Table of Contents

[Section 15: User Management 2](#_Toc145876252)

[15.1 Display List of User 2](#_Toc145876253)

[15.2 Lock or Unlock User 6](#_Toc145876254)

[15.3 Manage User Roles 8](#_Toc145876255)

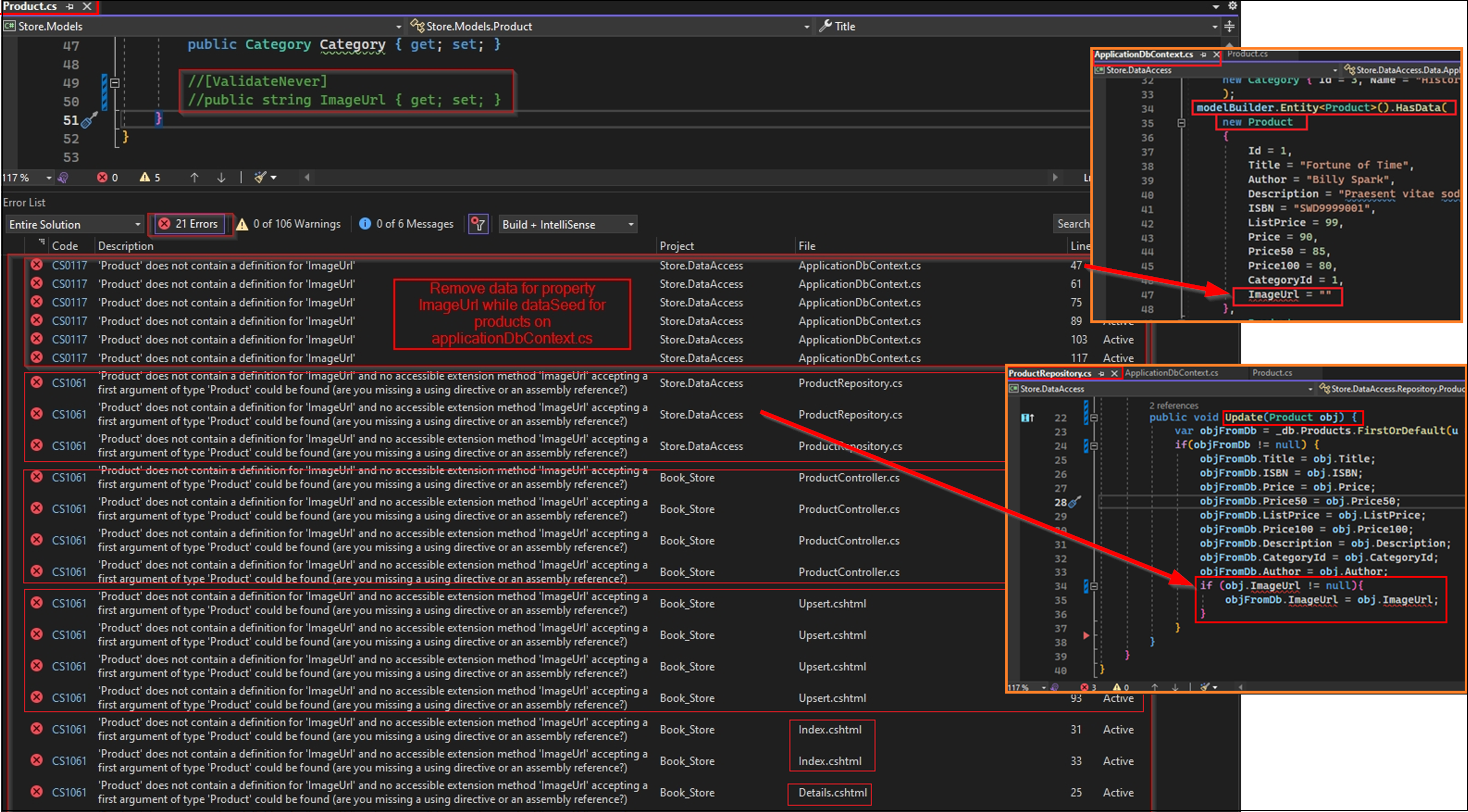
# Section 16: Add Multiple Product Image

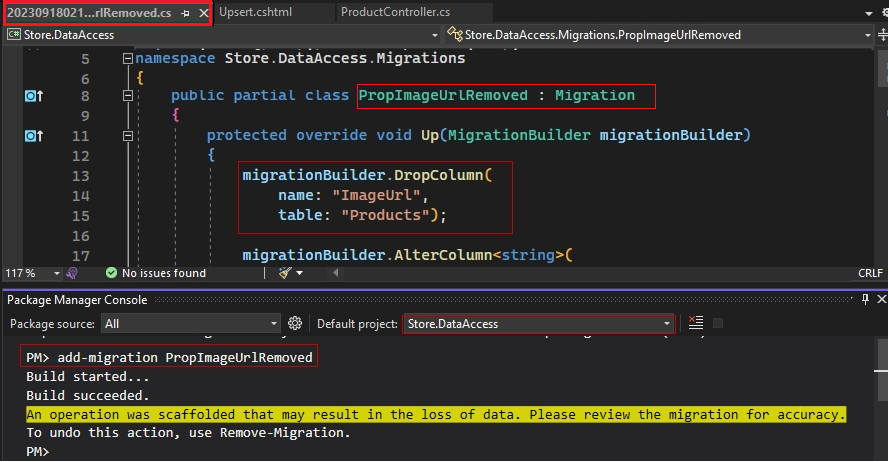
In this section we are going modify Product Model, and all related components to add feature that Admin can add more than one image files for a product and display them on product description view.

## 15.1 Modify Product Model and New Model for Images

To handle multiple images for a single product we can have a separate model instead of a property ImageUrl in product model.

### 15.1.1 Delete Property ImageUrl from Product Model

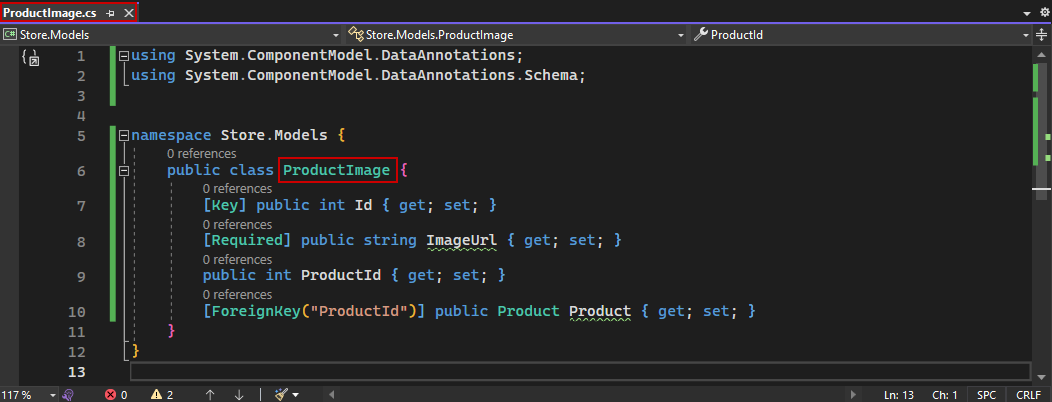
Since there will be new model for images, lets delete ImageUrl property from Product model. Then build the solution and of course it will not work due to broken references but error description helps us to locate references from all places.

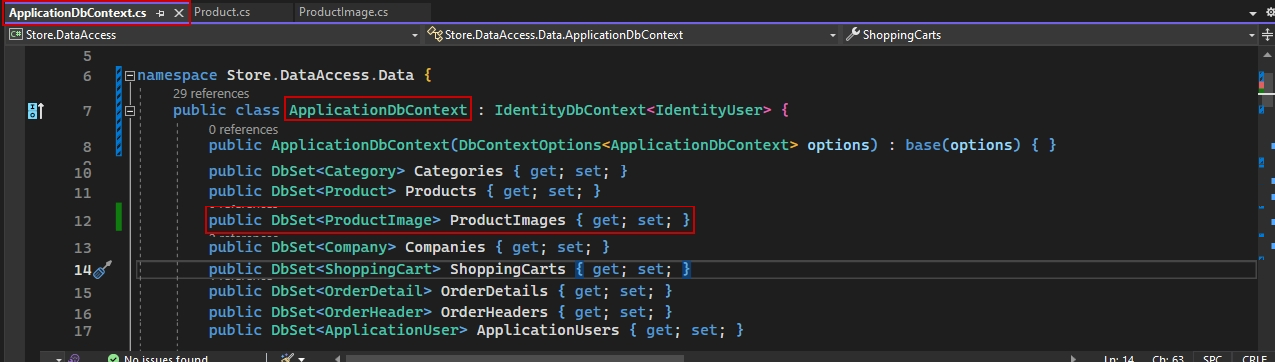
Next comment or delete all the references to properties ImageUrl. Then build the solution again to find if there is any more reference left. Once we remove a property from model and all references, we need to add new migration and update-database to reflect the change made on Product model.

### 15.1.2 Add New Model for Product Image

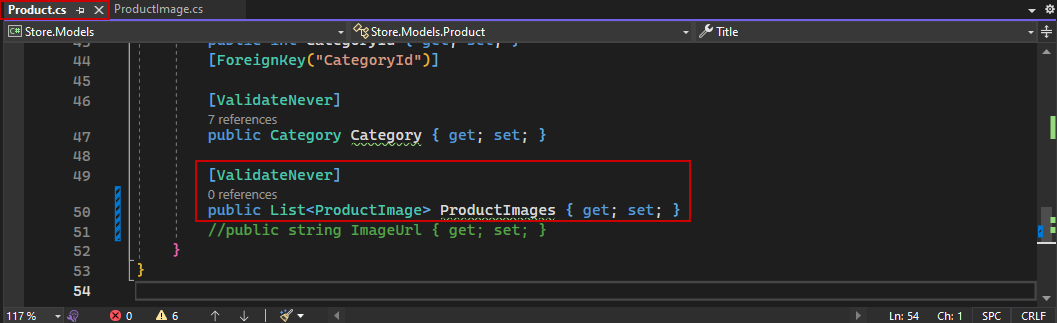
Now we need to add a new model that will be responsible for adding or managing all the images for a product let call it ProductImage and beside primary key Id it will have properties,

* ImageUrl of type string
* Product Id of type integer (as each image must be assigned to a product)
* Product of type Product (it a navigation property for foreign key relation)



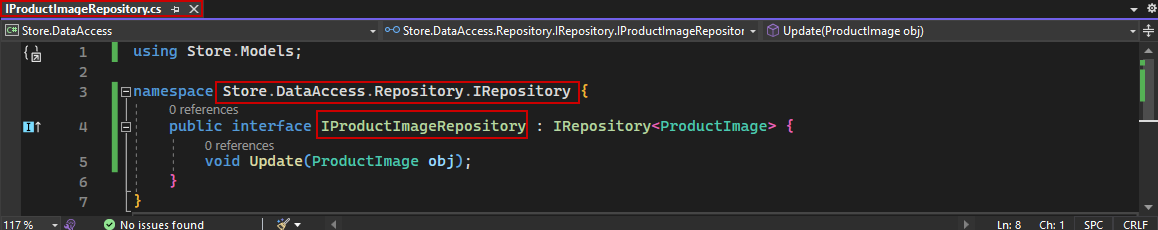
Next, we need to add this model to ApplicationDbContext, 

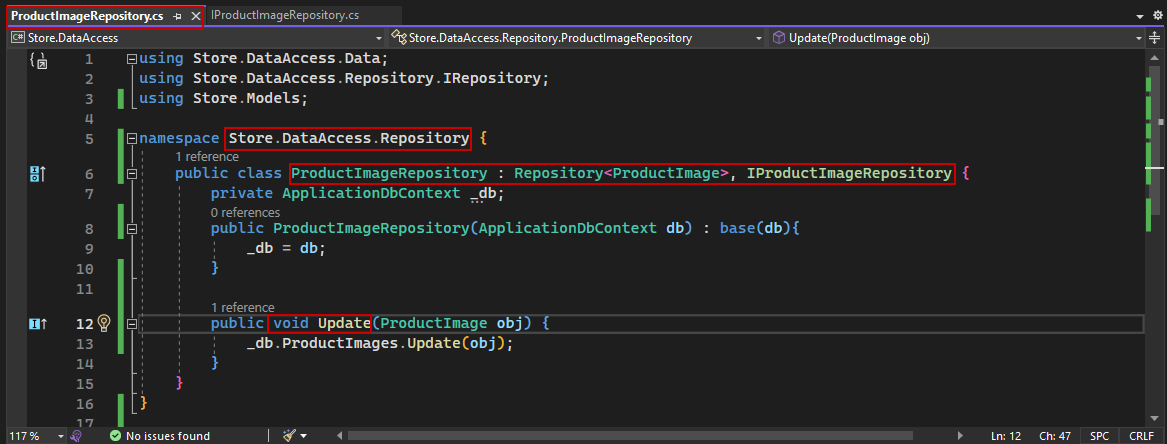
### 15.1.3 Add Property for List of Images to Product Model

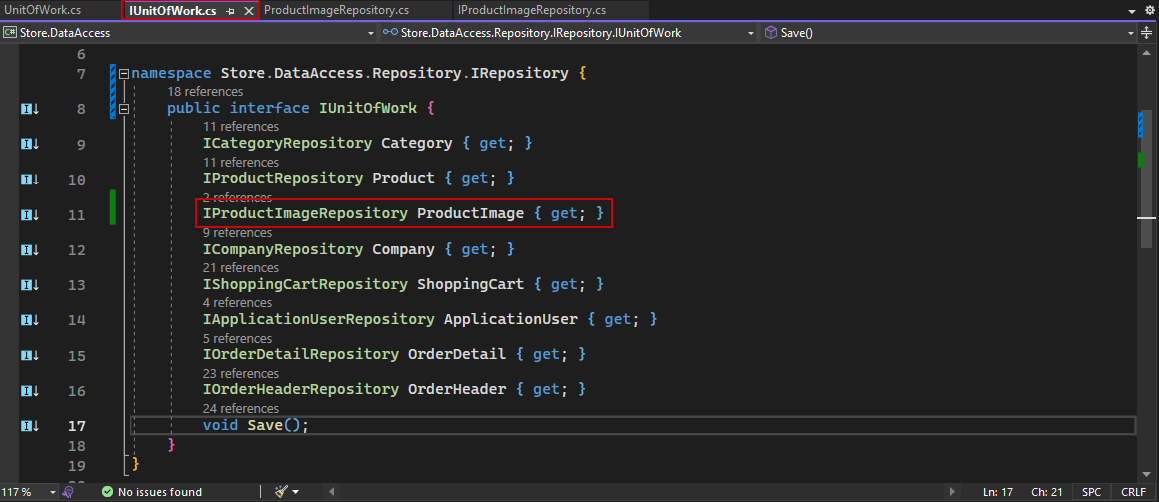
A single product can have multiple images, so in the product model, we can create list of product image where we can carry or load all the images for our product. So that way we are telling entity framework core about the one-to-many relation that we have between product and product image. 

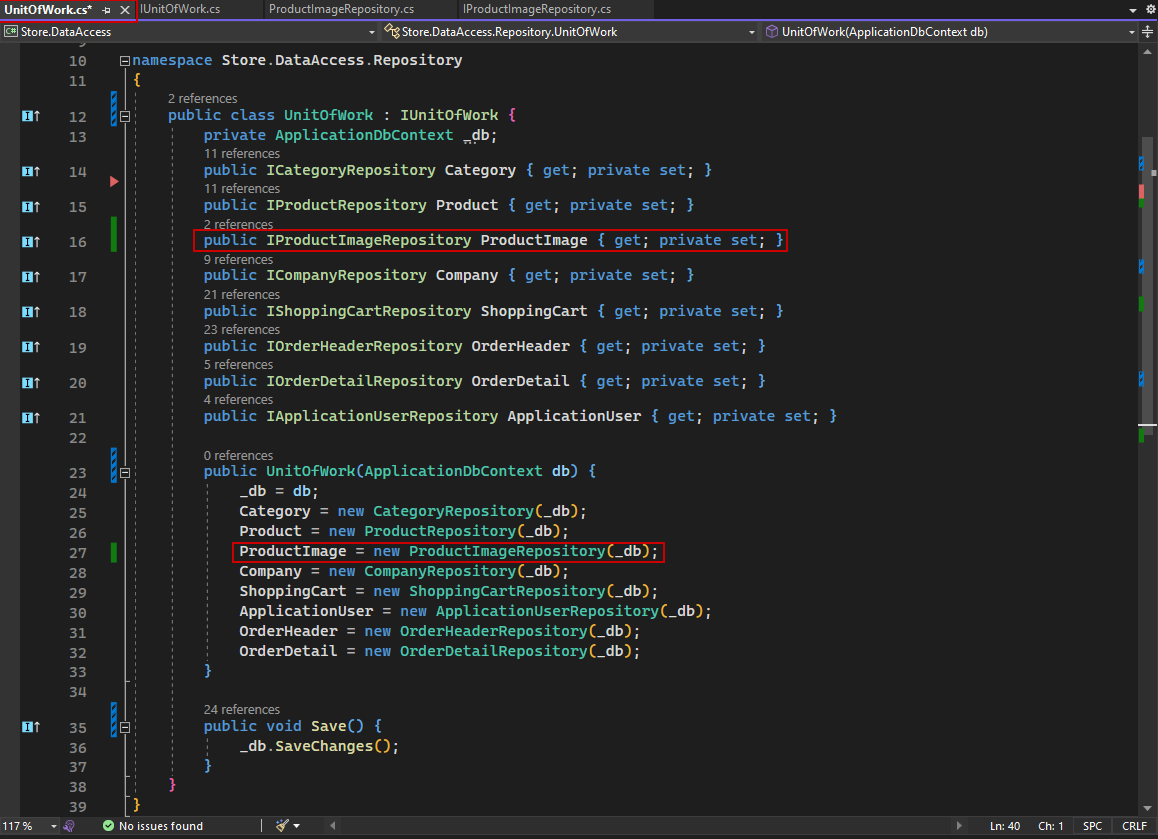
With that configured, all we must do is add a migration to add ProductImage to database.

### 15.1.4 Add Repositories for ProductImage Model and Modify IUnitOfWork

Since we have designed our project in the way to use repository to access any of our core model, lets add a repository for product image and then we will have to add that to unit of work. To save time we can copy any repository and modify it for ProductImage, for now let’s do it with ICategoryRepository.

Next implement of above interface as,

Finally, we must configure unit of work. In the interface here, we just need to create I product, image repository. 

And in the implementation here, let me create public IProductImageRepository and initialize that. 

Hence, we have added the product image to our unit of work.