Name: **GANESH SANJAY PANDHRE**

Roll Nos : **41** | Div: **D20B**

Aim: To build an adaptive and contextual Cognitive based Customer service application/ Insurance/ Healthcare Application/ Smarter Cities/ Government etc.

Experiment No: 4

Theory:

Introduction

In today's world, **Cognitive Computing** plays an important role in developing applications that are not only intelligent but also adaptive and context-aware. Unlike simple rule-based systems, cognitive systems understand user requirements, adapt according to context, and provide meaningful responses.

In this experiment, we expanded our **Apartment Assistant** application into an **adaptive and contextual system**. The assistant is designed to manage tasks related to apartment living and city services, while also learning and adjusting based on user needs. This brings the application closer to real-world use cases such as customer service, healthcare, government helpdesks, and smart city management.

Adaptive and Contextual Features

- **Adaptive System:** The assistant is able to learn from user inputs. For example, if a user frequently asks about maintenance schedules or billing details, the system can give more focused responses in future interactions.
- **Contextual Responses:** Instead of giving generic answers, the assistant uses the **context of the conversation** to respond. For example, if a user asks "When will my bill be generated?" after already mentioning "electricity", the system understands that the context is about *electricity bills*.

Expansion of Apartment Assistant

In the previous experiment, the focus was mainly on image-based features (like vehicle plate recognition). In this experiment, we **expanded the scope** of the Apartment Assistant by adding adaptive and contextual abilities. Some examples of this expansion are:

- Tracking requests from residents (e.g., complaints, maintenance, billing).
- Providing different services in one conversation instead of requiring separate queries.
- Remembering user preferences (for example, default flat number, frequent requests).
- Handling more general questions related to customer service, healthcare support, or smart city services.

This makes the assistant more powerful and closer to a real-world **cognitive service platform**.

Importance in Real Life

Such adaptive and contextual systems are widely useful in multiple domains:

- **Customer Service:** Intelligent chatbots that solve customer queries quickly by remembering their past issues.
- **Healthcare:** Systems that understand patient history and give personalized support.
- **Insurance:** Assistants that guide customers about policies or claims with relevant context.
- **Smarter Cities:** Applications that connect residents with municipal services like water, electricity, or vehicle parking.
- **Government Services:** Platforms that provide adaptive responses to citizen queries (tax, IDs, certificates, etc.).

```
# IMPORTS
import random
```

```
# KNOWLEDGE BASE (Apartment Data)
apartment_data = {
    "bills": {
        "electricity": 1200,
        "water": 600,
        "maintenance": 1500
    },
    "complaints": [],
    "facilities": ["gym", "swimming pool", "parking", "garden"],
    "visitors": []
}
```

```
# CONTEXT MANAGER
context = {
    "last_query": None,
    "pending_action": None
}
```

```
# ADAPTIVE SUGGESTIONS ENGINE
def adaptive_suggestions(user_query):
    if "bill" in user_query:
        return "Would you like me to set up an auto-payment reminder for your bills?"
    elif "complaint" in user_query:
        return "I noticed you lodge frequent complaints. Should I schedule a maintenance check automat
    elif "visitor" in user_query:
        return "Shall I notify the security guard about frequent visitors?"
    return None
```

```
# ASSISTANT RESPONSE FUNCTION
def apartment_assistant(user_query):
    user_query = user_query.lower()
    response = ""

    # Bills
    if "bill" in user_query:
```

```
        if "electricity" in user_query:
            response = f"Your electricity bill is Rs.{apartment_data['bills']['electricity']}."
        elif "water" in user_query:
            response = f"Your water bill is Rs.{apartment_data['bills']['water']}."
        elif "maintenance" in user_query:
            response = f"Your maintenance bill is Rs.{apartment_data['bills']['maintenance']}."
        else:
            bills = ", ".join([f"{k}: Rs.{v}" for k,v in apartment_data['bills'].items()])
            response = f"Here are your bills: {bills}."

    # Complaints
    elif "complaint" in user_query:
        complaint_text = user_query.replace("complaint", "").strip()
        if complaint_text:
            apartment_data['complaints'].append(complaint_text)
            response = f"Complaint noted: '{complaint_text}'. Our team will resolve it soon."
        else:
            response = f"You have lodged {len(apartment_data['complaints'])} complaints so far."

    # Facilities
    elif "facility" in user_query or "facilities" in user_query:
        response = "Available facilities are: " + ", ".join(apartment_data['facilities'])

    # Visitors
    elif "visitor" in user_query:
        visitor_name = user_query.replace("visitor", "").strip()
        if visitor_name:
            apartment_data['visitors'].append(visitor_name)
            response = f"Visitor '{visitor_name}' has been added to the guest list."
        else:
            response = f"You have {len(apartment_data['visitors'])} visitors recorded."

    # Contextual continuation
    elif "pay it" in user_query or "yes" in user_query:
        if context["last_query"] and "bill" in context["last_query"]:
            response = "Your last bill has been marked as paid. Thank you!"
        else:
            response = "I’m not sure what you are confirming. Could you clarify?"

    else:
        response = "Sorry, I didn’t understand that. You can ask about bills, complaints, facilities, (

    # Save context
    context["last_query"] = user_query

    # Add adaptive suggestion
    suggestion = adaptive_suggestions(user_query)
    if suggestion:
        response += "\nSuggestion: " + suggestion

    return response
```

```
query = "Show me all my bills"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: Show me all my bills
Assistant: Here are your bills: electricity: Rs.1200, water: Rs.600, maintenance: Rs.1500.
Suggestion: Would you like me to set up an auto-payment reminder for your bills?
```

```
query = "What about electricity bill?"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: What about electricity bill?
Assistant: Your electricity bill is Rs.1200.
Suggestion: Would you like me to set up an auto-payment reminder for your bills?
```

```
query = "Can you pay it?"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: Can you pay it?
Assistant: Your last bill has been marked as paid. Thank you!
```

```
query = "What about electricity bill?"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
query = "I want to register a complaint about water leakage"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: I want to register a complaint about water leakage
Assistant: Complaint noted: 'i want to register a  about water leakage'. Our team will resolve it
Suggestion: I noticed you lodge frequent complaints. Should I schedule a maintenance check automat
```

```
query = "Show facilities"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: Show facilities
Assistant: Available facilities are: gym, swimming pool, parking, garden
```

```
query = "Add visitor Ramesh"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: Add visitor Ramesh
Assistant: Visitor 'add  ramesh' has been added to the guest list.
Suggestion: Shall I notify the security guard about frequent visitors?
```

```
query = "Show visitors"
print(f"User: {query}")
print(f"Assistant: {apartment_assistant(query)}")
```

```
User: Show visitors
Assistant: Visitor 'show s' has been added to the guest list.
Suggestion: Shall I notify the security guard about frequent visitors?
```

Conclusion

This experiment successfully demonstrated how an **apartment assistant can be expanded** into a more **adaptive and contextual cognitive application**. Instead of just performing fixed tasks, the assistant can now understand the **user's intent**, respond according to the **situation**, and even improve based on past interactions. This kind of system can be applied to real-world areas like smart cities, customer service, insurance, healthcare, and government services, making them more intelligent and user-friendly.