Backend First project

Employee Management and Vacation Workflow System

This project aims to develop an employee management and vacation request workflow system for a company. The system will allow users to:

1. **View Employee Information** – Display key employee details, including name, department, position, and reporting structure.

2. **Manage Vacation Requests** – Employees can submit vacation requests, view their request history, and approve or decline requests for their subordinates.

First Objective: (**Database Design & Migration)**

Use **Code First development approach** in entity framework core to create workflow DB and apply migration, you are required to implement the following entities in your project.

1. **Department Entity class with:**
   a. Department Id (PK, identity).
   b. Department name (max 50 char, required).

2. **Position Entity class with:**
   a. Position Id (PK, identity)
   b. Position name (max 30 char, required).

3. **Employee Entity class with**:
   a. Employee number (max 6 char, required, PK and not identity column).
   b. Employee name (max 20 char, required).
   c. Department Id.
   d. Position Id.
   e. Gender Code (max 1 char such as: (M: Male, F: Female).
   f. Reported to employee number (max 6 char, nullable).
   g. Vacation days left (integer, and cannot exceed 24 days)
   h. Salary (decimals with max 2 digits after decimal points such as: 2500.50)

4. **Vacation type Entity class with:**
   a. Vacation type code (PK, 1 char, such as: S, U, A, O, B).
   b. Vacation type name (max 20 char such as: Sick, unpaid, annual, Day Off, Business Trip).

5. **Request State Entity class with:**
    a. State Id (PK, unique id)
    b. State name (10 char, required)

    ** example for states:

    1. Submitted. (mean request submitted with id: 1)
    2. Approved. (mean request approved by reported to employee with id: 2).
    3. Declined. (mean request declined by reported to employee with id: 3)

6. **Vacation Request Entity with:**
    a. Request Id (PK, identity)
    b. Request submission date (Date time, required)
    c. Description (max 100 char, required).
    d. Employee number.
    e. Vacation Type code (max **1** char)
    f. Start date (date only, required).
    g. End date (date only, required).
    h. Total vacation days (int, required).
    i. Request state Id (int, required).
    j. Approved by employee number (nullable)
    k. Declined by employee number (nullable).

**** make sure to include all required primary keys (PK) and Forign Keys (FK) also all relations needed between tables.**

Second Objective: (CRUD Operations Using Entity Framework Core)

After create database apply CRUD operation in EF core on **employee** entity as below:

1. **Add about new 20 departments** by use **DB context** and one save changes to database.

2. **Add 20 positions** by using **DB Set** and one save changes to database.

3. Use **constructor** in Employee entity class to add **(10)** employees (by default vacation days left = 24 days) then added to database.

4. Create method to update Employee main info and use <u>**employee number**</u> as unique key to find employee then update. (such as department, position, name, salary).

5. Create method to update vacation days balance after approve any vacation request which the logic of this method is to decrease employee vacation days left.

Third Objective: Apply CRUD operation in EF core on **Vacation request** full cycle as below:

1. **Submit new** vacation request action for employee and make sure to not overlap vacations for same employee.

2. System show pending vacation requests should employee take actions on and he has one of 2 options to take on as below:
   a. Approve vacation request by call method: **Approve**
   b. Decline vacation request by call method: **Decline.**

Fourth Objective:

Before execute this objective please make sure to fill database with **meaningful and large data (records in each table)** then use **LINQ** query to create get methods as below with consider performance of each query with huge data to make sure you write the query in the best way:

1. Get all employees in database and display employee number, name, department and salary.

2. Create method to Get employee by his unique number to return data as below (check figure):
   a. Employee number
   b. Employee name
   c. Department name
   d. Position name
   e. Reported to employee name.
   f. Total vacation days left

3. Use LINQ to Create method to get all employees have one or more pending vacation requests.

4. Use **LINQ** to get all history vacation requests (approved requests) for employee with return data as below:
   a. Vacation type
   b. Vacation description
   c. Request duration & total vacation days
   d. Approved by employee name.

5. Use LINQ to create get method to show all pending vacation requests employee should take action on with below data:
    a. Vacation description
    b. Employee number
    c. Employee name
    d. Submitted on date
    e. Vacation duration (such 2 weeks, 2 days).
    f. Start, end dates of vacation.
    g. Employee salary.

** Important Notes:

1. You Have **2 weeks** to complete all the requirements in this project.

2. the submission of project **at 15/02/2025 at 10:00 am**. and any submission after this time will be ignored **(no exceptions for anyone).**

3. Please make sure while submission on GitHub enter your **Full name (from 4 parts).**

4. make sure to share the project submission on your GitHub link and send this link to **Ayah** by email (a.hajali@skyits.com).

** feel free to ask about any objective and wish you all the best 😊.

Sky Academy

1. Wedyan Alswiti.
2. Ghaidaa Hammad.
3. Razan Alsamdi.