EduTutor-AI: AI-Powered Quiz Learning Platform

Project Documentation

Date: 28 June 2025

Team ID: LTVIP2025TMID31982

Team Size: 4

Team Leader: Kimidi Rakesh Vasanth

Team Member: Kalidindi Salma

Team Member: Kadingu Uday Kumar

Team Member: Jupudi Kiran

1. Introduction

EduTutor-AI is an AI-powered educational platform designed to assist students in practicing topic-based quizzes and help educators monitor learning outcomes. Built using Python and Streamlit, and integrated with IBM Watsonx and Pinecone, it delivers an end-to-end intelligent quiz generation, submission, evaluation, and feedback workflow.

2. Project Overview

EduTutor-AI aims to provide automated, personalized quiz generation based on subjects and topics using generative AI. It supports role-based access, allowing students to take quizzes and educators to analyze performance.

Key Features:

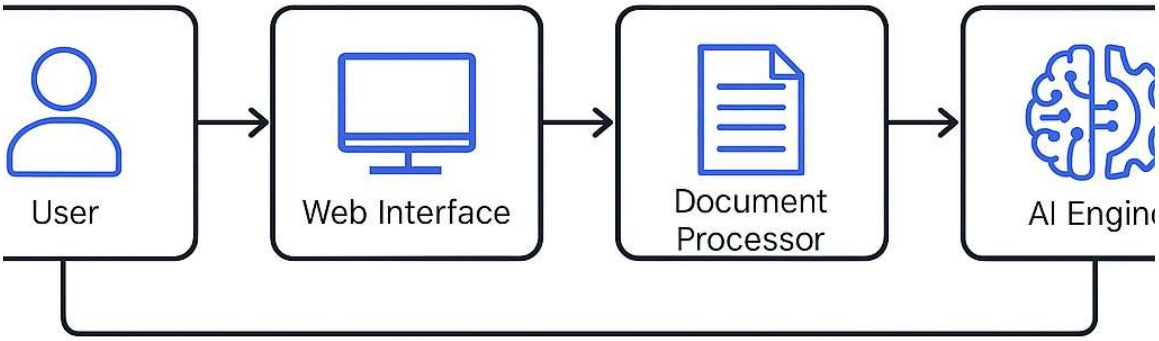
- Role-based login for students and educators
- AI-generated quizzes using IBM Watsonx
- Quiz result storage and feedback via Pinecone
- Educator dashboard to view student progress
- Real-time performance feedback and score explanation

3. Architecture

EduTutor-AI uses a modular architecture:
1. User Interface (Streamlit): Provides quiz interface and dashboard views.
2. Backend Logic (FastAPI): Handles quiz processing and API interactions.
3. AI Service Layer (IBM Watsonx): Generates topic-based MCQs.
4. Vector DB (Pinecone): Stores quiz results and user history.
5. Session Management: Maintains user state using Streamlit session or JWT.

EduTutor-AI Data Flow

Input flow from user to AI engine, an·. bac.



EduTutor-AI Data Flow

4. Setup Instructions

Prerequisites:
- Python 3.9+
- IBM Watsonx access and API key
- Pinecone account and key

Steps:
1. Clone the project and enter directory
2. Create virtual environment: `python -m venv venv`
3. Activate environment: `source venv/bin/activate` (Linux/Mac) or `venv\Scripts\activate` (Windows)
4. Install requirements: `pip install -r requirements.txt`
5. Create `.env` file and add:
   - WATSONX_APIKEY=your_key
   - WATSONX_PROJECT_ID=your_project_id
   - PINECONE_API_KEY=your_key
6. Run app: `streamlit run app.py`

5. AI Prompt Usage

EduTutor-AI uses IBM Watsonx `generate_text()` for MCQ generation.

Example prompts:
- "Create 5 MCQs on Pythagorean Theorem"
- "Generate quiz on Human Digestive System"

AI Parameters:
- max_new_tokens = 300
- temperature = 0.7
- top_p = 1.0

6. Authentication & Security

Authentication supports Google OAuth and Email/Password login.
Security Practices:
- Environment variables managed via `.env` + `python-dotenv`
- API keys are not hardcoded
- `.env` is git-ignored for safety

7. User Interface

The Streamlit interface includes:
- Login page (student or educator)
- Topic selection input for quiz
- Quiz display with options and timer

- Submission feedback (score, correct answers)
- Educator dashboard with student performance tables

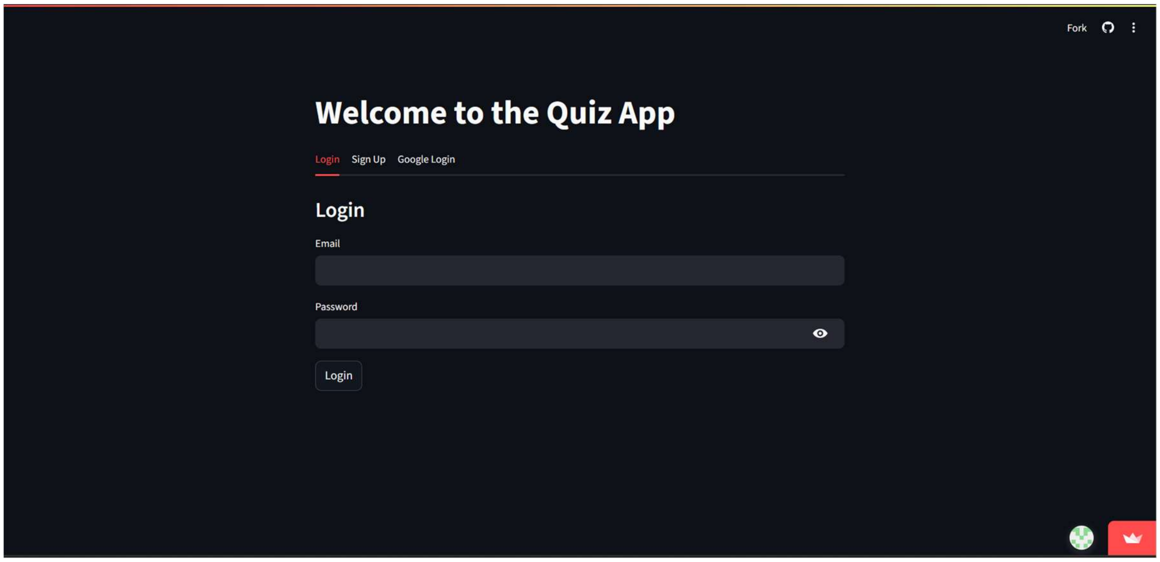
8. Testing

Testing Methods:

- Unit testing of logic and prompts (with pytest)
- Manual validation of quiz generation
- Student quiz submission flow
- Educator result retrieval and analytics view

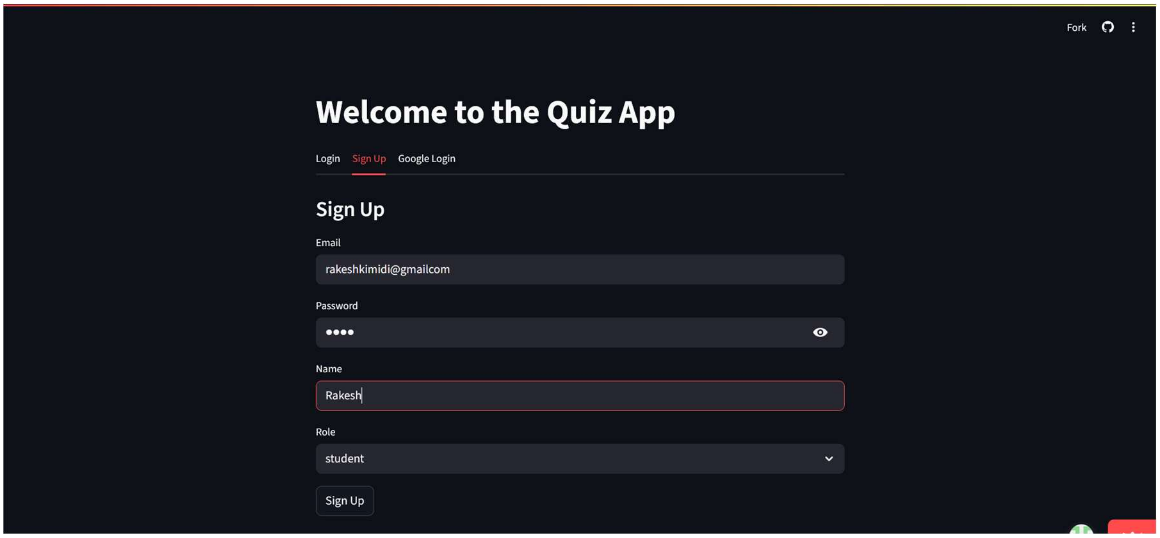
9. Screenshots

Add screenshots here of the quiz page, result summary, and educator dashboard.



Welcome to the Quiz App

Login



Welcome to the Quiz App

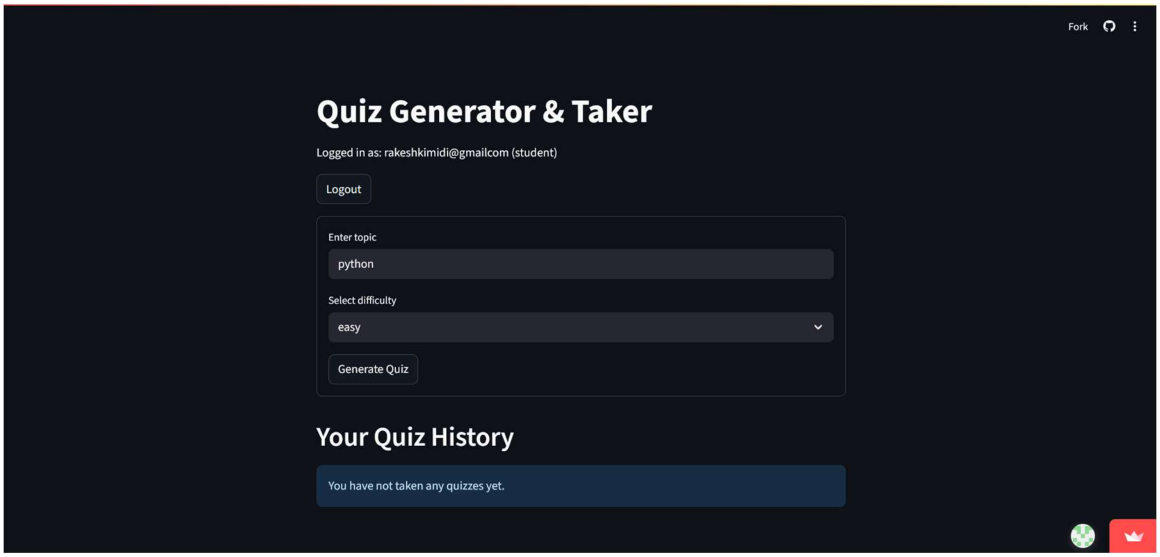
Sign Up

Email: rakeshkimidi@gmailcom

Password: ●●●●

Name: Rakesh

Role: student



Quiz Generator & Taker

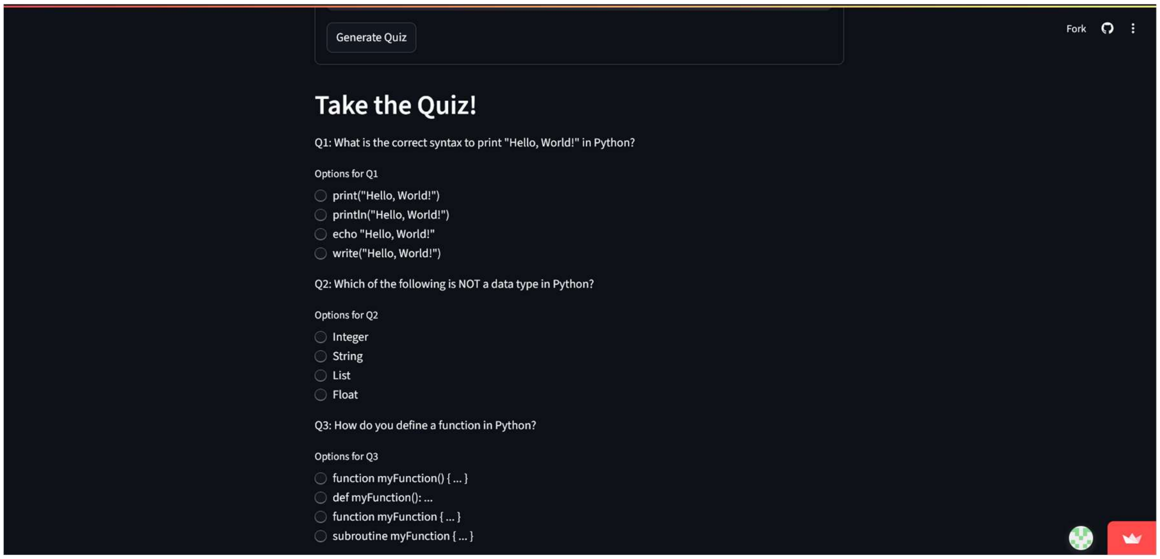
Logged in as: rakeshkimidi@gmailcom (student)

Enter topic: python

Select difficulty: easy

Your Quiz History

You have not taken any quizzes yet.



Take the Quiz!

Q1: What is the correct syntax to print "Hello, World!" in Python?

Options for Q1

○ print("Hello, World!")
○ println("Hello, World!")
○ echo "Hello, World!"
○ write("Hello, World!")

Q2: Which of the following is NOT a data type in Python?

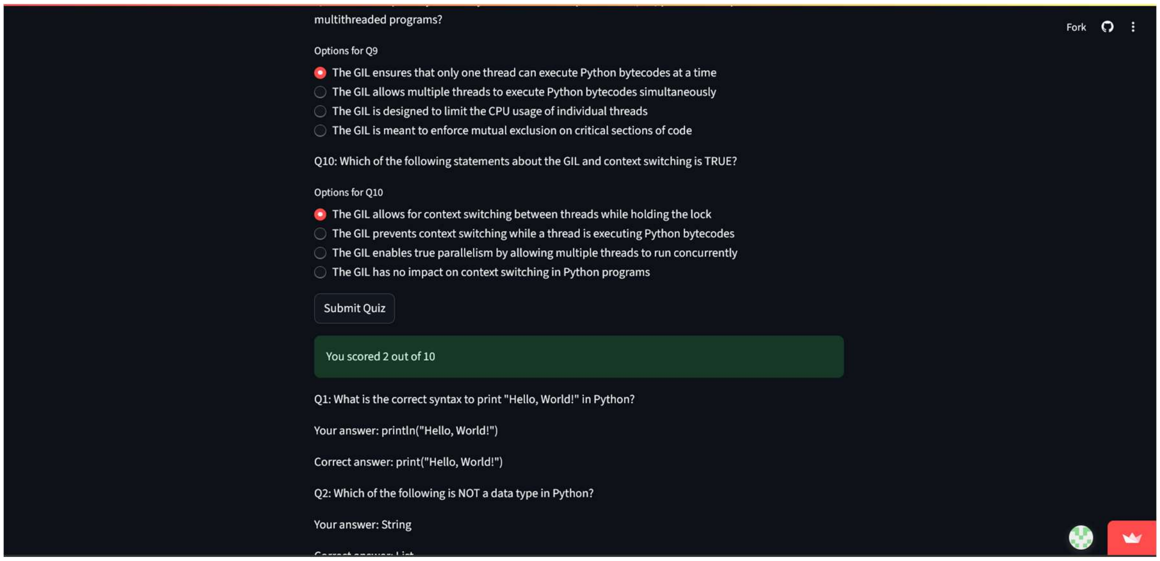
Options for Q2

○ Integer
○ String
○ List
○ Float

Q3: How do you define a function in Python?

Options for Q3

○ function myFunction() { ... }
○ def myFunction(): ...
○ function myFunction { ... }
○ subroutine myFunction { ... }



multithreaded programs?

Options for Q9

● The GIL ensures that only one thread can execute Python bytecodes at a time
○ The GIL allows multiple threads to execute Python bytecodes simultaneously
○ The GIL is designed to limit the CPU usage of individual threads
○ The GIL is meant to enforce mutual exclusion on critical sections of code

Q10: Which of the following statements about the GIL and context switching is TRUE?

Options for Q10

● The GIL allows for context switching between threads while holding the lock
○ The GIL prevents context switching while a thread is executing Python bytecodes
○ The GIL enables true parallelism by allowing multiple threads to run concurrently
○ The GIL has no impact on context switching in Python programs

Submit Quiz

You scored 2 out of 10

Q1: What is the correct syntax to print "Hello, World!" in Python?

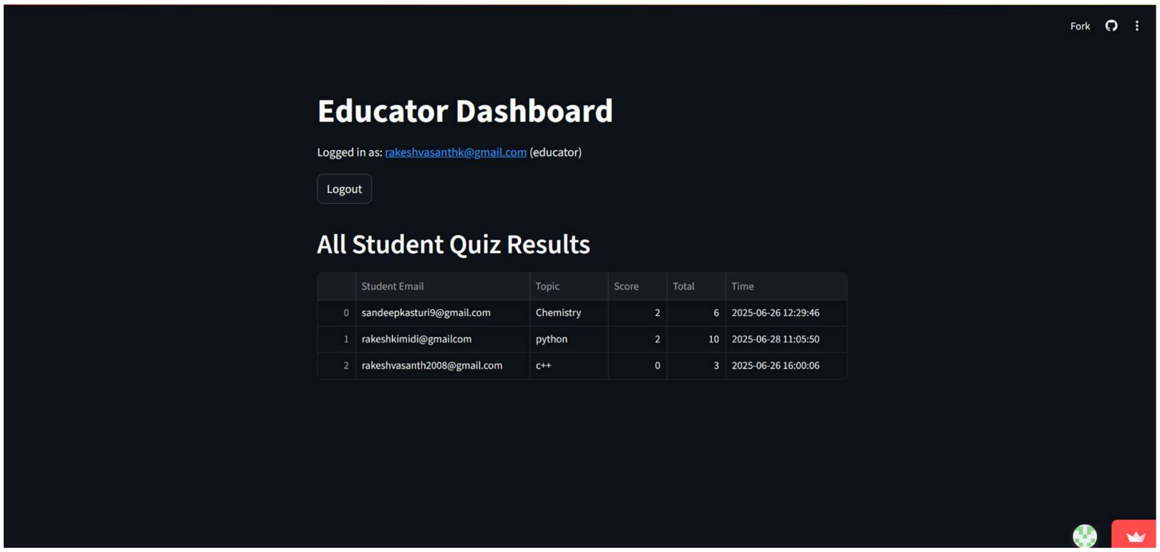
Your answer: println("Hello, World!")

Correct answer: print("Hello, World!")

Q2: Which of the following is NOT a data type in Python?

Your answer: String

Correct answer: List



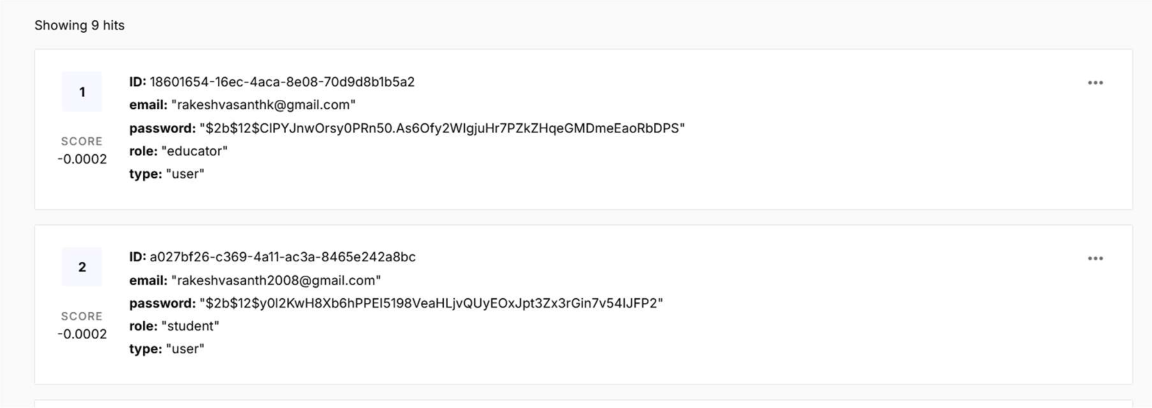
Educator Dashboard

Logged in as: rakeshvasanthk@gmail.com (educator)

Logout

All Student Quiz Results

	Student Email	Topic	Score	Total	Time
0	sandeepkasturi9@gmail.com	Chemistry	2	6	2025-06-26 12:29:46
1	rakeshkimidi@gmailcom	python	2	10	2025-06-28 11:05:50
2	rakeshvasanth2008@gmail.com	c++	0	3	2025-06-26 16:00:06



Showing 9 hits

1

SCORE
-0.0002

ID: 18601654-16ec-4aca-8e08-70d9d8b1b5a2
email: "rakeshvasanthk@gmail.com"
password: "$2b$12$ClPYJnwOrsy0PRn50.As6Ofy2WIgjuHr7PZkZHqeGMDmeEaoRbDPS"
role: "educator"
type: "user"

2

SCORE
-0.0002

ID: a027bf26-c369-4a11-ac3a-8465e242a8bc
email: "rakeshvasanth2008@gmail.com"
password: "$2b$12$y0l2KwH8Xb6hPPEI5198VeaHLjvQUyEOxJpt3Zx3rGin7v54IJFP2"
role: "student"
type: "user"

10. Known Issues

- No persistent user sessions (no DB for credentials)
- No error messages on invalid inputs
- Limited support for adaptive quizzes
- Only English language supported currently

11. Future Enhancements

- Add database (e.g., MongoDB or PostgreSQL)
- Dockerize for deployment
- Role-based route guards and analytics graphs
- Integrate Google Classroom or LMS
- Adaptive quiz difficulty and performance graph

12. Folder Structure

```
EduTutor-AI/
├── app.py              → Main Streamlit UI app
├── config.py            → Environment config loader
├── auth/              → Google and email login logic
├── services/           → IBM Watsonx and Pinecone integrations
├── templates/           → HTML templates
├── static/             → CSS and assets
├── .env               → Secret keys
├── requirements.txt        → Dependencies
```

13. Modules Breakdown

- auth/google_auth.py → Google OAuth handler
- services/watsonx_service.py → Calls IBM Watsonx for quiz generation
- services/pinecone_service.py → Manages student result storage
- quiz_routes.py → Quiz logic for submission and evaluation
- educator_dashboard.py → Displays analytics

14. Technology Stack

Frontend: Streamlit
Backend: Python + FastAPI
AI Service: IBM Watsonx Granite 3.3

Database: Pinecone (Vector DB)
Authentication: OAuth + JWT + .env
Deployment: IBM Cloud / Localhost

15. Conclusion

EduTutor-AI is a powerful learning tool that combines generative AI and real-time analytics to support both students and educators. It demonstrates how AI can personalize education, automate assessment, and deliver meaningful feedback — paving the way for future-ready ed-tech platforms.