

- **EDUTUTOR AI: PERSONALIZED LEARNING WITH GENERATIVE AI AND LMS INTEGRATION**

Project Documentation

1. Introduction

Project Title—> EduTutor AI: personalized Learning with generative AI and LMS integration

Teamleader:Nandhini S

Teammembers:

- 1.Abinaya v
- 2.Susmitha S
- 3.Dhivya Dharshini M

2. Project Overview

Purpose:

EduTutorAI is designed to provide personalized, AI-powered educational support for students. The system helps learners understand concepts, practice with quizzes, and receive interactive explanations. It reduces the need for constant teacher supervision by acting as a smart tutor that adapts to each student's needs.

Features:

Concept Explanation –

Explains any topic (like Physics, Math, Computer Science, etc.) in simple, step-by-step language.

Can adapt explanations to the student's level (school, college, or competitive exams).

Practice & Quizzes –

Generates custom questions, quizzes, and exercises instantly.

Gives step-by-step solutions and hints to improve learning.

Quiz Generator

Key Point: Practice learning

Functionality: Creates quizzes and MCQs dynamically for any subject/topic.

Personalized Learning Path

Key Point: Adaptive guidance

Functionality: Suggests study plans based on student progress.

Doubt Solving Chatbot

Key Point: Conversational interface

Functionality: Students can ask questions in natural language and get answers.

Report & Progress Tracking

Key Point: Performance insights

Functionality: Generates reports on learning outcomes and weak areas.

Multimodal Input Support

Key Point: Flexibility

Functionality: Accepts text/PDFs for summarization and question generation.

Gradio/Streamlit UI

Key Point: Easy-to-use dashboard

Functionality: Provides an interactive interface for students/teachers.

3. Architecture

Frontend (Streamlit/Gradio):

Provides dashboards, quiz pages, chat interface, and student progress reports.

Backend (FastAPI):

Hosts APIs for question generation, explanations, and report creation.

LLM Integration (Hugging Face Granite / Open-Source Models):

Handles NLP tasks such as summarization, concept explanation, and quiz generation.

Vector Search (FAISS / Pinecone optional):

Embeds study materials and enables semantic search.

ML Modules:

Used for progress tracking, quiz difficulty adjustment, and recommendation system.

4. Setup Instructions

Prerequisites:

Python 3.9+

pip & virtual environment

API keys (Hugging Face, optional DB)

Installation:

1. Clone repository

2. Install dependencies (pip install -r requirements.txt)

3. Configure .env with API keys

4. Run backend (FastAPI)

5. Launch frontend (Streamlit/Gradio)

5. Folder Structure

```
edututor_ai/
├── app/          # FastAPI backend
├── app/api/      # API routes (chat, quiz, report)
├── ui/           # Streamlit/Gradio frontend
├── quiz_generator.py # Quiz generation logic
├── concept_explainer.py # Handles topic explanations
├── report_generator.py # Creates student reports
├── document_embedder.py # For semantic search
└── main.py        # Entry point
```

6. Running the Application

Start FastAPI server

Run Streamlit/Gradio dashboard

Navigate UI (quiz, chat, reports)

Upload study material or ask questions

7. API Documentation

POST /ask – Student query → AI response

POST /generate-quiz – Generate quiz for subject/topic

GET /progress-report – Fetch student performance insights

POST /upload-material – Upload notes/PDF for question generation

8. Authentication

JWT-based authentication

Role-based access (student, teacher, admin)

9. User Interface

Sidebar navigation

Quiz & reports dashboard

Real-time chatbot interface

Progress visualization (charts, scorecards)

10. Testing

Unit Testing: Quiz generation, explanation accuracy

API Testing: Postman/Swagger UI

Manual Testing: Student usability, quiz difficulty check

11. Known Issues

Limited subject coverage in early version

Requires good internet connection for LLM queries

12. Future Enhancements

Voice-based tutoring

Mobile app integration

Multilingual support

Gamified learning features

13. Advantages

Saves time for both students and teachers.

Provides personalized learning.

Easy to deploy in Google Colab (no heavy setup).

Cost-effective (open-source based).

14. Screenshots:

HUGGING FACE

The screenshot shows the Hugging Face model card for the model `ibm-granite/granite-3.2-2b-instruct`. The card includes the following details:

- Model Summary:** Granite-3.2-2B-Instruct is a 2-billion-parameter, long-context AI model fine-tuned for thinking capabilities. Built on top of [Granite-3.1-2B-Instruct](#), it has been trained using a mix of permissively licensed open-source datasets and internally generated synthetic data designed for reasoning tasks. The model allows controllability of its thinking capability, ensuring it is applied only when required.
- Developers:** Granite Team, IBM
- Safetensors:** Model size: 2.53B params, Tensor type: BF16, Chat template, Files info
- Inference Providers:** Text Generation, Ask for provider support
- Downloads last month:** 8,000 (represented by a line chart)

The screenshot shows the Hugging Face search results for the query `IBM-granite`. The results include:

- Models:**
 - `ibm-granite/granite-embedding-english-x2` (selected)
 - `ibm-granite/granite-embedding-small-english-r2`
 - `ibm-granite/granite-speech-3.3-0b`
 - `ibm-granite/granite-speech-3.3-2b`
 - `ibm-granite/granite-3.3-2b-instruct`
- Trending:**
 - `microsoft/VibeVoice-1.5B`
 - `openbmb/MiniCPM-V-4_5`
 - `tencent/Hunyuan-MT-7B`
 - `meituan-longcat/LongCat-Flash-Chat`
 - `DeepSite v2`
 - `FastVLM WebGPU`

CODING

```
!pip install transformers torch gradio -q

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
        )

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Back Failure with Colab.
```

2025-08-27

- Python runtimes upgraded to Python 3.12. [GitHub](#)
- Julia runtimes upgraded to Julia 1.11. [GitHub](#)
- Launched [Interactive Slideshow Mode](#) for lectures. It provides a better viewing experience for presentations making lessons more dynamic.
- Launched [AI toggle per notebook](#). As requested by instructors, Colab now supports a toggle for disabling AI features at a notebook level to allow instructors and students to focus on learning without assistance.
- Python package upgrades
 - accelerate 1.9.0 → 1.10.1
 - aiohttp 3.11.15 → 3.12.15
 - aniso8601 4.9.0 → 4.10.0
 - bigframes 2.11.0 → 2.17.0
 - bigquery-magic 0.10.1 → 0.10.3
 - blis 2.6.1-1 → 3.7.2
 - colorama 1.4.6 → 4.0.0
 - diffusers 0.14.0 → 0.35.1
 - fastai 2.7.19 → 2.7.20
 - gcfs 2025.3.0 → 2025.7.0
 - google-generative-1.26.0 → 1.31.0
 - gradio 5.38.0 → 5.43.1
 - h2 4.2.0 → 4.3.0
 - huggingface-hub 0.33.4 → 0.34.4
 - imbalanced-learn 0.13.0 → 0.14.0
 - jax 0.5.2 → 0.5.3
 - jaxlib 0.5.1 → 0.5.3
 - keras 3.8.0 → 3.10.0

OUTPUT

```
merges.txt: 442/442 [00:00<00:00, 5.31MB/s]
tokenizer.json: 3.48M/3.48M [00:00<00:00, 32.7MB/s]
added_tokens.json: 100% [00:00<00:00, 2.14kB/s]
special_tokens_map.json: 100% [00:00<00:00, 18.64kB/s]
config.json: 100% [00:00<00:00, 32.04kB/s]
"torch.dtype" is deprecated! Use "dtype" instead!
model.safetensors.index.json: 29.8M/29.8M [00:00<00:00, 1.40MB/s]
Fetching 2 files 100% [00:00<00:00, 85.70kB/s]
model-00001-of-00002.safetensors: 100% [01:25<00:00, 5.00GB/5.00G [01:25<00:00, 62.3MB/s]
model-00002-of-00002.safetensors: 100% [00:10<00:00, 6.51MB/s]
Loading checkpoint shards: 100% [00:35<00:00, 14.57kB/s]
generation_config.json: 100% [00:00<00:00, 9.63kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://7c8dc87493ef6313aa.gradio.live
```

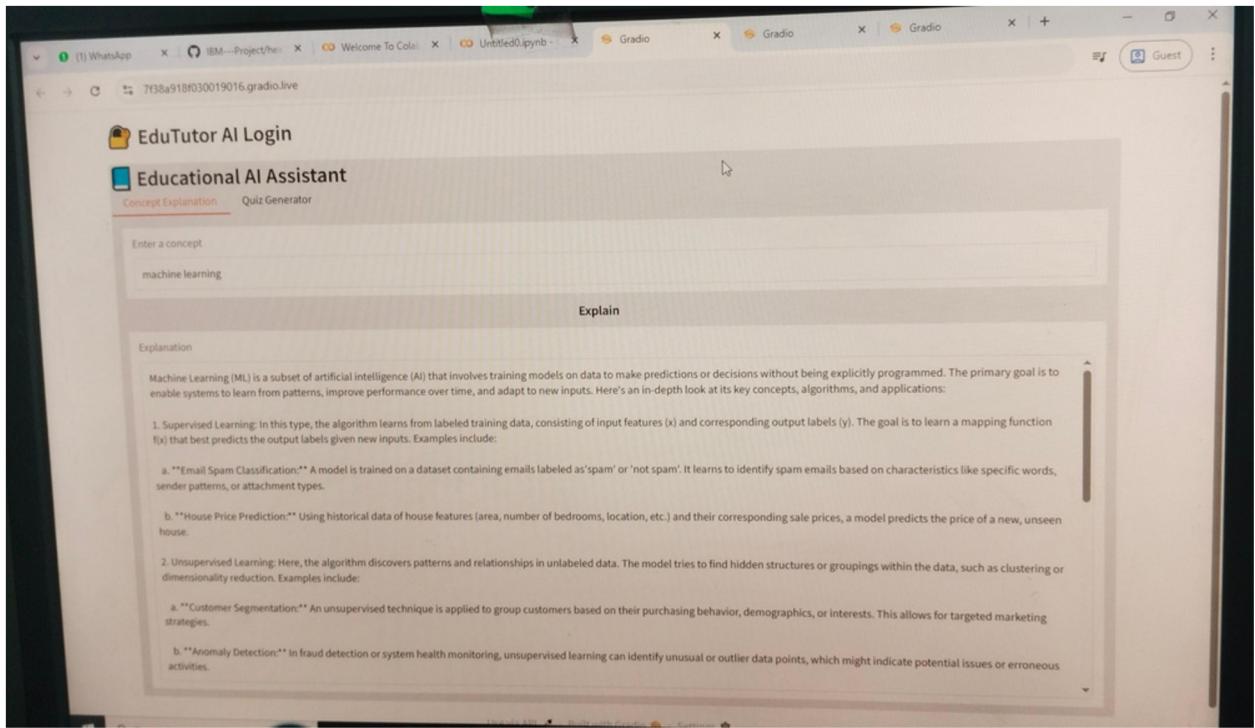
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run 'gradio deploy' from the terminal in the working directory to deploy to Hugging Face.

EduTutor AI Login

Username:

Password:

11:08 Python 3



15. Conclusion

EduTutor AI is a smart educational assistant that combines AI with interactive learning. It enhances student productivity, improves understanding, and reduces dependency on traditional methods. With future improvements, it can become a complete AI tutor platform.