PAYROLL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

MAADHAV B (2303811710421086)

in partial fulfillment of requirements for the award of the course

CGB1221-DATABASE MANAGEMENT SYSTEMS

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

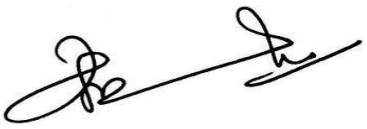
JUNE- 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)

SAMAYAPURAM– 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"PAYROLL MANAGEMENT SYSTEM"** is the bonafide work of **MAADHAV B (2303811710421086)**work who carried out the project during the academic year 2024 - 2025 under my supervision.



SIGNATURE

Mrs.A.DELPHIN CAROLINA RANI,

M.E.,Ph.D

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112

SIGNATURE

Mr.P.MATHESWARAN,

M. E., Ph.D.,

SUPERVISOR

ASSISTANT PROFESSOR

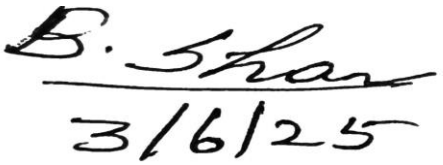
Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112

Submitted for the viva-voce examination held on 03.06.2025

INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on"**PAYROLL MANAGEMENT SYSTEM"**is the result of original work done by me and best of my knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS**.

Signature

MAADHAV B

Place: Samayapuram

Date: 03.06.2025

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution **K.Ramakrishnan College of Technology (Autonomous),** for providing us with the " opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express my deep expression and sincere gratitude to my project supervisor **Mr. P.MATHESWARAN, M.E., P h . D . ,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render my sincere thanks to Course Coordinator and other staff members for providing valuable information during the course

I wish to express my special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society

Be an institute with world class research facilities

Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to

Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems

Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments

Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

ABSTRACT

The Payroll Management System is a web-based application developed using PHP and MySQL, aimed at automating the payroll processing and employee management functionalities of an organization. The system is designed to streamline and simplify the complex process of managing employee records, attendance, salary calculations, deductions, and pay slip generation. It enables administrators to efficiently add and manage employee information, configure salary components, record monthly attendance, and generate pay slips dynamically based on preset rules and attendance data. The application features a user-friendly interface for easy navigation and data input. It incorporates CRUD operations for employee and salary records, attendance tracking with month-wise reporting, and automated pay slip generation with detailed breakdowns. Security and data integrity are maintained through role- based access controls and structured database interactions using MySQL. This system reduces the need for manual calculations and paperwork, thereby improving accuracy, saving time, and enhancing overall HR and payroll workflow efficiency. It is particularly beneficial for small to medium-sized organizations looking for a cost-effective and scalable payroll solution.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Payroll Management System is a web-based application developed using PHP and MySQL to automate employee salary processing. It simplifies tasks such as managing employee records, tracking attendance, calculating salaries, and generating pay slips. With a user-friendly interface and dynamic database interaction, the system reduces manual work, improves accuracy, and enhances HR efficiency. The system also integrates attendance and leave tracking to ensure accurate compensation based on actual workdays and leave entitlements. Additionally, it facilitates the timely generation of detailed pay slips, enhancing transparency and record-keeping. By reducing manual intervention and minimizing errors, the system improves operational efficiency, ensures compliance with statutory regulations, and supports the financial and administrative needs of both HR personnel and employees.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	-	FORM
CRUD	-	Create, Read, Update, and Delete
HTML	-	Hyper Text Markup Language
CSS	-	Cascading Style Sheets
PHP	-	Hypertext Preprocessor
MYSQL	-	MY Structured Query Language

CHAPTER 1
INTRODUCTION

1. Objective

The objective of this Payroll Management System is to design and develop an efficient, reliable, and fully automated solution to manage and streamline all aspects of the payroll process within an organization. The system is intended to handle the end-to-end payroll cycle, including the accurate computation of employee salaries by factoring in key components such as daily attendance, leave records, overtime hours, bonuses, deductions, and applicable tax regulations. By automating these complex and repetitive tasks, the system aims to significantly reduce manual intervention, minimize the risk of errors, and ensure timely and consistent salary disbursement.

In addition, it enables the secure and automated generation of detailed payslips for employees, maintains historical payroll data, and ensures compliance with legal and financial standards. The system also supports better transparency and accountability by providing real-time insights and reports to management, while safeguarding sensitive employee information through secure access controls. Ultimately, this system seeks to enhance the productivity of HR and finance departments, reduce administrative overhead, and foster a more organized, accurate, and employee-friendly payroll management environment.

2. Overview

The Payroll Management System is a dynamic web-based solution developed using PHP and MySQL, designed to automate the complex process of payroll management within an organization. It simplifies tasks such as employee registration, attendance and leave tracking, salary calculation, and tax deduction. By integrating these modules, the system ensures accurate and timely salary processing while minimizing manual effort and errors. The administrator can easily manage employee data, update records, monitor attendance, and calculate net salaries including bonuses, deductions, and allowances.

In addition to salary computation, the system features automated payslip generation, secure user login, and role-based access control to protect sensitive information. A user-friendly dashboard displays key stats like total employees, attendance records, and processed salaries. This centralized approach improves HR and finance department efficiency, enhances transparency, and ensures compliance with tax regulations. Overall, the system provides a reliable, efficient, and secure method for managing payroll operations.

3. SQL and Database Concepts

1. Database Concepts

In the context of a Payroll Management System, a database plays a crucial role in organizing, storing, and managing employee and payroll-related information in a structured and secure manner. The system relies on a relational database model where data is organized into interrelated tables such as *employees*, *attendance*, *salary*, *users*, and *leave records*. Each table uses a primary key to uniquely identify its records—for example, the emp_id in the *employees* table. Foreign keys are used to create logical relationships between tables, such as linking the *attendance* and *salary* tables to the *employees* table through the employee ID. The use of a Database Management System (DBMS) ensures data integrity, security, and consistency across all modules of the system, making it a reliable solution for handling critical payroll operations.

2. Normalization

Normalization in a Payroll Management System involves organizing the database to reduce redundancy and improve data integrity. It starts with ensuring that each field contains only one value (1NF), followed by making sure all attributes depend on the entire primary key (2NF). Finally, it ensures that non-key fields do not depend on other non-key fields (3NF). This process helps to efficiently manage employee data, attendance, salaries, and tax information, ensuring consistency and accuracy while minimizing unnecessary repetition. By applying normalization, the database becomes easier to maintain and more efficient for payroll processing.

3. Structured Query Language

In a Payroll Management System, Structured Query Language (SQL) is used to manage and manipulate employee data, attendance, salary, and tax records. SQL commands like `CREATE TABLE` define the structure of tables such as employees and salaries, while `INSERT`, `UPDATE`, `DELETE`, and `SELECT` allow for adding, updating, removing, and retrieving data. These SQL operations ensure accurate and efficient management of payroll data, helping in generating reports, calculating salaries, and tracking employee information.

4. Advanced SQL operations

Advanced SQL operations in a Payroll Management System involve using `JOIN` to combine data from tables like employees, attendance, and salaries based on common fields such as employee IDs. Aggregate functions like *SUM()* calculate total salaries, *AVG()* find average pay, and *COUNT()* track attendance. Constraints such as `NOT NULL` ensure mandatory fields, `UNIQUE` prevents duplicate records, and *CHECK* enforces rules on salary ranges or attendance statuses, ensuring data integrity and accurate payroll processing.

5. SQL and Database Concept used in project

In your Payroll Management System project, several key SQL and database concepts are utilized to ensure smooth data management and processing. The primary concept used is Normalization, where data is organized into multiple related tables to eliminate redundancy and ensure data integrity. For example, employee information, attendance records, salary details, and tax data are stored in separate tables, linked by common fields like employee ID. This helps maintain consistency and makes the system more efficient in terms of storage and querying.

SQL operations play a crucial role in managing and interacting with the database. SELECT queries are used to retrieve employee details, attendance records, and salary data for processing and reporting. INSERT and UPDATE are used to add new records (e.g., employee registrations or salary changes) and modify existing ones (e.g., updating attendance or salary details). JOINs are frequently used to combine data from different tables, such as linking employees with their attendance and salary records, ensuring that all related information is accessed together. Additionally, aggregate functions like SUM() and COUNT() are used to calculate total salaries and attendance for reporting purposes. Constraints like NOT NULL, UNIQUE, and CHECK are applied to enforce data integrity, ensuring that critical fields like employee IDs or salaries adhere to specific rules and that data remains consistent throughout the system.

CHAPTER 2
PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work for the Payroll Management System aims to create a complete solution that automates the process of salary calculation, employee attendance tracking, and payslip generation. The system will allow administrators to manage employee information, record their daily attendance, apply deductions, and compute net salaries accurately. By reducing the need for manual calculations and paperwork, this system ensures faster payroll processing and minimizes the chances of human errors. It will also allow users to view and manage records in a structured format through a clean and interactive web interface.

The system will be built using PHP for server-side scripting and MySQL as the database backend. HTML and CSS will be used for designing the frontend to ensure a responsive and user-friendly experience. SQL queries such as SELECT, INSERT, UPDATE, and DELETE will be used to interact with the database, while JOIN operations will be applied to fetch combined data from multiple tables like employees, attendance, and salary. Aggregate functions such as SUM() and AVG() may be used for generating salary summaries and reports. Database normalization techniques will also be applied to avoid redundancy and ensure data consistency.

In terms of structure, the application will have modules for employee management, attendance management, salary configuration, and payroll reporting. Each module will interact with the central database to ensure synchronization and real-time updates. A login system with authentication will be implemented to allow only authorized personnel to access or modify sensitive information. The final system will not only improve the speed and accuracy of payroll operations but also provide transparency and easy record-keeping for administrative and auditing purposes.

2.2 Block Diagram

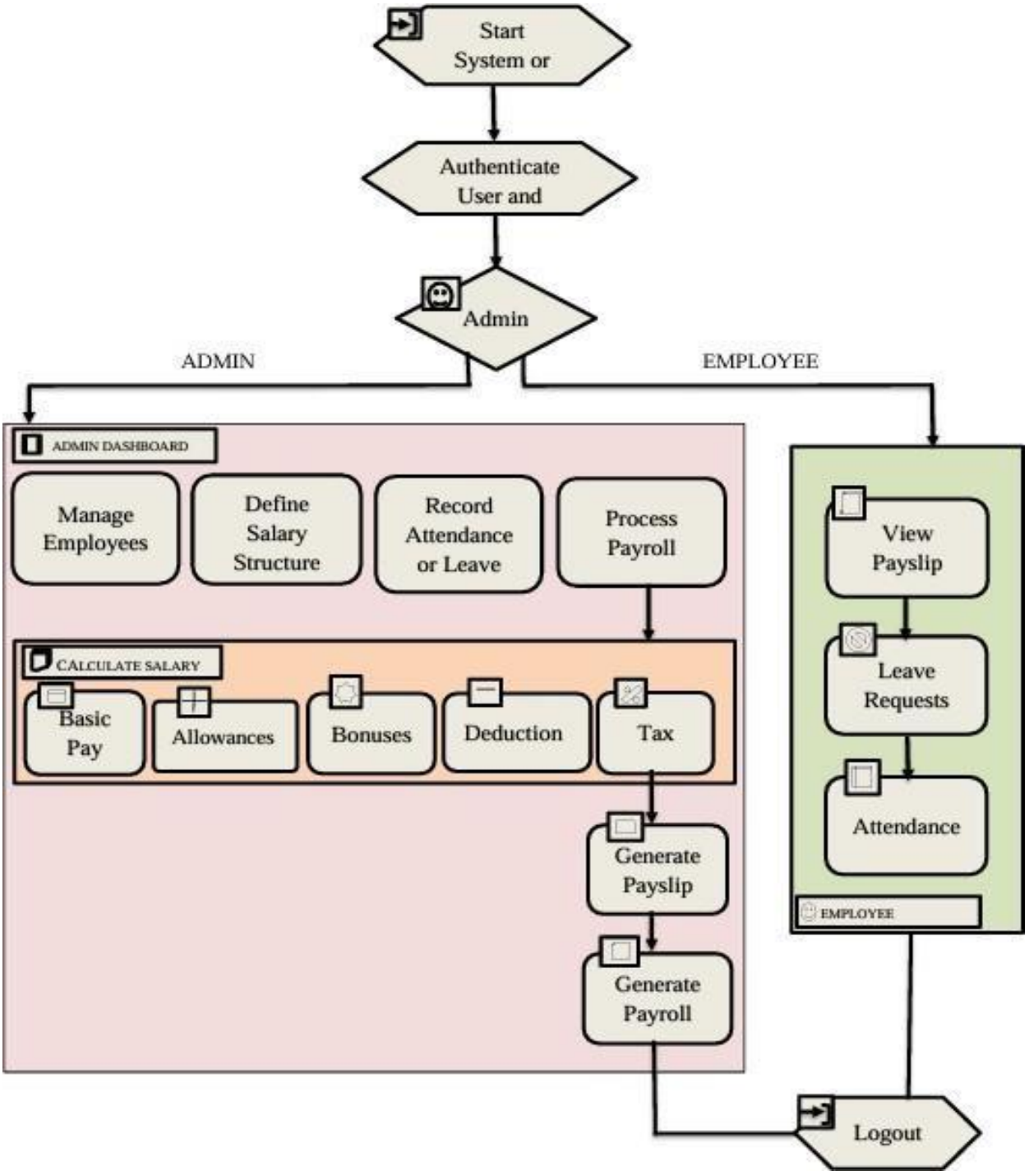


Fig 2.1

CHAPTER 3

MODULE DESCRIPTION

1. Authentication Module

In the Payroll Management System, **User Authentication** ensures secure access by requiring users to log in with a valid username and password. Once authenticated, a user session is created, allowing only authorized individuals to access the system. The **Role Management** component defines permissions based on user roles such as **Admin** or **HR Staff**, ensuring that sensitive data like salary details or employee records is only accessible to those with the appropriate privileges. This module helps protect confidential payroll information, restricts unauthorized access, and ensures that users can only perform tasks relevant to their role, thereby enhancing both security and system efficiency.

2. Employee Management Module

The **Employee Management Module** in the Payroll Management System is responsible for handling all employee-related information. It allows administrators or HR personnel to add new employees by capturing essential details such as employee ID, name, department, designation, contact information, joining date, and basic salary. This information is stored in the database and serves as the foundation for other modules like attendance tracking and salary processing. Additionally, the module includes a user-friendly interface for viewing all employee details in a structured table format, making it easy for admins to manage and monitor staff data.

By maintaining organized and centralized employee information, this module plays a crucial role in ensuring the efficiency and reliability of the overall payroll process. The module also ensures data integrity by validating input fields and preventing duplicate entries. It acts as the core of the payroll system, as all salary, attendance, and payslip operations are directly linked to employee records.

3. Attendance Management Module

The Attendance Management Module in the Payroll Management System plays a vital role in monitoring and recording employees' daily presence. It allows administrators or HR staff to mark attendance by selecting an employee, date, and status (Present, Absent, or Leave). Each attendance entry is stored in the database and directly linked to the respective employee, ensuring that payroll calculations reflect actual working days. This module supports error-free tracking and eliminates manual entry, which reduces the chances of manipulation or oversight.

The system also offers a user-friendly interface to view attendance history in a structured table format, sorted by date or employee name. Color-coded status indicators (e.g., green for present, red for absent) help users quickly assess attendance patterns. These records serve as a base for calculating deductions, bonuses, and leave balances during salary processing. In future enhancements, this module can be expanded to include leave request approval, automatic absence alerts, and attendance summaries, further improving accuracy and payroll efficiency.

4. Salary Management Module

The Salary Management Module is a core component of the Payroll Management System that handles the calculation and storage of employee salary details. It allows administrators to input components such as basic pay, allowances, and deductions for each employee. The system automatically calculates the net salary by applying the formula: ***Net Salary = Basic Pay + Allowances – Deductions***, ensuring accuracy in salary processing. This module stores the salary data month-wise for each employee, which helps in maintaining payroll records and tracking payment history.

In addition to salary entry, the module provides an organized view of all salary records through the view_salaries.php page. It displays employee names along with their monthly salary breakdown, helping HR staff easily verify and analyze payroll details. This module plays a crucial role in generating accurate payslips and reports, as well as ensuring that salaries are calculated in a timely and consistent manner. Future enhancements may include tax calculations, bonus handling, and export to PDF or Excel for payroll audits and reporting.

5. Payslip Generation Module

The Payslip Generation Module in the Payroll Management System is designed to create detailed, employee-specific salary slips based on the stored salary and employee data. It retrieves the relevant information from the database, including the employee's name, designation, salary components (basic pay, allowances, deductions), and calculates the final net salary. This module provides a clear breakdown of the earnings and deductions, ensuring transparency for both the organization and the employee.

The generated payslip is presented in a structured format that can be printed or saved for future reference. It serves as an official document that can be used for financial records, tax filing, or loan applications. The module enhances payroll efficiency by eliminating manual payslip creation and reducing errors. With a user-friendly interface and print functionality, it simplifies the process of distributing accurate payslips to employees each pay cycle.

6. Dashboard & Report Module

The Dashboard & Report Module serves as the central control panel of the Payroll Management System, providing administrators with a quick overview of key system statistics. It displays real-time data such as the total number of employees, total attendance entries, and the number of salary records processed. These summary insights help administrators monitor system activity and make informed decisions at a glance.

In addition to analytics, the dashboard offers quick navigation links to major features like employee management, attendance tracking, salary entry, and payslip generation. It acts as a centralized hub where authorized users can manage operations efficiently. This module improves usability by organizing all essential functions in one place and supports better reporting by visually presenting important data, making the system more user-friendly and efficient.

CHAPTER 4
CONCLUSION AND FUTURE ENHANCEMENT

1. Conclusion

The Payroll Management System (PMS) represents a significant improvement in how organizations handle employee compensation and related processes. Unlike traditional manual payroll methods that are time-consuming and error-prone, PMS offers an automated, centralized, and database-driven platform for managing salaries, deductions, bonuses, attendance, taxes, and payslip generation. It applies relational database concepts such as normalization and table relationships to ensure structured and consistent data storage. With secure user authentication and role-based access control, the system protects sensitive payroll data and limits access to authorized users only. Real-time data handling and automated salary calculations enhance accuracy and reduce administrative burden, while detailed reporting and payslip generation support transparency and accountability. Overall, PMS not only improves the efficiency and reliability of payroll operations but also lays the foundation for future enhancements like tax automation, analytics integration, and device-based attendance syncing, ultimately contributing to smarter and more streamlined HR and financial management within the organization.

2. Future Enhancement

The Payroll Management System (PMS) represents a significant improvement over traditional manual payroll methods by offering a centralized, automated, and secure platform for managing employee data, attendance, salary calculations, and payslip generation. By utilizing relational databases, normalization, and role-based access control, the system ensures data accuracy, integrity, and confidentiality. PMS streamlines payroll processing, reduces administrative workload, and minimizes human error, enabling organizations to operate more efficiently. It lays the groundwork for future enhancements like tax automation, real-time reports, and integration with financial systems, making it a smart and scalable solution for modern workforce management.

With faster data access and improved reporting, the system supports better decision-making for HR and finance teams. It enhances transparency, maintains consistent salary records, and ensures timely salary disbursement. Overall, PMS is a reliable step toward digital transformation in payroll, offering long-term benefits in accuracy, compliance, and operational efficienc

APPENDIX A
SOURCE CODE

1. db.php

```
<?php
$servername = "localhost";  // usually 'localhost'
$username = "root";          // your MySQL username
$password = "";              // your MySQL password (default is often empty for localhost)
$dbname = "payroll_system"; // the database name
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Optional: uncomment to test connection
// echo "Connected successfully";
?>
```

2. register.php

```
<?php
session_start();
require 'db.php';

$message = "";
if (isset($_POST['register'])) {
    $username = $_POST['username'];
    $email    = $_POST['email'];
    $password = $_POST['password'];
    $role     = "admin"; // Fixed as admin
    if (empty($username) || empty($email) || empty($password)) {
        $message = " + All fields are required.";
```

```
    } else {
        $checkQuery = "SELECT * FROM users WHERE username = ?";
        $stmt = $conn->prepare($checkQuery);
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if ($result->num_rows > 0) {
            $message = "⬛ Username already registered. Please <a href='login.php'>login here</a>.";
        } else {
            $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
            $insertQuery = "INSERT INTO users (username, email, password, role) VALUES (?, ?, ?, ?)";
            $stmt = $conn->prepare($insertQuery);
            $stmt->bind_param("ssss", $username, $email, $hashedPassword, $role);
            if ($stmt->execute()) {
                $message = "      Registration successful! Please <a href='login.php'>login</a>.";
            } else {
                $message = " +  Something went wrong. Please try again.";
            }
        }
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta  charset="UTF-8">
    <title>Register - Payroll System</title>
    <style>
```

```
* {
    box-sizing: border-box;
}
body {
    font-family: Arial, sans-serif;
    background-color: #f5f5f5;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;

}
.register-container {
    background-color: white;
    padding: 25px 30px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    width: 100%;
    max-width: 400px;
}
h2 {
    text-align: center;
}
input, button {
    display: block;
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    font-size: 16px;
    border-radius: 6px;
```

```
    <h2>Register as Admin</h2>
    <?php if (!empty($message)) echo "<p class='message'>$message</p>"; ?>
    <form method="POST" action="register.php">
        <input type="text" name="username" placeholder="Username" required>
        <input type="email" name="email" placeholder="Email" required>
        <input type="password" name="password" placeholder="Password" required>
        <button type="submit" name="register">Register</button>
    </form>
</div>
</body>
</html>
```

3 . login.php

```
<?php
session_start();
require 'db.php';

$message = "";
if (isset($_POST['login'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    if (empty($username) || empty($password)) {
        $message = " + Both fields are required.";
    } else {
        $query = "SELECT * FROM users WHERE username = ?";
        $stmt = $conn->prepare($query);
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $result = $stmt->get_result();
        if ($result->num_rows > 0) {
            $user = $result->fetch_assoc();
```

```
            if (password_verify($password, $user['password'])) {
                $_SESSION['user_id'] = $user['user_id'];
                $_SESSION['username'] = $user['username'];
                $_SESSION['role'] = $user['role'];
                header('Location: dashboard.php'); // Adjust if roles have different dashboards
                exit();
            } else {
                $message = " + Incorrect password.";
            }
        } else {
            $message = " + No user found with that username.";
        }
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login - Payroll System</title>
    <style>
        * {
            box-sizing: border-box;
        }
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
            padding: 0;
            margin: 0;
```

```
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
.login-container {
    background-color: white;
    padding: 25px 30px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    width: 100%;
    max-width: 400px;
}
h2 {
    text-align: center;
}
input, button {
    display: block;
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    font-size: 16px;
    border-radius: 6px;
}
input {
    border: 1px solid #ccc;
}
button {
    background-color: #007bff;
```

```
            border: none;
            color: white;
            cursor: pointer;
        }
        button:hover {
            background-color:  #0056b3;
        }
        .message {
            text-align: center;
            margin: 10px 0;
            color: red;
        }
    </style>
</head>
<body>
<div class="login-container">
    <h2>Login to Payroll System</h2>
    <?php if (!empty($message)) echo "<p class='message'>$message</p>"; ?>
    <form method="POST" action="login.php">
        <input type="text" name="username" placeholder="Username" required>
        <input type="password" name="password" placeholder="Password" required>
        <button type="submit" name="login">Login</button>
    </form>
</div>
</body>
</html>
```

4. add_employee.php

```
<?php
include("db.php");
```

```
$msg = "";
$editMode = false;
// Default empty values
$emp_id = $username = $name = $email = $phone = $department = $designation = $join_date = $basic_salary = "";
// Check for edit mode
if (isset($_GET['id'])) {
    $editMode = true;
    $edit_id = $_GET['id'];
    // Fetch employee details
    $stmt = $conn->prepare("SELECT * FROM employees WHERE emp_id = ?");
    $stmt->bind_param("i", $edit_id);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows === 1) {
        $row = $result->fetch_assoc();
        $emp_id = $row['emp_id'];
        $username = $row['username'];
        $name = $row['name'];
        $email = $row['email'];
        $phone = $row['phone'];
        $department = $row['department'];
        $designation = $row['designation'];
        $join_date = $row['join_date'];
        $basic_salary = $row['basic_salary'];
    } else {
        $msg = "Employee not found.";
    }
}
```

```
// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $emp_id = $_POST['emp_id'];
    $username = $_POST['username'];
    $name = $_POST['name'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $department = $_POST['department'];
    $designation = $_POST['designation'];
    $join_date = $_POST['join_date'];
    $basic_salary = $_POST['basic_salary'];
    if ($editMode) {
        // Update existing employee
        $stmt = $conn->prepare("UPDATE employees SET username=?, name=?, email=?, phone=?, department=?, designation=?, join_date=?, basic_salary=? WHERE emp_id=?");
        $stmt->bind_param("ssssssdi", $username, $name, $email, $phone, $department, $designation, $join_date, $basic_salary, $emp_id);
        if ($stmt->execute()) {
            $msg = "Employee updated successfully!";
        } else {
            $msg = "Error: " . $stmt->error;
        }
    } else {
        // Insert new employee
        $stmt = $conn->prepare("INSERT INTO employees (emp_id, username, name, email, phone, department, designation, join_date, basic_salary) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");
        $stmt->bind_param("issssssd", $emp_id, $username, $name, $email, $phone, $department, $designation, $join_date, $basic_salary);
        if ($stmt->execute()) {
```

```
        h2 {
            text-align: center;
        }
                input[type="text"],     input[type="email"],     input[type="number"],     input[type="date"],
input[type="tel"] {
            width: 100%;
            padding: 10px;
            margin-top: 10px;
            margin-bottom: 20px;
            border: 1px solid #ccc;
            border-radius: 6px;
        }
        input[type="submit"] {
            background: #007bff;
            color: white;
            padding: 10px 15px;
            border: none;
            border-radius: 6px;
            cursor: pointer;
            width: 100%;
        }
        .msg {
            text-align: center;
            color: green;
            font-weight: bold;
        }
    </style>
</head>
<body>
```

```
<div class="form-container">
    <h2><?= $editMode ? "Edit Employee" : "Add New Employee" ?></h2>
    <?php if ($msg): ?>
        <p class="msg"><?= $msg ?></p>
    <?php endif; ?>
    <form method="POST" action="">
        <label>Employee ID:</label>
        <input type="number" name="emp_id" value="<?= htmlspecialchars($emp_id) ?>" required <?= $editMode ? 'readonly' : '' ?>>
        <label>Username:</label>
        <input type="text" name="username" value="<?= htmlspecialchars($username) ?>" required>
        <label>Full Name:</label>
        <input type="text" name="name" value="<?= htmlspecialchars($name) ?>" required>
        <label>Email:</label>
        <input type="email" name="email" value="<?= htmlspecialchars($email) ?>" required>
        <label>Phone:</label>
        <input type="tel" name="phone" value="<?= htmlspecialchars($phone) ?>" required>
        <label>Department:</label>
        <input type="text" name="department" value="<?= htmlspecialchars($department) ?>" required>
        <label>Designation:</label>
        <input type="text" name="designation" value="<?= htmlspecialchars($designation) ?>" required>
        <label>Joining Date:</label>
        <input type="date" name="join_date" value="<?= htmlspecialchars($join_date) ?>" required>
        <label>Basic Salary:</label>
                    <input type="number" step="0.01" name="basic_salary" value="<?= htmlspecialchars($basic_salary) ?>" required>
        <input type="submit" value="<?= $editMode ? "Update Employee" : "Add Employee" ?>">
    </form>
</div>
```

```
</body>
</html>
```

5. view_employees.php

```
<?php
// Include database connection
require_once 'db.php';
// Fetch employee data
$sql = "SELECT * FROM employees";
$result = $conn->query($sql);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>View Employees</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        .container {
            width: 90%;
            margin: 20px auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
```

```
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
h2 {
    text-align: center;
    margin-bottom: 20px;
}
.employee-table {
    width: 100%;
    border-collapse: collapse;
    margin: 10px 0;
}
.employee-table th, .employee-table td {
    padding: 10px;
    text-align: left;
    border: 1px solid #ddd;
}
.employee-table th {
    background-color: #4CAF50;
    color: white;
}
.employee-table tr:nth-child(even) {
    background-color: #f2f2f2;
}
.employee-table tr:hover {
    background-color: #ddd;
}
.employee-table a {
    color: #4CAF50;
    text-decoration: none;
```

```
        }
        .employee-table a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Employee Details</h2>
        <table  class="employee-table">
            <thead>
                <tr>
                    <th>Employee ID</th>
                    <th>Username</th>
                    <th>Name</th>
                    <th>Email</th>
                    <th>Phone</th>
                    <th>Department</th>
                    <th>Designation</th>
                    <th>Join Date</th>
                    <th>Basic Salary</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                <?php
                if ($result->num_rows > 0) {
                    // Loop through each row and display employee details
                    while ($employee = $result->fetch_assoc()) {
```

```
                echo "<tr>";
                echo  "<td>" . $employee['emp_id'] . "</td>";
                echo "<td>" . $employee['username'] . "</td>";
                echo "<td>" . $employee['name'] . "</td>";
                echo "<td>" . $employee['email'] . "</td>";
                echo "<td>" . $employee['phone'] . "</td>";
                echo "<td>" . $employee['department'] . "</td>";
                echo "<td>" . $employee['designation'] . "</td>";
                echo "<td>" . $employee['join_date'] . "</td>";
                echo "<td>" . $employee['basic_salary'] . "</td>";
                echo "<td>

                    <a href='edit_employee.php?emp_id=" . $employee['emp_id'] . "'>Edit</a> |
                    <a href='delete_employee.php?emp_id=" . $employee['emp_id'] . "' onclick='return
confirm(\"Are you sure you want to delete this employee?\");'>Delete</a>
                </td>";
                echo "</tr>";
            }
        } else {
            echo "<tr><td colspan='10'>No employees found</td></tr>";
        }
        ?>
      </tbody>
    </table>
  </div>
 </body>
</html>
```

6. edit_employee.php

```
<?php
// Ensure the employee ID is passed and is a valid number
```

```
if (isset($_GET['id']) && is_numeric($_GET['id'])) {
    $emp_id = $_GET['id'];
    // Now redirect to add_employee.php with the correct employee ID
    header("Location: add_employee.php?id=$emp_id");
    exit(); // Always call exit() after a header redirect to prevent further code execution
} else {
    // If the ID is not passed or is invalid, show an error message
    echo " + Invalid employee ID.";
}
```

7. add_attendance.php

```
<?php
session_start();
if (!isset($_SESSION['username'])) {
    header("Location: login.php");
    exit();
}
include("db.php"); // Make sure this file connects to your database
// Fetch all employees for dropdown
$query = "SELECT emp_id, name FROM employees";
$result = $conn->query($query);
// Handle form submission
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $emp_id = $_POST['emp_id'];
    $date = $_POST['date'];
    $status = $_POST['status'];
    // Insert attendance data
    $stmt = $conn->prepare("INSERT INTO attendance (emp_id, date, status) VALUES (?, ?, ?)");
    $stmt->bind_param("iss", $emp_id, $date, $status);
    if ($stmt->execute()) {
```

```
        echo "Attendance added successfully.";
    } else {
        echo "Error adding attendance: " . $stmt->error;
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Add Attendance - Payroll System</title>
    <style>
        body {
            font-family: Arial;
            background: #f5f5f5;
            padding: 20px;
        }
        .form-container {
            max-width: 600px;
            margin: auto;
            background: white;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 0 10px #ccc;
        }
        h1 {
            text-align: center;
        }
        label {
            font-weight: bold;
            margin-top: 10px;
```

```
        }
        select, input[type="date"], input[type="submit"] {
            width: 100%;
            padding: 8px;
            margin-top: 5px;
            margin-bottom: 15px;
            border-radius: 5px;
            border: 1px solid #ccc;
        }
        input[type="submit"] {
            background-color: #007bff;
            color: white;
            border: none;
            cursor: pointer;
        }
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
<div class="form-container">
    <h1>Add Attendance</h1>
    <form action="add_attendance.php" method="POST">
        <label for="emp_id">Select Employee</label>
        <select name="emp_id" required>
            <option value="">Select Employee</option>
            <?php while ($row = $result->fetch_assoc()) { ?>
```

```
            <option value="<?php echo $row['emp_id']; ?>"><?php echo $row['name']; ?></option>
        <?php } ?>
    </select>
    <label for="date">Date</label>
    <input type="date" name="date" required>
    <label for="status">Attendance Status</label>
    <select name="status" required>
        <option value="Present">Present</option>
        <option value="Absent">Absent</option>
        <option value="Leave">Leave</option>
    </select>
    <input type="submit" value="Add Attendance">
  </form>
</div>
</body>
</html>
```

8. view_attendance.php

```
<?php
include 'db.php';
?>
<!DOCTYPE html>
<html>
<head>
  <title>View Attendance</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: #f0f2f5;
```

```
        }
        tr:hover {
            background-color:  #f1f1f1;
        }
        .status-present { color: green; font-weight: bold; }
        .status-absent { color: red; font-weight: bold; }
        .status-leave { color: orange; font-weight: bold; }
    </style>
</head>
<body>
<div class="container">
    <h2>Employee Attendance Records</h2>
    <table>
        <thead>
            <tr>
                <th>Employee Name</th>
                <th>Date</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>
        <?php
        $sql = "SELECT a.date, a.status, e.name
                FROM attendance a
                JOIN employees e ON a.emp_id = e.emp_id
                ORDER BY a.date DESC";
        $result = $conn->query($sql);
        if ($result->num_rows > 0) {
```

```
        while ($row = $result->fetch_assoc()) {
            $statusClass = strtolower($row['status']);
            echo "<tr>
                    <td>{$row['name']}</td>
                    <td>{$row['date']}</td>
                    <td  class='status-{$statusClass}'>{$row['status']}</td>
                  </tr>";
        }
    } else {
        echo "<tr><td colspan='3'>No attendance records found.</td></tr>";
    }
    ?>
    </tbody>
  </table>
</div>
</body>
</html>
```

9. add_salary.php

```
<?php
include 'db.php';
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $emp_id = $_POST['emp_id'];
  $month = $_POST['month'];
  $year = $_POST['year'];
  $basic_salary = $_POST['basic_salary'];
  $allowances = $_POST['allowances'];
  $deductions = $_POST['deductions'];
  $net_salary = $basic_salary + $allowances - $deductions;
```

```
    $stmt = $conn->prepare("INSERT INTO salary (emp_id, month, year, basic_salary, allowances, deductions, net_salary) VALUES (?, ?, ?, ?, ?, ?, ?)");

    $stmt->bind_param("issdddd", $emp_id, $month, $year, $basic_salary, $allowances, $deductions, $net_salary);
    if ($stmt->execute()) {
        echo "<script>alert('Salary added successfully'); window.location.href='dashboard.php';</script>";
    } else {
        echo "Error: " . $stmt->error;
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Add Salary</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #f4f6f8;
            margin: 0;
            padding: 0;
        }
        .container {
            max-width: 550px;
            margin: 60px auto;
            background: #ffffff;
            padding: 30px 40px;
            border-radius: 10px;
```

```
    box-shadow: 0px 0px 15px rgba(0,0,0,0.1);
}
h2 {
    text-align: center;
    margin-bottom: 30px;
    color: #333333;
}
label {
    font-weight: bold;
    display: block;
    margin-top: 15px;
    color: #555;
}
input[type="text"], input[type="number"], select {
    width: 100%;
    padding: 10px;
    margin-top: 6px;
    border: 1px solid #ccc;
    border-radius: 6px;
    box-sizing: border-box;
}
input[type="submit"] {
    width: 100%;
    background-color: #007BFF;
    color: white;
    padding: 12px;
   margin-top: 25px;
   border: none;
```

```
            border-radius: 6px;
            cursor: pointer;
            font-size: 16px;
        }
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
<div class="container">
    <h2>Add Employee Salary</h2>
    <form method="post" action="">
        <label for="emp_id">Select Employee:</label>
        <select name="emp_id" required>
            <option value="">-- Select Employee --</option>
            <?php
            $result = $conn->query("SELECT emp_id, name FROM employees");
            while ($row = $result->fetch_assoc()) {
                echo "<option value='{$row['emp_id']}'>{$row['name']} (ID: {$row['emp_id']})</option>";
            }
            ?>
        </select>
        <label for="month">Month:</label>
        <input type="text" name="month" placeholder="e.g. January" required>
        <label for="year">Year:</label>
        <input type="number" name="year" placeholder="e.g. 2025" required>
        <label for="basic_salary">Basic Salary:</label>
        <input type="number" step="0.01" name="basic_salary" required>
```

```
        <label for="allowances">Allowances:</label>
        <input type="number" step="0.01" name="allowances" required>
        <label for="deductions">Deductions:</label>
        <input type="number" step="0.01" name="deductions" required>
        <input type="submit" value="Add Salary">
    </form>
</div>
</body>
</html>
```

10. view_salaries.php

```
<?php
// view_salaries.php
include 'db.php'; // Make sure this file connects to your DB
?>
<!DOCTYPE html>
<html>
<head>
    <title>View Salaries</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color:  #f4f6f8;
            padding: 20px;
        }
        h2 {
            color: #333;
        }
        table {
```

```
            width: 100%;
            border-collapse: collapse;
            background-color:  #fff;
            margin-top:  20px;
        }
        th, td {
            padding: 12px;
            text-align: center;
            border-bottom: 1px solid #ddd;
        }
        th {
            background-color:  #007BFF;
            color: white;
        }
        tr:hover {
            background-color:  #f1f1f1;
        }
    </style>
</head>
<body>
<h2>Salary Records</h2>
<table>
    <tr>
        <th>Employee Name</th>
        <th>Month</th>
        <th>Year</th>
        <th>Basic Salary</th>
        <th>Allowances</th>
        <th>Deductions</th>
```

11. generate_payslip.php

```
<?php
include("db.php");

$msg = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $emp_id = $_POST['emp_id'];
    // Fetch employee details
    $stmt = $conn->prepare("SELECT * FROM employees WHERE emp_id = ?");
    $stmt->bind_param("i", $emp_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $employee = $result->fetch_assoc();
    // Fetch salary details for the employee
    $salary_stmt = $conn->prepare("SELECT * FROM salary WHERE emp_id = ?");
    $salary_stmt->bind_param("i", $emp_id);
    $salary_stmt->execute();
    $salary_result = $salary_stmt->get_result();
    $salary = $salary_result->fetch_assoc();
    if (!$employee || !$salary) {
        $msg = "No employee or salary record found!";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Generate Payslip</title>
```

```
        .payslip-container table, .payslip-container th, .payslip-container td {
            border: 1px solid #ccc;
            padding: 10px;
            text-align: left;
        }
        .payslip-container th {
            background-color: #007bff;
            color: white;
        }
        .print-btn {
            background: #28a745;
            color: white;
            padding: 10px 15px;
            border: none;
            border-radius: 6px;
            cursor: pointer;
            display: block;
            margin: auto;
            margin-top: 20px;
            width: 100%;

        }
    </style>
</head>
<body>
<div class="form-container">
    <h2>Generate Payslip</h2>
    <?php if ($msg): ?>
        <p class="msg"><?php echo $msg; ?></p>
    <?php endif; ?>
```

```
    <form method="POST" action="generate_payslip.php">
        <label>Employee ID:</label>
        <input type="number" name="emp_id" required>

        <input type="submit" value="Generate Payslip">
    </form>
</div>
<?php if(isset($employee) && isset($salary)): ?>
    <div class="payslip-container">
        <h3>Payslip for <?php echo htmlspecialchars($employee['name']); ?></h3>
        <table>
            <tr>
                <th>Employee ID</th>
                <td><?php echo $employee['emp_id']; ?></td>
            </tr>
            <tr>
                <th>Name</th>
                <td><?php echo $employee['name']; ?></td>
            </tr>
            <tr>
                <th>Designation</th>
                <td><?php echo $employee['designation']; ?></td>
            </tr>
            <tr>
                <th>Month</th>
                <td><?php echo $salary['month']; ?></td>
            </tr>
            <tr>
```

```
            <th>Year</th>
            <td><?php echo $salary['year']; ?></td>
          </tr>
          <tr>
            <th>Basic Salary</th>
            <td><?php echo number_format($salary['basic_salary'], 2); ?></td>
          </tr>
          <tr>
            <th>Allowances</th>
            <td><?php echo number_format($salary['allowances'], 2); ?></td>
          </tr>
          <tr>
            <th>Deductions</th>
            <td><?php echo number_format($salary['deductions'], 2); ?></td>
          </tr>
          <tr>
            <th>Net Salary</th>
            <td><?php echo number_format($salary['net_salary'], 2); ?></td>
          </tr>
        </table>
        <button class="print-btn" onclick="window.print()">Print Payslip</button>
    </div>
<?php endif; ?>
</body>
</html>
```

12. dashboard.php

```
<?php
session_start();
```

```
require 'db.php';
// Ensure only admin can access
if (!isset($_SESSION['username']) || $_SESSION['role'] !== 'admin') {
    header("Location: login.php");
    exit();
}
$username = $_SESSION['username'];
// Fetch quick stats (optional)
$employeeCount = $conn->query("SELECT COUNT(*) AS total FROM employees")->fetch_assoc()['total'];
$attendanceCount = $conn->query("SELECT COUNT(*) AS total FROM attendance")->fetch_assoc()['total'];
$salaryCount = $conn->query("SELECT COUNT(*) AS total FROM salary")->fetch_assoc()['total'];
?>
<!DOCTYPE html>
<html>
<head>
    <title>Admin Dashboard - Payroll System</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #f4f4f4;
            margin: 0;
            padding: 0;
        }
        .header {
            background: #2c3e50;
            color: white;
            padding: 20px;
```

```
    text-align: center;
}
.container {
    padding: 30px;
}
.card {
    background: white;
    border-radius: 8px;
    padding: 20px;
    margin: 20px 0;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
.nav {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
}
.nav a {
    flex: 1 1 200px;
    padding: 15px;
    text-align: center;
    background: #3498db;
    color: white;
    border-radius: 6px;
    text-decoration: none;
    font-weight: bold;
    transition: background 0.3s ease;
}
```

```
<body>
    <div class="header">
        <h1>Admin Dashboard - Payroll System</h1>
        <div class="logout">
            Logged in as: <strong><?php echo htmlspecialchars($username); ?></strong> |
            <a href="logout.php" style="color: #f1c40f;">Logout</a>
        </div>
    </div>
    <div class="container">
        <div class="stats">
            <div class="stat">
                <h2><?php echo $employeeCount; ?></h2>
                <p>Total Employees</p>
            </div>
            <div class="stat">
                <h2><?php echo $attendanceCount; ?></h2>
                <p>Total Attendance Records</p>
            </div>
            <div class="stat">
                <h2><?php echo $salaryCount; ?></h2>
                <p>Salary Processed</p>
            </div>
        </div>
        <div class="card">
            <h2>Admin Actions</h2>
            <div class="nav">
                <a href="add_employee.php">+ Add Employee</a>
                <a href="view_employees.php">+    View Employees</a>
```

```
            <a href="add_attendance.php">Add Attendance</a>
            <a href="view_attendance.php">View Attendance</a>
            <a href="add_salary.php">Add Salary</a>
            <a href="view_salaries.php">View Salaries</a>
            <a href="generate_payslip.php">Generate Payslips</a>
        </div>
    </div>
</div>
</body>
</html>
```

13. dashboard_admin.php

```
<?php
// Start the session
session_start();
// Check if the user is logged in as admin
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'admin') {
    header('Location: login.php');
    exit();
}
// Include the database connection
require 'db.php';
// Fetch the admin's username (or any other details you need)
$username = $_SESSION['username'];
// Query to get employee details
$query = "SELECT * FROM employees";
$result = $conn->query($query);
?>
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Dashboard - Payroll System</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
            padding: 20px;
        }
        .navbar {
            background-color: #007bff;
            color: white;
            padding: 10px;
            text-align: center;
        }
        .navbar a {
            color: white;
            text-decoration: none;
            margin: 0 15px;
        }

        .container {
            padding: 20px;
        }
        .dashboard {
            display: flex;
            flex-wrap: wrap;
            gap: 20px;
```

```
    <a href="add_salary.php">Add Salaries</a>
    <a href="logout.php">Logout</a>
</div>
<!-- Admin Dashboard -->
<div class="container">
    <h2>Welcome, Admin: <?php echo htmlspecialchars($username); ?></h2>
    <div class="dashboard">
        <!-- Employee List Section -->
        <div  class="dashboard-item">
          <h3>Employee List</h3>
          <ul  class="employee-list">
            <?php
            // Check if there are any employees in the database
            if ($result->num_rows > 0) {
                // Display each employee's details
                while ($employee = $result->fetch_assoc()) {
                    echo '<li>';
                    echo 'Name: ' . htmlspecialchars($employee['name']) . '<br>';
                    echo 'Email: ' . htmlspecialchars($employee['email']) . '<br>';
                    echo 'Department: ' . htmlspecialchars($employee['department']) . '<br>';
                    echo 'Designation: ' . htmlspecialchars($employee['designation']) . '<br>';
                    echo '<a href="edit_employee.php?emp_id=' . $employee['emp_id'] . '">Edit</a> | ';
                    echo '<a href="delete_employee.php?emp_id=' . $employee['emp_id'] . '">Delete</a>';
                    echo '</li>';
                }
            } else {
                echo '<li>No employees found.</li>';
            }
            ?>
```

```
            </ul>
        </div>
        <!-- Attendance Section -->
        <div class="dashboard-item">
            <h3>Add Attendance</h3>
            <a href="add_attendance.php">Add Attendance</a>
        </div>
        <!-- Salary Section -->
        <div class="dashboard-item">
            <h3>Add Salaries</h3>
            <a href="add_salary.php">Add Salaries</a>
        </div>
    </div>
</div>
</body>
</html>
```

14. style.css

```
body {
    font-family: Arial, sans-serif;
    background: #f2f2f2;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
.form-container {
    background: #fff;
    padding: 30px;
    border-radius: 10px;
```

```
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
    width: 300px;
}
h2 {
    text-align: center;
    margin-bottom: 20px;
}
input[type="text"], input[type="password"] {
    width: 100%;
    padding: 10px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 6px;
}
button {
    width: 100%;
    padding: 10px;
    background: #007bff;
    color: white;
    border: none;
    border-radius: 6px;
    cursor: pointer;
}
button:hover {
    background: #0056b3;
}
p {
    text-align: center;
    margin-top: 15px;
```

```
}
```

15. logout.php

```
<?php
session_start(); // Start or resume the session
session_unset(); // Unset all session variables
session_destroy(); // Destroy the session
// Redirect to register page
header("Location: register.php");
exit();
```

APPENDIX B
SCREENSHOTS

1. Register

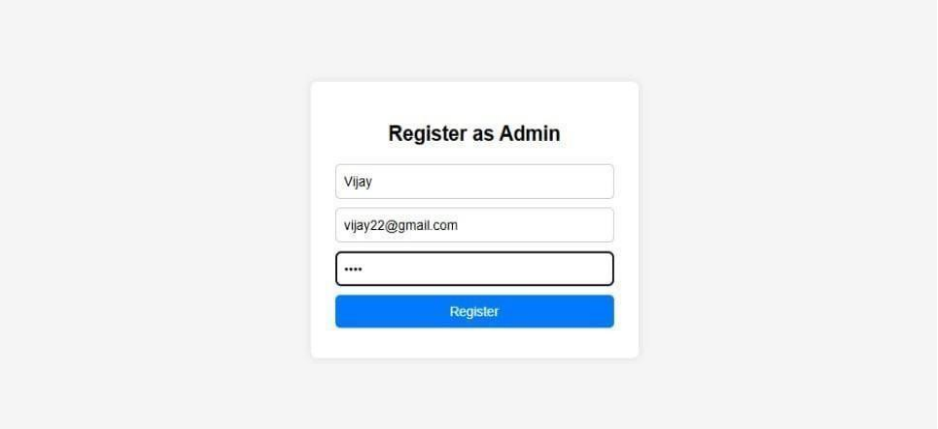


Fig 5.1

2. Login

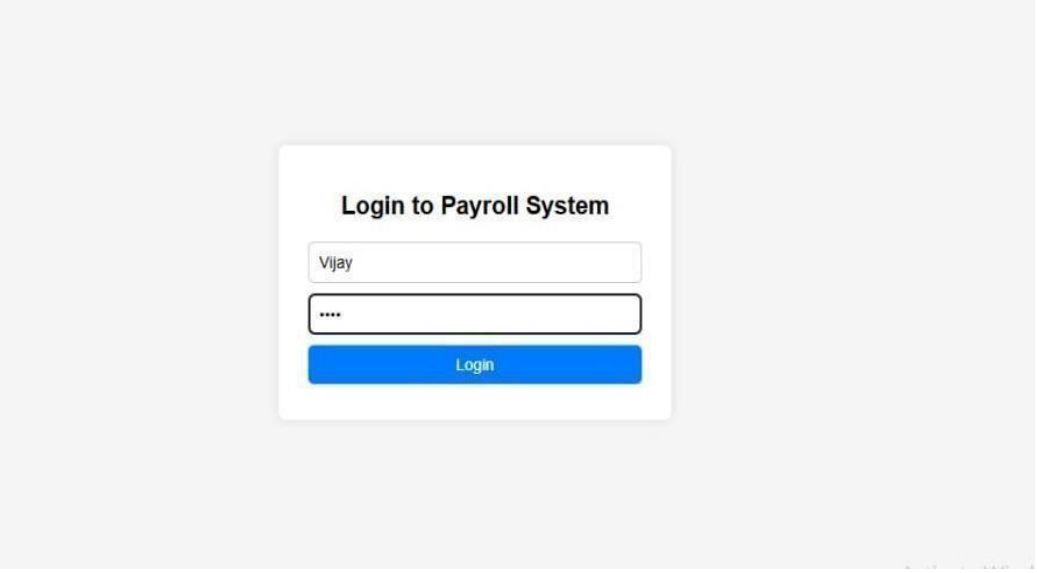


Fig 5.2

3. Dashboard

Fig 5.3

4. Add Employee

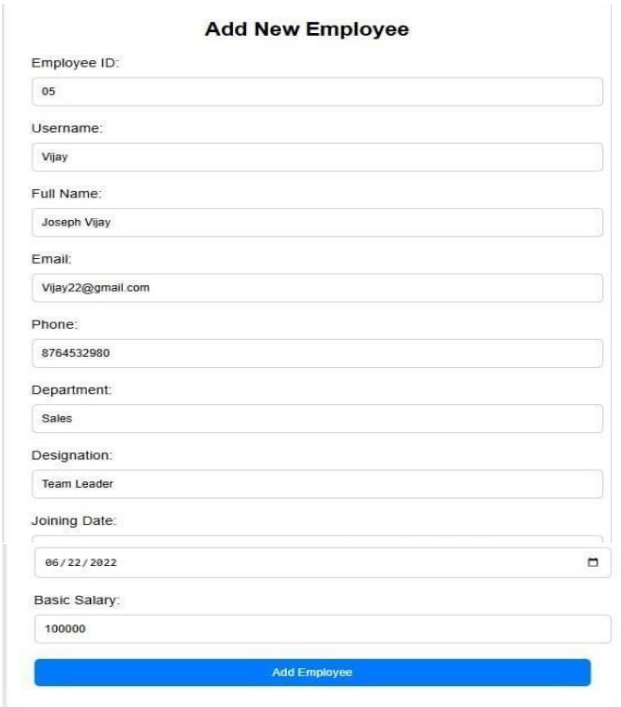


Fig 5.4

5. View Attendance

Employee Details

Employee ID	Username	Name	Email	Phone	Department	Designation	Join Date	Basic Salary	Actions
1		Yogeshwaran V B				Manager			Edit | Delete
2		Bhawna Sri				Executive			Edit | Delete
3	Shri	Shri Y	shri@gmail.com	1234567890	Sales	Executive	2024-01-10	45000.00	Edit | Delete
4	afiya	Afiya	afi123@gmail.com	7890776543	quality analyst	manager	2024-09-12	80000.00	Edit | Delete
5	Vijay	Joseph Vijay	Vijay22@gmail.com	8764532980	Sales	Team Leader	2022-06-22	100000.00	Edit | Delete

Fig 5.5

6. Add Attendance

Fig 5.6

7. View Attendance

Employee Attendance Records

Employee Name	Date	Status
Afiya	2025-05-11	Present
Joseph Vijay	2025-05-11	Present
Bhawna Sri	2025-05-10	Present
Yogeshwaran V B	2025-05-10	Leave
Shri Y	2025-05-09	Present

Fig 5.7

8. Add Employee Salary

Fig 5.8

9. View Employee Salary

Salary Records

Employee Name	Month	Year	Basic Salary	Allowances	Deductions	Net Salary
Yogeshwaran V B	May	2025	40000.00	5000.00	1000.00	44000.00
Shri Y	January	2025	45000.00	10000.00	800.00	54200.00
Bhawna Sri	April	2025	40000.00	10000.00	300.00	49700.00
Afiya	April	2025	80000.00	10000.00	0.00	90000.00
Joseph Vijay	April	2025	100000.00	10000.00	5000.00	105000.00

Fig 5.9

10. Generate Payslip

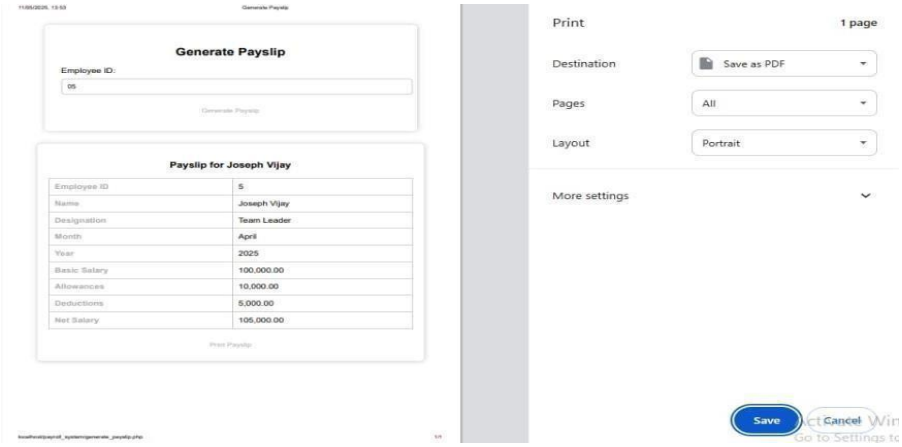


Fig 5.10

Fig 5.11

REFERENCES:

YOUTUBE:

- ✓ https://www.youtube.com/watch?v=FJ0OYXCK6vQ:contentReference[oaicite:3]{index=3}
- ✓ https://www.youtube.com/watch?v=ygrKPaJ-2EQ:contentReference[oaicite:4]{index=4}
- ✓ https://www.youtube.com/watch?v=0TEibwQZBPc:contentReference[oaicite:5]{index=5}

WEBSITE:

- ➢ https://link.springer.com/article/10.1186/s43067-023-00082-5
- ➢ https://opus.govst.edu/capstones/82/
- ➢ https://ar.iub.edu.bd/handle/11348/803

BOOKS:

- ➢ PHP & MySQL Web Development – Luke Welling.
- ➢ Learning PHP, MySQL & JavaScript – Robin Nixon
- ➢ Payroll Accounting 2023 – Bernard Bieg