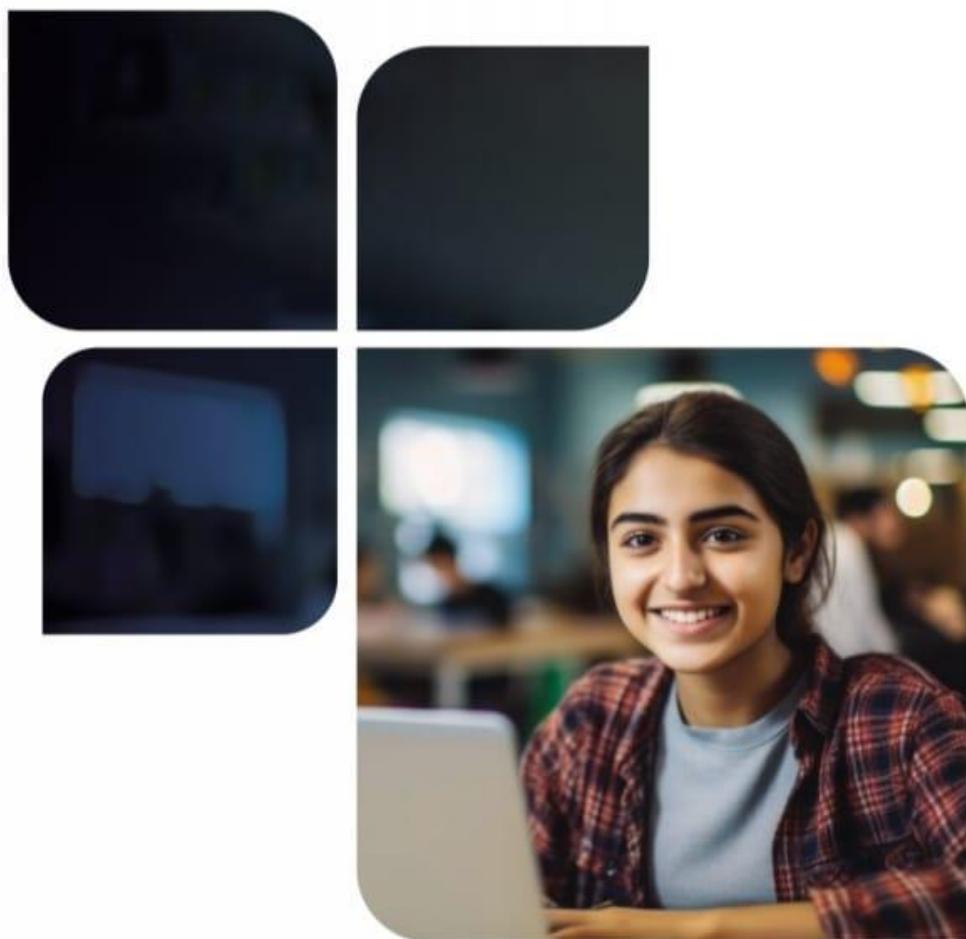




An ISO 9001 & ISO 21001
Certified Organization



The Quest
for your Dream Job
Ends Here!!

www.gqtech.in



**GLOBAL QUEST
TECHNOLOGIES**

PROJECT REPORT

TITLE : MEDICAL CLINIC APPOINTMENT SYSTEM

Submitted By : B. Geethika
Submitted to : Global Quest Technologies
Date : 15 February 2026

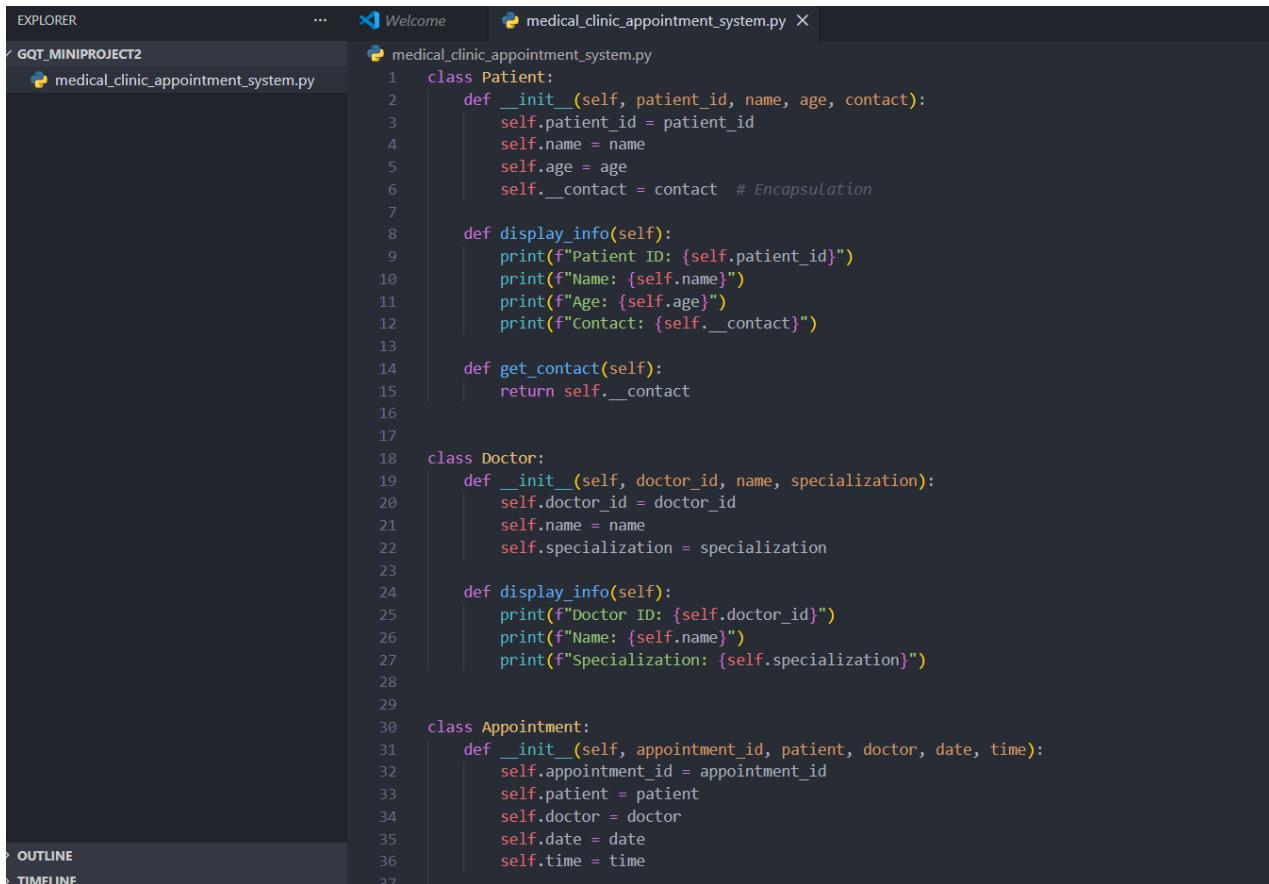
ABSTRACT

The Hotel Reservation System is a console-based Python application designed to automate and simplify hotel management tasks such as room management, guest management, and reservation scheduling. The system is built using Object-Oriented Programming (OOP) concepts, including classes, objects, inheritance, encapsulation, polymorphism, and composition, to ensure modularity, scalability, and maintainability.

The system allows hotel staff to add and manage rooms with details like room type, price, and availability, register guests with personal information securely stored, and schedule reservations while preventing double-booking. Guests' and rooms' information can be easily accessed and displayed using well-structured menus.

This system provides an efficient, interactive, and user-friendly console interface to manage multiple guests and rooms simultaneously, demonstrating real-world applications of OOP principles in software development. It serves as a foundation for more advanced hotel management systems with features like billing, online booking, and reports.

Source Code:-



A screenshot of a code editor showing the Python source code for a medical clinic appointment system. The code defines three classes: Patient, Doctor, and Appointment, each with its own __init__ method and a display_info method that prints out the object's ID, name, and specialization. The Patient class also includes a get_contact method. The code is color-coded, and the editor interface shows the file path 'GQT_MINIPROJECT2\medical_clinic_appointment_system.py' and various navigation tabs.

```
EXPLORER ... 🌐 Welcome 🗃 medical_clinic_appointment_system.py X
GQT_MINIPROJECT2
    medical_clinic_appointment_system.py

medical_clinic_appointment_system.py
1  class Patient:
2      def __init__(self, patient_id, name, age, contact):
3          self.patient_id = patient_id
4          self.name = name
5          self.age = age
6          self.__contact = contact # Encapsulation
7
8      def display_info(self):
9          print(f"Patient ID: {self.patient_id}")
10         print(f"Name: {self.name}")
11         print(f"Age: {self.age}")
12         print(f"Contact: {self.__contact}")
13
14     def get_contact(self):
15         return self.__contact
16
17
18 class Doctor:
19     def __init__(self, doctor_id, name, specialization):
20         self.doctor_id = doctor_id
21         self.name = name
22         self.specialization = specialization
23
24     def display_info(self):
25         print(f"Doctor ID: {self.doctor_id}")
26         print(f"Name: {self.name}")
27         print(f"Specialization: {self.specialization}")
28
29
30 class Appointment:
31     def __init__(self, appointment_id, patient, doctor, date, time):
32         self.appointment_id = appointment_id
33         self.patient = patient
34         self.doctor = doctor
35         self.date = date
36         self.time = time
37
```

The screenshot shows a code editor interface with a dark theme. The main pane displays the `medical_clinic_appointment_system.py` file. The `Appointment` class is defined, containing methods for displaying appointment details and adding patients. The `ClinicSystem` class is also defined, initializing patient, doctor, and appointment dictionaries. A tooltip or chat window is visible on the right side of the interface.

```
... medical_clinic_appointment_system.py ...
30     class Appointment:
31         ...
32         def display_appointment(self):
33             print(f"\nAppointment ID: {self.appointment_id}")
34             print(f"Date: {self.date} | Time: {self.time}")
35             print(f"Patient: {self.patient.name} (ID: {self.patient.patient_id})")
36             print(f"Doctor: {self.doctor.name} ({self.doctor.specialization})")
37             ...
38 
39     class ClinicSystem:
40         def __init__(self):
41             self.patients = {} # patient_id -> Patient
42             self.doctors = {} # doctor_id -> Doctor
43             self.appointments = {} # appointment_id -> Appointment
44 
45         # Add patient
46         def add_patient(self):
47             pid = input("Enter patient ID: ")
48             if pid in self.patients:
49                 print("Patient already exists!")
50                 return
51             name = input("Enter patient name: ")
52             age = int(input("Enter age: "))
53             contact = input("Enter contact number: ")
54             patient = Patient(pid, name, age, contact)
55             self.patients[pid] = patient
56             print(f"Patient {name} added successfully!")
57 
58         # Add doctor
59         def add_doctor(self):
60             did = input("Enter doctor ID: ")
61             if did in self.doctors:
62                 print("Doctor already exists!")
63                 return
64             name = input("Enter doctor name: ")
65             specialization = input("Enter specialization: ")
66             doctor = Doctor(did, name, specialization)
67             self.doctors[did] = doctor
68             print(f"Doctor {name} added successfully!")
69 
70         # Schedule appointment
71         def schedule_appointment(self):
72             if not self.patients or not self.doctors:
73                 print("Please add patients and doctors first!")
74                 return
75             aid = input("Enter appointment ID: ")
76             if aid in self.appointments:
77                 print("Appointment already exists!")
78                 return
79             pid = input("Enter patient ID: ")
80             did = input("Enter doctor ID: ")
81             date = input("Enter appointment date (YYYY-MM-DD): ")
82             time = input("Enter appointment time (HH:MM): ")
83 
84             patient = self.patients.get(pid)
85             doctor = self.doctors.get(did)
86             if not patient or not doctor:
87                 print("Invalid patient or doctor ID!")
88                 return
89 
90             appointment = Appointment(aid, patient, doctor, date, time)
91             self.appointments[aid] = appointment
92             print(f"Appointment scheduled for {patient.name} with Dr. {doctor.name}.")
93 
94         # Display all patients
95         def display_patients(self):
96             if not self.patients:
97                 print("No patients found!")
98             for p in self.patients.values():
99                 p.display_info()
100 
```

The screenshot shows a code editor interface with a dark theme. The main pane displays the `medical_clinic_appointment_system.py` file. The `ClinicSystem` class is defined, with its `add_doctor` method implemented. This method prompts for doctor ID, name, and specialization, creates a `Doctor` object, and adds it to the `doctors` dictionary. A tooltip or chat window is visible on the right side of the interface.

```
... medical_clinic_appointment_system.py ...
45     class ClinicSystem:
46         def __init__(self):
47             self.patients = {} # patient_id -> Patient
48             self.doctors = {} # doctor_id -> Doctor
49             self.appointments = {} # appointment_id -> Appointment
50 
51         # Add patient
52         def add_patient(self):
53             pid = input("Enter patient ID: ")
54             if pid in self.patients:
55                 print("Patient already exists!")
56                 return
57             name = input("Enter patient name: ")
58             age = int(input("Enter age: "))
59             contact = input("Enter contact number: ")
60             patient = Patient(pid, name, age, contact)
61             self.patients[pid] = patient
62             print(f"Patient {name} added successfully!")
63 
64         # Add doctor
65         def add_doctor(self):
66             did = input("Enter doctor ID: ")
67             if did in self.doctors:
68                 print("Doctor already exists!")
69                 return
70             name = input("Enter doctor name: ")
71             specialization = input("Enter specialization: ")
72             doctor = Doctor(did, name, specialization)
73             self.doctors[did] = doctor
74             print(f"Doctor {name} added successfully!")
75 
76         # Schedule appointment
77         def schedule_appointment(self):
78             if not self.patients or not self.doctors:
79                 print("Please add patients and doctors first!")
80                 return
81             aid = input("Enter appointment ID: ")
82             if aid in self.appointments:
83                 print("Appointment already exists!")
84                 return
85             pid = input("Enter patient ID: ")
86             did = input("Enter doctor ID: ")
87             date = input("Enter appointment date (YYYY-MM-DD): ")
88             time = input("Enter appointment time (HH:MM): ")
89 
90             patient = self.patients.get(pid)
91             doctor = self.doctors.get(did)
92             if not patient or not doctor:
93                 print("Invalid patient or doctor ID!")
94                 return
95 
96             appointment = Appointment(aid, patient, doctor, date, time)
97             self.appointments[aid] = appointment
98             print(f"Appointment scheduled for {patient.name} with Dr. {doctor.name}.")
99 
100        # Display all patients
101        def display_patients(self):
102            if not self.patients:
103                print("No patients found!")
104            for p in self.patients.values():
105                p.display_info()
```

EXPLORER ... Welcome medical_clinic_appointment_system.py

SQT_MINIPROJECT2

medical_clinic_appointment_system.py

```
45 class ClinicSystem:
101     def display_patients(self):
102         p.display_info()
103         print("-----")
104
105     # Display all doctors
106     def display_doctors(self):
107         if not self.doctors:
108             print("No doctors found!")
109         for d in self.doctors.values():
110             d.display_info()
111             print("-----")
112
113     # Display all appointments
114     def display_appointments(self):
115         if not self.appointments:
116             print("No appointments scheduled!")
117         for a in self.appointments.values():
118             a.display_appointment()
119             print("-----")
120
121     # Main menu
122     def menu(self):
123         while True:
124             print("\n==== Clinic Appointment System ===")
125             print("1. Add Patient")
126             print("2. Add Doctor")
127             print("3. Schedule Appointment")
128             print("4. View Patients")
129             print("5. View Doctors")
130             print("6. View Appointments")
131             print("7. Exit")
132             choice = input("Enter your choice: ")
133
134             if choice == '1':
135                 self.add_patient()
136
137             elif choice == '2':
138                 self.add_doctor()
139
140             elif choice == '3':
141                 self.schedule_appointment()
142
143             elif choice == '4':
144                 self.display_patients()
145
146             elif choice == '5':
147                 self.display_doctors()
148
149             elif choice == '6':
150                 self.display_appointments()
151
152             elif choice == '7':
153                 print("Exiting system. Goodbye!")
154                 break
155
156     # Run the clinic system
157     if __name__ == "__main__":
158         clinic = ClinicSystem()
159         clinic.menu()
```

EXPLORER ... Welcome medical_clinic_appointment_system.py

GQT_MINIPROJECT2

medical_clinic_appointment_system.py

```
45 class ClinicSystem:
125     def menu(self):
126
127         if choice == '1':
128             self.add_patient()
129         elif choice == '2':
130             self.add_doctor()
131         elif choice == '3':
132             self.schedule_appointment()
133
134         elif choice == '4':
135             self.display_patients()
136
137         elif choice == '5':
138             self.display_doctors()
139
140         elif choice == '6':
141             self.display_appointments()
142
143         elif choice == '7':
144             print("Exiting system. Goodbye!")
145             break
146
147         else:
148             print("Invalid choice! Try again.")
149
150
151
152
153
154
155
156     # Run the clinic system
157     if __name__ == "__main__":
158         clinic = ClinicSystem()
159         clinic.menu()
```

Output:-

```
PS C:\Users\reeth\OneDrive\Desktop\GQT_MiniProject2> & 'c:\Users\reeth\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\reeth\vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '53392' '--' 'c:\Users\reeth\OneDrive\Desktop\GQT_MiniProject2\medical_clinic_appointment_system.py'

    == Clinic Appointment System ==
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 1
Enter patient ID: 101
Enter patient name: Geethika
Enter age: 21
Enter contact number: 9191919191
Patient Geethika added successfully!

    == Clinic Appointment System ==
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 1
Enter patient ID: 102
Enter patient name: Geeth
Enter age: 22
Enter contact number: 9292929292
Patient Geeth added successfully!

    == Clinic Appointment System ==
1. Add Patient
2. Add Doctor
```

```
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE Launch ... + ...

    == Clinic Appointment System ==
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 2
Enter doctor ID: 201
Enter doctor name: Kiefer
Enter specialization: cardio
Doctor Kiefer added successfully!

    == Clinic Appointment System ==
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 2
Enter doctor ID: 202
Enter doctor name: Aries
Enter specialization: Pulmonary
Doctor Aries added successfully!

    == Clinic Appointment System ==
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 3
Enter appointment ID: 301
Enter patient ID: 101
Enter doctor ID: 201
```

```
GQT_MINIPROJECT2
medical_clinic_appointment_system.py

Enter your choice: 3
Enter appointment ID: 301
Enter patient ID: 101
Enter doctor ID: 201
Enter appointment date (YYYY-MM-DD): 2026-02-15
Enter appointment time (HH:MM): 05.20
Appointment scheduled for Geethika with Dr. Kiefer.

    === Clinic Appointment System ===
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 3
Enter appointment ID: 302
Enter patient ID: 102
Enter doctor ID: 202
Enter appointment date (YYYY-MM-DD): 2026-02-22
Enter appointment time (HH:MM): 01.73
Appointment scheduled for Geeth with Dr. Aries.

    === Clinic Appointment System ===
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 4
Patient ID: 101
Name: Geethika
Age: 21
Contact: 9191919191
-----
Patient ID: 102
Name: Geeth
Age: 22
```

```
OUTLINE
TIMELINE
EXPLORER
...
PROBLEMS OUTPUT DEBUG CONSOLE Ti :: || ⌂ ⌃ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ Launch + v
GQT_MINIPROJECT2
medical_clinic_appointment_system.py
-----
Patient ID: 102
Name: Geeth
Age: 22
Contact: 9292929292
-----

    === Clinic Appointment System ===
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 5
Doctor ID: 201
Name: Kiefer
Specialization: cardio
-----
Doctor ID: 202
Name: Aries
Specialization: Pulmonary
-----

    === Clinic Appointment System ===
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 6

Appointment ID: 301
Date: 2026-02-15 | Time: 05.20
Patient: Geethika (ID: 101)
Doctor: Kiefer (cardio)
```

EXPLORER ... PROBLEMS OUTPUT DEBUG CONSOLE Launch

GQT_MINIPROJECT2 medical_clinic_appointment_system.py

```
Name: Kiefer
Specialization: cardio
-----
Doctor ID: 202
Name: Aries
Specialization: Pulmonary
-----

==== Clinic Appointment System ===
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 6

Appointment ID: 301
Date: 2026-02-15 | Time: 05.20
Patient: Geethika (ID: 101)
Doctor: Kiefer (cardio)
-----

Appointment ID: 302
Date: 2026-02-22 | Time: 01.73
Patient: Geeth (ID: 102)
Doctor: Aries (Pulmonary)
-----

==== Clinic Appointment System ===
1. Add Patient
2. Add Doctor
3. Schedule Appointment
4. View Patients
5. View Doctors
6. View Appointments
7. Exit
Enter your choice: 7
Exiting system. Goodbye!
```

OUTLINE TIMELINE PS C:\Users\geeth\OneDrive\Desktop\GQT MiniProject2>



GLOBAL QUEST
TECHNOLOGIES

fuel your
passion for
IT with
our **guidance.**



Global Quest Technologies



#324, 2nd Floor, 3 A Cross, Near
Seshadripuram First Grade College,
Above City Union Bank,
Yelahanka New Town,
Bengaluru-560064

+91 9448 403 469 | 080-49720009
info@gqtech.in | www.gqtech.in