**Project Report: Desktop Voice Assistant**

**1. Project Overview**

This project involves the design and implementation of a **Desktop Voice Assistant** capable of performing essential tasks based on user voice commands. The assistant aims to improve productivity and provide hands-free control of the computer.

**2. Objectives**

* To implement a voice-controlled desktop assistant.
* To process and respond to user voice commands.
* To perform the following tasks:
  + Voice typing (speech-to-text).
  + Tell the current time.
  + Open websites.
  + Launch installed applications.
  + Search topics using Wikipedia.

**3. Tools and Technologies Used**

| **Technology** | **Purpose** |
| --- | --- |
| Python | Core programming language |
| speech\_recognition | Voice input processing |
| pyttsx3 | Text-to-speech conversion |
| datetime | Get current system time |
| webbrowser | Open websites |
| subprocess / os | Launch system applications |
| wikipedia | Fetch summary of search topics |
| keyboard (optional) | For voice typing |

**4. Functionalities Implemented**

**4.1 Voice Recognition**

Uses a microphone to capture user speech and converts it into text using Google Speech API.

**4.2 Text-to-Speech**

The assistant responds verbally using the pyttsx3 library.

**4.3 Time Reporting**

Tells the current time based on system clock.

**4.4 Opening Websites**

Interprets commands to open websites in the default web browser (e.g., "open YouTube").

**4.5 Launching Applications**

Launches predefined desktop applications using subprocess.

**4.6 Wikipedia Search**

Provides summaries of requested topics using the Wikipedia module.

**5. System Architecture**

**User** ⟶ Microphone Input ⟶ Speech Recognition ⟶ Command Parser  
⟶ Action Execution ⟶ Response via Text-to-Speech

**6. Sample Commands**

| **Command** | **Function** |
| --- | --- |
| “What time is it?” | Tells the current time |
| “Open YouTube” | Opens YouTube in browser |
| “Launch Chrome” | Opens Google Chrome |
| “Wikipedia Python” | Reads a summary about Python |
| “Start typing” | Begins typing dictated text |

**7. Conclusion**

The desktop voice assistant successfully meets the primary goals of recognizing and responding to voice commands. It can be further improved with more natural language understanding, context awareness, and integration with external APIs (like weather, music, or calendar events).

**9. Screenshots**

