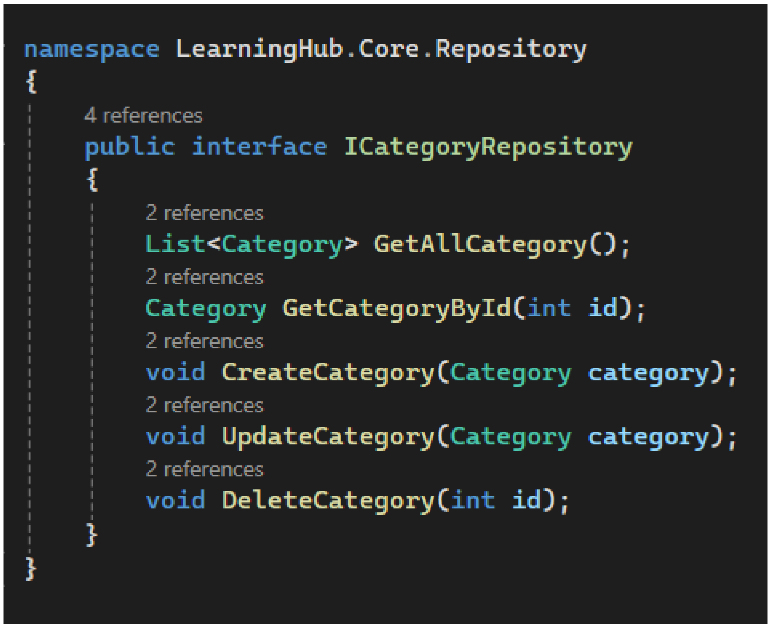
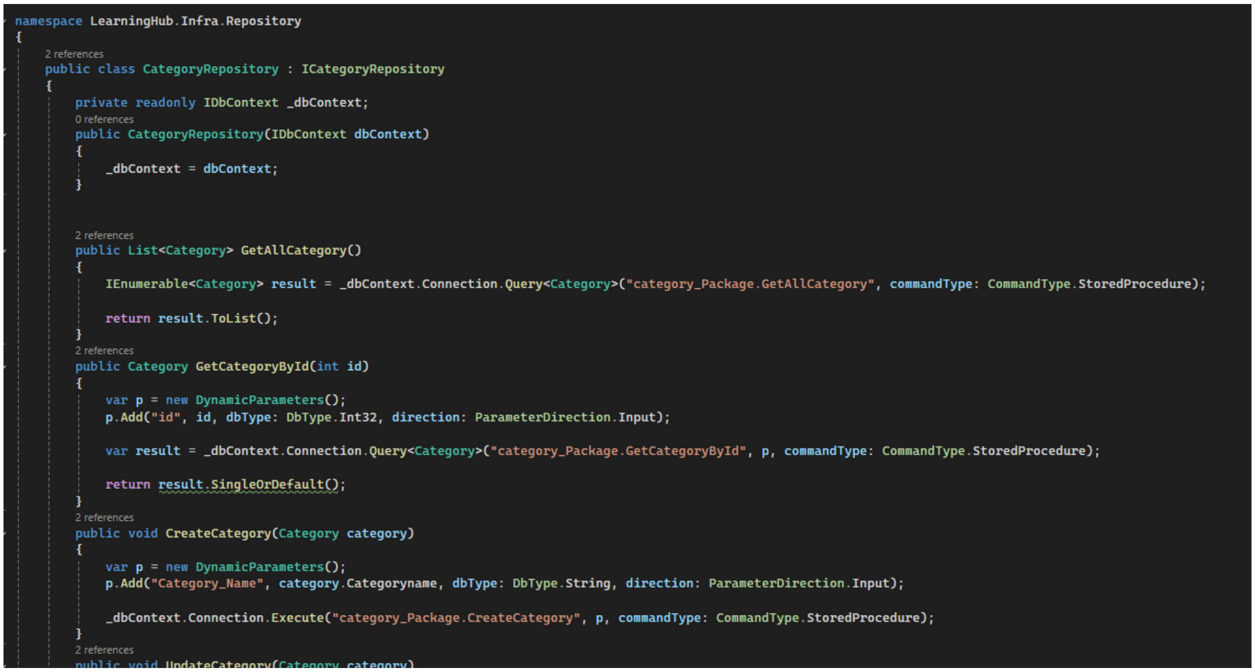
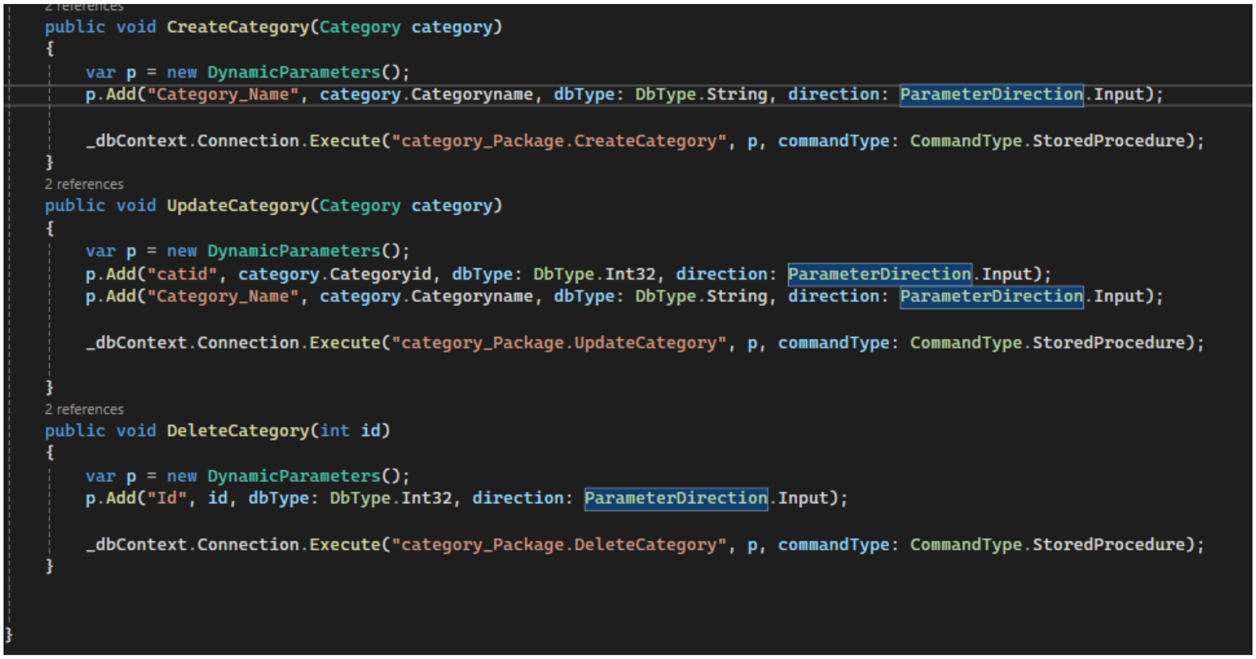
Task 03 API



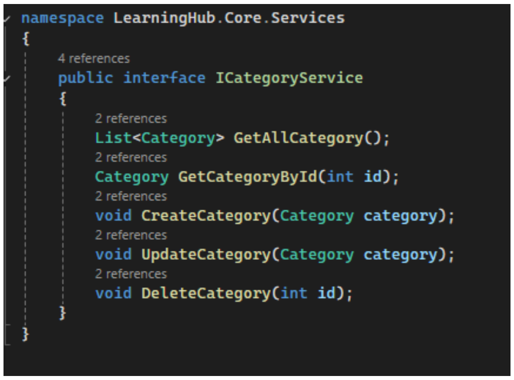
```
namespace LearningHub.Core.Repository
{
    4 references
    public interface ICategoryRepository
    {
        2 references
        List<Category> GetAllCategory();
        2 references
        Category GetCategoryById(int id);
        2 references
        void CreateCategory(Category category);
        2 references
        void UpdateCategory(Category category);
        2 references
        void DeleteCategory(int id);
    }
}
```



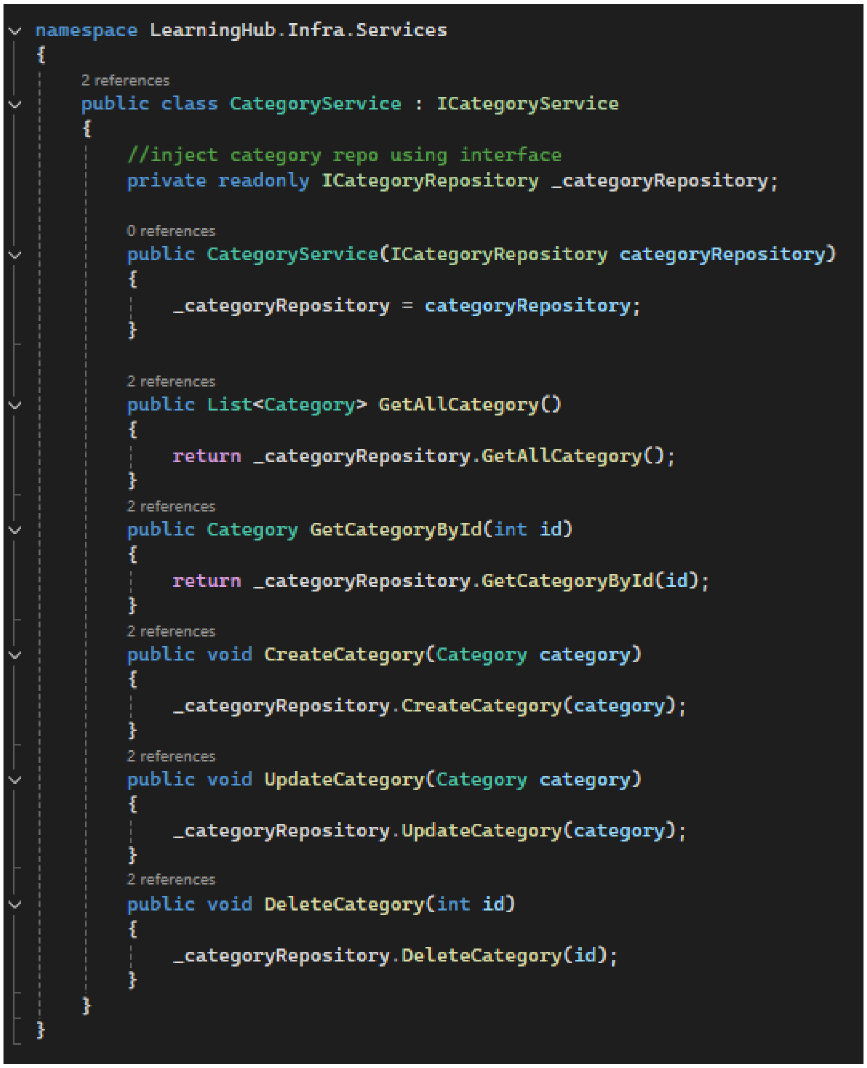
```
namespace LearningHub.Infra.Repository
{
    2 references
    public class CategoryRepository : ICategoryRepository
    {
        private readonly IDbContext _dbContext;
        0 references
        public CategoryRepository(IDbContext dbContext)
        {
            _dbContext = dbContext;
        }


        2 references
        public List<Category> GetAllCategory()
        {
            IEnumerable<Category> result = _dbContext.Connection.Query<Category>("category_Package.GetAllCategory", commandType: CommandType.StoredProcedure);

            return result.ToList();
        }
        2 references
        public Category GetCategoryById(int id)
        {
            var p = new DynamicParameters();
            p.Add("id", id, dbType: DbType.Int32, direction: ParameterDirection.Input);

            var result = _dbContext.Connection.Query<Category>("category_Package.GetCategoryById", p, commandType: CommandType.StoredProcedure);

            return result.SingleOrDefault();
        }
        2 references
        public void CreateCategory(Category category)
        {
            var p = new DynamicParameters();
            p.Add("Category_Name", category.Categoryname, dbType: DbType.String, direction: ParameterDirection.Input);

            _dbContext.Connection.Execute("category_Package.CreateCategory", p, commandType: CommandType.StoredProcedure);
        }
        2 references
        public void UpdateCategory(Category category)
```



```
    2 references
    public void CreateCategory(Category category)
    {
        var p = new DynamicParameters();
        p.Add("Category_Name", category.Categoryname, dbType: DbType.String, direction: ParameterDirection.Input);

        _dbContext.Connection.Execute("category_Package.CreateCategory", p, commandType: CommandType.StoredProcedure);
    }
    2 references
    public void UpdateCategory(Category category)
    {
        var p = new DynamicParameters();
        p.Add("catid", category.Categoryid, dbType: DbType.Int32, direction: ParameterDirection.Input);
        p.Add("Category_Name", category.Categoryname, dbType: DbType.String, direction: ParameterDirection.Input);

        _dbContext.Connection.Execute("category_Package.UpdateCategory", p, commandType: CommandType.StoredProcedure);

    }
    2 references
    public void DeleteCategory(int id)
    {
        var p = new DynamicParameters();
        p.Add("Id", id, dbType: DbType.Int32, direction: ParameterDirection.Input);

        _dbContext.Connection.Execute("category_Package.DeleteCategory", p, commandType: CommandType.StoredProcedure);
    }


}
```



```
namespace LearningHub.Core.Services
{
    4 references
    public interface ICategoryService
    {
        2 references
        List<Category> GetAllCategory();
        2 references
        Category GetCategoryById(int id);
        2 references
        void CreateCategory(Category category);
        2 references
        void UpdateCategory(Category category);
        2 references
        void DeleteCategory(int id);
    }
}
```



```
namespace LearningHub.Infra.Services
{
    2 references
    public class CategoryService : ICategoryService
    {
        //inject category repo using interface
        private readonly ICategoryRepository _categoryRepository;

        0 references
        public CategoryService(ICategoryRepository categoryRepository)
        {
            _categoryRepository = categoryRepository;
        }

        2 references
        public List<Category> GetAllCategory()
        {
            return _categoryRepository.GetAllCategory();
        }
        2 references
        public Category GetCategoryById(int id)
        {
            return _categoryRepository.GetCategoryById(id);
        }
        2 references
        public void CreateCategory(Category category)
        {
            _categoryRepository.CreateCategory(category);
        }
        2 references
        public void UpdateCategory(Category category)
        {
            _categoryRepository.UpdateCategory(category);
        }
        2 references
        public void DeleteCategory(int id)
        {
            _categoryRepository.DeleteCategory(id);
        }
    }
}
```