LINQ QUERY USAGE

1. Filtering & Searching Grievances

a. Filtering by Citizen

_context.Grievances

   .Where(g => g.CitizenId == userId)

Used in: GrievanceService.GetByCitizen

b. Filtering by Department

_context.Grievances

   .Where(g => g.DepartmentId == departmentId)

Used in:

- GrievanceService.GetByDepartment
- GrievanceRepository.GetByDepartment

c. Filtering by Escalation Status

.Where(g => g.IsEscalated)

Used in:

- GrievanceRepository.GetEscalatedGrievances
- GrievanceService.GetEscalatedGrievances

Requirement satisfied: Filtering & searching grievances

2. Grouping by Department and Category

a Grouping by Department

_context.Grievances

   .Include(g => g.Department)

   .GroupBy(g => g.Department.Name)

a. *Used in:*

- ReportService.GetGrievancesByDepartment
- ReportService.GetDepartmentPerformance

Grouping by Category

_context.Grievances

  .Include(g => g.Category)

  .GroupBy(g => g.Category.Name)

Used in:

- ReportService.GetGrievancesByCategory

Requirement satisfied: Grouping by department & category

3. Aggregation Queries

a) Grievance Count by Status

_context.Grievances

  .GroupBy(g => g.Status)

  .Select(g => new StatusSummaryDto

  {

    Status = g.Key,

    Count = g.Count()

  })

Used in: ReportService.GetStatusSummary

✓ Uses:

- GroupBy
- Count
- Projection (Select)

b) Average Resolution Time

g.Average(x =>

  (x.ResolvedAt.Value - x.CreatedAt).TotalHours

)

Used in: ReportService.GetDepartmentPerformance

Uses:

- DateTime arithmetic
- Average() aggregation

- LINQ projection

c) Department-wise Performance Metrics

.GroupBy(g => g.Department.Name)

.Select(g => new DepartmentPerformanceDto

{

   DepartmentName = g.Key,

   TotalGrievances = g.Count(),

   ResolvedGrievances = g.Count(),

   AverageResolutionHours = g.Average(...)

})

Used in: ReportService.GetDepartmentPerformance

Metrics computed:

- Total grievances
- Resolved grievances
- Average resolution time