**Task Management System**

**1. Project Overview**

**Purpose:**  
Develop a simple task management system using **Windows Forms (WinForms) in ASP.NET Core** with **Entity Framework Core**. This system allows a single user to manage personal tasks efficiently while practicing LINQ queries.

**Core Functionalities:**

* Add, update, delete, and view tasks.
* Filter and sort tasks.
* Track progress (Pending, In Progress, Completed).
* Search tasks by title or description.
* Generate reports on task status.

**3. Database Schema (Entities & Relationships)**

### ****Entities (Without Relationships)****

#### **Task Entity**

public class TaskItem

{

public int Id { get; set; }

public string Title { get; set; }

public string Description { get; set; }

public DateTime DueDate { get; set; }

public string Priority { get; set; } // Low, Medium, High

public Enum Status { get; set; } // Pending, In Progress, Completed

}

#### **User Entity**

public class User

{

public int Id { get; set; }

public string Name { get; set; }

public string Email { get; set; }

}

#### **Category Entity**

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

}

### ****Relationships Between Entities****

1. **User - TaskItem (One-to-Many)**
   * A **User** can have multiple **TaskItems**.
   * Each **TaskItem** belongs to a single **User**.
2. **Category - TaskItem (One-to-Many)**
   * A **Category** can have multiple **TaskItems**.
   * Each **TaskItem** belongs to a single **Category**.

**4. Features Breakdown**

**Basic Features**

* CRUD operations for tasks (Create, Read, Update, Delete).
* Filter tasks by status (Pending, Completed, etc.).
* Sort tasks by due date.

**Intermediate Features**

* Search tasks by title/description.
* Group tasks by priority.
* Show unique task priorities.
* Paginate tasks (e.g., 5 per page).

**Advanced Features**

* Track overdue tasks & send alerts.
* Generate reports for completed vs pending tasks.
* Calculate average task completion time.