SMART LEARNING ASSISTANT

Thejaswin K[1], Gunasri B[2], Jatchuthan S S[3], Ramprabu J[4]

[1,2,3]Electrical and Electronics Engineering, Kumaraguru Collage of Technology, Coimbatore, Tamil Nadu
[4]Guide: Professor Department of Electrical and Electronics Engineering, Kumaraguru Collage, Coimbatore, Tamil Nadu, India

---***---

Abstract - *Presently the voice assistants are largely used and their function keeps enlarging. So this development of technology gives a discrete environment for educational purpose. Constructing a smart learning environment supported by a learning platform is the key area of experimentation in the educational fields nowadays. This development of smart learning environment has been advanced in the covid 19 pandemic. The idea mainly proposes a smart learning assistant with raspberry pi which is economically affordable and commercially available products.*

1. INTRODUCTION

According to the researchers the students needs one-on-one interaction with a personal teacher for the effective learning. However many schools, colleges and university finds it hard to give such individual tutor. The increasing number of schools, colleges and university make mass lectures with more than 100 students per lecturer and also the open online courses have more than 1000 participants make interaction with the lecturer even more difficult to learn. In last five to ten years it is shown that the computer based teaching system is more effective than human teaching system. Therefore this paper mainly deals with computer based easy tutoring system which is a smart learning assistant.

1.1 System Architecture

Raspberry Pi4: The Raspberry Pi 4 Model B is the latest addition to the popular Raspberry Pi series of computers. It dramatically improves processor speed, multimedia performance, storage, and connectivity compared to its predecessor, while maintaining power consumption similar to backward compatibility. For end users, the Raspberry Pi 4 Model B offers desktop performance comparable to entry-level x86 PC systems. Key features of this product include a powerful 64-bit quad-core processor and dual display support at resolutions up to 4K via a pair of micro-HDMI ports.

Voice Recognition module: Voice recognition module is a compact easy control process detection board. It is a speaker design module and supports up to 80 audio commands. All tones can be trained as commands. Users must first train the module before detecting any voice command. Voice commands are stored in large groups like libraries. You can import all seven voice commands in the library into the recognizer. This means that seven commands are valid at the same time. This board has two control options: Serial port (full function) and General input pin (part of function). Common output pins on the board can generate different types of waves while recognizing the corresponding voice commands.

Speaker: Adding a USB powered speaker to the Raspberry Pi is one of the easiest ways to play great audio from this project. It's also the ideal affordable speaker option for the Raspberry Pi 4 module. With just one USB connection, no additional cables, power adapters, or charging process. This is the simplest and most compact audio option on the Raspberry Pi. The device has two speakers and is powered by one Raspberry Pi connector.

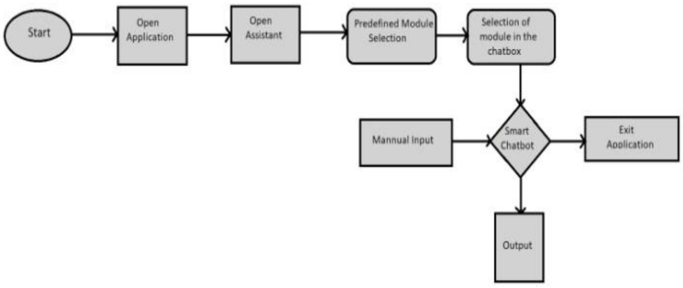
Display: The Raspberry Pi Touch Display is an LCD display that connects to the Raspberry pi via DSI. In some situations, it can be used with both HDMI and LCD displays simultaneously.

Input: A corrosion-free Type-C USB cable that makes it very easy to transfer data between the PC and host board, making it ideal for use with Raspberry Pi 4 and other USB Type-C devices. Not only does this cable use power lines, it also has high quality copper wire that is essential for long-term use. The high-quality tactile material is ideal for long-term use and protects against normal wear. With a length of 1 meter, it can be flexibly used in a variety of applications such as a Type-C cable for mobile phones and other Type-C devices, as well as the Raspberry Pi 4.

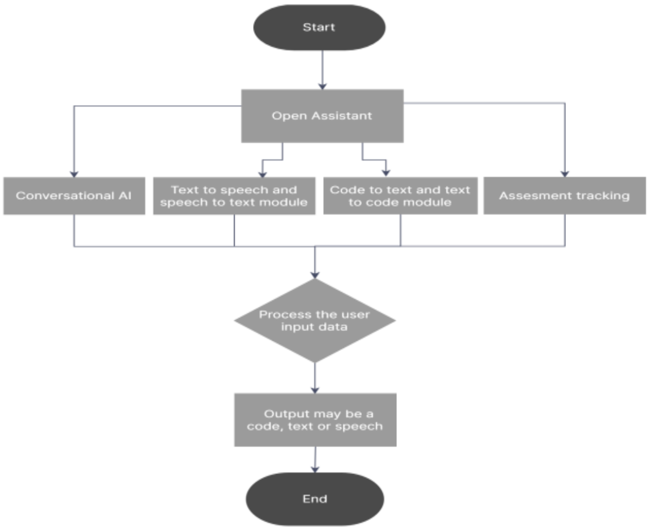
1.2 Implementation

In general, a SPA (Smart Personal Assistant) is a computer program that allows us to interact with users on a particular topic or domain in natural language. The SPA mainly contains all types of software that allow people to interact with your computer. The main task of the SPA is to improve the user's cognitive or behavioral abilities. The main advantages of using SPA are low response times, low service costs, and high user satisfaction. Compared to other e-learning technologies, SPA are characterized by their ability to interact synchronously with the learner and direct the learning process in response to individual needs.

So, this paper majorly deals with the development of a smart learning assistant for students to understand concepts in a step-by-step manner in an AI driven way i.e., Q/A generation, natural language to code, code to natural language, chat bot.



Once we start the smart learning assistant the main web page opens. It has predefined modules such as conversational AI which is a chatterbot for friendly interaction with students, progress tracking which is to track our grades helps us to do better. For example, in a chatbot if a user gives a manual input to the bot, it processes the given data and gives the specific defined output.


Start
Open Assistant
Conversational AI
Text to speech and speech to text module
Code to text and text to code module
Assesment tracking
Process the user input data
Output may be a code, text or speech
End

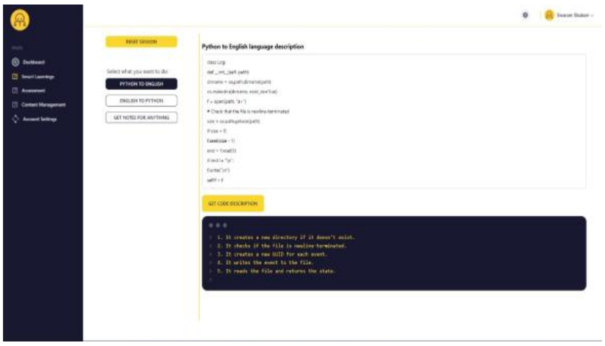
2.Code

```
import openai
from flask import *
app = Flask(_name_)
openai.api_key =
"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"
@app.route('/')
def index():
return "API is working"
@app.route('/nl2py',methods = ['POST'])
def api_1():
codePrompt = request.form['input']
response = openai.Completion.create(
engine="davinci-codex",
prompt=codePrompt,
temperature=0,
max_tokens=300,
top_p=1,
frequency_penalty=0,
presence_penalty=0)
return response.choices[0].text
@app.route('/py2nl',methods = ['POST'])
def api_2():
codePrompt = request.form['input']
response = openai.Completion.create(
engine="davinci-codex",
prompt=codePrompt,
temperature=0,
max_tokens=513,
top_p=1,
frequency_penalty=0,
presence_penalty=0,
stop=["\"\"\""]
)
return response.choices[0].text
if _name_ == '_main_':
app.run(debug = True, port=6000)
18
from flask import Flask
from flask import request, make_response
from flask_cors import CORS
import speech_recognition as sr
from OGGtoWAV import SpeechOggAudioFileToText
import helper_functions as hf
app = Flask(__name__)
r = sr.Recognizer()
```

```
CORS(app)
@app.route('/')
def index():
return 'Hello from LearnAssist text to speech API!'
@app.route('/audio', methods=['GET', 'POST'])
def welcome():
result =''
print('Request received')
try:
filename= 'audio/audio{}.ogg'.format(hf.makeId(12))
with open(filename, 'wb+') as destination:
destination.write(request.files['file'].stream.read())
SpeechOggAudioFileToText().ogg_to_wav(filename,filena
me.replace(
'.ogg','.wav'))
with sr.AudioFile(filename.replace('.ogg','.wav')) as
source:
audio_text = r.listen(source)
text = r.recognize_google(audio_text)
result = make_response(text,201)
print("\n\n\nResult: {}\n\n\n".format(text))
except Exception as e:
result = make_response(str(e),400)
print('Sorry.. try again...')
print(e)
return result
if __name__ == '__main__':
print('Server started @PORT 3005...')
app.run(port=3005)
```

3. RESULT AND CONCLUSION

PYTHON TO ENGLISH

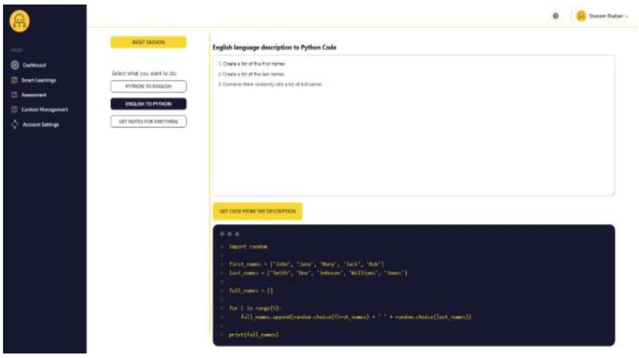


Python to English language description

```
class Log:
def __init__(self, path):
dirname = os.path.dirname(path)
os.makedirs(dirname, exist_ok=True)
f = open(path, "a+")
# Check that the file is newline-terminated
size = os.path.getsize(path)
if size > 0:
f.seek(size - 1)
end = f.read(1)
if end != "\n":
f.write("\n")
self.f = f
```

```
1. It creates a new directory if it doesn't exist.
2. It checks if the file is newline-terminated.
3. It creates a new UUID for each event.
4. It writes the event to the file.
5. It reads the file and returns the state.
```

ENGLISH TO PYTHON

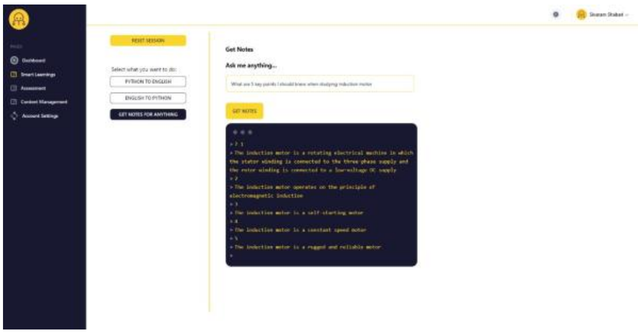


English language description to Python Code

1. Create a list of five first names
2. Create a list of five last names
3. Combine them randomly into a list of full names

```
import random

first_names = ['John', 'Jane', 'Mary', 'Jack', 'Bob']
last_names = ['Smith', 'Doe', 'Johnson', 'Williams', 'Jones']

full_names = []

for i in range(5):
    full_names.append(random.choice(first_names) + ' ' + random.choice(last_names))

print(full_names)
```

CONVERSATIONAL AI



Get Notes

Ask me anything...

What are 5 key points I should know when studying induction motor

```
> 1
> The induction motor is a rotating electrical machine in which the stator winding is connected to the three-phase supply and the rotor winding is connected to a low-voltage DC supply
> 2
> The induction motor operates on the principle of electromagnetic induction
> 3
> The induction motor is a self-starting motor
> 4
> The induction motor is a constant speed motor
> 5
> The induction motor is a rugged and reliable motor.
>
```

Conclusion:

Thus, a smart learning assistant has been made to assist the students learn through online and step by step manner which increases the learning outcomes of the students. These findings contribute to an intelligent tutoring system and smart assistant technology for the learners.

Smart learning assistant provides a convenient form of education through an application that supports natural conversational inputs and gives easily learnable outputs to the user where the student could easily learn.

REFERENCES

[1] Smart Personal Assistant using Machine Learning by Prajyot Mane, Shubham Sonone,Nachiket Gaikwad, Jyoti Ramteke from Computer Engineering Department, Sardar Patel Institute of Technology, Mumbai (2017).

[2] A vision and speech enabled, customizable, virtual assistant for smart environments by Giancarlo Iannizzotto, Lucia Lo Bello, Andrea Nucita, Giorgio Mario Grasso from Dept. for Cognitive Sciences, Education and Cultural Studies (COSPECS) University of Messina, Messina, Italy(2018).

[3] EduEasy - Smart Learning Assistant System by A. Karunasena, P.Bandara , J.A.T. P .Jayasuriya, P.D. Gallage, J.M.S.D. Jayasundara, L.A.P.Y.P. Nuwanjaya from Sri Lanka Institue of Information Technology, Malabe, Sri Lanka(2021).

[4] NLAST: A natural language assistant for students by Fernando A. Mikic Fonte, Martin Llamas Nistal, Martin Llamas Nistal, Manuel CaeiroRodriguez from Universidade de Vigo, Vigo, Galicia, ES(2016).