Project Report

On

Employee Management System

Acknowledgement

I am grateful to all my faculty who have contributed in inspiring and clarifying my thought over the years; (name of faculty) and many other faculty of XYZ College. Special gratitude I give to my respected head of the division Mr.XYZ, for allowing me to use the facilities available and also help me to coordinate my project. I am thankful also for the tireless generosity to my sponsors, well-wishers and my family for spiritual, moral and even financial support. Thanks also to the entire staff of my College, Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion. Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project. Finally, Honor and glory to the highest God for having enabled me successfully accomplishes.

.

Abstract

The main objective of this project is to build an employee database system that will store records of employees. It is purposed to reduce time spent on administrative tasks. The system is intended to accept process, generate employee. The system is also intended to provide better services to users, provide meaningful, consistent, and timely data and information and finally promotes efficiency by converting paper processes to electronic form. The system was developed using basic technologies such as MySQL database and PHP. The system is free of errors and very efficient and less time consuming due to the care taken to develop it. All the phases of software development cycle are employed and it is worthwhile to state that the system is user friendly and strong. Provision is made for future development in the system.

Introduction

Employee Management System is software which is helpful for employees as well as the companies or organizations. In the current system all the activities are done manually. It is very time consuming and costly. Our Employee Management System deals with the various activities related to the employees in the company.

The two main users involved in this system are

1. Employee
2. Admin

In the Software admin can add an employee by the authentication code i.e. empid and password with help of empid and password employee can login his/her account and employee can view and request for leave and view salary details which is maintain by admin.

Purpose

The objective of **Employee Management System** is to allow the administrator of any organization to edit and find out the personal details of an employee and allows the employee to keep up to date his profile .It'll also facilitate keeping all the records of employee, such as their id, name, mailing address, phone number, DOB etc. So all the information about an employee will be available in a few seconds.

Overall, it'll make Employee Information Management an easier job for the administrator and the employee of any organization.

Advantages:

- It helps the company's administrator to handle and manage employee records.
- It helps companies' administrator to generate report.
- It brings transparency and efficiency in the working of organization.

Disadvantages:

- The system can only handle single organization.
- The system does not include bank payment, dd, cheque status.
-

Applications:

The website **Employee Management System** is aimed towards recording a considerable number of employee records and needs online assistance for managing records of employee. Website should be user-friendly, 'quick to learn' and reliable website for the above purpose.**Employee Management System**is intended to be a stand-alone product andshould not depend on the availability of other website. The system will also have an administrator who has full-fledged rights with regards to performing all actions related to control and management of the website.

Feasibility study

Whenever we design a new system, normally the management will ask for a feasibility report of the new system. The management wants to know the technicalities and cost involved in creation of new system.

- Technical feasibility
- Economic feasibility
- Physical feasibility

Technical feasibility:

Technical feasibility involves study to establish the technical capability of the system being created to accomplish all requirements to the user. The system should be capable of handling the proposed volume of data and provide users and operating environment to increase their efficiency.
For example, system should be capable of handling the proposed volume of data and provide users.

Economic feasibility:

Economic feasibility involves study to establish the cost benefit analysis. Money spent on the system must be recorded in the form of benefit from the system. The benefits are of two types:

Tangible benefits:

- Saving man labor to do tedious tasks saves time.
-

Intangible benefits:

- Improves the quality of organization.

Physical feasibility:

It involves study to establish the time responses of the new system being created. For e.g., if the new system takes more than one day to prepare crucial finance statement for the management, wherever it was required in an hour, the system fails to provide the same.

It should be clearly established that the new system requirements in the form of time responses would be completely met with. It may call for increase in cost. If the required cost is sacrificed then the purpose of the new system may not be achieved even if it was found to be technically feasible.

Scope of the Project

The proposed system will affect or interface with the user (employee) and companies' administrator. The system works and fulfills all the functionalities as per the proposed system. It will provide reduced response time against the queries made by different users. This project is based on PHP language with MYSQL database which manage the details of the employee because it is a tedious job for any organization. Employee Management system will store all the details of the employee including their background information.

All possible features such as verification, validation, security, user friendliness etc. have been considered.

The different types of modules present in this project are

1. Admin
2. Employee

Admin:

1. **Dashboard**: In this section, admin can see all detail in brief like Total Registered Employee , Total Listed Department, Total Leave Type, Total Applied Leave, New Leave Request, Approved Leave Request and Rejected Leave Request.

2. **Department** In this section, admin can manage department (Add/Update/Delete).
3. **Leave Type** In this section, admin can manage leave type (Add/Update/Delete).

4. **Employee**: In this section, admin can manage the employee (Add/Update/Delete).
5. **Salary:** In this section, the admin can manage salary (Add/Update/Delete).
6. **Leave Request:** In this section, the admin can manage leave request and update the leave request.
7. **Reports:** In this section admin, can view how much employee has been register in particular period.
8. Admin can also update his profile, change the password and recover the password.

User (Employee):

1. **Dashboard**: It is welcome page for employee.
2. **My Profile**: In this section, employee can view and update his/her profile.
3. **Leave**: In this section, employee can apply for leave and view leave history.
4. **Salary History**: In this section, employee can view history of his/her salaries.
5. Employee can also view his profile, change the password and recover the password.

Software & Hardware requirements

- ✓ Any Version of browser after Mozilla Firefox 4.0, Internet Explorer 6.0,chrome

Hardware requirements:

- ✓ Any processor after Pentium 4.
- ✓ Any version of Windows XP or later.
- ✓ Processor speed: 2.0 GHz
- ✓ RAM : 1GB
- ✓ Hard disk: 40GB to 80 GB

Software requirements:

- ✓ Database : MySQL
- ✓ Server : Apache
- ✓ Frontend : HTML
- ✓ Scripting Language : JavaScript
- ✓ IDE : Sublime
- ✓ Technology : PHP

System Design

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data

Unified Modelling Language Diagrams (UML):

- The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.
- A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

i. This view represents the system from the users perspective.

ii. The analysis representation describes a usage scenario from the end-users perspective**.**

Structural model view

- ◆ In this model the data and functionality are arrived from inside the system.
- ◆ This model view models the static structures.

Behavioural Model View

- ◆ It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various

structural elements described in the user model and structural model view.

Implementation Model View

- In this the structural and behavioural as parts of the system are represented as they are to be built.

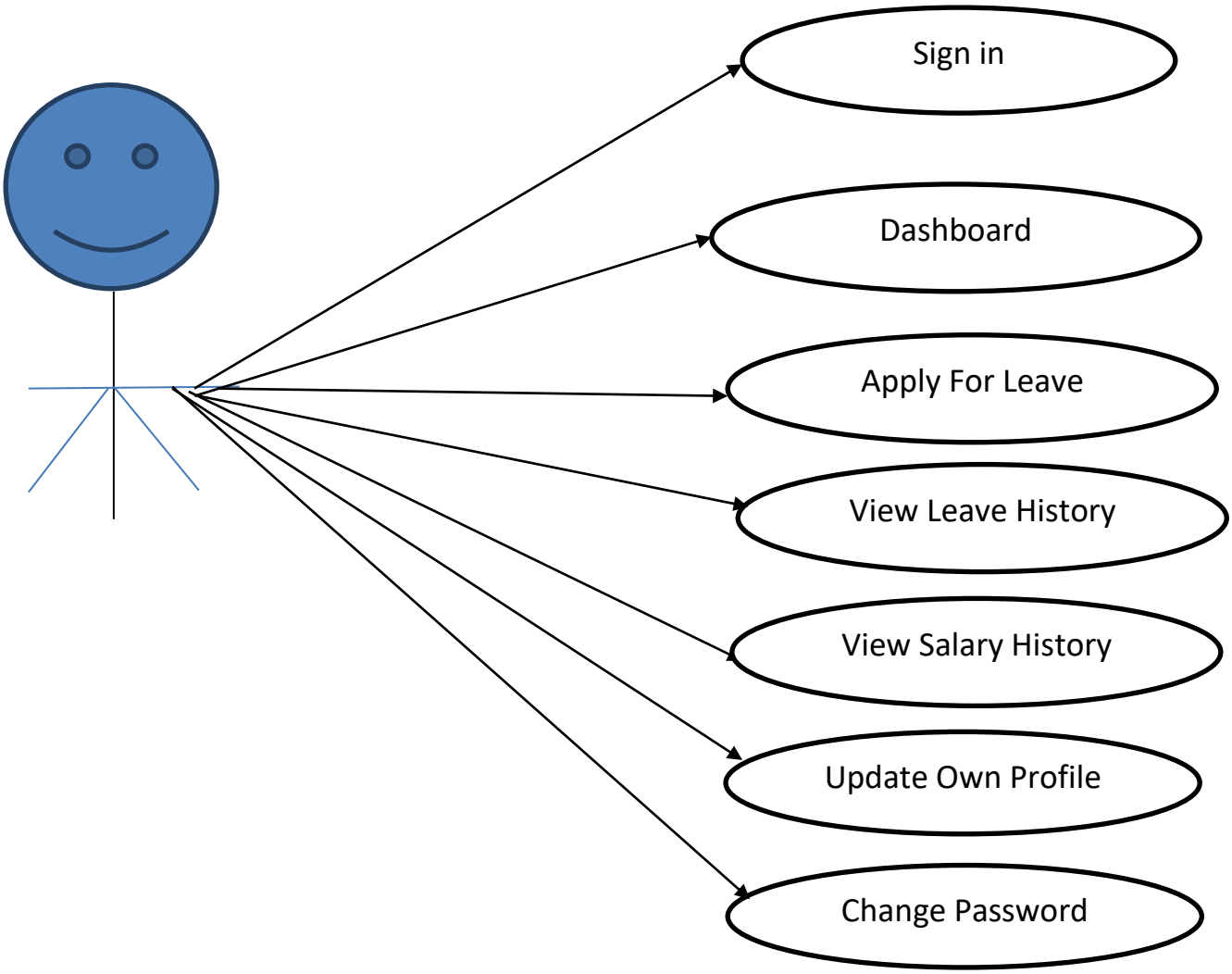
Environmental Model View

In these the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

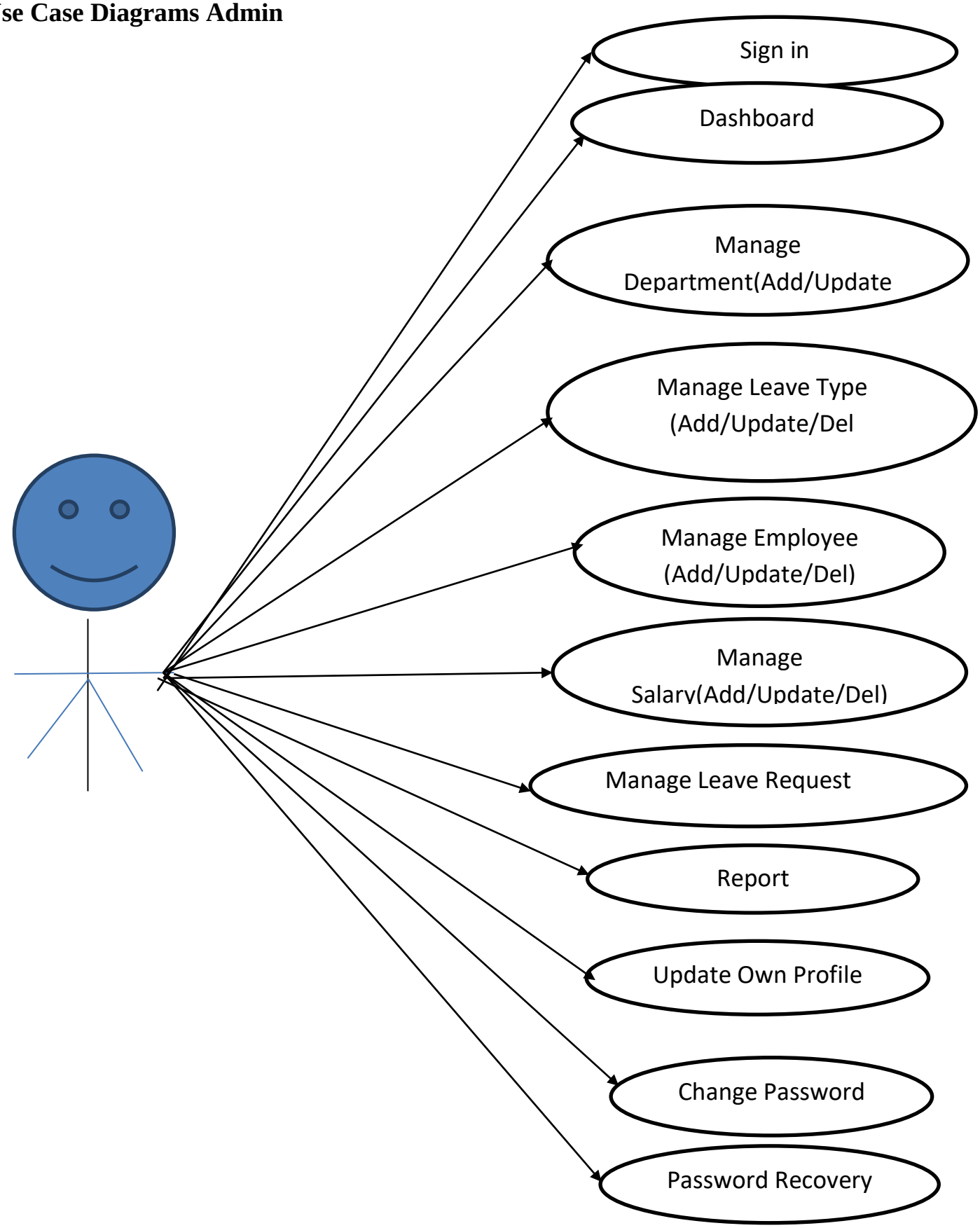
UML is specifically constructed through two different domains they are

- UML Analysis modelling, which focuses on the user model and structural model views of the system?
- UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views**.**

Use Case Diagrams Employee



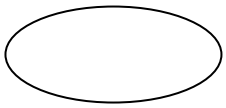
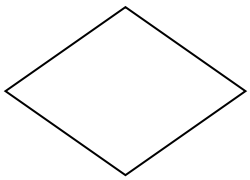
Use Case Diagrams Admin



ENTITY-RELATIONSHIP Diagrams

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

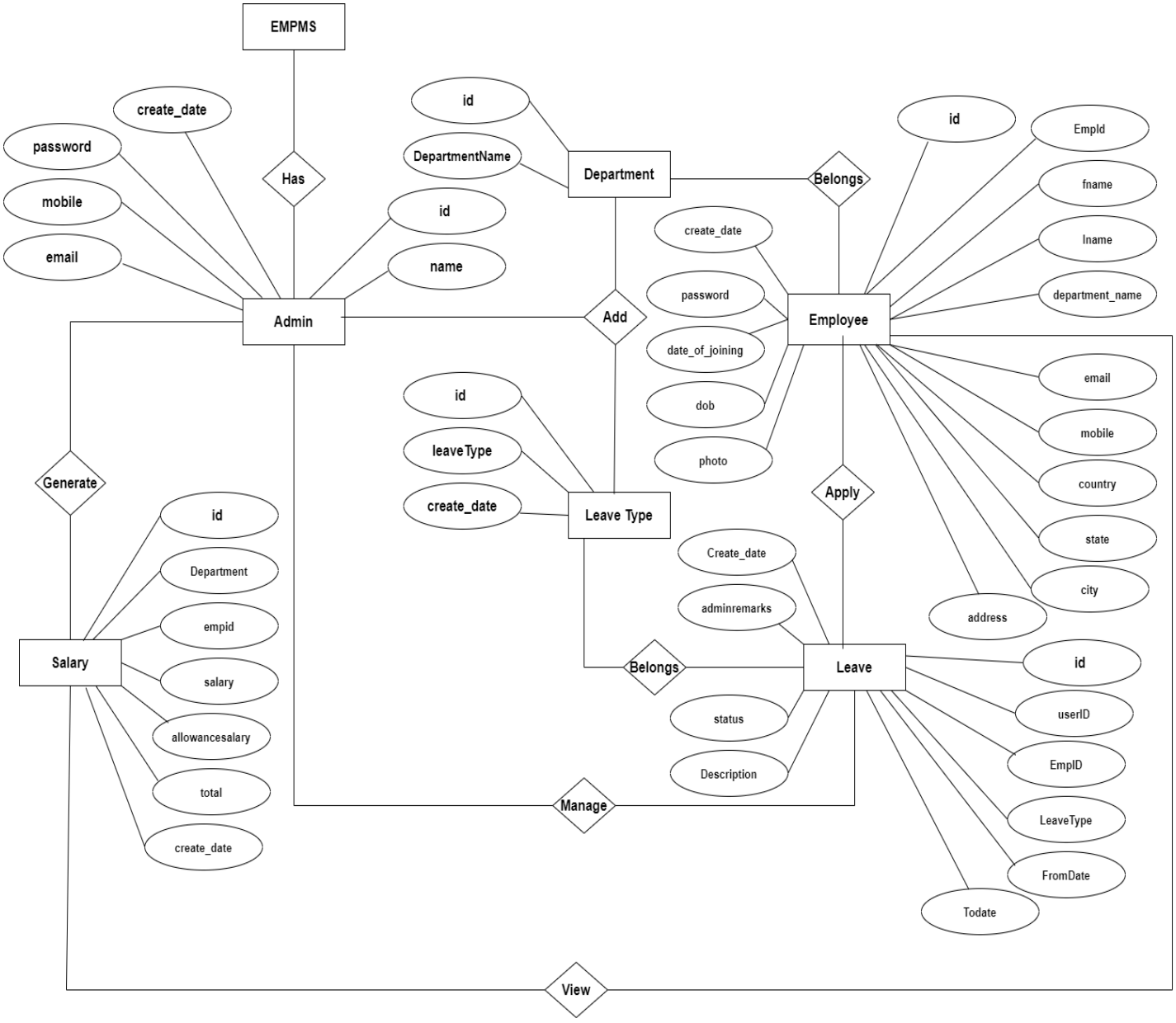
The symbols used in E-R diagrams are:

SYMBOL	PURPOSE
	Represents Entity sets.
	Represent attributes.
	Represent Relationship Sets.
	Line represents flow

Structured analysis is a set of tools and techniques that the analyst.

To develop a new kind of a system:

The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection a personal consideration.



Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

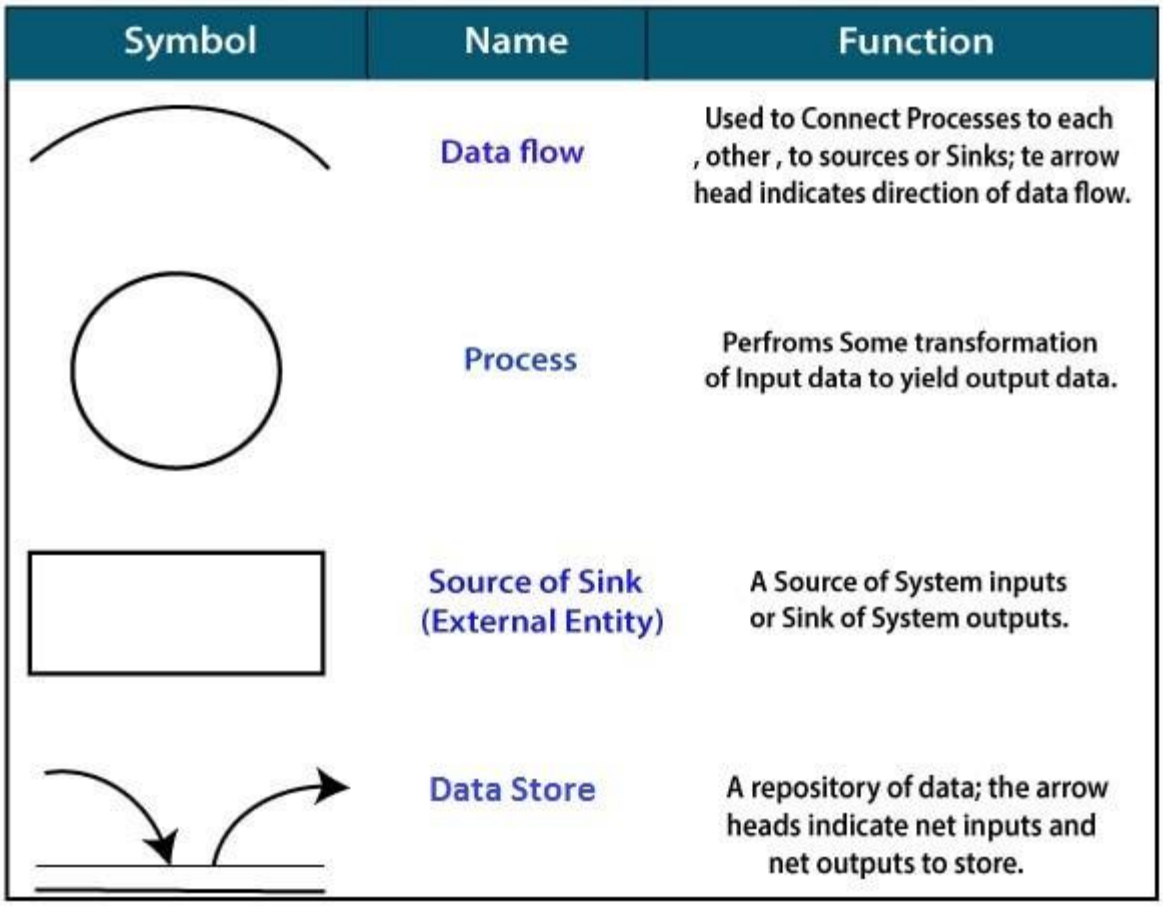
It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The following observations about DFDs are essential:

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown in fig:



Symbols for Data Flow Diagrams

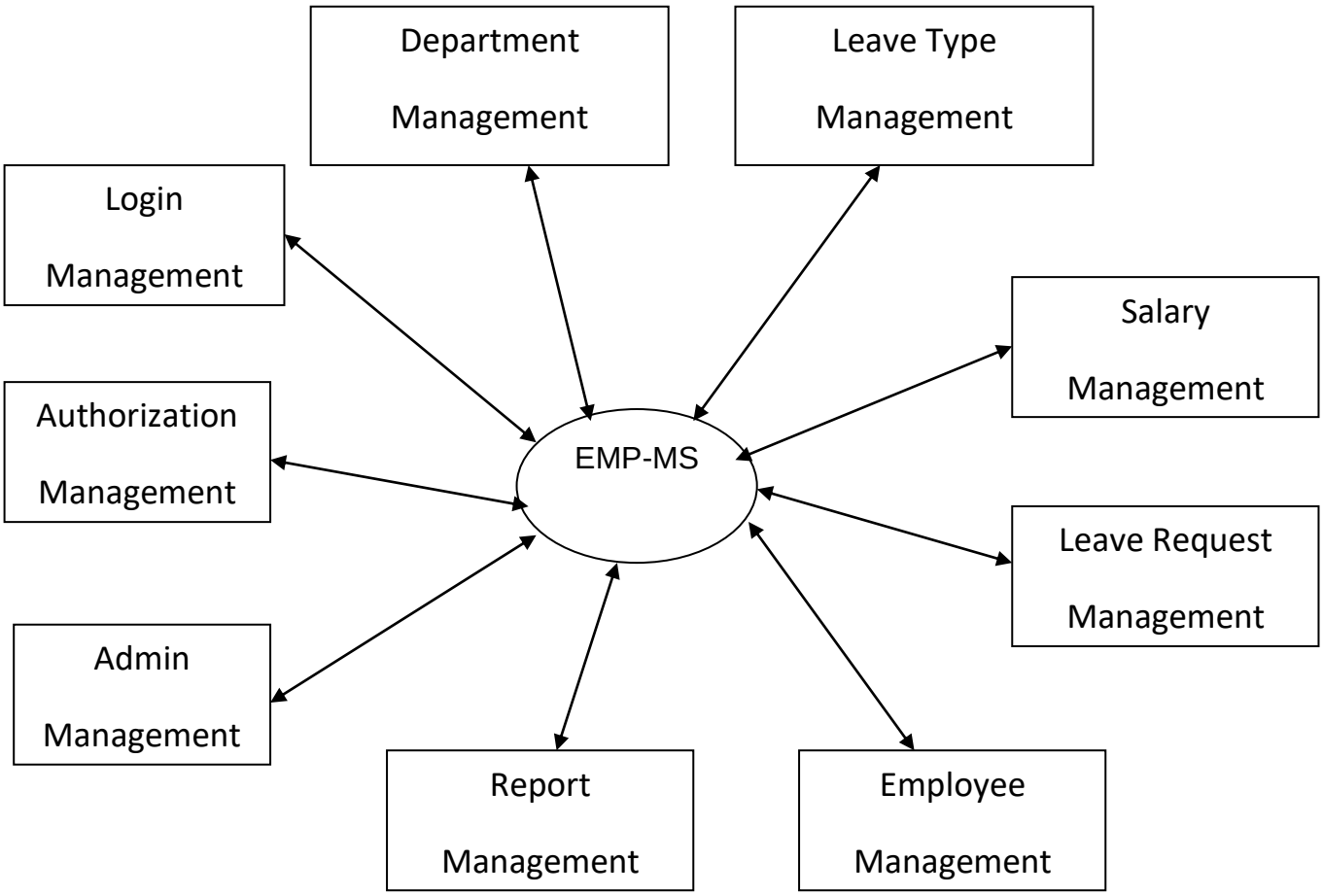
Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.

Data Flow: A curved line shows the flow of data into or out of a process or data store.

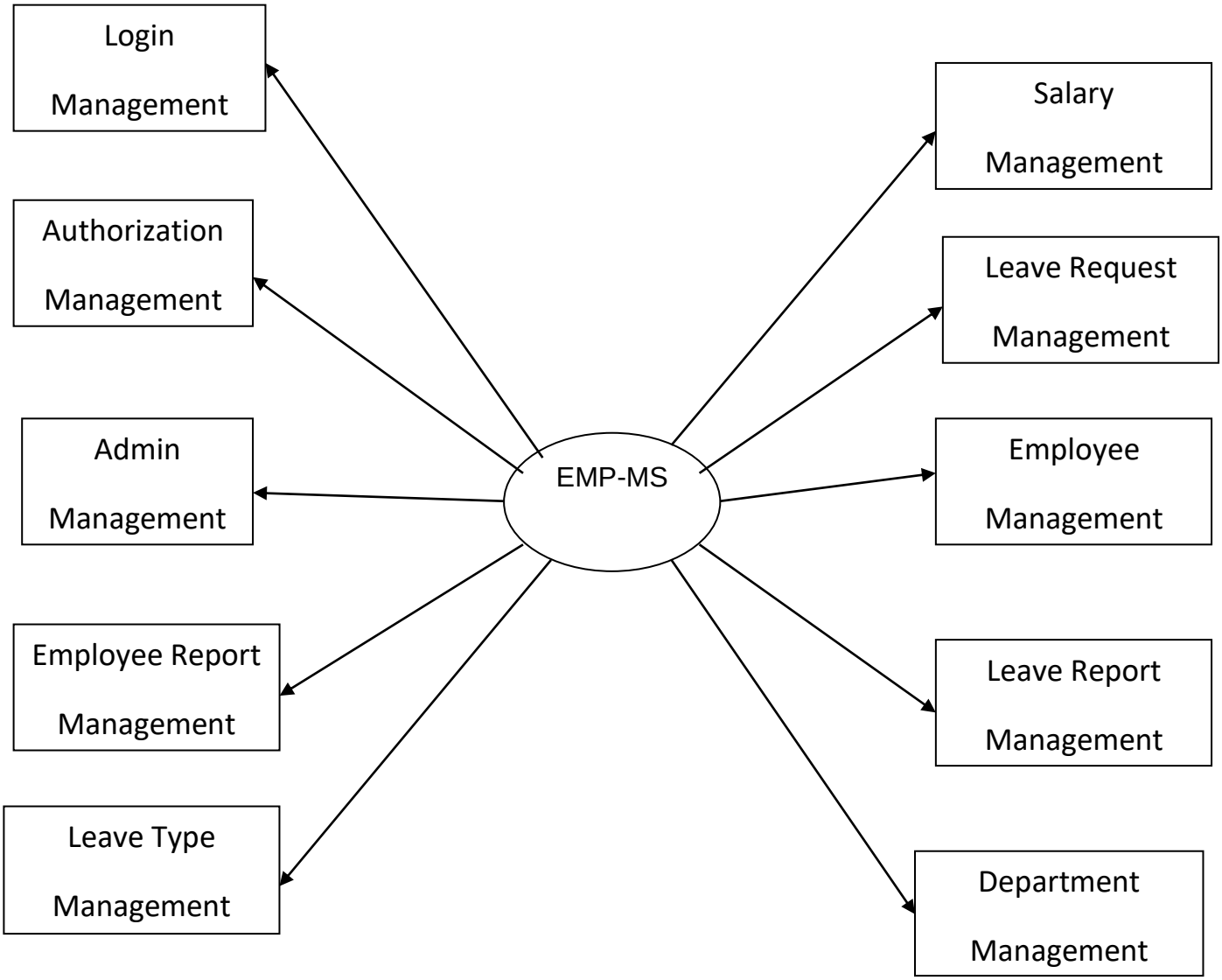
Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

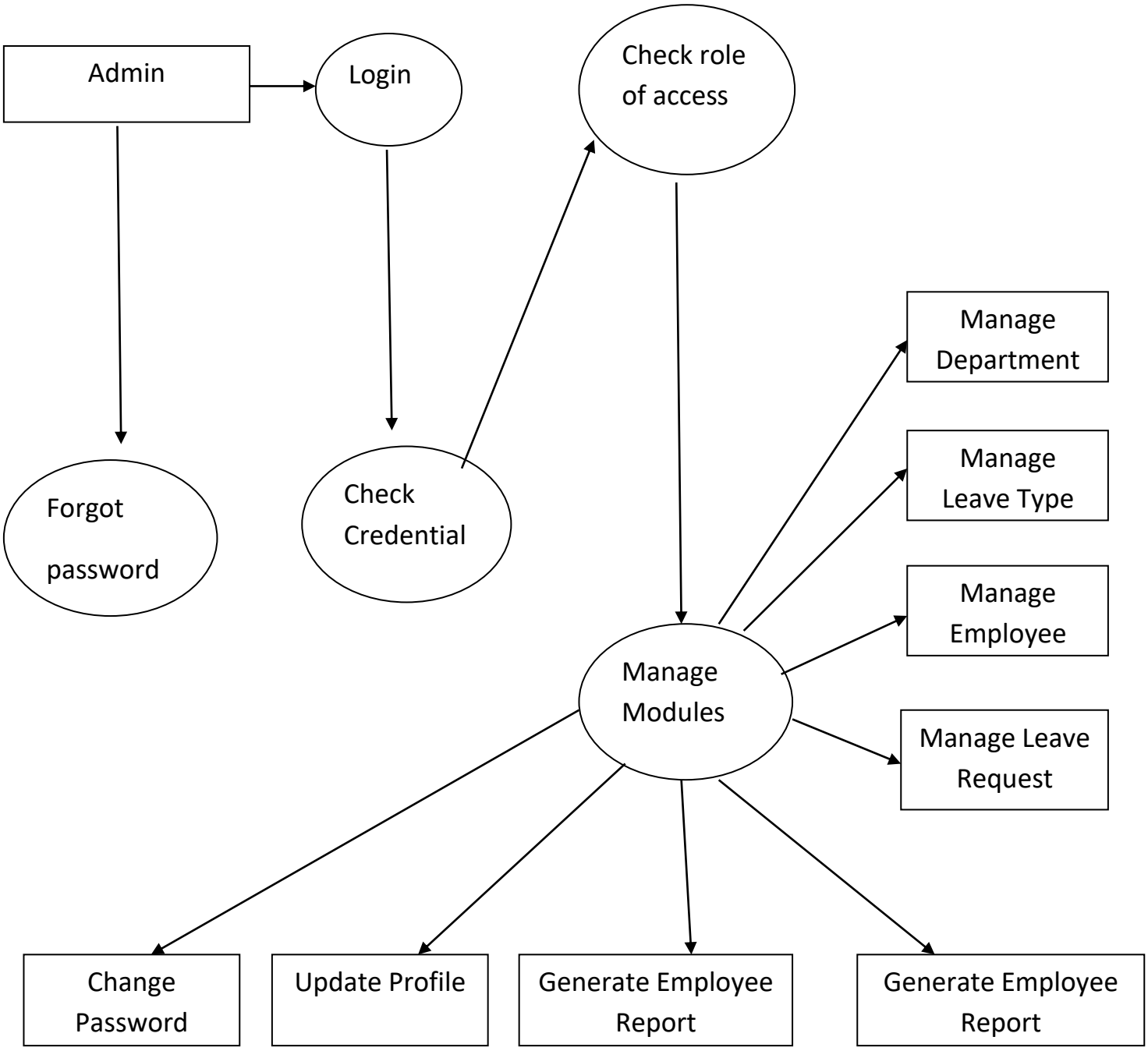
Zero Level DFD

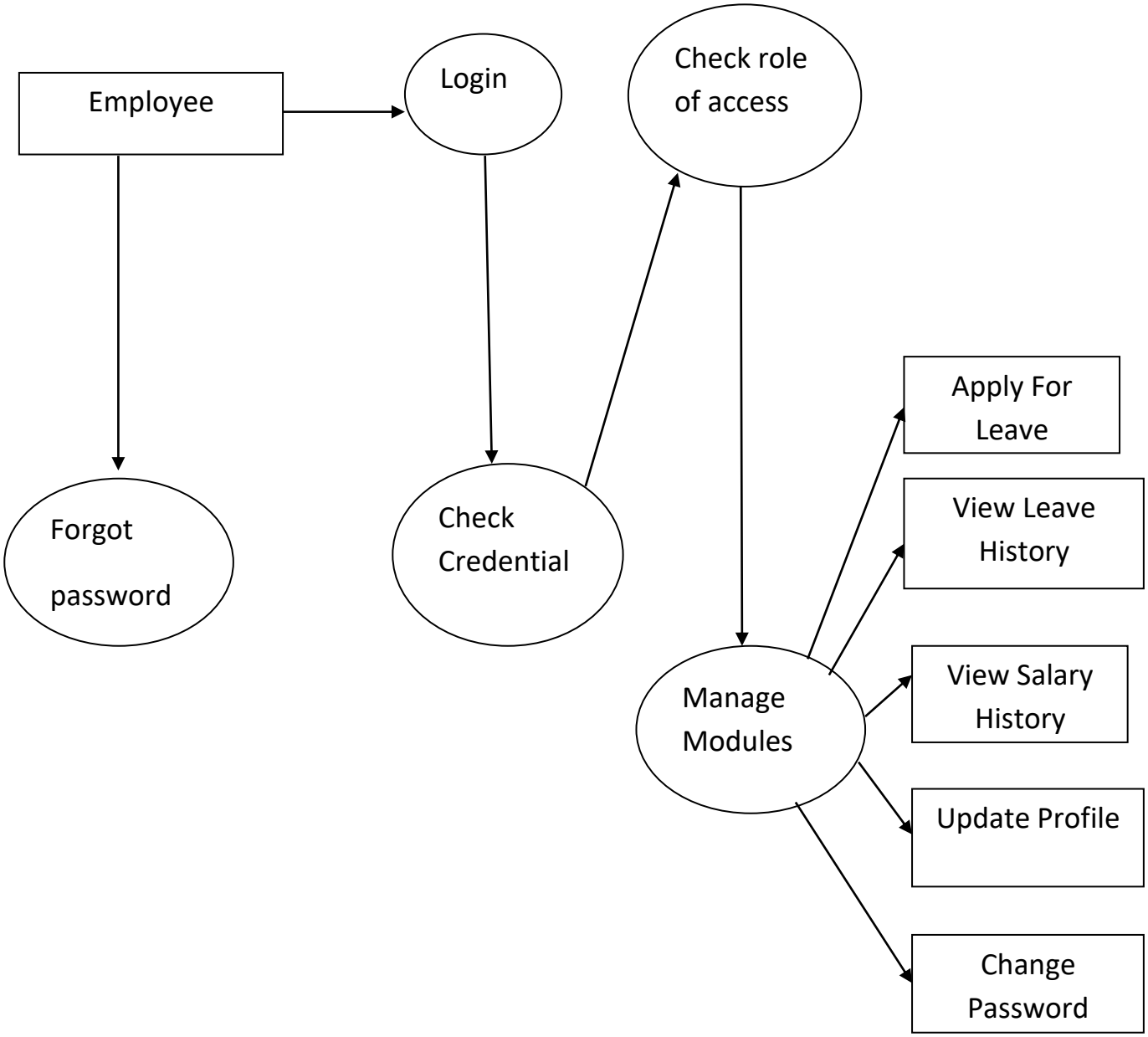


First Level DFD



Second Level DFD




Employee
Login
Check role of access
Forgot password
Check Credential
Apply For Leave
View Leave History
Manage Modules
View Salary History
Update Profile
Change Password

Database Design

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MySql database has been chosen for developing the relevant

databases.

EmployeeManagement System (EMS) contains 6 MySQL tables :

tbladmin table Structure : This table store the admin login and personal Details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**name**	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
3	**email**	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
4	**mobile**	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
5	**password**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
6	**create_date**	timestamp			Yes	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	1	A	No	

tbladdsalary table Structure : This table store the salary details of employee.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	Department	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
3	empid	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
4	salary	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
5	allowancesalary	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
6	total	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
7	create_date	timestamp			Yes	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	1	A	No	

tbldepartment table Structure : This table store the name department.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	DepartmentName	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	4	A	No	

tblemployee table Structure : This table store the employees details.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	EmpId	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
3	fname	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
4	lname	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
5	department_name	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
6	email	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
7	mobile	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
8	country	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
9	state	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
10	city	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
11	address	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
12	photo	varchar(200)	utf8mb4_general_ci		Yes	*NULL*		
13	dob	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
14	date_of_joining	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
15	password	varchar(450)	utf8mb4_general_ci		Yes	*NULL*		
16	create_date	timestamp			Yes	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	

tblleave Structure : This table store the employee leave.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	userID	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
3	EmpID	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
4	LeaveType	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
5	FromDate	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
6	Todate	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
7	Description	varchar(450)	utf8mb4_general_ci		Yes	*NULL*		
8	status	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
9	adminremarks	varchar(450)	utf8mb4_general_ci		Yes	*NULL*		
10	Create_date	timestamp			Yes	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	3	A	No	

tblleavetype Structure : This table store the name of leave type.

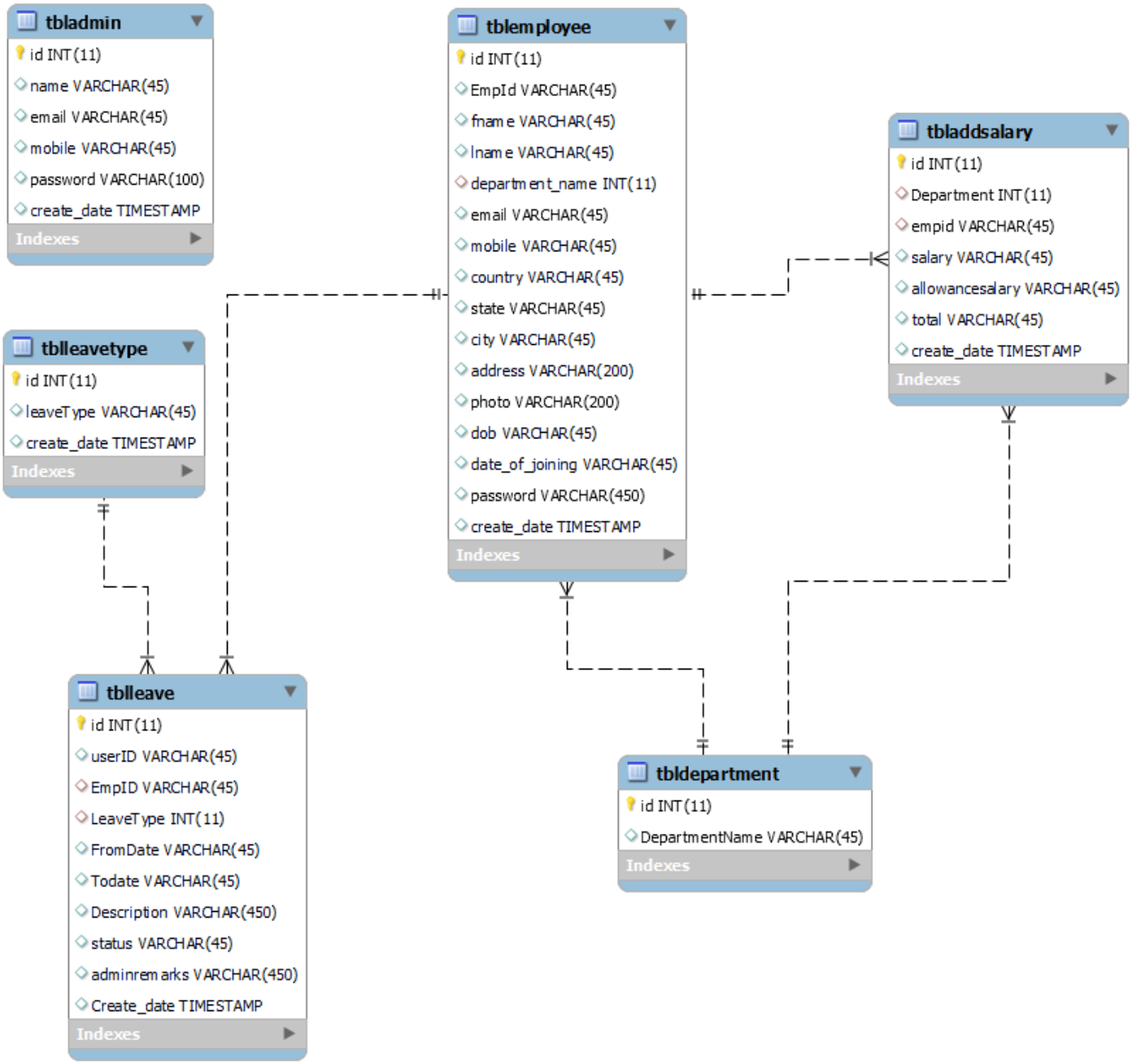
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**leaveType**	varchar(45)	utf8mb4_general_ci		Yes	*NULL*		
3	**create_date**	timestamp			Yes	current_timestamp()		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	

Class Diagram:

The class diagram shows a set of classes, interfaces, collaborations and their relationships.


tbladmin
id INT(11)
name VARCHAR(45)
email VARCHAR(45)
mobile VARCHAR(45)
password VARCHAR(100)
create_date TIMESTAMP
Indexes
tblemployee
id INT(11)
EmpId VARCHAR(45)
fname VARCHAR(45)
lname VARCHAR(45)
department_name INT(11)
email VARCHAR(45)
mobile VARCHAR(45)
country VARCHAR(45)
state VARCHAR(45)
city VARCHAR(45)
address VARCHAR(200)
photo VARCHAR(200)
dob VARCHAR(45)
date_of_joining VARCHAR(45)
password VARCHAR(450)
create_date TIMESTAMP
Indexes
tbladdsalary
id INT(11)
Department INT(11)
empid VARCHAR(45)
salary VARCHAR(45)
allowancesalary VARCHAR(45)
total VARCHAR(45)
create_date TIMESTAMP
Indexes
tblleavetype
id INT(11)
leaveType VARCHAR(45)
create_date TIMESTAMP
Indexes
tblleave
id INT(11)
userID VARCHAR(45)
EmpID VARCHAR(45)
LeaveType INT(11)
FromDate VARCHAR(45)
Todate VARCHAR(45)
Description VARCHAR(450)
status VARCHAR(45)
adminremarks VARCHAR(450)
Create_date TIMESTAMP
Indexes
tbldepartment
id INT(11)
DepartmentName VARCHAR(45)
Indexes

System Testing

SOFTWARE TESTING TECHNIQUES:

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

TESTING OBJECTIVES:

1. Testing is process of executing a program with the intent of finding an error.

2. A good test case design is one that has a probability of finding an as yet undiscovered error.

3. A successful test is one that uncovers an as yet undiscovered error.

These above objectives imply a dramatic change in view port.

Testing cannot show the absence of defects, it can only show that software errors are present.

There are three types of testing strategies

1. Unit test

2. Integration test
3. Performance test

Unit Testing:

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

Integration Testing:

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

Performance Testing:

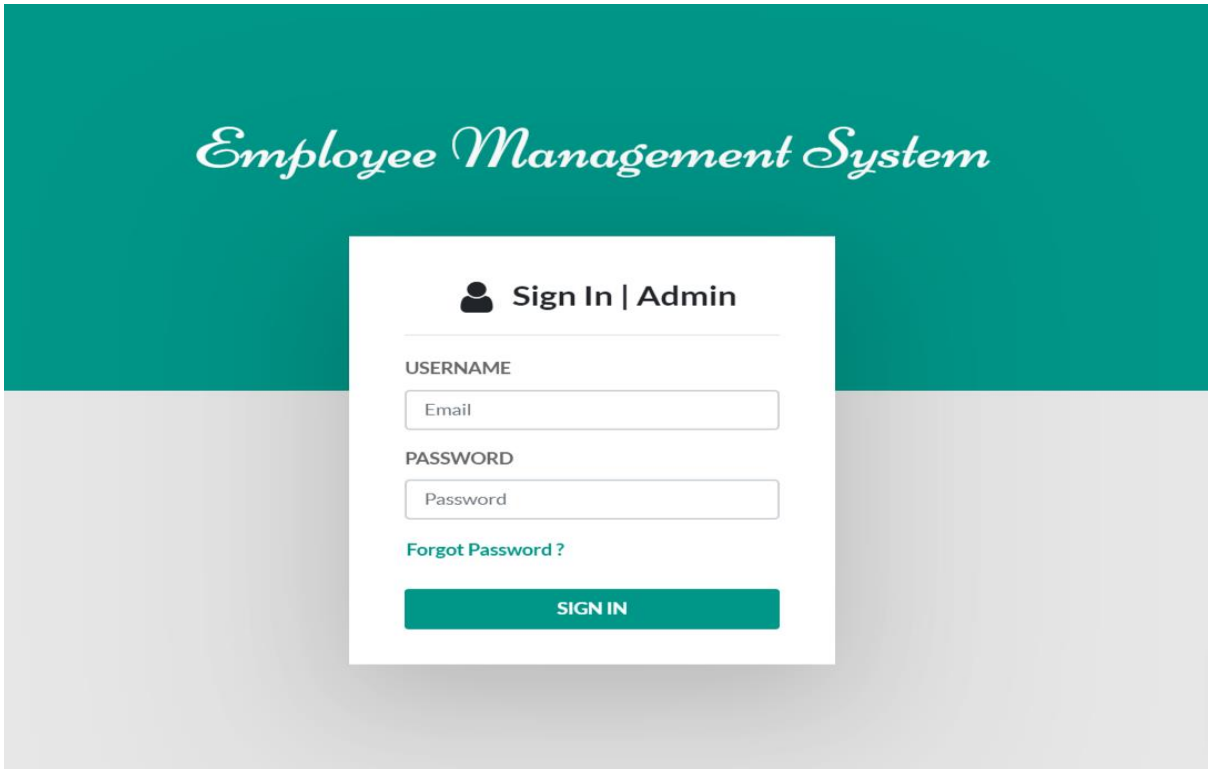
Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

Output Screen of Project

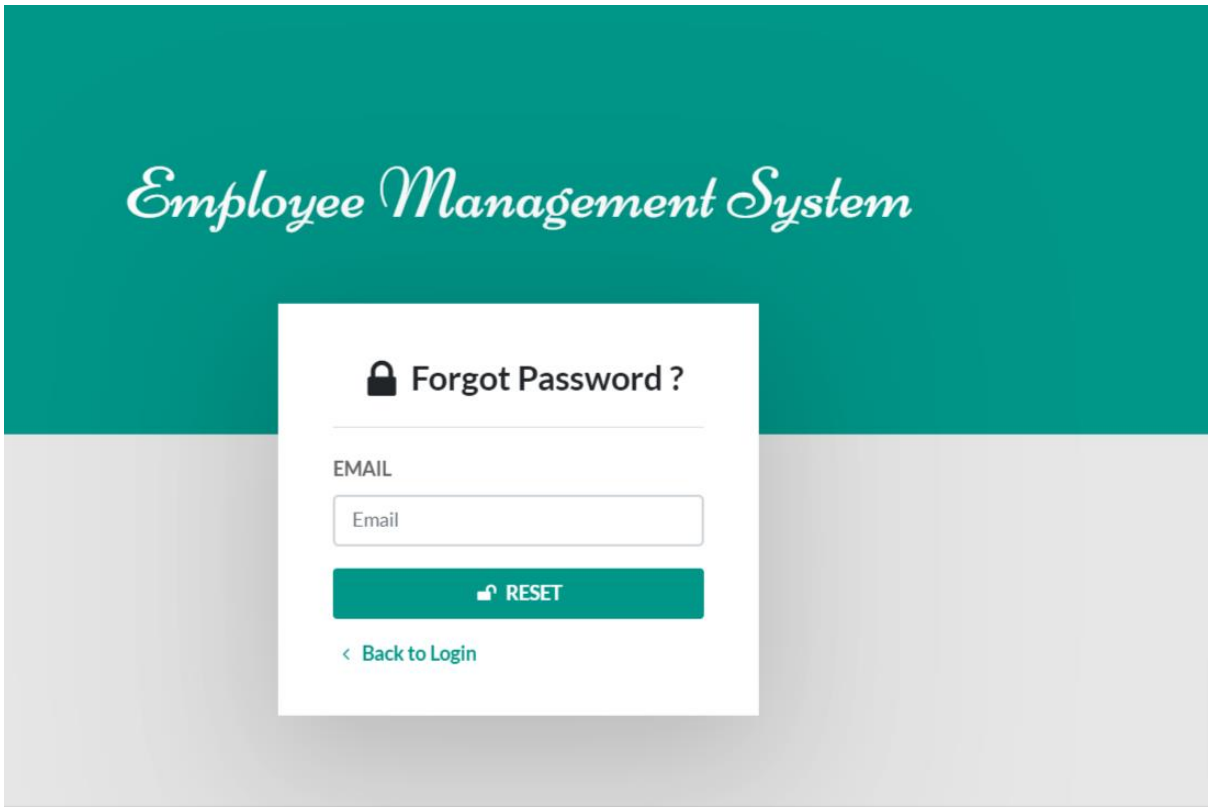
Home Page

Admin Panel

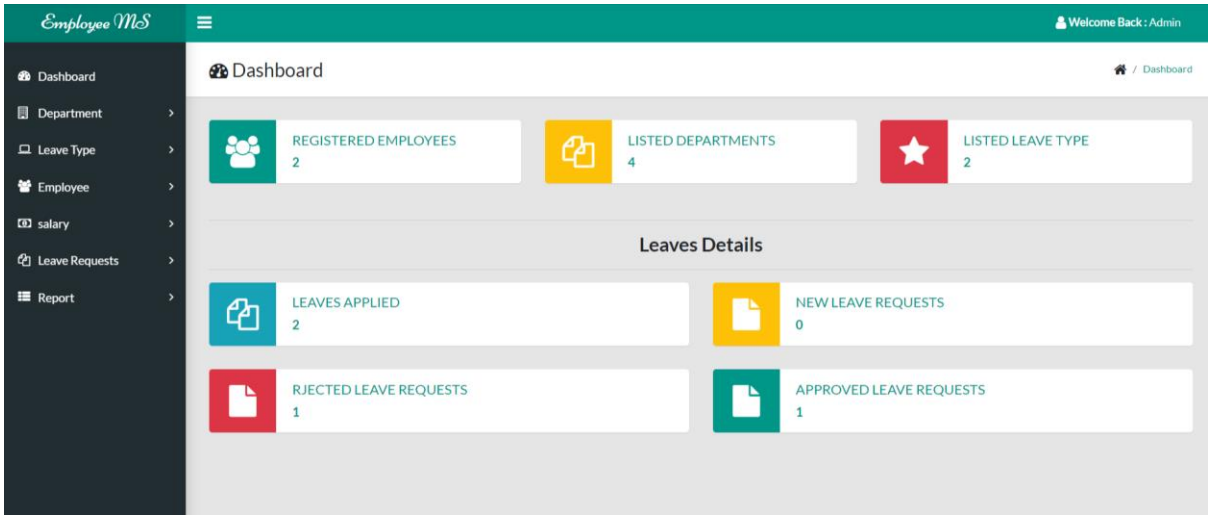
Login Page



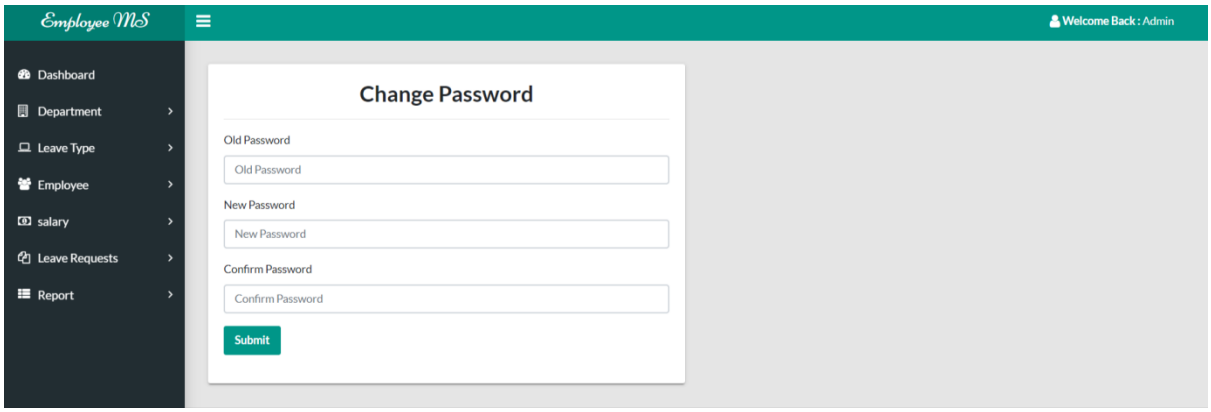
Forgot Password



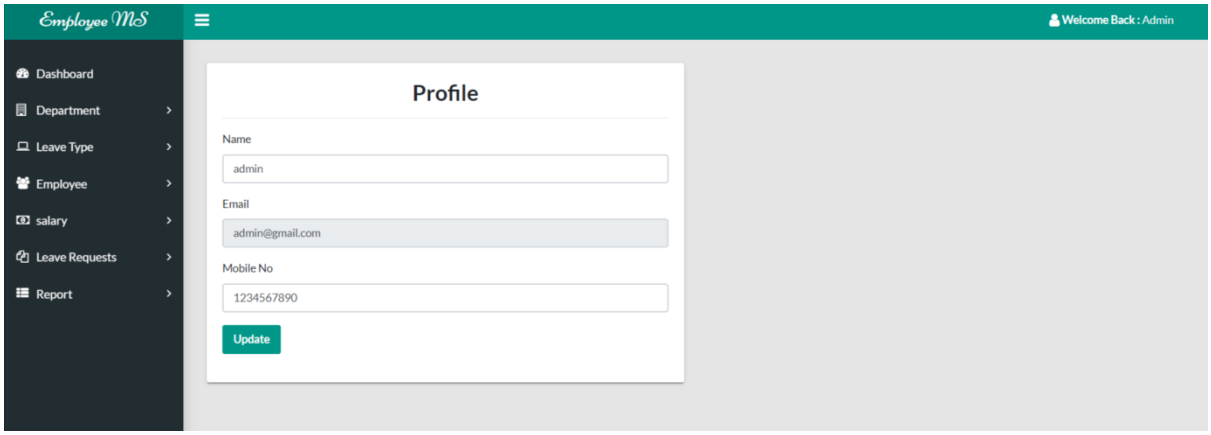
Dashboard



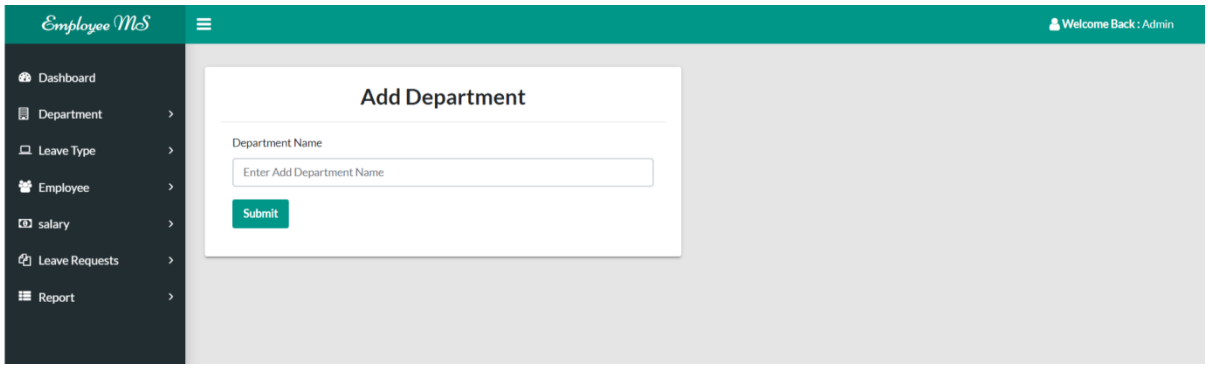
Change Password



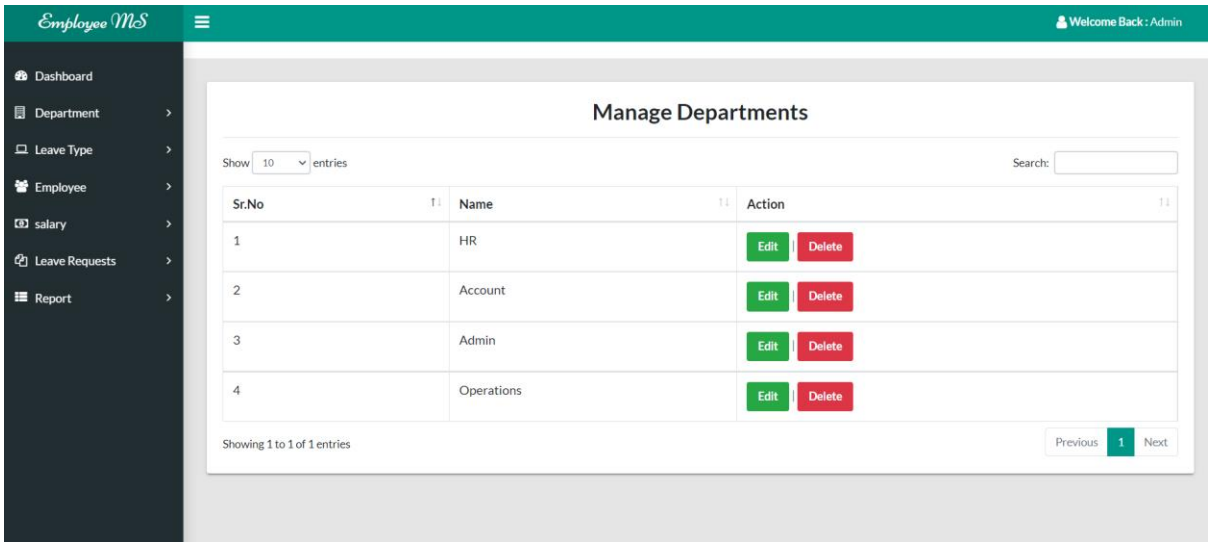
Admin Profile



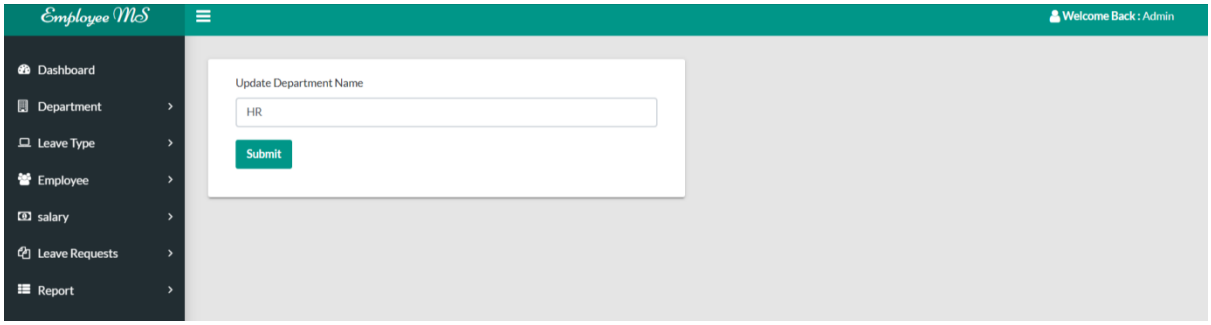
Add Department



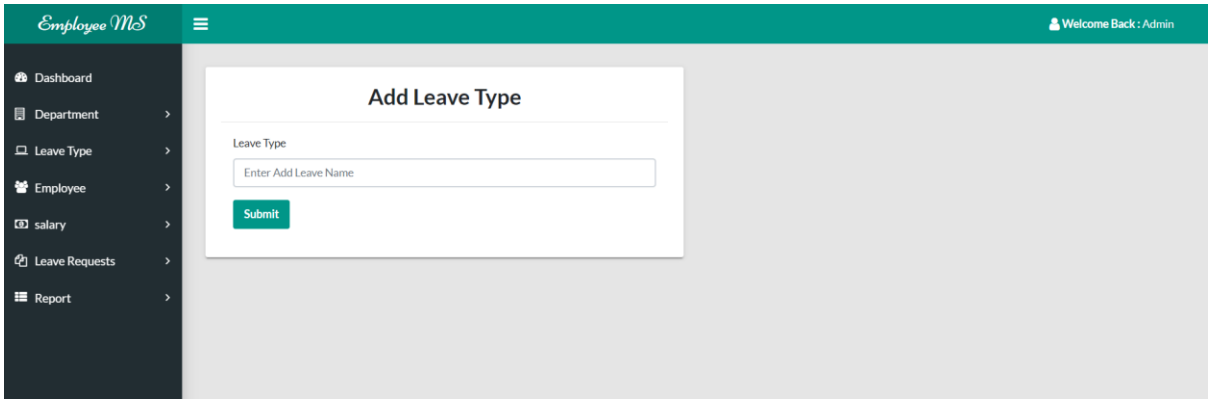
Manage Department



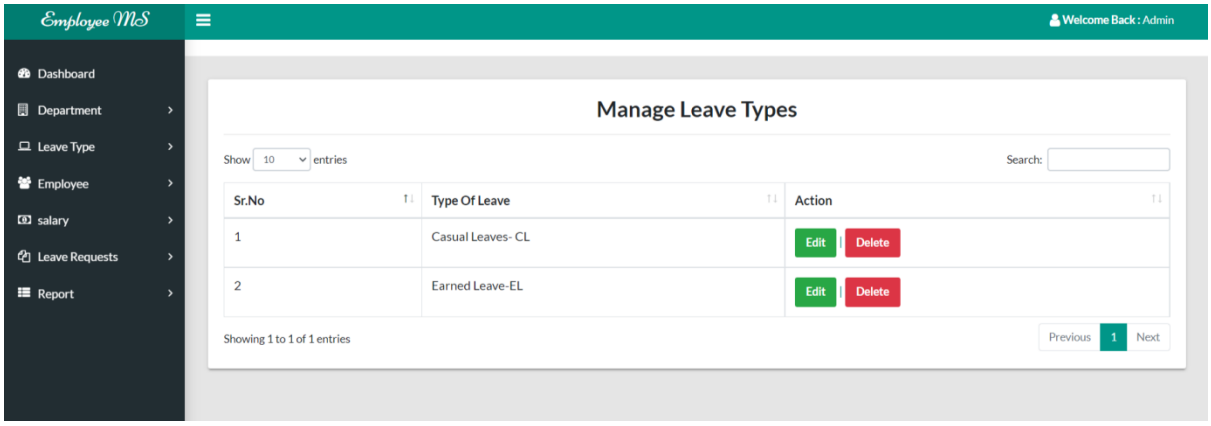
Update Department



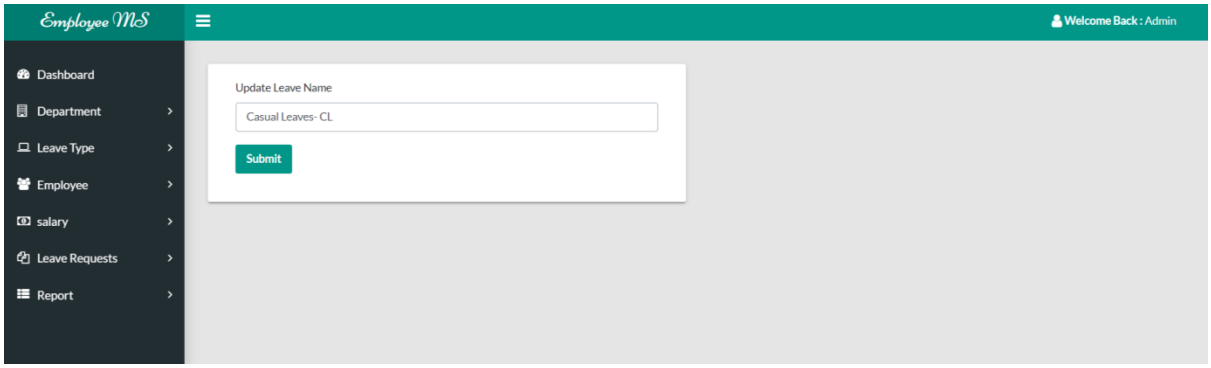
Add Leave Type



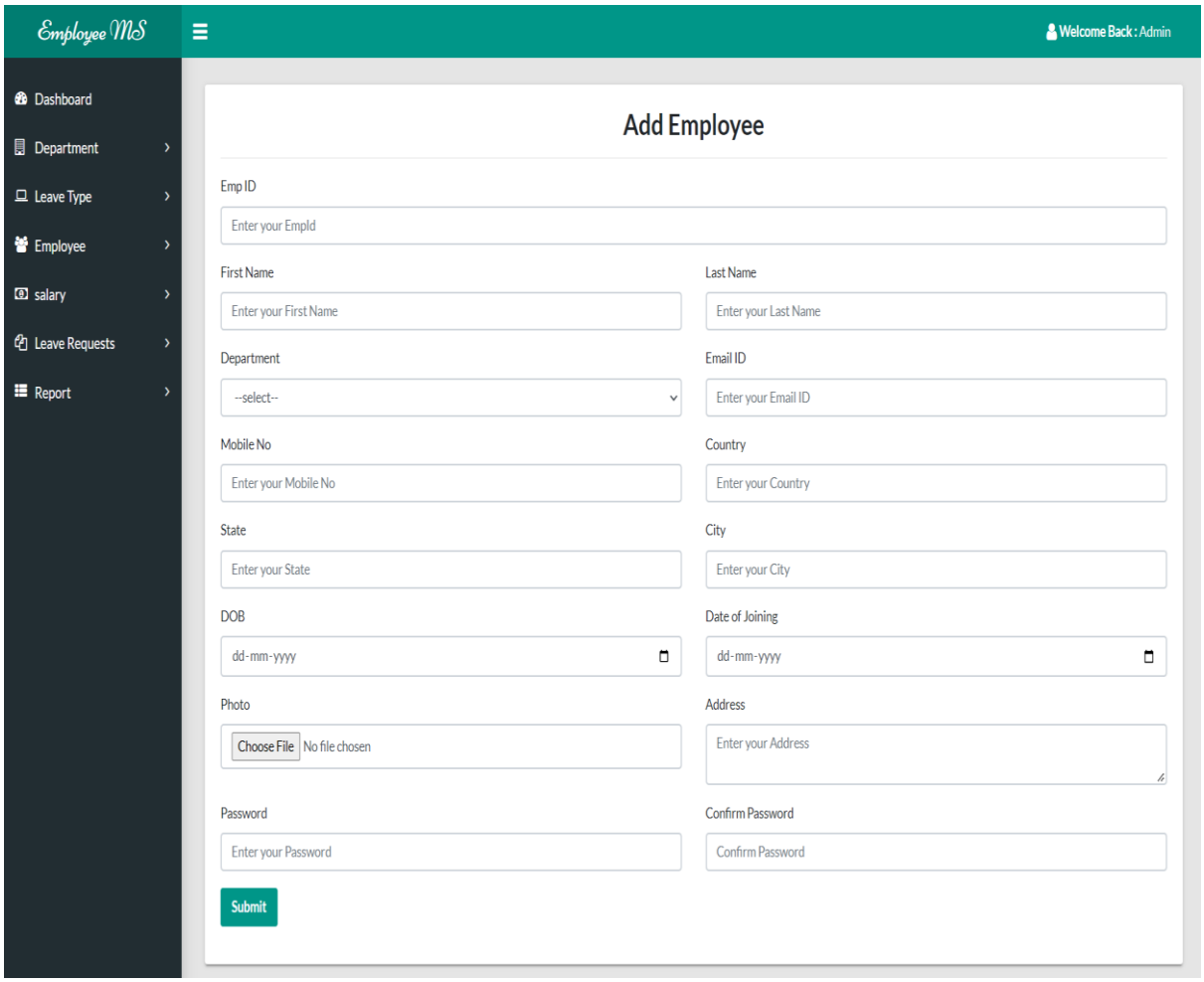
Manage Leave Type



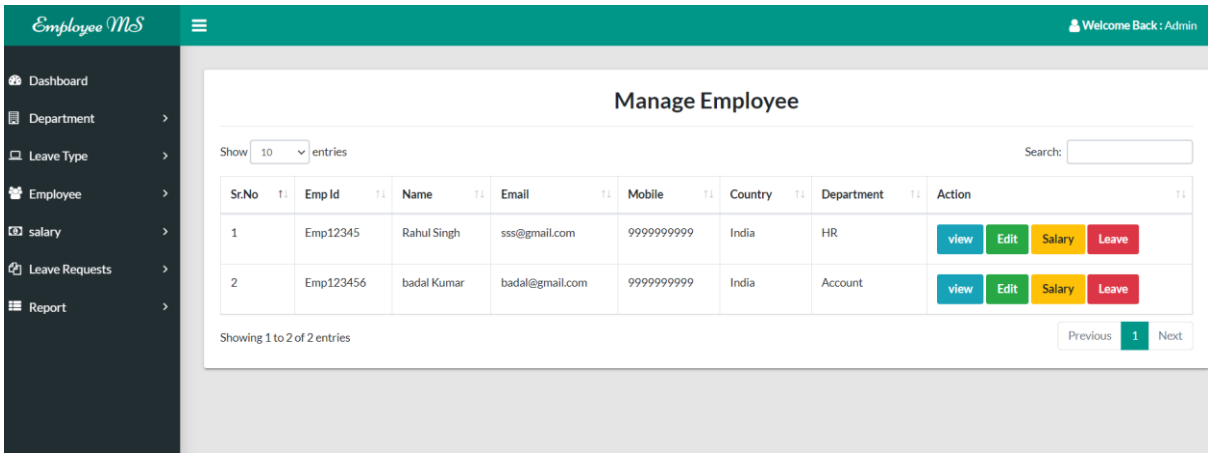
Update Leave Type



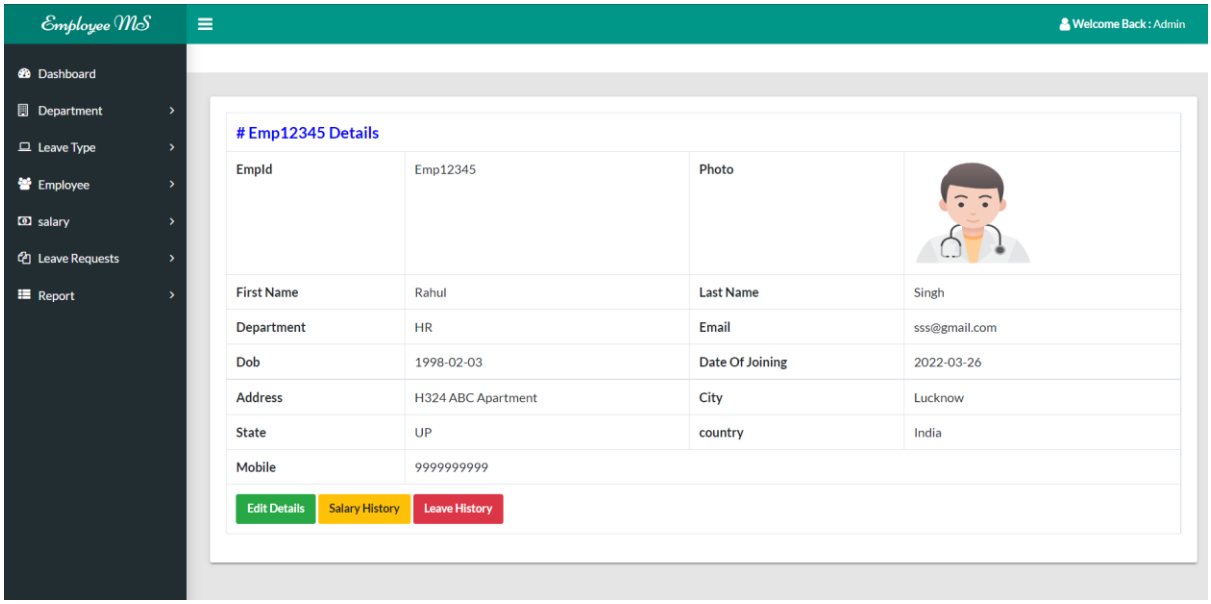
Add Employee



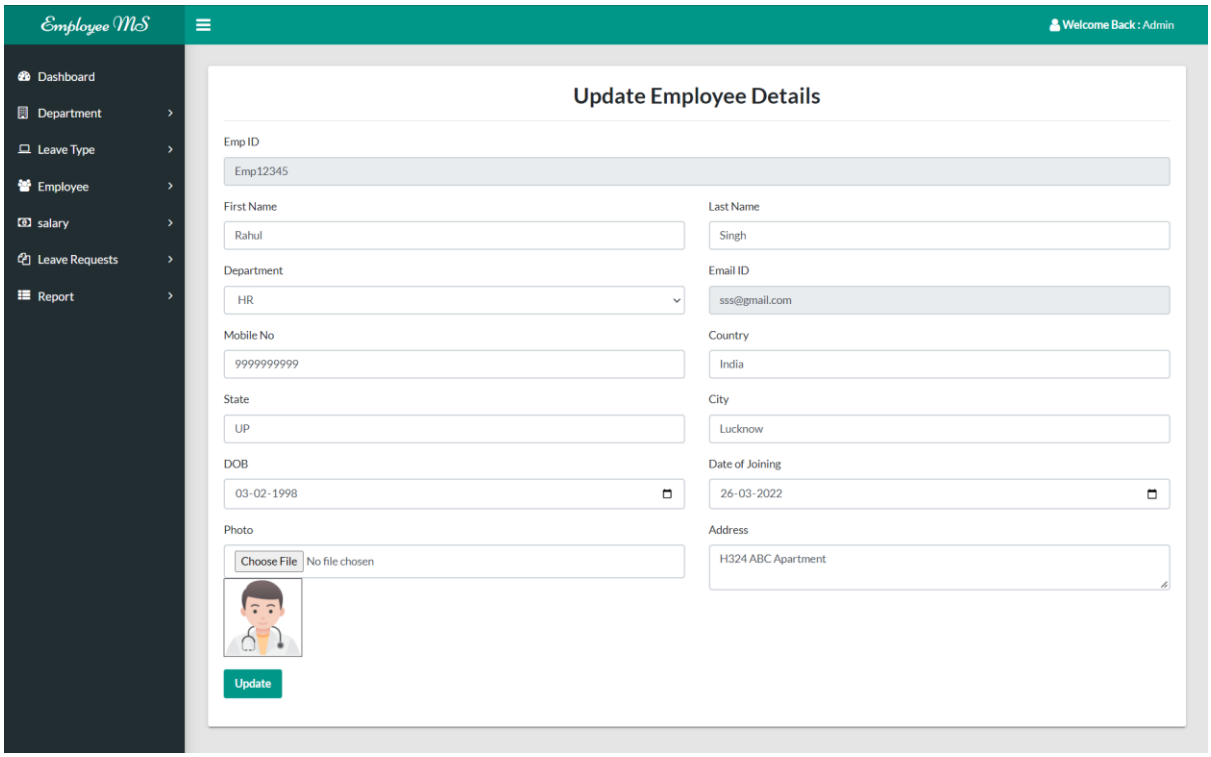
Manage Employee



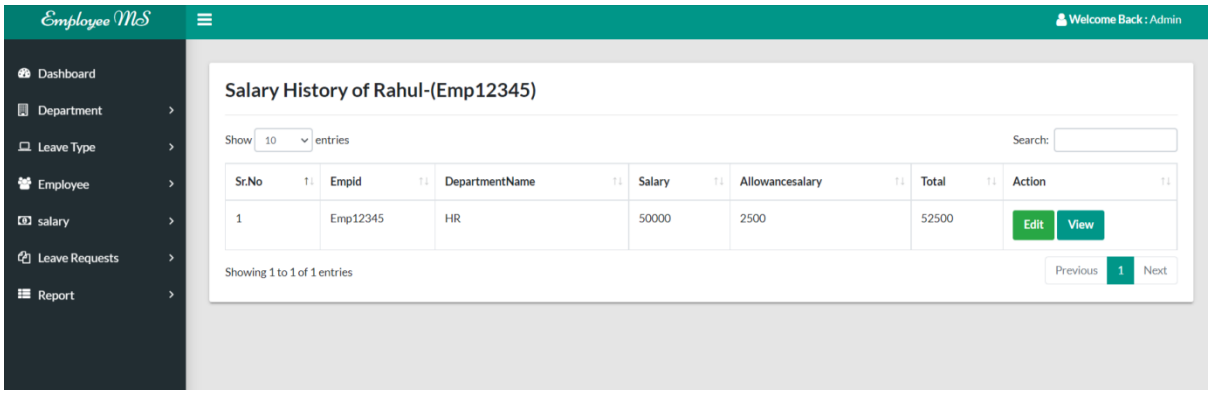
View Employee Details



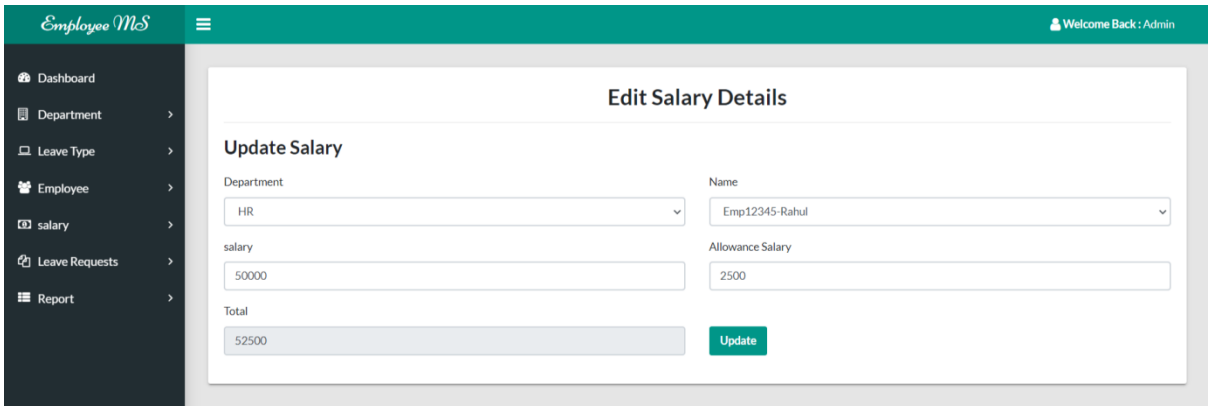
Update Employee Details



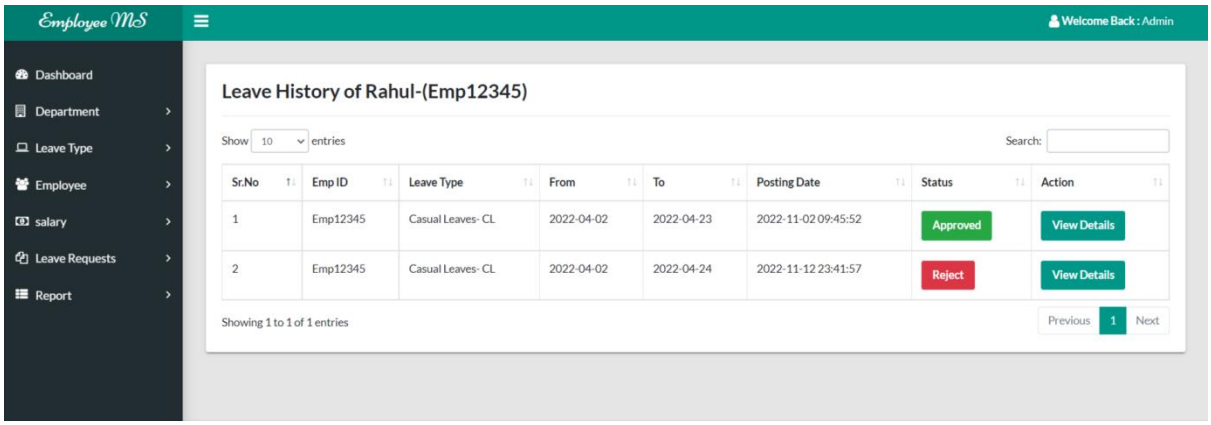
View Salary History



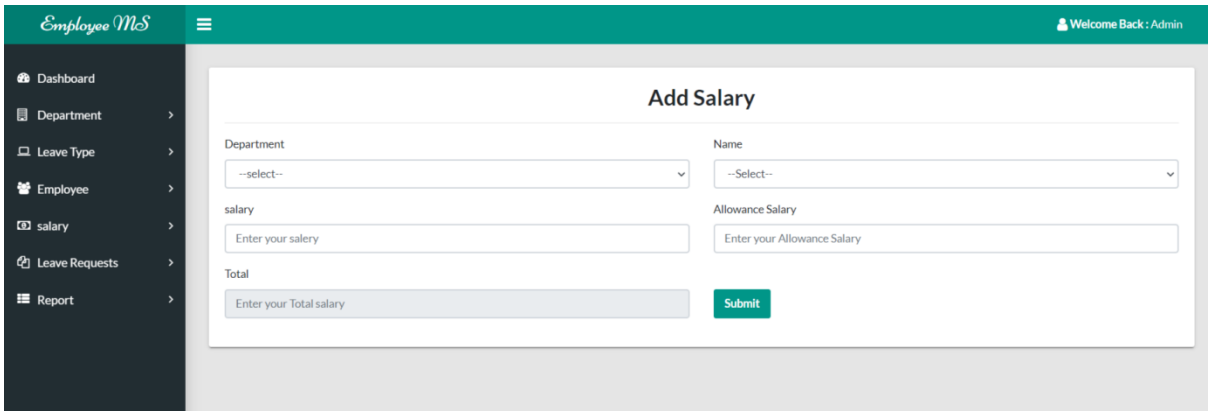
Update Salary



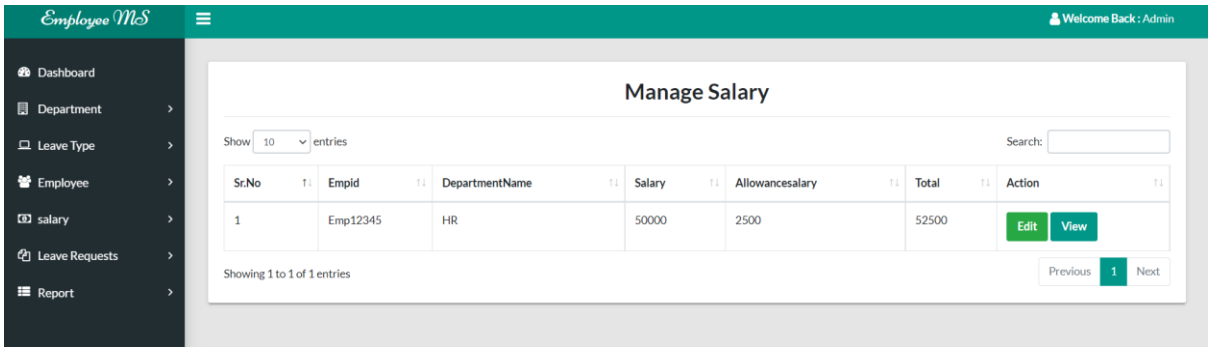
View Leave History



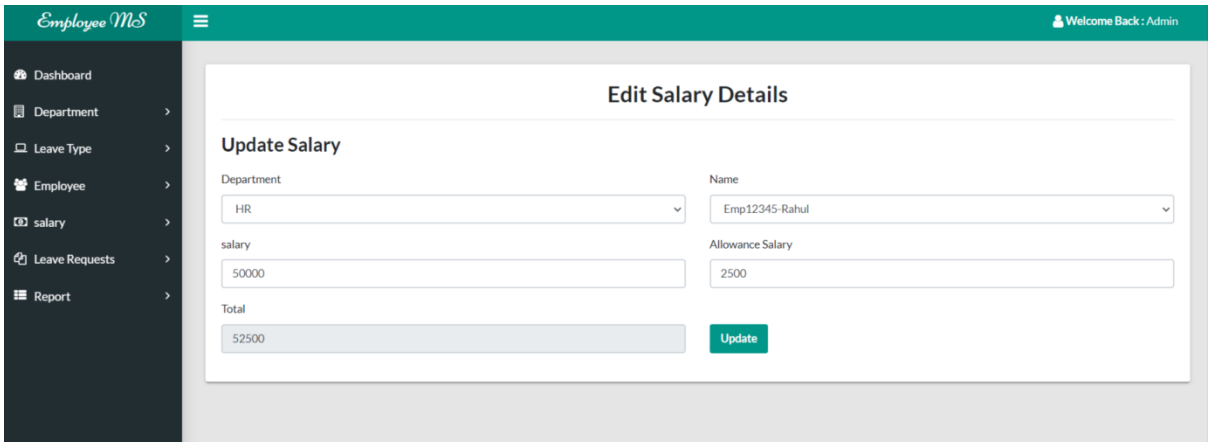
Add Salary



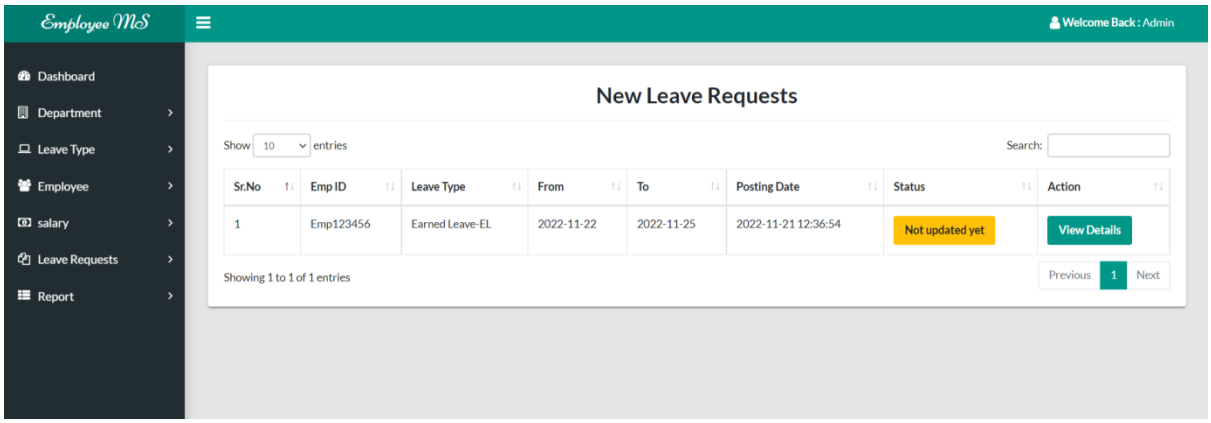
Manage Salary



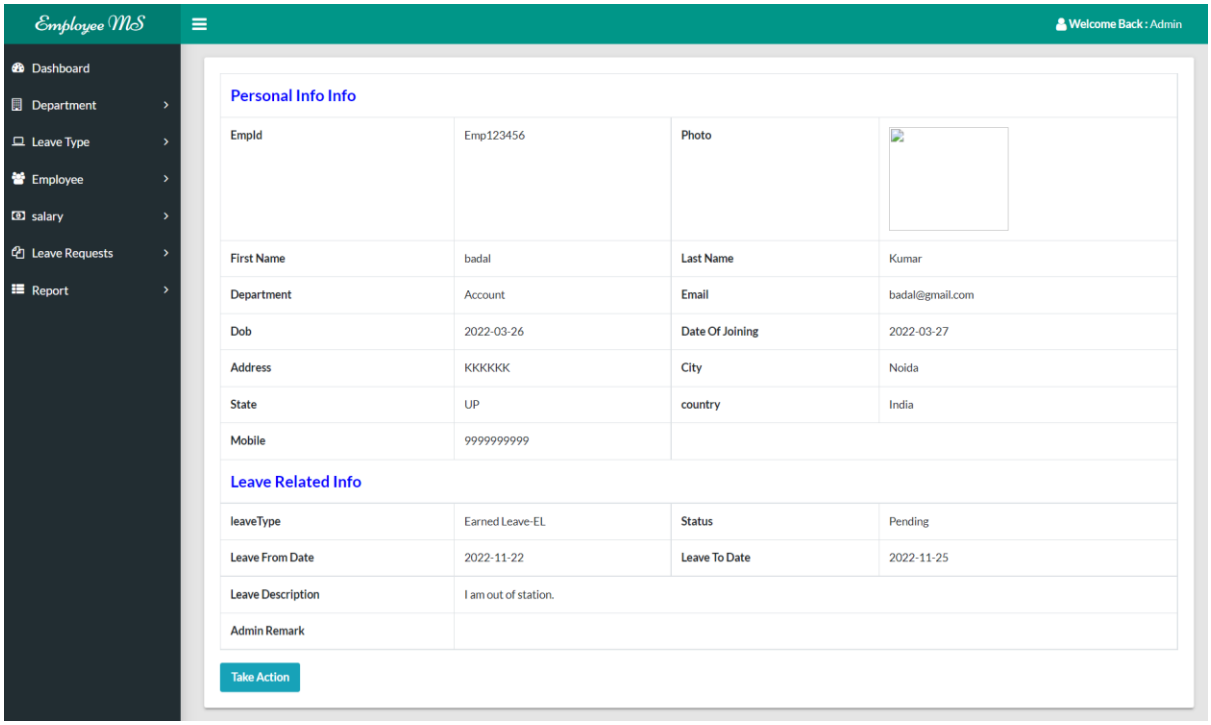
Update Salary



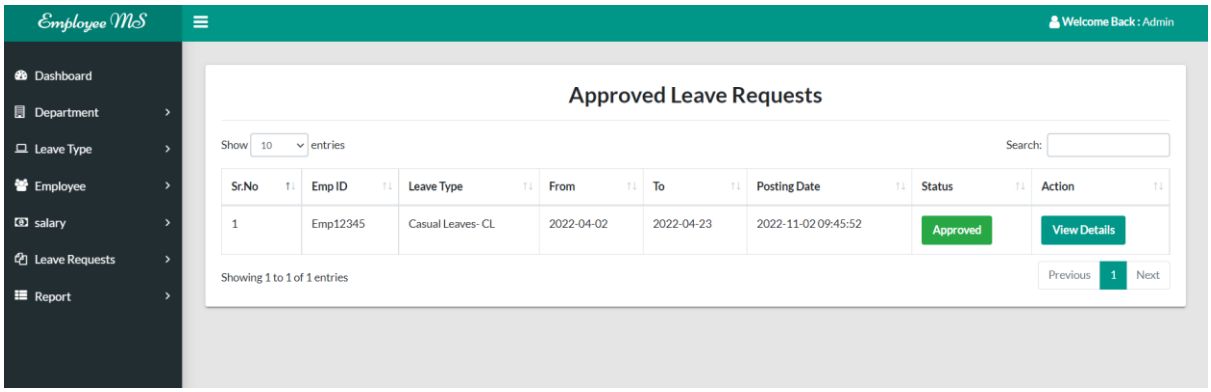
New Leave Request



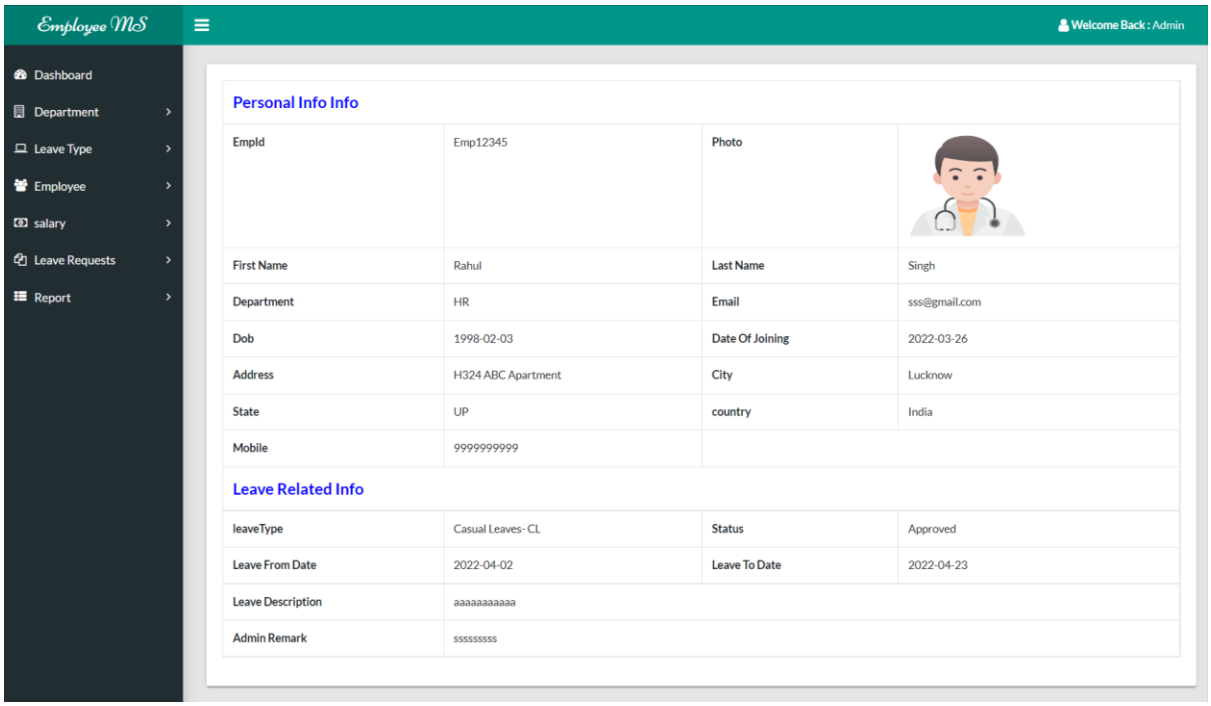
View New Leave Request



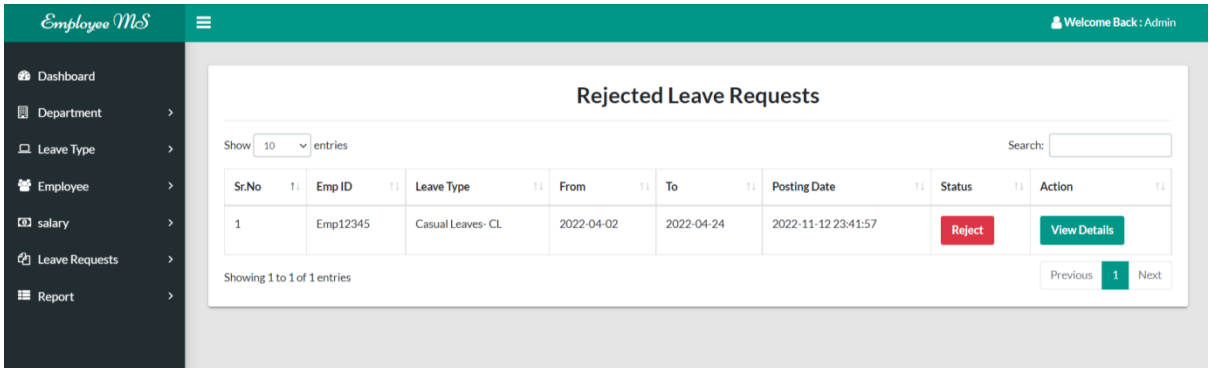
Accepted Leave Request



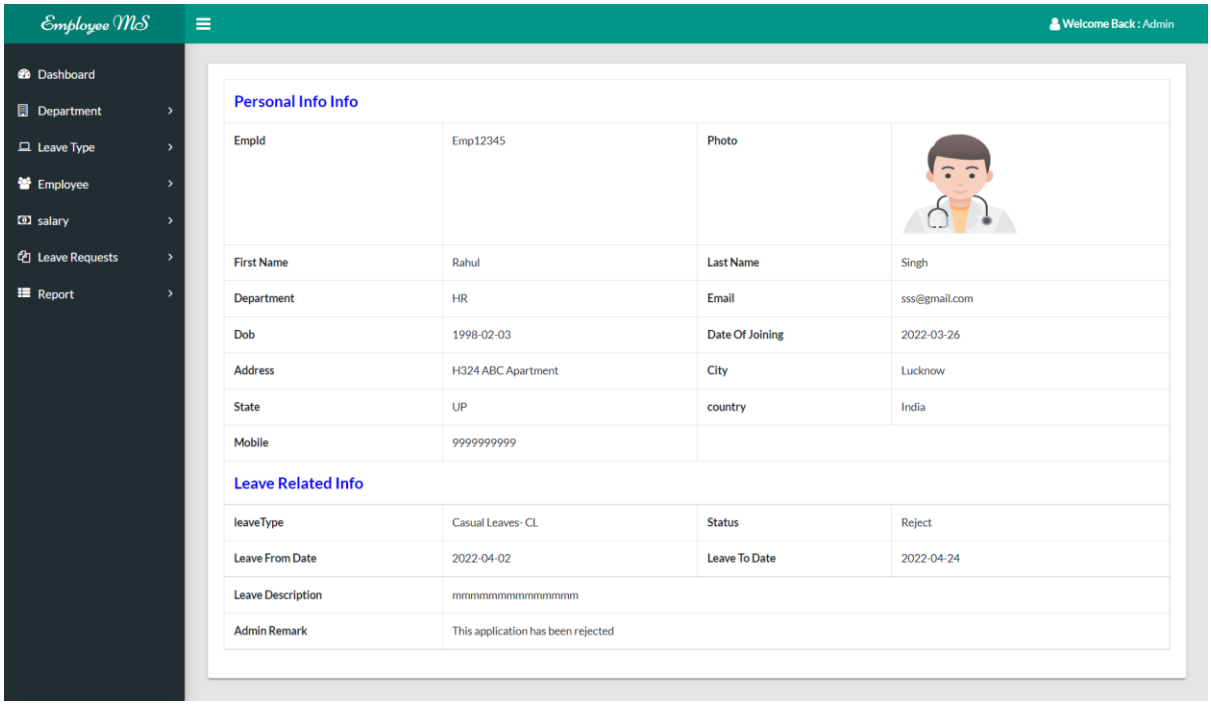
View Accepted Request



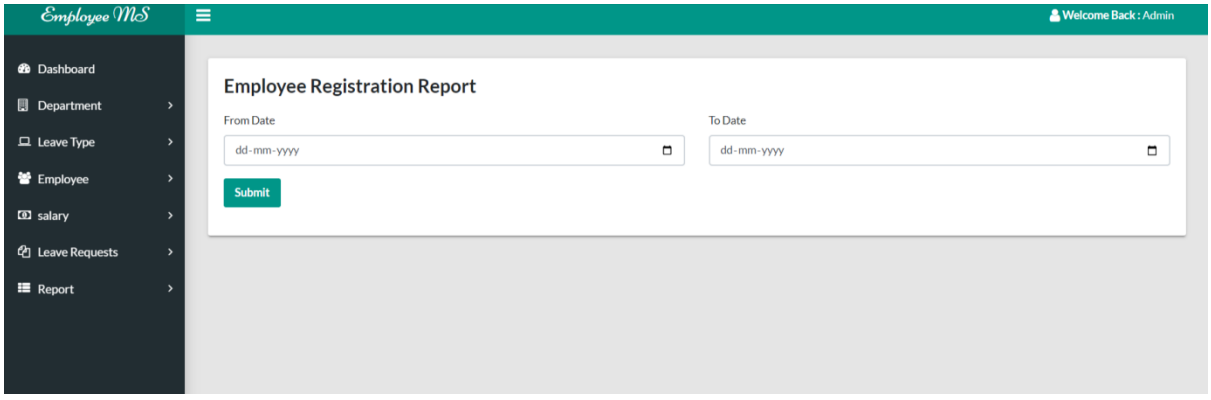
Rejected Leave Request



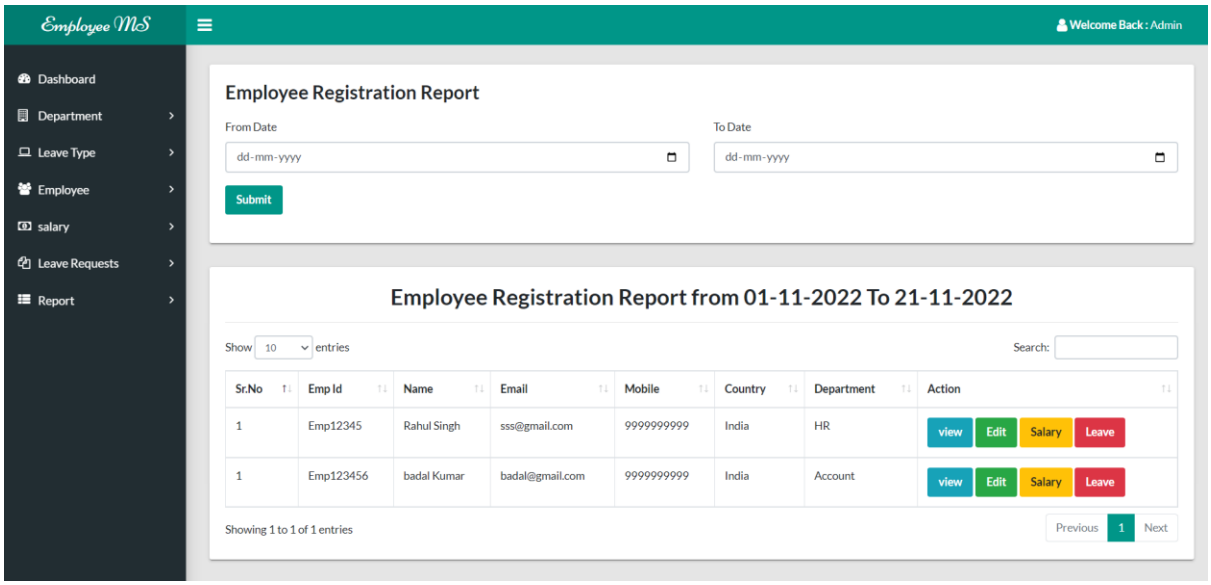
View Rejected Leave Request



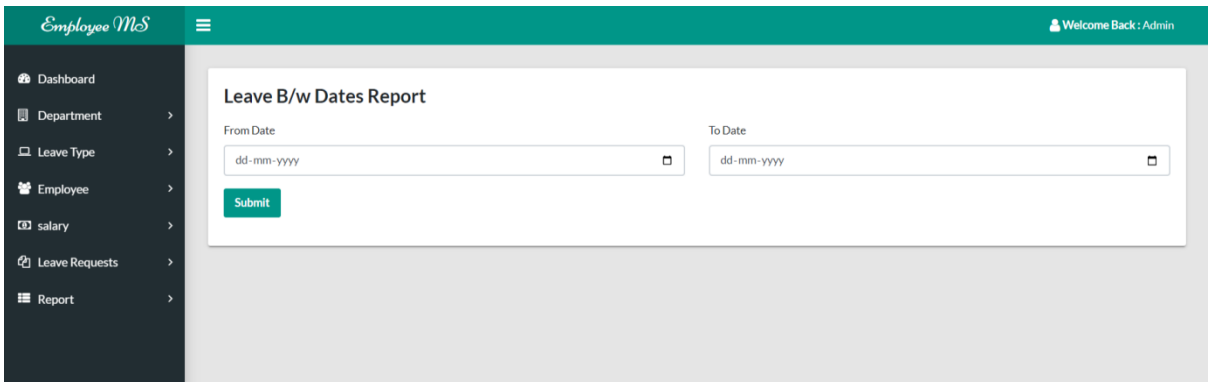
Employee Report



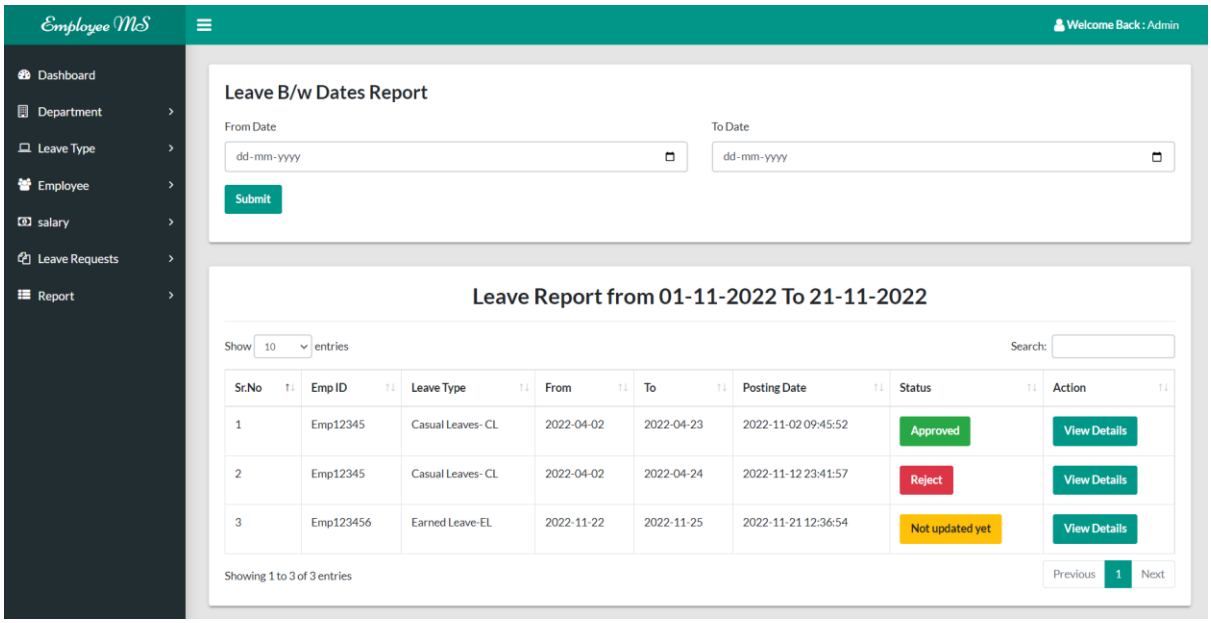
View Employee Report



Leave Report

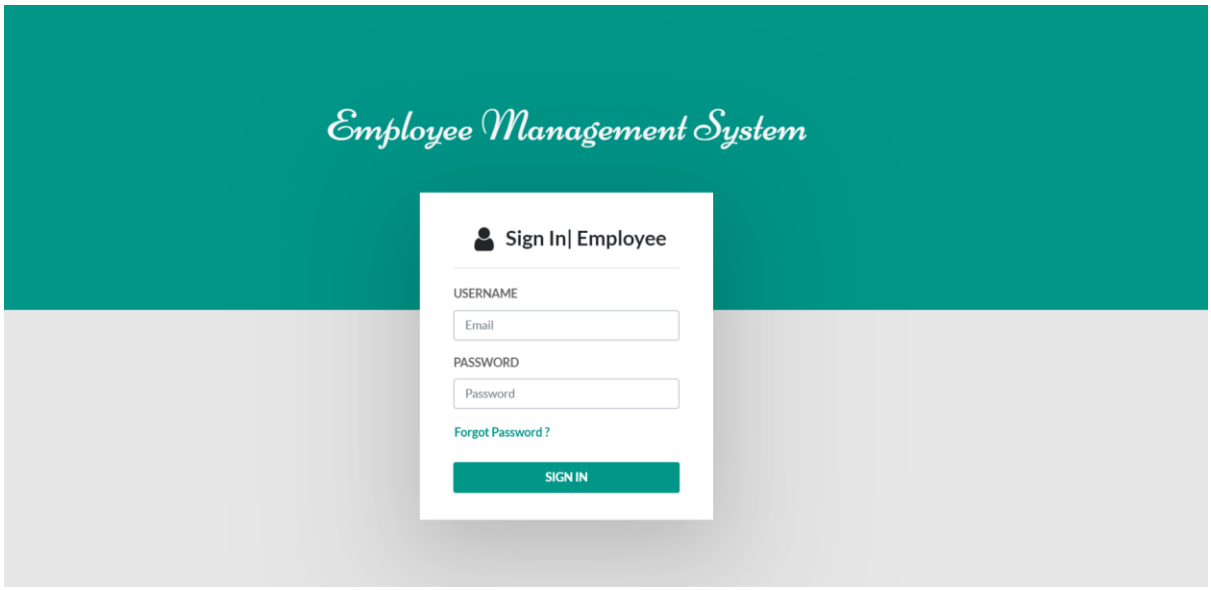


View Leave Report

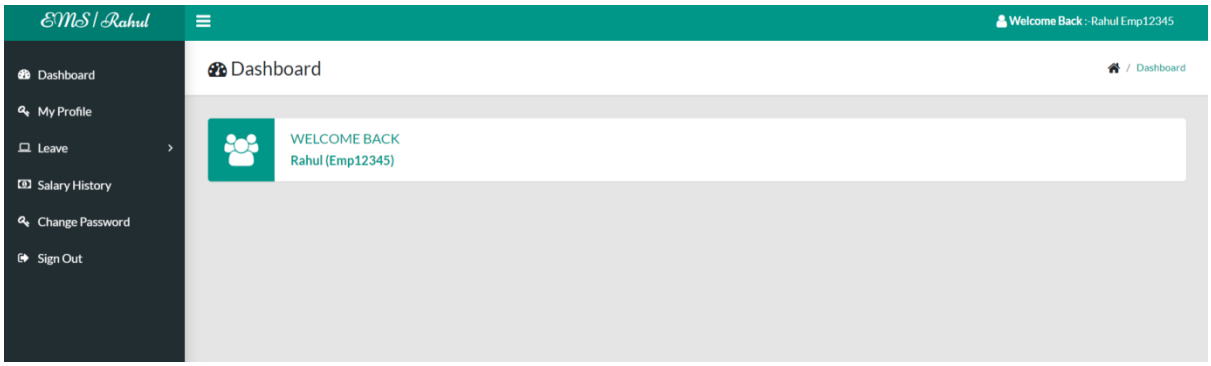


Employee Panel

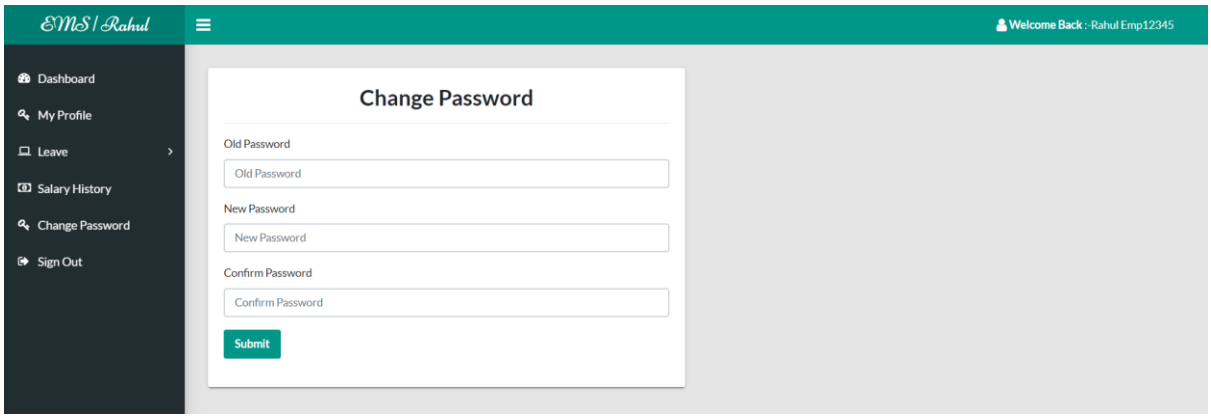
Login Page



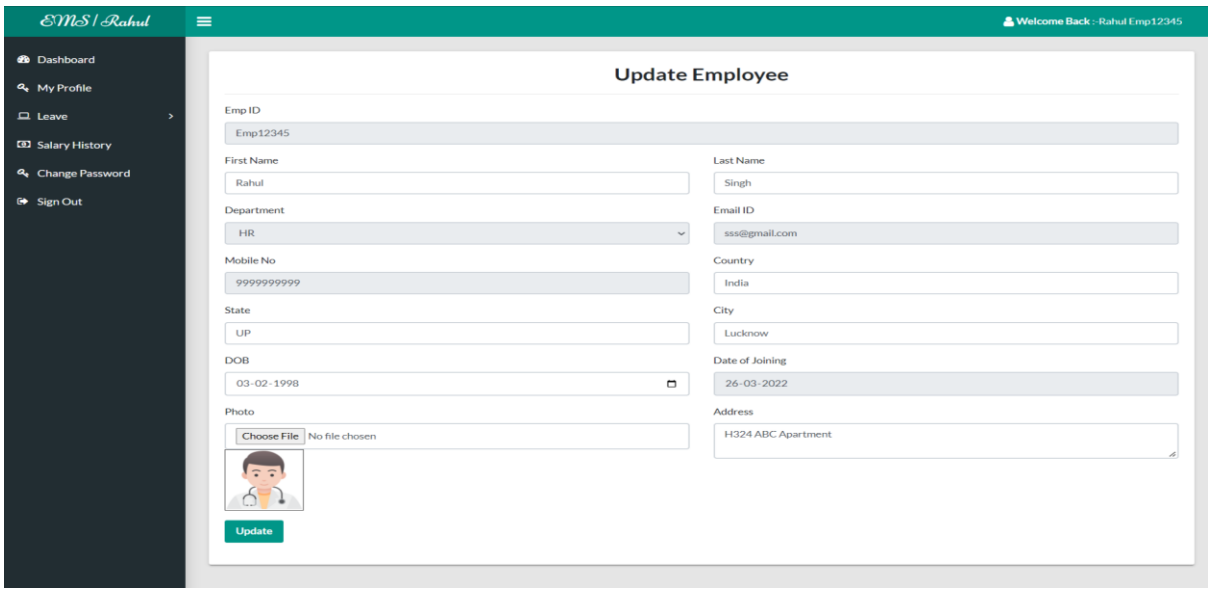
Dashboard



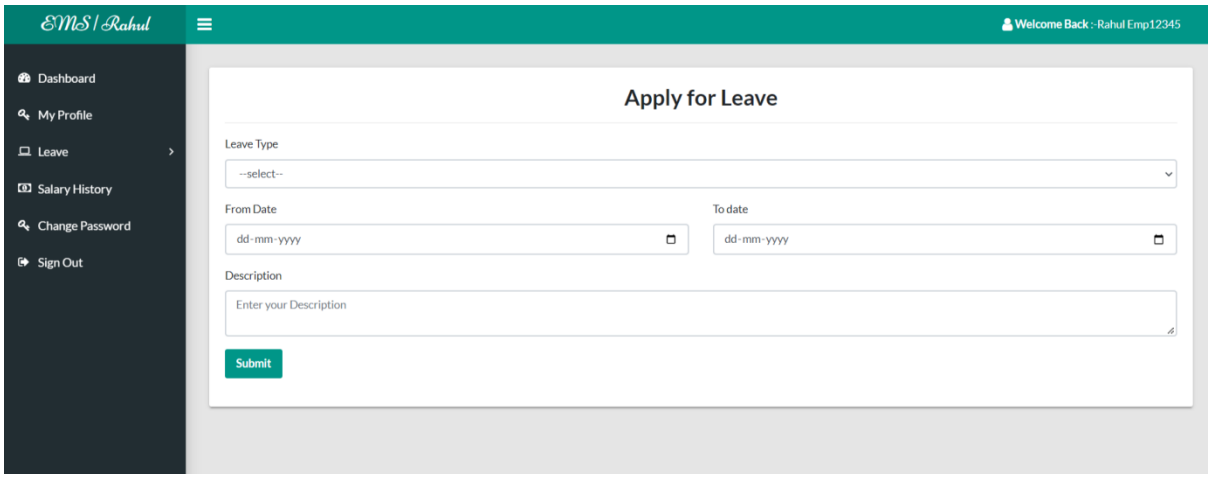
Change Password



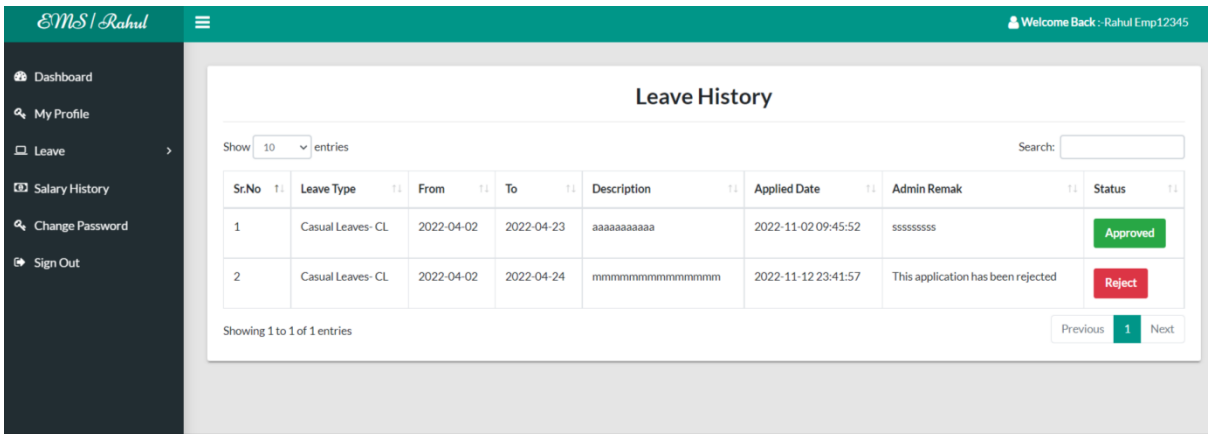
Profile



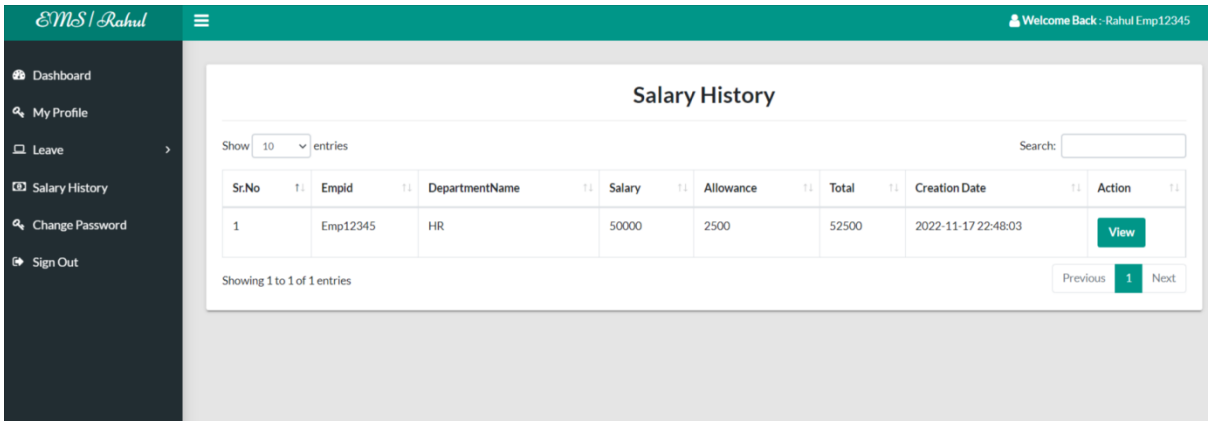
Apply Leave



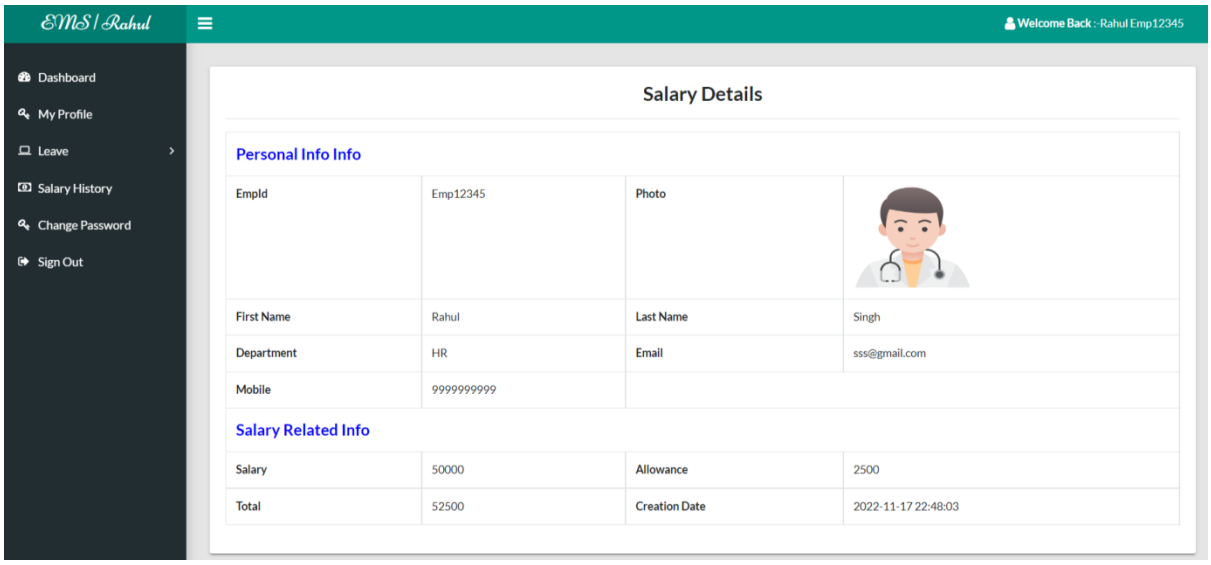
View Leave History



View Salary History



View Salary History in Details



Conclusion

The project titled as **Employee Management System** was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time.

Employee Management System can be used by companies to maintain the records of employee easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using this project

Bibliography

For PHP

- https://www.w3schools.com/php/default.asp
- https://www.sitepoint.com/php/
- https://www.php.net/

For MySQL

- https://www.mysql.com/
- http://www.mysqltutorial.org

For XAMPP

- https://www.apachefriends.org/download.html