Group 5: Employee Management System

First Name	Last Name	Class	Group Number
Jaswanth Busnur	Jagadeesh	ITMD 523 (4)	Group 5
Phanindra	Chandraprakash	ITMD 523 (4)	Group 5
Rahul Madhusudhan	Gudihal	ITMD 523 (4)	Group 5

Table of Contents

1. Introduction 2

2. Plans/Requirement Analysis 2

3. Database Design 3

4. Application Design 4

5. Demos 5

6. Conclusions and Future Work 33

6.1. Conclusions 33

6.2. Limitations 33

6.3. Potential Improvements or Future Work 33

7. Appendix 33

1. Introduction

A database is a collection of information that is organized so that it can be easily managed to be accessed, modified and updated. It is used to store the data in a pre-determined, structured way which helps in organized data update and retrieval process. The software systems used for creating and managing these databases to perform create, read, update and delete (CRUD) operations on the data is known as Database Management System(DBMS). It manages the data by allowing the database engine to access and modify the database schema and acts as an intermediary between database users and operations systems. Spreadsheets, XML databases and Microsoft Access, SQL server are some examples of database and database management systems respectively. With database management systems, there is little programming and development times with low system administration. It also provides solid security and data sharing over the conventional file systems. With the use of database management systems in p, there is very less chance of data becoming obsolete and provides the facility for different tools from third party vendors to gain access to the data and process them.

Our project involves creating a database of employees for an organization and provide an appraisal or grading system for the organization to evaluate the performance of the employees over time. It involves performing create, read, update and delete (CRUD) operations on the data of employees. The application contains employee and their department details which will be used for appraisal or grading system. The technologies used to develop and implement this application are Java classes for business logic implementation, HTML with Java Server pages for creating web pages with input of records to the database and Java Script for field validation. A front end graphical user interface is used to populate the fields in the application with stylesheets used to set the visual style of web pages and interface for the users that present a view of the application to the users. The implementation includes integrating dynamically generated webpages using HTML and JSP with Java classes. The login page provides interface to login for administrator and employee in the web page. The application displays three tabs in the home page; Home, Administration login and Employee login. Logging in as an employee allows only the displaying of details without any privileges to modify the data entered by the administrator.

2. Plans/Requirement Analysis

Our plan is to develop an Employee Management Application, used to manage employees, in Recruitment process, Project Management, and Client Management processes within the organization. The existing system requires lot of manual intervention in integrating data and management of day-to-day activities. Since the organization was maintaining different systems for Human Resource Management, Project Management, and Client Management processes. Searching and tracking of information took more time and the entire system was convoluted. Requirement analysis was done by exploring the available development, testing environments and techniques. Simultaneously database analysis and design was done to integrate with Graphical User Interface (GUI) application. The new database system must be able to integrate the employee details across different departments and provide comprehensive solution for data integrity and data search.

Functional Specification:

User Specification:

- Administrator
- Employee

Administrator:

- Administrator can create, read, update and delete employees and departments.
- Administrator can also provide appraisal ratings and alter the salary depending on the ratings.

Employee:

- Employee can only view his details under view information screen after logging.

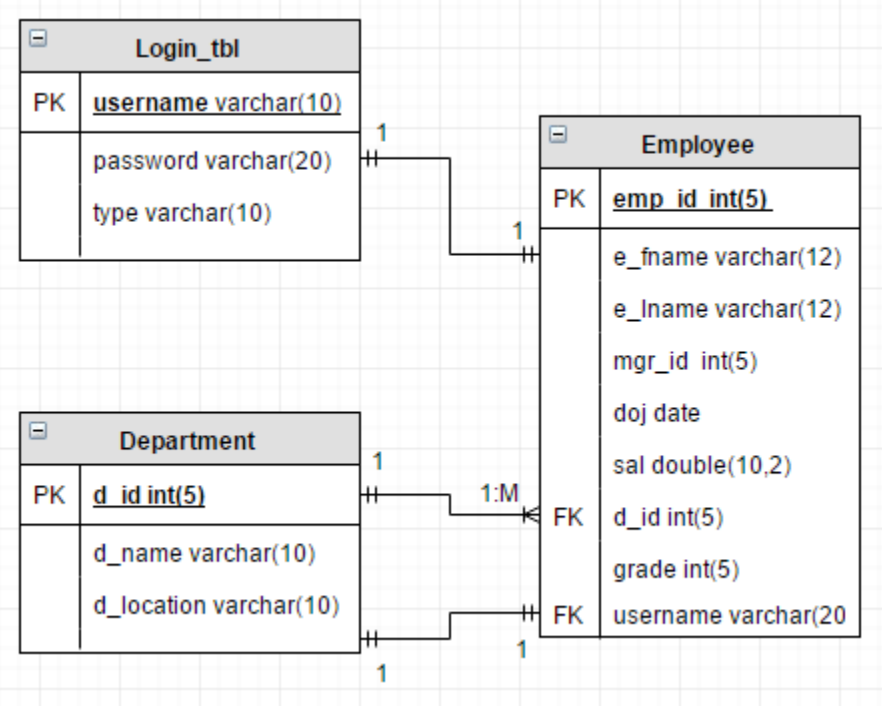
Module Specification:

There are three modules,

- Employee
- Department
- Appraisal Management

3. Database Design

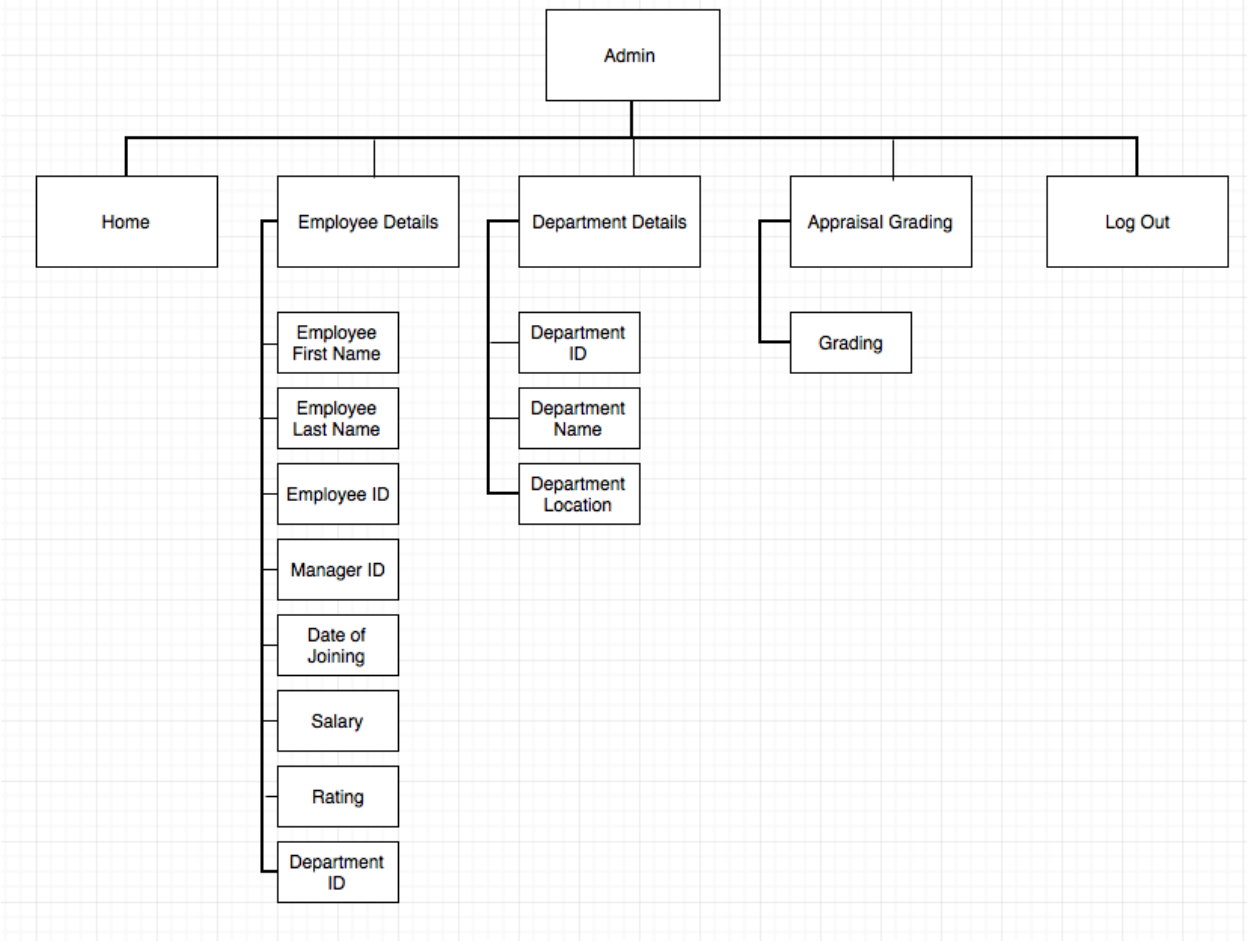
Physical Data Model



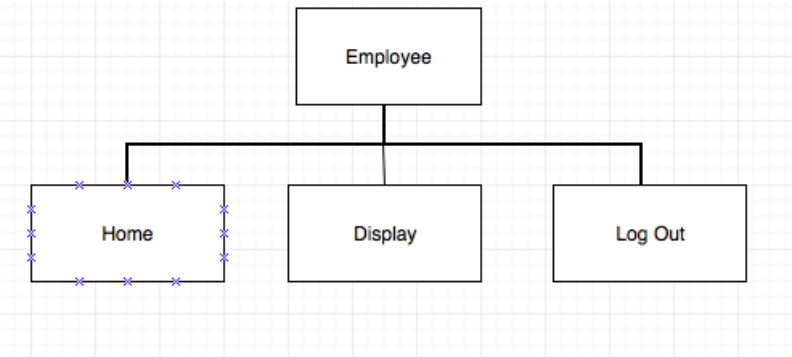
4. Application Design

Process Flow:

1. For Admin user



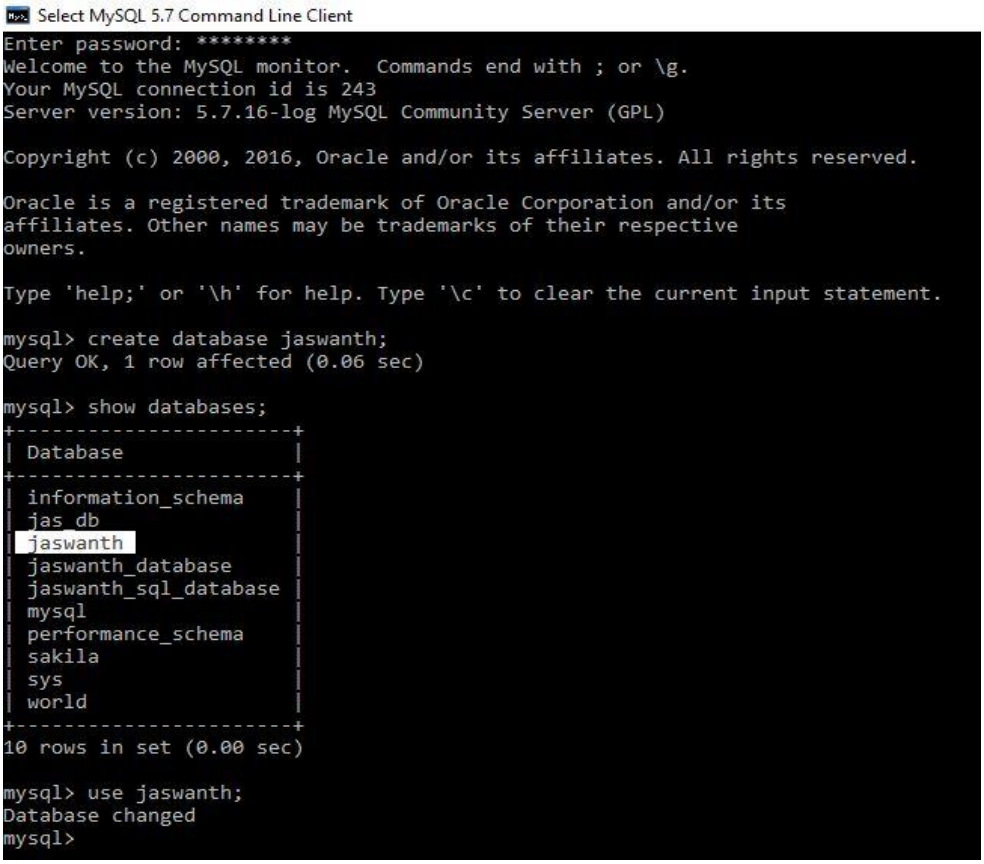
2. For Employee user



The employee management system application launches in a web browser featuring the login pages for the admin and employee users. The users as admin will be able to perform CRUD operations on the employee details. There is also an appraisal grading system for the employees which works on basis of set ratings on scale of 1-5 provided by the employee's manager and the appraisal system increases the employee's salary based on those ratings provided. The admin can also alter the details of departments to which the employees belong to and perform CRUD operations as well.

5. Demo

Database creation (DB name - jaswanth)



Tables creation:

1. **Login_tbl**

```
mysql> CREATE TABLE login_tbl (
    ->   username varchar(20) NOT NULL,
    ->   password varchar(20) default NULL,
    ->   type varchar(10) default NULL,
    ->   PRIMARY KEY  (username)
    -> );
Query OK, 0 rows affected (0.47 sec)

mysql> desc login_tbl;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(20) | NO   | PRI | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
| type     | varchar(10) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.04 sec)
```

2. **Department**

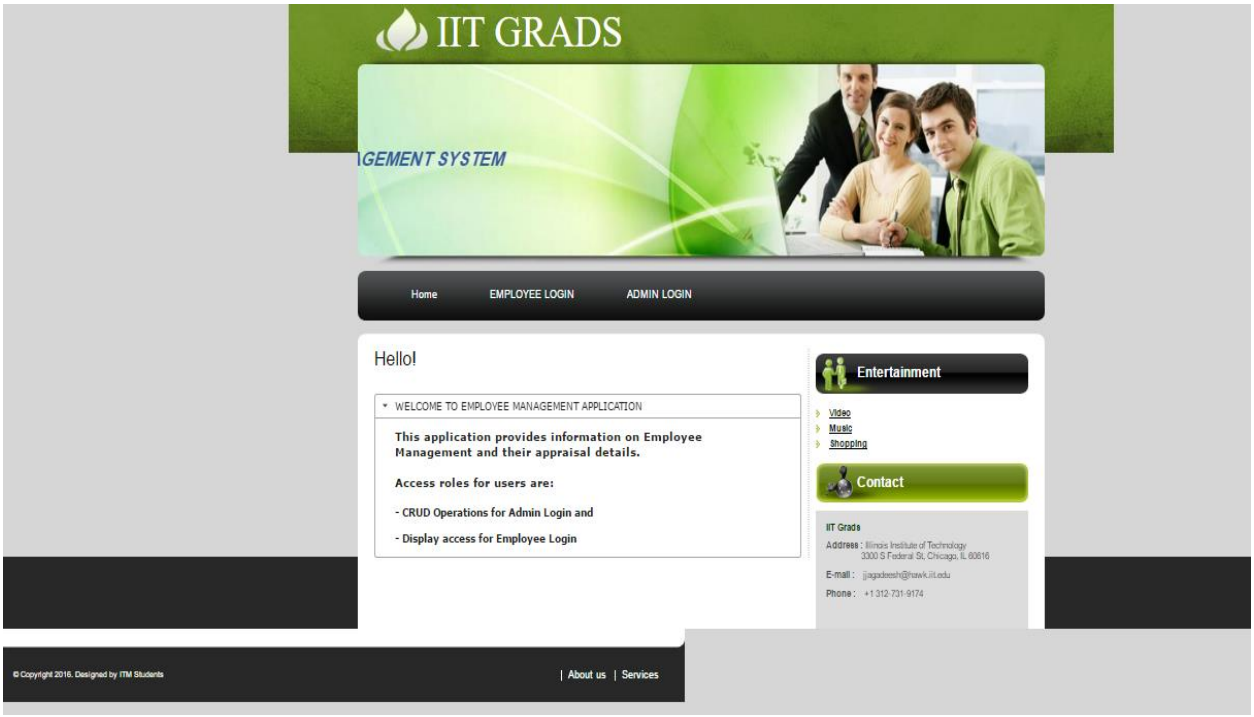
```
mysql> CREATE TABLE department (
    ->   d_id int(5) NOT NULL,
    ->   d_name varchar(10) default NULL,
    ->   d_location varchar(10) default NULL,
    ->   PRIMARY KEY  (d_id)
    -> );
Query OK, 0 rows affected (0.34 sec)

mysql> desc department;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| d_id       | int(5)      | NO   | PRI | NULL    |       |
| d_name     | varchar(10) | YES  |     | NULL    |       |
| d_location | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

3. Employee

MySQL 5.7 Command Line Client

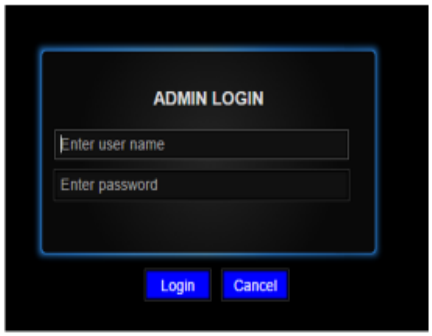
```
mysql> CREATE TABLE employee (
    ->   e_fname varchar(12) default NULL,
    ->   e_lname varchar(12) default NULL,
    ->   emp_id int(5) NOT NULL,
    ->   mgr_id int(5) NOT NULL,
    ->   doj date default NULL,
    ->   sal double(10,2) default NULL,
    ->   d_id int(5) default NULL,
    ->   grade int(5) default NULL,
    ->   PRIMARY KEY  (emp_id),
    ->   KEY d_id (d_id),
    ->   CONSTRAINT employee_ibfk_1 FOREIGN KEY (d_id) REFERENCES department (d_id)
    -> );
Query OK, 0 rows affected (0.56 sec)

mysql> alter table employee add column username varchar(20);
Query OK, 0 rows affected (0.77 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table employee add constraint un_fk_cnst foreign key (username) references login_tbl(username);
Query OK, 0 rows affected (1.14 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc employee;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| e_fname  | varchar(12) | YES  |     | NULL    |       |
| e_lname  | varchar(12) | YES  |     | NULL    |       |
| emp_id   | int(5)      | NO   | PRI | NULL    |       |
| mgr_id   | int(5)      | NO   |     | NULL    |       |
| doj      | date        | YES  |     | NULL    |       |
| sal      | double(10,2)| YES  |     | NULL    |       |
| d_id     | int(5)      | YES  | MUL | NULL    |       |
| grade    | int(5)      | YES  |     | NULL    |       |
| username | varchar(20) | YES  | MUL | NULL    |       |
+----------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

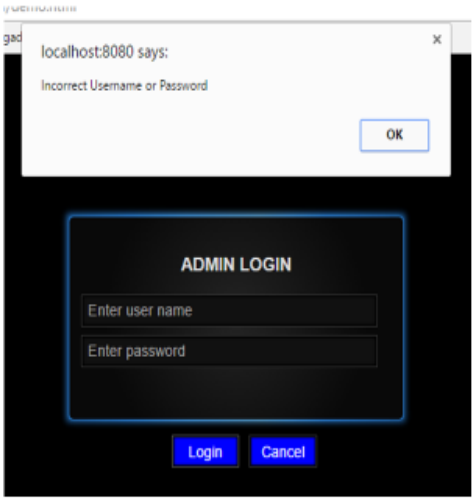
Application Homepage



Admin Page Login

Error trapping under Login Page





Successful login to Admin Page

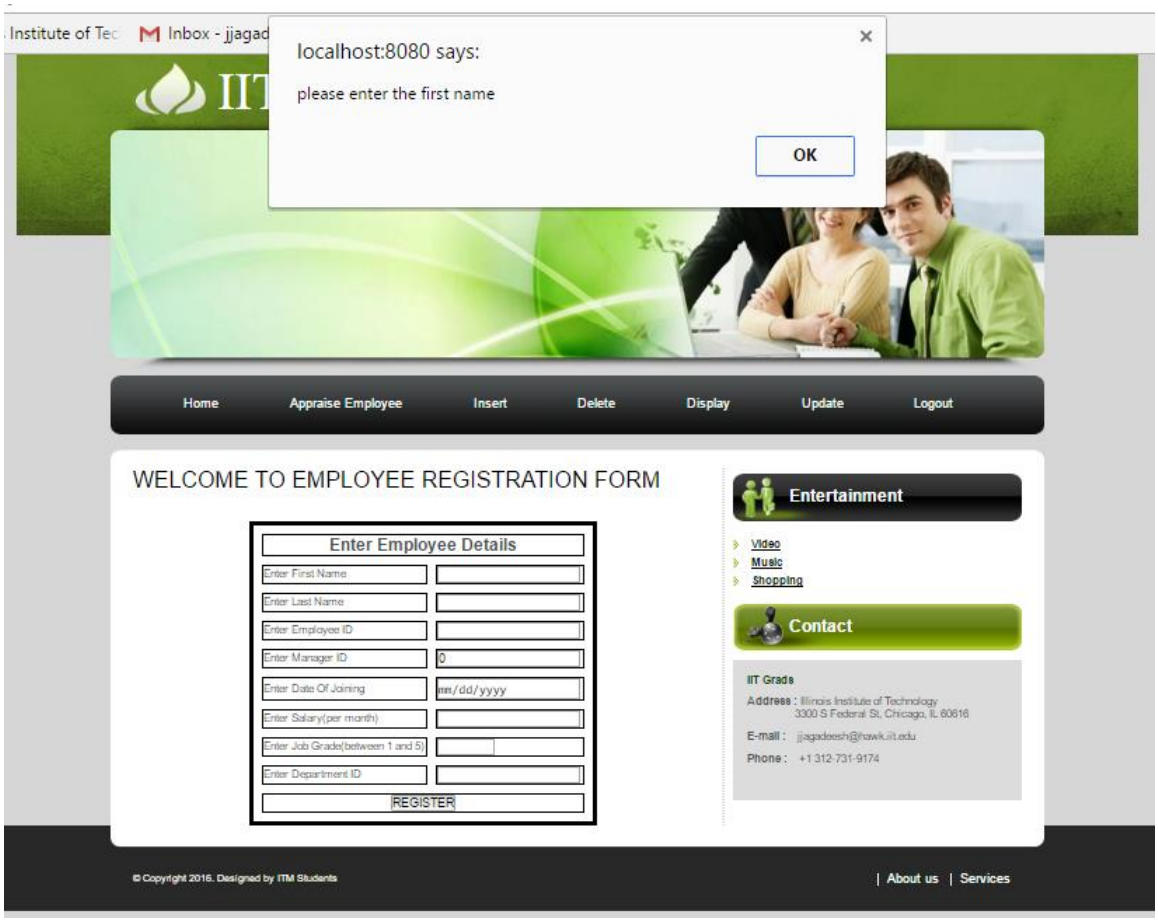
Insert Tab

IIT GRADS
ANAGEMENT SYSTEM
Home
Appraise Employee
Insert
Delete
Display
Update
Logout
REGISTRATION FORM
Employee
Department
Entertainment
Video
Music
Shopping
Contact
IIT Grads
Address : Illinois Institute of Technology
3300 S Federal St, Chicago, IL 60616
E-mail : jjagadeesh@hawk.iit.edu
Phone : +1 312-731-9174
© Copyright 2016. Designed by ITM Students
| About us | Services

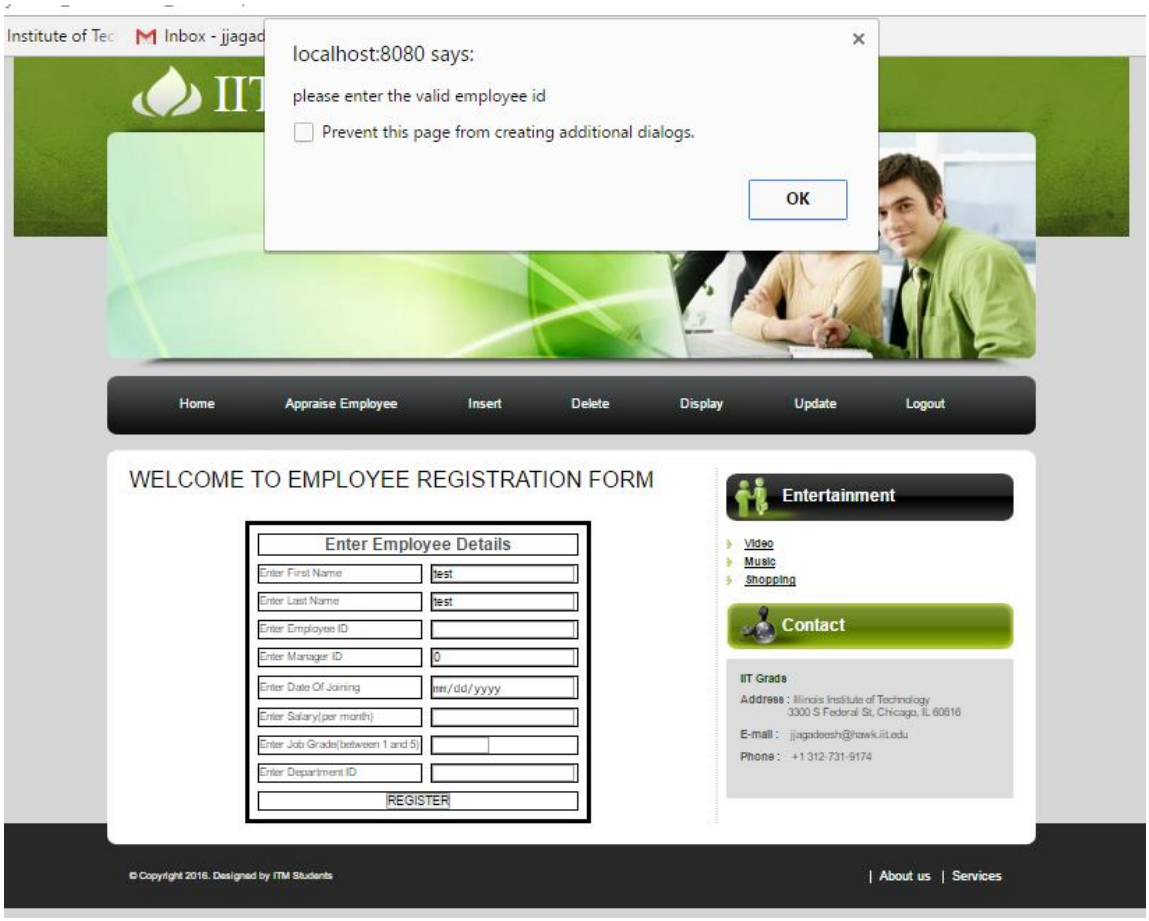
Insert Employee Details

Insert Department Details

Field level validation check



Field level validation check



Delete Tab

IIT GRADS
PLOYEE MANAGEMENT SYSTEM
Home
Appraise Employee
Insert
Delete
Display
Update
Logout
DELETION FORM
Employee
Department
Entertainment
Video
Music
Shopping
Contact
IIT Grads
Address : Illinois Institute of Technology
3300 S Federal St, Chicago, IL 60616
E-mail : jjagadeesh@iitedu
Phone : +1 312-731-9174
© Copyright 2016. Designed by ITM Students
| About us | Services

Delete Employee Details

Delete Department Details

Display Tab

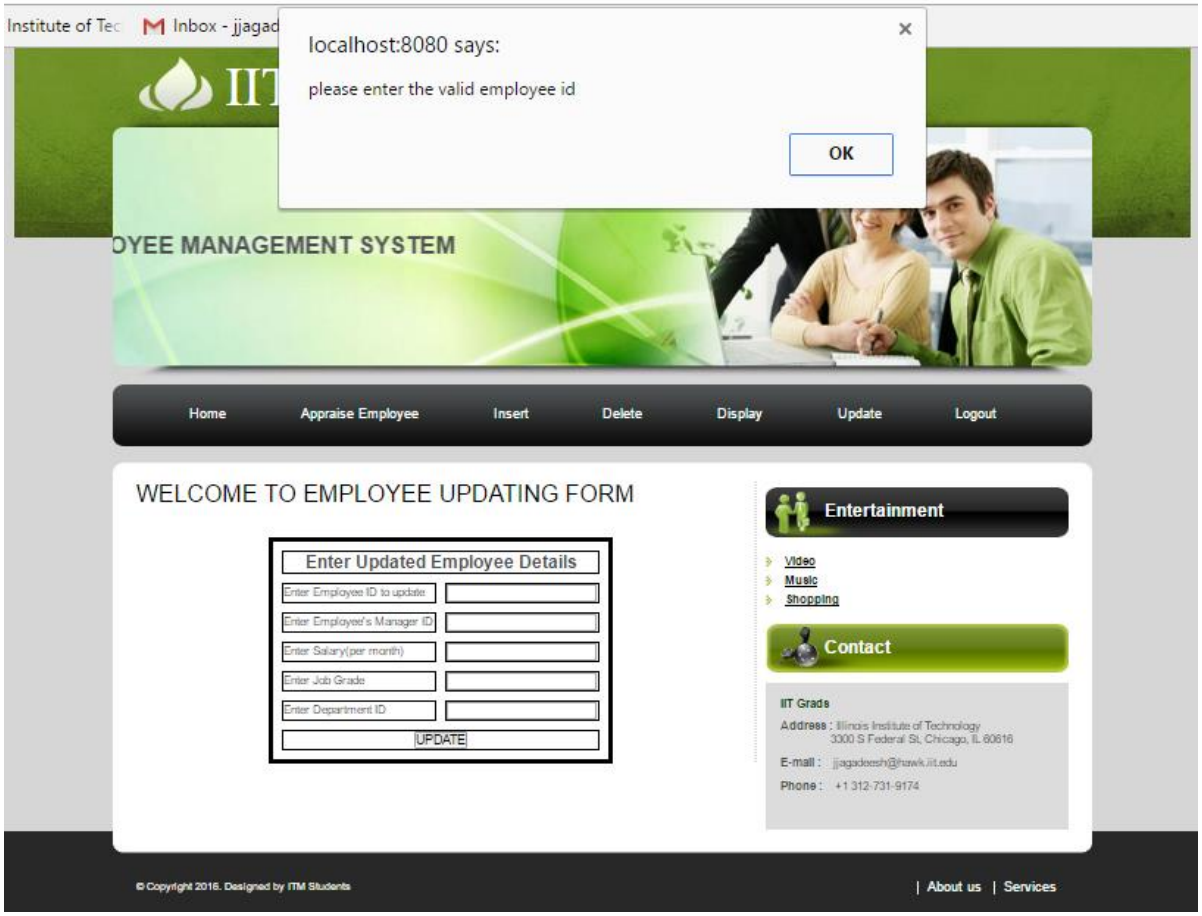
IIT GRADS
EMI
Home
Appraise Employee
Insert
Delete
Display
Update
Logout
DISPLAY FORM
Employee
Department
Entertainment
Video
Music
Shopping
Contact
IIT Grads
Address : Illinois Institute of Technology
3300 S Federal St, Chicago, IL 60616
E-mail : jjagadeesh@hawk.iit.edu
Phone : +1 312-731-9174
© Copyright 2016. Designed by ITM Students
| About us | Services

Update Tab

IIT GRADS
TEM
Home
Appraise Employee
Insert
Delete
Display
Update
Logout
UPDATION FORM
Employee
Department
Entertainment
Video
Music
Shopping
Contact
IIT Grads
Address : Illinois Institute of Technology
3300 S Federal St, Chicago, IL 60616
E-mail : jjagadeesh@hawk.iit.edu
Phone : +1 312-731-9174
© Copyright 2016. Designed by ITM Students
| About us | Services

Update Employee Details

Validation check for Update employee tab



Update Department Details

Appraise Employee Tab

Appraisal Module Functionality:

While creating the Employee details, admin will enter the employee ratings, the same will be reflected on employee's salary.

For example; If employee salary is $10000 and rating is 5, then his new salary will be $15000.

Calculations:

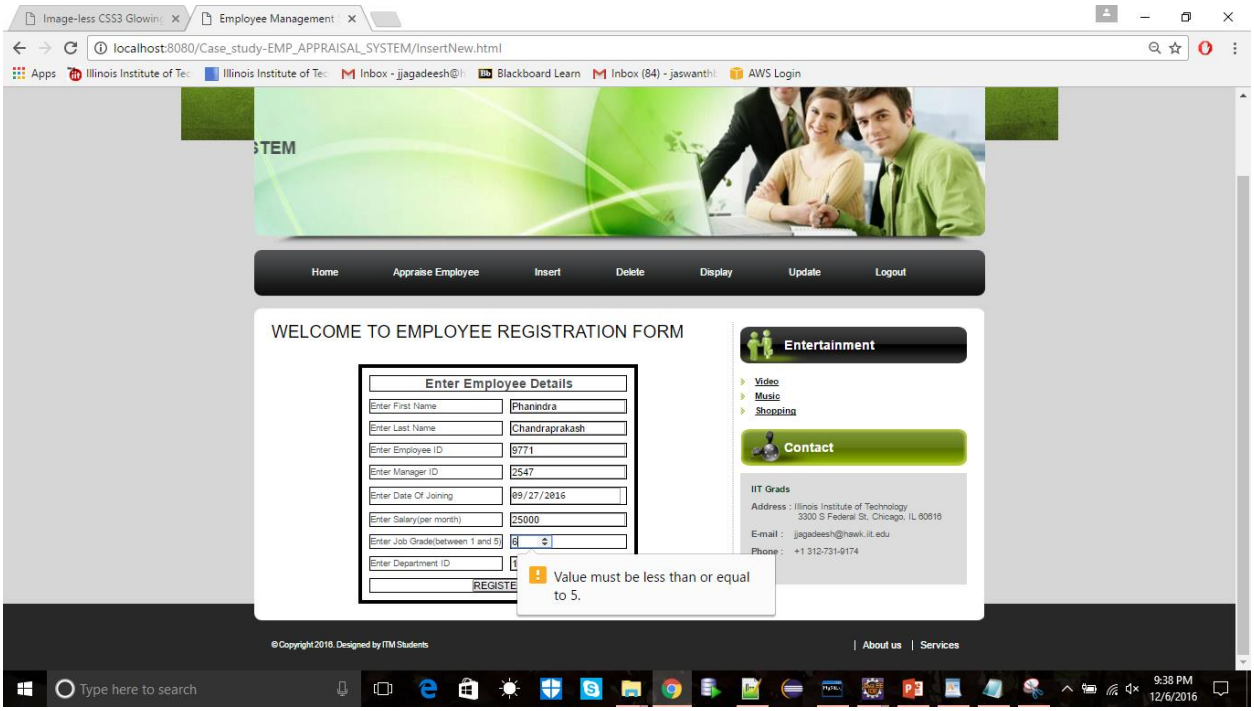
Old Salary: $10000

Rating: 5

New Salary = Old Salary + (0.5 * Old Salary)

New Salary = $15000

Validation check for Employee Grade

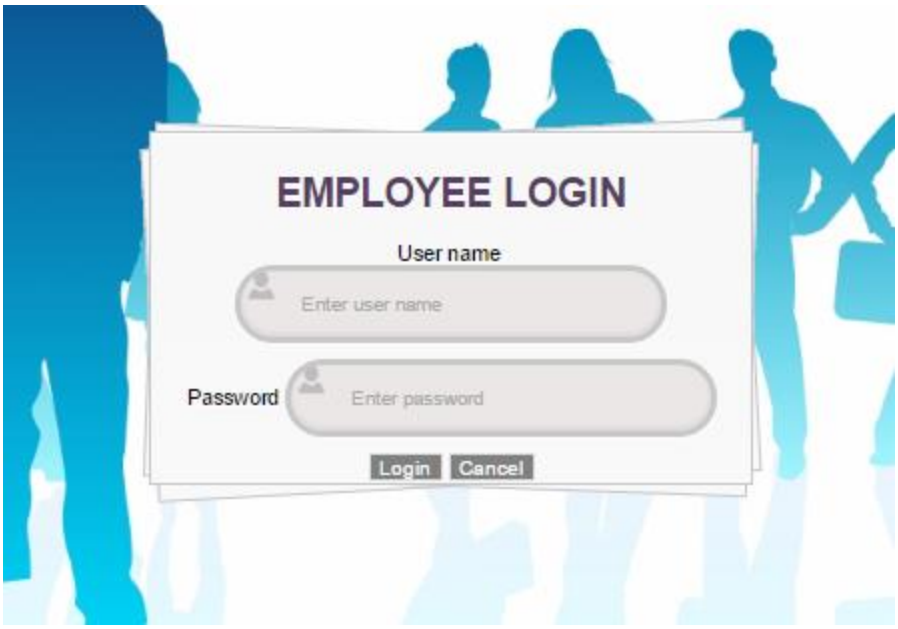


Display Employee Tab

Display Department Tab

Employee Details after Appraising based on Grading

Employee Login Page



Employee Login Error Check

Welcome page for Employee Login. Specific user is displayed based on who is logs in

Employee Display form. Only specific information of the logged in employee is displayed

HTML additional implementation (Accordion)

HTML additional implementation (Services to Links)

6. Conclusions and Future Work

6.1. Conclusions

This report includes the information development which places emphasis on basic CRUD operative functions performed on the data in the employee database. This application helps an organization to perform the basic and essential function of recording the data of its employees, their functionalities and performances at all levels in its organizational structure. The application also provides a grading or appraisal program for the managers to rate the performance of the employees and claim their benefits in accordance with their ratings. With each passing day, every organization is forced to invest in resources for managing their staff and the employee management system application does just that to help the organization stay ahead in the business competition.

6.2. Limitations

- The appraisal method is subjected to the presence of Halo Effect in evaluation of the employees by the appraisers.
- There can be ambiguity in average ratings given to the employees by the top management.
- Complexity involved in defining description of events.
- Overtimes are added to the salaries.

6.3. Potential Improvements or Future Work

The future enhancements for employee management system include the option of raising complaints and suggestions from the employee side. It will also feature the attendance list for employees and managers with log in and log out times. It also includes integrating the pay roll system with the existing module for more reliability and flexibility in the code. It can be made to work as an online job portal for recruiting management.

Improvements:

- All the employee records will be computerized and include online management only.
- This system provides flexibility and integration for coding architecture.
- This system is more efficient and reliable and facilitates the organizations.
- It helps in preventing data manipulations and thereby reduces data redundancy.
- Employees information is managed efficiently by business intelligence and supplies real time information.

7. Appendix

Minimum Hardware Requirements:

Client Side	Server Side
Processor: 1.5 GHz	Processor: 2.5 GHz
RAM: 2 GB	RAM: 8 GB
HDD: 252 GB	HDD: 1TB

Minimum Software Requirements:

Client Side	Server Side
Operating System: Windows XP onwards or Any Compatible OS	Operating System: Windows XP onwards or Any Compatible OS
Browser: Edge, Chrome, Firefox, and Safari	Web Server: Apache Tomcat Version 9.0
	Database Server: MySQL 5.7
	Framework: Web Application Framework

SQL Scripts

```
DROP TABLE IF EXISTS login_tbl;

CREATE TABLE login_tbl (

  username varchar(20) NOT NULL,

  password varchar(20) default NULL,

  type varchar(10) default NULL,

  PRIMARY KEY  (username));

DROP TABLE IF EXISTS department;

CREATE TABLE department (

  d_id int(5) NOT NULL,

  d_name varchar(10) default NULL,

  d_location varchar(10) default NULL,

  PRIMARY KEY  (d_id));

DROP TABLE IF EXISTS employee;
```

```
CREATE TABLE employee (
  e_fname varchar(12) default NULL,
  e_lname varchar(12) default NULL,
  emp_id int(5) NOT NULL,
  mgr_id int(5) NOT NULL,
  doj date default NULL,
  sal double(10,2) default NULL,
  d_id int(5) default NULL,
  grade int(5) default NULL,
  PRIMARY KEY  (emp_id),
  KEY d_id (d_id),
 CONSTRAINT employee_ibfk_1 FOREIGN KEY (d_id) REFERENCES department (d_id));
alter table employee add column username varchar(20);
alter table employee add constraint un_fk_cnst foreign key (username) references login_tbl(username);
```