# Edu Tutor AI – Full Source Code

This document includes the complete code used to implement all functionalities in Edu Tutor AI.

## 📌 Step 1: Install Required Libraries

!pip install -q transformers gradio accelerate

## 📌 Step 2: Load IBM Granite Model

from transformers import AutoModelForCausalLM, AutoTokenizer  
import torch  
  
model\_name = "ibm-granite/granite-3.3-2b-instruct"  
tokenizer = AutoTokenizer.from\_pretrained(model\_name)  
model = AutoModelForCausalLM.from\_pretrained(model\_name, torch\_dtype=torch.float32)  
  
def generate\_response(prompt):  
 inputs = tokenizer(prompt, return\_tensors="pt")  
 with torch.no\_grad():  
 outputs = model.generate(\*\*inputs, max\_new\_tokens=300, temperature=0.7)  
 return tokenizer.decode(outputs[0], skip\_special\_tokens=True)

## 📌 Step 3: Define Functionalities

def subject\_qna(subject, question):  
 prompt = f"As a {subject} tutor, answer this question clearly: {question}"  
 return generate\_response(prompt)  
  
def explain\_topic(topic):  
 prompt = f"Explain the topic in simple terms: {topic}"  
 return generate\_response(prompt)  
  
def generate\_essay(topic, word\_limit):  
 prompt = f"Write a detailed essay of about {word\_limit} words on: {topic}"  
 return generate\_response(prompt)  
  
def chat\_with\_tutor(user\_input):  
 prompt = f"Student: {user\_input}\nTutor:"  
 return generate\_response(prompt)  
  
def quiz\_creator(topic):  
 prompt = f"Create a 5-question quiz with answers on the topic: {topic}"  
 return generate\_response(prompt)  
  
def summarize\_text(text):  
 prompt = f"Summarize the following text: {text}"  
 return generate\_response(prompt)  
  
def translate\_text(text, language):  
 prompt = f"Translate the following into {language}: {text}"  
 return generate\_response(prompt)  
  
def flashcards(topic):  
 prompt = f"Generate 5 flashcards (Q&A style) for the topic: {topic}"  
 return generate\_response(prompt)

## 📌 Step 4: Gradio Interface

import gradio as gr  
  
with gr.Blocks() as demo:  
 gr.Markdown("# 🎓 Edu Tutor AI – Smart Personalized Learning Assistant")  
  
 with gr.Tab("📚 Subject Q&A"):  
 subject = gr.Dropdown(["Math", "Science", "English", "History", "Geography"], label="Select Subject")  
 question = gr.Textbox(lines=3, label="Ask your question")  
 output1 = gr.Textbox(label="Answer")  
 btn1 = gr.Button("Get Answer")  
 btn1.click(subject\_qna, [subject, question], output1)  
  
 with gr.Tab("🧠 Explain Topic"):  
 topic = gr.Textbox(label="Enter a topic to explain")  
 output2 = gr.Textbox(label="Explanation")  
 btn2 = gr.Button("Explain")  
 btn2.click(explain\_topic, topic, output2)  
  
 with gr.Tab("✍️ Essay Generator"):  
 essay\_topic = gr.Textbox(label="Essay Topic")  
 word\_limit = gr.Number(label="Word Limit", value=150)  
 output3 = gr.Textbox(label="Generated Essay")  
 btn3 = gr.Button("Generate Essay")  
 btn3.click(generate\_essay, [essay\_topic, word\_limit], output3)  
  
 with gr.Tab("💬 Chat with Tutor"):  
 chat\_input = gr.Textbox(label="Type your message")  
 output4 = gr.Textbox(label="AI Tutor Response")  
 btn4 = gr.Button("Send")  
 btn4.click(chat\_with\_tutor, chat\_input, output4)  
  
 with gr.Tab("📝 Quiz Creator"):  
 quiz\_topic = gr.Textbox(label="Enter quiz topic")  
 output5 = gr.Textbox(label="Quiz with Answers")  
 btn5 = gr.Button("Create Quiz")  
 btn5.click(quiz\_creator, quiz\_topic, output5)  
  
 with gr.Tab("🔍 Summary Generator"):  
 long\_text = gr.Textbox(lines=5, label="Paste content to summarize")  
 output6 = gr.Textbox(label="Summary")  
 btn6 = gr.Button("Summarize")  
 btn6.click(summarize\_text, long\_text, output6)  
  
 with gr.Tab("🌍 Translator"):  
 translate\_input = gr.Textbox(label="Text to Translate")  
 lang = gr.Dropdown(["Hindi", "French", "Spanish", "German", "Tamil"], label="Target Language")  
 output7 = gr.Textbox(label="Translated Text")  
 btn7 = gr.Button("Translate")  
 btn7.click(translate\_text, [translate\_input, lang], output7)  
  
 with gr.Tab("🧾 Flashcard Generator"):  
 flash\_topic = gr.Textbox(label="Enter topic for flashcards")  
 output8 = gr.Textbox(label="Generated Flashcards")  
 btn8 = gr.Button("Generate Flashcards")  
 btn8.click(flashcards, flash\_topic, output8)  
  
demo.launch(share=True)