AI Powered Voice Pair Programmer

1st Nithyakumar. M
dept. Electronics and Communication Engineering
Dr.Mahalingam college of Engineering and Technology
Pollachi, Tamilnadu,India
nithyakumar01@gmail.com

2nd Murugappan. M.M
dept. Electronics and Communication Engineering
Dr.Mahalingam college of Engineering and Technology
Pollachi,Tamilnadu,India
murugappankrishna@icloud.com

3rd Saamsundar. K.V
dept. Electronics and Communication Engineering
Dr.Mahalingam college of Engineering and Technology
Pollachi, Tamilnadu,India
saamanand7@gmail.com

4th Premgugan. P
dept. Electronics and Communication Engineering
Dr.Mahalingam college of Engineering and Technology
Pollachi,Tamilnadu,India
premgugan007@gmail.com

Abstract— **This project introduces an innovative solution to software development by harnessing the power of Artificial Intelligence (AI). The AI-Powered Voice Pair Programmer can be used to revolutionize software development by the power of AI. It also enhances collaboration and productivity in software development by facilitating voice-controlled pair programming. The system employs advanced speech recognition and natural language processing technologies to assess users' language skills or coding abilities and provide necessary assistance. It offers features such as code completion suggestions, syntax corrections, and contextual assistance, enhancing the efficiency and accuracy of coding tasks. Additionally, it allows developers to foster a collaborative environment. By leveraging AI, we can overcome traditional barriers of software development. However, it's crucial to address challenges related to privacy, security, and algorithmic biases to ensure the ethical and equitable use of these technologies. This project aims to create a dynamic, interactive, and personalized learning and working environment, revolutionizing software development collaboration.**

Keywords— AI-Powered, Voice, Pair Programmer, Software Development.

I.INTRODUCTION

The internet is full of information that can be accessed and used but to find the necessary data or solution that we are looking for particular problem is challenging.But now it all changed since the advent of AI we simply input what we need into specific AI models that are trained using Terabytes and Petabytes of data and we can get the answer to our problem.

To program these AI model still human programmers are needed and they work individually have to manually refer the internet or ask fellow programmers to find answers. In this type of environment's, the pair programmer method of programming is a very useful method that has been practiced in the programming industry for years. This increases the efficiency of programming. In this method of programming the efficiency increases by 40% according to a study.

Yet, this method also comes with its own set of disadvantages. Such as two people ding a single person's job. Here we combine the concept of AI and pair programming Through voice as a medium to bridge the gap in the access of information and eliminate the disadvantages by focusing the remaining manpower on another task. After the integration of AI in the concept of pair programming. The efficiency of programming increased to 15% more reaching 55%. The efficiency of the program is also increased with the assistance of AI.

At the core, AI Powered Voice Pair Programming is the fusion of the AI Technologies with Natural Language Processing (NLP) and voice recognition capabilities. This tech enables programmers to interact with the AI pair programmer with commands that are spoken aloud. Through voice commands the programmer the programmer can collaborate with the AI Pair Programmer in real-time therefore enhancing productivity.

Moreover, AI pair programmers can access and scour through vast amount of data to find a solution to our problem and they also reduce the risk of human error. It can provide seamless collaboration between developers, without any regard for their level of expertise.

In summary, voice pair programming enabled by AI has the potential to completely alter software development. Regardless of experience or skill level, this technology makes it possible for engineers to work together more successfully by integrating AI and NLP. This technology allows us greater access and usage of information. Following this introduction, the paper is structured in the following manner. Section 2 deals with works related to AI, Chatbots and Pair Programming. Section 3 addresses the existing system that is similar to AI Powered Voice Pair Programmer. Section 4 focuses on the System description of AI Powered Voice Pair Programmer. Section 5 handles the experimental results of the project. Section 6 covers the conclusion of the project and Section 7 contains the references.

II. RELATED WORKS

1. Proposal for Healthcare Chatbot Systems: This section introduces an innovative application of AI in the healthcare domain, outlining the creation of a chatbot system designed exclusively for medical inquiries. The system uses advanced AI algorithms and Natural Language Processing (NLP) approaches to offer correct responses to user inquiries while conforming to healthcare norms and laws. The architecture contains a powerful database for storing conversational data, allowing for quick retrieval of important information for user interactions.

2. An Overview of Chatbot Development Trends: This section delves into current chatbot development trends, with a focus on the integration of artificial intelligence technologies. It delves deeply into AI and Natural Language Processing (NLP), as well as the platforms and methods required for chatbot development and maintenance. Furthermore, it emphasizes chatbots' considerable impact on industries such as education and e-commerce, concentrating

on their cost-effectiveness and capacity to meet customer service demands.

3. An Investigation into Pair Programming: The part explains pair programming, which is a collaboration between two programmers on a single project, usually with an experienced and a less experienced programmer. This cooperative technique has been demonstrated to increase working productivity by 25% and test case success rates by 20% when compared to individual programming efforts. An investigation done at the University of Utah found that the majority of participants preferred pair programming.

4. Comparison of Human-Human and Human-AI Pair Programming: This part compares traditional human-human pair programming with the emerging trend of human-AI pair programming. Various parameters, including quality, productivity, satisfaction, learning, and cost, are evaluated. However, conclusive evidence regarding the effectiveness of human-AI pair programming remains inconclusive based on research findings.

5. Explanation of Chatbot Testing Tools and Methods: This section discusses the tools and procedures available for testing chatbots. It focuses on planning-based automated testing frameworks designed for chatbot applications. These frameworks seek to simplify the testing process while maintaining the dependability and robustness of chatbot functionality.

III. EXISTING SYSTEM

The existing system is a conversational virtual assistant or chatbots for healthcare system that uses AI and acts like a forum, where one post questions and any user of the forum answers the questions. This chatbot uses the same method in which a user posts a question and the question is answered by an expert in this a doctor. After that the answers and the questions are saved in the database and then the answer is replicated when a similar kind of question is asked.

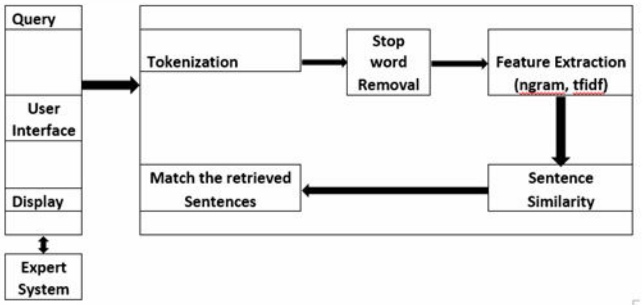


Fig 1. System Architecture

The user inputs the query as text. The chatbot gets the user query and some pre-processing is done. The pre-processing includes Tokenization, Removal of Stop Words, Feature Extraction and TF-IDF and Cosine likeness. All of the conversational exchanges using text are saved in a database to be recovered when needed.

Tokenization: It's the process of breaking up the text into words, typically referred as tokens. These are the elementary components which is used in the Natural Language Processing (NLP). In the tokenization process the words are separated with specified delimiters as references. The tokens are then analysed using NLP to get information from the tokens. This process is the first and very basic process in NLP.

Removal of Stop Words: The very simple example of Stop Words are the words like "the" and "is" that are removed during the pre-processing to collect meaningful information from the text. These words occur in daily language usage, but are not important for machines and AI, they remove these types of words to simplify and make the NLP tasks like analysis and classification of information efficient and accurate

Feature Extraction: It involves both N-gram and TF-IDF to analyse the frequency and importance of a word using its frequency in text. The N-gram technique can use to identify the next sequential words and TF-IDF assigns weights to these N-grams, these weights indicate the significance of the words.

The formula below is used to calculate the frequency of a particular word in a document. Here, tf is term frequency.

$$\mathbf{tf = tfi}$$

IDF denotes the rarity of a term, across the entire collection of documents. Here, idf is inverse document frequency.

$$\boldsymbol{idf} = \mathbf{log\ N/}\boldsymbol{df}$$

Using the above values of tf and idf we can calculate the term or word in the document. The formula is tf multiplied by idf is equal to the weight of each term in a document.

$$\mathbf{W\ i = tf\ i * log\ N/df}$$

N-gram: It's used for predicting and suggesting the next word. N-gram is mainly used in tasks like speech recognition and text generation.

Sentence similarity: It simply means that checking the similarity between two sentences. Here, Cosine similarity is used for this process. The number of query weights is directly proportional to the similarity between query and the document. It ranges from 0 to1.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Fig 2. Formula for Sentence Similarity

Answer display: After the query and the sentence is matched. The answers for the query which is already given to the query is displayed due to its similarities.

IV. SYSTEM DESCRIPTION

The proposed AI-Powered Voice Pair Programmer is a system designed to enhance developer productivity and collaboration during coding sessions. It has voice interaction capabilities, providing real-time coding suggestions, syntax checks, and debugging assistance. The system supports pair programming sessions, facilitating smooth collaboration between developers while offering personalized recommendations tailored to individual coding styles. It employs AI algorithms for intelligent code analysis, pattern recognition, and error prediction, ensuring enhanced code quality. Overall, the system empowers developers with intelligent coding assistance, leading to increased efficiency, reduced errors, and improved collaboration in software development projects.

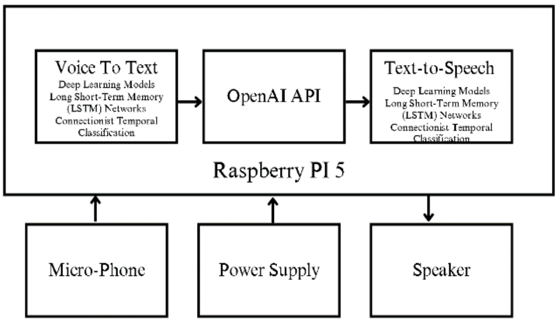

Voice To Text
Deep Learning Models
Long Short-Term Memory (LSTM) Networks
Connectionist Temporal Classification
OpenAI API
Text-to-Speech
Deep Learning Models
Long Short-Term Memory (LSTM) Networks
Connectionist Temporal Classification
Raspberry PI 5
Micro-Phone
Power Supply
Speaker

Fig 3. Block Diagram of the System

We have implemented the AI-powered voice pair programmer as a webpage and as a hardware system using raspberry pi 4.

The basics process involved in this process is getting the input as voice and transcribing the voice and giving it to our AI powered voice pair programmer, then we will get the required answers for the questions and the final output is received in voice.

First, we will discuss about the implementation of the project as a webpage. The basic idea of the project remains unchanged the HTML, CSS and JavaScript is used to display the front end or the interactive part of the webpage. We need a webserver to host the webpage. For this purpose, we use NGINX Web Server which is a high performance open-source webserver which is efficient in serving static content such as HTML, CSS, JavaScript, images and videos of smaller sizes making it perfect for hosting websites. The processes which are taking place in the NGINX webserver is master process and worker process. As the name suggests the master process is responsible for managing the worker processes and in both signal handling and configuration loading. While, the worker processes handle connections from the client and execute the requests independently of each other thus allowing NGINX to handle large number simultaneous connections effectively. The next step is to find the IP address of the raspberry pi 4. After the IP address is found it is entered in the local network so that the content, we wish to display can be viewed from any of the devices which have access to the IP address of the raspberry pi 4. The mic and the speaker required for the implementation of the webpage will be the default mic and speaker of the device which accesses the webpage.

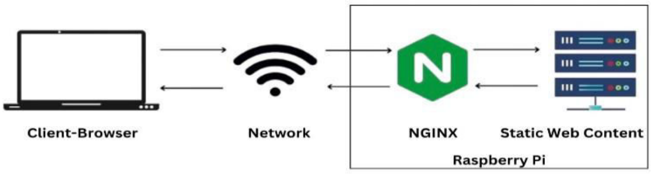
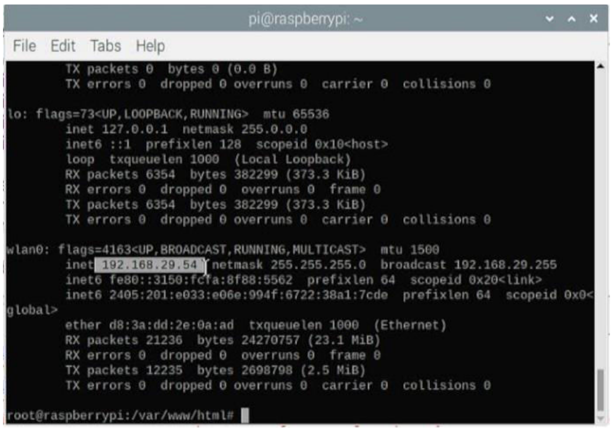


Fig 4. Block diagram of implementation of Webpage



```
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6354  bytes 382299 (373.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6354  bytes 382299 (373.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.29.54  netmask 255.255.255.0  broadcast 192.168.29.255
        inet6 fe80::3150:fcfa:8f88:5562  prefixlen 64  scopeid 0x20<link>
        inet6 2405:201:e033:e06e:994f:6722:38a1:7cde  prefixlen 64  scopeid 0x0<
global>
        ether d8:3a:dd:2e:0a:ad  txqueuelen 1000  (Ethernet)
        RX packets 21236  bytes 24270757 (23.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12235  bytes 2698798 (2.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@raspberrypi:/var/www/html# 
```

Fig 5. IP address of the Raspberry PI

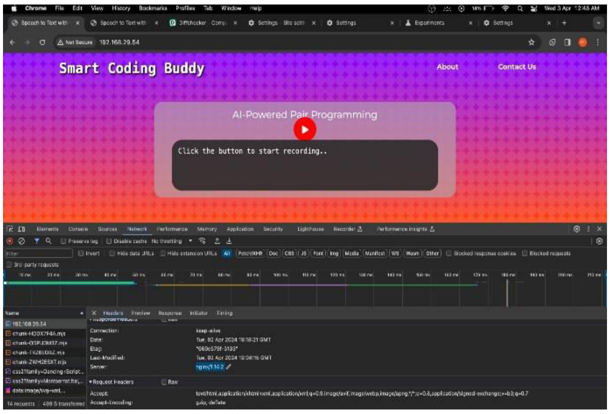

Not Secure 192.168.29.54
Smart Coding Buddy
About
Contact Us
AI-Powered Pair Programming
Click the button to start recording..

Fig 6. Website running with the same IP address

Next, we will discuss about implementation of the project as hardware. In this the first step is to get the input (Voice) and convert it into text. We have used python language in the project so when converting from voice to text, the Speech recognition module uses Hidden Markov Model (HMM). When the user speaks into a microphone the speech is converted into stream of analog audio. Then it is converted into digital samples. After these Acoustic features, that are present in the audio is extracted. The Mel-Frequency Cepstral Coefficients (MCFFs) that represent the speech signal are also included in the Acoustic features. Each digit HMM represents a different model for the spoken digit. Viterbi Algorithm is used to find the most likely sequence of digits. After the most likely sequence of digits are determined. It gives us the corresponding text representation of the speech.

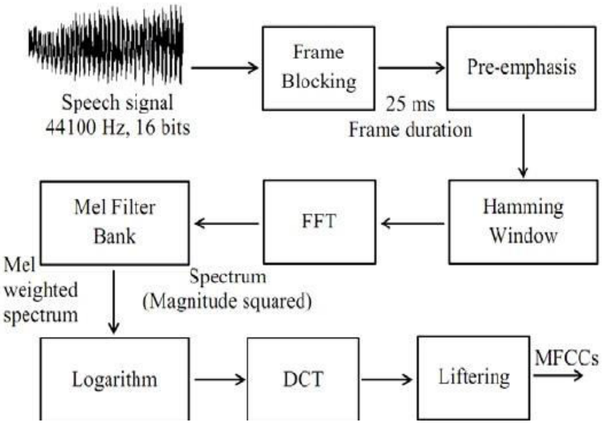


Fig 7. Voice to Text Process

In the next step, the output from the fist step i.e., the text transcribed from the speech is given as input to the Open AI Model. After getting the input the AI Model follows a series if steps to generate the answer to our query. The first step in that process is tokenization, it is simply defined as the splitting a sentence into small parts called tokens.

The second step is called Embedding. After the tokenization process takes place, each token is converted to a numerical form that the computer can understand and work with. These numerical values contain in them the semantic and syntactic information. Syntactic information relates to sentence structure, while semantic information concerns word meanings and relationships within sentences. The numerical representations are called as Embeddings.

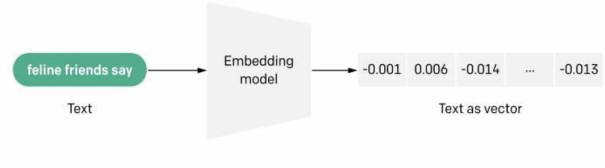


Fig 8. Embedding Process

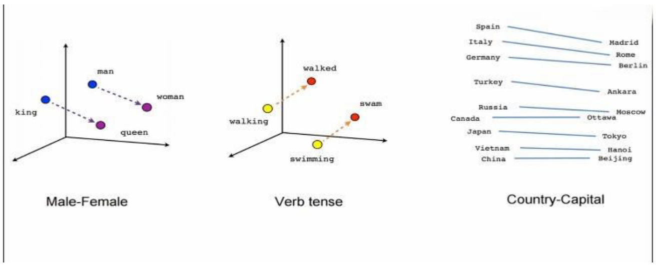


Fig 9. Embedding Process
(Syntactic Information in words)

The third step is Encoding, here These embeddings are then processed through multiple layers of the Open AI model. Within each layer, the embeddings are adjusted through various operations, such as self-attention mechanisms. These mechanisms allow each token to consider the context provided by the surrounding tokens, enabling the model to understand the relationships between different parts of the text.

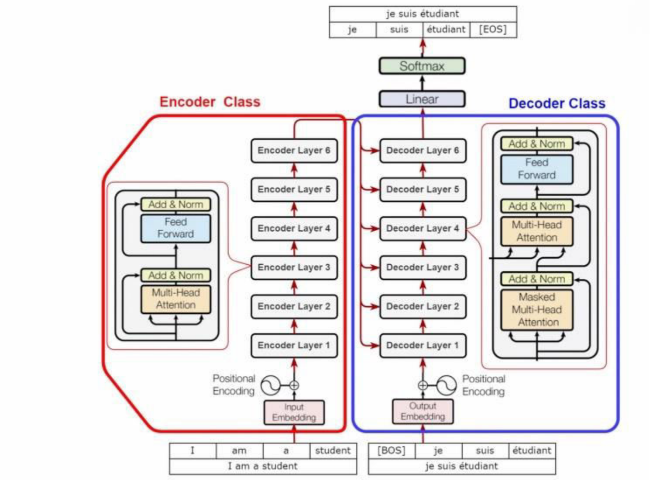


Fig 10. Encoding Process

As the embeddings pass through the layers, they gather more information from the surrounding tokens, building an understanding of the entire input text. Each layer contributes to this deepening understanding, refining the embeddings to capture every aspect of the text's meaning and structure. By the end of the encoding process, the embeddings carry a comprehensive representation of the input text, ready for further processing and analysis by the model.

The fourth step is Output Generation. In the AI Model, when predicting the next token, it analyzes previous tokens, including user input and generated tokens. This involves tokenization, converting each token into embeddings, and processing them through model layers. Through self-attention, the model understands token relationships. It then computes a probability distribution over the vocabulary for the next token based on context. This distribution represents the likelihood of various tokens occurring next. Thus, the probability distribution guides Open AI Model in choosing the next token. This process of probability computation ensures that Open AI Model generates appropriate responses, learned from training data to make informed predictions about the next word in a sequence. The model predicts each subsequent token based on the growing context. It generates tokens until reaching a stopping criterion, like punctuation or a maximum token limit, completing the phrase.

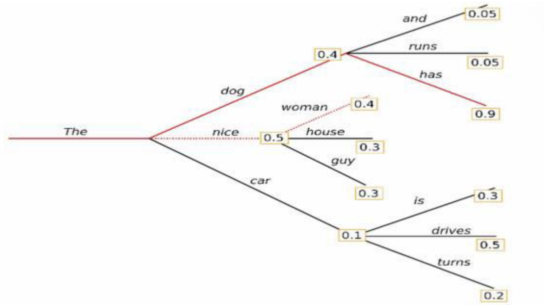
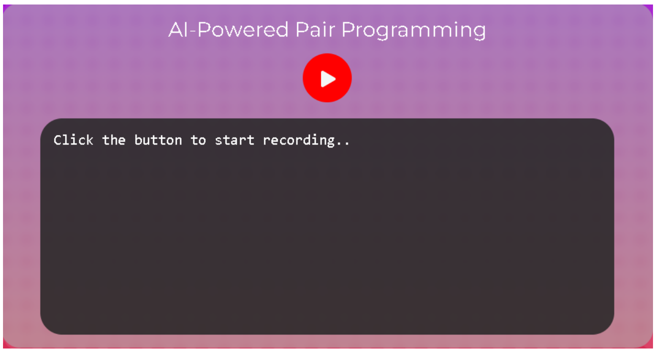


Fig 11. Output Generation Probability

The final step is Text-to-speech (TTS) conversion transforms generated text into spoken words through linguistic analysis, phonetic processing, and signal synthesis techniques. Initially, the input text undergoes preprocessing to standardize and normalize its format. Linguistic analysis identifies linguistic features, while phonetic processing converts words into their phonetic representations. Lexical lookup retrieves pronunciation information from a dictionary. Prosody generation determines intonation and rhythm for natural-sounding speech. Concatenative synthesis combines together recorded speech segments, while formant synthesis models speech sounds directly. Signal processing enhances speech quality through filtering and noise reduction. Post-processing refines the output, smoothing transitions and adjusting volume levels. The synthesized speech is outputted real-time playback.

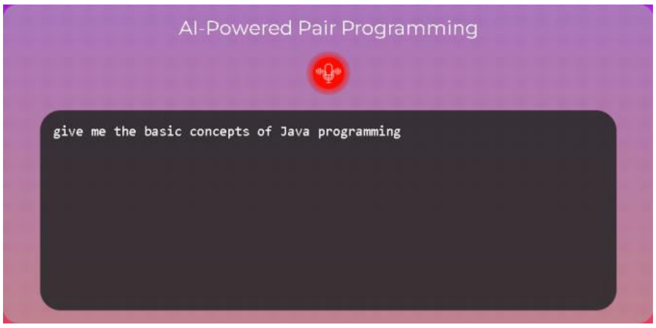
V. EXPERIMENTAL RESULTS

The results of the project as webpage and as hardware is given below. First, we are given the outputs of webpage.



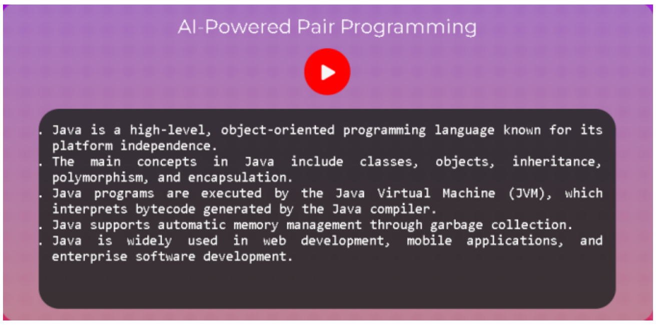
```
Click the button to start recording..
```

Fig 12. Webpage



```
give me the basic concepts of Java programming
```

Fig 13. Getting the question through the mic on the device



```
. Java is a high-level, object-oriented programming language known for its
  platform independence.
. The  main  concepts  in  Java  include  classes,  objects,  inheritance,
  polymorphism, and encapsulation.
. Java  programs  are  executed  by  the  Java  Virtual  Machine  (JVM),  which
  interprets bytecode generated by the Java compiler.
. Java supports automatic memory management through garbage collection.
. Java  is  widely  used  in  web  development,  mobile  applications,  and
  enterprise software development.
```

Fig 14. The answer displayed by the webpage both as text and as voice.

The first image shows the websites accessed using url. After that we need to click the play button in the webpage and start speaking the question. The webpage will display the question after that we need to press the mic button then the answer is generated. After the answer is generated its both viewable as a text and the webpage also uses the speaker on the device to play the answer as voice without any substantial delay or error.

Next we can see the results obtained by hardware implemntation and the result obtained by the software imlemntation are same except in the hardware the AI powered voice pair programmer is a separate device instead of the previous implementation which is a webpage. Here the process is exactly the same we give the input as voice through the mic it's converted into text and is fed in the Open AI Model to get the answer and the answer is brodcast as voice.

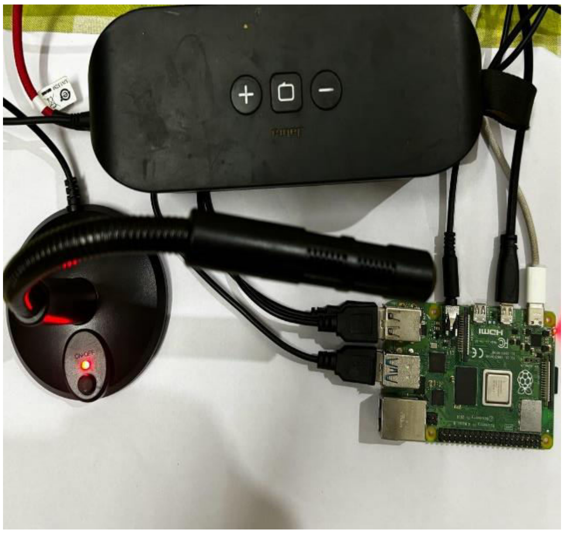
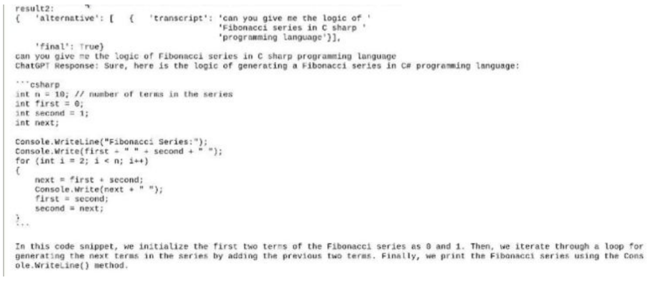


Fig 14. Hardware



~~~
result2:
{   'alternative': [   {   'transcript': 'can you give me the logic of '
                                         'Fibonacci series in C sharp '
                                         'programming language'}],
    'final': True}
can you give me the logic of Fibonacci series in C sharp programming language
ChatGPT Response: Sure, here is the logic of generating a Fibonacci series in C# programming language:

```csharp
int n = 10; // number of terms in the series
int first = 0;
int second = 1;
int next;

Console.WriteLine("Fibonacci Series:");
Console.Write(first + " " + second + " ");
for (int i = 2; i < n; i++)
{
    next = first + second;
    Console.Write(next + " ");
    first = second;
    second = next;
}
```

In this code snippet, we initialize the first two terms of the Fibonacci series as 0 and 1. Then, we iterate through a loop for
generating the next terms in the series by adding the previous two terms. Finally, we print the Fibonacci series using the Cons
ole.WriteLine() method.
~~~

Fig 14. Hardware Result in Raspberry PI

VI. CONCLUSION

In conclusion, the AI-powered Voice Pair Programmer project represents a significant advancement in collaborative coding. By leveraging natural language processing and speech synthesis technologies, this innovative system facilitates real-time collaboration between developers and an AI-powered partner. The developers can effortlessly communicate their coding intentions, receive immediate feedback, and jointly solve coding challenges with the assistance of an intelligent partner. The project not only enhances developer productivity and code quality but also fosters learning and knowledge sharing within software development teams. As AI continues to evolve, the Voice Pair Programmer project highlights the transformative potential of AI-driven tools in revolutionizing software development practices.

VII. REFERENCES

[1]. L. Athota, V. K. Shukla, N. Pandey and A. Rana, "Chatbot for Healthcare System Using Artificial Intelligence," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2020, pp. 619-622, doi: 10.1109/ICRITO48877.2020.9197833.

[2]. J. Skrebeca, P. Kalniete, J. Goldbergs, L. Pitkevica, D. Tihomirova and A. Romanovs, "Modern Development Trends of Chatbots Using Artificial Intelligence (AI)," 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical

University (ITMS), Riga, Latvia, 2021, pp. 1-6, doi: 10.1109/ITMS52826.2021.9615258.

[3]. L. Williams, R. R. Kessler, W. Cunningham and R. Jeffries, "Strengthening the case for pair programming," in IEEE Software, vol. 17, no. 4, pp. 19-25, July-Aug. 2000, doi: 10.1109/52.854064.

[4]. Qianou Ma, Tongshuang Wu, Kenneth Koedinger "Is AI the better programming partner? Human-Human Pair Programming vs. Human-AI Pair Programming" in arXiv preprint arXiv:2306.05153 (2023).

[5]. J. Bozic, O. A. Tazl and F. Wotawa, "Chatbot Testing Using AI Planning," 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 2019, pp. 37-44, doi: 10.1109/AITest.2019.00-10.