# **FLIGHT RESERVATION SYSTEM**

**#Name**  
name = input("kindly enter your name:")  
  
**#phone Number**  
for i in range(0, 11):  
 try:  
 phone\_number = int(input("Enter your 10-digit phone number: "))  
 if len(str(phone\_number)) == 10:  
 print("✅ Phone number accepted.")  
 break  
 else:  
 print(" Phone number must be 10 digits.")  
 except ValueError:  
 print("Invalid input. Please enter only digits.")  
else:  
 print("🙁 Too many failed attempts. Try again later.")  
  
  
class Flight:  
 def \_\_init\_\_(self, flight\_number, airline, origin, destination, seats\_available):  
 self.flight\_number = flight\_number  
 self.airline = airline  
 self.origin = origin  
 self.destination = destination  
 self.seats\_available = seats\_available # list of seat numbers  
  
 def book\_seat(self, seat\_number):  
 if seat\_number in self.seats\_available:  
 # After booking the seat, remove the seat number from available seats  
 self.seats\_available.remove(seat\_number)  
 print(f"Seat {seat\_number} booked successfully!")  
 else:  
 print(f"Seat {seat\_number} is not available. Please choose another seat.")  
  
 def get\_info(self):  
 return f"{self.flight\_number} | {self.airline} | {self.origin.capitalize()} → {self.destination.capitalize()} | Seats Available: {self.seats\_available}"  
  
  
**# input for origin and destination**  
countries = [country.lower() for country in [  
 "Afghanistan", "Albania", "Algeria", "Andorra", "Angola", "Antigua and Barbuda", "Argentina",  
 "Armenia", "Australia", "Austria", "Azerbaijan", "Bahamas", "Bahrain", "Bangladesh", "Barbados",  
 "Belarus", "Belgium", "Belize", "Benin", "Bhutan", "Bolivia", "Bosnia and Herzegovina", "Botswana",  
 "Brazil", "Brunei", "Bulgaria", "Burkina Faso", "Burundi", "Cabo Verde", "Cambodia", "Cameroon",  
 "Canada", "Central African Republic", "Chad", "Chile", "China", "Colombia", "Comoros",  
 "Congo (Congo-Brazzaville)", "Costa Rica", "Croatia", "Cuba", "Cyprus", "Czech Republic",  
 "Democratic Republic of the Congo", "Denmark", "Djibouti", "Dominica", "Dominican Republic",  
 "Ecuador", "Egypt", "El Salvador", "Equatorial Guinea", "Eritrea", "Estonia", "Eswatini",  
 "Ethiopia", "Fiji", "Finland", "France", "Gabon", "Gambia", "Georgia", "Germany", "Ghana",  
 "Greece", "Grenada", "Guatemala", "Guinea", "Guinea-Bissau", "Guyana", "Haiti", "Honduras",  
 "Hungary", "Iceland", "India", "Indonesia", "Iran", "Iraq", "Ireland", "Israel", "Italy",  
 "Ivory Coast", "Jamaica", "Japan", "Jordan", "Kazakhstan", "Kenya", "Kiribati", "Kuwait",  
 "Kyrgyzstan", "Laos", "Latvia", "Lebanon", "Lesotho", "Liberia", "Libya", "Liechtenstein",  
 "Lithuania", "Luxembourg", "Madagascar", "Malawi", "Malaysia", "Maldives", "Mali", "Malta",  
 "Marshall Islands", "Mauritania", "Mauritius", "Mexico", "Micronesia", "Moldova", "Monaco",  
 "Mongolia", "Montenegro", "Morocco", "Mozambique", "Myanmar", "Namibia", "Nauru", "Nepal",  
 "Netherlands", "New Zealand", "Nicaragua", "Niger", "Nigeria", "North Korea", "North Macedonia",  
 "Norway", "Oman", "Pakistan", "Palau", "Palestine", "Panama", "Papua New Guinea", "Paraguay",  
 "Peru", "Philippines", "Poland", "Portugal", "Qatar", "Romania", "Russia", "Rwanda",  
 "Saint Kitts and Nevis", "Saint Lucia", "Saint Vincent and the Grenadines", "Samoa",  
 "San Marino", "Sao Tome and Principe", "Saudi Arabia", "Senegal", "Serbia", "Seychelles",  
 "Sierra Leone", "Singapore", "Slovakia", "Slovenia", "Solomon Islands", "Somalia",  
 "South Africa", "South Korea", "South Sudan", "Spain", "Sri Lanka", "Sudan", "Suriname",  
 "Sweden", "Switzerland", "Syria", "Taiwan", "Tajikistan", "Tanzania", "Thailand",  
 "Timor-Leste", "Togo", "Tonga", "Trinidad and Tobago", "Tunisia", "Turkey", "Turkmenistan",  
 "Tuvalu", "Uganda", "Ukraine", "United Arab Emirates", "United Kingdom", "United States",  
 "Uruguay", "Uzbekistan", "Vanuatu", "Vatican City", "Venezuela", "Vietnam", "Yemen",  
 "Zambia", "Zimbabwe"  
]]  
  
while True:  
 origin = input("Enter the origin country: ").lower()  
 if origin in countries:  
 break  
 else:  
 print("Please enter a valid country name.")  
 break  
  
while True:  
 destination = input("Enter the destination country: ").lower()  
 if destination in countries:  
 if destination != origin:  
 break  
 else:  
 print("Origin and destination cannot be the same.")  
 else:  
 print("Please enter a valid country name.")  
  
**# To Create a list of flights based on user inputs for origin and destination**flights = [  
 Flight("AI202", "Air India", origin, destination, [1, 2, 3, 4, 5]),  
 Flight("AI303", "Air India", origin, destination, [1, 2]),  
 Flight("EK404", "Emirates", origin, destination, [1, 2, 3]),  
 Flight("EK505", "Emirates", origin, destination, [1, 2]),  
 Flight("BA606", "British Airways", origin, destination, [1, 2, 3, 4])  
]  
  
# To Show available flights  
#enumerates means its a built in function,add a index to an iterable list,returns enumerate object  
print("\nAvailable Flights:")  
for i, flight in enumerate(flights, start=1):  
 print(f"{i}. {flight.get\_info()}")  
  
**# To selects a flight**  
while True:  
 try:  
 choice = int(input("\nEnter the index number of the flight you want to book: "))  
 if 1 <= choice <= len(flights):  
 selected\_flight = flights[choice - 1]  
 print(f"\nYou selected: {selected\_flight.get\_info()}")  
  
 # Step 5: Show available seats  
 print(f"Available seats: {selected\_flight.seats\_available}")  
  
 # Step 6: User selects seat number  
 while True:  
 seat\_choice = int(input("Enter the seat number you want to book: "))  
 if seat\_choice in selected\_flight.seats\_available:  
 selected\_flight.book\_seat(seat\_choice)  
 break # Exit the loop after booking the seat  
 else:  
 print(f"Seat {seat\_choice} is not available. Please choose another seat.")  
  
 break # Exit outer loop after booking the seat  
 else:  
 print("\nInvalid flight selection. Please choose a valid number.")  
 except ValueError:  
 print("\nPlease enter a valid number.")  
  
  
**# Food class**  
class Food:  
 def \_\_init\_\_(self):  
 self.selected\_snack = ""  
 self.selected\_drink = ""  
  
 def snacks(self):  
 snack\_list = ["energy bite cookies", "lays", "kitkat", "pasta", "wheat cookies"]  
 print("\nAvailable snacks:")  
 for i, snack in enumerate(snack\_list, start=1):  
 print(f"{i}. {snack}")  
 try:  
 snack\_choice = input("Enter the snack index number (or press Enter to skip): ")  
 if snack\_choice == "":  
 print("You skipped snack selection.")  
 else:  
 snack\_choice = int(snack\_choice)  
 if 1 <= snack\_choice <= len(snack\_list):  
 self.selected\_snack = snack\_list[snack\_choice - 1]  
 print(f"You have selected snack: {self.selected\_snack}")  
 else:  
 print("Invalid snack index.")  
 except ValueError:  
 print("Invalid input. Snack choice must be a number.")  
  
 def drinks(self):  
 drink\_list = ["Watermelon Mocktail", "Lemon Mint", "Filter Coffee", "Green Tea"]  
 print("\nAvailable drinks:")  
 for i, drink in enumerate(drink\_list, start=1):  
 print(f"{i}. {drink}")  
 try:  
 drink\_choice = input("Enter the drink index number (or press Enter to skip): ")  
 if drink\_choice == "":  
 print("You skipped drink selection.")  
 else:  
 drink\_choice = int(drink\_choice)  
 if 1 <= drink\_choice <= len(drink\_list):  
 self.selected\_drink = drink\_list[drink\_choice - 1]  
 print(f"You selected drink: {self.selected\_drink}")  
 else:  
 print("Invalid drink index.")  
 except ValueError:  
 print("Invalid input. Drink choice must be a number.")  
  
  
obj = Food()  
obj.drinks()  
obj.snacks()  
  
**# displaying details**  
print(f"\nHi {name}, you have booked a seat on {selected\_flight.airline} "  
 f"from {selected\_flight.origin} to {selected\_flight.destination}.")  
  
print(f"drinks --> {obj.selected\_drink if obj.selected\_drink else 'None selected'}")  
print(f"snacks --> {obj.selected\_snack if obj.selected\_snack else 'None selected'}")  
  
print(f"Thank You {name}")