**Shujaat Ali Hashim Bs Computer Science**

**2022f-mulbscs-093 Lab – Project**

**Project: Voice Based Virtual Assistant**

**Note : Before running the program make sure to install all the required python libraries .**

**Overview**

This program is a graphical user interface (GUI) based virtual assistant built using PyQt5.It represents a modern approach of assistance required when operating a pc. It integrates speech recognition, web automation, and AI-driven conversation features. The assistant responds to user commands, searches the web, plays videos, and provides time and date information.

**Components and Modules**

1. **PyQt5 GUI Components:**
   * QMainWindow: Main application window.
   * QPushButton: Buttons for microphone, AI response, and accent change.
   * QLabel: Labels for displaying text and assistant logo.
   * QIcon and QPixmap: Used for icons and images.
2. **Custom Modules:**
   * record\_module.Record: Handles recording and storing of user queries.
   * web\_driver\_module.WebDriver: Manages web searches and automation.
   * conversation\_module.conversation: Manages spoken interactions with the assistant.
   * ai.ai\_convo: Processes AI-based responses.

**Initialization**

The MainWindow class inherits from QMainWindow and multiple custom modules. The \_\_init\_\_ method initializes:

* The GUI layout with buttons and labels.
* The voice state (is\_male\_accent, is\_listening, ai\_speaking).
* The Record and WebDriver modules.
* Microphone, accent, and assistant logo with icons.
* Styles for buttons and labels.

**Functionality**

**1. Changing Accent**

The change\_accent method toggles between male and female voices. It updates the icon, logo, and plays a sample introduction.

**2. Toggling Listening Modes**

* **toggle\_speaking**: Activates or deactivates user voice input.
* **toggle\_ai**: Enables AI-driven conversation mode.

**3. Listening to Commands**

The start\_listening method continuously listens for user commands when activated. Based on recognized text, it executes relevant actions:

* **Web Search:** Recognizes "search" or "play" commands and fetches information via WebDriver.
* **Website Launching:** Recognizes "launch" commands to open websites.
* **Time and Date Retrieval:** Responds with current time and date.
* **Stop Command:** Stops the assistant’s listening mode.

**4. AI Conversation Mode**

When ai\_speaking is enabled:

* The assistant listens for input.
* It processes the response using ai.ai\_convo.
* The response is displayed on the UI and spoken aloud.
* The conversation log is saved using Record.

**Execution**

The main() function initializes the QApplication, creates the main window, and runs the event loop until the program exits.

**Conclusion**

This program effectively combines GUI elements, voice interaction, and web automation to create an interactive virtual assistant. It listens, processes user commands, fetches information, and engages in AI-driven conversations.