```
# RAILWAY MANAGEMENT SYSTEM

# Importing Modules
import mysql.connector as sql
from random import randint

# Establishment of connection to MySQL Server
print("Enter the details of your MySQL Server:")
x = input("Hostname: ")
y = input("User: ")
z = input("Password: ")
con = sql.connect(host=x,
                  user=y,
                  password=z)
con.autocommit = True
cur = con.cursor()

# Creation of Database and subsequent Tables
cur.execute("CREATE DATABASE IRCTC;")
cur.execute("USE IRCTC;")
s = "CREATE TABLE accounts" \
    "(id   int primary key," \
    "pass  varchar(16)," \
    "name  varchar(100)," \
    "sex   char(1)," \
    "age   varchar(3)," \
    "dob   date," \
    "ph_no char(10));"
cur.execute(s)
s = "CREATE TABLE tickets" \
    "(id    int," \
    "PNR    int," \
    "train  varchar(25)," \
    "doj    date," \
```

```
    "tfr    varchar(100)," \
    "tto    varchar(100));"
cur.execute(s)


# Login Menu
def login_menu():
    print("WELCOME TO THE IRCTC PORTAL")
    print("1. Create New Account \n"
          "2. Log In \n"
          "3. Exit")
    opt = int(input("Enter your choice: "))
    if opt == 1:
        create_acc()
    elif opt == 2:
        login()
    else:
        e = input("Exit the portal? (Y/N) ")
        if e in "Nn":
            login_menu()


# Account Creation
def create_acc():
    print("Enter the details to create your account:")
    i = randint(1000, 10000)
    print(f"Your generated ID is: {i}")
    p = input("Enter your password: ")
    n = input("Enter your name: ")
    sex = input("Enter your gender (M/F/O): ")
    age = input("Enter your age: ")
    dob = input("Enter your date of birth (YYYY-MM-DD):
")
    ph = input("Enter your contact number: ")
```

```
    s1 = "INSERT INTO accounts VALUES" \
         f"({i}, '{p}', '{n}', '{sex.upper()}', " \
         f"{age}, '{dob}', '{ph}');"
    cur.execute(s1)
    print("Now you may log in with your newly created
account!")
    login()


# Log in to Account
def login():
    global a
    try:
        a = int(input("Enter your ID: "))
        b = input("Enter your password: ")
        s2 = f"SELECT name FROM accounts " \
             f"WHERE id = {a} AND pass = '{b}';"
        cur.execute(s2)
        j = cur.fetchone()
        print(f"Welcome back {j[0]}!")
        main_menu()
    except:
        print("Your account was not found!")
        print("You can: \n"
              "1. Try logging in again \n"
              "2. Create a new account")
        ch = input("Enter your choice: ")
        if ch == "1":
            login()
        elif ch == "2":
            create_acc()
        else:
            print("Invalid choice!")
            x1 = input("Exit the portal? (Y/N) ")
```

```
            if x1 in "Nn":
                login_menu()


# Main Menu
def main_menu():
    print("What would you like to do today? \n"
          "1. Purchase a Ticket \n"
          "2. Check Ticket Status \n"
          "3. Request a refund \n"
          "4. Account Settings \n"
          "5. Logout \n"
          "6. Exit")
    ch1 = int(input("Enter your choice: "))
    if ch1 == 1:
        buy_ticket()
    elif ch1 == 2:
        show_ticket()
    elif ch1 == 3:
        cancel_ticket()
    elif ch1 == 4:
        account()
    elif ch1 == 5:
        login_menu()
    else:
        exit_prompt()


# Exit Prompt
def exit_prompt():
    x2 = input("Would you like to exit? (Y/N) ")
    if x2.upper() == "N":
        main_menu()
```

```
# Back to Main Menu
def back_to_main_menu():
    x3 = input("Return to the Main Menu? (Y/N) ")
    if x3.upper() == "Y":
        print("Returning to Main Menu...")
        main_menu()


# Ticket Creation
def buy_ticket():
    print("Enter details for your journey: ")
    i = a
    pnr = randint(100000, 1000000)
    print(f"Your PNR is {pnr}")
    train = input("Enter the name of the train: ")
    doj = input("Enter the date of your journey
(YYYY-MM-DD): ")
    fr = input("Enter the Departing Station: ")
    to = input("Enter the Destination Station: ")
    s4 = f"INSERT INTO tickets VALUES" \
         f"({i}, {pnr}, '{train}', " \
         f"'{doj}', '{fr}', '{to}');"
    cur.execute(s4)
    back_to_main_menu()


# Ticket Checking
def show_ticket():
    try:
        pnr = int(input("Enter your PNR: "))
        s5 = f"SELECT * FROM tickets " \
             f"WHERE pnr = {pnr}"
        cur.execute(s5)
```

```
        j = cur.fetchone()
        if j[0] == a:
            print(f"Train: {j[2]} \n"
                  f"Date of Journey: {j[3]} \n"
                  f"From: {j[4]} \n"
                  f"To: {j[5]}")
            back_to_main_menu()
        else:
            print("Unauthorized! \n"
                  "Your ID does not match the PNR of
ticket.")
            back_to_main_menu()
    except:
        ticket_not_found()


# Ask for a refund
def cancel_ticket():
    try:
        pnr = int(input("Enter the PNR number of the
ticket: "))
        s2 = f"SELECT id, pnr, train " \
             f"FROM tickets " \
             f"WHERE pnr = {pnr}"
        cur.execute(s2)
        j = cur.fetchone()
        if j[0] == a:
            print(f"PNR: {j[1]} \n"
                  f"Train: {j[2]}")
            x4 = input("Do you really want to cancel
this ticket? (Y/N) ")
            if x4.upper() == "Y":
                s3 = f"DELETE FROM tickets " \
                     f"WHERE pnr = {pnr};"
```

```
                cur.execute(s3)
                print("You will be refunded shortly!")
                back_to_main_menu()
            else:
                back_to_main_menu()
        else:
            print("Unauthorized! \n"
                  "Your ID does not match "
                  "the PNR of ticket.")
            back_to_main_menu()
    except:
        ticket_not_found()


# If ticket is not found
def ticket_not_found():
    print("Ticket not found!")
    print("You can: \n"
          "1. Try entering your PNR number again \n"
          "2. Purchase a ticket \n"
          "3. Return to Main Menu \n"
          "4. Exit")
    ch = int(input("Enter your choice: "))
    if ch == 1:
        show_ticket()
    elif ch == 2:
        buy_ticket()
    elif ch == 3:
        print("Returning to Main Menu...")
        main_menu()
    else:
        exit_prompt()
```

```
# Account settings
def account():
    print("Do you want to: \n"
          "1. Show Account details \n"
          "2. Delete Account")
    ch = int(input("Enter your choice: "))
    if ch == 1:
        s4 = f"SELECT * FROM accounts WHERE id = {a}"
        cur.execute(s4)
        j = cur.fetchone()
        print(f"ID: {j[0]} \n"
              f"Name: {j[2]} \n"
              f"Gender: {j[3]} \n"
              f"Age: {j[4]} \n"
              f"DOB: {j[5]} \n"
              f"Phone Number: {j[6]}")
        back_to_main_menu()
    elif ch == 2:
        x6 = input("Do you want to request for refund(s)
for your ticket(s) too? (Y/N) ")
        if x6.upper() == "Y":
            s5 = f"DELETE FROM tickets WHERE id = {a}"
            cur.execute(s5)
            print("You will be refunded shortly!")
        s6 = f"DELETE FROM ACCOUNTS " \
             f"WHERE id = {a}"
        cur.execute(s6)
        print("Account Successfully Deleted!")
        login_menu()
    else:
        back_to_main_menu()


# Calling the first function, hence starting the program
```

```
if __name__ == "__main__":
  login_menu()
```