Models –

```
public enum AuditAction
{
    Created,
    Updated,
    Deleted
}
public class AuditRequest
{
    public object BeforeChange { get; set; }
    public object AfterChange { get; set; }
    public AuditAction Action { get; set; }
    public string EntityName { get; set; }
    public string UserId { get; set; }
}
public class AuditResult
{
    public string EntityName { get; set; }
    public AuditAction Action { get; set; }
    public string UserId { get; set; }
    public DateTime Timestamp { get; set; }
    public List<ChangedField> Changes { get; set; } = new();
}
public class ChangedField
{
    public string FieldName { get; set; }
    public object? OldValue { get; set; }
    public object? NewValue { get; set; }
}
```

Service Layer –

```
public interface IAuditTrailInfoService
{
   AuditResult GenerateAudit(AuditRequest request);
}
public class AuditTrailInfoService : IAuditTrailInfoService
{
   public AuditResult GenerateAuditInfo (AuditRequest request)
   {
      var result = new AuditResult
      {
         EntityName = request.EntityName,
         Action = request.Action,
         UserId = request.UserId,
         Timestamp = DateTime.UtcNow
      };

      if (request.Action == AuditAction.Created)
      {
         foreach (var item in request.AfterChange.GetType().GetProperties())
         {
            var newVal = item.GetValue(request.AfterChange);
            result.Changes.Add(new ChangedField
            {
               FieldName = item.Name,
               OldValue = null,
               NewValue = newVal
            });
         }
      }
      else if (request.Action == AuditAction.Deleted)
```

```
{
    foreach (var item in request.BeforeChange.GetType().GetProperties())
    {
        var oldVal = item.GetValue(request.BeforeChange);
        result.Changes.Add(new ChangedField
        {
            FieldName = item.Name,
            OldValue = oldVal,
            NewValue = null
        });
    }
}
else if (request.Action == AuditAction.Updated)
{
    var props = request.BeforeChange.GetType().GetProperties();

    foreach (var item in props)
    {
        var oldVal = item.GetValue(request.BeforeChange);
        var newVal = item.GetValue(request.AfterChange);

        if (!object.Equals(oldVal, newVal))
        {
            result.Changes.Add(new ChangedField
            {
                FieldName = item.Name,
                OldValue = oldVal,
                NewValue = newVal
            });
        }
    }
```

```
        }


        return result;
    }
}
```

Controller –

```
[ApiController]
[Route("api/[controller]")]
public class AuditTrailInfoController : ControllerBase
{
    private readonly IAuditTrailInfoService _auditService;


    public AuditTrailInfoController(IAuditTrailInfoService auditService)
    {
        _auditService = auditService;
    }


    [HttpPost("generate")]
    public IActionResult GenerateAuditInfo ([FromBody] AuditRequest request)
    {
        var result = _auditService.GenerateAuditInfo(request);
        return Ok(result);
    }
}
```

Program.cs –

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
builder.Services.AddScoped<IAuditTrailInfoService, AuditTrailInfoService>();

var app = builder.Build();
app.UseRouting();
app.UseEndpoints(endpoints =>
{
   endpoints.MapControllers();
});
app.Run();
```