Madhav Institute of Technology and Science, Gwalior

(A Govt. Aided UGC Autonomous Institute, affiliated to R.G.P.V, Bhopal, M.P)

Department of Computer Science & Engineering and Information Technology

A

Internship Report

On

DATA SCIENCE

(Code: 150801)

Session: Jan – june 2021

Submitted To:
Prof. Sneha Garg

Submitted By:
Monarch koli
CSE-4th year
0901CS171062

Offer letter

KANISHKA Information Technology PVT. LTD.

WEBSITE & SOFTWARE TRAINING AND DEVELOPMENT COMPANY

Incorporated under the companies act.1956 (No. 1 of 1956)

Reg.No.U72200MP2010PTC023769

Letter to Seek Industrial Training

RE:-Industrial Training

Student Name: Monarch Koli

Residential Address:Gole ka mandir Gwalior

Mobile Number: 8109305196

Email ID:monarchkoli12@gmail.com

Project Details

Title of Project: Data Science

Project Type: Core/ Non Core

We are glad to inform you that our Company would be assisting you by providing places or Temporary employment to learn and experience firsthand our industry. The Student is required to work full time for our company and observe all working conditions.

The student has joined our organization for the VIII semester Industrial Project And training will start from 20 feb 2021

Thank You in Anticipation
Geeta Kakrani (CEO)

209-210 GARIMA ARCADE M.L.B. ROAD GWALIOR
MOB.NO. 09300666805,09300316290

Acknowledgement

I have taken efforts in this internship, However it would not be possible without the kind support and help of many individuals and organization.I would like to extent my sincere thanks to all of them.

It is our privilege to express our sincerest regards to our internship coordinator, **Prof. SNEHA GARG** for their valuable inputs,guidance,encouragement,whole-hearted coorperation and constructive criticism throughout the duration of the Internship.

I would also like to thank **Mrs.Geeta Kakrani** head of Kanishka IT Pvt. Ltd for great guidance ,cooperation & support throughout the journey of internship and ahead.

I further extend my warm gratitude and regards to everyone who helped me during my internship.

Table of Content

About Organisation

Since 2006, Kanishka IT Pvt. Ltd. established . It Specialised in Web Application Development, Website Designing, Software Application Development and Mobile Application Development, Corporate Training, Research Centre, Campus Placement Training, Industrial Training . We specialize in cloud computing and our offerings are highly customizable and scalable. We believe in enahncing the skills of students. Every student is different from another so are our state-of-the-art solutions.

Features: -

1. Provide Multiple Courses.
2. Provide different types of Internships.
3. Provide knowledge for Job preparations.
4. Provide different types of Quizzes for better learning.
5. Provide Previous year IT Companies Test.
6. Provide Placement preparation.
7. Provide Certificate after completion of the Course.

About the Project

The project developed during the internship is titled **Desktop Voice Assistant**.

This project helps to have Personal digital assistants which are capturing a lot of attention lately. Chatbots are common in most commercial websites. With growing advancements in artificial intelligence, training the machines to tackle day-to-day tasks is the norm.

Voice based personal assistants have gained a lot of popularity in this era of smart homes and smart devices. These personal assistants can be easily configured to perform many of your regular tasks by simply giving voice commands. Google has popularized voice-based search that is a boon for many like senior citizens who are not comfortable using the keypad/keyboard.

Aim of the website: Voice based personal assistant {***MAVIS***}

This voice based personal assistant contains:

- It can send emails on your behalf.
- It can play music for you.
- It can do Wikipedia searches for you.
- It is capable of opening websites like Google, Youtube, etc., in a web browser.
- It is capable of opening your code editor or IDE with a single voice command.

About Technology Used

Visual studio Code: Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

PYTHON:

Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open.

Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

Basically Python is a high-level programming language for overall programming. Besides being an open-source programming language, python is an extraordinarily interpreted, object-oriented, and interactive programming language. Python joins surprising power with clear syntax. It has modules, classes, special cases, significant level dynamic data types, and dynamic composing. There are interfaces to numerous system calls and libraries, as well as to different windowing frameworks.

Machine Learning:

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning** focuses on the development of computer programs that can access data and use it to learn for themselves.

Currently, **machine learning** has been **used** in multiple **fields** and industries. For example, medical diagnosis, image processing, prediction, classification, **learning** association, regression etc.

Machine Learning Applications in Daily Life

1. Commute Estimation. ...
2. Email Intelligence. ...
3. Banking and Personal Finance. ...
4. Evaluation and Assessment. ...
5. Social Networking. ...
6. Medical Diagnosis and Healthcare. ...
7. Personal Smart Assistants.
8. Automated mobile word arrange and prediction.

Important libraries used for recognition of speech:

- **Speak function:**

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our M.A.V.I.S. talk, we will make a function called **speak().** This function will take audio as an argument, and then it will pronounce it**.**

```
def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

- DEFINING TAKE COMMAND FUNCTION:

The next most important thing for our A.I. assistant is that it should take command with the help of the microphone of the user's system. So, now we will make a **takeCommand()** function. With the help of the takeCommand() function, our A.I. assistant will return a string output by taking microphone input from the user.

Before defining the takeCommand() function, we need to install a module called **speechRecognition.** Install this module by:

```
pip install speechRecognition
```

- **Pyttsx3:**

✓ A python library that will help us to convert text to speech. In short, it is a text-to-speech library.
✓ It works offline, and it is compatible with Python 2 as well as Python 3.

```
pip install pyttsx3
```

- **Sapi5:**

✓ Microsoft developed speech API(Application Programming Interference).
✓ Helps in synthesis and recognition of voice.

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
```

- **Voiceid:**

✓ Voice id helps us to select different voices.
✓ voice[0].id = Male voice
✓ voice[1].id = Female voice.

```
✓ engine.setProperty('voice', voices[1].id)
```

- **wishme():**

Now, we will make a **wishme()** function, that will make our M.A.V.I.S. wish or greet the user according to the time of computer or pc. To provide current or live time to A.I., we need to import a module called datetime. Import this module to your program, by:

```
import datetime
```

Now, let's start defining the **wishme()** function:

```
def wishme():

hour = int(datetime.datetime.now().hour)
```

and for importing all the directories we use {**pip install (and library name)}.**

As we already learn and know about that for making an AI based desktop vice assistant we have to configure some of the modules to work with that. Like opening youtube , music, google…etc.

At the end the ques. Arises that **IS THIS AN A.I.?**The virtual assistant that we have created is not an A.I, but it is the output of a bunch of the statement. But, if we look at the fundamental level, the sole purpose of A.I develop machines that can perform human tasks with the same effectiveness or even more effectively than humans.

Import libraries:

```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import wikipedia #pip install wikipedia
import webbrowser
import os
import smtplib
```

smtplib:

- Simple Mail Transfer Protocol (SMTP) is a protocol that allows us to send emails and to route emails between mail servers. An instance method called **sendmail** is present in the SMTP module. This instance method allows us to send an email. It takes 3 parameters:
- **The sender:** Email address of the sender.
- **The receiver:** T Email of the receiver.
- **The message*:*** A string message which needs to be sent to one or more than one recipient.

Code to be used in the fact of running the code :

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[1].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()

#wishme_code

def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")

    elif hour>=12 and hour<18:
        speak("Good Afternoon!")

    else:
        speak("Good Evening!")

    speak("I am Mavis Sir. Please tell me how may I help you")

def takeCommand():
    #It takes microphone input from the user and returns string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

    except Exception as e:
        # print(e)
        print("Say that again please...")
        return "None"
    return query
```

```python
#email_code

def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('monarchkoli12@gmail.com', '12345678')
    server.sendmail('monarchkoli12@gmail.com’, to, content)
    server.close()

if __name__ == "__main__":
    wishMe()
    while True:
    # if 1:
        query = takeCommand().lower()

        # Logic for executing tasks based on query
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'open youtube' in query:
            webbrowser.open("youtube.com")

        elif 'open google' in query:
            webbrowser.open("google.com")

        elif 'open stackoverflow' in query:
            webbrowser.open("stackoverflow.com")


        elif 'play music' in query:
            webbrowser.open("music.youtube.com")


        elif 'the time' in query:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak("Sir, the time is {strTime}")
```

```
       elif 'open code' in query:
            codePath = "C:\\Users\\monar\\AppData\\Local\\Programs\\Microsoft
VS Code\\Code.exe"
            os.startfile(codePath)


        elif 'email to monarch' in query:
            try:
                speak("What should I say?")
                content = takeCommand()
                to = "monarchkoli12@gmail.com"
                sendEmail(to, content)
                speak("Email has been sent!")

except Exception as e:
                print(e)
                speak("Sorry SIR. I am not able to send this email")
```

output:

```
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
```

- Whole code runs on the condition of elif() conditions.

As we run the code firstly the AI greets us after that it waits for threshold pause of 1ms after that its starts listening our commands.

And there we have to ask the M.A.V.I.S. for the commands.

Conclusion

This project based internship helped me in gaining valuable information and practical knowledge on several topics like python, virtual studio code, modules, libraries etc. and how to link each and every thing with respective modules. And many more new terminologies for making a smart AI based virtual assistant. And maybe in future this project helps to lead new ideas and new mind opening projects.

The entire system is secured, and this project helped us to understand about the development phases of a project and software development life cycle. We learned how to test different features of a project and lot of things.

It was a wonderful and learning experience for me while working on this project based internship.This project took me through the various phases of project development and gave me an real insight into the world of web developnmnet.The joy of work and thrill involved while tackling the various problems and challenges gave me a feel of developers industry.

I enjoyed each and every bit of work I had put into this.