MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Web Automation

Submitted By:

Nitin Sen

0901CS203D07

Prince Kumar

0901CS203D09

Faculty Mentor:

Dr. Anjula Mehto

Assistant Professor, CSE, MITS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Web Automation

A project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Nitin Sen

0901CS203D07

Prince Kumar

0901CS203D09

Faculty Mentor:

Dr. Anjula Mehto

Assistant Professor, CSE, MITS

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Nitin Sen (0901CS203D07)** has submitted the project report titled **Web Automation** under the mentorship of **Dr. Anjula Mehto** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

09/05/22

Dr. Anjula Mehto
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Prince Kumar (0901CS203D09)** has submitted the project report titled **Web Automation** under the mentorship of **Dr. Anjula Mehto** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Anjula Mehto 09/05/22

Dr. Anjula Mehto
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2022

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

We hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Dr. Anjula Mehto, Assistant Professor**, Computer Science & Engineering

We declare that We have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Nifin Sen
0901CS203D07
3rd Year
Computer Science and Engineering

Prince Kumar
0901CS203D09
3rd Year
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to our career. We are thankful to our institute, **Madhav Institute of Technology and Science** to allow us to continue my disciplinary/ interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. We extend our gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

We would sincerely like to thank our department, **Department of Computer Science and Engineering, for allowing** us to explore this project. We humbly thanks **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

We are sincerely thankful to our faculty mentors. We are grateful to the guidance of **Dr. Anjula Mehto** Assistant professor, Computer Science & Engineering, for their continued support and guidance throughout the project. We are also very thankful to the faculty and staff of the department.

Nitin Sen
0901CS203D07
3rd Year
Computer Science and Engineering

Prince Kumar
0901CS203D09
3rd Year
Computer Science and Engineering

ABSTRACT

Software testing is a necessary technique to locate defects and to increase any of the software application quality. Now lot of applications is being created in internet which is executed in an internet browser. Web functions are turning into an extra complicated, so that it becomes hard to take a look physically. It might extend time and cost. So we are in need of automation testing to check out equipment elevated day by day. Selenium is a web application testing tool which is open source. This equipment is extensively used for checking out the functionality of website developed for variety of purposes. In this project we test the web application using selenium python. Testing is the precedence modules in the purchasing a website. Initially test planning created based on the testing a website. Test design and test method made by test diagram information. Before developing take a look at scripts the scope of testing need to be documented clearly. At last writing scripts based totally on the precedence then textual content execution and reporting carried out step by step.

सारः

सहफ़्टवेयर परीक्षण दोषों का पता लगाने और किसी भी सहफ़्टवेयर अनुप्रयोग गुणवत्ता को वढ़ाने के लिए एक आवश्यक तकनीक है। अब इंटरनेट में बहुत सारे एप्लिकेशन बनाए जा रहे हैं जो एक इंटरनेट ब्राउज़र में निष्पादित होते हैं। वेव फ़ंक्शंस एक अतिरिक्त जटिल में बदल रहे हैं, जिससे भौतिक रूप से देखना मुश्किल हो जाता है। यह समय और लागत बढ़ा सकता है। इसलिए हमें दिन-ब-दिन उन्नत उपकरणों की जांच के लिए स्वचालन परीक्षण की आवश्यकता है। सेलेनियम एक वेब अनुप्रयोग परीक्षण उपकरण है जो खुला स्रोत है। विभिन्न उद्देश्यों के लिए विकसित वेवसाइट की कार्यक्षमता की जांच के लिए इस उपकरण का व्यापक रूप से उपयोग किया जाता है। इस प्रोजेक्ट में हम सेलेनियम पायथन का उपयोग करके वेब एप्लिकेशन का परीक्षण करते हैं। वेबसाइट खरीदने में परीक्षण प्राथमिकता महड्यूल है। प्रारंभ में परीक्षण योजना एक वेबसाइट के परीक्षण के आधार पर बनाई गई। परीक्षण आरेख जानकारी द्वारा वनाई गई परीक्षण डिजाइन और परीक्षण विधि। स्क्रिप्ट को विकसित करने से पहले परीक्षण के दायरे को स्पष्ट रूप से प्रलेखित करने की आवश्यकता है। अंत में पूरी तरह से पूर्वता पर आधारित स्क्रिप्ट लिखना, फिर पाठ्य सामग्री निष्पादन और रिपोर्टिंग चरणबद्ध तरीके से की गई।

TABLE OF CONTANTS

TITLE	PAGE NO.
Abstract	
सार:	
List of Figures	**2**
Chapter 1: Project Overview	**3-4**
1.1 Introduction	
1.2 Manual Scripting	
1.3 Automatic Scripting	
Chapter 2: Literature Survey	**5**
2.1 Introduction	
2.1.1 Selenium	
2.1.2 Selenium Web Driver	
2.1.3 Pyttsx	
2.1.4 Speech Recongnition	
Chapter 3: Preliminary Design	**6-7**
3.1 Methodology	
3.1.1 Content of Automation Scripts	
3.1.2 Text-to-speech module	
3.1.3 Project Implementation	
Chapter 4: Project Implementation	**8-9**
Chapter 5:Project Requirement	**10**
Chapter 6: Conclusion	**11**
References	**12**

List of Figures

Figure Number	Figure Caption	Page No.
Fig. 3.1	Working of Web Driver	6
Fig. 4.1	Pyttsx	8
Fig. 4.2	Voice Module Code	8
Fig. 4.3	Voice Module Code	9

CHAPTER 1: PROJECT OVERVIEW

1.1 Introduction

Quality engineering is defined as the discipline of engineering which concerns itself with the service quality control and assurance and practices and principles of product. Gaps, errors or missing requirements as opposed to the actual requirements are identified with software testing. It can be either done using automation tools or manually. Browsers can be used to access information and perform a variety of tasks. Browser automation tools for web browsers can perform error prone and repetitive tasks, such as filling long HTML forms. The web driver architecture is shown in Fig 1. Automation tools need to accommodate different skill levels. Sophistication of libraries and scripts depends on whether the automation is being done by a non-programmer or a advanced programmer. The working of web browser automation tools is basically a recording of the steps in series that make up a transaction, and then by injecting JavaScript into the target web pages they play it back, followed by tracking and getting results. Browser automation is a big part of web automation. Its main aim is to mimic how people use web browser to automate scenarios which keep repeating. It usually happens with a computer acting like a human, the process gets fragile and complex. Browser interactions are based on HTML markup, which is very often not composed with this thought in mind. HTML structure changes often. Browsers interact through network inheriting all its failures and unreliability.

1.2 Manual Scripting

Manual scripting is an activity which is performed by testing persons. Manual scripting requires a tester so that to test the function guide check operations without the assist of Test automation .Manual scripting is a process where in a tester often writes the test plans. A test case is a set of conditions that are written for the applications and tester run all the test cases to verify the proper functionality of the software. Manual scripting requires the tester to possess a sure set of qualities; to be patient, observant, speculative, creative, innovative, open-minded, resourceful, opinionated, and skilful. In order to completely check that all the necessities of an software are met, there have to be at least take a look at all instances for every requirement. Manual scripting helps out to find out defects associated to the usability trying out and GUI area. Any new software have to be manually examined earlier than its trying out can be automated. Manual scripting requires greater effort, however is essential to test automation feasibility. Manual scripting may not need the prior knowledge about the testing tool.

1.3 Automation scripting

Automation scripting is nothing but executing the test instances where personal persence is not required. It makes use of exceptional software program to write and execute test instances to examine the proper result with the envisioned outcome. Once tests have been automated, they can be run shortly and repeatedly. Automated software program testing is the pleasant way to make

bigger the effectiveness, effectivity and quality of software program testing. Automation testing might require large quantity of funding for buying the software program and compatible hardware resources. Automation testing improves the accuracy which saves the time of the tester and organization's cost. Automation testing is first-class suitable in the surroundings where the necessities are often altering and large quantity of regression testing is required to be performed. Automation testing is exceptional ideal in the surroundings the place there are fundamental test instances that are to be done repeatedly. It will increase the satisfactory of testing structure and reduces the future upkeep cost. Various advantages of Automation checking out are quick run of test case. Reusable test instances are made and these test instances are reliable, complete and Programmable. The important difference between Manual testing and the Automated testing is that Automation testing is desirable for the surroundings where the Repetitive work is greater (e.g., strolling regression tests, re-entering the identical check data, and checking towards coding standards). Also, manual testing is high-quality desirable for the surroundings where the requirement modifications continuously

CHAPTER 2 : LITERATURE SURVEY

2.1 Introduction

2.1.1 Selenium

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Selenium is a suite of software tools to automate Web Browsers. It is an Open source suite of tools mainly used for Functional and Regression Test Automation. Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using a Selenium tool is usually referred as Selenium Testing. Selenium IDE: Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop Selenium Test cases. Selenium IDE is an easy to use tool from the Selenium Test Suite and can even be used by someone new to developing automated test cases for their web applications. One does not require any special setup to get started with Selenium IDE. You just need to add the extension of your specific browser. Selenium IDE provides you with a GUI (Graphical User Interface) for easily recording your interactions with the website. Selenium IDE allows a user or a test case developer to create the test cases and test suites and edit it later as per their requirements. The development environment also provides the capability of converting test cases to different programming languages, which makes it easier for the user and does not mandate the need for knowing a specific programming language

2.1.2 Selenium WebDriver

WebDriver is an API that allows us to write automated tests for web applications. The automated tests that use Selenium WebDriver are run by using a web browser. In other words, Selenium WebDriver helps us to verify that our application is working as expected when it is used by a real user.

2.1.3 Pyttsx3

Pyttsx3 stands for Python Text to Speech. It is a cross-platform Python wrapper for text-to-speech synthesis. It is a Python package supporting common text-to-speech engines on Mac OS X, Windows, and Linux. It works for both Python2.x and 3.x versions. Its main advantage is that it works offline.

2.1.4 Speech Recognition

This is a library for performing speech recognition, with support for several engines and APIs, online and offline. It supports APIs like Google Cloud Speech API, IBM Speech to Text, Microsoft Bing Voice Recognition etc

CHAPTER 3 :PRELIMINARY DESIGN

3.1 Methodology of Web Automation Using Python

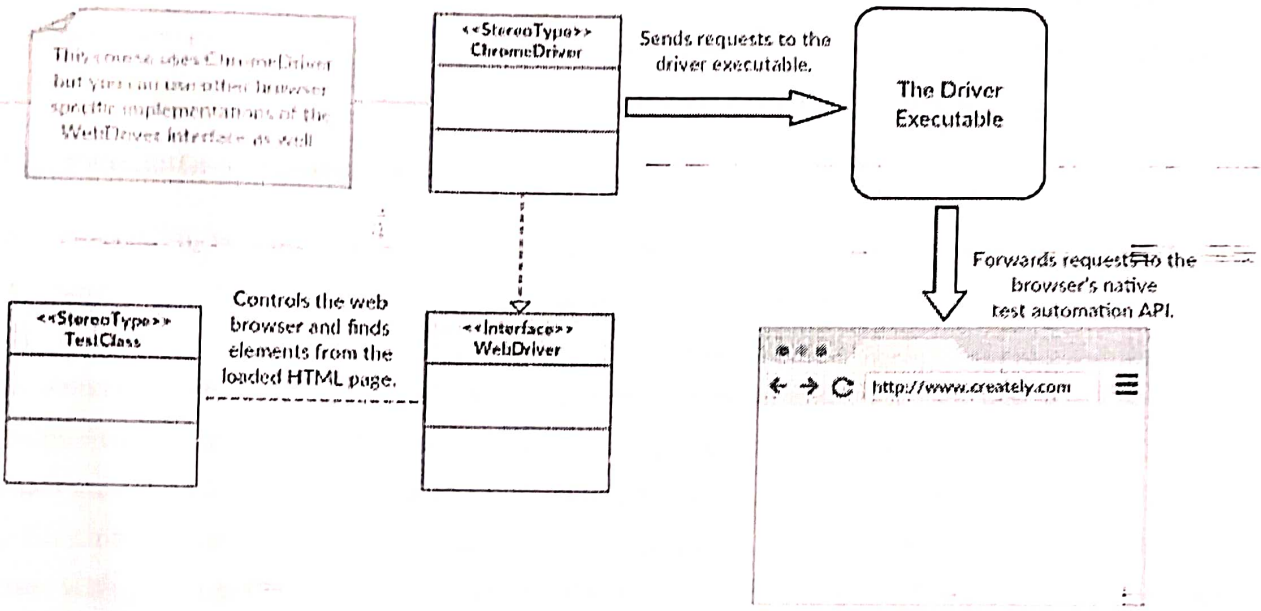


Fig. 3.1 Working of Web Driver

Selenium Web Driver supports many common programming languages and all common web browsers. The driver specific implementation of the Web Driver interface and the driver executable allows us to write automated tests that use the browser's native test automation API. We can get all required dependencies by declaring the selenium-java dependency in our build script. If we want to use only one Web Driver implementation, we don't need the selenium-java dependency. We can simply declare the dependency we want to use.

3.1.1 Content of Automation Scripts

The process of automation combines multiple software and programming languages. A strong sense of object-oriented programming in java is required to use java with selenium to run automation scripts. Most browsers are built using HTML and JavaScript hence requiring a thorough understanding of the same. Selenium, being a powerful tool, supports multiple frameworks like Testing and maven on which frameworks are built. All these elements combine to make up an automation script.

3.1.2 Text-to-speech module Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

3.1.3 Project Implementation

Mostly selenium web driver is been used to detect the errors in application. Selenium web driver can be implemented in any of the available and existing browsers; the only thing is that we have to import files of the browser that we are going to use for testing the particular application. In this any of the programming languages can be used like c#, java etc. The structure of selenium web driver is not so complimented like selenium RC. Multi-browser testing including improved functionality for browsers which are not well supported by Selenium Remote control .Selenium Web Driver makes directly calls to the browser using each browser's native support for automation. This helps in enhancing the test cases with programming techniques to cover all the required checks and test scenarios. In Selenium automation, if the elements are no longer observed via the common locators like id, class, name, etc. then XPath is used to discover an component on the internet web page. X Path is designed to enable the navigation of XML documents with the reason of deciding on individual elements, attributes, or some different phase of an

XML report for any particular processing. Syntax = //tag name[@attribute='Value']

For testing any kind of application there are many varieties of testing tools and types which can be used to test it. In selenium there are four types of components that is used to test an application, they are: -

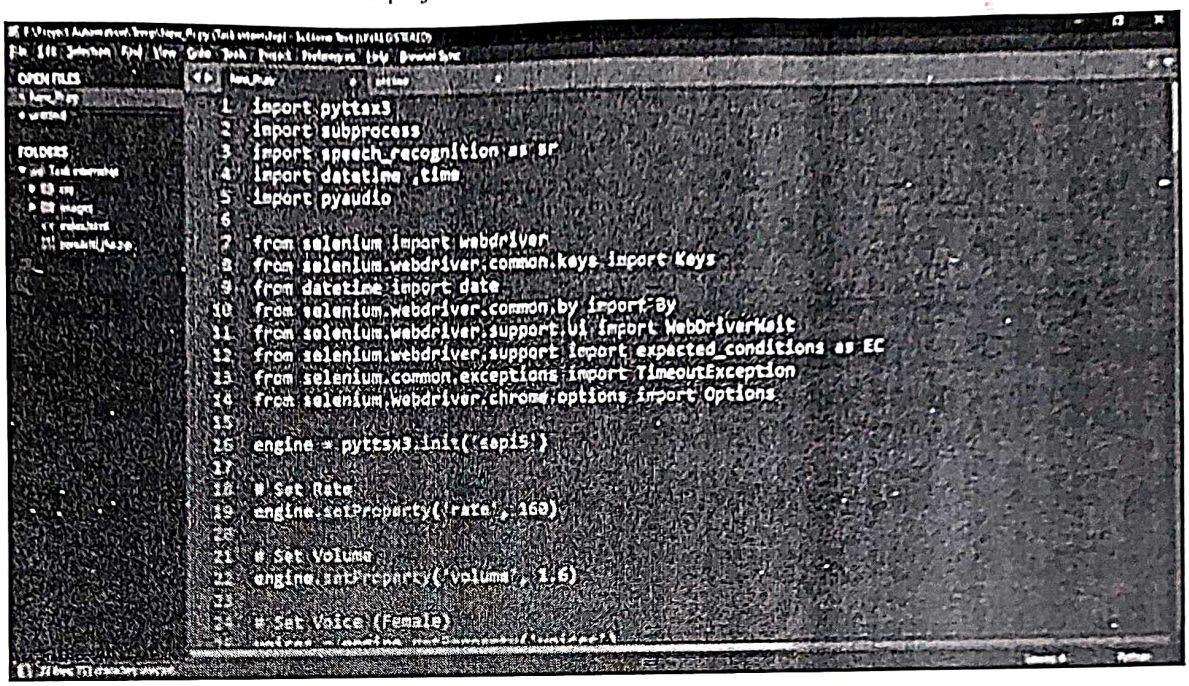
- Selenium Ide
- Selenium RC
- Selenium Web Driver
- Selenium Grid

A. **Absolute X Path :-** It is the direct way to discover the element, however the drawback of the absolute X Path is that if there are any adjustments made in the direction of the element then that X Path gets failed. The key attribute of X Path is that it starts with the single ahead slash(/) , which means you can select the component from the root node. The benefit of the use of absolute is, it identifies the element very fast.

B. **Relative X Path :-** A relative X path is one where the path begins from the node of your choise - it does not want to begin from the root node. It starts with Double forward slash (//).

CHAPTER 4 : Project Implementation

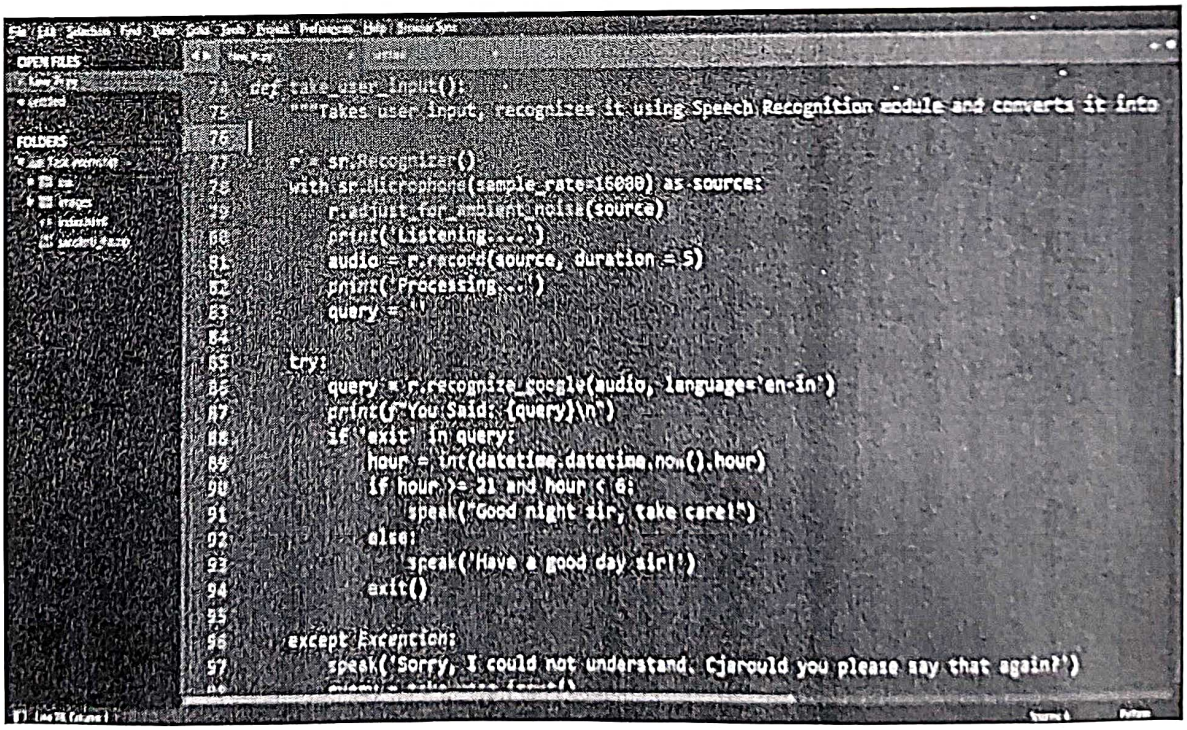
Step 1 :-Go to files and select new project



```
import pyttsx3
import subprocess
import speech_recognition as sr
import datetime ,time
import pyaudio

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from datetime import date
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException
from selenium.webdriver.chrome.options import Options

engine = pyttsx3.init('sapi5')

# Set Rate
engine.setProperty('rate', 160)

# Set Volume
engine.setProperty('volume', 1.6)

# Set Voice (Female)
```

Fig. 4.1 Pyttsx

Step 2:- Voice Module Code



```
def take_user_input():
    """Takes user input, recognizes it using Speech Recognition module and converts it into

    r = sr.Recognizer()
    with sr.Microphone(sample_rate=16000) as source:
        r.adjust_for_ambient_noise(source)
        print('Listening.....')
        audio = r.record(source, duration = 5)
        print('Processing...')
        query = ''

    try:
        query = r.recognize_google(audio, language='en-in')
        print(f"You Said: {query}\n")
        if 'exit' in query:
            hour = int(datetime.datetime.now().hour)
            if hour >= 21 and hour < 6:
                speak("Good night sir, take care!")
            else:
                speak('Have a good day sir!')
            exit()

    except Exception:
        speak('Sorry, I could not understand. Could you please say that again?')
```

Fig. 4.2 Voice Module Code

Step 3:- Then open in new window

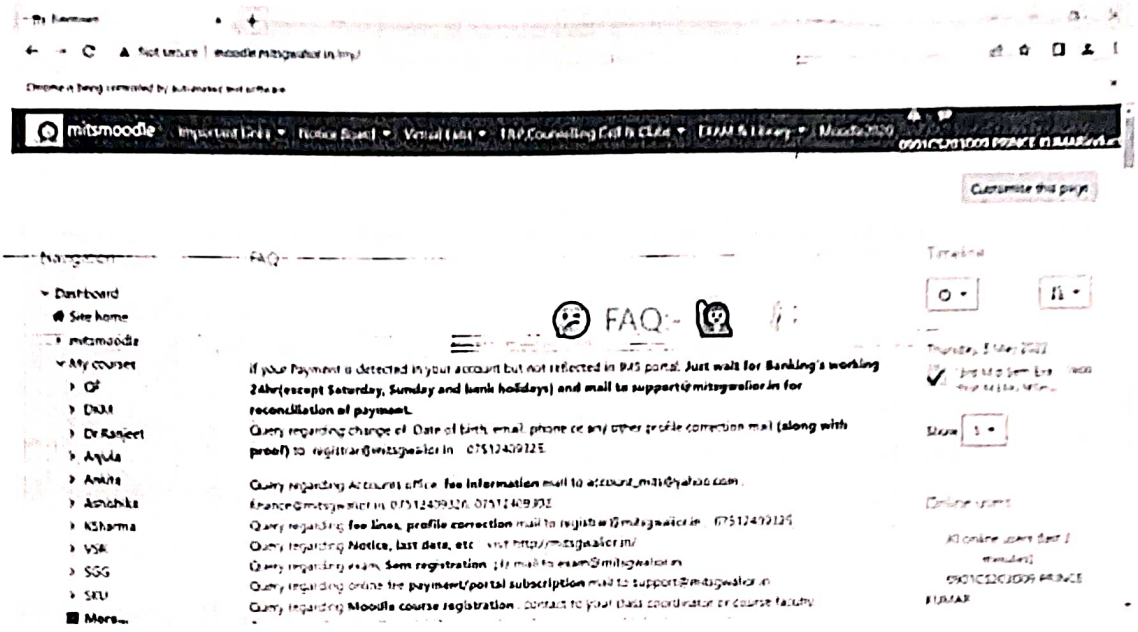


Fig. 4.3 Voice Module Code

4.1 def speak function

The first and foremost thing for an A.I. assistant is that it should be able to speak. To make our virtual assistant 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()
```

What is Web Driver?

- This component is an upgraded version of Remote Control. This is the most applied Selenium tool in automation testing. It utilizes a client API to send commands to web browsers.
- Every browser consists of a unique web driver for running tests. Some of the web drivers for specific browsers include Firefox driver, Safari driver, Opera driver and Internet Explorer driver.
- Some of the programming languages supported by this component include PHP, Java, C#, Javascript, Ruby, Perl, and Python.
- Selenium web driver is also compatible with various browsers such as Opera, Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari. The operating systems supported by the Selenium web driver include Mac, Solaris, Windows, and Linux.

To take command :- To take command from user speech recognition is used

CHAPTER 5 : PROJECT REQUIREMENT

The system requirements to build virtual assistant are given below.

Windows-Based Requirements

Computers running Microsoft Windows must meet the following minimum Hardware and Software requirements.

- Microsoft Windows 8.1 or newer(32- or 64- bit) in window 10 it can run without corrupting any program
- 4 GB RAM minimum, 8 GB RAM recommended;
- 2 GB of available disk space minimum, 4 GB Recommended
- 1280 * 800 minimum screen resolution
- it requires processor i3 7th gen and more for running

5.1 Software requirements

- Python 3.9, Selenium, WebDriver ,

5.2 Hardware Requirements

- Laptop / Computer
- Internet Required.

CHAPTER 6: CONCLUSION

Selenium is a cost-effective and flexible tool developers can use in the automation testing of their web applications.

The most intriguing feature of this software is the ability to test applications across various web browsers. This ensures that websites do not crash or breakdown in certain browsers.

The Selenium software is ideal for companies developing applications that support heavy traffic, especially social platforms and e-commerce websites. This software has undergone improvement over the years, which has added value to web development.

REFERENCES

- R. Belvin, R. Burns, and C. Hein, "Development of the HRL route navigation dialogue system," in Proceedings of ACL-HLT, 2001
- V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T.J.Hazen,and L.Hetherington, "JUPITER: A Telephone Based Conversational Interface for Weather Information," IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, pp. 85–96, 2000.
- M. Kolss, D. Bernreuther, M. Paulik, S. St"ucker, S. Vogel, and A. Waibel, "Open Domain Speech Recognition & Translation: Lectures and Speeches," in Proceedings of ICASSP, 2006.