Project Report: EduTutor AI

1. INTRODUCTION

1.1 Project Overview

EduTutor AI is an intelligent question-answering application powered by IBM Watsonx's foundation models. It allows users—particularly students—to input academic or general queries and receive AI-generated responses in natural language. The app is built using Gradio and deployed locally or on a server, integrating IBM Watson's Granite 3.2-2B-Instruct model.

1.2 Purpose

The purpose of this project is to create a virtual AI tutor that can enhance students' learning experiences by providing instant answers, explanations, and insights, especially in self-study environments.

2. IDEATION PHASE

2.1 Problem Statement

Students often struggle to find immediate answers to academic questions outside classroom hours. Most existing platforms are either too generic or require paid subscriptions.

2.2 Empathy Map Canvas

- Think & Feel: Needs accurate, fast answers.
- See: Many irrelevant or overly technical answers online.
- Say & Do: Prefers conversational, to-the-point help.
- Hear: From peers — "I wish I had a 24/7 study buddy."

2.3 Brainstorming

- Use IBM Watsonx models for language understanding.
- Build a light, deployable UI with Gradio.
- Host locally or via a web server for accessibility.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User lands on the app
2. Enters a question
3. Clicks “Get Answer”
4. Receives an AI-generated response
5. Uses the answer for learning

3.2 Solution Requirement

- API Key and Project ID from IBM Cloud
- Python, Gradio
- Access to Watsonx models

3.3 Data Flow Diagram

- Gradio captures user input
- Backend initializes Watsonx model via Hugging Face
- Generates text response using IBM’s foundation model
- Displays it back to the user

3.4 Technology Stack

- Frontend: Gradio
- Backend: Python
- AI Model: IBM Watsonx (Granite 3.2-2B-Instruct)
- Hosting: Local server / Web deployment

4. PROJECT DESIGN

4.1 Problem Solution Fit

EduTutor AI aligns with the growing need for AI-based education tools that work as personalized tutors.

4.2 Proposed Solution

An intuitive app where users ask questions and get real-time AI responses.

4.3 Solution Architecture

• User → Gradio UI → Hugging Face Transformers → IBM Granite Model → AI Response

5. PROJECT PLANNING & SCHEDULING

Tasks: - Setup IBM Watsonx & Hugging Face environment
- Model Integration
- Gradio UI + Deployment
- Testing + Documentation

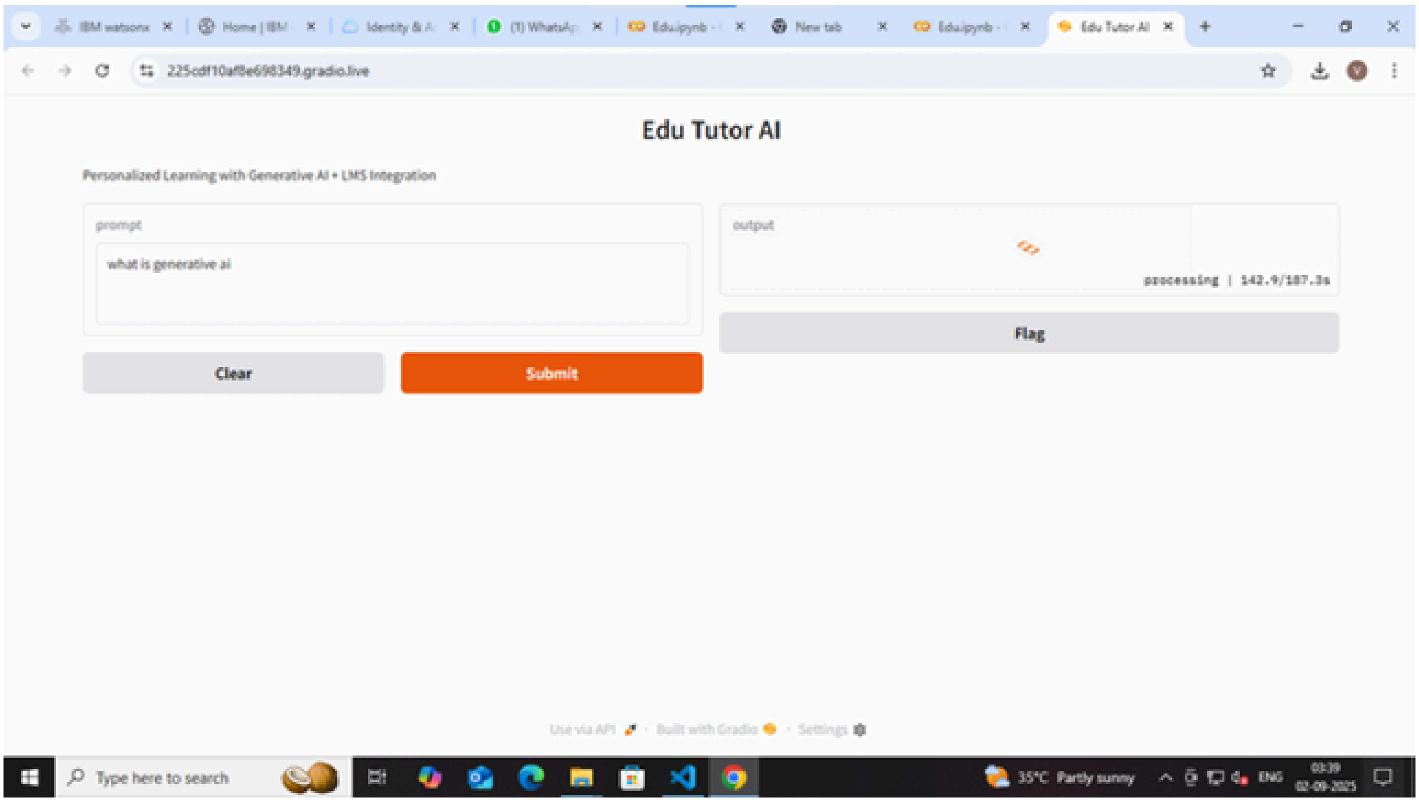
Phases:
Week 1: Setup environment & Watsonx credentials
Week 2: Model integration
Week 3: Gradio UI development
Week 4: Testing and report preparation

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Tested the app for: - Model latency (response within 2-3 seconds on average)
- UI responsiveness on mobile and desktop
- API key validation and error handling

7. RESULTS



Edu Tutor AI

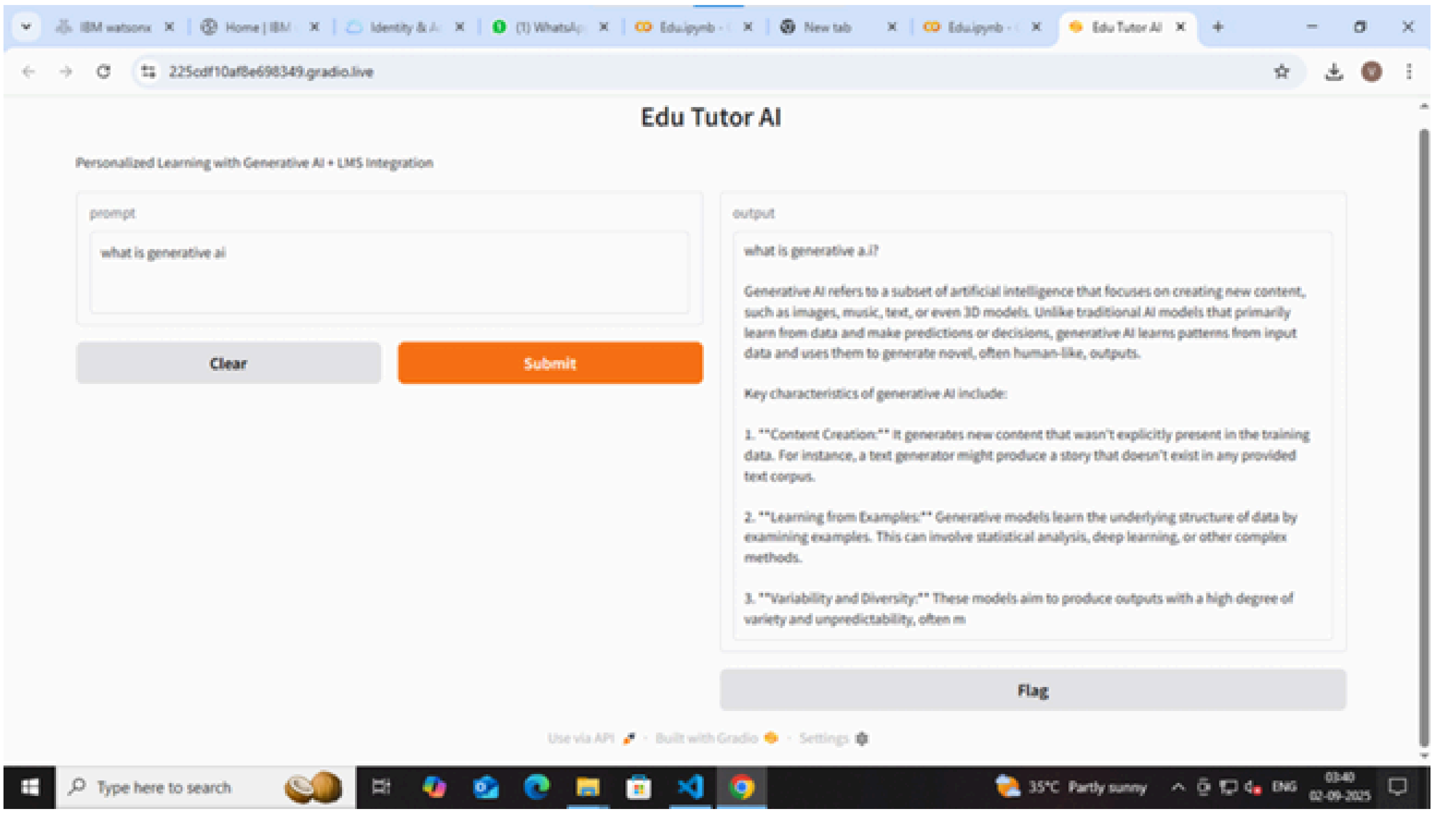
Personalized Learning with Generative AI + LMS Integration

prompt

what is generative ai

output

```
processing | 142.9/187.3s
```



Edu Tutor AI

Personalized Learning with Generative AI + LMS Integration

prompt

what is generative ai

output

what is generative a.i?

Generative AI refers to a subset of artificial intelligence that focuses on creating new content, such as images, music, text, or even 3D models. Unlike traditional AI models that primarily learn from data and make predictions or decisions, generative AI learns patterns from input data and uses them to generate novel, often human-like, outputs.

Key characteristics of generative AI include:

1. **Content Creation:** It generates new content that wasn't explicitly present in the training data. For instance, a text generator might produce a story that doesn't exist in any provided text corpus.

2. **Learning from Examples:** Generative models learn the underlying structure of data by examining examples. This can involve statistical analysis, deep learning, or other complex methods.

3. **Variability and Diversity:** These models aim to produce outputs with a high degree of variety and unpredictability, often m

8. ADVANTAGES & DISADVANTAGES

Advantages: - Fast, natural-language responses
 - Easy to use interface

- Free and accessible via browser

Disadvantages: - Depends on API availability
- Limited by token usage and response length
- No voice input or multilingual support (yet)

9. CONCLUSION

EduTutor AI serves as a practical, beginner-friendly AI tutoring app leveraging IBM Watsonx. It demonstrates the integration of cloud AI models with front-end frameworks like Gradio.

10. FUTURE SCOPE

- Add voice input
- Expand to subject-specific modules
- Add multi-language support
- Use authentication for user tracking

11. APPENDIX (Source Code)

```
import gradio as gr
 import torch
 from transformers import AutoTokenizer, AutoModelForCausalLM

 # Load model and tokenizer
 model_name = "ibm-granite/granite-3.2-2b-instruct"
 tokenizer = AutoTokenizer.from_pretrained(model_name)
 model = AutoModelForCausalLM.from_pretrained(
 model_name,
 torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
 device_map="auto"
 )

 # Function to get response
 def ask_question(prompt):
 inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
```

```
outputs = model.generate(
**inputs,
max_new_tokens=200,
temperature=0.7,
top_p=0.9,
do_sample=True
)
generated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)

if generated_text.startswith(prompt):
return generated_text[len(prompt):].strip()
return generated_text.strip()

# Gradio UI
demo = gr.Interface(
fn=ask_question,
inputs=gr.Textbox(lines=3, placeholder="Ask Edu Tutor AI..."),
outputs="text",
title="Edu Tutor AI",
description="Personalized Learning with Generative AI + LMS Integration"
)

demo.launch()
```

Dataset Link

https://www.ibm.com/docs/en/watsonx-as-a-service?topic=models-granite-13b-instruct

GitHub & Project Demo Link GitHub Repo:

https://github.com/asbdu340230534005001110 47/Edu-Tutor-AI--main-.git