**CHAPTER 3**

**BACK END DESIGN**

**3.1 SQL**

SQL stands for Structured Query Language. SQL used to communicate with a database. It is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Most of the database use SQL, most of them have their own additional proprietary extension that are usually only used on their system. However , the standard SQL commands such as “Select”, ”Insert”, ”Update”, ”Delete”, ”Create”, and “Drop” can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basis of each of these commands as well as allow you to put them to practice using SQL Interpreter.

**3.1.1 Features of SQL function**

SQL functions:

Can contain SQL Procedural Language statement and features which support the implementation of control-flow logic around traditional static and dynamic SQL statements. SQL are easy to implement, because they use a simple high-level, strongly typed language. SQL function are more reliable than equivalent external functions. Support input parameters. SQL scalar function returns a table result set. Support a simple, but powerful condition and error-handling model. Allow you to easily access the SQLSTATE and SQLCODE values as special variables.

Reside in the database and are automatically backed up and restored as part of backup and restore operation. Support nested function calls to other SQL functions or functions implemented in other languages. Many SQL statements can be included within SQL function, however these are exception.

**3.2 Database Tables**

The employee management system has 3 tables which are responsible for storing the required information based on the requirements

**Table 3.2.1: ADMIN\_LOGIN TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **NULL?** | **TYPE** | **DESCRIPTION** |
| User name | Not null | varchar | Id |
| Password |  | varchar | Password |

**Table 3.1:admin login table**

**Table 3.2.2: EMPLOYEE\_LOGIN TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **NULL?** | **TYPE** | **DESCRIPTION** |
| SSN | Not null | Integer | Id |
| Password |  | varchar | Password |

**Table 3.2:employee login table**

**Table 3.2.3: EMP\_DETAIL TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **NULL?** | **TYPE** | **DESCRIPTION** |
| Firstname |  | Varchar(10) | Firstname |
| Lastname |  | Varchar(10) | Lastname |
| Gender |  | Varchar(6) | Gender of employee |
| SSN | Not null | Int(10) | SSN of employee |
| DOB |  | Date | Date of birth |
| Department |  | Varchar(20) | Department name |
| Dept\_no |  | Varchar(10) | Department number |
| Phn\_number |  | Bigint(10) | Phone number |
| Address |  | Varchar(25) | Address of employee |

**Table 3.3:emp\_details table**

**Table 3.2.4: INSERT\_LOG\_TRIGGER**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **NULL?** | **TYPE** | **DESCRPTION** |
| Id | Not null | Int | Id number |
| SSN |  | Int(10) | Ssn of employee |
| Dep\_no |  | Varchar(10) | Department number |
| Action |  | Varchar(10) | Action performed |

**3.3 Schema diagram for Employee management system**

**Emp\_account**

|  |  |
| --- | --- |
| SSN | Password |

**Admin\_account**

|  |  |
| --- | --- |
| User\_name | Password |

**Emp\_details**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fname | Lname | Gender | SSN | DOB | Department | Dep\_no | Phn\_no | Address |

**3.4 E-R diagram for Employee Management System**

**EMP\_DETAIL**

**manages**

**view**

**LOGIN**

**ADMIN**

**as**

**EMPLOYEE**

3.4: E-R diagram

**3.5 Triggers**

In DBMS , a trigger is a SQL procedure that initiates an action when an event (INSERT,DELETE OR UPDATE) occurs. Since triggers are event-driven specialized procedures, they are stored in and managed by the DBMS. In employee management system trigger is used on the event insert.

CREATE TRIGGER ‘log\_trigger’ AFTER INSERT ON ‘emp\_details’

FOR EACH ROW

BEGIN

Insert into insert\_log\_trigger values(null,NEW,ssn,NEW.dep\_no,’inserted’);

END

GO

**3.6 Stored Procedure**

In database management systems an opration that is stored with the database server. Typically, stored procedures are written in SQL.

DELIMITER $$

CREATE DEFINER=’root’@’localhost’ PROCEDURE ‘getdata’(IN

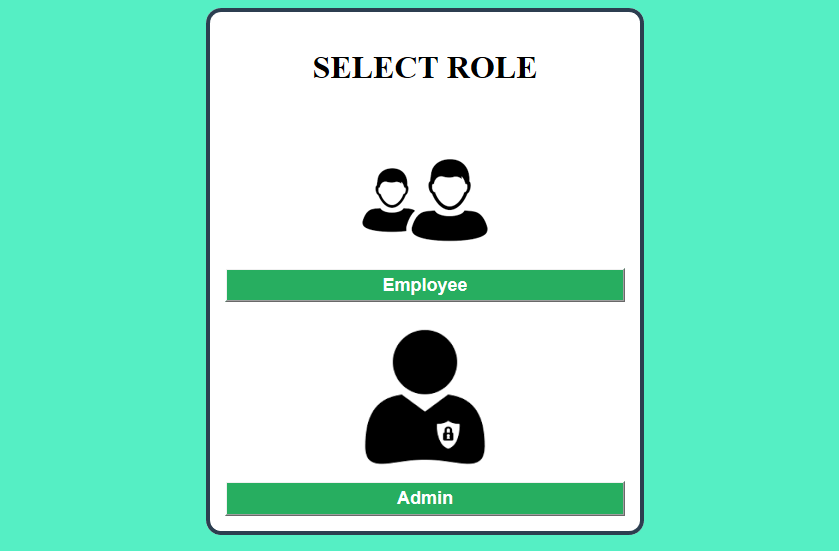
‘sn’ INT(10))

SELECT \* FROM emp\_details where ssn=sn$$

DELIMITER ;

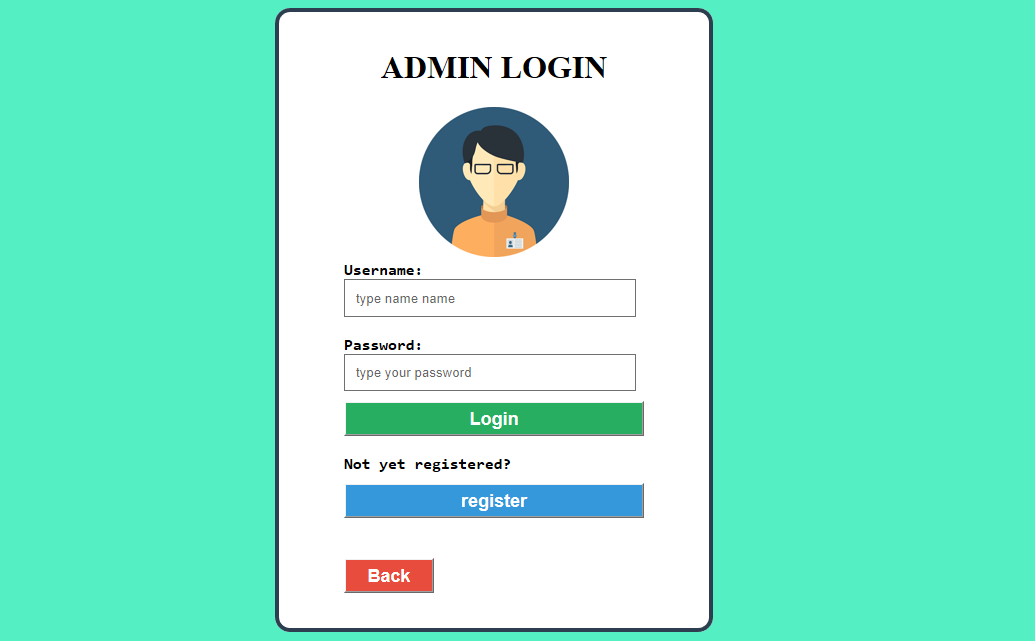
**CHAPTER 4**

**SNAPSHOTS**

****

**Fig 4.1: HOME PAGE**

The above figure shows home page, where admin login and employee login is present.Admin login is used for company maintenance and employment login is used for employee purpose. Employee should employee login that button will take to next page. If employee is not registered employee then employee can sign up then login. If employee is registered student then can directly login.

****

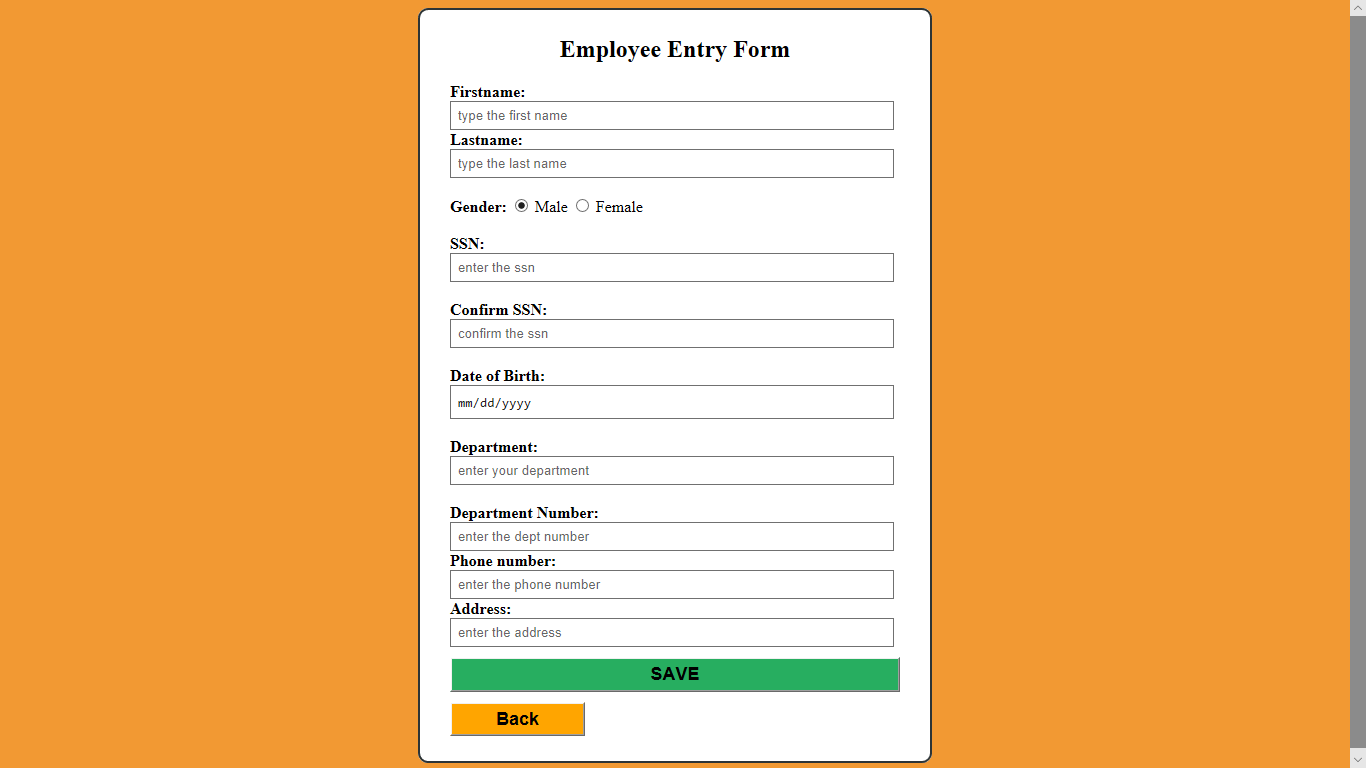
**Fig 4.2: ADMIN LOGIN PAGE**

The above figure shows admin login. Here admin insert username and password to login.



**Fig 4.3: ADMIN WORKSPACE PAGE**

The above figure shows admin workspace page. Here we have four buttons i.e add employee is used to add the employee details. View employee is used to retrieve the employee details. Update employee is used to edit the details. Delete employee is used to delete the details.



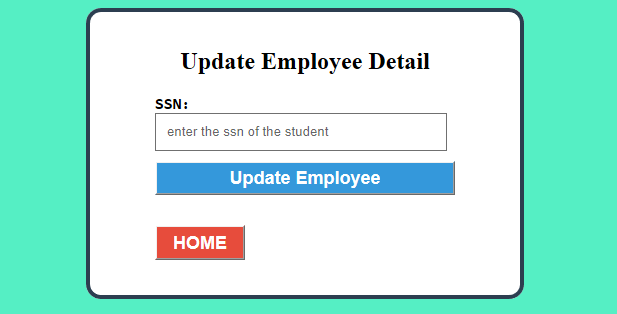
**Fig 4.4: EMPLOYEE ENTRY PAGE**

The above figure shows employee details entry form. Admin will fill the details of employee in the form inorder to add the employee to the database.



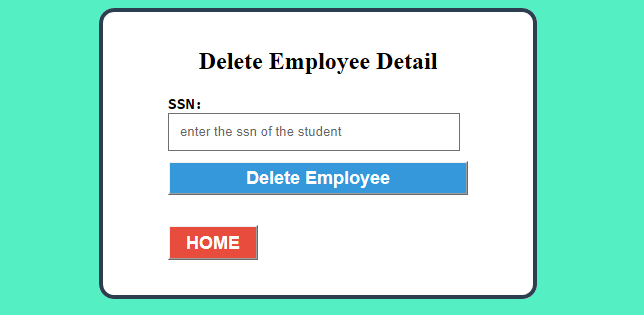
**Fig 4.5: EMPLOYEE VIEW PAGE**

The above figure shows employee view page. Here admin can retrieve employee details by entering ssn of employee.



**Fig 4.6: EMPLOYEE UPDATE PAGE**

This above figure shows employee update page. Here if admin want to edit any of employee details he has to enter the ssn of employee.

****

**Fig 4.7: EMPLOYEE DELETE PAGE**

This above figure shows employee delete page. If admin want to delete any of employee details he has to enter ssn of that employee.

****

**Fig 4.8: EMPLOYEE LOGIN**

This above figure shows employee login page. After employee being sign up to system he can login to system by entering ssn and password.

****

**Fig 4.9: EMPLOYEE SIGN UP PAGE**

The above figure shows sign up page for employee. For a new employee before login to system he or she should sign up to the system



**Fig 4.10: EMPLOYEE VIEW PAGE**

The above figure show employee view page where the employee will be entering ssn to view the details of employee.

**Chapter 5**

**CONCLUSION**

In this project we developed the employee management system. It is developed in NetBeans and the database has been built in SQL only, keeping in mind the specification of the system. For elaborating the system we have used simple schema and E-R diagrams. This system will be useful in maintaining employee details. This system is user friendly and accurate. The system has reached a steady state where all bugs have been eliminated.

**REFERENCES**

1) Fundamental Database System By Elmasri Navathe.

2)<https://www.google.co.in/search?q=netbeans&rlz=1C1CHZL_enIN770IN771&oq=netbeans&aqs=chrome.0.0l6.8600j0j7&sourceid=chrome&ie=UTF-8>

3) <https://netbeans.org>

4) https://xampp-windows.en.softonic.com/

5)https://netbeans.org/kb/docs/java/quickstart.html