Name: Shrusti Chintawar Roll No: 15 Div: D20B

Experiment No: 03

Aim - To build a Cognitive based application to acquire knowledge through images for a student service (timetable) application.

1. Introduction & Objectives

Introduction:

The increasing availability of digital images in various sectors such as Customer Service, Insurance, Healthcare, Smart Cities, and Government has created opportunities to automate the extraction and understanding of information embedded in images. Cognitive applications leveraging computer vision and artificial intelligence enable efficient and accurate acquisition of knowledge from these images, reducing manual effort and improving service delivery.

Objectives:

- To develop a cognitive application that can extract meaningful information from images related student timetable.To implement image preprocessing and analysis techniques such as Optical Character Recognition (OCR) and object detection to identify and interpret relevant data.
- To demonstrate how extracted knowledge can be utilized to automate tasks or assist decision-making in the chosen domain.

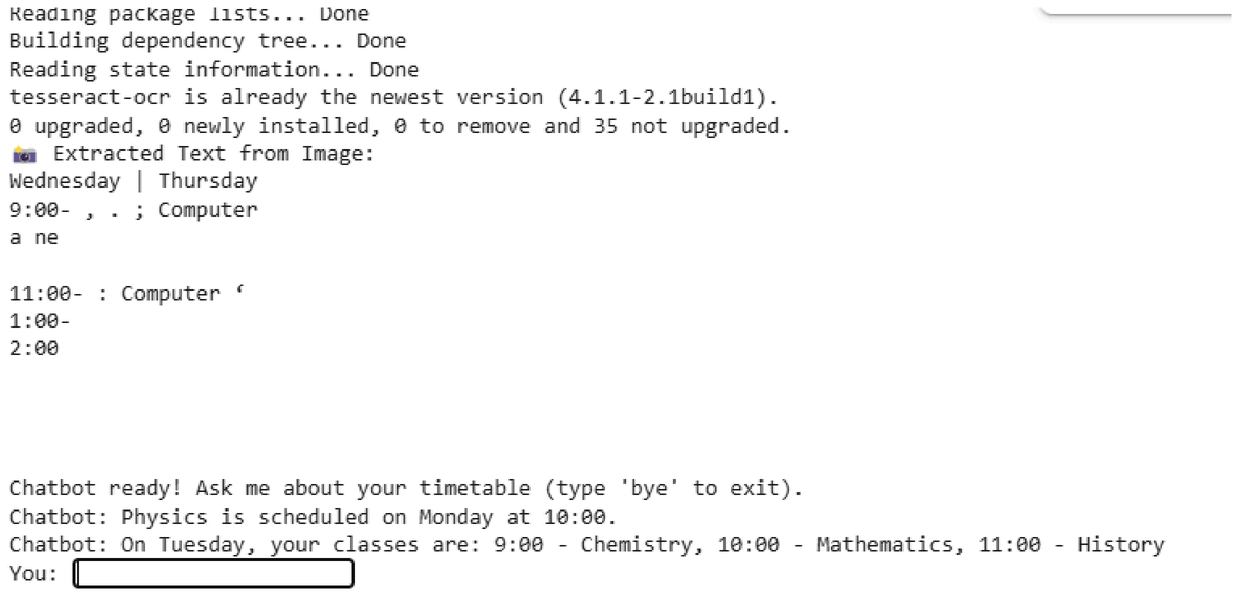
CODE:

```
!pip install pytesseract pillow spacy nltk --quiet

!apt-get install tesseract-ocr -y



import pytesseract

from PIL import Image

import spacy
```

```
# Load spaCy
nlp = spacy.load("en_core_web_sm")

# Timetable dictionary (from image)
timetable = {
        "monday": {"9:00": "Mathematics", "10:00": "Physics", "11:00":
"English"},
       "tuesday": {"9:00": "Chemistry", "10:00": "Mathematics", "11:00":
"History"},
        "wednesday": {"9:00": "Biology", "10:00": "Physics", "11:00":
"Computer Science"},
     "thursday": {"9:00": "Computer Science", "10:00": "Biology", "11:00":
"English"},
    "friday": {}
}

# Function to answer timetable queries
def answer_query(user_query):
    user_query = user_query.lower()

    # Day-based query
    for day in timetable:
        if day in user_query:
            if timetable[day]:
                  return f"On {day.capitalize()}, your classes are: " + ",
".join(
                                  [f"{time} - {subj}" for time, subj in
timetable[day].items()]
                )
            else:
                return f"No scheduled classes on {day.capitalize()}."

    # Subject-based query
    for day, subjects in timetable.items():
        for time, subj in subjects.items():
            if subj.lower() in user_query:
                    return f"{subj} is scheduled on {day.capitalize()} at
{time}."

    return "Sorry, I couldn't find that in your timetable."

# OCR function
def process_image(image_path):
    img = Image.open(image_path)
    extracted_text = pytesseract.image_to_string(img)
    return extracted_text

# ---- Example Run ----
# Upload timetable image:
```

```python
# from google.colab import files; uploaded = files.upload()
image_path = "sample_timetable.png"  # replace with your uploaded file
extracted_text = process_image(image_path)

print("📸 Extracted Text from Image:")
print(extracted_text)

# Interactive loop
print("\nChatbot ready! Ask me about your timetable (type 'bye' to exit).")
while True:
    user_query = input("You: ")
    if user_query.lower() in ["bye", "exit", "quit"]:
        print("Chatbot: Bye! Good luck with your studies.")
        break
    response = answer_query(user_query)
    print("Chatbot:", response)
```

OUTPUT



```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tesseract-ocr is already the newest version (4.1.1-2.1build1).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
📸 Extracted Text from Image:
Wednesday | Thursday
9:00- , . ; Computer
a ne

11:00- : Computer ‘
1:00-
2:00




Chatbot ready! Ask me about your timetable (type 'bye' to exit).
Chatbot: Physics is scheduled on Monday at 10:00.
Chatbot: On Tuesday, your classes are: 9:00 - Chemistry, 10:00 - Mathematics, 11:00 - History
You:
```

Conclusion

This experiment demonstrates a cognitive application that uses OCR and NLP to process student-related images and provide relevant responses, showcasing the potential of intelligent assistants for academic support.