EXPERIMENT NO . 6

Aim :- Identify suitable Agent Architecture for the problem

Problem Statement :- Identify Agent architecture for Virtual assistants.

Pseudo Code :-

```
Initialize VirtualAssistant:
   Load profiles, NLP engine, task system

Function HandleUserRequest(user_id, request):
   profile = GetUserProfile(user_id)
   goal = IdentifyGoal(ParseRequest(request))

   If IsKnownTask(goal):
      response = ExecuteTask(goal, profile)
   Else:
      response = LearnAndHandle(goal)

   UpdateUserProfile(user_id, request, response)
   return response

Function ParseRequest(request):
   return NLPParser(request)

Function IsKnownTask(goal):
   return CheckTaskDatabase(goal)

Function ExecuteTask(goal, profile):
   Return TaskExecutor(goal, profile) # Handles specific tasks like setting reminders

Function LearnAndHandle(goal):
   return LearnFromInteraction(goal)

Function UpdateUserProfile(user_id, request, response):
   ModifyUserProfile(user_id, request, response)

# Main loop
For each user_id in ActiveUsers:
   response = HandleUserRequest(user_id, GetUserRequest(user_id))
   SendResponseToUser(user_id, response)
```

Conclusion :- The virtual assistant system combines goal-based, utility-based, and learning agent models to handle user interactions efficiently. It parses requests, determines goals, executes tasks, and adapts through learning. This approach ensures personalized, dynamic responses and an evolving user experience.