EduTutor AI – Project Documentation

Team Members:

• Famitha. S. Parveen

• Gayathri. K

• Dharshini. S

• Dharshini. C

# 1. Introduction

EduTutor AI is a personalized learning assistant created under the Naan Mudhalvan initiative. The project’s primary goal is to provide an AI-based platform that can help students learn concepts in a simplified manner. By using IBM Granite models integrated through Hugging Face, the chatbot can explain topics, generate quizzes, and provide personalized learning guidance. Our team decided to work on this project because it combines Generative AI technology with education, making learning both effective and interactive.

# 2. Project Overview

The EduTutor AI project is designed to showcase how AI can be applied in the education field. The purpose of this project is to build a simple and interactive chatbot application that can answer student queries and generate quizzes. The chatbot runs in a cloud environment using Google Colab, making it accessible to anyone with an internet connection. This eliminates the need for high-end systems, as all computation is handled in the cloud. Features include concept explanations, quiz generation, and an easy-to-use Gradio interface.

# 3. Activity-1: Exploring Naan Mudhalvan Portal

In the first step, we explored the Naan Mudhalvan SmartInternz portal. After logging in with our credentials, we went to the ‘Projects’ section and selected EduTutor AI. We were provided with project resources, including documentation and setup instructions. The portal also helped us track our progress and submit deliverables. This initial step was important because it guided us in understanding the scope and flow of the project.

# 4. Activity-2: Choosing IBM Granite Model

The second activity focused on selecting an appropriate IBM Granite model from Hugging Face. We created our Hugging Face accounts and searched for Granite models. Among the available options, we chose ‘granite-3.2-2b-instruct’, which is efficient and suitable for our project. This model is lightweight and optimized for running inside Google Colab, making it ideal for our educational chatbot. Choosing the right model was a critical step, as it determined the quality of responses generated by our application.

# 5. Activity-3: Running in Google Colab

In this step, we worked on executing our EduTutor AI project in Google Colab. First, we opened a new notebook and changed the runtime type to T4 GPU to improve processing speed. We then installed the necessary dependencies using pip commands such as:  
  
!pip install transformers torch gradio -q  
  
After installation, we uploaded our code into the notebook and executed it step by step. The model was successfully downloaded, and the application launched with a Gradio interface. We could interact with the chatbot through a URL provided by Colab, confirming that our project was working correctly.

# 6. Activity-4: Uploading to GitHub

After completing the Colab execution, we uploaded our project files to GitHub. We first created a GitHub repository and then uploaded our .py notebook file. This ensured version control and allowed us to present our work professionally. By storing the project on GitHub, we made it accessible to mentors and peers, while also gaining experience with collaborative tools widely used in software development.

# 7. Conclusion

EduTutor AI successfully demonstrated how Generative AI can be applied in education. By using Google Colab and IBM Granite models, we created a simple yet powerful chatbot that can assist students in learning. The activities helped us understand the importance of portals, model selection, cloud execution, and version control. In the future, EduTutor AI can be expanded with advanced features such as voice-based interaction and support for multiple languages. Overall, this project gave us valuable practical knowledge in AI, cloud platforms, and collaborative development.