PROJECT REPORT

EduTutor AI: Personalized Learning with Generative AI and LMS Integration

1.INTRODUCTION:

- Project Title: EduTutor AI: Personalized Learning with Generative AI and LMS Integration
- Team Member: MADHUMITHA.A
- Team Member: SUBASRI.M
- Team Member: LOGASHREE.B
- Team Member: KEERTHANA.M
- Team Member: KEERTHIKA.S

2.PROJECT OVERVIEW:

The project **EduTutor AI: Educational Assistant** is an interactive learning platform powered by Generative AI**.** It is designed to make studying easier, smarter, and more engaging for students by providing concept explanations, topic-based quizzes, and instant evaluation.

The system is built using **Python, Hugging Face Transformers, PyTorch, and Gradio**, with the IBM Granite granite-3.2-2b-instruct model as the AI engine. The model generates detailed explanations and concise summaries, while a predefined quiz database helps students test their understanding in subjects such as Java and Physics.

Unlike traditional learning resources, this assistant provides **AI-powered explanations**, an **interactive quiz generator**, and a **creative score display** in a single platform. The clean and user-friendly Gradio interface makes it accessible for all learners without technical expertise.

Key Objectives:

- To simplify complex concepts using AI-generated explanations.
- To enable fast revision with concise summaries.
- To provide quizzes for self-assessment and knowledge testing.
- To motivate learners with a creative score display.

Main Features:

1. **Concept Explanation & Summary** – Enter a topic (e.g., Newton’s Laws, Inheritance in Java) and receive a detailed explanation along with a short revision summary in a highlighted box.
2. **Quiz Generator** – Generates three multiple-choice questions from a predefined database for Java and Physics, allowing students to test themselves.
3. **Score Display** – Instantly evaluates responses and shows results in a visually engaging score card with gradients, stars, and correct answers.

3. SYSTEM ARCHITECTURE:

The architecture of **EduTutor AI** is divided into four main layers: **AI Layer, Backend, Frontend, and Output Layer**. Each layer plays a key role in ensuring smooth execution of concept explanations, quizzes, and score display.

➢ AI Layer (Model Integration)

- Uses **IBM Granite granite-3.2-2b-instruct**, a Generative AI model.
- Integrated through **Hugging Face Transformers** and **PyTorch**.
- Responsible for:
  - Generating **detailed explanations** of topics.
  - Producing **short, easy-to-read summaries**.
- Function used: **generate_response()** (handles tokenization, generation, decoding).

➢ Backend Layer (Python Logic)

The backend manages processing, quizzes, and evaluation.

- **Concept Module**: explain_concept_with_summary() – prompts AI for both long explanation and short summary.
- **Quiz Module**:
  - QUIZ_DB – predefined database with Java & Physics questions.
  - load_quiz() – loads questions and answer options into UI.

- **Evaluation Module**:
    - grade_quiz() – compares student answers with correct answers, calculates score, and generates a styled result card.

➢ **Frontend Layer (User Interface)**

Built with **Gradio Blocks** for simplicity and interactivity.

- **Tabs**:
    - Concept Explanation
    - Quiz Generator
    - Score Display
- **Widgets**:
    - Textbox for user input and question display.
    - Radio Buttons for quiz answers.
    - HTML for summary box and score card.
- **Custom CSS**: Buttons styled with gradient orange color and hover effect for better UI.

➢ **Output Layer (User Results)**

- **Explanations**: Long explanation with real-world examples.
- **Summaries**: Short 3–5 line key points in a **blue contrast box**.
- **Quiz**: Three MCQs per topic with four options each.
- **Score Display**: A **creative score card** with gradient, stars 🎯, and answer key.

4.SYSTEM WORKFLOW:

Stepwise Process:

1. **User Interaction**
    - Users access the home page and select the desired module (Quiz, Tutorials, or Dashboard).
    - Enter required details like name, email, and topic selection.
    - Input validation ensures all fields are properly filled.
2. **Backend Processing**
    - The system receives the request and triggers the corresponding function.
    - Handles unexpected inputs with error messages to maintain smooth operation.
3. **Data Handling**

- Fetches relevant content from the database (questions, tutorials, previous scores).
- Stores user responses for scoring and analytics.

4. **Logic Execution**
   - Quiz answers are compared with correct answers to calculate scores.
   - Tutorials are displayed based on user selection and progress.
5. **Output Generation**
   - Displays results, scorecards, feedback, and recommendations.
   - Optionally, visual elements like charts or progress bars improve clarity.
6. **User Feedback**
   - Users can retry quizzes, proceed to the next tutorial, or exit.
   - Feedback is recorded for future system enhancements.

5.SETUP & REQUIREMENTS

Software Requirements:

- Python 3.11 or higher
- IDE: VS Code / PyCharm / Google Colab
- Libraries: gradio, torch, transformers

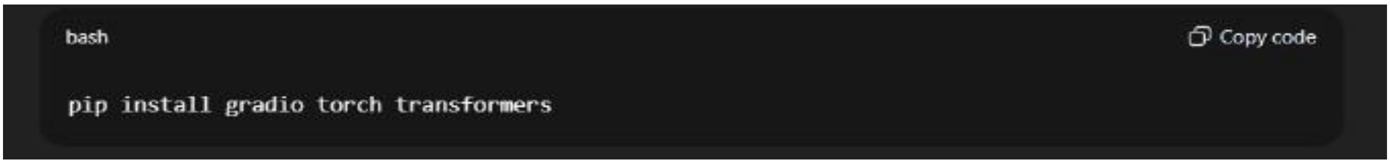
Hardware Requirements:

- Minimum 8 GB RAM
- Internet connection for library downloads
- Optional GPU for faster AI model execution

Installation Steps:

1. Install Python and set environment variables.

2.Install required libraries:



```bash
pip install gradio torch transformers
```

3. Download project files and place them in a single folder.

4. Run the main script (main.py or app.py).

5.Verify the home interface loads properly.

6.CODE EXPLANATION:

1. Main Interface (app.py)

- Handles user input and displays results using Gradio blocks.
- Example: Launch quiz interface and collect answers.

2. Backend Logic (backend.py)

- Processes inputs, validates data, calculates scores.
- Example function to calculate quiz score:

```
def calculate_score(user_answers, correct_answers):

   score = 0

for u, c in zip(user_answers, correct_answers):

    if u == c:

       score += 1

return score
```

3. Database Handling (db_handler.py)

- Fetches questions and tutorials, stores user responses, maintains logs.

7.SCREENSHOT:

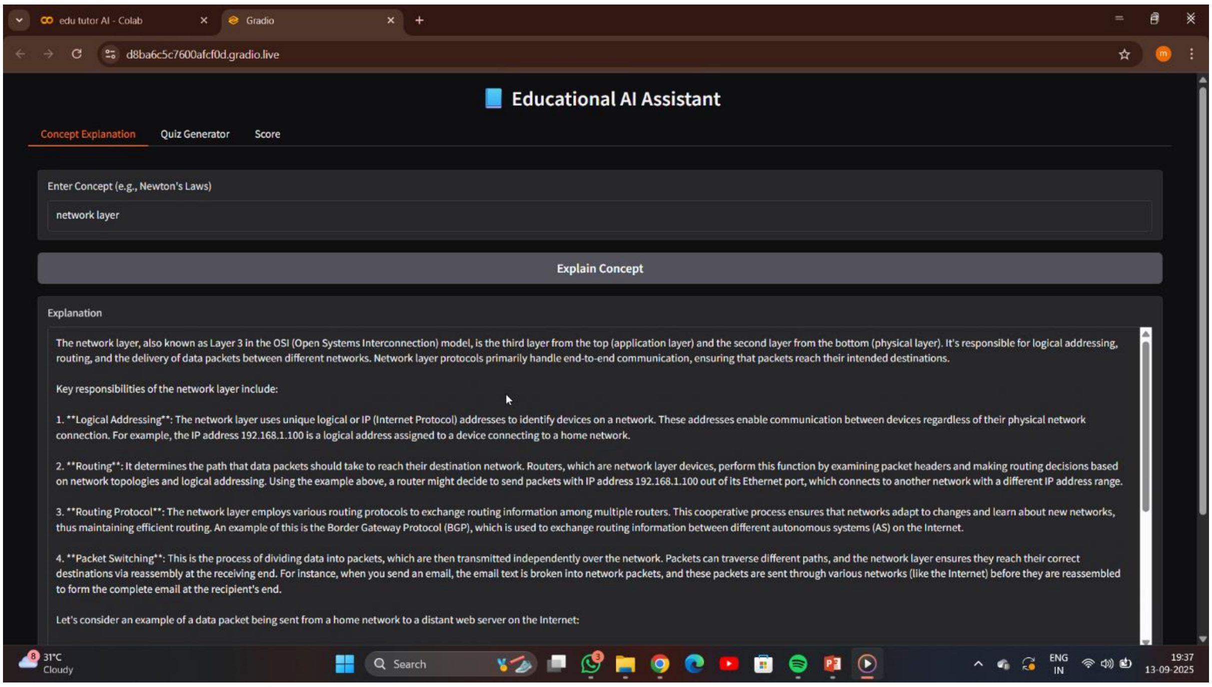
➢ Home Page / Dashboard Screenshot

Description:

- Shows the **landing interface** of EduTutorAI.
- Users can choose between modules like **Quiz**, **Tutorials**, or **Dashboard**.
- Input fields for **username** or **topic selection** are clearly visible.

➢ Concept Explanation:

- This page acts as the **entry point** for the system.
- Ensures **user-friendly navigation** by providing clear options.
- All user interactions start here, making it the **central hub**


edu tutor AI - Colab
Gradio
d8ba6c5c7600afcf0d.gradio.live
📘 Educational AI Assistant
Concept Explanation
Quiz Generator
Score
Enter Concept (e.g., Newton's Laws)

network layer

Explain Concept
Explanation

The network layer, also known as Layer 3 in the OSI (Open Systems Interconnection) model, is the third layer from the top (application layer) and the second layer from the bottom (physical layer). It's responsible for logical addressing, routing, and the delivery of data packets between different networks. Network layer protocols primarily handle end-to-end communication, ensuring that packets reach their intended destinations.

Key responsibilities of the network layer include:

1. **Logical Addressing**: The network layer uses unique logical or IP (Internet Protocol) addresses to identify devices on a network. These addresses enable communication between devices regardless of their physical network connection. For example, the IP address 192.168.1.100 is a logical address assigned to a device connecting to a home network.

2. **Routing**: It determines the path that data packets should take to reach their destination network. Routers, which are network layer devices, perform this function by examining packet headers and making routing decisions based on network topologies and logical addressing. Using the example above, a router might decide to send packets with IP address 192.168.1.100 out of its Ethernet port, which connects to another network with a different IP address range.

3. **Routing Protocol**: The network layer employs various routing protocols to exchange routing information among multiple routers. This cooperative process ensures that networks adapt to changes and learn about new networks, thus maintaining efficient routing. An example of this is the Border Gateway Protocol (BGP), which is used to exchange routing information between different autonomous systems (AS) on the Internet.

4. **Packet Switching**: This is the process of dividing data into packets, which are then transmitted independently over the network. Packets can traverse different paths, and the network layer ensures they reach their correct destinations via reassembly at the receiving end. For instance, when you send an email, the email text is broken into network packets, and these packets are sent through various networks (like the Internet) before they are reassembled to form the complete email at the recipient's end.

Let's consider an example of a data packet being sent from a home network to a distant web server on the Internet:

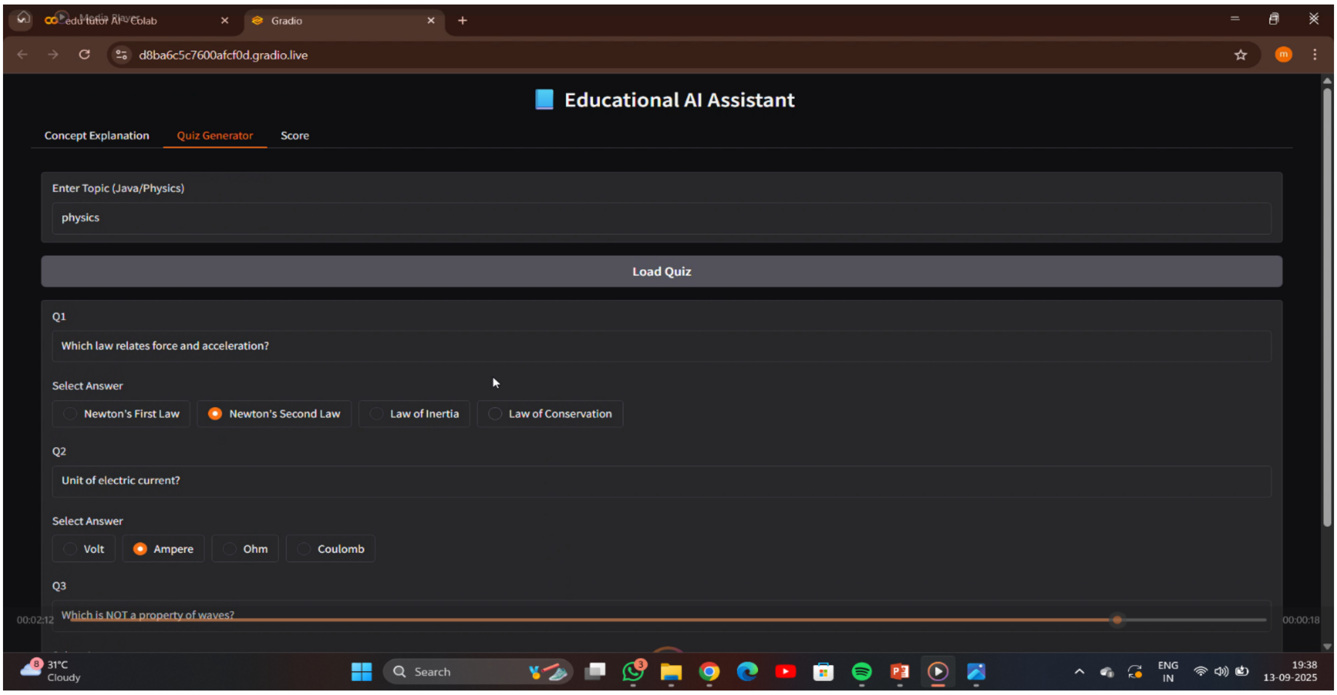
➢ Quiz Module Screenshot

Description:

- Displays a set of **multiple-choice questions**.
- Options for each question are clearly listed.
- Submit button triggers the scoring logic.

Concept Explanation:

- This module tests **user knowledge** on a particular topic.
- Designed to provide **interactive learning**, not just evaluation.
- Each question is validated before submission to prevent errors.
- Supports **instant scoring** for immediate feedback.



📘 Educational AI Assistant

Enter Topic (Java/Physics)

physics

Q1

Which law relates force and acceleration?

Select Answer

○ Newton's First Law ● Newton's Second Law ○ Law of Inertia ○ Law of Conservation

Q2

Unit of electric current?

Select Answer

○ Volt ● Ampere ○ Ohm ○ Coulomb

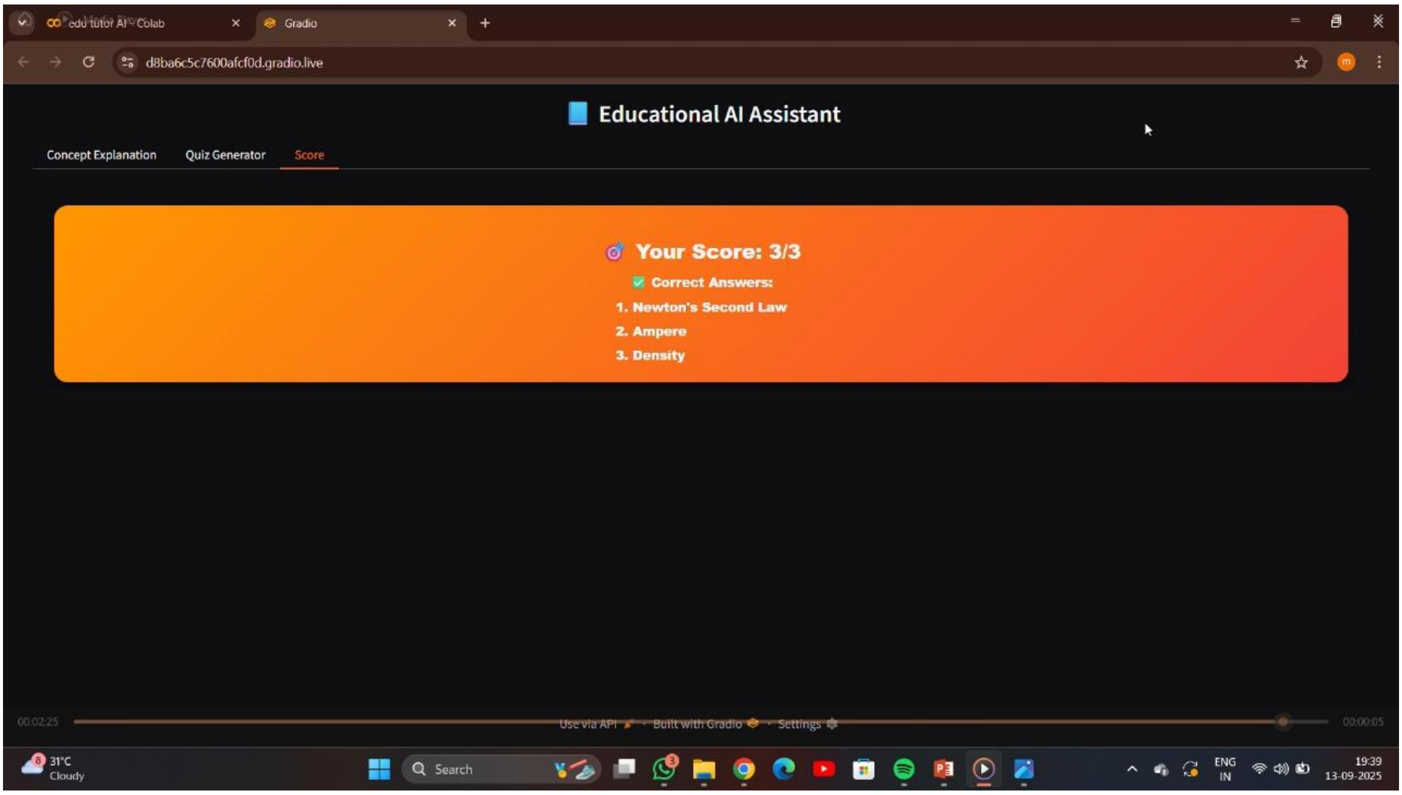
Q3

Which is NOT a property of waves?

How Scoring Works:

1. User selects answers for all questions.
2. On clicking submit, **backend logic** compares each answer with the correct answer stored in the database.

➢ Result / Score Page Screenshot



📘 Educational AI Assistant

🎯 Your Score: 3/3

✅ Correct Answers:

1. Newton's Second Law
2. Ampere
3. Density

Description:

- Displays **total score, percentage, and feedback messages**.

- Optional visualization (like progress bar or chart) to show performance.

Concept Explanation:

- Provides **real-time feedback** to the user.
- Helps learners **understand strengths and weaknesses**.
- Results are stored in the database for **future reference**.

8.TESTING

Overview:
The EduTutorAI system underwent comprehensive manual testing to ensure that all its modules—including quizzes, tutorials, and result calculation—functioned correctly and consistently. Each module was carefully examined to verify that it delivered the expected output under normal conditions, and edge cases such as empty input fields, invalid selections, or unexpected user actions were rigorously tested to ensure the system could handle them gracefully without errors or crashes. The testing process also included repeated runs to simulate multiple users interacting with the system simultaneously, verifying that performance remained stable and responsive.

Sample Test Cases:

Test Case	Input	Expected Output	Result
Quiz Module	5 correct answers	Score = 5	Pass
Input Validation	Empty name field	Error message	Pass
Data Storage	Submit answers	Stored in DB	Pass
Tutorial Navigation	Select topic	Display correct content	Pass
Invalid Input	Random string in number field	Error handled	Pass

9.Advantages & Limitations

- ➢ **Advantages:**

- User-friendly and interactive interface.
- Real-time feedback and scoring for quizzes.

- Modular design allows future expansion.
- Lightweight, works on standard hardware.

➢ **Limitations:**

- Limited initial content; requires updates.
- Internet required for setup and library installation.
- No advanced analytics dashboard yet.

10.KNOWN ISSUES:

- Limited Content in Quizzes and Tutorials: Currently, the system has a small set of quizzes and tutorials, which may restrict the variety of learning material available to users. Expansion is required to cover more topics.
- Advanced Analytics Dashboard Not Implemented: While the system tracks quiz scores and progress, a detailed analytics dashboard for performance trends, topic-wise weaknesses, and insights is not yet available.
- AI-Based Personalized Recommendations Not Available: The platform does not yet provide customized learning paths or recommendations based on user performance. All users follow the same standard tutorials and quizzes.
- Internet Connection Required for Setup: Installing the system and its libraries requires an active internet connection. Users without reliable internet may face difficulties in setup.
- Performance on Non-GPU Systems: Some AI or NLP features may run slower on machines without a dedicated GPU, affecting the responsiveness of advanced modules.
- No Mobile App Version: Currently, EduTutorAI is designed for desktop use. Mobile accessibility is not yet available, limiting on-the-go learning.
-

11.CONCLUSION:

EduTutorAI is an efficient, user-friendly, and modular e-learning platform that effectively combines **tutorials and quizzes** into a single interactive system. By providing **real-time feedback** on quiz performance, it allows learners to immediately identify their strengths and areas that need improvement, supporting a **self-paced learning environment**.

The platform's modular design ensures that it is **easy to maintain, update, and scale** for future needs. EduTutorAI not only helps learners grasp concepts more effectively but also offers educators a structured way to assess student understanding. Its interactive interface and seamless workflow make learning engaging and accessible.

Overall, EduTutorAI represents a **comprehensive digital learning solution** that balances usability, functionality, and scalability, making it a valuable tool for modern education

12.FUTURE ENHANCEMENTS:

1.AI-Based Personalized Learning: Implement intelligent algorithms that analyze user performance and provide **customized learning paths**, focusing on weak areas and recommending targeted tutorials or quizzes.

2.Multi-Language Support: Extend the platform to support **multiple languages**, making it accessible to a wider audience and promoting inclusive learning.

3.Cloud Database Integration: Shift to a **centralized cloud-based database** to enable multiple users to access the system from different devices, ensuring data consistency and scalability.

4.Advanced Analytics Dashboard: Provide educators and learners with **detailed insights** on performance trends, question difficulty, and overall learning progress.

5.Mobile Application Version: Develop a **mobile-friendly app** to allow learners to access tutorials and quizzes on-the-go, enhancing convenience and engagement.

6.Gamification Features: Add **badges, points, levels, and leaderboards** to motivate learners, increase participation, and make learning more enjoyable.

7.Interactive Multimedia Tutorials: Include **videos, animations, and simulations** within tutorials to improve concept understanding and retention.

These enhancements will not only **improve the learning experience** but also make EduTutorAI a **more comprehensive and future-ready educational platform**.