1. Title	***SAHAYAK***
2. Team Name	***Codaemons***
3. Team Members	1) Abhay Raj Yadav(E22CSEU1276) – worked of **Tkinter module**(front-end) 2)Mohd Adnan Hasan(E22CSEU1269)- developed **voice assistant**(back-end) 2) Vanipalli Rahul Reddy (E22CSEU1274)-integrated **weather** and **email** function. 4) Adhyay Bansal (E22CSEU1272)- developed **music function** of sahayak. 5)Aryan Jain(E22CSEU1285)- worked on **Google trans** module 6)Shubh Sonakiya(E22CSEU1290)- integrated **general queries** of sahayak
4. Abstract	**Introduction/Background:** The project aims to develop a personal-assistant for windows PC. Sahayak draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. **Objectives:** Users can interact with the assistant either through voice commands or using keyboard input. As a personal assistant, Sahayak assists the end-user with day-to-day activities like general human conversation, searching queries in google, searching for videos, live weather conditions, word meanings. The user statements/commands are analyzed with the help of machine learning to give an optimal solution. **Methods:** Our project Sahayak uses users voice inputs and performs tasks accordingly with the help of different inbuilt modules in python. Pycharm IDE provides many libraries for taking voice inputs, converting them into strings and then has functions to perform specific tasks. **Keywords:** Personal Assistant, Artificial Intelligence, Machine learning, Task Automation.
5. Introduction, Review of Literature and Background	In today's world use of artificial intelligence are in abundance, Human interaction with chat bots and AI assistants are very fascinating. We people are in need of these applications like Google assistant, Amazon alexa, Siri. Our project Sahayak comes into picture to interactively answer user queries and make this process more indigenous. The instructions can be processed according to the user's needs. Python has an API called Speech Recognition that can be used to convert speech to text. Making my own assistant was an interesting task. Send an email without typing a word, search Google without opening a browser, play music with a single voice command, open your favorite IDE, and many other everyday tasks. It's now easier to perform common tasks. In the current scenario, advances in technology allow us to perform any task with equal or greater effectiveness than we do. By doing this project, I realized that the concept of AI can reduce human effort and save time in all areas.
6. Objectives: Project Aim	The voice assistant uses artificial intelligence to give you highly accurate and efficient results. Assistants help reduce human effort and wasted time in completing tasks. He got rid of the concept of typing entirely; acting like another person we were talking to and asking us to perform a task. An assistant is no worse than a human assistant, but better at performing any task. It is said to be more effective and efficient. The libraries and packages used to build this program focus on time complexity and save time.

Features include: can send emails, can open favorite IDE, notepad, can play music, etc. You can search and run Wikipedia, open websites such as Google and YouTube in your web browser, view the weather forecast, and have your selections delivered to your desktop and can have a basic conversation with Sahayak.

7. **Methods and Materials**

Study/Project Design:

The system was developed with the help of the necessary Python packages following the concept of artificial intelligence. Data for this project is user input only. No matter what the user says, the wizard will perform the task accordingly. User input is nothing concrete, but a list of tasks that the user can perform in human language English, I want to be executed.
The IDE used in this project is PyCharm. All Python files were built with PyCharm and all required packages were easy to install in this IDE. Python provides many libraries and packages to accomplish this task. The following modules and libraries were used in this project:

Following modules were used in our project.

- **Tkinter** for front-end window
- **pyttsx3** for text to speech conversion
- **speech_recognition** for voice inputs
- **Wikipedia** for queries
- **Os** for developing path
- **Pyjokes** (jokes module)
- **Wolframalpha** (general queries)
- **Smtplib** (emails)
- **Webbrowser** (URL, websites)
- **Datetime** (time)
- **PIL Image, PhotoIMage** (images)
- **Google trans , GT3** (translator, API)
- **Pyaudio, Playsound** (play audio)
- **Pygame, pickle , Mutagen** (Music)
- **Requests,ssl** (https)

Hardware Specification:

Microphone , speaker, CPU, Display, voice terminal

Outcome Measures:

With the advancement SAHAYAK can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time. Functionalities of this project include, It can send emails,It can open command prompt, your favorite IDE, notepad etc., It can play music, It can do Wikipedia searches for you, It can open websites like Google, YouTube, etc., in a web browser, It can give weather forecast. It can have some basic conversation.

Data Collection and Analysis: (if Required)

Input through voice commands, used APIs for online search of data according to that

	input. **Drawbacks and the Limitation:** ✚ Due to no voice command encryption, security is somewhere an issue in Shayak. ✚ Background noises often interfere with user voice input, making it difficult for assistant to catch the phrase. ✚ Users' accents causes misinterpretation which alters the actual results. ✚ Our assisstant Sahayak doesn't have a wake word like 'alexa' , 'Hey Siri', 'OK google', etc. ✚ Right now it has just few conversation content. **Conclusion:** Making my own assistant was an interesting task. Send an email without typing a word, search Google without opening a browser, play music with a single voice command, open your favorite IDE, and many other everyday tasks can now be performed easily. Unlike other traditional voice assistants, Sahayak is desktop-specific, requiring users to not create an account to use it and relying on an Internet connection while receiving instructions to perform specific tasks. Sahayak Desktop Assistant is a voice assistant that allows us to perform many everyday desktop tasks like playing music or launching his favorite IDE with a single voice command.
8. Potential Impact	SAHAYAK is undoubtedly a very useful voice assistant that saves users time due to its conversational interaction, its effectiveness and efficiency. User feels like he or she is assigning the task to a human assistant with conversational interactions to provide input and obtain the desired output in the form of a completed task. Desktop Assistant is responsive which means it knows human language well, understands the context you provide, and responds in the same way In a human understandable language, English. So users find their reactions in an informed and intelligent way. The main use for this is its multitasking capabilities. Just ask for instructions, listen for a response from the user and perform the task.
9. Future Plans	• Make Sahayak self-learning so that we can train it and develop new skills. • Sahayak android app can also be developed. • Filter user clear voice command from the surrounding noise. • Integrate Sahayak in more voice terminals. • Encrypt voice commands to maintain security. • Increase the content for conversation. • Have some fun activities.
10. References	• **Codemy.com, GeeksforGeeks.com , Wolframalpha.com**