INTRODUCTION

Hotel Management is a CLI-based application that uses Python, pandas, and matplotlib to manage hotel reservations. The application uses pandas to read and manipulate data from a CSV file. matplotlib is used to plot graphs of the data, which can be used to visualize information about hotel reservations.

The application is designed to be easy to use and can be used by anyone with basic knowledge of Python. It is a valuable tool for hotels of all sizes, as it can help to automate tasks, improve efficiency, and make better decisions.

OBJECTIVE

The objective of the Hotel Management project is to provide a simple and user-friendly interface to manage hotel reservations and guest information. It aims to streamline the process of handling guest records, performing operations on the data, and generating visual representations of the data using graphs and charts.

The application will be able to perform a variety of operations on hotel reservations, including:

- Adding new records to the dataFrame
- Deleting records from the dataFrame
- Searching for records based on various columns, such as name, check-in date, check-out date, and room type
- Viewing the latest reservation
- Plotting a pie chart of guests by country or a bar graph of guests by check-in status
- Loading the dataFrame back to the CSV file

LIMITATION

- The project is CLI-based. This means that it can only be used from the command line. Some users may prefer a graphical user interface (GUI).
- The project only works with CSV files. This means that the data must be in CSV format. Some hotels may use a different format for their reservation data.
- The project does not support all features of a commercial hotel management system. For example, the project does not support online booking or payments.

Despite these limitations, the hotel management project has the potential to be a valuable tool for hotels of all sizes. It can help to automate tasks, improve efficiency, and make better decisions.

SYSTEM CONFIGURATION

The system configuration required to run the hotel management project is as follows:

Hardware:

A computer with a processor that is at least 1.5 GHz
At least 4 GB of RAM
At least 10 GB of free disk space

Software:

Python 3.11 or higher
pandas
Matplotlib

The project can be run on any operating system that supports Python, such as Windows, macOS, and Linux.

To install the required software, you can use the following commands:

pip install python
pip install pandas
pip install matplotlib

THEORETICAL BACKGROUND

1. What is Python Programming Language?

Python is a high-level, interpreted, and general-purpose programming language. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and simplicity, making it a great language for beginners. It has a large and active community, making it easy to find resources, libraries, and support.

Python is versatile and can be used for a wide range of applications, such as web development, data analysis, artificial intelligence, automation, and more.

2. What is Pandas?

Pandas is an open-source Python library built on top of NumPy. It provides data structures and functions for efficiently manipulating and analyzing large datasets. Pandas is widely used in data analysis and data science tasks.

3. What is NumPy?

NumPy (Numerical Python) is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays efficiently. NumPy is the foundation of many other scientific Python libraries, including Pandas.

4. What is a Series and DataFrame?

- Series: In Pandas, a Series is a one-dimensional labeled array that can hold data of any type (e.g., integers, strings, floats). It is similar to a Python list or array, but it has additional functionalities. A Series consists of two parts: the data and the index, which can be used to access the data.

- **DataFrame**: A DataFrame is a two-dimensional labeled data structure in Pandas, resembling a table or spreadsheet. It consists of rows and columns, where each column can have different data types. You can think of a DataFrame as a collection of Series. DataFrames are incredibly powerful for data manipulation, cleaning, and analysis.

5. What is a CSV file?

CSV stands for "Comma-Separated Values." It is a plain text file format used to store tabular data. Each line of the file represents a row, and the values within each row are separated by commas (or other delimiters like semicolons or tabs). CSV files are widely used for data storage and data exchange between different software systems.

6. How to Load CSV into DataFrame using Pandas?

Pandas provides a simple method to load CSV data into a DataFrame using the `pd.read_csv()` function. Here's how you can do it:

```python
import pandas as pd

# Load CSV data into a DataFrame
df = pd.read_csv('data.csv')

# Now, you can work with the DataFrame 'df'
# For example, you can print the first few rows
print(df.head())
```

HOTEL MANAGEMENT CSV USED

resources > hotel_records.csv

```
Name,Contact_Number,Address,Check_In,Check_Out,Room_Type,Total_Price,Is_Checked_In,Guest_Country
Sameer,+91-9988776655,Block-C Janak Puri New Delhi Delhi India,25-12-2022,25-01-2023,Deluxe,45000.0,True,India
John,+91-9876543210,123 Main Street New York NY USA,10-05-2023,15-05-2023,Super Deluxe,10000.0,True,USA
Rahul,+91-9876543211,456 Park Avenue Mumbai Maharashtra India,01-07-2023,10-07-2023,Standard,7000.0,False,India
Emily,+91-9876543212,789 Oak Street London England UK,15-08-2023,25-08-2023,Super Deluxe,18000.0,True,UK
Sandeep,+91-9876543213,246 Pine Street Bengaluru Karnataka India,05-10-2023,15-10-2023,Suit,30000.0,True,India
Sophia,+91-9876543214,789 Maple Avenue San Francisco CA USA,20-12-2023,30-12-2023,Deluxe,21000.0,False,USA
Arjun,+91-9876543215,987 Elm Street Jaipur Rajasthan India,10-02-2024,20-02-2024,Standard,6000.0,True,India
Oliver,+91-9876543216,654 Cedar Avenue Paris France France,25-04-2024,05-05-2024,Deluxe,7500.0,False,France
Aarav,+91-9876543217,321 Walnut Street Kolkata West Bengal India,15-06-2024,25-06-2024,Super Deluxe,16000.0,True,
India
Isabella,+91-9876543218,876 Birch Street Rome Italy Italy,01-08-2024,10-08-2024,Suit,20000.0,True,Italy
Ravi,+91-9876543219,753 Pine Street Chennai Tamil Nadu India,12-09-2024,18-09-2024,Standard,6000.0,True,India
Emma,+91-9876543220,987 Oak Avenue Sydney New South Wales Australia,05-11-2024,10-11-2024,Deluxe,9000.0,True,
Australia
Aryan,+91-9876543221,246 Maple Street Brisbane Queensland Australia,25-12-2024,02-01-2025,Super Deluxe,14000.0,
True,Australia
Mia,+91-9876543222,753 Elm Avenue Barcelona Catalonia Spain,15-02-2025,25-02-2025,Suit,22500.0,True,Spain
Abhishek,+91-9876543223,987 Walnut Street Bangkok Bangkok Thailand,01-04-2025,10-04-2025,Deluxe,10500.0,False,
Thailand
Liam,+91-9876543224,246 Cedar Avenue Vancouver British Columbia Canada,10-06-2025,18-06-2025,Super Deluxe,18000.0,
True,Canada
Anaya,+91-9876543225,753 Birch Street Auckland Auckland New Zealand,25-08-2025,05-09-2025,Standard,5000.0,True,
New Zealand
Alexander,+91-9876543226,987 Pine Avenue Tokyo Kanto Japan,15-10-2025,20-10-2025,Suit,12500.0,True,Japan
Sanvi,+91-9876543227,246 Oak Street Mexico City Mexico City Mexico,01-12-2025,10-12-2025,Standard,7000.0,True,
Mexico
Leo,+91-9876543228,753 Maple Avenue Cairo Cairo Egypt,20-01-2026,30-01-2026,Deluxe,13500.0,False,Egypt

```

SOURCE CODE

```
from init import loadDataFrameFromCsv

from tasks import init_execute_task, taskList

ACTIVE = True

DF = None




def main():

    global DF

    global ACTIVE

    print("     #### BOOTING UP THE APPLICATION  ###         ")

    print("     #### Hotel Management Software  ###         ")

    print("     #### loading csv data into the system  ###      ")

    DF = loadDataFrameFromCsv()

    print("     #### data has been successfully loaded into the
system  ###      ")

    while ACTIVE:

        option = taskList()

        DF = init_execute_task(DF, option)
```

```
        print("     ###############      ")
        if (type(DF) == type(False)):
            ACTIVE = False
        else:
            print("SHOW TASK MENU ?")
            choice = input("Y/N: ")
            if (choice.lower() == 'y'):
                ACTIVE = True
            else:
                ACTIVE = False

    print("     #### SHUTING DOWN THE APPLICATION ###      ")
    print("     #### Have A Nice Day :D  ###      ")


if __name__ == '__main__':
    main()
import pandas as pd
CSV_PATH = "./resources/hotel_records.csv"

def loadDataFrameFromCsv():
```

```
    try:
        df = pd.read_csv(CSV_PATH)
    except:
        print("error loading the csv file path")
        print(" program terminated ")
    return df


def loadDataFrameToCsv(df):
    try:
        print("     #### Saving Records To The Csv ####     ")
        df.to_csv(CSV_PATH, index=False)
        print(" #### records saved back to csv ####     ")
    except:
        print("error loading the dataFrame to the csv file path")
        print(" program terminated ")
        return df
from insert_record import insert_record
from delete_record import delete_record
from get_records import *
from init import loadDataFrameToCsv
from plot import *


def init_execute_task(df, option):
    if (option == "1"):
        return insert_record(df)
    elif (option == "2"):
        column = 'Name'
        return delete_record(df, column)
    elif (option == "3"):
        return get_latest_reservation(df)
    elif (option == "4"):
        return get_reservation_by_name(df)
    elif (option == "5"):
        return get_reservation_by_check_in_date(df)
    elif (option == "6"):
```

```
            return get_reservation_by_check_out_date(df)
        elif (option == "7"):
            return get_all_reservation_checked_in_status(df, True)
        elif (option == "8"):
            return get_all_reservation_checked_in_status(df, False)
        elif (option == "9"):
            return plot_guests_by_country(df)
        elif (option == "10"):
            return plot_guest_by_check_in_status(df)
        elif (option == "11"):
            return loadDataFrameToCsv(df)
        elif (option == "12"):
            print("Exiting...")
            return False

        else:
            print("Invalid Option")
            return False


def taskList():
    TASK_DICT = {
        "1": "Making a new booking reservation",
        "2": "Deleting a booking reservation",
        "3": "See latest 5 reservations",
        "4": "Search a booking using name",
        "5": "Search a booking Using check-in date",
        "6": "Search a booking Using check-out date",
        "7": "See all guests who are checked in",
        "8": "See all guests who are checked out",
        "9": "Plot the pie chart of guest by the country",
        "10": "Plot the bar graph of guest by the checked in status",
        "11": "Save reservations back to csv",
        "12": "Exit the application"
    }
    for key in TASK_DICT:
        print(f"     {key} --- {TASK_DICT[key]}      ",)
```

```
    operation_selected = input("select the task : ")
    if (not operation_selected.isdigit() or
(operation_selected.lower() not in TASK_DICT.keys())):
        print(
            f"     #### INVALID TASK SELECTED  {operation_selected}
###      ")
        return "-1"
    else:
        return operation_selected

from constants import CHECK_IN_COLUMN, CHECK_OUT_COLUMN


def insert_record(df):
    print("     ### Add new reservations record ###     ")
    input_values = []
    column_list = list(df.columns)
    for column in column_list:
        column_name = str(column).replace("_", " ")
        if (column not in [CHECK_OUT_COLUMN, CHECK_IN_COLUMN]):
            value = input(f"please provide {column_name} :")
            input_values.append(value)
        else:
            value = input(
                f"please provide {column_name} in format dd-mm-yyyy
:")
            input_values.append(value)

    print(f"        ### new reservation to be inserted in records is
### \n", input_values)
    try:
        df.loc[df.index[-1]+1, :] = input_values  # statement to add
new record
    except:
        print("     ### error while inserting record ###       ")
        exit()
```

```
    print("     ### new reservation made successfully ###       ")
    print("     ### last two reservations are ###       ")
    print(df.tail(2))
    return df

def delete_record(df, column):
    all_names = list(df[column])
    print(f"        #### All Guests Name ####       \n")
    print(all_names)
    name = input(f"enter the {column} to delete its reservation: ")
    if (name in all_names):
        try:
            record_index = df[df[column] == name].index
            print(f"deleting the {str(record_index)} record with name
{name} ")
            # axis 0 to drop a row/record
            df.drop(record_index, axis=0, inplace=True)
            return df
        except:
            print(" error occurred while deleting record of index ",
                  str(record_index))
            exit()
    else:
        print(f" record with {name} not found")
        return True

from constants import CHECK_IN_COLUMN, CHECK_OUT_COLUMN, NAME_COLUMN,
IS_CHECKED_IN_COLUMN


def get_latest_reservation(df):
    print("     ### the last 5 reservations are ###         ")
    print(df.tail(5))
    return df
```

```
def get_reservation_by_name(df):
    print("     ### Search reservation record by name ###       ")
    all_names = list(df[NAME_COLUMN])
    print(f"        #### All Guests Name ####       \n")
    print(all_names)
    print("pls check the casing of the name")
    name = input(f"enter the Name to check reservation: ")
    if (name in all_names):
        try:
            record = df[df[NAME_COLUMN] == name]
            print(f"Record Found")
            print(record)
            return df
        except:
            print(" error occurred while fetching record with name",
name)
            exit()
    else:
        print(f" record with {name} not found")
        return df


def get_reservation_by_check_in_date(df):
    print("     ### Search reservations record by check in date ###
")
    current_check_in_dates = set(df[CHECK_IN_COLUMN])
    print("All distinct check in dates in the system")
    print(current_check_in_dates)
    query_date = input(f"enter the check-in date in format
dd-mm-yyyy: ")
    if (query_date in current_check_in_dates):
        print(
            f"     ### reservations by check in date {query_date}
found ###       ")
        records = df[df[CHECK_IN_COLUMN] == query_date]
        print(records)
    else:
```

```
        print(
            f"     ### reservation by check in date {query_date} not
found ###       ")

    return df


def get_reservation_by_check_out_date(df):
    print("     ### Search reservations record by check out date ###
")
    query_date = input(f"enter the check-out date in format
dd-mm-yyyy: ")
    current_check_out_dates = list(df[CHECK_OUT_COLUMN])
    if (query_date in current_check_out_dates):
        print(
            f"     ### reservations by check out date {query_date}
found ###       ")
        records = df[df[CHECK_OUT_COLUMN] == query_date]
        print(records)
    else:
        print(
            f"     ### reservation by check out date {query_date} not
found ###       ")

    return df


def get_all_reservation_checked_in_status(df, checked_in_status):
    if checked_in_status:
        print("     ### all checked in reservations ###     ")
    else:
        print("     ### all checked in reservations ###     ")

    records = df[df[IS_CHECKED_IN_COLUMN] == checked_in_status]
    print(records)
    return df
```

```
import pandas as pd
import matplotlib.pyplot as plt
from constants import GUEST_COUNTRY, IS_CHECKED_IN


def plot_guests_by_country(data_frame):
    print("     #### Plotting the pie chart of guest by the country
###     ")
    try:
        country_counts = data_frame[GUEST_COUNTRY].value_counts()
        plt.figure(figsize=(8, 6))
        plt.pie(country_counts, labels=country_counts.index,
autopct='%1.1f%%')
        plt.title('Guests Residing in the Hotel by Country')
        plt.show()
    except:
        print("     #### ERROR ! Plotting the pie chart of guest by
the country ###     ")
    return data_frame


def plot_guest_by_check_in_status(df):
    print("     #### Plotting the bar graph of guest by the checked
in status ###     ")
    try:
        total_guest = len(df)
        guests_checked_in = len(df[df["Is_Checked_In"] == True])
        guests_not_checked_in = total_guest-guests_checked_in
        x_ticks_labels = ["Total Guests","Checked In Guests",
"Checked Out Guests"]
        data = [total_guest, guests_checked_in,
guests_not_checked_in]
        colors = ['orange', 'green', 'purple']
        plt.bar(x_ticks_labels,data, data=data, color=colors)
        plt.xlabel("Guests Status")
        plt.ylabel("Guests Count By Status")
        plt.title("Guests Status")
```

```
        plt.show()
    except:
        print("     #### ERROR ! Plotting the bar graph of guest by
the checked in status ###     ")
    return df
```

OUTPUTS

```
      #### BOOTING UP THE APPLICATION  ###
      #### Hotel Management Software  ###
      #### loading csv data into the system  ###
      #### data has been successfully loaded into the system  ###
      1 --- Making a new booking reservation
      2 --- Deleting a booking reservation
      3 --- See latest 5 reservations
      4 --- Search a booking using name
      5 --- Search a booking Using check-in date
      6 --- Search a booking Using check-out date
      7 --- See all guests who are checked in
      8 --- See all guests who are checked out
      9 --- Plot the pie chart of guest by the country
      10 --- Plot the bar graph of guest by the checked in status
      11 --- Save reservations back to csv
      12 --- Exit the application
 select the task : 
```

```
select the task : 1
    ### Add new reservations record ###
please provide Name :Angana
please provide Contact Number :+918826554477
please provide Address :gk-1 35-B New Delhi India -110019
please provide Check In in format dd-mm-yyyy :25-07-2023
please provide Check Out in format dd-mm-yyyy :01-08-2023
please provide Room Type :Suit
please provide Total Price :5200
please provide Is Checked In :True
please provide Guest Country :India
       ### new reservation to be inserted in records is ###
['Angana', '+918826554477', 'gk-1 35-B New Delhi India -110019', '25-07-2023', '01-08-2023', 'Suit', '5200', 'True', 'India']
    ### new reservation made successfully ###
    ### last two reservations are ###
      Name  Contact_Number                          Address    Check_In   Check_Out Room_Type Total_Price Is_Checked_In Guest_Country
19     Leo  +91-9876543228  753 Maple Avenue Cairo Cairo Egypt  20-01-2026  30-01-2026    Deluxe     13500.0         False         Egypt
20  Angana   +918826554477   gk-1 35-B New Delhi India -110019  25-07-2023  01-08-2023      Suit        5200          True         India
    ###############
SHOW TASK MENU ?
Y/N: 
```

```
select the task : 2
       #### All Guests Name ####

['Sameer', 'John', 'Rahul', 'Emily', 'Sandeep', 'Sophia', 'Arjun', 'Oliver', 'Aarav', 'Isabella', 'Ravi', 'Emma', 'Aryan', 'Mia', 'Abhishek', 'Liam', 'Anaya', 'Alexander',
 'Sanvi', 'Leo', 'Angana']
enter the Name to delete its reservation: Sophia
deleting the Index([5], dtype='int64') record with name Sophia
     ###############
SHOW TASK MENU ?
Y/N:
```

```
select the task : 3
     ### the last 5 reservations are ###
         Name  Contact_Number                                        Address    Check_In   Check_Out Room_Type Total_Price Is_Checked_In Guest_Country
16      Anaya  +91-9876543225  753 Birch Street Auckland Auckland New Zealand  25-08-2025  05-09-2025  Standard      5000.0          True   New Zealand
17  Alexander  +91-9876543226              987 Pine Avenue Tokyo Kanto Japan  15-10-2025  20-10-2025      Suit     12500.0          True         Japan
18      Sanvi  +91-9876543227   246 Oak Street Mexico City Mexico City Mexico  01-12-2025  10-12-2025  Standard      7000.0          True        Mexico
19        Leo  +91-9876543228              753 Maple Avenue Cairo Cairo Egypt  20-01-2026  30-01-2026    Deluxe     13500.0         False         Egypt
20     Angana   +918826554477               gk-1 35-B New Delhi India -110019  25-07-2023  01-08-2023      Suit        5200          True         India
     ###############
SHOW TASK MENU ?
Y/N:
```

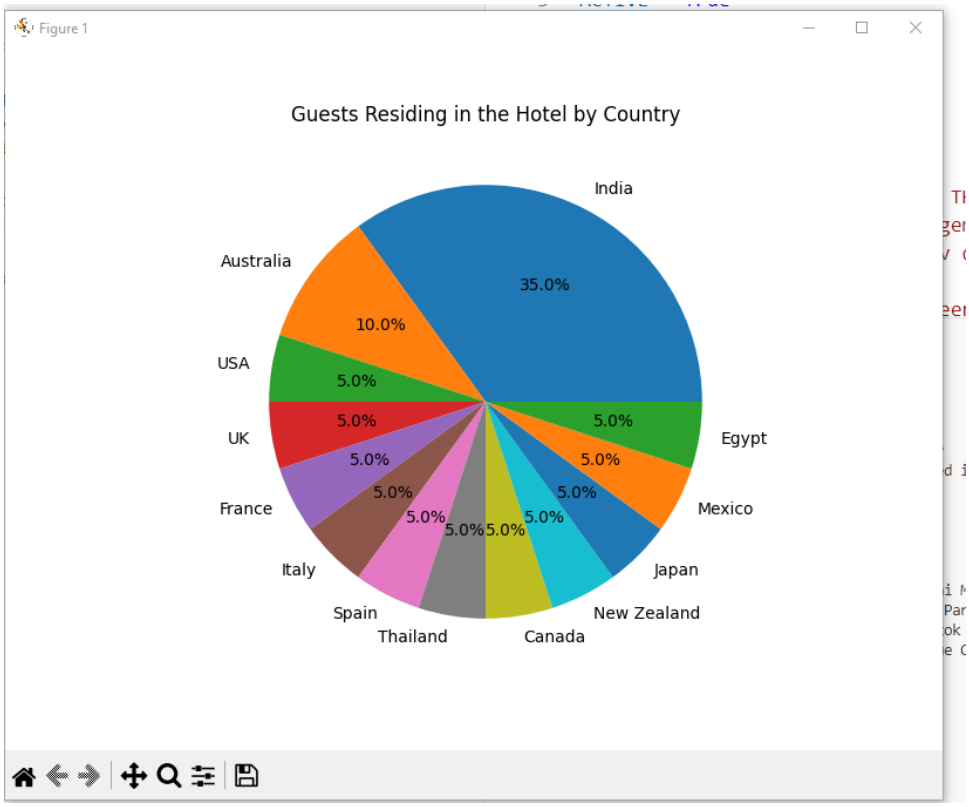
```
select the task : 4
     ### Search reservation record by name ###
       #### All Guests Name ####

['Sameer', 'John', 'Rahul', 'Emily', 'Sandeep', 'Arjun', 'Oliver', 'Aarav', 'Isabella', 'Ravi', 'Emma', 'Aryan', 'Mia', 'Abhishek', 'Liam', 'Anaya', 'Alexander', 'Sanvi',
'Leo', 'Angana']
pls check the casing of the name
enter the Name to check reservation: Liam
Record Found
    Name  Contact_Number                                      Address    Check_In   Check_Out     Room_Type Total_Price Is_Checked_In Guest_Country
15  Liam  +91-9876543224  246 Cedar Avenue Vancouver British Columbia Ca...  10-06-2025  18-06-2025  Super Deluxe     18000.0          True        Canada
     ###############
SHOW TASK MENU ?
Y/N:
```

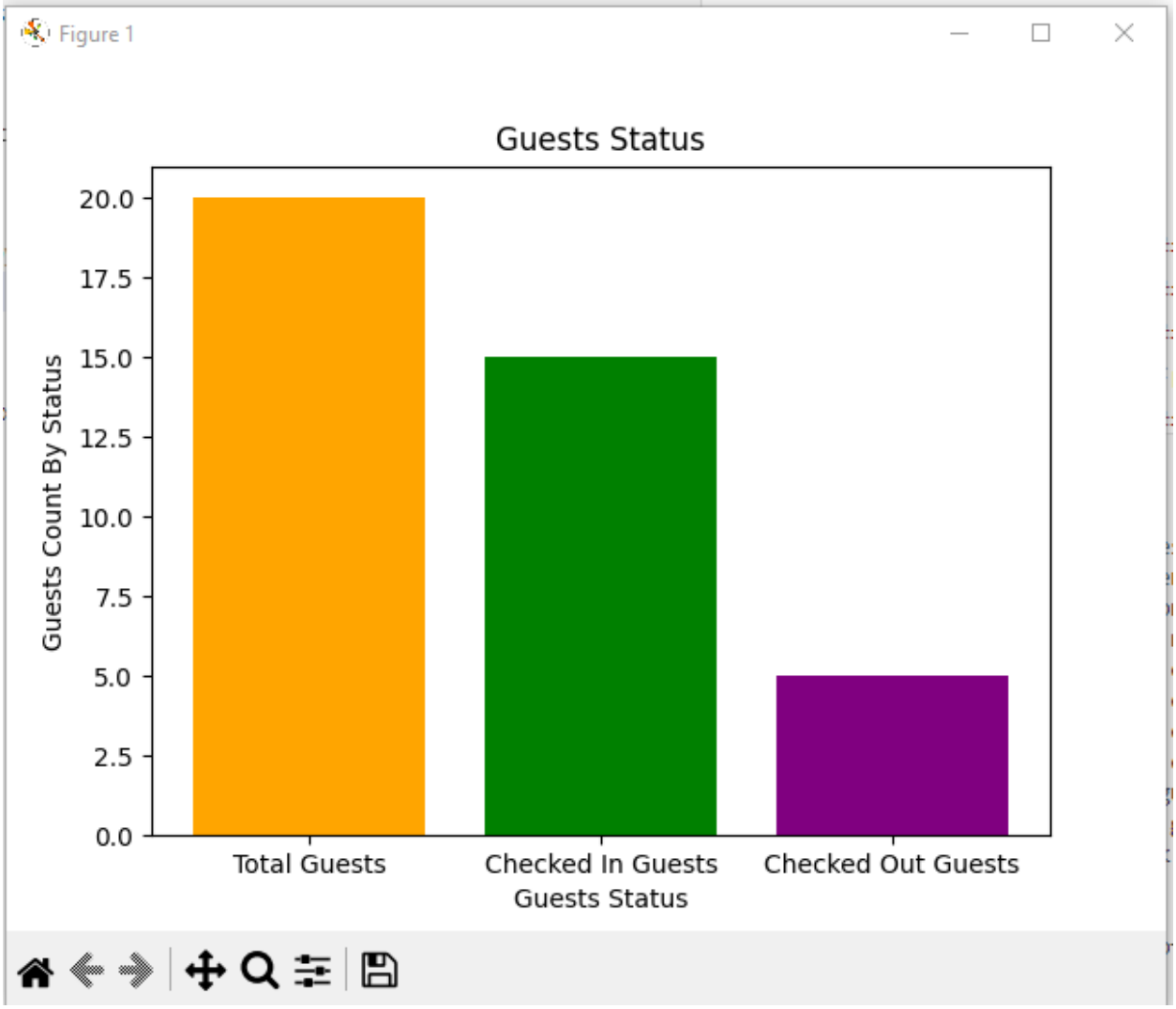
```
select the task : 5
     ### Search reservations record by check in date ###
All distinct check in dates in the system
{'15-10-2025', '01-04-2025', '10-05-2023', '25-08-2025', '25-07-2023', '25-12-2022', '05-11-2024', '12-09-2024', '20-01-2026', '10-06-2025', '15-08-2023', '25-12-2024', '1
0-02-2024', '01-12-2025', '15-02-2025', '01-08-2024', '25-04-2024', '01-07-2023', '15-06-2024', '05-10-2023'}
enter the check-in date in format dd-mm-yyyy: 05-10-2023
     ### reservations by check in date 05-10-2023 found ###
     Name  Contact_Number                                  Address    Check_In   Check_Out Room_Type Total_Price Is_Checked_In Guest_Country
4  Sandeep  +91-9876543213  246 Pine Street Bengaluru Karnataka India  05-10-2023  15-10-2023      Suit     30000.0          True         India
     ###############
SHOW TASK MENU ?
Y/N:
```

```
select the task : 6
     ### Search reservations record by check out date ###
enter the check-out date in format dd-mm-yyyy: 05-09-2025
     ### reservations by check out date 05-09-2025 found ###
     Name  Contact_Number                                        Address    Check_In   Check_Out Room_Type Total_Price Is_Checked_In Guest_Country
16  Anaya  +91-9876543225  753 Birch Street Auckland Auckland New Zealand  25-08-2025  05-09-2025  Standard      5000.0          True   New Zealand
     ###############
SHOW TASK MENU ?
Y/N:
```

```
select the task : 7
     ### all checked in reservations ###
         Name  Contact_Number                                        Address    Check_In   Check_Out     Room_Type Total_Price Is_Checked_In Guest_Country
0      Sameer  +91-9988776655           Block-C Janak Puri New Delhi Delhi India  25-12-2022  25-01-2023        Deluxe     45000.0          True         India
1        John  +91-9876543210                    123 Main Street New York NY USA  10-05-2023  15-05-2023  Super Deluxe     10000.0          True           USA
3       Emily  +91-9876543212                  789 Oak Street London England UK  15-08-2023  25-08-2023  Super Deluxe     18000.0          True            UK
4     Sandeep  +91-9876543213          246 Pine Street Bengaluru Karnataka India  05-10-2023  15-10-2023         Suit     30000.0          True         India
6       Arjun  +91-9876543215              987 Elm Street Jaipur Rajasthan India  10-02-2024  20-02-2024     Standard      6000.0          True         India
8       Aarav  +91-9876543217        321 Walnut Street Kolkata West Bengal India  15-06-2024  25-06-2024  Super Deluxe     16000.0          True         India
9    Isabella  +91-9876543218                 876 Birch Street Rome Italy Italy  01-08-2024  10-08-2024         Suit     20000.0          True         Italy
10       Ravi  +91-9876543219           753 Pine Street Chennai Tamil Nadu India  12-09-2024  18-09-2024     Standard      6000.0          True         India
11       Emma  +91-9876543220    987 Oak Avenue Sydney New South Wales Australia  05-11-2024  10-11-2024        Deluxe      9000.0          True     Australia
12      Aryan  +91-9876543221     246 Maple Street Brisbane Queensland Australia  25-12-2024  02-01-2025  Super Deluxe     14000.0          True     Australia
13        Mia  +91-9876543222           753 Elm Avenue Barcelona Catalonia Spain  15-02-2025  25-02-2025         Suit     22500.0          True         Spain
15       Liam  +91-9876543224  246 Cedar Avenue Vancouver British Columbia Ca...  10-06-2025  18-06-2025  Super Deluxe     18000.0          True        Canada
16      Anaya  +91-9876543225     753 Birch Street Auckland Auckland New Zealand  25-08-2025  05-09-2025     Standard      5000.0          True   New Zealand
17  Alexander  +91-9876543226                  987 Pine Avenue Tokyo Kanto Japan  15-10-2025  20-10-2025         Suit     12500.0          True         Japan
18      Sanvi  +91-9876543227      246 Oak Street Mexico City Mexico City Mexico  01-12-2025  10-12-2025     Standard      7000.0          True        Mexico
     ###############
SHOW TASK MENU ?
Y/N:
```

```
select the task : 8
     ### all checked in reservations ###
        Name  Contact_Number                                    Address    Check_In   Check_Out Room_Type Total_Price Is_Checked_In Guest_Country
2      Rahul  +91-9876543211    456 Park Avenue Mumbai Maharashtra India  01-07-2023  10-07-2023  Standard      7000.0          False         India
7     Oliver  +91-9876543216        654 Cedar Avenue Paris France France  25-04-2024  05-05-2024    Deluxe      7500.0          False        France
14  Abhishek  +91-9876543223  987 Walnut Street Bangkok Bangkok Thailand  01-04-2025  10-04-2025    Deluxe     10500.0          False      Thailand
19       Leo  +91-9876543228          753 Maple Avenue Cairo Cairo Egypt  20-01-2026  30-01-2026    Deluxe     13500.0          False         Egypt
     ###############
SHOW TASK MENU ?
Y/N:
```






```
12 --- Exit the application
select the task : 11
    #### Saving Records To The Csv ####
 #### records saved back to csv ####
    ###############
SHOW TASK MENU ?
Y/N:
```

```
select the task : 12
Exiting...
    ###############
    #### SHUTING DOWN THE APPLICATION ###
    #### Have A Nice Day :D  ###
(myenv) PS D:\geek-practice\project_dojo\hotel_managment_pandas>
```