Documentation

The Stock/Warehouse Management program/software is written in Python for release version 3. It runs on python 3.x versions without any issue. It can be run cross-platforms on any system that supports python. This software/program can be used to store information regarding inventory stored in all commercial establishment including shops, warehouses and factories.

Additional features included are ability to create sales record and maintaining it. Inventory records are stored in mysql database that are safe and cannot be modified without information regarding database credentials. MySQL or compatible databases(incl. MariaDB) are required preinstalled to run this software. Additionally, database connection info and credentials are dynamically stored into default folder *`'c:\Users\public\database\databaseinfo'`* which can be changed to suit it's purpose and maintain security of database. For establishing a connection to MySQL and compatible databases, this software uses python external library named `'mysql-connector'` that can be easily installed using command `'pip install mysql-connector'`. Try and except statement are used whenever necessary to provide seamless experience to user and prevent sudden halt due to any exception. The exceptions are also reported in format as found necessary.

This software uses four libraries i.e. os library, sys library, path library and mysql-connector external library imported via import statement followed by library name. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc. The os module is needed to interact with the underlying operating system. The python sys module provides functions and variables which are used to manipulate different parts of the Python Runtime Environment. It lets us access system-specific parameters and functions. Pathlib module in Python provides various classes representing file system paths with semantics appropriate for different operating systems. All output to display are made using `'print(text to display)'`.

It opens with Greeter 'Welcome to Stock Management Developed by Jaseema Hasan Class XII A Holy Angel\'s Convent School Muzaffarnagar'. Default folder `'c:\Users\public\database'` is created as object of class Path using `'Path(path to folder)'`. In Main Thread it checks whether default folder exists using 'is_dir()' on path object, default folder is created if it does not exists using 'os.makedirs(path object)'. If default database credentails storage file `'c:\Users\public\database\databaseinfo'` exists checked using `'is_file()'` called on path object, credentails are loaded into memory using file reader(`‘readline()’` with `rstrip('\n')` to remove trailing or prefixed '\n' newline character that are written to file) in chronological order viz. Hostname, Database name, Username, Password. If database credentials are not stored as in case of initial execution then user is informed that 'Database information does not exists' and is prompted to enter respective credentials read from console/commandline/command prompt via `'input(text to display)'`. Credential storage file is created to write with 'w' write mode, using `'open(filename,'mode')'`, `'write(variable suffixed with \n)'` and closed with `'close()'` functions called on file object.

Connection to database is established with database credentils using local function'conn(database name,hostname,username,password)' creating global connection object to be used in all other functions as well. For executing SQL statements on MySQL database, first Cursor(MySQLCursor) object is created and then SQL statements are passed via `'execute(“SQL statement”)'` called on cursor object. The MySQLCursor of mysql-connector (and similar libraries) is used to execute statements to communicate with the MySQL database to execute SQL statements, fetch data from the result sets, call procedures. Databases tables 'stock_inventory' & 'stock_sales' are created if they do not exist. In case of any exception during runtime, User is infromed about exception and the programs exits only if initialization of database fails which may be either due to wrong database credentials or missing database software, etc. As a safe recommended practice cursor object are closed using `'close()'` called on cursor objects. `'fetchone()'` is used to fetch only one row of results to query from database whereas `'fetchall()'` is used to query all rows of query result.

User is provided with a recurring interactive menu, with functions to insert a new inventory record, change inventory record, create a sales record, view inventory or sales record, delete view or sales records entirely or only specific inventory record. The code also includes ability to increment of existing inventory useful in case of new purchase before end of shelf line. An input is taken and based on user preference functions are called to do specific tasks. User is also provided with option to exit the recurring main thread. In case of wrong input, proper response are printed on screen and menu recurs.

There are five functions-

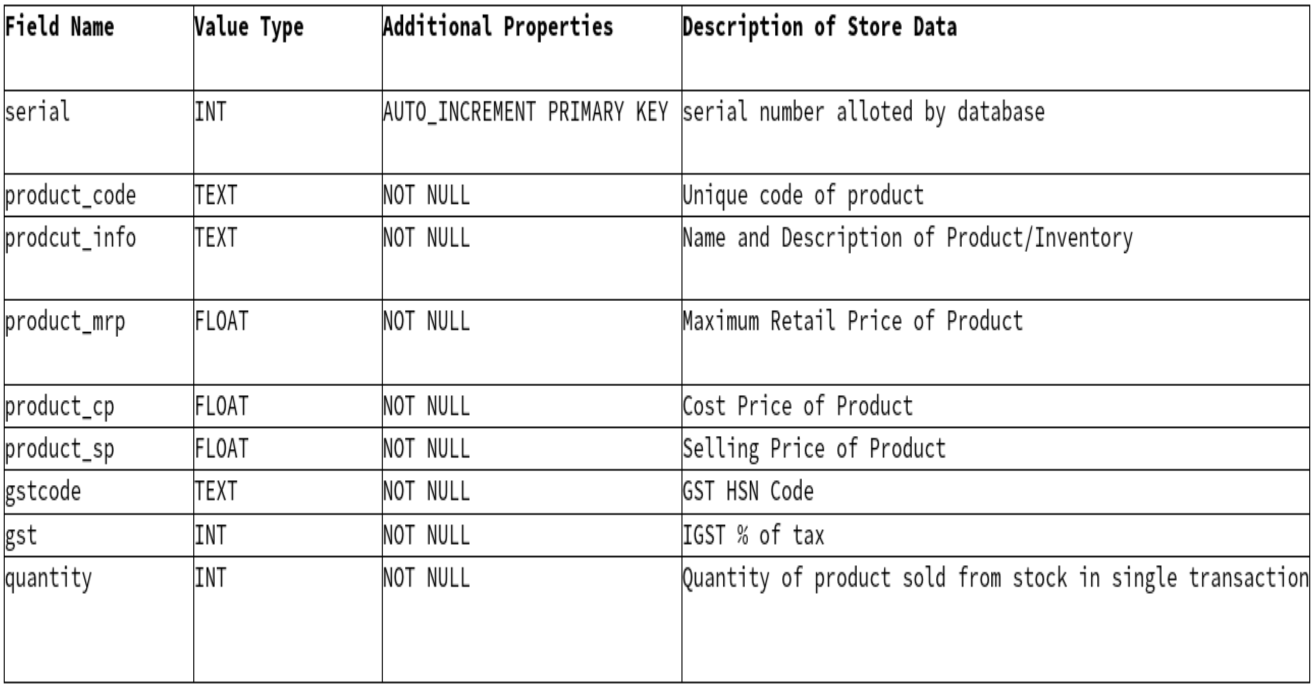
- **`conn(x,y,z,a)`** - to create a connection to database using credentials via 'mysql.connector.connect(host,user,password,database)'. It returns database connection object.
- **`entry(x)`** - it includes ability to create entries of new inventory, increment inventory or create sales record with variable argument to switch to specific task. 1 for adding new inventory, 2 for increment, 3 for creating sales record.
- **`edit()`** - it includes ability to edit a specific inventory record without changing it's product id(used for indexing products/inventory).
- **`view(x)`** - it includes ability to view inventory(1) and sales(2) data stored in database in tables 'stock_inventory' and 'stock_sales' respectively.
- **`delete()`** - it includes a submenu to delete entire inventory record (1), entire sales record (2) or a specific sales record (3).

Structure of Database Tables

stock_inventory

Field Name	Value Type	Additional Properties	Description of Store Data
serial	INT	AUTO_INCREMENT PRIMARY KEY	serial number alloted by database
product_code	TEXT	NOT NULL	Unique code of product
prodcut_info	TEXT	NOT NULL	Name and Description of Product/Inventory
product_mrp	FLOAT	NOT NULL	Maximum Retail Price of Product
product_cp	FLOAT	NOT NULL	Cost Price of Product
product_sp	FLOAT	NOT NULL	Selling Price of Product
gstcode	TEXT	NOT NULL	GST HSN Code
gst	INT	NOT NULL	IGST % of tax
quantity	INT	NOT NULL	Quantity of product currently in stock
sold	INT	DEFAULT ‘0’	Quantity of product sold till date

stock_sales



Field Name	Value Type	Additional Properties	Description of Store Data
serial	INT	AUTO_INCREMENT PRIMARY KEY	serial number alloted by database
product_code	TEXT	NOT NULL	Unique code of product
prodcut_info	TEXT	NOT NULL	Name and Description of Product/Inventory
product_mrp	FLOAT	NOT NULL	Maximum Retail Price of Product
product_cp	FLOAT	NOT NULL	Cost Price of Product
product_sp	FLOAT	NOT NULL	Selling Price of Product
gstcode	TEXT	NOT NULL	GST HSN Code
gst	INT	NOT NULL	IGST % of tax
quantity	INT	NOT NULL	Quantity of product sold from stock in single transaction

Variable Documentation

Variable Name	Scope and Description
database	Global variable to store database name for MySQL
username	Global variable to store username for MySQL
password	Global variable to store password for MySQL
hostname	Global variable to store hostname for MySQL
total	Global variable to store total of sales value
mydb	Database Connection Object
dbobj	Global Database Connection Object
pcode	Store product code
pinfo	Store product name and description
pmrp	Store product mrp
pcp	Store product cost price
psp	Store product selling price
gstc	Store GST HSN Code
gst	Store IGST %
qt	Store Quantity of product
cursor1	MySQL Connection Cursor
cursor2	MySQL Connection Cursor
cursor3	MySQL Connection Cursor
quantity	Store Quantity of product
result	Store result from MySQL database in array
n	Store number of rows of MySQL query result
x	Store formal function argument
d,y,z	Store user input
ctr	Counter variable for computation
my_dir	Store database credentials folder path
my_file	Store database credentials file path
file_object	Object of database credentials storage file