Hotel Management System

A DESIDERATUM FOR EVERY HOTEL.

1SG15CS004
ABHISHEK VARMA | 5TH SEMESTER 'A'
1SG15CS026
DYLAN IAN SALDANHA | 5TH SEMESTER 'A'

Hotel Management System

ACKNOWLEDGEMENT

Any achievement does not depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this mini project work. We would like to take this opportunity to thank them all.

We would first relish thanking **Infosys** for conducting this Python Workshop and giving us the opportunity of learning how to program with Python from experienced individuals and additionally providing fine resources to study from.

We would like to express our profound thanks to **Sri. G Dayanand,** Chairman, Sapthagiri College of Engineering Bangalore, for his continuous support in providing amenities to carry out this Mini Project.

Special Thanks to **Dr. N. Srinivasan,** Director, Sapthagiri College of Engineering Bangalore, for his valuable suggestion.

Also we would like to express our immense gratitude to **Dr. Aswatha Kumar M**, Principal, Sapthagiri College of Engineering Bangalore, for his help and inspiration during the tenure of the course.

We also extend our sincere thanks to **Dr. Yogish H K**, Professor and Head, Department of Computer Science and Engineering, Sapthagiri College of Engineering, for his constant support.

We would like to express our heartful gratitude to **Mr. Suriya Prakash,** Assistant professor and **Ms**. **Shilpa M,** Assistant professor, Department of Computer Science and Engineering, Sapthagiri College of Engineering, for their timely advice on the mini project and regular assistance throughout the work.

We also extend our sincere thanks to all the **Faculty members** and **supporting staff** Department of Computer Science and Engineering, Sapthagiri College of Engineering, for their constant support and encouragement.

We would withal relish thanking all the developers at StackOverflow for their unpaid strenuous exertion, without whom this project wouldn't be what it is today. Finally, we thank our parents and friends for their moral support.

ABHISHEK VARMA
DYLAN SALDANHA

ABSTRACT

Hotel Management System deals with the maintenance of a guest's bill during one's stay at the hotel and withal the allocation of rooms for them. This software will be used mainly by the receptionist who will be the first staff member a guest conventionally sees on ingress and additionally the last one afore one leaves. The receptionist can utilize this software to allocate rooms to the newly arrived guests based on their budget requisites. The room number will be then engendered and given to the guest along with a unique customer ID. Any restaurant, laundry, recreational activity bills will be accounted by the receptionist of particular facility. The entire bill can then be paid by the guest at the time one wishes to depart from the hotel.

Table of Contents

Chapter 1:

Introduction

1.1 Introduction to DBMS

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as modeling the details of all students and faculty members in a university.

A database management system (DBMS) is a computer software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, EnterpriseDB, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MemSQL, SQLite and IBM DB2.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Computer scientists may classify database-management systems according to the database models that they support; the most popular database systems since the 1980s have all supported the relational model - generally associated with the SQL language. Sometimes a DBMS is loosely referred to as a "database".

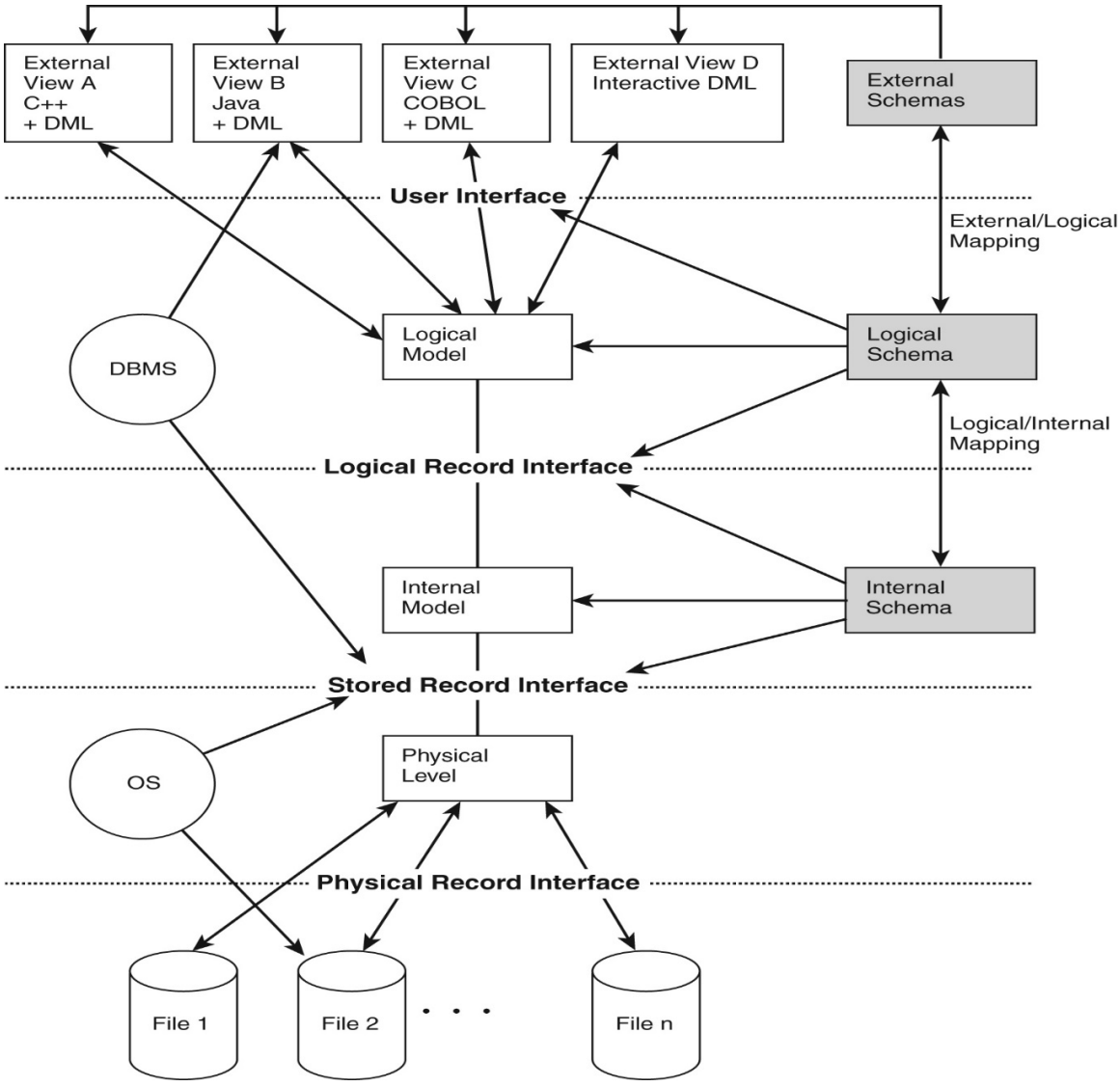
1.1.1 Three-tier Architecture

The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical. The architecture of a DBMS can be optically discerned as either single tier or multi-tier. An n-tier architecture divides the whole system into cognate but independent n modules, which can be independently modified, altered, transmuted, or superseded.

A 3-tier architecture disunites its tiers from each other predicated on the intricacy of the users and how they utilize the data present in the database. It is the most widely used architecture to design a DBMS.

- Physical Data Level (Tier)− The physical schema describes details of how data is stored: files, indices, etc. on the random access disk system. It also typically describes the record layout of files and type of files (hash, b-tree, flat). Early applications worked at this level - explicitly dealt with details. For example,

minimizing physical distances between related data and organizing the data structures within the file (blocked records, linked lists of blocks, etc.)



DBMS Architecture

- Conceptual Data Level (Tier) − Also referred to as the Logical level. It obnubilates details of the physical level. In the relational model, the conceptual schema presents data as a set of tables. The DBMS maps data access between the conceptual to physical schemas automatically.

External Data Level (Tier) − In the relational model, the external schema also presents data as a set of relations. An external schema specifies a view of the data in terms of the conceptual level. It is tailored to the needs of a particular category of users. Portions of stored data should not be seen by some users: and begins to implement a level of security and simplifies the view for these users. For example, students should not be

able to visually perceive faculty salaries, faculty should not be able to optically discern billing or payment data. Information that can be derived from stored data might be viewed as if it were stored. For example, GPA isn't stored, calculated when needed. Applications are written in terms of an external schema. The external view is computed when accessed. It is not stored. Different external schemas can be provided to different categories of users. Translation from external level to conceptual level is done automatically by DBMS at runtime.

1.2 Overview of the project

1.2.1 Problem statement

To develop a plenarily featured hotel management system that solves most of the prevalent hotel management quandaries faced by any hotel, utilizing state of the art technologies and making it available to all hotels at a low cost.

1.2.2 Objectives of project

- Ascertain that the software can run on any given platform.
- Facilely maintain the details of all the guests who have stayed at the hotel.
- Give facile and free access to guests to view their bill status.
- Develop a facile and error-free method of entering details of the guests for the staff members into the software.
- Allocate and mange rooms of each of the guests.
- Generate a bill for the use of each facility by the guest.
- Ascertain to have a user-amicable interface so that users are drawn into utilizing the software again.
- Avail in the transitioning from other 3rd party software.
- Manage the bills of all the guests felicitously without any mismatches.

Chapter 2:

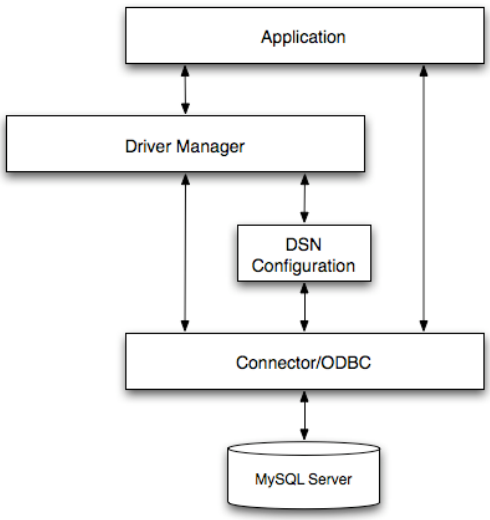
System Design and Methodology

2.1 System architecture

The user interface or the front end is developed utilizing Python. The main criterion for culling Python was its platform independence. Sure other things like its security, robustness, and simplicity follow. But it's platform independence takes first place. This is of utmost consequentiality because our first objective states that we require to ascertain that the software can run on any given platform. It would be tedious to develop the same software in multiple languages to fortify multiple platforms.

The database or backend is hosted utilizing a server called MAMP Server which provides the user to have a standalone database on his/her machine. The server further uses phpMyAdmin which is a free and open source administration implement for MySQL and MariaDB.

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.



System architecture

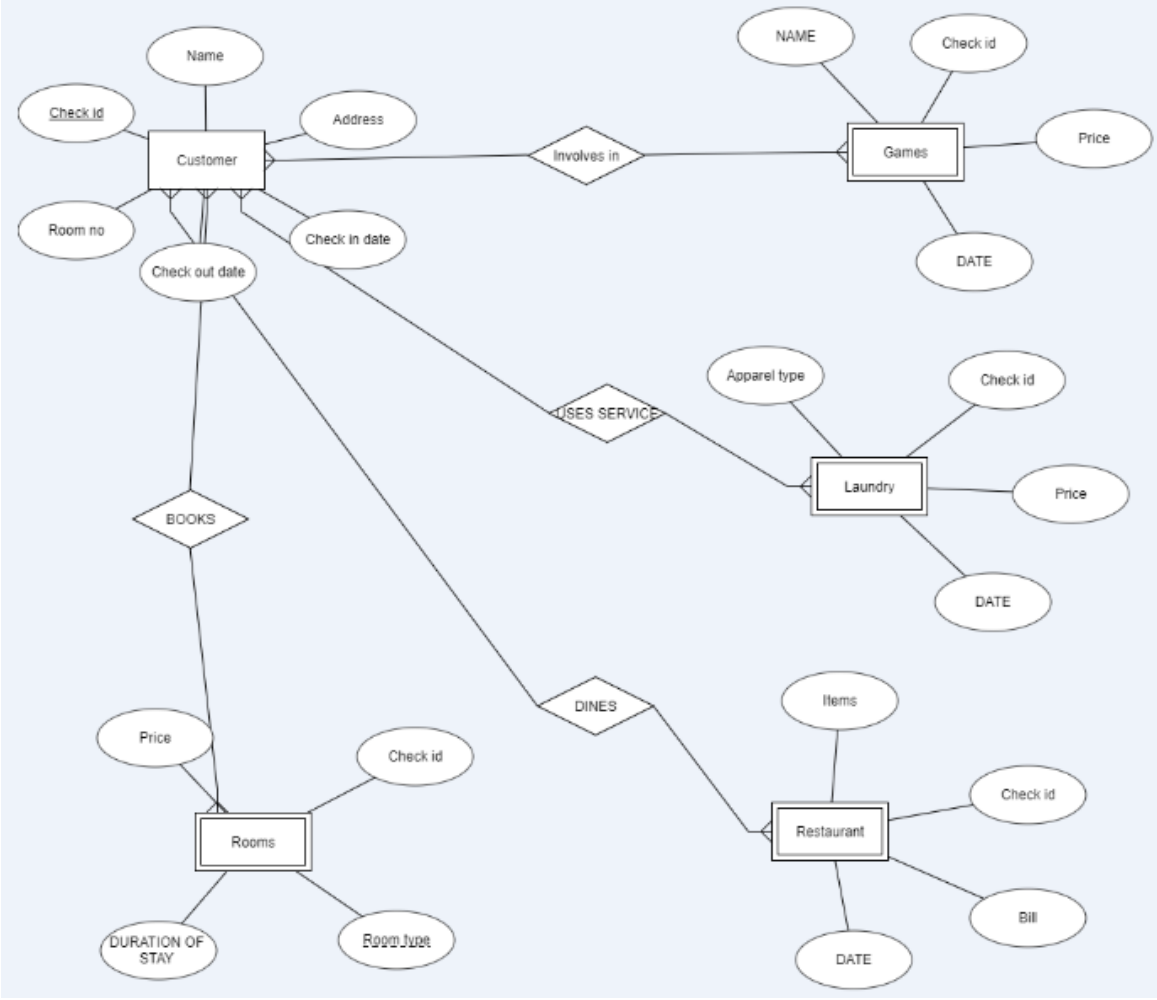
Here the database contains guest details currently residing in the hotel. This is accessed by the python code written on the frontend. We have used Spyder IDE to implement the python code.

The MySQL Connectors and APIs are the drivers and libraries that one uses to connect applications in different programming languages to MySQL database servers. The application and database server can be on the same machine or communicate

across the network. The Commons DbUtils library, on the other hand, is a small set of classes designed to make working with JDBC easier. JDBC resource cleanup code is mundane, error prone work so these classes abstract out all of the cleanup tasks from one's code leaving one with what one really wanted to do with JDBC in the first place: query and update data. A connection is established with the database, using the connect method of the mysql.connector Class (with an appropriate URL , username and password as the parameters).

To execute the MySQL queries on the database from the front end we have used a Connection Object. This class can be found in the mysql.connector package. Thereafter calling the execute method with the MySQL statement as a parameter will execute all DQL(Data Query Language) and DML(Data Manipulation Language) statements. The execute method will usually return a set of rows as its output which can be stored in a Cursor object.

2.2 Entity-relationship diagram (ER diagram)



ER Diagram

Chapter 3:

System Implementation

3.1 Modules

Main Menu Page

Objectives

- To allow the user to select from the given options with respect to the facility
- To check authenticity of the guest.

Input

- Menu number in the printed list.

Output

- The respective homepage of the facility will be displayed

Description

The first input page that is seen by the user will be the Main menu page. The user must select their facility from the printed list. The number of the option must be input and the user will be redirected to the page of their facility.

Check in and Room Allotments

Objectives

- To sanction the guest to check in into the hotel.
- To allot a room to the guest as per their requirements.
- To allot the paying guest with a unique customer ID.
- To sanction the guest to choose from a large variety of rooms that are available in the hotels.

Input

- Personal information about the guest.
- Type of room.

Output

- Customer ID of the guest.

- The allotted room number for the guest.

Description

When a new guest arrives to the hotel, the first thing one will do is check in to a room. This functionality is provided by the Check in page. The guest will first provide all the required details that asked by the receptionist and then will be provided with a unique customer ID and a room number based on the chosen room from the available set of rooms in the hotel.

Restaurant Page

Objectives

- To sanction the guest to order anything to his/ her likings from the menu.
- To store the bill amount of the particular guest along with one's total bill.

Input

- Customer ID of the particular guest.
- Items from the well-designed menu.
- The quantity of each item required.

Output

- An order for the requested food will be generated.
- The total restaurant bill will be generated.

Description

The guest will be first asked for his/her customer ID that was given during check in. The guest is then provided with the menu of items that are available in the restaurant on that particular day. The user can then key in the required set of items and their quantity as desired by the guest. The total restaurant bill for the guest is then generated and added to the total bill of the respective guest.

Laundry Page

Objectives

- To sanction the guest to order any type of laundry service to his/ her likings from the menu.
- To store the bill amount of the particular guest along with one's total bill.

Input

- Customer ID of the particular guest.
- Items from the well-designed menu.
- The quantity of each item required to given for laundry.

Output

- An order for the requested laundry will be generated.
- The total laundry bill will be generated.

Description

The guest will be first asked for his/her customer ID that was given during check in. The guest is then provided with the menu of items that are taken for laundry on that particular day. The user can then key in the required set of items and their quantity as desired by the guest. The total laundry bill for the guest is then generated and added to the total bill of the respective guest.

Recreational Activities Page

Objectives

- To sanction the guest to order any type of recreational service to his/ her likings from the menu.
- To store the bill amount of the particular guest along with one's total bill.

Input

- Customer ID of the particular guest.
- Items from the well-designed menu.
- The number of hours the activity lasted.

Output

- An order for the recreational activity will be generated.
- The total recreational activity bill will be generated.

Description

The guest will be first asked for his/her customer ID that was given during check in. The guest is then provided with the menu of items that are available for his/her satisfaction on that particular day. The user can then key in the required set of items and the number of hours as desired by the guest. The total recreational activity bill for the guest is then generated and added to the total bill of the respective guest.

Total Bill Page

Objectives

- To sanction the guest to view the total bill during departure and pay the required amount.
- To sanction the hotel staff to get the amount they deserve for their services.

Input

- Customer ID

Output

- A well-structured and descriptive bill consisting of all the service cost that the guest requested during his/her stay at the hotel.

Description

The guest will be first asked for his/her customer ID that was given during check in. The guest is then shown the sum of all the expenses under his/her name during their stay at the hotel.

Chapter 4:

Results and Screenshots

```
*****WELCOME TO HOTEL CALIFORNIA*****
1.Enter Customer Data
2.Calculate restaurant bill
3.Calculate laundry bill
4.Calculate Game bill
5.Show total cost
6.EXIT
Enter your choice:
```

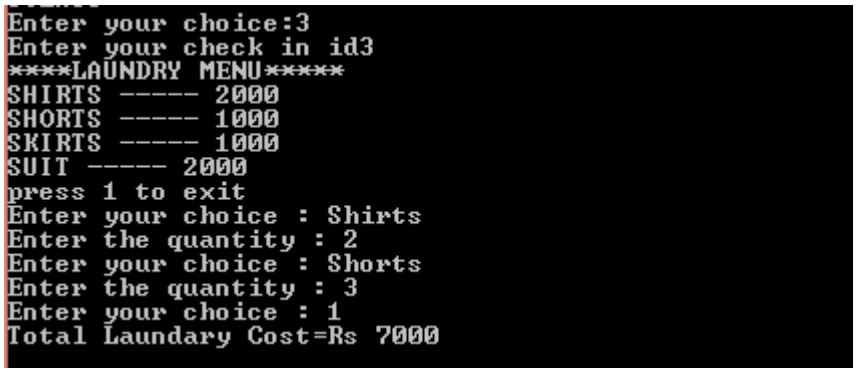
Main Menu Page

```
Enter your choice:1
Enter your name:Mark
Enter your address:Compton
Enter your check in date:12-06-2017
Enter your checkout date:15-06-2017
We have the following rooms for you:-
A   ----   2000

B   ----   3000

Enter Your Choice Please->A
Enter the number of days you will be spending with us
Your room no.: 3

check id is: 3

Thank you for registering!
```

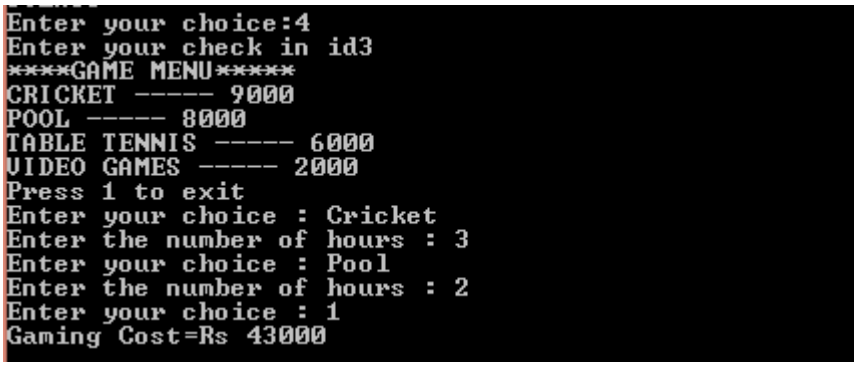
Check in and Room Allotments

```
Enter your choice:2
Enter your check in id3
*****RESTAURANT MENU*****
carrots ----- 2000
press 1 to exit
radish ----- 1000
press 1 to exit
Enter your choice:carrots
Enter the quantity : 3
Enter your choice:radish
Enter the quantity : 2
Enter your choice:1
Total food Cost=Rs 8000
```

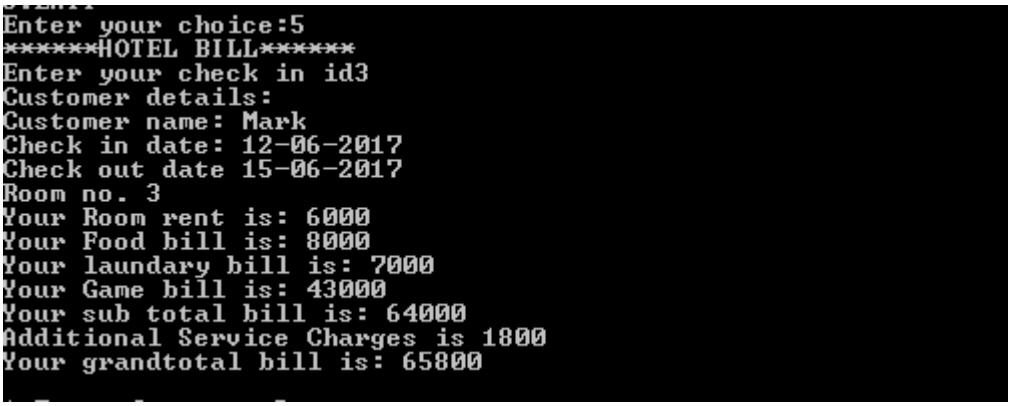
Restaurant Page



Laundry Page



Game Menu



Total Bill and Check out

Conclusion and Future works

To conclude the hotel management system is a desideratum for every hotel. All the objectives that were initially set at the commencement of the project have been met. All bugs that we have descried till date have been rectified. This hotel management system will not only avail hotels facilely manage their data but withal give guests valuable information that they can utilize for their own benefits. We developed this software without using any fancy front end decoration as we firmly wanted to build a software that any hotel, with any budget can easily afford. Any hotel need not spend their resources on building a brand new management system from scratch. Instead one can facilely get this software from GitHub as it is open source.

We hope to build a fancy front end graphical user interface without increasing the cost of the software even by a penny. One may assume that the above statement maybe in response to other individuals who just follow the work of other creators blindly on websites like YouTube. But we assure you that isn't true and that we only care about the customers we reach out to.

Bibliography

[1] MySQL Documentation - https://dev.mysql.com/doc/

[2] StackOverflow - https://stackoverflow.com/

[3] MAMP Documentation - https://www.mamp.info/en/documentation/

[4] Python Software Foundation - https://www.python.org/

[5] GitHub, Inc. - https://github.com/