PERSONAL DIGITAL VOICE ASSISTANT

A PROJECT REPORT

Submitted by,

SOUGATA PATRA (Roll-18010302026),
RITUSHREE DEY (Roll-18010302013),
SHUVRANSHU SOM (Roll-18010302024),
ROHITA KUNDU (Roll-18010302014),
SHOVAN DUTTA (Roll-18010302023)

in partial fulfilment for the award of the degree of

BACHELOR OF SCIENCE (HONOURS) IN COMPUTER SCIENCE

BRAINWARE UNIVERSITY
DEPARTMENT OF COMPUTATIONAL SCIENCE
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

PERSONAL DIGITAL VOICE ASSISTANT

Submitted by

SOUGATA PATRA (Roll-18010302026),
RITUSHREE DEY (Roll-18010302013),
SHUVRANSHU SOM (Roll-18010302024),
ROHITA KUNDU (Roll-18010302014),
SHOVAN DUTTA (Roll-18010302023)

in partial fulfilment for the award of the degree

of

BACHELOR OF SCIENCE (HONOURS) IN COMPUTER SCIENCE

in

DEPARTMENT OF COMPUTATIONAL SCIENCE

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

DEPARTMENT OF COMPUTATIONAL SCIENCE

BONAFIDE CERTIFICATE

Certified that this project report "PERSONAL DIGITAL VOICE ASSISTANT" is the bonafide work of "Sougata Patra **(Roll-18010302026)**, Ritushree Dey **(Roll-18010302013)**, Shuvranshu Som**(Roll-18010302024)**, Rohita Kundu **(Roll-18010302014)**, and Shovan Dutta **(Roll-18010302023)**," who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. SANDIP ROY	**ARPAN KISORE SARBADHIKARI**
HEAD OF THE DEPARTMENT	**SUPERVISOR**

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Our “Personal Digital Voice Assistant” project is based on ‘Python 3.5+’. Speech recognition is the process of converting audio into text, which is the most important part of our project. Python provides an API called ‘SpeechRecognition’ to allow us to convert audio into text for further processing, which in turn allowed us create all the other functions/ features to our assistant. The ‘pyttsx3’ library provided by python, helped us to make our assistant very interactive with the user. ‘Sapi5’ a part of ‘pyttsx3’ library is a Microsoft Text to speech engine used for voice recognition. And the best part is this package supports text to speech engines on Mac os x, Windows and on Linux, which in turn will give us the chance to make our assistant support a lot of popular platforms.

The GUI is based on ‘Tkinter’ a standard Python GUI library, with which we were able to make a very simple, intuitive and cheerful interface for the user. The graphical user interface has two modes i.e. dark mode and light mode to help it seamlessly integrate with modern design of the windows 10 User interface. To further differentiate between the two modes, we have also given the two modes two distinct voices.

Mainly starting with the help of the above mentioned library and many more very popular libraries like ‘os’, ‘time’, ‘subprocess’, ‘word2number’, ‘webbrowser’, ‘Requests’ and many more have enabled us to add very important features like searching / browsing the web, finding news, searching and playing any video on YouTube, opening and closing of any application on the computer, taking notes, controlling the volume and brightness, finding meanings of words, taking screenshots, taking small notes and much more. Which we hope will be very important and helpful for the user. But, the whole experience mainly depends on the internet connection speed and stability of the user as the very core function i.e., speech recognition is done in the cloud.

We tried to achieve a very reliable, fast (*depends on the internet connection speed), secure, intuitive experience for the user. Overall, we expect that users will have a very user-friendly, reliable, hands-free and a new way to interact with their computers.

INTRODUCTION

"Effective", "Elegant" and "Easy" were our desired goal when we started working on this Personal Digital Assistant project. In a pursuit of said goal we designed our personal digital voice assistant to help general computer users by giving them another method of controlling their machine i.e. with their voice. We strongly believe that by efficient use of resources we have achieved our goal of the three "E"s.

A voice assistant or personal digital assistant is a software agent that can perform tasks or services for an individual based on verbal commands i.e. by interpreting human speech and respond via synthesized voices. Users can ask their assistants' questions, control home automation devices, and media playback via voice, and manage other basic tasks such as email, to-do lists, open or close any application etc with verbal commands.

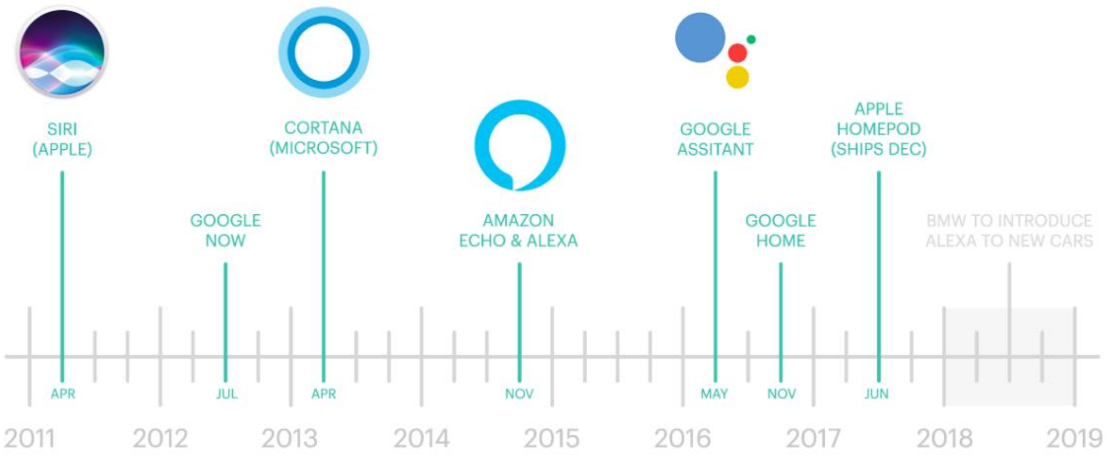


Fig 1: Timeline of Mainstream Voice Assistant

The first personal digital assistant was the Newton Message Pad, developed by Apple and introduced in 1992. Since then, Voice assistants got the major platform after Apple' Siri , Google's Google Assistant, Microsoft's Cortana, etc.

Just like its predecessor our programme can take screenshots, set a timer, take small notes, tell the local weather condition, catch up the user with news in BENGALI,HINDI and ENGLISH, etc. We have used the python programming language for this project. Our programme surpasses its predecessor by allowing us to shut down ,logout and restart the computer, telling the current CPU and RAM status, current battery status as well as if it is charging or not.

The personal digital voice assistant programme started its journey as a windows 10 assistant but with time we were able to make it compatible with windows 8 and 8.1.

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

- **Product Perspective**

The purpose of this project work is to make a Personal Digital Voice Assistant for general computer users by giving them another method of doing their simple and easy tasks on computer i.e. with their voice. People often automate repetitive and simple tasks for future use convenience. So, we aim at developing a program which will in time get powerful enough to perform all the mundane works of a general day to day computer user. All this work for time saving, more intuitive, more reliable, hands free, experience. Till now we believe we have achieved these goals to some extent and hope to improve further with time.

- **Specific Requirements**

The whole of this project work was developed using the Microsoft Visual Studio, were we used the Python language (*and also used many of the popular libraries like 'Tkinter', 'os', 'SpeechRecognition', 'pyttsx3') for the purpose of the development of the project including the Graphical User Interface.
So, the specific requirements for this project work are:-

i. Microsoft Visual Studio
ii. Python
iii. Windows 10 or any compatible Operating System.

As an end product the Personal Digital Voice Assistant developed by us will be implemented or used in desktops, laptops and tablets running Windows and in future also on Linux and Mac os.

- **Performance Requirements**

The 'Personal Digital Voice Assistant' developed is purely based on Python and can be used on a device having Windows 8 and up along with a good internet connection. So, the performance will vary with respect to the machine and the internet connection of the device.

- **Design Constrains**

  i. GUI

  The Graphical User Interface is based on Tkinter. The GUI is very simple to understand for the user.

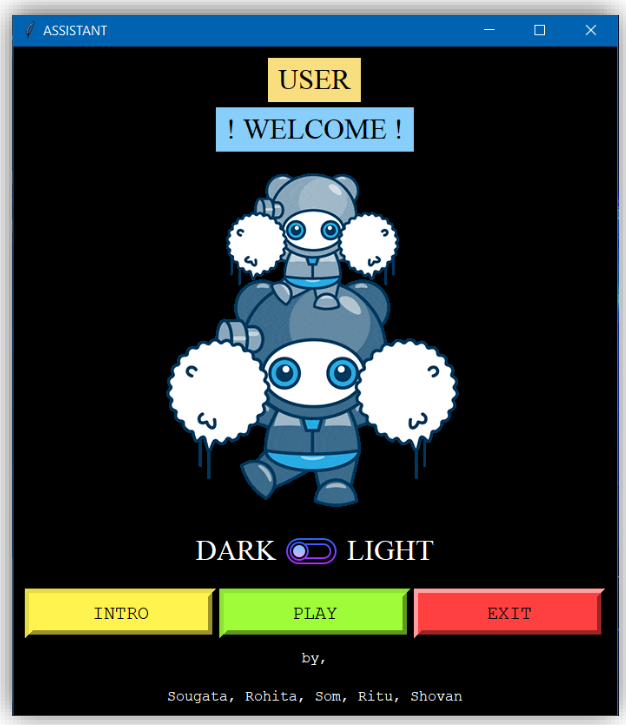


Fig 2: The (Dark mode) User Interface of the PDA

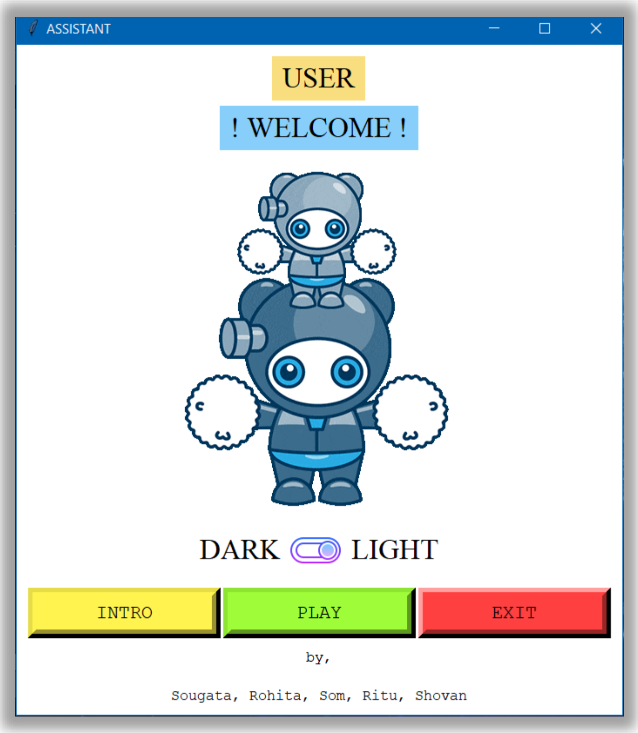


Fig 3: The (Light mode) User Interface of the PDA

The GUI brings one toggle button for changing the theme of the interface and three buttons each having different functions like Self Introduction(*yellow 'INTRO' button in Fig 2 &3), activation / starting the assistant(*green 'PLAY' button in Fig 2&3) and exit (*red 'Exit' button in Fig 2&3). Overall, we tried to make an interface which would blend in with the windows themes and would also be easy and clean for the user to look at.

ii. System

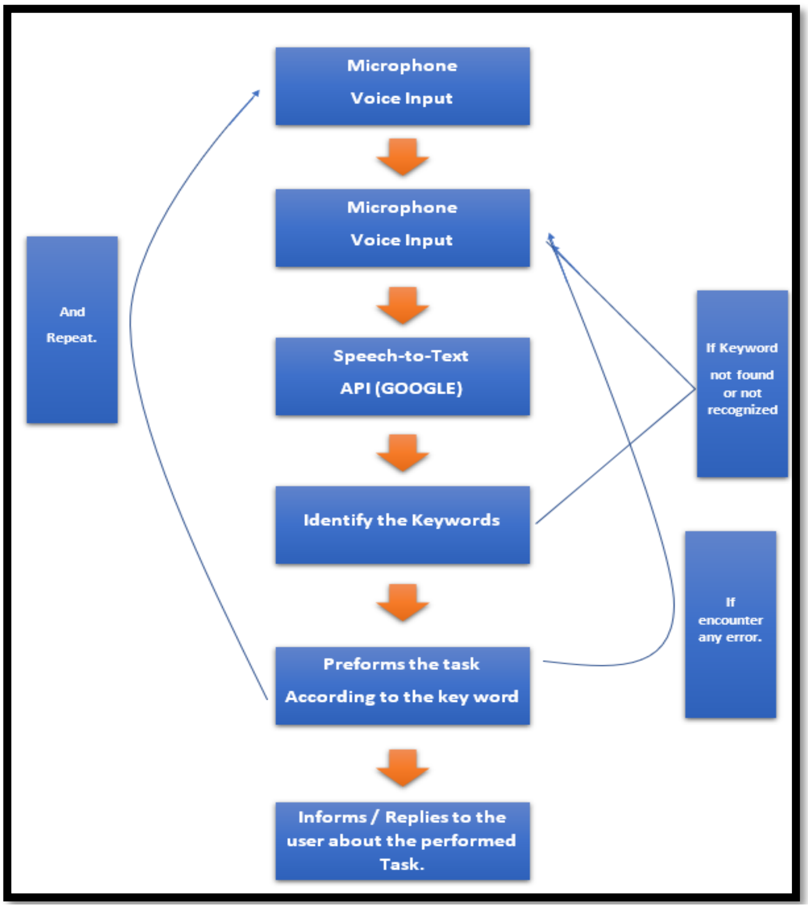


Fig 4: The Data Flow Diagram showing the basic Algorithm.

The overall design of our system consists of the following phases:

(a) Taking input from the user in the form of voice.

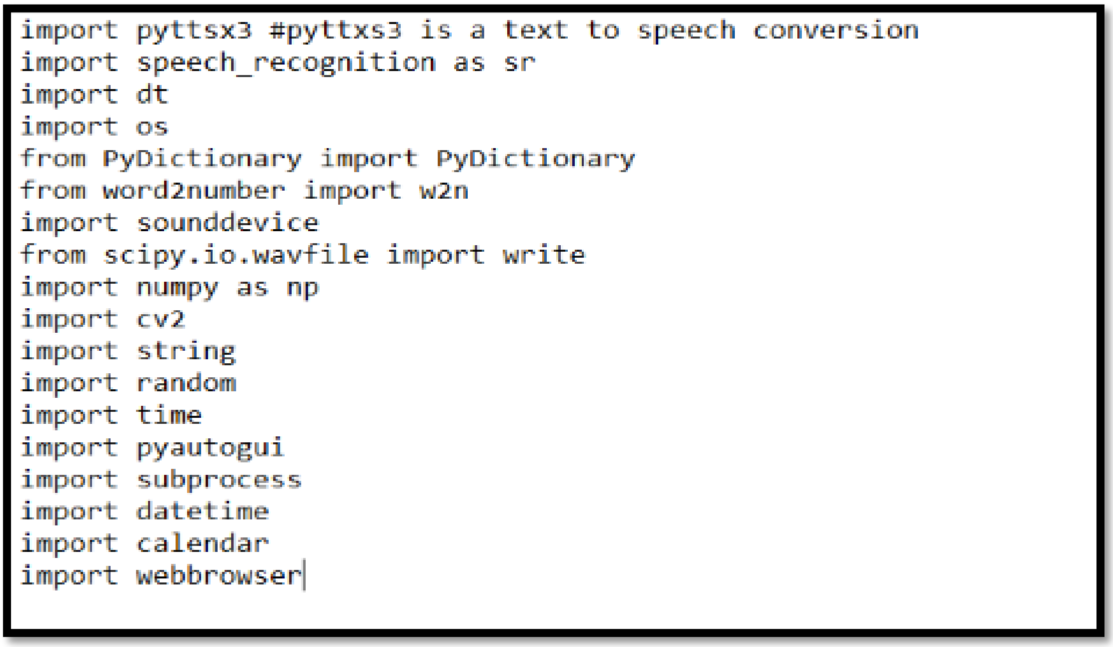
(b) Converting the speech into text to be processed by the assistant.

(c) The converted text is now processed to get the required results.

(d) The text contains one or two keywords that determine what query is to be executed. If the keyword doesn’t match any of the queries in the code then the assistant asks the user to speak again.

(e) The result which is in the form of text is converted to speech again, to give results to the user.

SYSTEM ARCHITECTURE



```
import pyttsx3 #pyttxs3 is a text to speech conversion
import speech_recognition as sr
import dt
import os
from PyDictionary import PyDictionary
from word2number import w2n
import sounddevice
from scipy.io.wavfile import write
import numpy as np
import cv2
import string
import random
import time
import pyautogui
import subprocess
import datetime
import calendar
import webbrowser
```

Fig. 5: Some important imported libraries

- **SPEECH RECOGNITION**

  The speech recognition module used the program is Google's Speech Recognition API which is imported in python using the command "import speech_recognition as sr". This module is used to recognize the voice which is given as input by the user. This is a free API that is provided and supported by Google. This is a very light API that helps in reducing the size of our Application.

- **Text To Speech & Speech To Text**

  The voice which is given as input is first converted to text using the speech recognition module. The text is then processed to give the result of the query given by the user. The final step is the conversion of the result of the processed query to speech which is the final output. The most time consuming among the two is Speech To Text because the system first has to listen to the user and different users have different, some are easy to understand while some are not easily audible. This is the step upon which our total execution time depends. Once the speech is converted to text executing commands and giving the results back to the user is not a time-consuming step.

- **IMPORTED MODULES**

i. PYTTSX3

The pyttsx3 is an offline module that is used for text to speech conversion in Python and it is supported by both Python 2 & 3.

The run and wait functionality is also in this module only. It determines how much time the system will wait for another input or in other words the time interval between inputs. This is a free module available in the python community which can be installed using the pip command just like other modules.

ii. DATETIME

The DateTime module is imported to support the functionality of the date and time. For example, the user wants to know the current date and time or the user wants to schedule a task at a certain time. In short, this module supports classes to manipulate date and time and perform operations according to it only. This is an essential module, especially in tasks where we want to keep a track of time. This module is very small in size and helps to control the size of our program. If the modules are too large or heavy then the system will lag and give slow responses.

iii. WEBBROWSER

This module allows the system to display web-based information to users. For example, the user wants to open any website and he gives input as “Open Google”. The input is processed using the web browser module and the user gets a browser with google opened in it. The browser which will be used is the default set web browser.

iv. WIKIPEDIA

Wikipedia is a library in python which it possible for the virtual assistant to process the queries regarding Wikipedia and display the results to users. This is an online library and needs an internet connection to fetch the results. The no. of lines that the user wants to get as a result can be set manually.

v. OS

OS Module provides operating system dependent functionalities. If we want to perform operations on files like reading, writing, or manipulate paths, all these types of functionalities are available in an OS module. All the operations available raise an error “OSError” in case of any error like invalid names, paths, or arguments which may be incorrect or correct but just no accepted by the operating system.

vi. SUBPROCESS

This module is used for getting system subprocess details which are used in various commands i.e. Shutdown, Sleep, etc. This module comes built-in with Python.

SYSTEM FEATURES

The application features are:-

1. Takes voice commands to perform operations.

   Using the 'speech recognition' package of python to interpret voice commands.

2. Fetch news and show them in browser.

   Can fetch English, Hindi & Bengali news on command in the browser from reputed news channels.

3. Has an interface where the user can interact.

   Has an intuitive GUI where user can interact with the assistant.

4. Tell the current weather condition.

   Can tell the current weather condition of the user's current location.

5. Stop or close and prevent listening for a period of time (i.e., sleep mode)

   Can stop or prevent listening for a period of time on user command rather than having to close the application repeatedly when not needed, thus saving time for re-launching.

6. Do normal conversations with the user. (*limited)

   Can converse with the user.

7. Toss a coin or roll a dice.

   Can toss a coin or roll a dice on command.

8. Light Mode and Dark Mode

   To properly blend in with the Windows 10 OS there is a built in light and dark mode along with a female and male voice respectively.

9. Tell users current location.

   Can tell the user's current location on command to the specificity of "city", "state", "country".

10. Take screenshots.

Can take screenshots on command and save them in the directory “C:\Users\Public\Pictures” (i.e., the public pictures folder) which is constant and available to every user on every version of windows (*Windows 8 & Up).

11. Open any application installed on the computer.

Can open any application (including system and 3rd party applications) which are already installed on the computer on command.

12. Close any currently open application on the computer.

Can close any application that is running/open on the computer on command.

13.Take small notes.

Can take small notes and save them as .txt file in the directory “C:\Users\Public\Documents” (i.e., the public documents folder) which is constant and available to every user on every version of windows(*Windows 8 & Up).

14. Shut down, restart, logout the computer and cancel the shutdown of the computer.

Can shut down, restart, logout the computer and cancel the shutdown of the computer (*with 60 seconds notice) on command of the user.

15. Find Meanings, Synonyms, Antonyms of words.

Can find Meanings, Synonyms, and Antonyms of a word separately on command. (* won’t be stopped until all the Meanings, Synonyms and Antonyms of a word are being told to the user)

16. Open some particular webpages in the browser

Can open sites like google, Wikipedia, YouTube, social media sites, online shopping sites and many more in the browser on command.

17. Tell the current battery status.

Can tell the current battery percentage as well as can say if it’s charging or not. (*only for laptops)

18. Search for summery or definition from Wikipedia, search for videos or play songs in YouTube and can search normally in google.

Can search for summery or definition from Wikipedia, search for videos or play songs in YouTube and can search normally in google on command.

19. Make voice recordings.

Can make voice recordings on command and save them in the directory “C:\Users\Public\Music” (i.e., the public music folder) which is constant and available to every user on every version of windows (*Windows 8 & Up).

20. Control Volume

Can increase, decrease or mute volume using voice commands.

21.Help user to set alarm/timer.

Can help user to set timer/alarm on command.

22. Tell the current system status.

Can inform user about the current CPU and RAM usage on command.

23. Control screen brightness

Can increase or decrease the monitor / screen brightness.

24.Wishes every time when it’s triggered.

Wishes the user depending on the time of the day and also tells the day, date &time while wishing.

25. Tell the day, date, time & weather with one command.

With commands like “brief me” & “what’s up” the assistant can inform user about the day, date, time & weather.

FUTURE SCOPE

The virtual assistants which are currently available are fast and responsive but we still have to go a long way. The understanding and reliability of the current systems need to be improved a lot. The assistants available nowadays are still not reliable in critical scenarios. The future of these assistants will have the virtual assistants incorporated with Artificial Intelligence which includes Machine Learning, Neural Networks, etc. and IoT. With the incorporation of these technologies, we will be able to achieve new heights. What the virtual assistants can achieve is much beyond what we have achieved till now. Most of us have seen Jarvis, that is a virtual assistant developed by iron man which is although fictional but this has set new standards of what we can achieve using voice-activated virtual assistants.

CONCLUSION

In this documentation we have discussed a Voice Activated Personal Assistant developed using python. This assistant currently works online and performs basic tasks like weather updates, stream music, search Wikipedia, open desktop applications, etc. The functionality of the current system is limited to working online only. In future this assistant will have machine learning incorporated in the system which will result in better suggestions with IoT to control the nearby devices similar to what Amazon’s Alexa does. The usage of the assistant will get offline also for features that don’t require an internet connection.

ACKNOWLEDGEMENT

We deem it a pleasure to acknowledge our sense of gratitude to our project guide Prof. ARPAN KISORE SARBADHIKARI under whom we have carried out the project work. His incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish to reciprocate in full measure the kindness shown by Dr. SANDIP ROY (H.O.D, COMPUTATIONAL SCIENCE) who inspired us with his valuable suggestions in successfully completing the project work.

We are also very thankful to Prof. Raktim Kumar Dey and Prof. Soma Mitra for their valuable advices about our project.

Finally, we must say that no height is ever achieved without some sacrifices made at some end and it is here where we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.

Date:

Place:

SOUGATA PATRA (Roll-18010302026),

RITUSHREE DEY (Roll-18010302013),

SHUVRANSHU SOM (Roll-18010302024),

ROHITA KUNDU (Roll-18010302014),

SHOVAN DUTTA (Roll-18010302023)

REFERENCES