Web Application Revision Practice 2 [JPJC/2023/Prelim/P2/Task 4]

Name your Jupyter Notebook as:

`Task4_<your name>_<centre number>_<index number>.ipynb`

The upcoming Merlion Theme Park wishes to develop a web application to allow its visitors to buy park entry tickets online. The database will have two tables: a table to store data about the tickets and a table about the sales. The fields in each table are:

`Ticket`:

- `tDate`: a **unique** string date of the format YYYY-MM-DD assigned to the ticket.
- `dayOfWeek`: the day of week of the ticket's date e.g.: Monday, Tuesday, ...
- `unitPrice`: the unit price of a ticket in Singapore dollars.
  Ticket is priced at $40 for weekday, and $60 for weekend, public holiday,and school holiday.
- `totQuan`: the total quantity of tickets for the date.
- `availQuan`: the quantity of tickets still available for the date.

`Sale`:

- `saleID`: a unique autoincrement integer ID assigned to the sale.
- `tDate`: the ticket's unique string date.
- `quan`: the quantity of ticket in the sale.
- `totalPrice`: the total price of the sale.

For each of the sub-tasks 4.1 to 4.3, add a comment statement at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

In [1] :

```
#Task 4.1
Program code
```

```
Output:
```

Task 4.1

Write a Python program that uses SQL code to create the database `MerlionThemePark` with the two tables given. Define the primary and foreign keys for each table. [5]

Task 4.2

The text files `Task4_2_1.txt` and `Task4_2_2.txt` store the comma-separated values for each of the tables in the database.

Write a Python program to read in the data from each file and then store each item of data in the correct place in the database. [5]

Task 4.3

Write a Python program to input a month and output the ticket information of all the dates of that month, displayed in columns with header. [5]

Test your program by running the application with the month 11. [1]

Save your Jupyter Notebook for Task 4.

Task 4.4

Write a Python program and the necessary files to create a web application that:

- has a home page for visitors to
  - input month into a textbox, and
  - click on submit button,

- on the next page, the ticket page, visitors can
  - view the ticket information (in a table with headers: Date, Day of Week, Unit Price $, and Available Quantity) according to the month entered in the previous page,
  - input month (MM), day (DD) and quantity of tickets into three textboxes, and
  - click on submit button to buy,

- assume the visitors' input for month and day are valid

- if quantity is within the quantity available, visitors can view the **sale confirmation page** containing the date of ticket, quantity, total price, and a message, "Transaction is successful".

- if the quantity exceeds the quantity available, visitors will view a **notification page** with the message, "Insufficient quantity. Transaction unsuccessfully."

For a successful sale (valid input for month, day and quantity), the `Ticket` and `Sale` tables are to be updated as follows:

- `availQuan` in `Ticket` table to be decreased accordingly, and

- a new record is creared in the `Sale` table.

Save your Python program as
`Task4_4_<your name>_<centre number>_<index number>.py`

with any additional files/subfolders as needed in a folder named
`Task4_4_<your name>_<centre number>_<index number>.py` [14]

Run the web application using the following inputs to obtain a successful ticket sale,
month = 11, day = 02, quantity = 10

Save the **sales confirmation page** of a successful ticket sale as:
`Task4_4_<your name>_<centre number>_<index number>.html` [1]