Shri Govindram Seksaria Institute of technology Indore (M.P.)

Yashmin Mohare

29 October 2023

Programming Practices : Mini Project(Jarvis)

https://github.com/YashminMohare/pp/tree/main

Objectives

1. To develop an AI-powered assistant, named Jarvis, capable of automating tasks, providing personalized recommendations, and enhancing user productivity in a variety of contexts, such as home automation, healthcare, or business operations.
2. In some fictional depictions, Jarvis is a smart home AI that manages daily tasks and household functions. One of its objectives could be to optimize home automation, making life more convenient for its users by controlling appliances, providing weather updates, and managing schedules.

Technology Used

Python

Modules For Install

i. pip install pyttsx3
ii. pip install webbrowser
iii. pip install os
iv. pip install SpeechRecognition
v. pip install Microphone
vi. pip install Pillow
vii. pip install wikipedia
viii. pip install Pillow-PIL or PIL

Purpose

Virtual Assistant: A Jarvis-like AI can serve as a virtual assistant, helping users with tasks such as setting reminders, managing schedules, answering questions, and providing information or recommendations. Home Automation: In a smart home context,

Jarvis can automate various tasks, such as controlling lighting, temperature, security systems, and home entertainment devices to enhance convenience and energy efficiency. Entertainment:

Jarvis-like AI can enhance the entertainment experience by suggesting movies, music, or books, and even creating personalized playlists or content recommendations.

Scope

i. Personal Assistant: A Jarvis AI can serve as a comprehensive personal assistant, managing tasks such as scheduling, reminders, and providing information on demand.
ii. Smart Home Automation: It can control and automate various functions within a smart home, including lighting, temperature, security, entertainment systems, and more.
iii. Voice-controlled Device Integration: Jarvis can integrate with and control various voice-controlled devices like smart speakers, home appliances, and IoT devices.

iv. Data Analysis and Insights: In a business context, it can process and analyze data to provide insights, make data-driven decisions, and improve operations.

v. Natural Language Processing: Jarvis can be designed to understand and generate human-like text, making it useful for tasks like content creation, translation, and customer support chatbots.

vi. Voice Recognition and Commands: It can recognize and respond to voice commands for various tasks, from making calls and sending messages to playing music or finding information.

vii. Security and Surveillance: Jarvis can monitor and alert users to potential security threats or incidents, whether at home or in a corporate setting.

viii. Health and Wellness: It can provide information and guidance on health and wellness topics, helping users maintain a healthy lifestyle.

ix. Education and Tutoring: Jarvis can serve as a virtual tutor, assisting with homework, explaining concepts, and providing personalized learning experiences.

x. Content Recommendations: It can offer recommendations for movies, music, books, news, and other forms of entertainment based on user preferences and behavior.

xi. Accessibility Features: Jarvis can assist individuals with disabilities by providing accessibility features such as text-to-speech, speech recognition, and screen reading.

xii. Business Applications: In the business world, it can streamline operations, assist with customer service, and automate repetitive tasks to improve efficiency.

xiii. Research and Data Retrieval: For researchers and professionals, Jarvis can assist with information retrieval, perform literature reviews, and generate reports.

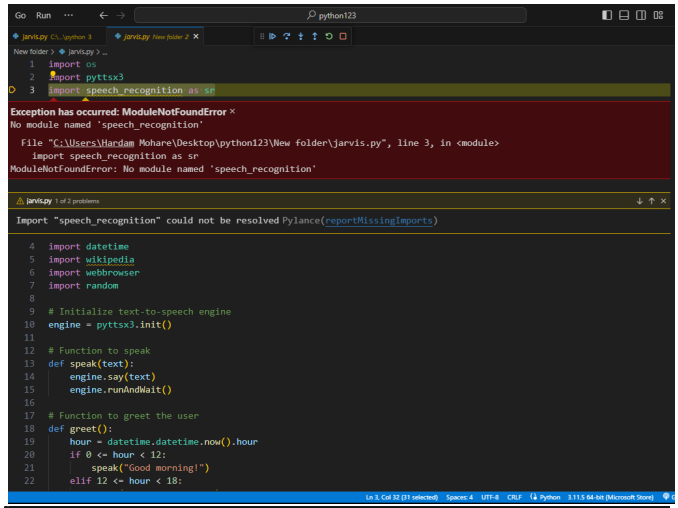
xiv. Language Translation and Interpretation: It can translate languages and serve as an interpreter for international communication.

xv. Emergency Assistance: Jarvis can be programmed to provide emergency information and guidance during critical situations.

Output

```
Listening...
User said: whats D time
Listening...
User said: what's the time
22:24:52
Listening...
User said: open website
Which website would you like to open?
Listening...
```

Debugging



Profiling

```
Listening...
         798344 function calls (762716 primitive calls) in 27.368 seconds

   Ordered by: cumulative time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    397/1    0.005    0.000   27.369   27.369 {built-in method builtins.exec}
        1    0.000    0.000   27.369   27.369 jarvis.py:1(<module>)
        1    0.000    0.000   20.749   20.749 jarvis.py:29(process_command)
        1    0.022    0.022   19.565   19.565 __init__.py:450(listen)
      842    0.005    0.000   19.534    0.023 __init__.py:198(read)
      842    0.005    0.000   19.529    0.023 __init__.py:553(read)
      842   19.524    0.023   19.524    0.023 {built-in method pyaudio._portaudio.read_stream}
        1    0.000    0.000    5.527    5.527 jarvis.py:18(greet)
        2    0.000    0.000    5.527    2.764 jarvis.py:13(speak)
        2    0.000    0.000    5.527    2.764 engine.py:168(runAndWait)
        2    0.000    0.000    5.527    2.764 driver.py:186(runAndWait)
        2    0.000    0.000    5.527    2.764 sapi5.py:128(startLoop)
      8/4    0.000    0.000    5.402    1.350 driver.py:108(setBusy)
        8    0.000    0.000    5.402    0.675 driver.py:81(_pump)
        2    5.402    2.701    5.402    2.701 sapi5.py:52(say)
       59    0.002    0.000    3.621    0.061 __init__.py:1(<module>)
  488/11    0.005    0.000    1.769    0.161 <frozen importlib._bootstrap>:1165(_find_and_load)
    474/7    0.003    0.000    1.769    0.253 <frozen importlib._bootstrap>:1120(_find_and_load_unlocked)
  447/11    0.003    0.000    1.753    0.159 <frozen importlib._bootstrap>:666(_load_unlocked)
  394/10    0.002    0.000    1.752    0.175 <frozen importlib._bootstrap_external>:934(exec_module)
 1019/16    0.001    0.000    1.740    0.109 <frozen importlib._bootstrap>:233(_call_with_frames_removed)
   118/19    0.000    0.000    1.369    0.072 {built-in method builtins.__import__}
  471/270    0.007    0.000    1.034    0.004 <frozen importlib._bootstrap>:1054(_find_spec)
     14/7    0.000    0.000    1.009    0.144 __init__.py:108(import_module)
     16/7    0.000    0.000    1.009    0.144 <frozen importlib._bootstrap>:1192(_gcd_import)
        1    0.000    0.000    0.967    0.967 __init__.py:74(__init__)
        1    0.000    0.000    0.889    0.889 __init__.py:182(get_pyaudio)
  473/270    0.001    0.000    0.863    0.003 __init__.py:89(find_spec)
      2/1    0.000    0.000    0.862    0.862 __init__.py:99(spec_for_distutils)
  442/187    0.001    0.000    0.714    0.004 <frozen importlib._bootstrap>:1207(_handle_fromlist)
```

```
      2/1    0.000    0.000    0.862    0.862 __init__.py:99(spec_for_distutils)
  442/187    0.001    0.000    0.714    0.004 <frozen importlib._bootstrap>:1207(_handle_fromlist)
      394    0.008    0.000    0.628    0.002 <frozen importlib._bootstrap_external>:1007(get_code)
      660    0.004    0.000    0.486    0.001 <frozen importlib._bootstrap_external>:1127(get_data)
        5    0.000    0.000    0.391    0.078 version.py:1(<module>)
        1    0.000    0.000    0.366    0.366 __init__.py:6(init)
        1    0.000    0.000    0.366    0.366 engine.py:28(__init__)
        1    0.000    0.000    0.366    0.366 driver.py:28(__init__)
      661    0.343    0.001    0.343    0.001 {method 'read' of '_io.BufferedReader' objects}
 449/401    0.001    0.000    0.322    0.001 <frozen importlib._bootstrap>:566(module_from_spec)
     2612    0.321    0.000    0.321    0.000 {built-in method nt.stat}
      450    0.003    0.000    0.304    0.001 <frozen importlib._bootstrap_external>:1464(_get_spec)
      449    0.001    0.000    0.303    0.001 <frozen importlib._bootstrap_external>:1496(find_spec)
     1051    0.014    0.000    0.297    0.000 <frozen importlib._bootstrap_external>:1604(find_spec)
     2213    0.002    0.000    0.289    0.000 <frozen importlib._bootstrap_external>:148(_path_stat)
        2    0.000    0.000    0.268    0.134 dist.py:1(<module>)
        1    0.000    0.000    0.233    0.233 wikipedia.py:1(<module>)
1259/1048    0.032    0.000    0.221    0.000 {built-in method builtins.__build_class__}
        1    0.000    0.000    0.213    0.213 sapi5.py:29(buildDriver)
        1    0.000    0.000    0.213    0.213 sapi5.py:34(__init__)
        2    0.000    0.000    0.206    0.103 requirements.py:1(<module>)
       10    0.000    0.000    0.204    0.020 __init__.py:31(create_module)
       10    0.000    0.000    0.204    0.020 __init__.py:29(load_module)
      465    0.003    0.000    0.196    0.000 __init__.py:272(_compile)
      353    0.000    0.000    0.196    0.001 __init__.py:228(compile)
      265    0.001    0.000    0.191    0.001 _compiler.py:738(compile)
        7    0.000    0.000    0.175    0.025 exceptions.py:1(<module>)
        1    0.174    0.174    0.174    0.174 sapi5.py:86(getProperty)
        1    0.000    0.000    0.163    0.163 __init__.py:171(__enter__)       1    0.000    0.000    0.146    0.146 element.py:1(<module>)
        1    0.000    0.000    0.146    0.146 sapi5.py:1(<module>)
        1    0.000    0.000    0.140    0.140 __init__.py:631(open)
        1    0.000    0.000    0.140    0.140 __init__.py:297(__init__)
        1    0.140    0.140    0.140    0.140 {built-in method pyaudio._portaudio.open}
        2    0.000    0.000    0.136    0.068 __init__.py:3233(_call_aside)
        1    0.000    0.000    0.136    0.136 __init__.py:3259(_initialize_master_working_set)
        1    0.000    0.000    0.111    0.111 monkey.py:43(patch_all)
```