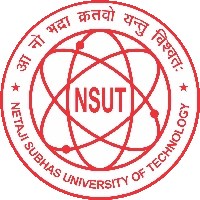
## NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

UNIVERSITY EMPLOYEE PAYROLL MANAGEMENT SYSTEM

SESSION 2024-25



SUBMITTED BY:

SAHIL WADHAWAN 2022UEE4665

YASHITA SINGH 2022UEE4668

SAHIL SHARMA 2022UEE4656

# ACKNOWLEDGEMENT

We would like to extend my sincere and heartfelt thanks towards all those who have helped me in making this project. Without their active guidance, help, cooperation and encouragement, we would not have been able to present the project on time.

We extend our sincere gratitude to our teacher for their moral support and guidance during the tenure of my project.

We also acknowledge with a deep sense of reverence, our gratitude towards our parents and other faculty members of the university for their valuable suggestions given to us in completing the project.

|  |  |  |
| --- | --- | --- |
| S.NO | NAME | ROLL NO. |
| 1. | Sahil Wadhawan | 2022UEE4665 |
| 2. | Yashita Singh | 2022UEE4668 |
| 3. | Sahil Sharma | 2022UEE4656 |
|  |  |  |

DATE: 31/03/2024 TEACHER’S SIGNATURE

# DECLARATION

We, the undersigned group members, collectively declare that the project report titled "UNIVERSITY EMPLOYEE PAYROLL MANAGEMENT SYSTEM " represents our original work and is the result of our collaborative research and analysis. We confirm that all sources of information, data, images, and materials used in this project have been duly acknowledged and cited in the bibliography section. Any assistance or contributions received from individuals or organizations are appropriately acknowledged.

Furthermore, we collectively declare that this project report has not been submitted for any academic or professional purpose elsewhere. Any similarities or resemblance to the work of others have been properly cited and referenced in accordance with our established academic standards and guidelines.

# INDEX

1. INTRODUCTION
   1. Introduction to problem area
   2. Problem area
   3. About the project
   4. Objective of project
   5. Requirement analysis
2. HARDWARE REQUIREMENTS
   1. Hardware requirements for project implementation
   2. Software tools
3. DATABASE DESIGN
   1. ER MODEL
   2. ER to schema
   3. Functional dependencies
   4. Primary keys
   5. Table relationship
   6. Normalisation (B.C.N.F.)
   7. Natural join to prove lossless decomposition
4. IMPLEMENTATION
   1. Implementation in XAMPP
   2. Implementation of application with code
   3. Snapshots of running application
   4. Working of the app
5. CONCLUSION
   1. Future scope of the work
   2. Conclusion
   3. References
   4. bibliography

# UNIVERSITY EMPLOYEE PAYROLL MANAGEMENT SYSTEM

## CHAPTER 1: INTRODUCTION

## 1.1 Introduction to problem area:

Employee payroll management is a fundamental aspect of any organization, including educational institutions like universities. With a diverse workforce comprising faculty, staff, and administrative personnel, efficiently managing payroll processes becomes crucial for ensuring timely and accurate compensation. Traditional manual methods of payroll management are often labor-intensive, error-prone, and time-consuming, leading to inefficiencies and potential discrepancies in employee payments. Therefore, there arises a need for an automated Employee Payroll Management System (EPMS) designed to streamline payroll operations and enhance overall administrative efficiency within the university setting.

## 1.2 Problem Area:

This DBMS Project titled Employee Payroll Management System (EPMS) seeks to tackle the challenges associated with manual payroll processing within the university environment. It aims to streamline payroll tasks, minimize errors, and improve the overall efficiency of payroll management processes. The system aims to automate tasks such as salary calculations, tax deductions, attendance management, and generating payroll reports, providing a secure and reliable platform for managing employee compensation effectively.

## 1.3 About the Project

The Payroll Management System is a simple PHP/MySQL project that manages and calculates the employee's payroll of a certain company. This payroll project can compute the employee's payroll monthly and semi-monthly. The payroll management system itself a static configuration for payroll calculation such as the number of working days of a company, which in this system, the setup is for a company that has 5 working days in a week and that's for Monday-Friday schedule and each day has a static schedule of 8 Am-12 PM and 1 PM-5 PM time log schedule. Using the said system static configuration, this will help to calculate the number of absences and tardiness/undertime of an employee in each day within the selected date range of the payroll. The payroll management system includes also the computation of different allowances and deductions of an employee and this feature of the system can be managed dynamically. Each employee will have their own setup for their allowances and deduction and also can be set as monthly, semi-monthly, and applicable only for one-time earning or deduction.

## 1.4 Objectives of Project:

The main objective of the Employee Payroll Management System (EPMS) project is to develop a comprehensive and automated platform for managing employee payroll within the university. It aims to:

* Automate payroll processes: Implement automated systems for calculating salaries, handling deductions, and processing payroll transactions.
* Minimize errors: Reduce manual errors and inaccuracies in payroll calculations and processing.
* Improve efficiency: Streamline payroll tasks to save time and resources for administrative staff.
* Enhance data security: Ensure the confidentiality and integrity of employee payroll information through robust security measures.
* Facilitate reporting: Generate detailed payroll reports and insights to aid in decision-making and auditing processes.
* Provide user-friendly interface: Design an intuitive user interface to simplify payroll management tasks for administrators and staff.

## 1.5 Requirement Analysis:

The Employee Payroll Management System aims to automate and streamline payroll processes within the university. The system is designed to meet the following requirements:

**1.Employee Management:** The system must be able to store and manage employee data, including personal details (name, contact information), employment details (position, department), and salary information.

**2.Payroll Processing:** The system should automate payroll processing tasks such as salary calculations, tax deductions, allowances, and deductions based on predefined rules and regulations.

**3.Attendance Tracking:** It should allow for the recording and tracking of employee attendance, including attendance dates, times, and status (present, absent, late).

**4.Allowances and Deductions:** The system should support the management of various allowances and deductions applicable to employees, such as housing allowance, transportation allowance, taxes, insurance premiums, and loan deductions.

**5.Department and Position Management:** It must maintain information about different departments and positions within the university, including department names, descriptions, positions, and salary scales.

**6.User Management:** The system should provide user authentication and authorization functionalities to control access to different system features and data based on user roles and permissions.

**7.Reporting and Analysis:** It should generate various reports and analytics related to payroll, attendance, allowances, deductions, and employee data to support decision-making and auditing processes.

**8.Security and Data Integrity:** The system must ensure the security and integrity of employee payroll data through robust authentication, encryption, and access control measures.

**9.User Interface:** The system should have a user-friendly interface that is intuitive and easy to navigate, facilitating efficient interaction for both administrators and employees.

**10.Scalability and Performance:** It should be scalable to accommodate the growing needs of the university and perform efficiently even with large volumes of data and concurrent user access.

**11.Compliance**: The system should adhere to relevant laws, regulations, and policies governing payroll management, ensuring compliance with legal requirements and standards.

**12.Integration:** It may need to integrate with other university systems or external applications for data exchange and interoperability.

By addressing these requirements, the Employee Payroll Management System will effectively meet the needs of the university in managing employee payroll and related processes.

## CHAPTER 2: HARDWARE REQUIREMENTS

## 2.1 Hardware Requirements for Project Implementation:

**1.Computer/Desktop:**

Operating System: Compatible with Windows/MAC OS/Linux

Processor: Intel Pentium Duo or equivalent

RAM: Recommended 8GB

Storage: Minimum 1GB

**2.Server:**

Processor: Multi-core processor (e.g., Intel Xeon)

RAM: Minimum 8GB Storage: SSD storage recommended for faster data access

Network: Gigabit Ethernet interface

**3.Database Server:**

DBMS: MySQL

CPU and RAM: Similar to server requirements, adequate for database operations

## 2.2 Software Tools

Software/Tech Stacks/Languages Used:

* **XAMPP:**

Description: XAMPP is a comprehensive software package that includes an RDBMS (Relational Database Management System) for creating databases and executing SQL queries.

Purpose: Used for database management and interaction with the project's data.

* **PHP:**

Description: PHP is a server-side scripting language used for developing dynamic web pages and web applications.

Purpose: Used for server-side scripting to handle form submissions, database interactions, and business logic in the project.

* **HTML/CSS/JavaScript:**

Description: HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript are fundamental web technologies used for building the frontend of web applications.

Purpose: HTML is used for structuring web pages, CSS for styling and layout, and JavaScript for adding interactivity and dynamic behavior to the user interface.

* **MySQL Workbench:**

Description: MySQL Workbench is a visual database design tool and database administration tool for MySQL databases.

Purpose: Used for database design, modeling, and administration tasks such as creating tables, defining relationships, and executing SQL queries.

* **Visual Studio Code (VS Code):**

Description: Visual Studio Code is a lightweight and versatile code editor with support for various programming languages and extensions.

Purpose: Used as the primary integrated development environment (IDE) for writing, editing, and debugging code files in PHP, HTML, CSS, and JavaScript.

By utilizing XAMPP, PHP, HTML/CSS/JavaScript, MySQL Workbench, and Visual Studio Code, the project benefits from a comprehensive and efficient development environment for creating a robust and user-friendly Employee Payroll Management System.

## CHAPTER 3: DATABASE DESIGN

## 3.1 ER Model



## 3.2 ER to Schema

1. **employee** (employee\_id,employee\_no, firstname, middlename, lastname, department\_id, position\_id, salary)
2. **deductions** (deduction\_id, deduction, description)
3. **allowances** (allowance\_id, allowance, description)
4. **department** (department\_id, name)
5. **position** (position\_id, department\_id, name)
6. **employee\_allowances** (employee\_allowance\_id, employee\_id, allowance\_id, type, amount, effective\_date, date\_created)
7. **employee\_deductions** (employee\_dudction\_id, employee\_id, deduction\_id, type, amount, effective\_date, date\_created)
8. **attendance** (attendance\_id, employee\_id, log\_type, datetime\_log)
9. **payroll** (payroll\_id, ref\_no, date\_from, date\_to, type, status, date\_created)
10. **payroll\_items** (payroll\_items\_id, payroll\_id, employee\_id, present, absent, late, salary, allowance\_amount, allowances, deduction\_amount, deductions, net, date\_created)
11. **users** (users\_id, doctor\_id, name, address, contact, username, password, type)

## 3.3 : Functional Dependencies

1. **Employee:** employee\_id -> employee\_no, firstname, middlename, lastname, department\_id, position\_id, salary
2. **Deductions:** deduction\_id -> deduction, description
3. **Allowances**: allowance\_id -> allowance, description
4. **Department:** department\_id -> name
5. **Position:** position\_id -> department\_id, name
6. **employee\_allowances:** employee\_allowance\_id -> employee\_id, allowance\_id, type, amount, effective\_date, date\_created
7. **employee\_deductions**: employee\_dudction\_id -> employee\_id, deduction\_id, type, amount, effective\_date, date\_created
8. **attendance**: attendance\_id -> employee\_id, log\_type, datetime\_log
9. **payroll:** payroll\_id -> ref\_no, date\_from, date\_to, type, status, date\_created
10. **payroll\_items**: payroll\_items\_id -> payroll\_id, employee\_id, present, absent, late, salary, allowance\_amount, allowances, deduction\_amount, deductions, net, date\_created
11. **users**: users\_id -> doctor\_id, name, address, contact, username, password, type

## 3.4 : Primary Keys

1. **Employee:** employee\_id
2. **Deductions:** deduction\_id
3. **Allowances**: allowance\_id
4. **Department:** department\_id
5. **Position:** position\_id
6. **employee\_allowances:** employee\_allowance\_id
7. **employee\_deductions**: employee\_dudction\_id
8. **attendance**: attendance\_id
9. **payroll:** payroll\_id
10. **Payroll\_items**: payroll\_items\_id
11. **Users**: users\_id

## 3.5 : Table Relationship

## 

## 3.6 : Normalisation (B.C.N.F)

**B.C.N.F Form :**

**Rule : LHS should be Candidate key or Super key**

**1. table: employee**

**FD :** employee\_id -> employee\_no , firstname, middlename, lastname, department\_id, position\_id, salary

Closure of **[**employee\_id**]+** = employee\_no , firstname, middlename, lastname, department\_id, position\_id, salary

**Candidate Key :** {employee\_id}

**Prime Attributes :** {employee\_id}

employee\_id -> employee\_no , firstname, middlename, lastname, department\_id, position\_id, salary**(T)**

**Hence , in BCNF form .**

**2. table: deductions**

**FD :** deduction\_id -> deduction, description

Closure of **[deduction\_id]+** = deduction, description

**Candidate Key :** { deduction\_id }

**Prime Attributes :** { deduction\_id }

deduction\_id -> deduction, description **(T)**

**Hence , in BCNF form .**

**3. table : allowances**

**FD :** allowance\_id -> allowance, description

Closure of **[**allowance\_id**]+** = allowance, description

**Candidate Key :** { allowance\_id }

**Prime Attributes :** { allowance\_id }

allowance\_id -> allowance, description **(T)**

**Hence , in BCNF form .**

**4. table: department**

**FD:** department\_id -> name

Closure of **[** department\_id**]+** = name

**Candidate Key:** {department\_id}

**Prime Attributes:** {department\_id}

department\_id -> name **(T)**

**Hence, in BCNF form.**

**5. table: position**

**FD:** position\_id -> department\_id, name

Closure of **[**position\_id**]+** = department\_id, name

**Candidate Key:** {position\_id}

**Prime Attributes:** {position\_id}

position\_id -> department\_id**,** name**(T)**

**Hence, in BCNF form.**

**6. table: employee\_allowances**

**FD:** employee\_allowance\_id -> employee\_id, allowance\_id, type, amount, effective\_date, date\_created

Closure of **[**employee\_allowance\_id**]+** = employee\_id, allowance\_id, type, amount, effective\_date, date\_created

**Candidate Key:** {employee\_allowance\_id}

**Prime Attributes:** {employee\_allowance\_id}

employee\_allowance\_id -> employee\_id, allowance\_id, type, amount, effective\_date, date\_created **(T)**

**Hence, in BCNF form.**

**7. table: employee\_deductions**

**FD:** employee\_dudction\_id -> employee\_id, deduction\_id, type, amount, effective\_date, date\_created

Closure of **[**employee\_dudction\_id**]+** = -> employee\_id, deduction\_id, type, amount, effective\_date, date\_created

**Candidate Key:** {employee\_dudction\_id}

**Prime Attributes:** {employee\_dudction\_id}

employee\_dudction\_id -> employee\_id, deduction\_id, type, amount, effective\_date, date\_created **(T)**

**Hence, in BCNF form .**

**8. table: attendance**

**FD :** attendance\_id -> employee\_id, log\_type, datetime\_log

Closure of **[**attendance\_id**]+** = employee\_id, log\_type, datetime\_log

**Candidate Key:** {attendance\_id}

**Prime Attribute:** {attendance\_id}

attendance\_id -> employee\_id, log\_type, datetime\_log **(T)**

**Hence, in BCNF form.**

**9. table: payroll**

**FD :** payroll\_id -> ref\_no, date\_from, date\_to, type, status, date\_created

Closure of **[**payroll\_id**]+** = ref\_no, date\_from, date\_to, type, status, date\_created

**Candidate Key :** { payroll\_id }

**Prime Attributes :** { payroll\_id }

payroll\_id -> ref\_no, date\_from, date\_to, type, status, date\_created **(T)**

**Hence , in BCNF form .**

**10. table : payroll\_items**

**FD :** payroll\_items\_id -> payroll\_id, employee\_id, present, absent, late, salary, allowance\_amount, allowances, deduction\_amount, deductions, net, date\_created

Closure of **[**payroll\_items\_id**]+** = employee\_id, present, absent, late, salary, allowance\_amount, allowances, deduction\_amount, deductions, net, date\_created

**Candidate Key:** {payroll\_items\_id}

**Prime Attributes:** {payroll\_items\_id}

payroll\_items\_id -> payroll\_id, employee\_id, present, absent, late, salary, allowance\_amount, allowances, deduction\_amount, deductions, net, date\_created**(T)**

**Hence, in BCNF form.**

**11. table: users**

**FD:** users\_id -> doctor\_id, name, address, contact, username, password, type

Closure of **[**users\_id**]+** = doctor\_id, name, address, contact, username, password, type

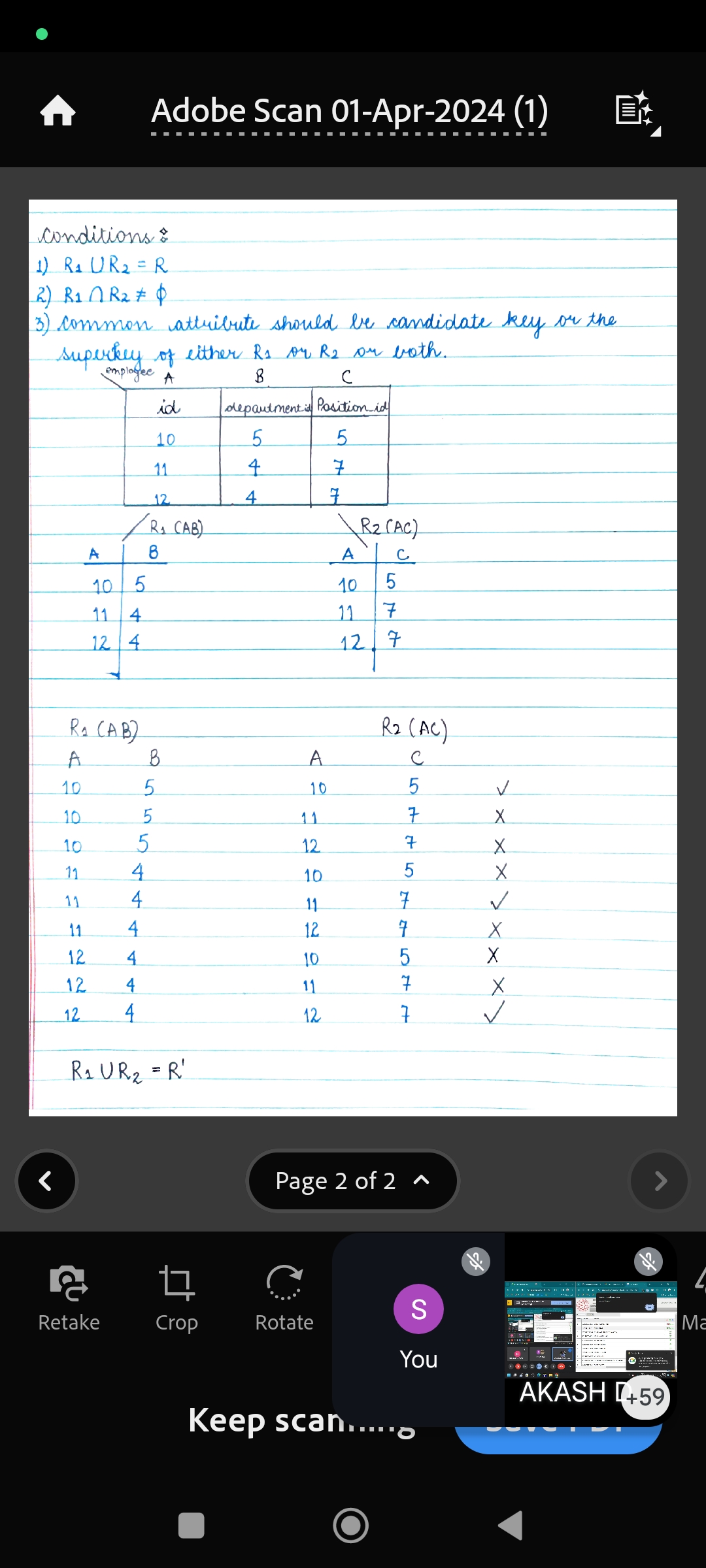
**Candidate Key:** {users\_id}

**Prime Attributes:** {users\_id}

users\_id -> doctor\_id, name, address, contact, username, password, type **(T)**

**Hence, in BCNF form.**

## 3.7 NATURAL JOIN TO PROVE LOSSLESS DECOMPOSITION

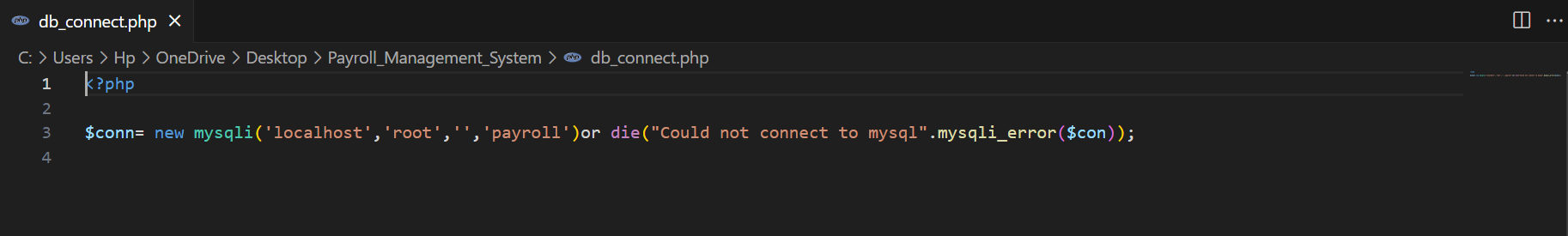




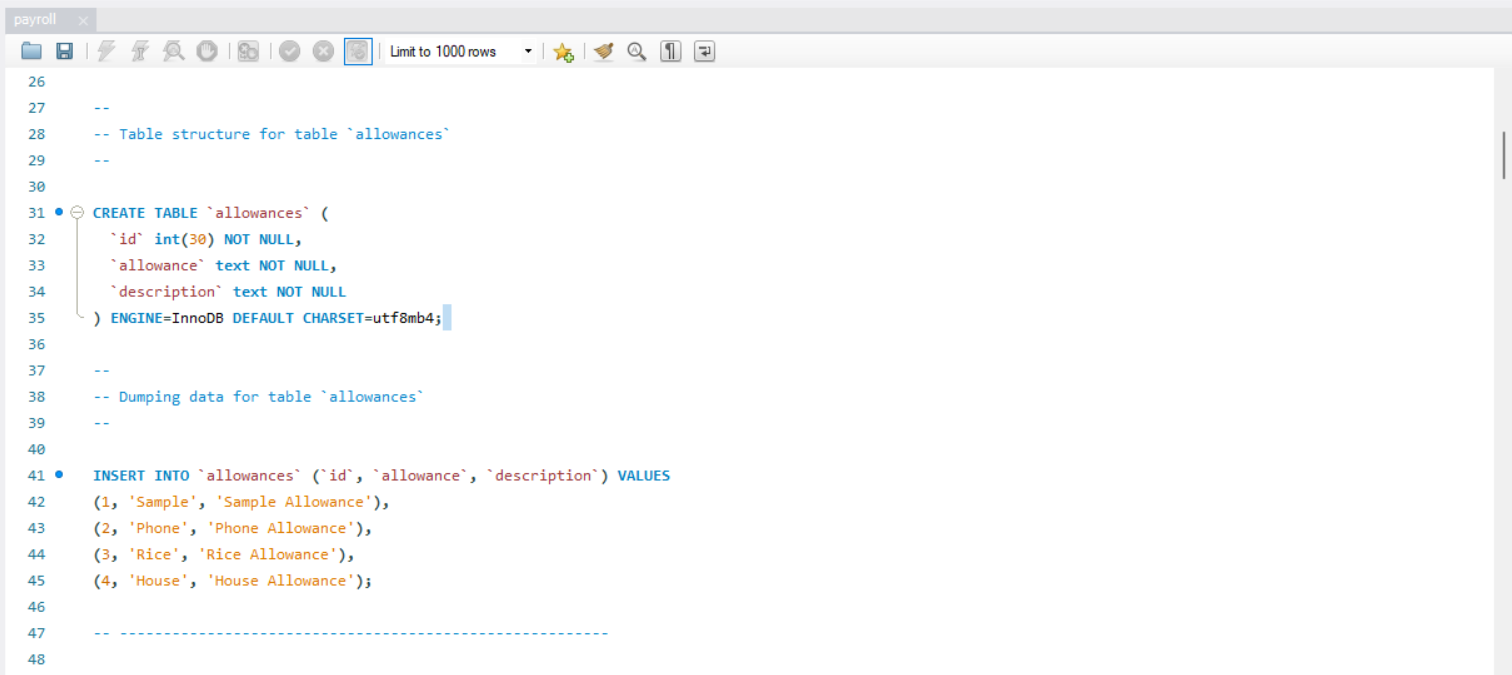
## CHAPTER 4 : Implementation

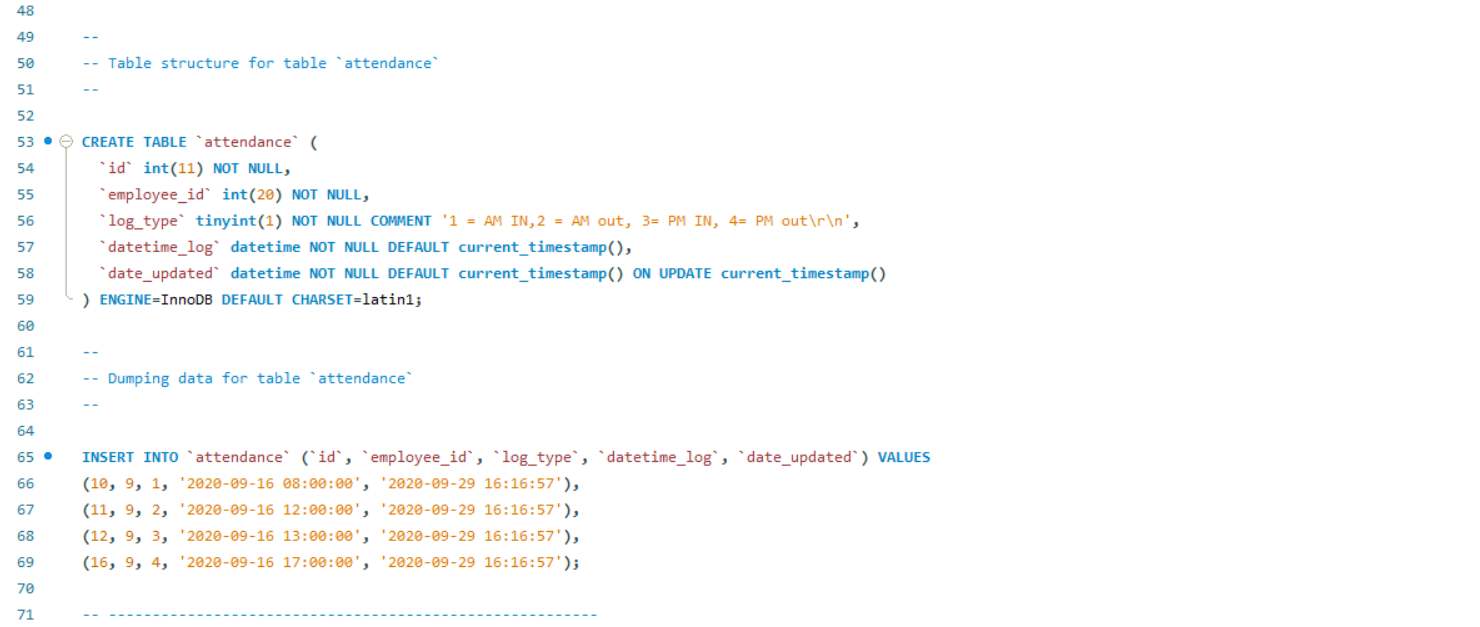
## 4.1 Implementation in xampp

1. Building connection in MySql

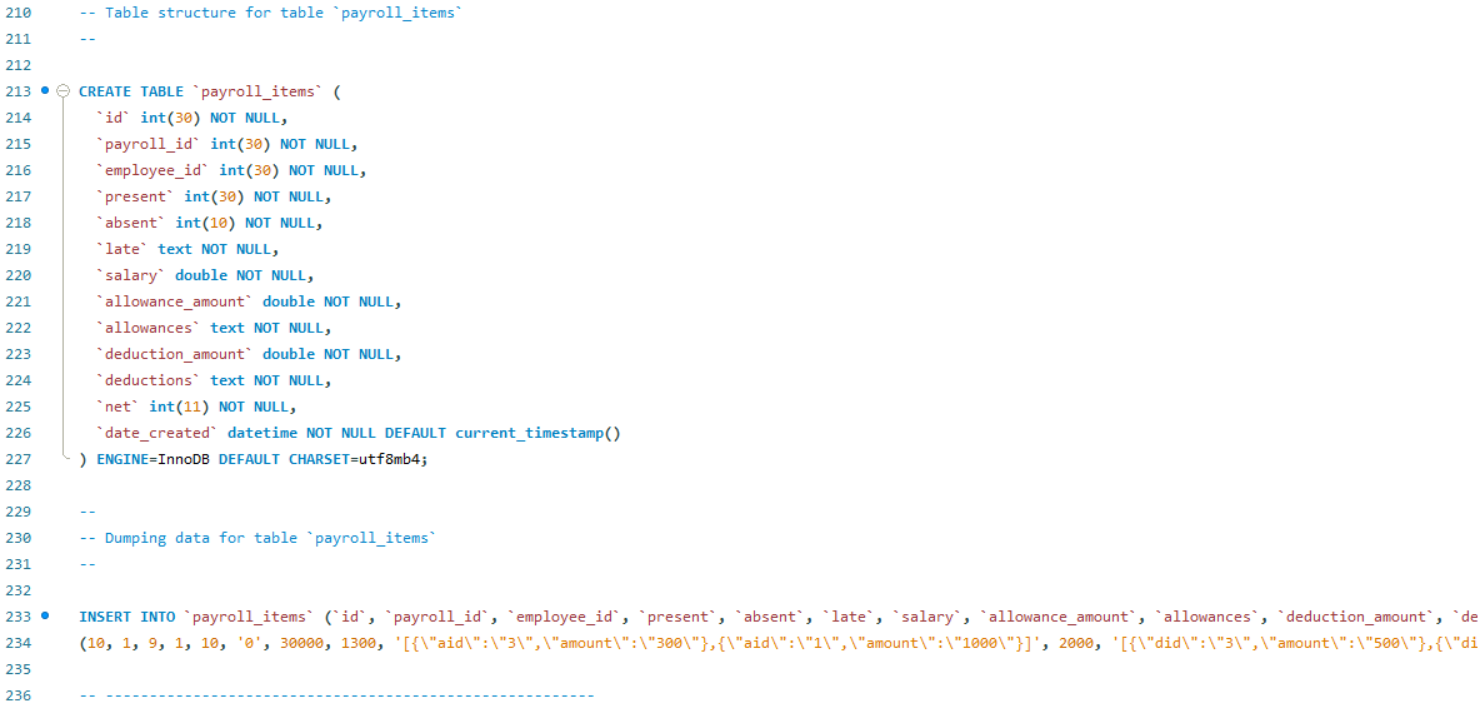


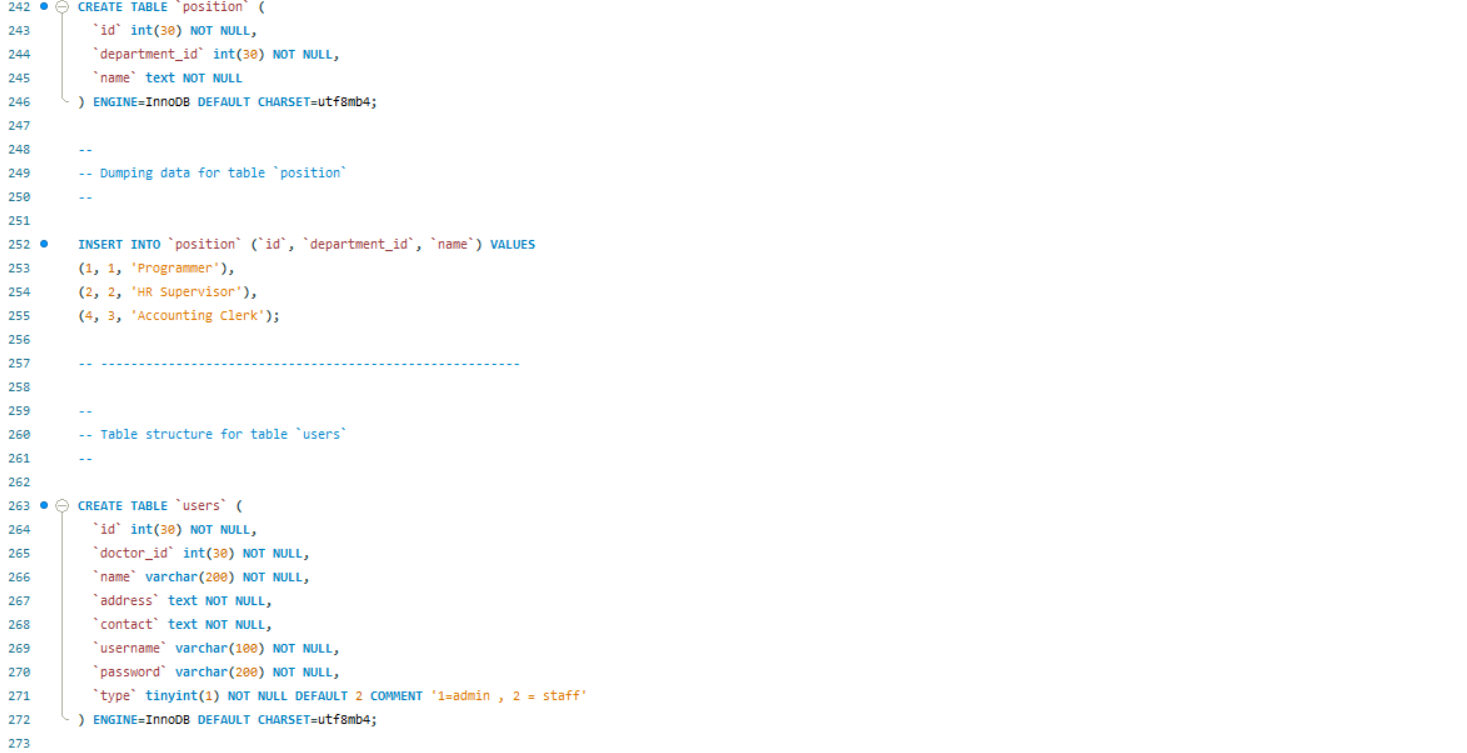
1. Creating tables and inserting values



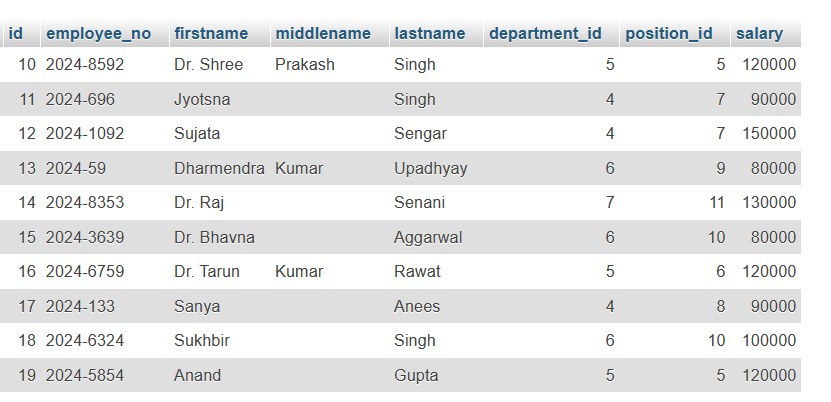




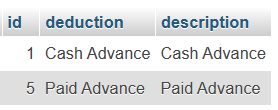




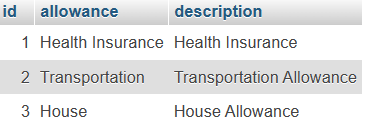
1. Final tables
2. **Employee**

****

1. **Deductions**



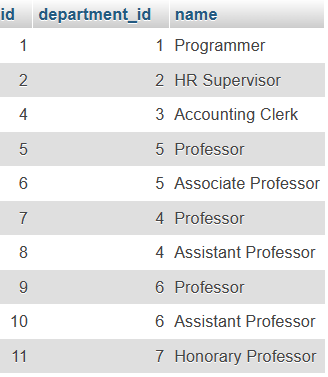
1. **Allowances**



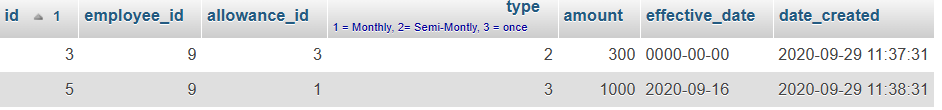
1. **Department**



1. **Position**



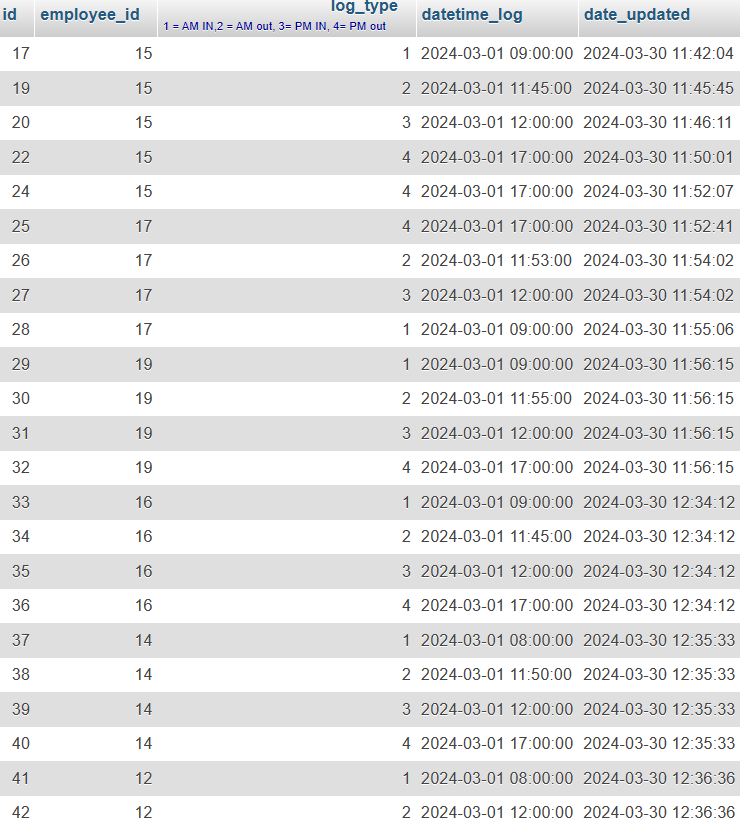
1. **employee\_allowances**



1. **employee\_deductions**



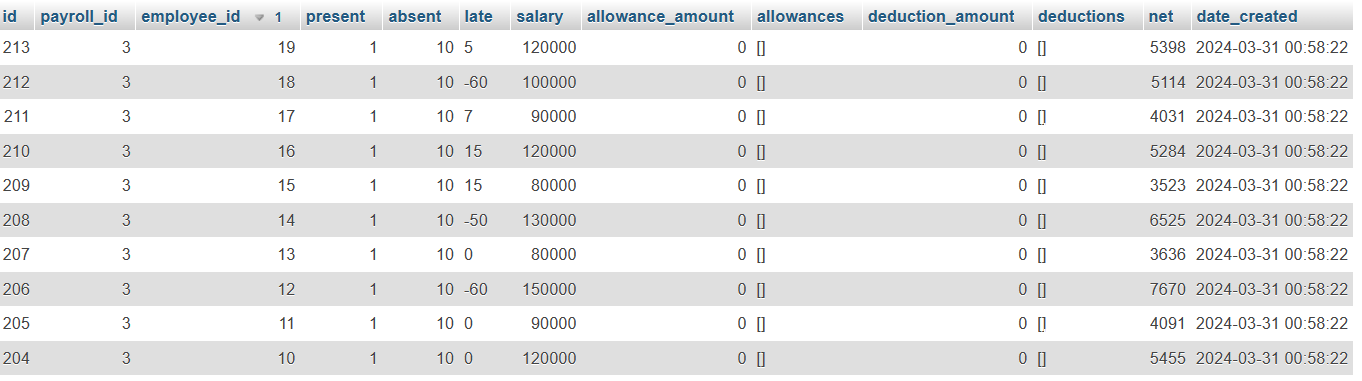
1. **attendance**



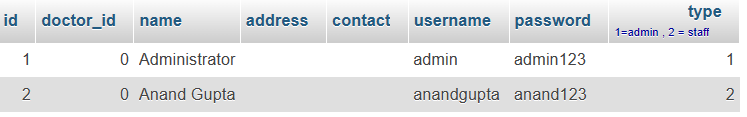
1. **payroll**



1. **payroll\_items**

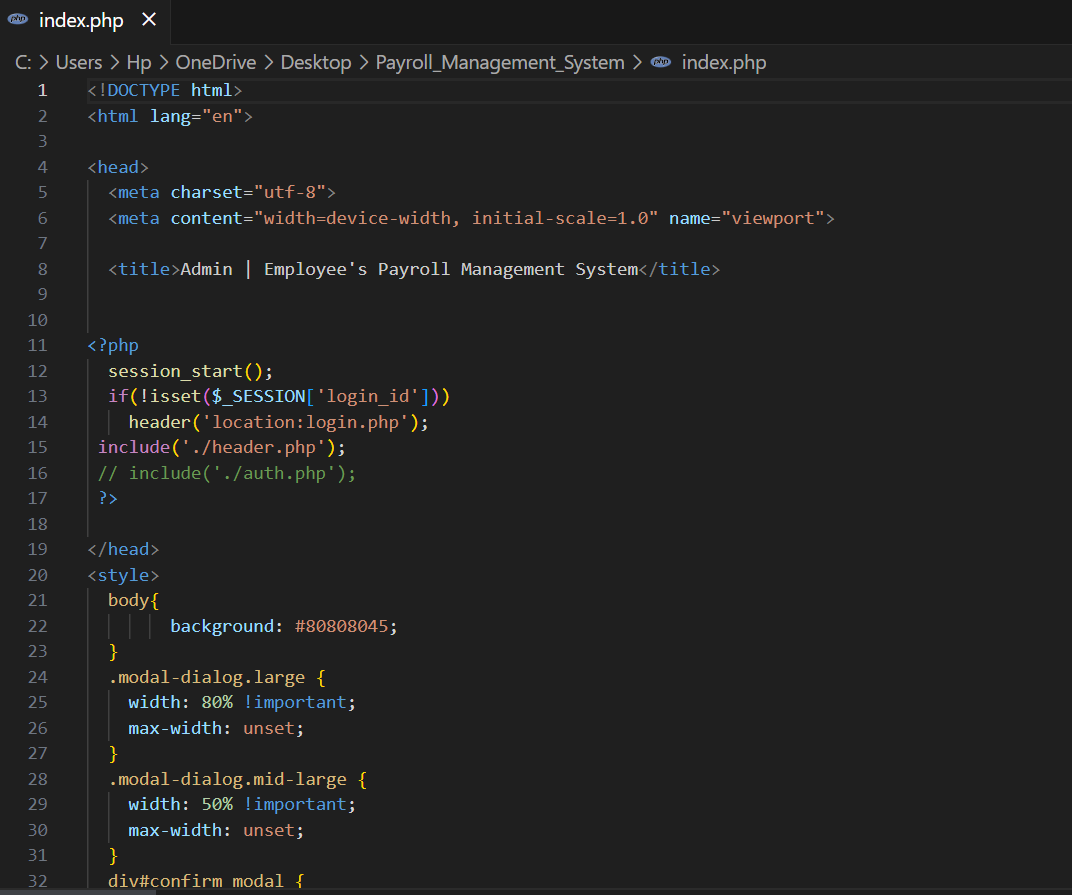


1. **users**



## 4.2 IMPLEMENTATION OF APPLICATION WITH CODE

* INDEX FILE









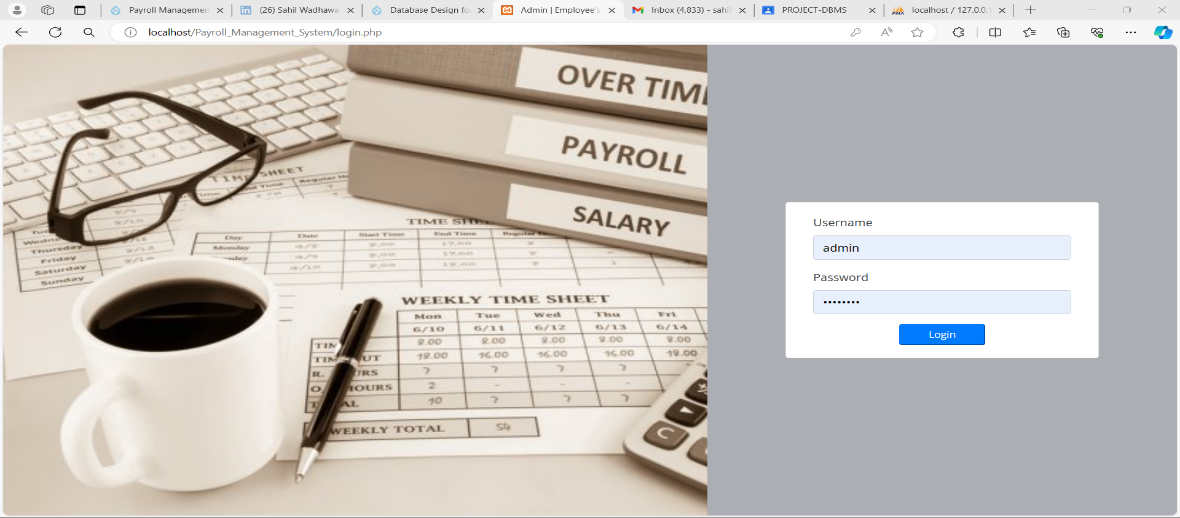


The complete source code can be found through the link:

https://github.com/SahilWadhawan

## 4.3 SNAPSHOTS OF RUNNING APPLICATION

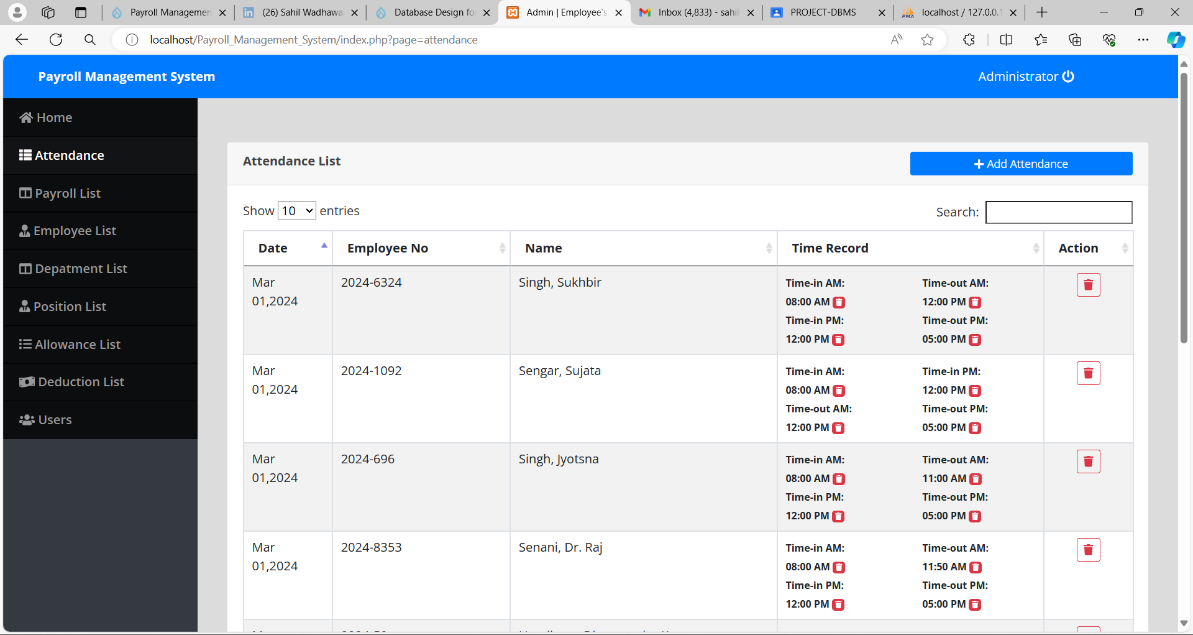
Login Page



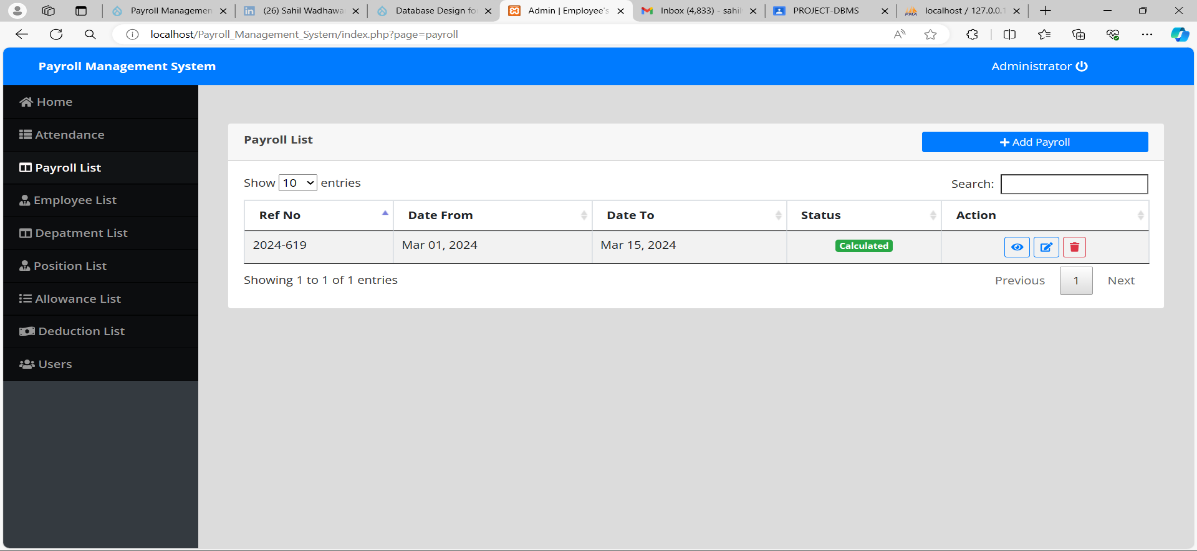
Home Page

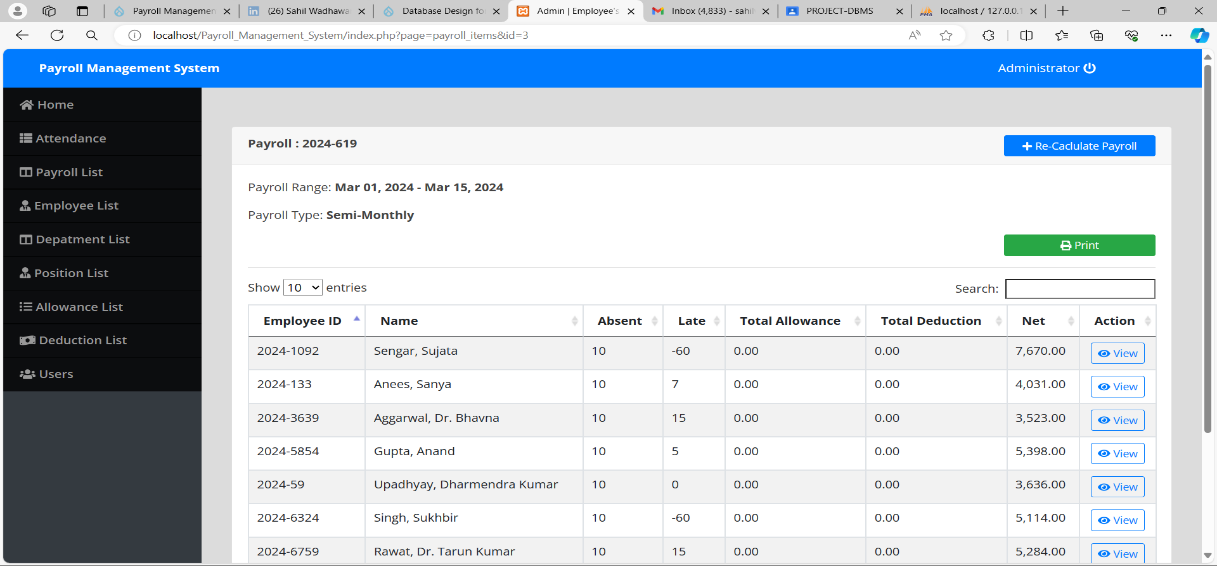


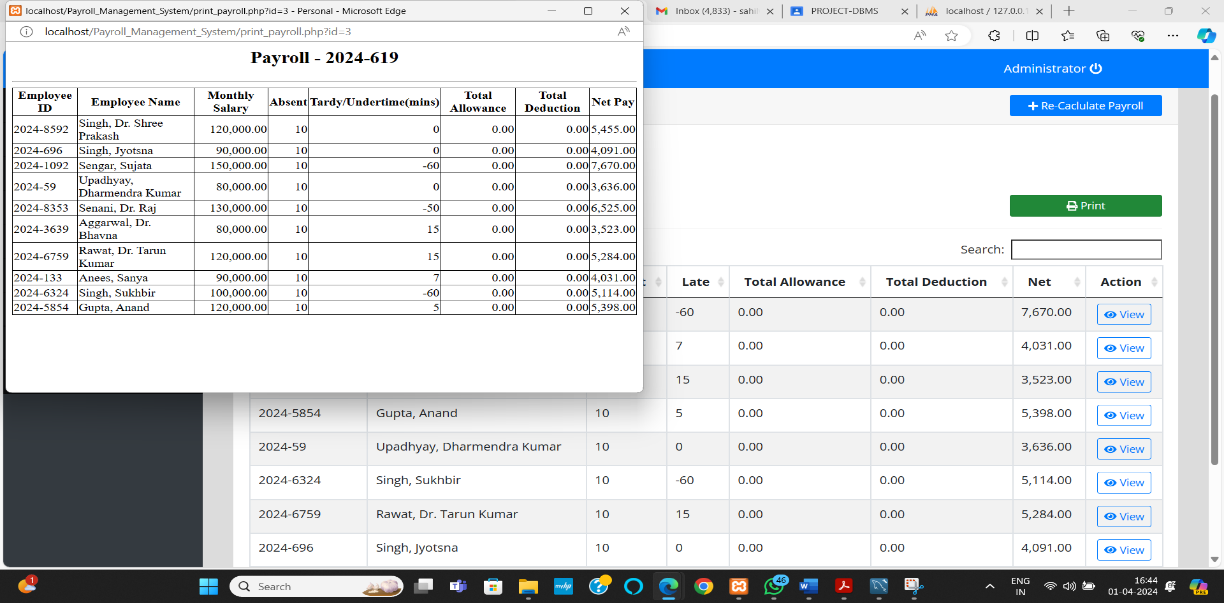
Attendance List Page



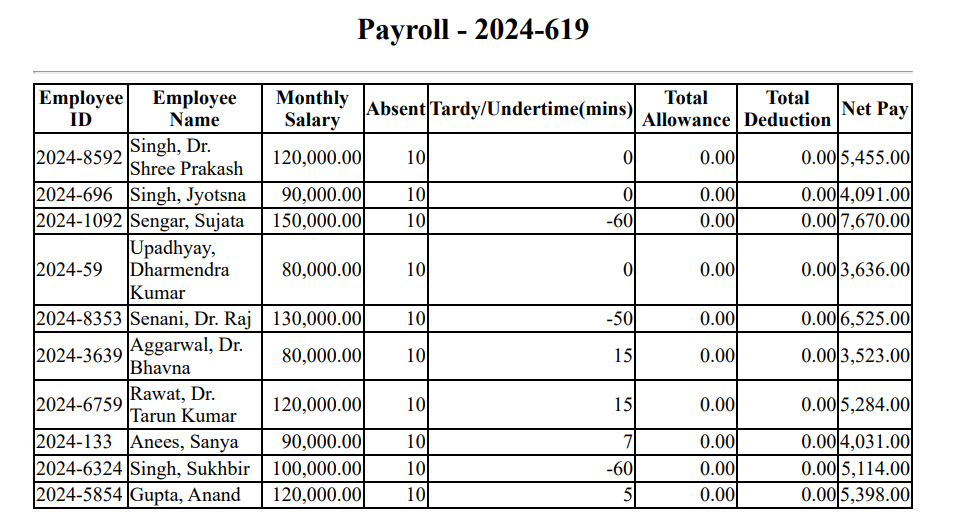
Payroll list page



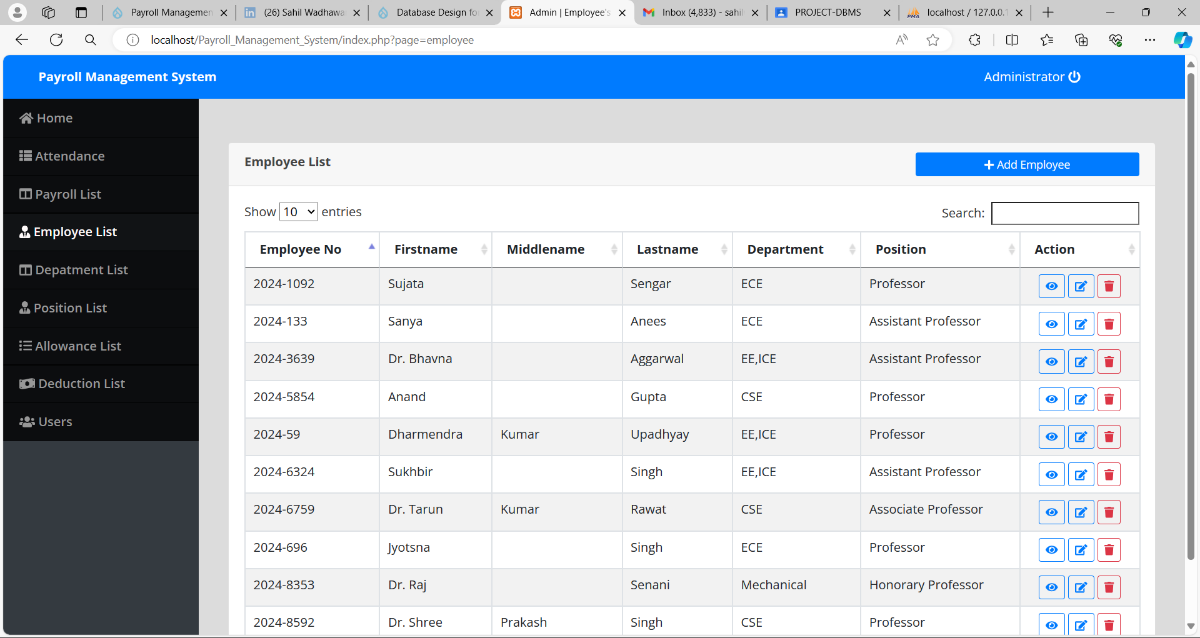




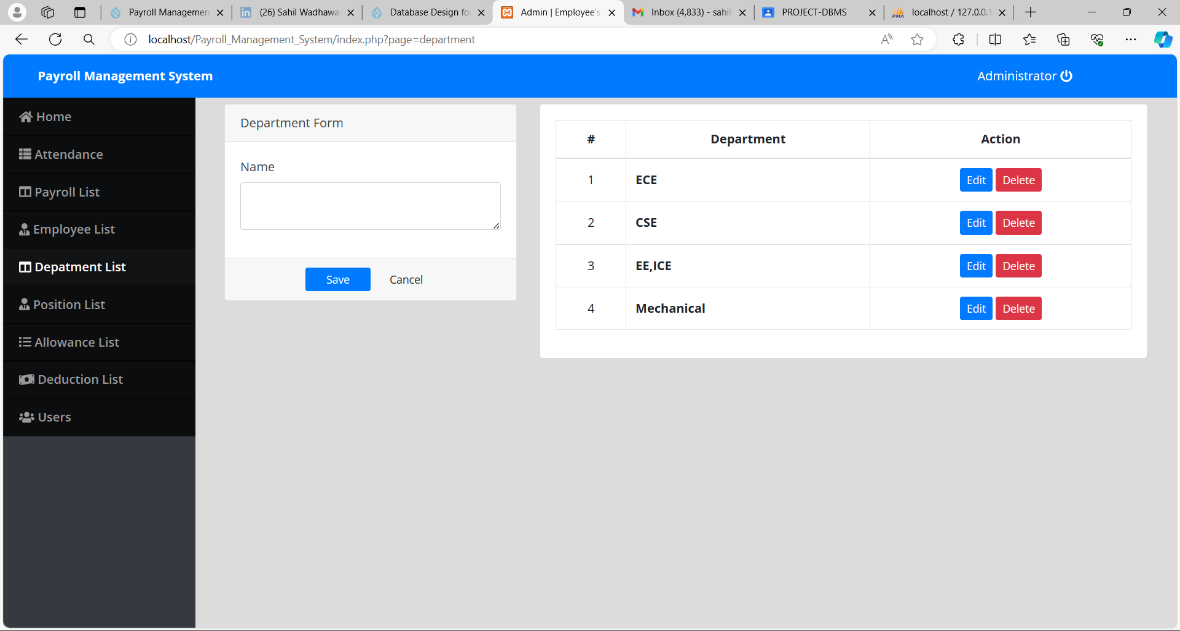
The payroll slip has been generated:



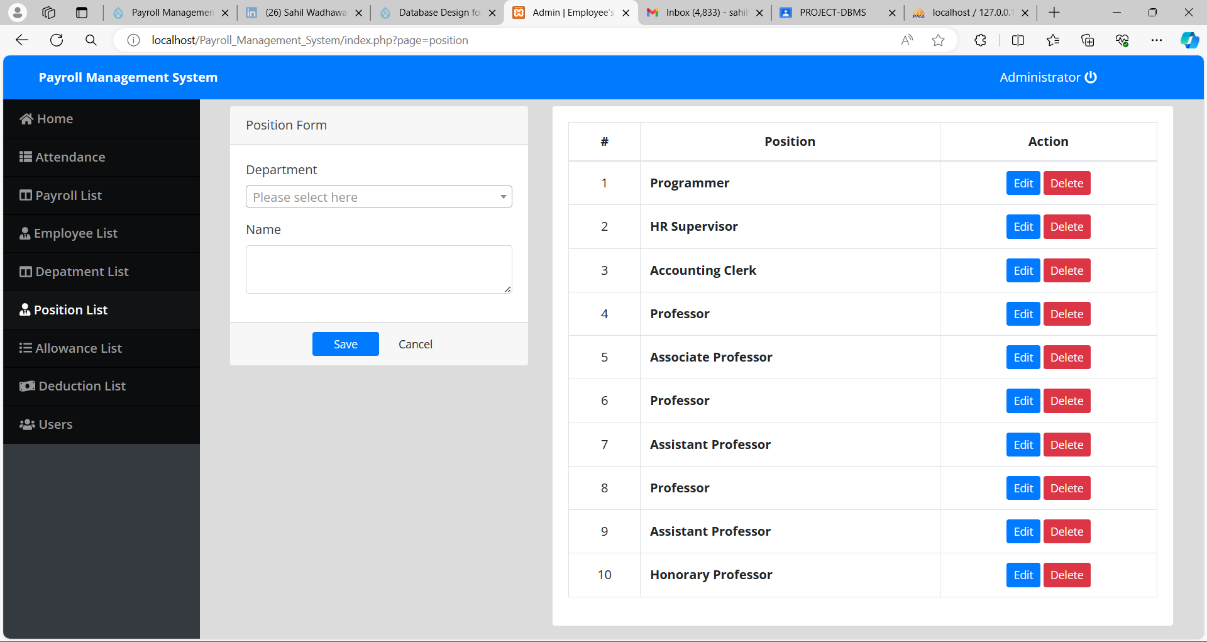
Employee list page



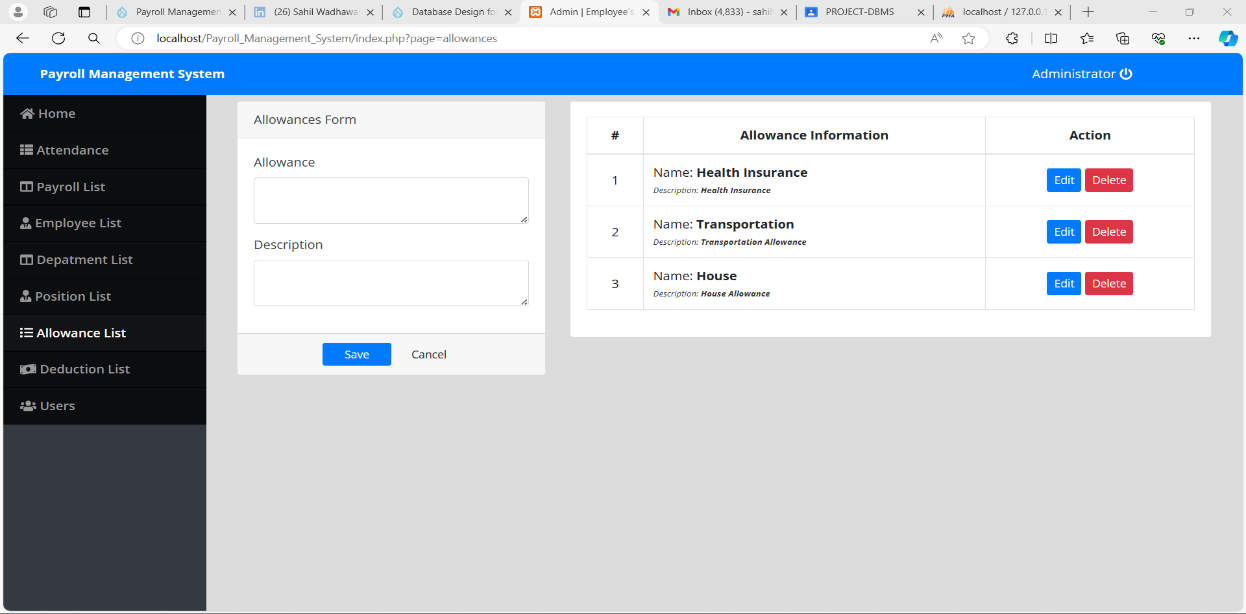
Department list page



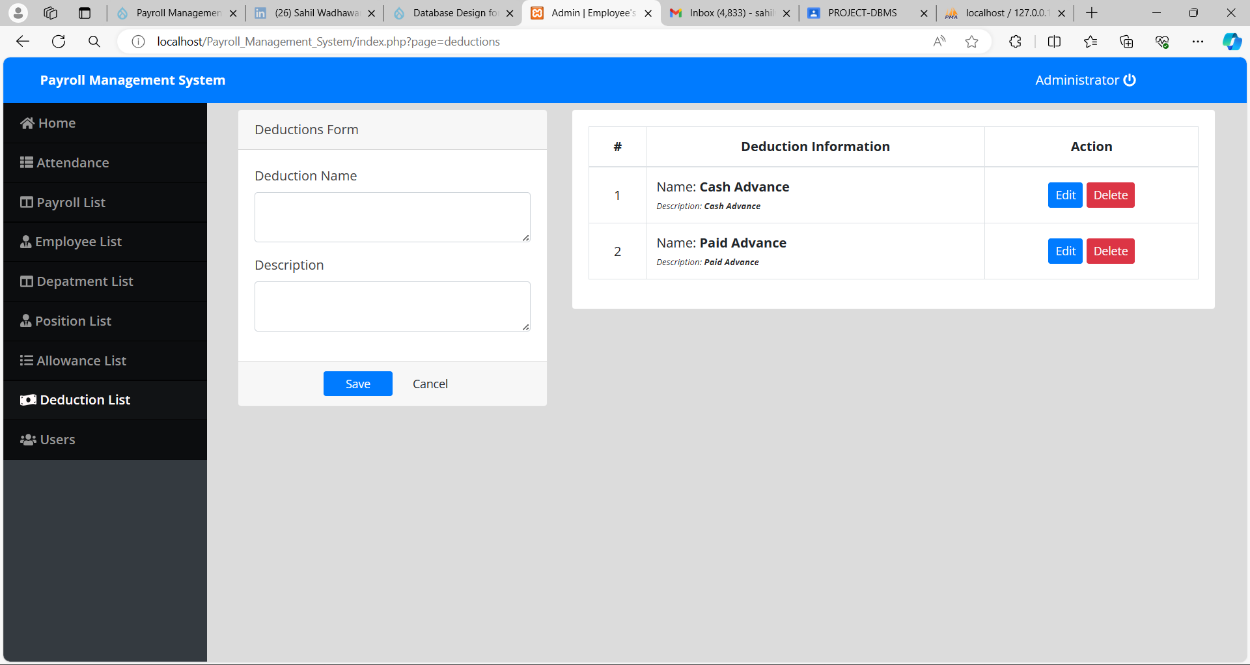
Position list page



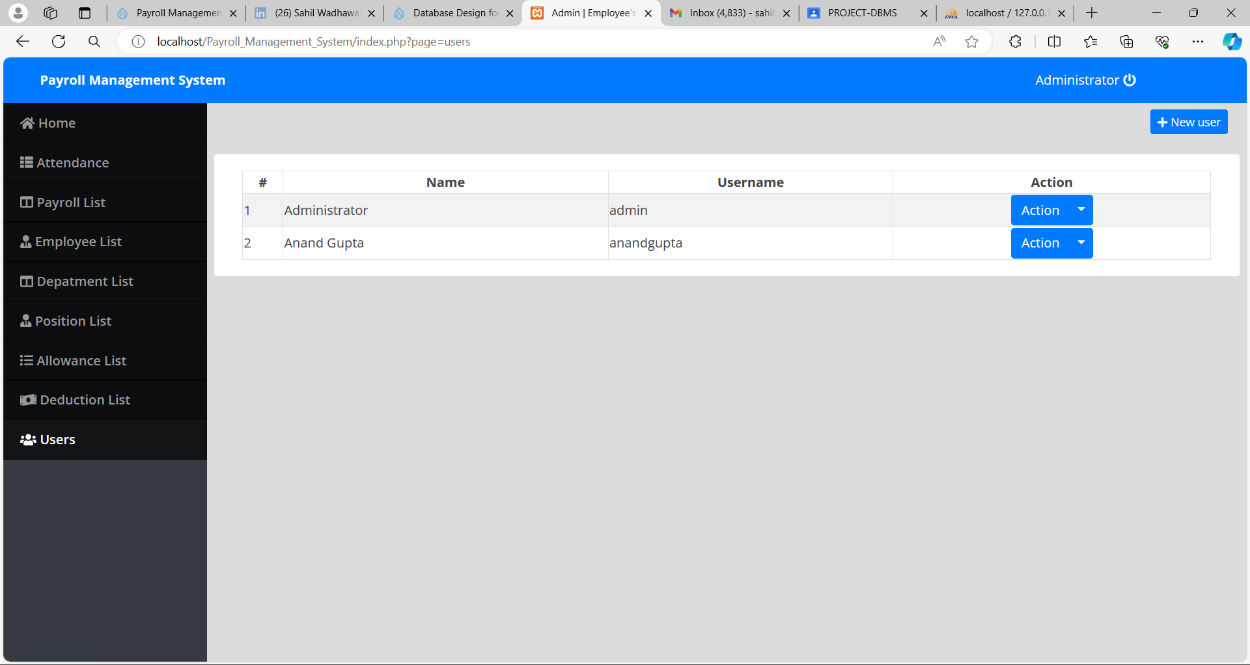
Allowance list page



Deduction list page



Users



## 4.4 working of the app

**The system works as the following**:

1.The System admin will populate the list of department, position, allowances, deduction, and employee.

2.After that, the system admin will set up the employee's allowances and deductions individually.

3.Then, the system admin will encode the employee's daily time record to the system.

4.The admin will create a payroll along with the payroll details such as the payroll date range and the payroll type which is either Monthly or Semi-monthly.

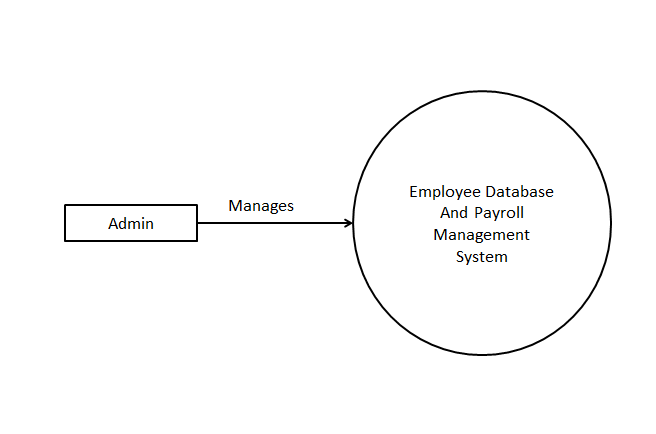
5.After the creation of the payroll details, the admin will calculate the payroll of all employees by clicking the calculate.

Then, there it is, the payroll was generated and calculated and it is also ready to print.

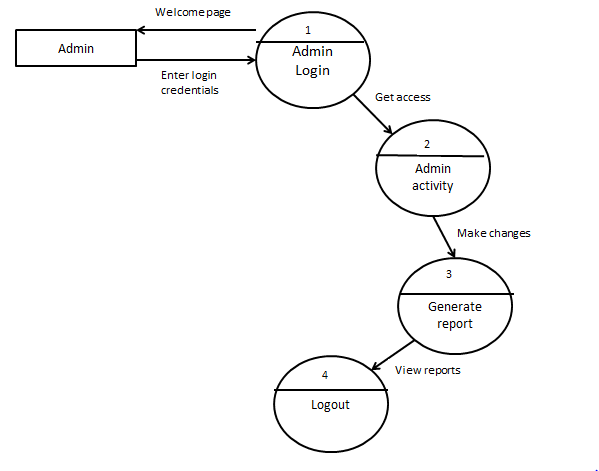
## 

DFD( Data Flow Diagram)

LEVEL 0



LEVEL 1



## chapter 5: conclusions

## 5.1 FUTURE SCOPE OF THE WORK

* The option to print the records In future.
* We intend to add a leave structure in the future.
* We would like to implement a regular backup mechanism to back up the employee database to avoid disasters.
* The system can be developed in such a way that its existing features can be modified to better versions.

## 5.2 CONCLUSION

In conclusion, the Employee Payroll Management System project has been designed with the specific needs of small-scale organizations in mind, where the administrative workload is manageable and the number of employees is limited. The system primarily serves the needs of a single user, the admin, who has complete control over all aspects of employee data management within the organization. Key functionalities of the system include the ability for the admin to add, modify, update, and delete employee records as needed.

Overall, the primary objective of this project is to streamline administrative processes, save time, reduce costs, and manage records efficiently. By providing a comprehensive yet straightforward solution tailored to the needs of small-scale organizations, the Employee Payroll Management System aims to improve operational efficiency and contribute to organizational success.

## 5.3 references

1) NSUT Administration Department for data

2) Dr. Anand Gupta, Associate Professor (DBMS)

## 5.4 bibliography

<http://nsut.ac.in/en/department/faculty/36>

<https://www.greythr.com/complete-guide-payroll/>

<https://www.oracle.com/in/human-capital-management/payroll/what-is-payroll-software/>