**Edu Tutor AI: Personalized Learning**

**AI-Powered Personalized Education Assistant**  
**Project Team Members:**

1. Fathima Bareera G
2. Soundarya R
3. Hatheja Z
4. Athiba Nasreen B

**1. Introduction**

**1.1 Project Title**

**Edu Tutor AI - Personalized Learning Assistant**

**1.2 Background**

Education in the digital era demands personalized learning approaches that adapt to individual student needs, pace, and understanding levels. Traditional learning platforms often lack adaptive feedback and real-time interactivity. **Edu Tutor AI** bridges this gap by offering an AI-driven personalized learning assistant that helps students learn effectively while also supporting educators with content generation and analytics.

**1.3 Objectives1. Introduction**

**1.1 Project Title**

**Edu Tutor AI - Personalized Learning Assistant**

**1.2 Background**

Education in the digital era demands personalized learning approaches that adapt to individual student needs, pace, and understanding levels. Traditional learning platforms often lack adaptive feedback and real-time interactivity. **Edu Tutor AI** bridges this gap by offering an AI-driven personalized learning assistant that helps students learn effectively while also supporting educators

* Deliver **personalized tutoring** using AI.
* Provide **adaptive learning pathways** based on student performance.
* Enable **interactive Q&A** on academic topics.
* Assist educators with **content creation** and **assessment generation**.
* Provide a user-friendly interface for seamless interaction.

**2. Project Overview**

**2.1 Purpose**

To create an AI-powered tutoring system that enhances learning by:

* Understanding each student’s knowledge level.
* Adapting teaching content and strategies.
* Engaging learners through natural conversation.

**2.2 Key Features**

* **Personalized Tutor:** Adapts lessons to each student’s pace and proficiency.
* **Doubt Solver:** Interactive question-answering on subjects like Math, Science, etc.
* **Curriculum-Based Learning:** Aligns with school/college syllabus.
* **Assessment Generator:** Creates quizzes and tests for practice.
* **Performance Tracking:** Visual analytics for students and educators.
* **Conversational Interface:** AI chat using natural language.
* **Gradio Interface:** Clean UI for interaction and learning sessions.

**3. Architecture**

**3.1 Frontend**

* **Framework:** Gradio
* **UI Tabs:** Learning Modules, Ask Doubts, Generate Assessments, Progress Tracker

**3.2 Backend**

* **Libraries Used:** Hugging Face Transformers, PyTorch
* **Model:** IBM Granite or Open-source LLMs like Mistral/LLama for text generation
* **Storage (Optional):** SQLite or Firebase for tracking user progress

**3.3 Functional Modules**

* get\_personalized\_lesson(student\_profile) – Delivers custom lessons
* solve\_academic\_doubt(query) – Answers academic questions
* generate\_assessment(topic, level) – Creates quizzes/tests
* track\_progress(user\_id) – Monitors and visualizes progress

**4. Setup Instructions**

**4.1 Prerequisites**

* Python 3.8+
* Google Colab (T4 GPU recommended)
* Required Libraries:
* transformers, torch, gradio, pandas, matplotlib

**4.2 Installation & Execution**

1. Open Google Colab
2. Switch runtime to **GPU**
3. Install dependencies:
4. !pip install transformers torch gradio pandas matplotlib -q
5. Upload or paste project files
6. Run all cells to initialize
7. Access the Gradio interface using the public link

**5. Folder Structure**

* app/-backend logic
* app/ - Logic for personalized learning
* doubt\_solver.py - Academic question answering
* assessment\_generator.py- Quiz/test creation
* progress\_tracker.py- Track user learning performance
* ui/-frontend components
* interface.py- Gradio front-end tabs
* edu\_tutor\_main.py-Main script to run the application

**6. Running the Application**

**Steps:**

* Launch Colab and install dependencies
* Upload source files (or clone from repo)
* Run edu\_tutor\_main.py
* Interface Tabs:
  + **Learn Module:** Choose subject & level to start personalized lessons
  + **Ask Doubt:** Freeform questions from students
  + **Generate Assessment:** Auto-generate topic-based quizzes
  + **Track Progress:** Visual reports of learning metrics

**7. API Documentation**

**Available Functions**

| get\_personalized\_lesson(): | Returns adaptive learning content |
| --- | --- |
| solve\_academic\_doubt(): | Returns subject-specific answers |
| generate\_assessment(): | Creates quiz/test based on topic and difficulty |
| track\_progress(): | Returns performance metrics for the learner |

**8. Authentication & Security (Future Scope)**

Planned security upgrades:

* JWT/API key-based user authentication
* Role-based access (Student, Teacher, Admin)
* Secure cloud-based user profile storage
* Data encryption for assessments and scores

**9. User Interface**

| * **Tab** | * **Input** | * **Output** |
| --- | --- | --- |
| * Learn Module | * Subject, Grade Level | * Personalized lesson content |
| * Ask Doubt | * Academic Question | * Instant AI-generated response |
| * Generate Assessment | * Topic, Difficulty | * Custom quiz/test |
| * Track Progress | * User ID | * Charts and performance stats |

**10. Testing**

| **Test Type** |  |
| --- | --- |
| Unit Testing: | Core functions tested with multiple inputs |
| Manual Testing: | Interface tested by simulated user sessions |
| API Testing: | All endpoints tested with standard requests |
| Edge Testing: | Handled null input, invalid topic, unknown grade |

**11. Screenshots**

*(Add in final documentation)*

Placeholder for:

* Gradio interface
* Quiz generation output
* Performance tracking dashboard
* Sample lesson content

**12. Known Issues**

* Output may vary based on phrasing and prompt clarity
* Requires stable internet connection
* Real-time adaptation still limited without continuous learning
* Assessment generator sometimes repeats questions

**13. Future Enhancements**

* Real-time learning updates using Reinforcement Learning
* Multilingual learning modules (regional languages)
* Teacher Dashboard with class-level insights
* Native mobile app for Android/iOS
* Integration with LMS platforms like Moodle or Google Classroom
* Voice support & Video-based explanations

**14. Conclusion**

**Edu Tutor AI** aims to revolutionize learning by leveraging the power of AI to deliver custom learning experiences for each student. It empowers learners to study at their own pace, receive instant help, and track their progress—all through an intuitive, interactive interface. This project lays a strong foundation for scalable, inclusive, and intelligent educational solutions.