Author: Kelvin (04/2022)

Employee Management

Contents

Author: Kelvin (04/2022)

1.0. Project Summary

This project is a simple application to manage employee details internally. It should allow for adding, viewing/updating and deactivating records.

When possible, deletion should not be possible for users, therefore a user would update a records active status to disable records where relevant. This is because in real life usage we would normally not want to actually delete records as they are unretrievable if any issues arise with a certain employee for example. So, within the code we should look to check the active status of records for functionalities.

2.0. User Interface

2.1. User Interface of Application

2.1.1. Home Page

This page is where the application first loads and should contain a navigation to the Employee and Job pages. This should be where future features would be added.

2.1.2. Employee Page

This page should contain a list of existing employees from the database. Each record should be selectable to view and edit an employee record. On this screen, there should be a option to add new employee records to the database.

2.1.2.1. Employee Add

This should be a new screen where the users can input relevant employee information, then later save it to the database.

2.1.2.2. Employee Edit

Clicking on an employee record should take the user to a page where all the relevant fields are populated with the employee data. Here the employee could update or deactivate a single record by saving.

2.1.3. Job Page

This page should contain list of existing jobs from the database. Each record should be selectable to view and edit a job record.

2.1.3.1. Job Add

This should be a new screen where the users can input relevant job information, then later save it o the database.

2.1.3.2. Job Edit

Clicking on a job record should take the user to a page where all the relevant fields are populated with the job data. Here the job could update or deactivate a single record by saving.

3.0. Database

3.1. Table Structure

3.1.1. Employee

Column Name	Data Type	Description
ID	Small int, not null	(PK) unique identifier of table
EmployeeNo	Nvarchar(50), not null	Employee number field relevant to users
Name	Nvarchar(50), not null	Name of record
PhoneNo	Nvarchar(50), null	Phone number of record
Email	Navarchar(50), null	Email of record
JobID	Small int, not null	(FK to Job) Job reference of record
SalaryAmount	Nvarchar(50), null	Salary amount of record
StartDate	Datetime, not null	Start date of employee record
LeaveDate	Datetime, null	Leave date of employee record
IsActive	Bool, not null	Active status of record
DateCreated	DateTime, not null	Date created of record

3.1.2. Job

Column Name	Data Type	Description
ID	Small int, not null	(PK) unique identifier of table
Title	Nvarchar(50), not null	Title field of record
IsActive	Bool, not null	Active status of record

4.0. Technologies

For development, this application will be created using the Visual Studio IDE, programmed in C# .NET Blazor WebAssembly (client-side web app) and TSQL for the database creation/management.

5.0. Tasks

5.1. Task Breakdown

Task	Description	Estimate
Create database & table	Write script that creates the database and tables with some sample data for consistency during development	1 hrs
Empty project and upload to version control	Select and create template project then upload to version control for consistency	1 hrs
UI pages	Make blank pages mentioned in the User Interface section	3 hrs
Save functionality	Create generic save functionality	2 hrs
Add functionality	Create generic add functionality	1 hrs
Edit functionality	Create generic edit functionality	1 hrs
Delete functionality	Create generic delete functionality	1 hrs
Connect backend to project	Connect backend to project	1 hrs
Connect UI to backend	Connect front and backend, making sure that actions save as intended	10 hrs
Testing project as whole	Test project as a whole – note fix any bugs discovered	5 hrs

6.0. After Development

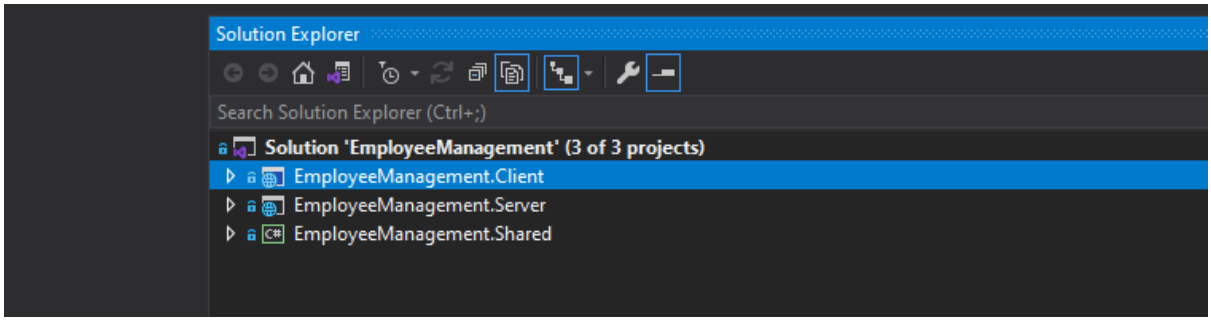
6.1. Script to Create Database, Tables & Sample Data

```
IF(DB_ID(N'EmployeeManagement') IS NULL)
	CREATE DATABASE [EmployeeManagement];
ELSE
	DROP DATABASE [EmployeeManagement];

GO

USE [EmployeeManagement]

CREATE TABLE [dbo].[Job] (
	[ID]		SMALLINT		IDENTITY(1, 1) NOT NULL,
	[Title]			NVARCHAR(50)	NOT NULL,
	[IsActive]	BIT				NOT NULL,
	CONSTRAINT [PK_Job] PRIMARY KEY CLUSTERED ([ID] ASC)
);

CREATE TABLE [dbo].[Employee] (
	[ID]			SMALLINT		IDENTITY(1, 1) NOT NULL,
	[EmployeeNo]		NVARCHAR(50)	NOT NULL,
	[Name]			NVARCHAR(50)	NOT NULL,
	[PhoneNo]		NVARCHAR(50)	NULL,
	[Email]			NVARCHAR(50)	NULL,
	[JobID]			SMALLINT		NOT NULL,
	[SalaryAmount]		NVARCHAR(50)	NULL,
	[StartDate]		DATETIME		NOT NULL,
	[LeaveDate]		DATETIME		NULL,
	[IsActive]		BIT			NOT NULL,
	[DateCreated]		DATETIME		NOT NULL,
	CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED ([ID] ASC),
	CONSTRAINT [FK_Job] FOREIGN KEY ([JobID]) REFERENCES [dbo].[Job] ([ID]),
);

GO

INSERT INTO [dbo].[Job] ([Title], [IsActive])
VALUES
('Sales Manager', 1),
('Sales Assistant Manager', 1),
('Sales Associate', 1);

INSERT INTO [dbo].[Employee] ([EmployeeNo], [Name], [PhoneNo], [Email], [JobID],
[SalaryAmount], [StartDate], [LeaveDate], [IsActive], [DateCreated])
VALUES
('G008', 'Joe Bloggs', NULL, NULL, 1, '£20,000', '2021-04-04', null, 1, '2020-01-01'),
('G009', 'Sarah Smith', '+44 123456789', 'sarah@yahoo.com', 2, '£19,000', '2021-04-
04', null, 1, '2021-04-12'),
('G010', 'Clark Kent', NULL, 'clark@outlook.com', 3, '£18,000', '2021-04-04', '2022-
01-19', 0, '2022-02-28');

GO

SELECT * FROM [dbo].[Job]
SELECT * FROM [dbo].[Employee]
```

Author: Kelvin (04/2022)

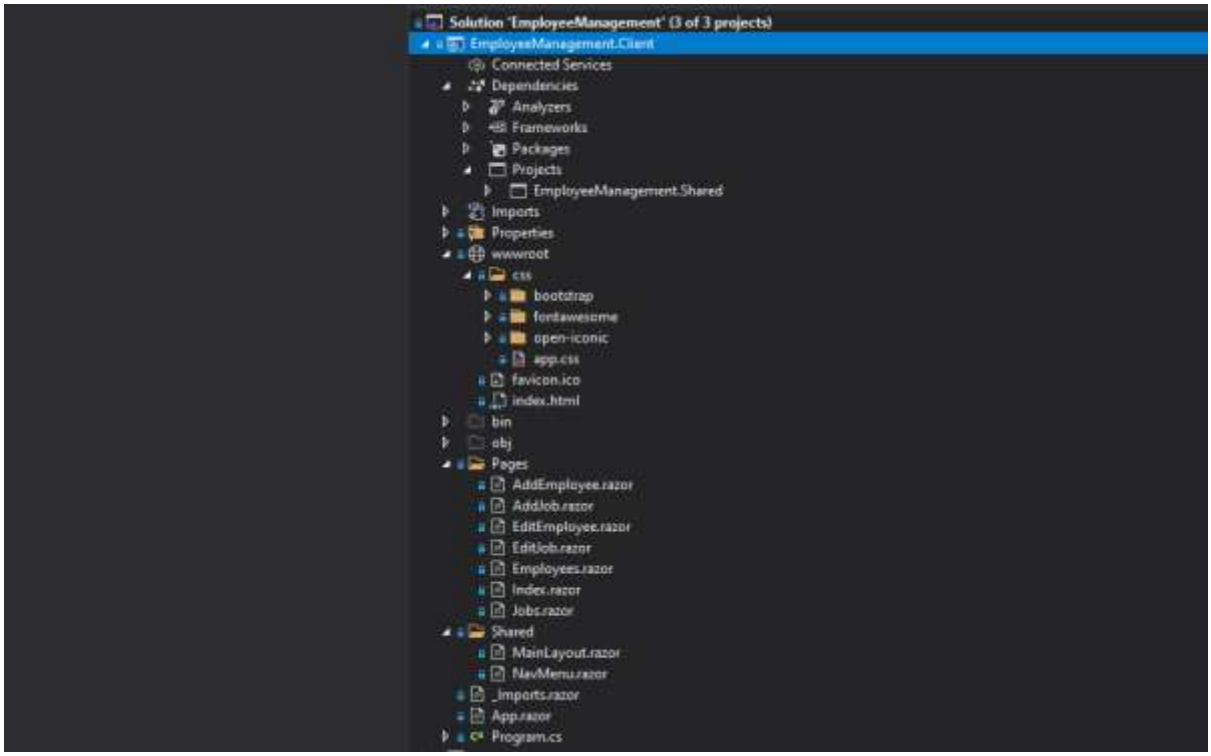
6.2. Project Layout

The solution is broken down into three projects: client, server and shared. An important aspect to note is that the server and client project needs to be ran at the same time as the server project communicates to the database and the client displays a user interface (right clicking the solution then selecting the properties option allows for this).



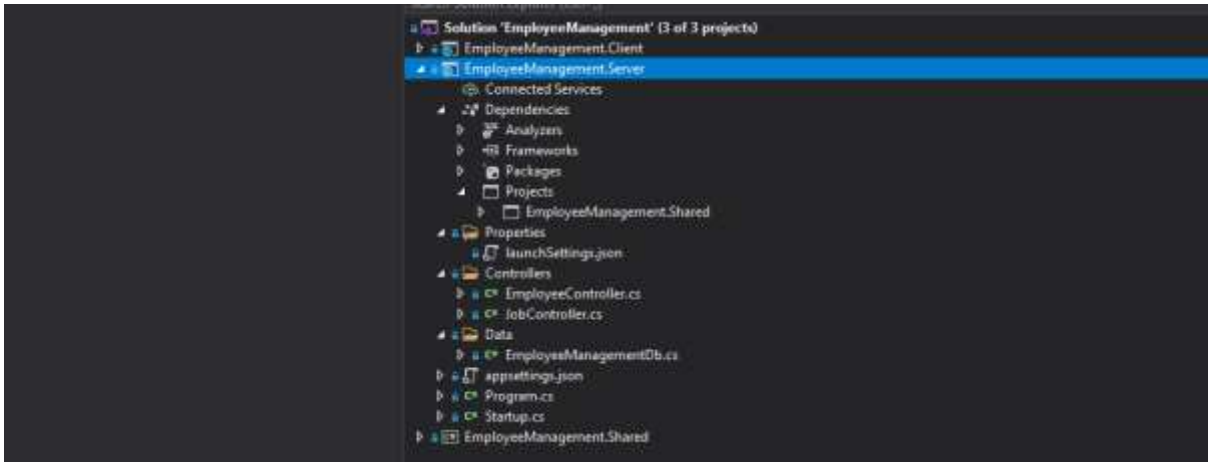
6.2.1. Client Project

The client project is essentially the user interface client application, it depends on the shared library project:



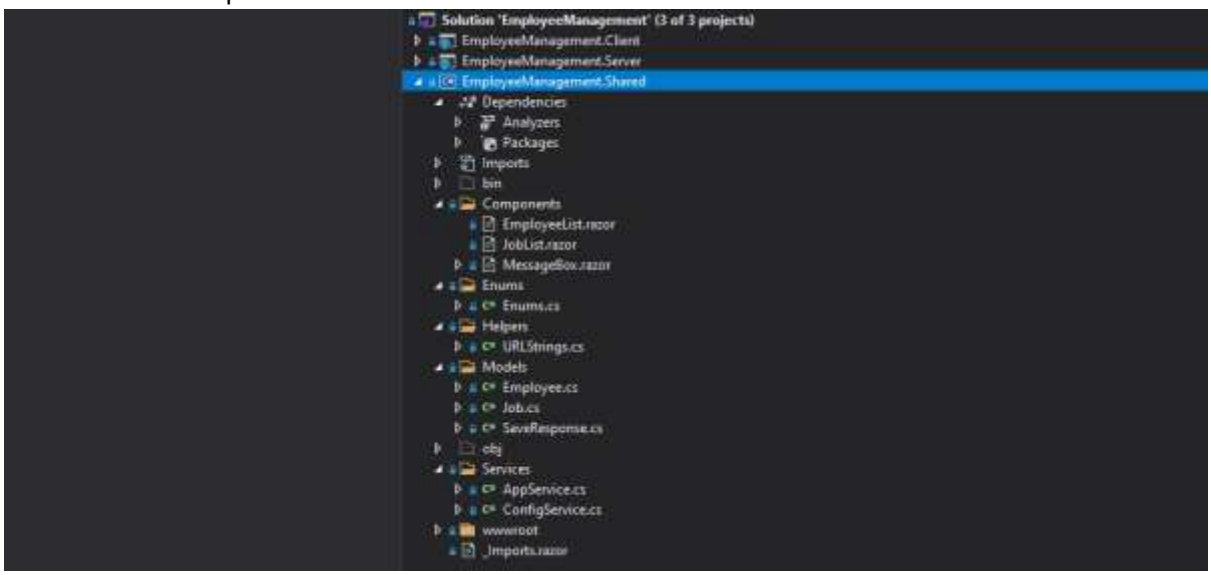
6.2.2. Server Project

The server project is where the database class is kept and where the controllers are kept that performs the API endpoints such as get, post, delete, etc. It also references the shared project.



6.2.3. Shared Project

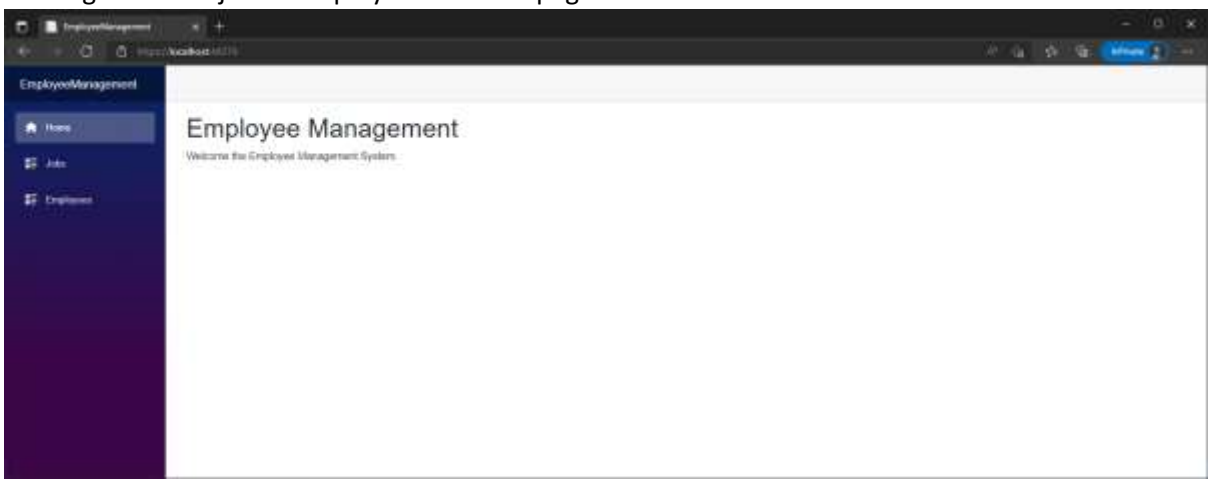
The shared project is where the files which both the client and server project use, i.e., where content that is shared is kept.



6.3. Run through of UI

6.3.1. Home Page

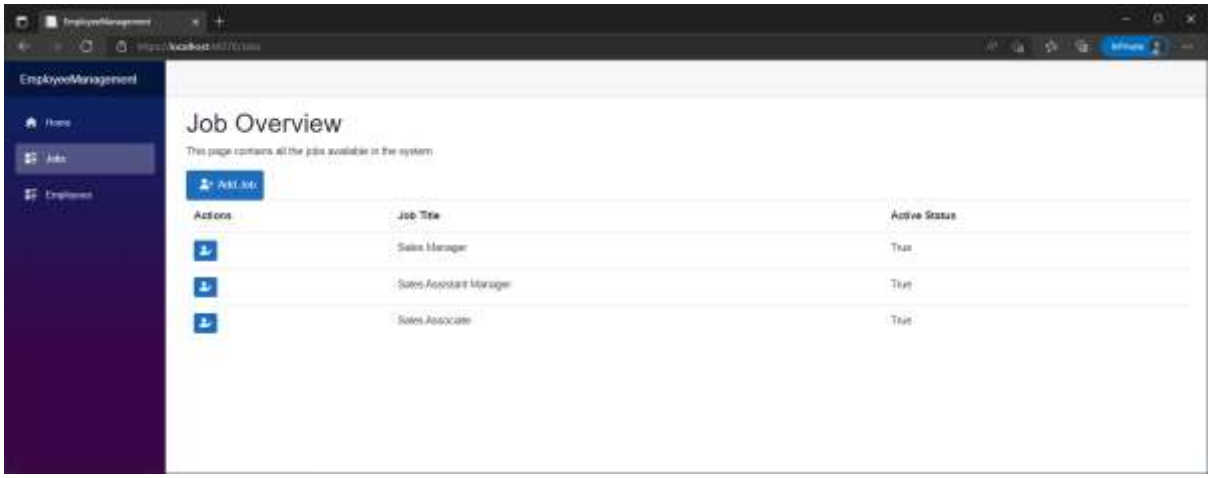
Where the initial project loads, the left-hand side shows the navigation pages, where users can either go into the job or employee overview pages:



6.3.2. Job Record Management

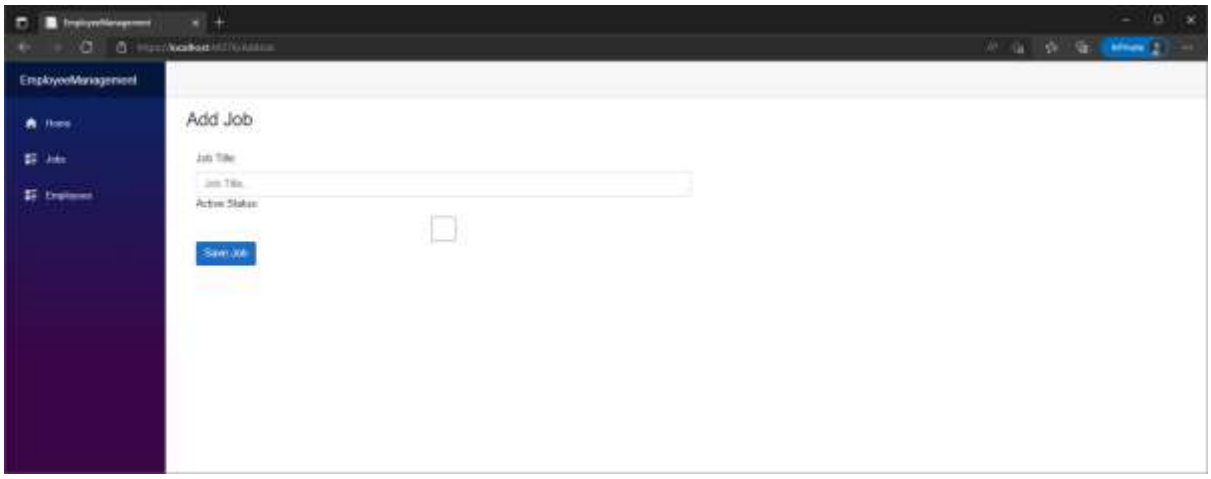
6.3.2.1. Job Overview Page

Clicking on the 'Jobs' menu options take the user to the 'Job Overview' page. Here the users get a list of all jobs on the database, they can now choose to add a new job or edit an existing one:

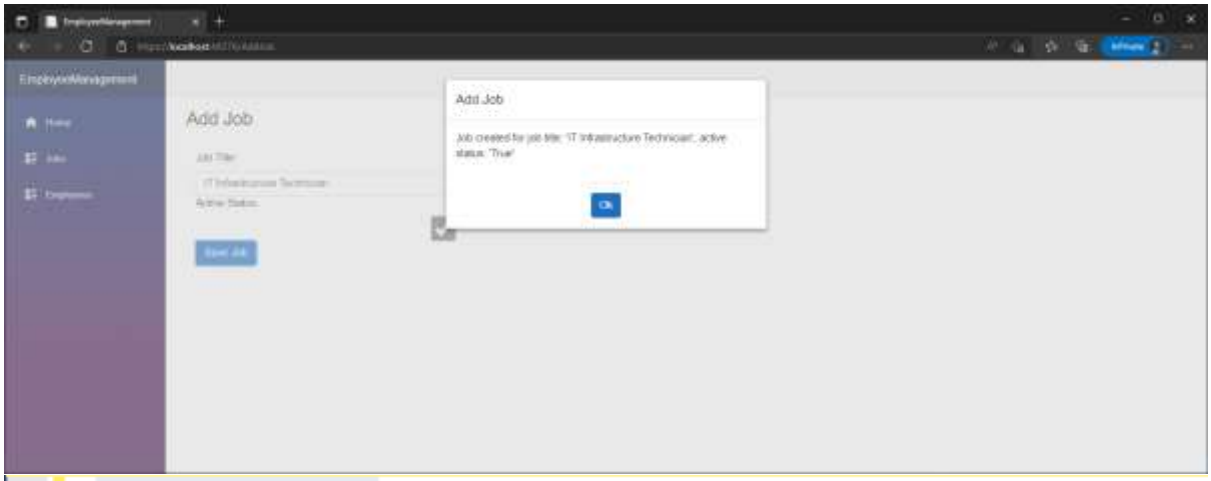


6.3.2.2. Add Job Page

Clicking on the 'Add Job' button from the overview page takes the users to a form where they can enter the job title and the active status:



Clicking the 'Save Job' button will display a success prompt and add the record to the database:



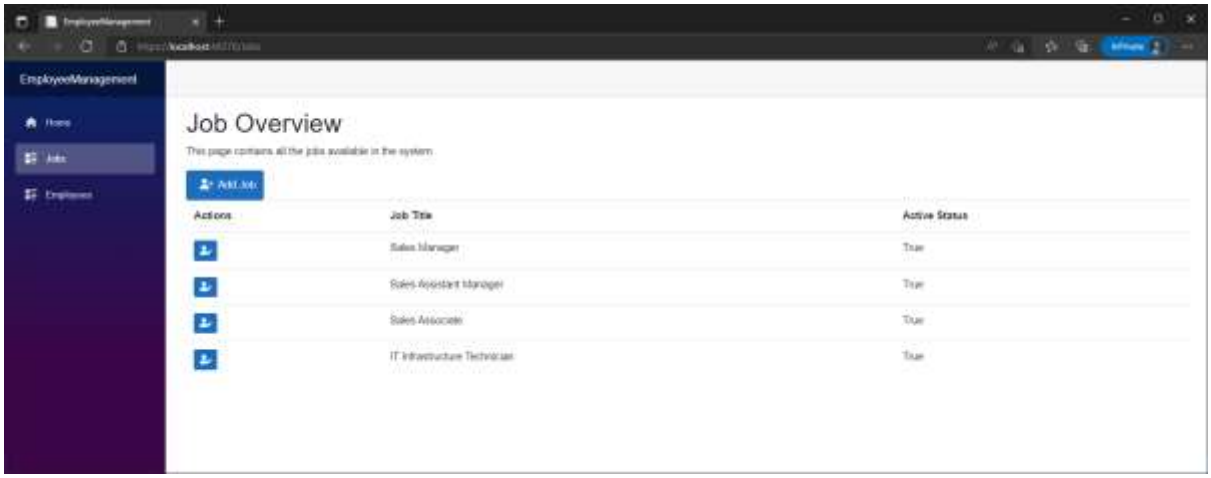

```
select * from Job
```

120 %

Results | Messages

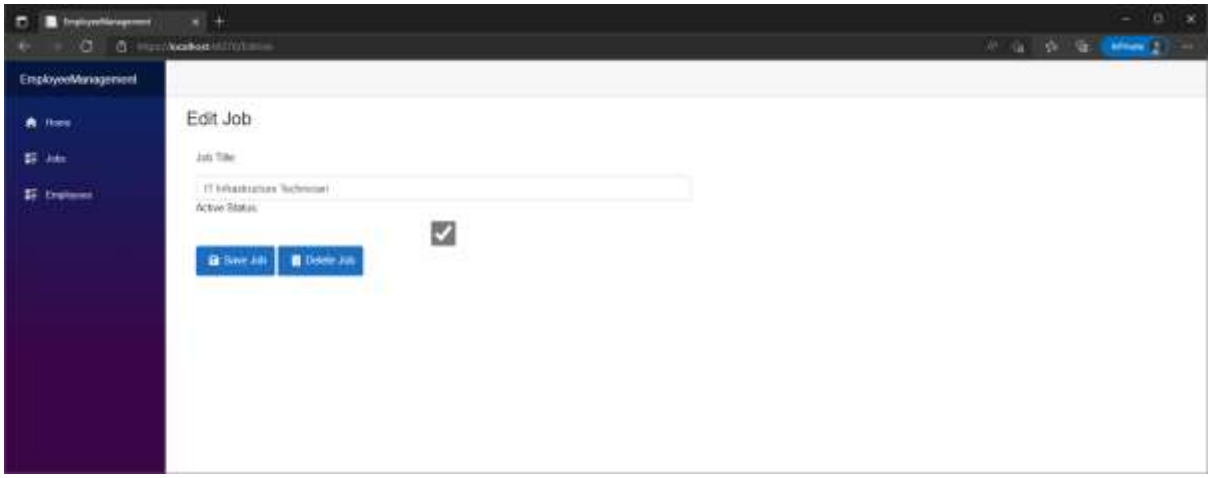
	ID	Title	IsActive
1	1	Sales Manager	1
2	2	Sales Assistant Manager	1
3	3	Sales Associate	1
4	4	IT Infrastructure Technician	1

The new record should now display in the overview page:

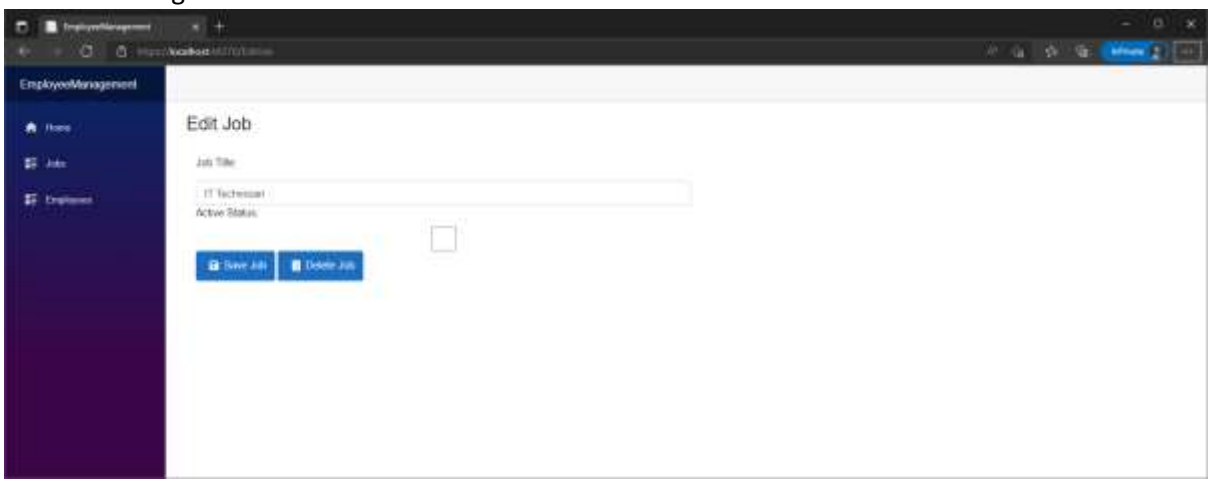


6.3.2.3. Edit Job Page

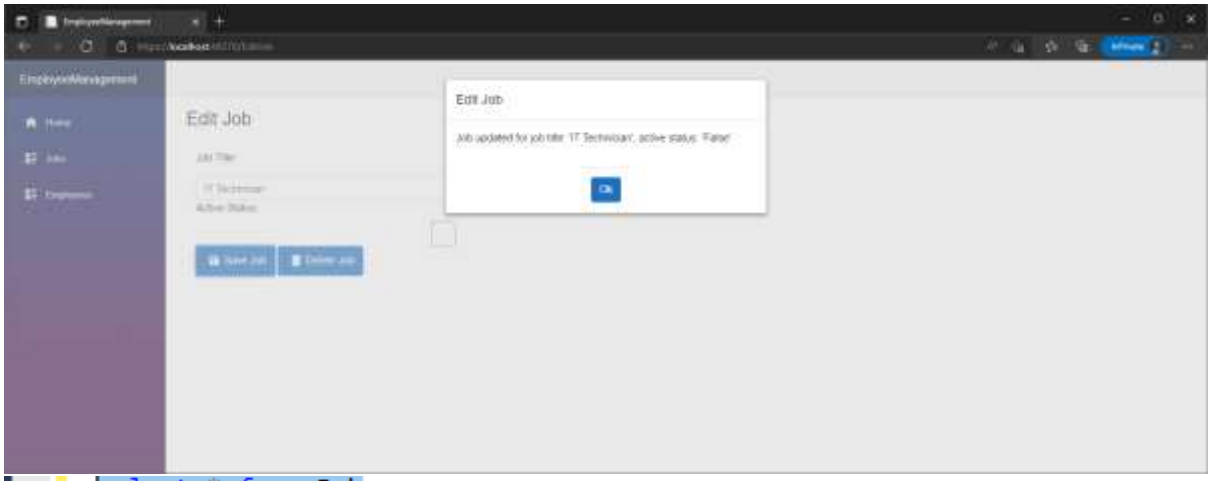
Clicking on the 'Edit Job' icon for a record will take the users to a form where the users can edit the job record:



For this I changed the title and deactivated the record:



Clicking the ‘Save Job’ button should display a success prompt and the database should be updated:




```
select * from Job
```

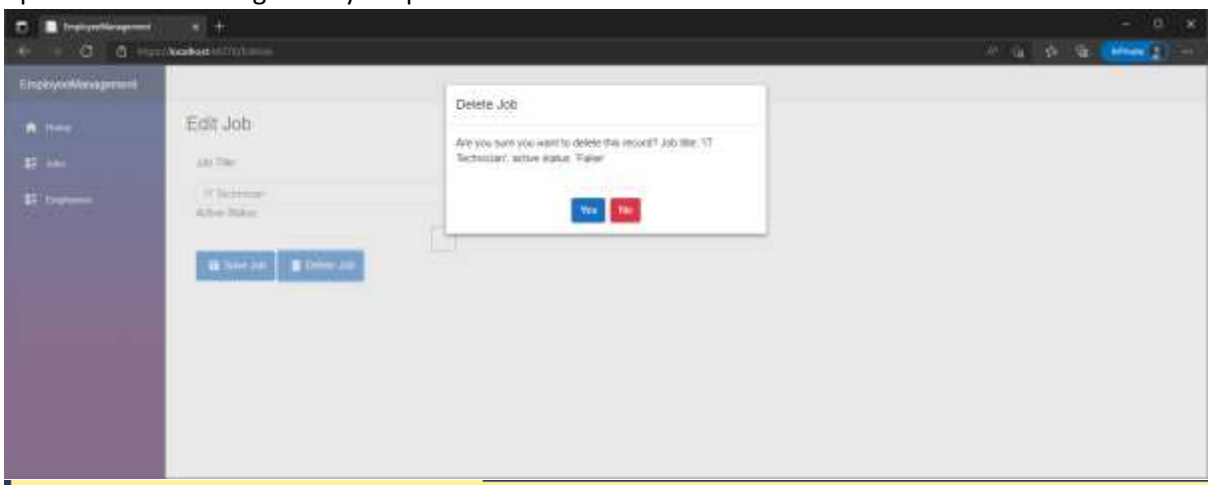
120 %

Results | Messages

	ID	Title	IsActive
1	1	Sales Manager	1
2	2	Sales Assistant Manager	1
3	3	Sales Associate	1
4	4	IT Technician	0

6.3.2.4. Delete Job

Clicking on the ‘Delete Job’ button on the edit job page should prompt the user to confirm, the no option will do nothing while yes option will delete the record:



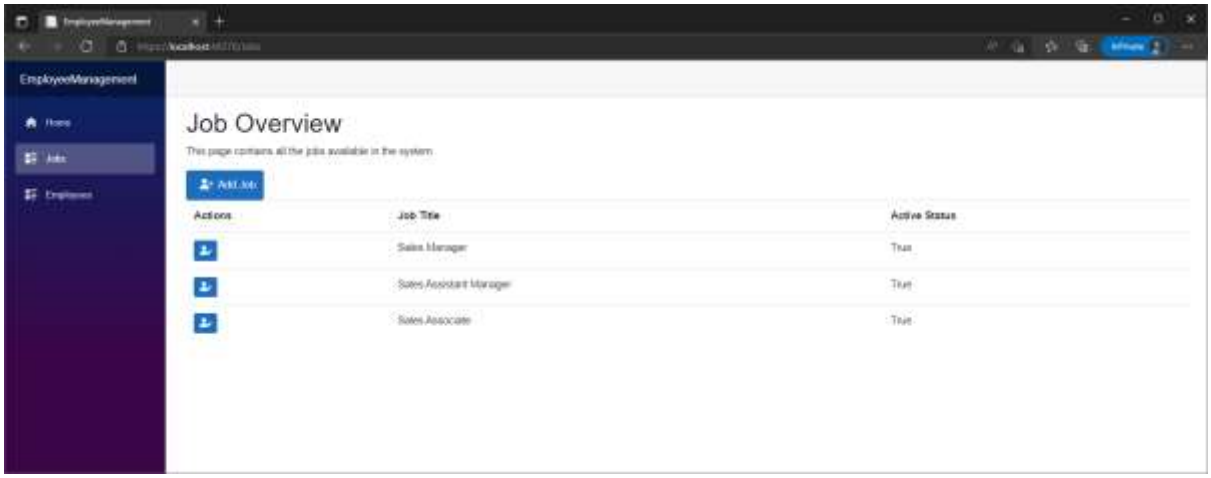

```
select * from Job
```

120 %

Results | Messages

	ID	Title	IsActive
1	1	Sales Manager	1
2	2	Sales Assistant Manager	1
3	3	Sales Associate	1

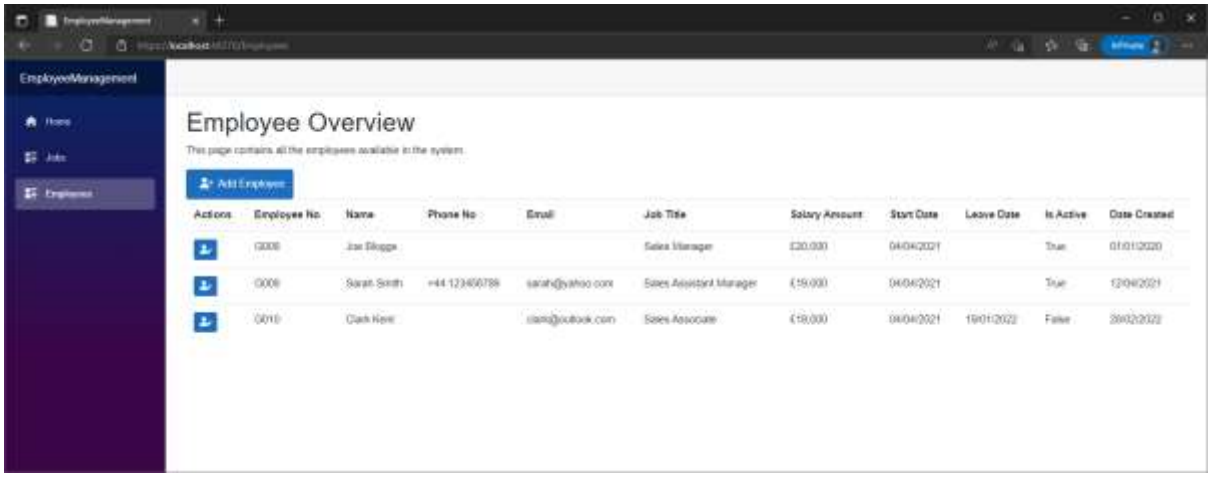
On delete the record should be removed from the overview page:



6.3.3. Employee Record Management

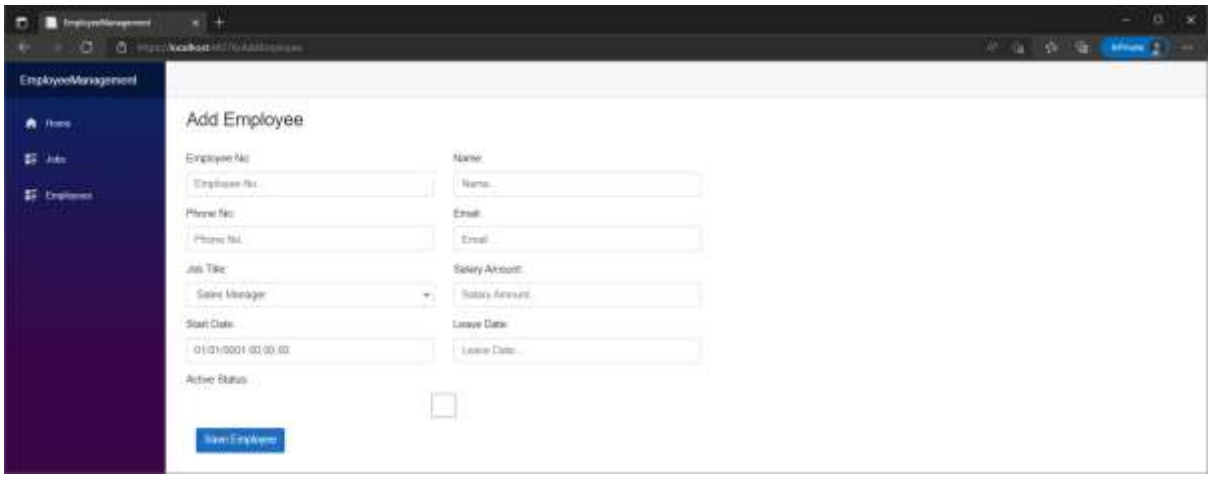
6.3.3.1. Employee Overview Page

Clicking on the 'Employees' menu options take the user to the 'Employee Overview' page. Here the users get a list of all employees on the database, they can now choose to add a new employee or edit an existing one:

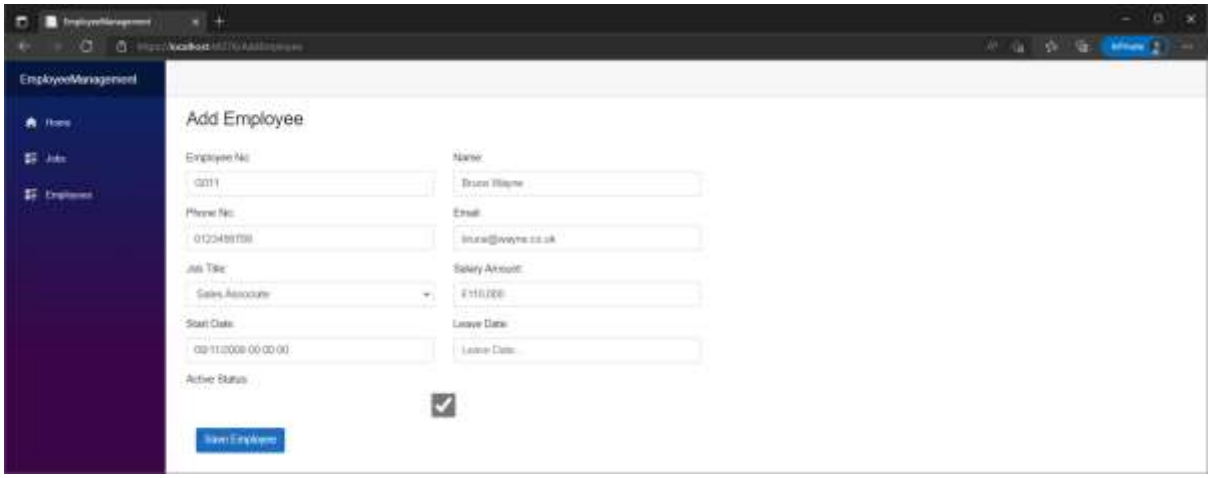


6.3.3.2. Add Employee Page

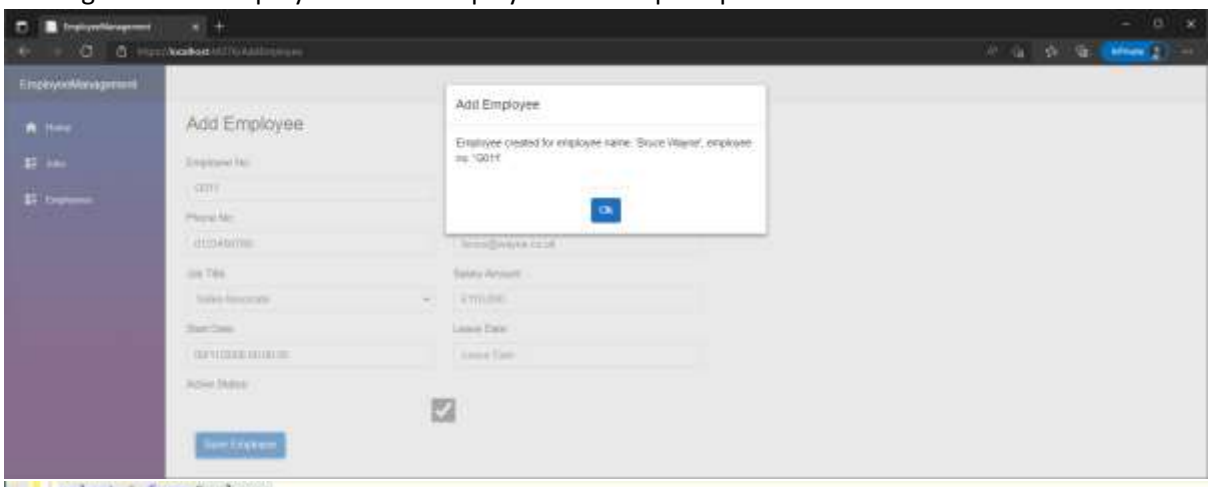
Clicking on the 'Add Employee' button from the overview page takes the users to a form where they can enter the details for an employee:



For this I have filled in form:



Clicking the 'Save Employee' button displays a success prompt and saves the record to the database:



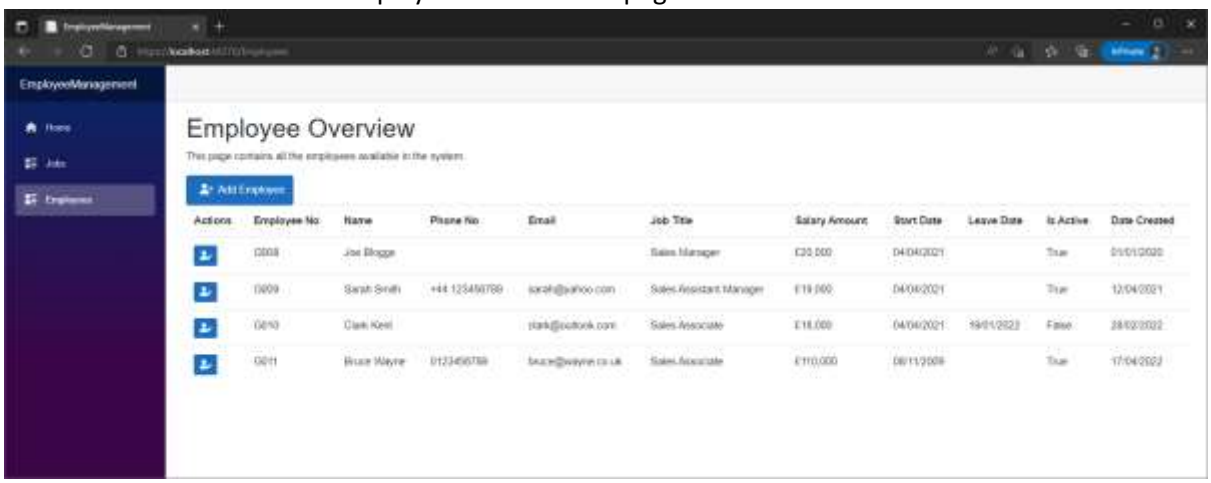

```
select * from Employee
```

120 %

Results | Messages

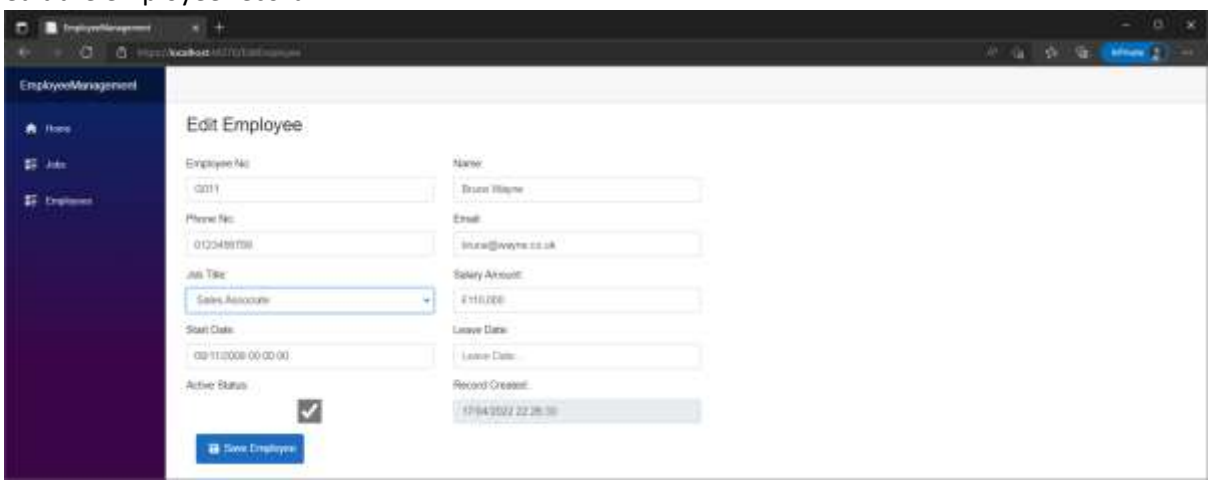
	ID	EmployeeNo	Name	PhoneNo	Email	JobID	SalaryAmount	StartDate	LeaveDate	IsActive	DateCreated
1	1	G008	Joe Blogs	NULL	NULL	1	£20,000	2021-04-04 00:00:00.000	NULL	1	2020-01-01 00:00:00.000
2	2	G009	Sarah Smith	+44 123456789	sarah@yahoo.com	2	£19,000	2021-04-04 00:00:00.000	NULL	1	2021-04-12 00:00:00.000
3	3	G010	Clark Kent	NULL	clark@outlook.com	3	£18,000	2021-04-04 00:00:00.000	2022-01-19 00:00:00.000	0	2022-02-28 00:00:00.000
4	4	G011	Bruce Wayne	0123456789	bruce@wayne.co.uk	3	£110,000	2009-11-08 00:00:00.000	NULL	1	2022-04-17 22:26:33.193

The new record should now display in the overview page:

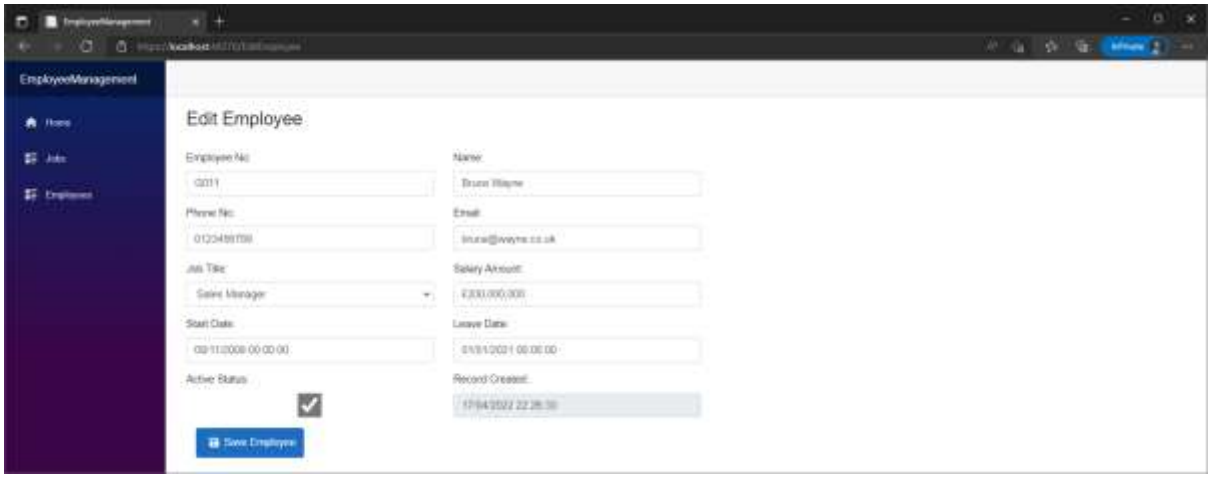


6.3.3.3. Edit Employee Page

Clicking on the 'Edit Employee' icon for a record will take the users to a form where the users can edit the employee record:

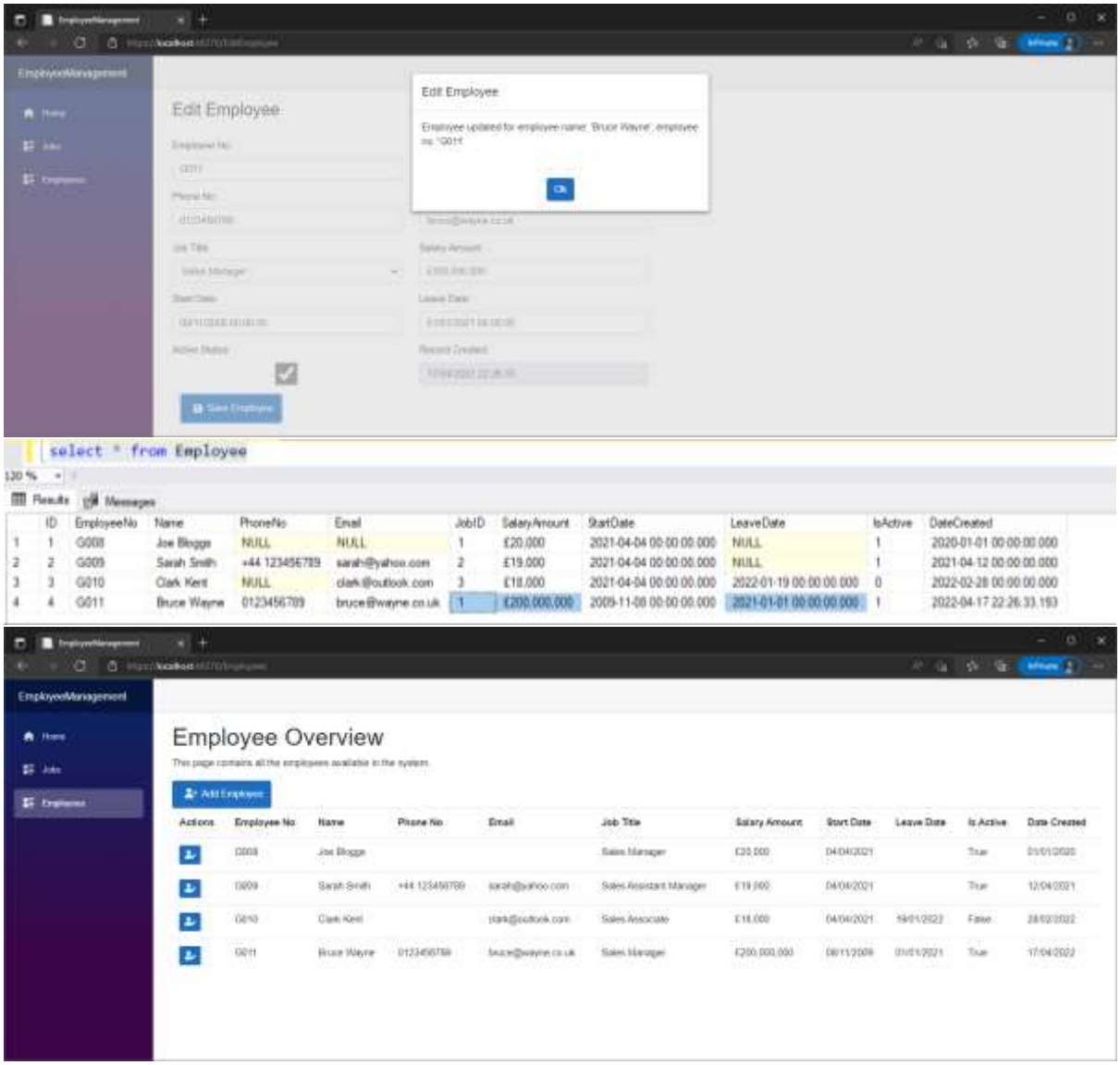


For this, I have changed the job title, salary amount and leave date:



Clicking the 'Save Employee' button should display a success promt to the users and update the record in the database:

Author: Kelvin (04/2022)



End of Document