VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi, Karnataka-590018

A Mini Project Report

On

"Payroll Management System"

Submitted in partial fulfillment for the award of the Degree of

Bachelor of Engineering

In

Information Science and Engineering

Submitted by

SIRVI PRIYANKA HIRALAL and SOUMYA P

(1VI18IS101 and 1VI18IS103)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

VEMANA INSTITUTE OF TECHNOLOGY

(Affiliated to VTU and approved by AICTE)

#1, Mahayogi Vemana road, Koramangala, Bengaluru-560034

2020-2021

Karnataka Reddyjana Sangha

VEMANA INSTITUTE OF TECHNOLOGY

#1, Mahayogi Vemana road, 3rd Block, Koramangala, Bengaluru-560034

Approved by AICTE and Affiliated to VTU, Belagavi

Department of Information Science and Engineering

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **Sirvi Priyanka Hiralal and Soumya P**, bearing **1VI18IS101 and 1VI18IS103,** students of 5th semester, Information Science and Engineering have completed the DBMS Mini Project entitled **"PAYROLL MANAGEMENT SYSTEM"** in partial fulfillment for the requirement of the award of degree of **Bachelor of Engineering** in **Information Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year 2020-2021. The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the said degree.

___________________	________________	___________________
Guide	HoD	Principal
(Prof. J.Ruby Dinakar)	**(Prof. Rajanna M)**	**(Dr. Vijayasimha Reddy B.G)**

External Viva

Name of the Examiner   Signature with date

1.______________________   ___________________

2.______________________   ___________________

ACKNOWLEDGEMENT

I sincerely thank Visvesvaraya Technology University for providing a platform to do the internship work.

I express my sincere thanks to **Dr. Vijayasimha Reddy B G**, Principal, Vemana Institute of Technology, Bengaluru, for providing necessary facilities and motivation to carry out DBMS mini project successfully.

I express heartfelt gratitude and humble thanks to **Prof. Rajanna M**, HoD, ISE, Vemana Institute of Technology, for his constant encouragement, inspiration and help to carry out DBMS mini project successfully.

I thank **Prof. J.Ruby Dinakar**, Assistant Professor, the DBMS Lab mini project coordinator for her boundless cooperation and support during the entire process.

I am thankful to all the teaching and non-teaching staff members of Information Science and Engineering Department for their help and needed support rendered throughout the project work.

Sirvi Priyanka Hiralal & Soumya P
1VI18IS101 & 1VI18IS103

ABSTRACT

“**Payroll Management System**” is designed to make the existing manual system automatic with the help of computerised equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This web application can maintain and view computerised records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client. For this application we use PHP and MySQL for the backend to store data and for the frontend design we use HTML and CSS.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Introduction to the Payroll Management System

The proposed project "**Payroll Management System**" has been developed to overcome the problems faced in the practicing of manual system. This software is built to eliminate and, in some cases, reduce the hardships faced by the existing system. Moreover, this system is designed for particular need of the company to carry out its operations in a smooth and effective manner. This web application is reduced as much as possible to avoid errors while entering data. It also provides error message while entering invalid data. It is user-friendly as no formal knowledge is required to use the system. Human resource challenges are faced by every organization which has to be overcome by the organization. Every organization has different employee and payroll management needs. Therefore, this project has design exclusive Employee and payroll Management System that are adapted to the organization's Managerial Requirements.

Payroll management is very simple, flexible and user-friendly management software. That takes care all requirements relating to accounting and management of employees' payroll. Payroll stores complete records of the employee, generate play-slips and computes all allowance and deductions and generates all statutory reports. It offers very high flexibility in defining various allowances, deductions, leave rules etc. for the employees and all formula for net pay are definable and changeable for user's end. Payroll application has been designed for the purpose of the maintaining details of various allowances and deductions that need to be given to the employee of the organization. Also, it generates the salary sheet automatically of employee of organization that assists the accounts department in many ways.

Some advantages of the Payroll Management are:

- It is cost effective as the user control the web application himself and does not go for professional service.
- It saves time as it speeds up every aspect of the employee database management and payroll process with a range of automated features.
- It is secure as the employee database and the payroll process is managed by the admin in house rather than sending private information to a third party.
- Validating procedures and checks restrict user from making mistakes.
- The software is easy to use and is user friendly so no expertise is required.
- The calculations are automated so no chance of error.

1.2 Problem Statement

The problem with the existing systems is that the organization is maintaining most of their day-to-day transaction data in the registers. The employees of the organization are doing the whole work manually. All the work is done manually so different employees are appointed for the maintenance of registers containing all the information related to the organization and employees. It is very time consuming and it is also not error free in some situations. In existence manual system databases calculations are difficult to perform so report generates as very much difficult.

1.3 Objective

The primary objective of the Payroll Management System is to process payroll information in an online environment, so specific online procedures can be performed in seconds rather than the hours required by traditional batch methods of computer processing. This Payroll Management System performs the basic payroll objectives by providing a complete and accurate record of all payroll transactions affecting each hourly and salaried employee.

The expected audiences of this document are the developers and the admin of the web application. Now with the help of this system the admin has the information on his finger tips and can easily prepare a good record based on their requirements. This system will not only automate the process but save the valuable time of the admin.

1.4 Scope

The scope of the project is managing a consistent storage of data by dedicated data administrator. It provides most of the features that a Database Management System should have. Payroll Software is programmed to enable HR professionals for managing the data and information of the company's employees on the regular basis. The easy information regarding the employees is including of contact details, investment details, attendance, salary data and much more.

Chapter 2

LITERATURE SURVEY

2.1 DBMS and SQL

A **Database Management System** (DBMS) refers to the technology for creating and managing databases. Basically, DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. We can break it like this DBMS= Database +Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can define DBMS like this: DBMS is a collection of inter-related data and set of programs to store and access those data in an easy and effective manner.

The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Database system are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data. According to the principles of database systems, the data is stored in such a way that it acquires a lot less space as the redundant data (duplicate data) has been removed before storage. Along with storing the data in an optimized and systematic manner, it is also important that we retrieve the data quickly when needed. Database system ensures that data is retrieved as quickly as possible.

2.1.1 Applications of DBMS

The development of computer graphics has been driven both by the needs of the user community and by the advances in hardware and software. The applications of database are many and varied. It can be divided into four major areas:

1. Hierarchical and network system
2. Flexibility with relational database
3. Object oriented application.
4. Interchanging the data on the web for e-commerce.

2.1.2 Display information

In this particular project, we are taken HTML web page as a front end in order to display the information which are stored in the backend database called MySQL. HTML stands for Hyper Text Markup Language. HTML describes the structure of web pages using markup.HTML elements are the building blocks of HTML pages. Browser do not display the HTML tags but use them to render the content of page.

2.1.3 Design

Professions such as engineering and architecture are concerned with design. Starting with a set of specification engineers and architects seek a cost effective and esthetic solutions that satisfies the specifications. Design is an iterative process rarely in the real world is a problem specified such that there is a unique optimal solution. Thus, the designer works iteratively.

2.1.4 User Interfaces

Our interactions with computers has become dominated by a visual paradigm that includes windows, icons, menus, pointing device, such as a mouse. Although we are familiar with the syntax of MySQL, advances in MySQL have made possible other forms of advantages.

2.1.5 SQL

SQL which is an abbreviation for **Structured Query Language** is a language to request data from a database, to add, update or remove data within a database, or to manipulate the metadata of the database.

Sometimes SQL is characterized as non-procedural because procedural languages generally require the details of the operations to be specified, such as opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems. Therefore, SQL is considered to be designed at a higher conceptual level of operation than procedural languages.

2.2 Triggers

A database trigger is a procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database. For example, when a new record representing a new worker is added to the employee's tables, new records should also be created in the tables of taxes, vacations and salaries. Triggers can also be used to log historical data.

2.3 Stored procedure

A stored procedure (also termed proc, storp, sproc, StoPro, StoredProc, StoreProc, sp, or SP) is a subroutine available to applications that access a relational database management system (RDBMS). Such procedures are stored in the database data dictionary.

Uses for stored procedures include data-validation (integrated into the database) or access-control mechanisms. Furthermore, stored procedures can consolidate and centralize logic that was originally implemented in applications. To save time and memory, extensive or complex processing that requires execution of several SQL statements can be saved into stored procedures, and all applications call the procedures. One can use nested stored procedures by executing one stored procedure from within another.

Chapter 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Hardware Requirements

Processor	:	i5 Core Processor
Clock speed	:	1.19 GHz
RAM	:	8 GB

3.2 Software Requirements

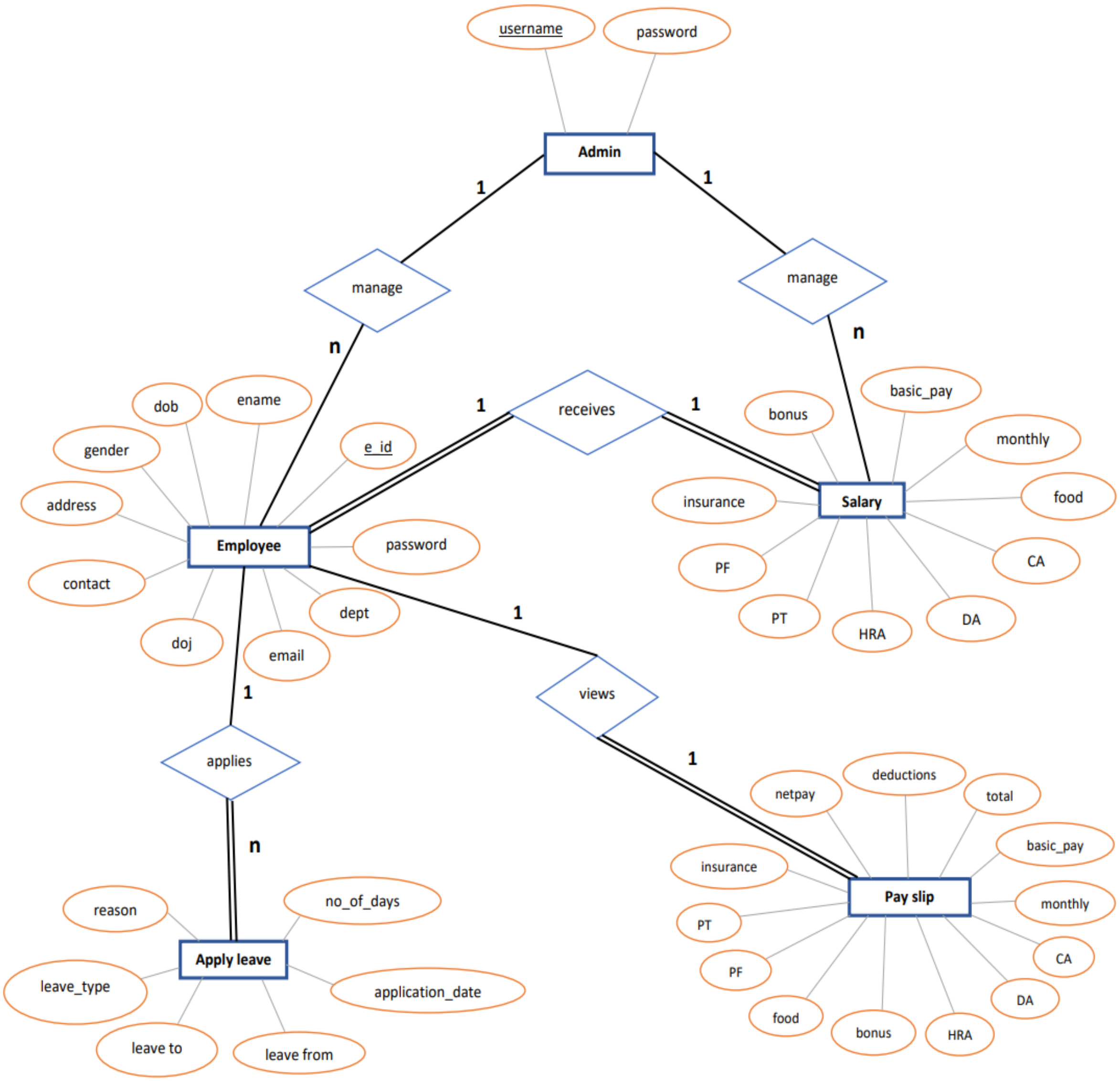
Operating System	:	64-bit Operating System, Windows 10
Database	:	MySQL
Web Server	:	XAMPP
IDE	:	Visual Studio Code
Scripting Language	:	PHP
Front End	:	HTML, CSS

Chapter 4

DESIGN

4.1 ER Diagram

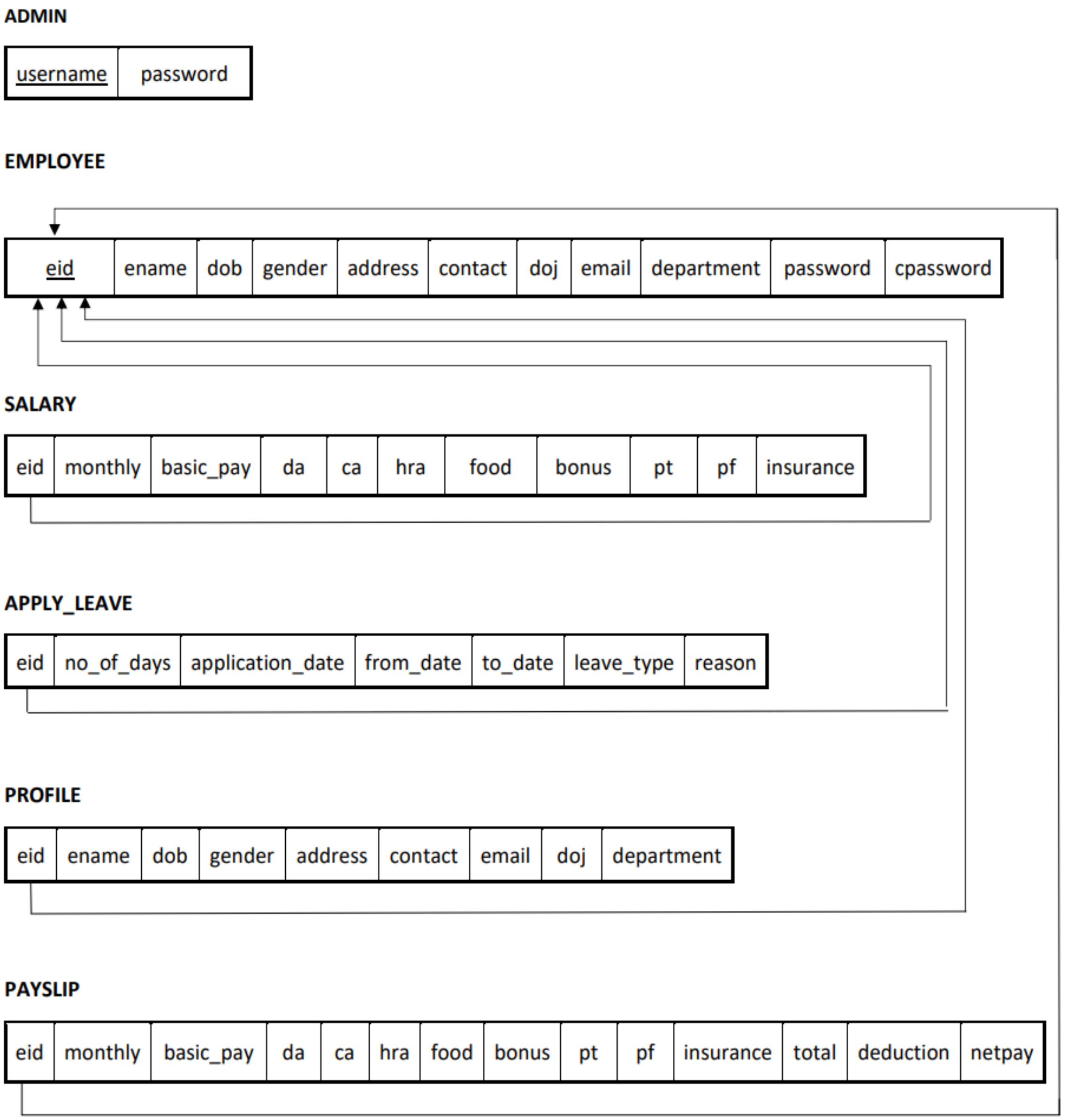
An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. In other words, ER diagrams illustrate the logical structure of databases.



4.1 ER DIAGRAM

4.2 Schema Diagram

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram shows us the database design.



ADMIN

username	password

EMPLOYEE

eid	ename	dob	gender	address	contact	doj	email	department	password	cpassword

SALARY

eid	monthly	basic_pay	da	ca	hra	food	bonus	pt	pf	insurance

APPLY_LEAVE

eid	no_of_days	application_date	from_date	to_date	leave_type	reason

PROFILE

eid	ename	dob	gender	address	contact	email	doj	department

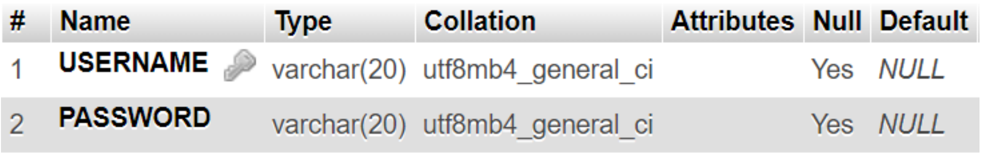
PAYSLIP

eid	monthly	basic_pay	da	ca	hra	food	bonus	pt	pf	insurance	total	deduction	netpay

4.2 SCHEMA DIAGRAM

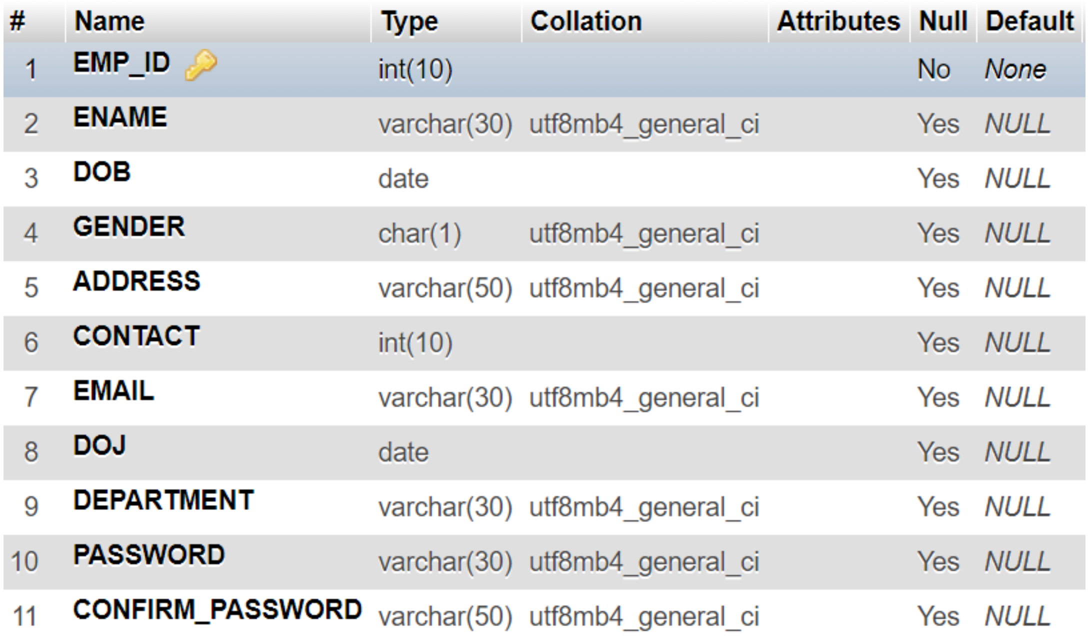
4.3 Database Design

A database design includes logical (entity relationship) and physical (table, column and key) design tools for the data.



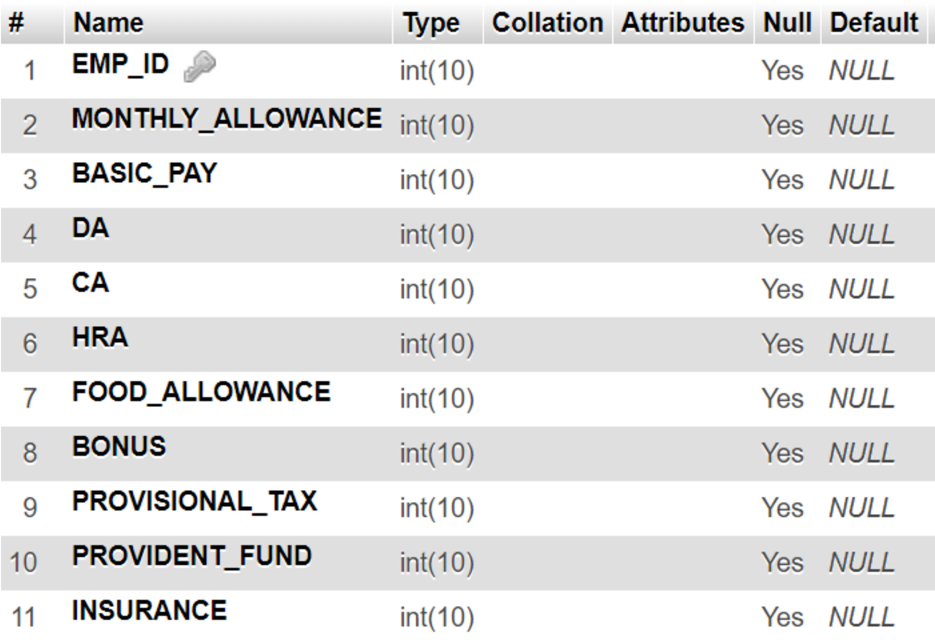
#	Name	Type	Collation	Attributes	Null	Default
1	**USERNAME** 🔑	varchar(20)	utf8mb4_general_ci		Yes	*NULL*
2	**PASSWORD**	varchar(20)	utf8mb4_general_ci		Yes	*NULL*

Table 4.3.1 Admin Table



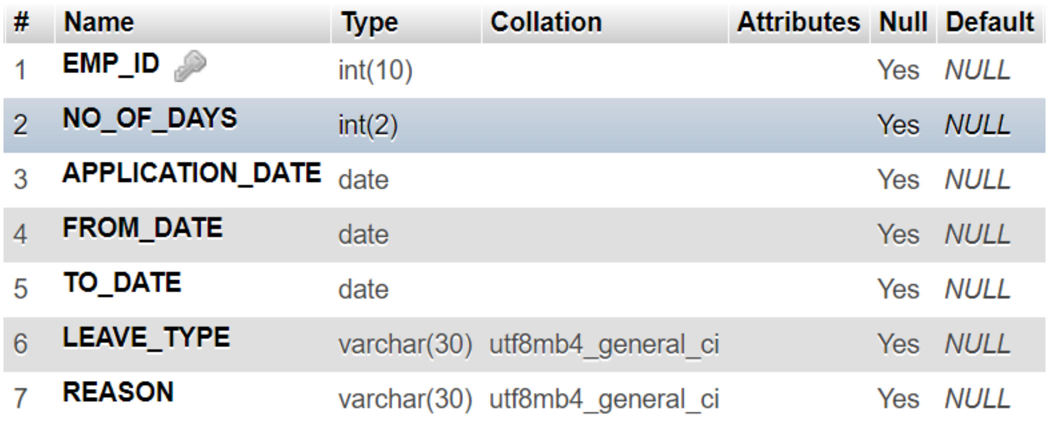
#	Name	Type	Collation	Attributes	Null	Default
1	**EMP_ID** 🔑	int(10)			No	*None*
2	**ENAME**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
3	**DOB**	date			Yes	*NULL*
4	**GENDER**	char(1)	utf8mb4_general_ci		Yes	*NULL*
5	**ADDRESS**	varchar(50)	utf8mb4_general_ci		Yes	*NULL*
6	**CONTACT**	int(10)			Yes	*NULL*
7	**EMAIL**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
8	**DOJ**	date			Yes	*NULL*
9	**DEPARTMENT**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
10	**PASSWORD**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
11	**CONFIRM_PASSWORD**	varchar(50)	utf8mb4_general_ci		Yes	*NULL*

Table 4.3.2 Employee Table



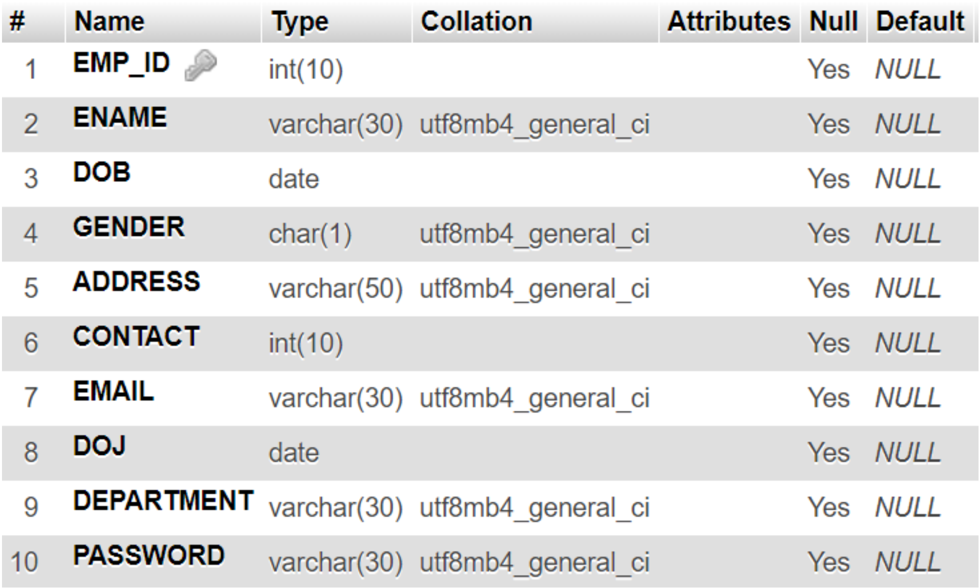
#	Name	Type	Collation	Attributes	Null	Default
1	EMP_ID	int(10)			Yes	NULL
2	MONTHLY_ALLOWANCE	int(10)			Yes	NULL
3	BASIC_PAY	int(10)			Yes	NULL
4	DA	int(10)			Yes	NULL
5	CA	int(10)			Yes	NULL
6	HRA	int(10)			Yes	NULL
7	FOOD_ALLOWANCE	int(10)			Yes	NULL
8	BONUS	int(10)			Yes	NULL
9	PROVISIONAL_TAX	int(10)			Yes	NULL
10	PROVIDENT_FUND	int(10)			Yes	NULL
11	INSURANCE	int(10)			Yes	NULL

Table 4.3.3 Salary Table



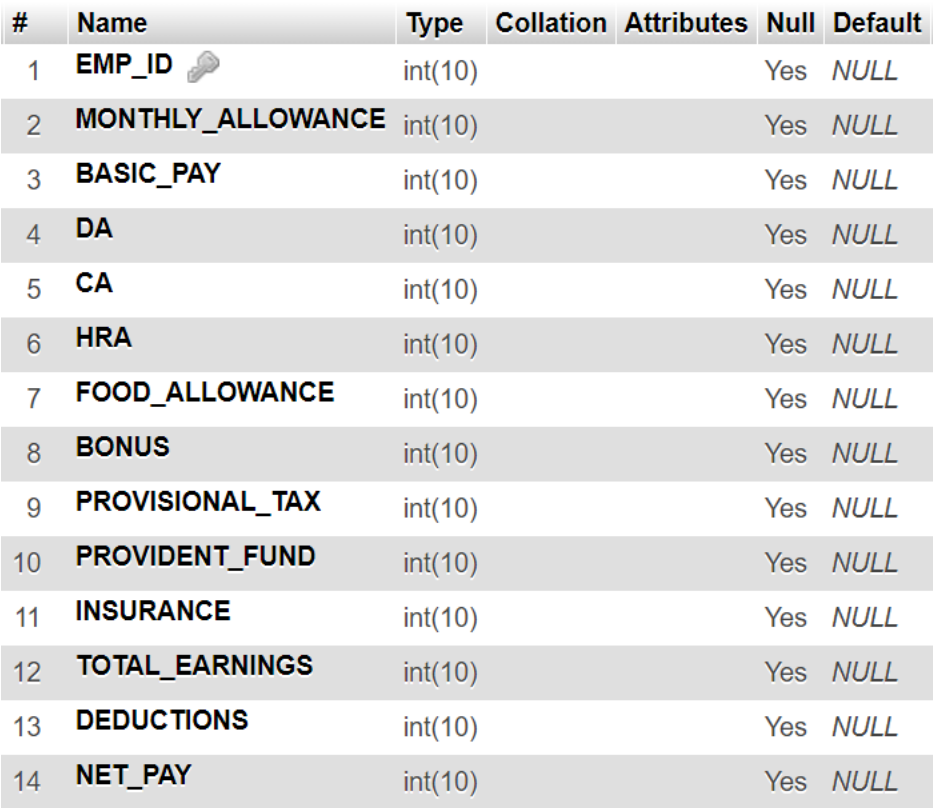
#	Name	Type	Collation	Attributes	Null	Default
1	EMP_ID	int(10)			Yes	NULL
2	NO_OF_DAYS	int(2)			Yes	NULL
3	APPLICATION_DATE	date			Yes	NULL
4	FROM_DATE	date			Yes	NULL
5	TO_DATE	date			Yes	NULL
6	LEAVE_TYPE	varchar(30)	utf8mb4_general_ci		Yes	NULL
7	REASON	varchar(30)	utf8mb4_general_ci		Yes	NULL

Table 4.3.4 Apply_Leave Table



#	Name	Type	Collation	Attributes	Null	Default
1	**EMP_ID** 🔑	int(10)			Yes	*NULL*
2	**ENAME**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
3	**DOB**	date			Yes	*NULL*
4	**GENDER**	char(1)	utf8mb4_general_ci		Yes	*NULL*
5	**ADDRESS**	varchar(50)	utf8mb4_general_ci		Yes	*NULL*
6	**CONTACT**	int(10)			Yes	*NULL*
7	**EMAIL**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
8	**DOJ**	date			Yes	*NULL*
9	**DEPARTMENT**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*
10	**PASSWORD**	varchar(30)	utf8mb4_general_ci		Yes	*NULL*

Table 4.3.5 Profile Table



#	Name	Type	Collation	Attributes	Null	Default
1	**EMP_ID** 🔑	int(10)			Yes	*NULL*
2	**MONTHLY_ALLOWANCE**	int(10)			Yes	*NULL*
3	**BASIC_PAY**	int(10)			Yes	*NULL*
4	**DA**	int(10)			Yes	*NULL*
5	**CA**	int(10)			Yes	*NULL*
6	**HRA**	int(10)			Yes	*NULL*
7	**FOOD_ALLOWANCE**	int(10)			Yes	*NULL*
8	**BONUS**	int(10)			Yes	*NULL*
9	**PROVISIONAL_TAX**	int(10)			Yes	*NULL*
10	**PROVIDENT_FUND**	int(10)			Yes	*NULL*
11	**INSURANCE**	int(10)			Yes	*NULL*
12	**TOTAL_EARNINGS**	int(10)			Yes	*NULL*
13	**DEDUCTIONS**	int(10)			Yes	*NULL*
14	**NET_PAY**	int(10)			Yes	*NULL*

Table 4.3.6 Payslip Table

Chapter 5

IMPLEMENTATION

5.1 Introduction

This project is designed and implemented using **MySQL** database along with **PHP** for back-end implementation and **HTML** and **CSS** for front-end design. IDE used is **Visual Studio Code**.

5.1.1 MySQL

The back-end of the web application is basically the brains behind the front-end. It comprises of three components: server, application and database. It is a link between the server and the user. Most of the coding for the web application can be found in the back-end and the quality of this code determines how the website functions. In this project MySQL is used as a back-end technology. MySQL is a multithreaded, multiuser SQL Database Management System. The basic program run as server providing multiuser access to a number of databases. MySQL is currently the world's most popular and widely used open-source database technology and data storage system. MySQL offers great reliability and ease of use. MySQL runs on virtually all platforms, including Linux, UNIX, and Windows.

5.1.2 Hypertext Preprocessor (PHP)

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open-source general purpose scripting language that is primarily designed for web development and can be embedded into HTML. PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page. PHP commands can be embedded directly into an HTML source document external file to process data or it can be used in combination with various web template systems, web content management systems and web frameworks. It has also evolved to include a command line interface capability and can be used in standalone graphical applications.

A good benefit of using PHP is that it can interact with many different database languages including MySQL. Both PHP and MySQL are compatible with an Apache server which is also free to license. PHP can also run on Windows, Linux and UNIX servers. Due to all these languages being free it is cheap and easy to setup and create a website using PHP. PHP also has very good online documentation with a good framework of functions in place.

5.1.3 Hypertext Markup Language (HTML)

HTML is the web's core language for creating documents and applications for everyone to use, anywhere. It is standardizing system for tagging text files to achieve font, color, graphic and hyperlink effects on World Wide Web pages. HTML elements form the building blocks of all websites. The markup tells the web browsers how to display web pages. Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of websites. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of corner stone technologies of the World Wide Web.

5.1.4 Cascading Style Sheets (CSS)

CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a simple mechanism for adding style (e.g. fonts, colors, spacing etc.) to web documents. CSS defines how HTML elements are displayed. CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web application and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors and fonts. This separation can improve content accessibility, provide more flexibility. This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods.

5.1.5 XAMPP

XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows**.**

5.2 Table Creation

There are 6 tables used in this payroll management project:

1. Admin
2. Employee
3. Salary
4. Apply_leave
5. Profile
6. Payslip

Creation of admin table

```
CREATE TABLE ADMIN
(USERNAME VARCHAR (20),
 PASSWORD VARCHAR (20));
```

Creation of employee table

```
CREATE TABLE EMPLOYEE
(EMP_ID INT (10) PRIMARY KEY,
 ENAME VARCHAR (30),
 DOB DATE,
 GENDER CHAR (1),
 ADDRESS VARCHAR (50),
 CONTACT INT (10),
 EMAIL VARCHAR (30),
 DOJ DATE,
 DEPARTMENT VARCHAR (30),
 PASSWORD VARCHAR (30),
 CONFIRM_PASSWORD VARCHAR (50));
```

Creation of salary table

```
CREATE TABLE SALARY
(EMP_ID INT (10) REFERENCES EMPLOYEE (EMP_ID) ON DELETE CASCADE,
 MONTHLY_ALLOWANCE INT (10),
 BASIC_PAY INT (10),
 DA INT (10),
 CA INT (10),
 HRA INT (10),
 FOOD_ALLOWANCE INT (10),
 BONUS INT (10),
 PROVISIONAL_TAX INT (10),
 PROVIDENT_FUND INT (10),
 INSURANCE INT (10));
```

Creation of payslip table

```
CREATE TABLE PAYSLIP
(EMP_ID INT (10) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE,
 MONTHLY_ALLOWANCE INT (10),
 BASIC_PAY INT (10),
 DA INT (10),
 CA INT (10),
 HRA INT (10),
 FOOD_ALLOWANCE INT (10),
 BONUS INT (10),
 PROVISIONAL_TAX INT (10),
 PROVIDENT_FUND INT (10),
 INSURANCE INT (10),
 TOTAL_EARNINGS INT (10),
 DEDUCTIONS INT (10),
 NET_PAY INT (10));
```

Creation of apply_leave table

```
CREATE TABLE APPLY_LEAVE
(EMP_ID INT (10) REFERENCES EMPLOYEE(EMP_ID) ON DELETE CASCADE,
 NO_OF_DAYS INT (2),
 APPLICATION_DATE DATE,
 FROM_DATE DATE,
 TO_DATE DATE,
 LEAVE_TYPE VARCHAR (30),
 REASON VARCHAR (30));
```

Creation of table profile

```
CREATE TABLE PROFILE
(EMP_ID INT (10) REFERENCES EMPLOYEEE(EMP_ID) ON DELETE CASCADE,
 ENAME VARCHAR (30),
 DOB DATE,
 GENDER CHAR (1),
 ADDRESS VARCHAR (50),
 CONTACT INT (10),
 EMAIL VARCHAR (30),
 DOJ DATE,
 DEPARTMENT VARCHAR (30),
 PASSWORD VARCHAR (30));
```

5.3 Implementation of Insert / Update and Delete

5.3.1 INSERT

Inserting into admin table :

INSERT INTO ADMIN VALUES (‘USERNAME’ , ‘PASSWORD’);

Inserting into employee table :

INSERT INTO EMPLOYEE VALUES (EMP_ID, ‘ENAME’, ‘DOB’, ‘GENDER’, ‘ADDRESS’, CONTACT, ‘EMAIL’, ‘DOJ’, ‘DEPARTMENT’, ‘PASSWORD’, ‘CONFIRM_PASSWORD’);

Inserting into salary table :

INSERT INTO SALARY VALUES (EMP_ID, MONTHLY_ALLOWANCE, BASIC_PAY, DA, CA, HRA, FOOD_ALLOWANCE, BONUS, PROVISIONAL_TAX, PROVIDENT_FUND, INSURANCE);

Inserting into apply_leave table :

INSERT INTO APPLY_LEAVE VALUES (EMP_ID, NO_OF_DAYS, ‘APPLICATION_DATE’, ‘FROM_DATE’, ‘TO_DATE’, ‘LEAVE_TYPE’, ‘REASON’);

Inserting into payslip table :

INSERT INTO PAYSLIP VALUES (EMP_ID, MONTHLY_ALLOWANCE, BASIC_PAY, DA, CA, HRA, FOOD_ALLOWANCE, BONUS, PROVISIONAL_TAX, PROVIDENT_FUND, INSURANCE, TOTAL_EARNINGS, DEDUCTIONS, NETPAY);

Inserting into profile table :

INSERT INTO PROFILE VALUES (EMP_ID, ‘ENAME’, ‘DOB’, ‘GENDER’, ‘ADDRESS’, CONTACT, ‘EMAIL’, ‘DOJ’, ‘DEPARTMENT’, ‘PASSWORD’);

Implementation of insertion using PHP

Sample Code: empregister.php

```
<?php
if(isset($_POST['submit']))
{
        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbDatabase = "payroll";
        // Create connection
        $conn = new mysqli ($servername, $username, $password, $dbDatabase);
        $id = $_POST['id'];
        $name = $_POST['name'];
        $dob = $_POST['dob'];
        $gender = $_POST['gender'];
        $address = $_POST['address'];
        $contact = $_POST['contact'];
        $doj = $_POST['doj'];
        $email = $_POST['email'];
        $department = $_POST['department'];
        $pwd = $_POST['password'];
        $cpwd=$_POST['cpassword'];
        if($pwd==$cpwd)
        {
                $sql = "insert into EMPLOYEE (EMP_ID, ENAME, DOB, ADDRESS,
                GENDER, CONTACT, EMAIL, DOJ, DEPARTMENT, PASSWORD,
                CONFIRM_PASSWORD) values ($id, '$name', '$dob', '$address', '$gender',
                $contact, '$email', '$doj', '$department', '$pwd', '$cpwd')";
```

```
                         if (mysqli_query ($conn, $sql))
                        {
                                        echo "Records inserted successfully.";
                                        header ("Location: adminhome.html");
                          }
                        else
                        {
                              echo "ERROR: Could not able to execute $sql. " . mysqli_error($conn);
                          }
            }
            else {
              echo '<script type="text/javascript">alert ("Please ConfirmPassword ") </script>';
            }
   }
 ?>
```

5.3.2 UPDATE

UPDATE PROFILE SET ADDRESS = ‘KOREA’ WHERE EMP_ID = 101;

Sample Code: profile-update.php

```
<?php
session_start ();
if(isset ($_POST['submit']))
{
                $servername = "localhost";
                $username = "root";
                $password = "";
                $dbDatabase = "payroll";
                // Create connection
                $conn = new mysqli ($servername, $username, $password, $dbDatabase);
                $id=$_SESSION ['emp_id'];
```

```
if ($POST[' newaddress'] != "")
{
        $newaddress= $_POST['newaddress'];
        $sql1 = "UPDATE employee SET ADDRESS = '$newaddress'
        WHERE EMP_ID =  '$id' ";
        $result1=mysqli_query($conn,$sql1);
        if(!$result1)
        {
                die('Could not update data: '.mysqli_error($conn));
        }
        echo 'Address updated Successfully!';
}
if($_POST['newcontact']!="")
{
        $newcontact= $_POST['newcontact'];
        $sql2 = "UPDATE employee SET CONTACT = '$newcontact'
        WHERE EMP_ID = '$id' ";
        $result2=mysqli_query($conn,$sql2);
        if(!$result2)
        {
          die('Could not update data: '.mysqli_error($conn));
        }
        echo 'Contact updated Successfully!';
}
if($_POST['password']!="")
{
        $newpassword= $_POST['newpwd'];
        $sql2 = "UPDATE employee SET PASSWORD = '$newpassword',
        CONFIRM_PASSWORD='$newpassword'
        WHERE EMP_ID = '$id' ";
        $result2 = mysqli_query ($conn,$sql2);
```

```
                if(!$result2)
                {
                    die('Could not update data: '.mysqli_error($conn));
                }
                echo 'Password updated Successfully!';
        }
        header ("Location: view-profile.php");
        mysqli_close ($conn);
}
?>
```

5.3.3 DELETE

```
DELETE * FROM EMPLOYEE WHERE EMP_ID=101;
```

Sample Code: delete-emp.php

```
<?php
  if(isset($_POST['submit']))
  {
          $servername = "localhost";
          $username = "root";
          $password = "";
          $dbDatabase = "payroll";
          // Create connection
          $conn = new mysqli ($servername, $username, $password, $dbDatabase);
          $id = $_POST['username'];
          $department = $_POST['text'];
          $sql ="delete * from EMPLOYEE where EMP_ID='$id'";
```

```
                if(mysqli_query($conn, $sql))
               {
                              echo " Records deleted successfully.";
                              header ("Location: adminhome.html");
                }
                else
                {
                     echo "ERROR: Could not able to execute $sql. " . mysqli_error($conn);
                }
     }
 ?>
```

5.5 Implementation of Triggers

Trigger to insert the details of the employee into the profile table as soon as they are inserted in the employee table

```
CREATE TRIGGER INTO_PROFILE
AFTER INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
INSERT PROFILE
(EMP_ID, ENAME, DOB, GENDER, ADDRESS, CONTACT, EMAIL, DOJ,
DEPARTMENT )
VALUES
(NEW.EMP_ID, NEW.ENAME, NEW.DOB, NEW.GENDER, NEW.ADDRESS,
NEW.CONTACT, NEW.EMAIL, NEW.DOJ, NEW.DEPARTMENT)
END
```

Trigger to insert the salary details into to payslip table as soon as the employee is paid salary to generate the payslip

```
CREATE TRIGGER INTO_PAYSLIP_SALARY
AFTER INSERT ON SALARY
FOR EACH ROW
BEGIN
INSERT INTO PAYSLIP
(EMP_ID, MONTHLY_ALLOWANCE, BASIC_PAY, DA, CA, HRA,
FOOD_ALLOWANCE, BONUS, PROVISIONAL_TAX, PROVIDENT_FUND,
INSURANCE)
VALUES
(NEW.EMP_ID, NEW.MONTHLY_ALLOWANCE, NEW.BASIC_PAY, NEW.DA,
NEW.CA, NEW.HRA, NEW.FOOD_ALLOWANCE, NEW.BONUS,
NEW.PROVISIONAL_TAX, NEW.PROVIDENT_FUND, NEW.INSURANCE)
END
```

Trigger to calculate the total earnings, total deductions and net pay of the employee to generate the payslip of the employee.

```
CREATE TRIGGER UPDATE_PAYSLIP
AFTER INSERT ON SALARY
FOR EACH ROW
BEGIN
UPDATE PAYSLIP SET
TOTAL_EARNINGS = NEW.MONTHLY_ALLOWANCE+ NEW.BASIC_PAY+
NEW.DA+ NEW.CA+ NEW.HRA+ NEW.FOOD_ALLOWANCE+ NEW.BONUS,
DEDUCTIONS = NEW.PROVISIONAL_TAX+ NEW.PROVIDENT_FUND+
NEW.INSURANCE,
NET_PAY = TOTAL_EARNINGS - DEDUCTIONS
WHERE NEW.EMP_ID=PAYSLIP.EMP_ID
END
```

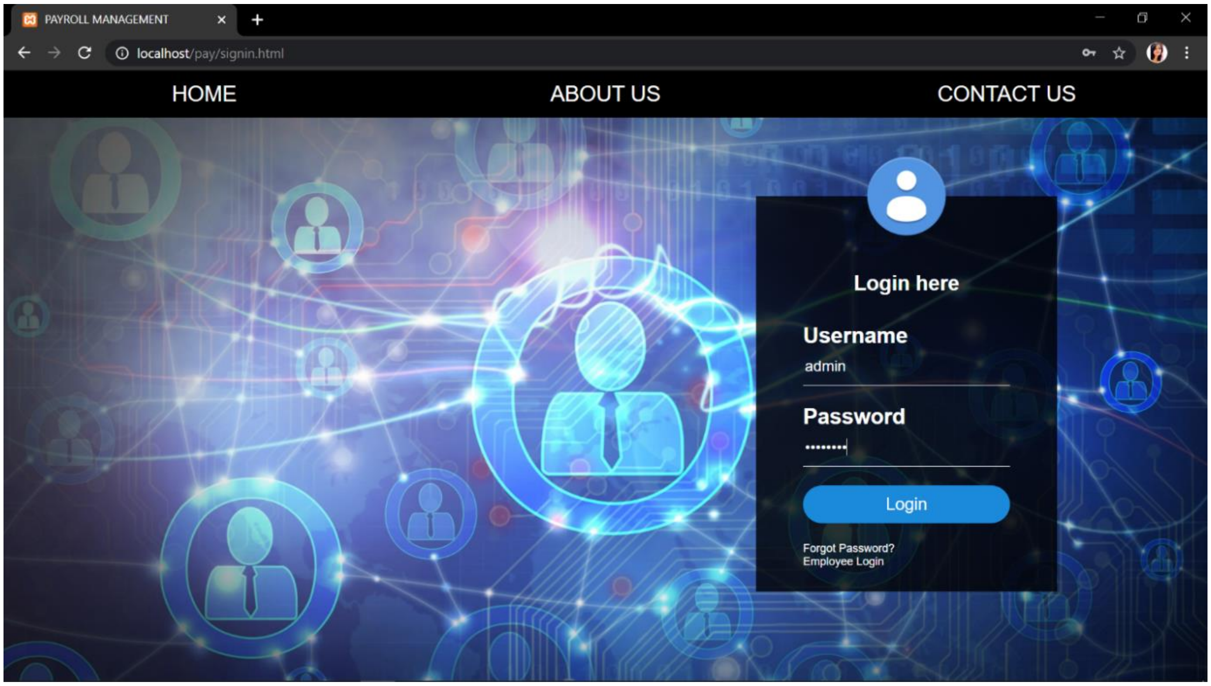
Trigger to update the monthly allowance of the employee based on the number of days of leave applied by the him and hence updating the total earnings and the netpay in the payslip.

```
CREATE TRIGGER LEAVE_DEDUCTION
AFTER INSERT ON APPLY_LEAVE
FOR EACH ROW
BEGIN
UPDATE PAYSLIP SET
MONTHLY_ALLOWANCE=MONTHLY_ALLOWANCE-
((MONTHLY_ALLOWANCE /26) * NEW.NO_OF_DAYS),
TOTAL_EARNINGS= MONTHLY_ALLOWANCE+ BASIC_PAY+ DA+ CA+ HRA+
FOOD_ALLOWANCE+ BONUS,
NET_PAY=TOTAL_EARNINGS-DEDUCTIONS
WHERE NEW.EMP_ID= PAYSLIP.EMP_ID;
```

Chapter 6

SAMPLE OUTPUT

Snapshot 1 : The login page where the admin who has the authority can login using predefined username and password.

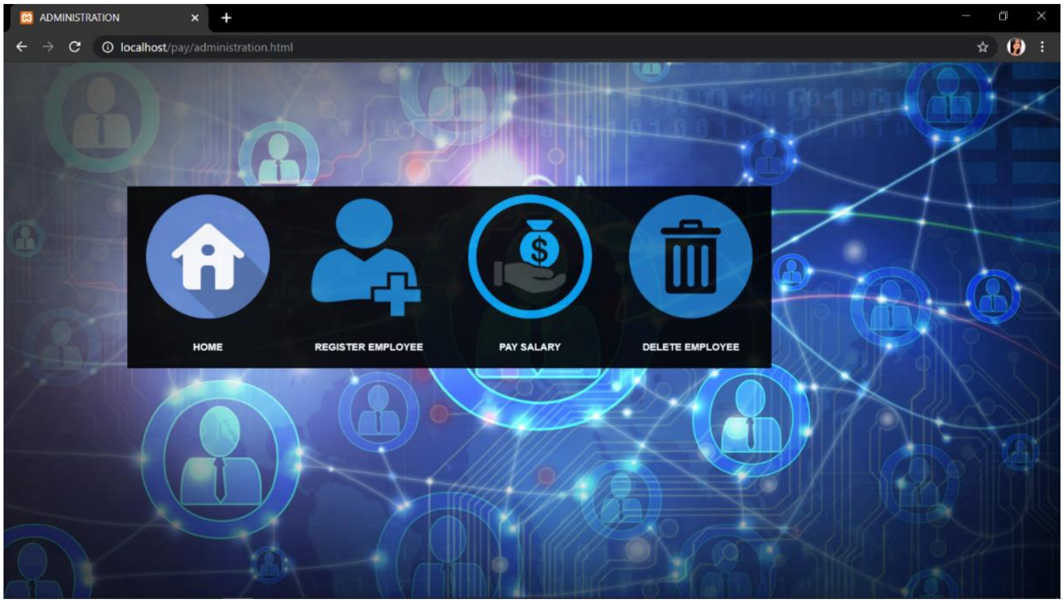


Username
admin

Password
••••••••

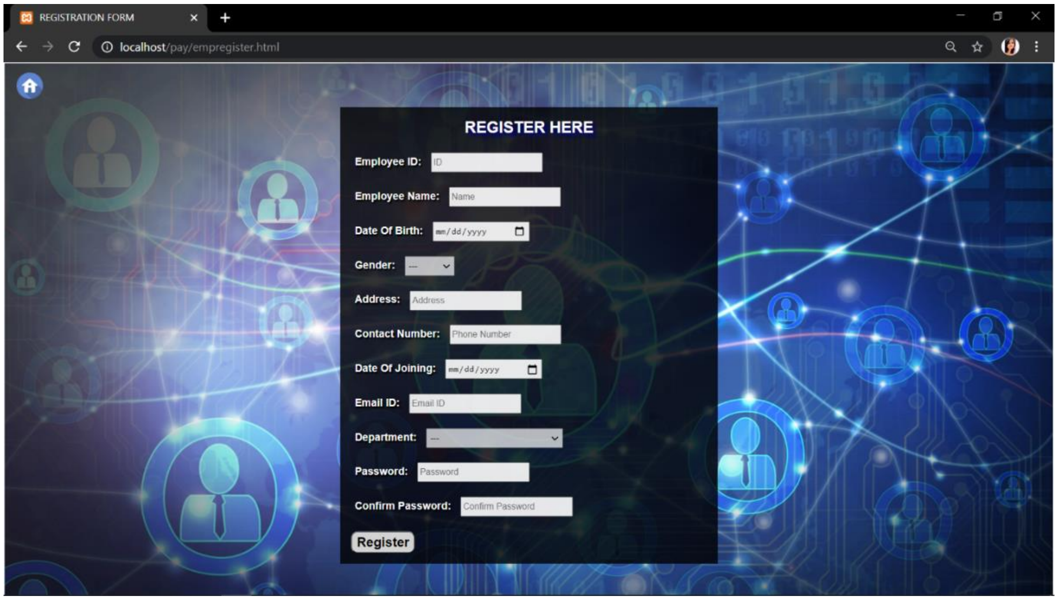
6.1 LOGIN PAGE

Snapshot 2 : Admin administration page where he can register new employees, pay salaries to the employees or delete any employee.



6.2 ADMINISTRATION PAGE

Snapshot 3 : Employee registration page where the admin registers new employees by inserting the details into the registration form.



REGISTER HERE

Employee ID: ID

Employee Name: Name

Date Of Birth: mm/dd/yyyy

Gender: --

Address: Address

Contact Number: Phone Number

Date Of Joining: mm/dd/yyyy

Email ID: Email ID

Department: --

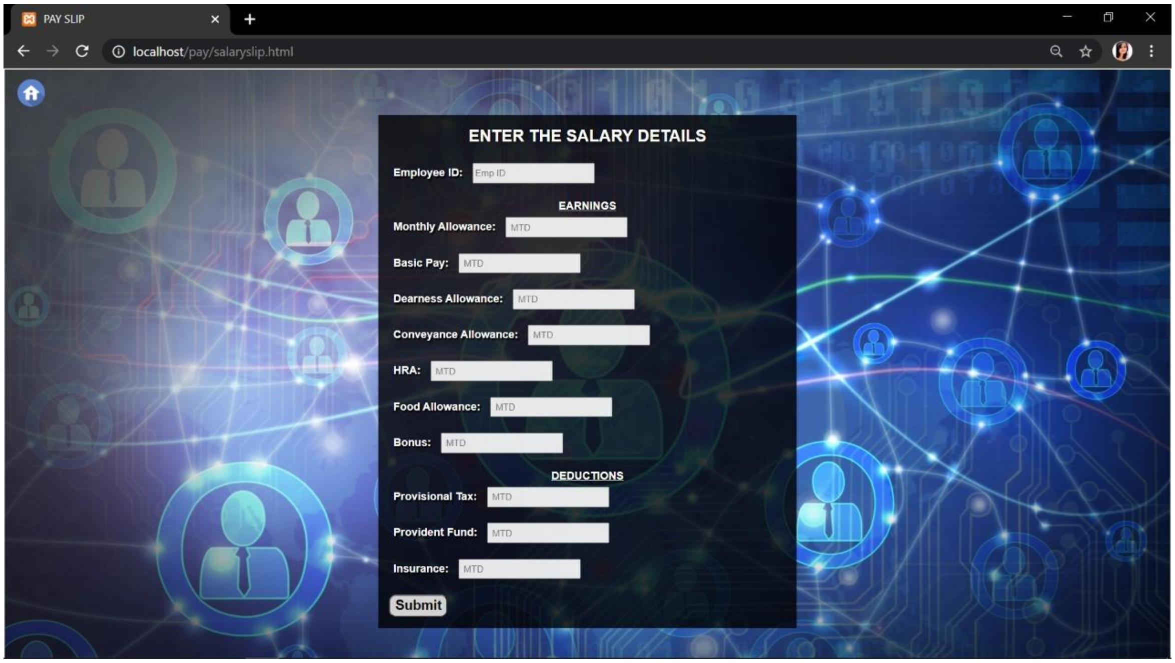
Password: Password

Confirm Password: Confirm Password

Register

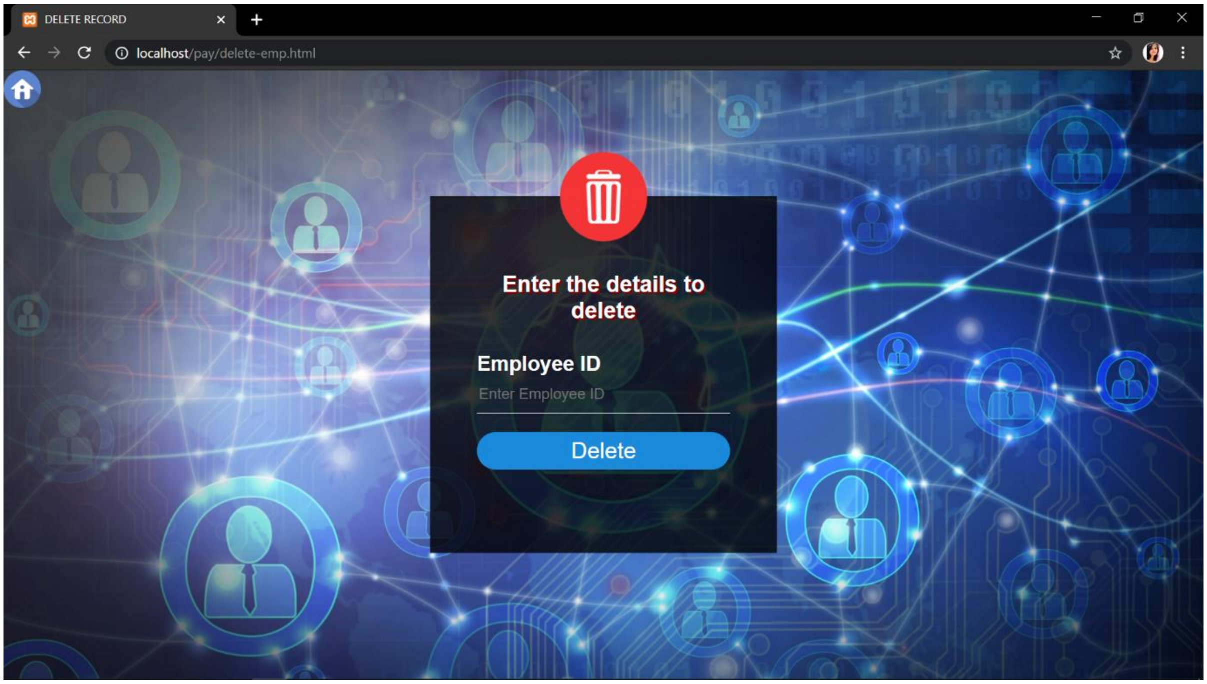
6.3 REGISTRATION PAGE

Snapshot 4 : Salary Page where the admin pays salary to the employees.



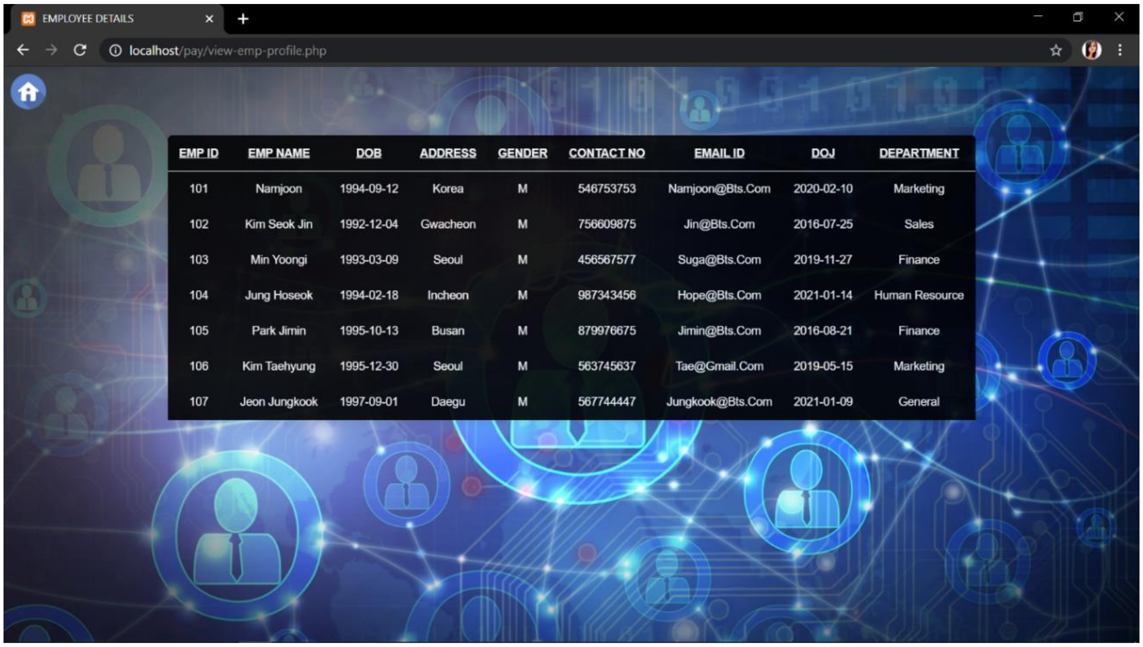
6.4 SALARY PAGE

Snapshot 5 : Employee delete page using which the admin can remove any employee from the database by specifying the employee ID.



6.5 DELETE PAGE

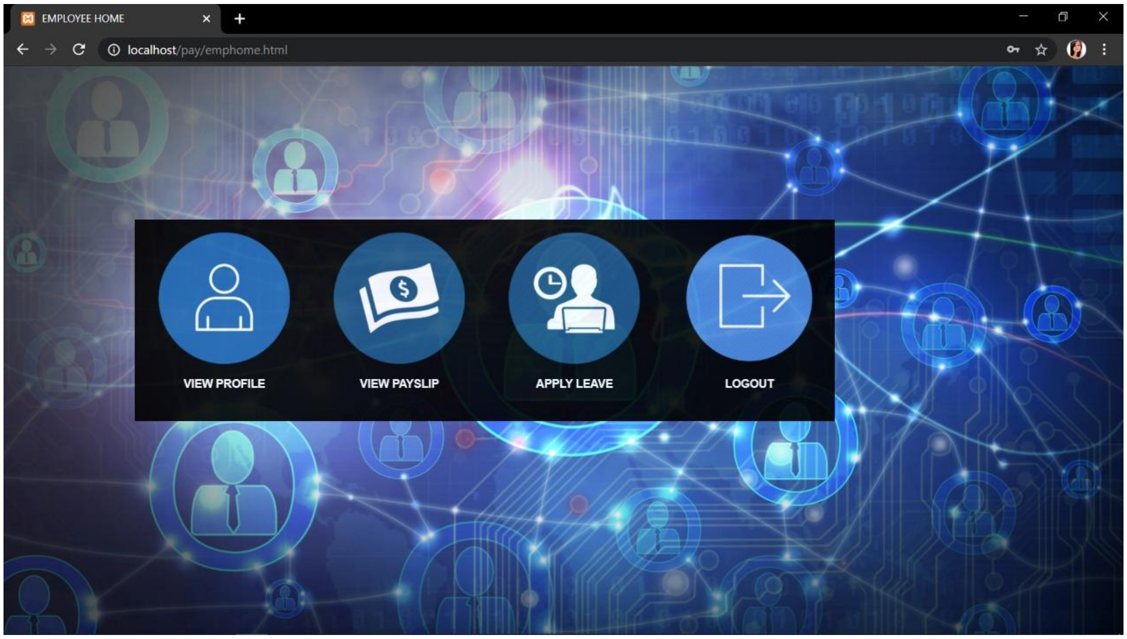
Snapshot 6 : Employee profile report containing the details of all the employees in the table which can be accessed by the admin.



EMP ID	EMP NAME	DOB	ADDRESS	GENDER	CONTACT NO	EMAIL ID	DOJ	DEPARTMENT
101	Namjoon	1994-09-12	Korea	M	546753753	Namjoon@Bts.Com	2020-02-10	Marketing
102	Kim Seok Jin	1992-12-04	Gwacheon	M	756609875	Jin@Bts.Com	2016-07-25	Sales
103	Min Yoongi	1993-03-09	Seoul	M	456567577	Suga@Bts.Com	2019-11-27	Finance
104	Jung Hoseok	1994-02-18	Incheon	M	987343456	Hope@Bts.Com	2021-01-14	Human Resource
105	Park Jimin	1995-10-13	Busan	M	879976675	Jimin@Bts.Com	2016-08-21	Finance
106	Kim Taehyung	1995-12-30	Seoul	M	563745637	Tae@Gmail.Com	2019-05-15	Marketing
107	Jeon Jungkook	1997-09-01	Daegu	M	567744447	Jungkook@Bts.Com	2021-01-09	General

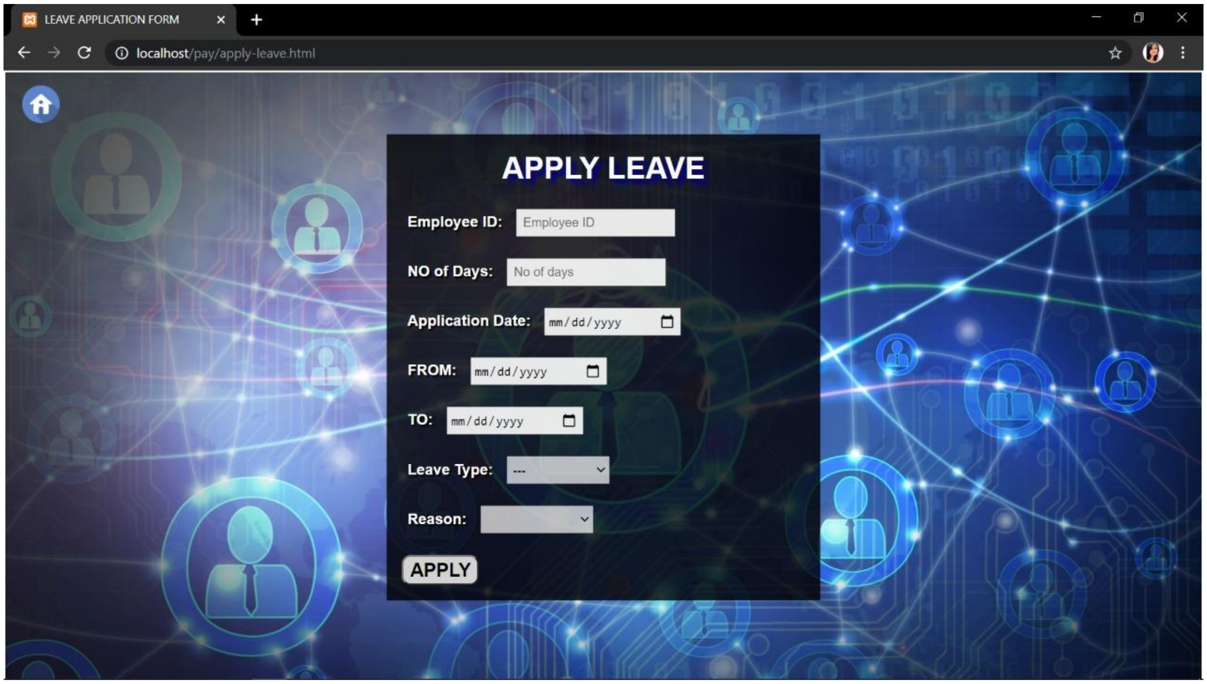
6.6 EMPLOYEE DETAILS REPORT

Snapshot 7 : Employee Salary report containing all the earnings, deductions and the netpay.



6.7 EMPLOYEE HOME PAGE

Snapshot 8 : Apply leave page where an employee can apply for the leave.



APPLY LEAVE

Employee ID: Employee ID

NO of Days: No of days

Application Date: mm / dd / yyyy

FROM: mm / dd / yyyy

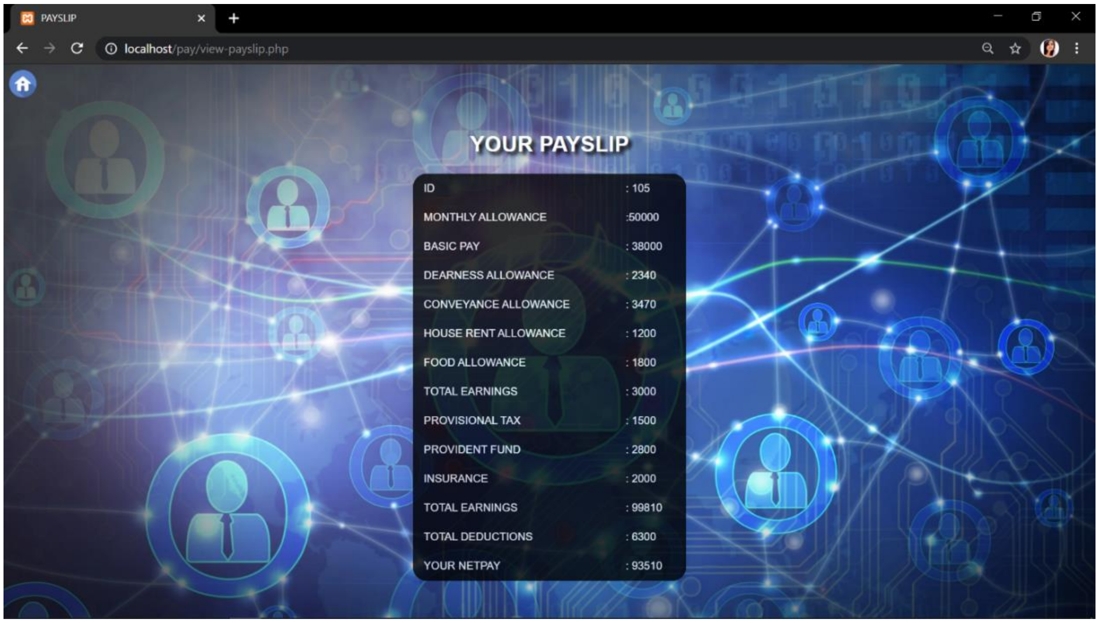
TO: mm / dd / yyyy

Leave Type: ---

Reason:

6.8 APPLY_LEAVE PAGE

Snapshot 9 : The payslip of the employee which is generated automatically as soon as the salary is paid by the admin to an employee which contains the total earnings, total deductions and the net pay of the employee.



YOUR PAYSLIP

ID	: 105
MONTHLY ALLOWANCE	:50000
BASIC PAY	: 38000
DEARNESS ALLOWANCE	: 2340
CONVEYANCE ALLOWANCE	: 3470
HOUSE RENT ALLOWANCE	: 1200
FOOD ALLOWANCE	: 1800
TOTAL EARNINGS	: 3000
PROVISIONAL TAX	: 1500
PROVIDENT FUND	: 2800
INSURANCE	: 2000
TOTAL EARNINGS	: 99810
TOTAL DEDUCTIONS	: 6300
YOUR NETPAY	: 93510

6.9 EMPLOYEE PAYSLIP

CONCLUSION

In conclusion, a database is a far more efficient mechanism to store and organize data than spreadsheets. It allows for a centralized facility that can easily be modified and quickly shared among multiple users. Having a web based front end removes the requirement of users having to understand and use a database directly, and allows users to connect from anywhere with an internet connection and a basic web browser. It also allows the possibility of queries to obtain information for various surveys. This application used for keeping track on employee information. It is built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small scale organization where the number of employees is limited. This database consists of different entities like the profile details, application of leave and pay-slip. According to the requested requirement the admin can add, manipulate, update and delete all employee data in his organization. The required records can be easily viewed by the admin anytime time he wants in an instant. The payment of the employee is based on monthly basis. Numerous validations implemented would enable the admin to enter accurate data. The main objective of this framework is to save time, make the system cost effective and management records efficiently.

FUTURE ENHANCEMENT

Our system is developed based on demand of user's satisfaction and facilities. In our system we have used the modern web technologies to make our system fast, convenient and efficient for all of the personnel mentioned. In future we plan to implement more functionalities and more improvement will be possible to pursue like keeping track of the attendance details of the employees. We also intend to give way to more integrated payroll and self-service technology where every employee has security rights to the payroll system.

REFERENCES

[1] *Fundamentals of Database Systems*, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[2] www.stackoverflow.com

[3] www.w3schools.com