A Project Report On

PayPilot : A Desktop Payroll Management System

Developed By

Vijay Manani

Submitted To

Geetanjali Collage of Computer Science & Commerce (BBA)

Saurashtra University,
Rajkot.
Academic Year : 2024-25

Guide Name
PROF. Mital Goswami

Acknowledgement

- I am pleased to submit my project titled "PayPilot : A Desktop Payroll Management System" to Saurashtra University, Rajkot as part of the requirements for the BCA degree in the Computer Applications branch.

- I extend my sincere gratitude to Prof. Brijesh Shah, Head of the Department, and all the esteemed faculty members of the Department of Computer Science for their invaluable guidance and unwavering support throughout this journey.

- I am also thankful to the distinguished authors of various books, research papers, and blogs that I referred to during the development of this project. Their works served as an essential foundation for my understanding and progress.

- A special note of appreciation goes to my parents, whose constant encouragement and support have been instrumental in not only the completion of this project but also in every endeavor I undertake in life. I am deeply grateful to my family for standing by me.

- Gratitude is a feeling that comes from the heart, and even the smallest timely help can make a big difference in life. For every piece of guidance and support I received, I extend my heartfelt thanks.

PREFACE

- This project report is a culmination of my efforts during the 5th semester of the Bachelor of Computer Applications (BCA) program. It has been an enlightening experience that highlighted the difference between theoretical learning and its practical application. While theoretical knowledge lays the foundation, it is through hands-on project work that the true depth of understanding is gained.

- The project titled **PayPilot** is based on **Payroll Management System** which I selected after careful study of the subject. This project gave me the opportunity to explore software development in real-world scenarios and implement the principles of software development that I have studied throughout the course.

- Working on this project has been both challenging and rewarding. I realized the importance of practical training in the field of software development, as it enables us to apply the concepts we have learned in class. The computerized systems we developed offer efficiency, speed, and accuracy that go far beyond traditional manual processes, making them vital in today's tech-driven world.

- This project report provides an overview of the entire development process, from design to implementation. The report includes various figures and technical details that will allow anyone with a basic understanding of software development to follow the content easily. I believe this experience has strengthened my skills and prepared me for future endeavors in this field.

- This project was completed as part of the curriculum of the **5th semester of BCA at Geetanjali Group of Colleges**. I hope this report will serve as a useful reference for anyone interested in understanding the technical aspects of the project.

Index

Project Profile

PayPilot is a robust desktop payroll management system designed to streamline and simplify payroll processes for organizations of all sizes. Built using the C# programming language and a MySQL relational database, Payroll management system offers a user-friendly interface for managing employee information, calculating salaries with utmost accuracy, generating detailed payslips, and creating comprehensive reports for informed decision-making.

- **Key-Features**

  - ✓ **Employee Management:**

    Add, edit, and manage details of employee records including name, designation, department, date of birth, date of joining, contact information, and Basic Salary.

  - ✓ **Attendants Records:**

    Add month-wise attendants including total working days, leave and payable days and also, we can delete attendants if wrong record inserted.

  - ✓ **Salary Processing:**

    Automatically calculate salaries based on predefined rules, basic-pay (Gross pay), allowance (Incentive, HRA) and deductions (PF, ESI).

  - ✓ **Pay-slip Generation:**

    Generate detailed and professional pay-slips for each employee, reflecting gross pay, all allowances and deductions, net pay.

- **Benefits**

  - ➢ **Enhanced Accuracy and Efficiency -** Payroll processes with automated calculations minimize errors and save valuable time compared to manual calculations.

  - ➢ **Improved Organization and Accessibility -** Centralized and secure storage of employee data within a MySQL database ensures easy access and data consistency

System Requirement

Note - To implement this project, the following system hardware and network are required.

➢ Hardware Requirements

Processor	Intel Core i3 or higher (or equivalent)
RAM	Minimum 4 GB

➢ Software Requirements

Operating System	Windows 7 or higher
Programming Language	C#
Database	MySQL
IDE	Visual Studio 2022
Framework	.NET Framework 4.7.2

Platform Specification

- **Development Framework:**

  The payroll management system PayPilot, will be developed as a standalone desktop application using the .NET Framework. The .NET Framework offers a stable and feature-rich environment for building Windows desktop applications, enabling robust functionality and ease of deployment for small to medium-sized organizations.

- **Programming Language:**

  The application will be coded in C#, a powerful and versatile object-oriented programming language. C# is well-suited for desktop application development on the .NET platform, ensuring scalability, maintainability, and strong performance for managing payroll operations.

- **Database Management System (DBMS):**

  The system will use **MySQL** as the relational database management system (RDBMS). MySQL is known for its performance, reliability, and ease of use, making it an ideal choice for managing employee data, payroll calculations, and other related information in the payroll system. It also offers seamless integration with C# through compatible data connectors.

- **Desktop Application Framework:**

  Windows Forms will be used as the user interface (UI) framework for developing the desktop application. Windows Forms provides a user-friendly environment for building interactive and responsive user interfaces, allowing the easy creation of input forms, tables, and reports for payroll management tasks.

- **Deployment Environment:**

  The PayPilot application will be deployed on Windows-based systems, ensuring compatibility with .NET Framework applications. It can be deployed either on individual desktops or within a local network where a central database server hosts the MySQL database and multiple client machines access the application.

Data Dictionary

- **Admin**

	Name	Data Type	Allow Nulls	Default
🔑	admin_uid	varchar(50)	☐	
	admin_pwd	varchar(50)	☐	
			☐	
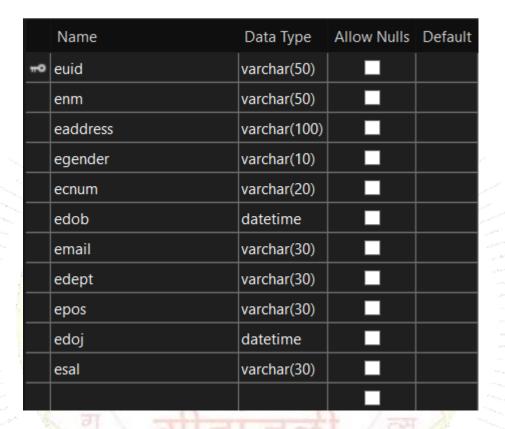
- **Attendants**

Name	Data Type	Allow Nulls	Default
atuid	varchar(50)	☐	
atnm	varchar(50)	☐	
atpayperd	datetime	☐	
attotal	varchar(50)	☐	
atpres	varchar(50)	☐	
atabs	varchar(50)	☐	
atleave	varchar(50)	☐	
		☐	

- **Salary**
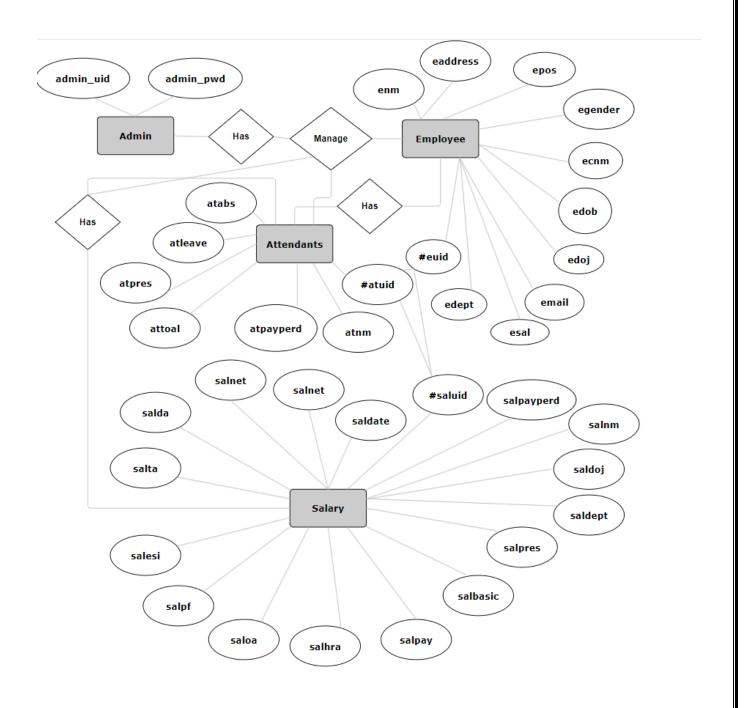
Name	Data Type	Allow Nulls	Default
saluid	varchar(30)	☐	
salpayperd	datetime	☐	
salnm	varchar(30)	☐	
saldoj	varchar(30)	☐	
saldept	varchar(30)	☐	
salpres	varchar(30)	☐	
salbasic	varchar(30)	☐	
salpay	varchar(30)	☐	
salhra	varchar(30)	☐	
salince	varchar(30)	☐	
saloa	varchar(30)	☐	
salpf	varchar(30)	☐	
salesi	varchar(30)	☐	
salta	varchar(30)	☐	
salda	varchar(30)	☐	
salnet	varchar(30)	☐	
saldate	datetime	☐	
		☐	

ગીતાંજલી કૉલેજ

- **Employee**

Name	Data Type	Allow Nulls	Default
euid	varchar(50)	☐	
enm	varchar(50)	☐	
eaddress	varchar(100)	☐	
egender	varchar(10)	☐	
ecnum	varchar(20)	☐	
edob	datetime	☐	
email	varchar(30)	☐	
edept	varchar(30)	☐	
epos	varchar(30)	☐	
edoj	datetime	☐	
esal	varchar(30)	☐	
		☐	
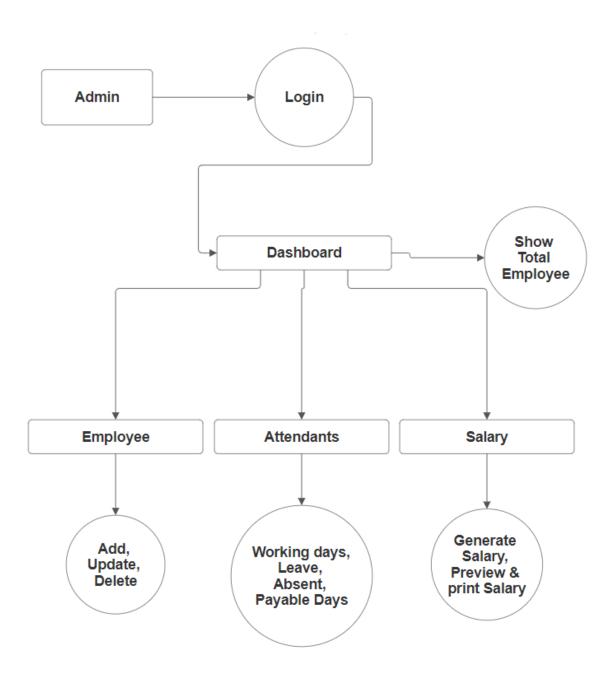
ER Diagrams

Data Flow Diagram

- **Context-Level DFD**

SDLC

- **Requirement Gathering and Analysis**

  Understand the specific needs of the payroll system. This includes identifying essential features such as employee management, attendance tracking, salary generation, payslip generation, role-based access control, and security measures.

- **Feasibility Study**

  Assess the technical, operational, and financial feasibility of developing PayPilot using C# and MySQL.

- **System Design**

  Design the architecture, database schema, and user interface for PayPilot.

- **Development**

  Build the core components of PayPilot based on the design specifications. Create the user interface using C# forms or other GUI tools. Implement the business logic for handling employee records, calculating salaries, tracking attendance, and generating payslips. Set up MySQL for storing employee, attendance, and salary data, with appropriate indexes, constraints, and relationships.
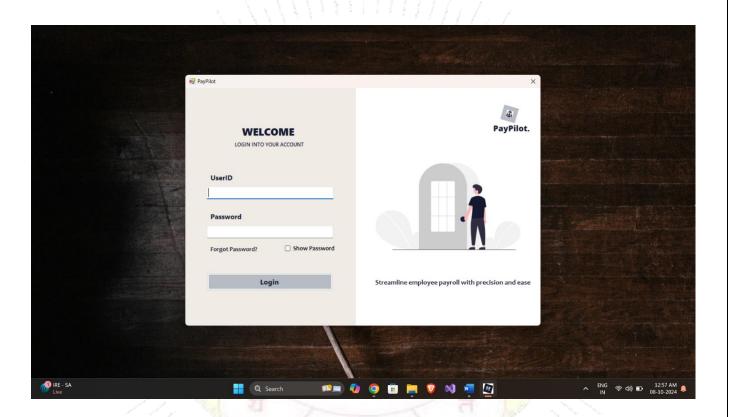
- **Testing**

  Ensure that the system functions correctly, is secure, and meets the requirements.

- **Deployment**

  Deploy PayPilot to a live production environment where HR and other authorized personnel can use it.
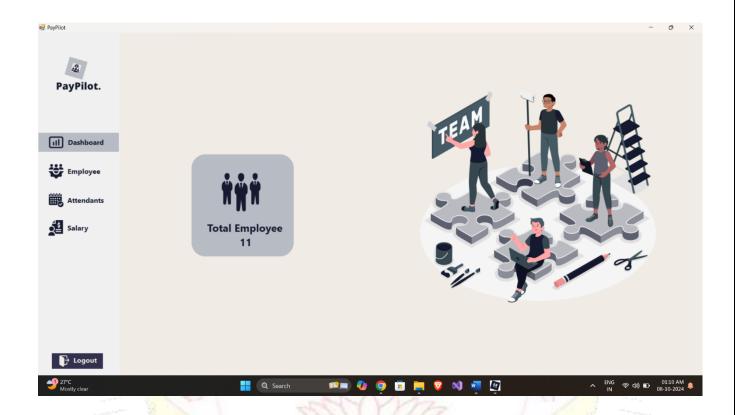
Screen Shots

- **Login Page**

> **User can log into the software by entering their user ID and password.**

> **If the entered credentials are correct, the user will successfully gain access to the system.**

> **User has the option to view the entered password by selecting the "Show Password" checkbox.**
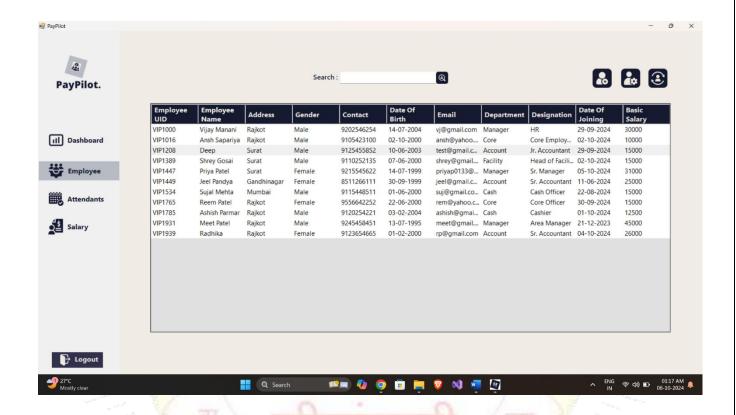
- **Dashboard Page**

- ➢ **On the Dashboard, user can view the total number of employees that have already been added to the system.**
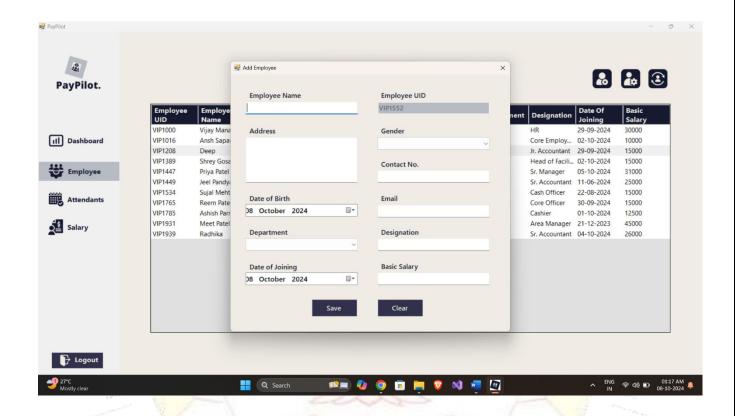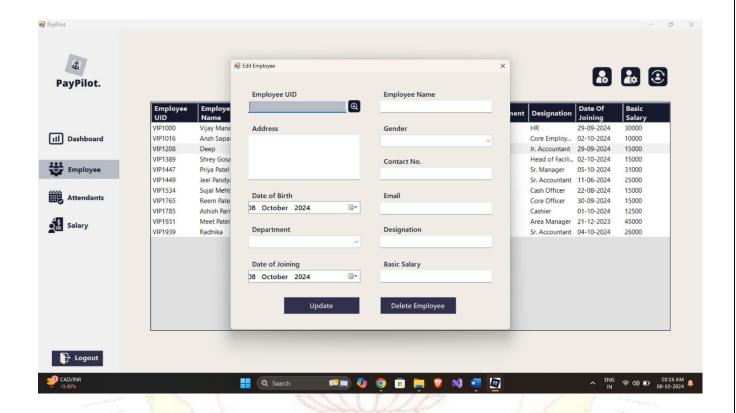
- **Employee Page**

> **On the Employee Page, user can view detailed information for all employees, including Employee UID, Name, Contact Information, Birth Date, Joining Date, Designation, Department, and Basic Salary.**

> **User can also search for specific employee details using the search box.**
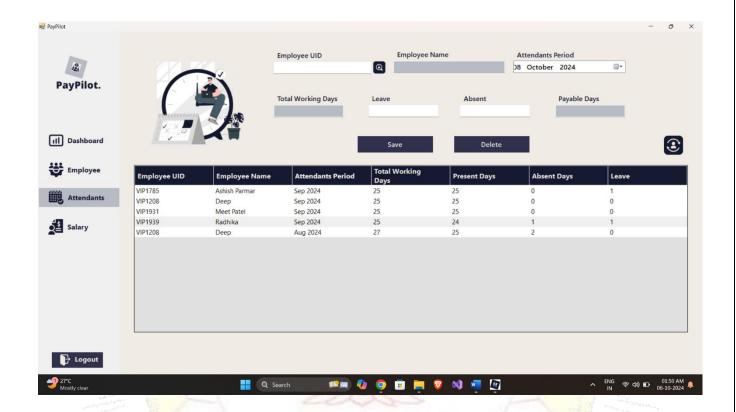
- **Employee Page >**

> **In Add Employee Form, user can add a new employee by entering details such as Name, Address, Gender, Contact Information, Birth Date, Joining Date, Department, Designation, and Basic Salary.**

> **Employee UID is automatically generated by the system.**
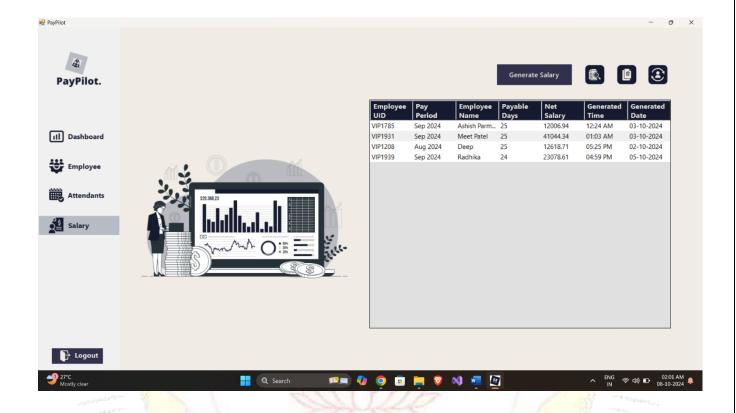
- **Employee Page >**

> **In Edit Employee Form, user can modify employee details if corrections are needed or if incorrect information was previously entered.**

> **User can also delete an employee within the Edit Employee Form.**

> **Employees can be edited or deleted by entering the Employee UID. Once the UID is entered, all corresponding employee details will be displayed in the form for review or modification.**
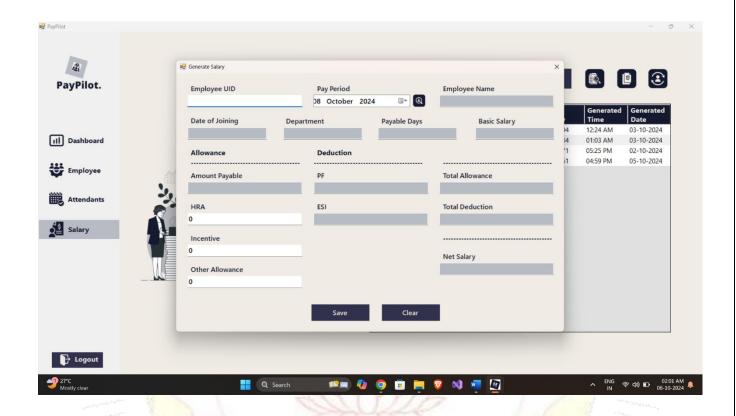
- **Attendants Page**

> **In the Attendants Page, the user can record attendance using the Employee UID. If attendance has already been recorded, both the attendance details and the employee's name will be displayed. If no attendance is recorded, only the employee's name will be shown.**

> **User can select an attendance period to view the total working days for that period. Leave and absence information can be entered, and the system will then calculate the payable days for the selected period.**

> **User also has the option to delete attendance records if they exist for the selected period. All attendance records can be viewed on the Attendants Page.**

- **Salary Page**
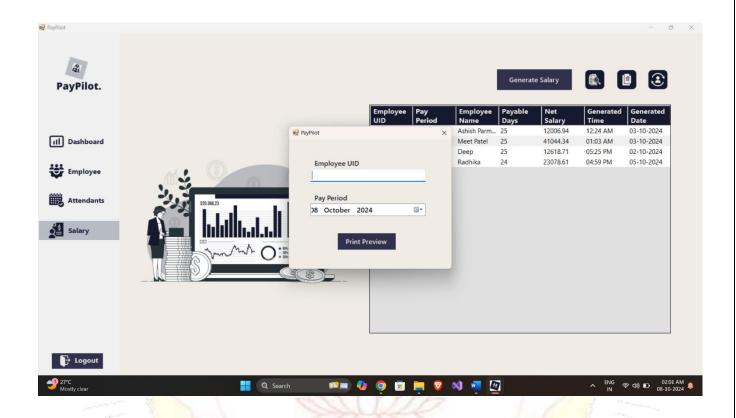

> In the Salary Form, user can generate salary only if attendance records exist; otherwise, the salary will not be generated.

> User can view all generated salary details on the Attendants Page, including the time and date of generation.

> User also has the option to delete a generated salary if an incorrect pay period or wrong details were entered.

- **Salary Page > Generate Salary**

> ➤ **User can generate the salary by entering the Employee UID and pay period. Once the UID and pay period are entered, the system will automatically display details such as the employee's name, joining date, department, payable days, and basic salary.**

> ➤ **The system will automatically calculate the payable amount, PF (Provident Fund), and ESI (Employee State Insurance). The user can manually enter additional allowances such as HRA (House Rent Allowance) and incentives. The net salary will be calculated after entering all relevant details.**
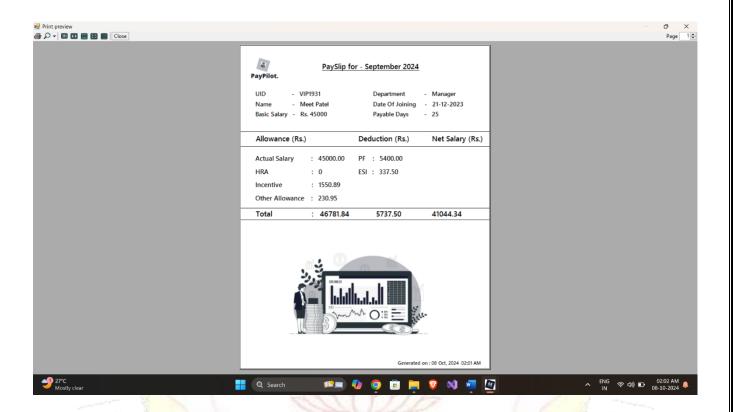
- **Salary Page >**

- ➢ **If the user wants to generate a payslip for a specific pay period, they can do so by entering the Employee UID and Pay Period.**

- ➢ **If a record exists for the entered pay period, the payslip will be generated and displayed.**

- **Print Preview Page**

> **After entering the Employee UID and Pay Period, the user will receive the payslip. The payslip will display employee details, payable days, allowances, deductions, and the net salary.**

> **The user has the option to save the payslip in PDF format.**

FUTURE REQUIREMENTS OF PROJECT

Anticipating future requirements for a PayPilot : Payroll Management System project is crucial for ensuring its long-term effectiveness and adaptability. Future requirements may include:

- **Mobile Application Support**:

  Incorporate mobile-friendly features or develop a mobile application for on-the-go access to payroll details, payslips, attendance records, and salary generation for employees and HR personnel.

- **Advanced Reporting and Analytics**:

  Implement advanced reporting and data analytics capabilities to generate detailed payroll insights, financial trends, and employee performance reports. This can assist in making strategic decisions for better workforce management.

- **Cloud-Based Solution**:

  Transition PayPilot to a cloud-based solution to allow for easier scalability, enhanced data backup, and remote access for multiple users, providing increased flexibility and reliability.

- **Integration with Biometric Devices**:

  Consider integrating biometric devices for attendance tracking, such as fingerprint or facial recognition systems, to further automate attendance and reduce errors in payroll calculations.

- **Role-Based Access Control Enhancements**:

  Improve role-based access control (RBAC) by allowing for more granular permission settings based on employee levels, such as providing specific HR roles with access to only relevant payroll details.

LIMITATION OF PROJECT

- **No Built-in Web Framework**:

  Developing the software without using ASP.NET may make software development more time-consuming. It would require manual handling of routing, session management, and other web-related functionalities, increasing the complexity of the system.

- **Limited Scalability**:

  Although C# and MySQL are reliable technologies, there may be limitations in scalability if the system is not designed with high volumes of data and users in mind. As employee data grows, it may require performance optimization and tuning, which could add complexity.

- **Lack of Cross-Platform Support**:

  While C# is supported across platforms through .NET Core, developing without ASP.NET may limit the ease of creating cross-platform web applications. This may require additional efforts if you wish to make the system accessible on both Windows and non-Windows platforms.

- **Maintenance and Updates**:

  Maintaining the system in the long run will require continuous manual updates and monitoring. Developers will need to ensure that the system stays up to date with security patches and new features, which could be time-consuming.

Test Cases

NO.	Test Case	Description	Precondition	Expected Result	Result
1.	User Login with Correct Credentials in **Login Page**.	User log into the software by entering correct User ID and password.	User has valid credentials in the system.	User is successfully logged into the system and redirected to the dashboard.	Pass
2.	User Login with Incorrect Credentials in **Login Page.**	User enters an incorrect User ID or password.	User has invalid credentials.	An error message appears indicating invalid login credentials.	Pass
3.	Search for Specific Employee in **Employee Page.**	User searches for an employee using the search box.	Employees have been added to the system.	The matching employee details are displayed.	Pass
4.	Add New Employee in **Add Employee Form**	User adds a new employee using the Add Employee Form.	Employee not exists in the system.	The new employee is successfully added to the system with an automatically generated Employee UID.	Pass
5.	Edit Employee Information in **Edit Employee Form**	User edits employee details.	Employee exists in the system.	The employee details are updated in the system.	Pass
6.	Delete Employee in **Edit Employee Form**	User deletes an employee from the system.	Employee exists in the system.	The employee is deleted from the system.	Pass
7.	Record Attendance in **Attendants Page**	User records attendance for an employee using the Employee UID.	Employee exists in the system.	Attendance is recorded for the employee, and attendance details are displayed if already recorded.	Pass
8.	Generate Salary in **Salary Page**	User generates a salary for an employee based on attendance records.	Employee and attendance records exist for the selected pay period.	Salary is generated, including PF, ESI, HRA, and other calculations.	Pass
9.	Generate Payslip in **Salary Page**	User generates a payslip for an employee for a specific pay period.	Salary record exists for the selected period.	Payslip is generated and displayed for the employee.	Pass
10.	Delete Generated Salary in **Salary Page.**	User deletes a generated salary.	Salary records exist for the employee.	The salary record is deleted.	Pass

BIBLIOGRAPHY

- **UI Colors**

  https://colorhunt.co/palette/

- **Icons**

  https://www.flaticon.com/

- **illustrations Images**

  https://undraw.co/illustrations
  https://storyset.com/illustration

- **GitHub**

  https://github.com/