PROJECT REPORT

TITLE: EduTutor-AI – Personalized AI Tutoring Platform

---

1. INTRODUCTION

1.1 PROJECT OVERVIEW

EduTutor-AI is an AI-powered tutoring platform designed to assist students with personalized academic help. The system allows users to upload questions (text or image), receive contextual answers, generate topic summaries, and engage in interactive Q&A with a virtual tutor powered by large language models.

1.2 PURPOSE

The project aims to enhance the learning experience for students by offering real-time, intelligent academic support across subjects. It provides structured explanations, concept breakdowns, and adaptive content delivery to meet individual learning needs.

---

2. IDEATION PHASE

2.1 PROBLEM STATEMENT

Students often struggle with understanding concepts, completing homework, and getting personalized help. Traditional tools are either too generic or lack interactivity.

2.2 EMPATHY MAP CANVAS

SAYS: “How can I understand this better?” “Is there a tutor who can explain this now?”
THINKS: “Can I rely on this explanation?”
DOES: Searches YouTube/Google for help, joins coaching groups.
FEELS: Overwhelmed by syllabus, anxious before exams.
PAINS: Incomplete answers, inconsistent tutoring, lack of visual explanations.
GAINS: Quick, reliable guidance; interactive learning support; 24/7 help.

2.3 BRAINSTORMING

Standalone ideas like OCR text extraction, doubt-solving bot, and concept summarizer were merged into EduTutor-AI to cover the end-to-end tutoring lifecycle.

---

3. REQUIREMENT ANALYSIS

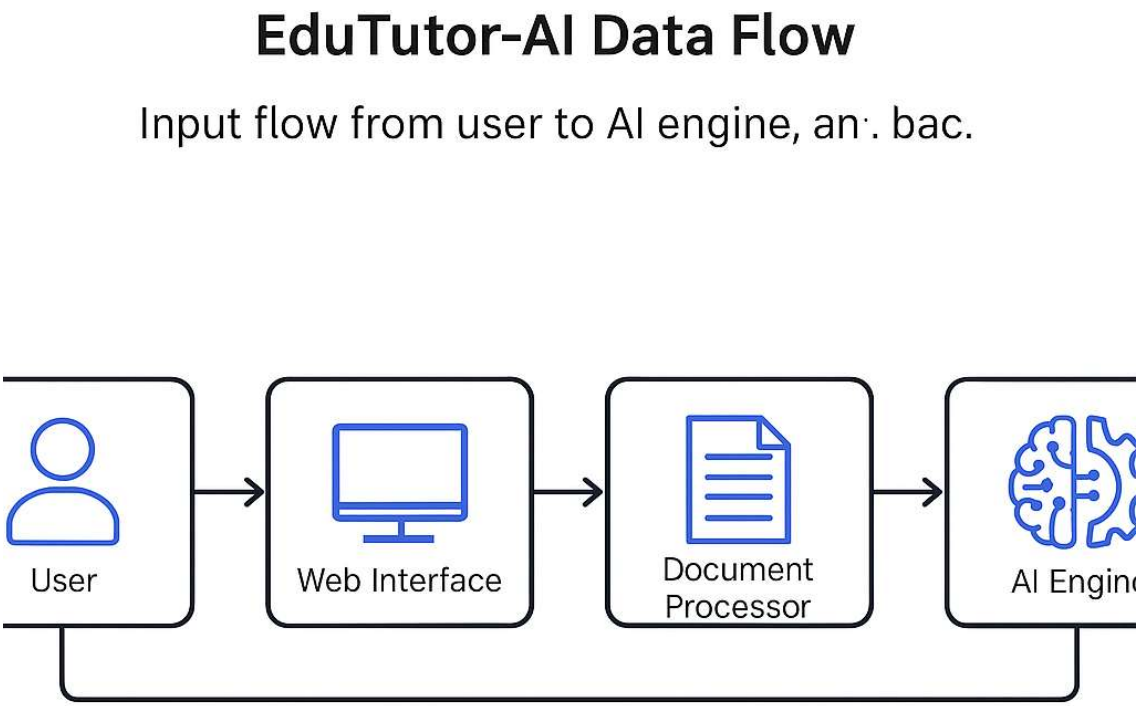
3.1 CUSTOMER JOURNEY MAP

1. User opens Streamlit app.
2. Uploads a question (image/text) or selects a module.
3. EduTutor-AI extracts content and generates responses.
4. The student reviews, saves, or asks follow-up questions.
5. The student switches modules or ends session.

3.2 SESSION REQUIREMENTS

- Upload image, text, or PDF input.
- Provide real-time AI response.
- Enable download of generated content.
- Preserve conversation/session memory.

3.3 DATA FLOW DIAGRAM



EduTutor-AI Data Flow

3.4 TECHNOLOGY STACK

- **Frontend**: Streamlit
- **Backend**: Python 3.11
- **AI Model**: OpenAI/GPT-like model or HuggingFace Transformers
- **OCR**: PyMuPDF, pytesseract
- **Session Management**: Streamlit session state
- **Environment Management**: virtualenv, dotenv

---

4. PROJECT DESIGN

4.1 PROBLEM–SOLUTION FIT

The need for personalized, immediate academic support is met using an AI engine capable of answering and explaining student questions dynamically.

4.2 PROPOSED SOLUTION

Layer 1: Streamlit UI for uploading and prompting
Layer 2: Python backend to parse inputs and generate prompts
Layer 3: AI module to generate answers/summaries
Layer 4: Output returned via UI with formatting

4.3 SOLUTION ARCHITECTURE

- **UI Layer**: Sidebar navigation, input widgets, result display
- **Core Logic**: Parsing text/images, managing prompts
- **AI Layer**: Answer and summary generation
- **Session Storage**: Inputs maintained during navigation

---

5. PROJECT PLANNING AND SCHEDULING

Week 1: UI Design, basic module layout, OCR integration
Week 2: AI integration, answer generation logic
Week 3: Summarization, chat logic, performance tuning
Week 4: Final UI cleanup, documentation, deployment

---

6. FUNCTIONAL AND PERFORMANCE TESTING

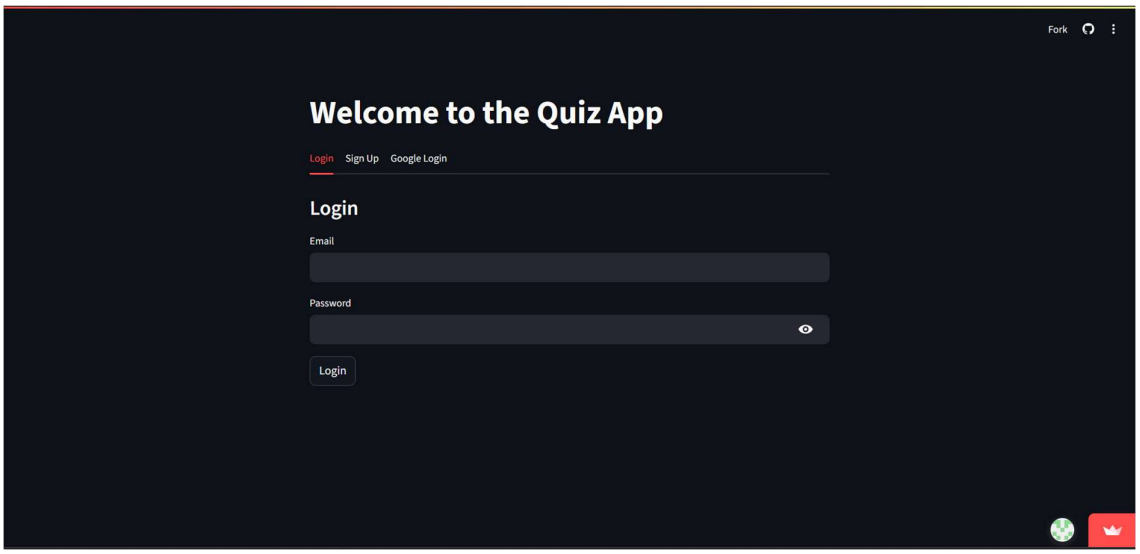
- **Unit Testing**: OCR utilities, answer parsing
- **Integration Testing**: Upload → AI → Response flow
- **Manual Testing**: Multiple question formats and subjects

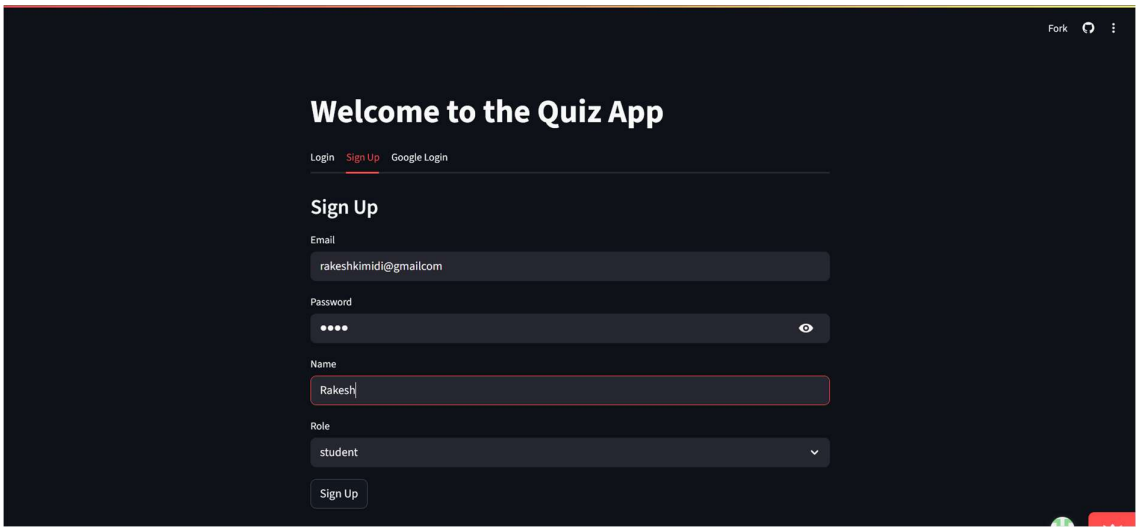
- **Resilience**: Handling invalid input and concurrent sessions

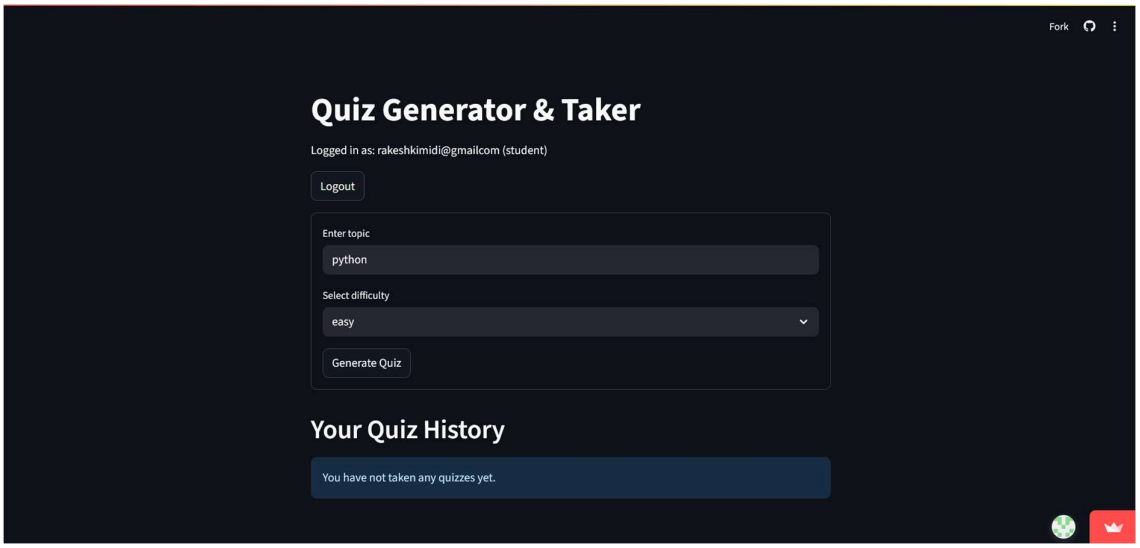
---

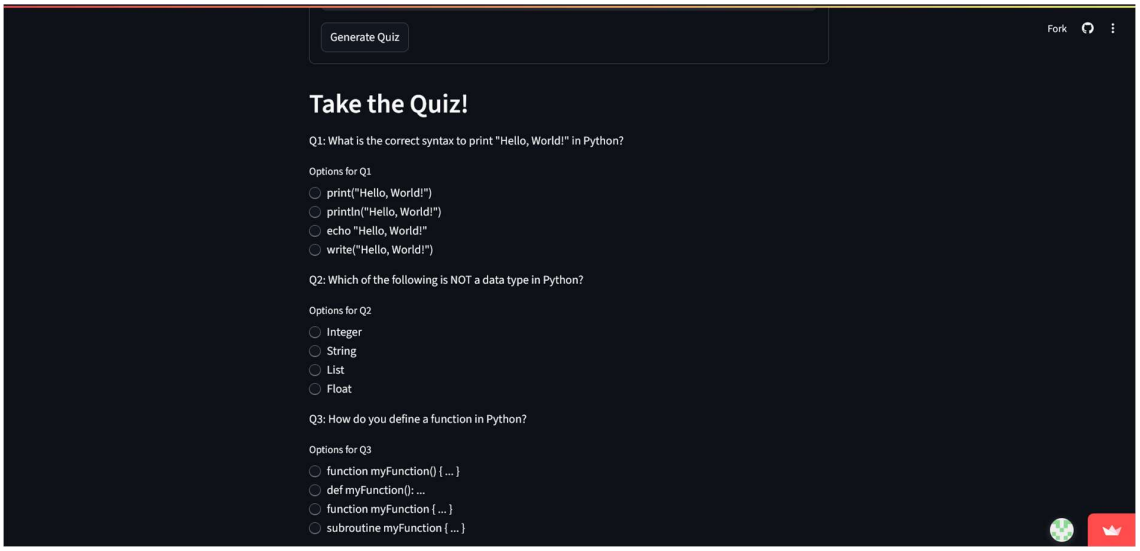
7. RESULTS

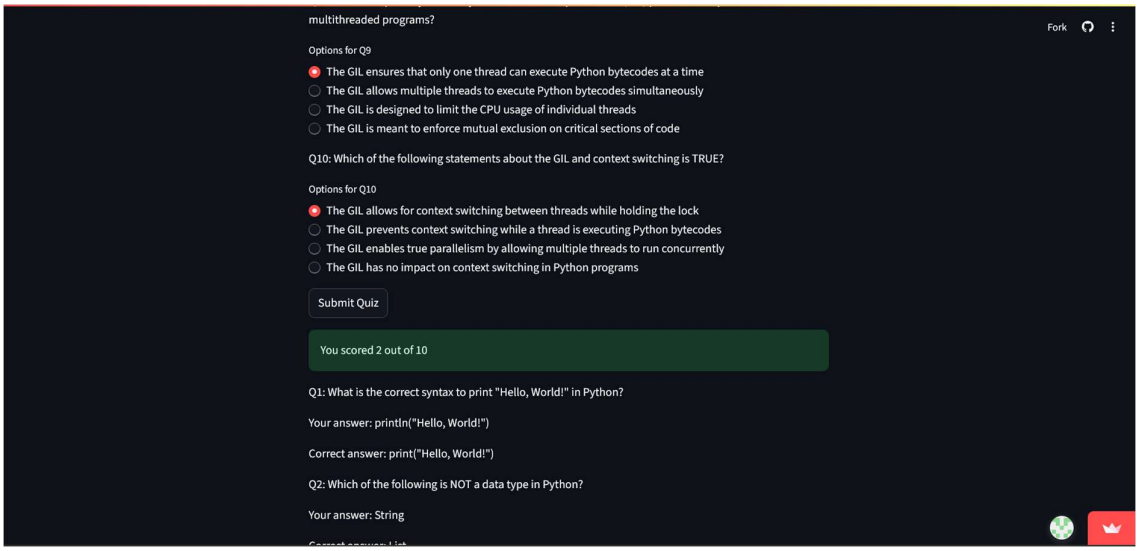
- Worked well for academic questions from math, science, social studies
- OCR processed handwritten and printed text effectively
- Students received answers with explanations in under 3 seconds

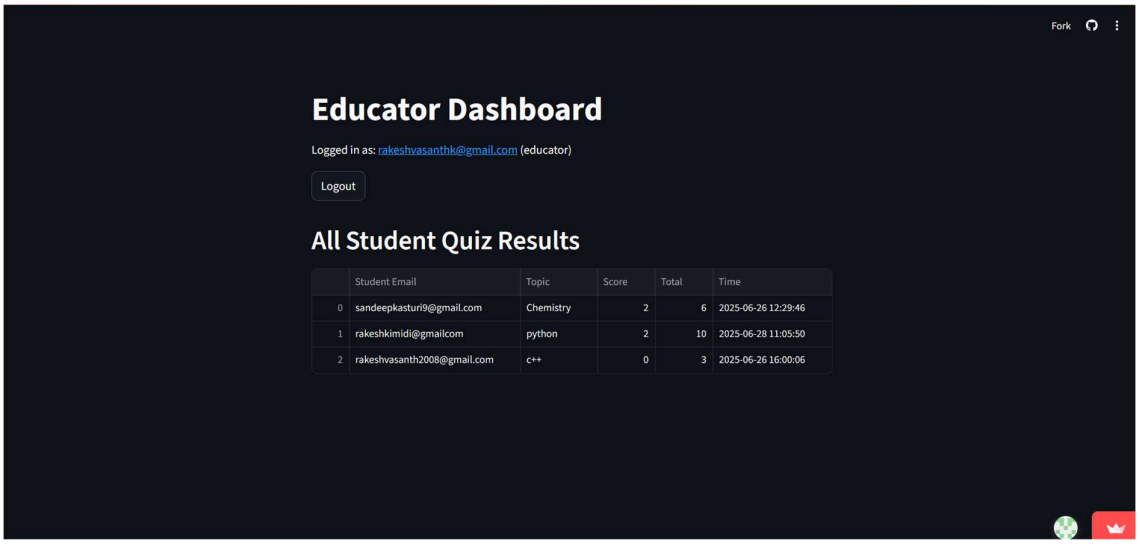





Fork
Quiz Generator & Taker
Logged in as: rakeshkimidi@gmailcom (student)
Logout
Enter topic
python
Select difficulty
easy
Generate Quiz
Your Quiz History
You have not taken any quizzes yet.


Fork
Generate Quiz
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 { ... }





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"

---

8. ADVANTAGES AND DISADVANTAGES

Advantages:

- Round-the-clock tutoring access
- Personalized, fast learning support
- Works with images and PDFs

Disadvantages:

- No user login or profile tracking
- Subject limitation to academic domains
- Requires stable internet for LLM API

---

9. CONCLUSION

EduTutor-AI bridges the gap between doubt-solving and smart tutoring using a simple interface and AI engine. It can be extended to multilingual subjects, integrated with LMS, and adapted for teacher assistance as well.

---

10. APPENDIX

- GitHub Repository: [Insert GitHub Link]
- Key Files: EduTutor.py, pages/, utils/, .env.example
- Model: GPT-style transformer model
- License: MIT or Apache 2.0