WaterlilyEmployee Project Plan

1. Understanding of the requirements

- **Employees CRUD:** Create, Read, Update, Delete employee records. Each employee has:
  - Id – int
  - Name – string
  - Email – string
  - Job Position – string

- **Working Days Calculator:**
  - Calculates number of working days between two dates.
  - Start date must be a weekday (Mon-Fri).
  - Exclude weekends (Saturday/Sunday) and public holidays
  - Public holidays stored in a table

- **Technology Stack:**
  - ASP.NET Core MVC
  - EF Core (DbContext + DbSet)
  - SQL Server
  - Bootsrap 5 + jQuery for UI

- **Architecture:**
  - Layered: Controller > Service > Repository > DbContext
  - Caching mechanism for frequently used data (like employee list, public holidays)
  - Delegates for generic caching (CachedLong, Cached)

- **Planned use of Stored Procedures (SPs):**
  - sp_InsertEmployee
  - sp_GetAllEmployees
  - sp_GetEmployeeById
  - sp_UpdateEmployee
  - sp_DeleteEmployee
  - sp_GetPublicHolidays

2. Solution Plan

- **Database Layer**
  - Create Employees and PublicHolidays tables.
  - Create SPs for CRUD operations and fetching holidays

- **Repository Layer**
  - Implement generic IRepository<T> for Employee
  - Implement WorkingDaysRepository for fetching public holidays.
  - Use EF Core FromSqlRaw for SPs.

- **Service Layer**
  - EmployeeService:
    - Handles business logic for CRUD, caching

  - WorkingDaysService:
    - Validate dates (start must be weekday, end >= start)
    - Retrieve public holidays (from cache or repo)
    - Calculate working days excluding weekends & holidays

- **Controller Layer**
  - EmployeesController – CRUD endpoints.
  - WorkingDaysController – GET for working days calculation.

- **UI Layer**
  - Views using Razor pages:
    - Employee: Index, Create, Edit
    - WorkingDays: Index
  - Bootstrap 5 for styling, validation with .was-validated.
  - Active menu highlighting

- **Caching**
  - Use cacheHelper for generic caching (long-term and 5-min cache).

- **Validation**
  - Server-side + client-side validation for required fields, start date weekday check.

- **Navigation**
  - Navbar links for Employees and Working Days Calculator.

3. Issues

- SP result shape mismatch with EF entity
  - Ensure SP returns all columns expected by EF entity OR use DTO
- Client-side validation not working
  - Use Bootstrap 5 .needs-validation + _ValidationScriptsPartial
- Caching stale data
  - Refresh cache on Create/Update/Delete actions
- Start date is weekend
  - Show error on client and server side

4. Pseudo Code

- **Employee CRUD**

```
Get Index:
   list = EmployeeService.GetAll()
   return View(list)

Post Create(Employee emp):
   if ModelState valid:
      EmployeeService.Add(emp)
      redirect to Index

Get Edit(id):
   emp = EmployeeService.GetById(id)
   return View(emp)

Post Edit(Employee emp):
   if ModelState valid:
      EmployeeService.Update(emp)
      redirect to Index

Delete(id):
   EmployeeService.Delete(id)
   redirect to Index
```

- **Working Days Calculator**

```
Post Index(startDate, endDate):
    if startDate is Sat/Sun:
        show error
    if endDate < startDate:
        show error
    holidays = Cache.Get("public_holidays") ?? repo.GetPublicHolidays()
    workingDays = 0
    for date in startDate..endDate:
        if date is Sat/Sun or in holidays:
            continue
        workingDays++
    ViewBag.Result = workingDays
    return View()
```

5. SP placeholders

- **Employee CRUD**

  - EXEC sp_InsertEmployee @Name, @Email, @JobPosition

```
ALTER PROCEDURE [dbo].[sp_InsertEmployee]
    @Name NVARCHAR(100),
    @Email NVARCHAR(100),
    @JobPosition NVARCHAR(100)
AS
BEGIN
    INSERT INTO Employees (Name, Email, JobPosition)
    VALUES (@Name, @Email, @JobPosition);
    SELECT SCOPE_IDENTITY() AS NewId;
END
```

  - EXEC sp_GetAllEmployees

```
ALTER PROCEDURE [dbo].[sp_GetAllEmployees]
AS
BEGIN
    SELECT * FROM Employees;
END
```

- EXEC sp_GetEmployeeById @Id

```
ALTER PROCEDURE [dbo].[sp_GetEmployeeById]
   @Id INT
AS
BEGIN
   SELECT * FROM Employees WHERE Id=@Id;
END
```

- EXEC sp_UpdateEmployee @Id, @Name, @Email, @JobPosition

```
ALTER PROCEDURE [dbo].[sp_UpdateEmployee]
   @Id INT,
   @Name NVARCHAR(100),
   @Email NVARCHAR(100),
   @JobPosition NVARCHAR(100)
AS
BEGIN
   UPDATE Employees
   SET Name=@Name, Email=@Email, JobPosition=@JobPosition
   WHERE Id=@Id;
END
```

- EXEC sp_DeleteEmployee @Id

```
ALTER PROCEDURE [dbo].[sp_DeleteEmployee]
   @Id INT
AS
BEGIN
   DELETE FROM Employees WHERE Id=@Id;
END
```

- **Public Holidays**

  - EXEC sp_GetPublicHolidays

```
ALTER PROCEDURE [dbo].[sp_GetPublicHolidays]
AS
BEGIN
   SELECT Id, HolidayDate, Description
   FROM PublicHolidays;
END
```