# EDU TUTOR AI PERSONALIZED LEARNING

## PROJECT DOCUMENTION

### 1. Introduction

Project Title: Edu Tutor AI 🎓

**Team Leader : JAINULDEEN J**
**Team member : MEGANATHANG**
**Team member : JOHN BRITTO J**
**Team member : MOHAMED BAKITH N**
**Team member : JAYASRIDHAR S**

### 2. Project Overview

Purpose:

The purpose of the Edu Tutor AI is to provide a personalized and interactive learning experience for students of all ages. By leveraging AI and powerful language models, this assistant can explain complex concepts, generate quizzes, and offer real-time feedback. It aims to make education more accessible, engaging, and tailored to individual learning styles. For educators, it serves as a supplementary tool to create dynamic lesson plans and assessments. Ultimately, this assistant uses technology to support both students and teachers in achieving educational goals.

Features:

Concept Explainer

Key Point: Simplified learning

Functionality: Explains complex concepts in a clear, easy-to-understand language.

Quiz Generator

Key Point: Personalized assessment

Functionality: Creates diverse quiz questions (multiple choice, true/false, short answer) on any given topic.

Code Explanation

Key Point: Code comprehension

Functionality: Breaks down code snippets and explains their function line-by-line.

Interactive Interface

Key Point: User-friendly experience

Functionality: Provides a simple, conversational interface for students and educators to interact with the assistant.

Multimodal Input Support

Key Point: Flexible data handling

Functionality: Accepts text, code snippets, and PDFs for document and code analysis.

### 3. Architecture

Frontend (Gradio):

The frontend is built with Gradio, which provides a user-friendly and intuitive web interface. It has different tabs for various functionalities like concept explanation and quiz generation, making it easy to navigate.
Backend (Python with Hugging Face Transformers):
The backend is a Python application that handles the core logic. It uses the Hugging Face Transformers library to load and manage the AI models. This setup allows for efficient processing of user queries and generation of educational content.
LLM Integration (IBM Granite 3.2-2B-Instruct):
The Granite 3.2-2B-Instruct model from IBM is used for all natural language understanding and generation tasks. Prompts are specifically designed to produce accurate explanations, detailed answers, and diverse quiz questions.

4. Setup Instructions
Prerequisites:
Python 3.9 or later
pip and virtual environment tools
Internet access for model download
Installation Process:
Clone the repository from the source.
Install dependencies from requirements.txt using pip.
Run the application via the gradio_app.py script.
Access the interface through the provided local URL.

5. Folder Structure
gradio_app.py: The main script for launching the Gradio interface and handling user interactions.
requirements.txt: Lists all necessary Python libraries for the project.
prompts/: (Optional) Directory for storing pre-defined prompts for various functionalities.

6. Running the Application
To run the project, simply execute the main Python script. The Gradio interface will launch in your browser, and you can start interacting with the Edu Tutor AI to get explanations and generate quizzes.

7. API Documentation
While this version is a single-script application, the core functions can be exposed as API endpoints for future integrations.
concept_explanation(concept): Explains a given educational concept.
quiz_generator(concept): Creates quiz questions on a specific topic.
code_explainer(code_snippet): Provides a breakdown of a given code snippet.

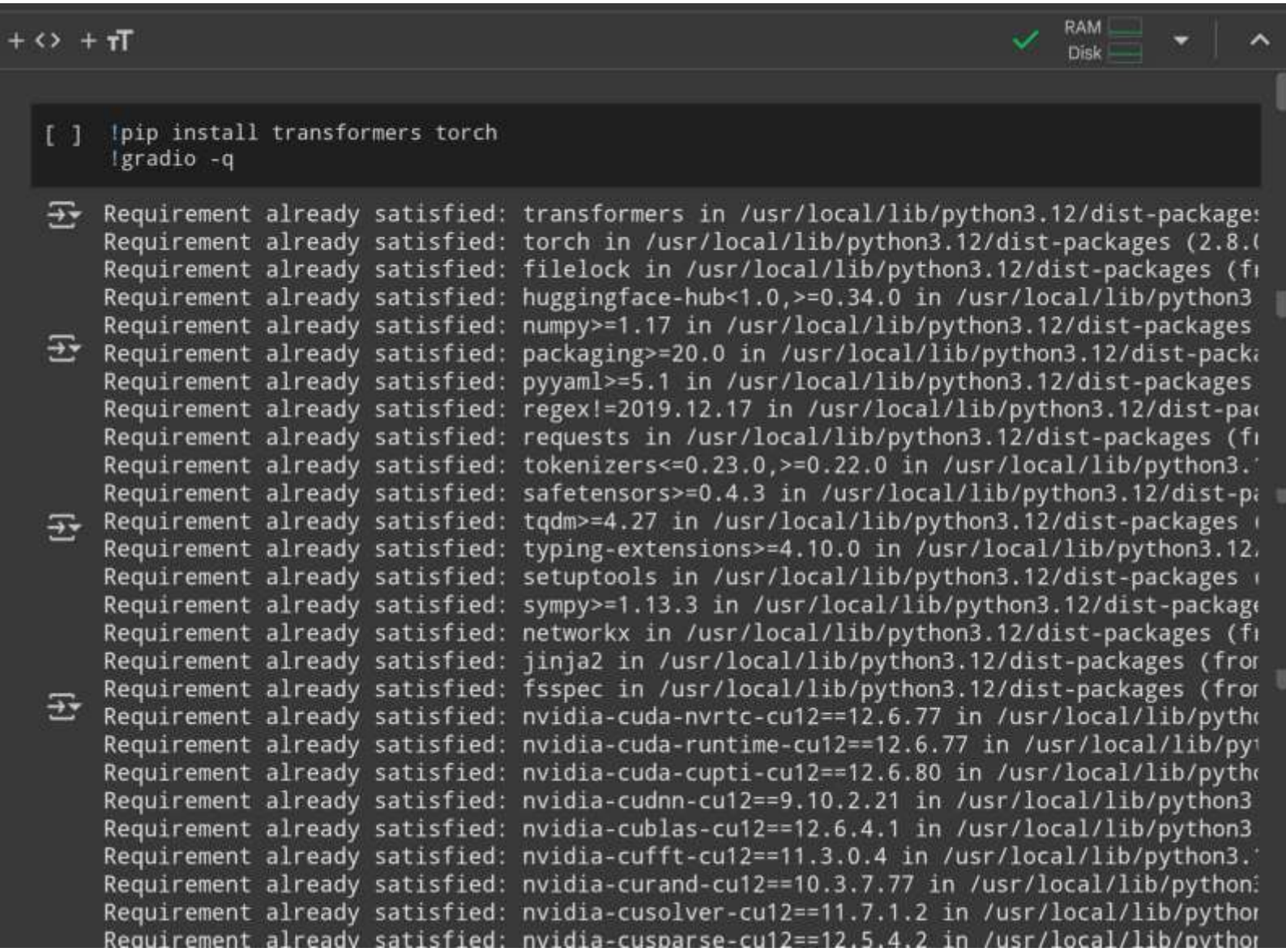
8. Authentication

This version of the project runs in an open environment for demonstration purposes.
However, for a secure deployment, the following authentication methods could be integrated:
Token-based authentication: This could use JSON Web Tokens (JWT) or API keys to secure endpoints.
OAuth2: This would allow for secure sign-in through third-party services.
Role-based access: Different user roles (e.g., student, teacher, admin) could have different levels of access and functionality within the application.
Planned future enhancements include user sessions and history tracking to create a more personalized and secure learning experience.

9. User Interface
The interface is designed to be simple and accessible for students. It includes:
A markdown-based welcome message.
Intuitive tabs for different functions.
Text boxes for input and output.
Buttons to trigger the AI's response.

10. Testing
The application was tested through manual checks to ensure:
Response Accuracy: The explanations and answers provided are correct.
Quiz Diversity: The generated quizzes include a variety of question types.
Functionality: The interface works as expected without errors.

11.SCREENSHOT


RAM
Disk
[ ] !pip install transformers torch
!gradio -q
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-p
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/pyth
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/py
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/pyth
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/pytho
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/pytho

EDU TUTOR AI

M

RAM
Disk

1m

```python
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" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=512):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

    if torch.cuda.is_available():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = response.replace(prompt, "").strip()
    return response
```

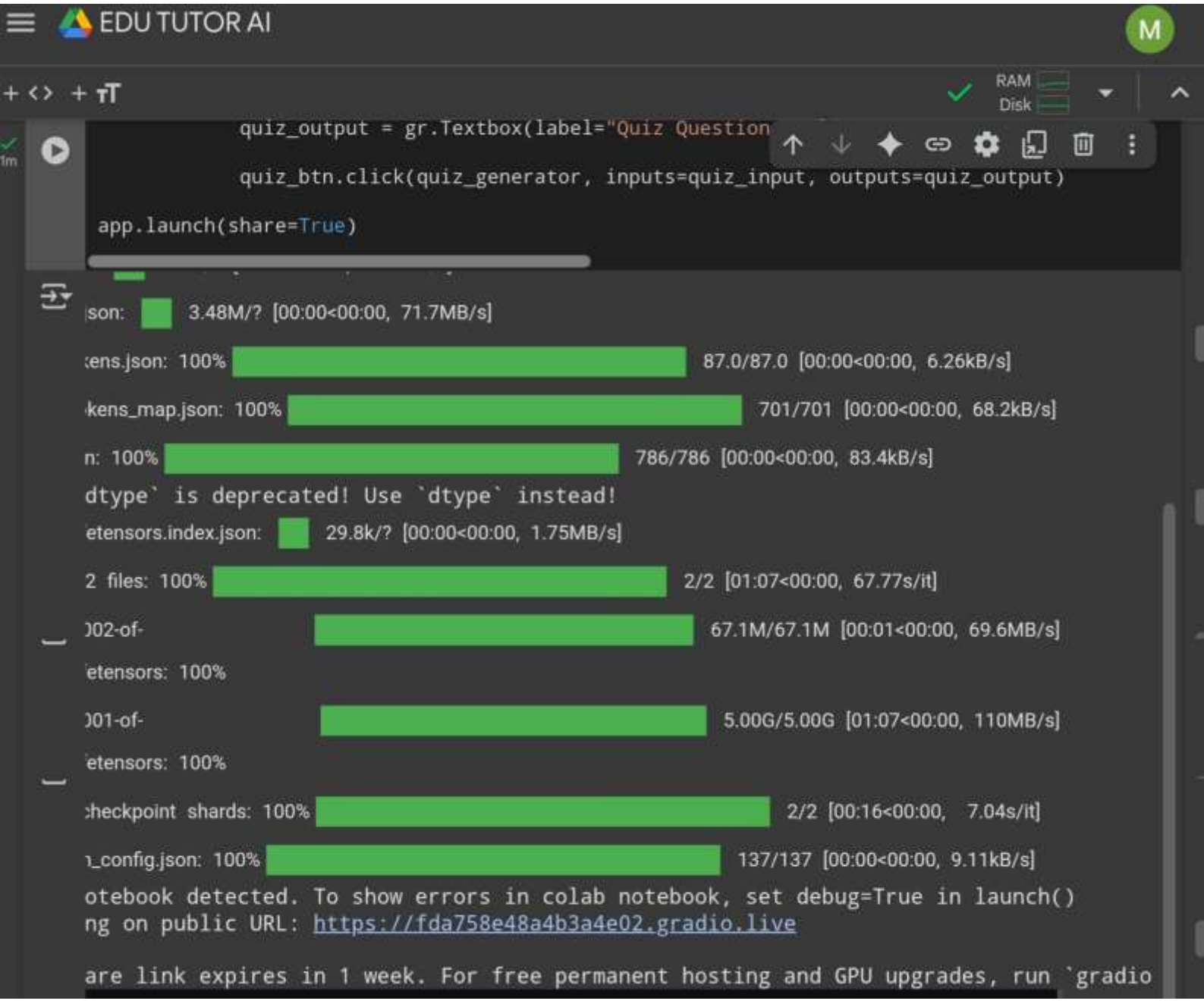
EDU TUTOR AI

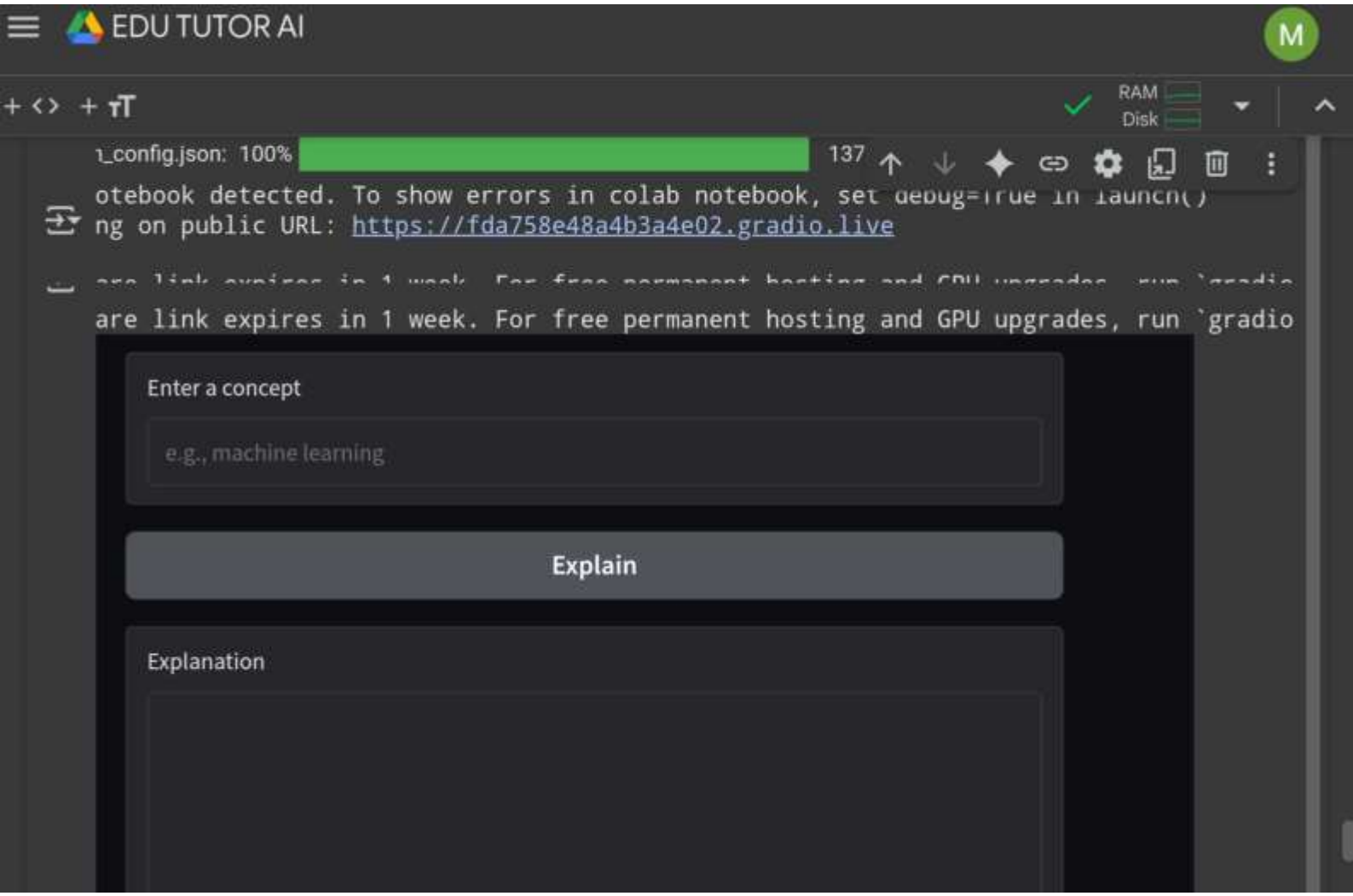
M

RAM
Disk

```
def concept_explanation(concept):
    prompt = f"Explain the concept of {concept} in detail with examples:"
    return generate_response(prompt, max_length=800)

def quiz_generator(concept):
    prompt = f"Generate 5 quiz questions about {concept} with different question types (mult
    return generate_response(prompt, max_length=1000)

# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# Educational AI Assistant")

    with gr.Tabs():
        with gr.TabItem("Concept Explanation"):
            concept_input = gr.Textbox(label="Enter a concept", placeholder="e.g., machine l
            explain_btn = gr.Button("Explain")
            explanation_output = gr.Textbox(label="Explanation", lines=10)

            explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanation

        with gr.TabItem("Quiz Generator"):
            quiz_input = gr.Textbox(label="Enter a topic", placeholder="e.g., physics")
            quiz_btn = gr.Button("Generate Quiz")
            quiz_output = gr.Textbox(label="Quiz Questions", lines=15)

            quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)

app.launch(share=True)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:104: UserWarn
Error while fetching `HF_TOKEN` secret value from your vault: 'Requesting secret HF_
You are not authenticated with the Hugging Face Hub in this notebook.
If the error persists, please let us know by opening an issue on GitHub (https://git
  warnings.warn(
```


EDU TUTOR AI
M
RAM
Disk
quiz_output = gr.Textbox(label="Quiz Question
quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)
app.launch(share=True)
son: 3.48M/? [00:00<00:00, 71.7MB/s]
ens.json: 100% 87.0/87.0 [00:00<00:00, 6.26kB/s]
kens_map.json: 100% 701/701 [00:00<00:00, 68.2kB/s]
n: 100% 786/786 [00:00<00:00, 83.4kB/s]
dtype` is deprecated! Use `dtype` instead!
etensors.index.json: 29.8k/? [00:00<00:00, 1.75MB/s]
2 files: 100% 2/2 [01:07<00:00, 67.77s/it]
002-of- 67.1M/67.1M [00:01<00:00, 69.6MB/s]
etensors: 100%
001-of- 5.00G/5.00G [01:07<00:00, 110MB/s]
etensors: 100%
heckpoint shards: 100% 2/2 [00:16<00:00, 7.04s/it]
_config.json: 100% 137/137 [00:00<00:00, 9.11kB/s]
otebook detected. To show errors in colab notebook, set debug=True in launch()
ng on public URL: https://fda758e48a4b3a4e02.gradio.live
are link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio


EDU TUTOR AI
M
RAM
Disk
1_config.json: 100%
137
otebook detected. To show errors in colab notebook, set debug=True in launch()
ng on public URL: https://fda758e48a4b3a4e02.gradio.live
are link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio
Enter a concept
e.g., machine learning
Explain
Explanation

12.KnownIssues

- Occasionallongresponse timeforlargePDFs.
- Forecastinglimitedto structuredCSVdata.
- Requires stable internetforIBMAPI access.

13.FutureEnhancements

- Addvoice-basedinteraction.
- Expandforecasting toinclude traffic&pollutiondata.
- Developamobileappversion.
- IntegratewithIoTsmartsensors.
- Supportmulti-languageoutputsfor localcommunities.