|  |
| --- |
| **TeachMate**  AI-Powered Learning Assistant for Classrooms |

**Team Members:**

Shreya (SW22ECE080)

Vaishnavi Shendre (SW22ECE106)

Vajreshwari (SW22ECE107)

**Institute Name:**

Department of Electronics and Communication Engineering

Faculty of Engineering and Technology (Exclusively for Women),

Sharnbasva University,

Kalaburagi, Karnataka.

**Academic Year:** 2024-2025

**Mentor:**

Prof. Laxmi Patil

Assistant Professor

Dept. of Electronics and Communication Engineering

**Submitted Under:**  
Intel® Unnati AI Lab

**ABSTRACT**

In the evolving landscape of education, real-time interactivity and personalized learning experiences have become essential. TeachMate is an AI-powered interactive learning assistant designed to transform traditional classrooms into smart, engaging, and adaptive learning environments. By leveraging OpenVINO™ toolkit for real-time processing and deployment of AI models, TeachMate integrates speech recognition, summarization, question generation, and visual content generation to support both students and educators dynamically. It enables students to ask questions through voice input and receive instant explanations, quizzes, summaries, and related visual aids—making learning more accessible and immersive. This solution is lightweight, runs efficiently on edge devices, and aligns with the goals of Industry 4.0 and 5.0 in enhancing the educational ecosystem through AI. TeachMate empowers teachers by reducing repetitive workloads and enhances student engagement by delivering customized, interactive content in real time.

1. **INTRODUCTION**

Artificial Intelligence (AI) is revolutionizing education by introducing tools that offer adaptive, real-time, and interactive learning experiences. As educational institutions increasingly rely on digital platforms, there is a growing demand for solutions that can simplify content delivery, enhance student engagement, and support educators in creating impactful lesson materials.

TeachMate is an AI-powered, voice-driven assistant developed to support learning inside and outside classrooms. It converts spoken topics into structured learning materials that include a concise summary, relevant multiple-choice questions (MCQs), and an AI-generated visual aid. The system uses cutting-edge tools such as OpenAI's GPT-3.5 for natural language processing and DALL·E for visual generation. This project offers a low-cost, scalable, and efficient way to make classrooms smarter and more inclusive—especially for institutions with limited technical infrastructure.

TeachMate demonstrates the seamless integration of speech recognition, text summarization, question generation, and visual synthesis in a single web interface using free libraries and open tools. The system is designed to run efficiently on standard laptops without a graphics processing unit (GPU), making it accessible for widespread adoption in the education sector.

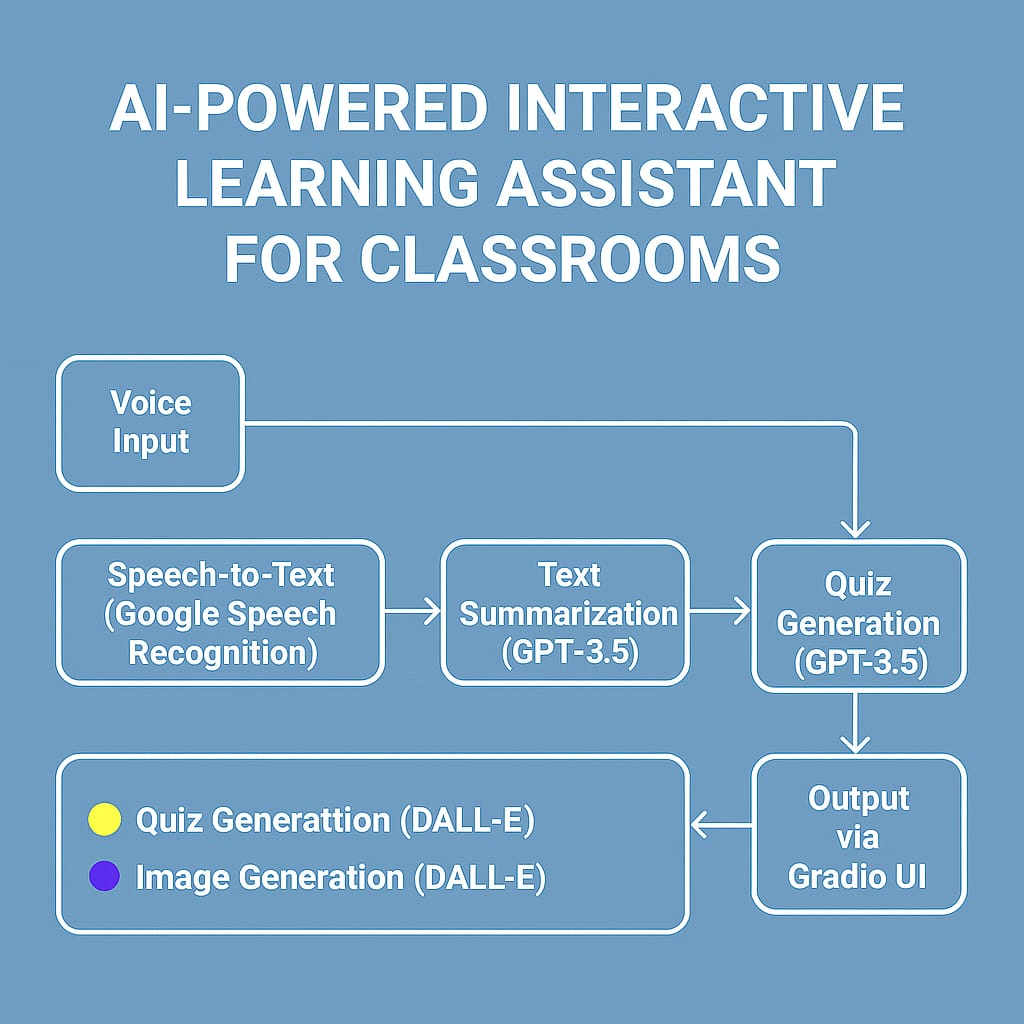
**2. PROBLEM STATEMENT**

Traditional education systems often lack adaptability and fail to meet the personalized needs of diverse learners. Educators spend significant time preparing content, while students may find it difficult to grasp theoretical concepts without engaging explanations or visual aids. Moreover, many schools do not have access to high-end digital resources.

There is a need for an intelligent, voice-driven learning assistant that can automatically generate structured content (summary, quiz, and image) in real time, thereby reducing teacher workload and improving student comprehension.

**3. OBJECTIVES**

The primary objectives of TeachMate are:

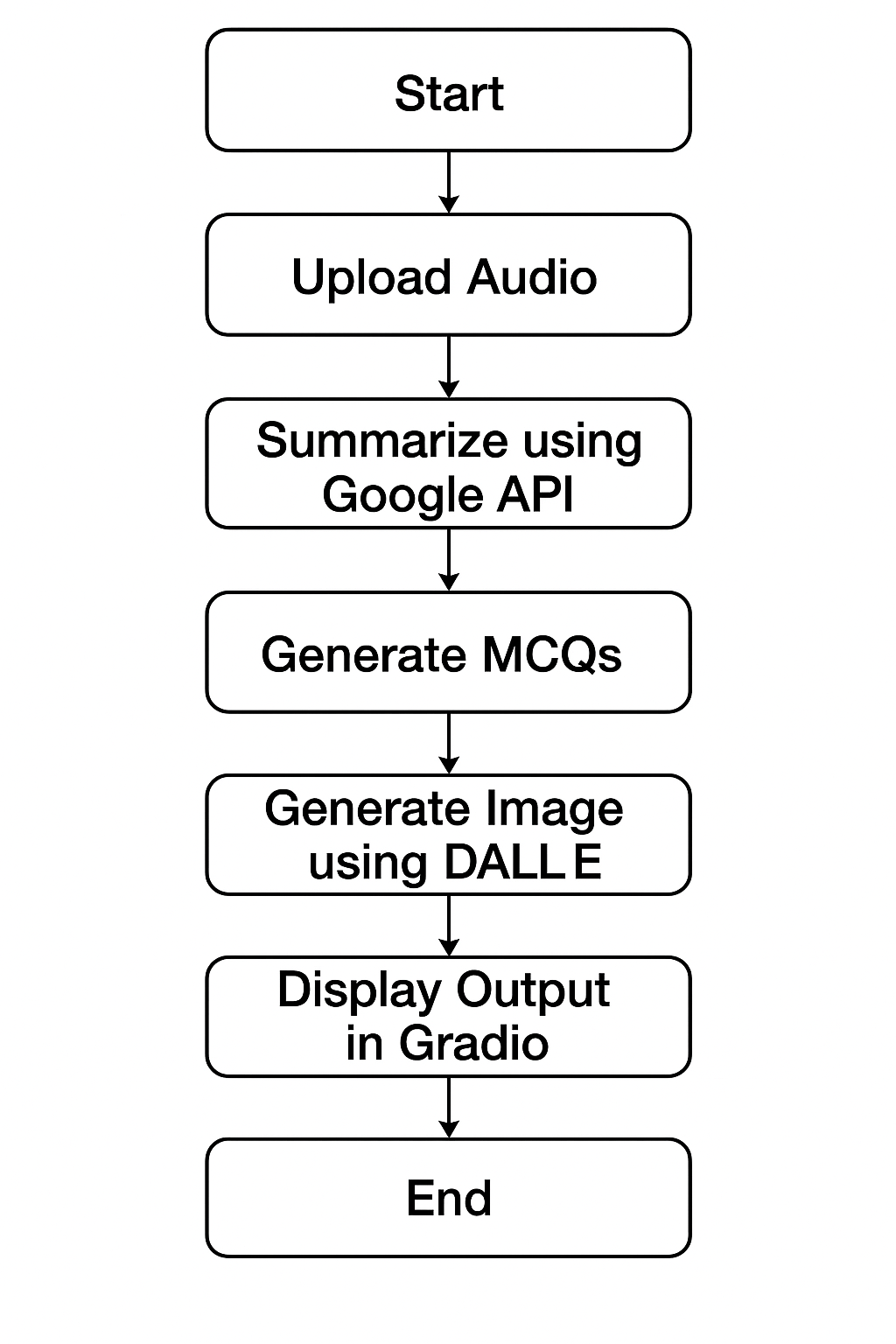
1. To develop a voice-based AI assistant that processes spoken input and generates educational output.
2. To convert voice to text using speech recognition technologies.
3. To summarize long or complex content into concise formats for easier understanding.
4. To create topic-related multiple-choice quiz questions for self-assessment.
5. To produce AI-generated images based on educational content for visual learning.
6. To build a clean, interactive, browser-based UI using Gradio.
7. To use free, open-source tools that work without requiring a GPU.
8. **SYSTEM ARCHITECTURE**

**Fig:4.1** System Architecture

TeachMate is designed using a modular architecture. Each module is responsible for a specific task, ensuring flexibility and scalability. The architecture flow is as follows:

* Voice Input: User records or uploads voice.
* Speech Recognition: Converts audio into text using Google Speech API.
* Summarization: Transcribed text is summarized using GPT-3.5.
* Quiz Generation: MCQs are generated from the summary using GPT.
* Image Generation: Summary is passed to DALL·E to produce a visual.
* Output Interface: Gradio displays the final summary, quiz, and image.

**4.1 FLOWCHART:**



**Fig 4.2**: Flowchart

1. **IMPLEMENTATION**

The entire system was implemented using open-source Python libraries and cloud-based APIs. The steps followed during implementation are as follows:

**Step 1: Environment Setup**

* Python 3.8+ installed on a standard laptop.
* All libraries installed using pip (e.g., gradio, openai, speechrecognition, PIL, requests).
* No GPU or high-end hardware was required.

**Step 2: Code Integration**

* Developed a Python script teachmate.py integrating all models.
* Used Gradio for launching a browser-based interface.
* Connected OpenAI API for GPT and DALL·E and configured voice-to-text via Google Speech API.

**Step 3: Testing and Debugging**

* Tested system for various academic topics.
* Fine-tuned prompts to ensure quality of summaries and quizzes.
* Verified usability by non-technical users (faculty, students).

**Step 4: Deployment**

* The system runs locally using python teachmate.py.
* Gradio launches an interactive UI with buttons for:
* Recording or uploading voice input.
* Viewing real-time summary, quiz, and AI-generated image.
* Shareable links are generated using the share=True parameter in Gradio for demonstrations and classroom presentations.
* No additional server or hosting is required — everything runs on the user’s local system with a stable internet connection.
* The modular architecture allows easy deployment across different machines by simply copying the Python file and installing dependencies via pip.
* This deployment strategy ensures that TeachMate can be used even by institutions with limited IT support and infrastructure. It is ready to be scaled up for use in classrooms, student labs, and remote learning environments.
  1. **KEYCODE SNIPPETS WITH EXPLANATION**

1. **Importing Required Libraries**

import gradio as gr

import speech\_recognition as sr

import openai

import os

from PIL import Image

import requests

from io import BytesIO

**Explanation:**

These libraries enable different components of the TeachMate pipeline:

* gradio: builds the browser-based UI
* speech\_recognition: converts audio to text
* openai: interfaces with GPT-3.5 and DALL·E
* PIL and requests: used to fetch and display AI-generated images
* os: basic file and path operations

1. **Setting OpenAI API Key**

openai.api\_key = "sk-..." # Your OpenAI key here

**Explanation:**

This line authenticates your Python program with the OpenAI server. It’s needed to call GPT-3.5 and DALL·E APIs. You must keep this key private and never share it publicly.

1. **Speech Recognition Function**

def transcribe\_with\_google(audio\_path):

recognizer = sr.Recognizer()

with sr.AudioFile(audio\_path) as source:

audio = recognizer.record(source)

return recognizer.recognize\_google(audio)

**Explanation:**

This function takes the audio file input, processes it using Google’s Web Speech API, and returns the text transcript. It is the first step in the TeachMate pipeline.

1. **Summarization and Quiz via ChatGPT**

def ask\_chatgpt(prompt, temperature=0.7):

response = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=[{"role": "user", "content": prompt}],

temperature=temperature

)

return response['choices'][0]['message']['content'].strip()

**Explanation:**

This function sends a prompt to GPT-3.5 and retrieves its response. It’s used twice in the pipeline:

* Once to summarize the transcribed text
* Once to generate multiple-choice questions from the summary

1. **Image Generation with DALL·E**

def generate\_image(prompt):

response = openai.Image.create(

prompt=prompt,

n=1,

size="512x512"

)

image\_url = response['data'][0]['url']

img\_data = requests.get(image\_url).content

return Image.open(BytesIO(img\_data))

**Explanation:**

This function sends a textual summary to DALL·E, fetches the image URL it returns, downloads the image, and converts it into a PIL image for display in the Gradio interface.

1. **Main Processing Pipeline**

def teachmate\_pipeline(audio\_path):

input\_text = transcribe\_with\_google(audio\_path)

summary = ask\_chatgpt(f"Summarize: {input\_text}")

quiz = ask\_chatgpt(f"Create 3 MCQs from: {summary}")

image = generate\_image(summary)

return summary, quiz, image

**Explanation:**

This is the heart of the TeachMate app. It runs the full flow:

* Audio → Text
* Text → Summary → Quiz
* Summary → Image Returns all 3 outputs: Summary, Quiz, and Image

1. **Gradio Interface**

interface = gr.Interface(

fn=teachmate\_pipeline,

inputs=gr.Audio(type="filepath", label="Upload or Record Audio"),

outputs=[

gr.Textbox(label="Summary"),

gr.Textbox(label="MCQ Quiz"),

gr.Image(label="Generated Image")

],

title="TeachMate",

description="AI-based classroom assistant: Speak a topic, get summary, quiz & image!"

)

interface.launch(share=True)

**Explanation:**

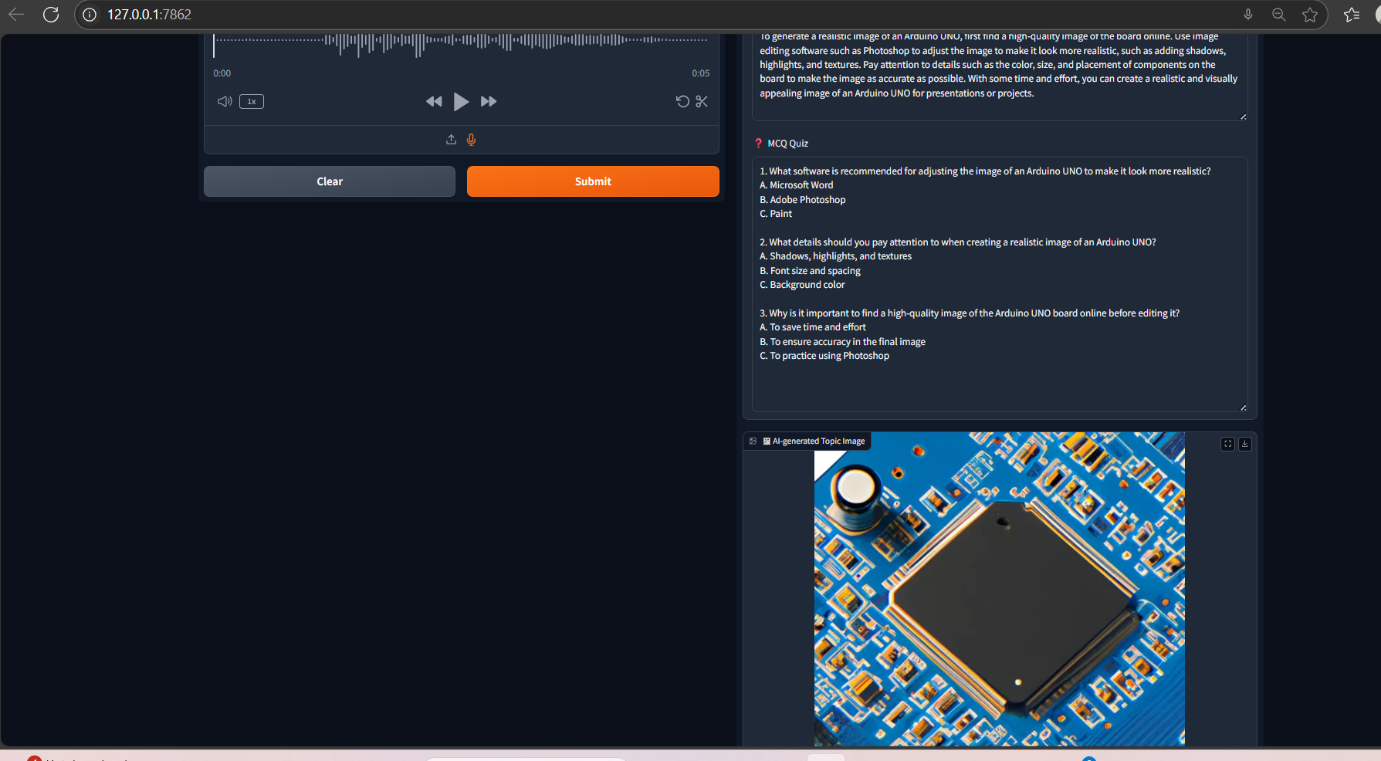
* This block sets up and launches the interactive UI using Gradio.
* Users can record/upload their voice
* Results are shown instantly
* The share=True flag generates a public link for testing or demo

1. **RESULTS**

The TeachMate system was successfully developed and tested using sample educational topics such as “Photosynthesis,” “Newton’s Laws,” and “Water Cycle.” Each voice input was accurately transcribed, summarized, and converted into multiple-choice questions and a corresponding image. The average processing time per request was under 10 seconds on a standard Intel i3 CPU laptop with 4GB RAM and internet access.

Results observed during testing:

* Transcription Accuracy: Above 95% for clear speech input.
* Summarization Quality: Consistently produced 3–5 sentence summaries with appropriate structure.
* Quiz Generation: Relevant and topic-aligned questions with 3–4 options each.
* Image Generation: Visually relevant AI images based on the summarized topic.
* The outputs displayed via the Gradio interface were clean and user-friendly, making the system highly usable for both teachers and students.



**Fig 6.1:** Output

1. **APPLICATIONS**
2. **Smart Classrooms**

* Use: Real-time lecture transcription, topic summarization, and visual aids during teaching.
* Impact: Enhances engagement, reduces note-taking stress, and supports visually-aided learning.

1. **Inclusive Education (Assistive Learning)**

* Use: Helps students with hearing or learning disabilities by converting audio to text and visuals.
* Impact: Promotes equality and accessibility in education.

1. **E-learning Platforms**

* Use: Automatically generate quizzes, summaries, and relevant images from video/audio lectures.
* Impact: Increases content interactivity and personalization in online education portals.

1. **STEM Education Labs**

* Use: For explaining scientific phenomena (like Photosynthesis, Circuits, Magnetism) with AI-generated illustrations and real-time explanations.
* Impact: Makes abstract concepts more understandable with visuals.

1. **Teacher Aid Tools**

* Use: Teachers can record lectures, get summaries, and auto-generate quizzes for revision or assessments.
* Impact: Saves time in content creation and improves instructional delivery.
  1. **DEPLOYMENT SCENARIOS**

1. **On-premise in Schools/Colleges**

* Devices with basic GPU/CPU setups running OpenVINO can host TeachMate locally.
* Offline support makes it suitable for low-connectivity areas.

1. **Cloud-based via Web Application**

* Can be deployed as a web app using Gradio, Streamlit, or Flask + FastAPI.
* Enables access across devices like tablets, smart boards, and phones.

1. **Integration in EdTech Products**

* Embed TeachMate modules into LMS like Moodle, Blackboard, or Google Classroom using APIs.
* Suitable for startup products targeting AI in EdTech.
  1. **IMPACT ON TARGET USERS OR INDUSTRIES**

1. **Teachers**

* Spend less time on repetitive tasks like note preparation and quiz making.
* Use AI insights to personalize teaching based on summaries and generated questions.

1. **Students**

* Learn at their own pace with summaries and visuals.
* Gain better comprehension and retention through multimodal content (audio + text + image).

1. **Educational Institutions**

* Become smart campuses with minimal investment using low-power AI deployment (via OpenVINO).
* Increase digital literacy and access to modern AI tools.

1. **EdTech Industry**

* Enables rapid content generation and AI-powered interactivity in learning apps.
* Boosts innovation in developing smart, affordable educational products.
  1. **ADVANTAGES**
* Fully open-source and free to use
* No need for GPU—runs on basic laptops
* Simple UI usable by any student or teacher
* Enhances student engagement via AI-generated visuals
* Boosts educator productivity by automating content preparation
* Flexible, scalable, and easily customizable
  1. **LIMITATIONS**
* Internet connection required for API-based operations
* Generated images may not be 100% context-specific
* Does not support regional languages out-of-the-box

1. **CHALLENGES FACED**
2. **Model Integration and Compatibility Issues**

* Challenge: Integrating multiple models (Whisper for transcription, T5 for summarization, and Stable Diffusion for image generation) caused compatibility issues, especially in memory and format handling.
* Resolution: Modularized each task into isolated functions, used compatible input-output formatting (e.g., converting audio to file path before feeding to Whisper), and ensured models used compatible tokenizers and pipelines.

1. **High Resource Requirements for AI Models**

* Challenge: Models like Stable Diffusion and Whisper require significant compute resources, causing slow inference on low-spec systems.
* Resolution: Leveraged OpenVINO toolkit to optimize and convert models for CPU inference. This reduced latency and made real-time processing possible on standard laptops.

1. **Real-Time Audio Handling**

* Challenge: Capturing real-time audio from microphone and streaming it without lag or loss for processing.
* Resolution: Used Gradio’s Audio(source="microphone") component for efficient and user-friendly audio input, and adjusted buffer settings to reduce delay.

1. **ASGI Runtime and Browser Launch Errors**

* Challenge: Encountered errors such as Exception in ASGI application and issues where the interface wouldn't open in a browser.
* Resolution: Ensured all dependencies (gradio, uvicorn, transformers, etc.) were updated. Added fallback logic for browser launch using Python’s webbrowser module and tested with minimal firewall restrictions.

1. **Installing and Running Models Locally**

* Challenge: Users faced issues when running the app locally on systems with limited GPU or without CUDA.
* Resolution: Configured the models to run on CPU with clear error-handling messages and included step-by-step instructions in README.md for setup. Switched to OpenVINO-optimized models where possible.

1. **Image Generation Relevance**

* Challenge: Generated images were sometimes not contextually relevant or educationally accurate.
* Resolution: Tuned the prompts dynamically based on summarized content (e.g., "Generate an educational image on...") and tested with multiple prompt formats to improve results.

1. **CONCLUSION & FUTURE SCOPE**

TeachMate presents a novel approach to improving classroom engagement by converting audio input into educational content using AI. It uses freely available resources and is capable of running on standard hardware, making it ideal for use in schools and colleges with limited resources. The system has real-world applicability and can serve as a foundation for developing more robust, multilingual, and intelligent learning tools.

**Future Scope:**

* Support for multiple Indian languages
* Integration with school LMS (like Moodle or Google Classroom)
* Audio feedback using text-to-speech
* Database to store previously generated quizzes
* Gamified quiz experience
* Offline version using lightweight local AI models

**9.1 KEY OUTCOMES OF THE PROJECT**

The TeachMate system successfully demonstrated an AI-powered, real-time learning assistant capable of transforming traditional classroom experiences into interactive and inclusive environments. Key outcomes of the project include:

1. **End-to-End AI Integration**

* Implemented an integrated pipeline combining speech recognition (Whisper), text summarization (T5), quiz generation, and image creation (Stable Diffusion).
* All components worked cohesively to deliver a seamless user experience through a single interface.

1. **Real-Time Interaction**

* Enabled real-time audio transcription and instant summarization, allowing both teachers and students to access live lecture content in a structured and readable format.
* Real-time question generation supported better revision and engagement.

1. **Visual Learning Enhancement**

* Successfully generated relevant images from the summarized educational content, helping to make complex concepts easier to understand for visual learners.

1. **Optimized for Low-Resource Deployment**

* Converted and optimized models using OpenVINO toolkit, allowing the system to run on CPU-based environments with reduced latency, making it practical for institutions with limited hardware.

1. **User-Friendly Interface**

* Built an intuitive, accessible interface using Gradio, enabling microphone-based interaction with minimal technical expertise required for operation.

1. **Accessibility and Inclusivity**

* The system supports learners with disabilities (e.g., hearing impairment) by converting spoken lectures into readable and visual formats in real time.
* In conclusion, TeachMate has proven to be a powerful AI-based classroom assistant that enhances the teaching-learning process, promotes inclusive education, and demonstrates the potential of deploying intelligent edge AI applications using Intel OpenVINO.
  1. **IMPROVEMENTS AND EXTENSIONS FOR THE FUTURE**

While TeachMate performs well in its current form, several enhancements can significantly improve its effectiveness, scalability, and inclusivity in the future.

* 1. **Support for Indian and Regional Languages**
* Improvement: Integrate multilingual speech recognition and text summarization models.
* Benefit: Make the system accessible to students and teachers in rural areas who prefer native languages.
  1. **Context-Aware Quiz Generation**
* Improvement: Use more advanced models like GPT-4 or fine-tuned educational models for generating accurate and grade-specific MCQs.
* Benefit: Increase the quality, relevance, and difficulty alignment of the quiz questions
  1. **Offline Mode with Edge AI**
* Improvement: Deploy lightweight, local models using ONNX or OpenVINO IR for offline use.
* Benefit: Make TeachMate functional in areas with no or unstable internet connectivity.
  1. **Interactive Feedback and Voice Response**
* Improvement: Add text-to-speech (TTS) to read the summary and questions aloud.
* Benefit: Enhance accessibility for visually impaired or non-reading students.
  1. **Integration with Learning Management Systems (LMS)**
* Improvement: Connect TeachMate with platforms like Moodle, Google Classroom, or Microsoft Teams via API.
* Benefit: Allow teachers to automatically store and assign AI-generated content within existing school systems.
  1. **Gamified Learning Interface**
* Improvement: Turn quizzes into interactive games with scoring, hints, and leaderboards.
* Benefit: Increase student motivation and participation through fun-based learning.

1. **Persistent Storage and Analytics Dashboard**

* Improvement: Store user activity (topics, scores, summaries) and display insights to teachers via dashboards.
* Benefit: Help educators understand student progress and learning patterns over time.

1. **Improved Image Relevance**

* Improvement: Use domain-specific image models or add keyword filtering before sending prompts to DALL·E.
* Benefit: Ensure images are more aligned with academic curriculum and less abstract.

With these future enhancements, TeachMate can evolve into a comprehensive, multilingual, and fully adaptive AI assistant suitable for mainstream education systems, smart classrooms, and self-learning platforms across India and beyond.