**PROJECT REPORT: EDUCATIONAL AI: PERSONALIZED LEARNING GENERATIVE AI WITH IBM**

**TEAM ID: NM2025TMID03509**

**TEAM LEADER**: PREETHI.R

**TEAM MEMBER:**

* PRISITHA.R
* PRIYADHARSHINI.M
* PRIYANKA.M
* RAJAPRAVEENA.N

**ABSTRACT**

This project presents an **AI-powered Educational Assistant** developed using the IBM Granite-3B Instruct model hosted on Hugging Face. It is designed to aid students and educators by providing two core functionalities:

1. **Concept Explanation** – Offers detailed explanations for any topic input by the user, complete with examples.
2. **Quiz Generator** – Dynamically generates quizzes (multiple choice, true/false, short answer) along with answers based on the given topic.

Developed in **Python** using **Google Colab**, **Transformers**, and **Gradio**, this application allows for interactive use via a web UI. It leverages **NLP and language modeling** to simplify learning and enhance engagement in self-paced education. The assistant can support a wide range of academic subjects and is scalable for broader use in edtech platforms.

| **S. NO.** | **CONTENT** |
| --- | --- |
| **1** | **Introduction** |
| **2** | **Tools and Technologies Used** |
| **3** | **Implementation Details** |
| **4** | **User Interface Design** |
| **5** | **Output Demonstrations** |
| **6** | **GitHub Deployment** |
| **7** | **Conclusion** |
| **8** | **Future Enhancements** |
| **9** | **References** |

**AGENDA OF THE PROJECT REPORT**

**1.INTRODUCTION**

The rise of artificial intelligence has opened up transformative opportunities in the education sector. Among these, large language

models have become central to automating content creation, student assistance, and interactive tutoring.

This project explores the use of **IBM's Granite-3.2-2B-Instruct** model, a powerful open-source LLM capable of generating human-like text, to assist students in learning by:

* Explaining difficult concepts with contextual examples.
* Generating quizzes to test knowledge retention.

**Key Goals:**

* Simplify complex topics for students.
* Help teachers save time by auto-generating quizzes.
* Provide an intuitive UI for seamless interaction.

**2. TOOLS AND TECHNOLOGIES USED**

**Tools and Libraries Used:**

* **Google Colab** – For cloud-based Python execution.

A screenshot of a computer

AI-generated content may be incorrect.

* A screenshot of a computer

  AI-generated content may be incorrect.**Hugging Face Transformers** – For loading IBM Granite model.
* **Gradio** – To build a web-based user interface.
* **Torch** – For model execution with CUDA/GPU support.

**3. IMPLEMENTATION**

* **Model Setup:**

model\_name = "ibm-granite/granite-3.2-2b-instruct"

tokenizer = AutoTokenizer.from\_pretrained(model\_name)

model = AutoModelForCausalLM.from\_pretrained(model\_name, ...)

* **Function to Generate Responses:**

def generate\_response(prompt, max\_length=512):

...

outputs = model.generate(...)

return decoded\_output

A screenshot of a computer

AI-generated content may be incorrect.

* **Concept Explanation:**

def concept\_explanation(concept):

prompt = f"Explain the concept of {concept} in detail with examples:"

return generate\_response(prompt, max\_length=800)

* **Quiz Generation:**

def quiz\_generator(concept):

prompt = f"Generate 5 quiz questions about {concept}..."

return generate\_response(prompt, max\_length=1000)

A screenshot of a computer program

AI-generated content may be incorrect.

**4. USER INTERFACE AND FLOW**

**Gradio Interface Setup:**

* Two tabs:
  + *Concept Explanation*
  + *Quiz Generator*

with gr.Tabs():

with gr.TabItem("Concept Explanation"):

...

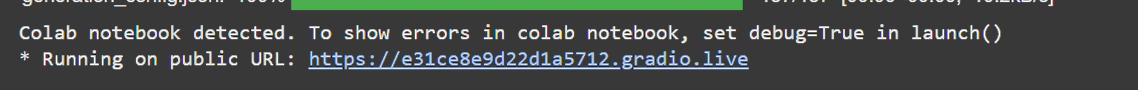
with gr.TabItem("Quiz Generator"):

...

**Execution Environment:**  
The application was executed using a **T4 GPU** runtime in **Google Colab**, ensuring fast model inference.

**Authen tication Warning**  
While the app works without a Hugging Face token, using a token (HF\_TOKEN) can improve reliability by accessing private models or increasing rate limits.

**5. OUTPUT DEMONSTRATIONS**

Click on the URl to open the Gradio Application click on the link. You can View the Application is running in the other tab

**1. Concept Explanation Output:**

User Input: python  
Generated Output:  
Explanation on Python’s syntax, code readability, history, and usage, with code samples.

A screenshot of a computer

AI-generated content may be incorrect.

**2. Quiz Generator Output:**

User Input: physics  
Generated Output:  
A list of 5 quiz questions (MCQs, True/False, Short Answer) and an **ANSWERS** section.

A screenshot of a computer

AI-generated content may be incorrect.

**6. GITHUB DEPLOYMENT**

**GitHub Repository**

The complete source code, including model setup, Gradio interface, and executable scripts, has been uploaded to a public GitHub repository for easy access, collaboration, and version control.

**Repository Link:**  
🔗 <https://github.com/Priyadharshini192006/IBM-Project>

**Contents of the Repository:**

| **File Name** | **Description** |
| --- | --- |
| README.md | Provides project overview and instructions. |
| edututorai.py | Python script containing full Gradio app code. |

A screenshot of a computer

AI-generated content may be incorrect.

This deployment ensures transparency, allows others to contribute, and enables continuous improvement and tracking of project evolution.

**7. CONCLUSION & FUTURE SCOPE**

**Conclusion**  
This AI-based Educational Assistant demonstrates how powerful LLMs like IBM Granite can enhance the educational experience. The system is capable of dynamic text generation, adaptive learning support, and content creation, all through a user-friendly interface.

**8. FUTURE ENHANCEMENTS:**

* Add multilingual support.
* Integrate voice-based interaction using Speech-to-Text APIs.
* Save quiz results to a database for student progress tracking.
* Allow teachers to fine-tune question formats.
* Add model-switching (option to choose between open LLMs like Mistral, LLaMA, or GPT).

**9. REFERENCES**

1. IBM Granite Model: <https://huggingface.co/ibm-granite/granite-3.2-2b-instruct>
2. Hugging Face Transformers Library: <https://huggingface.co/docs/transformers>
3. Gradio Documentation: <https://www.gradio.app/>
4. PyTorch Documentation: <https://pytorch.org/docs/stable/index.html>
5. Google Colab: <https://colab.research.google.com/>