```
import speech_recognition as sr
import datetime
import subprocess
import pywhatkit
import pyttsx3
import webbrowser

engine=pyttsx3.init()
voices=engine.getProperty('voices')
engine.setProperty('voice',voices[1].id)
recognizer=sr.Recognizer()

def cmd():
    with sr.Microphone() as source:
        print("Clearing background noises...Pleasw wait")
        recognizer.adjust_for_ambient_noise(source,duration=0.5)
        print('Ask me anything..')
        recordedaudio=recognizer.listen(source)
    try:
        text=recognizer.recognize_google(recordedaudio,language='en_US')
        text=text.lower()
        print('Your message:',format(text))

    except Exception as ex:
        print(ex)
    if 'chrome'in text:
        a='Opening chrome..'
        engine.say(a)
        engine.runAndWait()
        programName = "C:\Program Files\Google\Chrome\Application\chrome.exe"
        subprocess.Popen([programName])
    if 'time' in text:
        time = datetime.datetime.now().strftime('%I:%M %p')
        print(time)
        engine.say(time)
        engine.runAndWait()
    if 'play' in text:
        a='opening youtube..'
        engine.say(a)
        engine.runAndWait()
        pywhatkit.playonyt(text)
    if 'youtube' in text:
        b='opening youtube'
        engine.say(b)
        engine.runAndWait()
        webbrowser.open('www.youtube.com')
while True:
    cmd()
```

Certainly! Let's go through the code line by line and explain what each part does:

```
import speech_recognition as sr
import datetime
import subprocess
import pywhatkit
import pyttsx3
import webbrowser
```

- **Imports necessary libraries**:
  - speech_recognition is used for recognizing speech input.
  - datetime is used to get the current time.
  - subprocess is used to run system commands.
  - pywhatkit is used to perform various tasks such as playing YouTube videos.
  - pyttsx3 is used for text-to-speech conversion.
  - webbrowser is used to open web pages in the default browser.

```
engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
recognizer = sr.Recognizer()
```

- **Initializes the text-to-speech engine**:
  - engine = pyttsx3.init() initializes the pyttsx3 engine.
  - voices = engine.getProperty('voices') retrieves the available voices.
  - engine.setProperty('voice', voices[1].id) sets the voice to the second available voice (usually a female voice).
- **Initializes the speech recognizer**:
  - recognizer = sr.Recognizer() initializes the speech recognizer from the speech_recognition library.

```
def cmd():
```

- **Defines the cmd function** which listens for and processes voice commands.

```
    with sr.Microphone() as source:
        print("Clearing background noises...Please wait")
        recognizer.adjust_for_ambient_noise(source, duration=0.5)
        print('Ask me anything..')
        recordedaudio = recognizer.listen(source)
```

- **Captures audio input**:
  - with sr.Microphone() as source: uses the microphone as the audio source.
  - print("Clearing background noises...Please wait") informs the user that the system is adjusting for ambient noise.
  - recognizer.adjust_for_ambient_noise(source, duration=0.5) adjusts the recognizer to ignore ambient noise for 0.5 seconds.
  - print('Ask me anything..') prompts the user to speak.

- recordedaudio = recognizer.listen(source) listens and records the audio input from the user.

```
try:
   text = recognizer.recognize_google(recordedaudio, language='en_US')
   text = text.lower()
   print('Your message:', format(text))
```

- **Processes the recorded audio**:
  - The try block attempts to recognize the speech using Google's recognition service:
    - text = recognizer.recognize_google(recordedaudio, language='en_US') converts the audio to text.
    - text = text.lower() converts the text to lowercase for easier comparison.
    - print('Your message:', format(text)) prints the recognized text.

```
except Exception as ex:
   print(ex)
```

- **Handles exceptions**:
  - If speech recognition fails, the except block catches the exception and prints it.

```
if 'chrome' in text:
   a = 'Opening chrome..'
   engine.say(a)
   engine.runAndWait()
   programName = "C:\Program Files\Google\Chrome\Application\chrome.exe"
   subprocess.Popen([programName])
```

- **Opens Google Chrome if 'chrome' is in the recognized text**:
  - if 'chrome' in text: checks if 'chrome' is in the text.
  - a = 'Opening chrome..' sets the response text.
  - engine.say(a) converts the response text to speech.
  - engine.runAndWait() waits until the speech is finished.
  - programName = "C:\\Program Files\\Google\\Chrome\\Application\\chrome.exe" specifies the path to the Chrome executable.
  - subprocess.Popen([programName]) opens Chrome using the subprocess module.

```
if 'time' in text:
   time = datetime.datetime.now().strftime('%I:%M %p')
   print(time)
   engine.say(time)
   engine.runAndWait()
```

- **Tells the current time if 'time' is in the recognized text**:
  - if 'time' in text: checks if 'time' is in the text.
  - time = datetime.datetime.now().strftime('%I:%M %p') gets the current time in a 12-hour format with AM/PM.

- print(time) prints the current time.
- engine.say(time) converts the time to speech.
- engine.runAndWait() waits until the speech is finished.

```
if 'play' in text:
    a = 'opening youtube..'
    engine.say(a)
    engine.runAndWait()
    pywhatkit.playonyt(text)
```

- **Plays a YouTube video if 'play' is in the recognized text**:
  - if 'play' in text: checks if 'play' is in the text.
  - a = 'opening youtube..' sets the response text.
  - engine.say(a) converts the response text to speech.
  - engine.runAndWait() waits until the speech is finished.
  - pywhatkit.playonyt(text) plays the video on YouTube based on the recognized text.

```
if 'youtube' in text:
    b = 'opening youtube'
    engine.say(b)
    engine.runAndWait()
    webbrowser.open('www.youtube.com')
```

- **Opens YouTube if 'youtube' is in the recognized text**:
  - if 'youtube' in text: checks if 'youtube' is in the text.
  - b = 'opening youtube' sets the response text.
  - engine.say(b) converts the response text to speech.
  - engine.runAndWait() waits until the speech is finished.
  - webbrowser.open('www.youtube.com') opens the YouTube website in the default web browser.

```
while True:
    cmd()
```

- **Runs the cmd function in an infinite loop**:
  - while True: creates an infinite loop.
  - cmd() calls the cmd function to continuously listen for and process commands.