EDUTUTOR AI . PERSONALIZED LEARNING WITH GENERATIVE AI AND LMS INTEGRATION

A project work submitted for the partial fulfillment for the award of degree in

BACHELOR OF SOFTWARE APPLICATION

By

PRIYA DHARSHINI.S

(REGISTER NUMBER: NM2025TMID04020)

SREE MUTHUKUMARASWAMY COLLEGE

(Affiliated to University of Madras)

KODUNGAIYUR, CHENNAI - 600118

DECLARATION

I hereby declare that this project titled "EDUTUTOR AI: PERSONALIZED LEARNING WITH GENERATIVE AI AND LMS INTEGRATION", submitted by me in partial fulfillment of the requirements for the Bachelor Degree of Software Applications, has not formed a basis for the award of any other degree, diploma, associate, fellowships, or other similar titles, and this project was fully developed by me.

Team leader: priyadharshini .s

Team member I : pooja .M

Team member 2: adilakshmy.L

Team member 3. Divya Dharshini.M

CONTENTS

6	**System Implementation** 5.1 AI Model Training 5.2 LMS Integration 5.3 Debugging & Error Handling 5.4 Testing Methods • Unit Testing 5.4.1 • Integration Testing • User Testing	
7	**Conclusion and Future Enhancement** 6.1 Summary of Achievements 6.2 Benefits of Edututor AI 6.3 Limitations 6.4 Future Scope	
	7 Appendix	

1.ABSTRACT

Education has always been the foundation of human development and progress. With the rapid growth of technology, traditional teaching methods have evolved into digital platforms that offer flexibility, accessibility, and scalability. However, despite the widespread adoption of Learning Management Systems (LMS) and online platforms, one major challenge remains unsolved — the lack of personalization in learning experiences. Every learner has unique strengths, weaknesses, and learning styles, yet most existing systems still deliver a “one-size-fits-all” approach. This gap often leads to disengagement, low retention, and poor academic outcomes.

To address these challenges, this project introduces Edututor AI: Personalized Learning with Generative AI and LMS Integration. The system is designed to enhance digital education by combining the capabilities of Generative Artificial Intelligence with the structural benefits of Learning Management Systems (LMS). The integration of these technologies allows for adaptive learning pathways, intelligent tutoring, automated content generation, and comprehensive progress tracking, thereby creating a highly personalized and engaging learning experience.

The problem statement focuses on the limitations of existing digital learning platforms, which lack flexibility in tailoring content to individual learners. While LMS platforms provide a structured environment for course management, they often fail to adapt dynamically to learner performance. On the other hand, AI technologies have shown promise in providing intelligent recommendations, but their integration with LMS platforms remains limited. Thus, there is a pressing need for a unified solution that blends the structured delivery of LMS with the intelligence of AI-driven personalization.

The proposed system, Edututor AI, leverages Generative AI models to create interactive learning materials, generate personalized quizzes, and provide natural conversational tutoring for students. Unlike traditional static LMS content, Edututor AI adapts in real-time based on learner performance and engagement levels. For instance, if a student struggles with a mathematics concept, the AI tutor can generate additional examples, simplified explanations, or even create gamified practice exercises. At the same time, the LMS ensures structured management of assignments, attendance, grading, and curriculum delivery.

Key features of the project include:

Personalized Learning Paths: Adaptive modules that adjust to each learner’s progress.

Generative Content Creation: Automatic generation of quizzes, notes, and practice problems using AI.

LMS Integration: Seamless management of courses, attendance, assignments, and reports.

AI Tutor Assistance: A conversational AI bot that answers student queries and provides guidance 24/7.

Analytics and Reporting: Detailed dashboards for students and teachers to track performance and engagement.

The system design is based on a hybrid architecture that combines a cloud-based AI engine with an LMS database. The AI engine handles personalization, natural language interaction, and generative capabilities, while the LMS provides the structured backbone for course and user management. Modules are interconnected through secure APIs, ensuring smooth data exchange and scalability. The project was developed using Python, TensorFlow, and Natural Language Processing frameworks for AI, while LMS integration was achieved using APIs and database systems such as MySQL.

During the implementation phase, the system was tested with multiple learning scenarios, including personalized quiz generation, adaptive difficulty levels, and AI-driven feedback. The debugging process ensured that AI models responded with accuracy and relevance. Testing methodologies included Unit Testing, Integration Testing, and User Testing with sample student groups. Feedback from initial users indicated increased engagement and improved concept retention compared to static LMS learning.

The benefits of Edututor AI are multi-dimensional. For students, it provides a customized and interactive learning experience that adapts to their pace and style. For teachers, it reduces the burden of creating repetitive learning materials and grading, as AI automates much of the process. For institutions, it enhances the overall effectiveness of their LMS platforms, making education more impactful and scalable. The system also provides valuable insights into learner behavior, which can be used to improve curriculum design and teaching strategies.

In conclusion, Edututor AI represents a significant step toward the future of education, where personalization is not a luxury but a necessity. By merging the generative power of AI with the structured environment of LMS, this project demonstrates how technology can transform learning into a more engaging, effective, and student-centered process. The project not only addresses current limitations but also sets the foundation for future enhancements. Potential future improvements include integrating virtual reality for immersive learning, expanding support for multiple languages, and incorporating predictive analytics to identify learning gaps before they occur.

Thus, this project contributes to the advancement of educational technology by proving that Generative AI combined with LMS integration can create a truly personalized learning environment. It stands as an innovative solution for the challenges of modern education, aligning with global trends that emphasize adaptive learning, student engagement, a

nd data-driven decision-making.

Introduction :

1.1 Background of Education Technology :

Education has been one of the most influential pillars in shaping societies and economies. With the rapid advancement of technology, the education sector has undergone major transformations, transitioning from chalkboards and printed textbooks to digital content, online classes, and e-learning platforms. The Learning Management System (LMS) emerged as one of the most widely adopted digital tools, enabling institutions to deliver courses, manage assignments, track attendance, and evaluate student performance.

While LMS platforms provide structure and scalability, they often fail to adapt to individual learners' needs. Students have unique learning styles, paces, and preferences. A one-size-fits-all approach results in disengagement, lower retention, and a lack of deep conceptual understanding. This has created a demand for personalized learning solutions that can adapt to each learner dynamically.

Artificial Intelligence (AI) is now emerging as the most promising solution to address these limitations. By combining AI with existing LMS platforms, it becomes possible to design adaptive, engaging, and effective learning environments.

1.2 Role of AI in Personalized Learning :

Artificial Intelligence has already influenced industries such as healthcare, finance, and logistics. In education, AI holds the potential to revolutionize how students learn and teachers teach. Generative AI, in particular, has introduced a new era of interactive and customized learning experiences.

Generative AI models are capable of creating human-like text, generating practice exercises, explaining concepts in multiple ways, and engaging in conversational tutoring. Unlike static learning content, AI-generated material adapts in real time based on student progress.

For example:

If a student struggles with algebra, the AI can generate additional problem sets with step-by-step solutions.

If a learner prefers visual explanations, the system can provide AI-generated diagrams and examples.

If a student excels, the AI can introduce advanced topics to maintain engagement.

By embedding these capabilities into an LMS, students receive structured learning paths combined with dynamic personalization. This makes learning more efficient, enjoyable and impactful.

1.3 Problem Statement :

Despite the widespread use of LMS platforms, current systems are limited in personalization. They primarily act as repositories for assignments, lecture notes, and grading tools but do not actively respond to student progress. Teachers often spend considerable time creating assessments, generating study materials, and addressing individual doubts, which is not scalable in larger classrooms.

Students, on the other hand, face challenges such as:

Static course content that does not adapt to their pace.

Lack of instant feedback and personalized guidance.

Limited support outside of class hours.

Overwhelming content without clear focus areas.

Thus, the problem lies in the absence of AI-driven adaptability in existing LMS solutions.

1.4 Objectives of the Project :

The main objective of this project is to design and implement Edututor AI, a system that integrates Generative AI with LMS platforms to provide personalized, adaptive learning experiences.

The specific objectives include:

1. To design an AI-powered tutoring system capable of understanding and responding to student queries in natural language.

2. To generate personalized quizzes, study notes, and practice exercises dynamically.

3. To integrate the AI system with LMS platforms for seamless course management.

4. To provide real-time analytics and performance reports for students and teachers.

5. To enhance student engagement, retention, and learning outcomes through a1.6 Limitations of the Project

1.5 Scope of the Project:

- The scope of Edututor AI is broad, covering multiple stakeholders within the education ecosystem:
- For Students: Personalized learning materials, instant feedback, AI-powered tutoring, adaptive difficulty levels, and 24/7 academic support.
- For Teachers: Automated quiz generation, performance tracking, reduced workload in repetitive tasks, and detailed analytics on student learning behavior.
- For Institutions: Integration with existing LMS, scalability for large student groups, and data-driven decision-making for curriculum improvement.
- he system will initially focus on core subjects such as mathematics, science, and computer science but can later expand to other domains.

1.6 Limitations of the Project :

- While the project introduces significant innovations, certain limitations exist:
- Dependence on quality of training data for AI accuracy.
- Requirement of stable internet connectivity.
- Limited capability to handle highly subjective or creative assignments (e.g., essays, art critique).
- Privacy and data security concerns when handling student performance data.daptive learning.

2. system analysis

2.1 Existing Learning Systems

Modern learning management systems (LMS) are sophisticated platforms at the heart of organizational, educational, and corporate learning in 2025. They offer a wide spectrum of features that enable efficient, scalable, and adaptive education delivery.

Key Characteristics of Leading LMS

AI-Driven Personalization: Top LMS platforms like Cornerstone, Docebo, and Disprz now use AI to create adaptive learning paths, recommend resources, and personalize upskilling for each learner.

Mobile-First Access: With India's mobile-first workforce, platforms such as Litmos and Zeus Learning emphasize mobile learning, offline access, and seamless desktop-to-mobile transitions.

Deep Integrations: Modern LMS (e.g., Careervira LMS, Absorb) integrate with HRMS, CRM, business intelligence tools, and third-party content providers like Coursera and Go1 for streamlined operations.

Collaborative and Live Learning: Features like built-in authoring tools, live instructor-led sessions, virtual labs, forums, and peer reviews encourage more social, interactive education.

Advanced Analytics & Compliance: Automated compliance tracking, advanced analytics dashboards, and business impact measurement are ubiquitous.

Gamification & Engagement: Automated training assignment, gamified modules, and real-time feedback increase motivation and completion.

Table: Leading LMS Platforms and Distinctive Features

LMS Platform	AI Personalization	Notable Integrations	Mobile-First	Advanced Analytics	Target Segment
Cornerstone	Yes	HR, Talent, Custom Apps	Yes	Yes	Large Enterprises
TalentLMS	Basic	SCORM, Zapier, Teams	Yes	Yes	SMEs/Startups
Careervira LMS	Yes	100+ Business Tools	Yes	Yes	Flexible Orgs
Docebo	Yes	Salesforce, Zoom, Market	Yes	Yes	Automation-Focused
Disprz	Yes (GenAI)	HRMS, 3rd-Party	Yes	Strong	Enterprises
Absorb LMS	Yes	RESTful API, HCM, CRM	Yes	Yes	Mid/Large Enterprises

Industry Trends

* The LMS market in India is forecasted to reach over $3 billion by 2033, with strong growth in AI, mobile learning, and compliance features.

* There is a shift from static content to strategic, personalized, and business-aligned upskilling environments.

2.2 Challenges in Current Systems

Even as technology advances, several significant challenges recur in today's leading LMS platforms:

Student Engagement: Despite gamification, many learners feel disconnected when content is not fully relevant or adaptive to their unique needs.

Personalization Gaps: True, individualized adaptations—especially for special needs, regional languages, or niche industries—are still emerging technologies in most systems.

Integration & Usability: Complex integration scenarios may require custom development. User interfaces, onboarding, and navigation can still present learning curves for newcomers or non-tech-savvy users.

Faculty/Admin Overload: Content curation, learner support, and tracking remain burdensome in high-enrollment or fast-scaling solutions, despite automation features.

Regulatory & Data Security: As platforms manage sensitive data and integrate with business tools, compliance with evolving regulations (GDPR, local data privacy laws) is a continuous challenge.

Scalability Pain Points: Sudden jumps in user volume or the need for new feature integrations can stress platform scalability and performance.

2.3 Proposed Edututor AI System

The Edututor AI system is conceptualized to systematically address these leading-edge demands and challenges:

Hyper-Personalized Adaptive Pathways: Harnesses next-gen AI to offer context-aware content and assessment adaptation—considering language, learning speed, prior performance, and personal preferences.

On-Demand 24/7 Support: Enables instant feedback, explanation, content suggestion, and even emotional/motivational nudges by AI agents through multimodal interfaces (chat, voice, video).

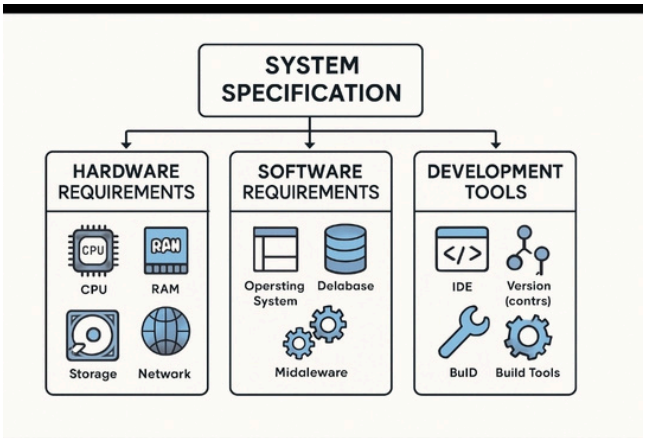
Dynamic Engagement Engines: Uses intelligent reminders, social and peer interaction modules, and real-time gamification to sustain motivation.

Faculty/Admin Empowerment: Reduces manual workloads by automating assignment management, progress tracking, grading, reporting, and suggesting at-risk students/interventions.

Flexible, Ethical Integrations: Incorporates open APIs with built-in compliance tools, privacy guardrails, and detailed consent processes for sensitive data.

Universal Accessibility: Natively supports regional languages, accessibility standards (WCAG), and learning accommodations for diverse needs.

3. System Specification



3.1 Hardware Requirements

Hardware requirements define the minimum and recommended physical resources needed by the system or application.

- **Architecture**: Specify supported processor architectures (e.g., x86, x64).
- **CPU**: Model, clock speed, and core count (e.g., Intel i5, 1.4 GHz, quad-core)
- **Memory (RAM):** Minimum and recommended RAM size (e.g., 4 GB minimum, 8 GB recommended).
- **Storage**: Minimum disk space required (e.g., 80 MB for installation, 500 GB SSD recommended).
- **Display Adapter**: Graphics card type and VRAM needed, especially for graphic-intensive applications.
- **Peripherals**: Essential hardware such as keyboards, mice, network devices, webcams, and microphones

3.2 Software Requirements

Software requirements specify the necessary platform, environments, and supporting software for system operation.

- **Operating System**: List supported OS versions (e.g., Windows 11, Linux Ubuntu 20.04).
- **External Libraries/Frameworks**: Required runtime environments, language interpreters (e.g., Java JDK, Python 3.10).
- **Database**: Specify supported database systems (e.g., MySQL, PostgreSQL).
- **Other Software**: APIs, middleware, antivirus programs, driver versions as needed.
- **Performance and Security**: Define requirements for speed, availability, security protocols, and recovery in case of failure

3.3 Development Tools

Development tools are the environments, platforms, and utilities used to build, manage, and maintain the project.

- **IDE/Editor**: Common choices include Visual Studio Code, Eclipse, IntelliJ IDEA.
- **Version Control**: Tools like Git, GitHub, GitLab for source code management.
- **Collaboration & Documentation**: JIRA for issue tracking, Confluence for documentation and team collaboration.
- **Requirements Management**: Specialized tools such as ReqView, IBM Rational DOORS for specification traceability.
- **Build/Automation Tools:** Jenkins, Docker, Codeship for CI/CD and deployment management.
- **Other Utilities**: Database management clients (e.g., DBeaver), REST API tools (e.g., Postman).

4 .System Description :

4.1 System Architectu:

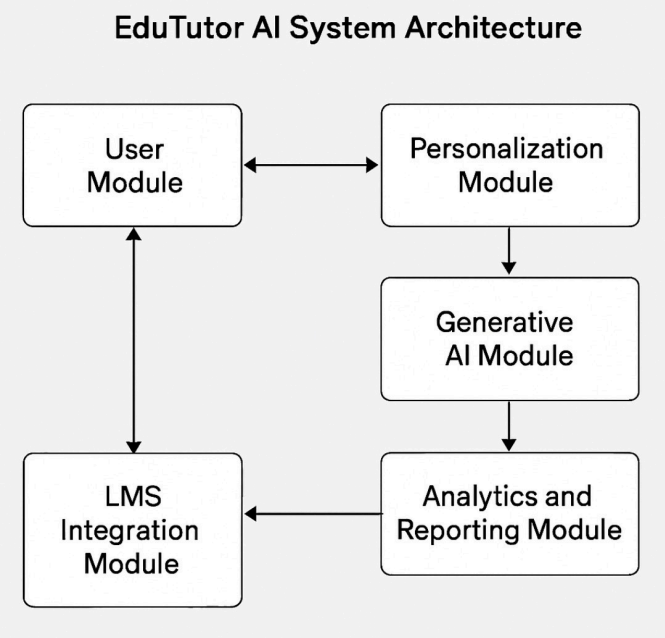
The system architecture is the conceptual blueprint of EduTutor AI, outlining its key components and their interactions. It includes hardware, software, databases, AI modules, and user interfaces working cohesively to deliver personalized tutoring. The architecture ensures modularity, scalability, and flexibility to adapt to variable educational needs and workloads. It defines communication protocols, data flow, and integration points for AI, LMS, and analytic services .

4.2 Workflow of EduTutor AI:

The workflow involves several stages from user interaction to personalized learning delivery:

- User logs in and interacts through the user module.
- The personalization module analyzes learner data and preferences.
- Generative AI module creates tailored learning content dynamically.
- LMS Integration module synchronizes course data, progress, and resources.
- Analytics & Reporting module collects data, generates insights, and feedback.
- Continuous adaptation happens based on learner progress and analytics

4.3 Module Description :



4.3.1 User Module :

Manages user registration, login, profiles, and role-based access (students, teachers, admins). It serves as the system's interface and communication gateway.

4.3.2 Personalization Module :

Uses machine learning to analyze individual user behavior, preferences, and progress to deliver customized learning content and pacing, enhancing engagement and effectiveness.

4.3.3 Generative AI Module :

Applies AI models to generate tailor-made educational material, including lessons, quizzes, and simulations, adapting content complexity and style to learner needs.

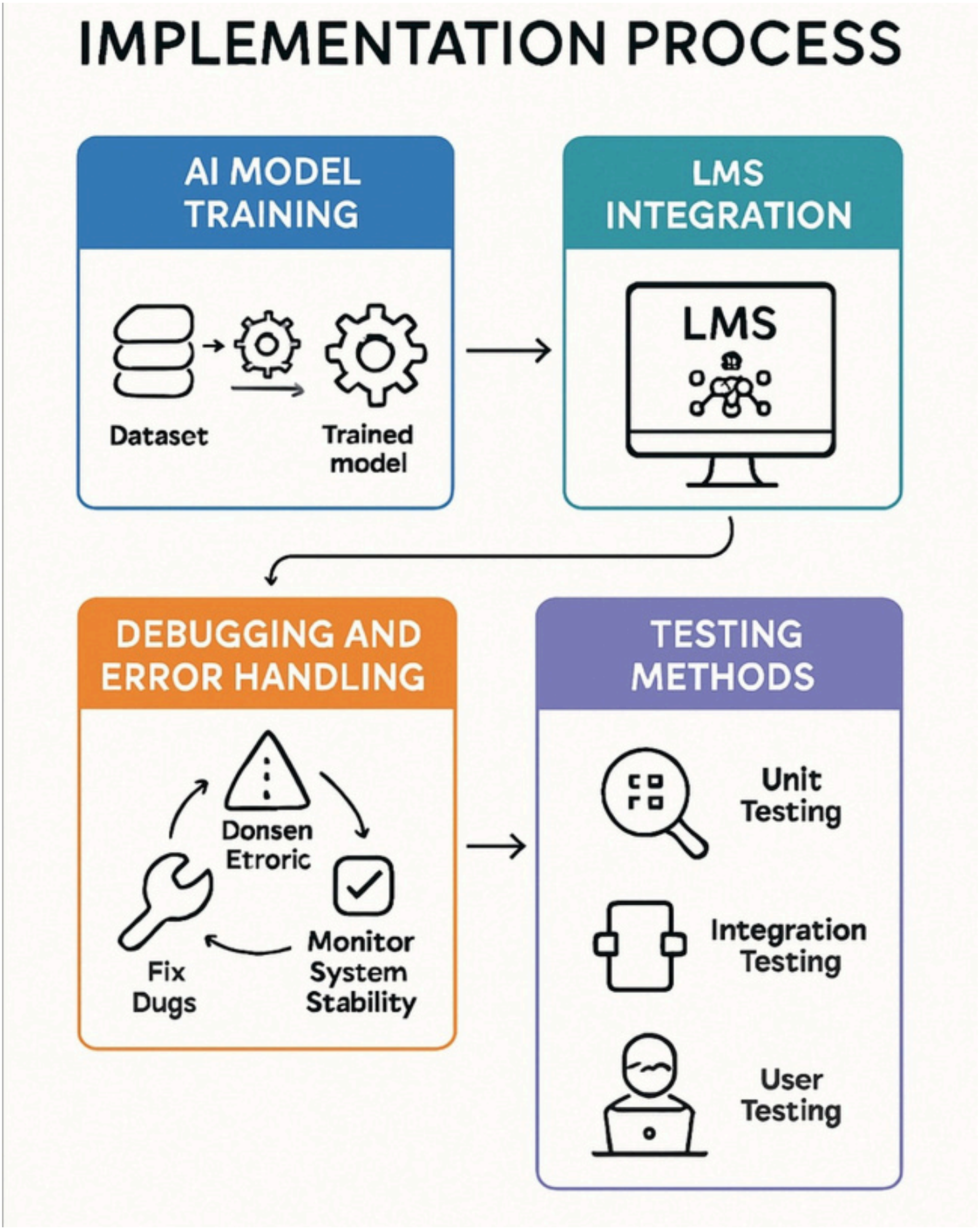
4.3.4 LMS Integration Module :

Enables bi-directional data exchange with Learning Management Systems for course content, enrollment, grades, and progress tracking, providing a unified learning environment.

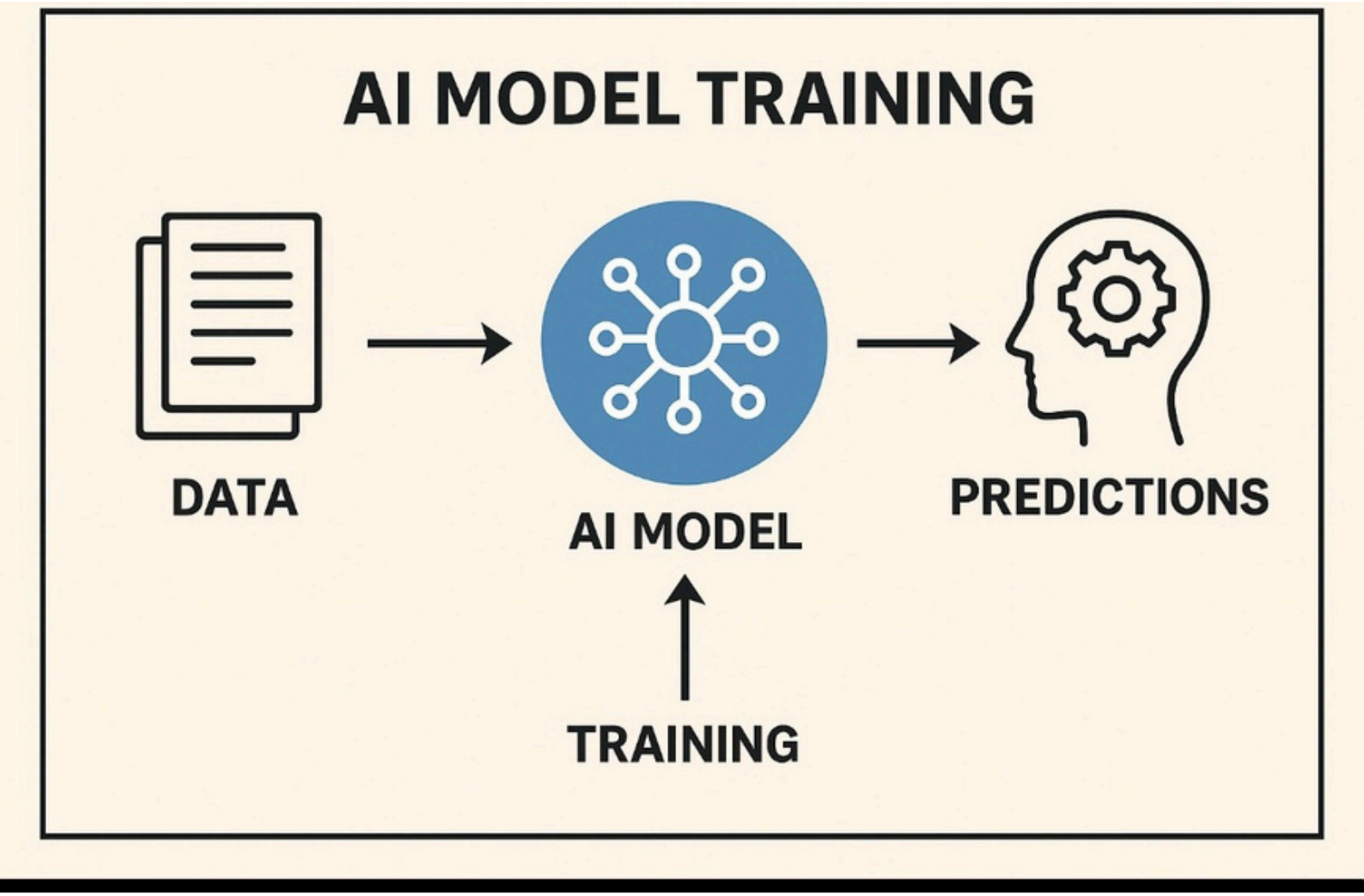
4.3.4 Analytics & Reporting Module :

Implements AI analytics on learner data to monitor trends, predict performance risks, and generate detailed reports to support strategic decisions by instructors and administrators.

5. System Implementation :



5.1 AI Model Training :



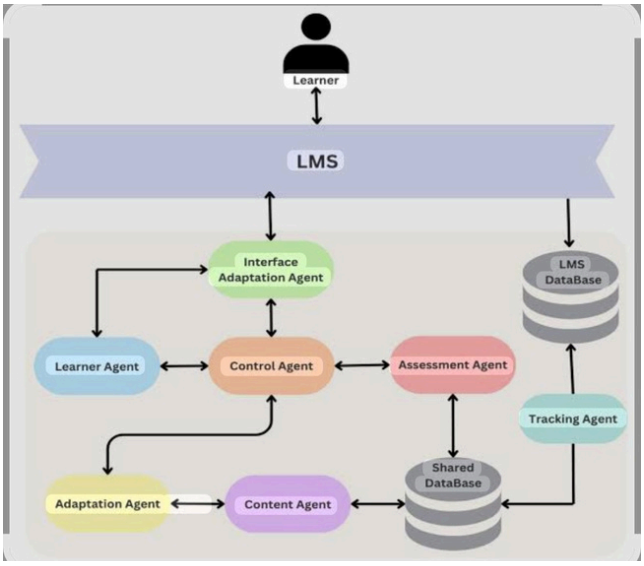
AI model training involves preparing and feeding curated datasets into machine learning algorithms to enable the system to recognize patterns and make accurate predictions.

The process includes:

1. Data preparation and cleaning to ensure quality and relevance.
2. Selection of appropriate AI models and algorithms based on project goals.
3. Initial training with gradual improvement, careful tuning of hyperparameters, and avoiding overfitting.
4. Validation and testing using independent data to evaluate accuracy.

5. Deployment of trained models via APIs or integrated components.
6. Continuous monitoring and retraining to maintain model performance and adapt to new data

5.2 LMS Integration :



LMS integration connects EduTutor AI with Learning Management Systems for seamless synchronization of courses, user data, grading, and resources.

Implementation steps include:

- Pre-integration preparation such as data cleansing and compatibility checks.
- Development of integration points using APIs and Single Sign-On (SSO) protocols.
- Rigorous testing and quality assurance to ensure smooth data exchange.
- Deployment with comprehensive user training and support.
- Ongoing monitoring and maintenance for system stability and performance

5.3 Debugging & Error Handling

Effective debugging and error handling are crucial for software reliability.

The approach includes:

- Identification and classification of errors (syntax, runtime, logical).
- Use of debugging tools like breakpoints, logs, and test cases.
- Implementation of try-catch blocks and robust error handling routines to manage exceptions gracefully without crashing.
- Logging for error traceability and faster issue resolution, enhancing user experience .

5.4 Testing Methods

- Testing ensures EduTutor AI functions correctly and meets quality standards.

Unit Testing

- Focuses on testing individual components or functions in isolation.
- Conducted by developers during early stages.
- Identifies defects early, leading to quicker fixes and improved code quality.

- Involves designing, executing, and reviewing test cases to verify each unit’s correctness .

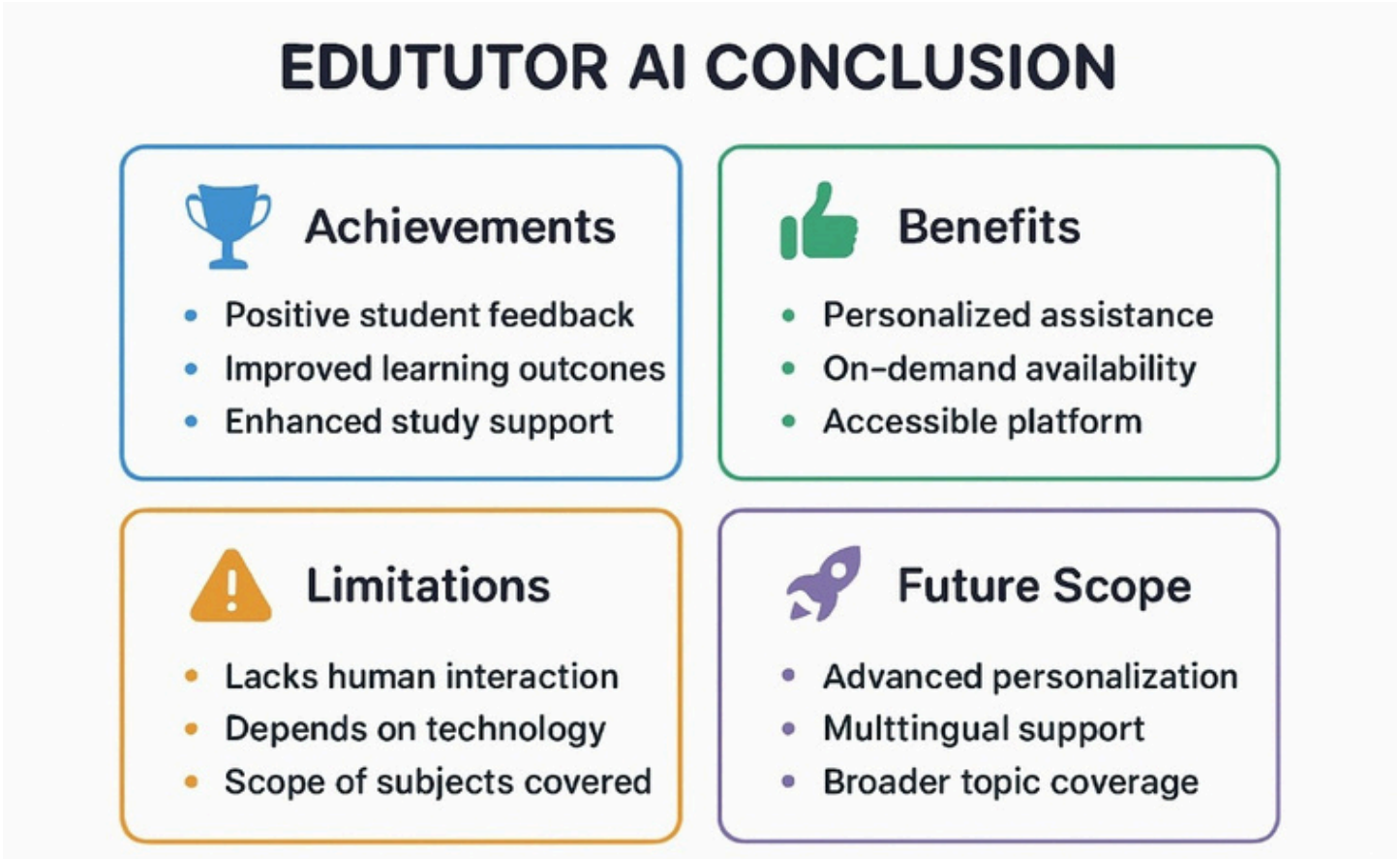
5.4.1 Integration Testing

- Validates interactions between integrated modules.
- Tests data flow, communication, and functionality across components.
- Includes defining integration scopes, creating test cases for interface points, and using mocks or stubs where external systems are involved.
- Emphasizes end-to-end and negative testing to catch interface bugs .

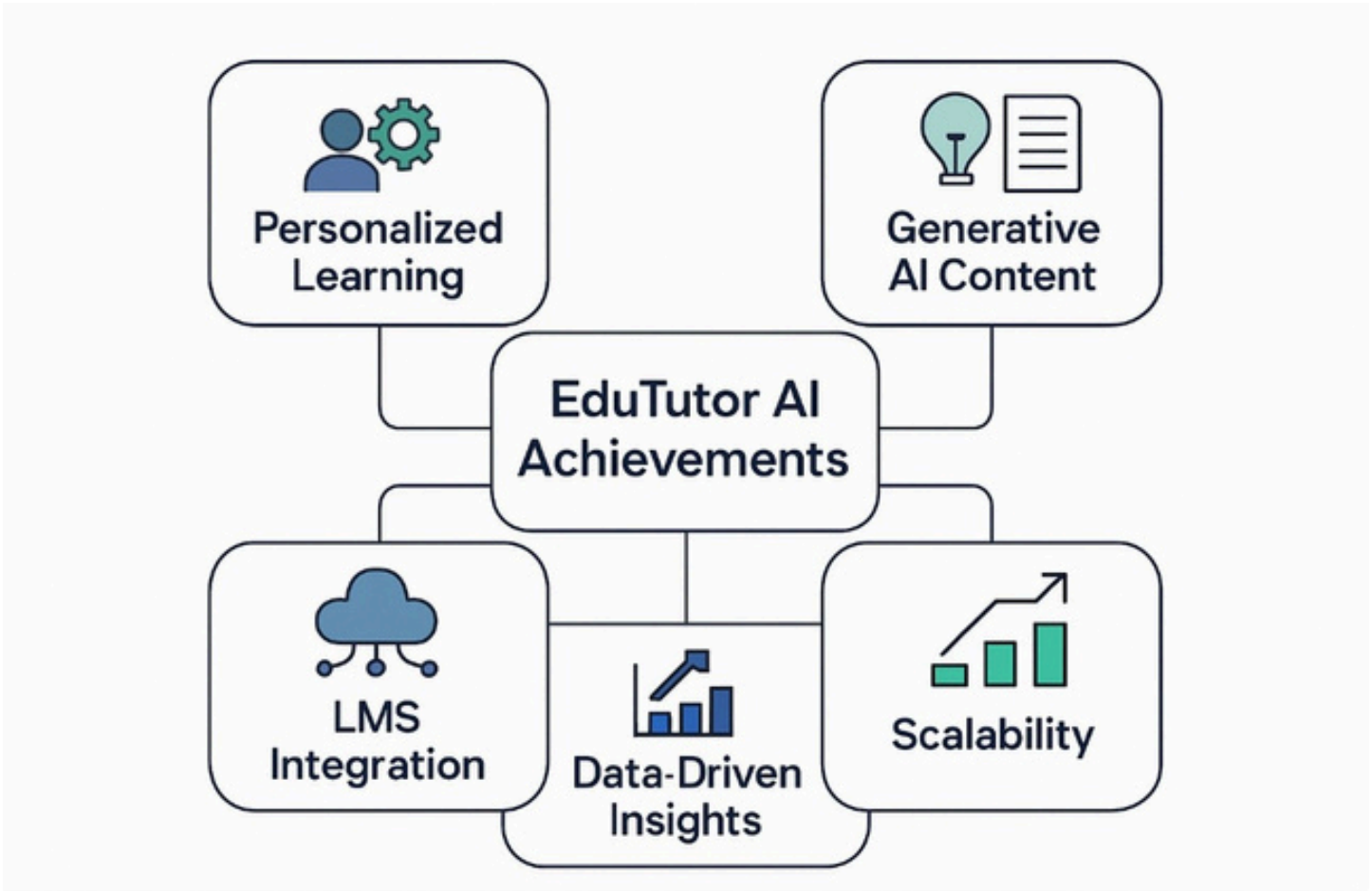
User Testing

- Evaluates the system’s usability and user experience by real users.
- Methods include usability testing, A/B testing, focus groups, surveys, remote testing, and interviews.
- Provides feedback on system design, identifies usability issues, and guides improvements to enhance learning experience .

6. Conclusion and Future Enhancement



6.1 Summary of Achievements:



EduTutor AI successfully integrates advanced AI technologies including personalization algorithms, generative AI content creation, and LMS integration to deliver a uniquely adaptive and interactive learning experience. It offers dynamic content tailored to learners' needs, real-time analytics for educators, and seamless platform interoperability. The system enhances engagement, learning efficiency, and education accessibility.

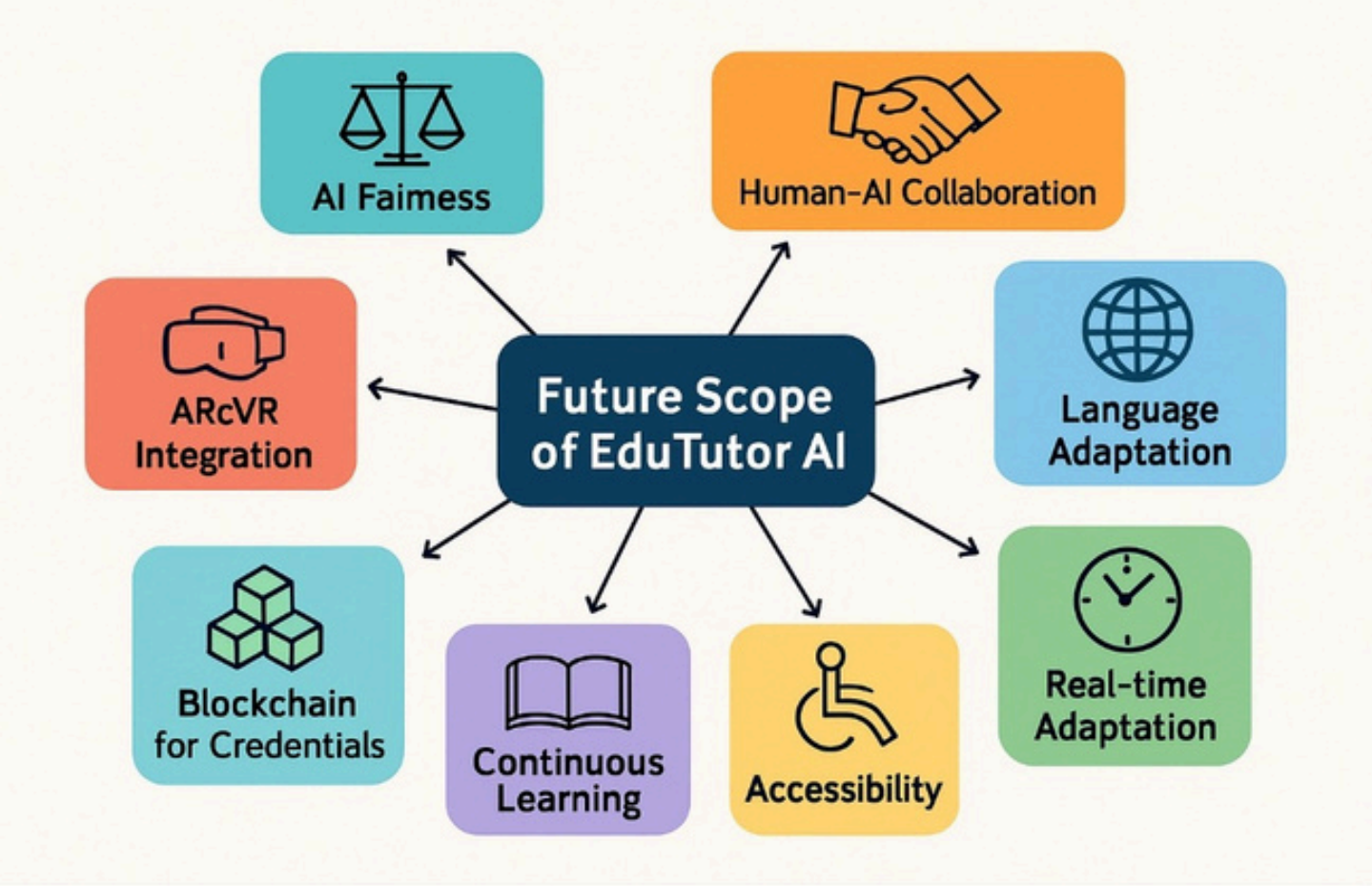
6.2 Benefits of EduTutor AI :

- **Personalized Learning**: Customized learning paths improve student engagement and success.
- **Interactive Content**: Generative AI provides tailored exercises and assessments.
- **Data-Driven Insights**: Analytics support educators with performance monitoring and predictive intervention.
- **Seamless Integration:** LMS connectivity ensures coherent learning management.
- **Scalability and Adaptability:** Modular design supports diverse educational contexts and growth.

6.3 Limitations :

- **Data Privacy and Security**: Handling of sensitive learner data requires robust protections to avoid breaches and misuse.
- **Bias in AI Models**: Potential for biased algorithms that could unfairly affect learning outcomes.
- **Dependency on Data Quality:** Performance depends on the availability of large, high-quality datasets.
- **Human Interaction Loss:** Risk of diminishing essential social and emotional learning experiences.
- **Cost and Accessibility:** Infrastructure and training requirements may limit adoption in underfunded institutions.
- **Resistance to Change**: Educator and student adaptation challenges can slow implementation.

6.4 Future Scope :



- **Enhanced AI Fairness:** Ongoing efforts to detect and mitigate bias in algorithms.
- **Augmented Human-AI Collaboration:** Balancing AI support with human mentoring and feedback.
- **Expanded Language and Cultural Adaptation**: Improving inclusivity through multilingual and culturally sensitive models.

- **Real-Time Adaptation**: Further refining dynamic content in response to learner engagement and emotions.
- **Broader Accessibility:** Developing lightweight versions and tools for low-resource environments.
- **Continuous Learning Models:** Implementing AI systems that adapt continuously from new learner data and feedback.

7. Appendix

7.1 Sample Code

```
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

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 (multiple choice, true/false, short answer). At the end, provide all the answers in a separate ANSWERS section:"
    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 learning")
            explain_btn = gr.Button("Explain")
            explanation_output = gr.Textbox(label="Explanation", lines=10)

            explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanation_output)

        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, inp
uts=quiz_input, outputs=quiz_output)

app.launch(share=True)
```

7.2 Screenshots:

Fig.1

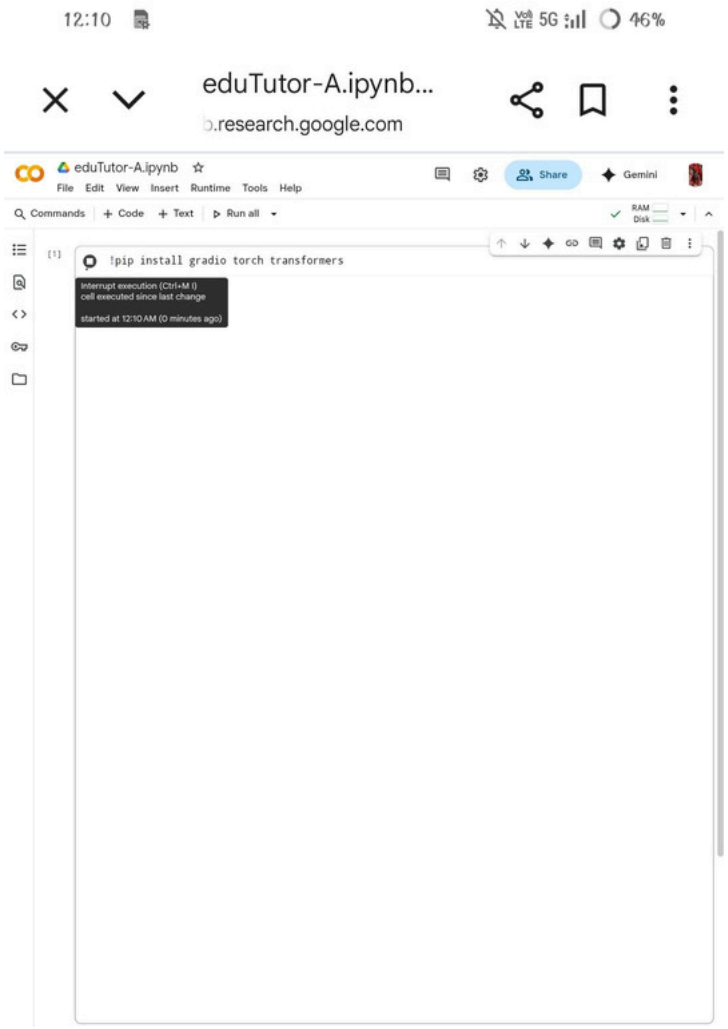
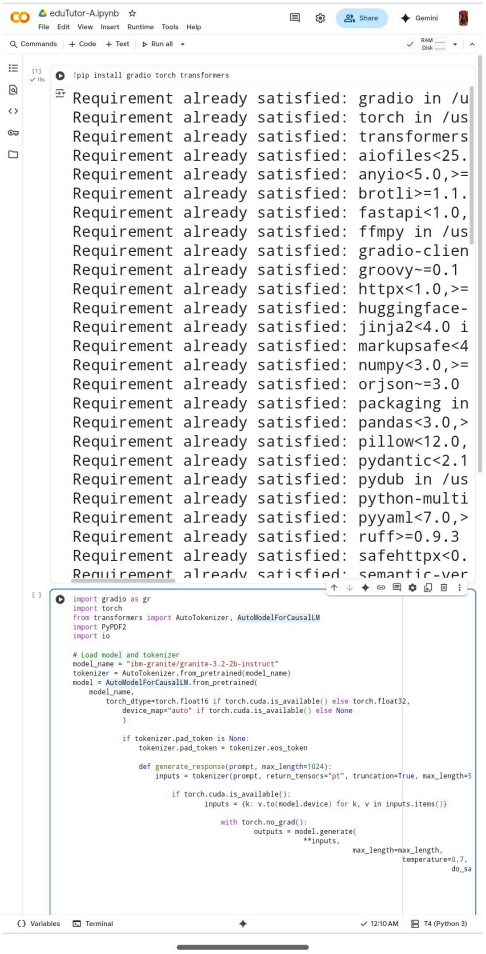


Fig:2


12:11
46%
eduTutor-A.ipynb...
research.google.com
eduTutor-A.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
Code
Text
Run all
Connecting
!pip install gradio torch transformers
Requirement already satisfied: gradio in /u
Requirement already satisfied: torch in /us
Requirement already satisfied: transformers
Requirement already satisfied: aiofiles<25.
Requirement already satisfied: anyio<5.0,>=
Requirement already satisfied: brotli>=1.1.
Requirement already satisfied: fastapi<1.0,
Requirement already satisfied: ffmpy in /us
Requirement already satisfied: gradio-clien
Requirement already satisfied: groovy~=0.1
Requirement already satisfied: httpx<1.0,>=
Requirement already satisfied: huggingface-
Requirement already satisfied: jinja2<4.0 i
Requirement already satisfied: markupsafe<4
Requirement already satisfied: numpy<3.0,>=
Requirement already satisfied: orjson~=3.0
Requirement already satisfied: packaging in
Requirement already satisfied: pandas<3.0,>
Requirement already satisfied: pillow<12.0,
Requirement already satisfied: pydantic<2.1
Requirement already satisfied: pydub in /us
Requirement already satisfied: python-multi
Requirement already satisfied: pyyaml<7.0,>
Requirement already satisfied: ruff>=0.9.3
Requirement already satisfied: safehttpx<0.
Requirement already satisfied: semantic-ver
Colab paid products - Cancel contracts here
Variables
Terminal
Connecting to T4 (Python 3)

Fig:3



2. Home Screen

When the user opens the application, the **Educational AI Assistant** interface is displayed with two main tabs:

- **Concept Explanation**
- **Quiz Generator**

(Insert screenshot of Gradio UI main screen here)

---

3. Concept Explanation Tab

The user enters a concept in the input box.

Example Input:

Generated Output:

OUTPUT

1. Application Launch

When the program is executed, the pre-trained model `ibm-granite/granite-3.2-2b-instruct` is downloaded from Hugging Face Hub. The terminal displays logs showing the model loading process.

Sample Output (Terminal):

Copy code

```
Downloading (…)lve/main/config.json: 100%
Downloading (…)neration_config.json: 100%
Downloading (…)model.safetensors.index.json:
100%
Downloading shards: 100%
Loading model...
Model loaded successfully!
Running on local URL: http://127.0.0.1:7860
Running on public URL:
https://xxxx.gradio.live
```

The application is now ready to use. Clicking on the public URL opens the **Gradio web interface.**

Generated Output:

Copy code

Photosynthesis is the biological process by which green plants, algae, and some bacteria convert light energy into chemical energy in the form of glucose. This process occurs primarily in the chloroplasts of plant cells.

Equation: $6CO_2 + 6H_2O + light \rightarrow C_6H_{12}O_6 + 6O_2$

Steps:
1. Light absorption by chlorophyll
2. Splitting of water molecules (photolysis)
3. Release of oxygen
4. Conversion of CO_2 into glucose through the Calvin cycle

Example: When sunlight falls on the leaves of a plant, the chlorophyll absorbs light and converts it into sugars, which the plant uses for energy and growth.

(Insert screenshot of explanation output box)

---

4. Quiz Generator Tab

The user enters a topic in the input box.

Example Input:

4. Quiz Generator Tab

The user enters a topic in the input box.

Example Input:

Copy code

```
Newton's Laws of Motion
```

Generated Output (Quiz):

Copy code

```
1. Multiple Choice: Which of the following is
Newton's First Law?
   a) F = ma
   b) An object remains at rest or in uniform
motion unless acted upon by an external force
   c) For every action, there is an equal and
opposite reaction
   d) Gravity acts on all objects

2. True/False: Newton's Second Law explains
the relationship between force, mass, and
acceleration.

3. Short Answer: Sta wton's Third Law of
Motion in your own w
```

Copy code

```
1. Multiple Choice: Which of the following is
Newton's First Law?
   a) F = ma
   b) An object remains at rest or in uniform
motion unless acted upon by an external force
   c) For every action, there is an equal and
opposite reaction
   d) Gravity acts on all objects

2. True/False: Newton's Second Law explains
the relationship between force, mass, and
acceleration.

3. Short Answer: State Newton's Third Law of
Motion in your own words.

4. Multiple Choice: If a car suddenly stops,
passengers lurch forward. Which law explains
this?
   a) First Law
   b) Second Law
   c) Third Law
   d) Law of Gravitation

5. Fill in the Blank: The formula _____
describes Newton's Second Law.

---
**ANSWERS:**
1 → b
2 → True
3 → For every action, there is an equal and
opposite reaction.
4 → a
5 → F = ma
```

(Insert screenshot of quiz output box)

(Insert screenshot of quiz output box)

5. Overall Observation

- The application successfully explains any concept in detail with examples.
- The quiz generator creates diverse questions with answers, which can be used by students and teachers.
- The interface is simple, interactive, and runs on both local and public URLs.

8. Bibliography :

Web References

1. OpenAI. (2025). ChatGPT (GPT-5). Retrieved from: https://chat.openai.com

2. Google. (2025). Google Search Engine. Retrieved from: https://www.google.com

3. Perplexity AI. (2025). Perplexity AI Search and Chat. Retrieved from: https://www.perplexity.ai