**Edututor AI: personalized with generative AI and LMS iteration**

**Project documentation**

1.Introduction

* Project Title: Edututor AI: Personalized with Generative AI and LMS Iteration
* TEAM LEADER: Ruthika.A
* TEAM MEMBER: Aishwaryalakshmi.V

2.Project Overview

Purpose:

The purpose of Edututor AI is to revolutionize personalized learning by integrating Generative AI with a Learning Management System (LMS). The system adapts to individual student needs, providing AI-generated learning materials, personalized feedback, automated assessments, and interactive tutoring support. By leveraging generative models, the platform enhances engagement and ensures a tailored learning experience for each student.

Features:

* AI-Powered Tutoring

Key Point: Personalized guidance

Functionality: Provides instant answers, explanations, and learning support based on student queries.

* Content Generation

Key Point: Automated material creation

Functionality: Generates quizzes, notes, and practice exercises based on syllabus and difficulty level.

* Performance Analytics

Key Point: Real-time progress tracking

Functionality: Monitors learner’s performance and suggests improvement areas.

* Adaptive Learning Path

Key Point: Customized course structure

Functionality: Creates personalized study plans depending on student strengths and weaknesses.

* Interactive Assessments

Key Point: AI-driven evaluation

Functionality: Provides instant grading, explanations, and adaptive question difficulty.

* Seamless LMS Integration

Key Point: Scalable deployment

Functionality: Works with existing LMS platforms (Moodle, Google Classroom, etc.) for smooth adoption.

3.Architecture

Frontend (React/Stream lit):

Provides an interactive and user-friendly dashboard for students and educators.

Backend (FastAPI/Django):

Manages requests, AI model integration, and LMS connectivity.

Generative AI Models:

Large Language Models are used for tutoring, content generation, and personalized recommendations.

Database:

Stores student progress, performance reports, and generated content securely.

Integration Layer:

Connects with LMS platforms via APIs for seamless functionality.

4.Setup Instructions

Prerequisites:

* Prerequisites:
* Python 3.9 or later
* API keys for AI model integration
* LMS credentials for integration
* Internet access

Installation Process:

1. Clone the repository
2. Install dependencies from requirements.txt
3. Configure .env file with credentials
4. Run backend server (FastAPI/Django)
5. Launch frontend (React/Stream lit)
6. Connect with LMS and start interaction

5.Folder Structure

* App/ – Backend logic including routes, models, and AI integration
* Ui/ – Frontend components for dashboards and learning pages
* Ai\_engine/ – Handles generative AI content and tutoring responses
* Database/ – Stores user data and learning records
* Report\_generator.py – Creates AI-driven performance reports

6.Running the Application

1. Start backend server (FastAPI/Django).
2. Launch frontend dashboard.
3. Sign in as student/teacher.
4. Upload course syllabus or connect with LMS.
5. Access personalized tutoring, auto-generated content, and performance reports.

7.API Documentation

1. POST /chat/ask – Returns AI-generated tutoring response
2. POST /upload-course – Uploads syllabus or course materials
3. GET /get-content – Generates quizzes, notes, and exercises
4. GET /analytics – Provides performance metrics and improvement suggestions

8.Authentication

* Role-based access (student, teacher, admin)
* API key integration for AI models
* LMS authentication support

9.User Interface

* Dashboard with learning progress visualization
* AI tutor chat window
* Auto-generated quizzes and notes section
* Report download option

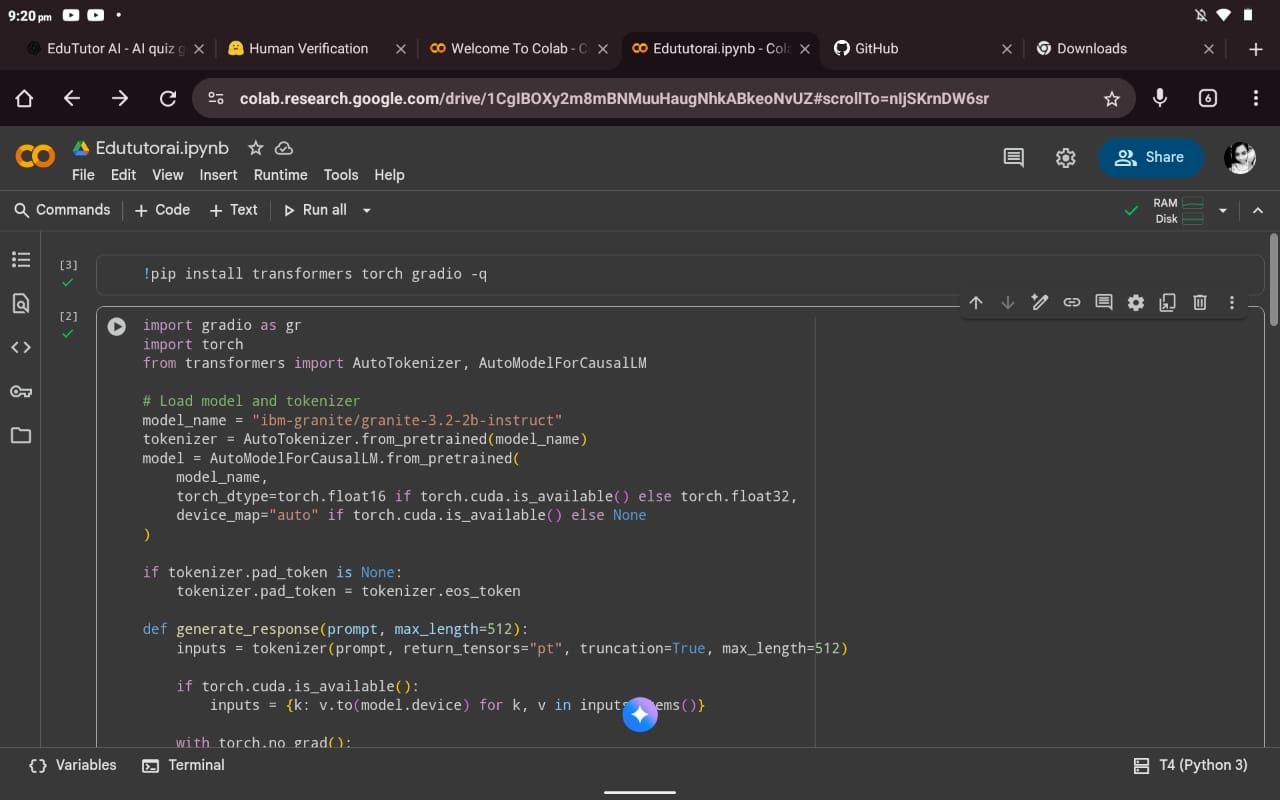
10.Testing

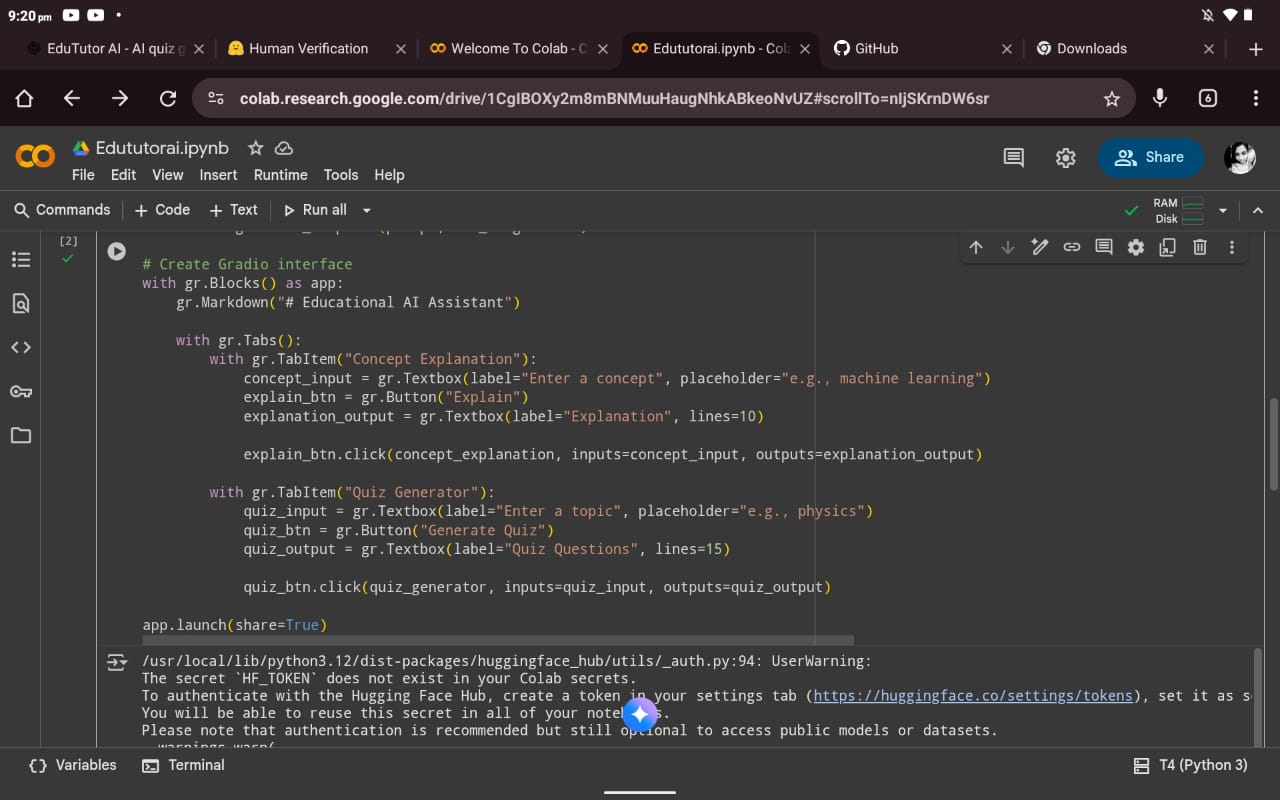
Unit Testing: AI responses and quiz generation

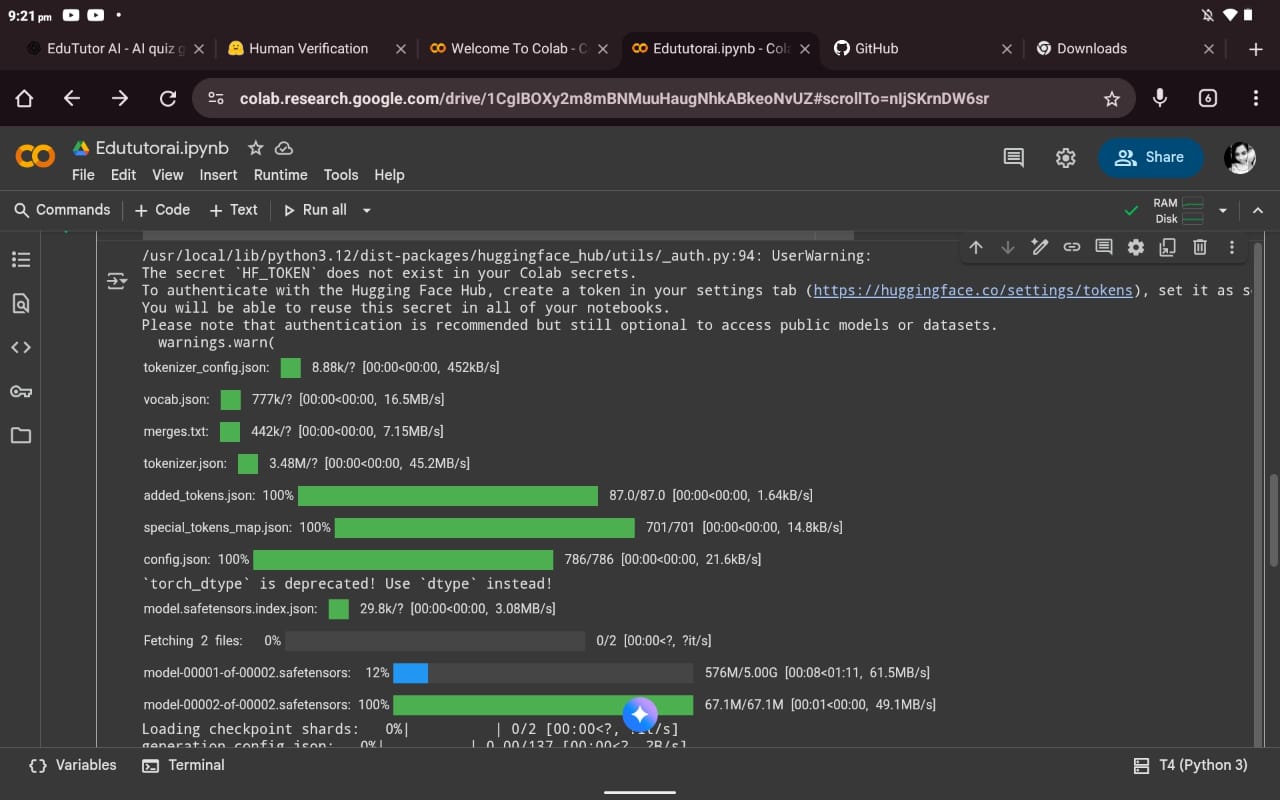
API Testing: Postman/Swagger

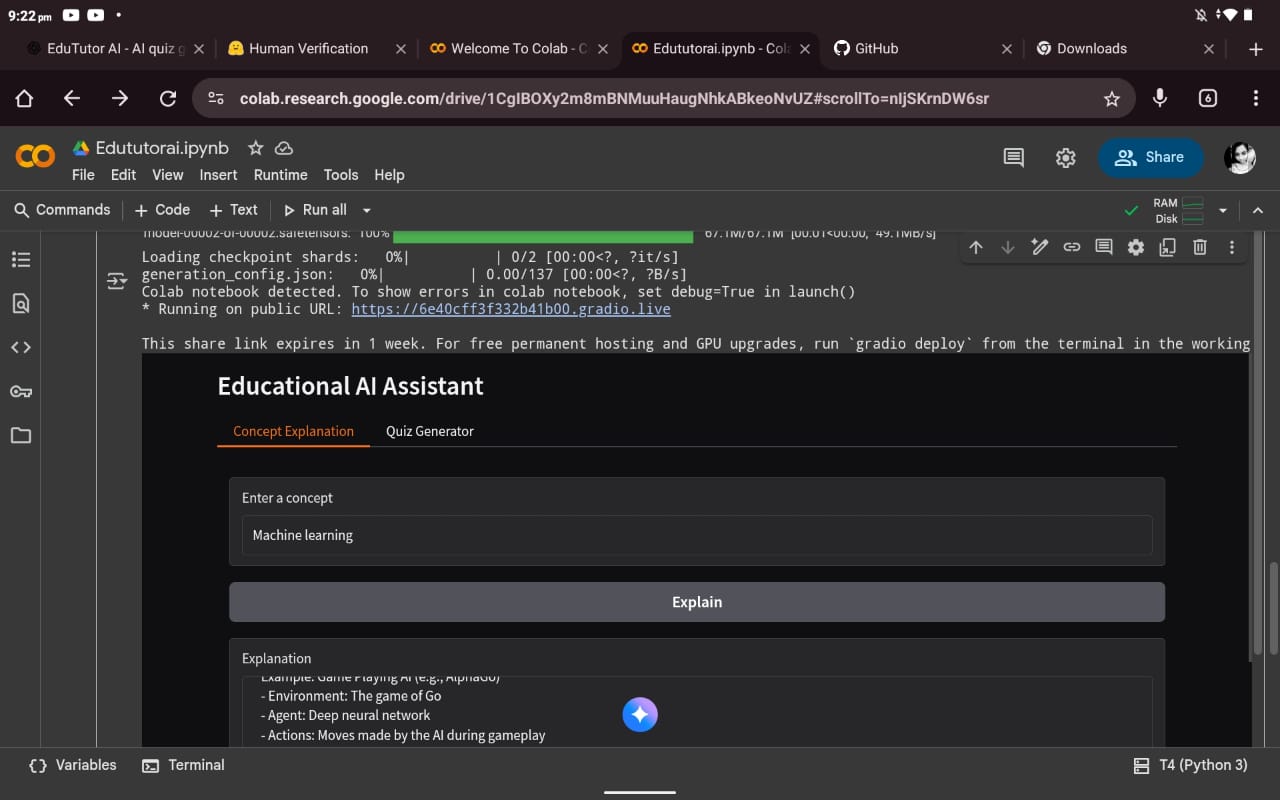
Manual Testing: LMS integration and dashboard usability

11. Screenshot









12.Known Issues

* Limited offline access
* Dependent on API availability

13.Future Enhancements

* Voice-based tutoring assistant
* Multilingual support
* AI-powered peer group recommendations
* Offline mode with cached lessons