Project Report on

RAILWAY RESERVATION MANAGEMENT SYSTEM

Submitted to

JAYSHREE PERIWAL HIGH SCHOOL
3, Chitrakoot Scheme Jaipur

In partial fulfillment of the requirements for
All India Senior School Certificate Examination 2025

Of

CENTRAL BOARD OF SECONDARY EDUCATION

Submitted by:

Samarth Khandelwal (Roll No.)
Aradhya Gupta (Roll No.)
Arav Kilak (Roll No.)

ACKNOWLEDGEMENT

We would like to thank everyone who helped us to accomplish this project.

Our sincere thanks to ***respected teachers*** who have helped us with their valuable suggestions and support throughout the development of the project.

We would like to thank our project guide **Mrs. Himanshi Sharma** for providing guidance and support at every stage of the project.

We are extremely grateful to **Mrs. Jayshree Periwal, Director and Mrs. Madhu Maini, Principal of JAYSHREE PERIWAL HIGH SCHOOL, JAIPUR**, for providing us with a computer lab, due to which we were able to complete our project.

Samarth Khandelwal
Aradhya Gupta
Arav Kilak

CERTIFICATE OF ORIGINALITY

This is to certify that the project entitled

"RAILWAY RESERVATION MANAGEMENT SYSTEM"

submitted to
JAYSHREE PERIWAL HIGH SCHOOL

In partial fulfillment of the requirement for **All India Senior School Certificate Examination (AISSCE) 2025** of CBSE, is original work carried out by **Samarth Khandelwal, Aradhya Gupta, and Arav Kilak** under my guidance.

The matter embodied in this project is genuine work done by the students and has not been submitted by any course of study.

_____

Signature of the guide:
Date: _____
Name: **Mrs. Himanshi Sharma**
HOD, Computer Science
JAYSHREE PERIWAL HIGH SCHOOL
Jaipur

CONTENTS

1. Objective & Scope of the Project

Objective

The Railway Reservation system facilitates the CONSUMERS :

- to enquire about trains available on the basis of source and destination
- Booking and cancellation of tickets.
- enquire about the status of the booked ticket.
- To view the details of the booked tickets etc.

Scope

This project is developed as a part of XII standard Course.

It can be easily used by NATIONAL RAILWAYS to add/remove view details, status of the tickets and the passengers.

PROBLEM DEFINITION

Railways are a critical mode of transportation that facilitate the movement of passengers and freight across vast distances. However, with increasing demand, existing railway management systems face challenges such as scheduling inefficiencies, maintenance issues, safety concerns, and customer dissatisfaction.

Life cycle of the project

System Development Life Cycle (SDLC)

The System Development Life Cycle (SDLC) is a set of activities that analysts, designers and users carry out to develop and implement an Information System.

The SDLC consists of the following activities.

Feasibility Study

↓

Requirement Definition (Analysis)

↓

Design (Database & Program)

↓

Development of Software

↓

Unit Testing

↓

System Testing

↓

Implementation

↓

Evaluation

↓

Maintenance

4. Details of Hardware and Software used

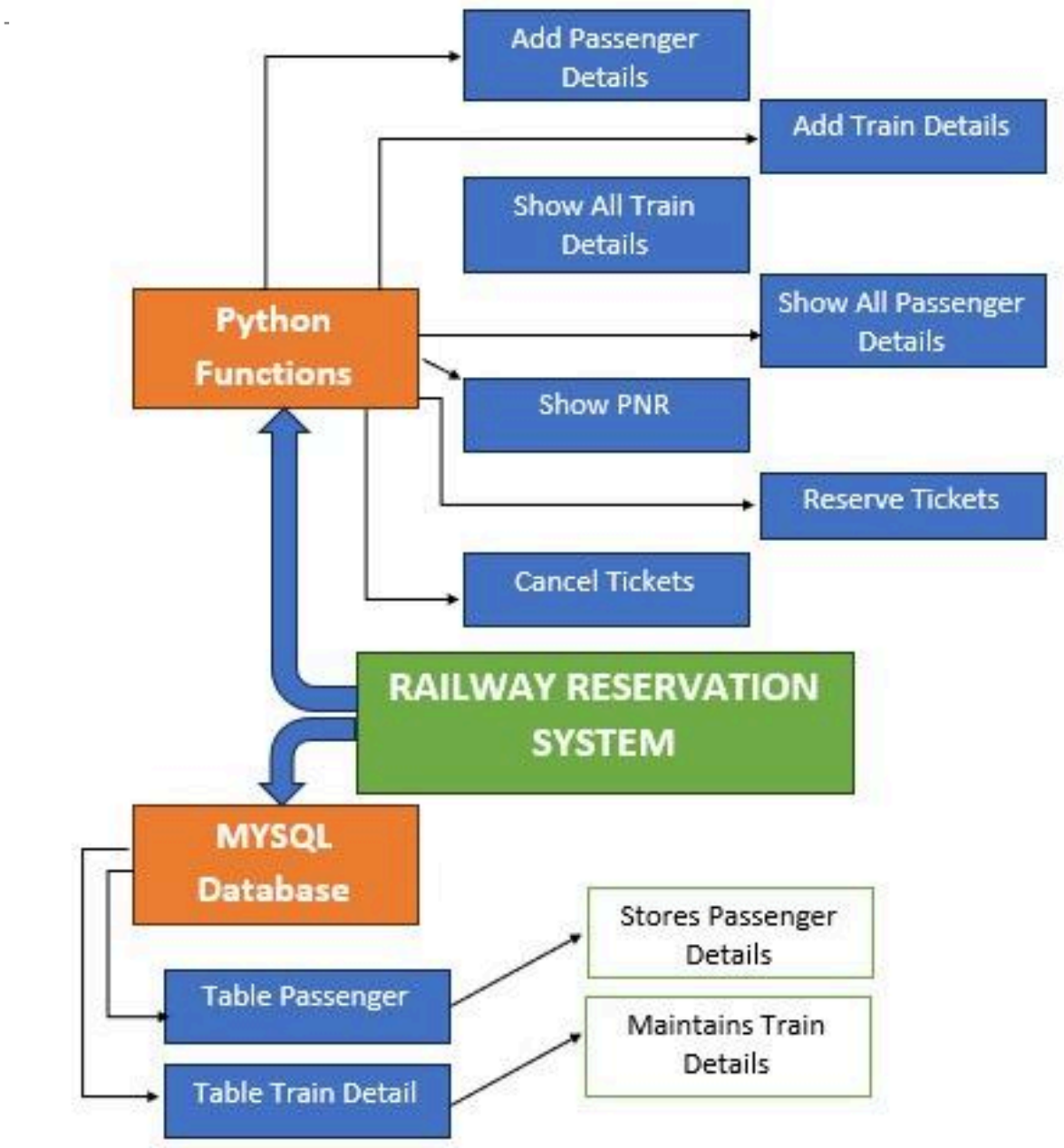
Hardware Specifications

Microprocessor (CPU) : Intel(R) Core(TM) i5-12600K
Memory (RAM) : 16 GB DDR5
Hard Disk : 1TB SSD
VDU : HDMI
Keyboard : Logitech G413
Mouse : Logitech MX Master 3
Printer : Inkjet / Laser
GPU : NVidia(R) GeForce(TM) RTX 3060 Ti

Software Specifications

Operating System : Windows 11
Front-End Design : Python 3.12
Back-End : MySQL 8.0
Documentation : Microsoft Word 365
Version Control : GIT

Context Diagram

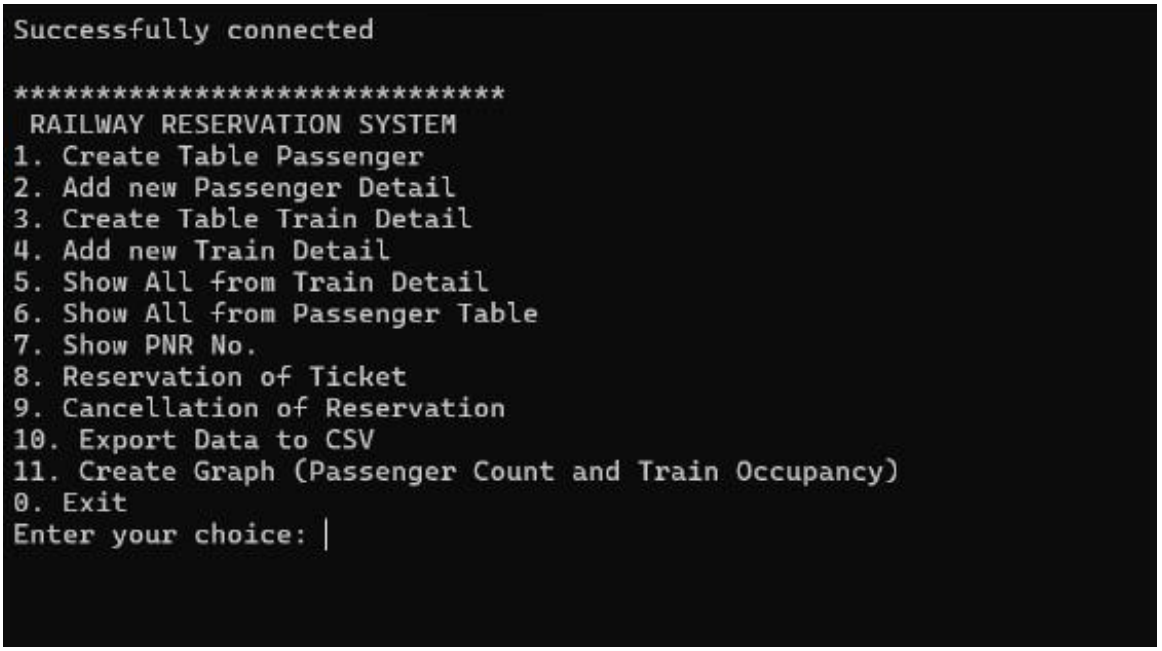


PROJECT INTERFACE

Input Screen Design

WELCOME SCREEN

- This is the first window to appear on executing the python program. It is the main login window.



1. TO CREATE A TABLE PASSENGER SELECT 1

```
mysql> desc passengers;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| pname       | varchar(30) | YES  |     | NULL    |       |
| age         | varchar(25) | YES  |     | NULL    |       |
| trainno     | varchar(30) | YES  |     | NULL    |       |
| noofpas     | varchar(25) | YES  |     | NULL    |       |
| cls         | varchar(25) | YES  |     | NULL    |       |
| destination | varchar(30) | YES  |     | NULL    |       |
| amt         | varchar(20) | YES  |     | NULL    |       |
| status      | varchar(25) | YES  |     | NULL    |       |
| pnrno       | varchar(30) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
9 rows in set (0.09 sec)
```

2. TO ADD A NEW PASSENGER DETAIL

```
Enter your choice: 2
ENTER NAME: ROHAN AGARWAL
ENTER AGE: 19
ENTER TRAIN NO.: 10001
ENTER NO. OF PASSENGERS: 2
ENTER CLASS: PREMIUM
ENTER DESTINATION: NEW DELHI
ENTER FARE: 5000
ENTER STATUS: CONFIRMED
ENTER PNR NO.: 12121
Record of Passenger inserted
```

3. TO CREATE A TABLE TRAIN DETAIL

```
mysql> desc trainsdetail;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| tname       | varchar(30) | YES  |     | NULL    |       |
| tnum        | varchar(25) | YES  |     | NULL    |       |
| source      | varchar(30) | YES  |     | NULL    |       |
| destination | varchar(30) | YES  |     | NULL    |       |
| fare        | varchar(10) | YES  |     | NULL    |       |
| ac1         | varchar(25) | YES  |     | NULL    |       |
| ac2         | varchar(30) | YES  |     | NULL    |       |
| slp         | varchar(25) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
8 rows in set (0.04 sec)
```

4. TO ADD A NEW TRAIN DETAIL SELECT 4

```
ENTER TRAIN NAME: NEW DELHI EXPRESS
ENTER NUMBER OF TRAIN: 10001
ENTER SOURCE OF TRAIN: NEW DELHI
ENTER DESTINATION OF TRAIN: NEW DELHI
ENTER FARE OF STATION: 5000
ENTER No. OF SEATS FOR AC1: 25
ENTER No. OF SEATS FOR AC2: 30
ENTER No. OF SEATS FOR SLEEPER: 30
Record inserted in Trains Detail
```

5. TO SHOW ALL TRAINSDETAIL SELECT 5

```
Enter your choice: 5
ALL TRAINS DETAILS
          tname   tnum      source destination   fare ac1 ac2 slp
0      Goa Express  12345   New Delhi       Goa   6000  10  20  50
1     Jammu Express  67890   New Delhi     Jammu   9000  15  25  60
2    kashmir express  10001  kanyakumari    kashmir  10000  50  50  50
3  NEW DELHI EXPRESS   10001   NEW DELHI   NEW DELHI   5000  25  30  30
```

6. TO SHOW ALL PASSENGER DETAILS SELECT 6

	pname	age	trainno	noofpas	...	destination	amt	status	pnrno
0	Bob Johnson	25	12345	2	...	Mumbai	1000	Confirmed	PNR002
1	Charlie Brown	35	67890	1	...	Delhi	2500	Waiting	PNR003
2	ROHAN AGARWAL	19	10001	2	...	NEW DELHI	5000	CONFIRMED	12121

[3 rows x 9 columns]

7. TO SHOW PNRNO DETAILS SELECT 7

```
Enter your choice: 7
PNR STATUS WINDOW
Enter Train No.: 67890

      pname   status
0  Charlie Brown  Waiting

**********************************
```

8. TO SELECT TICKETS SELECT 8

```
Enter your choice: 8
WE HAVE THE FOLLOWING SEAT TYPES FOR YOU:
TNAME is 1 FOR GOA EXPRESS FROM NEW DELHI:
1. FIRST CLASS AC RS 6000 Per PERSON
2. SECOND CLASS AC RS 5000 Per PERSON
3. THIRD CLASS AC RS 4000 Per PERSON
4. FOR SLEEPER RS 3000 Per PERSON
TNAME is 2 FOR JAMMU EXPRESS FROM NEW DELHI:
1. FIRST CLASS AC RS 10000 Per PERSON
2. SECOND CLASS AC RS 9000 Per PERSON
3. THIRD CLASS AC RS 8000 Per PERSON
4. FOR SLEEPER RS 7000 Per PERSON
ENTER YOUR CHOICE OF TICKET PLEASE: 1
HOW MANY TICKETS YOU NEED: 4
YOU HAVE CHOSEN FIRST CLASS AC TICKET
YOUR TICKET PRICE is = 24000
```

9. FOR CANCELLATION OF RESERVATION

```
Enter your choice: 9
Before any Changes in the STATUS

      pname age trainno noofpas ... destination  amt    status  pnrno
0   Bob Johnson  25   12345     2 ...     Mumbai  1000  Confirmed  PNR002
1  Charlie Brown  35   67890     1 ...      Delhi  2500   Waiting  PNR003
2  ROHAN AGARWAL  19   10001     2 ...  NEW DELHI  5000  CONFIRMED  12121

[3 rows x 9 columns]
Enter PNR No. to cancel: PNR002
Reservation with PNR No. PNR002 has been cancelled.
      pname age trainno noofpas ... destination  amt    status  pnrno
0   Bob Johnson  25   12345     2 ...     Mumbai  1000  cancelled  PNR002
1  Charlie Brown  35   67890     1 ...      Delhi  2500   Waiting  PNR003
2  ROHAN AGARWAL  19   10001     2 ...  NEW DELHI  5000  CONFIRMED  12121

[3 rows x 9 columns]
```

10. TO EXPORT TO CSV

```
Enter your choice: 10
Choose the table to export to CSV:
1. Export Passengers Table
2. Export Trains Detail Table
Enter your choice: 1
```

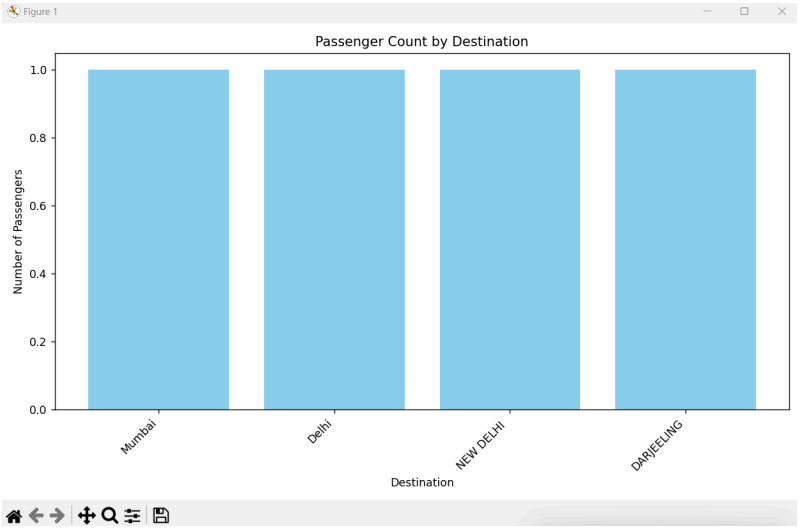
```
Enter the filename (e.g., passengers_data.csv): passengers_data
Passengers data exported to passengers_data
```

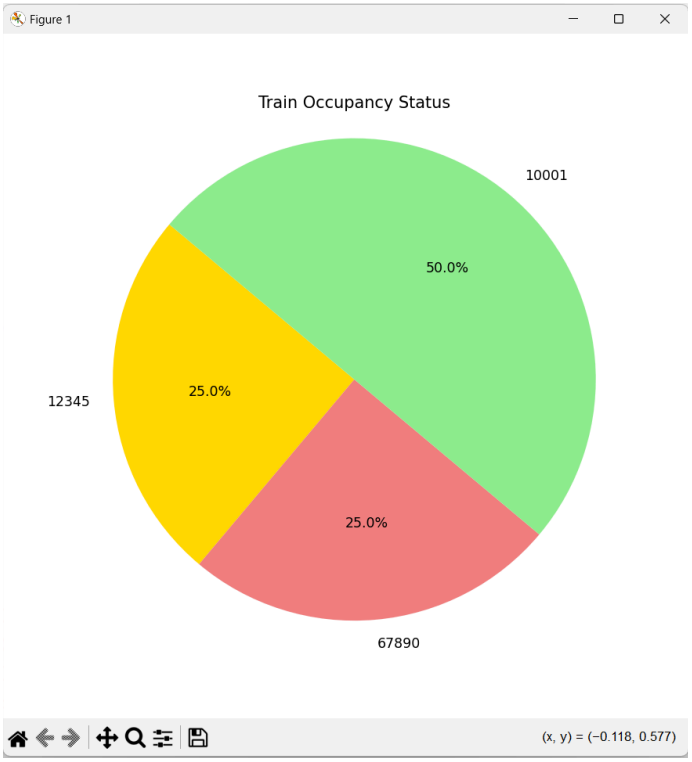
	A	B	C	D	E	F	G	H	I
1	pname	age	trainno	noofpas	cls	destination	amt	status	pnrno
2	Bob Johnson	25	12345	2	SL	Mumbai	1000	cancelled	PNR002
3	Charlie Brown	35	67890	1	AC	Delhi	2500	Waiting	PNR004
4	ROHAN AGARWAL	19	10001	2	PREMIUM	NEW DELHI	5000	CONFIRMI	PNR003
5	ARAV KILAK	20	10001	2	PREMIUM	DARJEELING	5000	CONFIRMI	PNR005
6									

11. TO CREATE A GRAPH OF TRAIN OCCUPANCY AND PASSENGER

```
Enter your choice: 11

Choose the type of graph to create:
1. Passenger Count by Destination (Bar Chart)
2. Train Occupancy Status (Pie Chart)
Enter your choice: 1
```





12. TO EXIT

SOURCE CODE

SQL DATABASE

```
mysql> use railway
Database changed
mysql> show tables;
+-------------------+
| Tables_in_railway |
+-------------------+
| passengers        |
| trainsdetail      |
+-------------------+
2 rows in set (0.12 sec)
```

TABLE TRAINSDETAIL

```
mysql> desc trainsdetail;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| tname       | varchar(30) | YES  |     | NULL    |       |
| tnum        | varchar(25) | YES  |     | NULL    |       |
| source      | varchar(30) | YES  |     | NULL    |       |
| destination | varchar(30) | YES  |     | NULL    |       |
| fare        | varchar(10) | YES  |     | NULL    |       |
| ac1         | varchar(25) | YES  |     | NULL    |       |
| ac2         | varchar(30) | YES  |     | NULL    |       |
| slp         | varchar(25) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

TABLE PASSENGERS

```
mysql> desc passengers;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| pname       | varchar(30) | YES  |     | NULL    |       |
| age         | varchar(25) | YES  |     | NULL    |       |
| trainno     | varchar(30) | YES  |     | NULL    |       |
| noofpas     | varchar(25) | YES  |     | NULL    |       |
| cls         | varchar(25) | YES  |     | NULL    |       |
| destination | varchar(30) | YES  |     | NULL    |       |
| amt         | varchar(20) | YES  |     | NULL    |       |
| status      | varchar(25) | YES  |     | NULL    |       |
| pnrno       | varchar(30) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
9 rows in set (0.03 sec)
```

DATA TRAINSDETAIL

```
mysql> select * from trainsdetail;
+-------------------+-------+-------------+-------------+-------+------+------+------+
| tname             | tnum  | source      | destination | fare  | ac1  | ac2  | slp  |
+-------------------+-------+-------------+-------------+-------+------+------+------+
| Goa Express       | 12345 | New Delhi   | Goa         | 6000  | 10   | 20   | 50   |
| Jammu Express     | 67890 | New Delhi   | Jammu       | 9000  | 15   | 25   | 60   |
|  kashmir express  | 10001 | kanyakumari | kashmir     | 10000 | 50   | 50   | 50   |
| NEW DELHI EXPRESS | 10001 | NEW DELHI   | NEW DELHI   | 5000  | 25   | 30   | 30   |
+-------------------+-------+-------------+-------------+-------+------+------+------+
4 rows in set (0.00 sec)
```

DATA PASSENGERS

```
mysql> select * from passengers;
+---------------+------+---------+---------+-----------+-------------+------+-----------+--------+
| pname         | age  | trainno | noofpas | cls       | destination | amt  | status    | pnrno  |
+---------------+------+---------+---------+-----------+-------------+------+-----------+--------+
| Bob Johnson   | 25   | 12345   | 2       | SL        | Mumbai      | 1000 | cancelled | PNR002 |
| Charlie Brown | 35   | 67890   | 1       | AC        | Delhi       | 2500 | Waiting   | PNR004 |
| ROHAN AGARWAL | 19   | 10001   | 2       | PREMIUM   | NEW DELHI   | 5000 | CONFIRMED | PNR003 |
| ARAV KILAK    | 20   | 10001   | 2       | PREMIUM   | DARJEELING  | 5000 | CONFIRMED | PNR005 |
+---------------+------+---------+---------+-----------+-------------+------+-----------+--------+
4 rows in set (0.00 sec)
```

PYTHON CODE

```
import warnings
import pandas as pd
import mysql.connector as sql
import matplotlib.pyplot as plt

# Suppress warnings for a cleaner output
warnings.filterwarnings("ignore", category=UserWarning, module="pandas")

# Connect to the MySQL database
conn = sql.connect(host='localhost', user='root', passwd='root', database='railway')
if conn.is_connected():
    print('Successfully connected')

def menu():
    print()
    print("*******************************")
    print(" RAILWAY RESERVATION SYSTEM   ")
    print("1. Create Table Passenger")
    print("2. Add new Passenger Detail")
    print("3. Create Table Train Detail")
    print("4. Add new Train Detail")
    print("5. Show All from Train Detail")
    print("6. Show All from Passenger Table")
    print("7. Show PNR No.")
    print("8. Reservation of Ticket")
    print("9. Cancellation of Reservation")
    print("10. Export Data to CSV")  # New option for exporting data
    print("11. Create Graph (Passenger Count and Train Occupancy)")  # New option for graph
    print("0. Exit")


def create_passengers():
    cl = conn.cursor()
    cl.execute("""
        CREATE TABLE IF NOT EXISTS passengers (
            pname VARCHAR(30),
            age VARCHAR(25),
            trainno VARCHAR(30),
            noofpas VARCHAR(25),
            cls VARCHAR(25),
            destination VARCHAR(30),
            amt VARCHAR(20),
            status VARCHAR(25),
            pnrno VARCHAR(30)
        )
    """)
    conn.commit()
    print('Table passengers created')
```

```
def add_passengers():
    cl = conn.cursor()
    L = []
    pname = input("ENTER NAME: ")
    L.append(pname)
    age = input("ENTER AGE: ")
    L.append(age)
    trainno = input("ENTER TRAIN NO.: ")
    L.append(trainno)
    noofpas = input("ENTER NO. OF PASSENGERS: ")
    L.append(noofpas)
    cls = input("ENTER CLASS: ")
    L.append(cls)
    destination = input("ENTER DESTINATION: ")
    L.append(destination)
    amt = input("ENTER FARE: ")
    L.append(amt)
    status = input("ENTER STATUS: ")
    L.append(status)
    pnrno = input("ENTER PNR NO.: ")
    L.append(pnrno)
    pas = tuple(L)
    sql_query = """
        INSERT INTO passengers (pname, age, trainno, noofpas, cls, destination, amt, status, pnrno)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
    cl.execute(sql_query, pas)
    conn.commit()
    print('Record of Passenger inserted')
    df = pd.read_sql("SELECT * FROM passengers", conn)
    print(df)

def create_trainsdetail():
    cl = conn.cursor()
    cl.execute("""
        CREATE TABLE IF NOT EXISTS trainsdetail (
            tname VARCHAR(30),
            tnum VARCHAR(25),
            source VARCHAR(30),
            destination VARCHAR(30),
            fare VARCHAR(10),
            ac1 VARCHAR(25),
            ac2 VARCHAR(30),
            slp VARCHAR(25)
        )
    """)
    conn.commit()
    print('Table trainsdetail created')


def add_traindetail():
    cl = conn.cursor()
    df = pd.read_sql("SELECT * FROM trainsdetail", conn)
    print(df)
    L = []
    tname = input("ENTER TRAIN NAME: ")
    L.append(tname)
    tnum = input("ENTER NUMBER OF TRAIN: ")
    L.append(tnum)
    source = input("ENTER SOURCE OF TRAIN: ")
    L.append(source)
    destination = input("ENTER DESTINATION OF TRAIN: ")
    L.append(destination)
    fare = input("ENTER FARE OF STATION: ")
    L.append(fare)
    ac1 = input("ENTER No. OF SEATS FOR AC1: ")
    L.append(ac1)
    ac2 = input("ENTER No. OF SEATS FOR AC2: ")
    L.append(ac2)
    slp = input("ENTER No. OF SEATS FOR SLEEPER: ")
    L.append(slp)
    f = tuple(L)
    sql = """
        INSERT INTO trainsdetail (tname, tnum, source, destination, fare, ac1, ac2, slp)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
    """
    cl.execute(sql, f)
    conn.commit()
    print('Record inserted in Trains Detail')

def show_passengers():
    print('ALL PASSENGERS DETAIL')
    df = pd.read_sql("SELECT * FROM passengers", conn)
    print(df)
```

```
def show_trains_detail():
    print("ALL TRAINS DETAILS")
    df = pd.read_sql("SELECT * FROM trainsdetail", conn)
    print(df)

def disp_pnrno():
    print("PNR STATUS WINDOW")
    a = input("Enter Train No.: ")
    qry = "SELECT pname, status FROM passengers WHERE trainno=%s;"
    df = pd.read_sql(qry, conn, params=(a,))
    print(df)

def ticketreservation():
    print("WE HAVE THE FOLLOWING SEAT TYPES FOR YOU:")
    print("TNAME is 1 FOR GOA EXPRESS FROM NEW DELHI:")
    print("1. FIRST CLASS AC RS 6000 Per PERSON")
    print("2. SECOND CLASS AC RS 5000 Per PERSON")
    print("3. THIRD CLASS AC RS 4000 Per PERSON")
    print("4. FOR SLEEPER RS 3000 Per PERSON")
    print("TNAME is 2 FOR JAMMU EXPRESS FROM NEW DELHI:")
    print("1. FIRST CLASS AC RS 10000 Per PERSON")
    print("2. SECOND CLASS AC RS 9000 Per PERSON")
    print("3. THIRD CLASS AC RS 8000 Per PERSON")
    print("4. FOR SLEEPER RS 7000 Per PERSON")

    a = int(input("ENTER YOUR CHOICE OF TICKET PLEASE: "))
    n = int(input("HOW MANY TICKETS YOU NEED: "))

    if a == 1:
        print("YOU HAVE CHOSEN FIRST CLASS AC TICKET")
        s = 6000 * n
    elif a == 2:
        print("YOU HAVE CHOSEN SECOND CLASS AC TICKET")
        s = 5000 * n
    elif a == 3:
        print("YOU HAVE CHOSEN THIRD CLASS AC TICKET")
        s = 4000 * n
    elif a == 4:
        print("YOU HAVE CHOSEN SLEEPER TICKET")
        s = 3000 * n
    else:
        print("Invalid option. PLEASE CHOOSE A TRAIN")
        return

    print("YOUR TICKET PRICE is =", s, "\n")
```

```
def cancel():
    print("Before any Changes in the STATUS")
    df = pd.read_sql("SELECT * FROM passengers", conn)
    print(df)
    pnr_no = input("Enter PNR No. to cancel: ")
    cl = conn.cursor()
    cl.execute("UPDATE passengers SET status='cancelled' WHERE pnrno=%s", (pnr_no,))
    conn.commit()
    print(f"Reservation with PNR No. {pnr_no} has been cancelled.")
    df = pd.read_sql("SELECT * FROM passengers", conn)
    print(df)

def export_to_csv():
    print("Choose the table to export to CSV:")
    print("1. Export Passengers Table")
    print("2. Export Trains Detail Table")

    choice = int(input("Enter your choice: "))

    if choice == 1:
        # Export passengers table to CSV
        df = pd.read_sql("SELECT * FROM passengers", conn)
        filename = input("Enter the filename (e.g., passengers_data.csv): ")
        df.to_csv(filename, index=False)
        print(f"Passengers data exported to {filename}")

    elif choice == 2:
        # Export trainsdetail table to CSV
        df = pd.read_sql("SELECT * FROM trainsdetail", conn)
        filename = input("Enter the filename (e.g., trains_detail_data.csv): ")
        df.to_csv(filename, index=False)
        print(f"Trains detail data exported to {filename}")

    else:
        print("Invalid choice! Please choose a valid option.")
def create_graph():
    print("\nChoose the type of graph to create:")
    print("1. Passenger Count by Destination (Bar Chart)")
    print("2. Train Occupancy Status (Pie Chart)")

    choice = int(input("Enter your choice: "))

    if choice == 1:
        # Generate Passenger Count by Destination
        df = pd.read_sql("SELECT destination, COUNT(*) as passenger_count FROM passengers GROUP BY destination", conn)
        destinations = df['destination']
        passenger_count = df['passenger_count']

        plt.figure(figsize=(10,6))
        plt.bar(destinations, passenger_count, color='skyblue')
        plt.xlabel('Destination')
        plt.ylabel('Number of Passengers')
        plt.title('Passenger Count by Destination')
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()

    elif choice == 2:
        # Generate Train Occupancy Status (Pie Chart)
        df = pd.read_sql("SELECT trainno, COUNT(*) as booked_count FROM passengers GROUP BY trainno", conn)
        trains = df['trainno']
        booked_count = df['booked_count']

        plt.figure(figsize=(7,7))
        plt.pie(booked_count, labels=trains, autopct='%1.1f%%', startangle=140, colors=['gold', 'lightcoral', 'lightgreen', 'lightblue'])
        plt.title('Train Occupancy Status')
        plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
        plt.show()

    else:
        print("Invalid choice! Please select a valid graph type.")
```

```
# Main loop
while True:
    menu()
    opt = int(input("Enter your choice: "))
    if opt == 1:
        create_passengers()
    elif opt == 2:
        add_passengers()
    elif opt == 3:
        create_trainsdetail()
    elif opt == 4:
        add_traindetail()
    elif opt == 5:
        show_trains_detail()
    elif opt == 6:
        show_passengers()
    elif opt == 7:
        disp_pnrno()
    elif opt == 8:
        ticketreservation()
    elif opt == 9:
        cancel()
    elif opt == 10:
        export_to_csv()
    elif opt == 11:
        create_graph()  # New option to create graphs
    elif opt == 0:
        print("Exiting...")
        break
    else:
        print('Invalid option. Please try again.')
```