

## main.py

```
1 import csv
2
3 class UserManagementSystem:
4     def __init__(self, csv_filename='users.csv'):
5         self.csv_filename = csv_filename
6         self.fields = ['Name', 'Email', 'Age']
7         self.users = []
8
9     def save_users(self):
10        with open(self.csv_filename, 'w', newline='') as file:
11            writer = csv.DictWriter(file, fieldnames=self.fields)
12            writer.writeheader()
13            writer.writerows(self.users)
14
15    def load_users(self):
16        try:
17            with open(self.csv_filename, 'r') as file:
18                reader = csv.DictReader(file)
19                self.users = [row for row in reader]
20        except FileNotFoundError:
21            self.users = []
22
23    def create_user(self):
24        name = input("Enter user name: ")
25        email = input("Enter user email: ")
26        age = input("Enter user age: ")
27
28        user = {
29            'Name': name,
30            'Email': email,
31            'Age': age
32        }
33
34        self.users.append(user)
35        print("User created successfully!")
36        self.save_users()
37
38    def list_users(self):
39        print("\nList of Users:")
```

```
40     for user in self.users:
41         print(f"Name: {user['Name']}, Email: {user['Email']}, Age: {user['Age']}")
42     print()
43
44     def edit_user(self):
45         self.list_users()
46         try:
47             user_index = int(input("Enter the index of the user you want to edit: ")) - 1
48             if 0 <= user_index < len(self.users):
49                 name = input("Enter new name: ")
50                 email = input("Enter new email: ")
51                 age = input("Enter new age: ")
52
53                 self.users[user_index]['Name'] = name
54                 self.users[user_index]['Email'] = email
55                 self.users[user_index]['Age'] = age
56
57                 print("User information updated successfully!")
58                 self.save_users()
59             else:
60                 print("Invalid user index.")
61         except ValueError:
62             print("Invalid input. Please enter a valid number.")
63
64     def delete_user(self):
65         self.list_users()
66         try:
67             user_index = int(input("Enter the index of the user you want to delete: ")) - 1
68             if 0 <= user_index < len(self.users):
69                 del self.users[user_index]
70                 print("User deleted successfully!")
71                 self.save_users()
72             else:
73                 print("Invalid user index.")
74         except ValueError:
75             print("Invalid input. Please enter a valid number.")
76
77     def run(self):
78         self.load_users()
79
80         while True:
81             print("\nUser Management System Menu:")
```

```
82         print("1. Create User")
83         print("2. List Users")
84         print("3. Edit User")
85         print("4. Delete User")
86         print("5. Exit")
87
88         choice = input("Enter your choice (1-5): ")
89
90         if choice == '1':
91             self.create_user()
92         elif choice == '2':
93             self.list_users()
94         elif choice == '3':
95             self.edit_user()
96         elif choice == '4':
97             self.delete_user()
98         elif choice == '5':
99             print("Exiting User Management System. Goodbye!")
100             break
101         else:
102             print("Invalid choice. Please enter a number between 1 and 5.")
103
104
105 if __name__ == "__main__":
106     user_management_system = UserManagementSystem()
107     user_management_system.run()
```