VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590 018, KARNATAKA

DBMS MINI PROJECT REPORT

ON

"EMPLOYEE DATABASE AND PAYROLL MANAGEMENT SYSTEM"

Submitted in the partial fulfillment of requirements for the

5th SEM DBMS MINI PROJECT (18CSL58)

IN

COMPUTER SCIENCE AND ENGINEERING

PROJECT ASSOCIATES

Jeevan H K **4BD20CS040**

Darshan A Hiremath **4BD20CS124**

PROJECT GUIDES

Dr. Gururaj T Ph.D.
Associate Professor

Prof. Chandrashekhar M V M.Tech.
Assistant Professor

2022-2023

Bapuji Institute of Engineering and Technology
Department of Computer Science and Engineering
Davanagere-577004

Bapuji Institute of Engineering and Technology
Davanagere – 577004

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **JEEVAN H K and DARSHAN A HIREMATH** bearing USN **4BD20CS040 and 4BD20CS124** respectively of Computer Science and Engineering department have satisfactorily submitted the Mini Project report entitled **" EMPLOYEE DATABASE AND MANAGEMENT SYSTEM "** for **5th SEM DBMS MINI PROJECT (18CSL58).** The project report has been approved as it satisfies the academic requirements for the year 2022-23.

Dr. Gururaj T Ph.D.
Associate Professor
Guide

Prof. Chandrashekhar M V M.Tech.
Assistant Professor
Co-Guide

Dr. Nirmala C R Ph.D.
Prof.& Head of Department

Signature of Examiners:

Date:

Place:davanagere

1._____

2._____

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute, **"BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY"** for having well-qualified staff and labs furnished with the necessary equipment.

We express our sincere thanks to our resourceful guides **Dr. Gururaj T**, Associate Professor, Department of Computer Science and Engineering, BI.E.T., Davanagere, and **Prof. Chandrashekhar M V**, Assistant Professor, Department of Computer Science and Engineering, BI.E.T., Davanagere, who helped us in every aspect of our project. We are indebted to her discussions about the technical aspects and suggestions pertaining to our project.

We are grateful to **Dr. Nirmala C R**, Professor and H.O.D, Department of Computer Science and Engineering, B.I.E.T., Davanagere, for endeavoring encouragement, facilities, and extended support.

We also express our wholehearted gratitude to our respected Principal, **Dr. H B Aravind** for his moral support and encouragement.

We would like to extend our gratitude to all staff of the **Department of Computer Science and Engineering** for the help and support rendered to us. We have benefited a lot from the feedback, and suggestions given by them.

We would like to extend our gratitude to all our family members and friends especially for their advice and moral support.

Jeevan H K (4BD00CS040)

Darshan A Hiremath (4BD00CS124)

Bapuji Educational Association (Regd.)
Bapuji Institute of Engineering and Technology, Davangere-577004

Vision and Mission of the Institute

Vision

"To be a centre of excellence recognized nationally internationally, in distinctive areas of engineering education and research, based on a culture of innovation and invention."

Mission

"BIET contributes to the growth and development of its students by imparting a broadbased engineering education and empowering them to be successful in their chosen field by inculcating in them positive approach, leadership qualities and ethical values."

Vision and Mission of the Computer Science and Engineering Department

Vision

"To be a centre-of-excellence by imbibing state-of-the-art technology in the field of Computer Science and Engineering, thereby enabling students to excel professionally and be ethical."

Mission

1.	Adapting best teaching and learning techniques that cultivates Questioning and Reasoning culture among the students.
2.	Creating collaborative learning environment that ignites the critical thinking in students and leading to the innovation.
3.	Establishing Industry Institute relationship to bridge skill gap and make them industry ready and relevant.
4.	Mentoring students to be socially responsible by inculcating ethical and moral values.

Program Educational Objectives (PEOs):

PEO1	To apply skills acquired in the discipline of computer science and engineering for solving Societal and industrial problems with apt technology intervention.
PEO2	To continue their carrier ion industry /academia or pursue higher studies and research.
PEO3	To become successful entrepreneurs, innovators to design and develop software products and services that meets societal, technical and business challenges.
PEO4	To work in the diversified environment by acquiring leadership qualities with effective communication skills accompanied by professional and ethical values.

Program Specific Outcomes (PSOs):

PSO1	Analyse and develop solutions for problems that are complex in nature but applying the knowledge acquired from the core subjects of this program.
PSO2	To develop secure, scalable, resilient and distributed applications for industry and societal Requirements.
PSO3	To learn and apply the concepts and contract of emerging technologies like artificial intelligence, machine learning, deep learning, big-data analytics, IOT, cloud computing etc for any real time problems.

Course Learning Objectives:

- Foundation knowledge in database concepts, technology and practice to groom students into well-informed database application developers.
- Strong practice in SQL programming through a variety of database problems.
- Develop database applications using front-end tools and back-end DBMS.

Course Outcomes:

CO1: Create, Update and query on the database.

CO2: Demonstrate the working of different concepts of DBMS

CO3: Implement, analyze and evaluate the project developed for an application.

ABSTRACT

"Employee Database And Payroll Management System" is designed to make the existing manual system automatic with the help of computerized equipment and full-edged 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 computerized records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client.

CONTENTS

LIST OF FIGURES

LIST OF TABELS

CHAPTER 1

INTRODUCTION

1.1 Introduction

This project mainly focuses on managing all employee's financial records in a simple and automated fashion. This payroll management system manages the employee's salaries ,deductions, other conveyance, net pay, bonuses and generation of pay slips, etc. and this has been developed to overcome the problems faced in the practicing of manual system.

1.2 Overview (DBMS and front end used)

Database is a collection of related data and data is a collection of facts and figures that can be to produce information. Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks. A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

A DBMS relieves users of framing programs for data maintenance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database. Some other DBMS examples include:

- MySQL
- SQL Server
- Oracle
- DBASE
- FoxPro

1.2 PHP (HYPERTEXT PREPROCESSOR)

PHP is the most popular and widely used server-side scripting language for web development. It is used to make the Dynamic pages in websites. Rasmus Lerdorf was the creator of PHP in 1995. PHP codes are embedding in HTML source codes for making the page dynamic. PHP can deal with most of the requirements in web development like Database, File handling, String operations, Arrays, Graphics, File Uploads, Data processing etc. PHP can be used in any operating system with a web server Supports PHP. Apache web server is one of the popular web servers dealing with PHP + MySQL. Moreover, PHP is absolutely free to use.

1.3 Problem Statemen

A Payroll Management System is a software that is used to manage all employee's financialrecords in a simple and automated way.

1.5 Objectives

1.To automate each and every activity of the manual system, which increases its throughput.

2.To provide a quick response with very accurate information as and when required.

3.make the present manual system more interactive, speedy and user friendly.

4.To avail information, whatever and whenever needed.

5.To reduce the cost of maintenance.

6.To develop a stand-alone application for " Payroll Management System"

CHAPTER 1

INTRODUCTION

1.1 Introduction

This project mainly focuses on managing all employee's financial records in a simple and automated fashion. This payroll management system manages the employee's salaries ,deductions,other conveyance,net pay, bonuses and generation of payslips, etc.and this has been developed to overcome the problems faced in the practicing of manual system.

1.2 Overview (DBMS and front end used)

Database is a collection of related data and data is a collection of facts and figures that can be to produce information. Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks. A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

A DBMS relieves users of framing programs for data maintenance. Fourth- generation query languages, such as SQL, are used along with the DBMS package to interact with a database. Some other DBMS examples include:

MySQL

- SQL Server
- Oracle
- DBASE
- FoxPro

1.3 PHP (HYPERTEXT PREPROCESSOR)

PHP is the most popular and widely used server-side scripting language for web development. It is used to make the Dynamic pages in websites. Rasmus Lerdorf was the creator of PHP in 1995. PHP codes are embedding in HTML source codes for making the page dynamic. PHP can deal with most of the requirements in web development like Database, File handling, String operations, Arrays, Graphics, File Uploads, Data processing etc. PHP can be used in any operating system with a web server Supports PHP. Apache web server is one of the popular web servers dealing with PHP + MySQL. Moreover, PHP is absolutely free to use.

1.4 Problem Statement

A Payroll Management System is a software that is used to manage all employee's financialrecords in a simple and automated way.

1.5 Objectives

1. To automate each and every activity of the manual system, which increases its throughput.
2. To provide a quick response with very accurate information as and when required.
3. To make the present manual system more interactive, speedy and user friendly.
4. To avail information, whatever and whenever needed.
5. To reduce the cost of maintenance.
6. To develop a stand-alone application for " Payroll Management System"

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

The hardware required for the development of this project is:

1. Processor : Intel Core i5
2. Processor speed :1.7 GHz
3. RAM :4 GB RAM
4. System Type :32/ 64-Bit Operating System

2.2 Software Requirements

The software required for the development of this project is:

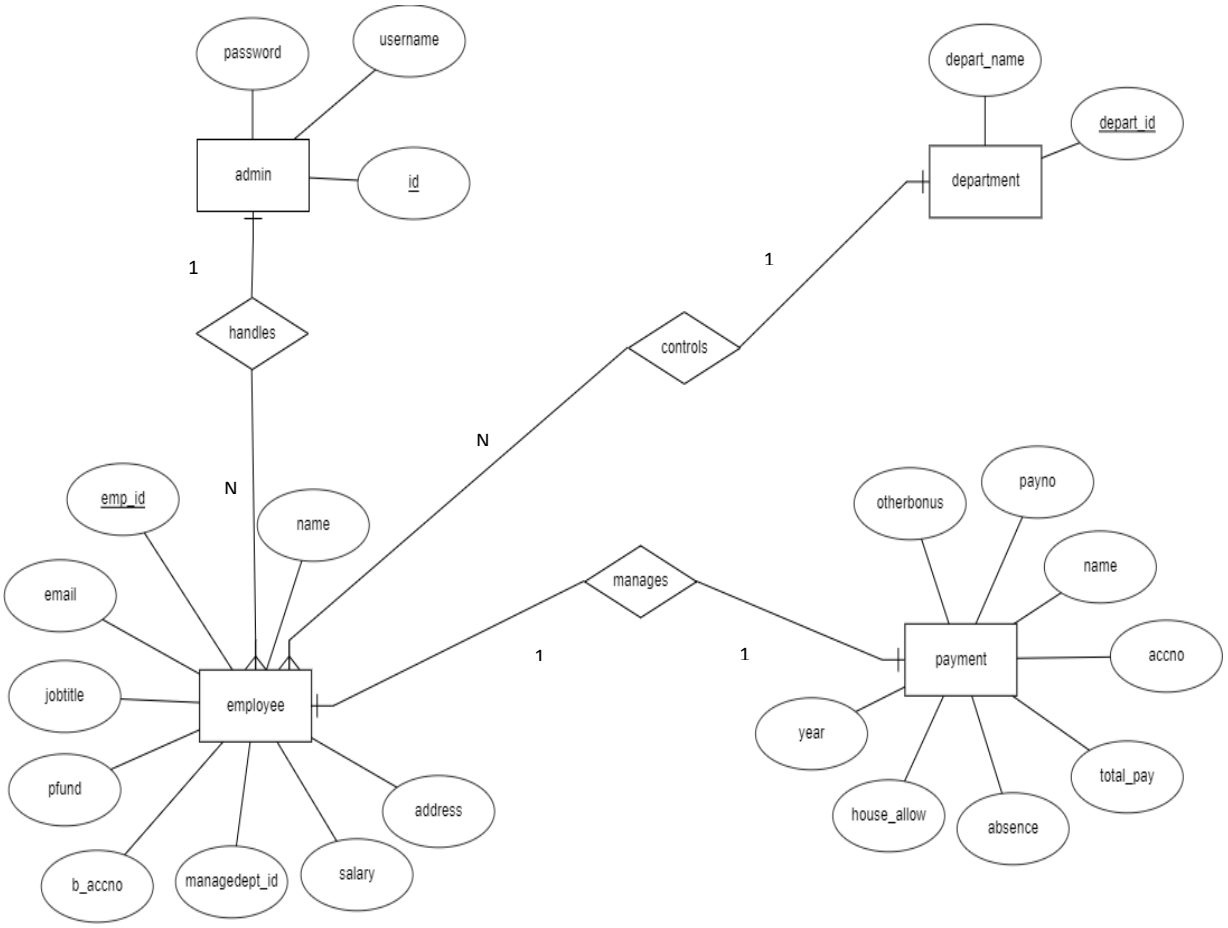
1. Software : XAMPP
2. Operating System : Windows 7 (andother higher version)
3. Front End :HTML, CSS
4. Programming Language : PHP
5. Database Environment : MySQL and PhpMyAdmin
6. Server : APACHE

CHAPTER 3

DESIGN

3.1 ER Diagram and Description

An entity relationship model also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored. The cardinality or fundamental principle of one data aspect with respect to another is a critical feature. The relationship of one to the other must be precise and exact between each other in order to explain how each aspect links together. In simple words, Cardinality is a way to define the relationship between two entities. The ER diagram below shows the relationship between the many tables that exist in the database for the functioning of the Pharmacy Drug Management System



DESCRIPTION

The ER Model figure shows conceptual view of the database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases. So, let's see each entity

DEPARTMENT TABLE

This entity stores the information about department and the attributes are

depart_name , depat_id

EMPLOYEE TABLE

This entity stores the information about employees .The attributes are

Name , emp_id , jobtittle , pfund. b_account ,managedept_id ,salary ,address.

PAYMENT TABLE

This entity stores the information about payments of the employee. The attributes are name, accno,total_pay, absence , year , other bonous .

3.2 SEVEN STEPS FOR E R TO SCHEME CONVERSION

Step 1: Mapping of Regular Entity Types.

For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R. Ifmultiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify secondary (unique) keys of relation R. Knowledge about keys is also kept for indexing purposes and other types of analyses.

Step 2: Mapping of Weak Entity Types.

For each weak entity type W in the ER schema with owner entity type E,

R and include all simple attributes (or simple components of composite attributes) of was attributes of R. In addition, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any. If there is a weak entity type E2 whose owner is also a weak entity type E1, then E1 should be mapped before E2 to determine its primary key first

Step 3: Mapping of Binary 1:1 Relationship Types.

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T thatcorrespond to the entity types participating in R. There are three possible approaches:

1.The foreign key approach.

2. The merged relationship approach, and

The first approach is the most useful and should be followed unless special conditions exist.

1. Foreign key approach:

Choose one of the relations—S, say—and include as a foreign key in S the primary key of T. It isbetter to choose an entity type with total participation in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1

2. Merged relation approach:

An alternative mapping of a 1:1 relationship type is to merge the two entity types and the relationship into a single relation. This is possible when both participations are total, as this wouldindicate that the two tables will have the exact same number of tuples at all times.

3. Reference or relationship relation approach:

The third option is to set up a third relation R for the purpose of cross-referencing the primary keysof the two relations S and T representing the entity types. As we will see, this approach is requiredfor binary M: N relationships. The relation R is called a relationship relation (or sometimes a lookup table), because each tuple in R represents a relationship instance that relates one tuple fromS with one tuple from T. The relation R will include the primary key attributes of S and T as foreignkeys to S and T. The primary key of R will be one of the two foreign keys, and the other foreign key will be a unique key of R. The drawback is having an extra relation, and requiring an extra join operation when combining related tuples from the tables.

Step 4: Mapping of Binary 1: N Relationship Types.

For each regular binary 1: N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to at most one entity instance on the 1-side ofthe relationship type. Include any simple attributes (or simple components of composite attributes)of the 1: N relationship type as attributes of S.

Step 5: Mapping of Binary M: N Relationship Types.

For each binary M: N relationship type R, create a new relation S to represent R. Include as foreignkey attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M: N relationship type (or simple components of composite attributes) as attributes of S.

Notice that we cannot represent an M: N relationship type by a single foreign key attribute in oneof the participating relations (as we did for 1:1 or 1: N relationship types) because of the M: N cardinality ratio; we must create a separate relationship relation S.

Step 6: Mapping of Multivalued Attributes.

For each multivalued attribute A, create a new relation R. This relation R will include anattributecorresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute.

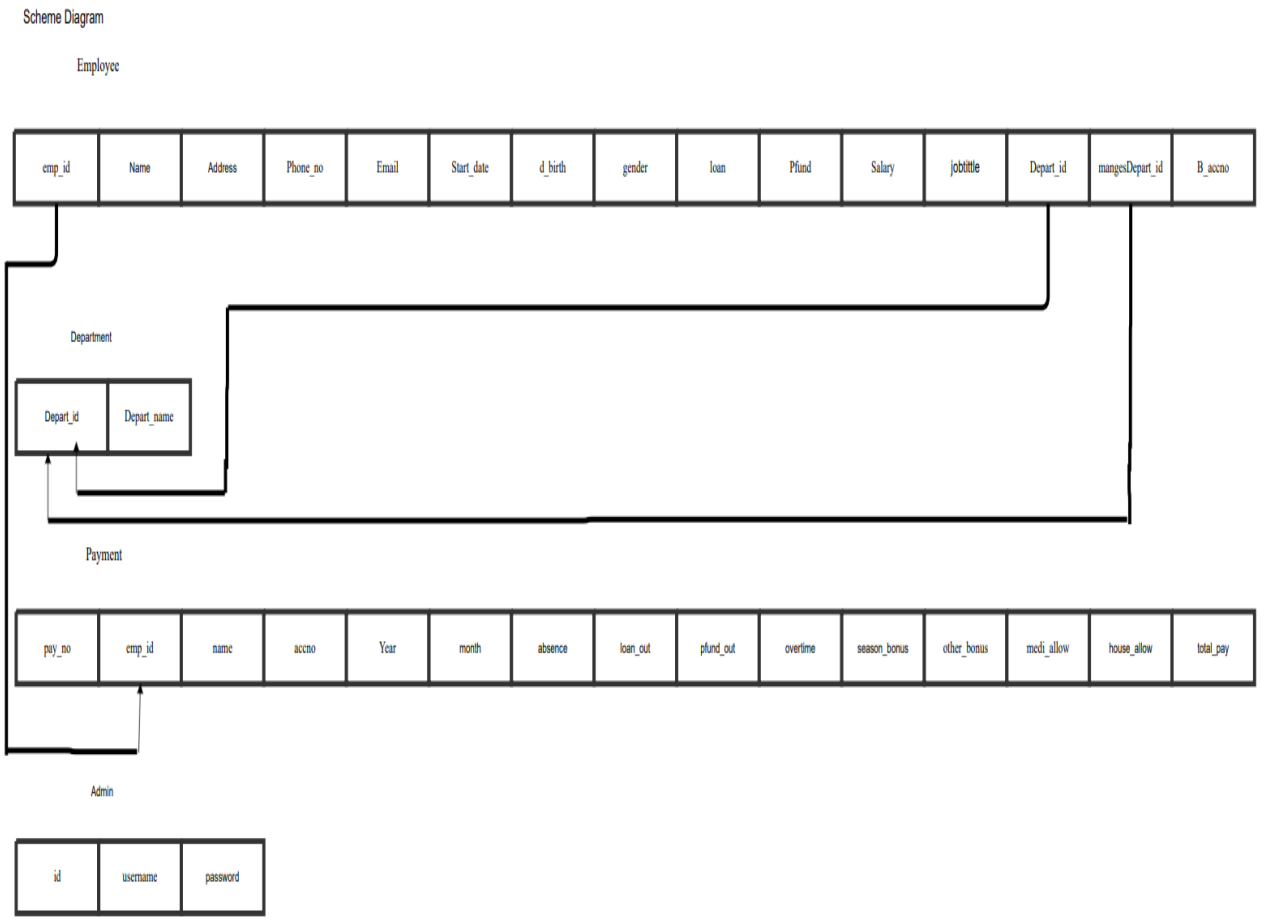
The primary key of R is the combination of A and K. If the multivalued attribute is composite, weinclude its simple components.

Step 7: Mapping of N-array Relationship Types.

For each n-array relationship type R, where n > 2, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entitytypes. Also include any simple attributes of the n-array relationship type (or simple components of composite attributes) as attributes of S. The primary key of S is usually a combination of all theforeign keys that reference the relations representing the participating entity types. However, if thecardinality constraints on any of the entity types E participating in R is 1, then the primary key ofS should not include the foreign key attribute that references the relation E ‘corresponding to E.

3.3 Schema Diagram

In any data model it is important to distinguish between the description of the database and the database itself. The description of a database is called the database schema, whichis specified during database design and is not expected to change frequently. A displayed schema is called a schema diagram. A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints.



3.4 Database Description

3.4.1 Admin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(50)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	password	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Table 3.3.1 Admin Table

The Above table describes all the details about Admin such as admin id, admin name, admin password so that he/she can login to view his profile.

3.4.2 Department:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Depart_id	int(11)			No	None			Change Drop More
2	Depart_name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Table 3.4.2 Department Table

The Above table describes all the details about Department such as dept id, dept_name , so thathe/she can department to manage employees.

3.4.3 Employee

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	emp_id	int(11)			No	None			Change Drop More
2	Name	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
3	Address	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
4	Phone_no	varchar(15)	latin1_swedish_ci		No	None			Change Drop More
5	Email	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
6	Start_date	date			No	None			Change Drop More
7	d_birth	date			No	None			Change Drop More
8	gender	varchar(15)	latin1_swedish_ci		No	None			Change Drop More
9	loan	float			No	None			Change Drop More
10	pfund	float			No	None			Change Drop More
11	salary	float			No	None			Change Drop More
12	jobtitle	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
13	Depart_id	int(11)			No	None			Change Drop More
14	managesDepart_id	int(11)			No	None			Change Drop More
15	b_accno	int(20)			No	None			Change Drop More

Table 3.3.3 employee Table

The Above table describes all the details about the Employee Such As empid,name , address , phoneno ,email, startdate, dbirth, gender, loan, pfund, salary, jobtitle, accno, deptid, managesdeptid

3.4.3 Payment:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	pay_no	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	emp_id	int(11)			No	None			Change Drop More
3	name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	accno	int(20)			No	None			Change Drop More
5	year	int(11)			No	None			Change Drop More
6	month	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
7	absence	int(11)			No	None			Change Drop More
8	loan_cut	float			No	None			Change Drop More
9	pfund_cut	float			No	None			Change Drop More
10	overtime	float			No	None			Change Drop More
11	season_bonus	float			No	None			Change Drop More
12	other_bonus	float			No	None			Change Drop More
13	medi_allow	float			No	None			Change Drop More
14	house_allow	float			No	None			Change Drop More
15	total_pay	float			No	None			Change Drop More

Table 3.3.4 Payment Table

The Above table describes all the details about the payment such as payno, empid, name, accno, year, month, absence, loancut, pfundcut, overtime, seasonbonus Otherbonus,mediallow, houseallow,totalpay.

CHAPTER 4

IMPLEMENTATION

IMPLEMENTATION CODE

4.1.1 CONNECTION CODE FOR FRONT END TO BACK END

```
<?php
$DB_host = "localhost";

$DB_user = "root";

$DB_pass = "";

$DB_name = "regis";try

{

$DB_con = new
PDO("mysql:host={$DB_host};dbname={$DB_name}",$DB_user,$DB_pass);
$DB_con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

}
catch(PDOException $e)
{

$e->getMessage();

}
?>
```

4.2 SQL STATEMENTS

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Feb 08, 2022 at 05:25 AM
-- Server version: 10.4.22-MariaDB
-- PHP Version: 7.4.27

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
--
-- Database: `payroll`
--


-- --------------------------------------------------------


--
-- Table structure for table `admin`
```

```
--
CREATE TABLE `admin` (
  `id` int(50) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`id`, `username`, `password`) VALUES
(1, '1604071', '12345'),
(2, '1604070', 'montu');

-- --------------------------------------------------------

--
-- Table structure for table `department`
--

CREATE TABLE `department` (
  `Depart_id` int(11) NOT NULL,
  `Depart_name` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `department`
```

```
--
INSERT INTO `department` (`Depart_id`, `Depart_name`) VALUES
(108, 'codsection'),
(109, 'ise'),
(2005, 'ece');

-- --------------------------------------------------------

--
-- Table structure for table `employee`
--

CREATE TABLE `employee` (
  `Employee_id` int(11) NOT NULL,
  `Name` varchar(200) NOT NULL,
  `Address` varchar(200) NOT NULL,
  `Phone_no` varchar(15) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Start_date` date NOT NULL,
  `d_birth` date NOT NULL,
  `gender` varchar(15) NOT NULL,
  `loan` float NOT NULL,
  `pfund` float NOT NULL,
  `salary` float NOT NULL,
  `jobtitle` varchar(50) NOT NULL,
  `Depart_id` int(11) NOT NULL,
  `managesDepart_id` int(11) NOT NULL,
  `b_accno` int(20) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `employee`
--
INSERT INTO `employee` (`Employee_id`, `Name`, `Address`, `Phone_no`, `Email`,
`Start_date`, `d_birth`, `gender`, `loan`, `pfund`, `salary`, `jobtitle`, `Depart_id`,
`managesDepart_id`, `b_accno`) VALUES
(100, 'hanumanthayya', 'raichur', '7899517043', 'abc@gmail.com', '2022-01-27', '2000-11-
04', 'male', 0, 2500, 50000, 'executive', 108, 108, 123456789),
(102, 'ARUNA B', ' davangere', '7899517012', 'arunab978@gmail.com', '2022-01-28',
'2001-03-15', 'male', 0, 0, 100000, 'manager', 108, 108, 2147483647);

-- --------------------------------------------------------

--
-- Table structure for table `payment`
--

CREATE TABLE `payment` (
  `pay_no` int(11) NOT NULL,
  `emp_id` varchar(100) NOT NULL,
  `name` varchar(100) NOT NULL,
  `accno` int(20) NOT NULL,
  `year` int(11) NOT NULL,
  `month` varchar(50) NOT NULL,
  `absence` int(11) NOT NULL,
  `loan_cut` float NOT NULL,
  `pfund_cut` float NOT NULL,
  `overtime` float NOT NULL,
```

```
  `season_bonus` float NOT NULL,
  `other_bonus` float NOT NULL,
  `medi_allow` float NOT NULL,
  `house_allow` float NOT NULL,
  `total_pay` float NOT NULL
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `payment`
--

INSERT INTO `payment` (`pay_no`, `emp_id`, `name`, `accno`, `year`, `month`,
`absence`, `loan_cut`, `pfund_cut`, `overtime`, `season_bonus`, `other_bonus`,
`medi_allow`, `house_allow`, `total_pay`)
 VALUES
(1, '160403', '', 0, 2018, 'december', 2, 0, 0, 6, 2000, 0, 0, 0, 3400),
(2, '1604023', 'Rukon', 236954128, 2018, 'december', 2, 0, 1000, 6, 2000, 0, 1200, 3200,
46800),
(3, '100', 'hanumanthayya', 123456789, 2021, 'november', 2, 0, 2500, 10, 1000, 1500,
3000, 8000, 113600);

--
-- Indexes for dumped tables
--
--
-- Indexes for table `admin`
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);
```

```
--
-- Indexes for table `department`
--
ALTER TABLE `department`
 ADD PRIMARY KEY (`Depart_id`);


--
-- Indexes for table `employee`
--
ALTER TABLE `employee`
 ADD PRIMARY KEY (`Employee_id`),
 ADD KEY `Depart_id` (`Depart_id`),
 ADD KEY `managesDepart_id` (`managesDepart_id`);


--
-- Indexes for table `payment`
--
ALTER TABLE `payment`
 ADD PRIMARY KEY (`pay_no`);


--
-- AUTO_INCREMENT for dumped tables
--


--
-- AUTO_INCREMENT for table `admin`
--
ALTER TABLE `admin`

MODIFY `id` int(50) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
```

```
--
-- AUTO_INCREMENT for table `payment`
--
ALTER TABLE `payment`
 MODIFY `pay_no` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=4;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `employee`
--
ALTER TABLE `employee`
 ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`Depart_id`)
REFERENCES `department` (`Depart_id`) ON DELETE CASCADE,
ADD CONSTRAINT `employee_ibfk_2` FOREIGN KEY (`managesDepart_id`)
REFERENCES `department` (`Depart_id`) ON DELETE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;
```

CHAPTER 5

SNAP SHOTS

5.1 Admin login page:

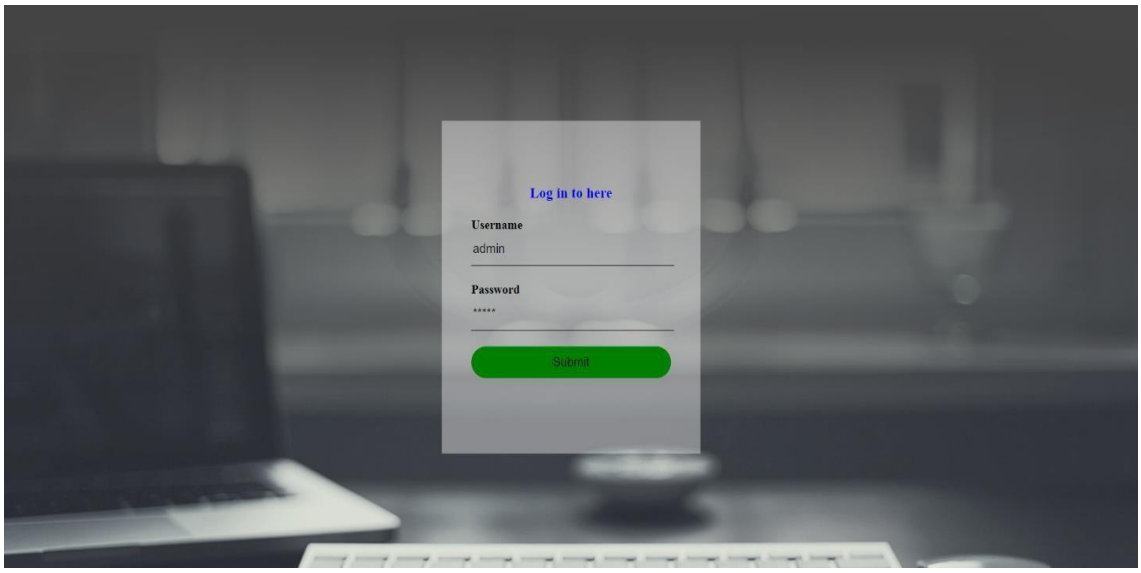


Table 5.1. login page

Here the admin or user can log on to the page after getting his/her credentials like username andthe password for accessing next contents.

5.2 Home page:

Table 5.2 home page

After login we get a home page which has list of contents where we can access different details through the navigation bar.

5.3 department:

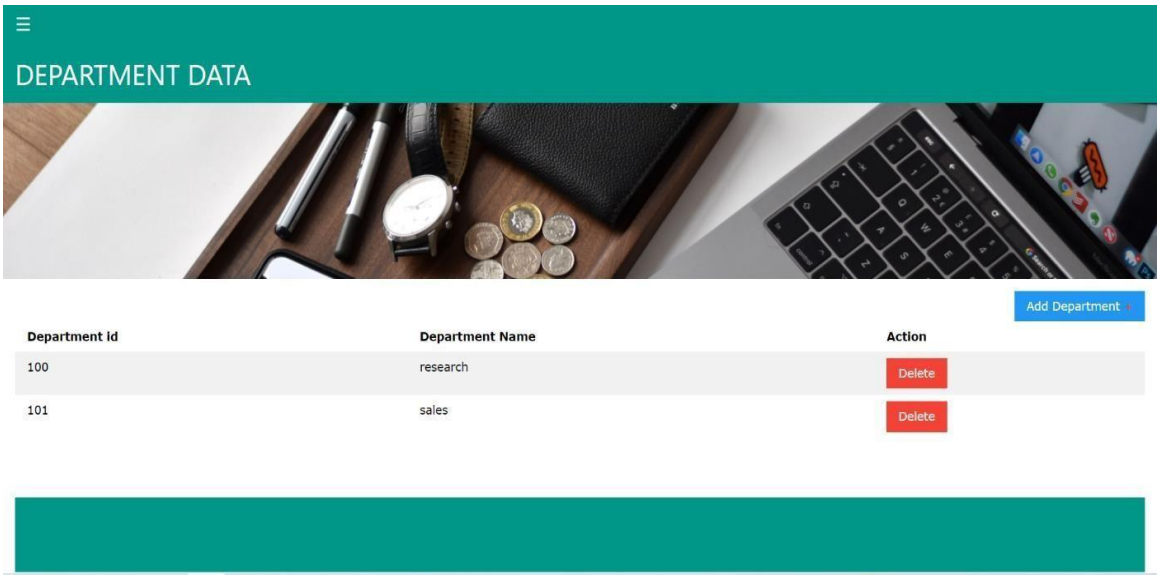


Table 5.3 department page

This is the department page where we can add department data and we can also remove the department details.

5.4 employee:

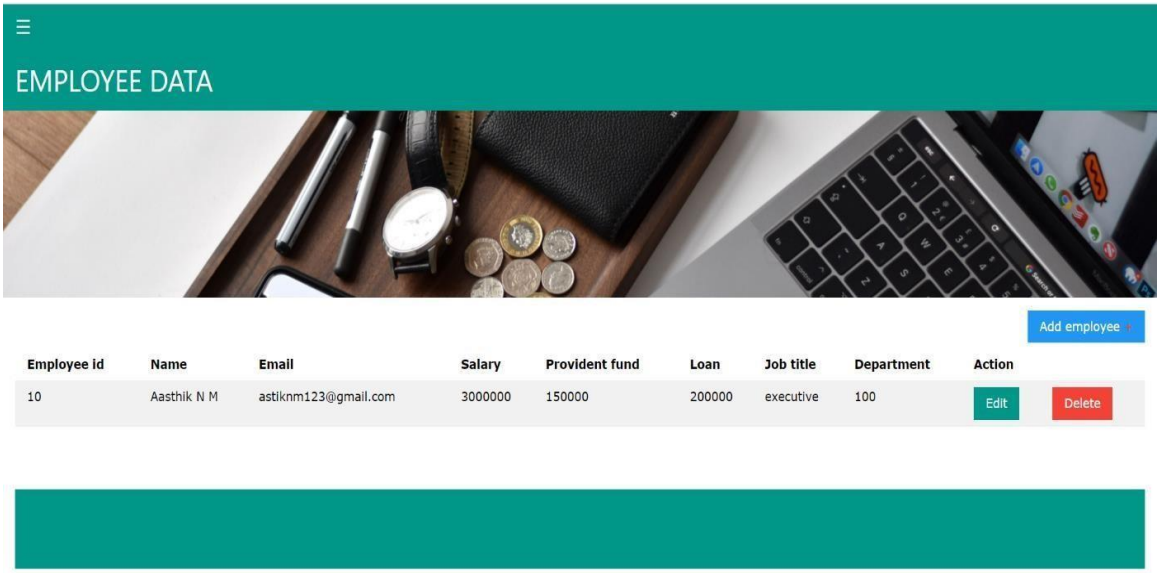


Table 5.4 employee page

This is the employee page where we can add employee data. We can also edit and remove the employee details

5.5 payments:

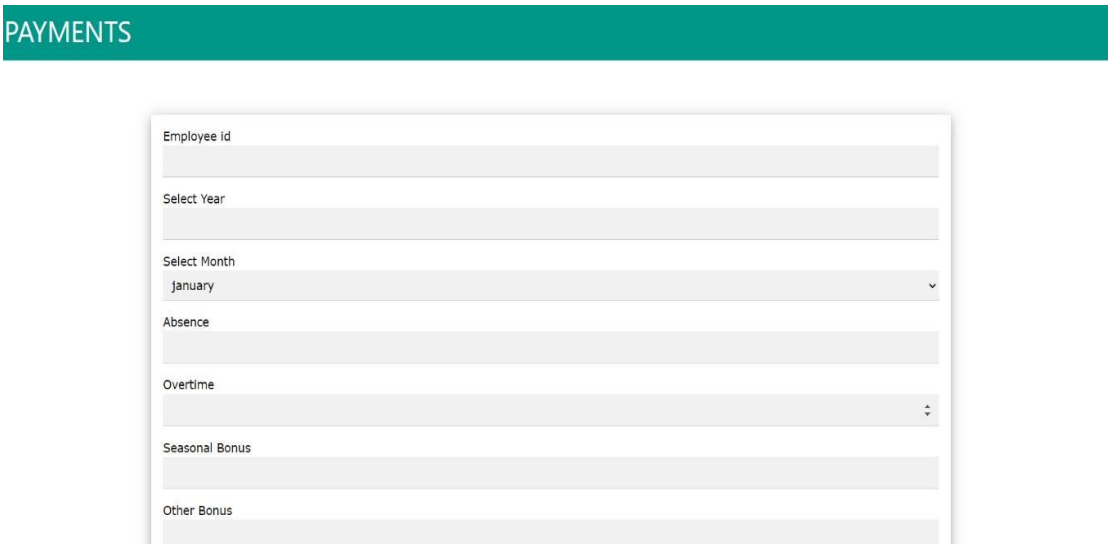


Table 5.5 payment page

This is the payment page where we can make payments with the help of employee details

5.6 pay slip:

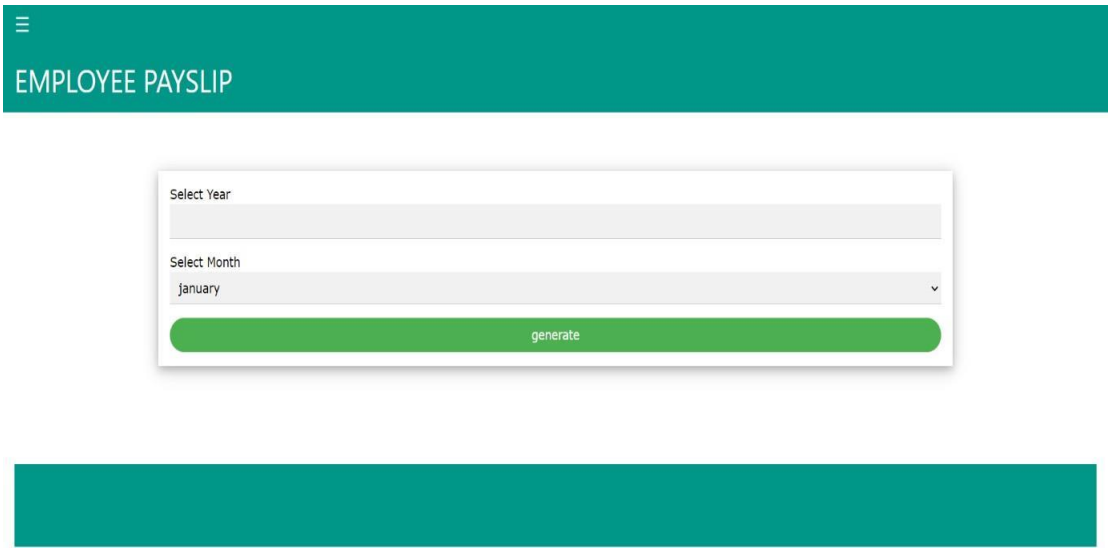


Table 5.6 pay slip

Here we can generate a pay slip for each employees working in the department

5.7 set salary:

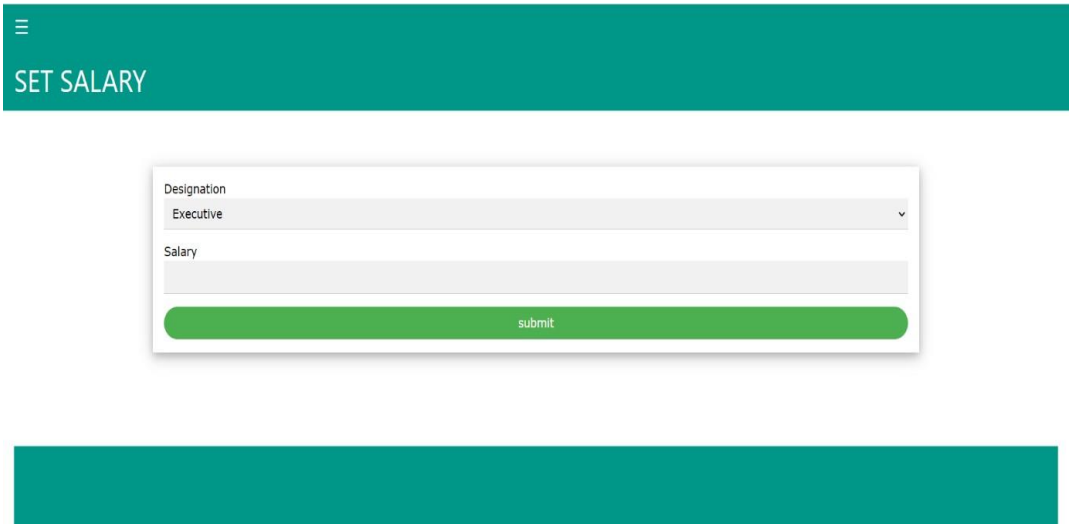


Table 5.7 set salary

Here we can set the salary of an employee who are working in the department.

5.8 payment history:

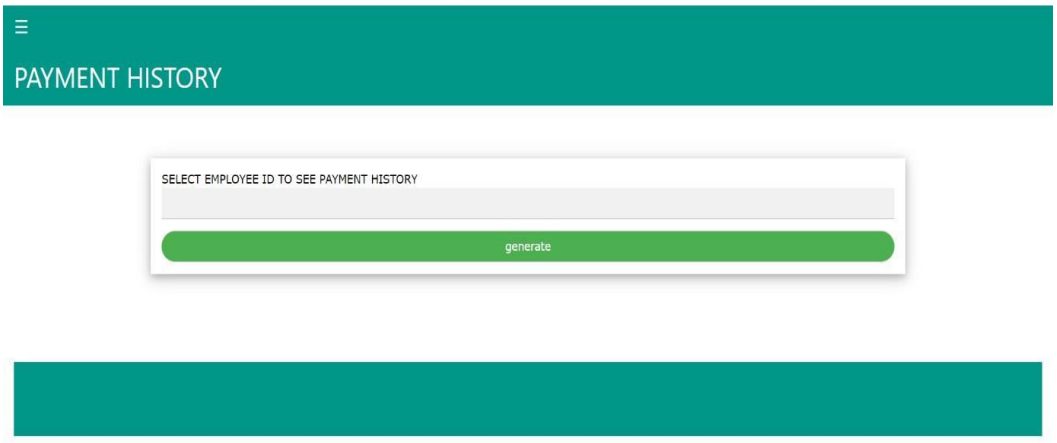


Table 5.8 payment history

This is page is for checking the payment history of an employee

CONCLUSION

This project 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. Accordingto the requested requirement the admin can add, manipulate, update and delete all employee data in his organization. The admin can add new departments and delete them. 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.

REFERENCES

BOOKS:

1. Fundamentals of database systems, Ramez Elmasri and S B Navathe, 7th Edition, 2017,Pearson.

2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

3. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and

4. Silberschatz Korth and Sudharshan, Database System Concepts, 6thEdition, McGraw Hill, 2013.

LIST OF WEBSITES:

- www.stackoverflow.com
- https://www.w3schools.com
- www.geeksforgeeks.org