**EMPLOYEE MANAGEMENT SYSTEM**

**WITH EXPORT TO EXCEL SHEET**

**A Project Report**

***Submitted by:***

**SURYA BANSAL (1420147)**

***In partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**at**



**RIMT INSTITUTE OF ENGINEERING AND TECHNOLOGY,**

**MANDI GOBINDGARH, PUNJAB(INDIA)**

**ABSTRACT**

An organized and systematic office solution is essential for all corporations and organizations. There are many departments of administration for the maintenance of employee information and databases in any department/city. All these departments provide various records regarding their employees. Most of these track records need to maintain information about the employee. This information could be the personal details like employee name, age, contact etc or corporate information related to departments like designation or operating city etc.

All the modules in company administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules. For example when someone needs any kind of details related to an employee and many other details. So it needs to contact all the modules that are office, department and finance and status of employees.

With that in mind, i overhauled the existing Employee Database Management System and made necessary improvement to streamline the processes using only python to code the program which give it its light weight and efficient processing with mathematical modules. Administrators using the system will find that the process of recording and retrieving employees information and their operating cities and their designation. In general, this project aims to enhance efficiency and at the same time maintain information accurateness. Later in this report, features and improvement that allow achievement to this goal will be demonstrated and highlighted.

**DECLARATION**

I hereby declare that the project work entitled “EMPLOYEE MANAGEMENT SYSTEM WITH EXPORT TO EXCEL SHEET” is an authentic record of my own work as requirements of Major Project for the award of degree of B.Tech. (Computer Science & Engineering), **RIMT Institute of Engineering & Technology**, Mandi Gobindgarh, under the guidance of CSE Department.

**(Signature of student)**

**Place:**

**Date:**

**ACKNOWLEDGEMENT**

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I would be grateful to my project guide the ‘CSE Department’ for the guidance, inspiration and constructive suggestions that were helpful to me in the preparation of this project.

I also thank my professors who have helped in successful completion of this project.

**TABLE OF CONTENTS**

Title page i

Abstract ii

Declaration iii

Acknowledgement iv

|  |  |  |
| --- | --- | --- |
| Chapter no. | Title | Page no. |
| Chapter 1 | **INTRODUCTION**   * 1. Introduction to python & software req.   1.1.1 History of python  1.1.2 Versions of pyhton  1.1.3 Release Dates  1.1.4 Openpyxl Library  1.1.5 Tkinter Library  1.1.6 PYMySQL | 6-22  6-14 |
|  | * 1. Introduction to project   1.2.1 Introduction  1.2.2 Modules  1.2.3 Features  1.2.4 XAMPP  1.2.5 MySQL | 14-22 |
| Chapter 2 | **OBJECTIVES**  2.1 Purpose  2.2 Scope of the project | 23-24  23  24 |
| Chapter 3 | **IMPLEMENTATIONS**  3.1 SYSTEM REQUIREMENTS  3.2 DFD  3.3 ER DIAGRAM | 25-27  25  26  27 |
| Chapter 4 | **CONCLUSION & FUTURE SCOPE** | 28-29 |
| Chapter 5 | **SNAPSHOTS** | 30-43 |

**CHAPTER 1: INTRODUCTION**

* 1. **Introduction to python**
     1. **History of python**

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rosum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system. Van Rossum remains Python's principal author. His continuing central role in Python's development is reflected in the title given him by the Python community: Benevolent Dictator For Life (BDFL).

On the origins of Python, Van Rossum wrote in 1996:

|  |  |  |
| --- | --- | --- |
|  | In December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/Chackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of Monty Python's Flying Circus). |  |

* + 1. **VERSIONS**

**VERSION 1:**

Python reached version 1.0 in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce. Van Rossum stated that "Python acquired lambda, reduce(), filter() and map(), courtesy of a Lisp hacker who missed them and submitted working patches”.

The last version released while Van Rossum was at CWI was Python 1.2. In 1995, Van Rossum continued his work on Python at the Corporation for National Research Initiatives(CNRI) in Reston, Virginia whence he released several versions.

By version 1.4, Python had acquired several new features. Notable among these are the Modula-3 inspired keyword arguments (which are also similar to Common Lisp's keyword arguments) and built-in support for complex numbers. Also included is a basic form of data hiding by name mangling, though this is easily bypassed.

During Van Rossum's stay at CNRI, he launched the Computer Programming for Everybody (CP4E) initiative, intending to make programming more accessible to more people, with a basic "literacy" in programming languages, similar to the basic English literacy and mathematics skills required by most employers. Python served a central role in this: because of its focus on clean syntax, it was already suitable, and CP4E's goals bore similarities to its predecessor, ABC. The project was funded by DARPA. As of 2007, the CP4E project is inactive, and while Python attempts to be easily learnable and not too arcane in its syntax and semantics, reaching out to non-programmers is not an active concern.

**BeOpen:**

In 2000, the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. CNRI requested that a version 1.6 be released, summarizing Python's development up to the point at which the development team left CNRI. Consequently, the release schedules for 1.6 and 2.0 had a significant amount of overlap. Python 2.0 was the only release from BeOpen.com. After Python 2.0 was released by BeOpen.com, Guido van Rossum and the other PythonLabs developers joined Digital Creations.

The Python 1.6 release included a new CNRI license that was substantially longer than the CWI license that had been used for earlier releases. The new license included a clause stating that the license was governed by the laws of the State of Virginia. The Free Software Foundation argued that the choice-of-law clause was incompatible with the GNU General Public License. BeOpen, CNRI and the FSF negotiated a change to Python's free software license that would make it GPL-compatible. Python 1.6.1 is essentially the same as Python 1.6, with a few minor bug fixes, and with the new GPL-compatible license.

**VERSION 2:**

Python 2.0 introduced list comprehensions, a feature borrowed from the functional programming languages SETL and Haskell. Python's syntax for this construct is very similar to Haskell's, apart from Haskell's preference for punctuation characters and Python's preference for alphabetic keywords. Python 2.0 also introduced a garbage collection system capable of collecting reference cycles.

Python 2.1 was close to Python 1.6.1, as well as Python 2.0. Its license was renamed Python Software Foundation License. All code, documentation and specifications added, from the time of Python 2.1's alpha release on, is owned by the Python Software Foundation (PSF), a non-profit organization formed in 2001, modeled after the Apache Software Foundation. The release included a change to the language specification to support nested scopes, like other statically scoped languages. (The feature was turned off by default, and not required, until Python 2.2.)

A major innovation in Python 2.2 was the unification of Python's types (types written in C) and classes (types written in Python) into one hierarchy. This single unification made Python's object model purely and consistently object oriented. Also added were generators which were inspired by Icon.

Python 2.6 was released to coincide with Python 3.0, and included some features from that release, as well as a "warnings" mode that highlighted the use of features that were removed in Python 3.0. Similarly, Python 2.7 coincided with and included features from Python 3.1, which was released on June 26, 2009. Parallel 2.x and 3.x releases then ceased, and Python 2.7 was the last release in the 2.x series. In November 2014, it was announced that Python 2.7 would be supported until 2020, but users were encouraged to move to Python 3 as soon as possible.

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release, the development process became more transparent and community-backed.

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go transcompiler. The Register speculated that this was in response to Python 2.7's planned end-of-life, but Google cited performance under concurrent workloads as their only motivation.

**VERSION 3:**

Python 3.0 (also called "Python 3000" or "Py3K") was released on December 3, 2008. It was designed to rectify fundamental design flaws in the language—the changes required could not be implemented while retaining full backwards compatibility with the 2.x series, which necessitated a new major version number. The guiding principle of Python 3 was: "reduce feature duplication by removing old ways of doing things".

Python 3.0 was developed with the same philosophy as in prior versions. However, as Python had accumulated new and redundant ways to program the same task, Python 3.0 had an emphasis on removing duplicative constructs and modules, in keeping with "There should be one— and preferably only one —obvious way to do it".

Nonetheless, Python 3.0 remained a multi-paradigm language. Coders still had options among object-orientation, structured programming, functional programming and other paradigms, but within such broad choices, the details were intended to be more obvious in Python 3.0 than they were in Python 2.x.

**COMPATIBILITY ISSUES:**

Python 3.0 broke backward compatibility, and much Python 2 code does not run unmodified on Python 3. Python's dynamic typing combined with the plans to change the semantics of certain methods of dictionaries, for example, made perfect mechanical translation from Python 2.x to Python 3.0 very difficult. A tool called "2to3" does the parts of translation that can be done automatically. At this, 2to3 appeared to be fairly successful, though an early review noted that there were aspects of translation that such a tool would never be able to handle. Prior to the roll-out of Python 3, projects requiring compatibility with both the 2.x and 3.x series were recommended to have one source (for the 2.x series), and produce releases for the Python 3.x platform using 2to3. Edits to the Python 3.x code were discouraged for so long as the code needed to run on Python 2.x. This is no longer recommended; as of 2012 the preferred alternative is to create a single code base that can run under both Python 2 and 3 using compatibility modules.

**FEATURES OF PYTHON3:**

Some of the major changes included for Python 3.0 were:

* Changing print so that it is a built-in function, not a statement. This made it easier to change a module to use a different print function, as well as making the syntax more regular. In Python 2.6 and 2.7 print() is available as a builtin but is masked by the print statement syntax, which can be disabled by entering from \_\_future\_\_ import print\_function at the top of the file.
* Removal of the Python 2 input function, and the renaming of the raw\_input function to input. Python 3's input function behaves like Python 2's raw\_input function, in that the input is always returned as a string rather than being evaluated as an expression.
* Moving reduce (but not map or filter) out of the built-in namespace and into functools (the rationale being that operations using *reduce* are expressed more clearly using an accumulation loop).
* Adding support for optional function annotations that can be used for informal type declarations or other purposes.
* Unifying the str/unicode types, representing text, and introducing a separate immutable bytes type; and a mostly corresponding mutable bytearray type, both of which represent arrays of bytes.
* Removing backward-compatibility features, including old-style classes, string exceptions, and implicit relative imports.
* A change in integer division functionality. (In Python 2, 5 / 2 is 2. In Python 3, 5 / 2 is 2.5, and 5 // 2 is 2).

Subsequent releases in the Python 3.x series have included additional, substantial new features; all ongoing development of the language is done in the 3.x series.

**1.1.3 RELEASE DATES:**

* Python 1.0 - January 1994
  + Python 1.2 - April 10, 1995
  + Python 1.3 - October 12, 1995
  + Python 1.4 - October 25, 1996
  + Python 1.5 - December 31, 1997
  + Python 1.6 - September 5, 2000
* Python 2.0 - October 16, 2000
  + Python 2.1 - April 17, 2001
  + Python 2.2 - December 21, 2001
  + Python 2.3 - July 29, 2003
  + Python 2.4 - November 30, 2004
  + Python 2.5 - September 19, 2006
  + Python 2.6 - October 1, 2008
  + Python 2.7 - July 3, 2010
* Python 3.0 - December 3, 2008
  + Python 3.1 - June 27, 2009
  + Python 3.2 - February 20, 2011
  + Python 3.3 - September 29, 2012
  + Python 3.4 - March 16, 2014
  + Python 3.5 - September 13, 2015
  + Python 3.6 - December 23, 2016
    1. **OPENPYXL LIBRARY :**

Openpyxl is a Python library for reading and writing Excel 2010 xlsx/xlsm/xltx/xltm files. It was born from lack of existing library to read/write natively from Python the Office Open XML format.

Excel is a popular and powerful spreadsheet application for Windows. The openpyxl module allows your Python programs to read and modify Excel spreadsheet files. For example, you might have the boring task of copying certain data from one spreadsheet and pasting it into another one. Or you might have to go through thousands of rows and pick out just a handful of them to make small edits based on some criteria. Or you might have to look through hundreds of spreadsheets of department budgets, searching for any that are in the red. These are exactly the sort of boring, mindless spreadsheet tasks that Python can do for you.

Although Excel is proprietary software from Microsoft, there are free alternatives that run on Windows, OS X, and Linux. Both LibreOffice Calc and OpenOffice Calc work with Excel’s .xlsx file format for spreadsheets, which means the openpyxlmodule can work on spreadsheets from these applications as well. You can download the software from https://www.libreoffice.org/ and http://www.openoffice.org/, respectively. Even if you already have Excel installed on your computer, you may find these programs easier to use. The screenshots in this chapter, however, are all from Excel 2010 on Windows 7.

First, let’s go over some basic definitions: An Excel spreadsheet document is called a workbook. A single workbook is saved in a file with the .xlsx extension. Each workbook can contain multiple sheets (also called worksheets). The sheet the user is currently viewing (or last viewed before closing Excel) is called the active sheet.

Each sheet has columns (addressed by letters starting at A) and rows (addressed by numbers starting at 1). A box at a particular column and row is called a cell. Each cell can contain a number or text value. The grid of cells with data makes up a sheet.

Installing the openpyxl Module

Python does not come with OpenPyXL, so you’ll have to install it. Follow the instructions for installing third-party modules in Appendix A; the name of the module is openpyxl. To test whether it is installed correctly, enter the following into the interactive shell:

>>> **import openpyxl**

New versions should stay backward compatible, you can check out the full documentation for OpenPyXL at http://openpyxl.readthedocs.org/.

* + 1. **TKINTER LIBRARY:**

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter).

There are several popular GUI library alternatives available, such as wxPython, PyQt (PySide), Pygame, Pyglet, and PyGTK.

Tkinter is free software released under a Python license.

**Window**

This term has different meanings in different contexts, but in general it refers to a rectangular area somewhere on the user's display screen.

**Top Level Window**

A window that exists independently on the screen. It will be decorated with the standard frame and controls for the desktop manager. It can be moved around the desktop, and can usually be resized.

**Widget**

The generic term for any of the building blocks that make up an application in a graphical user interface. Examples of widgets: buttons, radiobuttons, text fields, frames, and text labels.

**Frame**

In Tkinter, the Frame widget is the basic unit of organization for complex layouts. A frame is a rectangular area that can contain other widgets.

**Child and parent**

When any widget is created, a parent-child relationship is created. For example, if you place a text label inside a frame, the frame is the parent of the label.

* + 1. **PYMYSQL LIBRARY :**

PyMySQL is a python library build to act as a connector between the python program and MySQL database.

Its a MySQL connector for python 3 version that works with

* Pypi
* Django 1.6
* CPython
  1. **INTRODUCTION TO PROJECT :**

**1.2.1 INTRODUCTION :**

The presented project here is made in view to simplyfy the system of managing the information of the employees by the corporations which includes storing and maintaing the information etc. i.e **“EMPLOYEE MANAGEMENT SYSTEM WITH EXPORT TO EXCEL SHEET”.**

This project can be widely used form small businesses to large corporation to keep simple data related to their employees. This project can later be evolved to much more utility easily by adding new required modules. This project’s best quality is its simplycity and easy to understand working.

This is the ‘second version’ of my Emplyee Management System. I have developed it from a previous verion which i developed in my last semester as a lab project. And i hope to develop it in more ‘comercialized format’ next time.

Its written entirely in python language. Using python is benificial because:

* Its easier to code
* Its source code is ½ the size of other languages like C++ & java
* It uses a interpreter instead of a compiler which helps us to pin-point and remove the error present in the code
* It comes with a very large group of libraries which helps to eliminate almost any problem that comes forward
* Libraries like Openpyxl and Pymysql are very beneficial in this project
* Openpyxl library is used to export the MySQL data to Excel sheets
* Pymysql is a connector library used to connect python program to MySQL database

**1.2.2 MODULES OF THE PROJECT :**

1. **ADDING A RECORD :**

This module is responsible for adding the employee’s data into the database. Its have 9 fields which then later get stored in two different tables:

* Emp\_ID : Employee ID, also a primary key which helps to keep the info in both table neat by avoiding ambiguity and false info
* First\_Name
* Last\_Name
* Age
* Sex
* Designation
* Operating City
* Income
* Contact

1. **SEARCHING A RECORD :**

This module helps us to find the information about a specific employee using the Emp\_ID. Its serves as a primary key, so its also unique and makes it easier to find data in those two tables. It then shows the result in a textpad window.

1. **DELETING A RECORD :**

This module helps us to delete the record using its Emp\_ID as search constraint. It deletes the data in both the tables.

1. **UPDATING A RECORD :**

This module allows us to update the data about the employee. This modules also uses Emp\_ID to specify the employee and then updates its data.

1. **DISPLAY THE DATA :**

This modules extract all the data present in the database and show us in textpad window. It shows data of both the tables separatly to improve readability.

1. **SAVING THE DATA INTO EXCEL SHEET :**

This module uses the OPENPYXL library to export all the data from MySQL database to the excel sheets. We achieve the exporting with the format being defined in coding adn openpycl library being used as a carrier / connector between excel, python, and MySQL database.

1. **CLEAR MODULE :**

This module is used to clear all the data enterred into the text field to save time to enter new data into fields.

1. **EXIT MODULE :**

It is used to close the main program window or sub windows. Its programmed to save / commit the data saved into database and closing the connection created by the PYMYSQL connect to ensure data safety before terminating the program.

1. **LOGIN MODULE :**

A login module is used to ask for valid login credentials at startup of the program. This way only authentic users will be able to access the portal

**1.2.3 FEATURES :**

* It has login feature in which user have to be authenticated before using the app
* It can add records of employee in two separate orders.
* It has two tables separating the employee’s data
  + Personal data like age, sex, contact etc.
  + Corporate data like operating city, designation, income etc.
* It has Employee ID as primary key that links both the tables, helps avoiding ambiguity
* It can search data in the tables using Employee ID
* It can delete any record in both tables at same time using Employee ID constraint
* We can also Update details of the present employee records.
* We have a option to save the record present in the database to be saved on an Excel Sheet for more utility.
* Its best feature is its simple design and easy to understand working.
* Its very light weight and a efficient tool
* We can further expand this project by adding modules according to our requirement in future

* + 1. **INTRODUCTION TO XAMPP :**

XAMPP  is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the  PHP  and  Perl programming languages.

XAMPP stands for:

* (X) Cross-Platform Symbol
* (A) Apache
* (M) MariaDB
* (P) PHP
* (P) Perl

It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. XAMPP has the ability to serve web pages on the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). A special tool is provided to [password-protect](https://en.wikipedia.org/wiki/Password) the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in [MariaDB](https://en.wikipedia.org/wiki/MariaDB) and [SQLite](https://en.wikipedia.org/wiki/SQLite) among others.

Once XAMPP is installed, it is possible to treat a [localhost](https://en.wikipedia.org/wiki/Localhost) like a remote host by connecting using an [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol) client. Using a program like [FileZilla](https://en.wikipedia.org/wiki/FileZilla) has many advantages when installing a [content management system](https://en.wikipedia.org/wiki/Content_management_system) (CMS) like [Joomla](https://en.wikipedia.org/wiki/Joomla) or [WordPress](https://en.wikipedia.org/wiki/WordPress). It is also possible to connect to localhost via FTP with an [HTML editor](https://en.wikipedia.org/wiki/HTML_editor).

* + 1. **INTRODUCTION TO MySQL :**

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter,[8] and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr and YouTube.

MySQL is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). Its SQL parser is written in [yacc](https://en.wikipedia.org/wiki/Yacc), but it uses a home-brewed [lexical analyzer](https://en.wikipedia.org/wiki/Lexical_analysis). MySQL works on many [system platforms](https://en.wikipedia.org/wiki/System_platform), including [AIX](https://en.wikipedia.org/wiki/AIX_operating_system), [BSDi](https://en.wikipedia.org/wiki/BSD/OS), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD), [HPUX](https://en.wikipedia.org/wiki/HP-UX), [eComStation](https://en.wikipedia.org/wiki/EComStation), [i5/OS](https://en.wikipedia.org/wiki/IBM_i5/OS), [IRIX](https://en.wikipedia.org/wiki/IRIX), [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS), [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [NetBSD](https://en.wikipedia.org/wiki/NetBSD), [Novell NetWare](https://en.wikipedia.org/wiki/Novell_NetWare), [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD), [OpenSolaris](https://en.wikipedia.org/wiki/OpenSolaris), [OS/2](https://en.wikipedia.org/wiki/OS/2) Warp, [QNX](https://en.wikipedia.org/wiki/QNX), [Oracle Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)), [Symbian](https://en.wikipedia.org/wiki/Symbian), [SunOS](https://en.wikipedia.org/wiki/SunOS), [SCO OpenServer](https://en.wikipedia.org/wiki/SCO_OpenServer), SCO [UnixWare](https://en.wikipedia.org/wiki/UnixWare), Sanos and [Tru64](https://en.wikipedia.org/wiki/Tru64). A port of MySQL to [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS) also exists.

The MySQL server software itself and the client libraries use [dual-licensing](https://en.wikipedia.org/wiki/Dual_license) distribution. They are offered under GPL version 2, beginning from 28 June 2000 (which in 2009 has been extended with a [FLOSS](https://en.wikipedia.org/wiki/Alternative_terms_for_free_software) License Exception) or to use a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organisations exist to provide support and services, including [MariaDB](https://en.wikipedia.org/wiki/MariaDB) and [Percona](https://en.wikipedia.org/wiki/Percona).

MySQL has received positive reviews, and reviewers noticed it "performs extremely well in the average case" and that the "developer interfaces are there, and the documentation (not to mention feedback in the real world via Web sites and the like) is very, very good". It has also been tested to be a "fast, stable and true multi-user, multi-threaded sql database server".

**MySQL VERSION TIMELINE:**

* First internal release on 23 May 1995
* Version 3.19: End of 1996, from www.tcx.se
* Version 3.20: January 1997
* Windows version was released on 8 January 1998 for Windows 95 and NT
* Version 3.21: production release 1998, from www.mysql.com
* Version 3.22: alpha, beta from 1998
* Version 3.23: beta from June 2000, production release 22 January 2001
* Version 4.0: beta from August 2002, production release March 2003 (unions).
* Version 4.01: beta from August 2003, Jyotiadopts MySQL for database tracking
* Version 4.1: beta from June 2004, production release October 2004 (R-trees and B-trees, subqueries, prepared statements).
* Version 5.0: beta from March 2005, production release October 2005 (cursors, stored procedures, triggers, views, XA transactions).

The developer of the Federated Storage Engine states that "The Federated Storage Engine is a proof-of-concept storage engine", but the main distributions of MySQL version 5.0 included it and turned it on by default. Documentation of some of the short-comings appears in "MySQL Federated Tables: The Missing Manual".

* Sun Microsystems acquired MySQL AB in 2008.
* Version 5.1: production release 27 November 2008 (event scheduler, partitioning, plugin API, row-based replication, server log tables)

Version 5.1 contained 20 known crashing and wrong result bugs in addition to the 35 present in version 5.0 *(almost all fixed as of release 5.1.51)*.

MySQL 5.1 and 6.0-alpha showed poor performance when used for data warehousing – partly due to its inability to utilize multiple CPU cores for processing a single query.

* Oracle acquired Sun Microsystems on 27 January 2010.
* The day Oracle announced the purchase of Sun, Michael "Monty" Widenius forked MySQL, launching MariaDB, and took a swath of MySQL developers with him.
* MySQL Server 5.5 was generally available (as of December 2010). Enhancements and features include:
  + The default storage engine is InnoDB, which supports transactions and referential integrity constraints.
  + Improved InnoDB I/O subsystem
  + Improved SMP support
  + Semisynchronous replication.
  + SIGNAL and RESIGNAL statement in compliance with the SQL standard.
  + Support for supplementary Unicode character sets utf16, utf32, and utf8mb4.
  + New options for user-defined partitioning.
* MySQL Server 6.0.11-alpha was announced on 22 May 2009 as the last release of the 6.0 line. Future MySQL Server development uses a New Release Model. Features developed for 6.0 are being incorporated into future releases.
* The general availability of MySQL 5.6 was announced in February 2013. New features included performance improvements to the [query optimizer](https://en.wikipedia.org/wiki/Query_optimizer), higher transactional throughput in InnoDB, new [NoSQL](https://en.wikipedia.org/wiki/NoSQL)-style memcached APIs, improvements to partitioning for querying and managing very large tables, TIMESTAMP column type that correctly stores milliseconds, improvements to replication, and better performance monitoring by expanding the data available through the PERFORMANCE\_SCHEMA. The InnoDB storage engine also included support for full-text search and improved group commit performance.
* The general availability of MySQL 5.7 was announced in October 2015.
* MySQL Server 8.0.0-dmr (Milestone Release) was announced 12 September 2016.

**CHAPTER 2 : OBJECTIVES**

**2.1 PURPOSE OF THE PROJECT :**

The objective of Employee Management System is to allow the administrator of any organisation to edit and find out the personal details of a employee and allows him to keep up-to-date data of the employee easily. It will also facilitate keeping all the records such as their Id, Name, contact no., age ,designation, operating city etc. So all the information about an employee will be available at tips in just few seconds. Overall it make employee data management an easier job for the administrator of any organisation.

The main purpose of this SRS document is to illustrate the requirements of the project Employee Management System and is intended to help any organisation to maintain and manage it’s employee’s personal and official corporate data.

The project is made with the intend of having a simplified system that is as easy as possible to use. It needed to be compact and efficient. Anybody can learn to operate it. It should have future possibilities to expand and have commercial format.

**2.2 SCOPE OF THE PROJECT :**

Without a Employee Management System (of any kind), managing and maintaining the details of the employee will be a tedious and very hefty job for any organisation. If the organisation is a MNC then it will be impossible to store and maintain at up to date and accessible at real time record of its employee in different departments in different places.

A system is need to boost the record keeping department of any company. Employee Management System will store all the details of the employees including their personal details and official corporate details in one place where it will be real-time accessible and maintained.

Login module in Employee Management System will help in authentication of user accounts. Users who have valid login Id and password can only login into their respective accounts.

This system is equipped with all the neccesary functions which are need by default for managing the data in the database. More tables can be added later according to the need of the administrator.

**CHAPTER 3: IMPLEMENTATION**

**3.1 SYSTEM REQUIREMENTS :**

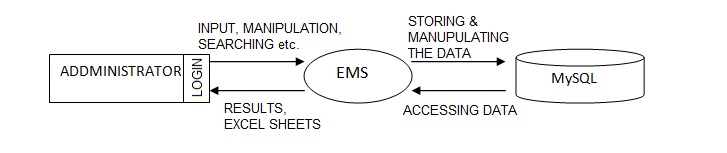
**3.1.1 HARDWARE REQUIREMENTS (minimum):**

* Processor : Intel Pentium 4th genration or AMD K10
* Ram : 1 Gigabytes of DDR3
* Hard disk space : 50 Gigabytes (SATA arch)
* Monitor : LCD (preffered)

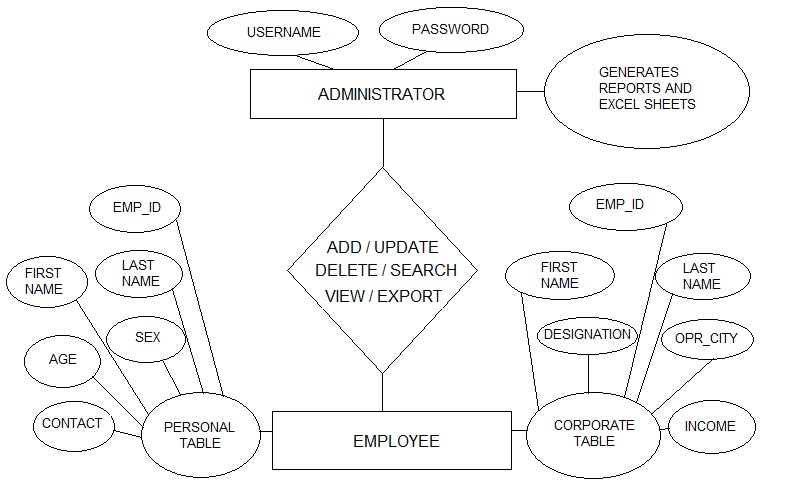
**3.1.1 SOFTWARE REQUIREMENTS (minimum):**

* Operating System : Window 7 or Linux kernel v10 or MacOS v9
* Language installed : Python version 3.6
* Libraries needed : PyMySQL v1.6, OpenPyXl v2.5, Tkinter
* Front End : Python IDLE 3.6
* Back End : MySQL database
* Excel Software : MS Office 2007 (preffered) or Open Office
* Other Softwares : XAMPP v3

**3.2 DATA FLOW DIAGRAMS (DFD) :**



**3.3 ER DIAGRAM**



**CHAPTER 5 : CONCLUSION AND FUTURE SCOPE**

**CONCLUSION**

Simplicity is never simple. As we have seen in this project, the process of creating a user- friendly and straightforward platform that facilitates the administrator’s job is one filled with complexity. From understanding user requirements to system design and finally system prototype and finalization, every step requires in-depth understanding and commitment towards achieving the objectives of the project.

Although the employee database management module is not fully integrated to the system and used on real time, the system prototype demonstrates easy navigation and data are stored in a systematic way. Overall, efficiency has improved and work processes simplified. Although all the objectives have been met, the system still has room for improvement. The system is robust and flexible enough for future upgrade using advanced technology and devices.

**FUTURE SCOPE**

Without a Employee Management System (of any kind), managing and maintaining the details of the employee will be a tedious and very hefty job for any organisation. If the organisation is a MNC then it will be impossible to store and maintain at up to date and accessible at real time record of its employee in different departments in different places.

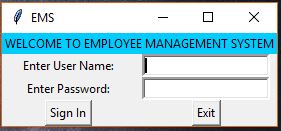
A system is need to boost the record keeping department of any company. Employee Management System will store all the details of the employees including their personal details and official corporate details in one place where it will be real-time accessible and maintained.

Login module in Employee Management System will help in authentication of user accounts. Users who have valid login Id and password can only login into their respective accounts.

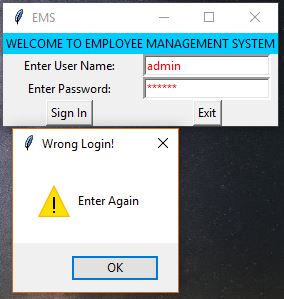
This system is equipped with all the neccesary functions which are need by default for managing the data in the database. More tables can be added later according to the need of the administrator.

This project can be evolved into a very much detailed version of itself and lots more features can be added thorugh modules. It can be given a full comercial format and can be sold to organisations.

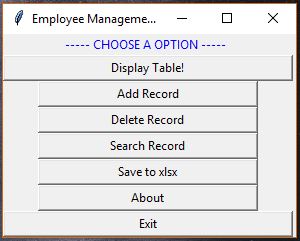
**CHAPTER 5 : SNAPSHOTS**



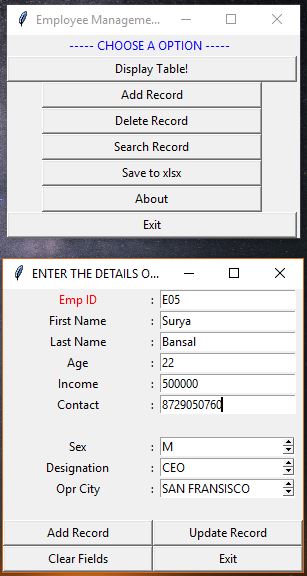
LOGIN PAGE



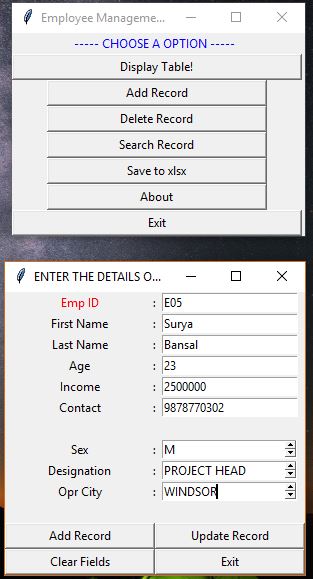
INVALID LOGIN ATTEMPT ALERT



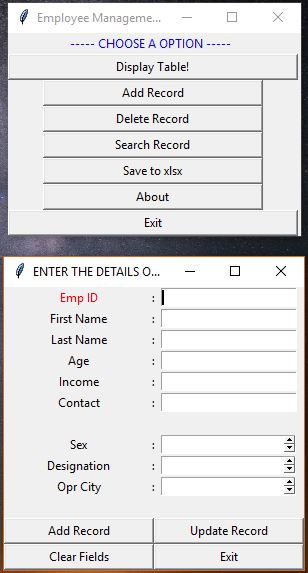
FRONT END OF PORTAL



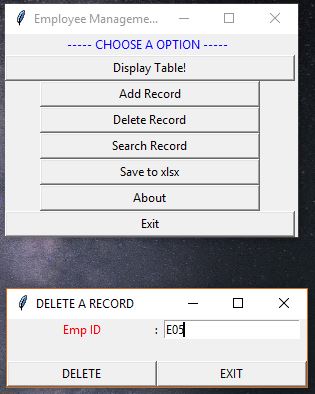
ENTERING A RECORD



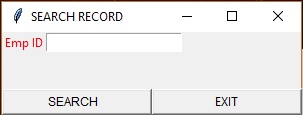
UPDATING A RECORDING



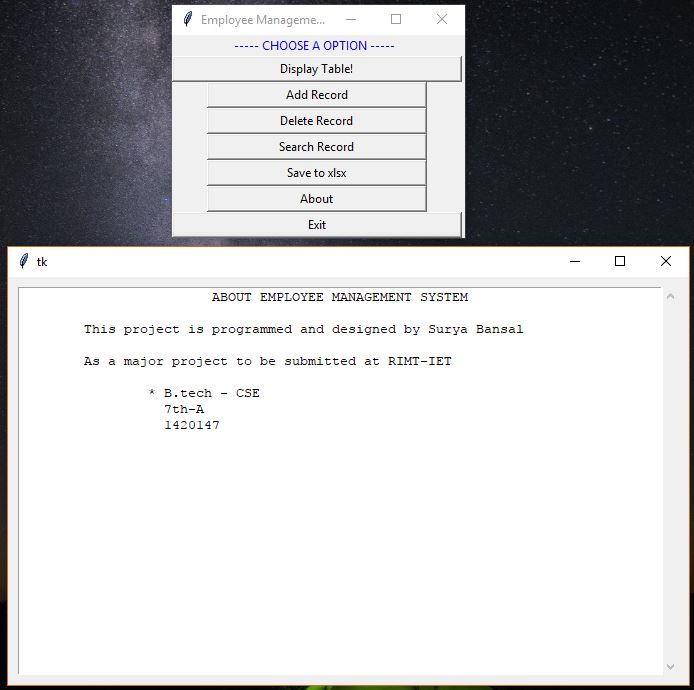
CLEARING THE FIELDS



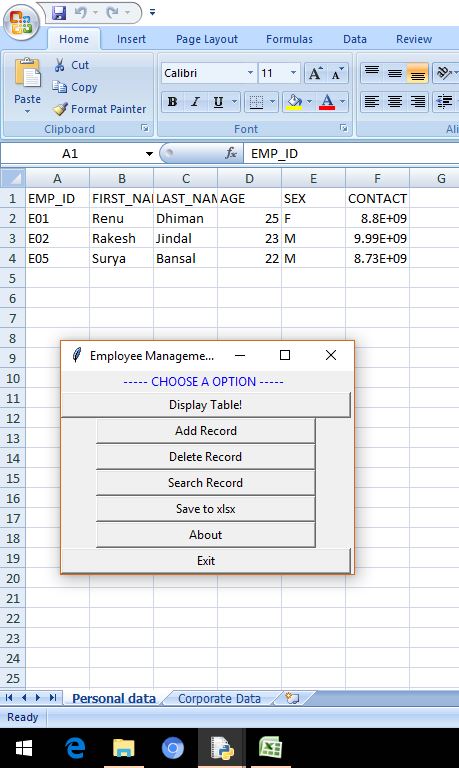
DELETING A RECORD



SEARCHING A RECORD

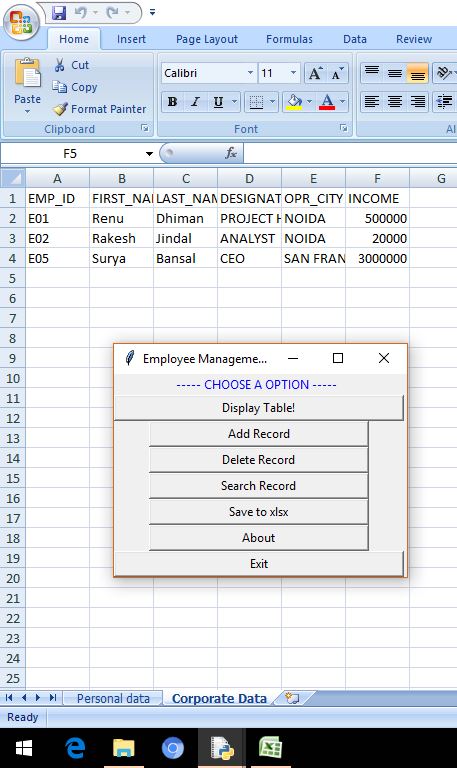


ABOUT THE PROGRAM

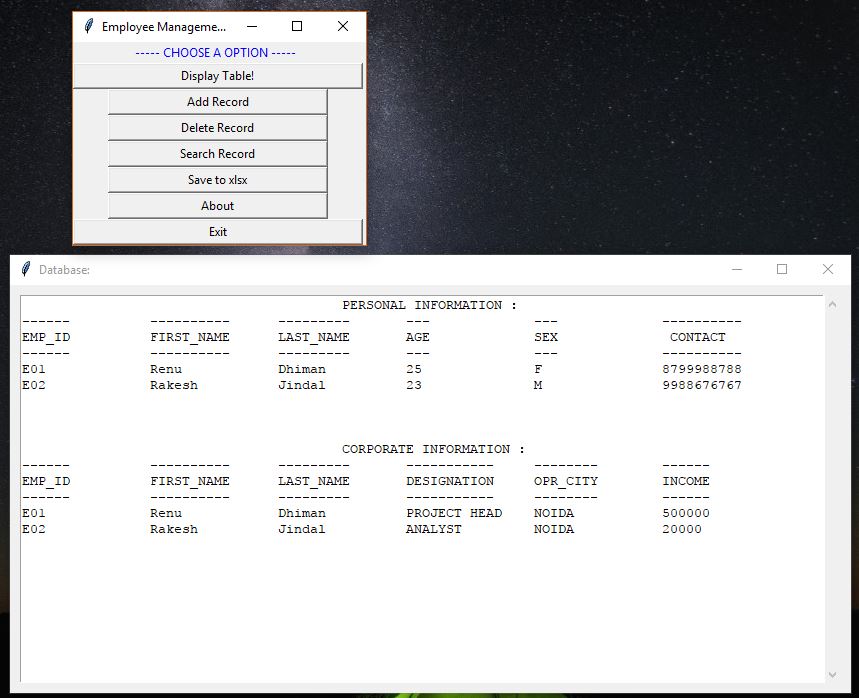


EXPORTING DATABASE TABLES TO EXCEL

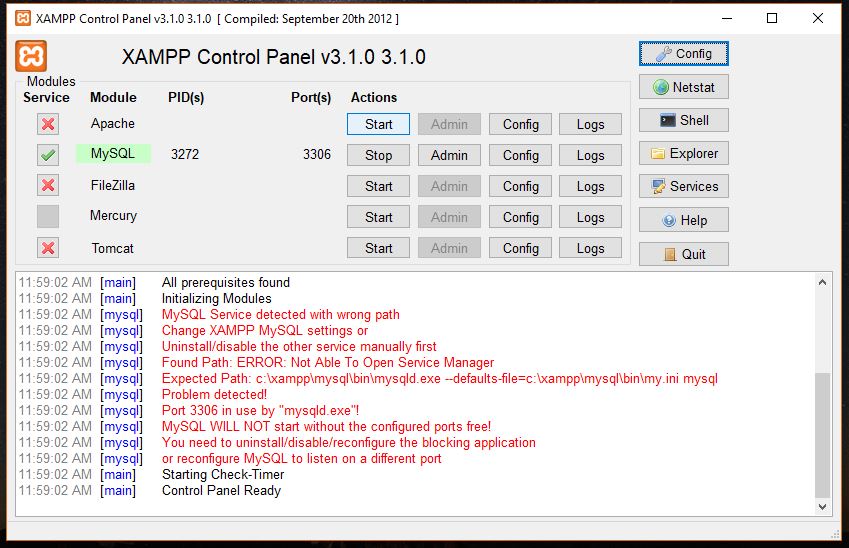
PERSONAL DATA TABLE



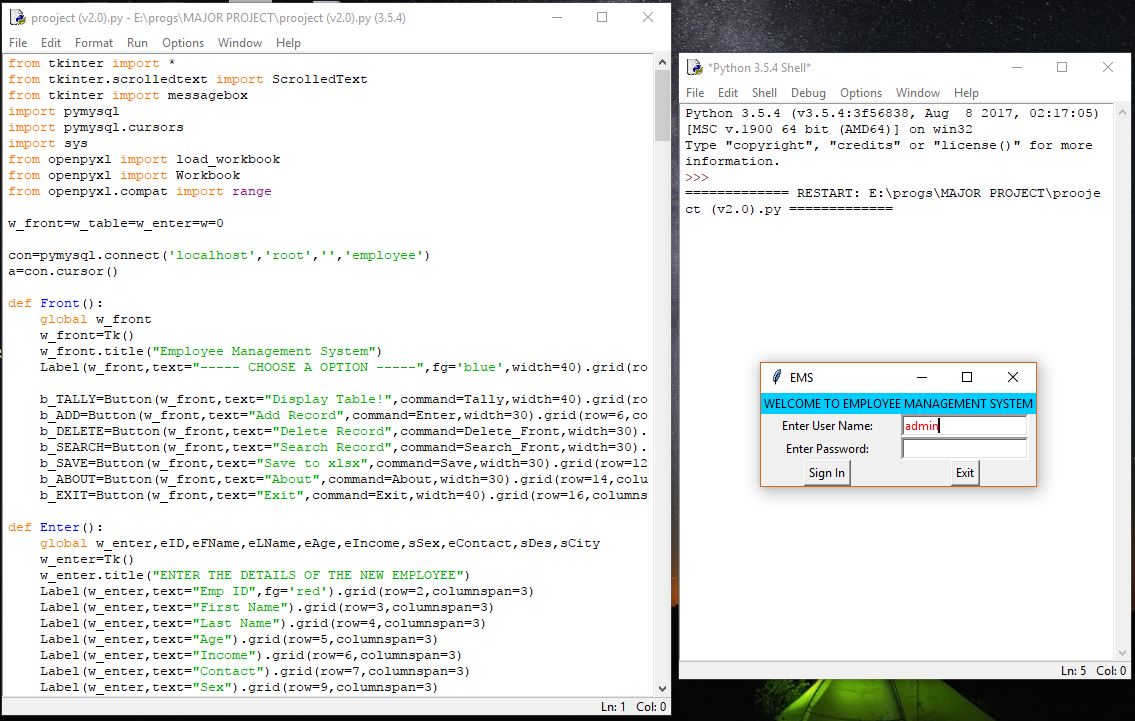
CORPORATE DATA TABLE



ALL RECORD REPORT IN DATABASE



XAMPP CONTROL PANEL



IDLE – An IDE for python