

Deliverable 2

Employee Management System

Team Members: Ritesh Sharma / Sajjan Gautam

Second part of the Project Report (Employee Management Project) for the Desktop Application Development

1.) Overview (Summary of the project so far)

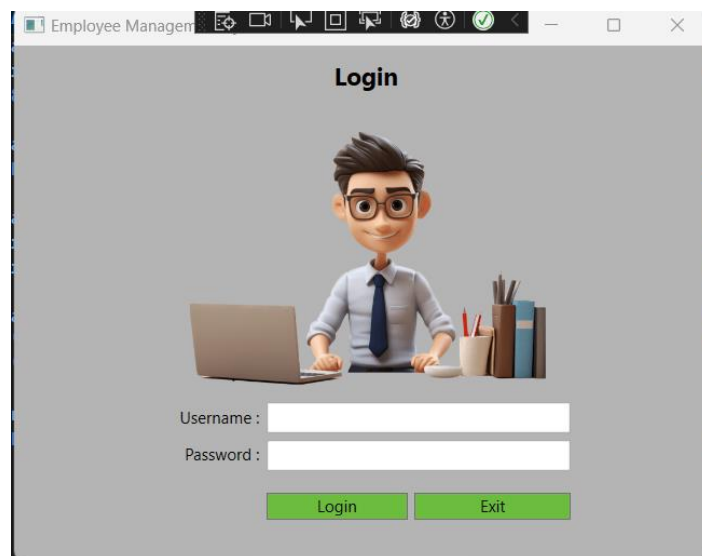
We have developed the main components of the Employee Management System, including login functionality, admin functionalities for managing employee data, and a basic employee interface. The following functionalities are implemented.

- Login Screen: Users can log in as either an admin or an employee.
- Admin Interface: Admins can view, add, update, and delete employee information.
- Employee Interface: Employees can view their details.

2.) Screenshots of the Application

- **Main Window**

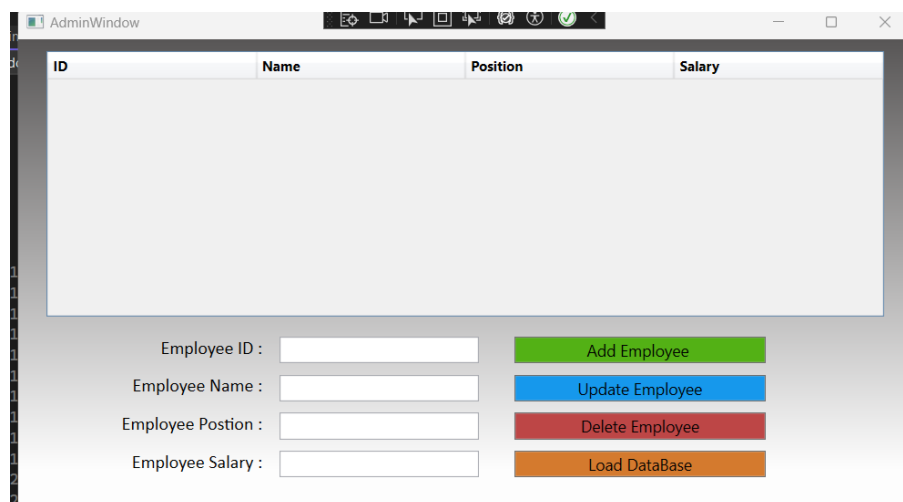
When the application is started this is the first page you will see.



In this page you will need to enter the valid credentials for either the administrator or an employee. When you click on login after you enter the username and password you will be redirected to your respective page (Admin/Employee).

- **Admin Window**

When you enter the valid credentials for an admin in main window you will be redirected to this window.

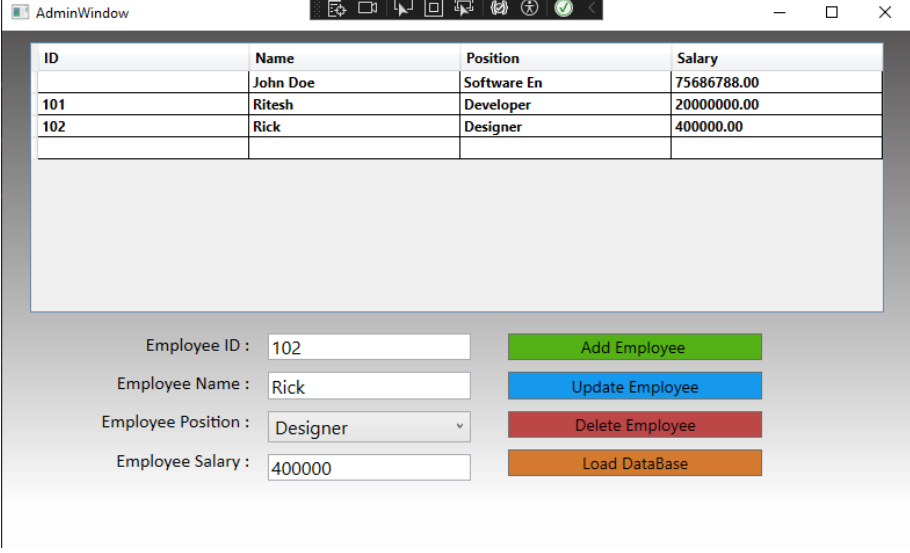


User have four options in the Admin Window:

1. Add Employee Information.
2. Update Employee Information.
3. Delete Employee Information.
4. Load the Employee Information Database

1. Add Employee Information:

Admin can enter the Employee Information and click on Add Employee Button. The information gets saved in the MYSQL database



The screenshot shows a window titled "AdminWindow" with a table of employee data and a form below it. The table has columns for ID, Name, Position, and Salary. The form has input fields for Employee ID, Employee Name, Employee Position (a dropdown), and Employee Salary, along with four buttons: Add Employee (green), Update Employee (blue), Delete Employee (red), and Load DataBase (orange).

ID	Name	Position	Salary
	John Doe	Software En	75686788.00
101	Ritesh	Developer	20000000.00
102	Rick	Designer	400000.00

Employee ID :

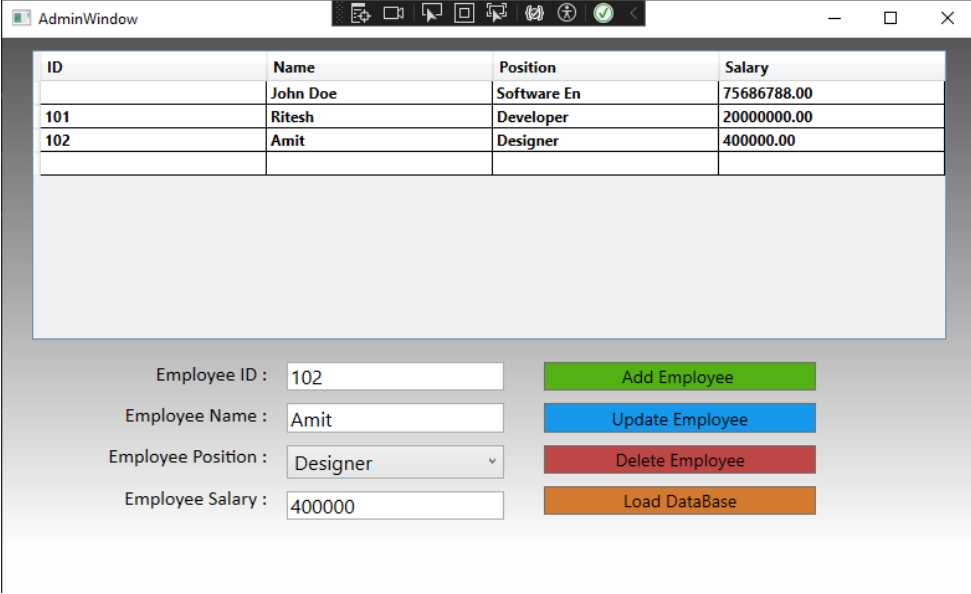
Employee Name :

Employee Position :

Employee Salary :

2. Update Employee Information.

Admin can update the employee information based on the employee ID.



The screenshot shows the same "AdminWindow" window, but the employee data has been updated. In the table, the name for ID 102 is now "Amit" instead of "Rick". In the form, the Employee Name input field now contains "Amit". All other elements, including the table structure, form fields, and buttons, remain the same.

ID	Name	Position	Salary
	John Doe	Software En	75686788.00
101	Ritesh	Developer	20000000.00
102	Amit	Designer	400000.00

Employee ID :

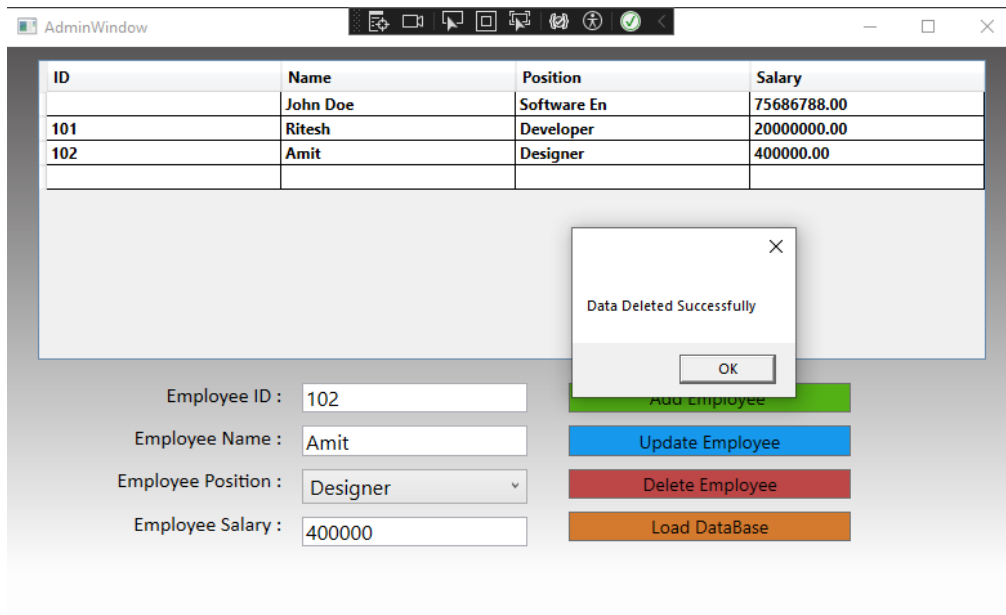
Employee Name :

Employee Position :

Employee Salary :

3. Delete Employee Information.

Admin can delete employee information and entry in the MYSQL Database using the employee ID.



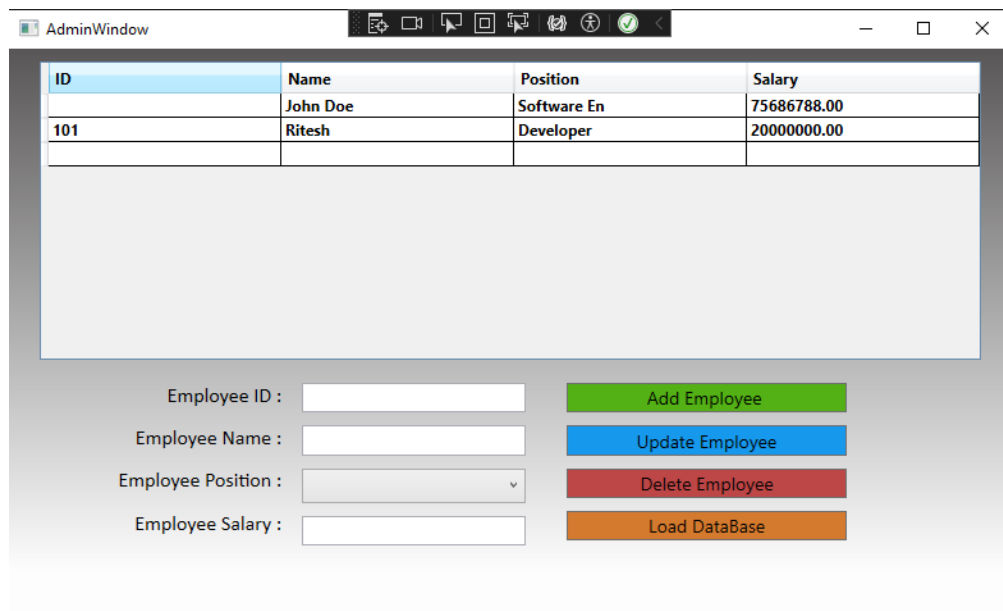
The screenshot shows the AdminWindow interface. At the top, there is a table with employee data:

ID	Name	Position	Salary
	John Doe	Software En	75686788.00
101	Ritesh	Developer	20000000.00
102	Amit	Designer	400000.00

Below the table, there is a modal dialog box with the text "Data Deleted Successfully" and an "OK" button. Underneath the dialog, there are input fields for Employee ID (102), Employee Name (Amit), Employee Position (Designer), and Employee Salary (400000). To the right of these fields are four buttons: "Add Employee" (green), "Update Employee" (blue), "Delete Employee" (red), and "Load DataBase" (orange).

4. Load the Employee Information Database

Admin can load the information store in the MYSQL database using the Load Database Button



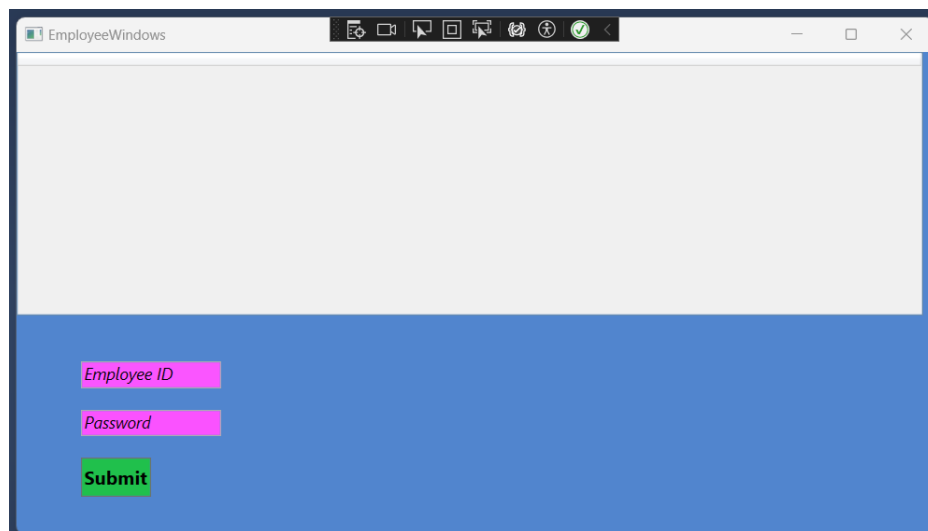
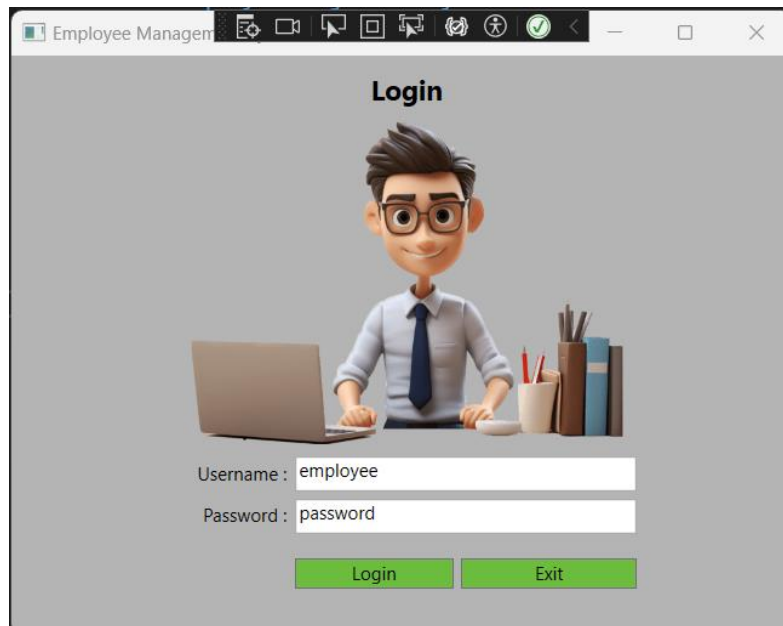
The screenshot shows the AdminWindow interface. At the top, there is a table with employee data:

ID	Name	Position	Salary
	John Doe	Software En	75686788.00
101	Ritesh	Developer	20000000.00

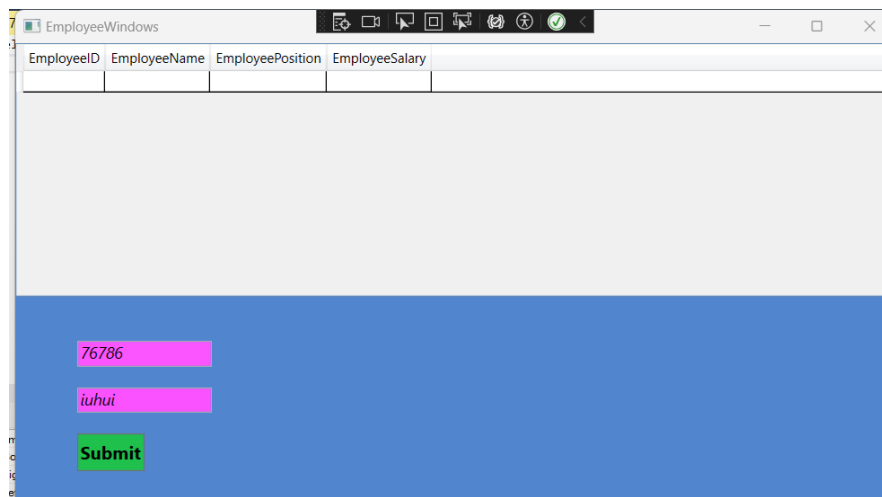
Below the table, there are input fields for Employee ID, Employee Name, Employee Position, and Employee Salary. To the right of these fields are four buttons: "Add Employee" (green), "Update Employee" (blue), "Delete Employee" (red), and "Load DataBase" (orange).

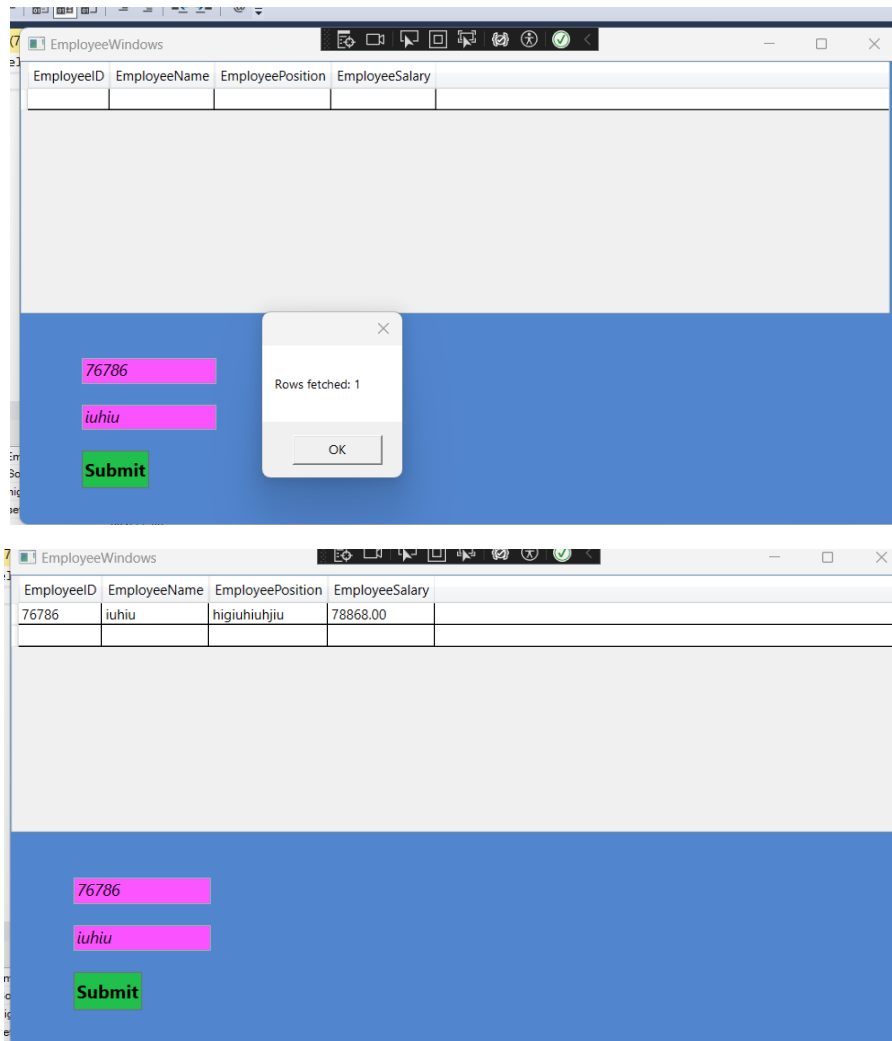
- **Employee Window**

When you enter the valid credentials for an employee in main window you will be redirected to this window.



If an employee enters their credentials, they are able to view their stored employee information including their current position and salary.





- **Database Integration**

We are using a SQL Server database to store employee information. Below are the screenshots of the database table and entries:

100 %				
		Results	Messages	
	EmployeeID	EmployeeName	EmployeePosition	EmployeeSalary
1	NULL	John Doe	Software En	75686788.00
2	76786	iuhiu	higiuiuhjju	78868.00
3	312423	esrff	sefsef	4567456.00
4	233241	ewrewr	ewrwr	234555.00
5	35466534	gregrsdgr	rsdgrsg	3454.00

3.) Tools Finalized

Backend

Language: C#

Framework: .NET Framework 4.8

Frontend

Technology: WPF (Windows Presentation Foundation)

Database

Type: SQL Server 2022

Version: 16.0.1000.6.

API

Framework: ASP.NET Core Web API

Version Control

Platform: GitHub

Repository: [Application-Development-Repository](#)

Additional Tools

IDE: Visual Studio 2022 (For code editing)

Database Management: MSSQL

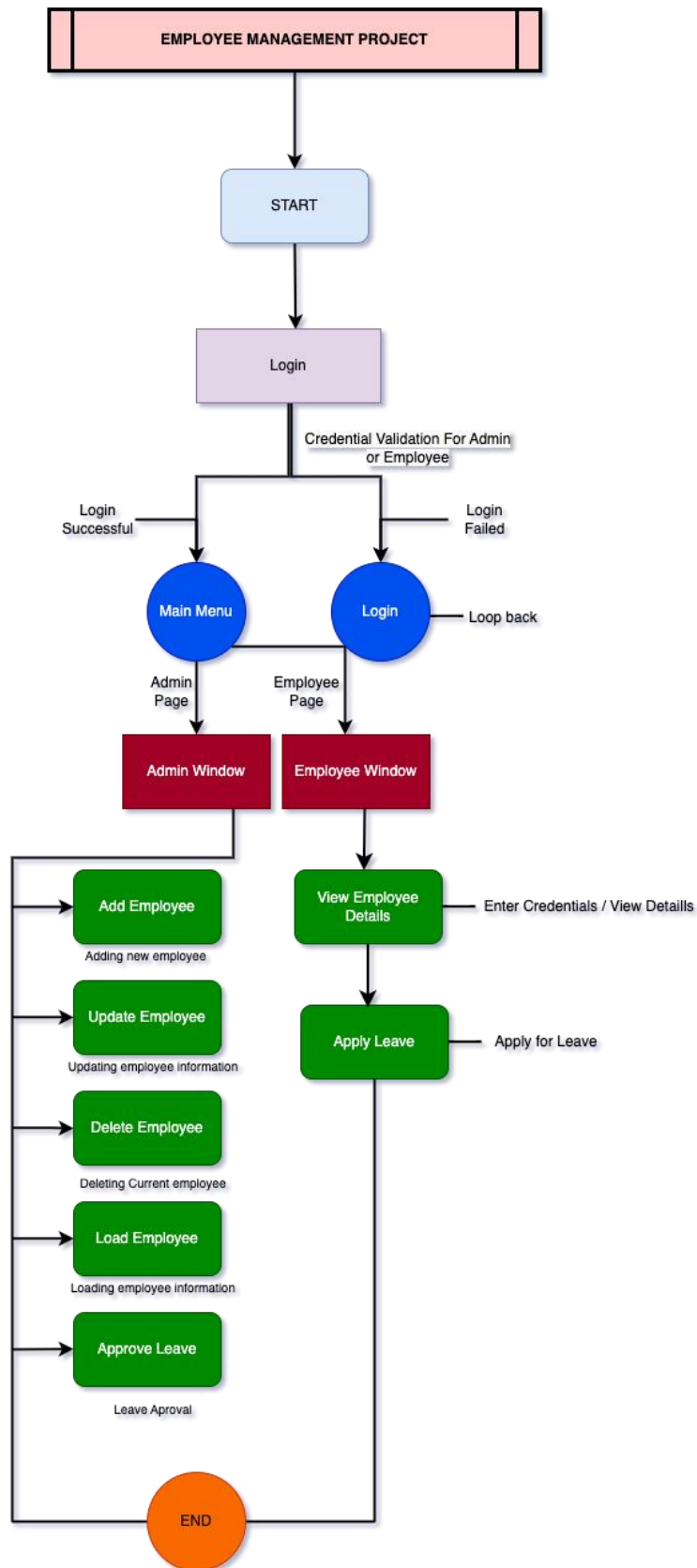
4.) Plan

Working Plan for the Next 6 Weeks

The following table outlines our working plan for the next 6 weeks, including the migration to a REST API-based database and the generation of test programs.

Week	Task Assigned To: Sajjan Gautam	Task Assigned To: Ritesh Sharma
1	Completing remaining UI elements for employee functionalities	Adding Employee position Combo Box list
2	Integrating employee authentication and authorization mechanisms	Generating employee ID automatically
3	Checking the connection and implementing features like data validation and error handling	Making the connection between adminWindow and employeeWindow
4	Reviewing the code and making changes if necessary	Adjusting the code and adding more features
5	Migrating the database to REST API based database	Migrating the database to REST API based database
6	Generating test programs to test the application and debugging	Generating test programs to test the application and debugging

5.) Flowchart:



6.) Class Diagram:

