HOPE HALL FOUNDATION SCHOOL

CBSE AISSCE

RAILWAY RESERVATION SYSTEM

Computer Science (083) Project

Sagnik Pal
Class-12 Science C
Roll -27

UNDERTAKING

I declare that the work presented in this project titled "**Railway Reservation System**", is submitted to Ms. Rachna Gupta – Computer science teacher at Hope Hall Foundation School for the award of the CBSE AISSCE Class 12 certificate. I have not plagiarized or submitted the same work for the award of any other examination. In case this undertaking is found incorrect, I accept that my certificates may be unconditionally withdrawn.

14th March, 2022
Hope Hall Foundation School

Sagnik Pal
12 Science C
Roll 27

CERTIFICATE

Certified that the work contained in the project titled “Railway Reservation System” by Sagnik Pal has been carried out under my supervision and that this work has not been submitted elsewhere for an AISSCE certificate

Rachna Gupta
Computer Science teacher
Hope Hall Foundation School

ACKNOWLEDGEMENTS

I would like to thank Mr. Bhagat Singh, principal of Hope Hall Foundation School

I am deeply indebted to my mentor Ms. Rachna Gupta

I owe my sincere gratitude towards Hope Hall Foundation School

My heartfelt thanks to CBSE

I also express my deepest gratitude to my family and friends

SAGNIK PAL
12 Science C
Roll 27

CONTENTS

About the Project-

Our aim in this project is to develop a Railway Reservation System in Python. It is a Simple Railway Reservation System that allows User to –

- Search Trains
- Book Tickets
- Cancel tickets
- Check PNR
- It also has Admin side which allows extra features like Add Trains and change stations.

How will the project help others?

A Railway Reservation System allows passengers to book and cancel tickets from any of the terminals. These tickets can be purchased or cancelled for trips all around the country with travel times of up to 72 hours and distances of up to thousands of kilometers.

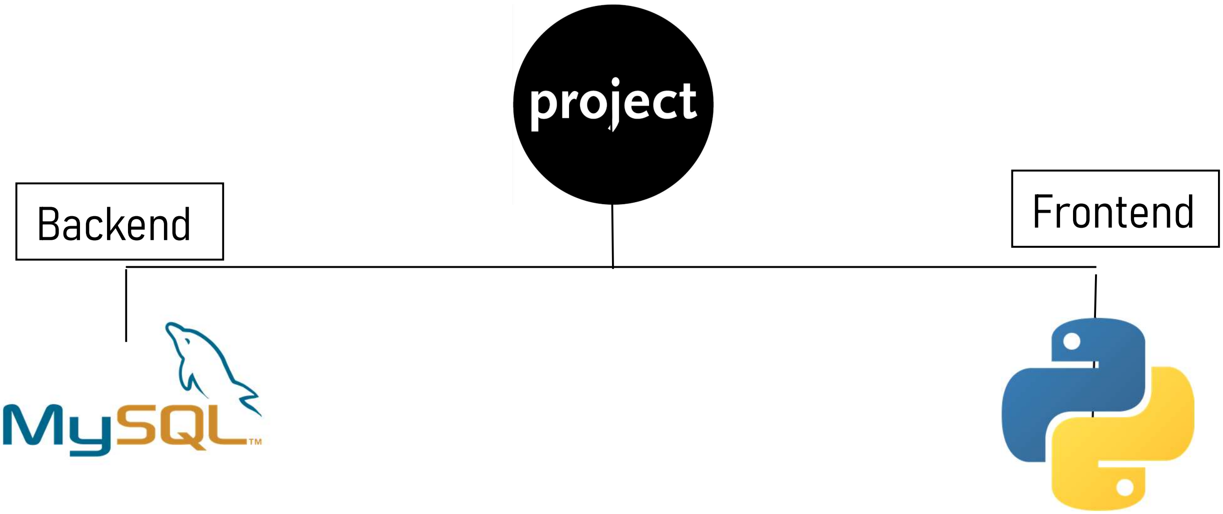
Often many a times people live far from railway stations which makes the process of booking tickets both Cumbersome and Time Consuming.

However, our project will enable users to book tickets conveniently from their homes with any device with an internet connection in it.

A few points about the Project –

For the project we have used the below mentioned tools-

- For Backend development we have used MySQL for database.
- Database is hosted on the internet, and publicly accessible by users
- And for the frontend or the user interface development we used Python.
- Our Project will have command line based interface



project

Backend

Frontend

Hardware requirements-

Given Below are the minimum Hardware Requirements-

- A computer, Desktop or Laptop.
- Processor – Any modern Processor Capable of running a Browser.
- Ram – 1GB
- An Active Internet Connection

Software requirements-

Given Below are the minimum Software Requirements-

- Operating System- Windows 7/ 8 / 8.1 / 10 / 11 or MacOS or Linux.
- A modern Browser.
- Coding Language – Python.
- Backend – MySQL.
- Development Environment – replit.com.

Contribution of our Project in Real Life-

How our project will be useful-

- Our project will enable users to book tickets conveniently from their homes with any device with an internet connection in it.
- It will help users save their time and make the train reservation process much simpler and smoother.
- It will allow a user to perform multiple actions like, booking a ticket, cancelling a ticket, viewing information about their trips, etc.

What institutions can use our project-

Government intuitions like the IRCTC can take use of our project and use the application to its full extent, hence achieve the project's objective of simplifying the train reservation.

How much resources are being saved with it-

- As the Application is fully autonomous it will help to reduce manpower previously used.
- It is easy to maintain and easy to adopt system and on top of that it is cost effective.
- Admin panel will make it easy for the organization to make any changes regarding the stations, trains instantaneously over the whole server over the country with just a few clicks.

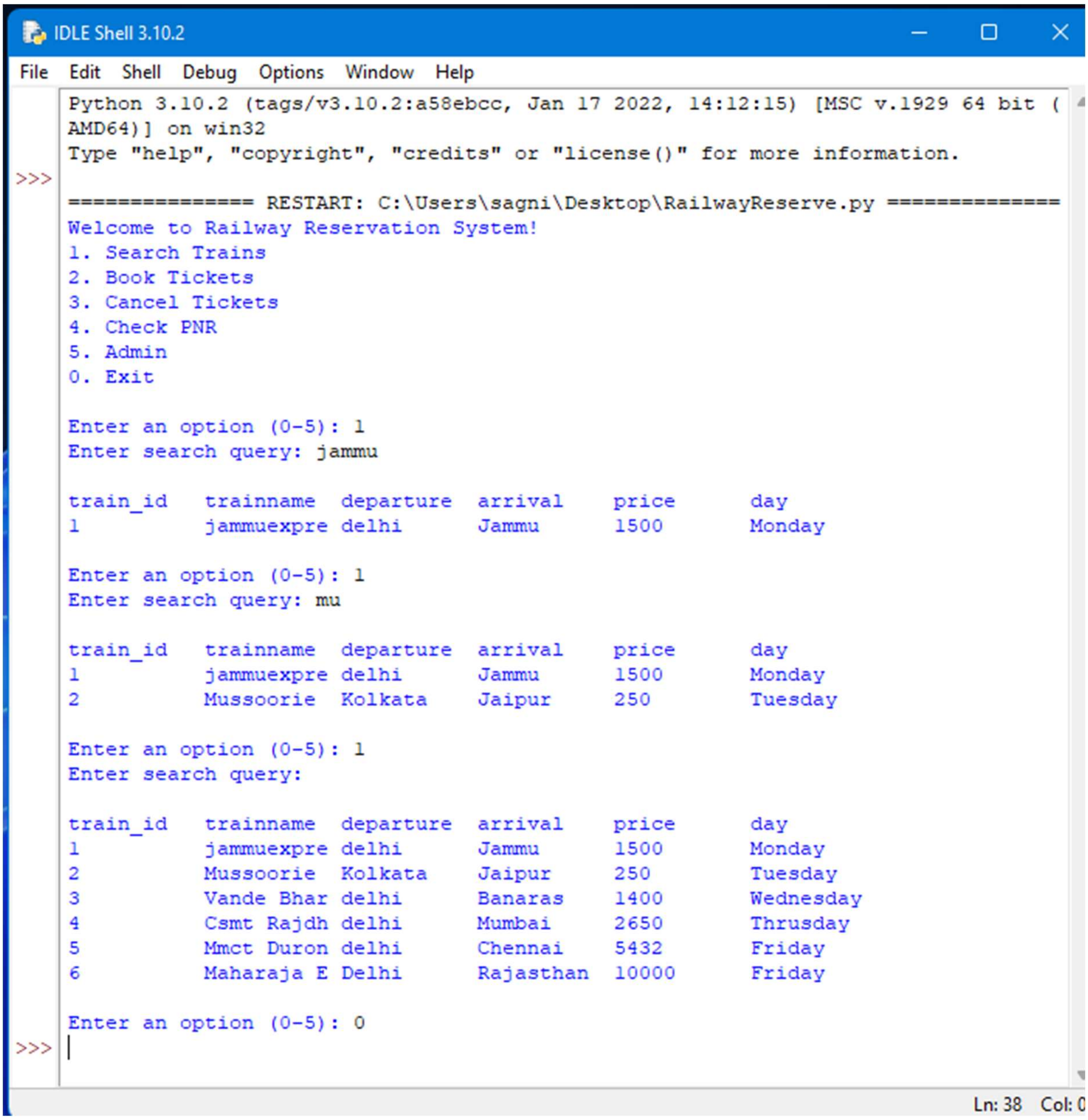
Python CODE

```
import mysql.connector as sql
conn = sql.connect(host="sql6.freesqldatabase.com",
user="sql6470083", passwd="G4HqJ3xy9V",
database="sql6470083")
cursor = conn.cursor()

#searches for trainnames that match a part of query
def searchTrain():
  x = input("Enter search query: ")
  cursor.execute('SELECT * FROM Trains WHERE trainname LIKE
"%{}%"'.format(x))
  printTable()

#creates new record in Tickets table
def bookTicket():
  name = input("Enter Traveller's name: ")
  train_id = input("Enter Train ID: ")
  date = input("Enter date of travel (YYYY-MM-DD): ")
  cursor.execute('INSERT INTO Tickets(name, train_id, date)
VALUES("{}", {}, "{}")'.format(name, train_id, date))
  conn.commit()
  print("Ticket booked!")

#deletes record in Tickets table
def cancelTicket():
  pnr = int(input("Enter PNR: "))
  cursor.execute('DELETE FROM Tickets WHERE
pnr={}'.format(pnr))
  conn.commit()
  print("Ticket cancelled!")

#outputs PNR from details
def checkPNR():
  name = input("Enter Traveller's name: ")
  train_id = input("Enter Train ID: ")
  date = input("Enter date of travel (YYYY-MM-DD): ")
  cursor.execute('SELECT * FROM Tickets WHERE name="{}" AND
train_id={} AND date="{}"'.format(name, train_id, date))
  printTable()
```

```
def admin():
  #authenticate admin credentials
  user = input("Enter username: ")
  passwd = input("Enter password: ")
  if (user, passwd) != ('admin', '12345678'):
    print("Incorrect username or password!")
    return
  print()

  #admin menu
  print("Administrator settings")
  print("1. Add Train")
  print("2. Change Train Stations")
  print("3. Delete Train")
  print("0. Logout")
  print()
  while True:
    x = int(input("Enter an option (0-3): "))

    #add a new train
    if x == 1:
      trainname = input("Enter trainname: ")
      departure = input("Enter departure: ")
      arrival = input("Enter arrival: ")
      price = int(input("Enter price: "))
      day = input("Enter day of the week: ")
      cursor.execute('INSERT INTO Trains(trainname,
departure, arrival, price, day) VALUES("{}", "{}", "{}",
{}, "{}")'.format(trainname, departure, arrival, price,
day))
      conn.commit()
      print("Success!")

    #change destination and arrival
    elif x == 2:
      train_id = int(input("Enter train_id: "))
      departure = input("Enter new departure: ")
      arrival = input("Enter new arrival: ")
      cursor.execute('UPDATE Trains SET departure="{}",
arrival="{}" WHERE train_id={}'.format(departure, arrival,
train_id))
      conn.commit()
      print("Success!")

    #delete a train
    elif x == 3:
      train_id = int(input("Enter train_id: "))
      cursor.execute('DELETE FROM Trains WHERE
train_id={}'.format(train_id))
      print("Success!")

    else:
      print("Logged out")
      return
    print()

#prints the result table with a fixed width
def printTable():
  table = cursor.fetchall()
  if cursor.rowcount == 0:
    print("No search results")
    return

  print()
  for i in cursor.column_names:
    print(i, end=' '*(11-len(i)))
  print()
  for row in table:
    for i in row:
      print(i, end=' '*(11-len(str(i))))
    print()

#menu
print("Welcome to Railway Reservation System!")
print("1. Search Trains")
print("2. Book Tickets")
print("3. Cancel Tickets")
print("4. Check PNR")
print("5. Admin")
print("0. Exit")
print()
while True:
  x = int(input("Enter an option (0-5): "))
  if x == 1:
    searchTrain()
  elif x == 2:
    bookTicket()
  elif x == 3:
    cancelTicket()
  elif x == 4:
    checkPNR()
  elif x == 5:
    admin()
  else:
    break
  print()
conn.close()
```

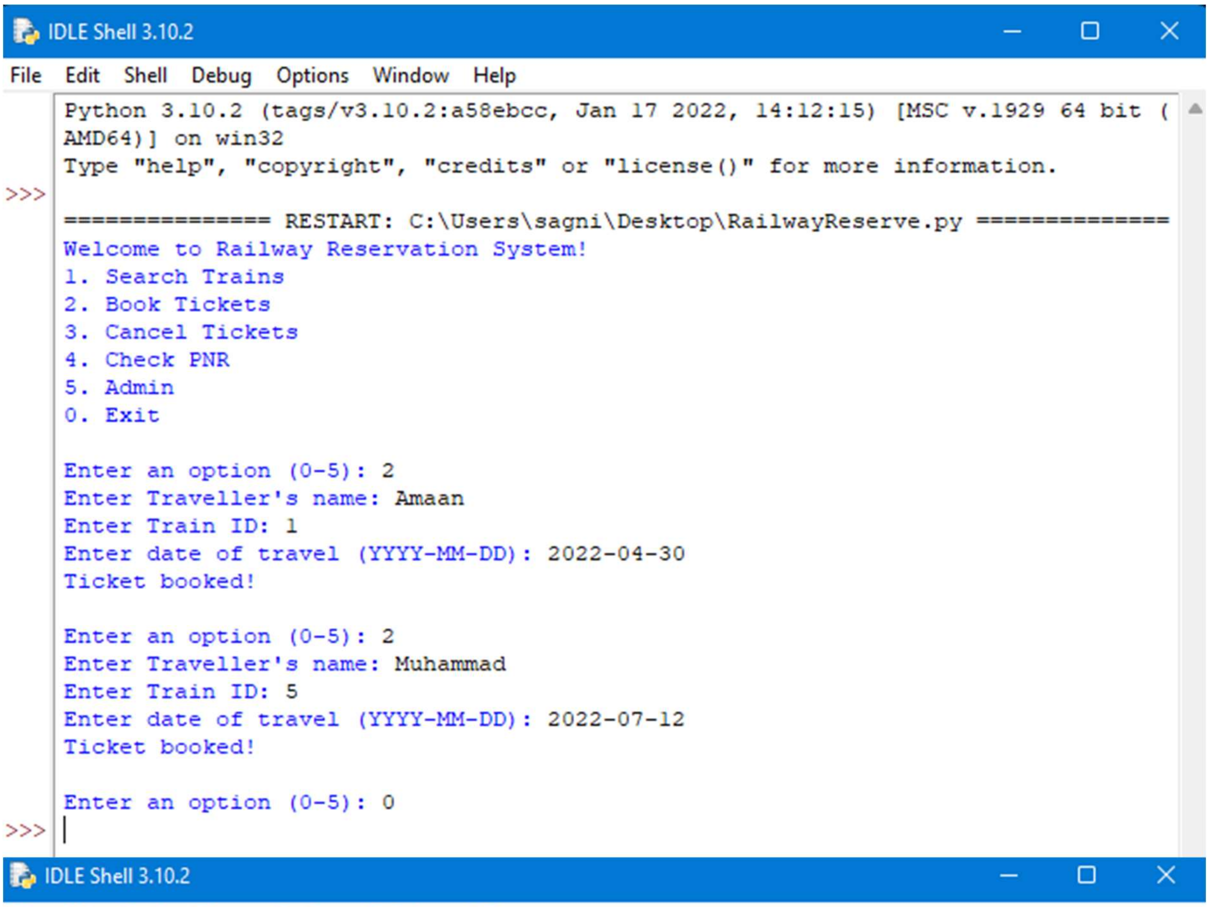
OUTPUT

Search Trains

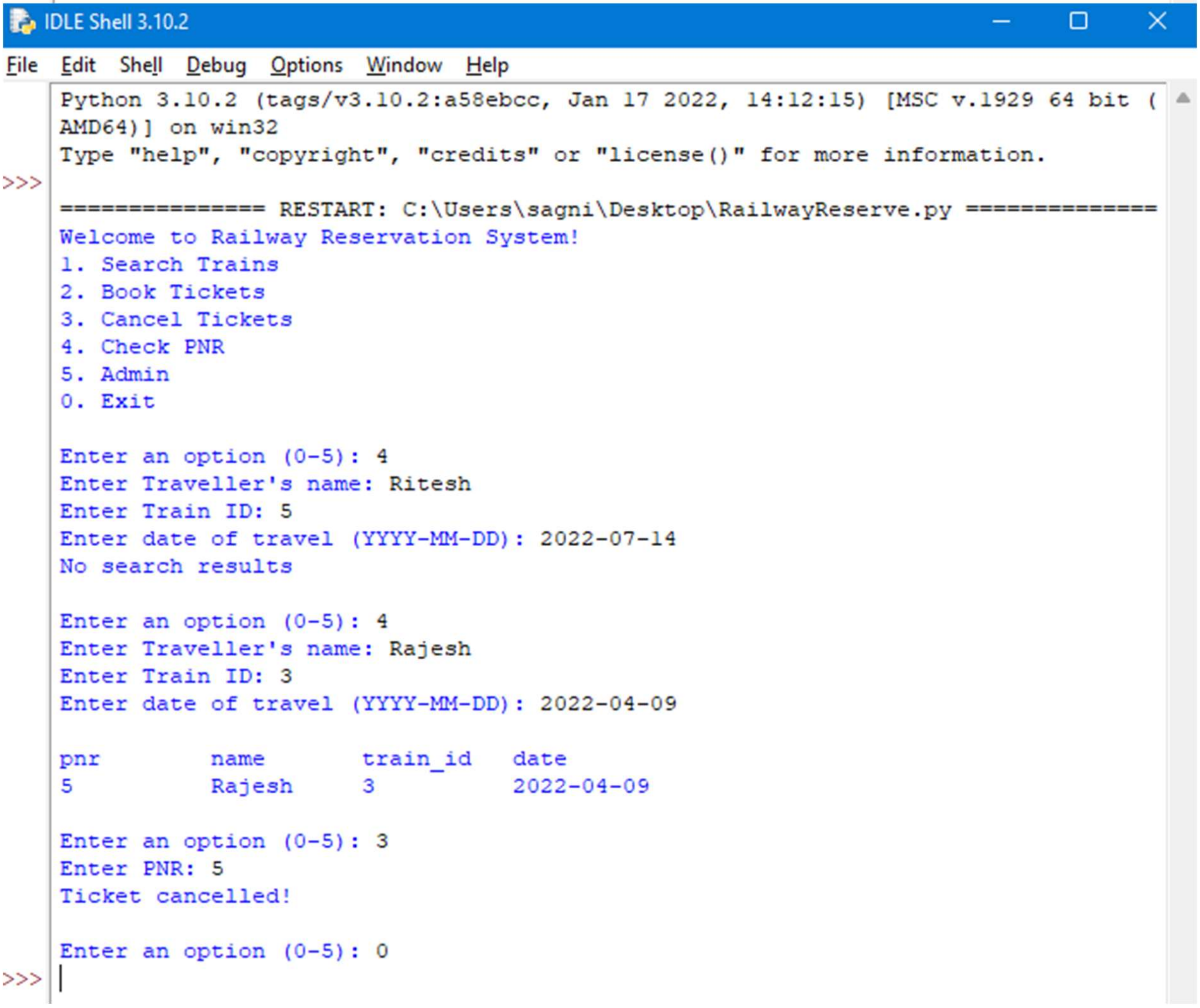


```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\sagni\Desktop\RailwayReserve.py ==============
Welcome to Railway Reservation System!
1. Search Trains
2. Book Tickets
3. Cancel Tickets
4. Check PNR
5. Admin
0. Exit

Enter an option (0-5): 1
Enter search query: jammu

train_id   trainname  departure  arrival    price      day
1          jammuexpre delhi      Jammu      1500       Monday

Enter an option (0-5): 1
Enter search query: mu

train_id   trainname  departure  arrival    price      day
1          jammuexpre delhi      Jammu      1500       Monday
2          Mussoorie  Kolkata    Jaipur     250        Tuesday

Enter an option (0-5): 1
Enter search query:

train_id   trainname  departure  arrival    price      day
1          jammuexpre delhi      Jammu      1500       Monday
2          Mussoorie  Kolkata    Jaipur     250        Tuesday
3          Vande Bhar delhi      Banaras    1400       Wednesday
4          Csmt Rajdh delhi      Mumbai     2650       Thrusday
5          Mmct Duron delhi      Chennai    5432       Friday
6          Maharaja E Delhi      Rajasthan  10000      Friday

Enter an option (0-5): 0
>>>
```

BOOK AND CANCEL TICKETS

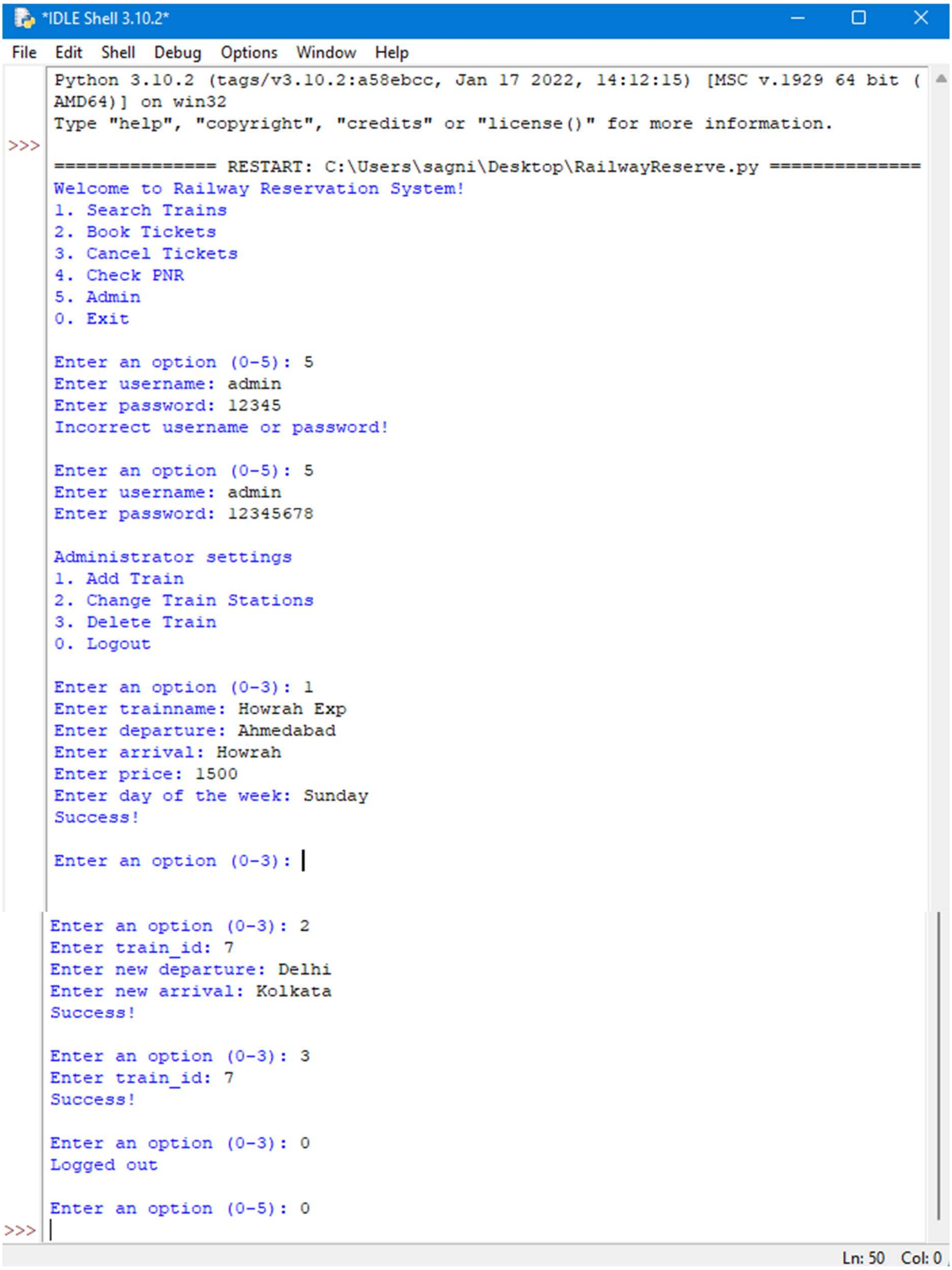


```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\sagni\Desktop\RailwayReserve.py ==============
Welcome to Railway Reservation System!
1. Search Trains
2. Book Tickets
3. Cancel Tickets
4. Check PNR
5. Admin
0. Exit

Enter an option (0-5): 2
Enter Traveller's name: Amaan
Enter Train ID: 1
Enter date of travel (YYYY-MM-DD): 2022-04-30
Ticket booked!

Enter an option (0-5): 2
Enter Traveller's name: Muhammad
Enter Train ID: 5
Enter date of travel (YYYY-MM-DD): 2022-07-12
Ticket booked!

Enter an option (0-5): 0
>>> 
```



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\sagni\Desktop\RailwayReserve.py ==============
Welcome to Railway Reservation System!
1. Search Trains
2. Book Tickets
3. Cancel Tickets
4. Check PNR
5. Admin
0. Exit

Enter an option (0-5): 4
Enter Traveller's name: Ritesh
Enter Train ID: 5
Enter date of travel (YYYY-MM-DD): 2022-07-14
No search results

Enter an option (0-5): 4
Enter Traveller's name: Rajesh
Enter Train ID: 3
Enter date of travel (YYYY-MM-DD): 2022-04-09

pnr        name       train_id    date
5          Rajesh     3           2022-04-09

Enter an option (0-5): 3
Enter PNR: 5
Ticket cancelled!

Enter an option (0-5): 0
>>> 
```

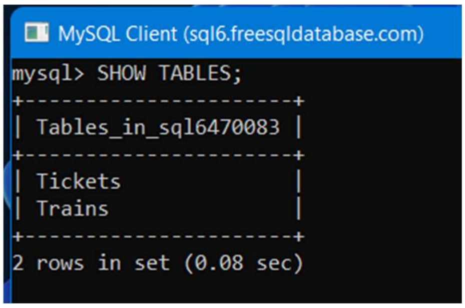
ADMINISTRATOR SETTINGS



```
Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\sagni\Desktop\RailwayReserve.py ==============
Welcome to Railway Reservation System!
1. Search Trains
2. Book Tickets
3. Cancel Tickets
4. Check PNR
5. Admin
0. Exit

Enter an option (0-5): 5
Enter username: admin
Enter password: 12345
Incorrect username or password!

Enter an option (0-5): 5
Enter username: admin
Enter password: 12345678

Administrator settings
1. Add Train
2. Change Train Stations
3. Delete Train
0. Logout

Enter an option (0-3): 1
Enter trainname: Howrah Exp
Enter departure: Ahmedabad
Enter arrival: Howrah
Enter price: 1500
Enter day of the week: Sunday
Success!

Enter an option (0-3): 

Enter an option (0-3): 2
Enter train_id: 7
Enter new departure: Delhi
Enter new arrival: Kolkata
Success!

Enter an option (0-3): 3
Enter train_id: 7
Success!

Enter an option (0-3): 0
Logged out

Enter an option (0-5): 0
>>>
```

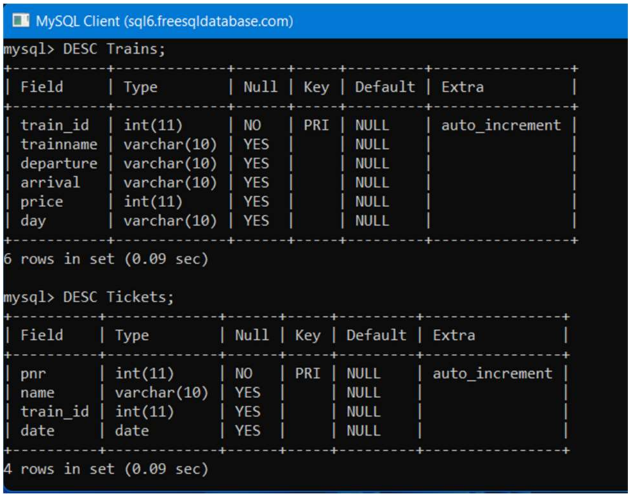
MySQL Tables

Tables



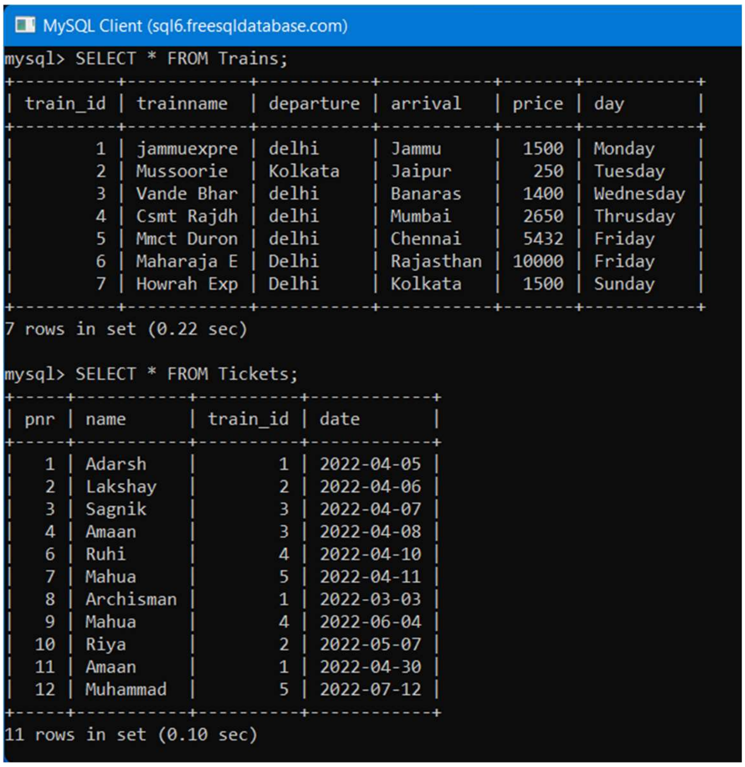
```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_sql6470083 |
+----------------------+
| Tickets              |
| Trains               |
+----------------------+
2 rows in set (0.08 sec)
```

Table Descriptions



```
mysql> DESC Trains;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| train_id  | int(11)     | NO   | PRI | NULL    | auto_increment |
| trainname | varchar(10) | YES  |     | NULL    |                |
| departure | varchar(10) | YES  |     | NULL    |                |
| arrival   | varchar(10) | YES  |     | NULL    |                |
| price     | int(11)     | YES  |     | NULL    |                |
| day       | varchar(10) | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
6 rows in set (0.09 sec)

mysql> DESC Tickets;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| pnr      | int(11)     | NO   | PRI | NULL    | auto_increment |
| name     | varchar(10) | YES  |     | NULL    |                |
| train_id | int(11)     | YES  |     | NULL    |                |
| date     | date        | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
4 rows in set (0.09 sec)
```

Table Contents



```
mysql> SELECT * FROM Trains;
+----------+-----------+-----------+-----------+-------+-----------+
| train_id | trainname | departure | arrival   | price | day       |
+----------+-----------+-----------+-----------+-------+-----------+
|        1 | jammuexpre | delhi    | Jammu     |  1500 | Monday    |
|        2 | Mussoorie | Kolkata   | Jaipur    |   250 | Tuesday   |
|        3 | Vande Bhar | delhi    | Banaras   |  1400 | Wednesday |
|        4 | Csmt Rajdh | delhi    | Mumbai    |  2650 | Thrusday  |
|        5 | Mmct Duron | delhi    | Chennai   |  5432 | Friday    |
|        6 | Maharaja E | Delhi    | Rajasthan | 10000 | Friday    |
|        7 | Howrah Exp | Delhi    | Kolkata   |  1500 | Sunday    |
+----------+-----------+-----------+-----------+-------+-----------+
7 rows in set (0.22 sec)

mysql> SELECT * FROM Tickets;
+-----+-----------+----------+------------+
| pnr | name      | train_id | date       |
+-----+-----------+----------+------------+
|   1 | Adarsh    |        1 | 2022-04-05 |
|   2 | Lakshay   |        2 | 2022-04-06 |
|   3 | Sagnik    |        3 | 2022-04-07 |
|   4 | Amaan     |        3 | 2022-04-08 |
|   6 | Ruhi      |        4 | 2022-04-10 |
|   7 | Mahua     |        5 | 2022-04-11 |
|   8 | Archisman |        1 | 2022-03-03 |
|   9 | Mahua     |        4 | 2022-06-04 |
|  10 | Riya      |        2 | 2022-05-07 |
|  11 | Amaan     |        1 | 2022-04-30 |
|  12 | Muhammad  |        5 | 2022-07-12 |
+-----+-----------+----------+------------+
11 rows in set (0.10 sec)
```