Hospital Management System (HMS) Documentation

Table of Contents

📄 General Information

Back to Top

ℹ About this Manual

This manual is meant to be a documentation of the Hospital Management System project. The sections in this document aim to serve as a reference to the system and does not aim to teach you the basics of either MySQL or Python.

This project was made by Richmond on 12 to 13 March 2022.

This document was written on 13 March 2022.

📢 Note to User

To run this project the following requirements must be met:

- MySQL Server should be installed preferably with MySQL Workbench
- MySQL connector for python has to be installed.

This project uses a database named hospital_db, so create a new database with that name having two tables, employee and patient. The tables may be left empty and populated as desired.

In the two files `employee.py` and `patient.py` all of the functions use the connector function, a total of 8 times between the two files. Remember to change the username and password to the selected user of the database. Optionally the host parameter of the `connect()` function can be omitted.

Furthermore, additional tables can be added to the database and connceted to the project by using the same template. It is recommended that further additions are interfaced through the `main.py` file for the sake of convenience and to try and avoid creating unnecessary files.

📋 Overview of HMS

Back to Top

❓ What is HMS?

Our Hospital Management System (HMS) is a command-line interface application built with Python that connects to a hospital database and provides various functionality to perform actions on the database.

This system although fully functioning, is built with modularity in mind and therefore has a modular approach in respect to the use of various files.

HMS uses a MySQL database and uses the MySQL python connector to connect to MySQL server.

✔ The Main Features of HMS

HMS provides a very simple command-line based interface to the user for performing various operations on the database.

The following are the various operations the user can perform.

- Edit tables in the database.
- Add entries.
- Update entries.
- View tables.
- Delete entries.

📜 Feature Details

Back to Top

💻 User Interface

In order to run the main interface, the `main.py` file needs to be run. Upon successfully running the `main.py` file, the user is presented with the following lines as output.

```
HOSPITAL MANAGEMENT SYSTEM
==========================
 1. Add a new patient
 2. View all patients
 3. Update patient details
 4. Delete patient
 5. Add a new employee
 6. View all employees
 7. Update employee details
 8. Delete employee
 9. To print this list again
 10. Exit

-->
```

The list represents all the operations HMS can perform on the database. The arrow `-->` represents an input prompt from user. In order to trigger a specific operation, the user can enter the number of the corresponding list element.

🕹️ Functionality

The `main.py` runs as a continuous loop that can be exited only when the user enters `10` in the console. Note that the program does not exit even when any other integer is entered other than the ones specified in the options menu. In such a case the program will generate the following error message along with the input prompt in the next line.

```
Invalid input!

-->
```

In case the user enters anything other than an integer, the following error message is shown and the user is shown the list of options again.

```
Please input integer only...
```

Upcoming Features

Back to Top

Features Planned

Features that are planned to be added in the future are as follows

- Full database control, edit delete modify tables.
- Full query support with error capture and display.
- Console commands support, support for word-based commands as opposed to simple integer-based inputs.

Note that in adding these features, the existing feature set will not be affected. Legacy features will remain as is, even if the commands are changed, funtionality will remain the same.