  {  UserLoginHistoryBO objUserLoginHistoryBO = new UserLoginHistoryBO();  IEnumerable<UserLoginHistory> user= objUserLoginHistoryBO.GetUserLoginHistories(id);  return View(user);  }  public ActionResult OrderHistory(int? id=0)  {  OrderHistoryBO objOrderHistoryBO = new OrderHistoryBO();  objOrderHistoryBO.GetOrderHistories(id.Value);  return View("Index");  }  }  } |



Index.cshtml:

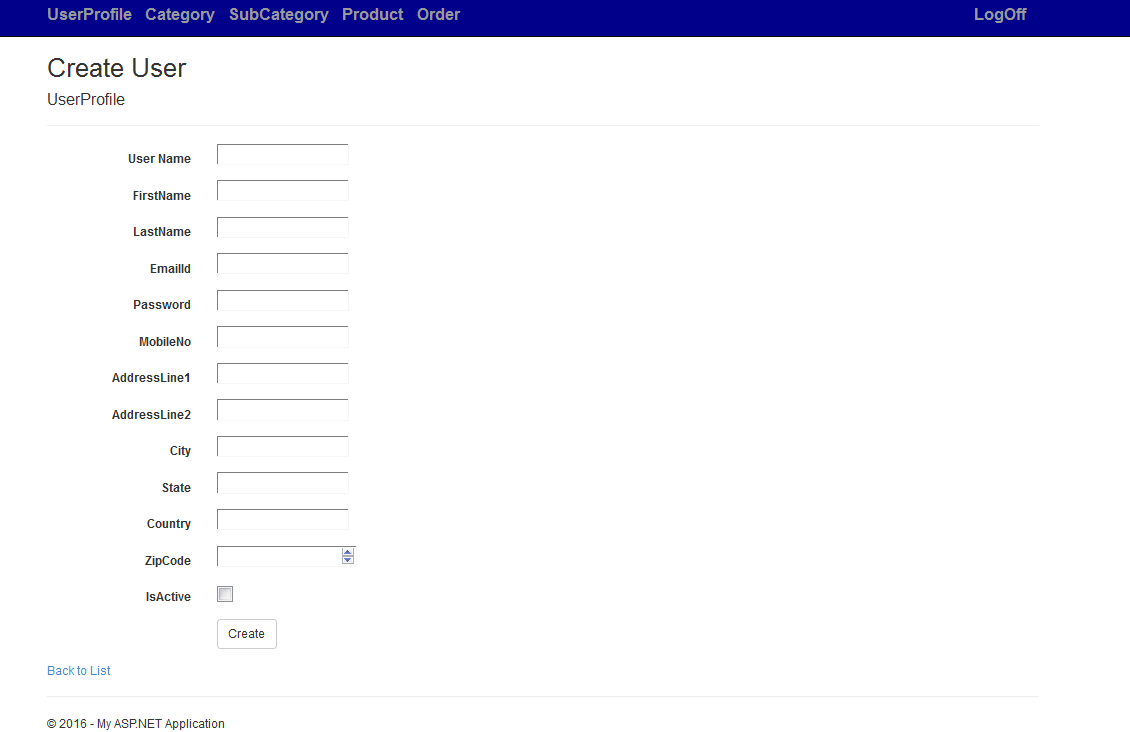
1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select Index.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.UserProfile>  @{  ViewBag.Title = "Index";  }  <h2>UserProfile</h2>  <p>  @Html.ActionLink("Add UserProfile", "Create")  </p>  @using (Html.BeginForm())  {  <div>  @Html.Label("EmailId")  <input type="text" name="txtEmailId" />  @Html.Label("PhoneNo")  <input type="text" name="txtPhoneNo" />  @Html.Label("CreatedDate")  <input type="text" name="txtFromDate" class="Date"/>  @Html.Label("To")  <input type="text" name="txtToDate" class="Date"/>  <input type="submit" name="btnSearch" value="Search" />  <input type="submit" name="btnReset" value="Reset"/>  </div>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.UserName)  </th>  <th>  @Html.DisplayNameFor(model => model.FirstName)  </th>  <th>  @Html.DisplayNameFor(model => model.LastName)  </th>  <th>  @Html.DisplayNameFor(model => model.EmailId)  </th>  <th>  @Html.DisplayNameFor(model => model.Password)  </th>  <th>  @Html.DisplayNameFor(model => model.MobileNo)  </th>  <th>  @Html.DisplayNameFor(model => model.AddressLine1)  </th>  <th>  @Html.DisplayNameFor(model => model.AddressLine2)  </th>  <th>  @Html.DisplayNameFor(model => model.City)  </th>  <th>  @Html.DisplayNameFor(model => model.State)  </th>  <th>  @Html.DisplayNameFor(model => model.Country)  </th>  <th>  @Html.DisplayNameFor(model => model.ZipCode)  </th>  <th>  @Html.DisplayNameFor(model => model.CreatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.UpdatedDate)  </th>  <th>  @Html.DisplayNameFor(model => model.Role.RoleName)  </th>  <th>  @Html.DisplayName("CreateByUser")  </th>  <th>  @Html.DisplayName("UpdatedByUser")  </th>  <th>  @Html.DisplayNameFor(model => model.IsActive)  </th>  <th></th>  </tr>  @foreach (var item in Model) {  <tr>  <td>  @Html.DisplayFor(modelItem => item.UserName)  </td>  <td>  @Html.DisplayFor(modelItem => item.FirstName)  </td>  <td>  @Html.DisplayFor(modelItem => item.LastName)  </td>  <td>  @Html.DisplayFor(modelItem => item.EmailId)  </td>  <td>  @Html.DisplayFor(modelItem => item.Password)  </td>  <td>  @Html.DisplayFor(modelItem => item.MobileNo)  </td>  <td>  @Html.DisplayFor(modelItem => item.AddressLine1)  </td>  <td>  @Html.DisplayFor(modelItem => item.AddressLine2)  </td>  <td>  @Html.DisplayFor(modelItem => item.City)  </td>  <td>  @Html.DisplayFor(modelItem => item.State)  </td>  <td>  @Html.DisplayFor(modelItem => item.Country)  </td>  <td>  @Html.DisplayFor(modelItem => item.ZipCode)  </td>  <td>  @Html.DisplayFor(modelItem => item.CreatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.UpdatedDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.Role.RoleName)  </td>  <td>  @Html.DisplayFor(modelItem => item.UserProfile2.UserName)  </td>  <td>  @Html.DisplayFor(modelItem => item.UserProfile3.UserName)  </td>  <td>  @Html.DisplayFor(modelItem => item.IsActive)  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id=item.PKUserId }) |  @Html.ActionLink("Details", "Details", new { id=item.PKUserId }) |  @Html.ActionLink("Delete", "Delete", new { id=item.PKUserId })|  @Html.ActionLink("UserLoginHistory", "UserLoginHistory", new { id=item.PKUserId})  @Html.ActionLink("OrderHistory", "OrderHistory", new {id=item.PKUserId})  </td>  </tr>  }  </table>  }  @section scripts  {  <link rel="stylesheet" href="//code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css">  <script src="//code.jquery.com/jquery-1.10.2.js"></script>  <script src="//code.jquery.com/ui/1.11.4/jquery-ui.js"></script>  <script type="text/javascript">  $(document).ready(function () {  $('.Date').datepicker();  });  </script>  } |

Details.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select Details.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.UserProfile  @{  ViewBag.Title = "Details";  }  <h2>User Details</h2>  <div>  <h4>UserProfile</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.FirstName)  </dt>  <dd>  @Html.DisplayFor(model => model.FirstName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.LastName)  </dt>  <dd>  @Html.DisplayFor(model => model.LastName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.EmailId)  </dt>  <dd>  @Html.DisplayFor(model => model.EmailId)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Password)  </dt>  <dd>  @Html.DisplayFor(model => model.Password)  </dd>  <dt>  @Html.DisplayNameFor(model => model.MobileNo)  </dt>  <dd>  @Html.DisplayFor(model => model.MobileNo)  </dd>  <dt>  @Html.DisplayNameFor(model => model.AddressLine1)  </dt>  <dd>  @Html.DisplayFor(model => model.AddressLine1)  </dd>  <dt>  @Html.DisplayNameFor(model => model.AddressLine2)  </dt>  <dd>  @Html.DisplayFor(model => model.AddressLine2)  </dd>  <dt>  @Html.DisplayNameFor(model => model.City)  </dt>  <dd>  @Html.DisplayFor(model => model.City)  </dd>  <dt>  @Html.DisplayNameFor(model => model.State)  </dt>  <dd>  @Html.DisplayFor(model => model.State)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Country)  </dt>  <dd>  @Html.DisplayFor(model => model.Country)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ZipCode)  </dt>  <dd>  @Html.DisplayFor(model => model.ZipCode)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Role.RoleName)  </dt>  <dd>  @Html.DisplayFor(model => model.Role.RoleName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile2.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile2.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile3.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile3.UserName)  </dd>  </dl>  </div>  <p>  @Html.ActionLink("Edit", "Edit", new { id = Model.PKUserId }) |  @Html.ActionLink("Back to List", "Index")  </p> |



Create.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select Create.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.UserProfile  @{  ViewBag.Title = "Create";  }  <h2>Create User</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>UserProfile</h4>  <hr />  @Html.ValidationSummary(true)  <div class="form-group">  @Html.LabelFor(model => model.UserName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.UserName)  @Html.ValidationMessageFor(model => model.UserName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.FirstName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.FirstName)  @Html.ValidationMessageFor(model => model.FirstName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.LastName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.LastName)  @Html.ValidationMessageFor(model => model.LastName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.EmailId, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.EmailId)  @Html.ValidationMessageFor(model => model.EmailId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Password, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Password)  @Html.ValidationMessageFor(model => model.Password)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.MobileNo, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.MobileNo)  @Html.ValidationMessageFor(model => model.MobileNo)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AddressLine1, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AddressLine1)  @Html.ValidationMessageFor(model => model.AddressLine1)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AddressLine2, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AddressLine2)  @Html.ValidationMessageFor(model => model.AddressLine2)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.City)  @Html.ValidationMessageFor(model => model.City)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.State, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.State)  @Html.ValidationMessageFor(model => model.State)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Country, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Country)  @Html.ValidationMessageFor(model => model.Country)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ZipCode, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.ZipCode)  @Html.ValidationMessageFor(model => model.ZipCode)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Create" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |

Edit.cshtml:

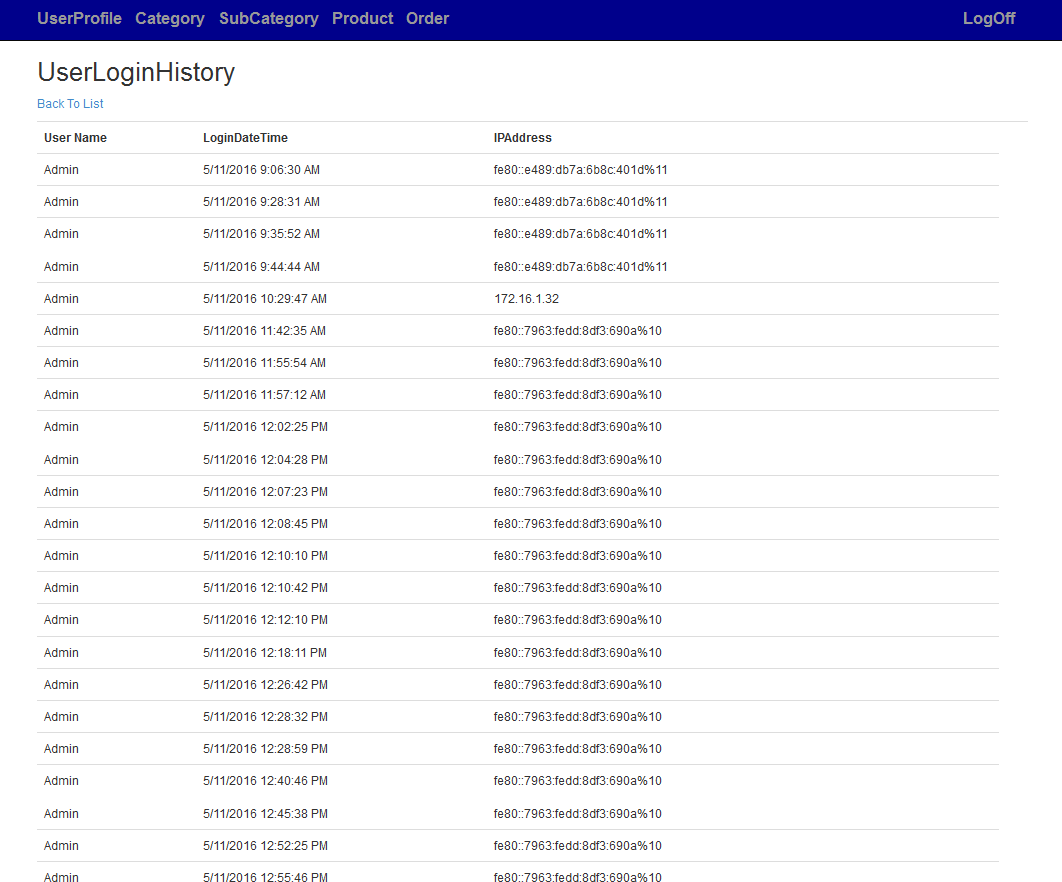
1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select Edit.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.UserProfile  @{  ViewBag.Title = "Edit";  }  <h2>Edit User Profile</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>UserProfile</h4>  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKUserId)  @Html.HiddenFor(model=>model.FKRoleId)  @Html.HiddenFor(model=>model.CreatedByUserId)  @Html.HiddenFor(model=>model.CreatedDate)  <div class="form-group">  @Html.LabelFor(model => model.UserName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.UserName)  @Html.ValidationMessageFor(model => model.UserName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.FirstName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.FirstName)  @Html.ValidationMessageFor(model => model.FirstName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.LastName, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.LastName)  @Html.ValidationMessageFor(model => model.LastName)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.EmailId, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.EmailId)  @Html.ValidationMessageFor(model => model.EmailId)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Password, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Password)  @Html.ValidationMessageFor(model => model.Password)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.MobileNo, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.MobileNo)  @Html.ValidationMessageFor(model => model.MobileNo)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AddressLine1, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AddressLine1)  @Html.ValidationMessageFor(model => model.AddressLine1)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.AddressLine2, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.AddressLine2)  @Html.ValidationMessageFor(model => model.AddressLine2)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.City, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.City)  @Html.ValidationMessageFor(model => model.City)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.State, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.State)  @Html.ValidationMessageFor(model => model.State)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.Country, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.Country)  @Html.ValidationMessageFor(model => model.Country)  </div>  </div>  <div class="form-group">  @Html.LabelFor(model => model.ZipCode, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.ZipCode)  @Html.ValidationMessageFor(model => model.ZipCode)  </div>  </div>    </div>  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div>  @section Scripts {  @Scripts.Render("~/bundles/jqueryval")  } |

Delete.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select Delete.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.UserProfile  @{  ViewBag.Title = "Delete";  }  <h2>Delete User Profile</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <h4>UserProfile</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.FirstName)  </dt>  <dd>  @Html.DisplayFor(model => model.FirstName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.LastName)  </dt>  <dd>  @Html.DisplayFor(model => model.LastName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.EmailId)  </dt>  <dd>  @Html.DisplayFor(model => model.EmailId)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Password)  </dt>  <dd>  @Html.DisplayFor(model => model.Password)  </dd>  <dt>  @Html.DisplayNameFor(model => model.MobileNo)  </dt>  <dd>  @Html.DisplayFor(model => model.MobileNo)  </dd>  <dt>  @Html.DisplayNameFor(model => model.AddressLine1)  </dt>  <dd>  @Html.DisplayFor(model => model.AddressLine1)  </dd>  <dt>  @Html.DisplayNameFor(model => model.AddressLine2)  </dt>  <dd>  @Html.DisplayFor(model => model.AddressLine2)  </dd>  <dt>  @Html.DisplayNameFor(model => model.City)  </dt>  <dd>  @Html.DisplayFor(model => model.City)  </dd>  <dt>  @Html.DisplayNameFor(model => model.State)  </dt>  <dd>  @Html.DisplayFor(model => model.State)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Country)  </dt>  <dd>  @Html.DisplayFor(model => model.Country)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ZipCode)  </dt>  <dd>  @Html.DisplayFor(model => model.ZipCode)  </dd>  <dt>  @Html.DisplayNameFor(model => model.CreatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.CreatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UpdatedDate)  </dt>  <dd>  @Html.DisplayFor(model => model.UpdatedDate)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  <dt>  @Html.DisplayNameFor(model => model.Role.RoleName)  </dt>  <dd>  @Html.DisplayFor(model => model.Role.RoleName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile2.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile2.UserName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.UserProfile3.UserName)  </dt>  <dd>  @Html.DisplayFor(model => model.UserProfile3.UserName)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index")  </div>  }  </div> |



UserLoginHistory.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select UserLoginHistory.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.UserLoginHistory>  @{  ViewBag.Title = "UserLoginHistory";  }  <h2>UserLoginHistory</h2>  <p>  @Html.ActionLink("Back To List","Index")  </p>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.UserProfile.UserName)  </th>  <th>  @Html.DisplayNameFor(model => model.LoginDateTime)  </th>  <th>  @Html.DisplayNameFor(model => model.IPAddress)  </th>  <th></th>  </tr>  @foreach (var item in Model) {  <tr>  <td>  @Html.DisplayFor(modelItem => item.UserProfile.UserName)  </td>  <td>  @Html.DisplayFor(modelItem => item.LoginDateTime)  </td>  <td>  @Html.DisplayFor(modelItem => item.IPAddress)  </td>  </tr>  }  </table>  <p>  @Html.ActionLink("Back To List", "Index")  </p> |

OrderHistory.cshtml:

1. Go to Solution Explorer->Project->Areas->Admin->Views->UserProfile->In UserProfile,Select Orderhistory.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.OrderHistory>  @{  ViewBag.Title = "OrderHistory";  }  <h2>OrderHistory</h2>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.Description)  </th>  <th>  @Html.DisplayNameFor(model => model.Status)  </th>  <th>  @Html.DisplayNameFor(model => model.DateChanged)  </th>  <th>  @Html.DisplayNameFor(model => model.Order.CustomerComment)  </th>  <th></th>  </tr>  @foreach (var item in Model) {  <tr>  <td>  @Html.DisplayFor(modelItem => item.Description)  </td>  <td>  @Html.DisplayFor(modelItem => item.Status)  </td>  <td>  @Html.DisplayFor(modelItem => item.DateChanged)  </td>  <td>  @Html.DisplayFor(modelItem => item.Order.CustomerComment)  </td>  </tr>  }  </table> |

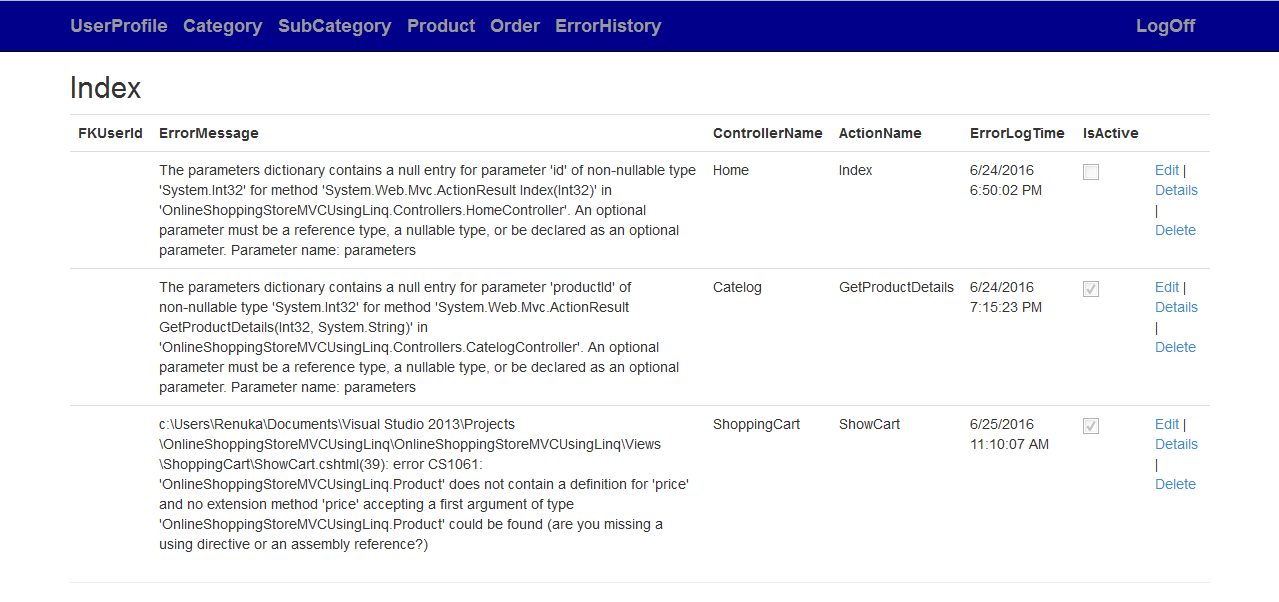
1. Go to Solution Explorer->Project->Select Areas->Admin->Controllers → Right Click On Controller →Add Controller →ControllerName=Error History→ Template=MVC Controller with read/write actions and views using EntityFramework →ModelClass=Error((OnlineShoppingStoreMVCUsingLinq) → DataContextClass=(OnlineShoppingStoreMVCUsingLinq)->Click Add.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data;  using System.Data.Entity;  using System.Linq;  using System.Net;  using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq;  using OnlineShoppingStoreMVCUsingLinq.BO;  namespace OnlineShoppingStoreMVCUsingLinq.Areas.Admin.Controllers  {  public class ErrorHistoryController : Controller  {  ErrorHistoryBO objErrorHistoryBO = new ErrorHistoryBO();  // GET: /Admin/Error/  public ActionResult Index()  {  return View(objErrorHistoryBO.GetErrorHistories());  }  // GET: /Admin/Error/Details/5  public ActionResult Details(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  ErrorHistory errorhistory = objErrorHistoryBO.GetErrorHistory(id.Value);  if (errorhistory == null)  {  return HttpNotFound();  }  return View(errorhistory);  }  // GET: /Admin/Error/Create  public ActionResult Create()  {  return View();  }  // POST: /Admin/Error/Create    [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult Create(ErrorHistory errorhistory)  {  if (ModelState.IsValid)  {  objErrorHistoryBO.InsertErrorHistory(errorhistory);  return RedirectToAction("Index");  }  return View(errorhistory);  }  // GET: /Admin/Error/Edit/5  public ActionResult Edit(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  ErrorHistory errorhistory = objErrorHistoryBO.GetErrorHistory(id.Value);  if (errorhistory == null)  {  return HttpNotFound();  }  return View(errorhistory);  }  // POST: /Admin/Error/Edit/5    [HttpPost]  [ValidateAntiForgeryToken]  [ValidateInput(false)]  public ActionResult Edit(ErrorHistory errorhistory)  {  if (ModelState.IsValid)  {  objErrorHistoryBO.UpdateErrorHistory(errorhistory);  return RedirectToAction("Index");  }  return View(errorhistory);  }  // GET: /Admin/Error/Delete/5  public ActionResult Delete(int? id)  {  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  ErrorHistory errorhistory = objErrorHistoryBO.GetErrorHistory(id.Value);  if (errorhistory == null)  {  return HttpNotFound();  }  return View(errorhistory);  }  // POST: /Admin/Error/Delete/5  [HttpPost, ActionName("Delete")]  [ValidateAntiForgeryToken]  public ActionResult DeleteConfirmed(int id)  {  objErrorHistoryBO.DeleteErrorHistory(id);  return RedirectToAction("Index");  }  }  } |

Index.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->ErrorHistory->In ErrorHistory,Select Index.cshtml->Change the code as below

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.ErrorHistory>  @{  ViewBag.Title = "Index";  }  <h2>Index</h2>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.FKUserId)  </th>  <th>  @Html.DisplayNameFor(model => model.ErrorMessage)  </th>  <th>  @Html.DisplayNameFor(model => model.ControllerName)  </th>  <th>  @Html.DisplayNameFor(model => model.ActionName)  </th>  <th>  @Html.DisplayNameFor(model => model.ErrorLogTime)  </th>  <th>  @Html.DisplayNameFor(model => model.IsActive)  </th>  <th></th>  </tr>  @foreach (var item in Model) {  <tr>  <td>  @Html.DisplayFor(modelItem => item.FKUserId)  </td>  <td>  @Html.DisplayFor(modelItem => item.ErrorMessage)  </td>  <td>  @Html.DisplayFor(modelItem => item.ControllerName)  </td>  <td>  @Html.DisplayFor(modelItem => item.ActionName)  </td>  <td>  @Html.DisplayFor(modelItem => item.ErrorLogTime)  </td>  <td>  @Html.DisplayFor(modelItem => item.IsActive)  </td>  <td>  @Html.ActionLink("Edit", "Edit", new { id=item.PKErrorId }) |  @Html.ActionLink("Details", "Details", new { id=item.PKErrorId }) |  @Html.ActionLink("Delete", "Delete", new { id=item.PKErrorId })  </td>  </tr>  }  </table> |



Details.cshtml

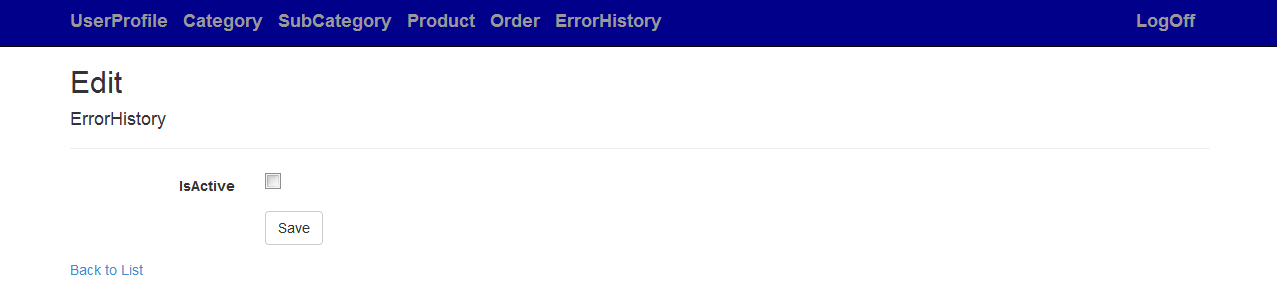
1. Go to Solution Explorer->Project->Areas->Admin->Views->ErrorHistory->In ErrorHistory,Select Details.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.ErrorHistory  @{  ViewBag.Title = "Details";  }  <h2>Details</h2>  <div>  <h4>ErrorHistory</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.FKUserId)  </dt>  <dd>  @Html.DisplayFor(model => model.FKUserId)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ErrorMessage)  </dt>  <dd>  @Html.DisplayFor(model => model.ErrorMessage)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ControllerName)  </dt>  <dd>  @Html.DisplayFor(model => model.ControllerName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ActionName)  </dt>  <dd>  @Html.DisplayFor(model => model.ActionName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ErrorLogTime)  </dt>  <dd>  @Html.DisplayFor(model => model.ErrorLogTime)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  </dl>  </div>  <p>  @Html.ActionLink("Edit", "Edit", new { id = Model.PKErrorId }) |  @Html.ActionLink("Back to List", "Index")  </p> |

Edit.cshtml

1. Go to Solution Explorer->Project->Areas->Admin->Views->ErrorHistory->In ErrorHistory,Select Edit.cshtml->Change the code as below

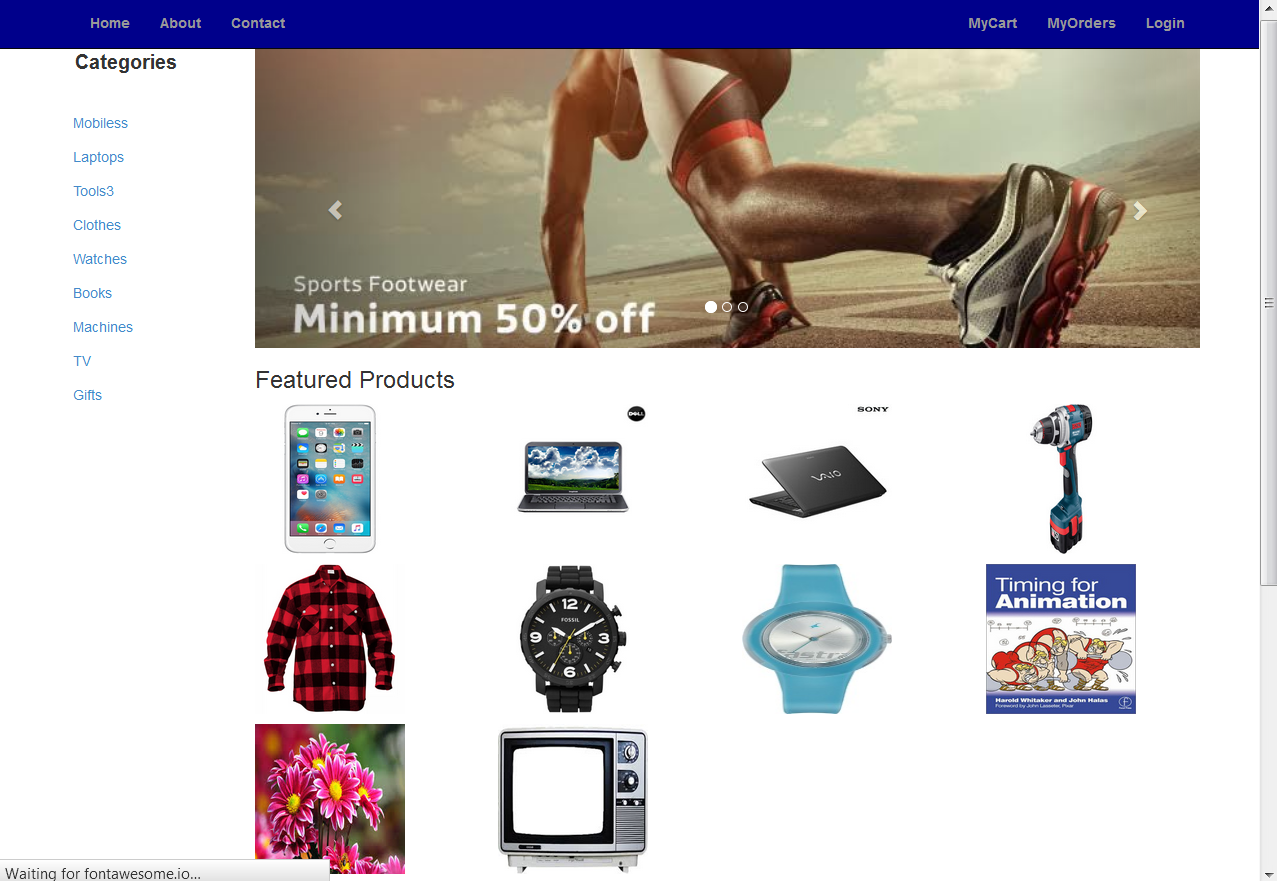
|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.ErrorHistory  @{  ViewBag.Title = "Edit";  }  <h2>Edit</h2>  @using (Html.BeginForm())  {  @Html.AntiForgeryToken()    <div class="form-horizontal">  <h4>ErrorHistory</h4>  <hr />  @Html.ValidationSummary(true)  @Html.HiddenFor(model => model.PKErrorId)  @Html.HiddenFor(model=>model.FKUserId)  @Html.HiddenFor(model=>model.ErrorMessage)  @Html.HiddenFor(model=>model.ErrorLogTime)  @Html.HiddenFor(model=>model.ActionName)  @Html.HiddenFor(model=>model.ControllerName)  <div class="form-group">  @Html.LabelFor(model => model.IsActive, new { @class = "control-label col-md-2" })  <div class="col-md-10">  @Html.EditorFor(model => model.IsActive)  @Html.ValidationMessageFor(model => model.IsActive)  </div>  </div>  <div class="form-group">  <div class="col-md-offset-2 col-md-10">  <input type="submit" value="Save" class="btn btn-default" />  </div>  </div>  </div>  }  <div>  @Html.ActionLink("Back to List", "Index")  </div> |



Delete.cshml

1. Go to Solution Explorer->Project->Areas->Admin->Views->ErrorHistory->In ErrorHistory,Select Delete.cshtml->Change the code as below

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.ErrorHistory  @{  ViewBag.Title = "Delete";  }  <h2>Delete</h2>  <h3>Are you sure you want to delete this?</h3>  <div>  <h4>ErrorHistory</h4>  <hr />  <dl class="dl-horizontal">  <dt>  @Html.DisplayNameFor(model => model.FKUserId)  </dt>  <dd>  @Html.DisplayFor(model => model.FKUserId)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ErrorMessage)  </dt>  <dd>  @Html.DisplayFor(model => model.ErrorMessage)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ControllerName)  </dt>  <dd>  @Html.DisplayFor(model => model.ControllerName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ActionName)  </dt>  <dd>  @Html.DisplayFor(model => model.ActionName)  </dd>  <dt>  @Html.DisplayNameFor(model => model.ErrorLogTime)  </dt>  <dd>  @Html.DisplayFor(model => model.ErrorLogTime)  </dd>  <dt>  @Html.DisplayNameFor(model => model.IsActive)  </dt>  <dd>  @Html.DisplayFor(model => model.IsActive)  </dd>  </dl>  @using (Html.BeginForm()) {  @Html.AntiForgeryToken()  <div class="form-actions no-color">  <input type="submit" value="Delete" class="btn btn-default" /> |  @Html.ActionLink("Back to List", "Index")  </div>  }  </div> |



UserModule:

GoTo Solution Explorer->Project->Views->Shared Folder -> Under Shared folder -> Add-> MVC5 LayoutPage->Click Ok -> Change the code as below

UserLayout.cshtml:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>@ViewBag.Title - My ASP.NET Application</title>  @Styles.Render("~/Content/css")  @Scripts.Render("~/bundles/modernizr")  </head>  <body>  <div class="navbar navbar-inverse navbar-fixed-top" style="background-color:darkblue">  <div class="container">  <div class="navbar-collapse collapse" style="font-weight:bolder">  <ul class=" nav navbar-nav">  <li>@Html.ActionLink("Home", "Index", "Home")</li>  <li>@Html.ActionLink("About", "About", "Home")</li>  <li>@Html.ActionLink("Contact", "Contact", "Home")</li>  </ul>  <ul class="nav navbar-nav navbar-right">  <li>  @Html.ActionLink("MyCart", "ShowCart", "ShoppingCart")  </li>  <li>@Html.ActionLink("MyOrders", "GetOrderDetails", "MyOrder")</li>  @if (Request.IsAuthenticated)  {  <li>@Html.ActionLink("LogOut", "SignOut", "Account")</li>  }  else  {  <li>@Html.ActionLink("Login", "Login", "Account")</li>  }  </ul>  </div>  </div>  </div>  <div class="container">  <div class="row">  <div class="col-xs-2">  <p style="font-weight:bold;font-size:20px;padding-left:15px;">Categories</p>  <br />  @Html.Partial("CategoryPartialView")  </div>  <div class="col-xs-10">  <div>  @RenderBody()  </div>  </div>  </div>  <hr />  <link href="https://fortawesome.github.io/Font-Awesome/assets/font-awesome/css/font-awesome.css" rel="stylesheet">  <!--footer start from here-->  <footer>  <div class="container">  <div class="row">  <div class="col-md-4 col-sm-6 footerleft ">  <div class="logofooter"><img src="/Images/Logo.jpg" height="100"/></div>  <p>Online Shopping Store.</p>  <p><i class="fa fa-map-pin"></i> 210, Deccansoft, opp.Pardise Hotel, Hyderabad - 110085, INDIA</p>  <p><i class="fa fa-phone"></i> Phone (India) : +91 9\*\*\*\*\*\*5\*\*\*</p>  <p><i class="fa fa-envelope"></i> E-mail : info@ShoppingStore.com</p>  </div>  <div class="col-md-2 col-sm-6 paddingtop-bottom">  <h6 class="heading7">GENERAL LINKS</h6>  <ul class="footer-ul Primary" >  <li>@Html.ActionLink("Home", "Index", "Home", new { style = " color:#a0a3a4" })</li>  <li ><a href="#" > Privacy Policy</a></li>  <li><a href="#" > Terms & Conditions</a></li>  <li><a href="#"> Frequently Ask Questions</a></li>  </ul>  </div>  <div class="col-md-3 col-sm-6 paddingtop-bottom">  <h6 class="heading7">LATEST POST</h6>  <div class="post">  <p>facebook crack the movie advertisment code:what it means for you <span>june 23,2016</span></p>  <p>facebook crack the movie advertisment code:what it means for you <span>june 22,2016</span></p>  </div>  </div>  <div class="col-md-3 col-sm-6 paddingtop-bottom">  <div class="fb-page" data-href="https://www.facebook.com/facebook" data-tabs="timeline" data-height="300" data-small-header="false" style="margin-bottom:15px;" data-adapt-container-width="true" data-hide-cover="false" data-show-facepile="true">  <div class="fb-xfbml-parse-ignore">  <blockquote cite="https://www.facebook.com/facebook"><a href="https://www.facebook.com/facebook" >Facebook</a></blockquote>    </div>  </div>  </div>  </div>  </div>  </footer>  </div>  <!--footer start from here-->  <div class="copyright">  <div class="container">  <div class="col-md-6">  <p>&copy; @DateTime.Now.Year - Online Shopping Store</p>  </div>  <div class="col-md-6">  <ul class="bottom\_ul">  <li><a href="#">About us</a></li>  <li><a href="#">Blog</a></li>  <li><a href="#">Faq's</a></li>  <li><a href="#">Contact us</a></li>  <li><a href="#">Site Map</a></li>  </ul>  </div>  </div>  </div>  @Scripts.Render("~/bundles/jquery")  @Scripts.Render("~/bundles/bootstrap")  @RenderSection("scripts", required: false)  </body>  </html> |

1. Create Partial View:

GoTo Solution Explorer->Project->Views->Shared Folder ->Right Click On Shared Folder->Add->NewItem ->Select as->MVC4 Partial Page(Razor)-> name=CategoryPartialView.cshtml.

CategoryPartialView.cshtml:

|  |
| --- |
| @using OnlineShoppingStoreMVCUsingLinq.BO;  @{  CategoryBO objCategoryBO = new CategoryBO();  }  <ul class="list-group" >  @foreach (Category item in objCategoryBO.GetCategories())  {  <li style="list-style: none;" class="list-group-item-success ">  <a class="btn" href="@Url.Action("GetSubCategory", "Catelog", new { categoryId = item.PKCategoryId, categoryName = item.CategoryName })">@item.CategoryName</a><br />  </li>  }  </ul> |

ProductsPartialView.cshtml:

1. GoTo Solution Explorer->Project->Views->Shared Folder ->Right Click On Shared Folder->Add->NewItem ->Select as->MVC4 Partial Page(Razor)-> name=ProductsPartialView.cshtml

|  |
| --- |
| @using OnlineShoppingStoreMVCUsingLinq.BO  @{  ProductBO objProductBO = new ProductBO();  }  <div class="row">  @foreach (var product in objProductBO.GetProducts())  {  <div class="col-xs-12 col-sm-3 img-responsive" style="padding-bottom:10px;">  <a href=@Url.Action("GetProductImages", "Catelog", new { productId = product.PKProductId })><img src="@product.ImagePath" alt="Loading..." height="150" width="150"/></a>  </div>  }  </div> |

1. Go to Solution Explorer->Project->Select Controllers → Right Click On Controller →Add Controller →ControllerName=Home→Select Template As Empty MVC Controller →Click Add.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVCUsingLinq.Controllers  {  public class HomeController : Controller  {  //  // GET: /Home/  public ActionResult Index()  {  return View();  }  public ActionResult About()  {  return View();  }  public ActionResult Contact()  {  return View();  }  }  } |

Index.cshtml

1. GoTo Solution Explorer->Project->Views->Home->Select Index-> Double click on it -> Change the code as below

or if View is not able

GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Account Controller->Double click on it->Index Method In Home Controller->Right Click On Index Method->Add View-> View name-Index>Click Add.

|  |
| --- |
| @{  ViewBag.Title = "Home";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <div id="myCarousel" class="carousel slide" data-ride="carousel" >  <!-- Indicators -->  <ol class="carousel-indicators">  <li data-target="#myCarousel" data-slide-to="0" class="active"></li>  <li data-target="#myCarousel" data-slide-to="1"></li>  <li data-target="#myCarousel" data-slide-to="2"></li>  </ol>  <!-- Wrapper for slides -->  <div class="carousel-inner" role="listbox" style="height:300px">  <div class="item active">  <img src="/Images/shoe.jpg" alt="Loading" width="100%" height="100%">  </div>  <div class="item">  <img src="/Images/sm.jpg" alt="Loading" width="100%" height="100%">  </div>  <div class="item">  <img src="/Images/shirt.jpg" alt="Loading" width="100%" height="100%">  </div>  </div>  <!-- Left and right controls -->  <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">  <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>  <span class="sr-only">Previous</span>  </a>  <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">  <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>  <span class="sr-only">Next</span>  </a>  </div>  <div>  <h3>Featured Products</h3>  @Html.Partial("ProductsPartailview")  </div> |

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Home Controller->Double click on it->About Method In Home Controller->Right Click On About Method->Add View-> Select Scaffold template as none->Click Add.and change the code as below

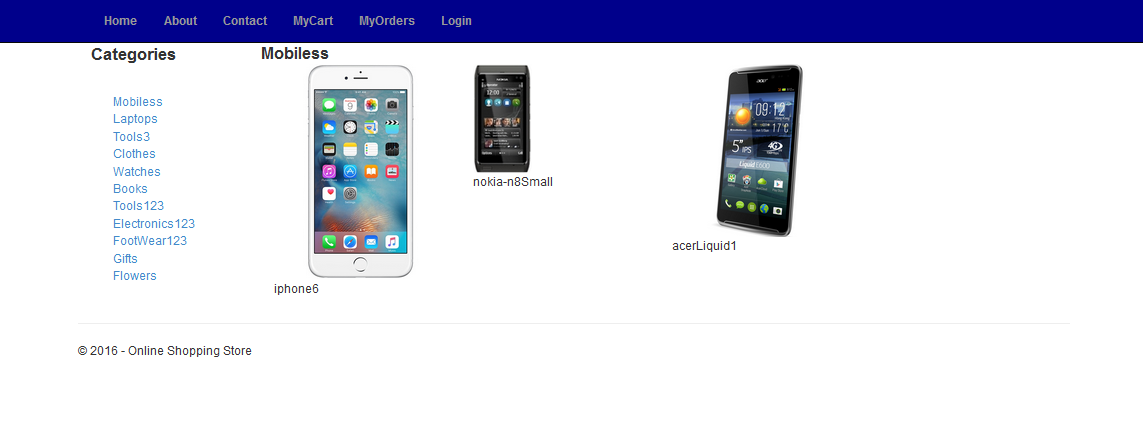
|  |
| --- |
| @{  ViewBag.Title = "About";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <h2>About</h2> |

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Home Controller->Double click on it->Contact Method In Home Controller->Right Click On Contact Method->Add View-> Select Scaffold template as none->Click Add.and change the code as below

|  |
| --- |
| @{  ViewBag.Title = "Contact Us";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <h2>Contact Us</h2> |

1. Go toSolution Explore->Project-> Controllers → Right Click On Controller →Add Controller → ControllerName=CatelogController→Select Template As Empty MVC Controller →Click Add.

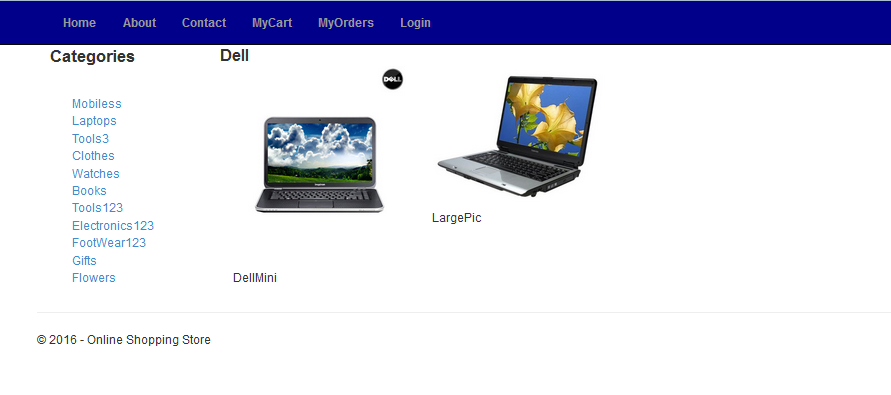
|  |
| --- |
| using OnlineShoppingStoreLibrary.BO;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVC.Controllers  {  public class CatelogController : Controller  {  SubCategoryBO objSubCategoryBO = new SubCategoryBO();  ProductBO objProductBO = new ProductBO();  ProductImageBO objProductImageBO = new ProductImageBO();  //  // GET: /Catelog/  public ActionResult Index()  {  return View();  }  public ActionResult GetSubCategory(int categoryId,string categoryName)  {  ViewBag.CategoryName = categoryName;  ViewBag.CategoryId = categoryId;  return View(objSubCategoryBO.GetSubCategories(categoryId));  }  public ActionResult GetProducts(int categoryId,string categoryName,int subCategoryId,string subCategoryName)  {  ViewBag.CategoryName = categoryName;  ViewBag.CategoryId = categoryId;  ViewBag.SubCategoryName = subCategoryName;  ViewBag.SubCategoryId = subCategoryId;  return View(objProductBO.GetProducts(categoryId,subCategoryId));  }  public ActionResult GetProductImages(int productId)  {  ViewBag.ProductId = productId;  return View(objProductImageBO.GetImages(productId));  }  public ActionResult GetProductDetails(int productId,string imagePath)  {  ViewBag.ProductId = productId;  ViewBag.Imagepath = imagePath;  return View(objProductBO.GetProduct(productId));  }  }  } |



GetSubCategory.cshtml:

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Catelog Controller->Double click on it-> GetSubCategory Method->Right Click On GetSubCategory Method->Add View-> View name-GetSubCategory.cshtml:>Click Add.

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.SubCategory>  @{  ViewBag.Title = "SubCategory";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <div style="font-weight:bold;font-size:18px;margin:0px 0px 0px 15px">@ViewBag.CategoryName</div>  <div class="container">  @foreach (var items in Model)  {  <div class="col-xs-3 img-responsive">  <a href="@Url.Action("GetProducts", "Catelog", new { categoryId = @items.FKCategoryId, categoryName = @items.CategoryName, subCategoryId = @items.PKSubCategoryId, subCategoryName = @items.SubCategoryName })"><img src="@items.ImagePath" alt="Loading..... " height="150" width="150"/></a>  <p>@items.ImageName.Split('.')[0].ToString()</p>  </div>  }  </div> |



GetProducts.cshtml:

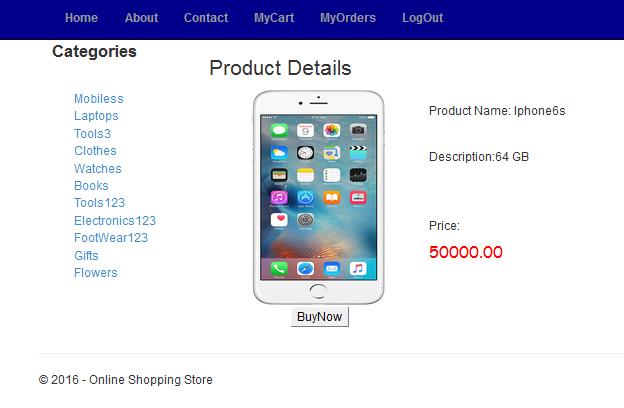
1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Catelog Controller->Double click on it-> GetProducts Method->Right Click On GetProducts Method->Add View-> View name-GetProducts .cshtml:>Click Add.

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.Product>  @{  ViewBag.Title = "Products";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <div style="font-weight:bold;font-size:18px;margin:0px 0px 0px 15px">@ViewBag.subCategoryName</div>  <div class="container">  @foreach (var items in Model)  {  <div class="col-sm-3 img-responsive">  <a href=@Url.Action("GetProductImages", "Catelog", new {productId=@items.PKProductId })><center><img src="@items.ImagePath" alt="Loading.." style="max-height:150px" /><br/> @items.ImageName.Split('.')[0].ToString()</center></a>    </div>  }  </div> |

31. GetProductImages.cshtml:

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Catelog Controller->Double click on it-> GetProductImages Method->Right Click On GetProductImages Method->Add View-> View name-GetProductImages .cshtml->Click Add.

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.ProductImage>  @{  ViewBag.Title = "ProductImages";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <h3></h3>  <div class="container">  @foreach (var items in Model)  {  <div class="col-sm-3 img-responsive">  <a href=@Url.Action("GetProductDetails", "Catelog", new { productId = @items.FKProductId,imagePath=@items.ImagePath})>  <center>  <img src="@items.ImagePath" alt="Loading.." style="max-height:180px"/>  <p>@items.ImageName.Split('.')[0].ToString()</p>  </center>  </a>  </div>  }  </div> |



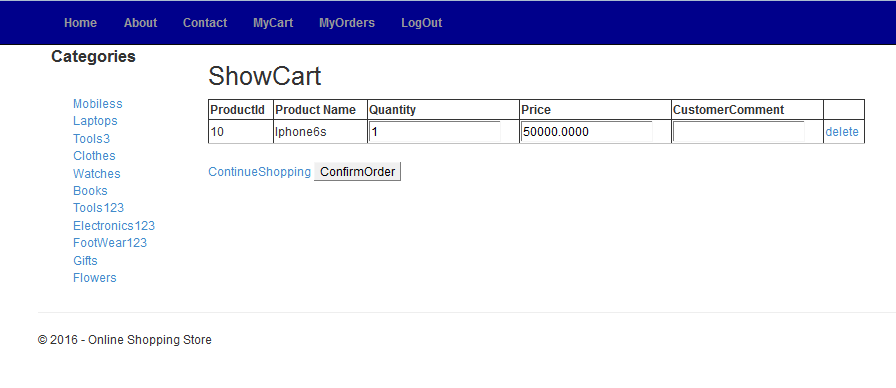
GetProductDetails.cshtml:

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select Catelog Controller->Double click on it-> GetProductDetails Method->Right Click On GetProductDetails Method->Add View-> View name-GetProductDetails .cshtml->Click Add.

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Product  @{  ViewBag.Title = "ProductDetails";  Layout = "~/Views/Shared/UserLayout.cshtml";  }  <h3>Product Details</h3>  <div class="row">  <div class="col-md-4">  <img class="Zoom" style="height:250px;width:250px" src=@ViewBag.ImagePath alt="Loading.." data-zoom-image="@ViewBag.ImagePath" />  </div>  <div class="col-md-6">  <h4>Product Name: @Html.DisplayFor(model => model.ProductName)</h4><br />  <h4>Description:@Html.DisplayFor(model => model.Description)</h4><br />  <h4>Price:</h4><h4 style="font-family:cursive;color:red">@Html.DisplayFor(model => model.Price)</h4>  </div>  </div>  <p style="padding-left:10%"><a href="@Url.Action("AddToCart", "ShoppingCart", new {productId=@Model.PKProductId,price=@Model.Price,quantity=@Model.Quantity })"><input type="submit" value="BuyNow" name="btnBuyNow" class="btn-primary" /></a></p>  @section Scripts{  <script src="~/Scripts/jquery.elevatezoom.js"></script>  <script type="text/javascript">  $(".Zoom").elevateZoom();  </script>  } |

33. Go to Solution Explorer->Project->Controller→ Right Click On Controller →Add Controller → ControllerName=ShoppingCartController→Select Template As Empty MVC Controller →Click Add.

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVCUsingLinq.Controllers  {  [Authorize]  public class ShoppingCartController : Controller  {  //  // GET: /ShowCart  HttpCookie CartCookie;  public ActionResult AddToCart(int? productId=0, decimal? price=0, int? quantity=0)  {  if (productId != 0)  {  CartCookie = Request.Cookies[Helper.UserName];  if (CartCookie == null)  {  CartCookie = new HttpCookie(Helper.UserName);  CartCookie.Expires = DateTime.MaxValue;  }  quantity = 1;  CartCookie.Values[productId.ToString()] = quantity.ToString() + '^' + price.ToString();  Response.Cookies.Add(CartCookie);  }  return RedirectToAction("ShowCart");  }  public ActionResult GetShowCartDetails(int id, int total, int qty, string customerComment)  {  CartCookie = Request.Cookies[Helper.UserName];  CartCookie.Values[id.ToString()] = qty.ToString() + '^' + total.ToString() + '^' + customerComment;  Response.Cookies.Add(CartCookie);  return RedirectToAction("ShowCart");  }  public ActionResult ShowCart()  {  return View();  }  public ActionResult ShowCartProductRemove(int deleteId)  {  ViewBag.Id = Helper.UserId;  HttpCookie CartCookie = Request.Cookies[Helper.UserName];  CartCookie.Values.Remove(deleteId.ToString());  Response.Cookies.Add(CartCookie);  return RedirectToAction("ShowCart");  }  }  } |



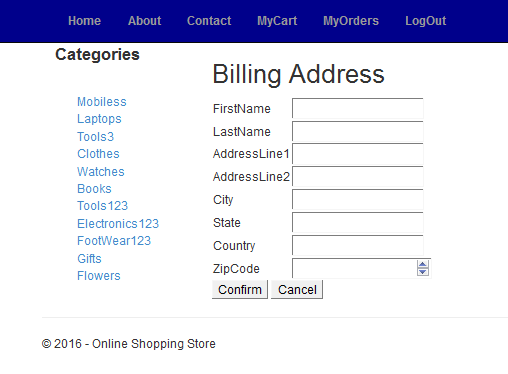
ShowCart.cshtml:

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select ShoppingCartController->Double click on it-> ShowCart Method->Right Click On ShowCart Method->Add View-> View name-ShowCart .cshtml->Click Add.

|  |
| --- |
| @model OnlineShoppingStoreMVCUsingLinq.Product  @using OnlineShoppingStoreMVCUsingLinq.BO  @{ProductBO objProductBO = new ProductBO();}  @{  ViewBag.Title = "ShowCart";  Layout = "~/Views/Shared/UserLayout.cshtml";  int availableProducts = 0;  int availableQuantity = 0;  }  <h2>ShowCart</h2>  @using (Html.BeginForm())  {  HttpCookie CartCookie = Request.Cookies[Helper.UserName];  <table class="table table-bordered table-responsive">  <tr>  <th>ProductId</th>  <th>ProductName</th>  <th>Quantity</th>  <th>Price</th>  <th>Totalprice</th>  <th>CustomerComment</th>    </tr>  @if (CartCookie != null)  {  foreach (var item in CartCookie.Values)  {  Product product = objProductBO.GetProduct(Convert.ToInt32(item));  if (product != null)  {  availableProducts = product.PKProductId;  availableQuantity = product.Quantity;  string cookieValue = CartCookie[item.ToString()].ToString();  <tr>  <td>@(product.PKProductId)</td>  <td>@(product.ProductName)</td>  <td><input type="text" name="txtQuantity" id="Quantity" value="@cookieValue.Split('^')[0]" class="quantityClass" /></td>  <td><input class="priceClass" type="text" name="txtPrice" id="Price" value="@product.Price" readonly="readonly" /></td>  <td><input class="total" type="text" name="txtTotal" value="@cookieValue.Split('^')[1]" /></td>  <td><input type="text" name="txtCustomerComment" class="CustomerComment" /></td>  <td>  <a href="@Url.Action("ShowCartProductRemove", "ShoppingCart", new { deleteId = product.PKProductId })">  delete  </a>  </td>  </tr>  }  }  }  </table>  <br />  <a href="@Url.Action("Index", "Home")">ContinueShopping </a>  if (availableProducts != 0)  {  <a href="@Url.Action("PlaceOrder", "Confirm")"><input type="button" value="ConfirmOrder" name="btnConfirmOrder" class="btn-primary"/></a>  }  }  @section Scripts{  <script type="text/javascript">  $(document).ready(function () {  $('.quantityClass').change(function (e) {  var id = "@availableProducts";  var availableqty = "@availableQuantity"  var qty = $(this).val();  var price = $(this).parent().next().find(".priceClass").val();  if (qty <= availableqty) {  var total = qty \* price;  $(this).parent().nextAll().find(".total").val(total);  var customerComment = $(this).parent().nextAll().find(".CustomerComment").val();  $.ajax({  url: "GetShowCartDetails",  data: { id: id, total: total, qty: qty, customerComment: customerComment },  type: 'Get',  }).success(function (data) {  })  }  else  {  alert("Available quantity is: "+availableqty);  }  });  });  $(document).ready(function () {  $(".CustomerComment").change(function (e) {  var id = "@availableProducts";  var qty = $(this).parent().prevAll().find(".quantityClass").val();  var price = $(this).parent().prevAll().find(".priceClass").val();  var total = qty \* price;  var customerComment = $(this).val();  $.ajax({  url: "GetShowCartDetails",  data: { id: id, total: total, qty: qty, customerComment: customerComment },  type: 'Get',  }).success(function (data) {  })  });  });  </script>  } |

35. Go To Solution Explorer->Project->Controllers → Right Click On Controller →Add Controller → ControllerName=ConfirmOrderController→Select Template As Empty MVC Controller →Click Add.

|  |
| --- |
| using OnlineShoppingStoreMVCUsingLinq.BO;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVCUsingLinq.Controllers  {  [Authorize]  public class ConfirmController : Controller  {  BillingAddressBO objBillingAddressBO = new BillingAddressBO();  ProductBO objProductBO = new ProductBO();  Order objOrder = new Order();  OrderBO objOrderBO = new OrderBO();  OrderDetail objOrderDetail = new OrderDetail();  OrderDetailBO objOrderDetailBO = new OrderDetailBO();  OrderHistory objOrderHistory = new OrderHistory();  OrderHistoryBO objOrderHistoryBO = new OrderHistoryBO();  //  // GET: /Confirm/  public ActionResult PlaceOrder()  {  return View();  }  [HttpPost, ActionName("PlaceOrder")]  public ActionResult ConfirmOrder(BillingAddress objBillingAddress)  {  objBillingAddressBO.InsertBillingAddress(objBillingAddress);  int addressId = objBillingAddress.PKAddressId;  HttpCookie CartCookie = Request.Cookies[Helper.UserName];  foreach (var items in CartCookie.Values)  {  string cookieValue = CartCookie[items.ToString()].ToString();  var product = objProductBO.GetProduct(Convert.ToInt32(items));  objOrder.FKProductId = product.PKProductId;  objOrder.FKBillingAddressId = addressId;  objOrder.Status = "Processing";  objOrder.CustomerComment = cookieValue.Split('^')[2];  objOrderBO.InsertOrder(objOrder);  objOrderDetail.FKProductId = product.PKProductId;  objOrderDetail.FKOrderId = objOrder.PKOrderId;  objOrderDetail.Cost=Convert.ToDecimal(cookieValue.Split('^')[1]);  objOrderDetail.Quantity = Convert.ToInt32(cookieValue.Split('^')[0]);  objOrderDetailBO.InsertOrderDetail(objOrderDetail);  objOrderHistory.FKOrderId = objOrder.PKOrderId;  objOrderHistory.Status = objOrder.Status;  objOrderHistoryBO.InsertOrderHistory(objOrderHistory);  }  CartCookie.Expires = DateTime.Now.AddDays(-1);  Response.Cookies.Add(CartCookie);  return RedirectToAction("GetOrderDetails","MyOrder");  }  }  } |



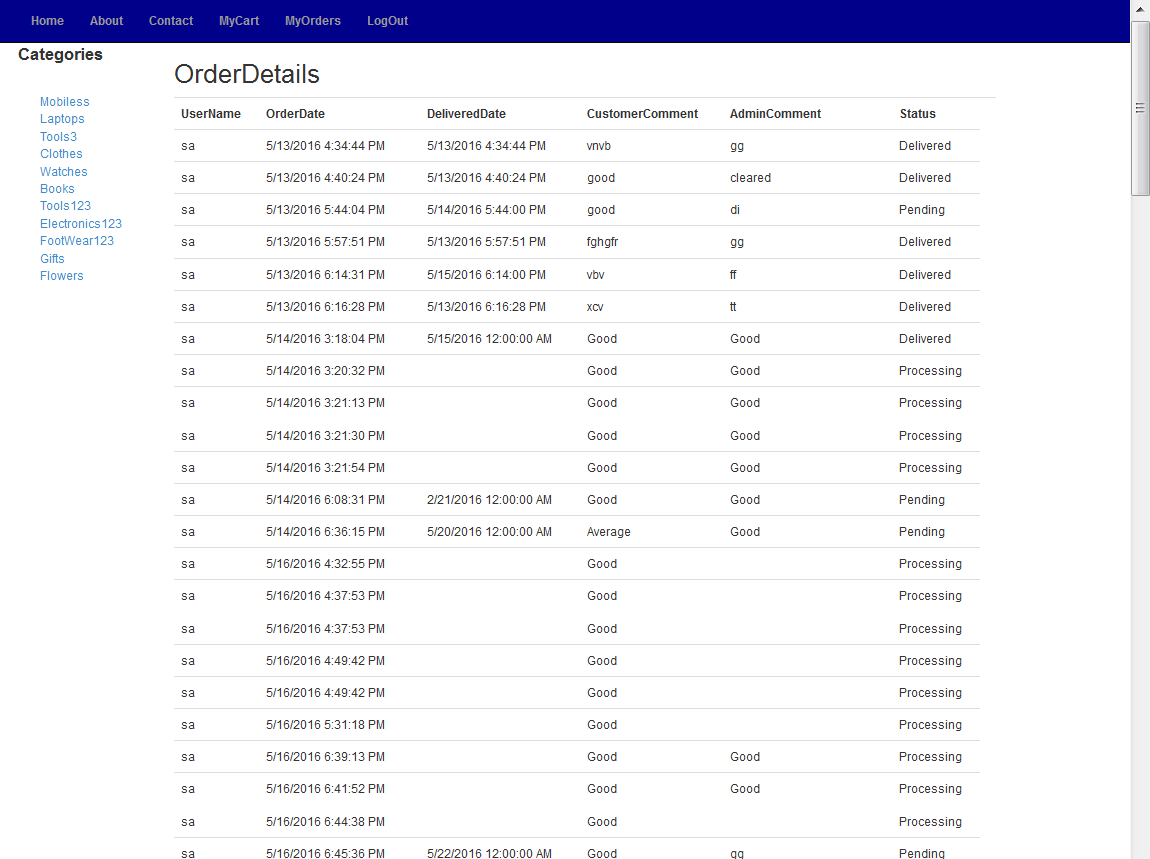
PlaceOrder.cshtml:

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select ConfirmOrderController->Double click on it-> ConfirmOrder Method->Right Click On ConfirmOrder Method->Add View-> View name- PlaceOrder.cshtml->Click Add.

|  |
| --- |
| @@model OnlineShoppingStoreMVCUsingLinq.BillingAddress  @{  ViewBag.Title = "PlaceOrder";  }  <h2>Billing Address</h2>  @using(Html.BeginForm())  {  <table>  <tr>  <td>  @Html.DisplayNameFor(model=>model.FirstName)  </td>  <td>  @Html.EditorFor(model=>model.FirstName)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model=>model.LastName)  </td>  <td>  @Html.EditorFor(model=>model.LastName)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model=>model.AddressLine1)  </td>  <td>  @Html.EditorFor(model=>model.AddressLine1)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model => model.AddressLine2)  </td>  <td>  @Html.EditorFor(model=>model.AddressLine2)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model=>model.City)  </td>  <td>  @Html.EditorFor(model=>model.City)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model=>model.State)  </td>  <td>  @Html.EditorFor(model=>model.State)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model=>model.Country)  </td>  <td>  @Html.EditorFor(model=>model.Country)  </td>  </tr>  <tr>  <td>  @Html.DisplayNameFor(model=>model.ZipCode)  </td>  <td>  @Html.EditorFor(model=>model.ZipCode)  </td>  </tr>  </table>  <input type="submit" value="Confirm" name="btnSubmit"/>  <a href="@Url.Action("AddToCart","ShoppingCart")"><input type="button" value="Cancel" name="btnCancel"/></a>  } |

1. Go To Solution Explorer->Project-> Controllers → Right Click On Controller →Add Controller → ControllerName=MyOrderController→Select Template As Empty MVC Controller →Click Add.

|  |
| --- |
| uusing OnlineShoppingStoreMVCUsingLinq.BO;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVCUsingLinq.Controllers  {  [Authorize]  public class MyOrderController : Controller  {  OrderBO objOrderBO = new OrderBO();  //  // GET: /MyOrder/  public ActionResult GetOrderDetails()  {  return View(objOrderBO.GetOrders(null,null,Helper.UserName));  }  }  } |



GetOrderDetails.cshtml:

1. GoTo Solution Explorer->Project->Select Controller Folder->In Controller, Select MyOrderController->Double click on it-> GetOrderDetails Method->Right Click On GetOrderDetails Method->Add View-> View name- GetOrderDetails.cshtml->Click Add.

|  |
| --- |
| @model IEnumerable<OnlineShoppingStoreMVCUsingLinq.Order>  @{  ViewBag.Title = "GetOrderDetails";  }  <h2>OrderDetails</h2>  <table class="table">  <tr>  <th>  @Html.DisplayNameFor(model => model.UserName)  </th>  <th>  @Html.DisplayNameFor(model => model.OrderDate)  </th>  <th>  @Html.DisplayNameFor(model => model.DeliveredDate)  </th>  <th>  @Html.DisplayNameFor(model => model.CustomerComment)  </th>  <th>  @Html.DisplayNameFor(model => model.AdminComment)  </th>  <th>  @Html.DisplayNameFor(model => model.Status)  </th>  <th></th>  </tr>  @foreach (var item in Model) {  <tr>  <td>  @Html.DisplayFor(modelItem => item.UserName)  </td>  <td>  @Html.DisplayFor(modelItem => item.OrderDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.DeliveredDate)  </td>  <td>  @Html.DisplayFor(modelItem => item.CustomerComment)  </td>  <td>  @Html.DisplayFor(modelItem => item.AdminComment)  </td>  <td>  @Html.DisplayFor(modelItem => item.Status)  </td>  </tr>  }  </table> |

1. Go to Solution Explorer->Project-> Add a Folder name =Custom Filters-.>Under the Custom Filter Folder-> Add the class name=CustomExceptionFilterAttribute

|  |
| --- |
| using OnlineShoppingStoreMVCUsingLinq.BO;  using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  namespace OnlineShoppingStoreMVCUsingLinq.Custom\_Filters  {  public class CustomExceptionFilterAttribute : ActionFilterAttribute, IExceptionFilter  {  ErrorHistory objErrorHistory = new ErrorHistory();  ErrorHistoryBO objErrorHistoryBO = new ErrorHistoryBO();  public void OnException(ExceptionContext filterContext)  {  if (!filterContext.ExceptionHandled)  {  var controllerName = (string)filterContext.RouteData.Values["controller"];  var actionName = (string)filterContext.RouteData.Values["action"];  var model = new HandleErrorInfo(filterContext.Exception, controllerName, actionName);  // Code to log the error in database goes here..\  objErrorHistory.ActionName = actionName;  objErrorHistory.ControllerName = controllerName;  objErrorHistory.ErrorMessage = model.Exception.Message;  objErrorHistory.IsActive = true;  objErrorHistoryBO.InsertErrorHistory(objErrorHistory);  if (string.IsNullOrEmpty(Helper.UserData) || Helper.RoleId == 2)  {  filterContext.Result = new ViewResult  {  ViewName = "~/Views/Shared/Error.cshtml",  MasterName = "~/Views/Shared/UserLayout.cshtml",  ViewData = new ViewDataDictionary<HandleErrorInfo>(model),  };  filterContext.ExceptionHandled = true;  }  else  {  filterContext.Result = new ViewResult  {  ViewName = "~/Areas/Admin/Views/Shared/Error.cshtml",  MasterName = "~/Areas/Admin/Views/Shared/\_Layout.cshtml",  ViewData = new ViewDataDictionary<HandleErrorInfo>(model),  };  filterContext.ExceptionHandled = true;  }  }  }  }  } |

1. Go to solution explorer-> Project->Areas-> Admin-> Views-> Shared-> Add a View name=Error.cshtml

|  |
| --- |
| @model System.Web.Mvc.HandleErrorInfo  @{  ViewBag.Title = "Error";  }  <h2 class="text-danger">Error</h2>  <h2>Exception details</h2>  <p>  Controller Name: @Model.ControllerName  <br>  Action Name: @Model.ActionName  <br />  Exception: @Model.Exception.Message  </p> |

1. Go to Solution explorer-> Project-> Views → Shared->select Error.cshtml-. Change the code as below

|  |
| --- |
| @model System.Web.Mvc.HandleErrorInfo  @{  ViewBag.Title = "Error";  }  <h2 class="text-danger">Error. </h2>  <br />  <h2 class="text-danger">An error occurred while processing your request.</h2>  <br/>  <div class="row">  <div class="col-sm-10 img-responsive">  <img src="/Images/error.jpg" style="max-height:300px"/>  </div>  </div> |

1. To handle the Gobal Exception (Go to solution explorer->Project->go to App\_Start->Select Filterconfig.cs Change the code below)

|  |
| --- |
| using System.Web;  using System.Web.Mvc;  using OnlineShoppingStoreMVCUsingLinq.Custom\_Filters;  namespace OnlineShoppingStoreMVCUsingLinq  {  public class FilterConfig  {  public static void RegisterGlobalFilters(GlobalFilterCollection filters)  {  filters.Add(new HandleErrorAttribute());  filters.Add(new CustomExceptionFilterAttribute());  }  }  } |

Go to solution explorer->Project->go to App\_Start->Select Routeconfig.cs->Change the code as below

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  using System.Web.Routing;  namespace OnlineShoppingStoreMVCUsingLinq  {  public class RouteConfig  {  public static void RegisterRoutes(RouteCollection routes)  {  routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");  routes.MapRoute(  name: "Default",  url: "{controller}/{action}/{id}",  defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }  );  }  }  } |