

Pearson

Higher National in

Computing

Unit 4: Database Design &
Development

ASSIGNMENT

NAME: MUNEEB IFTIKHAR



Table of Contents

Database Design and Planning	4
Introduction to Database Design	4
Problem Definition and Business.....	4
Project Scope and Boundaries	4
Assumptions and Constraints	5
Justification of Database Approach	6
Success Criteria for the Database Design	6
Identification of User Requirements	6
Identification of System Requirements	8
System Requirements Table (Functional + Non-Functional).....	8
Identification of Entities and Attributes	10
Relationships Between Tables	13
Entity Relationship Diagram (ERD)	15
Normalization and Data Integrity	16
Validation Rules and Constraints	18
Design Evaluation and Justification	18
Database Development and Implementation.....	19
Introduction to Database Development.....	19
Setting Up the Database Server Using XAMPP.....	20
Creating the Database Using phpMyAdmin	20
Implementing Database Tables in MySQL.....	21
Inserting Sample Data Using SQL.....	25
Establishing ODBC Connectivity Between MySQL and MS Access	26
Linking MySQL Tables in Microsoft Access	27
Front End Validation in Microsoft Access	28
Data Validations	29
Evaluation of Development Tools and Approach	31
Security and Maintenance Considerations.....	32

Access Front-End File Submission	33
Introduction to Queries and Testing	33
Development of SQL Queries	34
Use of Queries to Meet User Requirements	37
Testing Strategy and Test Planning	37
Test Plan Table	38
Test Data Explanation	39
Evaluation of Test Results and Effectiveness	40
Reflection on Improvements Identified Through Testing	40
Technical Documentation, User Documentation, and System Evaluation	41
Purpose of Documentation	41
Technical Documentation System Overview	41
Technical Documentation Database Structure and Maintenance	42
Technical Documentation Security Considerations	44
 Technical Documentation Installation Guide	45
User Documentation System Usage Overview	45
User Documentation Generating Reports and Forms	46
 Reports	47
 Forms:	49
 Errors Form Validation:	52
Evaluation of the System	53
Recommendations and Future Improvements	55
Final Conclusion	56
References	57

Database Design and Planning

Introduction to Database Design

Planning and designing of the database was the first phase of designing the Attendance Management System of NetSolace Inc. It was necessary to have a complete understanding of the business problem and convert it into a meaningful database solution before the software and tools were put to use. It is important to ensure that there is a carefully-designed database as that is what the whole system is based on. Weaknesses during this stage would have an impact on development, testing and expansion of the system in the future.

Being a Trainee Database Administrator, I was to analyse the needs of the attendance system, what the data to be stored should be, and create a database structure that will be efficient, reliable and scalable. This included finding users and their requirements, conclusion on the entities that are necessary, how the entities would be associated to one another and that the final design would be based on the principles of relational database.

Problem Definition and Business

The purpose of this project is to design and develop a relational database system for **NetSolace Inc.** to manage employee attendance captured using biometric authentication methods such as fingerprint, RFID, and PIN. The organisation currently requires a structured system that can reliably store attendance events, link them to employees and devices, and support payroll-related reporting.

The existing manual or semi-automated methods of attendance tracking are prone to data redundancy, errors, and difficulties in generating accurate management information. As the number of employees and attendance records increases, these limitations negatively impact operational efficiency, payroll accuracy, and managerial decision-making.

This database system is therefore intended to provide a **centralised, secure, and scalable solution** that supports daily HR operations, management reporting, and future technological expansion.

Project Scope and Boundaries

In Scope

The database system will:

- Store employee personal and employment details

- Record daily attendance events with date, time, authentication method, and In/Out status
- Store biometric device information and operational status
- Store salary slip data linked to attendance and pay periods
- Support querying and reporting for HR staff and management
- Be implemented using **MySQL (backend)** and **Microsoft Access (frontend)**

Out of Scope

The database system will **not**:

- Directly integrate with live biometric or IoT devices
- Handle real-time attendance capture
- Include full payroll processing logic such as tax computation
- Be deployed as a web-based or cloud-hosted application

Defining the scope ensures that the system meets the assignment requirements without unnecessary complexity while remaining extensible for future development.

Assumptions and Constraints

Assumptions

- Attendance data is entered manually during development and testing
- HR staff have basic computer literacy and can operate Microsoft Access
- MySQL server is hosted locally using XAMPP
- Each attendance record relates to exactly one employee and one device

Constraints

- The system is developed as an academic prototype within limited time
- Microsoft Access limits concurrent multi-user access
- Security is limited to database-level permissions (no biometric-level encryption)
- Performance testing is based on sample data rather than production-scale data

These assumptions and constraints guide design decisions and justify certain technical choices made during development.

Justification of Database Approach

A **relational database model** was selected for this system due to the structured and interrelated nature of the data involved. Employees, attendance records, devices, and salary slips share clear relationships that can be effectively managed using primary and foreign keys.

MySQL was chosen as the backend database due to its reliability, performance, and industry acceptance. Microsoft Access was selected as the front-end tool because it allows rapid development of forms, queries, and reports suitable for non-technical users.

This approach supports data integrity, reduces redundancy through normalization, and ensures that the system can be scaled or migrated to more advanced platforms in the future.

Success Criteria for the Database Design

The database design will be considered successful if it:

- Stores employee, attendance, device, and salary data without redundancy
- Enforces data integrity through primary and foreign key constraints
- Prevents invalid or incomplete data entry using validation rules
- Supports accurate queries and reports for HR and management
- Can be maintained, backed up, and extended without major redesign

Identification of User Requirements

Design stage was also important in understanding of the user requirements. Various categories of users in NetSolace Inc. whose expectations and roles are differing are expected to use the attendance system.

The key users of the system are the Human Resources staff. They will ensure that they keep employee records, daily attendance and assist in salary preparation. On their part, the system has to be user-friendly and should display the information in a format that can be easily understood like reports and not in raw tabular form. The HR personnel must be able to access the history of attendance, checking whether an employee is punctual or not, and accessing information that is related to salary in an expedited and precise manner.

At a taller level, management relates with the system. They are not concerned with individual attendance entries but on the summarized information. They should know the attendance trends, the number of absenteeism occurrences, and the utilization of attendance equipments

effectively. To the management, data entry functionality is of less concern than the accuracy and readability reports.

The requirements of the system administrators and technical staff are different. They should make sure it has a stable, secure and maintainable database. New employees, devices and payroll records should be addable to the system without disrupting data on the existing ones. The administrators also need the capability of the database backup and maintenance of the data integrity in the long term.

User Requirements Table

UR ID	User Type	User Requirement (Clear Statement)	Priority	How the Database Supports It
UR1	HR Staff	Add, edit, and view employee records quickly	High	Employees table with PK EmployeeID + forms/queries
UR2	HR Staff	Record daily attendance with date/time, method, and In/Out status	High	AttendanceRecords table with validation + FK links
UR3	HR Staff	View attendance history per employee by date range	High	Queries using filters + joins Employees↔Attendance
UR4	Management	View summarized attendance trends (absent/late frequency)	Medium	Aggregate queries + grouped reports
UR5	Management	View biometric device usage counts per location	Medium	Device usage query (Devices↔AttendanceRecords)
UR6	System Admin	Add devices and deactivate devices without losing history	High	Devices table with Status field; records remain linked
UR7	System Admin	Ensure data integrity (no orphan	High	PK/FK constraints + referential integrity

		attendance/salary records)		
UR8	HR + Payroll	Generate salary summary per pay period per employee	High	SalarySlips table + salary queries/reports

Identification of System Requirements

Regarding the system, a number of technical requirements were identified. The system should be anchored on relational database model since the data entities are obviously interconnected with each other. Customer data table, employee data table, attendance data table, device data table, and salary data table have to be stored in relational tables, and not in a table.

The database should have the facility of more than one authentication system because attendance units will be attended with the help of fingerprint or RFID and PIN. This is because the attendance table will have to have a field in which the authentication method applied to a particular attendance record is stored.

Future compatibility is also another requirement of the system. This project is a prototype but in the future, the database should be furnished with the capability to automatically receive data on the attendance by the IoT devices via APIs. This affected visualizations like separating attendance information and employee information and device information being stored separately.

System Requirements Table (Functional + Non-Functional)

A) Functional Requirements (FR)

FR ID	Functional Requirement	Priority	Notes / Evidence in Design
FR1	Store employee details (name, designation, hire date, status)	High	Employees table
FR2	Store attendance records with employee + device reference	High	AttendanceRecords table with EmployeeID, DeviceID FKs
FR3	Store device details (code, location, active status)	High	Devices table

FR4	Store payroll/salary slips by pay period	Medium	SalarySlips table
FR5	Support multiple authentication methods (Fingerprint/RFID/PIN)	High	AttendanceRecords.AuthMethod constrained
FR6	Generate management information queries (attendance summary/device usage)	Medium	Aggregate SQL queries
FR7	Allow front-end entry via Microsoft Access linked tables	Medium	ODBC connection + linked tables

B) Non-Functional Requirements (NFR)

NFR ID	Non-Functional Requirement	Priority	How It Is Addressed
NFR1	Security: only authorized users can modify data	High	MySQL user accounts + role permissions + Access front-end restrictions
NFR2	Data Integrity: prevent invalid references	High	PK/FK + referential integrity (no orphan records)
NFR3	Reliability: system should preserve historical attendance/salary	High	Separate tables + stable relationships; avoid overwriting history
NFR4	Performance: queries should run efficiently for growing data	Medium	Indexing on PK/FK fields (EmployeeID, DeviceID, dates)
NFR5	Maintainability: system easy to update/extend	Medium	Normalized design + modular tables
NFR6	Backup/Recovery: database can be restored if failure occurs	Medium	phpMyAdmin export strategy (detailed in LO4 later)
NFR7	Usability: non-technical staff can operate it via forms/reports	High	Access forms + reports rather than raw table editing

Identification of Entities and Attributes

On the analyzed needs, four key entities were determined to be imperative to the system.

Employees entity is the storage facility where employee related information is stored. Attributes will be employee id, first name, last name, designation, hiring date and employment status. This is an entity that contains the personnel of the organization and in terms of the attendance and salary information as a reference.

The Attendance Records unit consists of attendance information that has been produced by biometric devices. They are attendance ID, employee ID, device ID, the date of attendance, time of attendance, authentication mode, and in/out state. All the records denote an event of attendance.

The biometric attendance units are the Devices entity. Attributes will consist of device ID, device code, location and activation status. Keeping the devices separately enables the performance follow-up of the devices.

Payroll-related information is held in the Salary Slips entity. The attributes are salary ID, employee ID, pay period start date, pay period end date, total hours worked, basic salary, overtime hours, overtime rate, gross salary and net salary.

Identification of Entities and Attributes (SIMPLIFIED VERSION)

The database design is based on identifying the main objects involved in the attendance management system and the information that needs to be stored about them. These objects are represented as **entities**, and the details stored about them are known as **attributes**.

Based on the system requirements, four main entities were identified for the NetSolace Inc. attendance system:

- Employees
- Devices
- AttendanceRecords
- SalarySlips

These entities cover all core data required for attendance tracking and payroll reporting.

1.4.1 Entities Overview Table

Entity	Purpose
--------	---------

Employees	Stores employee personal and job information
Devices	Stores details of biometric attendance devices
AttendanceRecords	Stores daily attendance events
SalarySlips	Stores salary information for each pay period

Separating data into these entities helps reduce duplication and improves data accuracy.

1.4.2 Employees Entity

Field	Data Type	Key	Description
EmployeeID	INT	PK	Unique ID for each employee
FirstName	VARCHAR		Employee first name
LastName	VARCHAR		Employee last name
Designation	VARCHAR		Job role
HireDate	DATE		Date of joining
EmploymentStatus	VARCHAR		Active or Inactive

The **Employees** table is the main reference table. Attendance and salary records are linked to employees using the EmployeeID field.

1.4.3 Devices Entity

Field	Data Type	Key	Description
DeviceID	INT	PK	Unique device ID
DeviceCode	VARCHAR		Device code
Location	VARCHAR		Device location
Status	VARCHAR		Active or Inactive

Storing device information separately allows device usage to be monitored without affecting attendance records.

1.4.4 AttendanceRecords Entity

Field	Data Type	Key	Description
AttendanceID	INT	PK	Unique attendance record

EmployeeID	INT	FK	Links to Employees
DeviceID	INT	FK	Links to Devices
AttendanceDate	DATE		Date of attendance
AttendanceTime	TIME		Time of attendance
AuthMethod	VARCHAR		Fingerprint / RFID / PIN
InOut	VARCHAR		IN or OUT

Each record in this table represents one attendance event captured by a device.

1.4.5 SalarySlips Entity

Field	Data Type	Key	Description
SalaryID	INT	PK	Salary slip ID
EmployeeID	INT	FK	Links to Employees
PayPeriodStart	DATE		Start of pay period
PayPeriodEnd	DATE		End of pay period
TotalHoursWorked	DECIMAL		Total hours worked
BasicSalary	DECIMAL		Base salary
OvertimeHours	DECIMAL		Overtime hours
NetSalary	DECIMAL		Final salary

Salary data is stored separately to keep financial information organised and secure.

1.4.6 Entity Relationships (Simple Explanation)

- One employee can have many attendance records
- One device can record many attendance entries
- One employee can have multiple salary slips over time

These relationships are shown clearly in the ERD section.

Relationships Between Tables

After identifying the entities, the relationships were established to actual relationships in the world.

An employee may have more than one attendance record, and so there exists a one-many pad between the Employees and Attendance Records. On the same note, a device can produce a large number of attendance records and this forms a one-to-many relationship between Devices and Attendance Records. Employees have a one to many relationship with salary slips because an employee may have various salary slips with time.

These associations will make sure attendance and salary data will refer to valid employees and devices at all times.

Data Flow Diagram Level 0 (Context Diagram) — Description

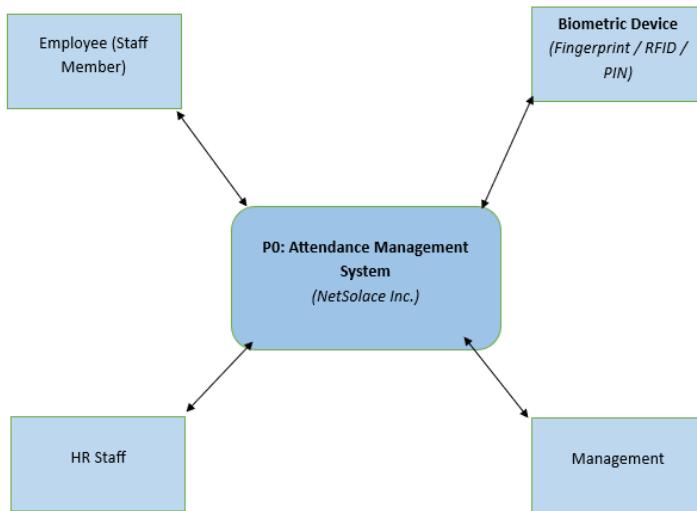
System: Attendance Management System

External Entities:

- Employee (provides attendance input via biometric device)
- HR Staff (maintains employees, views attendance, generates salary reports)
- Management (views summary reports)
- System Administrator (maintains devices, security, backups)

Main Data Flows (Level 0):

- Employee → System: Attendance Event (date/time, auth method, in/out, device)
- HR Staff → System: Employee Details / Corrections
- System → HR Staff: Attendance Reports / Salary Summaries
- System → Management: Attendance Trends / Device Usage
- Admin → System: Device Setup / Security Settings
- System → Admin: Audit/Status Information (device active, errors)



1.5.2 Data Flow Diagram Level 1 — Description

Break the system into 4 processes:

P1 Manage Employees

- Inputs: Employee details from HR
- Outputs: Employee records stored in D1 Employees

P2 Capture Attendance

- Inputs: Attendance event (EmployeeID, DeviceID, AuthMethod, Date/Time, InOut)
- Outputs: Stored in D2 AttendanceRecords
- Validation: must match existing EmployeeID + DeviceID, AuthMethod allowed list

P3 Manage Devices

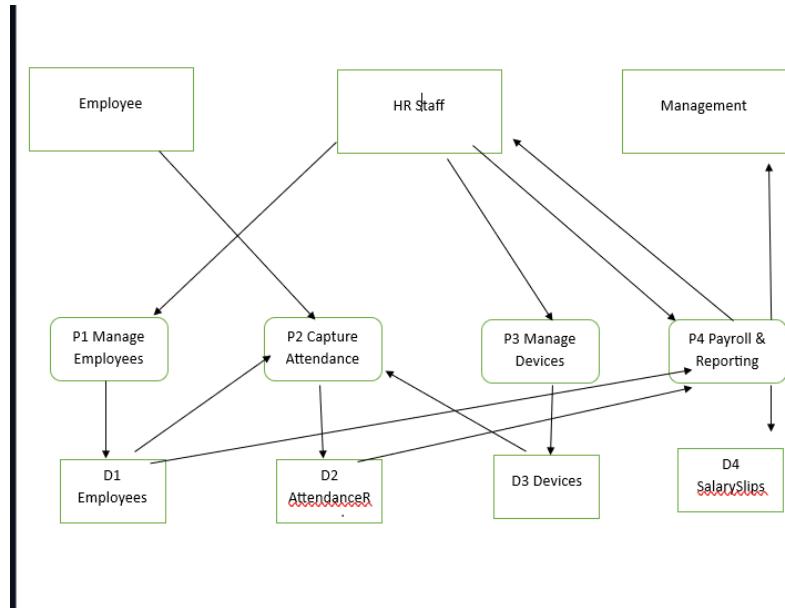
- Inputs: Device details/status from Admin
- Outputs: Stored in D3 Devices

P4 Payroll & Reporting

- Inputs: Pay period parameters (start/end), employee selection
- Outputs: Salary slip records (D4 SalarySlips) + reports for HR/Management

Data Stores (Level 1):

- D1 Employees
- D2 AttendanceRecords
- D3 Devices
- D4 SalarySlips



Entity Relationship Diagram (ERD)

Explanation of the ERD

Entity Relationship Diagram (ERD) is a visual implement of the logical design of the attendance management database and depicts how various entities relate to other entities. The ERD intended to represent the real-life processes of the biometric attendance system of the NetSolace Inc. and help to maintain the correct data storage, retrieval, and scaling.

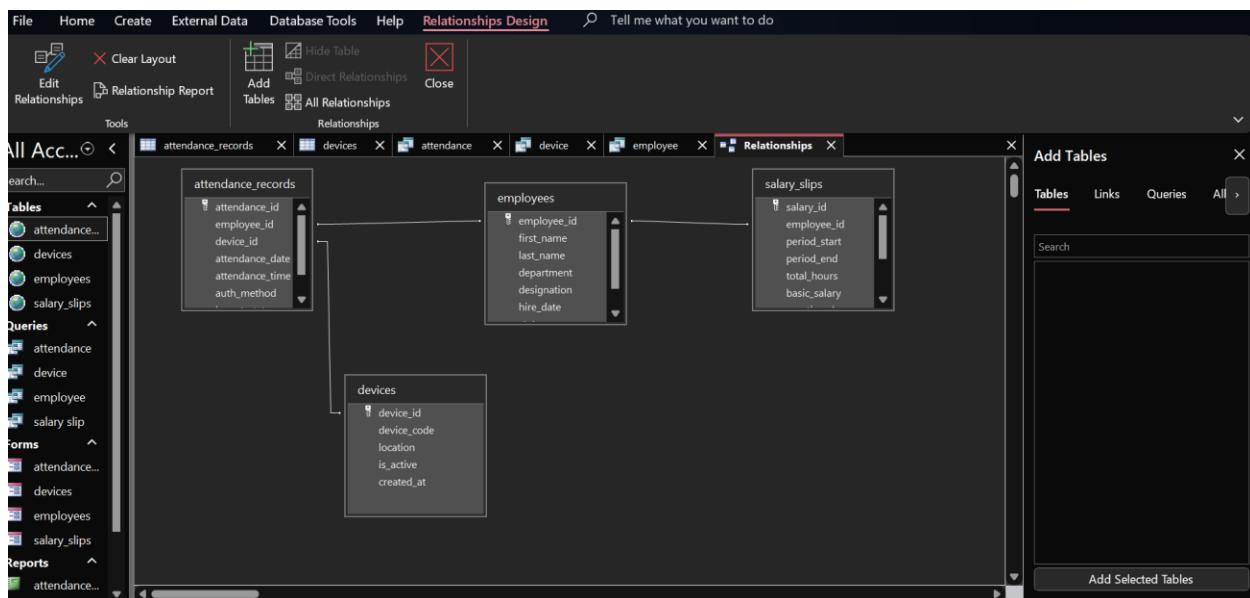
The first job of the ERD is to provide a central element around which the system will revolve in the form of the Employees entity. All workers are identified by employing employee ID that is employed in linking any attendance record and salary slip. This makes sure that there is always a valid employee when it comes to all the attendance and salary information.

Attendance Records entity, It has a one to many relationship with Employees entity. This is because, a single employee may have several attendance records though only a single attendance record belongs to one employee. This connection perfectly reflects the working nature of attendance since different times within an interval, employees are able to enter an attendance.

The Attendance Records entity also relates to the Devices entity in a one-to-many way. A single biometric device is capable of creating numerous attendance records, however, each attendance record is registered by just one device. Such relationship facilitates the monitoring of the devices and enables the system to monitor the frequency of use of each unit of attendance.

The Salary Slips entity is connected with the Employees one associated with the other via another one-to-many association. This shows that every employee is able to have several salary slips throughout his or her lifetime, one per pay period. This is because segregation of salary data and attendance data enhances the security and elegance of such data, particularly those that relate to sensitive monetary matters.

In general, the ERD is quite encouraging in demonstrating the isolation of concerns in the database and still very logical in ensuring the existence of strong logical relationships amongst entities. This structure enhances the data integrity of the information along with the prevention of duplication and at the same time enables the system to be dynamic and scalable in future without the need to consider the real-time IoT integration or an extension of the pay-roll among other features.



Normalization and Data Integrity

Normalization has been done to assure that there is no redundancy in the database design, and the data is not lost. At first, it may seem convenient to store the names of employees, details of their devices, and salary information in the attendance table. But that would mean a duplication of data and that would make the chances of having discrepancies to rise.

The database avoids record management being represented by the number of times an information is stored since it separates employees, devices, attendance records, and salary slips into separate tables. As an example, when the designation of an employee changes, one will not have to update every single attendance record; one has to update the Employees table and everything will work out on its own.

Normalization to the third normal form is also beneficial to the performance and queries. The database is also more efficient to maintain and scale up with the expansion of the organization.

Unnormalized Form (UNF)

A single table approach could look like:

Attendance_UNF

EmployeeID, EmployeeName, Designation, HireDate, DeviceCode, DeviceLocation, AttendanceDate, AttendanceTime, AuthMethod, InOut, PayPeriodStart, PayPeriodEnd, HoursWorked, BasicSalary, OvertimeHours, NetSalary

Problems:

- Repeating employee/device/salary details across many attendance rows
- Update anomalies (changing designation requires multiple updates)
- Insert/delete anomalies (can't store device without attendance)

1NF (Atomic Values + No Repeating Groups)

All fields are atomic, but redundancy remains.

Still a problem: EmployeeName/DeviceLocation repeated per attendance record.

2NF (Remove Partial Dependencies)

If we consider a composite key (EmployeeID + AttendanceDate + AttendanceTime), then fields like EmployeeName depend only on EmployeeID (partial dependency).

Fix: split into separate tables.

- Employees(EmployeeID, FirstName, LastName, Designation, HireDate, EmploymentStatus)
- Devices(DeviceID, DeviceCode, Location, Status)
- AttendanceRecords(AttendanceID, EmployeeID, DeviceID, AttendanceDate, AttendanceTime, AuthMethod, InOut)

3NF (Remove Transitive Dependencies)

Check if any non-key field depends on another non-key field.

Example:

- DeviceLocation depends on DeviceCode (not on AttendanceID) → keep in Devices only
- Salary fields depend on PayPeriod and Employee → keep in SalarySlips only

Final 3NF tables:

- Employees
- Devices
- AttendanceRecords
- SalarySlips(SalaryID, EmployeeID, PayStart, PayEnd, TotalHoursWorked, BasicSalary, OvertimeHours, OvertimeRate, GrossSalary, NetSalary)

Result: No redundancy, improved integrity, easier maintenance.

Validation Rules and Constraints

Validation rules are significant in upholding of data quality. Mandatory fields had been selected like name of the employees and the attendance dates so that it is impossible to save an incomplete record. The types of data were specifically selected in order that dates, times, and numeric data will act appropriately in computing and providing a comparison.

The authentication was limited to preset values (Fingerprint, RFID and PIN). This does ensure that inconsistent spellings or invalid values are not stored in the attendance table.

Design Evaluation and Justification

Achievement of User and System Requirements

The final database design is successful in meeting both the user and system requirement of the project at the initial stage of the project. On the user side, the design can allow the simple access of the attendance, employee, device and salary data via queries and report without interacting with raw data tables. The relational structure, in system perspective, guarantees the absence of inconsistency and errors in data storage based on the operations of an attendance management system that is rooted in biometric devices. The entire layout is also neatly designed and well organized logically and can be easily adapted to the present day as well as to expansion in the future.

Strength and Clarity of the Database Structure

The diversity of data presented in separate yet related tables can be incorporated as one of the strongest advantages of the design. The records on attendance, information of employees, details of devices, and payroll are all separated but interconnected by meaningful relations. This segregation enhances reliability because any change made in one context like employee information or salary so that it affects the attendance records adversely, will be avoided. Consequently, past information is safe and reliable with the course of time.

Scalability and Flexibility of the Design

The database will be highly scalable; this is a factor that is significant to the growing organizations such as NetSolace Inc. The introduction of new workshops or biometric devices is easy because it does not need to change the current database organization. Likewise, it is possible to extend the logic of calculating salaries, such as, but not limited to, adding bonuses or deductions, without impacting attendance information. This flexibility makes sure that the system may be modified easily as the future operational needs arise without necessarily redesigning it.

Support for Future Technologies and Integration

The other major reason as to why the design should be designed with a view of being integrated in the future is the fact that the design is ready to undergo the process. The format of the separation of the attendance data renders it appropriate to accept real-time data input by the IoT devices through APIs. Since the attendance data are already separate and logically connected, the automated input can be introduced and have little affect on other database fields. All in all, the design is a powerful and solid base towards future evolution and is useful in supporting real-time IoT drive and system development over the years.

Database Development and Implementation

Introduction to Database Development

Once the planning and design phase took place, the second step involved several steps to the conversion of the logical database design into the working physical database system. This stage was used to polish on the application of the database using the right type of software programs and also to make sure that the system was running correctly on a working environment. The objective was to have a stable background database server and integrate it to a user friendly interface where data gets to be stored, accessed and managed effectively.

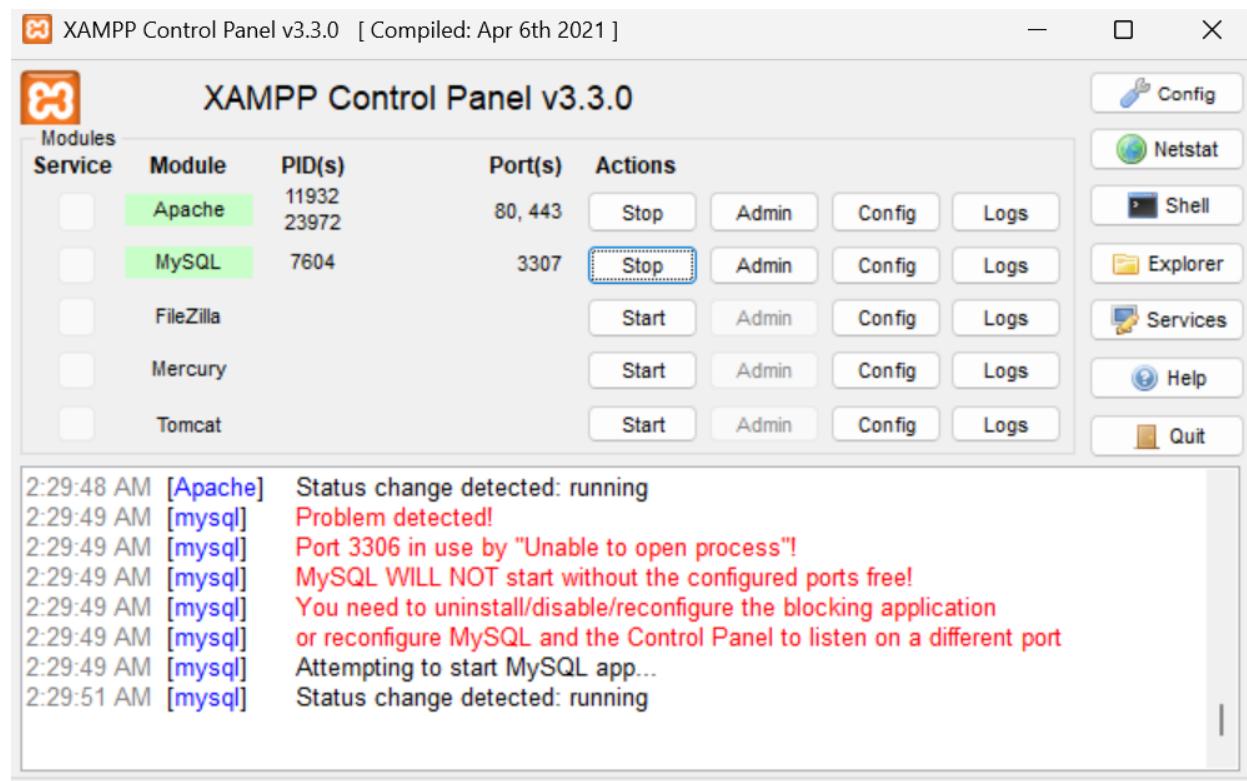
In this project, MySQL, phpMyAdmin, XAMPP and ODBC coupled with Microsoft Access were used. Such an arrangement is indicative of a realistic enterprise style; a potent server based data storage system, and a distinct front end application system to deal with user interaction.

The database development phase involved converting the logical database design into a working physical database. This stage focused on implementing tables, constraints, and relationships using MySQL, and linking the database to a Microsoft Access front-end to allow easy data entry and reporting. The aim was to ensure the system works correctly, securely, and meets user requirements identified.

Setting Up the Database Server Using XAMPP

The initial phase in the process of system implementation was to install and configure XAMPP. XAMPP has been chosen on the basis of offering a simple local server test environment that consists of MySQL which is appropriate in developing and testing database programs.

Once the XAMPP is installed, I started the XAMPP Control Panel wherein I started the MySQL service. This was critical since no database and table can be created without the database server being operational. I ensured that MySQL was functioning correctly using the control panel to check the status of the service. This guaranteed that this service was online so as to have a stable service of storing attendance data.



Creating the Database Using phpMyAdmin

After the MySQL server was started, I was able to use phpMyAdmin in one of the web browsers by accessing the following web address: <http://localhost/phpmyadmin>. phpMyAdmin is a

graphical interface application to manage MySQL databases and is thus easier in creating databases, defining tables, and running SQL queries.

I also created a new database, which was called netsolace_attendance, within phpMyAdmin. This database was purely devoted to storing data that were produced by attendance management system. Making specific database makes it organized and does not clash with other applications.

The screenshot shows the phpMyAdmin interface for the 'netsolace_attendance' database. The left sidebar lists various databases, with 'netsolace_attendance' selected. The main area displays the table structure with the following data:

Table	Action	Rows	Type	Collation	Size	Overhead
attendance_records	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
departments	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
devices	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
employees	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
salary_slips	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
5 tables	Sum	20	InnoDB	utf8mb4_unicode_ci	144.0 Kib	0 B

At the bottom, there are links for 'Create new table' and 'Console'.

Implementing Database Tables in MySQL

I engaged in the implementation of the four main tables that had been identified in the design phase as Employees, Attendance Records, Devices, and Salary Slips. All the tables have been designed with well chosen fields, primary key, and types of data to represent real life needs.

Employees table was created initially since it would be used in the center of reference in the entire system. The main key was an auto-increasing number that uniquely identified the individual employees. The names and designations were entered in the text fields and hire dates were entered as a date field.

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 'netolace_attendance' database.

Table Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	employee_id	int(11)	utf8mb4_unicode_ci		No	None	AUTO_INCREMENT		Change Drop More
2	first_name	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
3	last_name	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
4	department	varchar(50)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
5	designation	varchar(50)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
6	hire_date	date			Yes	NULL			Change Drop More
7	status	enum('active', 'inactive')	utf8mb4_unicode_ci		Yes	active			Change Drop More
8	created_at	timestamp			No	current_timestamp()			Change Drop More

Datasheet View:

employee_id	first_name	last_name	department	designation	hire_date	status	created_at
1	Ali	Khan	IT	Developer	10/12/2025	active	12/10/2025
2	Sara	Ahmed	IT	Software Engin	10/12/2025	active	12/10/2025
3	Faiz	Ali	IT	Network Admir	10/12/2025	active	12/10/2025
4	Ahmed	Ali	IT	System Analyst	10/12/2025	active	12/10/2025
5	Muhammad	Aun	IT	Database Assis	10/12/2026	active	12/10/2025
6	Hussnain	Ahmed	IT	Programmer	10/12/2025	active	12/10/2025

My second table was the Attendance Records table, and it holds the attendance data provided by biometric devices. The field that this table will have captures the attendance date, time, method of authentication, and in/out status. Attendance records were joined with the employee and device identifiers to establish relations between them.

The screenshot shows the phpMyAdmin interface for the 'attendance_records' table in the 'netsolace_attendance' database. The table has 8 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	attendance_id	int(11)			No	None		AUTO_INCREMENT	Change Drop
2	employee_id	int(11)			No	None		Change Drop	
3	device_id	int(11)			No	None		Change Drop	
4	attendance_date	date			No	None		Change Drop	
5	attendance_time	time			No	None		Change Drop	
6	auth_method	enum('FINGERPRINT', 'RFID', 'PIN')	utf8mb4_unicode_ci		No	None		Change Drop	
7	in_out_status	enum('IN', 'OUT')	utf8mb4_unicode_ci		No	None		Change Drop	
8	created_at	timestamp			No	current_timestamp()		Change Drop	

Below the table structure, there are buttons for 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', etc. There is also a search bar for 'Add' and a 'Go' button.

The screenshot shows the data in the 'attendance_records' table. The table contains 6 rows of data:

attendance_id	employee_id	device_id	attendance_date	auth_method	in_out_status	created_at
1	1	1	12/1/2025 ##### FINGERPRINT	IN	12/1/2025	
2	2	1	12/1/2025 ##### FINGERPRINT	IN	12/1/2025	
3	3	1	12/1/2025 ##### RFID	IN	12/1/2025	
4	4	1	12/1/2025 ##### FINGERPRINT	IN	12/1/2025	
5	5	1	12/1/2025 ##### RFID	IN	12/1/2025	
6	6	1	12/1/2025 ##### FINGERPRINT	IN	12/1/2025	

The left sidebar shows the database structure with tables like attendance, devices, employees, and salary_slips.

The table Devices was then formed where the information about every attendance unit is stored. It is possible to define this table individually and then provide productive monitoring of the use of devices and easily administer the device without any impact on attendance indicators.

The screenshot shows the phpMyAdmin interface for the 'netolace_attendance' database. The left sidebar lists databases and tables, with 'devices' selected. The main area displays the 'Table structure' for the 'devices' table, showing columns: device_id, device_code, location, is_active, and created_at. It also shows the primary key index for device_id and a unique index for device_code.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	device_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	device_code	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
3	location	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
4	is_active	tinyint(1)			Yes	1			Change Drop More
5	created_at	timestamp			No	current_timestamp()			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	device_id	2	A	No	
Edit Rename Drop		BTREE	Yes	No	device_code	2	A	No	

Datasheet View

device_id	device_code	location	is_active	created_at
1	NS-UNIT-01	Head Office - IV	1 #####	
2	NS-UNIT-02	Factory - Entrar	1 #####	

Lastly, the table on Salary Slips was introduced, where the payroll-related information was to be stored. This table is the one that connects salary data with the employees, and stores calculated data that include total number of hours worked, overtime as well as the net salary. During the separation of the salary data and the attendance data, it will be more understandable and secure.

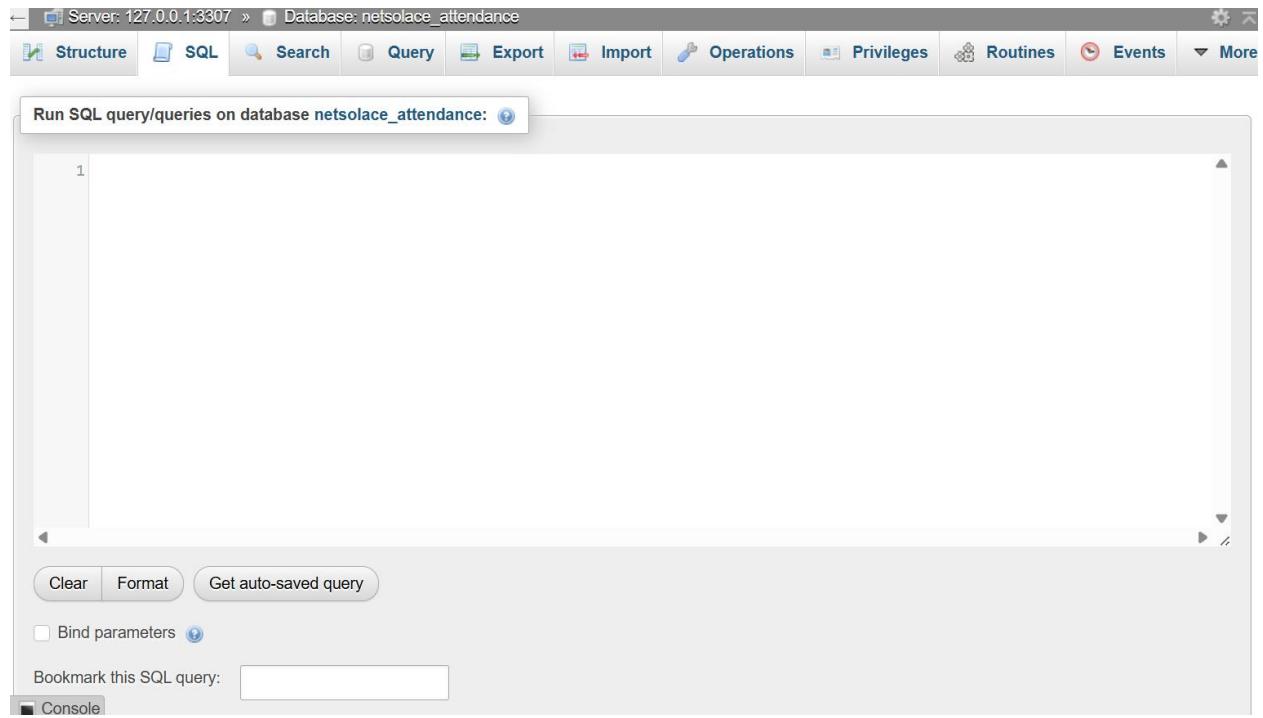
The screenshot shows the phpMyAdmin interface for the 'netsolace_attendance' database. The top section displays the 'salary_slips' table structure with 11 columns: salary_id, employee_id, period_start, period_end, total_hours, basic_salary, overtime_hours, overtime_rate, gross_salary, net_salary, and generated_at. The bottom section shows a data entry grid for the 'salary_slips' table, displaying six rows of sample data.

salary_id	employee_id	period_start	period_end	total_hours	basic_salary	overtime_hours	overtime_rate	gross_salary	net_salary	generated_at
1	1	12/1/2025	12/31/2025	176	50000	10	500	55000	53000	12/31/2025
2	2	12/1/2025	12/31/2025	176	45000	8	450	48600	47000	12/31/2025
3	3	12/1/2025	12/31/2025	176	42000	6	400	44400	43000	12/31/2025
4	4	12/1/2025	12/31/2025	176	48000	12	450	53400	51000	12/31/2025
5	5	12/1/2025	12/31/2025	176	40000	5	350	41750	40500	12/31/2025
6	6	12/1/2025	12/31/2025	176	46000	7	400	48800	47000	12/31/2025

Inserting Sample Data Using SQL

To verify how the database operated, I entered some sample data in each of the tables using the phpMyAdmin SQL INSERT commands. This encompassed the addition of a few employees, enrollment of test devices, development of the attendance records under various authentication possibilities, as well as generation of pay slip information in case of testing queries and reports.

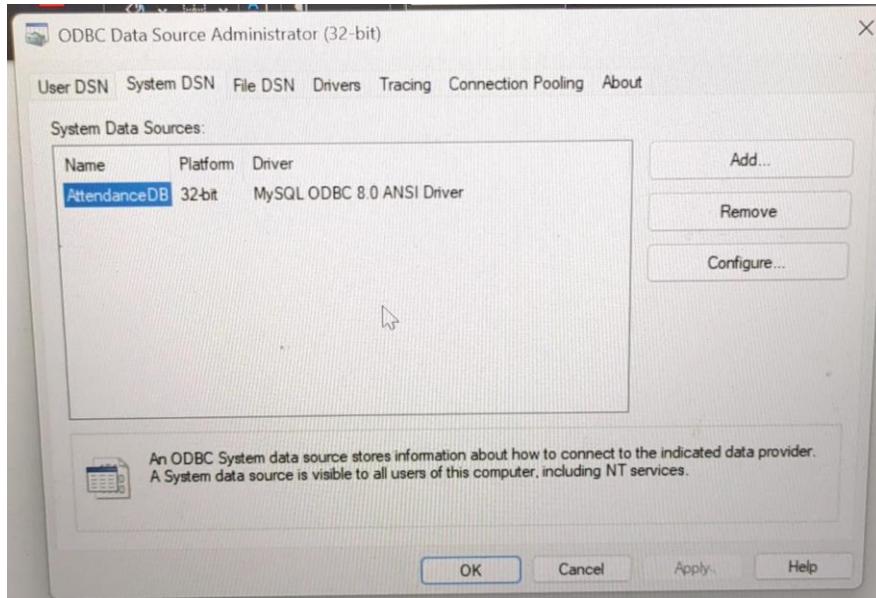
The SQL statements that were used provided the assurance that the data could be inserted into the database accurately and repeatably, and were also an indication that it provided direct interaction with the database via query language.



Establishing ODBC Connectivity Between MySQL and MS Access

To offer the user-friendly front-end, I linked the MySQL database with Microsoft Access with the help of ODBC (Open Database Connectivity). This operation enables Access to relate with MySQL tables as connected tables with the information being safely stored in the MySQL server.

The first step I made the ODBC Data Source Administrator open and created a new Data Source Name (DSN). I put the server as the localhost, an authentication credential and chose the database containing the netsolace attendance, netsolace attendance. To ensure that Access could optimally interact with MySQL, I tested the connection to ensure that this was the case.



SQL File Submission

The complete database structure and sample data were exported from phpMyAdmin as a .sql file. This file contains all table definitions, relationships, and insert statements and is submitted as part of the assignment evidence.



.sql.txt

Linking MySQL Tables in Microsoft Access

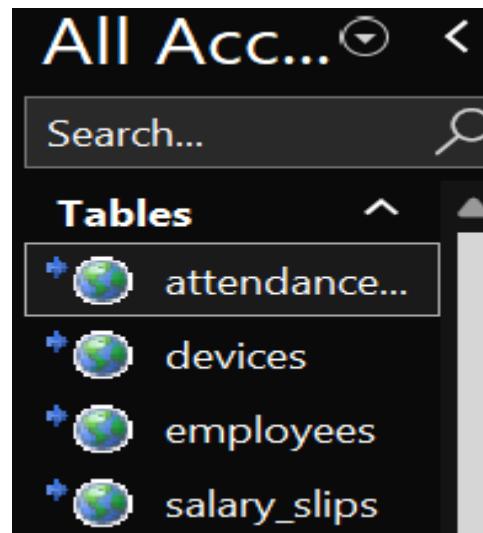
When the ODBC connection was configured, I opened Microsoft Access and formed a new empty database file to serve as a front-end application. I used the External Data tab and selected the choice of connecting tables to an ODBC database and selecting the already set up DSN.

I connected all the four tables along with the Employees, Attendance Records, Devices and Salary Slips. After the connection, the tables were listed on the access Navigation Pane with a special icon, which shows they were linked to another database.

Linking MySQL Tables in Microsoft Access

Item	Description
------	-------------

Front-end Tool	Microsoft Access
Backend Database	MySQL
Connection Type	ODBC
Tables Linked	Employees, Devices, AttendanceRecords, SalarySlips



This can be used to store it in the MySQL database so that any modifications in Access are reflected in the backend and vice versa.

Front End Validation in Microsoft Access

Validation rules were applied in Microsoft Access to prevent incorrect or incomplete data entry. Required fields were enforced at table and form level, and custom validation error messages were displayed when users entered invalid data. For example, employee names cannot be left blank, attendance dates are mandatory, and authentication methods are restricted to predefined values. These validations improve data accuracy and reduce user input errors before data is stored in the MySQL database.

Test ID	
T01	Employee form – valid save
T02	Employee form – validation error
T03	Attendance form – valid entry

T04	Attendance form – invalid employee
T05	Device usage query result
T06	Salary calculation result
T07	Boundary case (0 overtime)
T08	Access edit reflected in MySQL

Data Validations

Attendance Validation:

The screenshot shows the Microsoft Access Table Design view for the 'attendance_records' table. The main pane displays the table structure with fields: attendance_id, employee_id, device_id, attendance_date, attendance_time, auth_method, in_out_status, and created_at. The 'Data Type' column shows the respective types for each field. The 'Field Properties' pane is open, showing detailed settings for the 'attendance_id' field, including its data type as Long Integer, validation rule as 'Is Not Null', and required status as Yes. The 'Property Sheet' pane on the right shows general properties for the table, such as 'Read Only When Disconnected' (unchecked), 'Subdatasheet Expanded' (No), and 'Orientation' (Left). Other tabs in the ribbon include Home, Create, External Data, Database Tools, Help, and Table Design.

Devices Validation:

Properties for device_id:

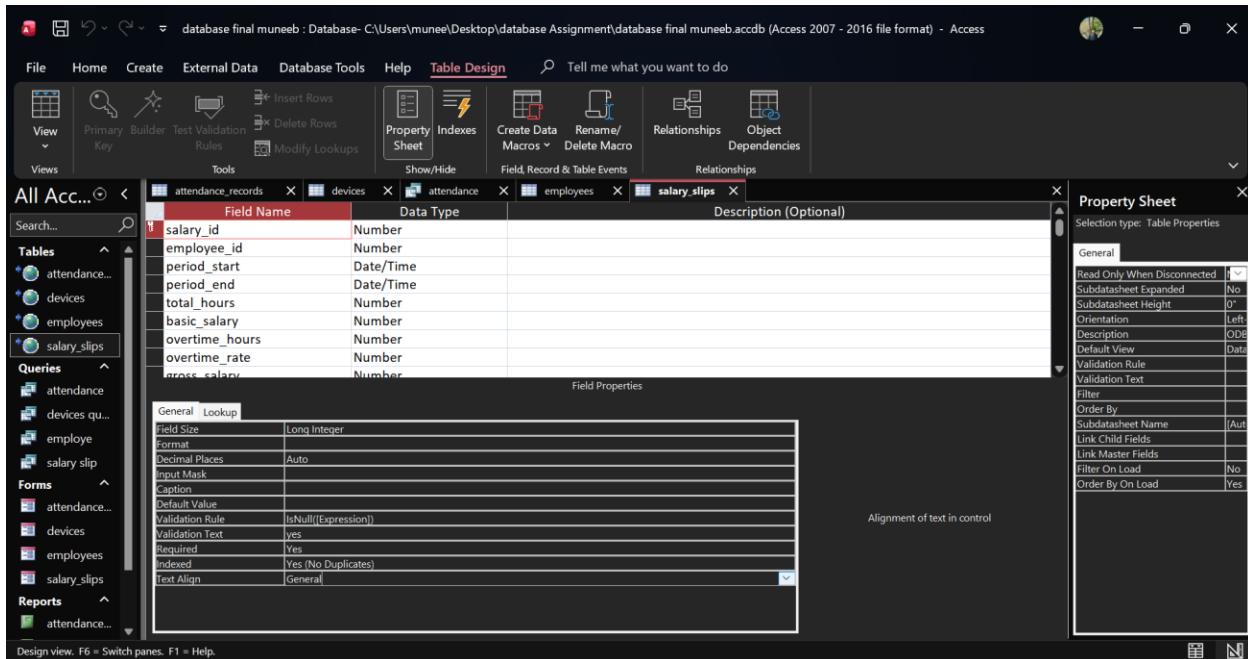
- Field Properties:**
 - Validation Rule: IsDate(Expression)
 - Validation Text: yes
 - Required: Yes
 - Indexed: Yes (No Duplicates)
- Property Sheet (General):**
 - Read Only When Disconnected: Yes
 - Subdatasheet Expanded: No
 - Subdatasheet Height: 0"
 - Orientation: Left
 - Description: ODB
 - Default View: Data
 - Validation Rule: Validation Text
 - Filter: Order By
 - Order By: Subdatasheet Name (Auto)
 - Link Child Fields: Link Master Fields
 - Filter On Load: No
 - Order By On Load: Yes

Employees Validation:

Properties for employee_id:

- Field Properties:**
 - Validation Rule: Is Not Null
 - Validation Text: Yes
 - Required: Yes
 - Indexed: Yes (No Duplicates)
- Property Sheet (General):**
 - Read Only When Disconnected: Yes
 - Subdatasheet Expanded: No
 - Subdatasheet Height: 0"
 - Orientation: Left
 - Description: ODE
 - Default View: Data
 - Validation Rule: Validation Text
 - Filter: Order By
 - Order By: Subdatasheet Name (Auto)
 - Link Child Fields: Link Master Fields
 - Filter On Load: No
 - Order By On Load: Yes

Salary Slips Validation:



Evaluation of Development Tools and Approach

Effectiveness of MySQL as the Backend Database

The choice of MySQL as the back-end database was a good choice of this project. MySQL is very reliable, highly performative, and supports the relational schema of the database. It can process large amounts of data, and is suited to the attendance management system since records can increase dramatically with time; which allows it to perform complex SQL queries. MySQL is also common in industry and this implies that the system is designed on well-supported technology that is future proof.

Role of phpMyAdmin in Database Development

The phpMyAdmin was a significant factor to be taken into account on the development stage by making it easy to create and handle databases. Its graphical interface enabled making tables, updating and filling them in a short time, without the need to use completely command-line input. The immediate and direct access to SQL queries and immediate results also facilitated testing and debugging a great deal. All in all, phpMyAdmin worked wonders towards the quick realization of the database framework and successful testing.

Use of Microsoft Access as a Front-End Interface

Since it is a simple tool that can be used by non-technical users, Microsoft Access was chosen as the front-end tool. Access was suitable due to the ease with which queries, forms and reports could be created without needed extensive coding, and thus it suited a prototype system. In this project, Access was the means that would help to show how HR employees and administrators

could deal with the database through the use of familiar tools. The report and its shape assisted in displaying the data clearly and structured that enhanced usability.

ODBC-Based Client Server Architecture

ODBC to support the interface between the Microsoft Access and the MySQL database allowed a flexible client-server architecture. This method was essential in order to make sure that data did not go out of reach since they were stored centrally in MySQL but could still be accessed in Access. It also showed the interaction of various systems without redundancy. ODBC connection was very reliable during the development and testing phase and this has confirmed that such setup is suitable in small and medium scale systems.

Limitations and Future Development Considerations

Despite the efficiency of the selected tools to create a prototype, it has certain restrictions to this kind of approach. The use of Microsoft Access is not suited in the environment with a large number of users who want to access it simultaneously or in a remote fashion. To enhance scalability and accessibility, the use of Access in a future implementation would be done away with, changing it to a web-based front. Regardless of these shortcomings, the general development strategy is effective in satisfying the requirements of the system and offers a solid base in the future which can be expanded and advanced.

Security and Maintenance Considerations

Security Measures Implemented

- Restricted MySQL user access
- HR users allowed data entry via Access only
- Salary data separated into its own table

Maintenance Strategy

- Weekly database backups using phpMyAdmin export
- Indexes on EmployeeID and Date fields
- Old attendance records can be archived if required

The screenshot shows the phpMyAdmin interface for the 'netsolace_attendance' database. The left sidebar shows various databases like 'information_schema', 'mysql', and 'netsolace_attendance'. The main panel is titled 'Users having access to "netsolace_attendance"' and displays a table with three rows:

User name	Host name	Type	Privileges	Grant	Action
root	127.0.0.1	global	ALL PRIVILEGES	Yes	Edit privileges Export
root	::1	global	ALL PRIVILEGES	Yes	Edit privileges Export
root	localhost	global	ALL PRIVILEGES	Yes	Edit privileges Export

Below the table, there are buttons for 'Check all', 'With selected:', and 'Export'. A 'New' button is at the bottom left, and a 'Console' link is at the bottom right.

The Microsoft Access front-end file (AttendanceSystem.accdb) is submitted as part of the assignment and contains linked tables, forms, queries, and reports.

Access Front-End File Submission

The Microsoft Access front-end file (AttendanceSystem.accdb) is submitted as part of the assignment and contains linked tables, forms, queries, and reports.



database final muneeb.accdb

Introduction to Queries and Testing

After the successful implementation of the database system and connection to the Microsoft Access, the second step was to guarantee that the information stored in the database system could be transformed into useful information and the system worked well under various conditions. Learning Outcome 3 is concerned with querying SQLs and testing of the system to ensure it is performing as per the user and system requirements.

Queries are also an important aspect of any database system as they will convert the raw data that was stored into the meaningful outputs. Within the framework of the NetSolace Inc. attendance system, there were queries needed to access the history of attendance, analyzing device usage, as well as to produce the information related to salaries. Testing was also

essential, since it was also done to ensure that queries returned the right answers and that the system was able to process both valid and invalid data accordingly.

Development of SQL Queries

Queries are also an important aspect of any database system as they will convert the raw data that was stored into the meaningful outputs. Within the framework of the NetSolace Inc. attendance system, there were queries needed to access the history of attendance, analyzing device usage, as well as to produce the information related to salaries. Testing was also essential, since it was also done to ensure that queries returned the right answers and that the system was able to process both valid and invalid data accordingly.

The screenshot shows the Microsoft Access interface with the following details:

- Left pane (Tables/Queries):** Shows tables like attendance, devices, employees, and salary_slips, and queries like attendance.
- Middle pane (Query Results):** Displays the results of the query: "SELECT attendance_id, employee_id, device_id, attendance_date, attendance_time, auth_method, in_out_status FROM attendance_records;". The results show 6 rows of data.
- Right pane (Property Sheet):** Shows properties for the current query, including General settings like Description, Default View, Record Locks, ODBC Timeout, Filter, Order By, Max Records, Orientation, Subdatasheet Name, Link Child Fields, Link Master Fields, Subdatasheet Height, Subdatasheet Expanded, Filter On Load, and Order By On Load.
- Status Bar:** Shows "Showing rows 0 - 5 (6 total, Query took 0.0010 seconds.)".
- Bottom Buttons:** Includes "Check all", "With selected:", "Edit", "Copy", "Delete", and "Export".

attendance_id	employee_id	device_id	attendance_date	attendance_time	auth_method	in_out_status
1	1	1	2025-12-01	09:00:00	FINGERPRINT	IN
2	2	2	2025-12-01	09:05:00	FINGERPRINT	IN
3	3	3	2025-12-01	09:02:00	RFID	IN
4	4	4	2025-12-01	09:10:00	FINGERPRINT	IN
5	5	5	2025-12-01	09:15:00	RFID	IN
6	6	6	2025-12-01	09:20:00	FINGERPRINT	IN

The next query that was important was the Device Usage Query. This query will join the Devices and the Attendance Records tables and will be using a count function to determine the number of records related to the attendance of the devices. The rationale behind this query is to assist

the management track the intensity at which every device of biometric is in use and those that might need maintenance or relocation.

The screenshot shows a dual-pane interface. The top pane displays a 'Property Sheet' for a selected query, listing various properties like General, Description, Default View, Record Locks, ODBC Timeout, Filter, Order By, Max Records, Orientation, Subdatasheet Name, Link Child Fields, Link Master Fields, Subdatasheet Height, Subdatasheet Expanded, Filter On Load, and Order By On Load. The bottom pane is a 'Table: devices' view from the 'phpMyAdmin' interface. It shows a single row of data with columns: device_code (NS-UNIT-01), location (Head Office - Main Gate), and Total_Usage (6). The SQL query executed is:

```
SELECT devices.device_code, devices.location, COUNT(attendance_records.attendance_id) AS Total_Usage
FROM devices INNER JOIN attendance_records ON devices.device_id = attendance_records.device_id
GROUP BY devices.device_code, devices.location;
```

The left sidebar of the phpMyAdmin interface shows the database structure with schemas like information_schema, mysql, netsolace_attendance, performance_schema, phpmyadmin, and test, along with their respective tables (e.g., attendance_records, devices, employees, salary_slips).

SalarySummary Query has also been generated which is used to combine Employee data and salary data. This query will give out names of employees, pay period, number of hours worked and salary amount. Values calculated like overtime pay or gross salary were measured in some instances using the calculated fields. This question is the basis of payroll reporting and it manifests the presentation of payroll information in an accurate and transparent way.

The screenshot displays two database interfaces side-by-side. On the left, Microsoft Access is shown with a query window containing the following SQL code:

```

SELECT employees.employee_id, employees.first_name, employees.last_name, salary_slips.period_start, salary_slips.period_end,
       salary_slips.basic_salary, (salary_slips.overtime_hours * salary_slips.overtime_rate) AS Overtime_Pay, salary_slips.gross_salary,
       salary_slips.net_salary
FROM employees INNER JOIN salary_slips ON employees.employee_id = salary_slips.employee_id;

```

The results of this query are displayed in a table with the following data:

employee_id	first_name	last_name	period_start	period_end	basic_salary	Overtime_Pay	gross_salary	net_salary
1	Ali	Khan	2025-12-01	2025-12-31	50000.00	5000.0000	55000.00	53000.00
2	Sara	Ahmed	2025-12-01	2025-12-31	45000.00	3600.0000	48600.00	47000.00
3	Faiz	Ali	2025-12-01	2025-12-31	42000.00	2400.0000	44400.00	43000.00
4	Ahmed	Ali	2025-12-01	2025-12-31	48000.00	5400.0000	53400.00	51000.00
5	Muhammad	Aun	2025-12-01	2025-12-31	40000.00	1750.0000	41750.00	40500.00
6	Hussnain	Ahmed	2025-12-01	2025-12-31	46000.00	2800.0000	48800.00	47000.00

On the right, the phpMyAdmin interface shows the same data in a grid format. The SQL query is identical to the one in Access. The results table is as follows:

employee_id	first_name	last_name	period_start	period_end	basic_salary	Overtime_Pay	gross_salary	net_salary
1	Ali	Khan	2025-12-01	2025-12-31	50000.00	5000.0000	55000.00	53000.00
2	Sara	Ahmed	2025-12-01	2025-12-31	45000.00	3600.0000	48600.00	47000.00
3	Faiz	Ali	2025-12-01	2025-12-31	42000.00	2400.0000	44400.00	43000.00
4	Ahmed	Ali	2025-12-01	2025-12-31	48000.00	5400.0000	53400.00	51000.00
5	Muhammad	Aun	2025-12-01	2025-12-31	40000.00	1750.0000	41750.00	40500.00
6	Hussnain	Ahmed	2025-12-01	2025-12-31	46000.00	2800.0000	48800.00	47000.00

These queries along with others show the good utilization of SQL joins, filtering and calculations to address organizational information requirements.

Test Environment

Item	Description
Operating System	Windows
Database Server	MySQL (XAMPP)
Database Tool	phpMyAdmin
Front-End Tool	Microsoft Access
Connection Type	ODBC

Test Data Type	Sample employee, attendance, and salary data
-----------------------	--

This shows the marker **where and how testing was done**.

Use of Queries to Meet User Requirements

The queries that are developed are very direct in the manner they respond to the user needs that were identified in the project. The attendance queries assist the HR staff in providing employee attendance data in a readable format. The queries on the usage of the devices can provide insight to the management users as they can see the performance of the attendance devices in practice. Salary summary queries are used to assist in preparing payrolls as the queries portray consolidated salary information based on the attendance data.

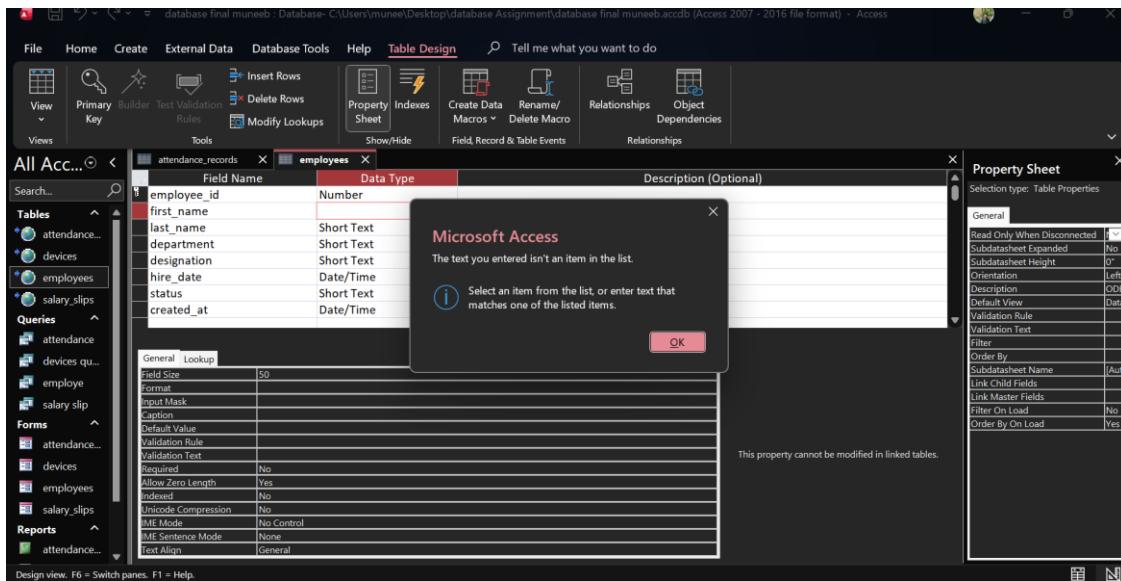
The system is reduced to queries rather than accessing the table; this reduces the chances of data being altered accidentally. Queries are an additional barrier between the data stored in the system and users guaranteeing the safe and effective access to information. This will enhance usability and reliability of the data hence this system can be used in real world organization.

Testing Strategy and Test Planning

Since it is important that the system works properly, a well-designed testing plan was elaborated on. Test was done to correlate data entry, querying, rules of data validation and data integration between MySQL and Microsoft Access. Various types of test data were concerned to test the system under realistic and most difficult conditions.

Testing was classified into three broad categories namely normal testing, abnormal testing and boundary testing. Normal testing entailed input of real information that portrays normal system use. Abnormal testing was introduced with a purpose to test the validity or the incompleteness of data with an aim of seeing the reaction of the system. Boundary testing investigated the methods used by the system to deal with extreme or edge cases.

Test ID	Expected Result	Actual Result	Pass/Fail
T02	Error message shown	Validation message displayed	Pass



Test Plan Table

The table below summarizes the main test cases used during system testing:

Test ID	Test Area	Test Description	Input Used	Expected Outcome
T01	Employees	Add employee with valid data	Valid name, designation, hire date	Record saved successfully
T02	Employees	Save employee with missing name	Blank first name	Record rejected
T03	Attendance	Add valid attendance record	Valid employee and device IDs	Attendance saved correctly
T04	Attendance	Attendance without valid employee	Invalid employee ID	Entry rejected
T05	Devices	Run device usage query	Existing attendance data	Correct usage count
T06	Salary	Generate salary slip with normales hours	Valid salary data	Correct net salary shown
T07	Salary	Boundary test with zero overtime	Overtime hours = 0	Salary handled correctly

T08	Integration	Update record through Access	Edit employee name	Change reflects in MySQL
-----	-------------	------------------------------	--------------------	--------------------------

The screenshot shows the phpMyAdmin interface. The left sidebar lists databases: New, information_schema, mysql, netsolace_attendance (selected), New, attendance_records, devices, employees, salary_slips, performance_schema, phpmyadmin, and test. The top navigation bar includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, and More. The main area is titled "Run SQL query/queries on database netsolace_attendance:" and contains the following SQL code:

```

1 SELECT employees.employee_id, employees.first_name, employees.last_name, salary_slips.period_start,
salary_slips.period_end, salary_slips.basic_salary, (salary_slips.overtime_hours * salary_slips.overtime_rate) AS
Overtime_Pay, salary_slips.gross_salary, salary_slips.net_salary
2 FROM employees INNER JOIN salary_slips ON employees.employee_id = salary_slips.employee_id;
3

```

Below the code are buttons for Clear, Format, Get auto-saved query, Bind parameters, Bookmark this SQL query:, and Console.

Evidence of testing was collected in the form of screenshots showing successful data entry, validation error messages, and query results. These screenshots demonstrate that the system behaves correctly under both valid and invalid input conditions.

Examples of evidence include:

- Access validation error when a required field is empty
- Successful attendance record insertion
- Query results showing device usage summary

Test Data Explanation

Normal testing data had been made to be similar to the real operating data. This was comprised of normal hours of work, real names of employees, and normal authentication like fingerprint and RFID. Realistic data also made sure that the system would not experience any problem in its operation under normal usage.

The abnormal test data encompassed the lack of names in the list of employees, wrong reference between tables and incomplete attendance record. Such tests were used to ensure that validation rules and relational links did not allow wrong data to be stored to the database. The data of boundary tests concentrated on the edge cases like zero overtime hours or

minimum attendance entries, as the calculations and queries would not break down under severe circumstances.

These testing methods were combined to provide a wide coverage of the behavior of the system.

Test data was carefully chosen to reflect real system usage. Valid data such as normal working hours and correct authentication methods were used to confirm correct system behaviour. Invalid data such as missing employee names, invalid authentication methods, and non-existing employee IDs were used to test validation rules and referential integrity. Boundary data, such as zero overtime hours, was included to ensure salary calculations handled edge cases correctly. This approach confirms the robustness and reliability of the system.

Evaluation of Test Results and Effectiveness

The outcome of the testing process indicated that the system is reliable in all scenarios that are tested in the testing process. Valid data was properly stored and invalid data was rejected or signalled and queries were able to give the desired results. The system logic was also found to be correct by manually scrutinizing the results of each query, mostly the number of employees who are present and the count of their salaries.

It was also tested that the ODBC connection between the MySQL and the Microsoft Access was stable. Any alterations done in Access were instantly echoed in MySQL database this proved the syncing between the frontend and backend. This is especially relevant to real-world implementation where it is necessary to have uniformity within systems.

A weakness that was described during the testing was that the prototype had manual inputs of attendance data. Although this is achievable in the development and testing settings, in the real life systems, automated data capture by the biometric devices would come in handy.

The testing process confirmed that the database system meets the functional and non-functional requirements. Validation rules successfully prevented incorrect data entry, relationships maintained data integrity, and queries returned accurate results. The ODBC connection between Microsoft Access and MySQL functioned correctly, ensuring data consistency between front-end and back-end systems.

Reflection on Improvements Identified Through Testing

Testing assisted to identify the areas that could be improved. As an example, more validation rules might be added to limit wrong input further and more industrial error messages might further enhance good user feedback. Also, in the future, automated testing scripts may be presented to simplify the manual testing process and enhance that of consistency.

In general, the testing plan was successful and was able to guarantee that the system was satisfied with the functional and user needs.

The testing process confirmed that the database system meets the functional and non-functional requirements. Validation rules successfully prevented incorrect data entry, relationships maintained data integrity, and queries returned accurate results. The ODBC connection between Microsoft Access and MySQL functioned correctly, ensuring data consistency between front-end and back-end systems.

Technical Documentation, User Documentation, and System Evaluation

Purpose of Documentation

Documentation is a very essential aspect of a database system since it makes the system understandable, maintainable and in future. The best planned database may be hard to maintain without a proper documentation. Due to this fact, Learning Outcome 4 is aimed at creating technical documentation to be used by the administrators, developers, and creating user manual to be used by non-technical users like the HR personnel and management.

The documentation was made with the clarity and practicality in mind making sure that it can be used even without the presence of the original developer in the event future users or developers wish to use the system.

The purpose of documentation is to ensure that the database system can be understood, used, and maintained by different users. Technical documentation is intended for system administrators and developers, while user documentation is designed for HR staff and management. Proper documentation also ensures that the system can be maintained or extended in the future.

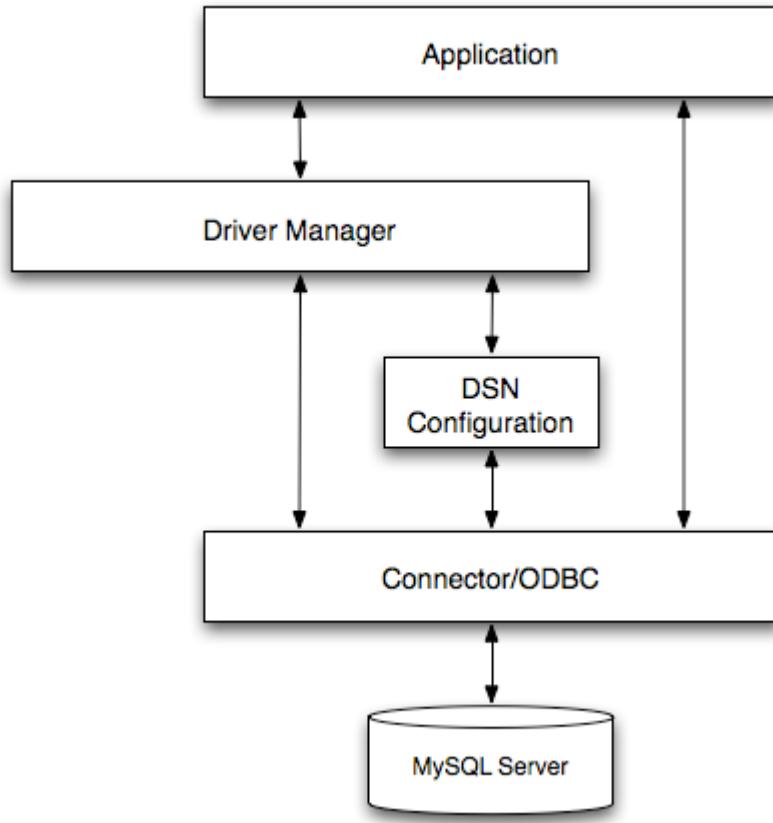
Technical Documentation System Overview

The attendance system client server architecture has been constructed on a client server principle. The back-end database is writable through MySQL that has been deployed locally with the help of XAMPP, and the front-end interface is the Microsoft Access. The Access is connected to the MySQL database through ODBC, which enables the storage of data in a central location and can additionally produce a desktop interface that is also user-friendly.

This system has all the core data in the form of employees, attendance records, devices, and salary slips stored in MySQL. Microsoft access does not store data locally, however, it

communicates with MySQL by means of linked tables. Such a separation provides greater control of data, greater scalability, and less probability of a data inconsistency.

The Attendance Management System uses a client–server architecture. MySQL is used as the back-end database server and is hosted locally using XAMPP. Microsoft Access is used as the front-end application and is connected to MySQL through ODBC. All data is stored centrally in MySQL, while Access is used for data entry, queries, and reports.



Technical Documentation Database Structure and Maintenance

The database will include four tables which are interrelated to each other and are as follows: Employees, Attendance Records, Devices, and Salary Slips. Every table has a well determined role and primary keys are used to identify records uniquely and logical relationships are made to relate data in different tables. This organization assures the safety of the information and contributes to the effective interrogation.

Regarding maintenance, the administrators are recommended to back up MySQL database on a regular basis through the export option provided in phpMyAdmin. Backups are to be made prior to any structural changes, which can include the addition of new fields and tables. Also, the periodic table data review and validation conditions contribute to the long term consistency.

The administrators should also be sure that the MySQL server is secure, by regulating the access of database users and securing credentials. In case of a system deployment to the real organizational setting, there should be additional security measures in terms of encrypted connection and limited user rights.

Employees Table

Field Name	Data Type	Description
EmployeeID	INT (PK)	Unique employee identifier
FirstName	VARCHAR	Employee first name
LastName	VARCHAR	Employee last name
Designation	VARCHAR	Job title
HireDate	DATE	Date of joining
EmploymentStatus	VARCHAR	Active or Inactive

Devices Table

Field Name	Data Type	Description
DeviceID	INT (PK)	Unique device ID
DeviceCode	VARCHAR	Device reference code
Location	VARCHAR	Device location

Status	VARCHAR	Active or Inactive
--------	---------	--------------------

AttendanceRecords Table

Field Name	Data Type	Description
AttendanceID	INT (PK)	Unique attendance record
EmployeeID	INT (FK)	Linked employee
DeviceID	INT (FK)	Linked device
AttendanceDate	DATE	Attendance date
AttendanceTime	TIME	Attendance time
AuthMethod	VARCHAR	Fingerprint / RFID / PIN
InOut	VARCHAR	IN or OUT

SalarySlips Table

Field Name	Data Type	Description
SalaryID	INT (PK)	Salary slip ID
EmployeeID	INT (FK)	Linked employee
PayPeriodStart	DATE	Start of pay period
PayPeriodEnd	DATE	End of pay period
TotalHoursWorked	DECIMAL	Total hours worked
NetSalary	DECIMAL	Final salary

Technical Documentation Security Considerations

The main issue to security in cases where employee attendance and salary information are stored is security. The present prototype has authentication credentials to access the MySQL database where only authorized persons have access to modify data or database structure.

Although this type of security is adequate when working on a prototype, role-based access control may be incorporated in future implementation. As an example, a report may be provided to the HR staff with read-only access, and the administrators should have all the

control over the tables and queries. This would go a long way in averting the possibilities of making accidental or unauthorized alterations on sensitive data.

Technical Documentation Installation Guide

Installation Steps

1. Install **XAMPP**
2. Start **MySQL** from XAMPP Control Panel
3. Open **phpMyAdmin**
4. Import the provided .sql file
5. Create ODBC DSN for MySQL
6. Open Microsoft Access
7. Link MySQL tables using ODBC

This setup allows Access to act as a front-end while MySQL stores all data.

User Documentation System Usage Overview

On the side of the user, the system will be easy and efficient. The majority of communication with the system is in a report and query format as opposed to going directly to the table. This design option minimises user errors and provides the user to concentrate on the interpretation of information instead of dealing with raw data.

Upon opening the Microsoft Access front-end, one can go to the Reports section of the interface and get ready-made reports of the employees, attendance status, the usage of the device as well as of the salaries. These reports will reflect the information in a clear and understandable format that is applicable in the routine administration activities.

Backup

- Open phpMyAdmin
- Select the database
- Click **Export**
- Choose **SQL**
- Save the file

Restore

- Open phpMyAdmin
- Click **Import**
- Select the saved .sql file
- Click **Go**

This ensures data can be recovered if required.

User Documentation Generating Reports and Forms

The user can easily open the Employee Report in the Reports section in order to create an employee report. Employee information at this report is presented in a systematic format, and all the information on workforce can be readily accessed by the HR personnel.

Similarly, attendance record reports are prepared. Daily/historical attendance information, such as the authentication method and the device that captured the appearance, could be viewed by users with the help of these reports. This allows checking the attendance more quickly and efficiently.

The device usage reports will give an idea of the frequency of using each of the biometric devices, which will help the administrators to track the operational efficiency. Salary reports provide the payroll information in an easily digestible format to specific pay periods and minimise the manual computing of the payrolls by the HR personnel.

Adding a New Employee

1. Open Microsoft Access
2. Open Employee form
3. Enter employee details
4. Click Save

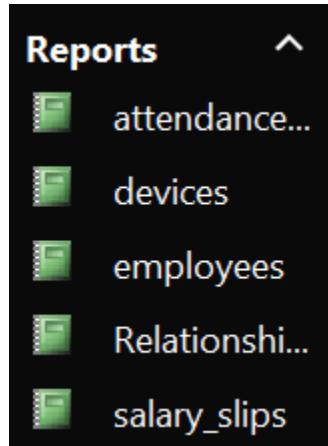
Recording Attendance

1. Open Attendance form
2. Select Employee and Device
3. Enter date, time, and authentication method
4. Save record

Generating Reports

1. Open Reports section
2. Select required report
3. View or print results

Reports



All Acc... Search... attendance device employee Relationships Relationships for database final.muneeb_Backup salary slip employees

employees Thursday, December 11, 2025
4:51:32 AM

employee_id	first_name	last_name	department	designation	hire_date	st
1	Ali	Khan	IT	Developer	10/12/2025	ad
2	Sara	Ahmed	IT	Software Engineer	10/12/2025	ad
3	Faiz	Ali	IT	Network Admin	10/12/2025	ad
4	Ahmed	Ali	IT	System Analyst	10/12/2025	ad
5	Muhammad	Aun	IT	Database Assistant	10/12/2026	ad
6	Hussnain	Ahmed	IT	Programmer	10/12/2025	ad

Report View

Devices					Thursday, December 11, 2025 4:52:18 AM
device_id	device_code	location	is_active	created_at	
1	NS-UNIT-01	Head Office - Main Gate	1	##########	
2	NS-UNIT-02	Factory - Entrance	1	##########	

Forms:

Attendance Record:

attendance_records

attendance_id	1
employee_id	1
device_id	1
attendance_date	12/1/2025
attendance_time	12/24/2025 9:00:00 AM
auth_method	FINGERPRINT
in_out_status	IN
created_at	12/1/2025

Add Record

Save Record

Delete Record

Close Form

Devices:

devices

device_id	1
device_code	NS-UNIT-01
location	Head Office - Main Gate
is_active	1
created_at	12/9/2025 2:48:30 AM

Add Record

Save Record

devices

[Save Record](#)

[Delete Record](#)

[Close Form](#)

Employees:

employees

employee_id

first_name

last_name

department

designation

hire_date

status

created_at

employees

[Add Record](#)

[Save Record](#)

[Delete Record](#)

[Close Form](#)

Salary Slip:

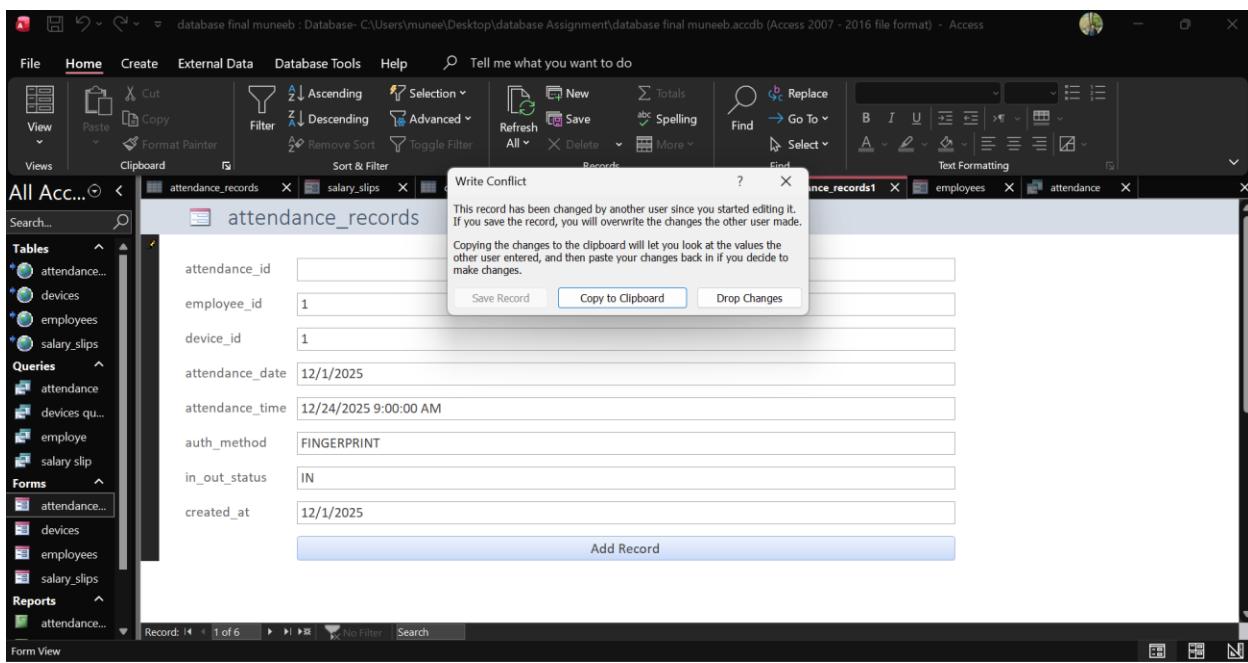
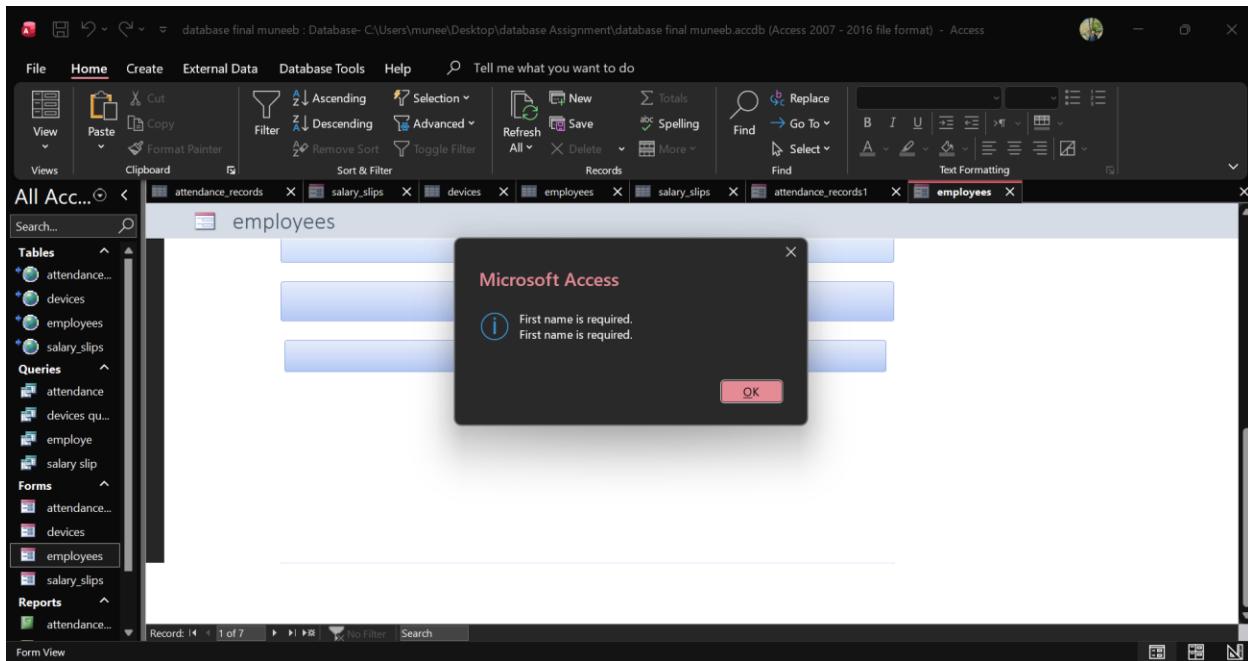
salary_slips

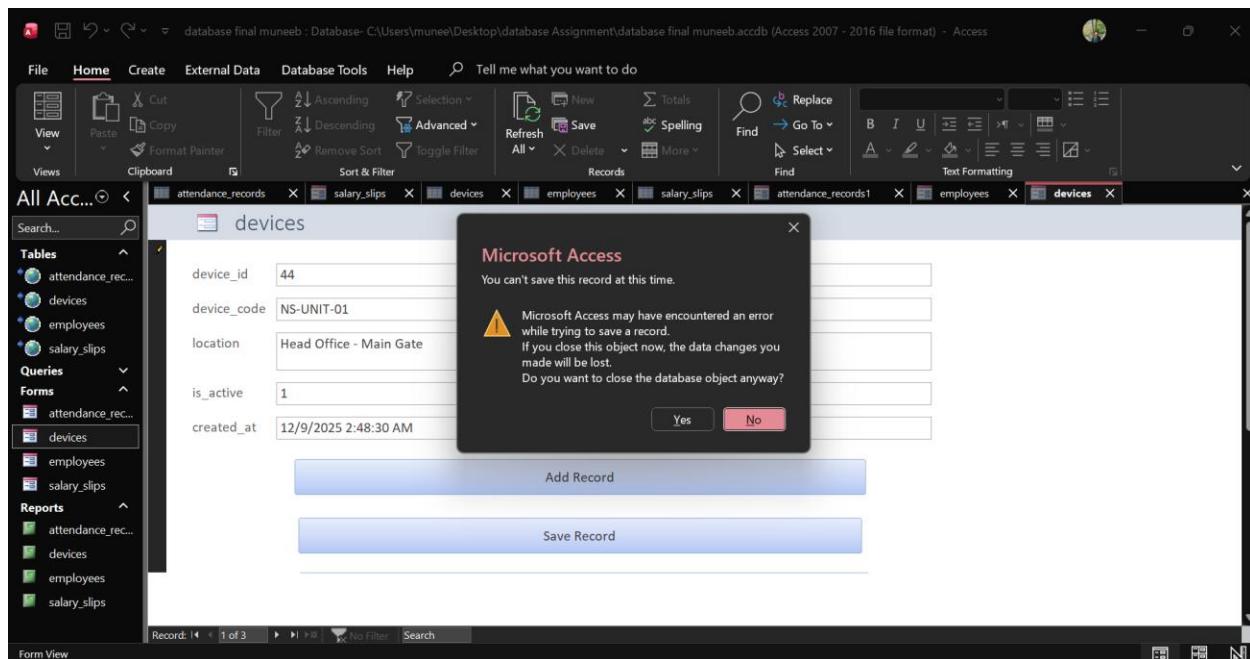
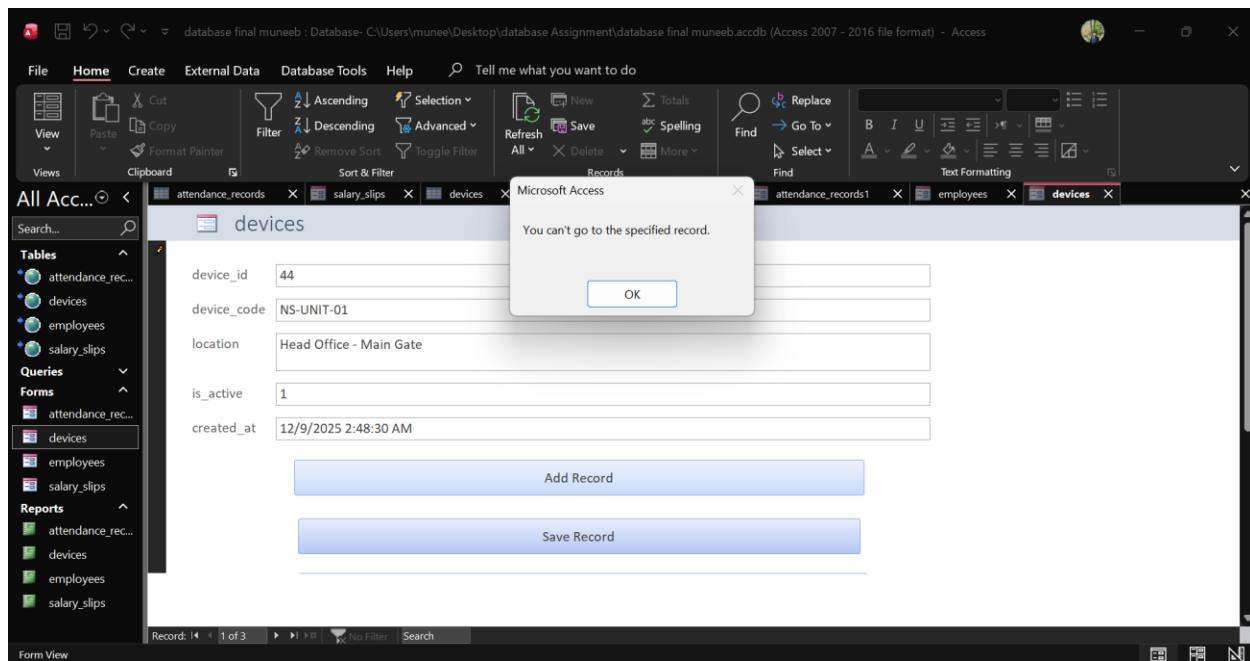
salary_id	<input type="text" value="1"/>
employee_id	<input type="text" value="1"/>
period_start	<input type="text" value="12/1/2025"/>
period_end	<input type="text" value="12/31/2025"/>
total_hours	<input type="text" value="176"/>
basic_salary	<input type="text" value="50000"/>
overtime_hours	<input type="text" value="10"/>
overtime_rate	<input type="text" value="500"/>
gross_salary	<input type="text" value="55000"/>
net_salary	<input type="text" value="53000"/>
generated_at	<input type="text" value="12/31/2025"/>

salary_slips

overtime_rate	<input type="text" value="500"/>
gross_salary	<input type="text" value="55000"/>
net_salary	<input type="text" value="53000"/>
generated_at	<input type="text" value="12/31/2025"/>

Errors Form Validation:





Evaluation of the System

Achievement of Project Objectives

Generally, the attendance management system attains the aim of the project as initially set. The system can be reliably safe in the storage and management of the attendance information produced by the biometric devices by various authentication systems such as fingerprint, RFID

and PIN. It also facilitates processing of the data on salary-related data and offers structured reports where they can be utilized by the HR staff and the management. The application of relational database model makes it to be accurate, consistent, and organised in a logical way during the system lifecycle.

MySQL and Microsoft Access front-end were found to be a good mixture in order to create an effective prototype solution. The MySQL is good in terms of strong data handling and able to process structured queries, whereas the access offers simple and familiar interface to the data. Combined these technologies provide a performance, reliability and ease of use balanced system.

Strength of the Relational Database Design

A modular, relational structure of the system can be considered one of its major assets. All the reports are kept in different but interconnected table such as attendance record, employee records, devise records and salary records. Such division enhances the integrity of data and minimizes redundancy since a single information is stored in one place and additional information is used when required. Consequently, maintenance and update of the database is simplified, as modification in a single table will not affect the information found in another table in a negative way.

Relationships and well-defined primary keys are also used in order to make sure that every record of attendance and salary is related to a valid employee and device. Such design decision is in line with the best practices of a database and is also associated with the stability of the whole system.

Effectiveness of Queries and Reports

The queries and reports provided by the system are very instrumental in transforming the data stored into significant information. Data in a table format can be grouped into one query, which can show the data in a structured format as opposed to the users being exposed to unstructured data in a table format. Reporting based on these queries also makes the use more manageable because the information will be displayed in a readable and professional format.

This solution will enhance usability and security. The user is also capable of getting the information he/she requires like attendance reports, device history and remuneration information without having the chance of making unwanted changes to the underlying data. Due to this fact, the system is quite appropriate in operational application in an organizational setting.

Overall System Reliability and Suitability

In a general sense, the system represents a high degree of reliability of a prototype. Data verification was performed to ensure the storage is done in the correct manner, relationships are acting as required and queries give the correct results. This design can also be extended in the future, since it is possible to add new features or data items without having to redesign the entire design.

To sum the case, the attendance management system offers a solid basis of satisfying technical and user needs. Although it is in the form of a prototype, its form, functionality and design guidelines render it fit to be advanced to a complete production system.

When incorrect or incomplete data is entered, the system displays validation error messages. These messages guide users to correct mistakes before saving data, reducing errors and improving data quality.

Recommendations and Future Improvements

As much as the system proves effective as a prototype and fulfils the present project requirements, it has a number of areas which can be improved to make it more appropriately applicable in the real-world and large-scale implementation. The direct incorporation of the database with the biometric attendance tools, based on the IoT, would be one of the most valuable additions. In the existing prototype, the data on attendance is entered manually to test it, which is convenient when developing a prototype but not in the real life. Depending on the system as a means of integration of live devices by APIs or middleware services, attendance data could be stored automatically in real time. This would not only save on manual work, but also avoid human error, guarantee greater accuracy of data and enable the management to keep track of the attendance on the go.

The other improvement that is worthwhile would be the fact that the front-end needs to be migrated to a web-based application instead of remaining in Microsoft Access. Although Access can be applied in a small setting or situations where there are only one or two users, it is restricted in situations where there are many users and they require access at once. The front-end will be built on the latest web technologies to ensure that several HR personnel or managers could access the system simultaneously without having to be at the same location. This would contribute greatly to usability and flexibility particularly organisations having remote teams or multiple branches. The web interface in the future would also allow making the API integration with the IoT devices more seamless and scalable.

Additional controls could also enhance the security and reliability. The use of role-based access control would make sure that the users only view and deal with the data concerning their duties. To give an example; attendance and salary report would be accessible to the HR staff, system administrators would have all control over configuration as well as maintenance.

Sensitive information including salary details would be secured as transmitted database connections would use encrypted connections between the front end and database server. Moreover, they should have automated and scheduled backups to protect in case of losing data as a result of system failure, accidental loss or security breach.

Lastly, more can be done to the system such as the inclusion of graphical dashboards and advanced reporting systems. The visual dashboards of attendance pattern, rate of late arrivals, absenteeism, and monthly report would enable the management to have a rapid exposure of data that are easy to process as compared to raw tables, or text report. Such visualizations would enhance the data-driven decision-making, allow finding workforce problems in advance, and enhance the general planning in the organization. All these additions would turn the current prototype into a powerful, scalable and enterprise-ready solution of attendance management.

Issue Identified	Impact	Proposed Improvement	Benefit
Manual attendance entry	Time-consuming	Integrate biometric devices	Faster and accurate data
Desktop-only Access	Limited access	Web-based front-end	Multi-user access
Basic validation	User errors	Improved form validation	Better data accuracy
No dashboards	Hard to analyse trends	Visual dashboards	Better management decisions

Final Conclusion

This project shows that the development of the database is well organized with the planning, implementation, testing, documentation, and evaluation. The last attendance management tool is dependable, easy to use and scalable such that it gives NetSolace Inc. a robust platform on which to operate the attendance data collected through biometric machines. As it continues to develop, the system can be upgraded to a fully integrated enterprise level system.

References

Apache Friends (2024) *XAMPP – Apache + MySQL + PHP + Perl*. Available at: <https://www.apachefriends.org> (Accessed: 12 December 2025).

MySQL (2024) *MySQL Database Documentation*. Oracle Corporation. Available at: <https://dev.mysql.com/doc/> (Accessed: 12 December 2025).

Oracle (2024) *MySQL ODBC Connector Documentation*. Available at: <https://dev.mysql.com/doc/connector-odbc/en/> (Accessed: 12 December 2025).

Microsoft (2024) *Microsoft Access documentation*. Available at: <https://support.microsoft.com/access> (Accessed: 12 December 2025).