

Employee Management

Contents

1.0.	Project Summary.....	2
2.0.	User Interface	2
2.1.	User Interface of Application	2
2.1.1.	Home Page	2
2.1.2.	Employee Page.....	2
2.1.3.	Job Page	2
3.0.	Database	3
3.1.	Table Structure.....	3
3.1.1.	Employee.....	3
3.1.2.	Job	3
4.0.	Technologies	3
5.0.	Tasks.....	4
5.1.	Task Breakdown	4
6.0.	After Development	5
6.1.	Script to Create Database, Tables & Sample Data	5
6.2.	Project Layout	6
6.2.1.	Client Project.....	6
6.2.2.	Server Project.....	7
6.2.3.	Shared Project.....	7
6.3.	Run through of UI.....	8
6.3.1.	Home Page	8
6.3.2.	Job Record Management	8
6.3.3.	Employee Record Management.....	12
End of Document	15

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	Nvarchar(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);

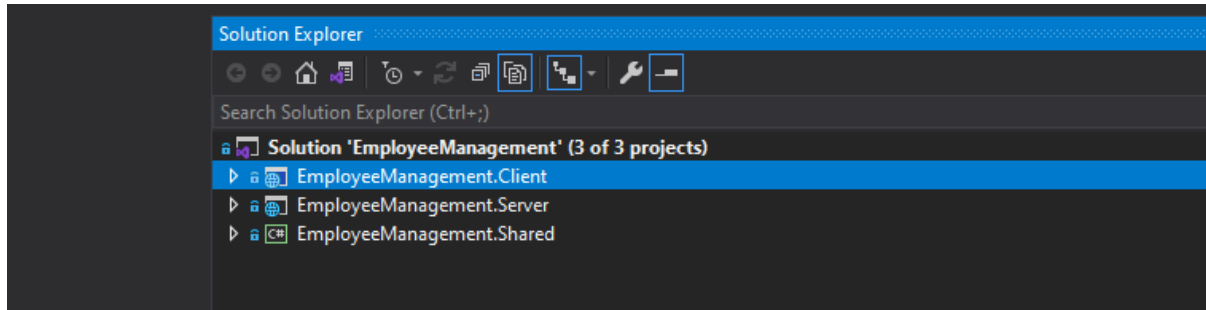
INSERT INTO [dbo].[Employee] ([EmployeeNo], [Name], [PhoneNo], [Email], [JobID],
[SalaryAmount], [StartDate], [LeaveDate], [IsActive], [DateCreated])
VALUES
('G008', 'Joe Bloggs', NULL, NULL, 1, '£20,000', '2021-04-04', null, 1, '2020-01-01'),
('G009', 'Sarah Smith', '+44 123456789', 'sarah@yahoo.com', 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]
```

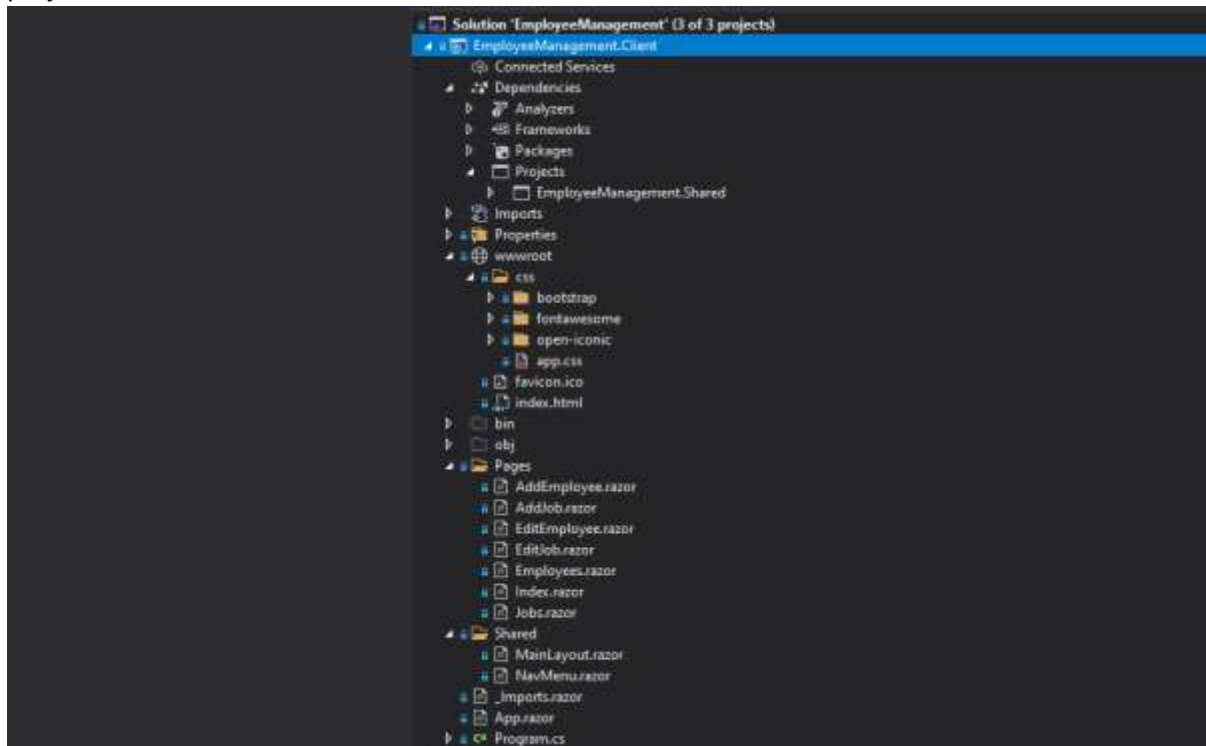
6.2. Project Layout

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).



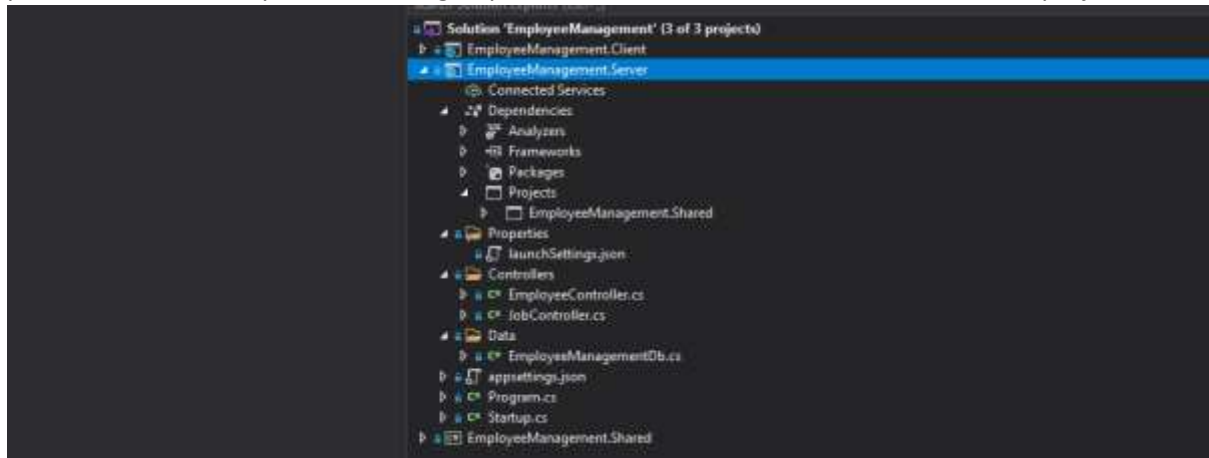
6.2.1. Client Project

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



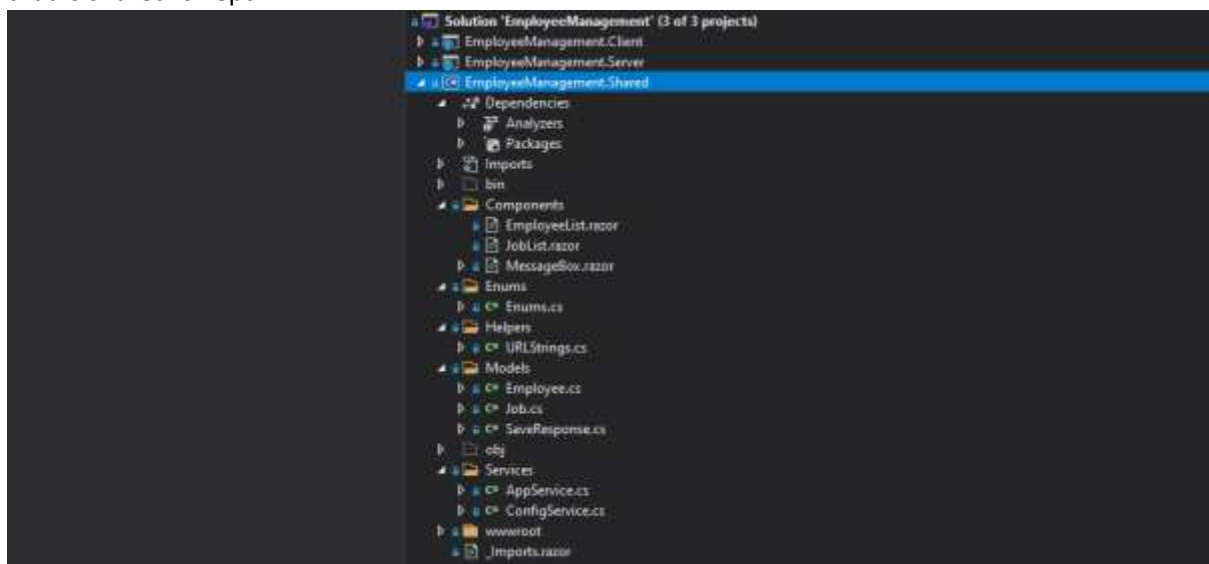
6.2.2. Server Project

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

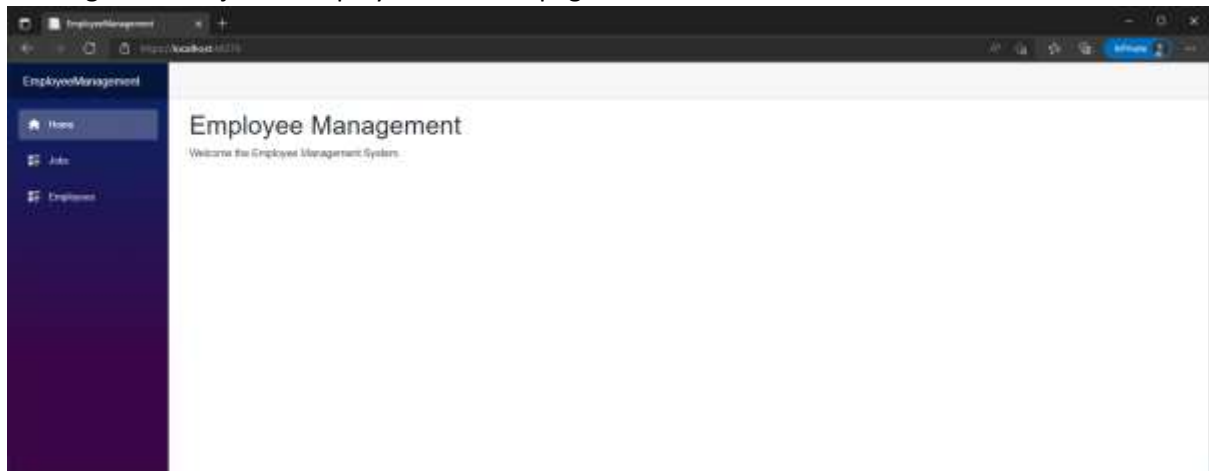
The shared project is where the files which both the client and server project use, i.e., where content that is shared is kept.



6.3. Run through of UI

6.3.1. Home Page

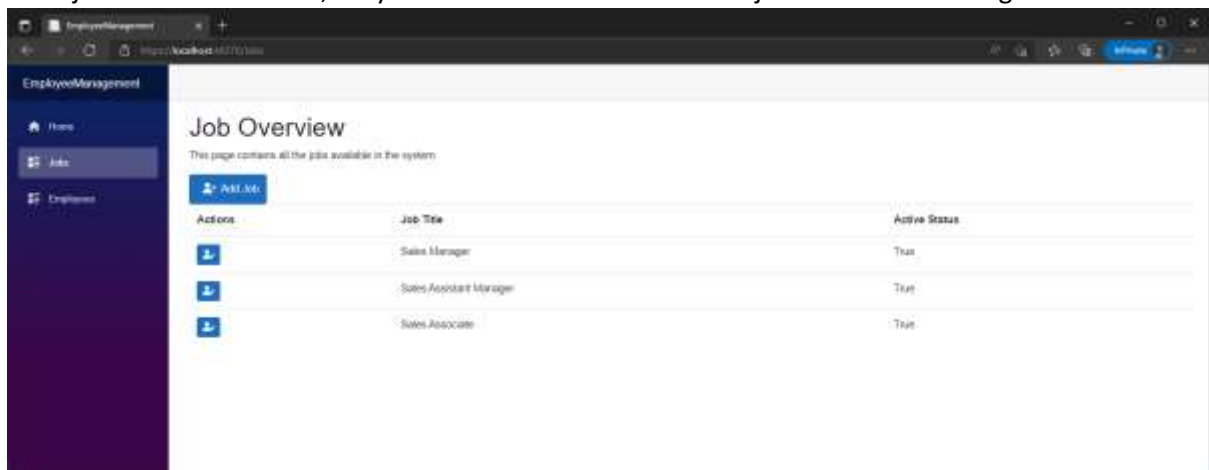
Where the initial project loads, the left-hand side shows the navigation pages, where users can either go into the job or employee overview pages:



6.3.2. Job Record Management

6.3.2.1. Job Overview Page

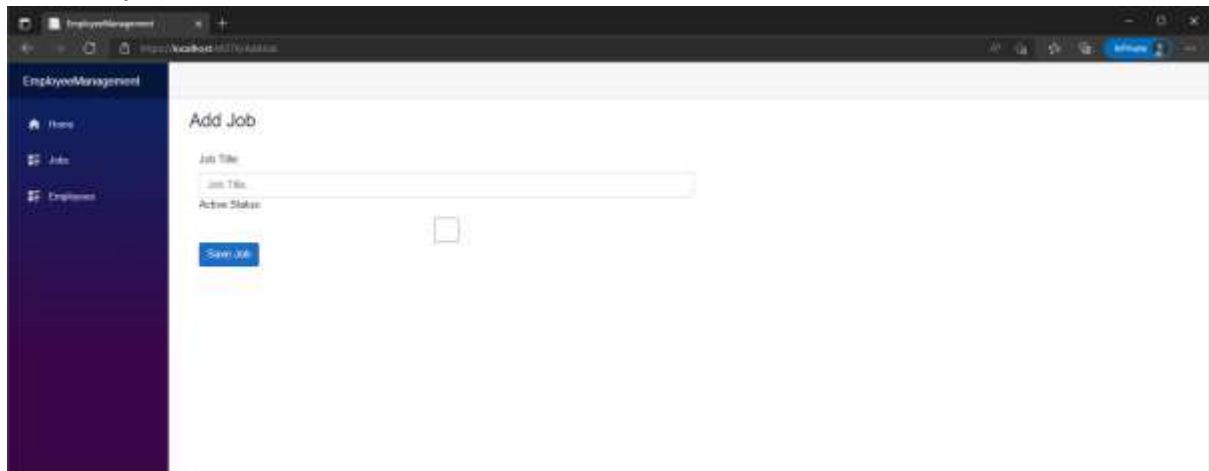
Clicking on the 'Jobs' menu options take the user to the 'Job Overview' page. Here the users get a list of all jobs on the database, they can now choose to add a new job or edit an existing one:



Author: Kelvin (04/2022)

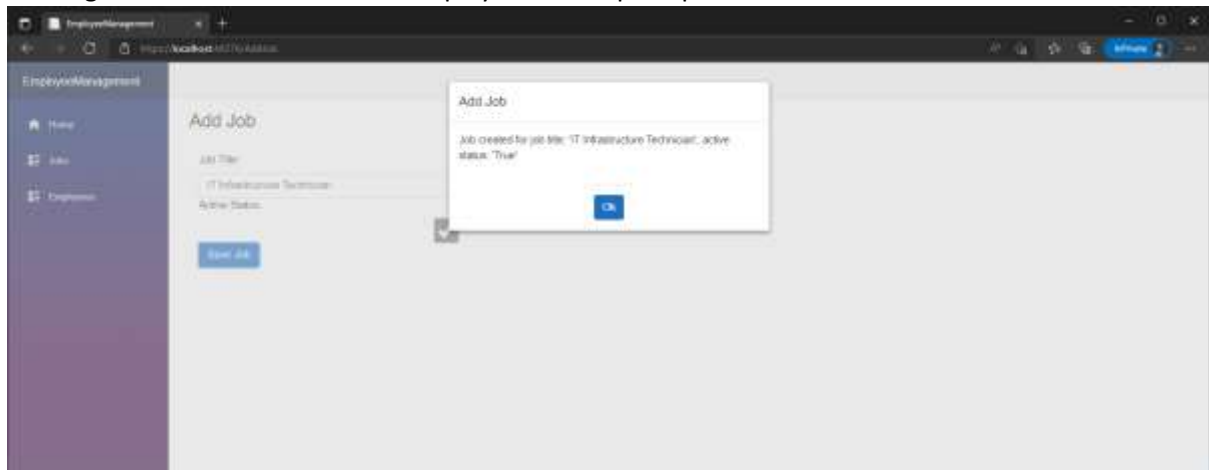
6.3.2.2. Add Job Page

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:



The screenshot shows a web application titled 'EmployeeManagement'. On the left is a dark sidebar with navigation links: 'Home', 'Jobs', and 'Employees'. The main content area is titled 'Add Job'. It contains a text input field for 'Job Title', a checkbox for 'Active Status', and a blue button labeled 'Save Job'.

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



This screenshot shows the same 'Add Job' form, but now a modal dialog is displayed over it. The modal has a title 'Add Job' and a message: 'Job created for job title: IT Infrastructure Technician, active status: True'. There is an 'OK' button in the modal. The background form is slightly dimmed.

```
select * from Job
```

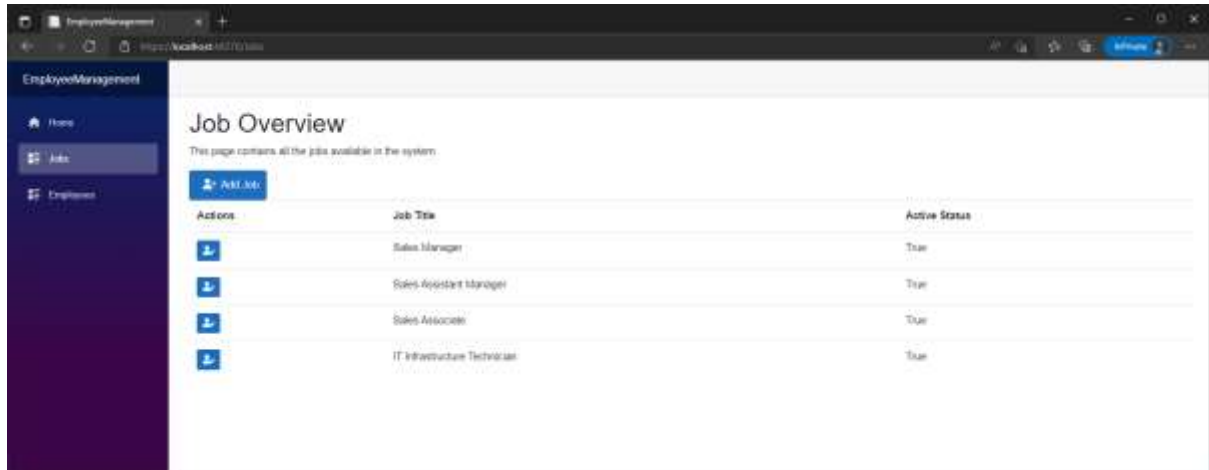
120 %

	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 screenshot shows a SQL query editor with the query 'select * from Job'. Below the query, there is a zoom level of '120 %'. At the bottom, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 4 rows and 4 columns: ID, Title, and IsActive. The first row is highlighted.

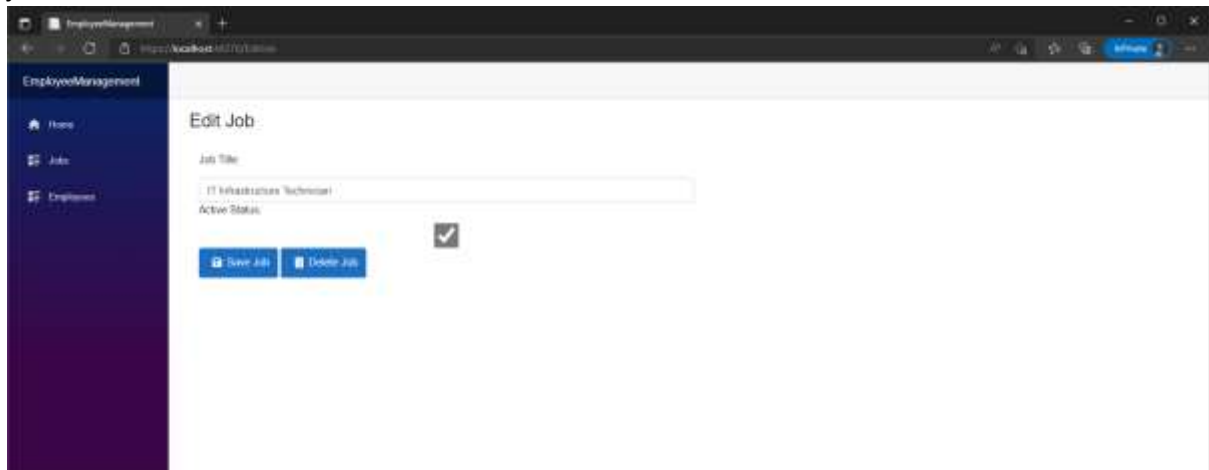
Author: Kelvin (04/2022)

The new record should now display in the overview page:

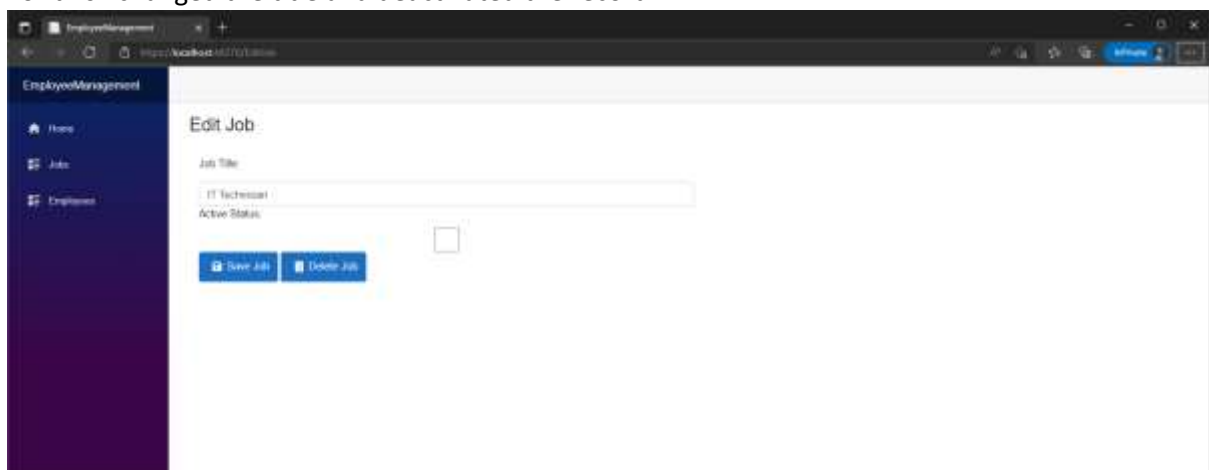


6.3.2.3. Edit Job Page

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:

The screenshot shows the 'Edit Job' page in a web application. A success message box is displayed, stating: 'Job updated for job title: IT Technician, active status: False'. Below the message is a blue 'Ok' button. The page also shows the 'Job Title' field set to 'IT Technician' and the 'Active Status' field set to 'False'. At the bottom, there are 'Save Job' and 'Delete Job' buttons. Below the page, a SQL query is shown: `select * from Job`. The results of the query are displayed in a table:

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

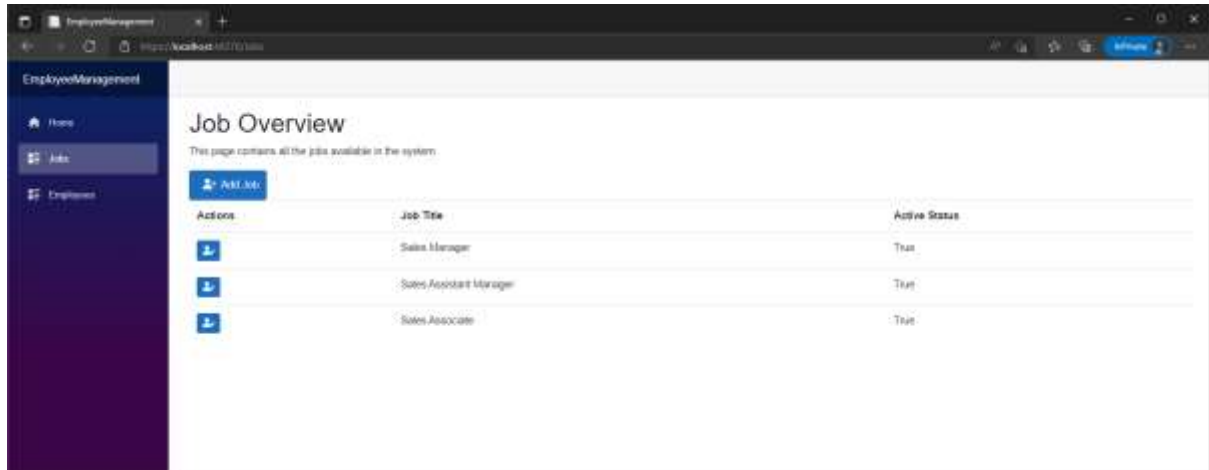
6.3.2.4. Delete Job

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:

The screenshot shows the 'Edit Job' page in a web application. A confirmation message box is displayed, asking: 'Are you sure you want to delete this record? Job title: IT Technician, active status: False'. Below the message are 'Yes' and 'No' buttons. The page also shows the 'Job Title' field set to 'IT Technician' and the 'Active Status' field set to 'False'. At the bottom, there are 'Save Job' and 'Delete Job' buttons. Below the page, a SQL query is shown: `select * from Job`. The results of the query are displayed in a table:

	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:



EmployeeManagement

Home Jobs Employees

Job Overview

This page contains all the jobs available in the system.

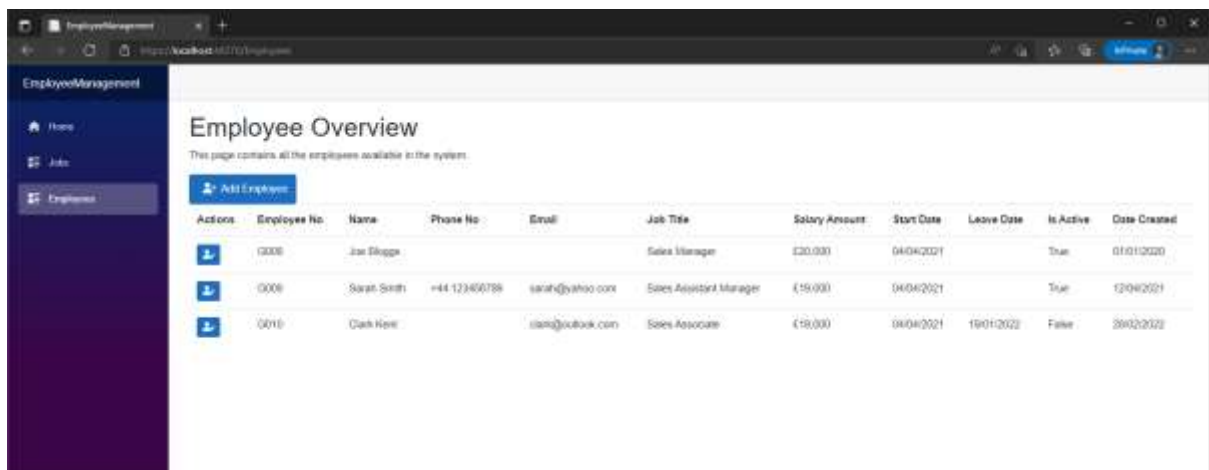
[Add Job](#)

Actions	Job Title	Active Status
Add	Sales Manager	True
Add	Sales Assistant Manager	True
Add	Sales Associate	True

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:



EmployeeManagement

Home Jobs Employees

Employee Overview

This page contains all the employees available in the system.

[Add Employee](#)

Actions	Employee No.	Name	Phone No.	Email	Job Title	Salary Amount	Start Date	Leave Date	Is Active	Date Created
Add	0008	Jane Elgoge			Sales Manager	£20,000	04/04/2021		True	01/01/2020
Add	0009	Sarah Smith	+44 123456789	sarah@yahoo.com	Sales Assistant Manager	£15,000	04/04/2021		True	12/04/2021
Add	0010	Clark Hale		clark@outlook.com	Sales Associate	£10,000	04/04/2021	19/01/2022	False	29/02/2022

Author: Kelvin (04/2022)

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:

The screenshot shows the 'Add Employee' form in the EmployeeManagement application. The form is empty, with fields for Employee No, Name, Phone No, Email, Job Title, Salary Amount, Start Date, Leave Date, and Active Status. The 'Save Employee' button is at the bottom.

For this I have filled in form:

The screenshot shows the 'Add Employee' form with the following data filled in: Employee No: G011, Name: Bruce Wayne, Phone No: 0123456789, Email: bruce@wayne.co.uk, Job Title: Sales Associate, Salary Amount: £110,000, Start Date: 2021-03-08 00:00:00, Leave Date: (empty), Active Status: checked.

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

The screenshot shows the 'Add Employee' form with a success prompt overlay. The prompt says: "Add Employee. Employee created for employee name: Bruce Wayne, employee no: G011". Below the form, a SQL query "select * from Employee" is shown, followed by a table of employee data.

ID	EmployeeNo	Name	PhoneNo	Email	JobID	SalaryAmount	StartDate	LeaveDate	IsActive	DateCreated
1	G008	Joe Bloggs	NULL	NULL	1	£20,000	2021-04-04 00:00:00.000	NULL	1	2020-01-01 00:00:00.000
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	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	G011	Bruce Wayne	0123456789	bruce@wayne.co.uk	3	£110,000	2020-11-08 00:00:00.000	NULL	1	2022-04-17 22:26:33.193

Author: Kelvin (04/2022)

The new record should now display in the overview page:

Actions	Employee No	Name	Phone No	Email	Job Title	Salary Amount	Start Date	Leave Date	Is Active	Date Created
	0008	Joe Bloggs			Sales Manager	£30,000	04/04/2021		True	01/01/2020
	0009	Sarah Smith	+44 123456789	sarah@yahoo.com	Sales Assistant Manager	£18,000	04/06/2021		True	12/04/2021
	0010	Clark Keel		clark@outlook.com	Sales Associate	£16,000	04/04/2021	19/01/2022	False	28/02/2022
	0011	Bruce Wayne	0123456789	bruce@wayne.co.uk	Sales Associate	£110,000	08/11/2008		True	17/04/2022

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:

Employee No: 0011

Name: Bruce Wayne

Phone No: 0123456789

Email: bruce@wayne.co.uk

Job Title: Sales Associate

Salary Amount: £16,000

Start Date: 09/11/2008 00:00:00

Leave Date:

Active Status: ☒

Record Created: 17/04/2022 22:28:30

[Save Employee](#)

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

Employee No: 0011

Name: Bruce Wayne

Phone No: 0123456789

Email: bruce@wayne.co.uk

Job Title: Sales Manager

Salary Amount: £300,000,000

Start Date: 09/11/2008 00:00:00

Leave Date: 01/01/2021 00:00:00

Active Status: ☒

Record Created: 17/04/2022 22:28:30

[Save Employee](#)

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

The screenshot displays the 'EmployeeManagement' application interface. The top section shows the 'Edit Employee' form with fields for Employee No, Name, Phone No, Email, Job Title, Salary Amount, Start Date, Leave Date, and Is Active. A modal window indicates the employee has been updated. Below the form, a SQL query 'select * from Employee' is shown, followed by a table of results. The bottom section shows the 'Employee Overview' page with a table listing all employees.

SQL Query Results:

ID	EmployeeNo	Name	PhoneNo	Email	JobID	SalaryAmount	StartDate	LeaveDate	IsActive	DateCreated
1	0008	Joe Bloggs	NULL	NULL	1	£20,000	2021-04-04 00:00:00.000	NULL	1	2020-01-01 00:00:00.000
2	0009	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	0010	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	0011	Bruce Wayne	0123456789	bruce@wayne.co.uk	1	£200,000,000	2009-11-00 00:00:00.000	2021-01-01 00:00:00.000	1	2022-04-17 22:26:33.193

Employee Overview Table:

Actions	Employee No	Name	Phone No	Email	Job Title	Salary Amount	Start Date	Leave Date	Is Active	Date Created
	0008	Joe Bloggs			Sales Manager	£20,000	04/04/2021		True	01/01/2020
	0009	Sarah Smith	+44 123456789	sarah@yahoo.com	Sales Assistant Manager	£19,000	04/06/2021		True	12/04/2021
	0010	Clark Kent		clark@outlook.com	Sales Associate	£18,000	04/06/2021	19/01/2022	False	28/02/2022
	0011	Bruce Wayne	0123456789	bruce@wayne.co.uk	Sales Manager	£200,000,000	06/11/2008	01/01/2021	True	17/04/2022

End of Document